Naresh Kumar Sehgal
Manoj Saxena
Dhaval N. Shah

AI on the Edge with Security

Foundations and Practices

Springer

AI on the Edge with Security

Naresh Kumar Sehgal • Manoj Saxena
Dhaval N. Shah

AI on the Edge with Security

Foundations and Practices

Springer

Naresh Kumar Sehgal
Deeply Human, Inc.
Santa Clara, CA, USA

Dhaval N. Shah
Fremont, CA, USA

Manoj Saxena
Center of Excellence for Distributed AI
NetEdge Computing Solutions Pvt. Ltd
New Delhi, Delhi, India

ISBN 978-3-031-78271-8 ISBN 978-3-031-78272-5 (eBook)
https://doi.org/10.1007/978-3-031-78272-5

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Naresh dedicates this book to his grandfather Niranjan Das Sehgal, and his three children: Hetesh, Gauri, and Garima, who taught him the value of patience and selfless love.
Manoj dedicates this book to Prof. Pramod Chandra P. Bhatt who introduced him to the world of Artificial Intelligence (AI) and kept him at the Edge with continuous guidance.
Dhaval dedicates this book to his Grandfather Bhogilal P. Shah, Parents, Niranjan B. Shah and Geeta N. Shah and Brother Kaushal N. Shah. Their encouragement and support played a crucial role in this quest.

Foreword

We, Information Technologists, are adept at coining picturesque names for new technological innovations. In the early part of this century, we saw the emergence of huge data centers established by vendors that were accessible to customers via the Internet from any place at any time. They provided on-demand and payment services such as computing cycles, data storage, computing platforms, and software systems. They were multi-tenanted, elastic, and virtualized. Instead of calling them a "Computing Utility on the Internet," the term Cloud Computing was coined befuddling non-professionals who wondered how Clouds could compute! Cloud Computing was inevitably followed by Fog Computing, Mist Computing, and Dew Computing—more picturesque names for cloud extensions. These computing environments are visualized as layers: Cloud➔Fog➔Mist➔ and Dew. In between Dew and Mist, a more ordinarily named environment—Edge Computing—has emerged. It is a computing layer at the edge of the Internet that can access the cloud while processing locally critical data of systems they govern. The most important properties of Edge Computing are low latency and high security, both properties vital to process data acquired from a plethora of IoT devices. The data acquired from these devices are securely processed primarily by Edge computers.

Novel ideas from the emerging Artificial Intelligence paradigms are used to solve problems arising from the growing applications of computers in health care systems, manufacturing systems employing armies of robots, logistics in moving goods, and plenty of novel systems. Security and Intelligence are the two vital requirements in emerging computer-assisted systems and Edge Computing provides them.

This book by Naresh Kumar Sehgal, Manoj Saxena, and Dhaval N. Shah is a timely addition to the emerging area of Edge Computing—a direction in which technology is moving. The authors have several years of experience designing and delivering IT systems to clients worldwide, which is reflected in this book intended for students and professionals who wish to understand this new technology. I am sure you will gain immensely after reading this timely book. Happy reading!

Supercomputer Education and Research Centre
Indian Institute of Science
Bangalore, India

V. Rajaraman

Preface

The idea for this book took birth in Fremont, California, in a meeting of Naresh Kumar Sehgal and Dhaval N. Shah with Prof. PCP Bhatt, who was visiting California from India in the Summer of 2023. Prof. Bhatt is the author of one of the most widely used *Operating Systems* book, which is in its 5th edition in India. He and Naresh had co-authored three books on Cloud Computing and noted a trend to move computing out of data centers closer to the customers. Serendipitously, Dhaval, in his AI Consulting work was sensing a need for a book in this area too. Thus a new book on Edge Computing was jointly conceived. Subsequently, Prof. Bhatt introduced Naresh and Dhaval to Dr. Manoj Saxena, who had completed his Ph.D. with Prof. Bhatt. Manoj is operating a successful business related to Distributed and Edge Computing in India for over three decades.

A traditional data center consists of racks of servers, storage, and network devices in a large, air-conditioned facility. Then scores of clients and IoT (Internet of Things) devices interact with this data center over public and private networks. However, this setup causes latency which is unacceptable for real-time computing applications that can involve instant decision-making using AI. There is a need to support AI applications at or near the edge of computing networks to minimize latency and to address data privacy concerns. Such concerns currently cause many customers such as hospitals to not put their data in the Clouds. In this book, we review the solution architectures and algorithms to support AI in the edge computing devices.

Computing growth often mimics biology, which has recently resulted in an explosion of AI and ML applications to the real-life problems. Next step in this evolution is Intelligent IoT devices. Which can perform some AI and ML at the edge of networks. The net result will be alleviation of a need to transfer lots of data to the faraway datacenters and mimic intelligent entities making autonomous decisions while interacting with each other. Such a revolution will turn Cloud computing on its head to de-centralize the control back to the edge devices.

There are many books on IoT, Cloud Computing, and AI/ML, but these are treated as three different topics currently. We aim to connect the dots and show readers the power of combining these in a single setup. Result will be lower electrical power, latency, and higher security. Next generation of Cloud computing is needed to minimize latency and address privacy/security concerns of many customers. This book highlights the problems and proposes new solutions for performing AI and ML at the Edge of computing networks. We introduce some new concepts such as Collaborative Federated Learning in the context of data privacy and Edge Computing.

As a textbook, our target audience are students in Computer Science, Information Technology, and professionals practicing AI or Edge Computing. Readers will learn about new topics to prepare themselves for the next steps in evolution of Cloud Computing in context of Security, AI, and Edge.

The book assumes some background and understanding of basic hardware, operating systems, and some aspects of software engineering. To that extent it would suit senior under graduates or graduates in their early semesters. As a technical manuscript, the book has enough in-depth coverage to interest IT managers and architects who need to develop solutions for Edge Computing. Our book provides a strong foundation of Cloud, Security, IoT, AI/ML, and Networking before building a case for “Secure AI on the Edge of the Cloud.” This enables our book to serve as a textbook at the upper UG or graduate level of classes.

We recognize that a book has a limited shelf life so would like to plan ahead for an update. Thus, we invite the readers to send their inputs, comments, and any feedback to AIEdgeBookAuthors@gmail.com. These will be incorporated in the next edition. Many thanks!!

Santa Clara, CA, USA — Naresh Kumar Sehgal
New Delhi, India — Manoj Saxena
Fremont, CA, USA — Dhaval N. Shah

Acknowledgments

Naresh wants to acknowledge leadership and help from Prof. PCP Bhatt for authoring the first chapter to get us started. Prof. Bhatt has a vast knowledge of computer hardware and software systems, which he is willing to share without expecting anything in return. This book also leverages several security ideas from Prof. John M. Acken at Portland State University, Oregon. Naresh also acknowledges support and guidance from Prof. Pramod Varshney at Syracuse University, NY.

Manoj wants to acknowledge the support and encouragement received from his family, especially his wife Deepika Saxena, for completing this book.

Dhaval wants to acknowledge Naresh Sehgal for his spirit of steadfastness in working on this project towards completion, and also the initial guidance and support provided by Prof. PCP Bhatt. Dhaval also acknowledges review inputs from Cole Kepford and Vinay Venkatesh.

We appreciate the excellent support from Charles Glaser and Brian Halm, along with the staff of Springer Publications, to bring this book in your hands. Various trademarks used in this book are respectively owned by their companies.

Definitions

Acceptability
Acceptability indicates the willingness of users to perform a particular action. It is used for evaluating factors acceptable for authentication.

Access Control
Access control is a mechanism that determines whether to allow or deny access to the requested data, equipment, or facility.

Attack Surface
Attack surface for a system refers to the set of access points, which can be used to compromise security of that system.

Attestation
Attestation is a process of certification. It enhances the trust level of remote users that they are communicating with a specific trusted device in the Cloud.

Authentication Factor
An authentication factor is data or a measurement used to validate an identity claim.

Auto-scaling
Auto-scaling is used by AWS to automatically add new servers to support an existing service, when more customers are using it and service may be slowing down.

Botnet
A collection of systems successfully infected by an attacker to launch Web-based robot (aka Wobot) attack such as Denial of Service (DOS).

Breach
A successful attack on a secured system.

Cache
A cache is a high-speed memory included in a CPU (central processing unit) package, for storing data items that are frequently accessed, to speed up a program. There may be multiple levels of cache memories.

Ciphertext
Ciphertext is an encrypted message.

Client-Server
Client-server is a computing model where a single server provides services to many client devices.

Cloud Bursting
Cloud bursting is a process to request and access resources beyond an enterprise’s boundary, typically reaching into a Public Cloud from a Private Cloud, when user load increases.

Cluster
Cluster refers to a group of interconnected servers, within a rack or combining a few racks, to perform a certain function such as supporting a large database or to support a large workload.

Cyber-Physical System
A cyber-physicalsystem is a physical system with an embedded computer control. It contains various physical devices being controlled by system software and measurement devices.

Decryption
Decryption is the algorithmic modification of a ciphertext using a key to recover the plaintext content of the message.

Denial of Service Attack
A denial of service (DoS) attack is flooding a server with repeated service requests, thereby denying its services to other requestors.

Distributed Denial-of-Service Attack
A distributed denial-of-service (DDoS) attack uses multiple clients on the Internet to repeatedly send excessive numbers of requests to a server. It prevents the recipient from providing services to other requestors. Frequently, this causes a Website to completely crash.

Edge Computing
Edge computing refers to the notion of having compute ability at the edge of a network with local storage and decision-making abilities. Edge denotes the end point of a network.

Elasticity
Elasticity is a property of computing and storage facilities in a Cloud to expand in view of a growing need and shrink when the need goes away.

Electronic Signature
An electronic signature is a secure hash of a message with its author’s identification. This electronic hash is often used as a replacement of a physical signature on a paper document.

Encryption
Encryption is the algorithmic modification of a message using a key to convert plaintext to ciphertext. This makes it difficult for an attacker to read the content of the message except with a key.

Failure in Time
Failure in time (FIT) is the failure rate measured as the number of failures per unit time. It is usually defined as a failure rate of 1 per billion hours. A component having a failure rate of 1 FIT is equivalent to having an MTBF of 1 billion hours. FIT is the inverse of MTBF.

Fault Tolerance
Fault tolerance is the property of a computer system to keep functioning properly in the presence of a hardware or software fault.

Grid Computing
Combining compute servers, storage, and network resources to make them available on a dynamic basis for specific applications. Grid computing is a form of utility computing. Also, see Utility Computing.

Hashing
Hashing is the calculation of a short fixed size number based upon the content of a message. Hashing can be used for error checking, authentication, and integrity checking.

Hybrid Cloud
Hybrid cloud is a Cloud-computing environment, which uses a mix of on-premises, Private Cloud, and third-party, Public Cloud services with orchestration between the two platforms.

Identity Authentication
Identity authentication is determination that a claimant is who it claims to be. It uses a process or algorithm for access control to evaluate whether to grant or deny access.

Imposter
An imposter is a person or program that presents itself as someone or something else in order to gain access, circumvent authentication, trick a user into revealing secrets, or utilize someone else's Cloud service. See also Masquerader.

Interoperability
Interoperability is the ability of different information technology systems and software applications to communicate, exchange data, and use the information that has been exchanged.

Latency
Latency refers to time delay in transmission of network packets or in accessing data from a memory.

Machine Learning
Machine learning refers to the ability of computers to learn without being explicitly programmed.

Malware
Malware is a program (such as virus, worm, Trojan horse, or other code-based malicious entity infecting a host) that is covertly inserted into a system with the intent of compromising the confidentiality, integrity, or availability of the victim's data, applications, or operating system.

Mean Time Between Failures
Mean time between failures is the downtime or basically mean time to failure plus the mean time to repair.

Mean Time to Failure
Mean time to failure is the average time between failures.

Mean Time to Repair
Mean time to repair is the average time from when a failure occurs until the failure is repaired.

Noisy Neighbors
Noisy neighbor is a phrase used to describe a Cloud Computing infrastructure co-tenant that monopolizes bandwidth, disk I/O, CPU, and other resources and can negatively affect other users' Cloud performance.

Observability
Observability is a measure of how well internal states of a system can be inferred from knowledge of its external outputs.

Optimizations
In computing, optimization is the process of modifying a system to make some features of it work more efficiently or use fewer resources.

Private Cloud
Private Cloud is dedicated to a single organization.

Public Cloud
A Public Cloud offers its services to a full range of customers. The computing environment is shared with multiple tenants, on a free or pay-per-usage model.

Quality of Service
The overall performance for a Cloud service, as documented in a service-level agreement between a user and a service provider. The performance properties may include uptime, throughput (bandwidth), transit delay (latency), error rates, priority, and security.

Reliability
Reliability is a quality measure to reflect consistency of operation such as failure frequency and consequent performance and/or security.

Self-Service
Self-service Cloud Computing is a facility where customers can provision servers, storage, and launch applications without going through an IT person.

Side-Channel Attack
An attempt to create a security breach by indirectly attacking the secured information, such as guessing a secret by measuring power supply current.

Streaming
Streaming is a technique for transferring data so that it can be processed as a steady and continuous stream. Streaming technologies are becoming important because most users do not have fast enough access to download large datasets. With streaming, the client browser or plug-in can start displaying the data before the entire file has been transmitted.

Threat Model
Threat modeling is an approach for analyzing the security of an application. It is a structured approach to identify, quantify, and address the security risks associated with an application.

Trusted Certificate
A certificate that is trusted by the relying party (usually a third party) on the basis of secure and authenticated delivery. The public keys included in trusted certificates are used to start certification paths. See also Trust Anchor and Trusted Arbitrator.

Trusted Computing
Trusted computing refers to a situation where users trust the manufacturer of hardware or software in a remote computer and are willing to put their sensitive data in a secure container hosted on that computer.

Usability
Usability is a metric of user experience. It represents ease of use or how easily a product can be used to achieve specific goals with effectiveness, efficiency, and satisfaction for typical usage.

Utility Computing
Delivering compute resources to users, who pay for these as a metered service on a need basis.

Verification
Verification is the process of establishing the truth, accuracy, or validity of something.

Virtual Machine Monitor
A virtual machine monitor (VMM) enables users to simultaneously run different operating systems, each in a different VM, on a server.

Virtual Machines
A virtual machine (VM) is an emulation of a computer system.

Virtual Private Cloud
A Virtual Private Cloud (VPC) is an on-demand configurable pool of shared computing resources allocated within a Public Cloud environment, providing a certain level of isolation between the different organizations.

Virus
A computer program that can copy itself and infect a computer without permission or knowledge of the user. A virus might corrupt or delete data on a computer, use resident programs to copy itself to other computers, or even erase everything on a hard disk.

Vulnerability Vulnerability refers to the inability of a system to withstand the effects of a hostile environment. A window of vulnerability (WoV) is a time frame within which defensive measures are diminished and security of the system is compromised.

Web Service Web service is a standardized way of integrating and providing Web-based applications using the XML, SOAP, WSDL, and UDDI open standards over an Internet Protocol backbone.

Workload In enterprise and Cloud Computing, workload is the amount of work that the computer system has been given to do at a given time. Different types of tasks may stress different parts of a system, e.g., CPU-bound or memory-bound workloads.

Contents

Part I Foundations

Part II Practices

Abbreviations

AIaaS	AI as a Service
AI	Artificial Intelligence
BEMS	Building Energy Management Systems
BIM	Building Information Modelling
CapEx	Capital Expenditure
CCPA	California Consumer Privacy Act
CFL	Collaborative Federated Learning
CSP	Cloud Service Provider
DBMS	Database Management System
DDOS	Distributed Denial of Service
DOS	Denial of Service
DR	Disaster Recovery
DTM	Distributed Trust Model
EMS	Energy Management Systems
EU	European Union
FCC	Federal Communications Commission
FDA	Food and Drug Administration
FIT	Failure in Time
FTC	Federal Trade Commission
GDP	Gross Domestic Product
GDPR	General Data Protection Regulation
HIPPA	Health Insurance Portability and Accountability Act
HPC	High-Performance Computing
HSM	Hardware Security Module
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
IaaS	Infrastructure as a Service
IIoT	Intelligent IoT
INFOSEC	Information Security
IoT	Internet of Things

IP	Intellectual Property
IP	Internet Protocol
LAN	Local Area Network
MFA	Multifactor Authentication (also, see 2FA)
ML	Machine Learning
MTBF	Mean Time Between Failure
MTTF	Mean Time to Failure
MTTR	Mean Time to Repair
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OEM	Original Equipment Manufacturer
OpEx	Operational Expenditure
OTP	One True Password
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PIN	Personal Identification Number
PK	Public Key
PoC	Proof of Concept
QoS	Quality of Service
REST	Representational State Transfer
RFID	Radio-Frequency Identification
RoT	Root of Trust
RSA	Rivest–Shamir–Adleman
SaaS	Software as a Service
SEC	Security
SGX	Software Guard Extensions
SLA	Service-Level Agreement
TCB	Trusted Compute Boundary
TCP/IP	Transmission Control Protocol/Internet Protocol
TEE	Trusted Execution Environment
TEFCA	Trusted Exchange Framework and Common Agreement
VLSI	Very Large-Scale Integration
VM	Virtual Machine
VMM	Virtual Machine Monitor
VPC	Virtual Private Cloud
VPN	Virtual Private Network
WAN	Wide Area Network
WWW	World Wide Web

Part I
Foundations

Chapter 1
Edge Computing with AI: Introduction

1.1 Introduction

Most human endeavors require intelligence. Our civilization has been built upon the intelligent use of available technologies and techniques. As civilizations evolve, so do their needs and requirements. Systems around us have grown in scale and complexity. This growth has been sustained largely because more powerful and human-friendly tools have been made available. Over the years, computer technology has supported this [1] growth in two major ways. One was by offering faster and higher instruction processing capabilities, which grew exponentially as per Moore's law [2] (as hardware). The other was by offering higher levels of user-friendly and intelligent software environments [3]. Newer software offers enhanced layers of abstraction to support different degrees of automation, bringing it tantalizingly close to human intelligence in its operation. The decade of the 1970s and early 1980s witnessed many advances by bringing in computer and communications technology to work in mutually enriching mode. The result was the Internet, announced in 1983[1] [4]. This ought to be considered a breakthrough. Among other things, the Internet offered an operational platform in the form of the client-server architecture [5] to drive further growth. This computing framework led to the emergence of seamless, scalable distributed computing. The computing scene has changed ever since. The distributed computing and client-server architecture with large server farms brought a significant change in the way computing was perceived and done. That has ushered in the era of Cloud computing [6].

The Cloud computing paradigm seemingly offers unlimited capability. Consequently, the new millennium saw another tectonic shift in the software system designs and applications as artificial intelligence (AI) came out of its

[1] The evolution of the Internet was gradual, spreading over time. However, officially January 1, 1983, is the date assigned as the date of birth of the Internet.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_1

second winter [7]. Within the precincts of Cloud computing, notwithstanding its myriad and multiple facets, what stands out is the embedding of AI at the Edge of the Cloud to direct system operations. This synergy brings Edge computing right into the center stage for most, if not all, IoT (Internet of Things) devices as well.

This book is primarily an effort to explore the foundations of the support technologies, techniques, and architecting methodologies that beget synergy by embedding AI at the Edge of the Cloud.

This chapter, in fact, sets the tone and context for the rest of the book. As might be obvious, this book deals with three key areas, e.g., Cloud computing, Edge computing, and use of AI for timely decision-making. The chapter begins with an explanation of the terms and definitions used in the Cloud computing context. This is followed by a detailed explanation of why Edge computing is required and what kind of architectural framework is used. Later sections take the rationale further and offer first impressions of systems with embedded AI. The final section briefly describes the organization of the book.

1.2 Cloud Computing

This section covers the primary and ground level definitions in Cloud computing. These concepts are further elaborated with more technical details in later chapters. Readers who are familiar and know the basic terms used here may skip this section.

1.2.1 Cloud Computing Terms

It is well known and understood that computers operate with a heartbeat provided by a powerful engine called CPU—short for central processing unit. The memory units and the peripherals do not match CPU speeds. In fact, way back in 1959, that was the raison d'être for “the time-sharing systems” [8]. This development was subsequently followed by invoking interactivity to additionally give each user an impression of exclusive control for operations on a main frame—essentially a centrally located large system.

The Cloud computing works derive their inspiration from the early time-sharing interactive systems, albeit with a different paradigm. A typical Cloud service architecture that supports its users via the Internet is shown in Fig. 1.1.

The primary rationale in favor of Cloud computing is why own and maintain a system if it can be tenanted—essentially avail rental service pattern or pay-as-you-use [6]. The reasons for considering this to be a paradigm shift are as follows:

(a) The mainframe is replaced by a large pool of interconnected powerful servers to cater to computing and storage needs, however large and varied.
(b) The services are offered using high-speed Internet infrastructure.

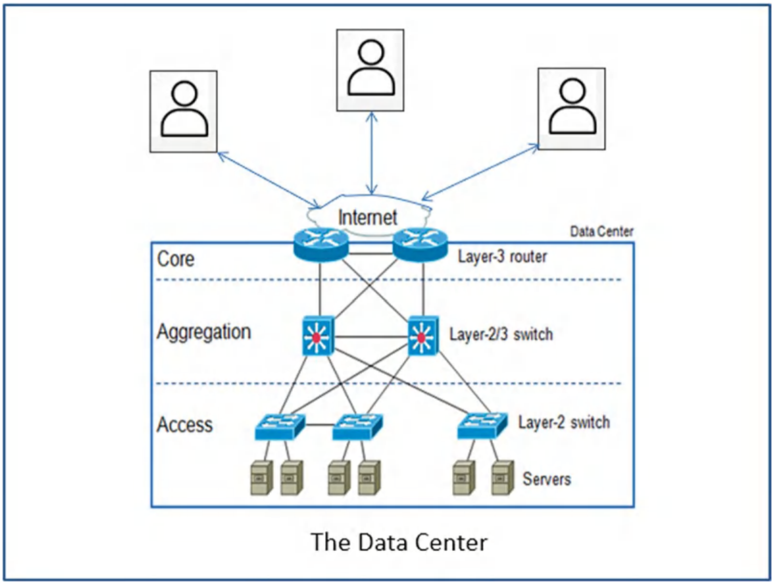


Fig. 1.1 The Cloud data center and user connectivity

(c) The Cloud is expected to accommodate varying user needs, i.e., computing services can be customized to meet the user needs.
(d) The computing services can be availed by choosing from a variety of rental plans. For example, it may be consummated by subscription or pay-as-you-use, or by time of the day, duration, and required bandwidth, besides the extent of usage, etc.
(e) The service quality is underwritten usually by offering a Service-Level Agreement (SLA).
(f) The services also cover the extent and depth of security provisions.

Clearly, challenges lie in meeting varying workload demands and assurances of security in a multi-tenanted facility [6]. However, our limited objective here is to focus on the forms of services and how the Cloud computing paradigm succeeds in meeting a spectrum of service scenarios. Broadly, the Cloud computing services are provisioned using the following three modes of operations: IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service). A more detailed discussion on Cloud operations and services would be described in a later chapter. For now, it would be worthwhile to briefly define these terms to understand the modes of operation. Besides, this would also set the context for the Edge on Cloud.

Figure 1.2 shows the pyramid depicting the Cloud operations [6] at the three service levels, which are described below.

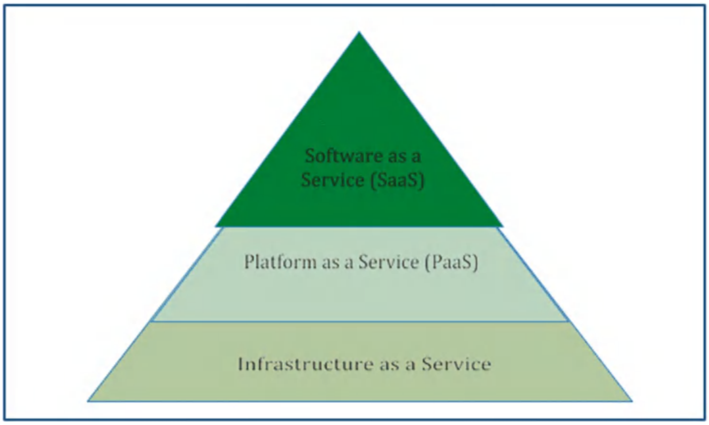

Software as a Service (SaaS)
Platform as a Service (PaaS)
Infrastructure as a Service

Fig. 1.2 Cloud computing pyramid depicting IaaS, PaaS, and SaaS operations [6]

IaaS Basically, the IaaS Cloud service providers provision infrastructure components. The infrastructure components include configurable servers, storage devices, and network support. The client is free to create customized and virtual architectural configurations of choice using hypervisors.

PaaS Quite in contrast to the IaaS, the PaaS allows developers to build a complete and customizable operating environment. They can do builds for applications, execute and manage them using the Web and database servers and operating systems of choice, availing tools, network services, and middleware.

SaaS As an abstraction, SaaS is at a higher level than PaaS and IaaS as it proffers a software distribution model. SaaS allows independent application developers (also known as independent service vendors, or ISVs) to offer and support software services on contractual terms using the Cloud. The Cloud service provider (CSP) hosts the applications, and through them, the ISVs make these services available to the end users.

In addition to the broad definitions given above, the three modes of operation are compared in Table 1.1. The comparison reveals the context of use, and the Cloud environment caters to various user or organizational needs at different levels. Following this is a description of the way organizations and end users obtain access to the Cloud.

One additionally needs to know and understand the way organizations and end users obtain access to the Cloud. There are three ways this happens.

Private, Public, and Hybrid Cloud The Cloud service provisioning may be private, public, or hybrid. The connotations of these terms indicate where and how the data and associated systems are shared. For example, the private Cloud is owned and operated by an organization as a captive facility for internal users with very

Table 1.1 A comparison of the three forms of Cloud operations and services

	IaaS	PaaS	SaaS
User category	System designers	Product developers	Application users
Nature of service available	Configuration control to create platform and system-based services	Development tools, to create and deploy portable apps in a virtual environment	Web services-based solutions, including specified executable middleware
Flexibility in use	One may even create a virtual machine IP address	Limited control for customizing the operating system (OS) environment	No control but flexibility for creating applications
Asset generated	A configured infrastructure or system architecture	A virtualized development platform	An application to provide specific class of solutions
Scalability	Highly scalable	Highly scalable	Almost unlimited
Security	Provided by the Cloud service provider (CSP)	CSP, OS in use or embedded in the solution stack	CSP infrastructure and application embedded
Some CSP examples	AWS, EC2, Rackspace	Azure, Heroku	Salesforce, DropBox, Google Apps

limited or no external access. The public Cloud is open to all end users primarily as a multi-tenancy mode with an agreed upon SLA often under best effort considerations. The hybrid mode is used when an organization may choose to operate in a mixed mode. They secure their intellectual properties and critical systems and data private while sharing space for non-critical assets.

Like every human endeavor, the Cloud services too over time have been evaluated, and efforts are always afoot to improve the range of service offerings and performance. The next subsection is devoted to explaining how services get rendered and what time measurements are critical in the context of Cloud computing.

1.2.2 Latency and Response Times

Let us consider an operational scenario as depicted in Fig. 1.3 where a user seeks resolution to a query. In general, query resolution requires access to and processing at the multiple backend servers. The application finally assembles the responses and sends it across to the user. The latency experienced at the client end is the result of cumulative delays and response times within the system. Note that professionals differentiate between the response times and latency. Latency refers to delays while the response time primarily refers to the processing time within the system. From our standpoint the Cloud latency shall be defined as the cumulative time required for the desired service.

The latency and response time studies are carried out under varying load conditions. This may entail using different strategies. For instance, from a certain

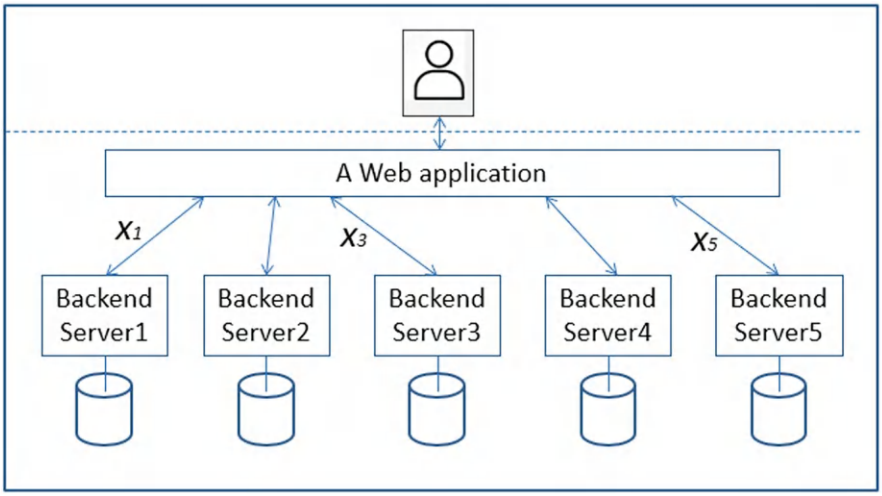


Fig. 1.3 System response time—the latency

workstation the same query is raised at randomly chosen times and latencies experienced are recorded. One may also generate workloads with multiple users seeking resolution to the same query. There are several strategies that can be adapted to simulate different workloads [9].

Clearly, the response times at client end would vary under different operating conditions. A brief analysis follows shows how the response times get impacted and where the delays occur in the system.

(a) The servers queue up the requests received. Figure 1.3 shows that the servers may respond with different time lags. The three relevant servers (1, 3, and 5) respond in time duration x_1, $x_{3,}$ and x_5. Though the three servers are operating in parallel, the slowest one would be counted for recording the delay, i.e., if $x_3 > x_{1,}$ *and* $x_3 > x_{5,}$ then the delay is recorded as x_3.
(b) The server disk storage may have read conflicts and would contribute to the delays.
(c) There may be a server node failure. In that case workload balancing is done. This too may result in delays as each server node may need to cater to greater workload.
(d) The Internet infrastructure adds to the delay depending upon the network traffic.
(e) Finally, the load at the Web application end would contribute its share of delay.

One would like to know how this reflects on CSPs and clients. It would be worthwhile to reflect on the following observations cited by Martin Kleppmann [9] (Refs. 20–22 in his book):

1. Amazon recorded that a 100-millisecond delay diminishes sales by 1%.
2. Other extended studies quoted show that a 1 s delay results in a 16% decline in customer satisfaction.

The client end determines the way the Cloud services are performing. So, a measurement at that end makes sense. As alluded to earlier, one may send the query at random times and note the latencies observed. The average computed from latencies observed would be a good measure. However, the commercial world prefers a percentile of customers for whom services were rendered [9]. That would mean we look at the median or define latencies by indicating that 99.95 percentile. Regardless of the measure used, the latency is an issue of concern with the Cloud service provisioning being ubiquitous. The nation states use it for sharing public information and governance; e-commerce businesses are firmly placed and use it to provide goods and services of all kinds, from travel to daily grocery; the media uses it to create and distribute content; the political parties perform analytics and use it to plan strategies, etc. The list goes on and on. The application scenarios get further enhanced by the emergence of IoT on the horizon catering to intelligent applications like smart buildings, home appliances, and climate monitors, etc. As a result, presently, the latency considerations and intelligent operations at the Edge are truly the two major compelling drivers for Cloud-based operations.

1.3 Processing at the Edge

In continuum with the discussions so far, here is a list of a few representative systems where awareness of environment is required and latency matters.

1. Patient monitoring systems in the health care domain
   (a) With connected contact lenses (for eyes), inhalers (for asthma), ingestible sensors for glucose, heart rate, depression, and mood monitors
   (b) Robotic surgery: real-time monitoring during robotic surgery
2. In fully automated manufacturing plants with mobile robots or building construction sites with material and crane movement
3. Telemetering network gateways and instrument monitors that are often used for disaster management
4. Practically all IoT devices used for transportation systems including railways, truck convoys and traffic diversion advisory required on high density traffic on roads
5. Monitoring of CCTVs and communication with security cameras

All these systems are expected to operate as if they are self-aware. This is akin to how humans react to their environment. Therefore, the analogy offered in the next paragraph is very apt to understand why and how Edge computing with embedded intelligence helps in enhancing the efficacy in conjunction with Cloud-based operations.

In his book, Matthew Syed [10] makes a point about many winners ranging from Mozart, Beckham, Roger Federer, and others from a variety of vocations. The churn that precedes their success is the point of analogy here. Take the athletes, for instance. Besides storing strategies and winning patterns (as data) in their mind, they hone and chisel their skills with repeated practice[2] (see the footnote) to build and enhance their muscle memories. This is equally true of musicians who train their vocal chords or instrumentalists who have practice sessions on their instruments. Now if one imagines the brain as the main data center and our muscles with their kinetic links as active agents at the Edge of a human system, then the analogy for Edge computing reveals itself. The Edge, once trained to respond to its environment, can mitigate the latency issues inherent to Cloud operations and services.

Let us now embark on a journey to explore ways to achieve lower latency responses. To begin with, an understanding of Edge computing should be helpful. In certain ways, the processing at the Edge offers a compromise between a totally centralized system and a system with distributed capabilities. It would be preposterous to suggest a fixed framework to achieve the desired goals for all operational scenarios.

However, there is a broader conceptual understanding about the nature of Cloud connectivity with the Edge as shown in Fig. 1.4. The Edge devices facilitate interaction of the external agents with a data center. Invariably, the IoT or Edge devices are placed where the data is generated. These devices store data for local processing and action. They feed large volumes of data to be uploaded to the Cloud for longer-term analysis. The Cloud servers are at the bottom, and the switches connect Edge devices to the Cloud data center via the Internet routers.

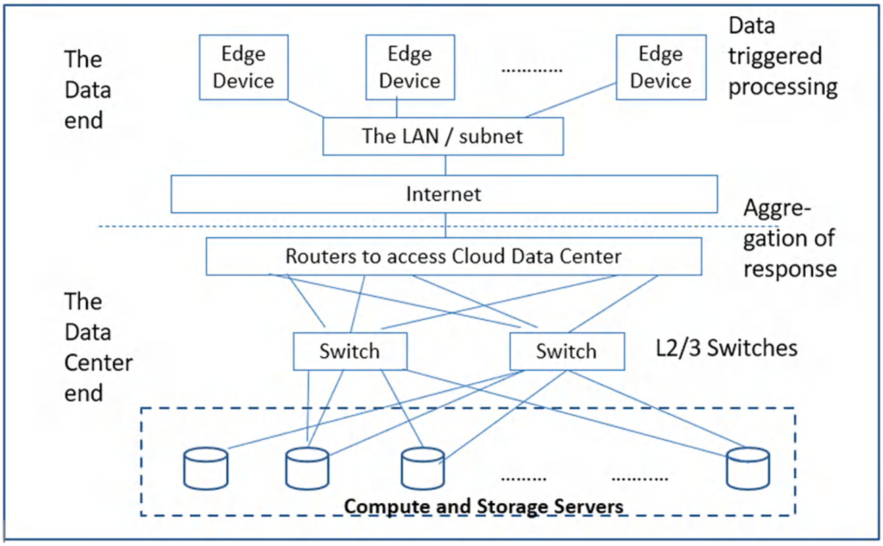


Fig. 1.4 A conceptual architecture for data centers with connectivity at the Edge

[2] Each practice session is like a supervised training session in AI or tweaking of weights in ANN.

At this point, it would be worthwhile to explore some diverse scenarios to show that the framework shown in Fig. 1.4 is generic enough, i.e., it can be adapted to suit specific operational needs.

1.3.1 The Edge Architecture: Scenario 1

The example scenario captured has the backdrop of pre-Internet days when one of us was witness to a certain event. Back then (circa 1972), the communication and control were largely managed using telemetry. One of us was with the University of Manchester Institute of Science and Technology (UMIST) with a workspace on the fourth floor of a multistory building. One late afternoon the firefighters came rushing and connected their hoses to the water outlet to extinguish the fire in room 409—just two rooms away and diagonally placed. The sequence of events and the consequent response are described next:

- A smoker faculty member had thrown a cigarette butt in trash can and left for lunch.
- The butt was not fully extinguished and sat atop some crumpled sheets of paper—perhaps a rejected technical manuscript.
- The smoke arose from the trash can and was sensed by the smoke detector.
- The relay on the fourth floor corridor identified that the problem was in room no. 409.
- The foyer relay at the ground floor identified the problem location: floor no. 4.
- The relays at the main entrance of UMIST identified the building location.
- The city fire station identified that there was a fire at the UMIST campus.
- The fire station personnel rushed to UMIST guided by guards at the entrance, reached the building master keys were collected from the foyer attendant guided by the detector relay system, the team reached the fourth floor guided by the floor relay display, and entered room 409 to extinguish the fire.

There are several key pointers here to show how differently this kind of situation would be handled today. The technological solutions evolve over time through a succession of improved choices. To begin with, the delays due to the human elements in the chain can be eliminated completely. Second, automated intelligent devices could be in place to facilitate movement at the campus and building entrance with precise location information. Also, the horizon of services offered by city agencies can be extended to ensure a coordinated response. For instance, there would be a medical team accompanying the fire personnel for handling medical emergencies. If the situation warrants, then the hospitals can be alerted to expect to receive the injured. This is precisely what a modern Edge architecture is expected to support. How this may be achieved today is shown in Figs. 1.5, 1.6, and 1.7.

The schematic in Fig. 1.5 corresponds to and can be easily derived from the generic framework shown in Fig. 1.4.

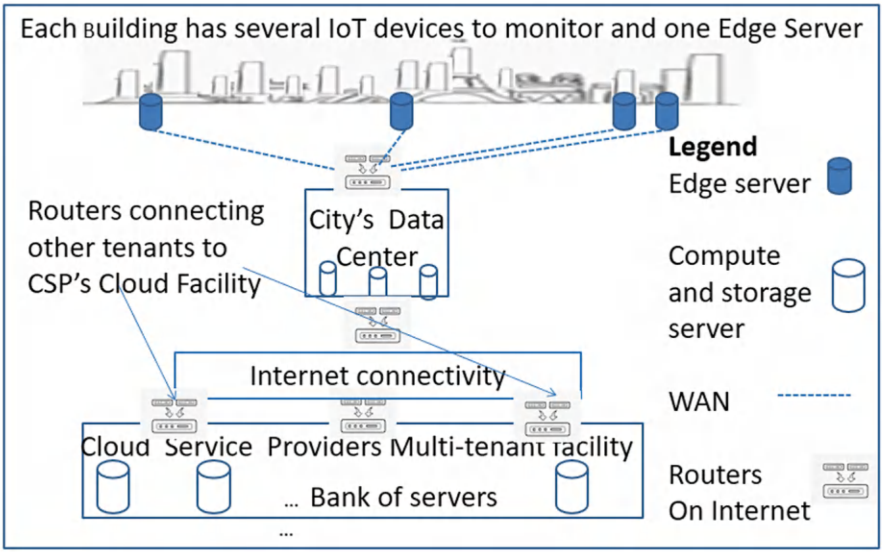

Each Building has several IoT devices to monitor and one Edge Server
Legend
Edge server
Compute and storage server
WAN
Routers On Internet
Routers connecting other tenants to CSP's Cloud Facility
City's Data Center
Internet connectivity
Cloud Service Providers Multi-tenant facility
... Bank of servers
...

Fig. 1.5 Edge servers at various premises connected to the city service center

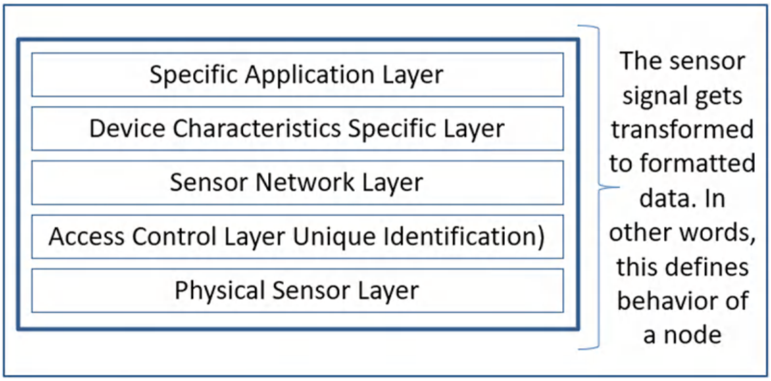

Specific Application Layer
Device Characteristics Specific Layer
Sensor Network Layer
Access Control Layer Unique Identification)
Physical Sensor Layer
The sensor signal gets transformed to formatted data. In other words, this defines behavior of a node

Fig. 1.6 Protocols to transform sensor signals to data

The narrative thus far is devoid of technical details. Therefore, it is imperative to provide some technical details about the IoT devices, sensor connectivity, and protocols to transform to connect to the Edge for processing. The key point is that the signals from sensors get converted into formatted data. IoT devices used for buildings typically include the following kinds of sensors:

- Smoke detectors, fire alarms
- Burglar alarms
- Hierarchical power utility sensors for rooms, floors, and buildings
- Sensors to monitor utilities like water supply in toilets and kitchens

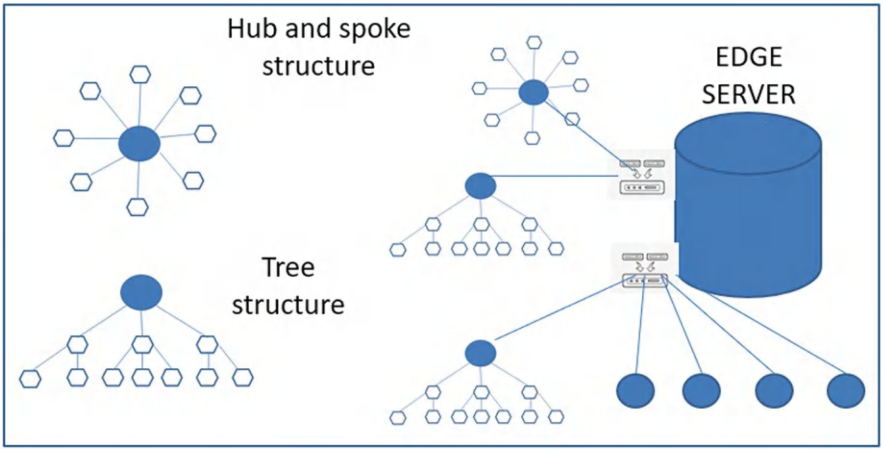


Fig. 1.7 Connection of the sensor networks to Edge servers

- CCTV cameras at floors, public spaces, foyers
- Parking lot entry authentication using RFID
- Goods movement and monitoring
- Biometric authentication for staff at the entrance to the building
- Ultrasound devices for paramedic support

The systems in place ought to be compliant with protocols for near-field communication (NFC), Wi-Fi, Bluetooth, etc. for device to device, device to gateway, etc. with minimally >128 kbps bandwidth while supporting Advanced Encryption Standard (AES) encryption. These devices provide core components of the micro-services needed to monitor the safety and security of buildings.

For our model, each node provides at least one micro-service and is internally characterized by the layers that transform signals into actionable data, as shown in Fig. 1.6.

The reference no. 11 offers a more detailed description of how protocol layers transform sensor-generated signals into data. Figure 1.6, therefore, shall be reckoned to define a node fairly correctly. Note that the bottom two layers are required to adhere to the Institute of Electrical and Electronics Engineers (IEEE) 802.15 suite of protocols [11]. The nodes so designed are expected to support data transfer upwards of 128 kbps for near real-time response. In fact, there are standards set by IoT organizations, such as https://csa-iot.org.

So, for our scenario, a typical building may be installed with IoT device-supported nodes, each providing a micro-service.

In general, these nodes may be connected in hub-and-spoke formation or as a tree network to build a hierarchy. In the tree structure, the leaf nodes would be connected to IoT devices, and the intermediate nodes would be typical multi-channel data acquisition systems or even gateways connecting to a hub or low-end Edge device. The hub could be a low-end Edge device to connect to an Edge server via a router, as shown in Fig. 1.7. Some independent nodes may have a higher level of functionality and may be directly connected to the Edge server. Each building in the city may have an Edge server and gather data from all the sensors in the building. These servers would be connected to city service centers. The overall latency may range from 6 to 20 milliseconds to support near-real-time operations [12].

This example predominantly used IoT devices. The next example will be to show how mobile Edge devices may be utilized to support low-latency real-time operations to give another facet of Edge architectures.

1.3.2 The Edge Architecture: Scenario 2

The modern mantra of economy of scale promotes mass production of consumer goods. This in turn entails immediate product distribution to the consumer locations. A major part of such distribution is undertaken by trucks. The fleet operators like to operate truck convoys for safety, security, and reliability, besides the economy of operations. The convoys sometimes fork into fragments and sometimes merge. Tracking the convoys as well as individual trucks is a real-time monitoring and control problem [13].

In the USA, the National Highway Traffic Safety Authority (NHTSA) records suggest that there are over 9 million registered trucks engaged in goods movement in the USA [14]. No wonder NHTSA mandates strict safety requirements, such as an onboard two-way radio system for vehicle-to-vehicle (V2V) communications (to be aware of other vehicles on the road).

Modern trucks have a lot of embedded electronic controls and monitoring instruments connected on a local intercommunication network (LIN), an automotive Ethernet network or a very sophisticated controller area network (CAN). These networks collect data and display it on the truck's dashboard. Various sources of data are the fuel injection system and the engine management system, transmission systems for speed regulation, automotive brakes, safety air bags, vehicle suspensions, and power windows, besides various channels for audio (notifications) and video visuals (traffic, GPS location, etc.). The convoys additionally require information about the convoy chain. When and if it fragments, the chain status is updated. For complete convoy management, our solution recommends positioning an Edge device on each truck. Therefore, it would be desirable to understand the characteristics of Edge devices. Edge devices have the following two key characteristics:

1. It sits at the boundary of two networks, basically connecting them.
2. Minimally, it is an entry-exit gateway like a firewall or connects a campus network to a WAN. Maximally, it can regulate and monitor to control transmission rates and decide on routing with some degree of filtering, translation, or transformation in addition to a fair amount of dynamic storage.

In the backdrop of the fact that NHTSA also reckons that 80% of goods movement in the USA [14] is by trucks operating with continuous connectivity with Cloud backup support, real-time monitoring is desirable. Besides the USA, large geographies like the European Union or other parts of the world, including India, have good cellular coverage. So, if each truck in the convoy has an Edge device, then it is assured of continuous connectivity comparable to a 4G or 5G cellular device. This, then, gives us the basis to beget a convoy management solution. In fact, such a solution would be part of a fleet operator's Cloud-based management platform. At all points in time, a fleet operator may have many convoys to track and manage. This is achieved by operating base stations that are geographically spread out to collect real-time data about the convoys that are in their well-demarcated area.

It is obvious that the convoy management requires near-real-time distributed data management that must satisfy consistency and reliability of operations [9]. The solution architecture using Edge devices is described in Figs. 1.8, 1.9, and 1.10 with the accompanying explanations offered next.

The explanation of Fig. 1.8 is as follows:

(a) Each truck has an RFID tag for unique identification.
(b) The convoy chain is formed by the RFID tags on the trucks.
(c) The Edge device on the truck may store biometric information for secure operation.
(d) Each convoy has a clearly identified leader.
(e) The Edge device on a leader has an additional task of convoys data collation.
(f) Each truck in the convoy is capable of acting as a leader.
(g) A single truck forking out is also a leader.

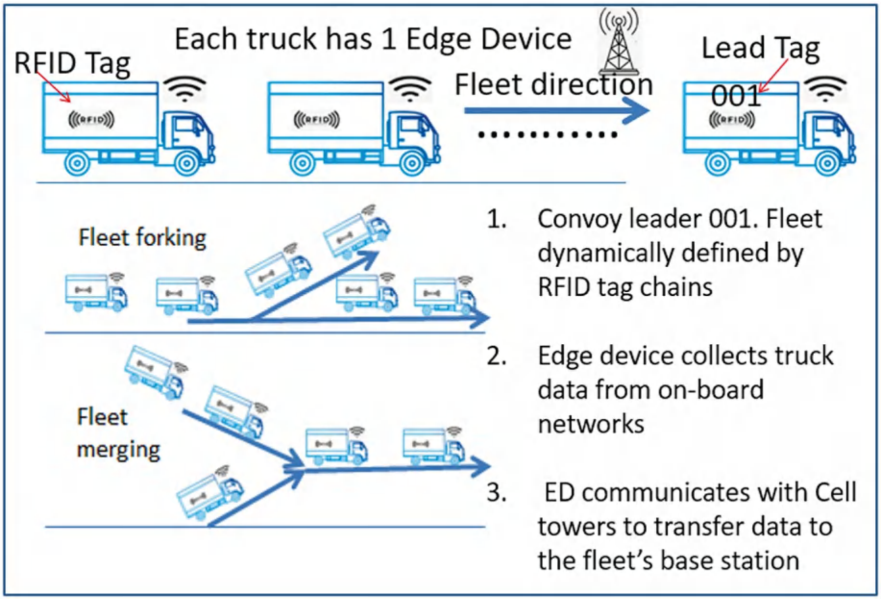

Each truck has 1 Edge Device
RFID Tag
Lead Tag
001
Fleet direction
Fleet forking
Fleet merging
1. Convoy leader 001. Fleet dynamically defined by RFID tag chains
2. Edge device collects truck data from on-board networks
3. ED communicates with Cell towers to transfer data to the fleet's base station

Fig. 1.8 Data for convoy composition, location, speed, timekeeping, etc

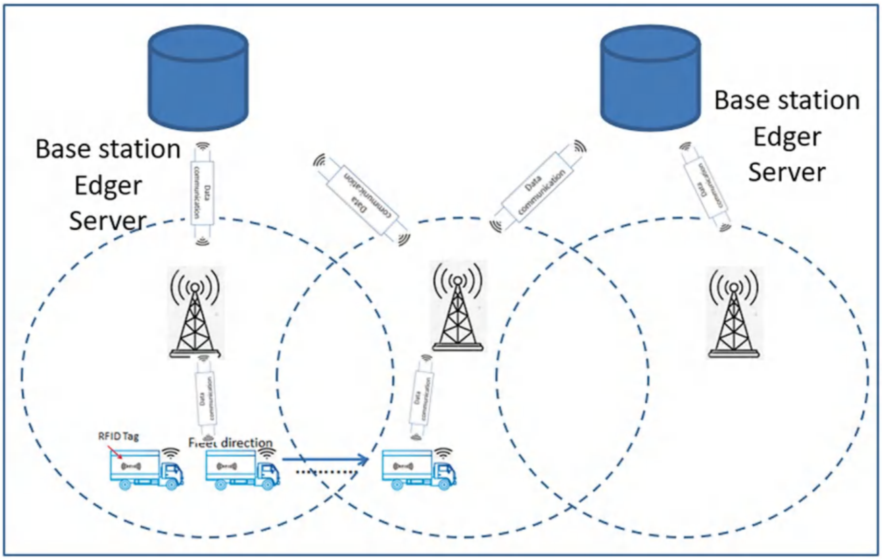


Fig. 1.9 Convoys move forward and transition to the next cellular zone

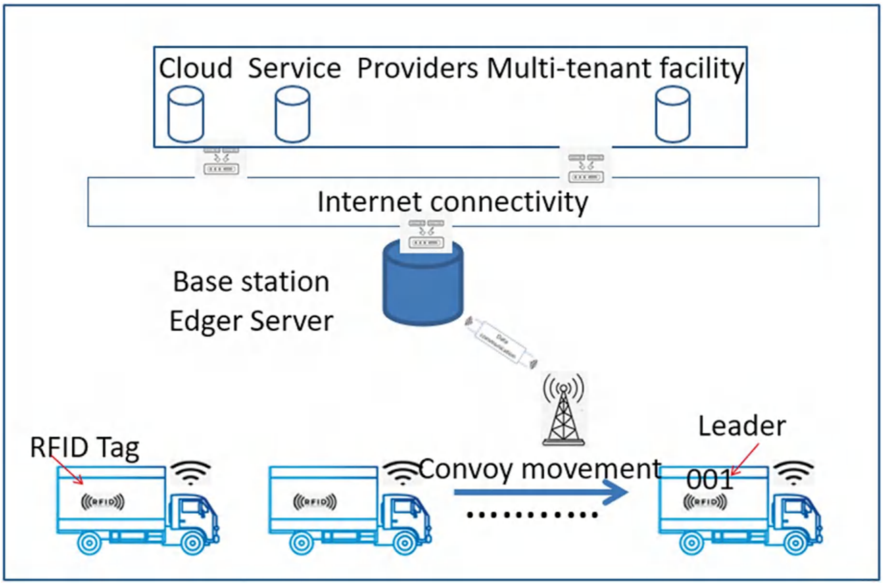


Fig. 1.10 Schematic of Edge computing–based convoy management

(h) The convoy chain may fork and merge, generating new convoy formations.
(i) Trucks use cellular networks to communicate with each other and the leader.
(j) The onboard network collects operational data for display on the truck's dashboard.
(k) The Edge device interfaces the onboard network with the cellular network.
(l) The Edge device also stores summarized data for each truck for later analytics. It is important to do this to ensure that the vehicles return for regular maintenance and even being retired.
(m) The leader summarizes convoy data to the fleet's base Edge server (see Fig. 1.9).

The operational details captured in Fig. 1.9 are described next.

(a) The convoy communicates with the fleet's base station via cell towers.
(b) A convoy may move over a vast region, connecting with different cell towers.
(c) The leader keeps uploading the summarized convoy track data and the health information to the fleet's closest monitoring base station. The fleet base station can also respond when and only if an emergency arises.
(d) As the convoy moves, the fleet base station Edge servers hand over and take over convoy monitoring. This is similar to the way air traffic control systems work. An aircraft's movement is monitored in near-real-time over distinctly identifiable flight information regions.
(e) The Edge servers may upload each convoy's data to the Cloud for longer-term analysis (see Fig. 1.10).

The operational details captured in Fig. 1.10 are described next.

(a) The fleet's base station is connected to a Cloud.
(b) The Cloud may be a private Cloud or multi-tenanted facility provided by a CSP.
(c) The Cloud collects data from all the fleet base stations.
(d) This data can be analyzed for decision support that can help in the following ways:

    - Forming convoys
    - Loading and unloading of goods (how not to return empty trucks instead carry different goods)
    - Route planning
    - Profitability of routes of operations
    - Augmenting capacities or rerouting, etc.

The convoy and fleet management system schematic shown in Fig. 1.10 is easily derivable from the high-level framework shown in Fig. 1.4.

Though not mentioned earlier, the Edge devices may also interface with digital assistants like Alexa and Siri. Edge devices may also be connected with IoT sensors, actuators, or robotic channels using ultrasound, Bluetooth and other near-field communication (NFC) devices. When endowed with AI capabilities and as part of filtering operations, Edge devices perform tasks like spam filtering and encryption [15]. The enterprises have used Edge devices for preferred customer face and voice recognition for shopping. Busy airport hubs in the USA use it for frequent flier face recognition to manage heavy traffic [16].

It is obvious that Edge is expected to respond to its environment with timely decisions and seeming intelligent actions. This assertive mode of operation entails that the judgment be judicious. Only then embedding AI capabilities at the Edge would make sense. So, we first need to briefly explore the AI models and how these models influence decision-making.

1.3.3 AI and ML on the Edge: Scenario 3

This example illustrates the value and modality of adding AI and ML capability to an Edge server. Consider an office building with a large cafeteria that has many tables and chairs spread across a wide area. There are celling lights in the cafeteria, along with heating and cooling vents for temperature control. Since people come and go for breakfast, lunch, and an evening snack at different times, there is no need to have lighting and air flow be available on a 24 × 7 basis. These can be easily regulated by timers and temperature sensors. However, even during the times when people are present in the cafeteria, not all the tables would be occupied. Controlling energy expenditure on a need basis will result in substantial financial savings.

With this view, an Edge server with several sensors can be installed in the building. It can detect as people start to stream in, grab their food, and occupy a table. Then light and air vents overhead can be turned on. When people finish eating and leave, then an empty table is detected to turn off lights and air vents. This by itself does not need AI, but if a pattern can be established over time, then selectively air vents can be turned on in advance to ensure that people walking over to a table will immediately experience cooler or warmer temperatures instead of waiting for the air-conditioning or a heater to turn on and take effect in a few minutes. Furthermore, via an app, immediate control can be granted to the people sitting on the table to adjust their local temperature and lighting as they desire without affecting other tables or cafeteria occupants.

An intelligent system may remember personal settings, so when a specific person or a group of people occupy a table, then environmental controls can be personalized to their previous settings. Note that the goal of such a system is to save money for the cafeteria operator but also offer a pleasant environment to its users.

1.4 Building Blocks for AI on the Edge

Artificial intelligence (AI)-endowed systems are increasingly finding acceptance. It is largely because the current computing capabilities and support systems presently available make it possible to translate some hyped systems and concepts to be in fact realizable. A minimal understanding of AI, which encompasses machine learning (ML), requires exploration of the AI models. The topic of AI (and ML) is too vast to be covered in detail. We will restrict to the essentials. For a more detailed coverage readers may explore the books [17, 18]. The book by Stuart Russell and Peter Norvig recommends incorporating software intelligent agents in the software, which is like embedding an AI model in the devices.

1.4.1 AI Models

Depending upon the context, AI models are used primarily for three kinds of ML tasks. These are: classification, prediction, and clustering. For example, a task like spam detection qualifies to be a classification task. Similarly, a prediction task may reflect on the expected time of arrival at the destination based on the current location and traffic pattern on the way. As for the clustering task, a discovery of a hidden data pattern is required.[3] This may be based on some data attribute, as that is very helpful in creating groups or clusters. For example, in a social gathering, one tends to gravitate to form a cluster with no intent except perhaps the topic of discussion like jazz, tennis, the stock market, etc. Sometimes clustering appears like a classification of some kind, but it has to be noted that classification always requires a priori definition of classes, whereas clusters are formed in situ and have dynamic membership.

Note that currently AI models are trained using a lot of historic data, then validated using some test data sets, and lastly deployed in the field for inference in real time. However, in the future, this flow may need to evolve as training is not a one-time activity if the nature of the problem keeps changing. Then we will need to consider incremental training and model updates in the field.

1.4.2 Information Security Trends

Edge computing is a combination of distributed intelligent sensors and computing devices connected to centralized servers. Security problems in Edge computing arise from a large attack surface due to computing devices, data transfer over the communication equipment, and storage facilities. These span across the location of Edge devices, large data centers, and networks connecting them. Threat vectors include malicious actors accessing information content for observation and alteration, interfering with the operational capability for unauthorized access, etc. Solutions need to consider prevention from and response to any security threats [6].

[3] An example of a cluster is the collection of documents retrieved based on content from a search on the Web.

1.4.3 Edge Computing Challenges

A traditional data center consists of racks of servers, storage, and network devices in a large, air-conditioned facility. Then scores of clients and IOT (Internet of Things) devices interact with this datacenter over public and private networks. However, this setup causes latency, which is unacceptable for real-time computing applications that can involve instant decision-making using AI. There is a need to support AI applications at or near the Edge of computing networks to minimize latency and address data privacy concerns. Such concerns currently cause many customers, such as hospitals, to not put their data in the Cloud.

The current setup limits Edge computing to deploying sensors for data collection, but all or most of the AI tasks are done on central servers in a large datacenter. This needs to change so intelligent devices can be deployed at the Edge of a network and be used for making on-the-spot AI-based decisions.

1.5 This Book's Organization

The next generation of Cloud computing is needed to minimize latency and address privacy/security concerns of many customers. This book highlights the problems and proposes new solutions for AI on the Edge of computing networks. The book has two parts: Foundations in the first six chapters and practices in the next four chapters.

The first chapter introduces the notion of Edge computing with AI. Then the next four chapters lay the foundations of computing on the Edge, IoT, AI, and information security, respectively. Each of these foundational chapters introduces the basic concepts, examines the historic evolution and current limitations, and then enumerates the emerging needs in respective domains. The first part of the book concludes with Edge artificial intelligence in the sixth chapter.

The second part of the book consists of current practices. It starts with the security concerns and performance issues on the Edge in the seventh chapter, proposing a new distributed trust model and solutions using federated learning. Then we look at some use cases for the intelligence of the Edge computing, with examples of smart building energy management, medical data sharing by hospitals, and social media consumption in Chap. 8. Next, the role of regulatory agencies is examined in the ninth chapter, including the FDA and laws such as Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR). The book concludes with the tenth chapter by proposing AIaaS (AI as a Service), its price and performance considerations, and some emerging trends at the Edge.

Each of these chapters includes points to ponder as an exercise for the readers to stretch their learning boundaries and our proposed answers for these questions.

1.6 Summary

Currently, many sensors are deployed at the Edges of a network for data collection, which is then sent to central servers in a datacenter. This model introduces latency in training an AI model and using it for inference on the Edges of the network. Adding sufficient intelligence and storage capacities to the Edge devices will enable them to do incremental updates to the model and use it to perform AI inference operations in the field.

References

1. https://www.livescience.com/20718-computer-history.html
2. https://www.wallstreetmojo.com/moores-law/
3. https://medium.com/@micahyost/a-brief-history-of-software-development-f67a6e6ddae0
4. https://www.usg.edu/galileo/skills/unit07/internet07_02.phtml
5. https://en.wikibooks.org/wiki/A_Bit_History_of_Internet/Chapter_5_:_Client-Server
6. Sehgal, N. K., Bhatt Pramod Chandra, P., & Acken, J. M. (2023). *Cloud computing with security and scalability*. Springer.
7. https://en.wikipedia.org/wiki/AI_winter
8. https://en.wikipedia.org/wiki/Time-sharing#Notable_time-sharing_systems
9. Martin, K. (2017). *Designing data intensive systems*. Book, O'Reilly.
10. Syed, M. (2010). "Bounce" Harper Collins.
11. https://www.digi.com/solutions/by-technology/zigbee-wireless-standard
12. https://en.wikipedia.org/wiki/Real-time_computing
13. https://www.truckinginfo.com/149411/mobile-communication-making-it-all-work
14. https://www.its.gov
15. https://edge-technology.com/spam-internet-security/
16. https://www.cyberlink.com/faceme/insights/articles/204/Facial-Recognition-at-the-Edge-The-Ultimate-Guide
17. Stuart, R., & Peter, N. (2010). *Artificial intelligence: A modern approach*. Prentice Hall.
18. Gupta, P., & Sehgal, N. (2021). *Introduction to machine learning in the cloud with python: Concepts and practices*. Springer.

Chapter 2
Foundations of Computing at the Edge of Networks

2.1 Historic Evolution of Computing

For more than a half century, computing technologies have been evolving in several phases, as described below. Though there appears to be a cyclic relationship between them, in reality, computing has grown as an outward-going spiral, as shown in Fig. 2.1.

1. *Phase 1 (Mainframes):* This was the era of large mainframe systems in the backrooms connected to multiple users via dumb terminals. These terminals were electronics, or electromechanical hardware devices, using separate devices for entering and displaying data. They had no local data processing capabilities. Even before the keyboard or display capabilities were card-punching systems with JCL (job control language). JCL was an early form of scripting language to give instructions to mainframe computers [1].

   The main takeaway of this era is the concept of multiple users sharing the same large machine in a backroom and each unaware of other users. At an abstract level, this is similar to Cloud computing with users on thin clients connected to racks of servers in the backend data centers. Information Security (INFOSEC) relied entirely upon restriction of physical access to the computational machinery.

2. *Phase 2 (PCs and Workstations):* This was the era starting in the 1980s with the advent of personal computers (PCs), many of which were stand-alone or connected to mainframes via slow modems [2, 3]. Each user interacted with a PC on a one-on-one basis, with a keyboard, a mouse, and a display terminal. All the storage, computing power, and memory were contained within a PC box. Any needed software was installed via floppy disks with limited storage capacity to run on the PCs. These PCs evolved in the early 1990s to laptops with integration of display, keyboard, mouse, and computing in a single unit. There was also an

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_2

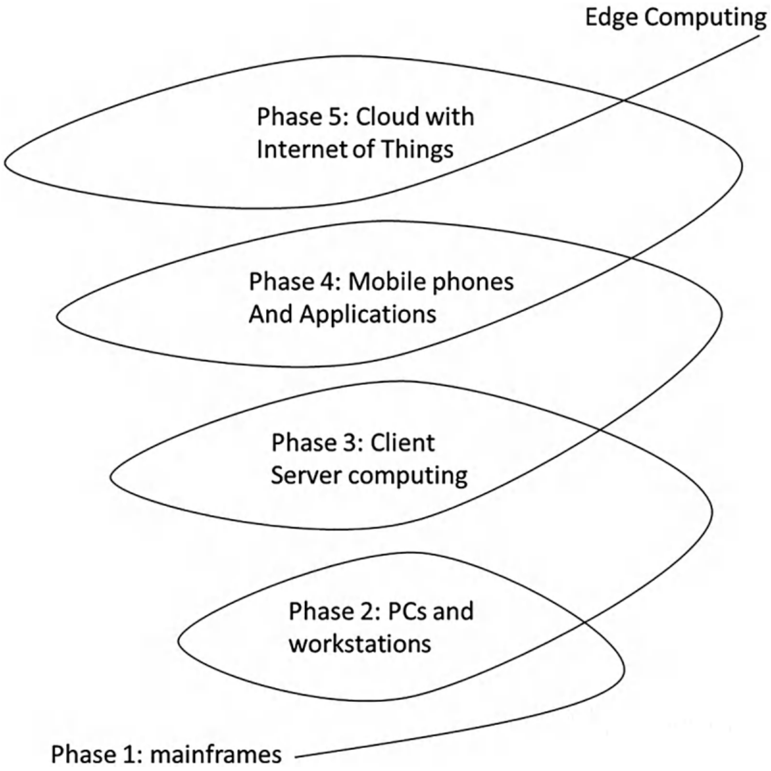

Edge Computing
Phase 5: Cloud with Internet of Things
Phase 4: Mobile phones And Applications
Phase 3: Client Server computing
Phase 2: PCs and workstations
Phase 1: mainframes

Fig. 2.1 Evolution of computing models in a spiral of growing computing needs

attempt in the early 1990s to create a network computer, which was diskless and connected to more powerful computers in the back end. Perhaps the idea was before its time, as networks were still slow with 28.8 kbit/s dial-up modems [3]. A more practical paradigm that became prevalent is the client-server solution [4], which enabled remote devices with a little computing power to connect with servers over corporate Ethernet networks.

The main takeaway of this era was the birth of an online desktop to emulate the desk of working professionals. It represented multiple tasks that were simultaneously kept in an open state on the PC. With graphical user interface (GUI) and operating system (OS) utilities, it was possible to create the notion of a desktop on a computer, to go from a single user-single job model to single user-multiple jobs running simultaneously. This caused user interactions to move from command prompts to mouse-driven clicks.

3. *Phase 3 (Client-Server Computing):* In the mid-1990s, the era of Web browsers [5] became prevalent. These are software applications to retrieve, present, and traverse information resources on the World Wide Web (WWW). These came out of a research project but became popular with everyday computer users to

access information located on other computers and servers. Information often contained hyperlinks, clicking which enabled users to traverse to other Websites and locations on the WWW. These browsers needed full-fledged PCs to run on, which were more powerful than dumb terminals.

> The main takeaway of this era was the birth of WWW, with PCs forming a gateway for connecting users to the Internet. It used Internet infrastructure to connect with other devices through the Transmission Control Protocol/Internet Protocol (TCP/IP) protocol for accessing a large set of resources.

4. *Phase 4 (Mobile Phones and Apps):* The new century heralded an era of full-fledged Internet browsing with PCs and a mobility revolution with cell phones. It was inevitable that the twain shall meet, launching innovative mobile applications running on cell phones. Cell phones created yet another gateway to the Cloud. This enabled users to book hotels, rent rooms, and buy goods on the move.

   > The main takeaway of this era was the birth of mobile clients, similar to the client-server model, except with limited compute power in small form factors. Thousands of powerful servers were located in large and, often remote, data centers. It may be noted that this represented one revolution of spiral akin to mainframes, as depicted in Fig. 2.1.

5. *Phase 5 (Internet of Things):* When companies discovered that the population of the world limits the number of smartphones and mobile devices they can sell, they started to look for new business opportunities. These came in the form of the Internet of Things (IoT) paradigm, which enables everyday common objects such as a television, a refrigerator, or even a light bulb to have an IP (Internet Protocol) address. This gives rise to new usage models; for example, to conserve energy, IoT objects can be remotely monitored and turned on or off. This also includes consumer services for transportation and a utilitarian phase for user interactions with appliances, leading to higher productivity and improved quality of life. Computing reach resulted in better-informed decision-making and extended social relationships. Lately, the information security considerations have also become critical and will be briefly discussed later in this chapter.

This era completes the cycle of computing evolution, with many front-end devices connected to the powerful servers in the Cloud on the back end, as shown in Fig. 2.1.

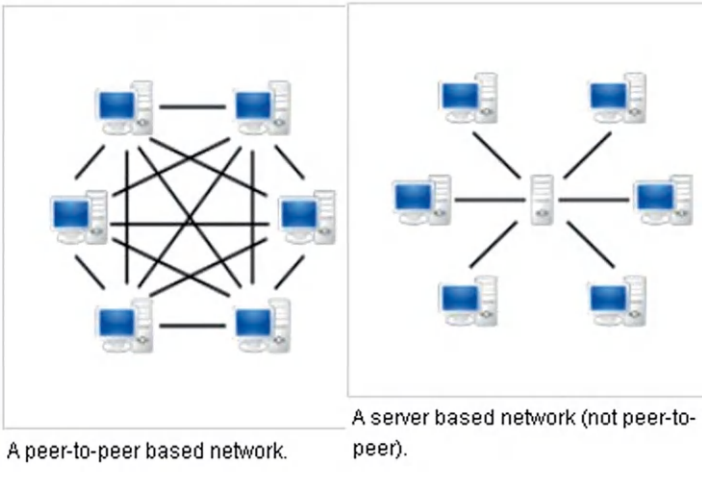
Another way to look at the computing transitions is by imagining a pendulum that oscillates between centralized servers on one end and client devices on the other. In phase 1, computing power was concentrated on the mainframe side, while in the next phase, it shifted to the PC side. Such oscillations continued across different phases of the spiral shown in Fig. 2.1. Now we are evolving beyond phase 5 with Edge computing, with some limited storage and compute capacity on local devices, while large data centers in the Cloud are used for data backups or deeper analysis. This is also referred to as Fog computing. In this usage model, computing resources may be located somewhere between the data sources on the Edge and the Cloud data center.

2.2 Historic Evolution of Networking

Two models prevailed in the networking domain: peer-to-peer and client-server. In the former, each computer that wants to talk to another computer needs a dedicated line, so N machines will need N^2 connections. This while being fast was obviously not scalable; also, when no data is being transferred, then network capacity is unutilized. Meanwhile, the client-server model with a central server supporting multiple clients was economical and scalable for the enterprises, as data and files could be easily shared between multiple users, as shown in Fig. 2.2.

In networked systems with a client-server model, a computer system acting as a client makes a request for service, while another computer acting as a server provides a response when a request is received, as shown in Fig. 2.3.

The greatest advantage of the networking model was its scalability [6], i.e., one could add any number of servers and clients. It offered an opportunity to create open-source software that offered interoperability. The network protocol was based on TCP/IP, and going forward, this growth led to several interconnecting networks, eventually leading to the Internet using the HTTP protocol. The protocol-based control meant that one could operate without vendor lock-in, i.e., not be dependent on any single supplier. Networks within an enterprise grew as LAN (local area network) and WAN (wide area network). LAN-connected computers in confined areas, such as within a single office building, or there could be several LANs within individual confined areas, e.g., one for each floor. WAN-connected computers spread over larger geographical areas, such as across the cities. Several LAN standards were developed over time, such as Institute of Electrical and Electronics Engineers' IEEE 802.2, Ethernet or IEEE 802.3, IEEE 802.3u for faster Ethernet, IEEE 802.3z and 802.3ab for Gigabit Ethernet, IEEE 802.5 for token rings, and IEEE 802.12 for


A peer-to-peer based network.

A server based network (not peer-to-peer).

Fig. 2.2 Two networking models

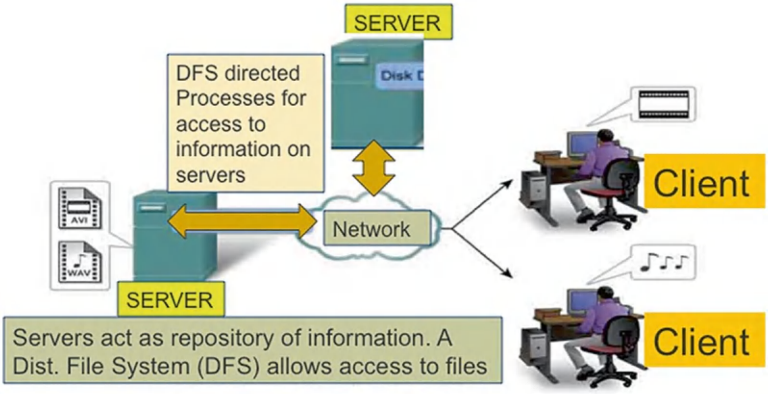

SERVER
Disk C
DFS directed Processes for access to information on servers
AVI
WAV
Network
Client
Client
SERVER
Servers act as repository of information. A Dist. File System (DFS) allows access to files

Fig. 2.3 The networking model

100VG on any LAN. The Internet in turn is simply a net of networks, using a TCP/IP suite of protocols. These were mostly packet switching protocols, with each packet having a structure offering a virtual connection and no single dedicated path committed to maintain the connection between a source and its destination. Paths were temporarily created to maintain connections as needed between users and/or machines. Sometimes, time division multiplexing was used to share the same physical line between different pairs of users, and every pair was given a short time slot. As an example, if a time unit has ten slots, then ten pairs can communicate simultaneously, oblivious to the presence of others. Connections are made over a network of switched nodes, such that nodes are not concerned with the content of data. The end devices on these networks are stations such as a computer terminal, a phone, or any other communicating device. Circuit switching operates in three phases: to establish a connection, transfer data, and then disconnect. Networking routers use an intelligent algorithm to determine the optimal path that takes the least time or to minimizes cost to establish a path of communication.

TCP/IP is a fault-tolerant protocol [6], so if a packet is lost or corrupted, then the connection is retried and the packet is sent again. It was originally designed by DARPA (Defense Advanced Research Projects Agency) during the Cold War era to survive a nuclear attack. TCP/IP is a layered protocol where TCP provides reliability of a connection by attempting to send the same packet again if the previous transmission failed, while IP provides routability between the communicating nodes by finding an optimal path.

When computing needs exceed what can be reasonably supported by a single server, an effort is made to share the workload among multiple servers. Note that in sharing a workload, the latencies on servers that are loosely connected shall be determined by the TCP/IP suite of protocols. The concern often is that these latencies would be in far excess of what may be acceptable. Therefore, an alternate approach is to use server clusters. The latency in tightly bound server clusters is far less than the networked servers. One of the techniques for clustering is by sharing

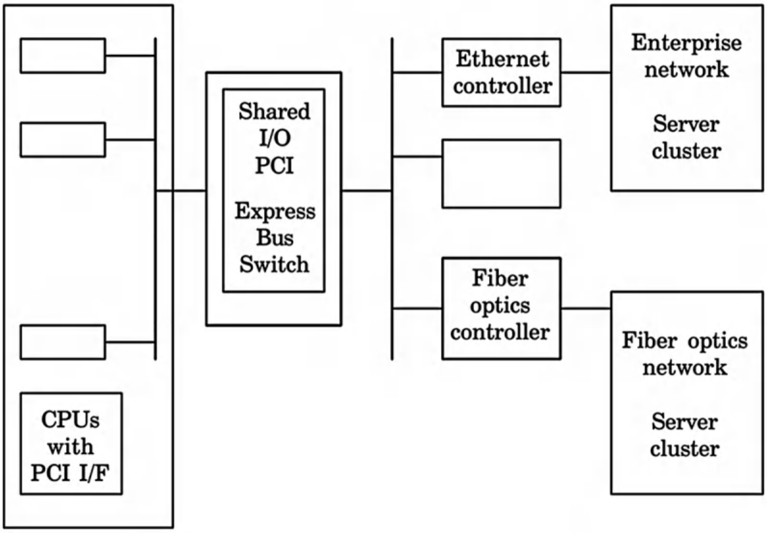


Fig. 2.4 Connecting multiple servers in a cluster

I/O switches between the servers [7]. The switches connect individual classes of controllers, such as Ethernet controllers, in an enterprise network, as shown in Fig. 2.4.

Clustering ensures that large applications and services are available whenever customers and employees need them. This enables IT managers to achieve high availability and scalability for mission-critical applications, such as corporate databases, email, Web-based services, and support external-facing retail Websites. Clustering forms the backbone of enterprise IT's servers and storage elements. Clustering may operate with storage area network (SAN) or Internet small computer system interface (iSCSI) architectures [7] as shown in Fig. 2.5.

2.3 Roots of Cloud Computing

The term "Cloud computing" became popular about two decades ago [6]. However, its roots extend at least half a century back when users sat in front of blinking terminals far away from mainframe computers connected via cables. Telecommunication engineers about a century ago used the concepts of the Cloud. Historically, telecommunications companies only offered single dedicated point-to-point data connections. In the 1990s, they started offering virtual private network (VPN) connections with the same quality of service (QoS) as their dedicated services but at a reduced cost. Instead of building out physical infrastructure to allow for more users to have their own connections, telecommunications companies were now able to provide users with shared access to the same physical infrastructure. Later on, notions of

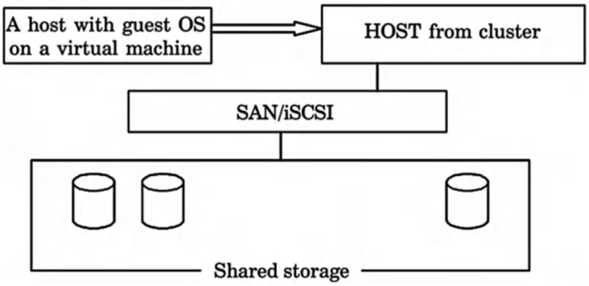


Fig. 2.5 Connecting multiple storage elements in a cluster

utility computing, server farms, and corporate data centers formed the foundation of the current generation of large Cloud data centers. The same theme underlies the evolution of Cloud computing, which started with mainframes and now has evolved to 24 × 7 services (24-h service available 7 days a week, i.e., no downtime). The following list briefly explains the evolution of Cloud computing:

- Grid computing: Solving large problems using parallelized solutions, e.g., in a server farm
- Utility computing: Computing resources offered as a metered service
- SaaS: Network-based subscriptions to applications
- Cloud computing: "Anytime, anywhere" access to IT resources delivered dynamically as a service

Server farms didn't have any application programming interface (API). Each user was made to think that they had full access and control of a server, but in reality, time-sharing and virtual machine isolation kept each user's processes independent of others.

According to National Institute of Standards and Technology (NIST) [8], any Cloud must have the following five characteristics, as listed below:

1. Rapid Elasticity: Elasticity is defined as the ability to scale resources both up and down as needed. To the consumers, the Cloud appears to be infinite, and they can purchase as much or as little computing power as they need. This is one of the essential characteristics of Cloud computing in the NIST definition.
2. Measured Service: In a measured service, aspects of the Cloud service are controlled and monitored by the Cloud provider. This is crucial for billing, access control, resource optimization, capacity planning, and other tasks.
3. On-Demand Self-Service: The on-demand and self-service aspects of Cloud computing mean that a consumer can use Cloud services as needed without any human interaction with the Cloud provider.
4. Ubiquitous Network Access: Ubiquitous network access means that the Cloud provider's capabilities are available over the network and can be accessed through standard mechanisms by both thick and thin clients.

5. Resource Pooling: Resource pooling allows a Cloud provider to serve its consumers via a multi-tenant model. Physical and virtual resources are assigned and reassigned according to consumers' demand. There is a sense of location independence in that the customers generally have no control or knowledge of the exact location of the provided resources but may be able to specify a geographical location (e.g., country, state, or data center).

2.3.1 Types of Cloud

Clouds can be broadly defined in the following three categories:

1. Private Cloud: Private Cloud is owned and used by a single organization. An example is Intel's own data centers, which are used to run its chip design tasks. These are not put in a shared public facility for perceived security and calculated cost-saving reasons.
2. Public Cloud: Public Cloud offers its services to a full range of external customers. The computing environment is shared with multiple tenants on a free or pay-per-usage model. Examples include Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft's Azure Cloud offerings.
3. Hybrid Cloud: Hybrid Cloud is a computing environment that uses a mix of on-premises private Cloud and third-party public Cloud services with orchestration between the two platforms.

While economics is the main driver of public Cloud computing, its customers want the same performance and security aspects that they enjoy on a private server. This is a classic case of wanting to have your cake and eat it too. Specifically, public Cloud users want full observability and controllability for their workloads. This refers to the applications and data they deploy on a remote server. They also expect a Cloud server to be secure as if it were on their own premises behind a firewall. These requirements are well captured in NIST's five essential characteristics that we listed in the previous section. However, there are some ambiguities; for example, "on-demand self-service" requires that a consumer can unilaterally provision computing capabilities without requiring human interaction with a Cloud service provider. This implies automation on the Cloud service provider's site but doesn't specify anything on the user side. In reality, for a user to provision hundreds of jobs at a moment's notice or to monitor them requires some automation. Furthermore, if a particular application is not behaving well, then the user will need some diagnostics to identify the root cause of problems and be able to migrate the job to another Cloud server. This implies the availability of suitable monitoring and alerting tools. Automation of actions is such that the user environment and the Cloud provider's environment work in unison. This will help to realize the full potential of Cloud computing. In Fig. 2.6, different phases and options for Cloud architectures are shown. In practice, one of these or a combination of these is adopted by users.

Most enterprises already own some servers and storage facilities in their private data centers. These form a private Cloud for the owning entity, in which only its jobs can run. In contrast, a public Cloud is open to anyone on the internet who can pay for it as a utility service on demand. The last category shown in Fig. 2.6 is hybrid Clouds, which are a mix of both private and public Cloud facilities used by some enterprises.

For customers with confidential or performance-sensitive workloads, public Cloud providers offer a virtual private Cloud, which can be thought of as a hosted service, or a group of dedicated servers that are cordoned off for such a customer. These offer dedicated facilities but at a higher cost since the Cloud service provider can't share this infrastructure with other customers. This is preferable for some Cloud users who do not wish to maintain their own IT services but want the assurance of privacy.

Users may experience large variations in the Cloud consumption profiles, some of which exceed their internally installed server base. Then they have two choices, either to buy more servers or let some computing demand remain unsatisfied at peak usage time periods.

A private Cloud requires more capital investment and operational costs; however, some of these servers may remain idle during off-peak times. This is a case where the need is less than the installed capacity. The second case is when the need is greater than the installed capacity. It has an implication of lost business opportunities as user tasks will need to wait in a queue, or worse yet, customers will switch to somewhere else to meet their need. This is particularly true for online retailers who can't respond to their shoppers in a timely manner. This dilemma is depicted in

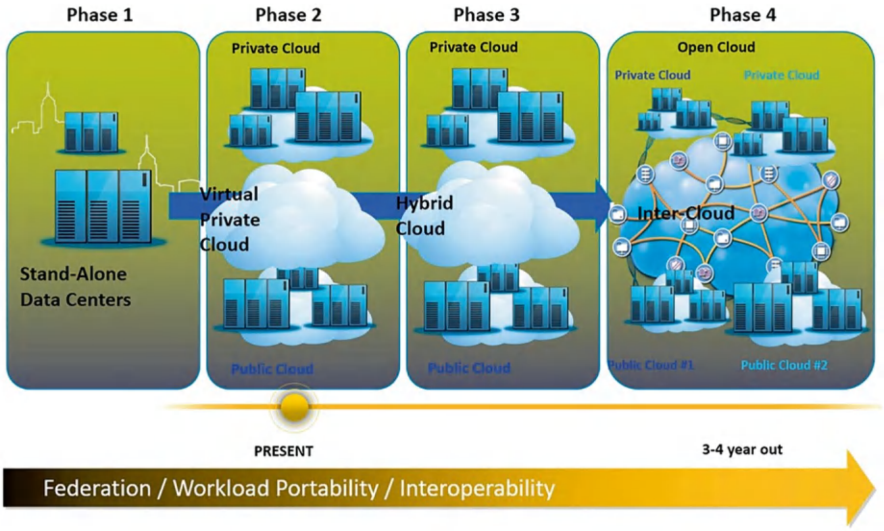


Fig. 2.6 A phased approach to adoption of Cloud computing

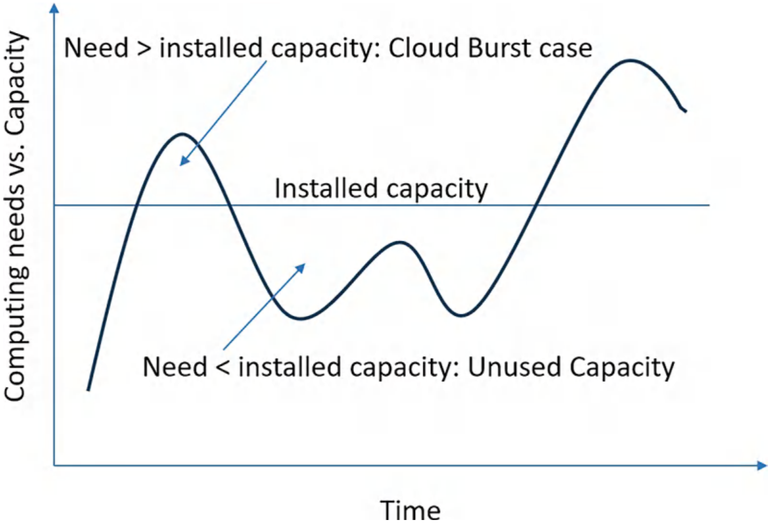


Fig. 2.7 Variations in computing needs of an organization

Fig. 2.7, where one way to meet the unsatisfied demand is via the migration of computing tasks to a public Cloud.

"Cloud bursting" is a term used to define the jobs that were running in an internal data center but are moved out to a public Cloud at peak usage and then return back to a private Cloud when the internal capacity is available. Since public Cloud charge on a pay-per-use basis, this proposition is attractive. However, as we will see in later chapters, this is nontrivial because computing jobs typically require a setup, which includes an operating system environment, sometimes with a plethora of supporting tools and often a large amount of data. Such jobs can't be easily migrated between private and public Cloud at a moment's notice, which means that computing environments on both sides need to be kept in synchronization. One way to do this is by data mirroring between the data centers, so only computing programs need to migrate, while associated databases always stay in synchronization. However, this adds to the operational cost of keeping the public Cloud environment always ready to go at a moment's notice.

2.4 Information Security Basic Concepts

The previously defined Cloud computing business models and implementation architectures have extended access to a wide variety of capabilities. Consequently, its security needs have also increased beyond the basic information security issues. However, basic security concepts still apply. Information security or INFOSEC begins with access control. Access control includes several abilities, including issuing control commands, sending information, reading information, and physical access to locations or machinery. Access control is based upon identity authentication. The entity to be identified can be a person, a device, or a computational process. There are four basic factors of identity authentication: information, physical item, biological or physical characteristics, and location. These factors involve answering the four key questions of identity authentication:

1. What do you have?
2. What do you know?
3. What you are?
4. Where you are?

Examples of the information factor are username and password, birthdate, or mother's maiden name. Examples of the physical item factor include a car key, credit card, debit card, or employee badge. What you are as a person is called biometrics. Examples of what you are include fingerprints, blood, DNA, eye scans, and voice patterns. Examples of location metrics are GPS coordinates, city or state, current time, current temperature, and altitude. The most common form of access control is based upon the single factor of what you know, specifically username and password. Another common form of authentication is based upon the single factor of what you have (a credit card) or sometimes what you know (your credit card number). A common two-factor authentication (2FA) is a debit card with a pin. The card is the physical device you have, and the pin is the information you know. In general, access control is improved with added factors for authentication, that is, by using multifactor authentication (MFA). There are many trade-offs for identity authentication and access control. The trade-offs include level of security, speed of performance, usability, and acceptability of method.

The next category of security for Cloud computing is protecting information both during transmission and during storage. Protection of information includes keeping secrets and private data away from unauthorized entities, preventing changes by unauthorized entities, and detection of attempts at tampering with the data. Separate from security is the detection of errors due to transmission noise or equipment problems. While the methods for this are related to security methods, they are not sufficient in the presence of malicious participants. The primary basic technique for preventing unauthorized reading of data is encryption. The originator (person A) encrypts the data using a key to convert the original plaintext to ciphertext. The ciphertext is then stored or transmitted. The recipient or reader (person B) uses their key to decrypt the ciphertext to obtain the original information. An unauthorized person (person E) only sees the ciphertext, and upon this idea rests the secrecy of the message. There are two basic classes of encryption: symmetric encryption and asymmetric encryption. In symmetric encryption, the same key is used to encrypt and decrypt the message. For asymmetric encryption, a different key is used to encrypt a message from the key to decrypt the ciphertext. Symmetric encryption is also called private key encryption because the key must be kept secret. Asymmetric encryption is also called public key encryption because the encryption key can be made public and only the decryption key needs to be kept secure for the message to be secure. Where encryption hides information, hashing is used to detect data tampering. Specifically, changes in information (such as the terms of a contract) must be reliably detected. This involves creating a digest of the data that can be used to check for tampering. Calculating the digest for the tampered data can be compared to the digest of the original data to detect (but not identify) changed data. The common information protection techniques in today's Cloud environment and the Internet are Advanced Encryption Standard (AES) for symmetric encryption, RSA (Rivest–Shamir–Adleman) for asymmetric encryption, and SHA-2 (Secure Hash Algorithm 2) or SHA-3 (Secure Hash Algorithm 3) for information hashing.

While access control and information protection are required for preventing security breaches, some security attacks will occur. The detection of positional attacks and an appropriate response mechanism are required. For example, a person trying to login and repeatedly getting the password wrong is suspicious. Hence, many systems limit the number of failed attempts (detection of an attack) and then close the login access (response to a suspected attack). This is a straightforward approach for access control. However, most attacks are of the type denial of service (DOS). The goal is not about getting or changing information by accessing the resources, but about preventing others from utilizing the resources. Here a device or several devices repeatedly and rapidly attempt to perform a normally permitted activity at such a volume and rate with a view to prevent other entities from accessing the resources. A common example is many rapid Website inquiries resulting in the overload and crash of servers. The trade-off here is that techniques for the detection of malicious activity slow down the performance of legitimate normal activities.

The fundamental information security concepts will be described in more detail in later chapters, especially as they apply to Cloud computing. Information security traditionally identifies the security boundary, and that leads to identifying potential security attack scenarios. The systems are then designed to defend against those attacks. The first thing to note is that in many information security breaches, the problem was not the theoretical security of the system, but the implementation. A current trend away from passwords uses smartphones and/or biometric feature recognition for access control. The second thing to note is that with Cloud computing, the identification of the security boundary is difficult and always changing.

2.5 An Example of a Remote Security Attack

DNS is used to resolve IP addresses. It allows users to type a human-readable address, such as www.cnn.com for the Cable News Network (CNN) site, and translate it to an IP address. Attackers have found ways to manipulate the DNS records [9] located on a DNS.

Hackers can replace a legitimate IP address with a booby-trapped address and then carry out some malicious activities, such as harvesting users' login information. Furthermore, attackers can cover their tracks by substituting the site's security certificates. The Internet uses Transport Layer Security (TLS) and Secure Sockets

Layer (SSL) protocols to provide privacy and integrity [10]. A Certificate Authority issues TLS/SSL certificates. These are digital files, also called root certificates, that contain the keys, which are trusted by browsers. This attack happens in the following five steps:

1. The attacker sets up a fake site resembling the real target name.
2. Then somehow the attacker compromises the login credentials of a DNS provider server. The attacker changes the IP address of a targeted domain, such as a bank's Website, to the fake one. The attacker also generates a new valid TLS certificate for the malicious Website.
3. The victim inadvertently approaches the DNS provider looking for a real site.
4. Using the DNS record of the compromised site, the user is directed to the fake site.
5. The fake site asks for the victim's user credentials and records this information in the attacker's database. Later, it is harnessed to access the victim's records from the target site.

This process is illustrated in Fig. 2.8, which may lead unsuspecting users to think that they are on a legitimate Website and then enter their passwords, such as for a bank account. The attackers can later on use this information to steal money from the actual bank account.

The root cause of this attack is an improperly secured DNS server and ability to generate valid digital certificates for a new target site. Like other attacks that we shall study in this book, understanding the root cause is key to preventing these attacks.

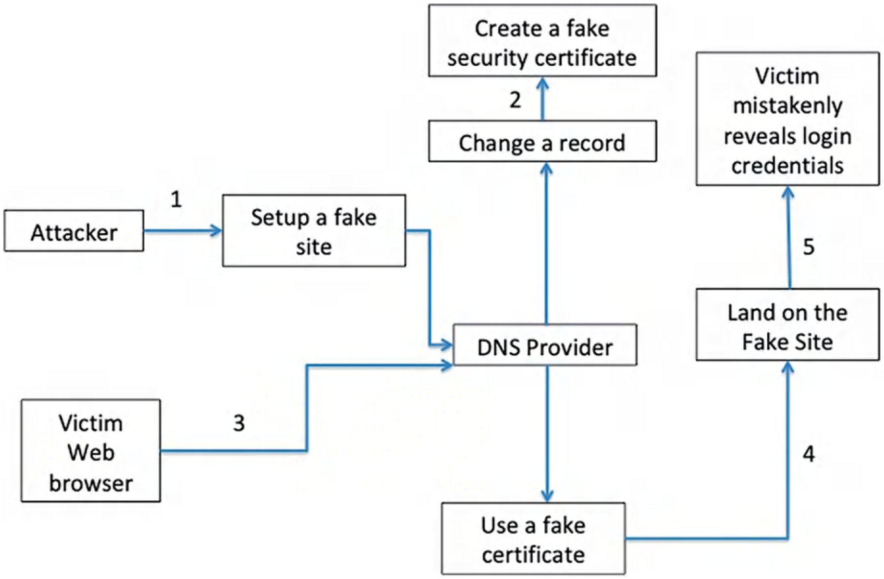


Fig. 2.8 Process of redirecting incoming IP traffic to a different Website

2.6 Edge Software Security Requirements

Traditionally, software must meet functional and performance requirements. However, security needs to focus on minimizing the attack surface and vulnerabilities. It is also required that even under an attack, software will perform as specified [11].

The US Department of Defense's Cyber Security and Information Analysis Center (CSIAC) [10] has specified that all software must meet the following three security needs:

1. *Dependability:* Software should be dependable under anticipated operating conditions and remain fairly dependable under hostile operating conditions.
2. *Trustworthy:* Software should be trustworthy in its own behavior and robust. That is its inability to be compromised by an attacker through exploitation of vulnerabilities or insertion of malicious code.
3. *Resilience:* Software should be resilient enough to recover quickly to full operational capability with a minimum of damage to itself, the resources and data it handles, and the external components with which it interacts.

This means that security needs should be considered during all phases of development, starting with architecture, implementation, validation, and deployment. From an Edge user's point of view, security is subtle, invisible, and almost taken for granted. However, Cloud developers and operators need to observe the following practices to assure security:

1. *Language options:* Start by considering strengths and weaknesses of available options, preferring a language with strong type checking and built-in security measures. For example, because C is a weakly typed high-level language, it is therefore unable to detect or prevent improper memory allocation, resulting in buffer overflows. So, a program will need to check for boundary limits. Whereas Java is a strongly typed high-level language, it has intrinsic security mechanisms based on trusted byte code interpretation, which prevent the use of uninitialized variables and language constructs to mitigate buffer overflows.
2. *Secure coding:* Adopt coding practices that eliminate incidents of buffer overflows, strings overwrites, and pointer manipulations, preferably by adding checks on the array sizes, string lengths, and pointer overrides, respectively.
3. *Data handling:* Separate sensitive or confidential data and identify methods to securely handle it, e.g., using encryption during storage and transmission and at run time.
4. *Input validation:* Add additional checks to ensure that the range and type of data entered by users are correct before passing it to the downstream APIs.
5. *Physical security:* All equipment connected to Cloud servers should have restricted physical access. System logs must be maintained and reviewed to trace back the root cause of attacks.

Overall, ensuring security is a matter of following the “trust but verify” approach, based on a quote by the late President Ronald Reagan, which is even truer for Cloud computing.

2.7 Rising Security Threats

Security is no longer just a theoretical concern but is costing our world’s economy more than half a trillion dollars per year, as shown in Table 2.1.

Each year, prominent breaches are happening and affecting hundreds of millions of people. Some of the largest impacts during the past decade are shown in Table 2.2.

Lastly, cybercrime is starting to drag down the gross domestic product (GDP) of many national economies, as shown in Fig. 2.9. Beyond these figures, there is a multiplier effect on the economy as productivity slows down. It is due to additional security checks that need to be implemented as preventive measures.

There is a continuous battle to outdo each other between the hackers and security professionals. In the subsequent chapters of this book, we will study methods that will help the latter.

2.8 Summary

Cloud computing emerged only a decade ago. It is based on basic technologies that have been around and under development for more than half a century. These technologies were hardened in other environments, such as defense applications, personal computing, mobile products, etc. Note that as we advance into a Web application domain, it also expands the hacking attack surface. Business models and relationships have to be completely reevaluated due to the fundamental technologies of Cloud computing. Threat agents can have multiple entry points on the client-side browsers, network in-between, and server-side hardware or software. Each risk needs to be evaluated and mitigation strategies developed in advance to prevent harm to the users of Web applications.

Table 2.1 Cost of security breaches [12]

Security incidents	Results
Cost of security breaches/year globally	$600 B
Insiders contributing to security incidents in 2017	46%
Security experts expect a major attack in the next 90 days	30%

Table 2.2 Biggest data breaches of this century, so far [13]

Year	Enterprise compromised	Number of people impacted
2018	Marriott	500 M
2017	Equifax	143 M
2016	Adult Friend Finder	412.2 M
2014	eBay	145 M
2013	Yahoo	3 B

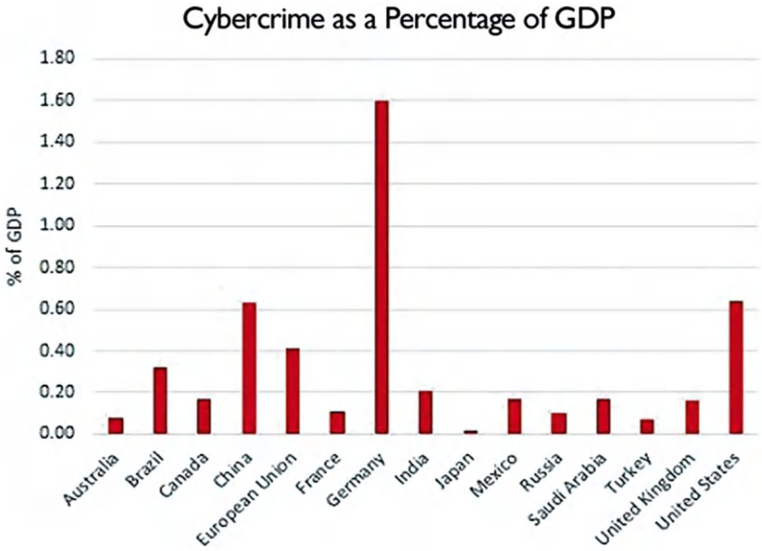


Fig. 2.9 Drag of cybercrime on national economies [14]

2.9 Points to Ponder

1. What led to the desire of people with PCs to connect with one another?
2. What led to the growth of thin clients? How thick should thick clients be, and how thin should thin clients be? Which use cases suit each category (e.g., an information panel at the airport vs. an enterprise handheld computer)? Or given a usage class which client they should use? Besides usage, software update frequency and security are also a consideration.
3. What's the minimal precaution an Edge Cloud user should take?
4. What are the trade-offs of securing information during transmission?
5. Why is the NIST cyber security framework applicable to Edge computing?
6. Multi-tenancy drives load variations in a public Cloud. Different users running different types of workloads on the same server can cause performance varia-

tions. How can you predict and minimize the undesirable effects of performance variability?

7. Does a private Cloud always provide a higher level of security than a public Cloud?

2.10 Answers

1. *What led to the desire of people with PCs to connect with one another?*

   - Professionals wanted to share data and programs with other users.
   - People wanted to share music with their friends.
   - Socially motivated sharing of content, e.g., pictures, stories, and news between family and friends, gave rise to social networks. This in turn gave rise to sites such as Facebook. This phenomenon brought many new consumers and devices to the Cloud connected networks and databases.

2. *What led to the growth of thin clients? How thick should thick clients be, and how thin should thin clients be? Which use cases suit each category (e.g., an information panel at the airport vs. an enterprise handheld computer)? Or given a usage class which client they should use? Besides usage, software update frequency and security are also a consideration.*

   - The previously mentioned trend of people wanting to stay connected at all times through various means, such as while traveling on vacation with a smartphone but not carrying a PC, meant that phones evolved to have a larger screen and mobile applications. However, phones have limited compute capability due to smaller form factors and users' desire to make the phone charge last a full day. This is the essence of a thin client. Another example of thin clients is an airport display terminal with little or no local processing or storage capability.
   - A laptop PC is a thick client due to its larger form factor, as compared to a smartphone. However, a laptop should be comfortable to carry around, have a reasonable battery life, and not get overly hot during its operation.
   - Thickness of a thin client depends on the usage model for each category (e.g., an information panel at the airport needs a larger display vs. an enterprise handheld computer in the hands of an Amazon parcel delivery driver needs more storage and a larger battery vs. a smartphone to fit in the vacation-going tourist's pocket). Besides usage, software update frequency and security requirements are also a consideration. Enterprises typically need encryption on their mobile handheld devices to protect confidential data, which needs more compute and battery power.

3. *What's the minimal precaution any Cloud user should take?*

    - A user needs to ensure that any secure Website connection starts with https://, wherein the last "s" means secure. Also, if a user arrives at a Website by clicking on a link through an email or text message, then the spellings need to be checked to ensure that it is the correct service provider's site instead of a phishing attack to steal financial information.

4. *What are the trade-offs of securing information during transmission?*

    - During transmission, any sensitive information, such as users' passwords, credit card information, or any sensitive data, needs to be protected. This information may be traveling over public nodes, which are susceptible to a man-in-the-middle attack. It can be protected with encryption. However, the process of encryption by the sender and decryption by the receiver requires time and compute resources. Hence, an appropriate trade-off must be done on the need and level of encryption requirements, e.g., 128 bits vs. 256 bits.

5. *Why is the NIST cyber security framework applicable to Edge computing?*

    - Even though users of Edge computing may be more worried about the security of their devices, the Cloud managers care about the security of their infrastructure as well as the security of their customers' data. This situation is akin to the operators of a public airline, who can't assume that all passengers have good intentions. Thus, security checking is done for anyone wishing to fly on a commercial plan. Similarly, the public Cloud managers are on a constant lookout to avert any threats in their datacenter, potentially coming from insiders, hackers, or their own customers.
    - Multi-tenancy drives load variations in a public Cloud. Different users running different types of workloads on the same server can cause performance variations. How can you predict and minimize the undesirable effects of performance variability?
    - Analytics can be used to predict periodic patterns of workloads, e.g., for enterprise users, weekdays may be busy, while for recreational users, weekdays may use more compute on the weekends. For example, Netflix customers watch more movies on the weekend using Amazon's data centers. Their jobs can be scheduled to run at complementary times to balance the overall server utilization in a Cloud.
    - Sometimes, in a multi-tenancy environment, execution of one task impacts the performance of other tasks. This is also known as a noisy neighbour problem. It exists due to shared resources on a server, such as frequent accesses to shared memory, disk drives, or network cards.
    - In a public Cloud, once a persistent, noisy neighbour presence is detected, it is beyond the scope of an individual user to avoid it. The reason is that one user has no control over another user's tasks. However, it can be avoided by stopping the task and requesting a different machine. This may interrupt the

service, so a better way is to start another server in parallel and migrate the task in a seamless manner, if possible.

6. *Does a private Cloud always provide a higher level of security than a public Cloud?*

    - It is a misconception that information on private Cloud servers and storage systems is always more secure than the information stored in public Clouds. Some private hospitals and energy companies have recently suffered well-publicized hacker attacks. Research from Verizon's 2021 report* suggests that insiders are responsible for around 22% of security breaches at private enterprises. Since public Clouds operate at a hyperscale, they hire many security professionals and deploy state-of-the-art tools that smaller or private organizations may not be able to afford. Also, public Clouds operators are cognizant of the fact that their revenues depend on reputation for security and reliability. Therefore, they operate with more conservative strategies and take extra care to minimize or eliminate any undesirable incidents.
    - *https://www.verizon.com/business/resources/reports/2021/2021-data-breach-investigations-report.pdf

References

1. https://searchdata center.techtarget.com/definition/JCL
2. https://www.britannica.com/technology/personal-computer
3. https://en.wikipedia.org/wiki/Dial-up_Internet_access
4. https://condor.depaul.edu/elliott/513/projects-archive/DS513Spring99/figment/CSCONC.HTM
5. https://www.techopedia.com/definition/288/Web-browser
6. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). *Cloud computing with security and scalability*. Springer.
7. Bhatt, P. C. P. (2019). *An introduction to operating systems: Concepts and practice (GNU/Linux and Windows)*. PHI Learning Pvt.
8. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf
9. https://arstechnica.com/information-technology/2019/01/a-dns-hijacking-wave-is-targeting-companies-at-an-almost-unprecedented-scale/
10. https://www.acunetix.com/blog/articles/tls-security-what-is-tls-ssl-part-1/
11. Krutz, R. L., & Vines, R. D. (2010). *Cloud security: A comprehensive guide to secure cloud computing*. Wiley.
12. https://www.vmware.com/radius/rising-costs-cybersecurity-breaches/
13. https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html.
14. https://www.financialsense.com/contributors/guild/cybercrime-s-global-costs

Chapter 3
Foundations of the Internet of Things (IoT)

3.1 Introduction

In today’s interconnected world, we take good connectivity as a given in every aspect of our lives, right from coffee machines and air-conditioner thermostats to smartwatches and cars suggesting alternate routes in case of traffic congestion. All this has been possible due to smart devices that have emerged and are connected to the Cloud all the time. These devices are connected using a technology called the Internet of Things (IoT). IoT has emerged as a transformative technology used in a wide variety of industries, right from healthcare and manufacturing to retail and smart homes.

The IoT technology is not very old and first appeared in 1999 [1] but was adopted very fast by the industry due to various other advancements that were happening in the supporting technologies/ecosystem, such as (1) faster connectivity and network capabilities, (2) improvement in Cloud computing, (3) availability of data analytics tools and processing capability of large datasets by computers, and (4) availability of low-cost devices and corresponding cost reduction in computing and storage.

Starting with the simple connectivity of a device with an on/off switch communicating information over the internet, the IoT has undergone a significant evolution over the period, resulting in an intricate smart ecosystem of interconnected devices. These systems have revolutionized all industries by increasing efficiency, cost savings, and improved customer service. IoT has also impacted the daily lives of everybody by making everyday tasks easier, safer, and more efficient.

In this chapter, we will cover the history of the evolution of IoT devices, starting with a humble beginning, present-day reality, and exciting future possibilities that will emerge as IoT gets integrated with artificial intelligence and machine learning. We will also look at the new and exciting applications and emerging use cases and why industries are adopting this technology so fast. We will get a glimpse of the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_3

limitations and risks that come along with the adoption of IoT technology in Sect. 3.6.

Let us start by looking at the evolution of the IoT as a technology before getting into other interesting aspects of the adoption of IoT.

3.2 Historic Evolution of IoT Devices

Machines have been communicating with each other since the time the telegraph was developed in the 1830s. From that time the communication between machines has evolved, and now the machines communicate information over the Internet. This progress has led to the development of the Internet of Things (IoT).

The history of IoT can be traced to the development of Advanced Research Projects Agency Network (ARPANET) as a public network in the early 1980s. With time, it evolved into the Internet of today. Table 3.1 gives a brief history of the evolution of IoT as a timeline of milestones [1, 2].

As per the available studies [4], it is estimated that there are 15.14 billion connected IoT devices as of 2023, and over 83% of the organizations have improved their efficiency by introducing IoT devices in their technical landscape. IoT is considered the main driver for the “Next Industrial Revolution.” This has been made possible due to advancements in technologies such as 5G networks, Edge computing, and machine learning. These technologies have allowed real-time data analysis and decision-making at the Edge devices while improving reliability and reducing latency significantly. These issues have been discussed in detail in Chap. 4 of this book.

The growth of IoT is not without its challenges. One of the major challenges concerning the wide use of IoT is Cyber Security. This issue has been dealt with in detail in Chap. 5 of this book.

3.3 IoT Computing Environment

A typical IoT installation is a complex system with connected devices at the heart of the system. There are various layers as well as components that make up this complex infrastructure, right from the sensors that collect the data from the devices/equipment (things) in the field in analog format to the applications that allow humans to interact with the devices/equipment in the field.

The IoT environment can be considered an integrated, mutually reinforcing structure consisting of four layers that are designed to carry value-laden data from the various networked “things” to traditional IT systems [5]. For example, energy companies use connected sensors to measure vibrations in turbines, which are fed through the network to the computing systems of the companies that analyze it to predict when machines will need maintenance. Similarly, jet engine manufacturers

Table 3.1 Evolution of the Internet of Things

1982	A graduate student with the help of two fellow students and a research engineer in Carnegie Mellon University's Computer Science Department developed a code to monitor a Coca-Cola vending machine connected through ARPANET to let anybody on the network know if the machine has cold soda bottles.
1989	Tim Berners Lee, an English computer scientist, proposes the framework of the World Wide Web and lays the foundation of the Internet.
1990	John Romkey at the Massachusetts Institute of Technology (MIT) invents a toaster that can be turned on or off via the Internet. It can be considered the first connected "thing" in the Internet of Things.
1993	The two researchers, Quentin Stafford-Fraser and Paul Jardetzky, at the University of Cambridge developed a room coffee pot in their lab to enable people to check the level of coffee.
1999	Kevin Ashton, cofounder and Executive Director of Auto-ID Labs at MIT, coined the term Internet of Things (IoT) in a presentation to Proctor & Gamble to describe the system where the Internet is connected using RFID sensors in their supply chain.
2003	The term "IoT" started making its public appearance and was used widely in mainstream publications like The Guardian and Scientific American.
2005	The United Nations International Telecommunications Union published its 7th Edition of the Internet Report [3] titled Internet of Things, acknowledging the impact of IoT in its report.
2008	The first IoT conference is held in Zurich, bringing together researchers and practitioners from academia and industry to take part in the sharing of knowledge. The US National Intelligence Council recognized IoT as one of the six disruptive civil technologies.
2011	The Cisco Internet Business Solutions Group (CIBSG) presented in the White Paper that IoT gained popularity between 2008 and 2009 when the number of things connected to the internet exceeded the number of people connected to it.
2011	Internet Protocol Version 6 (IPv6) public launch: The new protocol allows (2^{128}) addresses.
2013 onwards	As companies like Apple and Samsung make waves with their smartphones, there is a proliferation of AI-powered personal assistants like Google Home and Amazon Alexa. Some devices control individual things in our homes, all working in concert with our computers and phones to share data and interact.

embed sensors that measure temperature, pressure, and other conditions of the engines to predict their failures or to carry out scheduled maintenance.

The four layers of a typical IoT system are the Sensing Layer or Physical Layer, Network Layer, Processing or Middleware Layer, and Application Layer [6]. Depending on the usage and installation of the IoT system, these layers may be implemented using different elements. In this section and the following section, we will discuss how these layers are implemented right from simple to complex systems.

3.3.1 IoT Computing Architecture

As explained earlier, a typical architecture of IoT is a layered structure with many *Devices* (things) connected at the end that generate the raw data, which is then communicated through a network using some form of *Gateway* to the backend server (typically a Cloud server) that is running the *IoT Platform* that helps in integrating the IoT information into the existing enterprise database so it can be used by the *Applications*.

The roles of the devices, gateways, and Cloud platforms are well defined [7], and each one of them provides specific features and functionality required by the IoT system. The interaction between each of the layers is depicted in Fig. 3.1.

Let us discuss the function of each of the layers in the architecture in some detail.

3.3.1.1 Physical Layer

This is the outermost layer, or sensing layer, which is the basis of an IoT system. In this layer of physical devices, both sensors and actuators are connected. The sensors gather raw data from their environment, and actuators take required actions as may be decided by the higher levels within the architecture.

The sensors can be anything from temperature sensors, surveillance cameras, and security cameras in a smart home system to heart rate monitors in wearable technology. In the case of industrial scenarios, they may cover everything from legacy industrial devices to robotic camera systems, water-level detectors, air quality sensors, and accelerometers. For example, deploying a sensor on an automotive assembly line can be used to assess quality control through robotic functions with its output relayed to higher layers for processing purposes.

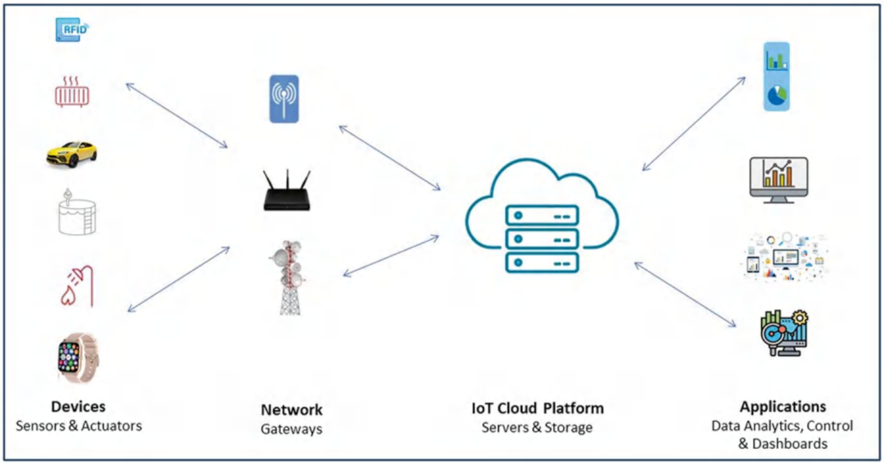


Fig. 3.1 A typical IoT system architecture

An actuator might, for example, shut off a power supply, adjust an airflow valve, or move a robotic gripper in an assembly process as may be directed by the system.

3.3.1.2 Network Layer

The network layer is responsible for providing communication and connectivity between devices in the IoT system. It includes protocols such as Hypertext Transfer Protocol (HTTP), Message Queuing Telemetry Transport (MQTT) [8, 9], and Advanced Message Queuing Protocol (AMQP) to facilitate transmission from one application/device to another and with the internet. The network technologies that are commonly used in IoT include Wi-Fi, Bluetooth [10], Zigbee [11], and cellular networks such as 4G and 5G.

The network layer may include gateways and routers that act as intermediaries between devices and the Internet and may include security features such as encryption and authentication to protect against unauthorized access.

These devices are often located near the sensors and actuators for the reasons of reliable communication and energy conservation. For example, a pump might contain numerous sensors and actuators that feed data into a data aggregation device that also digitizes the data. This device might be physically attached to the pump and will communicate the digitized data to the adjacent gateway for further communication.

The basic gateway capabilities can be enhanced by adding capabilities of analytics, malware protection, and data management to make them intelligent gateways. These gateways will be able to analyze the data streams in real-time.

3.3.1.3 IoT Cloud Platform

The IoT Cloud platform, or the data processing layer of IoT architecture, refers to the software and hardware components that are responsible for collecting, analyzing, and interpreting data from IoT devices.

The layer includes a variety of technologies and tools, such as data management systems, analytics platforms, and machine learning algorithms. These tools are used to extract meaningful insights from the raw data received from devices to enable businesses to make decisions and streamline their operations. We can also consider Edge analytics along with AI techniques supported by intelligent gateways to form part of this layer.

The processing may take place on-premises, in the Cloud, or in a hybrid Cloud system, but the type of processing executed remains the same, regardless of the platform.

3.3.1.4 Application Layer

The application layer of IoT architecture is the topmost layer that interacts directly with the end user. It is responsible for providing user-friendly interfaces and functionalities that enable users to access and control IoT devices. The layer includes various software and applications such as mobile apps, Web portals, machine learning algorithms, data visualization tools, and other advanced analytics capabilities that are designed to enable humans to interact with the underlying IoT infrastructure.

For example, when someone uses an application specifically designed for smart homes, he will be able to activate coffee makers simply via the app's button-tapping function, or when there is unusual energy consumption, it may indicate faults or inefficiencies. Similarly, in the case of industrial applications, if vibration levels in machinery go beyond normal limits, maintenance can be scheduled before costly breakdowns occur.

3.3.2 Development Frameworks

To enable the development of IoT solutions, there is a need to have software running on various hardware devices at each layer of IoT architecture [12]. We will be looking at the three stacks of software that will be running on the hardware of these three layers to enable a secure and seamless flow of data from sensors to the backend servers of the IoT systems.

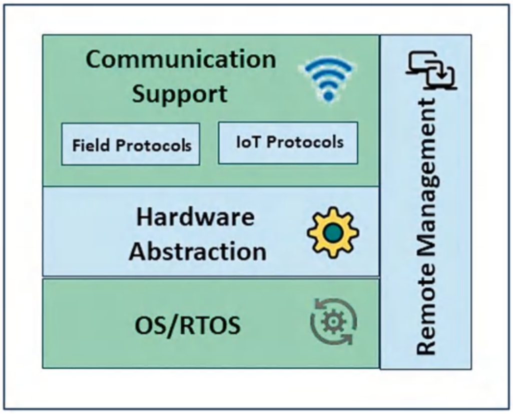
3.3.2.1 Stack for Physical Layer Devices

The "Things" as the sensors and actuators are called in IoT parlance and are the starting point of the IoT solutions. These devices are generally constrained in terms of size or power supply and often use microcontrollers that have very limited capabilities. These microcontrollers are designed to carry out specific task(s) with the aim of mass production and low cost.

The software for these microcontroller devices is also designed to support specific tasks. The software stack is depicted in Fig. 3.2. The key features of the software stack running on a device include:

1. *IoT Operating System:* Most of the devices will run with "bare metal," but some will have embedded or real-time operating systems that are particularly suited for microcontrollers and can provide IoT-specific capabilities.
2. *Hardware Abstraction:* It is a software layer that enables access to the hardware features of the microcontroller, such as flash memory, GPIOs, and serial interfaces.

Fig. 3.2 Software Stack for physical layer devices



3. *Communication Support*: This includes drivers and protocols enabling the devices to connect using a wired or wireless protocol like Bluetooth, MQTT, CoAP, etc., thereby enabling device communication.
4. *Remote Management*: This functionality enables the device to remotely control the upgrade of its firmware or to monitor its battery level.

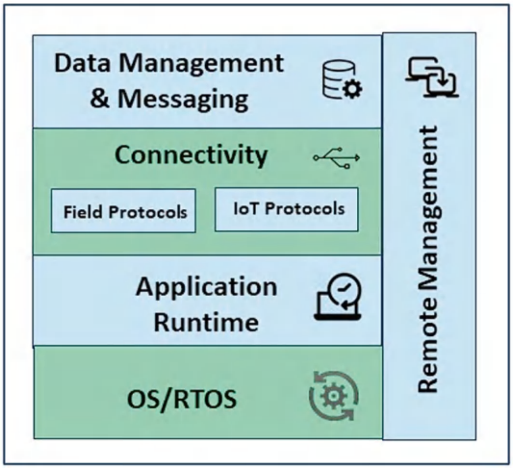
3.3.2.2 Stack for Network Layer or Gateways

The gateways act as the aggregator for a group of sensors and actuators and coordinate the connectivity of the connected devices to each other and an external network. Typically, a gateway is physical hardware, but sometimes the functionality may be incorporated into a larger "thing" that is connected to the network. For example, an automobile or a home automation appliance [13] may act as a gateway.

An IoT gateway may often have the capability of processing the data at the Edge and storage capabilities to overcome network latency and provide reliability. The IoT gateways are generally dependent on software to implement the core functionalities. The software stack for gateway is depicted in Fig. 3.3. The key features of the gateway software stack are:

1. *Operating System*: It is typically a general-purpose operating system having a small footprint, such as Linux.
2. *Application Runtime Environment*: IoT gateways often can run small application code (applets), and allow the applications to be dynamically updated. For example, a gateway may have support to run Java, Python, or Node.js code.
3. *Communication and Connectivity*: The gateways need to support different connectivity protocols (e.g., Bluetooth, Wi-Fi, Zigbee) to allow different devices to connect with them. They also need to connect to different types of networks (e.g., Ethernet, Wi-Fi, cellular, etc.). While providing connectivity using different protocols, the gateways need to ensure the reliability, security, and confidentiality of the communications.

Fig. 3.3 Software Stack for network layer gateways



4. *Data Management and Messaging*: This capability is needed to provide local persistence to overcome network latency, offline mode, and real-time analytics at the Edge. It also provides the ability to forward device data consistently to an IoT platform.
5. *Remote Management*: It enables the ability to remotely provision, configure, and start/shut down the gateway as well as the applications running on it.

3.3.2.3 Stack for IoT Cloud Platform

The IoT Cloud Platform provides the software and services required to enable IoT solutions to perform and deliver their functionality. It typically operates on a Cloud infrastructure or inside an enterprise data center. The platform should provide flexibility to scale both horizontally (e.g., to support a large number of devices connected) as well as vertically (e.g., to address the variety of IoT solutions). It also enables the interoperability of the IoT solution with existing enterprise applications and other IoT solutions.

The core features of the IoT platform are depicted in Fig. 3.4 and are described below:

1. *Connectivity and Message Routing*: The IoT platform should be able to interact with very large numbers of devices and gateways using different protocols and data formats. It should be able to normalize to allow for easy integration into the rest of the enterprise.
2. *Device Management and Device Registry*: A Central Registry to identify the devices/gateways running in an IoT solution and the ability to provision new software updates and manage the devices. It should also be capable of registering new devices and gateways as the network scales.

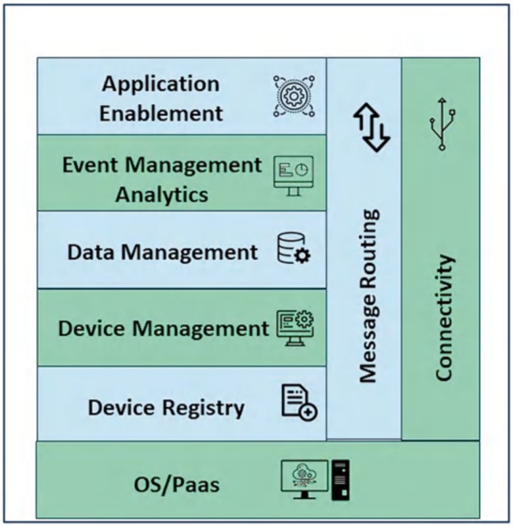


Fig. 3.4 Software Stack for IoT Cloud platform

3. *Data Management and Storage*: A scalable data store that supports the volume and variety of IoT data.
4. *Event Management and Analytics*: A scalable event processing capability with the ability to consolidate and analyze data; shall also have the ability to create reports, graphs, and dashboards as may be required by the user.
5. *Application Enablement*: Ability to create reports, graphs, and dashboards to use API for application integration.

In addition to the above functionalities that need to be provided at different layers, there is a need to have cross-Stack functionality to ensure secure and seamless operations of the IoT system. The required functionalities are:

1. *Security*: Security needs to be implemented across the network, right from the devices to the Cloud. Features such as authentication, encryption, and authorization need to be part of each stack.
2. *Ontologies*: The format and description of device data are required to enable data analytics and data interoperability. The ability to define ontologies and metadata across heterogeneous domains is a key area for IoT system interoperability.
3. *Development Tools and SDKs*: IoT developers will require development tools that support the different hardware and software platforms involved.

There are many open-source and proprietary solutions available for the same [14]. Some of the popular open-source frameworks are DeviceHive, Mainflux, Thinger.io, and Kaa Enterprise IoT Platform, to name a few [15]. The source code can be obtained on GitHub [16]. All major Cloud service providers (CSPs) have a proprietary IoT solution. Some of the popular frameworks are Amazon Web Services IoT, Azure IoT, Cisco IoT Solutions, and Google Cloud IoT.

3.4 Intelligent IoT Devices at the Edge

In the last section, we looked at a simplified version of the IoT architecture. The architecture had several limitations, as discussed. The major issues with the simplified architecture are the latency and security concerns, especially when a system is deployed for critical or sensitive applications.

As depicted in Fig. 3.1 and explained in the earlier section, the processing is centralized on the Cloud platform. This results in latency, as the data must be transported to the Cloud, and after processing, the results are transferred back to the Edge device to carry out the actions.

To overcome this drawback, many approaches have been suggested. The two most popular approaches are Edge computing and Fog computing [17]. A lot of people use these terms interchangeably as both are concerned with leveraging the computing capabilities within a local network. These capabilities are used to carry out computation tasks that would have been ordinarily done in the Cloud. In other words, both Edge computing and Fog computing are computing methods that bring computing and data processing closer to the site where data is initially generated and collected. As both methods process data closer to the source, they reduce latency and conserve IoT network resources—crucial for timely insights.

Before we proceed further in terms of the requirements of intelligent devices, it is crucial to understand the differences that set apart these two methods.

3.4.1 Edge Computing

Edge computing [18] [19], as the name implies, brings the computation to the Edge of a network, i.e., at the source where data is produced. The storage and computation powers are embedded within the devices to collect and process sensor-generated data. This decentralized approach to data processing and storage improves response time by drastically cutting the latency associated with transmitting large volumes of data to the Cloud server. This enhances the Edge device's overall functionality and optimizes its performance.

This distributed model plays a pivotal role in IoT systems requiring real-time reactions. Some of the examples where Edge computing becomes a necessity due to response time criticality are (A) Can the robotic arm performing a surgery cut an artery? (B) Will the car crash? (C) Is the aircraft approaching the threat detection system a friend or a foe? Under such situations, there is no time to send the data to the Cloud platform; the decision processing needs to be done on the device itself.

However, Edge computing devices currently do not have the computing and storage that are necessary to perform advanced analytics. Due to this limitation, the processed data is forwarded to Cloud servers for analysis, review, and archival.

3.4.2 Fog Computing

Fog computing [20], as the name suggests, is the computing layer between the Cloud and the Edge. It is typically nearer to the Edge, and that is how it derives its name (the word "fog" in Fog computing is a metaphor, as fog in general terms refers to the clouds close to the ground). The Fog computing resources are typically located in the same or nearby premises where the sensors are located. They are connected to the Edge devices on a Local Area Network.

In cases where Edge devices may send huge streams to the Cloud, Fog computing can receive its data from the Edge layer before it reaches the Cloud and then decide what is relevant and what is not. The Fog layer may forward the relevant data to the Cloud, while the irrelevant data can either be deleted or analyzed at the Fog layer for remote access or to inform localized learning models. In some cases, the analyzed data may also be forwarded to the Cloud layer for storage and archival purposes.

Figure 3.5 captures the relationship among the various storage and processing layers, namely, Edge computing, Fog computing, and Cloud computing, in a more complex IoT environment [21].

A real-life example of Fog computing would be an embedded application on a production line, where a temperature sensor would measure the temperature every single second. This data would then be forwarded to the Fog server over a local area network. The Fog server will then decide whether the data should be forwarded to the Cloud or not, based on the local model or certain other parameters. For simple temperature readings, these data savings might seem negligible, but imagine the volume of all the temperature measurements; every single second of a 24/7

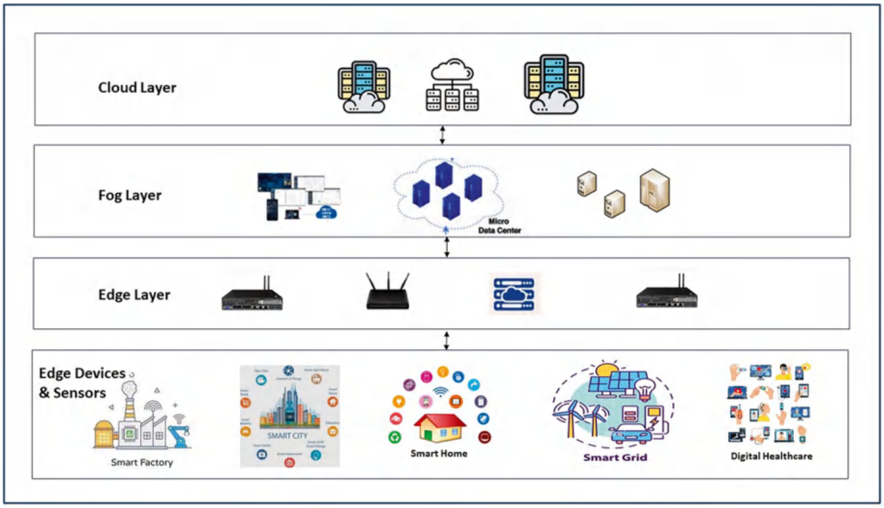


Fig. 3.5 Enhanced IoT environment with Edge computing and Fog computing layers

measurement cycle is sent to the Cloud. Consider the case of the data being generated by security cameras, which are constantly streaming complex information or large files such as images or videos. The impact on network bandwidth and latency could be massive.

3.4.3 Edge Computing vs. Fog Computing

As may be clear from the description above, Edge and Fog computing have many commonalities, but they are still very different computing methods. Let us discuss the similarities and differences between the two models.

3.4.3.1 Similarities Between Edge and Fog Computing

Both Edge and Fog computing are viable solutions to handle tremendous amounts of data generated by the IoT devices at the Edge.

Both technologies have been designed to keep data closer to where it originated and perform computations that otherwise would need to be done in the Cloud. This approach offers better bandwidth efficiency, resulting in minimal expenses. This also results in reducing the latency.

Both approaches offer increased security and privacy by encrypting data by using local computing power. They can also identify potential cyberattacks, thus enabling them to respond with security measures quickly.

Each of these computing methods is designed to support autonomous operations, even in locations where connectivity is intermittent or bandwidth is limited. These two technologies can process data locally.

3.4.3.2 Differences Between Edge and Fog Computing

The significant difference between Edge and Fog computing is where computation and data analysis occur [22]. In the case of Edge computing, it takes place right on the devices attached to the sensors. In some cases, processing may occur on a gateway device closer to sensors. In the case of Fog computing, it takes place on devices or servers that are further away from the sensors that generate data.

Edge computing allows data to be analyzed and acted upon in real time, thereby optimizing the performance of the system. The data is also more secure as it is not transported [23]. In some use cases, Edge computing can send results directly to the Cloud. Therefore, Edge computing can be done without the presence of Fog computing.

Fog computing cannot exist without Edge computing because it cannot produce data on its own. Its main objective is to reduce the workload at both the Edge and the Cloud. This is done by performing the necessary processing tasks.

Edge computing is typically used for minor resource-intensive applications because devices have limited capabilities in terms of data storage and processing. It has been found useful in healthcare applications in the form of patient monitoring [24], predictive maintenance, and large-scale multiplayer gaming. On the other hand, Fog computing is primarily used for applications that process large volumes of data gathered across a network of devices. It is useful for use cases such as smart grids [25] and autonomous vehicles.

3.4.3.3 Comparison of Edge and Fog Computing

The above discussion and comparison [26] of the two approaches have been summarized in Table 3.2. in terms of their advantages, disadvantages, and typical use cases.

The main advantages of both these computing methods are improved user experience and systematic data transfer with minimal latency. Both methods apply to multiple problems. The decision as to which one to choose for a specific problem depends on the cost, response time required, and management complexity.

3.4.4 Additional Requirements

In addition to the storage, computing, and analytics capabilities, intelligent devices also have additional requirements to ensure a secure and better user experience.

3.4.4.1 Security

With the expansion of IoT systems, security has become an important requirement to ensure data is safe as it moves from the Edge layer to the Cloud layer [26]. In addition to data security, there is a need to keep devices and connections also secure. This requires that any deployment shall have a security layer to provide encryption for reliable data transmission, authentication services for user verification purposes, as well as access control elements that can be used to restrict certain resources if required.

Integrating a sophisticated security system helps protect against any potential threats that might breach through vulnerabilities within the entire IoT network architecture. The security layer provisions and enforces multiple levels of security protocols at various stages to deter malicious attacks against devices and connectivity. The access control mapping guards the protection of data collected and shared by a network participant or user. More details about the security features required for reliable communication and processing are discussed in Chap. 5. The best practices to be followed to build a reliable security layer are discussed in Chap. 7 of this book.

Table 3.2 Comparison of Edge and Fog Computing

S. No.	Edge computing	Fog computing
Advantages		
1	Provides flexible remote connectivity	Connectivity flexibility in terms of wired, Wi-Fi, or high-speed 5G networks
2	Data sovereignty compliance by keeping data close to its source	Reduced Cloud dependency resulting in cost savings by minimizing data transfer costs and Cloud service consumption
3	Enhanced security by providing data encryption before transmission	Reduced bandwidth consumption resulting in bandwidth efficiency and latency improvement
4	Local intelligence enables the device to host AI and machine learning models locally, enabling them to make intelligent decisions	Distributed intelligence across the network, enabling decision-making at various levels
5	Offline operation even when disconnected from the Cloud	Local data processing helps filter and preprocess data before sending it to the Cloud
Disadvantages		
1	Edge computing disperses computing resources, making centralized monitoring and management more complex	As the computing resources are restricted to a physical location, it restricts computing to that specific location
2	Adding more devices to the Edge requires careful management and configuration to maintain consistent performance and reliability	It is exposed to potential security threats like IP spoofing and Man in the middle attacks
3	Edge devices depend on network connectivity for communication and data transfer and, therefore, require reliable network connectivity under normal operations	Implementing Fog solutions requires integration with both Edge and Cloud systems. This requires additional finances at the setup time
4	Deployment of Edge networks involves setting up and configuring Edge devices that require specialized skills and expertise	
Use Cases		
1	In industrial settings, Edge computing is extensively used for real-time monitoring and control of machinery, optimizing production processes, and enabling predictive maintenance to prevent costly downtime	For smart cities and urban infrastructure Management, typically Fog computing is deployed to enhance the functioning of applications such as traffic management, street lighting, waste management, and public safety
2	It is crucial for self-driving cars and vehicles, where immediate decision-making based on sensor data is essential to ensure safe navigation and collision avoidance	Fog computing plays a pivotal role in energy management systems by analyzing data from smart meters, sensors, and power grids. It aids in load balancing, fault detection, and energy consumption optimization

(continued)

Table 3.2 (continued)

S. No.	Edge computing	Fog computing
3	For smart home devices like thermostats, security cameras, and voice assistants, Edge computing enables local processing for quicker response times and enhanced privacy	For oil and gas exploration in remote locations such as oil rigs, Fog computing can process data from various sensors to monitor equipment performance, detect anomalies, and ensure the safety of workers
4	Edge computing facilitates local AI processing for applications like image recognition, natural language processing, and machine learning on devices like smartphones and cameras	In agriculture, Fog computing assists in monitoring soil conditions, weather forecasts, and crop health using sensors. This enables precise irrigation, fertilization, and pest management for improved yield and resource efficiency

3.4.4.2 Business Layer

To derive the business advantage and provide a better user experience, there is a need to have a business layer that serves as a bridge between IoT data and existing operations. The business layer helps in better decision-making and strengthening collaboration. It also simplifies application complexity by automating procedures through rule enforcement. By confirming data validity, it guarantees protection from breaches and maintains its robustness.

3.5 Use Cases for IoT Devices

This section discusses three use cases where IoT devices are connected to AI/ML applications running on an IoT-enabled Cloud platform. The use cases cover applications right from smart home to remote monitoring of solar power plants.

3.5.1 Smart Home Systems

A smart home is any home that includes automated IoT devices connected to the Internet [13]. Using these IoT devices, users can control lighting, heating, and other home appliances (such as washing machines, dishwashers, and dryers) to make life simple. Smart home applications give automatic control of things around the home, turning them from “dumb” to smart.

Smart homes make life more convenient and can even save money on heating, cooling, and electricity bills. They also lead to greater safety with Internet of Things devices like security cameras and systems. These devices are connected to the Internet, which allows them to be controlled remotely. For example, one can put lights on schedules so that they turn off during the night at sleep time or turn the A/C up about an hour before one returns from the office so the house is comfortable. It can also monitor the air quality and open or close windows as required to ensure the house has a constant supply of fresh and filtered air.

Typically, smart home applications are linked to a voice assistant like Alexa, Siri, or OK Google. These voice assistants can be used to control lights, adjust room temperature, answer phone calls, check who is at the front door, or trigger an entire sequence of events from a single command [27].

A smart home system has three main elements: sensors, controllers, and actuators.

- Sensors can monitor changes in daylight, temperature, or motion detection. Home automation systems can then adjust sensor settings (and more) based on user preferences.
- Controllers refer to devices like personal computers, tablets, or smartphones. These devices are used to send and receive messages about the status of automated features.
- Actuators may be light switches, motors, or motorized valves that control the actual mechanism, or function, of a home automation system. They are programmed to be activated by a remote command from a controller.

The IoT devices are connected to an Internet gateway using any of the different communication protocols, i.e., Wi-Fi, Bluetooth, ZigBee, or others. Many of these IoT devices have sensors that monitor changes in motion, temperature, and light so the user can gain information about the device's surroundings. The gateways are connected to an IoT platform hosted on a Cloud server.

Smart homes work at three levels:

- *Monitoring*: It allows a user to check in on remote devices through an application. For example, one can view the live feed from a smart security camera.
- *Control*: It allows the user to control these devices remotely, like panning a security camera to see more of a living space.
- *Automation*: It allows setting up devices to trigger one another, like having a smart siren go off whenever an armed security camera detects motion and informs the security services of a breach.

A typical smart home system topology is shown in Fig. 3.6.

Typically, smart home systems offer a variety of functions. Some of the common functions offered by these systems are:

- Alarm Systems
- Appliance Control
- Digital Personal Assistant
- Fire and Carbon Monoxide Monitoring
- Home Automation Security Systems and Cameras
- Keyless Entry
- Real-time Text and Email Alerts
- Remote Lighting Control
- Surveillance Systems
- Thermostat Control

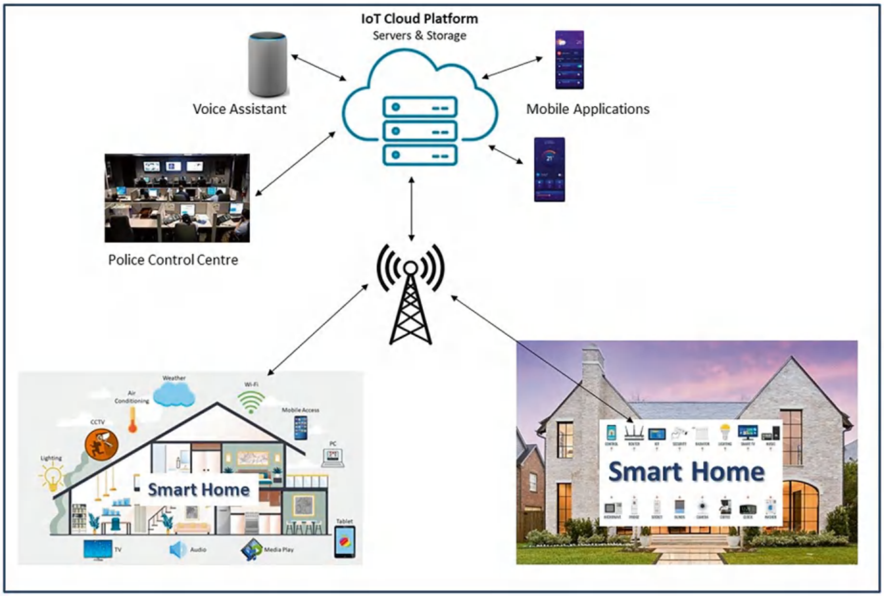


Fig. 3.6 Smart home systems

- Voice-activated Control

  Smart home systems offer many benefits. Some of them are:

- *Remote Access*: The devices at home can be controlled using mobile applications from anywhere, even if it is a remote location thousands of miles away from home.
- *Comfort and Convenience*: The system allows you to control devices remotely or via voice commands, set them on schedules, and even sync them with the sunrise and sunset. The systems are user-friendly to allow anybody to operate them without any knowledge of underlying technologies.
- *Energy Efficiency*: Due to the programming capabilities of smart devices like thermostats and motion detection, the system reduces energy consumption, thereby reducing electricity bills.
- *Safety*: Many smart security products increase safety at home. For example, sensors for doors and windows, security cameras that can detect people, and video doorbells that greet whoever is knocking on the door.

There are certain issues when using these smart devices, as listed below. These major concerns relate to cost, security, and privacy.

- *Cost*: Smart IoT devices are more expensive than their non-connected counterparts.
- *Security Issues*: Anything that is connected to the Internet can be hacked, including smart IoT devices. There is a need to be aware of these concerns and adhere to best digital security practices as described in Chap. 7 of this book.

- *Privacy*: Privacy is a huge concern when dealing with smart cameras, as users can livestream footage from a camera's application, which hackers can intercept.

3.5.2 Remote Monitoring and Management for Solar Plants

The solar power plant [28] consists of a sophisticated infrastructure that works in a manner that maximizes electricity production. The primary components of a solar power plant are solar panels, inverters, solar controllers, transmission, and energy storage systems. There is a need to closely monitor the performance of solar power plants to ensure maximum productivity and availability. In addition to monitoring the performance of solar plants, there is a need to monitor other parameters as well to ensure smooth running. The monitoring helps in scheduling plant maintenance, parts replacement, or solar panel cleaning to ensure the long-term performance of the power plant.

As solar plants are generally located in remote locations due to space requirements, remote monitoring of the plant is the only viable option available. The remote monitoring system keeps users well-informed about every minute detail of the solar power system. It also keeps details of the overall health of the solar system by keeping track of potential defects and issues that may arise [29].

The solar plant monitoring and management system is a collection of hardware and software. They work together to monitor faulty solar panels, and dust accumulates on panels, thus lowering the output, connection loss, and many such issues. The system gathers data from various sensors, analyzes, and synthesizes it to make decisions concerning solar energy systems.

The system integrates with various sensors like current sensors, voltage sensors, energy meter sensors, and weather sensors. These sensors are nothing but smart IoT devices. They also monitor the inverter performance, which converts the direct current (DC) generated by solar panels into usable alternating current (AC). The data generated from these IoT devices is analyzed in the Cloud, giving detailed and intricate performance feedback on the solar energy system and offering real-time data on energy generation, solar panel efficiency, and overall system performance.

The topology used for the system is a simplified classic remote monitoring solution with a cellular connectivity device at its core for connectivity to the Cloud. An example of system topology is shown in Fig. 3.7.

The remote monitoring and management system based on smart IoT devices provides multiple benefits:

- *Downtime Reduction*: The timely detection of issues enables prompt maintenance, minimizes downtime, and maximizes energy production, resulting in higher system efficiency.
- *Repair and Replacement Cost Reduction*: The early identification of defects or anomalies prevents minor issues from snowballing into major problems. This results in reduced repair and replacement costs.

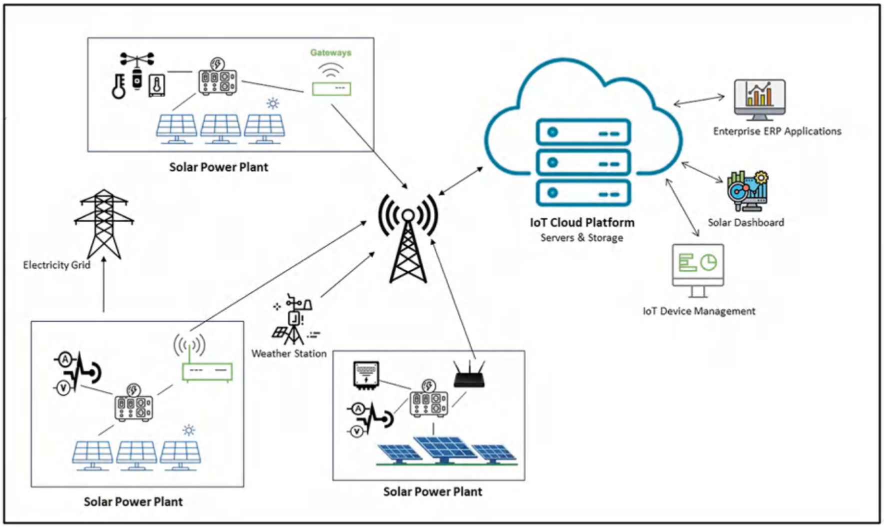


Fig. 3.7 Remote monitoring and management of solar plants

- *Managing Multiple Installations*: The control system can manage multiple solar energy sites from a centralized location. This scalability multiplies operational efficiency and streamlines maintenance procedures.
- *Improving Plant Efficiency and Performance*: The insights provided by remote monitoring systems empower users to refine system parameters, address inefficiencies, and optimize energy output, thereby elevating overall plant performance.

3.5.3 Smart Helmet

In case of a road accident involving a two-wheeler rider, medical help must be provided to the injured at the earliest to save their life. This is a perfect use case for an IoT-enabled helmet with an AI-enabled application to provide immediate medical help.

The smart helmet is a collection of sensors, hardware, and software [30]. These work together to constantly monitor the driving pattern and behavior of the rider. The system integrates with various sensors like an accelerometer [31], a gyroscope, LDR, IR, and barometric pressure and altitude sensors [32]. The data generated from these IoT devices is analyzed in the Cloud, giving detailed and intricate performance feedback to the rider about the speed and behavior pattern on the road. The analytics are performed by an AI engine in the Cloud.

In the event of an accident, the device detects the occurrence and severity of the accident using an AI model trained to detect the accident based on the data from various onboard sensors. The location of the accident is detected using the GPS-GSM satellite system [33]. The location is also communicated to the Cloud along with other data from smart helmet.

Based on the location data, the application running in the Cloud alerts the nearest emergency response team (hospital/ambulance) using SMS messages along with the system communication channel. SMS messages are also sent to the mobile phones of friends and relatives of accident victims to inform them about the accident. This helps in providing medical help at the earliest to the injured to minimize the risk of casualty.

The complete system consists of the IoT-enabled helmet, the IoT Cloud platform, the communication channel between the device and the Cloud, and the user interface, which may be based on mobile and laptop/desktop. The topology of such a system is shown in Fig. 3.8.

The advantage of such a system is that it is self-activated and informs the stakeholders (hospital, friends, and relatives) about the accident, its severity, time, and location, thereby reducing response time to provide medical help. It also gives confidence to the rider that he is not alone but is always traveling with a virtual road safety companion who can act on his behalf.

3.6 Current Limitations of IoT Devices

IoT devices are being installed in homes and businesses for many different applications. These devices have their limitations in terms of computational power, privacy, and security. The security risks associated with IoT devices are discussed in more

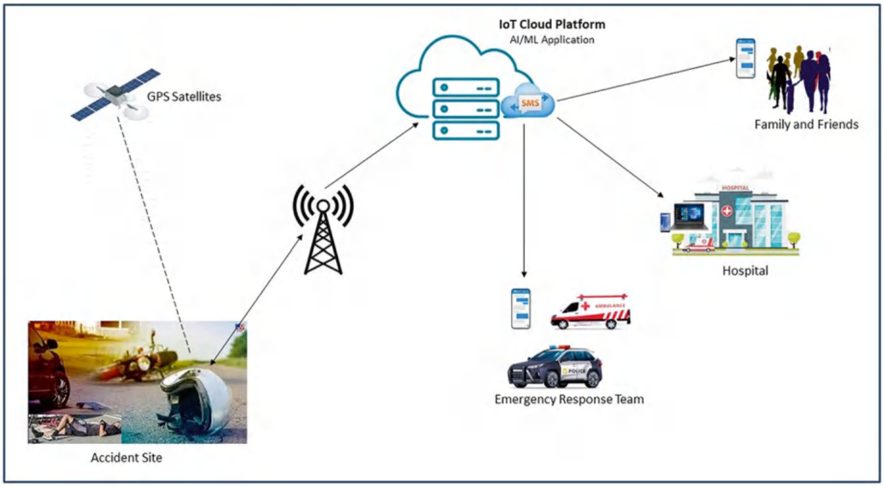


Fig. 3.8 Smart helmet for safety of two-wheeler riders

detail in Chap. 5 of this book. The other limitations associated with IoT devices and systems are listed below.

- *Limited Computation Capability*: IoT devices at the Edge typically have limited computing power as they are mainly concerned with converting analog data generated by sensors to digital format. Digital data is then sent to the Cloud server for further processing.
- *Limited Storage Capacity*: IoT devices at the Edge are often not designed to store large datasets, as these devices have limited storage capacity.
- *Privacy Concerns*: IoT devices are more vulnerable to cyberattacks due to limited storage and computational capability. These attacks can compromise the security and privacy of the data being collected and transmitted. The privacy issue is especially critical if the system leads to financial losses or reputational damage.
- *Security Risks*: With many interconnected IoT devices at the Edge and to the Cloud servers, these devices are more vulnerable to cyberattacks. These attacks can be in the form of hacking or unauthorized access leading to data breaches. Different kinds of cyberattacks are described in more detail in Chap. 5 of this book. Due to limited resources at the Edge, IoT devices often lack robust security mechanisms such as encryption, authentication protocols, and regular software updates.
- *Complexity and Integration*: Implementing IoT Edge devices is a complex task as it involves the integration of a wide range of technologies, including sensors, actuators, communication networks, and data analytics. This complexity is especially more challenging for businesses with existing infrastructure and legacy systems.
- *Limited Interoperability*: At present, there is no universally accepted standard for IoT device compatibility and connectivity. IoT systems often rely on proprietary technologies and protocols [34], which makes it difficult to establish communication and exchange data between IoT devices from different vendors.
- *Scalability*: As mentioned earlier, due to limited interoperability, it is difficult for different IoT devices to communicate and exchange data. This limits the scalability and flexibility of IoT systems and makes it costly and resource-intensive to implement and maintain them. Moreover, the sheer volume of data generated by IoT devices may strain existing IT infrastructure and may require investment in data storage and processing capabilities.
- *Reliability and Downtime Risks*: For most of the applications, IoT Edge devices rely on continuous connectivity and power supply to function effectively. Any disruptions in network connectivity or power outages can impact the performance of IoT devices. There is a need to have contingency plans in the event of IoT device failures or connectivity issues.
- *Data Overload and Analysis Challenges*: Sensors generate enormous amounts of data. Extracting meaningful insights from this data requires advanced analytics capabilities. As IoT devices have limited processing capabilities, this data needs to be sent to Cloud servers for effective analysis and to derive actionable insights.

- *High Costs*: The cost of IoT devices is higher as compared to regular devices, resulting in an initial higher cost of deployment. The ongoing maintenance and support of IoT systems can also be costly. This cost needs to be reduced so these devices can be deployed easily.

3.7 Emerging Needs for IoT Devices

The Internet of Things (IoT) has revolutionized many industries, right from healthcare to automotive and smart cities. This rapid adoption of IoT has resulted in a much broader range of devices on the Internet. These devices include medical devices, vehicles, household appliances, electric meters, street lights, traffic lights and controllers, smart TVs, and digital assistants such as Amazon Alexa and Google Home [35]. These are driving innovations in creating new opportunities for products, services, and businesses. These innovations are putting demands on IoT devices in terms of higher processing and storage capacity, as described below.

The IoT devices of the future must meet the following needs to support emerging products and services:

- *Distributed Data Storage and Processing*: There will be many new IoT applications that have to make decisions in real time and cannot tolerate delays due to network latency. These devices must handle data-related functions locally instead of transmitting data to the Cloud server. For example, motor vehicles and surgical robots.

- The decentralized storage of data requires that all locations maintain the same level of security; otherwise, the hackers may break in and corrupt the data, resulting in erroneous results. To avoid this, IoT systems need to incorporate mechanisms to ensure the accuracy and consistency of data at the decentralized nodes. This needs to be consistent with the proper functioning of the overall system.
- *Edge Computing*: Edge computing is becoming a popular approach to reduce the latency and bandwidth requirements associated with Cloud computing. This requires that the IoT Edge devices must have sufficient computing and storage capacity. These capabilities can be achieved by using low-power microcontrollers and single-board computers, which are connected to IoT devices and sensors. Integrating AI applications directly on the Edge devices (IoT devices) further enhances Edge computing capabilities. As Edge computing distributes data processing and analytics across multiple devices, it often creates redundant systems that are less prone to failure. The sensitive data can be kept more secure, as it does not need to be transmitted to a centralized server.
- *Secured Sensors and Devices*: IoT devices incorporate sensors that collect data from the physical world; these can be subjected to electromagnetic radiation that may cause them to malfunction. For example, spoofing location data can cause a

connected car to veer off course. As IoT devices are also connected to the Internet, a cyberattack may corrupt them to malfunction. There is a need to incorporate mechanisms to protect devices from such an attack.

- *Implementing Strong Authentication and Encryption*: These methods protect IoT devices and data from unauthorized access and tampering. Encryption also ensures the integrity of data and communications. The IoT devices need to provide support for biometrics, tokens, and private keys, which offer a secure and reliable means of user authentication. These will reduce the risk of cyberattacks and ensure the security of sensitive data.
- ***Regular Software Updates and Patch Management*: To improve security and minimize** vulnerabilities of IoT devices, there is a need for regular software updates and patches. The IoT devices should be capable of running automated patching tools and employ secure patching protocols [36].
- *AI and Machine Learning*: The integration of AI with IoT devices can create new opportunities as it can help to analyze large amounts of data that is generated by sensors to gain insights and make predictions. The future trend is toward decentralized architectures where AI processing can be done on Edge devices. This requires the IoT devices to have more processing and storage capabilities. This capability can be used to optimize heating, ventilation, and air-conditioning (HVAC) systems, reducing energy consumption and improving comfort.
- *Support for 5G Networks*: The increased speed and lower latency of 5G networks make it possible to support real-time applications like remote surgery and augmented reality. The newer IoT devices should support 5G network connectivity natively. This will improve the speed, reliability, and security of IoT systems.
- *Sustainability and Green IoT*: Climate change is forcing humans to look for renewable energy sources, such as solar or wind power, to power IoT devices. Organizations are working on developing IoT devices that are designed to reduce energy consumption and minimize waste [37]. They are also looking at using eco-friendly materials and manufacturing processes, reducing the environmental impact of IoT devices.

3.8 Summary

The Internet of Things (IoT) is a transformative technology that connects various devices to the Cloud, enabling better connectivity in various industries such as healthcare, manufacturing, retail, and smart homes.

The chapter discusses a typical architecture to support an IoT computing environment, covering various layers and functionalities provided by each layer to create a seamless system. It also discusses the development framework for building robust and reliable IoT systems, covering Fog computing and Edge computing models. Each model has its advantages and disadvantages, which were highlighted by relevant use cases. The three use cases covering smart home systems, remote monitoring and management of solar plants, and the use of smart helmets in providing immediate emergency healthcare services in accidents were discussed to highlight the advantages offered by intelligent Edge devices. The chapter highlights the current limitations of IoT devices at the Edge and the functionalities that need to be provided by these Edge devices to support emerging applications in the future.

3.9 Points to Ponder

1. Given that IoT is a relatively *new technology and lacks standards*, how can you make IoT devices from different vendors work together?
2. Mission-critical IoT devices are expected to communicate continuously and seamlessly with the Cloud, even in difficult situations, without failure. How can this be planned?
3. With billions of connected IoT devices, rapid identification and authentication of devices is a challenge. What can be done to simplify this?
4. Data security is a major concern across prominent businesses and government agencies. How can the security of IoT systems be enhanced?
5. How can an individual's privacy be reasonably maintained without impacting the benefits obtained by deploying IoT systems?
6. What are the ethical and legal considerations that a business must consider when deploying IoT systems?
7. Is Fog computing required when Edge computing becomes prevalent?

3.10 Answers

1. *Given that IoT is a relatively new technology and lacks standards, how can you make IoT devices from different vendors work together?*

   We can follow the guidelines given below while designing IoT systems for compatibility and integration:

   1. Establish universally accepted specifications and protocols for full interoperability between devices and applications.
   2. The systems should be built within the universal framework to establish open and transparent communications.
   3. IoT devices work much better with one another when developed with software-driven technologies instead of hardware-driven technologies. This allows flexibility that may be needed for integration.
   4. The devices should maintain an open-sourced messaging protocol for effective data transfer.
   5. Guarantee secure communication between devices.
   6. Enable developers to create applications that are compatible with different devices.

2. *Mission-critical IoT devices are expected to communicate continuously and seamlessly with the Cloud, even in difficult situations, without failure. How can this be planned?*

   Wireless connectivity is highly complex, and its standards are fast-evolving. Under such circumstances, the best way to plan a fail-proof system is to incorporate highly flexible, configurable, and upgradeable designs. These designs should be able to work in both R&D and production environments and meet future needs. The design should avoid having a single point of failure by having distributed and redundant systems.
3. *With billions of connected IoT devices, rapid identification and authentication of devices is a challenge. What can be done to simplify this?*

   There is a need to have a mechanism of strong authentication and identification of IoT devices. This ensures that connected devices can be trusted to communicate securely with other devices as well as the backend infrastructure. This requires that every device should have a unique identity that can be used to connect to a gateway or central server. This will also assist IT and system administrators in tracking each device throughout its life cycle, communicating with it securely, and preventing any harmful behavior.
4. *Data security is a major concern across prominent businesses and government agencies. How can the security of IoT systems be enhanced?*

   There are many approaches available to enhance security and safeguard IoT systems against cyberattacks. For example, the redundancy inherent in the distributed nature of the IoT can guard against cyberattacks, including zero-day attacks on a single device. It can be done using an accountability approach. IoT systems can assign some nodes to recheck the calculations of other nodes periodically. If the majority of the nodes assigned to rerun the calculation come to a different result, the node being checked is declared to be at fault and isolated from the system.

   Another technique known as state estimation can protect against sensor attacks. This approach takes the early experiences with a particular environment to estimate the reasonable range of possible values that a sensor might report. If the system receives data from the sensor that falls outside that range, it can flag that sensor for additional scrutiny or even go so far as to isolate it from the system.
5. *How can an individual's privacy be reasonably maintained without impacting the benefits obtained by deploying IoT systems?*

   With the large-scale use of IoT systems, the concerns are no longer limited to the protection of privacy and sensitive data, but our health and habits can become the target of a security attack.

   To protect the privacy of an individual, we can adopt a scheme known as differential privacy. This scheme can prevent data from being attributed to any specific person in situations when individual data points are combined and reported as an aggregate value. This can be achieved by adding a predefined range of random noise to each data point. If the number of observations being aggregated is large enough, the central limit theorem of statistical analysis dictates that the randomness of the noise will tend to cancel itself out.

Companies need to develop policies that respect the privacy of every individual. In addition, they need to plan and implement these policies while deploying IoT technology or innovative services.

6. *What are the ethical and legal considerations that a business must consider when deploying IoT systems?*

   The businesses must consider issues associated with data collection, privacy, and consent. Businesses must ensure that they have clear policies and procedures in place to address data privacy and protection. Additionally, they must comply with relevant regulations and standards to avoid potential legal consequences. Transparent communication with customers regarding data collection and usage is essential to maintain trust and uphold ethical standards.

7. *Is Fog computing required when Edge computing becomes prevalent?*

   In Edge computing, the Edge devices send huge amounts of data to the Cloud, this may consume a lot of bandwidth and introduce delays. The Fog computers receive the data from Edge devices and analyze what is important to create analytical summaries. This metadata is then shared with a central Cloud platform, where it is further analyzed to generate actionable insights. The unimportant data may be either deleted or kept with Fog computers for further analysis.

   Wearable smart devices such as fitness trackers are an excellent example of using Fog computing. Such devices rely on linked smartphones to process the data they collect and instantly show the output to the user. In this case, smartphones serve as Fog computing devices.

References

1. Foote, K. D. (2022). A brief history of the internet of things. January 2022. https://www.dataversity.net/brief-history-internet-things/
2. Sharma, N., Shamkuwar, M., & Singh, I. (2019). The history, present and future with IoT. In *Internet of things and big data analytics for smart generation* (pp. 27–51). Springer.
3. ITU Internet Reports, International Telecommunication Union. The Internet of Things: 7th Edition. https://www.itu.int/net/wsis/tunis/newsroom/stats/The-Internet-of-Things-2005.pdf
4. Internet of Things statistics for 2023 – Taking Things Apart. https://dataprot.net/statistics/iot-statistics/
5. *Cisco Internet Business Solutions Group* White Paper, The Internet of Things, Dave Evans, April 2011
6. IoT Architecture: Detailed Explanation of the 4 IoT Layers, July 2023. https://www.theiotacademy.co/blog/iot-architecture/
7. Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering, 2017*, Article ID: 9324035.
8. Locke, D. (2010). MQ telemetry transport (MQTT) v3. 1 protocol specification, IBM developer Works Technical Library. http://www.ibm.com/developerworks/webservices/library/wsmqtt/index.html
9. Stanford-Clark, A., & Linh Truon, H. (2008). MQTT for sensor networks (MQTT-S) protocol specification, International Business Machines Corporation Version 1.
10. Gomez, C., Oller, J., & Paradells, J. (2012). Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology. *Sensors, 12*(9), 11734–11753.

11. Baronti, P., Pillai, P., Chook, V. W. C., Chessa, S., Gotta, A., & Hu, Y. F. (2007). Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standards. *Computer Communications, 30*(7), 1655–1695.
12. Bandyopadhyay, S., Sengupta, M., Maiti, S., & Dutta, S. (2011). Role of middleware for the Internet of things: A study. *International Journal of Computer Science & Engineering Survey, 2*(3), 94–105.
13. *Implementation of a Cloud-Based Home Automation System*, ResearchGate (2016). researchgate.net/publication/337568027_Implementation_of_a_Cloud-Based_Home_Automation_System
14. A preliminary study of open-source IoT development frameworks. In *ICSEW'20: Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops* (pp. 679–686)
15. Top 25 open source IoT frameworks you need to know. https://www.esparkinfo.com/blog/open-source-iot-frameworks.html
16. awesome-open-iot. https://github.com/Agile-IoT/awesome-open-iot
17. Buyya, R., & Srirama, S. N. (2019). *Fog and edge computing: Principles and paradigms*. Wiley. ISBN: 978-1-119-52498-4 January 2019 512 pages.
18. Bhatia, A. (2023). Edge computing in IoT: What it is and how to use it successfully, June 2023. IEEE Computer Society. https://www.computer.org/publications/tech-news/trends/edge-computing-in-iot
19. What is IoT edge computing? Redhat, July 2022. https://www.redhat.com/en/topics/edge-computing/iot-edge-computing-need-to-work-together
20. Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for the Internet of things and analytics. In *Big data and internet of things: A road map for smart environments* (pp. 169–186). Springer.
21. Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the 1st ACM MCC workshop on mobile cloud computing* (pp. 13–16).
22. Singh, S. P., Nayyar, A., Kumar, R., et al. (2019). Fog computing: from architecture to edge computing and big data processing. *Journal of Supercomputing, 75*, 2070–2105.
23. Alwakeel, A. M. (2021). An overview of fog computing and edge computing security and privacy issues. *Sensors, 21*(24), 8226.
24. Lakshminarayanan, V., Ravikumar, A., Sriraman, H., Alla, S., & Chattu, V. K. (2023). Health care equity through intelligent edge computing and augmented reality/virtual reality: A systematic review. *Journal of Multidisciplinary Healthcare, 16*, 2839–2859. https://www.dovepress.com/health-care-equity-through-intelligent-edge-computing-and-augmented-re-peer-reviewed-fulltext-article-JMDH
25. Okay, F. Y., Ozdemir, S. (2016). A fog computing based smart grid model. In: *Conference: 2016 International Symposium on Networks, Computers and Communications (ISNCC)*, May 2016
26. Stojmenovic, I., & Wen, S. (2014, September). The fog computing paradigm: scenarios and security issues. In *Proceedings of the federated conference on Computer Science and Information Systems (Fed-CSIS '14)* (pp. 1–8). IEEE.
27. Kaiborta, A. K., & Samal, S. (2022). IoT-based voice assistant for home automation. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2022 (pp. 165–172).
28. Patil, S. M., Vijayalashmi, M., & Tapaskar, R. (2017). IoT based solar energy monitoring system. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India (pp. 1574–1579).
29. IoT-based solar energy measurement and monitoring model. *International Journal of Photoenergy* (2022)
30. Start-ups innovate to enhance helmet safety quotient. https://www.autocarpro.in/feature/vegan-cars-are-the-next-big-fad-in-the-automotive-industry-45290

31. Accelerometers, Chris Woodford. http://www.explainthatstuff.com/accelerometers.html
32. Anjum, A., & Ilyas, M. U. (2013). Activity recognition using smartphone sensors. In *Proceedings of the IEEE 10th Consumer Communications and Networking Conference (CCNC '13)* (pp. 914–919), Las Vegas, Nev, USA, January 2013.
33. Hlaing, N. N. S., Naing, M., & Naing, S. S. (2019, May–June). GPS and GSM based vehicle tracking system. *International Journal of Trend in Scientific Research and Development* 3(4). https://www.researchgate.net/publication/334123684_GPS_and_GSM_Based_Vehicle_Tracking_System
34. Top 12 most commonly used IoT protocols and standards. https://www.techtarget.com/iotagenda/tip/Top-12-most-commonly-used-IoT-protocols-and-standards
35. Growing opportunities in the Internet of Things. https://www.mckinsey.com/industries/private-equity-and-principal-investors/our-insights/growing-opportunities-in-the-internet-of-things
36. Software Updates for Internet of Things (suit). https://datatracker.ietf.org/wg/suit/about/
37. Henkel, J., Pagani, S., Amrouch, H., Bauer, L., & Samie, F. (2017). Ultra-low power and dependability for IoT devices (Invited paper for IoT technologies). In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, Lausanne, Switzerland (pp. 954–959)

Chapter 4
Foundations of Artificial Intelligence

4.1 Introduction

Artificial intelligence (AI) is a multidisciplinary field that encompasses the creation of intelligent agents, which are systems that can reason, learn, and act autonomously. AI research has been highly successful in developing effective techniques for solving a wide range of problems, from game playing to medical diagnosis [1]. Artificial intelligence stands at the forefront of technological innovations, aiming to replicate and augment human-like intelligence in machines.

The idea of creating AI dates back to ancient times, with myths and legends from many cultures featuring stories of intelligent machines. However, the field of AI as we know it today has emerged in the mid-twentieth century, with the development of computers and the formalization of the concept of intelligence [2].

AI is a vast field that can be classified along various aspects as follows:

- Capabilities
- Learning
- Techniques
- Ethical considerations

We cover each of these classifications in detail below.

4.1.1 AI Classification Based on Capabilities

As shown in Fig. 4.1, AI capabilities can be classified as:

- Artificial Narrow Intelligence (ANI): Also known as weak AI, ANI systems are designed to perform specific tasks, often mimicking human capabilities in a nar-

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_4

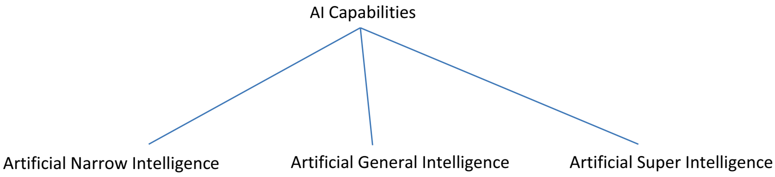

AI Capabilities
Artificial Narrow Intelligence
Artificial General Intelligence
Artificial Super Intelligence

Fig. 4.1 AI capabilities

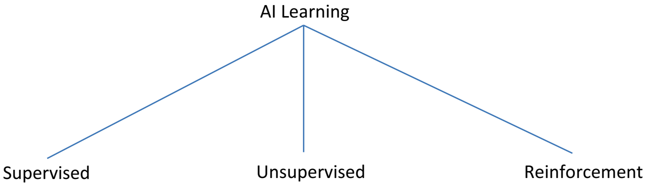


Fig. 4.2 AI learning

row domain. Examples include chess-playing computers, spam filters, facial recognition software, etc.

- Artificial General Intelligence (AGI): Also known as the strong AI, AGI systems aim to possess human-level intelligence across a broad range of cognitive tasks. While AGI has not yet been achieved, it remains a central goal of AI research [3].
- Artificial Super Intelligence (ASI): Hypothetical AI systems that would surpass human intelligence. The impact and feasibility of ASI are widely debated.

4.1.2 AI Classifications Based on Learning

As shown in Fig. 4.2, AI learnings can be classified as:

- Supervised Learning: This refers to learning from labeled data, where each data point is associated with a correct output. The algorithm learns to map the input data to the expected output labels.
- Unsupervised Learning: This refers to learning from unlabeled data, where the algorithm identifies patterns and structures in the data.
- Reinforcement Learning: This refers to learning by interacting with an environment. The algorithm receives rewards or penalties for its actions and learns to take actions that maximize its rewards. An algorithm strives to improve its output in an iterative manner.

4.1.3 AI Classification Based on Techniques

As shown in Fig. 4.3, AI techniques can be classified as:

- Symbolic AI: This uses symbols to represent knowledge and to reason about the world.
- Search-Based AI: This uses search algorithms to find solutions to problems by exploring a space of possible solutions.
- Planning and Scheduling: This refers to AI systems that can plan and schedule complex tasks.
- Expert Systems: These refer to AI systems that capture the knowledge and expertise of human experts based on codified rules. Often probabilities are used to traverse the decision trees.
- Neural Networks: These refer to AI systems inspired by the structure and function of the human brain. These work by using training data to set the weights in a neural network, which are then used during the inference phase.
- Deep Learning: This is a type of machine learning that uses artificial neural networks with multiple layers to learn from data.
- Convolutional Neural Networks (CNNs): These are a type of neural network that is well-suited for analyzing visual imagery.
- Generative Adversarial Networks (GANs): This uses a type of deep learning model that consists of two competing neural networks, i.e., a generator and a discriminator. The goal is to improve the output with competition between the two networks.

4.1.4 AI Classification Based on Ethical Considerations

As shown in Fig. 4.4, AI ethical considerations can be classified as:

- Benevolent AI: This AI benefits humanity and promotes positive outcomes.
- Malevolent AI: This AI could pose risks to humanity, intentionally or unintentionally.
- Aligned AI: This AI aligns with human values and goals, ensuring its beneficial use.

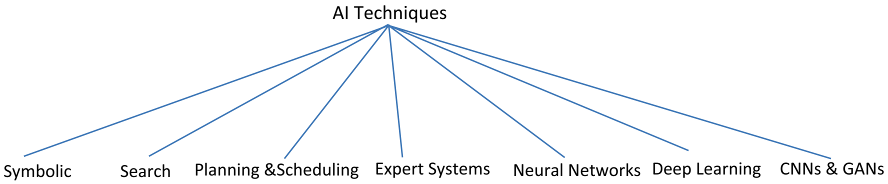


Fig. 4.3 AI techniques

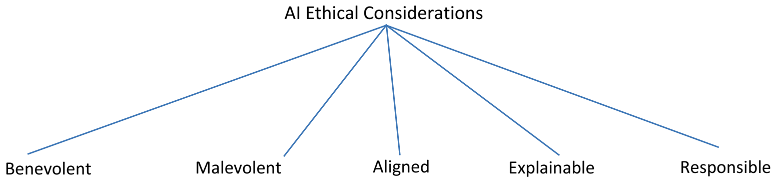


Fig. 4.4 AI ethical considerations

- Explainable AI: This refers to AI that can provide explanations for its decisions, fostering transparency and trust.
- Responsible AI: This is the AI developed and used in a responsible manner, addressing ethical concerns and potential risks.

4.2 Historic Evolution of AI

Artificial intelligence (AI) has a rich and fascinating history, spanning over centuries of philosophical inquiries, technological advancements, and ground-breaking research. The pursuit of AI has captivated minds for generations, fuelled by the dreams of creating machines that are capable of intelligent thoughts and behaviors.

The historical evolution of AI is a tale of continuous progress, setbacks, and reinvention. From the early philosophical musings about artificial intelligence to the modern-day AI systems that permeate our lives, AI has undergone a remarkable transformation. As AI continues to evolve, it will undoubtedly play an increasingly significant role in shaping our future.

4.2.1 Early Foundations of AI (Pre-1950s)

The notion of artificial intelligence can be traced back to ancient mythology, with stories of self-moving automatons and artificial beings. However, the formal study of AI emerged in the mid-twentieth century, driven by advancements in mathematics, computer science, and philosophy.

- Alan Turing's Turing Test: In 1950, Alan Turing published his seminal paper, "Computing Machinery and Intelligence," introducing the Turing Test as a standard for measuring machine intelligence. The test proposes that a machine can be considered intelligent if it can carry on a conversation indistinguishable from a human [4].
- The Dartmouth Conference (1956): The Dartmouth Summer Research Project on Artificial intelligence, held in 1956, is widely considered the birth of AI as a distinct field of study. This landmark event brought together leading researchers

(including John McCarthy, deemed as the Father of AI) and established AI as a legitimate academic discipline [5, 6].

4.2.2 The Rise of AI (1950s–1970s)

The 1950s and 1960s marked a period of rapid growth and enthusiasm for AI research. Early AI systems demonstrated promising capabilities in problem-solving, game playing, and language processing, fueling optimism about the potential of AI to achieve human-level intelligence.

- Symbolic AI and Expert Systems: Symbolic AI, also known as the expert systems, focused on representing knowledge and reasoning using symbolic expressions and rules. Expert systems gained popularity in the 1970s, demonstrating success in applications such as medical diagnosis and financial planning [7].
- Search Algorithms and Heuristics: During this era, search algorithms and heuristics became essential tools for AI problem-solving [8, 9]. Search algorithms systematically explore a space of possible solutions, while heuristics provide informed guidance to reduce the search space and find solutions efficiently.

4.2.3 Challenges and Setbacks (1970s–1980s)

The 1970s and 1980s saw a period of challenges and setbacks for AI research. Initial optimism waned as the complexity of real-world problems and the limitations of early AI techniques became apparent. This period was marked by a focus on specific AI subfields and a reassessment of the goals and approaches to AI research.

AI Winter: The term “AI winter” refers to a period of reduced funding and enthusiasm for AI research in the late 1970s and early 1980s. This period was characterized by a recognition of the difficulty of achieving true artificial general intelligence (AGI) and a focus on more practical and attainable AI applications.

4.2.4 The Resurgence of AI (1990s–Present)

The 1990s witnessed a resurgence of AI research, driven by advancements in computing power, data availability, and machine learning algorithms. AI applications began to make significant impacts in various domains, leading to renewed interest and investment in AI research.

- The Rise of Machine Learning: Machine learning emerged as a distinct subfield of AI, focusing on algorithms that can learn from data without explicit programming. Techniques like decision trees, neural networks, and statistical learning became increasingly prominent [10].

- The Rise of Deep Learning: Deep learning, a subfield of machine learning, emerged in the 2000s and revolutionized AI. Deep learning algorithms [11], particularly artificial neural networks with multiple layers, demonstrated remarkable performance in tasks like image recognition, natural language processing, and speech recognition.
- AI in the Real World: AI applications became increasingly prevalent in various aspects of our lives, from self-driving cars to personalized recommendations to virtual assistants. AI is transforming industries like healthcare, finance, transportation, and manufacturing.

  Below is a brief look at the progress of AI during the last two centuries, as shown in Table 4.1:

4.3 AI Computing Environment

The AI computing environment consists of the infrastructure, hardware, software, and networking components that collectively support the development, training, and deployment of artificial intelligence (AI) systems. This specialized environment plays a crucial role in enabling the capabilities of AI, encompassing various technologies and configurations tailored to the unique demands of AI workloads.

4.3.1 Components of AI Computing Environment

These consist of several hardware and software components as described below.

4.3.1.1 Hardware Accelerators

- **Graphics Processing Units (GPUs):** These are widely used for accelerating deep learning tasks. GPUs excel in parallel processing and are well-suited for training large neural networks [14].
- **Tensor Processing Units (TPUs):** Tensors are mathematical objects that generalize scalars, vectors and matrices to higher dimensions. TPUs, designed by Google, are hardware accelerators optimized for machine learning workloads offering high performance with lower power consumption [15].

Table 4.1 Evolution of artificial intelligence [12, 13]

1800s	Babbage's analytical engine: Charles Babbage's design for the analytical engine, a mechanical computer, laid the foundation for modern computing and artificial intelligence
1842	Ada Lovelace, considered the first computer programmer, recognized the potential of the analytical engine for artificial intelligence
1943	McCulloch and Pitts' work on the mathematical model of a neuron laid the foundation for artificial neural networks
1950	Alan Turing introduced the Turing test as a standard for measuring machine intelligence
1956	The Dartmouth Conference is considered the birth of artificial intelligence as a distinct field of study
1957	John McCarthy's coining of the term "artificial intelligence"
1958	Frank Rosenblatt developed the perceptron, a simple artificial neural network, marking a significant step in AI research
1960s	Symbolic AI and expert systems became prominent, focusing on representing knowledge and reasoning using symbolic expressions and rules
1965	John McCarthy developed the List Processing (LISP) programming language, widely used in AI research
1966	The SHAKEY robot, developed at Stanford Research Institute (SRI) International, demonstrated the potential of AI in robotics
1958	Frank Rosenblatt developed the perceptron, a simple artificial neural network, marking a significant step in AI research
1969	Marvin Minsky and Seymour Papert's book *Perceptrons* highlighted the limitations of early perceptrons and led to a period of decline in AI research
1970s	The AI Winter period was characterized by reduced funding and enthusiasm for AI research due to the perceived limitations of early AI techniques
1979	Nils Nilsson's book provided a comprehensive overview of problem-solving methods in AI working in concert with our computers and phones to share data and interact
1980s	Machine learning emerged as a distinct subfield of AI, focusing on algorithms that can learn from data without explicit programming
1982	John Hopfield introduced the Hopfield network, a type of recurrent neural network, demonstrating the potential of neural networks for complex tasks
1986	The Rumelhart-Hinton-Williams back propagation algorithm, a method for training multi-layer neural networks, revitalized research in deep learning
1990s	AI research regained momentum due to advancements in computing power, data availability, and machine learning algorithms
1997	IBM's Deep Blue computer defeated world chess champion Garry Kasparov, marking a significant milestone in AI
2000s	Deep learning, a subfield of machine learning, gained prominence, leading to breakthroughs in areas like image recognition, natural language processing, and speech recognition
2001	Google DeepMind's AlphaGo program defeated world Go champion Lee Sedol, demonstrating the power of deep learning in complex games
2007	The ImageNet large -scale visual recognition challenge (ILSVRC) became a major benchmark for image classification and object detection, fueling the development of deep learning algorithms

(continued)

Table 4.1 (continued)

2012	AlexNet, a convolutional neural network, won the ILSVRC 2012, sparking a resurgence of interest in deep learning
2014	Apple's Siri virtual assistant became widely popular, demonstrating the potential of conversational AI
2016	Google DeepMind's AlphaGo Zero program defeated its predecessor, AlphaGo, without human intervention, highlighting the ability of deep learning to learn and improve autonomously
2017	An Uber self-driving car accident in Arizona resulted in a fatality, highlighting the challenges and safety concerns of autonomous driving systems
2018	Google Translate achieved human parity on a benchmark test of 111 languages, demonstrating the remarkable progress in natural language processing
2020	AI played a role in various aspects of the COVID-19 pandemic, including contact tracing, drug discovery, and vaccine development
2021	Advancements in reinforcement learning, self-supervised learning, and AI ethics. OpenAI's ChatGPT, potentially the AI "Killer App," demonstrated remarkable capabilities in natural language processing, further advancing the state of AI
2022	AI is being explored for various applications related to climate change, such as renewable energy development, weather forecasting, and ocean monitoring
2023	The ethical implications of AI, such as bias, fairness, and privacy, are becoming increasingly important and need to be addressed responsibly

4.3.1.2 High-Performance Computing (HPC) Systems

Clusters and Supercomputers AI applications often require significant computational power. HPC systems, comprising clusters of interconnected computers or supercomputers, provide the necessary resources for training complex models.

4.3.1.3 Storage Solutions

Distributed Storage AI workloads are likely to generate and consume massive datasets. Distributed storage solutions, often in the form of clustered file systems or object storage, are essential for managing and accessing large volumes of data efficiently.

4.3.1.4 Cloud Computing Platforms

Based on the requirements and Cloud abstraction level required, here are the various types of Cloud computing platforms [16]:

- **Software as a Service (SaaS)**: This is focused on the end users of Cloud, to provide them with application-level access such that multiple users can execute the same application binary in their own virtual machine or server instance. These application sessions may be running on the same or different underlying hardware, and SaaS enables application providers to upgrade or patch their binaries in a seamless manner. Examples of SaaS providers are Salesforce.com providing CRM (customer relationship management), Google.com serving documents and Gmail, etc., all of which are hosted in the Cloud.

- **Platform as a Service (PaaS):** This is focused on application developers with varying computing needs according to their project stages. These are met by servers that can vary in number of CPU cores, memory, and storage at the user's will. Such servers are called elastic servers. Their services can autoscale, i.e., new virtual machines can start for load balancing with a minimal administrative overhead. Examples of PaaS providers are Google's App Engine, Microsoft's Azure, Red Hat's Makara, Amazon Web Services (AWS) Elastic Beanstalk, AWS Cloud Formation, etc. These Cloud service providers (CSPs) have the capability to support different operating systems on the same physical server.
- **Infrastructure as a Service (IaaS):** This is the bottom-most layer in a Cloud stack, providing direct access to virtualized or containerized hardware. In this model, servers with given specifications of CPUs, memory, and storage are made available over a network. Examples of IaaS providers are AWS EC2 (Elastic Compute Cloud), OpenStack, Eucalyptus, Rackspace's CloudFiles, etc.

4.3.1.5 Frameworks and Libraries

- **TensorFlow, PyTorch, and Keras:** These popular open-source frameworks provide a foundation for building and training machine learning models. They are optimized to work seamlessly with hardware accelerators.
- **CUDA and cuDNN:** NVIDIA's CUDA is a parallel computing platform that allows developers to use GPUs for general-purpose processing, while cuDNN is a GPU-accelerated library for deep neural networks.

4.3.1.6 Data Pipelines and Preprocessing Tools

- **Apache Spark, Apache Flink:** Big data processing frameworks are essential for handling large datasets efficiently. They enable distributed data processing and are often used in conjunction with AI applications [17].

4.4 AI Edge Computing

AI Edge computing refers to the practice of processing data and running AI algorithms locally on the Edge devices, such as sensors, smartphones, or IoT devices, rather than relying solely on centralized Cloud servers. This distributed computing paradigm has gained prominence due to the advantages it offers in terms of reduced latency, improved privacy, bandwidth efficiency, and the ability to operate in real-time. The AI Edge computing environment is a specialized setup that caters to the unique requirements of deploying and running AI models at the Edge

4.4.1 Components of AI Edge Computing Environment

A typical AI Edge computing environment consists of various components.

4.4.1.1 Edge Devices

- **IoT Devices:** These consist of sensors, cameras, and other Internet of Things (IoT) devices that act as the front-line data collectors at the Edge.
- **Intelligent IoT Devices:** These devices often have limited processing power and storage, such as face recognition cameras and motion detection sensors.

4.4.1.2 Edge Servers

- **Embedded Systems:** These are systems with some embedded computing capabilities, such as Edge gateways and microcontrollers, that play a crucial role in preprocessing data before it is sent to centralized servers.
- **Local Servers:** In some Edge computing setups, local servers with moderate computing power may be deployed to handle more complex AI tasks, allowing for faster processing without relying on a distant data center.

4.4.1.3 Edge AI Processors

- **Low-Power AI Chips:** These are specialized AI processors optimized for power efficiency. These are designed to run AI models on Edge devices with limited resources.
- **Field-Programmable Gate Arrays (FPGAs):** These are usually add-on components in an Edge server. FPGAs provide flexibility and can be programmed to accelerate specific AI workloads, making them suitable for Edge computing environments.

4.4.1.4 Edge Computing Frameworks

- **TensorFlow Lite, ONNX Runtime:** These frameworks are tailored for Edge computing and allow developers to deploy and run lightweight versions of AI models on Edge devices [18].

- **EdgeX Foundry:** This is an open-source framework that facilitates interoperability between IoT devices and Edge computing systems [19].

4.4.1.5 Edge-Focused Machine Learning Models

- **Model Optimization:** AI models are often optimized for Edge deployment, involving techniques such as quantization, pruning, and compression to reduce their size and resource requirements.
- **Federated Learning:** This approach enables AI models to be trained across multiple Edge devices while keeping all data localized and preserving privacy. Code and model travel to the site of participating parties.

4.4.1.6 Edge-to-Cloud Connectivity

- **Low-Latency Networks:** Edge computing relies on low-latency networks to ensure quick communications amongst the Edge devices and, if needed, to centralize Cloud resources. Usually, if Edge devices are distributed across different sites, then they may not communicate with each other and only connect to the central Cloud servers.
- **5G Networks:** The deployment of 5G networks enhances connectivity, making it more feasible to process and transmit data in real time from Edge devices [20].

4.4.2 Challenges and Considerations

- **Resource Constraints:** Edge devices often have limited computing power, memory, and energy resources, requiring AI models to be lightweight and optimized for efficiency.
- **Security and Privacy:** Securing Edge devices against physical and cyber threats is crucial. Additionally, handling sensitive data locally raises concerns about privacy, necessitating robust security measures [21].
- **Orchestration and Management:** Coordinating and managing AI workloads across a diverse range of Edge devices can be challenging. Orchestration tools are essential for optimizing resource utilization [22].
- **Interoperability:** Ensuring interoperability between different Edge devices and frameworks is crucial for building a cohesive and collaborative Edge computing environment. This is done through standard protocols [23].

4.4.3 Future Trends

- **Decentralized AI Architectures:** AI processing occurs on Edge devices in decentralized AI architectures. This trend is likely to continue as technology advances and devices become more powerful.
- **AI at the Network Edge:** The integration of AI directly into network infrastructure, known as AI at the network Edge could further enhance Edge computing capabilities. This would be by enabling intelligent decision-making within the network itself [24].

4.4.4 Edge AI Ecosystem Growth

The growth of an ecosystem around Edge AI, including the Edge AI market [25] and developer communities, is expected to drive innovation and standardization in the field.

In conclusion, AI Edge computing represents a paradigm shift in how AI is deployed and utilized, bringing computation and intelligence closer to the data sources. As technology evolves, the AI Edge computing environment will play a pivotal role in enabling a wide range of applications, from smart cities and autonomous vehicles to industrial automation and healthcare.

4.5 AI Analytics

AI analytics refers to the use of advanced analytics techniques powered by AI to derive insights, patterns, and valuable information from data [26]. This combination of AI and analytics aims to enhance the efficiency and effectiveness of data analysis. This provides organizations with a deeper understanding of their datasets and enables data-driven decision-making.

4.5.1 Key Aspects of AI Analytics

These consist of various items as detailed below.

4.5.1.1 Data Processing and Integration

- **Data Preparation:** AI analytics involves the preprocessing and cleaning of raw data. AI algorithms can automate tasks such as data cleansing, imputation for missing data, and normalization. This ensures that the data is ready for analysis.
- **Data Integration:** AI analytics often deals with diverse datasets, integrating structured and unstructured data from various sources. So, integrating this data is a crucial step in gaining a comprehensive view for analysis.

4.5.1.2 Machine Learning and Predictive Analytics

- **Predictive Modeling:** AI analytics leverages machine learning algorithms for predictive modeling. This includes regression analysis, decision trees, and more advanced techniques such as neural networks to forecast future trends or outcomes based on historical data.
- **Anomaly Detection:** Machine learning models can identify anomalies or outliers in data, helping organizations detect irregular patterns that may indicate fraud, errors, or unusual behaviors.

4.5.1.3 Natural Language Processing (NLP)

- **Text and Sentiment Analysis:** NLP allows AI analytics to analyze and understand human language. Organizations can extract insights from textual data, such as customer reviews, social media comments, and news articles, to gauge sentiment and make informed decisions [27].
- **Chatbots and Virtual Assistants:** AI-driven chatbots [28] and virtual assistants utilize NLP to understand and respond to user queries, providing a more interactive and user-friendly experience.

4.5.1.4 Image and Video Analysis

- **Computer Vision:** AI analytics can analyze images and videos to recognize patterns and objects. Applications include facial recognition, object detection, and quality control in manufacturing.
- **Medical Imaging:** In healthcare, AI analytics is used to analyze medical images, aiding in the early detection of diseases and improving diagnostic accuracy [29].

4.5.1.5 Prescriptive Analytics

AI analytics goes beyond predicting outcomes by suggesting actions to optimize results. This involves the use of optimization algorithms that recommend the best course of action based on the predicted outcomes [30].

4.5.1.6 Continuous Learning and Adaptation

AI analytics models can adapt and improve over time through reinforcement learning. This involves learning from user interactions and feedback from the relevant environment to continually enhance model performance.

4.6 Edge AI Analytics

Edge AI analytics refers to the deployment of AI and advanced analytics directly on Edge devices, such as sensors, cameras, or Internet of Things (IoT) devices. This approach brings the power of AI algorithms closer to the data source, allowing for real-time processing, reduced latency, and improved efficiency. Edge AI analytics techniques and tools are relevant in scenarios where immediate insights are crucial, bandwidth constraints exist, or privacy concerns necessitate local data processing.

4.6.1 Key Components of Edge AI Analytics

Here are the various key components:

4.6.1.1 Edge Devices

- **IoT Devices:** Sensors, cameras, and other IoT devices serve as the primary data sources at the Edge [31]. These devices may be equipped with processing capabilities to execute AI algorithms locally.
- **Edge Gateways:** These consist of intermediate devices positioned between Edge devices and centralized servers. Edge gateways preprocess and filter data before it is sent to the Cloud. These may also host certain AI analytics functions [32].

4.6.1.2 Edge AI Processors

- **Low-Power AI Chips:** This refers to customized AI processors optimized for low power consumption. These may be designed for deployment in Edge devices with limited compute and storage resources. Note that some Edge AI devices may be deployed in the field to run on battery or solar power.
- **FPGAs (Field-Programmable Gate Arrays):** FPGAs provide flexibility and can be programmed to accelerate specific AI workloads. This makes them suitable for Edge AI analytics. FPGAs also offer advantages due to rapid deployment as compared to ASICs and faster run times as compared to software-only applications.

4.6.1.3 Lightweight AI Models

- **Model Optimization:** Edge AI analytics often involves deploying lightweight and optimized versions of AI models [33]. This ensures efficient execution on the Edge devices with limited computational resources.
- **Quantization and Compression:** Quantization [34] is the process of mapping a large set of values to a smaller set of discrete, finite values. Techniques such as quantization and model compression reduce the size of models, making them more suitable for deployment on Edge devices.

4.6.1.4 Real-Time Inference

- **Low-Latency Processing:** Edge AI analytics focuses on minimizing processing delays, enabling real-time inference and decision-making. This is particularly critical in applications such as autonomous vehicles, industrial automation, and healthcare.

4.6.2 Applications of Edge AI Analytics

- **Smart Cities:** Edge AI analytics is used in smart city applications for real-time monitoring of traffic, waste management, and public safety. It enables quicker response times and more efficient resource allocation.
- **Industrial IoT (IIoT):** Edge AI analytics, in industrial settings, supports predictive maintenance, quality control, and process optimization. This reduces downtime and improves overall operational efficiency.
- **Healthcare:** Edge AI analytics in healthcare allows for real-time analysis of patient data, enabling early detection of health issues. It also facilitates remote patient monitoring and personalized medicine.

- **Retail:** Retailers use Edge AI analytics for in-store analytics, customer behavior analysis, and inventory management. It enables personalized customer experiences and efficient supply chain operations.
- **Autonomous Vehicles:** Edge AI is crucial in autonomous vehicles for real-time object detection, path planning, and decision-making. Local processing ensures timely responses, enhancing safety and reliability.
- **Surveillance and Security:** Edge AI analytics enhances video surveillance systems by enabling on-device object detection, facial recognition, and anomaly detection, reducing the need for constant data transmission to centralized servers.

4.6.3 Challenges and Considerations

- **Resource Constraints:** Edge devices often have limited processing power, memory, and energy resources. Designing AI models and algorithms that can operate efficiently under these constraints is a key challenge.
- **Security:** Securing Edge devices against physical and cyber threats is crucial. Since these devices are often deployed in uncontrolled environments, ensuring data integrity and confidentiality is a priority for IT managers.
- **Model Maintenance:** Updating and maintaining AI models on a large number of distributed Edge devices can be complex. Strategies for efficient model updates and version control are essential.
- **Interoperability:** Ensuring interoperability between different Edge devices, processors, and frameworks is crucial for creating a seamless and collaborative Edge AI analytics environment.

4.6.4 Future Trends

- **AI at the Network Edge:** The integration of AI directly into network infrastructure, known as AI at the network Edge, could further enhance Edge AI analytics capabilities by enabling intelligent decision-making within the network itself.
- **Decentralized AI Architectures:** The trend towards more decentralized AI architectures, where AI processing occurs on Edge devices, is expected to continue as technology advances and devices become more powerful.
- **Edge AI Ecosystem Growth:** The growth of an ecosystem around Edge AI, including Edge AI marketplaces and developer communities, is expected to drive innovation and standardization in the field.
- **Federated Learning:** Federated learning [24], where models are trained across multiple Edge devices while keeping data localized, is gaining attention as a privacy-preserving approach to collaborative AI analytics.

In summary, Edge AI analytics represent a paradigm shift in how AI is deployed, enabling on-device processing and real-time decision-making. As technology continues to advance, the integration of AI at the Edge is expected to play a pivotal role in numerous applications across various industries.

4.7 Emerging Applications

Below are some of the emerging applications of AI and Edge AI:

- **Autonomous Vehicles:** AI is playing a crucial role in the development of self-driving cars and drones. Edge AI is used to process data in real-time, enabling these vehicles to make split-second decisions without relying heavily on Cloud-based processing or manual interventions.
- **Healthcare:** AI is being applied in medical imaging for the early detection of diseases, personalized medicine, and drug discovery. Edge AI can enable real-time analysis of patient data at the Edge devices, improving diagnostic speed and efficiency.
- **Industrial Internet of Things (IIoT):** AI and Edge AI are used in industrial settings for predictive maintenance, quality control, and process optimizations. Edge devices can help process data locally, thereby reducing latency and improving response times.
- **Smart Cities:** AI is used in smart city applications for traffic management, public safety, and resource optimization. Edge AI is employed to process data from various sensors and devices deployed throughout the city.
- **Retail:** AI is used for customer analytics, inventory management, and personalized shopping experiences. Edge AI can be applied in retail environments to analyze customer behavior in real time and optimize store operations.
- **Agriculture:** AI and Edge AI are employed in precision agriculture for crop monitoring, pest detection, and yield predictions. Edge devices on farm equipment can process data locally, enabling timely decision-making on the farm.
- **Energy Management:** AI is used to optimize energy consumption, predict equipment failures, and improve overall energy efficiency. Edge AI can be applied in energy consumption systems to process data at the Edge, and make local real-time decisions simultaneously, reducing the need for constant communication with remote servers.
- **Edge Computing in General:** Edge AI is a critical component of Edge computing, where processing is done closer to the data source rather than relying solely on remote Cloud servers. This is particularly important for applications that require low latency and real-time decision-making.
- **Natural Language Processing (NLP) at the Edge:** NLP applications, such as voice assistants and language translation, are increasingly being implemented at the Edge. This allows devices to interpret and respond to user commands without relying on constant internet connectivity.

- **Cyber Security:** AI is used for threat detection and anomaly detection in cyber security. Edge AI can enhance cyber security by monitoring and processing data locally to identify potential security threats in real-time.

Despite a diverse set of AI and Edge AI applications, there are limitations, as discussed in the next section.

4.8 Limitations of AI and Edge AI

AI and Edge AI both have their own limitations.

4.8.1 Limitations of AI

Several limitations persist in the field of artificial intelligence.

- **Lack of Common Sense and Generalization:** AI systems often struggle with common sense reasoning and generalizing knowledge across different domains. They may perform well on specific tasks they were trained on but can lack a broader context.
- **Data Dependency and Bias:** AI models rely heavily on the data they are trained on. If training data is biased or incomplete, an AI system can inherit and perpetuate these biases. This raises ethical concerns, especially in applications such as recruitment, financial lending, or criminal justice.
- **Interpretable AI:** Many AI models, particularly using deep neural networks, are often considered "black boxes" because understanding their decision-making process can be challenging. This lack of interpretability is a concern in critical applications where transparency is crucial.
- **Explainability and Accountability:** AI systems should be able to explain their decisions, especially in high-stakes applications like healthcare or finance. Establishing accountability for the actions of AI systems is essential for gaining trust.
- **Robustness and Adversarial Attacks:** AI models are susceptible to adversarial attacks. Small, carefully crafted changes to input data can lead to incorrect predictions. Ensuring the robustness of AI systems, especially in security-critical applications, remains a challenge.
- **Computational Resources:** Training and running complex AI models, especially deep neural networks, demand significant computational power. This can limit the accessibility of advanced AI technologies in resource-constrained environments.
- **Ethical and Societal Implications:** AI raises ethical questions, such as job displacement due to automation, biases in decision-making, and a potential misuse

of AI technologies. Formulating ethical guidelines and policies to address these concerns is an ongoing challenge.

- **Transfer Learning Challenges:** While transfer learning [35] has shown promise, the ability to efficiently transfer knowledge learned in one domain to another remains a challenge. AI models may struggle to adapt quickly to new, unseen scenarios.
- **Security Concerns:** AI systems can be vulnerable to attacks and manipulations. Ensuring the security of AI models and preventing unauthorized access or malicious use are ongoing challenges.
- **Resource and Energy Consumption:** Training large AI models requires substantial computational resources, leading to high energy consumption. Developing more energy-efficient algorithms is crucial for sustainable AI development.

- Researchers and practitioners continue to address these limitations, and the field of AI is dynamic, with ongoing efforts to enhance the capabilities, reliability, and ethical considerations associated with AI technologies. While these AI challenges exist, limitations on the Edge are exacerbated by a lack of computational power and storage in the Edge devices.

4.8.2 Limitations of Edge AI

Below are some common limitations of Edge AI:

- **Limited Computational Power:** Edge devices, such as sensors, cameras, or IoT devices, often have limited computational resources compared to powerful Cloud servers. This constraint can impact the complexity and speed of AI algorithms that can be deployed on the Edge.
- **Storage Constraints:** Edge devices typically have limited storage capacity. Storing and managing large AI models or datasets on these devices can be challenging, especially for applications that require extensive data processing, such as chatGPT and Bard.
- **Energy Consumption:** Edge devices are often battery-powered, and running resource-intensive AI algorithms can drain the battery quickly. Developing energy-efficient algorithms is crucial for the widespread adoption of Edge AI, particularly in remote settings.
- **Security Concerns:** Edge devices may be more vulnerable to physical attacks or unauthorized access compared to centralized Cloud servers. Ensuring the security of AI models and data on Edge devices is a significant challenge.
- **Model Update and Maintenance Challenges:** Managing and updating AI models on a large number of distributed Edge devices can be complex. Ensuring that all devices are running the latest models and updates is crucial for maintaining optimal performance and security.

- **Lack of Standardization:** There is currently a lack of standardization in Edge AI technologies. Different devices may have varying hardware specifications, communication protocols, and software frameworks, making it challenging to create universal interfaces that work seamlessly across all Edge devices.
- **Data Privacy Concerns:** Edge AI processes data locally, which can be advantageous for privacy. However, it also raises concerns about how sensitive data is stored on the device and whether there are potential privacy breaches. This is especially relevant in applications such as surveillance or healthcare.
- **Scalability Issues:** Scaling Edge AI solutions to a large number of devices can be challenging. Coordinating the deployment and management of AI models across a vast network of Edge devices requires efficient protocols and infrastructure.
- **Limited Connectivity:** In some remote scenarios, Edge devices may have limited or intermittent connectivity. This limitation can affect the ability to update AI models, receive real-time support, or share data with a remote server.
- **Trade-off between Accuracy and Resource Constraints:** Due to limited computational resources, Edge devices may need to trade off accuracy for efficiency. This compromise is essential for real-time processing but can impact the overall performance of certain AI applications.

It is important to note that ongoing research and advancements are continuously addressing these limitations. The field of Edge AI is evolving rapidly to overcome these challenges. As technology progresses, solutions to these limitations are likely to become more refined and accessible.

4.9 Summary

In this chapter, we have introduced AI and explained various AI classifications based on capabilities, learning, techniques, and ethical considerations. We then give a detailed historical evolution of AI. This is followed by coverage of AI computing environments, AI Edge computing, and AI analytics. We conclude by describing emerging applications in AI and various limitations of AI.

4.10 Points to Ponder

1. What is the key difference between AI and ML?
2. What type of problems are suitable for unsupervised learning to solve?
3. What are the advantages and concerns of using ML in a public Cloud?
4. What is the meaning of overfitting and why is it not desirable?
5. What are the differences between machine learning (ML) and deep learning (DL)?

6. What are the advantages of DL?
7. What are the drawbacks of DL?
8. If a Cloud service provider wants to offer ML at the IaaS layer, what will be the features of such a service?
9. If a Cloud service provider wants to offer ML at the PaaS layer, what will be the features of such a service?
10. If a Cloud service provider wants to offer ML at the SaaS layer, what will be the features of such a service?

4.11 Answers

1. What is the key difference between AI and ML?

    - Artificial intelligence (AI) is the broader concept of machines acting in a smart manner, such as for playing chess. Machine learning (ML) is a subdomain of AI based on the idea that, given a sufficient amount of data, machines can learn the rules without explicit instructions or programming. Thus, ML is one of the ways for machines to develop AI capabilities.

2. What type of problems are suitable for unsupervised learning to solve?

    - In unsupervised learning, machines find meaningful relationships and patterns in a given dataset. It is useful when labeled data is not available, in cases such as finding groups or clusters, extracting generative features, or for exploratory purposes.

3. What are the advantages and concerns of using ML in a public Cloud?

    - Cloud-based machine learning is good for applications that need to analyze large quantities of data. If that data is coming from many different sources over a period of time, then Cloud storage is an attractive place to store that data for the long term. An example of such cases is the Internet of Things (IoT) and healthcare diagnostics. However, some businesses have privacy and security concerns about storing their data in a public Cloud, for example, all medical data storage and transmission must abide by Health Insurance Portability and Accountability Act (HIPAA) protocols.
    - A new research area of federated learning is emerging to address these concerns while still being able to use ML in a public Cloud.

4. What is the meaning of overfitting and why is it not desirable?

    - Overfitting is a modeling error that occurs if a function is defined to closely fit to a limited set of data points during the training phase. It makes a model conform to mimic a dataset that may not be fully representative of other data points that the model may encounter in the future. Thus, it may result in substantial errors when a model is used for inference.

5. What are the differences between machine learning (ML) and deep learning (DL)?

    - Recall that ML is a branch of AI that can self-learn based on a given dataset and improve its decisions over time without human intervention. Similarly, DL is a sub-branch of ML, which can be applied to extremely large datasets. The word deep comes from multiple layers in an artificial neural network (ANN).

6. What are the advantages of DL?

    - DL breaks down a complex problem in stages and uses layered solution processes to create an artificial human brain-like structure to make intelligent decisions. Each layer in the DL system represents a stage where parameters can be tuned for making complex decisions, e.g., for Netflix to decide which movie to suggest next based on past viewing habits of a subscriber.

7. What are the drawbacks of DL?

    - DL requires a very large amount of data to perform better than other AI solutions. A DL model is computationally expensive to train due to a complex data model with many variables. It may require expensive GPUs and other specialized hardware machines. A DL system is hard to debug in case of any errors in the output.

8. If a Cloud service provider wants to offer ML at the IaaS layer, what will be the features of such a service?

    - A user of IaaS wants to avoid capital expenditure and uses Cloud facilities to pay for it on a per-use basis. IaaS users are mostly concerned with the quality of service in terms of a Cloud server's compute, memory, and network latencies. ML in IaaS can be used to track the response time for users' hosted applications, enabling them to schedule tasks so as to maximize their compute efficiency. Any idle servers can be shut down and workloads consolidated to maximize the utilization of the running servers. New servers can be started up as users' workload demands increase. ML can be helpful for tracking usage metrics, predicting costs when scalability is needed to maintain a constant Quality of Service (QoS), etc.

9. If a Cloud service provider wants to offer ML at the PaaS layer, what will be the features of such a service?

    - Users of PaaS are mostly concerned about the specific tasks related to their hosted services, such as database I/O transactions and customer activities on their hosted Web sites. Examples of PaaS are a Python integrated development environment or MatLab tools and facilities. ML can be helpful for generating metrics related to end user experiences such as search times for catalog items, wait times, and optimizations related to other services such as payments.

10. If a Cloud service provider wants to offer ML at the SaaS layer, what will be the features of such a service?

    - A user of SaaS, such as Netflix or Salesforce, may be mostly concerned about the statistics and preferences related to their applications' end users. These services need to be provided on an expeditious basis. ML can be helpful to track the SaaS customers' preferences and make suggestions based on AI models and past behavior to suggest new movies, etc.

References

1. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9955430/
2. https://www-formal.stanford.edu/jmc/ailogic.pdf
3. https://en.wikipedia.org/wiki/Artificial_general_intelligence
4. https://plato.stanford.edu/entries/turing-test/
5. http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf
6. https://spectrum.ieee.org/dartmouth-ai-workshop
7. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1307157/pdf/westjmed00160-0094.pdf
8. https://www.almabetter.com/bytes/tutorials/artificial-intelligence/search-algorithm-in-ai
9. https://www.almabetter.com/bytes/tutorials/artificial-intelligence/heuristic-function-in-ai
10. https://www.dtreg.com/methodology/view/decision-trees-compared-to-regression-and-neural-networks
11. https://ieeexplore.ieee.org/document/8694781
12. https://www.officetimeline.com/blog/artificial-intelligence-ai-and-chatgpt-history-and-timelines
13. https://en.wikipedia.org/wiki/History_of_artificial_intelligence
14. https://en.wikipedia.org/wiki/Graphics_processing_unit
15. https://en.wikipedia.org/wiki/Tensor_Processing_Unit
16. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). *Cloud computing with security and scalability* (3rd ed.). Springer.
17. https://jelvix.com/blog/top-5-big-data-frameworks
18. https://prace-ri.eu/wp-content/uploads/Edge-Computing-An-Overview-of-Framework-and-Applications.pdf
19. https://en.wikipedia.org/wiki/EdgeX_Foundry
20. https://sdn.ieee.org/newsletter/march-2016/mobile-edge-computing-an-important-ingredient-of-5g-networks
21. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). *Cloud computing with security: Concepts and practices* (2nd ed.). Springer.
22. https://aithority.com/technology/transforming-businesses-key-components-of-ai-orchestration-and-how-it-works/
23. https://www.researchgate.net/publication/353274282_Networking_models_and_protocols_foron_edge_computing
24. Kumar, S. N., & Pramod, G. (2021). *Introduction to machine learning in cloud with python: Concepts and practices*. Springer.
25. https://www.marketsandmarkets.com/Market-Reports/edge-ai-software-market-70030817.html
26. https://www.anodot.com/learning-center/ai-analytics/
27. https://towardsdatascience.com/text-sentiment-analysis-in-nlp-ce6baba6d466?gi=3e333a94ca68
28. https://www.thisisdmg.com/en/ai-powered-chatbot/

29. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7594889/
30. https://www.linkedin.com/pulse/optimization-algorithms-ai-techniques-design-planning-khamis-phd-pjovc/
31. https://www.mdpi.com/1424-8220/20/22/6441
32. https://www.machinemetrics.com/blog/edge-analytics
33. https://www.sciencedirect.com/science/article/pii/S2667345223000196
34. https://en.wikipedia.org/wiki/Quantization_(signal_processing)
35. https://www.almabetter.com/bytes/articles/transfer-learning-in-deep-learning

Chapter 5
Foundations of Information Security

5.1 Information Security Background

Today's information technology environment contains a wide variety of computing devices and multiple communication channels between various participants. Economics drove the creation of large datacenters, and Cloud computing was devised to distribute this enormous computing power. As the capability of inexpensive computing devices continued ahead of the communications capabilities, computational power has moved back to the end nodes of wide area networks. The age of Internet of Things (IoT) arrived a decade ago, as demonstrated by the fact that more things are now connected to the Internet than people in the world [1]. The "things" connected to Internet include sensors, controllers, and intelligent devices [2]. These devices have limited power to pose security problems but have an even more limited ability to provide any security solutions. To date the biggest security breaches in the IoT world have been instructions sent to the IoT devices, which are then used to launch massive denial of service attacks on central servers. The top three examples are Mirai, Hajime, and Persirai codes [3].

Information security can be viewed as composed of three functions, namely, access control, secure communications, and protection of private data. Alternatively, a common three-pillar split is confidentiality, integrity, and availability (the CIA triad of security policies and objectives). Sometimes, information security is shortened to INFOSEC. Access control includes both the initial entrance by a participant and the reentry of that participant, followed by the access of additional participants. Note that a participant can be an individual or some computer process. The secure communications include any transfer of information among any of the participants and devices. The protection of private data includes storage devices, processing units, and even cache memory [4].

The first function encountered is access control, i.e., who can rightfully access a computer system or data. The access control can be resolved at a hardware level

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_5

with a special access device, such as a dongle connected to the USB port or built-in security keys. Access control is usually addressed at the operating system level with a login step. An example of access control at the application level is requiring login and password.

After access control is granted, secure communication is the next function, which requires encryption. The most commonly recognized function of a secure system is the encryption algorithm, and the most common problem in a secure system is the encryption key management. At the hardware level, the communication encryption device can be implemented at the I/O port. At the operating system level, encrypted communications can be implemented in the secure driver software. At the application level, the encryption algorithm is implemented in any routine performing secure communication.

Some of the other functions and issues for Edge security systems are hashing (for checking data integrity), identity authentication (for allowing access), electronic signatures (for preventing revocation of legitimate transactions), information labeling (for tracing location and times for transactions), and monitors (for identifying potential attacks on the system). Each of these functions affects the overall security and performance of a system. The weakest security element for any function at any level limits the overall security and risk. In addition to accepting the security process, an Edge computing user may have concerns regarding the protection of private data. Protection of data includes both limiting availability to authorized recipients and integrity checks on the data. The level of security required is not universal. Ease of access is more important for low-security activities, such as remotely turning on light bulbs. More difficult access is required for medium security, such as controlling the environment in a building. High security is required for high-value tasks, such as controlling entry doors in a building or operating industrial equipment on a factory floor. Very strict and cumbersome access procedures are expected for nuclear weapon applications. These examples provide a clue to security in an Edge Cloud computing environment with shared resources [5]. Specifically, in the same computing environment, different applications are running at a variety of security levels. Security solutions must also consider the trade-offs of security versus performance. Some straightforward increases in the security cause inordinate degradation of performance. As described previously, the security implementations can be done at multiple levels for each of the functions. Because security is a multifunction, multilevel problem, high-level security operations need access to low-level security measurements. This is true in monitoring both performance and security. The current environment requires any Cloud user or provider to consider security in many places. Such security considerations have performance impact and many trade-off points, as depicted in Fig. 5.1.

Impact of security is a measure of drop in the performance due to computation overhead of encryption and decryption. As shown, the performance cost of full data and data encryption is very high but needed for highly sensitive data in a public Cloud. On the other hand, access control and login passwords may be sufficient for a single-user access device, such as a surveillance camera at home. Then hash checking and secure handshakes, using techniques such as virtual private networks

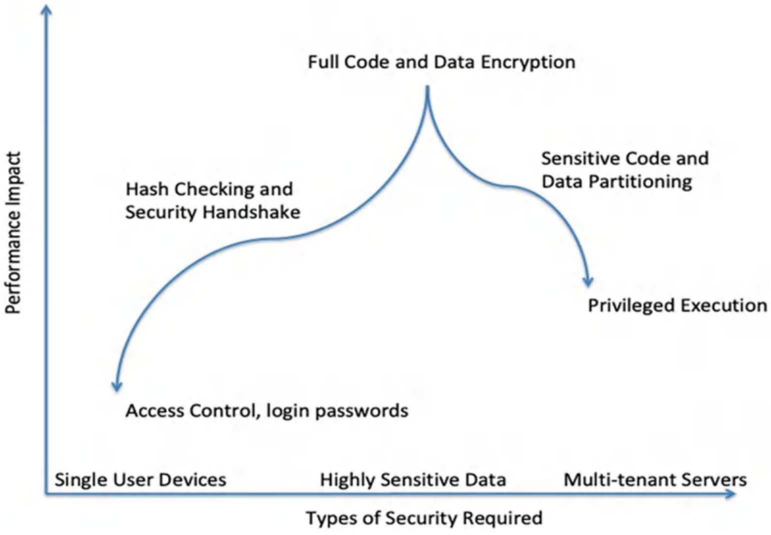

Full Code and Data Encryption
Hash Checking and Security Handshake
Sensitive Code and Data Partitioning
Privileged Execution
Access Control, login passwords
Performance Impact
Single User Devices
Highly Sensitive Data
Multi-tenant Servers
Types of Security Required

Fig. 5.1 Trade-offs between security required and its performance impact

(VPN), between an Edge device and Cloud may be considered sufficiently secured. If there is a desire to balance security and performance, then sensitive data and code can be partitioned to run in a secure environment in a public Cloud.

Three environmental factors directly affect the evolution of information security: computing power available, growing user base, and sharing of Edge resources. The first factor has been and continues to be the computer power available to both sides of the information security battle. Computing power continues to follow Moore's law with increasing capacity and speeds increasing exponentially with time. Therefore, while the breaking of a security system with brute force may take many years with the present computer technology, in only a few years quantum computer capacity may be available to achieve the same break-in in real time. The second environmental factor is the growing number of people accessing Edge devices. The world has changed from a relatively modest number of financial, governmental, business, and medical institutions having to secure information to nearly every business and modern human needing support for Edge-based information security. The sheer number of different people accessing Edge devices has increased the importance of different levels of security. The third environmental change that has a significant impact on security is the sharing of devices on the Edge of a network, which is the crux of this chapter.

5.2 Evolution of Edge Security Considerations

Edge computing represents a combination of distributed computing connected to centralized servers. Historically, centralized versus distributed models have alternated as computing and communication capabilities have grown, while the limiting factor has alternated between computational capability and communication capacity. The present environment of Cloud and Edge computing is a complex mixture of computing capability, communication capacity, and security considerations. In this chapter, we will focus on the security aspects of Edge computing. Any such investigation must include multiple subtopics, e.g., protecting information content from observation and alteration, protection of operational capability from unauthorized access, protection of normal operation in the presence of malicious overloaded requests, etc. Solutions need to consider prevention from and response to any security threats [5].

Examples of prevention include encryption to protect content from observation and alteration, access checking protocols to prevent unauthorized accesses, tracking mechanisms to identify attempted attacks, and blocking messages except from trusted devices.

In the past, information communication was routinely, although not universally, protected with encryption. As the evolution of computation leads to Edge-based Cloud computing, now everything is accessible to everyone. The physical boundaries are gone. When connected to and using the Cloud, providers and users can be using resources anywhere. In fact, that is the goal for Cloud computing: to separate delivering the desired service from the underlying implementation of the capability to deliver the service. This makes a huge difference in security, as nothing can be assumed to be secure. In addition to the protection of information, now there is a new security problem of protecting access to Edge devices. An unauthorized user can attempt to access a resource with enough effort that it interferes with the usage by authorized users. This is called a denial-of-service attack. For example, excessive requests to a Web server can cause a Website to crash. This type of attack, barring physical attacks, did not exist in earlier computer environments. Another security issue is privacy. The information required to access and utilize one device should neither reveal that information to unauthorized parties nor to unauthorized use by parties authorized to have that information. Additionally, the environment includes the security risk of attackers falsifying or changing information. This includes changing messages in transit or initiation of false messages. This requires an electron signature or checksum to detect any data tampering.

In parallel to the changing environment affecting the evolution of Edge security considerations, the changes in performance directly affect security considerations. When communication and performance were slow, minimum information security was required because the attackers had simple tools to get access. Simple tools included automated password guessing or brute force encryption hacking. Due to a low level of computational performance, all of these techniques were far too slow to be a serious threat. One method to increase security is to increase the encryption key size. The key size is limited to the performance used to encrypt, as larger keys require more computation. On the other hand, the security of the key size must be large enough to exceed the performance available to the attackers. As Edge-based computational capacity increases, the keys must get bigger because the attackers have more capacity, and the keys can get bigger because the devices also have more capacity. Thus, for encryption, there is a continuous race between defenders and attackers. The Edge computing environment oscillated between when the communication channel was the performance limiter and when the computational capability was the limiting factor. In the present world of Edge-based Cloud computing, the attackers have access to significant computational ability; therefore, security is now a universal serious problem.

5.3 Edge Security Players

Traditional computing environments had a clear delineation between “inside” and “outside.” Physically, “inside” might be in Alice’s office or “inside” the bank building. With the dawn of networks, and especially the Internet, the networks were partitioned as “inside the firewall” and “outside,” which could be anywhere. This is one of the differences between a public Cloud and a private Cloud. Secure communication was only needed when the communication went from “inside” to “outside.” With Edge-based Cloud computing, “inside” is not clearly defined as computers in multiple devices and data centers across different geographies can be pooled together to appear as a large virtual pool.

To visualize a wide variety of elements and security requirements in the IoT domain, consider Fig. 5.2. The standard Internet communication security approach (including virtual private networks, i.e., VPN) is to establish a link between Alice and Bob using access control to identify the authorized individuals and then to use encryption for information exchange between the “islands” of security containing Alice and Bob. Alternatively, Dave may want to do a remote transaction with his bank. Dave’s transaction requires a higher level of security than Dave’s normal activities. Carol may want to turn on her light bulbs at home since she will be arriving after dark. While this does not require a high level of security, Carol certainly does not want some random person turning her lights on and off. Other examples of low levels of security are household appliances, such as a toaster or a refrigerator. The high levels of security examples include remotely opening a home garage, accessing banks, or operating factories.

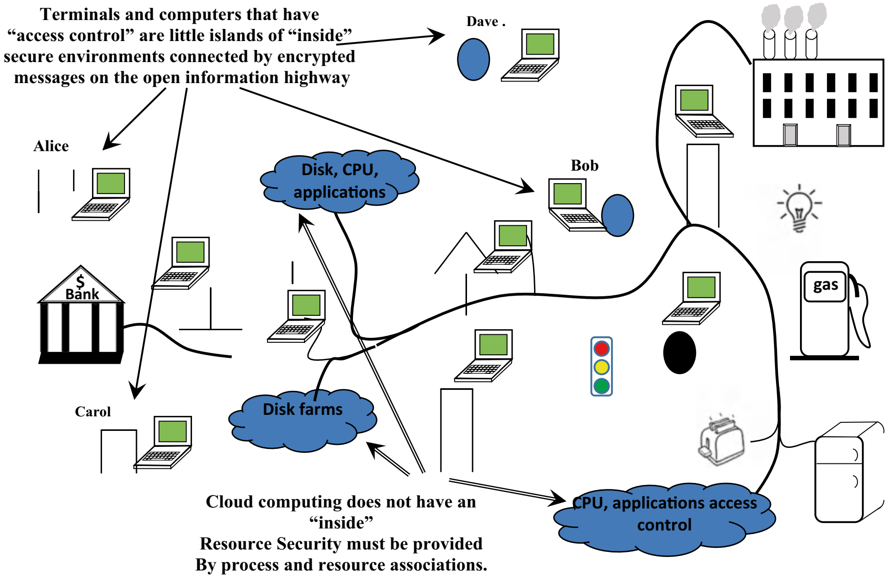

Terminals and computers that have
"access control" are little islands of "inside"
secure environments connected by encrypted
messages on the open information highway
Dave .
Alice
Bob
Disk, CPU, applications
$ Bank
gas
Carol
Disk farms
Cloud computing does not have an
"inside"
Resource Security must be provided
By process and resource associations.
CPU, applications access control

Fig. 5.2 Information security on the Edge of a network

5.4 Edge to Cloud Secure Communications

In the era of Edge computing, another consideration is due to multiple connection paths for each device. Each element on the Edge can connect using a choice of paths or even multiple paths between the same endpoints. Specifically, any computing element on the Edge can connect via the Internet, telephone lines, cell phone connections, wireless local area service networks (Wi-Fi), or local wireless point-to-point connections such as Bluetooth or near-field communication). See Fig. 5.3 for multiple paths from Alice to Bob, to a local server hub, to the Internet, or to the house alarm system. Edge computing continues to mature and encompass more of our world. Standards are being created, such as Waggle [6], which is an open sensor platform for Edge computing, which has been introduced to reduce some of the foreseen compatibility problems. Edge computing security issues encompass end-to-end devices and the networks in between.

The common protocol for communication on the Internet is the Hypertext Transfer Protocol (HTTP) for basic nonsecure communication. A client contacts the server, the server sets up a channel, and then communication continues between the client and server as shown in Fig. 5.4.

However, at the present time, almost all communication uses the secure version, which is Hypertext Transfer Protocol Secure (HTTPS). For HTTPS, the client makes an initial contact the same way as with HTTP [5]. However, now the client and server must establish a secure link. For efficiency, the secure link uses symmetric encryption. For symmetric encryption, both parties must have the same key. This

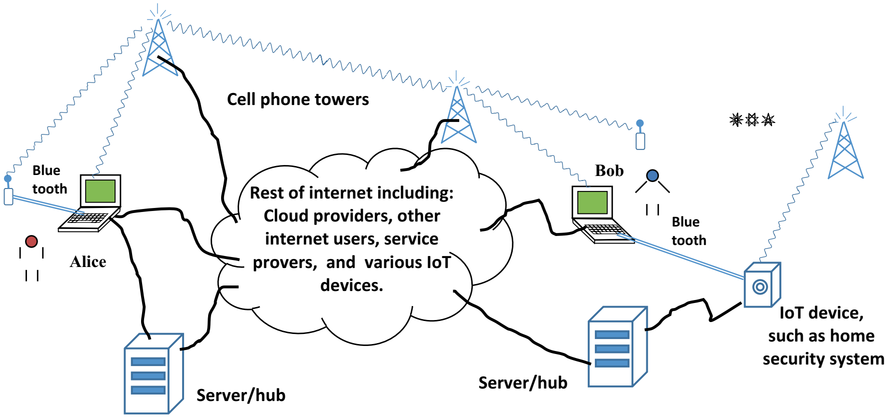


Fig. 5.3 Communication connectivity from the Edge

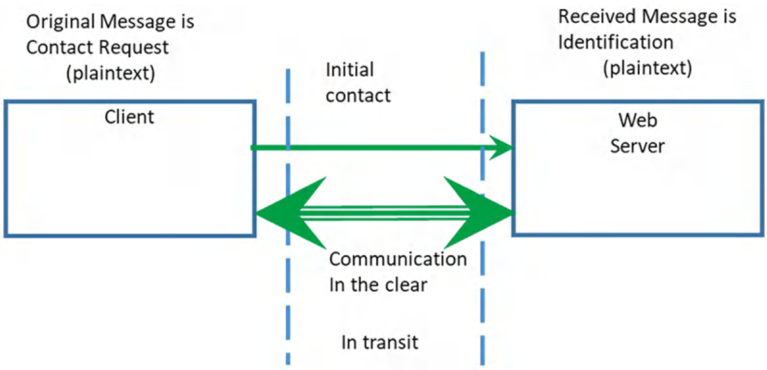


Fig. 5.4 HTTP nonsecure communication in the Cloud

is a shared secret requiring a secure mechanism to exchange the secret key securely. Asymmetric encryption can be used to exchange the secret key. The server generates a public/private key pair. In asymmetric encryption for communication, the public key is used to encrypt and the private key is used to decrypt. The server makes available the public key so that anyone can encrypt a message. However, the server keeps the private key secret so only the server can decrypt and read the message. This is shown in Fig. 5.5. The client generates a secret key for symmetric encryption to be used just for this one session. The client sends this key to the server using the server's public key for asymmetric encryption. Both the client and the server now have the secret session key and use it for the rest of the secure communications between the server and the client. After the session is over, the private key is discarded.

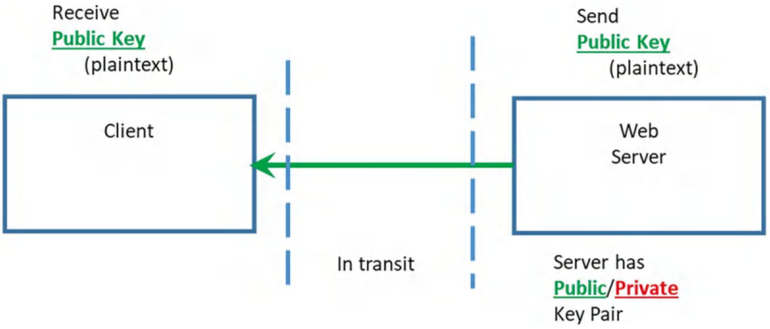


Fig. 5.5 HTTPS secure communication using server-provided public key

5.5 Edge Security Storage and Computations

The trade-offs between performance and security described for transmission also apply to storage and computation. A common solution for Edge security and integrity checking of networked storage environments is encrypted data file systems [7]. The cryptographic file systems (CFSs) are a significant performance burden. Using a CFS is especially needed when the data storage is farmed out to untrusted storage sub-providers [8]. A big difference is the wide range of storage lifetime. For storage such as copyrighted movies on DVD, there is a longtime value (several months or even years); however, for storage such as main memory, there is a short-time value (perhaps microseconds). The emphasis in the main memory security should be on read-write efficiency. A small loss of time here has a huge impact on the performance of a computer system due to repeated operations.

Hence lighter and faster encryption schemes can be applied to data of ephemeral value, such as being held in a system memory, which will be short-lived. For long-term data, companies such as financial institutions have invested in hardware security modules (HSMs). A HSM is a physical computing device acting as a vault to hold and manage digital keys for strong authentication and provides crypto processing services. This can be a plug-in card or an external device, attached directly to a network server. These HSMs are certified as per the internationally accepted standards, such as Federal Information Processing Standard (FIPS) in the USA, to provide users with a security assurance.

5.6 Side-Channel Security Attacks on the Edge

Edge computing enables unexpected attacks because physical hardware alone does not define the security boundary and is normally not in the control of the remote user. Direct attacks attempt to get a user's information by attacking the secured data directly. Direct attacks include attempting to decrypt by guessing keys with repeated trials or by attempting to gain access with password guessing. Side-channel attacks (SCA) attempt to gain information by looking at peripheral locations or measuring usage effects rather than the user's data itself [9]. Traditionally, side-channel attacks were not significant because they required some level of access to the secured hardware. In the IoT, the hardware is everywhere, and in the Cloud hardware is shared, providing remote access to hackers.

The side-channel attacks (SCA) can be grouped into several categories [5]. The first category is cache side-channel attacks. In this type of attack, the attacker attempts to access memory locations that are outside its authorization. While separate processes are assigned different segments of main memory, the data must pass through the same cache for both processes. Of course, when data is in the cache, it is assigned a memory tag associated with its own process. However, depending upon the cache coherency algorithm, the cache may not be cleared between process swaps. Also, the bound checking implementation limits what can be learned about the data in the cache. However, with some speculative executions or branch prediction algorithms, the bound checks are suspended to improve run times. It opens up an avenue to carry out successful SCA.

The next category of attack is the timing attack. Here the attacker monitors the target process to measure how long certain operations take. This can be a cache timing tack or a calculation timing attack. For a calculation timing attack, some algorithms for multiplication increase performance by doing a bit-by-bit add and skip or just skip the add step when the bit is 0. Thus the timing attacker has a sense of how many (and sometimes when) bits are zero in a key. This can significantly reduce the search space for potential keys, making them easier to guess.

The next category of attack is the power-analysis attack. This is an indirect attack because it is not looking at the binary data, although it is using the binary data. For example, the pattern of the power supply current is compared for guessed keys to the normal operation. This happens because the electrical charge required for transistors to turn on is higher than to keep them off, which is for binary 1 or 0, respectively. When there is a match in the power usage pattern, then the key has been found. Another category of attack is fault side-channel attack. Here the security algorithm or hardware has an intentional error injected while encrypting. The change in the resulting performance gives a hint for the secret information.

A huge problem for Edge computing is the variety of systems employed in a data center. Thus, while one component may be protected against a particular side-channel attack, other components will not be protected. This is exacerbated by the legacy hardware problem, which in widely connected Edge computing environments, the IoT includes systems with the latest security precautions and many more old systems lacking sufficient security implementations. One needs to do end-to-end penetration testing in a Cloud deployment to ensure its security.

5.7 Hardware-Based Security Solutions

Hardware-based security can occur on the Edge device as well as the central server in a datacenter. This, including encrypted communications from the Edge to a datacenter, will ensure an end-to-end security profile. Trusted computing in Fig. 5.6 refers to a setup where users trust the manufacturer of hardware or software on a remote computer and are willing to put their sensitive data in a secure container hosted on that computer [11] in the datacenter.

As we noted in a previous section, using hardware as a root of trust in a Cloud environment increases Cloud customers' confidence, as their VMs are running on the known and attested remote servers. Furthermore, such servers may also be running other tenants' VMs in a shared pool of resources. Thus, our security-conscious customers want to ensure that there is no in-memory attack or inadvertent data corruption from other tasks. Both confidentiality and integrity of sensitive data can be protected using special hardware features now beginning to be available.

One such example is Intel's Software Guard Extension (SGX), which provides a set of security-related instructions built into the latest CPUs [10]. These allow user-level as well as privileged OS or Virtual Machine Monitor (VMM) code to define

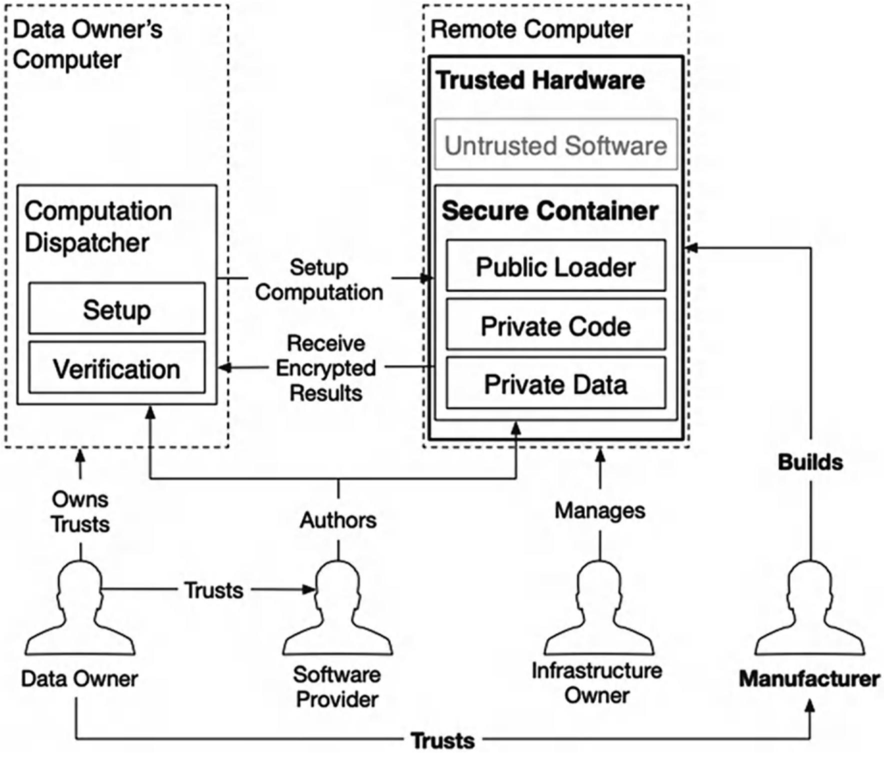


Fig. 5.6 Basis of trusted computing [10]

private regions of memory called enclaves. These enclaves are used to protect code and data, which can't be read or saved by processes outside the enclave. This includes even processes running at higher privileged levels. The enclave is decrypted at runtime only within the CPU package and only for the code and data running from within the enclave itself.

In Amazon's EC2 Cloud, server platforms can execute software at four different privilege levels, as shown in Fig. 5.7. This is based on a ring structure, akin to a scout camp, such that the innermost, or Ring 0, is most secure. Software running at a less privileged level, such as Ring 3, can freely read or modify the code or data. System management mode (SMM) at the top is used by motherboard manufacturers to implement protected regions of BIOS (basic input-output system). VMX refers to virtual machine extensions to support hypervisors or VMMs. VMX non-root is where the guest operating system runs in Ring 0 and user applications in Ring 3.

An enclave's code execution always happens in protected mode, at Ring 3, and uses the address translation setup by the OS kernel or a VMM. Even to service an interrupt, to protect the private data, the CPU must perform an asynchronous enclave exit to switch from the enclave context to regular Ring 3. Then it services the interrupt, fault, or VM exit. CPU saves the state into a predefined area inside the enclave and transfers control to a pre-specified instruction outside the enclave. After servicing the external system call, the CPU switches the state back to an enclave, restoring the register values and flags, etc.

The first step in any secure computing is certification or attestation. It proves to remote users that they are communicating with a specific trusted platform. This reduces the probability of a man-in-the-middle attack. Proof of attestation is a

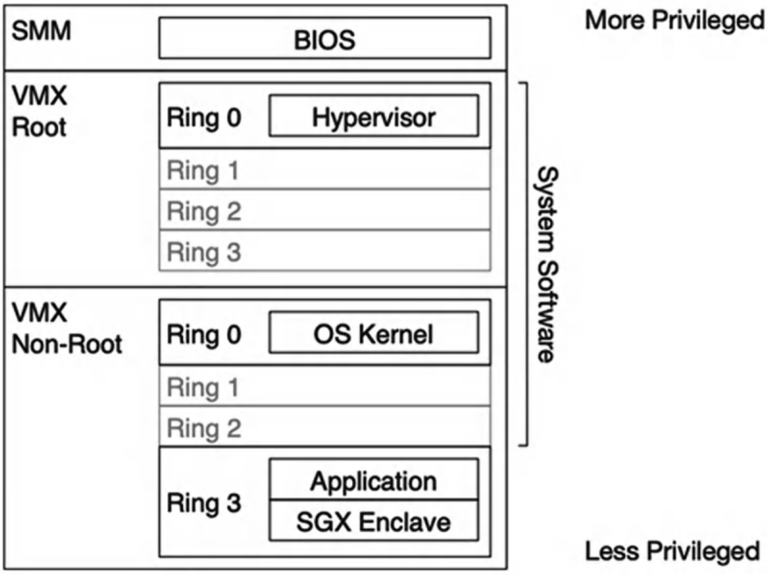


Fig. 5.7 Privilege levels in a server platform [10]

signature produced by the platform's secret and unique attestation key, as shown in Fig. 5.8. It convinces the remote users that their sensitive code and data will reside in a secure container or enclave. Execution flow can enter an enclave only via special SGX instructions, similar to the mechanics for switching from the user mode to kernel mode. Thus, the trusted compute boundary (TCB) for the SGX threat model is limited only to the processes resident in an enclave. In other words, anything outside of an enclave including the OS or VMM is excluded from the TCB. A code or process outside the enclave trying to read it will see cipher or encrypted text. SGX is useful for implementing secure remote computation, secure Web browsing, Digital Rights Management (DRM), etc. Other applications can conceal security keys or proprietary algorithms using SGX.

Edge-based Cloud customers want their data to be protected at rest (in storage), during transit (during transportation), and at run time (in execution). SGX enables users to protect their data while it is being processed in the Cloud.

As of this writing, Microsoft's Azure Cloud [12] provides confidential computing using Intel's SGX capability. It provides trusted execution environment (TEE), which enables users to:

- Safeguard information from malicious and insider threats while in use.
- Maintain control of data through its lifetime.
- Protect and validate the integrity of code in a public Cloud.
- Ensure that data and code are opaque to the Cloud platform provider.

Another US public Cloud provider using SGX is IBM [13]. They reported that while external attacks outnumber internal incidents as causes of breaches, internal security incidents are on the rise. In 2022, 46% of attacks were malicious insider

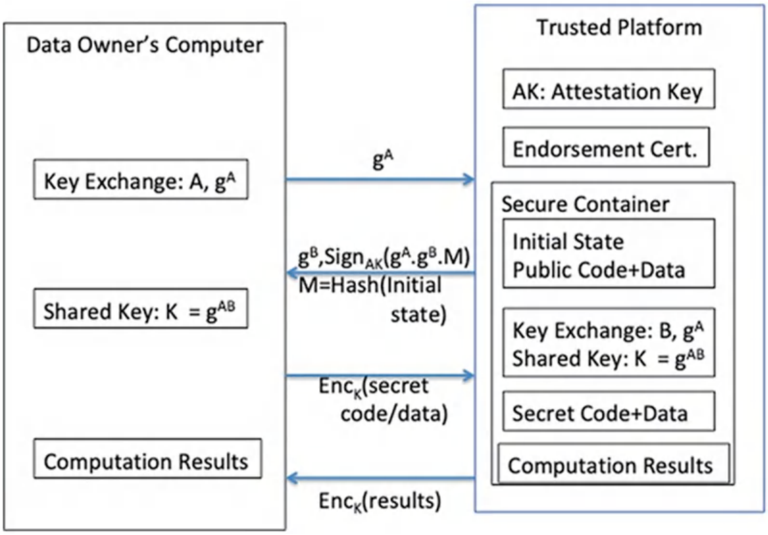


Fig. 5.8 Attesting the authenticity of a trusted platform to a remote user [11]

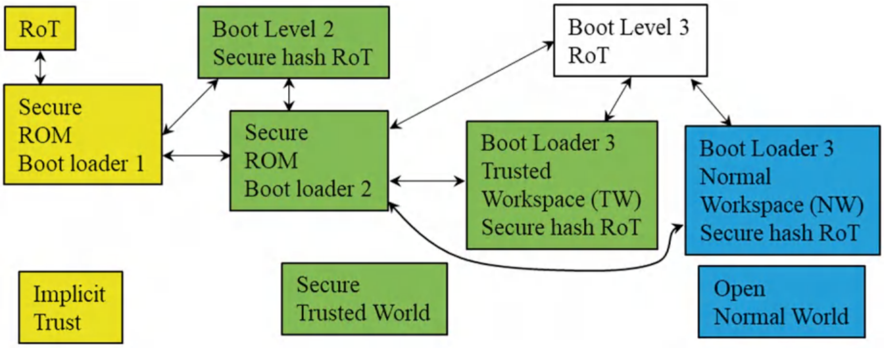

RoT
Secure ROM Boot loader 1
Boot Level 2 Secure hash RoT
Secure ROM Boot loader 2
Boot Level 3 RoT
Boot Loader 3 Trusted Workspace (TW) Secure hash RoT
Boot Loader 3 Normal Workspace (NW) Secure hash RoT
Implicit Trust
Secure Trusted World
Open Normal World

Fig. 5.9 Arm trust zone attestation at boot time [14]

incidents. IBM Global Cloud deploys SGX-enabled non-virtualized servers, also known as bare metal servers, to provide run-time protection for users' sensitive application code and data.

Alternatively, let us consider attestation in the ARM architecture [14]. ARM CPUs are used in many Edge-based devices. Local attestation for ARM TrustZone is dependent on several measurements taken during the boot process and requires a secure boot. During power-on, implicitly trusted code resident in a secure Read Only Memory (ROM) or Static Random Access Memory (SRAM) is executed. The boot loading process occurs in 3 stages, as shown in Fig. 5.9. Boot loader 1 is responsible for authenticating the boot loader 2 stage. Boot loader 1 verifies the root of trust (RoT) public key in the boot loader 2. Boot loader 1 then verifies the boot loader 2's content certificate using the enclosed RoT public key. Boot loader 1 loads boot loader 2 into memory and verifies the hash. Execution is then transferred to boot loader 2. Boot loader 2 is responsible for authenticating all of the possible boot loader 3 stages. Boot loader 2 verifies the RoT public key in the certificate against the RoT public key stored in the hash. Boot loader 2 then verifies the certificate using its RoT public key and saves the trusted world (TW) and normal world (NW) public keys. Boot loader 2 uses the TW public key to verify the boot loader 3's certificate and verifies the boot loader 3's content certificate using the boot loader 3's public key. Boot level 2 extracts and saves the boot level 3 hash used for boot level 3 image verification. Finally, execution is transferred to the verified boot level 3 images. During execution, both TW and NW coexist. After the secure boot, any secure monitor software should be loaded to run in the highest privilege level [15].

ARM's approach using the multistage boot provides a higher level of security than a single stage of attestation, as in SGX. The rationale here is similar to multifactor authentication (MFA), which relies on the fact that it is harder to breach the security of multiple devices or locations. With a multistage boot, a hacker would have to replace boot code at multiple locations.

5.8 Security Practices for Edge Computing

While Edge Cloud computing and security practices continue to evolve, many users have already migrated their mission-critical applications to the Cloud driven by economic value and convenience factors [5]. This has made both public and private Cloud attractive targets for security hackers. Hence, we propose the following practices for the users and practitioners of Edge Cloud computing to ensure that assets remain secure:

1. *Continuous Monitoring:* This is needed for any unexpected usage patterns or changes in your Cloud resources. You can't protect what you can't see.
2. *Attack Surface Management:* It refers to the set of access points that are exposed to an unauthorized user. An organization needs to limit the devices or methods that can access its mission-critical data [16]. Besides the obvious methods of using encryption, one needs to ensure that the devices that are authorized to access this data themselves are not vulnerable or hacked.
3. *No Residual Footprints:* Looking into bins for any trashed paperwork is an old spying practice. The online equivalent of this is to try reading the leftover bits in the memory or disk after a target VM stops using these resources in a Cloud. By zeroing out its contents of memory and disk, a VM upon exit can ensure that the next VM will have no residual data to exploit. This operation may cost some extra time and money but is well worth the trouble of avoiding your valuable data falling into wrong hands.
4. *Strong Access Control:* While obvious to any IT manager, many recent attacks came through unexpected entry points on the Edge. Many companies use Internet-connected heating, ventilation, and air-conditioning (HVAC) systems without adequate security, giving hackers a potential gateway to the key corporate systems. An example [17] shows how hackers stole login credentials belonging to a company that provides HVAC services and used that access to gain a foothold on another target company's payment systems. A strong chain is as weak as its weakest link, so analyze your system and its access points to find its most vulnerable spots.
5. *Damage Controls:* With always evolving sophisticated hacking techniques, no system is 100% hack-proof, and it is not a question of if but when a security attack can happen on your Cloud infrastructure or data. Each organization and user needs a plan to minimize the damage in such cases. For an individual, it might be a matter of canceling their credit cards, changing banking passwords, or perhaps closing some online accounts if they are compromised. For an organization, mitigation strategies may be more complex, involving an alternative control and command network or quickly shutting down infected servers, etc. [18].

As security intelligence shows [19, 20], available technology is not restricted to firewalls. Any Edge computing solution must protect the access in the most effective way possible, such as by including the following capabilities:

- Intrusion detection tools
- Application firewall
- New generation firewall
- Attack mitigation tools for distributed denial of service (DDOS) attacks
- Log correlation

5.9 Machine Learning for Security

From previous chapters, we learned that machine learning (ML) techniques are being applied in many areas of life, such as smart grids, factory floors, and automobiles. Since cyber security incidents across all these domains are also growing, there is a need to apply ML techniques to curb unauthorized transactions.

In any transaction involving humans, the most common cybercrime is phishing [21]. It is an attempt to fraudulently obtain personal information such as passwords and bank or credit card details by posing as a trustworthy entity. There are some excellent ML-based solutions in the following three areas:

1. *Detective Solutions:* By monitoring an Edge device's activities and flagging any unusual transactions. Incoming packets can be prevented by denial of service (DOS)/distributed denial of service (DDOS) attack detection. Web content can be filtered using anti-malware software. ML can help to strengthen these techniques by training with known vulnerabilities and then looking for some unexpected patterns.
2. *Preventive Solutions:* Incoming login requests to an Edge device can be checked using multifactor authentication. In addition to usual techniques such as sending one-time password (OTP) verification codes by email or mobile addresses, ML techniques can be used to avoid SIM hijacking attacks [22].
3. *Corrective Solutions:* After recognizing repeated attacks from a particular set of IP addresses, or geographies, it is possible to detect the location of phishing sites and pull them down. ML techniques can be used for forensics and investigation. A comparison of the following six ML classifiers pegs their error rates between 0.075 and 0.105; the detailed discussion can be found at reference [23]:
   (a) Logistic regression (LR)
   (b) Classification and regression trees (CART)
   (c) Bayesian additive regression trees (BART)
   (d) Support vector machines (SVM)
   (e) Random forests (RF)
   (f) Artificial neural networks (ANN)

Interestingly, hackers are also using ML techniques. An example is the usage of machine learning tools to break human interaction proofs (HIP aka CAPTCHA). CAPTCHA stands for completely automated public Turing test to tell computers and humans apart. It is a challenge-response test used in computing to detect if the user is a human. This involves distorted characters often hidden, so it becomes a recognition challenge for a regular computer. However, the ML approach trains by using the segmentation steps of hidden characters and then uses neural networks to recognize characters in new challenges [24]. Historically, the training stages were computationally expensive due to complex segmentation functions. With the advent of Edge-based Cloud computing and the elasticity of affordable resources, hackers can now overcome the difficulties in the identification of valid characters. This requires the bar to be raised for public-facing Edge devices connected to the Cloud.

5.10 Summary

The present state of Edge computing is an environment of vastly different computing capabilities connecting via a wide variety of communication paths. This situation creates both great operational capability opportunities and unimaginable security problems.

Edge security issues exacerbate with the growth of the Internet as more people and devices join the Web, opening new ways to compromise an ever-increasing amount of information and potential for damages. However, an even bigger challenge to information security has been created with the implementation of remote access via Cloud computing. This chapter gave a brief general description of information security issues and solutions. Some information security challenges that are specific to Edge Cloud computing have been described. Security solutions must make a trade-off between the amount of security and the level of performance cost. The key thesis of this chapter is that any security solutions applied to Edge devices must span multiple levels and across all the functions from end to end.

5.11 Points to Ponder

1. How could one improve the Cloud's performance and support for Edge-based devices?
2. Why is Edge computing needed for self-driven cars in the future?
3. Can you think of another example of Edge computing devices on a road?
4. What is the trust and security model for Edge devices?
5. What kinds of attacks are possible using IoT and Edge devices?
6. Can hardware be the sole root of trust?
7. Does ARM's approach using the multistage boot provide a higher level of security than a single stage of attestation in SGX, and why?

5.12 Answers

1. *How could one improve the Cloud's performance and support for Edge-based devices?*

    (a) By having distributed and redundant systems for a failsafe solution.
    (b) Avoid having a single point of failure.
    (c) Backend Cloud services are needed to log data and results for audits and machine learning inferences.
    (d) Sensors can generate enormous data requiring Cloud storage and compute power. However, moving data in and out of Cloud is slow and expensive. So input-output considerations will require local compute and storage power.

2. *Why is Edge computing needed for self-driven cars in the future?*

    (a) Sensors in a moving car can generate enormous data, requiring Cloud storage and compute power. Examples of this are forward-looking and side-view cameras. However, moving data in and out of Cloud is slow and expensive. A car may need to react quickly due to changing road conditions. So input-output considerations for the sensor data will require local compute and storage power. However, any learning and performance data can be reconciled with backend servers during the night or when the car is safely parked.

3. *Can you think of another example of Edge computing devices on a road?*

    (a) A network of traffic lights can communicate and coordinate between them for adjusting to an accident that may be sending scores of cars to other roads. Normally, such a scenario can cause bottlenecks, while other roads may be empty. Using a combination of artificial intelligence and machine learning methods, in the future a network of traffic lights may be able to adapt their sequence of red-yellow-green to ease the wait times on critical junctions.

4. *What is the trust and security model for Edge devices?*

    (a) Edge devices in a Cloud need backend Cloud services that are needed to log data and results for audits and machine learning inferences. However, devices need to trust the Cloud servers, and Cloud needs to trust the incoming device data.

5. *What kinds of attacks are possible using IoT and Edge devices?*

    (a) It has been shown that an army of botnets (a term used for devices on the Internet) can be hijacked by hackers and used for launching distributed denial of service (DDOS) attacks on unsuspecting Cloud servers. An example is of home surveillance cameras that had unsecured IP addresses used for bringing down a security journalist's blog site.

6. *Can hardware be the sole root of trust?*
   (a) As we previously noted, multifactor authentication offers a better defense strategy. Having any single piece of hardware or software as the sole root of trust is risky. One possible solution is mutual attestation by various devices that are not located at the same place or are not under the same control. Thus, an attacker will need to simultaneously compromise multiple devices, which is harder to accomplish than altering any single root of trust.

7. *Does ARM's approach using the multistage boot provide a higher level of security than a single stage of attestation in SGX, and why?*
   (a) Yes, ARM's approach using the multistage boot indeed provides a higher level of security than a single stage of attestation as in SGX. The rationale here is similar to multifactor authentication (MFA), which relies on the fact that it is harder to breach the security of multiple devices or locations. With a multistage boot, a hacker would have to replace boot code at multiple locations. There is no absolute or 100% security, but it can be improved in stages, as ARM has done. They have enhanced the integrity of their boot process.

References

1. https://www.postscapes.com/internet-of-things-history/
2. Ashton, K. (2009). That 'internet of things' thing. *RFID Journal, 22*(7), 97–114.
3. http://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/
4. Christodorescu, M., Sailer, R., Schales, D. L., Sgandurra, D., & Zamboni, D. (2009). Cloud security is not (just) virtualization security: A short chapter. Proceedings of the 2009 ACM workshop on Cloud Computing Security (pp. 97–102), Chicago.
5. Sehgal, N. K., Bhatt Pramod Chandra, P., & Acken, J. M. (2023). *Cloud computing with security and scalability*. Springer.
6. Beckman, P., Sankaran, R., Catlett, C., Ferrier, N., Jacob, R., & Papka, M. (2016). Waggle: An open sensor platform for edge computing. In *2016 IEEE sensors* (pp. 1–3). IEEE. https://doi.org/10.1109/ICSENS.2016.7808975
7. Juels, A., & Kaliski, Jr., B. S. (2007). PORS: Proofs of retrievability for large files. Proceedings of the 14th ACM conference on Computer and Communications Security (pp. 584–597). Alexandria.
8. Cachin, C., Keidar, I., & Shraer, A. (2009). Trusting the cloud. *SIGACT News, 40*, 81–86.
9. https://en.wikipedia.org/wiki/Side-channel_attack
10. https://en.wikipedia.org/wiki/Software_Guard_Extensions
11. Costan, V., & Devadas, S. (n.d.) Intel SGX explained. https://eprint.iacr.org/2016/086.pdf
12. https://azure.microsoft.com/en-us/solutions/confidential-compute/
13. https://www.ibm.com/blogs/bluemix/2018/05/data-use-protection-ibm-Cloud-using-intel-sgx/
14. https://www.arm.com/technologies/trustzone-for-cortex-a
15. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2020). *Cloud computing with security and scalability*. Springer.
16. Enck, W., Butler, K., Richardson, T., McDaniel, P., & Smith, A. (2008). Defending against attacks on main memory persistence. Proceedings of the 2008 annual computer security applications conference (pp. 65–74).

17. http://www.computerworld.com/article/2487452/cybercrime-hacking/target-attack-shows-danger-of-remotely-accessible-hvac-systems.html
18. Al-Rwais, S., & Al-Muhtadi, J. (2010). A context-aware access control model for pervasive environments. *IETE Technical Review, 27*, 371–379.
19. http://www.informationweek.com/Cloud/infrastructure-as-a-service/5-critical-Cloud-security-practices/a/d-id/1318801
20. https://securityintelligence.com/23-best-practices-for-Cloud-security/
21. Anti-Phishing Working Group. (n.d.). Phishing and Fraud solutions. http://www.antiphishing.org/
22. https://www.pandasecurity.com/mediacenter/security/sim-hijacking-explained/
23. Abu-Nimeh, S., Nappa, D., Wang, X., & Nair, S. (2007). A comparison of machine learning techniques for phishing detection. *APWG eCrime Researchers Summit*, Pittsburg.
24. Chellapilla, K., & Simard, P. Y. (2005). Using machine learning to break visual human interaction proofs (HIPs). *Advances in Neural Information Processing Systems, 17*, 265–272.

Chapter 6
Edge Artificial Intelligence

6.1 Introduction

Edge AI refers to the deployment and execution of AI and ML models on Edge devices. These include smartphones, IoT sensors, industrial controllers, and other resource-constrained devices located at the Edge of the network and closer to the data sources. In addition, on-premise servers, including those located in a hybrid Cloud setup, are content delivery networks [1]. This approach contrasts with traditional Cloud-based AI, where data is transmitted to powerful centralized servers for training. Then inference is done on the remote devices. In the expanded realm of Edge AI, remote devices can perform limited training or re-training activities on the remote devices. This includes smart/intelligent Edge data collection, data de-identification/re-identification, and fine tuning of models on Edge.

6.1.1 Key Aspects of Edge AI

Edge AI tasks include decentralized data processing, real-time responsiveness, enhanced privacy and security, reduced bandwidth and cost, resilience and reliability, specialized hardware, federated learning, and diverse applications that are covered in detail below:

- **Decentralized Data Processing:** AI models are deployed on Edge devices, enabling local data processing and decision-making. This reduces the need for constant data transmission to the Cloud, minimizing latency and bandwidth requirements.
- **Real-time Responsiveness**: Edge AI enables real-time processing and decision-making, crucial for applications that require instantaneous responses, such as autonomous vehicles, predictive maintenance, and robotics. By eliminating the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_6

need to send data to the Cloud and wait for a response, Edge AI provides faster reaction times.

- **Enhanced Privacy and Security:** Sensitive data can be processed locally on Edge devices, reducing the risk of data breaches during transmission or storage on centralized servers. Edge AI enables compliance with data privacy regulations and enhances security for sensitive applications.
- **Reduced Bandwidth and Cost:** By processing data locally, Edge AI minimizes the amount of data that needs to be transmitted over the network, reducing bandwidth requirements and associated costs. This is particularly beneficial for applications with limited connectivity or in remote locations.
- **Resilience and Reliability:** Edge AI systems can continue to operate even when network connectivity is disrupted or unavailable, ensuring uninterrupted service delivery. This is crucial for mission-critical applications in industries like manufacturing, healthcare, and transportation.
- **Specialized Hardware:** Edge AI often leverages specialized hardware, such as AI accelerators, field-programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs), optimized for efficient AI processing on resource-constrained devices. These hardware solutions enable the deployment of complex AI models on Edge devices with limited compute resources.
- **Diverse Applications:** Edge AI has potential applications across various industries, including manufacturing (predictive maintenance, quality control), healthcare (remote patient monitoring, medical imaging), smart cities (traffic management, environmental monitoring), and consumer electronics (voice assistants, augmented reality).

Next, we will examine the traditional AI approach, where learning mainly happens in a centralized environment and data travels from remote locations to the central data centers. This will be followed by a brief review of decentralized learning, also known as federated learning. After that, we will examine the emerging area of generative AI using vector databases and transformer architecture. We will conclude this chapter by looking at how load balancing can be applied to Edge AI and describing the need for embedded AI in the Edge devices.

6.2 Centralized Learning

Centralized learning in the context of AI refers to the traditional approach where data is collected from various sources and transmitted to a central server or data center for processing, training, and model development. This method contrasts with decentralized learning, where data remains distributed across multiple devices or locations, and only model updates are shared. Decentralized learning, also known as federated learning [2], will be covered in detail later in this chapter. Centralized learning has been the dominant paradigm in AI due to its simplicity and the availability of powerful centralized computing resources. Below, we explore the key aspects, benefits, and challenges of centralized learning in AI.

6.2.1 Key Aspects of Centralized Learning in AI

These consist of data collection, model training, model deployment, and monitoring and management that are covered in detail below.

- **Centralized Data Collection:** Data from various sources is aggregated and stored in a central repository, such as a data warehouse or Cloud storage. This centralization facilitates comprehensive data analysis and model training.
- **Centralized Model Training:** AI models are trained on the aggregated data using powerful centralized computing resources, such as graphics processing units (GPUs) [3] and tensor processing units (TPUs) [4]. This approach leverages the full computational power of data centers to train complex models efficiently.
- **Centralized Model Deployment:** Once trained, AI models are deployed from the central server to various applications and devices. Updates and improvements to the models are managed centrally and distributed as needed.
- **Centralized Monitoring and Management:** The performance of AI models is monitored centrally, allowing for consistent oversight and management. Centralized systems can quickly identify and address issues, ensuring optimal model performance.

6.2.2 Benefits of Centralized Learning in AI

The benefits consist of access to large datasets, efficient use of computational resources, consistency and standardization, and simplified management and maintenance.

- **Access to Large Datasets:** Centralized learning allows for the aggregation of large and diverse datasets, which can improve the accuracy and robustness of AI models. This is particularly beneficial for training deep learning models that require vast amounts of data.
- **Efficient Use of Computational Resources:** Centralized data centers can leverage high-performance computing resources, such as clusters of GPUs and TPUs, to train models more efficiently. This reduces the time required for model training and enables the development of more complex models.
- **Consistency and Standardization:** Centralized learning ensures that all data is processed and analyzed using the same standards and methodologies. This consistency is crucial for maintaining the quality and reliability of AI models.

- **Simplified Management and Maintenance:** Centralized systems simplify the management and maintenance of AI models, as updates and improvements can be implemented in a single location. This reduces the complexity of deploying and maintaining AI models across multiple devices or locations.

6.2.3 Challenges of Centralized Learning

These consist of data privacy and security, latency and bandwidth constraints, scalability issues, single points of failure, etc.

- **Data Privacy and Security:** Centralized learning requires the transfer of data to a central server, which can raise concerns about data privacy and security. Sensitive data may be at risk during transmission or while stored in centralized repositories. These concerns can be addressed by solutions such as data encryption/decryption or data anonymization [5] and data de-identification/re-identification [6].
- **Latency and Bandwidth Constraints:** Transmitting large volumes of data to a central server can result in high latency and bandwidth consumption. This can be problematic for applications that require real-time processing and decision-making. One possible solution is to pick a centralized data center close to the data generation and consumption locations [1].
- **Scalability Issues:** As the volume of data and the number of devices increase, centralized systems may struggle to scale efficiently. This can lead to bottlenecks and reduced performance. The methods to scale using network, compute and data are well known [2] but come at a higher cost.
- **Single Point of Failure:** Centralized systems can be vulnerable to failures or attacks, as a single point of failure can disrupt the entire AI infrastructure. Ensuring the reliability and resilience of centralized systems is a significant challenge. This can be done with regular backups of critical data and having hot standby servers in a data center.

6.3 Federated Learning

In many instances, machine learning data is owned by different entities that do not trust each other or do not wish to share their data even for a common purpose of training a model. Such is the case with medical data spread between different hospitals. They are also bound by laws (such as HIPAA in the USA) to protect their patient identities. Hence, a new technology has emerged to train ML models at scale across multiple medical institutions without moving the data between them. It is called federated learning (FL) [7], an instance of which is depicted in Fig. 6.1.

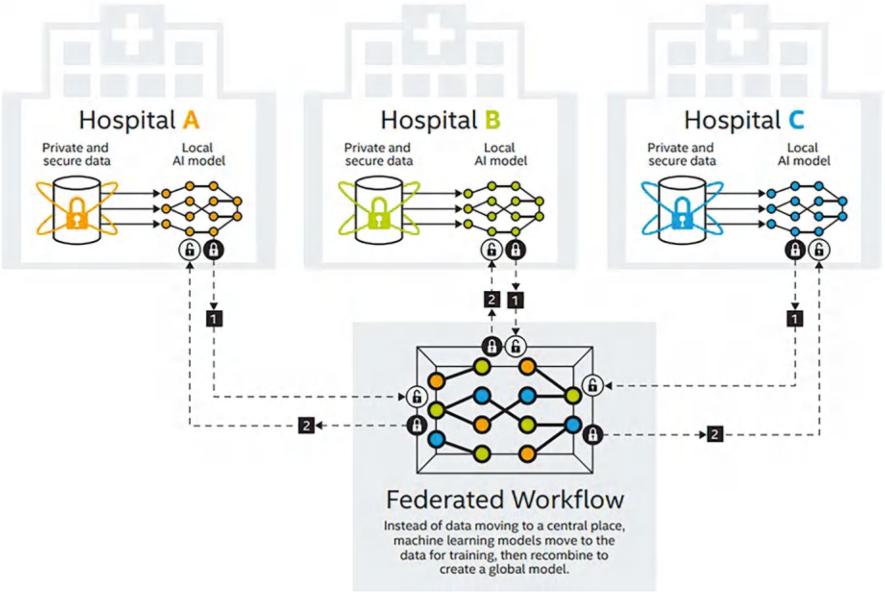


Fig. 6.1 Federated learning architecture. 1 is local model sharing. 2 is global model sharing updates. Image used courtesy of Intel [7]

FL enables data to stay local and algorithms to travel across the participating institutions for training a deep learning algorithm while preserving privacy and security of the patients' data. However, FL has a performance penalty that we shall discuss in the next chapter.

6.3.1 Balancing Centralized and Decentralized Learning

To address the limitations of centralized learning, many organizations are exploring hybrid approaches that combine the strengths of both centralized and decentralized learning. This hybrid model allows for various aspects outlined below.

- **Centralized Training for Core Models:** Core AI models are trained centrally using aggregated data and powerful computing resources to ensure high accuracy and robustness.
- **Decentralized Learning for Local Adaptation:** Localized models are trained on Edge devices or local servers using decentralized learning techniques, allowing for real-time processing and enhanced privacy. Only model updates, rather than raw data, are shared with the central server, reducing bandwidth consumption and enhancing data privacy.

- **Enhanced Collaboration and Flexibility:** Central and local AI teams collaborate to share insights, best practices, and model updates, fostering a more integrated and adaptive AI ecosystem.

Centralized learning in AI offers significant advantages in terms of data aggregation, computational efficiency, and consistency. However, it also presents challenges related to data privacy, latency, scalability, and resilience. By adopting a balanced approach that incorporates elements of both centralized and decentralized learning, organizations can create a more effective and adaptable AI strategy that meets the diverse needs of their applications and users.

6.4 Generative AI

Generative AI (GenAI) [8] represents a revolutionary advancement in AI focusing on creating new content rather than simply analyzing existing data. This rapidly evolving technology has the potential to positively transform numerous industries and aspects of our daily lives. It can lead to increased productivity and reduced costs.

6.4.1 Introduction to Generative AI

At its core, GenAI relies on complex neural networks and machine learning algorithms to process and learn from vast amounts of training data. These systems employ techniques such as deep learning, natural language processing, and reinforcement learning to understand patterns and relationships within the data.

Interacting with GenAI typically begins with a prompt, which can be in the form of text, an image, or other input that the AI system can process. The AI then uses its trained algorithms to generate new content in response to this prompt. This content can range from text and images to audio and even synthetic data that closely resembles human-created content.

6.4.2 Key Components of GenAI

- **Large Language Models (LLMs):** These are deep neural networks trained on massive amounts of text data, enabling them to understand and generate human-like text.
- **Multimodal LLMs:** These are models designed to process and generate data across multiple types of modalities, such as text, images, audio, and video. While traditional LLMs are primarily focused on text, multimodal models combine various types of input and output, enabling richer, more versatile capabilities.

- **Neural Networks:** These structures, loosely modeled after the human brain, consist of interconnected nodes that work together to process information.
- **Tokenization:** This process breaks down input text into smaller units called tokens, helping the AI understand the structure and meaning of the input.

6.4.3 Vector Database

A vector database [9] is a specialized storage system designed to efficiently manage and query high-dimensional vector data. This is crucial for various AI and ML applications. Here are the key aspects of vector databases:

6.4.3.1 Definition and Functionality

Vector databases store and index vector mappings, i.e., numerical representations of data objects, thus allowing for fast retrieval and similarity searches. Unlike traditional databases that rely on exact matches, vector databases enable searches based on the semantic similarity of the data, making them particularly useful in AI contexts where understanding relationships and patterns is essential. For example, words “lamb” and “sheep” are closer to each other vs. words “lamb” and “apple” that are distant to each other. In the same way, words “apple” and “orange” are closer to each other vs. words “apple” and “lamb” that are distant to each other.

6.4.3.2 Key Features

- **High-Dimensional Data Handling:** Vector databases are optimized to manage vectors that can have hundreds to thousands of dimensions, where each dimension corresponds to a specific feature or attribute of the data being represented.
- **Approximate Nearest Neighbor (ANN) Search:** Vector databases implement algorithms that facilitate efficient ANN searches, allowing users to find vectors that are closest to a given query vector. This capability is vital for applications like recommendation systems and semantic search.
- **Scalability and Flexibility:** Vector databases are designed to handle large volumes of vectored data, providing horizontal scalability and dynamic data management capabilities. This is essential for modern AI applications.
- **Integration with AI Models:** These databases often work in conjunction with large language models (LLMs) and other AI systems, enabling tasks such as retrieval-augmented generation (RAG), where relevant information is retrieved to enhance model responses [10].
- **Applications:** Vector databases are utilized in various domains, including recommendation systems, natural language processing (NLP), image recognition and retrieval and anomaly detection.

6.4.3.3 Differences from Traditional Databases

Unlike traditional databases that store data in tabular formats and require exact matches for queries, vector databases focus on storing data as vectors and allow for similarity-based searches. This shift enables a broader scope of results and the ability to handle complex data relationships, making vector databases particularly suited for AI and ML applications.

Vector databases represent a significant advancement in data management, particularly in the context of AI and ML. Their ability to store, retrieve, and query high-dimensional data efficiently opens up new possibilities for applications that require an understanding of complex relationships and patterns within data. For example, these are very useful in NLP activities such as sentiment analysis, document similarity, and semantic searches. As AI technologies continue to evolve, the role of vector databases will become increasingly critical in enabling sophisticated data-driven solutions.

6.4.4 Transformer Architecture

The Transformer Architecture [11] has revolutionized the field of AI since its introduction. Originally designed for NLP, transformers have become the backbone of numerous AI applications, extending their influence beyond text to areas such as computer vision, robotics, and more. This section provides a comprehensive overview of the transformer architecture, its components, functionality, applications, and implications for the future of AI.

6.4.4.1 Overview of Transformer Architecture

Transformers are a type of neural network architecture that utilizes self-attention mechanisms to process sequential data. Unlike traditional recurrent neural networks (RNNs), which process input data sequentially, transformers can evaluate all elements of the input simultaneously. This parallel processing capability allows for more efficient training and improved performance on tasks involving long-range dependencies.

The architecture is typically divided into two main components:

- **Encoder**: The encoder processes the input sequence and generates contextual embeddings for each token. It captures the relationships between tokens through self-attention mechanisms.
- **Decoder**: The decoder takes the encoder's output and generates the final output sequence. It uses both the encoder's context and its own previous outputs to produce the next token in the sequence.

6.4.4.2 Core Components of Transformers

- **Embedding Layer**: The input tokens are first converted into numerical representations (embeddings) that capture their semantic meanings. This layer also includes positional encodings to retain information about the order of tokens in the sequence.
- **Self-Attention Mechanism**: This mechanism allows the model to weigh the importance of different tokens in the sequence relative to each other. Each token attends to all other tokens, enabling the model to capture contextual relationships effectively.
- **Multi-Head Attention**: Instead of a single attention mechanism, transformers utilize multiple attention heads that operate in parallel. Each head learns different aspects of the relationships between tokens, enhancing the model's ability to capture diverse patterns in the data.
- **Feed-Forward Neural Network**: After the attention mechanism, the output is passed through a feed-forward neural network, which applies nonlinear transformations to the data. This layer is applied independently to each position in the sequence.
- **Layer Normalization and Residual Connections**: Each sub-layer (attention and feed-forward) is followed by layer normalization and residual connections, which helps stabilize training and improve convergence.
- **Output Layer**: The final output from the decoder is transformed into probabilities for each token in the vocabulary, allowing the model to generate the next token in the sequence.

6.4.4.3 Advantages of Transformer Architecture

- **Parallelization**: The transformer architecture allows for parallel processing of input data, significantly speeding up training times compared to RNNs, which require sequential processing.
- **Handling Long-Range Dependencies**: Transformers effectively capture relationships between distant tokens in a sequence, overcoming the limitations of RNNs in managing long-term dependencies.
- **Flexibility**: The architecture can be adapted for various tasks, including text generation, translation, summarization, and even image processing through variants like vision transformers (ViTs).

The transformer architecture has fundamentally changed the landscape of artificial intelligence, enabling significant advancements in natural language processing and beyond. Its ability to process data efficiently and capture complex relationships has made it a cornerstone of modern AI. As research continues to evolve, the transformer architecture will likely remain a key player in shaping the future of AI technologies, driving innovations across diverse fields and applications.

6.4.5 GenAI Applications and Impact

GenAI has a wide range of applications across various industries:

- **Art and Design:** It can create unique artworks, assist in product design, and generate architectural layouts.
- **Writing and Content Creation:** It can produce articles, essays, and even code.
- **Music:** It can compose original melodies and complete unfinished musical pieces.
- **Healthcare:** It can aid in drug discovery and design new molecules.
- **Business:** It can help redesign business processes, transform supply chains, and generate marketing content.

The impact of GenAI extends beyond these specific applications. It has the potential to enhance creativity and productivity, streamline workflows, and solve complex problems in ways previously unimaginable. For instance, in scientific research, GenAI can assist in generating hypotheses, analyzing data, and even writing parts of research papers.

6.4.6 Challenges and Ethical Considerations

While GenAI offers immense potential, it also presents several challenges:

- **Accuracy and Bias:** Early implementations have shown issues with accuracy and can perpetuate biases present in training data [12].
- **Hallucinations:** AI systems can sometimes produce fabricated or inaccurate information with high confidence [13].
- **Copyright and Authorship:** The use of AI-generated content raises questions about intellectual property rights and authorship [14]. This includes getting permission from the content owner before the content is used for training.
- **Job Displacement:** There are concerns about AI potentially replacing human workers in certain creative fields [15].

Addressing these challenges requires on-going research, ethical guidelines, and potentially new legal frameworks. It's crucial to develop GenAI systems that are transparent, accountable, and aligned with human values.

6.4.7 The Future of Generative AI

The field of GenAI is rapidly evolving, with on-going research aimed at improving accuracy, reducing biases, and expanding capabilities. Future developments may include:

- More sophisticated models are capable of generating increasingly complex and nuanced content.
- Better integration with other AI technologies for more comprehensive solutions.
- Improved user interfaces make GenAI more accessible to nontechnical users.
- Advancements in ethical AI to address concerns about bias and misinformation.
- Emotion analysis is natural language processing that can provide information on customer sentiment with rules-based or keyword-based approaches.

As GenAI continues to advance, it has the potential to revolutionize various aspects of our lives, from how we create art and solve problems to how businesses operate and innovate. However, it is crucial to approach this technology with careful consideration of its ethical implications and potential societal impacts. GenAI represents a significant leap forward in artificial intelligence, offering unprecedented capabilities in content creation and problem-solving. As we continue to explore and develop this technology, it is essential to balance its immense potential with responsible implementation and ethical considerations. The future of GenAI is bright, and its impact on society is likely to be profound and far-reaching.

6.5 Load Balancing in Edge AI

Load balancing is important because it helps distribute traffic across the network Edge, which can improve performance and reliability. When combined with Edge computing, load balancing can create a highly responsive system that can handle large amounts of data and traffic.

In Edge AI, load balancing is a critical consideration to ensure efficient use of resources, maintain performance, and provide reliability across distributed systems. As Edge AI applications continue to grow, managing workloads effectively becomes paramount to maximize the benefits of low latency, bandwidth optimization, and local processing. Here is a detailed exploration of the load balancing in Edge AI:

6.5.1 Key Considerations for Load Balancing in Edge AI

The key considerations are distributed architecture, dynamic workload distribution, resource constraints, data localization and privacy, network connectivity, scalability, flexibility, latency sensitivity, reliability, and redundancy.

6.5.1.1 Distributed Architecture

As shown in Fig. 6.2, we have several end devices. Traffic from these devices gets redirected by an intelligent Edge gateway and subsequently loads balanced on the server farm with servers marked in green.

- **Edge Nodes**: Unlike centralized Cloud computing, Edge AI involves multiple Edge nodes located near data sources [16]. Load balancing must account for the distribution of these nodes to different servers for optimized processing. Each Edge node should have multiple application instances running, and traffic to them can be managed via an Edge intelligent gateway that ensures that the load is spread out and that servers do not become overloaded [17].
- **Hierarchical Structure**: Often, Edge AI systems use a hierarchical structure where local Edge devices communicate with regional Edge servers before reaching the Cloud. Load balancing needs to address each level of this hierarchy.

- The load balancers in the Edge nodes can also use global server load balancing (GSLB) [18] to work with peers in other Edge data centers and in the Cloud to manage and spread the load over multiple sites in a region as required to maintain the best response times.

6.5.1.2 Dynamic Workload Distribution

- **Real-time Processing**: Edge AI applications often require real-time processing, which demands dynamic workload distribution to avoid delays. This means load balancers must quickly adjust to changing conditions.
- **Variable Demand**: Different Edge nodes may experience varying demand based on user activity or sensor inputs, necessitating adaptive load balancing strategies.

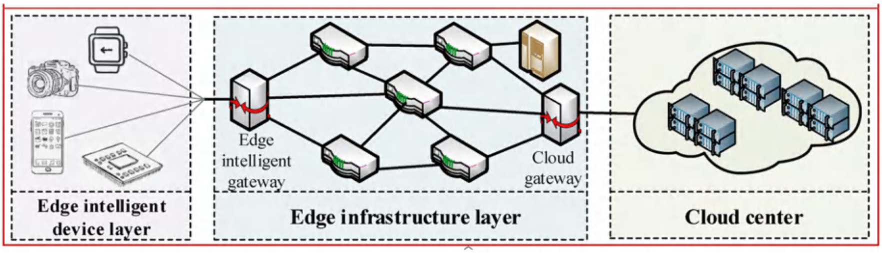


Fig. 6.2 Edge computing load balancing architecture [16]

6.5.1.3 Resource Constraints

- **Limited Capacity:** Edge devices often have limited computing power, memory, and energy resources. Load balancing strategies need to optimize resource use without overwhelming individual devices.
- **Energy Efficiency:** Balancing workloads to minimize energy consumption is crucial for battery-powered devices and contributes to sustainability efforts.

6.5.1.4 Data Localization and Privacy

- **Local Data Processing**: Load balancing must consider data locality to ensure that sensitive data is processed locally, enhancing privacy and compliance with data protection regulations.
- **Regional Constraints**: Certain data might be restricted to specific regions due to privacy laws, requiring careful management of where processing occurs.

6.5.1.5 Network Connectivity

- **Intermittent Connections**: Edge devices may experience intermittent network connectivity, requiring load balancing solutions that can adapt to connectivity changes.
- **Bandwidth Management**: Efficiently distributing workloads to manage bandwidth usage is essential, particularly in environments with limited connectivity.

6.5.1.6 Scalability and Flexibility

- **Scaling Workloads**: As the number of Edge devices increases, load balancing mechanisms must scale to accommodate additional devices and workloads.
- **Flexible Architectures**: Load balancers should support flexible architectures that can adapt to new devices, applications, and changing demands.

6.5.1.7 Latency Sensitivity

- **Proximity Considerations**: Load balancing must minimize latency by considering the physical proximity of Edge devices to the data source and the processing node.
- **Time-critical Applications**: Applications with stringent latency requirements, such as autonomous vehicles or industrial automation, need specialized load balancing strategies to ensure timely processing.

6.5.1.8 Reliability and Redundancy

- **Fault Tolerance**: Load balancing should provide redundancy to ensure reliability in the face of hardware failures or network disruptions.
- **Failover Mechanisms**: Implementing failover mechanisms that reroute traffic to available nodes is crucial to maintaining service continuity.

6.5.2 Load Balancing Techniques in Edge AI

- **Round Robin Simple Allocation:** Distributes tasks evenly across Edge nodes in a cyclic manner. It's simple to implement but doesn't consider the current load or capacity of nodes.
- **Least Connections Dynamic Load Adjustment:** Assigns new tasks to the node with the fewest active connections, balancing the load based on current activity levels.
- **Weighted Distribution Resource-Based Allocation:** Assigns weights to nodes based on their capacity, directing more traffic to more powerful devices. This considers the heterogeneity of Edge devices.
- **Geo-Location-Based Proximity Optimization:** Directs tasks to the closest Edge node geographically, reducing latency and improving performance for location-sensitive applications.
- **AI-Driven Load Balancing**
  - **Predictive Analysis**: Uses AI algorithms to predict workloads and adjust distribution proactively. Machine learning models can analyze historical data to forecast demand and optimize resource allocation.
  - **Adaptive Algorithms**: Continuously learns and adapts to changes in demand patterns, optimizing load balancing decisions in real time.
- **Fog Computing Intermediate Layer**: Utilizes fog nodes as an intermediate layer between Edge devices and the Cloud, distributing tasks based on current load and proximity to end users.
- **Multi-access Edge Computing (MEC) Network Edge Integration**: Integrates computing capabilities at the network Edge, enabling more efficient load balancing by closely coupling processing with network operations.

6.5.3 Load Balancing Implementation Challenges

- **Complexity of Distributed Systems**: Managing a large number of Edge devices with varying capabilities adds complexity to load balancing strategies.

- **Real-time Decision-Making**: The need for immediate processing and decision-making requires load balancers to operate with minimal delay.
- **Security Concerns**: Ensuring secure communication and data integrity during load balancing operations is vital to preventing unauthorized access or data breaches.
- **Heterogeneity of Devices**: Edge AI systems often involve devices with different hardware and software configurations, complicating uniform load balancing strategies.
- **Cost Management**: Balancing the cost of computation, data transmission, and energy consumption requires careful consideration to maximize efficiency while minimizing expenses.

6.5.4 Load Balancing Future Trends

- **Edge-to-Cloud Continuum Seamless Integration**: Future load balancing solutions will seamlessly integrate Edge and Cloud resources, optimizing the flow of data and computation across the entire continuum.
- **AI-Enhanced Load Balancing Intelligent Automation**: Increasing use of AI to automate and enhance load balancing processes, leveraging machine learning for predictive and adaptive distribution.
- **5G and Beyond Enhanced Connectivity**: The proliferation of 5G networks will improve connectivity and enable more sophisticated load balancing strategies with higher data throughput and lower latency [19].
- **Collaborative Edge Environments Resource Sharing**: Edge nodes may collaborate to share resources and balance workloads across different locations, enhancing overall system performance and resilience. This is covered in detail in Chap. 8 with regard to provisioning considerations of mission-critical vs. non-mission-critical apps.
- **Advanced Optimization Algorithms Innovation**: Development of advanced algorithms to tackle the complexity and dynamism of Edge environments, improving efficiency and adaptability [20].

Load balancing in Edge AI is essential for maximizing the potential of distributed computing environments. By considering factors such as resource constraints, latency, scalability, and network connectivity, effective load balancing strategies can enhance performance, reliability, and efficiency in Edge AI applications. As technology evolves, innovative solutions and advancements will continue to shape the landscape of load balancing in Edge AI, enabling new possibilities and driving further growth in this field.

6.6 Embedded AI at the Edge

Embedded AI at the Edge [21] refers to the integration of AI capabilities directly into Edge devices. This enables these devices to process data and make decisions locally without the need for constant communication with centralized Cloud infrastructure. This approach leverages Edge computing, where data is processed near its source, delivering numerous advantages across various applications. Here is a broad look into embedded AI at the Edge.

6.6.1 Key Components of Embedded AI at the Edge

- **Edge Devices**: These are hardware components equipped with computational resources to perform AI tasks. Edge devices include sensors, cameras, microcontrollers, and more sophisticated devices such as smartphones, industrial machinery, and autonomous vehicles. They are designed to handle specific AI tasks locally, such as image recognition, anomaly detection, and speech processing.
- **AI Models:** AI models used at the Edge are typically lightweight and optimized for efficiency, enabling them to run on devices with limited computational power. Techniques such as model compression, pruning, and quantization are used to reduce the size and complexity of AI models while maintaining their accuracy and performance.
- **Edge Computing Infrastructure**: This refers to the distributed computing framework that allows data processing to occur on the Edge devices or nearby servers. Please refer to Chap. 3 of this book for detailed treatment on this. Edge infrastructure often reduces the need to send data back to the Cloud, thus minimizing latency and bandwidth usage while enabling real-time decision-making.

6.6.2 Advantages of Embedded AI at the Edge

- **Low Latency**: By processing data locally, embedded AI at the Edge significantly reduces the time required to analyze and respond to data inputs. This is critical for real-time applications such as autonomous vehicles, industrial automation, and augmented reality, where delays can lead to inefficiencies or safety hazards.
- **Improved Privacy and Security**: Data processed on Edge devices can remain local, reducing the risk of exposure during transmission to centralized servers. This enhances privacy and security, particularly for sensitive data in industries like healthcare, finance, and retail.
- **Reduced Bandwidth Costs**: With data processing occurring locally, there is less need to transmit large volumes of data over networks. This can significantly reduce bandwidth costs and alleviate network congestion, especially in environ-

ments with numerous connected devices, such as smart cities and industrial settings.

- **Reliability and Availability**: Edge devices can continue to operate and make decisions even when Cloud connectivity is intermittent or unavailable. This ensures continuous service delivery in critical applications and enhances system resilience against network failures.
- **Energy Efficiency**: Local data processing can be more energy-efficient than constantly sending data to and from the Cloud, especially for battery-powered devices. This efficiency is crucial for extending the operational lifespan of IoT devices and supporting sustainable development goals.

6.6.3 Applications of Embedded AI at the Edge

- **Smart Homes**: In smart home environments, devices such as smart speakers, thermostats, and security cameras use embedded AI to personalize user experiences, automate home management tasks, and enhance security without relying heavily on Cloud services. For instance, AI-enabled cameras can detect and respond to unusual activities in real time.
- **Industrial IoT**: Embedded AI at the Edge enables predictive maintenance, real-time monitoring, and quality control in manufacturing and industrial settings. By analyzing sensor data locally, Edge devices can quickly detect anomalies and optimize operations, reducing downtime and improving productivity.
- **Healthcare**: Wearable devices and medical sensors equipped with embedded AI can monitor patient vitals, detect anomalies, and provide immediate feedback. This improves patient care, reduces the burden on healthcare systems, and enables remote monitoring and telemedicine applications.
- **Autonomous Vehicles**: Self-driving cars and drones rely on embedded AI at the Edge to process data from sensors and cameras in real time. This enables them to navigate, detect obstacles, and make decisions without relying on Cloud-based systems, which is essential for safety and efficiency.
- **Retail**: In retail environments, Edge devices use AI to analyze customer behavior, manage inventory, and optimize operations. This enhances customer experiences by providing personalized recommendations and streamlining checkout processes while improving operational efficiency.

6.6.4 Challenges and Considerations of AI at the Edge

- **Resource Constraints**: Edge devices often have limited processing power, memory, and storage, requiring AI models to be highly optimized for efficient performance. This necessitates the careful selection and adaptation of AI algorithms to fit within these constraints.

- **Model Deployment and Management**: Deploying and updating AI models on numerous Edge devices can be complex, requiring robust management and orchestration systems. Ensuring consistency and reliability across a distributed network of devices is crucial.
- **Interoperability**: Ensuring compatibility and seamless communication between different Edge devices and platforms is essential for effective implementation. This requires standardized protocols and interfaces to facilitate integration and collaboration.
- **Security**: Protecting Edge devices from cyber threats is critical, as they can be vulnerable entry points into broader networks. Implementing strong security measures, such as encryption and authentication, is essential to safeguarding data and systems.
- **Data Quality**: Ensuring the accuracy and relevance of data collected and processed by Edge devices is crucial for reliable AI performance. High-quality data is essential for training accurate models and making informed decisions.

6.6.5 Future Trends

- **Advancements in Hardware**: The development of specialized AI chips and processors, such as neural processing units (NPUs) and tensor processing units (TPUs), will enhance the capabilities of Edge devices. These advancements will enable more complex AI models to run efficiently at the Edge.
- **Federated Learning**: This technique allows Edge devices to collaboratively train AI models using local data while preserving privacy. By aggregating model updates rather than raw data, federated learning reduces the need for centralized data storage and enhances privacy protection.
- **5G and Beyond**: The rollout of 5G networks will enable faster and more reliable connectivity, supporting more advanced Edge AI applications by facilitating seamless communication between devices. This will enable new use cases, such as augmented reality and remote-controlled robotics.
- **Edge AI Platforms**: The growth of platforms and frameworks designed to simplify the deployment and management of AI on Edge devices will accelerate adoption across industries. These platforms will provide tools and services for model deployment, monitoring, and optimization.

Embedded AI at the Edge is poised to revolutionize numerous sectors by bringing intelligence closer to the source of data generation. As technology continues to advance, the potential for innovative applications and solutions will expand, driving further growth and transformation in this space.

6.7 Summary

Edge AI is an emerging area that is a crucial part of the overall AI rollout. We have covered an introduction to centralized AI learning vs. federated learning. Together with that, the emerging areas of generative AI, load balancing and embedded Edge AI have been described in detail. All these have given birth to many new exciting Edge AI applications with more diverse expansion in the future.

6.8 Points to Ponder

1. What role does load balancing play in achieving scalability?
2. Does Edge computing help in scaling for IoT systems?
3. What are the characteristics of systems where centralized learning would be appropriate and where federated learning would be appropriate?
4. How can a mixed-mode architecture overcome the limitations of federated learning?
5. How does Edge computing affect the overall performance of an AI system?

6.9 Answers

1. What role does load balancing play in achieving scalability?

   Note that scalability is not linear, as bottlenecks inevitably arise in a network or some other I/O (input-output) path within a computer. Then, parallel computers or servers are deployed in a data center to handle larger workloads. However, load balancing is needed to ensure that all the servers are evenly loaded. This is akin to checkout lines at a grocery store. Having more cash registers and clerks will help only if customers are evenly distributed in different lines. This reduces overall system-level latencies.
2. Does Edge computing help in scaling for IoT systems?

   Yes, because scaling in the context of IoT systems means ability to handle a larger number of sensors and end points. However, if all of them need to send their data to a centralized server, it will inevitably result in delays. By introducing additional processing and storage at local sites closer to the end points, we can minimize the dependence on a centralized server. An Edge server may serve as a data aggregation device and can make localized decisions for IoT without going back to the central server. The central database can be updated in a slack operational mode or with periodical backups when the workload is known to be less.
3. What are the characteristics of systems where centralized learning (CL) would be appropriate and where federated learning would be appropriate?

CL is appropriate when all the parties that are contributing data in a centralized data center trust each other or data can be appropriately de-identified. For example, from many different Tesla vehicles, car data is de-identified and rolled up in the data center for training. However, in an environment where parties do not trust each other or laws require separation, such as healthcare data of different patients in different hospitals needs to be protected under Health Insurance Portability and Accountability Act (HIPAA), then FL is appropriate.

4. How can a mixed-mode architecture overcome the limitations of federated learning?

   Federated learning architecture is used when participants in a machine learning solution do not trust each other and want to keep their datasets locals. This helps in improving security. However, it results in additional delays as machine learning code needs to be copied from one customer site to another. A compromise is mixed-mode architecture, where only most sensitive or private parts of a dataset are kept local, while considerable parts of de-identified data can be shared in a centralized location. This improves the run time for machine learning algorithms.

5. How does Edge computing affect the overall performance of an AI system?

   Edge computing tends to improve the overall AI performance in a manner similar to a local cache memory on a server. Datasets are stored locally, closer to where they are generated and processed, without transferring back and forth between the Edge of a network and a central server. This definitely helps in reducing latency.

References

1. https://www.akamai.com/glossary/what-is-a-cdn
2. Sehgal, N. K., Bhatt Pramod Chandra, P., & Acken John, M. (2023). *Cloud computing with security and scalability: Concepts and practices* (3rd ed.). Springer.
3. https://www.gigabyte.com/Glossary/gpu
4. https://en.wikipedia.org/wiki/Tensor_Processing_Unit
5. https://en.wikipedia.org/wiki/Data_anonymization
6. Kumar, S. N., & Bhatt Pramod Chandra, P. (2024). *Project management in cloud applications*. Springer. https://link.springer.com/book/10.1007/978-3-031-53890-2
7. https://www.allaboutcircuits.com/news/can-machine-learning-keep-patient-privacy-for-tumor-research-intel-says-yes-with-federated-learning/
8. https://www.techtarget.com/searchenterpriseai/definition/generative-AI
9. https://www.datastax.com/guides/what-is-a-vector-database
10. https://en.wikipedia.org/wiki/Retrieval-augmented_generation
11. https://en.wikipedia.org/wiki/Attention_Is_All_You_Need
12. https://www.ibm.com/topics/ai-bias
13. https://en.wikipedia.org/wiki/Hallucination_(artificial_intelligence)
14. https://hbr.org/2023/04/generative-ai-has-an-intellectual-property-problem
15. https://builtin.com/artificial-intelligence/ai-replacing-jobs-creating-jobs
16. https://www.mdpi.com/2071-1050/14/15/9602
17. https://ieeexplore.ieee.org/document/9448021

18. https://kemptechnologies.com/global-server-load-balancing-gslb
19. https://www.matec-conferences.org/articles/matecconf/pdf/2017/39/matecconf_cscc2017_03010.pdf
20. https://www.edge-ai-vision.com/2024/02/optimizing-generative-ai-for-edge-devices/
21. https://www.computer.org/csdl/magazine/co/2023/09/10224582/1PI5P76M3jq

Part II
Practices

Chapter 7
Security and Performance at the Edge

7.1 Background

Federated learning (FL) is a machine learning solution architecture where many clients within an organization, or across multiple organizations, collaboratively train a model. This is done under the orchestration of a central server (e.g., a service provider), while keeping the training data decentralized. FL can mitigate many of the systemic privacy risks and costs resulting from traditional, centralized machine learning and data science approaches. The central server receives the dataset contributions from all clients. FL is typically used when one needs to train models on a larger dataset than any one single entity owns and is not willing to share its data with others (e.g., for legal, strategic, or economic reasons). FL trains an algorithm keeping the training data locally on users' decentralized systems rather than contributing it to a single data center for training. The distributed locations are used as nodes performing computation on their local datasets to update a global model [1]. This is in contrast to traditional centralized machine learning techniques, where all the local datasets are uploaded to a shared server location. If data types at different locations are different, then FL enables multiple participants to build a common, robust machine learning model without sharing their data, thus addressing critical issues such as data privacy, security, and access rights. FL solutions for Clinical and Biomedical research are already exploring cross-device FL solutions [2].

7.2 Security Concerns with Centralized Learning

With many IoT devices and use cases, it is imperative to have localized compute power and data storage. An example is a car [3], as shown in Fig. 7.1, which can generate up to 5 TB of data/day. This comes from onboard cameras, Infrared (IR)

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_7

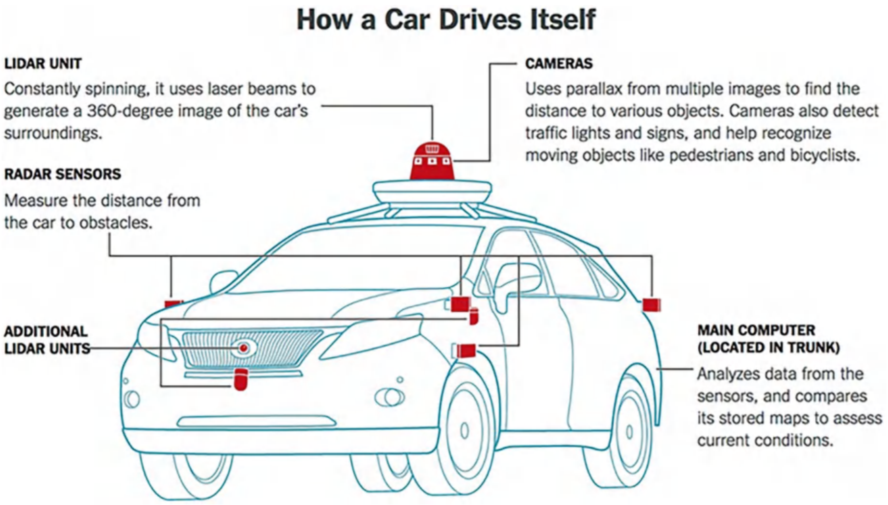

How a Car Drives Itself
LIDAR UNIT
Constantly spinning, it uses laser beams to generate a 360-degree image of the car's surroundings.
CAMERAS
Uses parallax from multiple images to find the distance to various objects. Cameras also detect traffic lights and signs, and help recognize moving objects like pedestrians and bicyclists.
RADAR SENSORS
Measure the distance from the car to obstacles.
ADDITIONAL LIDAR UNITS
MAIN COMPUTER (LOCATED IN TRUNK)
Analyzes data from the sensors, and compares its stored maps to assess current conditions.

Fig. 7.1 A car's self-driving system with multiple sensors [3]

sensors, and data collected from the engine, brakes, etc. However, an autonomous car cannot pause for a server in the Cloud to make a decision to accelerate or brake. Hence, it needs sufficient compute power in the car to drive safely. This capability has been dubbed a "data center on the wheels." It can synch up with a remote data center in the Cloud overnight while parked, but on the road, it must focus on safe driving with real-time decision-making. Hence, a part of the Cloud is migrated from remote data center to the field, termed Edge computing.

However, some functions are still best guided by a central server, such as navigational decisions for routing to a destination. The server can guide the car on which exit to take, but the onboard computer on a self-driven car must decide when to turn the wheels to take that exit.

Security concerns abound with the emergence of Edge computing. In the car example, its computers are not behind a firewall but physically accessible to many people besides the owner. When a car is taken to a mechanic for an oil change or another repair, there is a risk of someone tampering with the hardware or software components, setting up a future failure of the self-driven car. It is also possible for someone to access private data stored in the car, e.g., its travel points. Vulnerabilities in other unprotected devices such as home appliances (TV, fridge) on a network can be used to launch a cyberattack. A recent DDOS (distributed denial-of-service) attack was launched using hijacked home security cameras, while in another instance private video clips were stolen and posted on the Internet [4].

Even a simple home automation system, such as an intelligent door lock, needs the following security features for safety:

1. A firewall to dissuade remote hackers with login authentication.

2. Authentication requires identification of phone numbers, passwords, or biometrics such as face recognition, thumbprint, and retina scan.

Note that any single biometric can be easily defeated, e.g., a pictured mask to fool a face recognition or copy of a thumbprint image, presented to the door camera. It is desirable to have a multifactor authentication (MFA) system. Furthermore, a data-logging system is needed to record who opened or locked the door and when. This data is immediately backed to a remote Cloud to avoid local tampering. Machine intelligence can be used to create a regular usage pattern and flag anomalies, e.g., when a door is opened at unexpected hours or with unusual frequency.

7.3 Performance Issues with Federated Learning

Imagine that multiple parties need to collaborate for a common purpose but do not trust each other with their data sharing. An example is the research for a new drug that needs multiple hospitals to provide patient data, pharmaceuticals to provide their drug data, and medical researchers to explore new treatment protocols. Neither party may want to give away its dataset, but all are interested in knowing new drug protocols that are effective in treating a disease. In this situation, a multiparty Cloud is a feasible solution. In such a scheme, multiple participants collaborate using shared hardware to accomplish their common goal. This also requires the data of each party to be kept private from other users while sharing the computed models for all to use.

A proposed framework [5] for secure multi-party computation (SMPC) has four entities, namely, proxy server, Cloud server, analyzer, and parties that are taking part in the shared computations. Upon receiving all required users' data, a central Cloud server is used to perform desired computations. A proxy server hides the identities of all users to provide anonymity. Each user can send its data for computations after authenticating into the system. The proxy server hides traceability of every message sent by each user to the Cloud server. Incoming data is encrypted to provide protection and integrity against a man-in-the-middle attack. There are obvious questions on the performance and efficiency of Cloud environments to deploy such a model while enforcing the security requirements of all user entities. We propose a sophisticated collaborative federated learning (CFL) algorithm that enables knowledge transfer among parties (clients). At the same time, our approach maintains private data locally and improves communication efficiency, as shown in Fig. 7.2.

FL enables data to stay local, while algorithms travel across participating institutions, to train a deep learning algorithm. This solution preserves privacy and security of users' data. However, with no data sharing with a central server, the training time for ML is very high as code needs to travel across multiple client sites in every iteration.

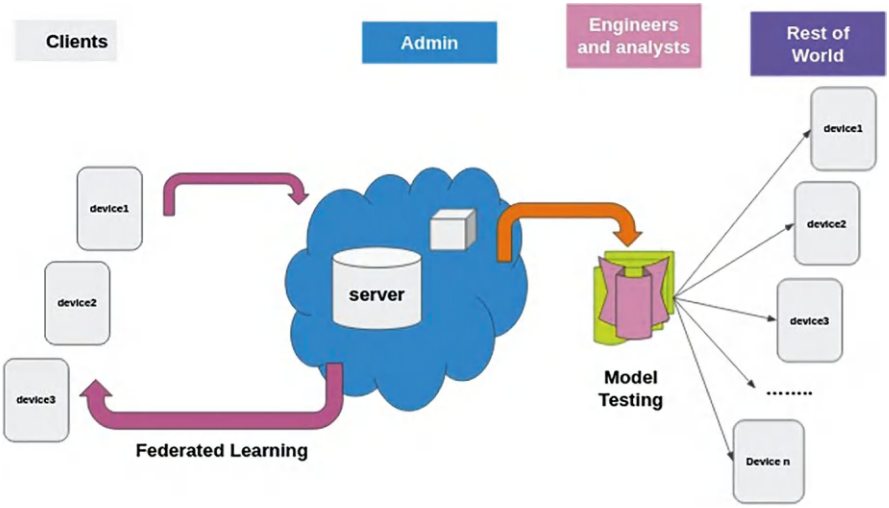


Fig. 7.2 Collaboration among clients in federated learning

Since federated learning can be implemented where shared data is collected centrally and analyzed in the central location, data can remain distributed and the analysis (i.e., shared code and neural network weights) moves from location to location. In the first case, performance is primarily limited by the one-time movement of the data and the subsequent multiple iterations are done centrally. In the second case, performance is limited by the multiple movements of the analysis mechanism (i.e., shared code and neural network weights), done once for each iteration, and the calculation is done in a distributed manner in Fig. 7.3. As the previous examples showed, at some point performance of the centralized implementation outperforms the distributed case [6]. The downside is the security and privacy issues with collecting the data centrally.

7.4 Collaborative Federated Learning (CFL)

We propose collaborative federated learning (CFL) as a combination of centralized and decentralized machine learning algorithms. Each client has data on a different set of subjects, while the data of every client has the same set of features. Examples of such data include smartphone users' word-typing histories (from the same word dictionary), which are stored on individual devices with the same dataset features and analyzed by private machine learning algorithms.

In centralized learning (CL), all the data is aggregated from multiple databases in a central location, where AI algorithms do the training and inference computations. Then results are distributed back to different clients. Each client can see only

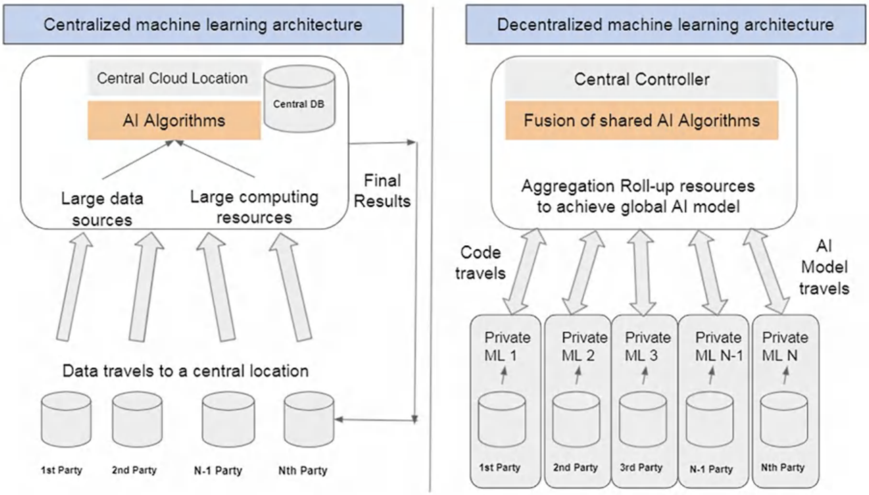


Fig. 7.3 Architecture of centralized and decentralized machine learning [6]

its own data and the final results of centralized computations, but not other clients' datasets. Hence, data and models are shared.

In decentralized or distributed learning (DL), each client keeps its data private, whereas the AI algorithm travels to each client's site, does some computations, and partial results are then copied back to a central location. From there, AI algorithms go to the next client's site, do more computations, and update the central database. A client cannot see the data of others but only accesses the shared FL model. A key difference is that if clients don't trust each other, they never have to give their private data away. In this case, data won't get shared; only FL code is shared.

We propose to divide the data into two parts: private and public, as shown in Fig. 7.4. Private data may consist of a patient's identifiable information in the healthcare domain [7]. This information can be removed and replaced by an ID, which is only known to the data-owning party [8]. We reckon that 80% of data is not private such that it can be safely shared. In a blood test report, once the patient's identification information is anonymized, the actual values of measurement in the blood sample can be shared. For a non-healthcare example, people share their work profiles, designations, etc. on social networking platforms (e.g., LinkedIn) without sharing their Social Security Number (SSN) or personal salary information. This enables some meaningful computations to be done on a shared basis without having sensitive code or data travel between the sites, resulting in unacceptable delays. Then computation results can be shared between the contributing parties, which can then do reverse mapping of public IDs to private information for their own interpretation.

Thus, applying a similar collaborative approach to our prior medical examples may result in hospitals offering personalized treatments to critical patients by comparing anonymized patients with similar cases in other hospitals for better

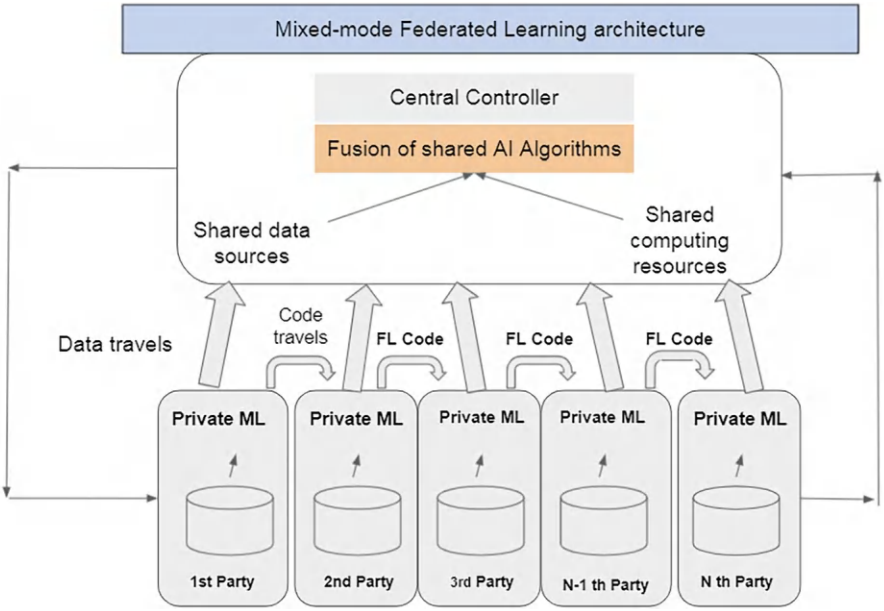


Fig. 7.4 A collaborative federated learning architecture [7]

diagnostic matches. In addition, collaborative-mode approaches also provided valuable data to pharmaceuticals and academic researchers, resulting in new treatments using AI and ML [9].

7.5 Performance of Collaborative Federated Learning

Consider three entities: a hospital = A, a drug company = B, and a medical research unit = C, with a single centralized server. All three need to communicate and share data for new drug discovery, based on which patients need help, how they react to the current drugs, and which ones may be appropriate for a future clinical trial. An obvious solution is for all three to put their data in a shared central server. We will examine the time taken for data transfers and execute a machine learning algorithm to compute neural network weights before the results are distributed to each participating party. Now assume a second case where these entities do not fully trust the others or, due to some regulations, are not willing to share their data on a central server. In this case, the data stays local and code travels to each site for the training algorithm to be completed. Next, we will examine both cases of centralized and decentralized trainings and the resulting run times as $T1$ and $T2$, respectively. Some notations are below:

t_{da} = Data copying delays from hospital A to the central server

t_{db} = Data copying delays from drug company B to the central server
t_{dc} = Data copying delays from medical company C to the central server
t_{pa} = Time for codes and weights of the neural network program to travel from the central server to hospital A
t_{pb} = Time for codes and weights to travel from the central server to drug company B
t_{pc} = Time for codes and weights to travel from the central server to medical researcher C
t_{px} = Program execution time for one iteration
n = Number of training iterations required

So, the worst-case (asynchronized) data copy time to the central database is $=t_{da} + t_{db} + t_{dc.}$ and in a completely centralized model, total worst-case run time will be $T1 = t_{da} + t_{db} + t_{dc} + n * t_{px}$

For a fully decentralized federated learning system, the total worst run time will be $T2 = n * (t_{pa} + t_{pb} + t_{pc} + t_{px})$

For larger n, $T2 \gg T1$, because in $T1$ we copy data only once to a central server where all the code runs, whereas in $T2$, the program has to travel for every iteration to access data on various local sites.

In addition, consider a third case, where a fraction of data is shared while the rest is kept local. Let this fraction be m, so $T3 = m*T1 + (1 - m)*T2$. Note that for 100% sharing, m = 1, which is also the centralized training model, and then $T3 = T1$. Similarly, for a completely decentralized model, $m = 0$, which means $T3 = T2$.

Below are the results in Fig. 7.5, when the program size is smaller than the dataset size with 50% data sharing. As expected, CFL run times fall in between the

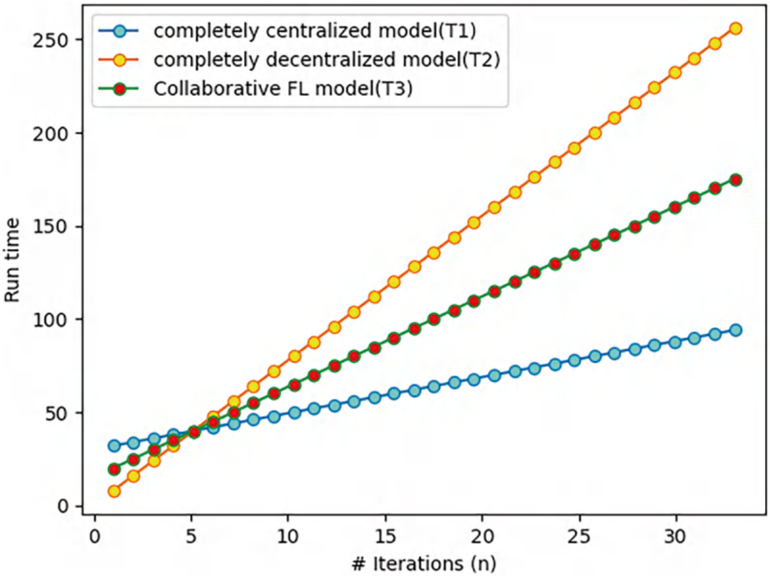


Fig. 7.5 Run time with 50% data sharing

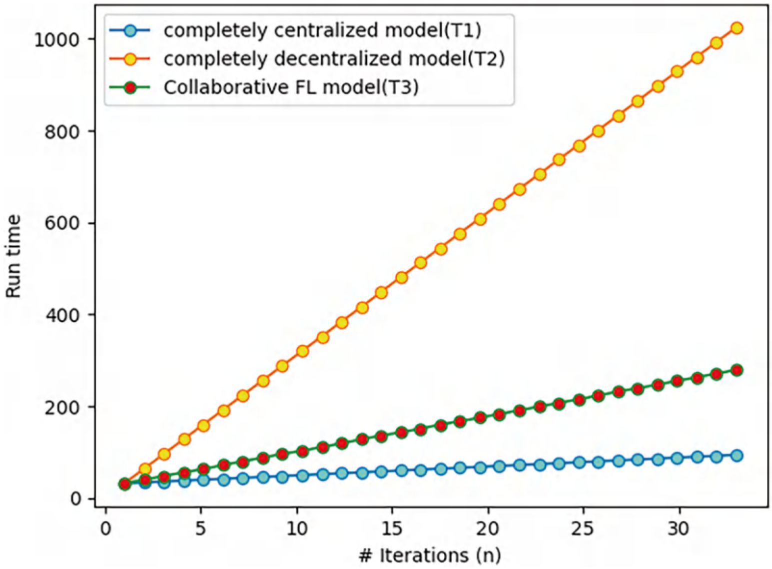


Fig. 7.6 Run time with 80% data sharing

centralized and decentralized training models. The sample code for our model is in the appendix following this chapter.

However, if 80% of data is shared centrally, the CFL model performance is closer to the centralized training case, as shown in Fig. 7.6.

In summary, a decentralized model is slower most of the time because code and model data need to travel to different locations. However, it is often preferred for security and privacy reasons. A preferred solution is partial data sharing, to keep the private data locally and share the rest of the data on a central server. This is also known as differential privacy [7].

7.6 Edge Computing Security Challenges

Unbalanced and non-identically distributed data partitioning across a massive number of unreliable devices with limited communication bandwidth poses a problem. Perimeter defense has long been insufficient for IoT security. Fixed protocols for boundaries of security with individual devices' security implementations will fail. Specifically, Edge computing exacerbates security concerns in the following ways:

1. Definition of a Cloud has been expanding and getting out of a data center.
2. Perimeter defense is insufficient, as there is no fixed perimeter.
3. Fixed protocols for boundaries of security fail in a shared security model.
4. A fixed universal security policy is inadequate, as each party owns their data.

5. Resources on Edge need to be adaptive for varying amount of compute.

It is clear that no existing FL methods can meet the above requirements, but the proposed CFL method can improve the overall security profile with a shared responsibility model. CFL also offers a crucial trade-off between runtime performance and data control considerations.

7.7 Distributed Trust Model

We need to remember that IoT devices are periodically collecting data about an environment or individuals, which can be potentially shared with third parties, compromising privacy. It can range from personal preferences of Web browsing habits, TV channel selection, or images from home security cameras. Some devices can be programmed to selectively transmit data to a Cloud service for processing, e.g., a security camera that has a buffer of 15 s but records and transmits a 30-s clip only if some motion (event activity) is detected for 15 s before and 15 s after the occurrence of the event. This reduces storage requirements but increases chances of a mistake. Such devices are designed to render service with minimal intervention, and yet they need to be directed using voice activation or image recognition. On the other hand, if there is a continuous recording dashcam, which is a forward-looking recording device in a car, the purpose of this dashcam is to establish the other party's guilt in case of an accident in a vehicle. It will also record voice conversations of passengers potentially violating their privacy rights. It is recommended for the vehicle driver to inform passengers and seek their consent in advance.

For ensuring trust in Edge computing, it has to start with a trusted environment, trusted protocols, and tamper-proof components. Vendors need to provide "anti-tamper" solutions to start with. Software upgrades in the field are needed for bug fixes during the lifetime of an Edge computing device. A secure channel must exist to provide signed binary packets that are transmitted and installed in the field, e.g., on a car or TV at home. In our door example, the vendor needs to provide an anti-tamper solution to prevent someone locally changing the firmware or settings in an unauthorized manner. Even remote software upgrades are authenticated. Otherwise, unprotected home appliances can be used to launch cyberattacks. For example, someone can open garage doors via remote Internet attacks. Besides security, there are privacy concerns, as home sensors are collecting data about individuals that can be shared with third parties for commercial purposes.

Undesirable consequences may emerge if a third party can remotely gain control of a self-driven car, causing an accident on the road, or someone with malice can access the medicine drip meters in a hospital with fatal consequences for the patients. This can be avoided with a balanced approach to interoperability and access control. This needs to be addressed at different layers of architecture and within the protocol stacks between the devices. Standardization and adoption of communication protocols should specify when it is optimal to have standards. Some vendors like to create a proprietary ecosystem of compatible IoT products. This creates user lock-in to their particular ecosystem, which from a vendor's point of view is desirable because a closed ecosystem approach can offer benefits of security and reduce costs. However, from a user's point of view, such practices can create interoperability issues in communicating with solutions from other vendors, thereby limiting the end user's choices for upgrades or future system expansion.

Solution-level cost considerations involve technical factors such as limited internal processing, memory resources, or power consumption demands. Vendors try to reduce the unit cost of devices by minimizing parts and product design costs. It may be more expensive to incorporate interoperability features and test for compliance with a standard specification. A non-interoperable device may lack standards and documented best practices, limiting the potential use of IoT devices.

7.8 Quantifying Edge Security

We look at the hardware and software stack of a simple home surveillance camera system to analyze its attack surface and threat model. Then a novel method is presented to apply series-parallel reliability calculations to propose a system security scoring computation. We close this section and the book with an analysis of methods to improve the system-level reliability.

Attackers have often exploited component-level vulnerabilities. Most security systems are designed using capability models. A capability model usually takes into account how various services are utilized. An example of such a model starts with a multidimensional representation composed of:

1. **Hardware:** an application-specific integrated circuit (ASIC) or programmable microcontroller.
2. **Operating System:** Windows, Linux, Android, etc.
3. **Applications:** nature of application and its privilege level.
4. **Setup:** Manner in which various components, services, and utilities are deployed:
   (a) Kernel, library services, file access, etc.
   (b) Manner in which objects such as username, application, and function get authenticated
   (c) The kind of cryptography used: MD5 (Message Digest Algorithm 5) vs. SHA256 (Secure Hash Algorithm with a digest size of 256 bits)

We propose to evaluate components of a given HW and SW solution of one or more Cloud-connected IoT devices based on the robustness and trustworthiness of their entire solution stack, with a multiplicative serialized model in the following order:

1. Native compiled code is trusted more than interpreted code.
2. Code uses external libraries.
3. Third-party SW is attempting to integrate with the platform.

Using the above method, it is possible for us to evaluate the trust of different operating systems with applications from diverse fields. Our goal is to create a framework for evaluating and assigning a security score to each layer, which is used to compute a composite score. An application can be disassembled to see whether it uses a kernel service, a utility in the user space, or a built-in library, etc.

For each component in the stack, a list of orthogonal properties is established, followed by an objective scoring system for each property. The numerical score for a utility function depends on the manner in which it is accessed, e.g., read (as a call by value) or write (a call by reference). A security score can be computed by answering a set of questions by a user or automatically computed by a testing tool. Examples of questions include:

- Whether a salt is used for hash passwords?
- Which algorithm is used for hashing: MD5 or SHA256?
- Does the communication channel use SSL, and which version of TLS is being used?
- What is the version of MYSQL in operation?

Another security score determination method is whether port 3306 used by MySQL is open to the world or just to the application servers that use the MySQL database. This score can be continuously updated during the operations. More importantly, it needs to be updated after a maintenance or upgrade action is completed.

Security score questionnaires may also focus on the best practices during development. Automated score calculation focuses on the system operations hygiene: an OS without the latest patch can be at a security risk.

Security score computations have two outputs:

1. **Probability of a successful attack:** What is the probability that an attack on this device will succeed?
2. **Probable impact of a successful attack**: What is the probable impact if the attack succeeds?

Here:

Pa = Probability of the attack from 0 to 1
Pi = Probable impact of a successful attack from 0 to 1

Thus, $Pa*Pi$ = the expected loss

The security score (S) can be computed as follows:

$$S = 1 - Pa * Pi,$$

This is the score for a single component. By describing the series-parallel relationships among the different components and their individual security scores, the whole system security score can be computed.

The factors that affect the probability of an attack include:

- Presence of an existing vulnerability that is known to attackers
- Hackers' focus on products of this type
- History of exploitation of this type of product

The probable impact of a security attack is defined as the sum of any regulatory fines, reputational damage, and operational losses, representing a loss of trust in the product and services. This needs constant monitoring for security breaches and policy updating [10].

The first step in the modeling of probable impact is to describe the whole system in terms of its components, hierarchy, organization, and security-wise connections between the components, which could be in series or parallel.

The directional lines in the figure below represent the security-wise relationship between different blocks in a system. In Fig. 7.7, all the blocks should be secure for the system to be secure and provide the required functionality. In Fig. 7.8, any one block should be available for the system to provide the required functionality. Composite security score can be computed by applying series-parallel reliability rules [7, 11], as shown in Figs. 7.7 and 7.8:

7.8.1 An Example of Trust-Based Security Scoring

Raspberry *Pi* is the de facto choice and starting point for many IoT devices. This choice is driven by its ubiquity and low price, making it a popular controller for many home and entry-level appliances. A higher installed base also makes it an attractive target for hackers, making it a good evaluation choice for our IoT trust model. For our sample system, we restricted probability values to high (0.9), medium (0.6), and low (0.3). Similarly, the impact values were also high (0.9), medium (0.6), and low (0.3).

We took an implementation of a Raspberry *Pi* Model 3B with Raspbian OS Ver. 4.14, released on April 18, 2018, as a reference system for trust-based scoring [12]. The base Raspberry Pi system comes with a microSD card, which holds the OS and can be used to install additional software. The factory settings and factory-shipped software packages for the OS were used for trust scoring. No packages were updated. Once the basic model trust scoring was complete, we proceeded to complete the Raspberry Pi-based security camera setup. We added the software components listed below, and the data flow for the system is depicted in Fig. 7.9.

1. MongoDB
2. Rabbit MQ

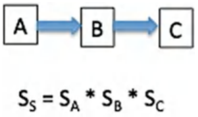


Fig. 7.7 Risk levels of series systems

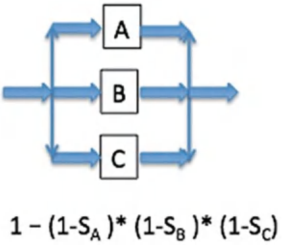


Fig. 7.8 Risk levels of parallel systems

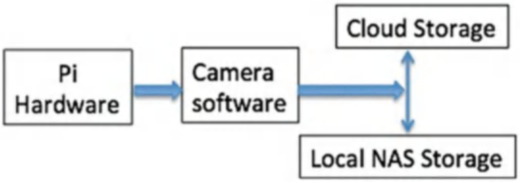


Fig. 7.9 Series-parallel implementation of our prototype

3. AWS IoT client
4. MotionPie software

We use MongoDB to store images on a network-attached storage (NAS) drive. Cloud storage uses Amazon's backend services and is accessed by an AWS IoT client. Motion Pie image processing software is used to detect motion and decide which video clips are saved or discarded. Note that camera images in Fig. 7.9 are stored simultaneously and immediately in both the local and Cloud storage locations.

A problem with this security camera prototype, as shown in Fig. 7.10, is that someone with physical access to the local system can easily switch off the system or alter the system software. There is no authentication of system software at boot time, so the base hardware setup has a high probability (0.9) of an attack. The impact probability of such an attack is also high (0.9), as the base system can be fully compromised.

In the default setup, the username is "admin," and the password is blank. It is easy for someone to remotely hijack the system and use its camera in a Mirai botnet attack [4]. Once the password has been changed and the camera is moved behind a secure firewall, the probability of such an attack is lowered to medium (0.6). However, the impact probability remains high (0.9). Our proposed system uses the AWS IoT security model [14], compliant with X.509 certification with asymmetric keys [4]. The backend environment used to store images is protected by high security, so the probability of an attack is low (0.3). The impact probability is also low (0.3) because the images are stored in two physically separate places: local storage and Cloud-based storage. As a consequence, the higher attack (0.9) and impact (0.6) probabilities of the local system do not sway the overall assessment.

Overall, we have the Raspberry hardware and software components in series security-wise, which itself is in series with two parallel storage systems security-wise. Note that *Sc* is the security of the camera, *Ss* is the security of the software,

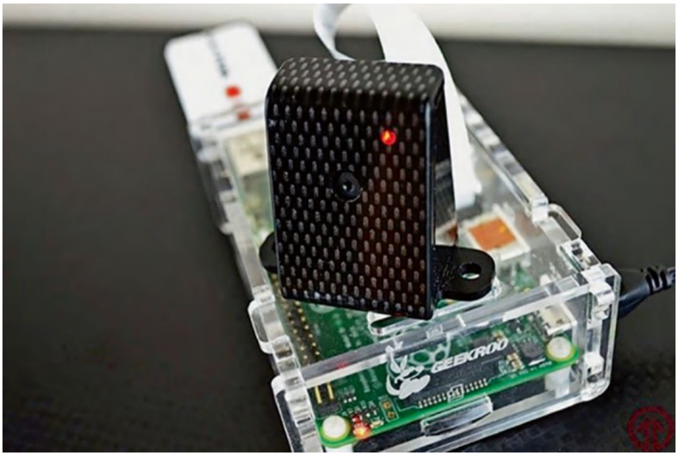


Fig. 7.10 A simple Raspberry *Pi*-based camera system [13]

Sgc is the security of Cloud storage, and *Sgl* is the security of local storage. At component level, here is what we have so far:

$$Sc = 1 - (0.9 * 0.9) = 1 - 0.81 = 0.19$$

$$Ss = 1 - (0.6 * 0.9) = 1 - 0.54 = 0.46$$

And for the storage systems:

$$Sgc = 1 - (0.3 * 0.3) = 0.91$$

$$Sgl = 1 - (0.9 * 0.6) = 0.46$$

As Cloud storage and local storage are in parallel, as shown in Fig. 7.9, and they provide redundant functionality, their combined security score can be computed using the reliability parallel chaining rule:

$$\begin{aligned} Sg &= 1 - (1 - Sgc) * (1 - Sgl) \\ &= 1 - (1 - 0.91) * (1 - 0.46) \\ &= 1 - (0.09 * 0.54) = 0.9514 \end{aligned}$$

Finally, the end-to-end system-level security protection score for an attack is a composite of three scores = *Sc***Ss***Sg* = 0.19∗0.46∗0.9514 = 0.07, which is only 7% or very low. This means that the entire camera system is prone to attacks. However, we can still use it due to our added security measures of a strengthened password, dual storage on local HW and the Cloud, and moving the camera behind a secure firewall. Thus, one of the two paths needs to be secured to continue the required functionality:

$$\text{Path A : Camera} \rightarrow \text{Software} \rightarrow \text{Local } NAS$$

$$\text{Path B : Camera} \rightarrow \text{Software} \rightarrow \text{Cloud Storage}$$

Note that all past images will still be preserved even if the system is compromised up to the point of intrusion, say if someone physically removes the microSD card on a security camera. If a home or business uses such a system, it may need multiple cameras, so if one of them is compromised, others will continue the surveillance. An example setup would consist of a system with five *Pi* cameras monitoring the same location, using shared local NAS, and common Cloud storage. The improved security score for the camera and software part is computed as $1 - (1\text{–}0.19{*}0.46)^5 = 0.36$. Note that we have parallel image streaming from five cameras going to a single Cloud and local storage system. The entire system security will be 0.36∗0.9514 = 0.34, or 34%, improving the total system security.

Another way to achieve better security is by making it harder to compromise a single camera system, say by putting it in a cage with a backup battery, so its microSD card can't be easily replaced and the system remains powered on. This action drives the probability of a physical attack lower, from high to low, such that $Sc = 1\text{–}0.3 * 0.9 = 1\text{–}0.27 = 0.73$. The overall score for such a single camera system would be $= Sc{*}Ss{*}Sg = 0.73{*}0.46{*}0.9514 = 0.32$, or 32%, which is almost the same as our five parallel camera systems and at a much lower cost. A system with one camera also presents a single point of failure, making a combination of multiple cameras with physical security a better approach. Both redundancy and cost control can be achieved by using just two physically secured camera systems in parallel instead of five cameras.

7.9 Summary

Edge computing represents a combination of distributed computing connected to centralized servers. Actors on the Edge may interact with each other as well as a central data center. Their concerns include multiple subtopics, e.g., protecting information content from observation and alteration, protection of operational capability from unauthorized access, protection of normal operation in the presence of malicious overloaded requests, etc. Therefore, trust requires a distributed solution. In centralized learning, the central server potentially represents a single point of failure, which is one of the bottlenecks for performance as well. Another issue is the need for all participants to trust the central authority with their datasets. In contrast, a decentralized federated learning solution needs parties to run a common binary on each of their datasets and trust the incoming program, thus avoiding a single point of failure but potentially creating a security hazard with malicious code. Another issue is the long training run time due to multiple hops between different dataset locations. In this chapter, we propose a novel collaborative federated learning (CFL) solution that combines the advantages of centralized and decentralized federated schemes without compromising security. We concluded this chapter with a method to compute security scores for composite hardware and software systems using series and parallel methods. This enables system architects to compare security of various proposals to make appropriate choices between cost, performance, and security.

7.10 Points to Ponder

1. How can one improve Cloud performance and support for IoT?
2. Why is Edge computing needed for self-driven cars in the future?
3. What is the trust and security model for Edge devices?
4. Who owns data in a secure multi-party Cloud (SMPC)?
5. Can hardware be the sole root of trust?

7.11 Answers

1. How can one improve Cloud performance and support for IoT?

    - By having distributed and redundant systems for a failsafe solution.
    - Avoid having a single point of failure.
    - Backend Cloud Services are needed to log data and results for audits and machine learning inferences.
    - Sensors can generate enormous data, requiring Cloud storage and compute power. However, moving data in and out of Cloud is slow and expensive. So, input-output considerations will require local compute and storage power.

2. Why is Edge computing needed for self-driven cars in the future?

    - Sensors in a moving car can generate enormous data, requiring Cloud storage and compute power. Examples of this are forward-looking and side-view cameras.
    - However, moving data in and out of Cloud is slow and expensive. A car may need to react quickly due to changing road conditions.
    - So, input-output considerations for the sensor data will require local compute and storage power.
    - Any learning and performance data can be reconciled with backend servers during the night or when the car is safely parked.

3. What is the trust and security model for Edge devices?

- It has been shown that an army of botnets (a term used for devices on the Internet) can be hijacked by hackers and used for launching distributed denial-of-service (DDOS) attacks on unsuspecting Cloud servers.
- An example is of home surveillance cameras that had unsecured IP addresses used for bringing down a security journalist's blog site.
- So, security and trust models for Edge need to account for local vulnerabilities for the devices. A method to compute system-level security was shown in this chapter.

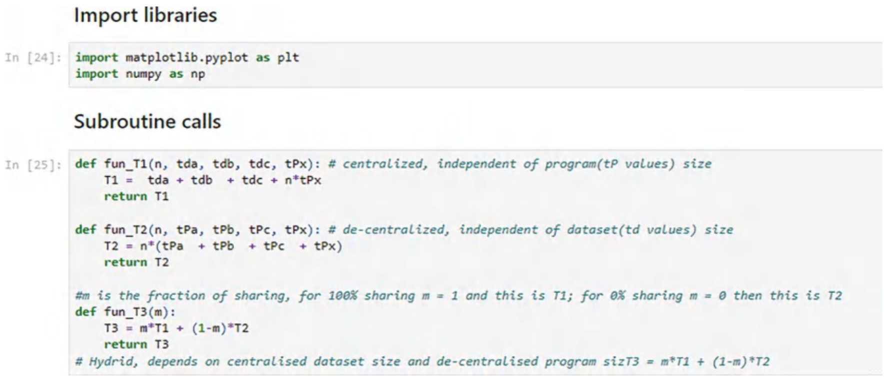
4. Who owns data in a secure multi-party Cloud (SMPC)?

   - No single party owns the entire dataset in a SMPC environment, as each contributes a subset for the common good.
   - All participants have the right to use others' data for their computations and can only extract results in an agreed-upon output format.
   - Any personally identifiable information (PII) must be removed from the output.

5. Can hardware be the sole root of trust?

   - Having any single piece of hardware or software as the sole root of trust is risky.
   - Multifactor authentication (MFA) offers a better defense strategy.
   - Another possible solution is mutual attestation by various devices that are not located at the same place or are not under the same control.
   - Thus, an attacker will need to simultaneously compromise multiple devices, which is harder to accomplish than altering any single root of trust.

7.12 Sample Code for CFL Performance



Import libraries

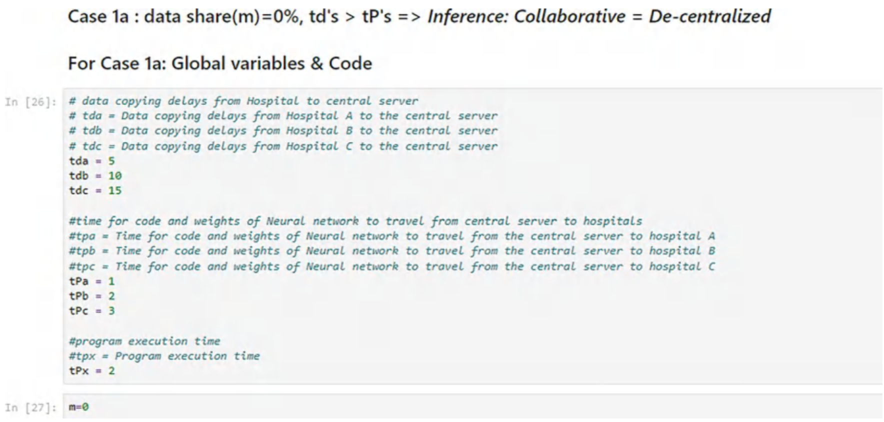
In [24]:

```
import matplotlib.pyplot as plt
import numpy as np
```

Subroutine calls

In [25]:

```
def fun_T1(n, tda, tdb, tdc, tPx): # centralized, independent of program(tP values) size
    T1 =  tda + tdb  + tdc + n*tPx
    return T1

def fun_T2(n, tPa, tPb, tPc, tPx): # de-centralized, independent of dataset(td values) size
    T2 = n*(tPa  + tPb  + tPc  + tPx)
    return T2

#m is the fraction of sharing, for 100% sharing m = 1 and this is T1; for 0% sharing m = 0 then this is T2
def fun_T3(m):
    T3 = m*T1 + (1-m)*T2
    return T3
# Hydrid, depends on centralised dataset size and de-centralised program sizT3 = m*T1 + (1-m)*T2
```



Case 1a : data share(m)=0%, td's > tP's => *Inference: Collaborative = De-centralized*

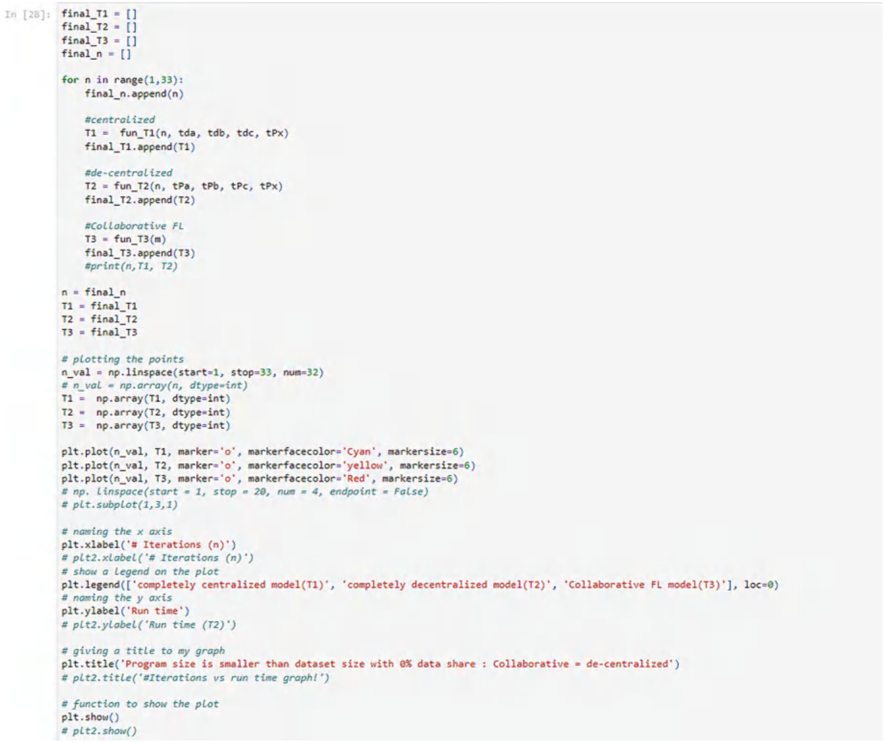
For Case 1a: Global variables & Code

In [26]:

```
# data copying delays from Hospital to central server
# tda = Data copying delays from Hospital A to the central server
# tdb = Data copying delays from Hospital B to the central server
# tdc = Data copying delays from Hospital C to the central server
tda = 5
tdb = 10
tdc = 15

#time for code and weights of Neural network to travel from central server to hospitals
#tpa = Time for code and weights of Neural network to travel from the central server to hospital A
#tpb = Time for code and weights of Neural network to travel from the central server to hospital B
#tpc = Time for code and weights of Neural network to travel from the central server to hospital C
tPa = 1
tPb = 2
tPc = 3

#program execution time
#tpx = Program execution time
tPx = 2
```

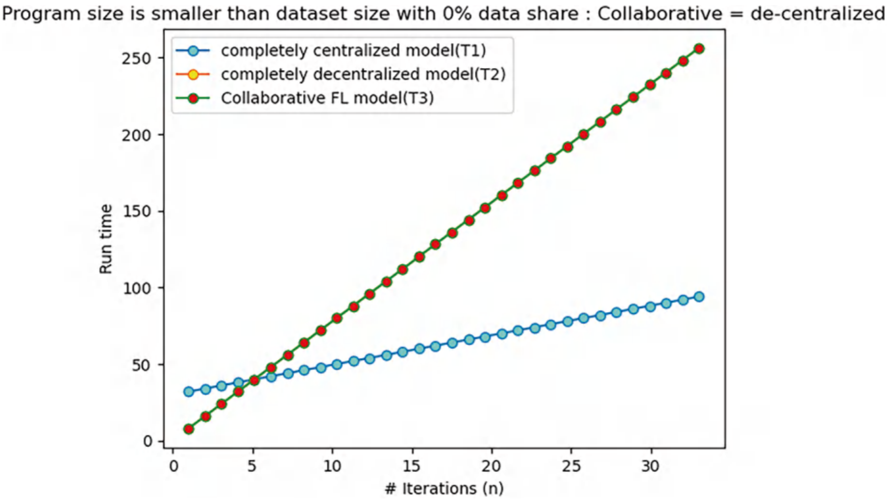
In [27]:

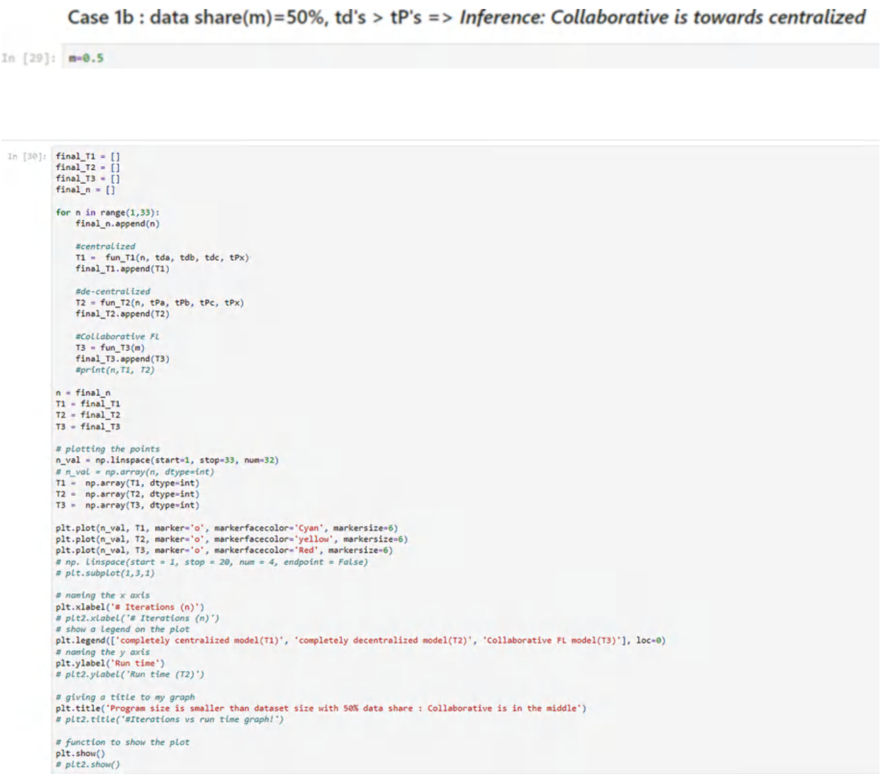
```
m=0
```



In [28]:

```
final_T1 = []
final_T2 = []
final_T3 = []
final_n = []

for n in range(1,33):
    final_n.append(n)

    #centralized
    T1 =  fun_T1(n, tda, tdb, tdc, tPx)
    final_T1.append(T1)

    #de-centralized
    T2 = fun_T2(n, tPa, tPb, tPc, tPx)
    final_T2.append(T2)

    #Collaborative FL
    T3 = fun_T3(m)
    final_T3.append(T3)
    #print(n,T1, T2)

n = final_n
T1 = final_T1
T2 = final_T2
T3 = final_T3

# plotting the points
n_val = np.linspace(start=1, stop=33, num=32)
# n_val = np.array(n, dtype=int)
T1 =  np.array(T1, dtype=int)
T2 =  np.array(T2, dtype=int)
T3 =  np.array(T3, dtype=int)

plt.plot(n_val, T1, marker='o', markerfacecolor='Cyan', markersize=6)
plt.plot(n_val, T2, marker='o', markerfacecolor='yellow', markersize=6)
plt.plot(n_val, T3, marker='o', markerfacecolor='Red', markersize=6)
# np. linspace(start = 1, stop = 20, num = 4, endpoint = False)
# plt.subplot(1,3,1)

# naming the x axis
plt.xlabel('# Iterations (n)')
# plt2.xlabel('# Iterations (n)')
# show a legend on the plot
plt.legend(['completely centralized model(T1)', 'completely decentralized model(T2)', 'Collaborative FL model(T3)'], loc=0)
# naming the y axis
plt.ylabel('Run time')
# plt2.ylabel('Run time (T2)')

# giving a title to my graph
plt.title('Program size is smaller than dataset size with 0% data share : Collaborative = de-centralized')
# plt2.title('#Iterations vs run time graph!')

# function to show the plot
plt.show()
# plt2.show()
```

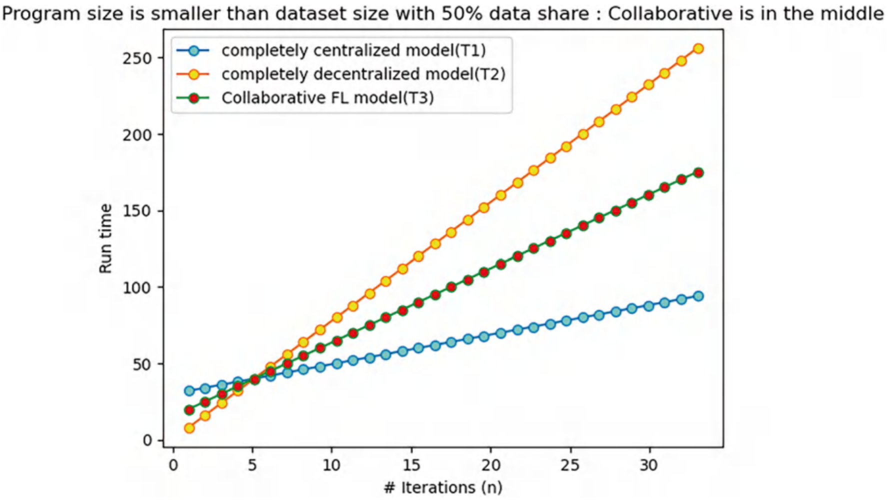


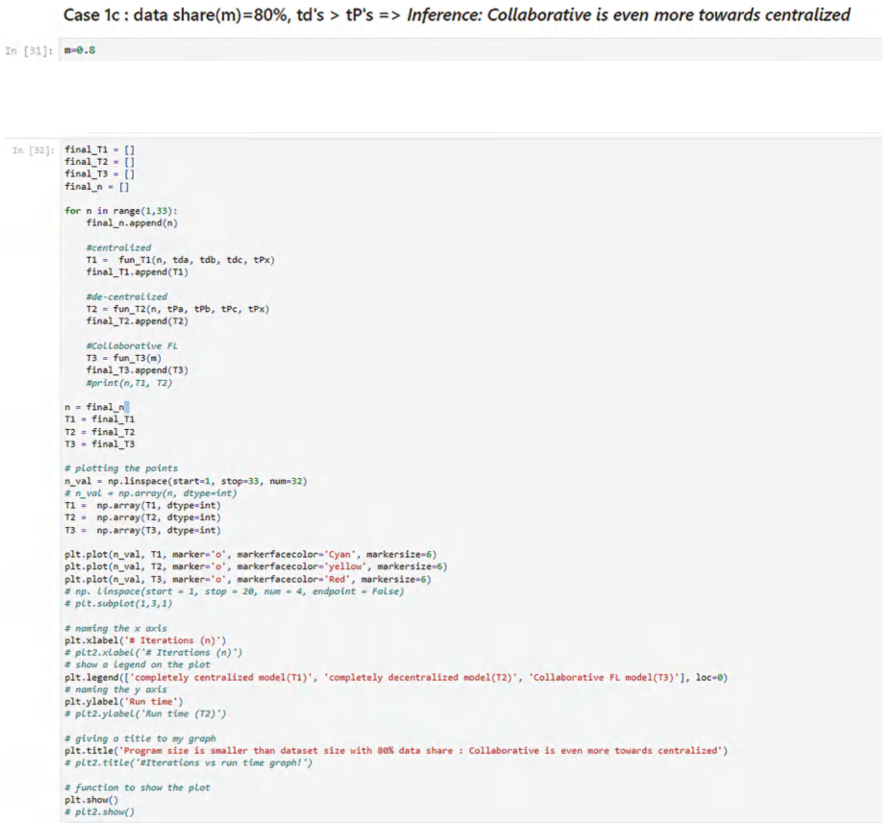


Case 1b : data share(m)=50%, td's > tP's => *Inference: Collaborative is towards centralized*

```
In [29]: m=0.5
```

```
In [30]: final_T1 = []
final_T2 = []
final_T3 = []
final_n = []

for n in range(1,33):
    final_n.append(n)

    #centralized
    T1 =  fun_T1(n, tda, tdb, tdc, tPx)
    final_T1.append(T1)

    #de-centralized
    T2 = fun_T2(n, tPa, tPb, tPc, tPx)
    final_T2.append(T2)

    #Collaborative FL
    T3 = fun_T3(m)
    final_T3.append(T3)
    #print(n,T1, T2)

n = final_n
T1 = final_T1
T2 = final_T2
T3 = final_T3

# plotting the points
n_val = np.linspace(start=1, stop=33, num=32)
# n_val = np.array(n, dtype=int)
T1 =  np.array(T1, dtype=int)
T2 =  np.array(T2, dtype=int)
T3 =  np.array(T3, dtype=int)

plt.plot(n_val, T1, marker='o', markerfacecolor='Cyan', markersize=6)
plt.plot(n_val, T2, marker='o', markerfacecolor='yellow', markersize=6)
plt.plot(n_val, T3, marker='o', markerfacecolor='Red', markersize=6)
# np. linspace(start = 1, stop = 20, num = 4, endpoint = False)
# plt.subplot(1,3,1)

# naming the x axis
plt.xlabel('# Iterations (n)')
# plt2.xlabel('# Iterations (n)')
# show a legend on the plot
plt.legend(['completely centralized model(T1)', 'completely decentralized model(T2)', 'Collaborative FL model(T3)'], loc=0)
# naming the y axis
plt.ylabel('Run time')
# plt2.ylabel('Run time (T2)')

# giving a title to my graph
plt.title('Program size is smaller than dataset size with 50% data share : Collaborative is in the middle')
# plt2.title('#Iterations vs run time graph!')

# function to show the plot
plt.show()
# plt2.show()
```

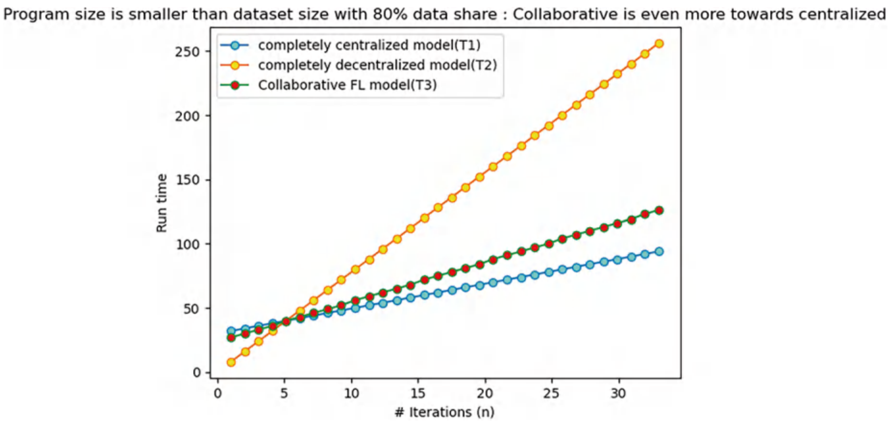


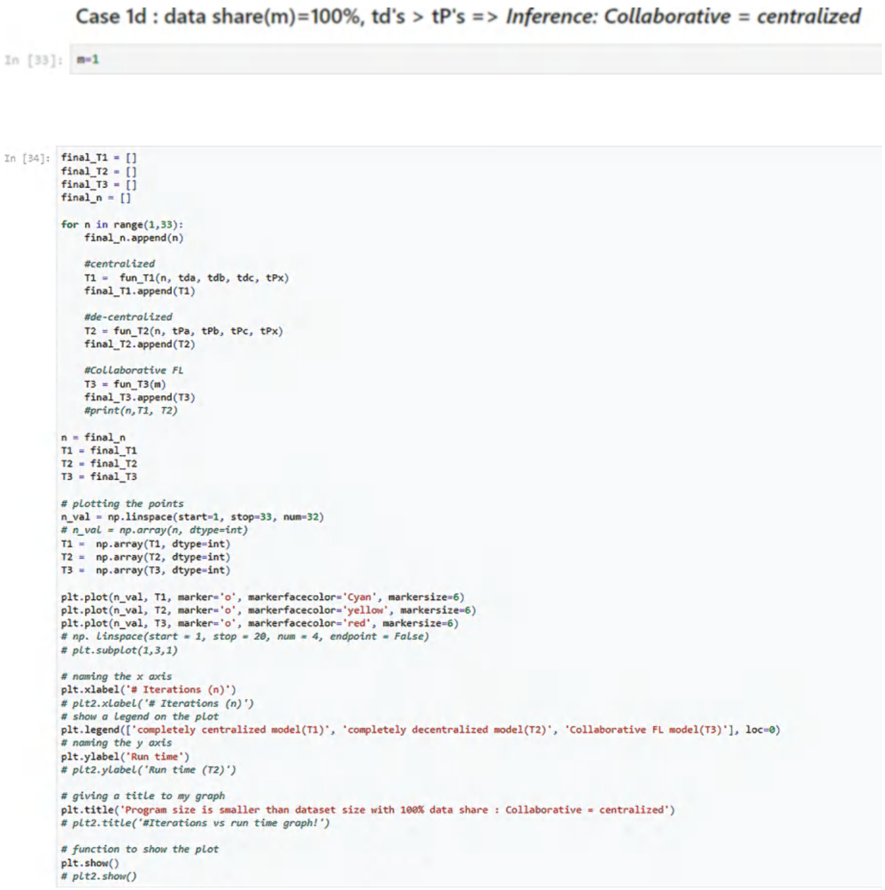


Case 1c : data share(m)=80%, td's > tP's => *Inference: Collaborative is even more towards centralized*

```
In [31]: m=0.8
```

```
In [32]: final_T1 = []
final_T2 = []
final_T3 = []
final_n = []

for n in range(1,33):
    final_n.append(n)

    #centralized
    T1 =  fun_T1(n, tda, tdb, tdc, tPx)
    final_T1.append(T1)

    #de-centralized
    T2 = fun_T2(n, tPa, tPb, tPc, tPx)
    final_T2.append(T2)

    #Collaborative FL
    T3 = fun_T3(m)
    final_T3.append(T3)
    #print(n,T1, T2)

n = final_n
T1 = final_T1
T2 = final_T2
T3 = final_T3

# plotting the points
n_val = np.linspace(start=1, stop=33, num=32)
# n_val = np.array(n, dtype=int)
T1 =  np.array(T1, dtype=int)
T2 =  np.array(T2, dtype=int)
T3 =  np.array(T3, dtype=int)

plt.plot(n_val, T1, marker='o', markerfacecolor='Cyan', markersize=6)
plt.plot(n_val, T2, marker='o', markerfacecolor='yellow', markersize=6)
plt.plot(n_val, T3, marker='o', markerfacecolor='Red', markersize=6)
# np. linspace(start = 1, stop = 20, num = 4, endpoint = False)
# plt.subplot(1,3,1)

# naming the x axis
plt.xlabel('# Iterations (n)')
# plt2.xlabel('# Iterations (n)')
# show a legend on the plot
plt.legend(['completely centralized model(T1)', 'completely decentralized model(T2)', 'Collaborative FL model(T3)'], loc=0)
# naming the y axis
plt.ylabel('Run time')
# plt2.ylabel('Run time (T2)')

# giving a title to my graph
plt.title('Program size is smaller than dataset size with 80% data share : Collaborative is even more towards centralized')
# plt2.title('#Iterations vs run time graph!')

# function to show the plot
plt.show()
# plt2.show()
```

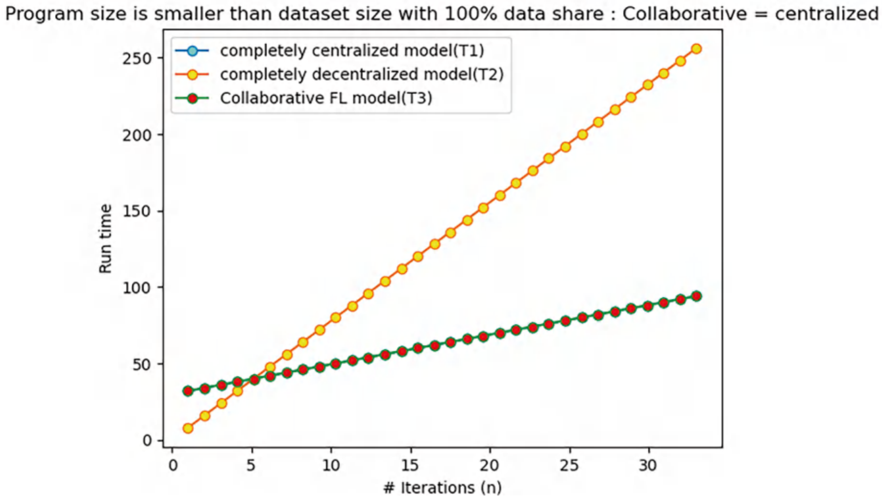




Case 1d : data share(m)=100%, td's > tP's => *Inference: Collaborative = centralized*

```
In [33]: m=1
```

```
In [34]: final_T1 = []
final_T2 = []
final_T3 = []
final_n = []

for n in range(1,33):
    final_n.append(n)

    #centralized
    T1 =  fun_T1(n, tda, tdb, tdc, tPx)
    final_T1.append(T1)

    #de-centralized
    T2 = fun_T2(n, tPa, tPb, tPc, tPx)
    final_T2.append(T2)

    #Collaborative FL
    T3 = fun_T3(m)
    final_T3.append(T3)
    #print(n,T1, T2)

n = final_n
T1 = final_T1
T2 = final_T2
T3 = final_T3

# plotting the points
n_val = np.linspace(start=1, stop=33, num=32)
# n_val = np.array(n, dtype=int)
T1 =  np.array(T1, dtype=int)
T2 =  np.array(T2, dtype=int)
T3 =  np.array(T3, dtype=int)

plt.plot(n_val, T1, marker='o', markerfacecolor='Cyan', markersize=6)
plt.plot(n_val, T2, marker='o', markerfacecolor='yellow', markersize=6)
plt.plot(n_val, T3, marker='o', markerfacecolor='red', markersize=6)
# np. linspace(start = 1, stop = 20, num = 4, endpoint = False)
# plt.subplot(1,3,1)

# naming the x axis
plt.xlabel('# Iterations (n)')
# plt2.xlabel('# Iterations (n)')
# show a legend on the plot
plt.legend(['completely centralized model(T1)', 'completely decentralized model(T2)', 'Collaborative FL model(T3)'], loc=0)
# naming the y axis
plt.ylabel('Run time')
# plt2.ylabel('Run time (T2)')

# giving a title to my graph
plt.title('Program size is smaller than dataset size with 100% data share : Collaborative = centralized')
# plt2.title('#Iterations vs run time graph!')

# function to show the plot
plt.show()
# plt2.show()
```



References

1. Konecny, J., McMahan, H. B., & Ramage, D. (2015). *Federated optimization: Distributed optimization beyond the datacenter.* https://arxiv.org/pdf/1511.03575.pdf.
2. Kairouz, P., McMahan, H. B., Avent, B., et al. (2021). *Advances and open problems in federated learning*. Now Foundations and Trends.
3. https://www.nytimes.com/2017/01/19/business/tesla-model-s-autopilot-fatal-crash.html
4. http://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/
5. Pussewalage, H. S. G., Ranaweera, P. S., Oleshchuk, V. A., & Balapuwaduge, I. A. M. (2016) *Secure multi-party based cloud computing framework for statistical data analysis of encrypted data*. http://dl.ifip.org/db/conf/icin/icin2016/1570221695.pdf
6. *Illustration of different machine learning architectures*. https://www.researchgate.net/figure/The-illustration-of-different-machine-learning-architectures-a-Centralized-ML_fig5_339481056
7. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2022). *Cloud computing with security and scalability*. https://link.springer.com/book/10.1007/978-3-031-07242-0.
8. *De-identified data*. https://www.hopkinsmedicine.org/institutional review board/hipaa research/de_identified_data.html
9. Gupta, P., & Sehgal, N. K. (2021). *Introduction to machine learning in the cloud with python.* https://link.springer.com/book/10.1007/978-3-030-71270-9
10. Sandhu, R., Sohal, A. S., & Sood, S. K. (2017). Identification of malicious edge devices in fog computing environments. *Information Security Journal: A Global Perspective, 26*(5), 213–228.
11. https://link.springer.com/book/10.1007/978-1-4899-1860-4
12. https://www.raspberrypi.org/downloads/raspbian/
13. https://pimylifeup.com/raspberry-pi-security-camera/
14. https://aws.amazon.com/iot-core/

Chapter 8
Intelligent Edge Computing: Design Use Cases

8.1 Introduction

In Chap. 3, we reviewed some of the use cases for Intelligent IoT devices. We discussed the functionality and a high-level connectivity diagram to illustrate various components used to build systems using such devices.

In this chapter, we will look at some use cases for Intelligent Edge computing. For each use case, we will give a high-level description and describe the best practices, architecture, and system design. The best practices section will detail regulatory compliances, security, and privacy of the data and correct ways to document the complete requirements and workflows for the system. The architecture section will define the system's architecture to ensure that it is compliant with the best practices, supports interoperability with other systems and devices, follows industry standards, and uses design patterns to ensure it is easily implementable. The last section will cover system design principles to support industry standards, interoperability, and how to avoid falling into the trap of proprietary products and protocols.

We have picked three use cases covering:

1. *Smart Building Energy Management System*: It describes an energy management system (EMS) that can be deployed in a smart building to optimize energy usage without inconveniencing the occupants.
2. *Medical Data Sharing by Hospitals*: The data related to patients needs to be shared using the Cloud to allow AI and machine learning (ML)-based solutions to look for hidden data patterns. This also enables collaboration with other medical professionals to decide the course of treatment for a patient. This brings in the issues related to data privacy, security, and compliance with Health Insurance Portability and Accountability Act (HIPAA) and General Data Protection Regulation (GDPR) laws. In this case study, we shall describe how to design the system so that it is compliant with the regulatory requirements while enabling collaboration and use of AI/ML algorithms in the Cloud.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_8

3. *Solar Energy Power Plant Management*: It describes an intelligent system to reduce the cost and improve the effectiveness of operations and maintenance (O&M) for a solar energy power plant. It also covers the combined photovoltaic (PV) and energy storage systems to increase the power plant's performance.

In the next section, let us start by looking at the smart building energy management system.

8.2 Smart Building Energy Management System

Energy management is a proactive process involving systematic coordination of procurement, conversion, distribution, and use of energy to meet the requirements, considering environmental and economic objectives. For example, using daylighting responsive controls for electrical lighting as well as adjusting the heating, ventilation, and air-conditioning (HVAC) operational schedule in response to weather changes.

The continuing growth of energy usage by commercial buildings has created a need to develop innovative techniques to reduce and optimize energy usage. Buildings are increasingly being made intelligent by installing systems that encourage sustainable technologies and decrease carbon emissions and operational costs but at the same time increase the productivity, well-being, and comfort of the occupants. The buildings can save up to 29% of total energy costs [1] through the implementation of energy management systems.

Often there is confusion about building management systems (BMS) and building energy management systems (BEMS). They are sometimes used interchangeably. It is important to understand the difference between the two:

1. *Building Management System (BMS)*: It is a centralized, computer-based control system that monitors and controls a building's mechanical and electrical equipment, such as HVAC, lighting, power, and security systems. These systems enable building managers to optimize the performance of the equipment, enhance occupants' comfort, and reduce energy consumption.
2. *Building Energy Management System (BEMS)*: It is a specialized subset of BMS that focuses on monitoring, controlling, and optimizing the energy usage of the building. It integrates with various building systems and energy sources to provide a holistic view of energy consumption and identify improvement opportunities.

Before going further into BEMS, it is important to understand the building concepts based on design goals, as intelligent buildings are designed to meet different goals. Figure 8.1 depicts three types of intelligent building concepts, namely, green buildings, net zero buildings, and smart buildings.

1. *Green Buildings*: These are designed to be environmentally friendly.

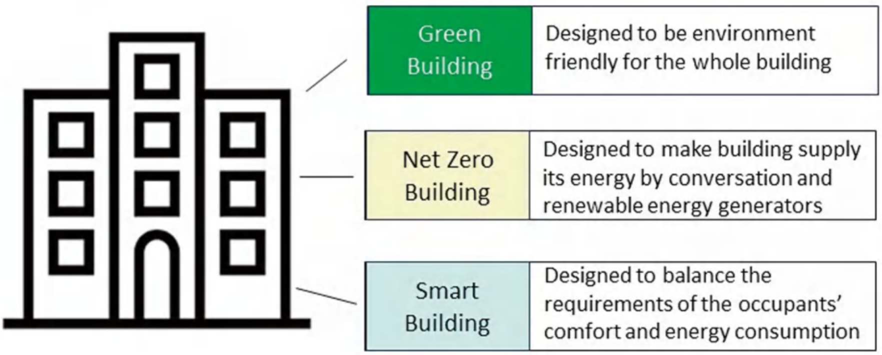

Green Building
Designed to be environment friendly for the whole building
Net Zero Building
Designed to make building supply its energy by conversation and renewable energy generators
Smart Building
Designed to balance the requirements of the occupants' comfort and energy consumption

Fig. 8.1 Building concepts based on design goals

2. *Net Zero Buildings*: These are designed to make the building supply its energy by conservation and renewable energy generators.
3. *Smart Buildings*: These are designed to balance the requirements of the occupants' comfort and energy consumption.

BEMS has existed in various forms since the 1970s. Initially, these systems monitored and controlled mainly HVAC systems. They used electronic-based equipment to control and manage HVAC systems.

Current BEMS are mostly installed in smart buildings with automated smart systems. They use digital technologies like sensor technologies, data analytics, and machine learning [2]. These advancements enable BEMS to become more advanced and effective at reducing energy consumption by allowing real-time monitoring and control of a wider range of building systems, like lighting, HVAC, security, and fire safety. As a result of this integration, various building systems can be controlled, optimized, and coordinated effectively, allowing a holistic approach to building management. Figure 8.2 describes a modern BEMS as a hierarchical system of control layers/levels.

8.2.1 Best Practices for Energy Management

Three major factors have driven BEMS adoption:

1. *Advancement in Technology*: Development of new technologies for making sensors and actuators. The availability of faster communication through 4G and 5G networks has reduced latency. It is now possible to carry out data analytics and run machine learning algorithms in the Cloud to bring in efficiency.
2. *Increased Energy Prices*: The cost of fossil fuel is increasing worldwide and has adverse effects on the environment, forcing governments to put in levies to reduce carbon emissions.

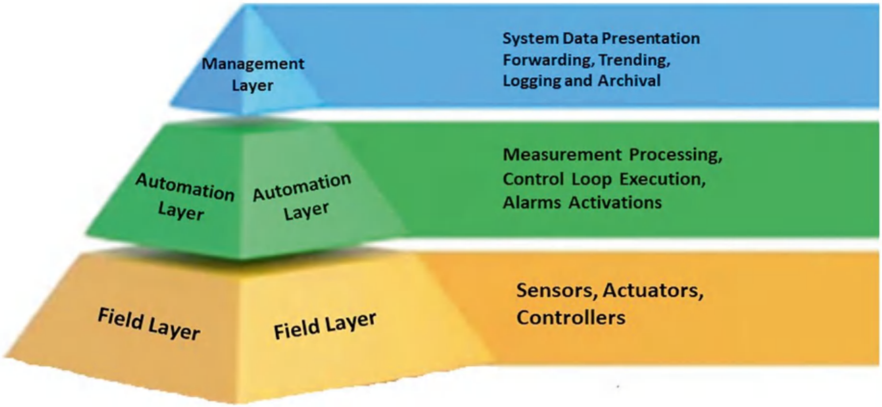


Fig. 8.2 Control layers in a building energy management system

3. *Government Regulations*: Across the world and especially in Europe and the USA, governments are coming up with Energy Performance of Buildings Directives [3], which impose certain energy efficiency standards on all new and existing buildings [4].

While thinking of implementing a BEMS, one should consider every aspect of the delivery, right from planning and implementation to post-occupancy administration, including maintenance. The aspects to be considered include the building, people, environments, systems, usage time, and budget. Figure 8.3 shows the interaction among these factors [5]. The occupant(s) may interact with BEMS to satisfy the overall comfort needs (including thermal, visual, acoustic, and indoor air quality). This interaction may result in a significant change in energy consumption.

The BEMS should be designed in such a manner that it monitors and analyzes every aspect of energy consumption, right from lighting to air circulation, heating, cooling, etc. It should help in identifying where energy savings can be made. The requirements, workflows, and other factors should be well documented while designing the BEMS. The best practice is to capture these requirements in a tabular form, as shown in Table 8.1. The specific requirements elicited from stakeholders should be tabulated and assigned appropriate priority, frequency of operations, and area of functionality. This helps in streamlining the design and development process, thereby helping in tracking them. It is always better to include links to use case documentation and other key reference material as needed to make the requirements as complete and understandable as possible. The value of priority of the requirement is given in Table 8.2.

Table 8.3 enumerates some of the requirements of BEMS in the format specified by Table 8.1. It is not comprehensive, and it is just a sample to illustrate how requirements should be captured. The requirements should be categorized as follows:

- Base requirements
- Security requirements

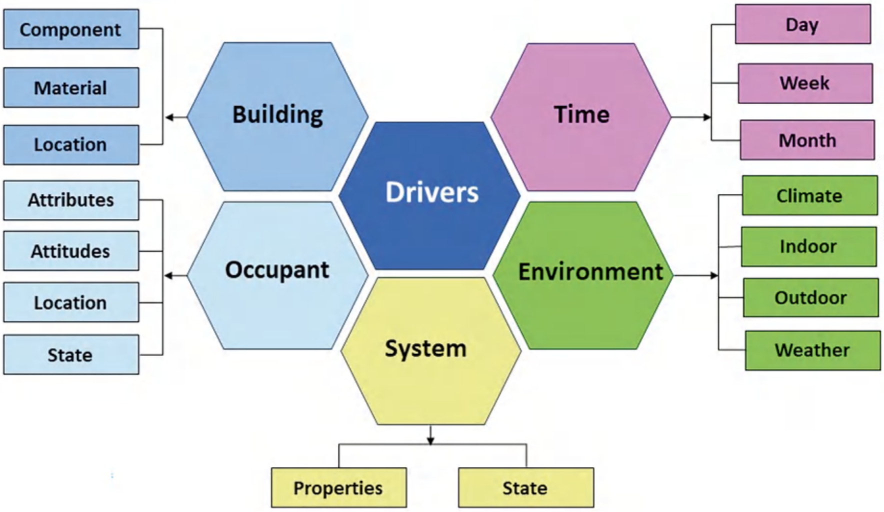


Fig. 8.3 Principal drivers of building energy management system [5]

- Reporting and analytics requirements
- Usability requirements
- Audit requirements

It is best to incorporate the functional and nonfunctional requirements separately. Table 8.4 gives the format in which the nonfunctional requirements should be captured. While designing the system, the functional and nonfunctional requirements can be put into a traceability matrix that can be followed throughout the project.

Once the requirements of BEMS have been captured and recorded completely, one should document the guiding principles that should be considered while defining the architecture and design of the system. It ensures that we incorporate issues related to interoperability, industry standards, social responsibilities, transparency, privacy, etc. [6]. Some of the guiding principles for the BEMS can be as given below. It is not a complete and exhaustive list but includes guidelines that have maximum impact:

1. BEMS should improve energy efficiency in buildings by monitoring, controlling, and optimizing HVAC systems, lighting, and equipment that consume energy.
2. BEMS should significantly reduce utility costs and energy consumption by identifying energy wastage, optimizing equipment schedules, and applying energy-saving strategies.
3. BEMS should implement centralized control and automation for HVAC, lighting, and equipment. It should be capable of implementing real-time energy-saving measures by adjusting the settings of the equipment through integration with the system.
4. BEMS should organize the data according to various purposes, for example:

Table 8.1 Sample table to capture the requirements of BEMS

Req #	Requirement	Description	Priority	Frequency	Use Case Reference	Service Category	Impacted Stakeholders
Requirement Serial Number	Name the Requirement	Description of the Requirement	Priority of the Requirement. Value can be taken from the Table 8.2	How often it will be used or will it be triggered based on a condition	Links to use case documentation	System functionality with which it is concerned	Users who will be impacted by this requirement

Table 8.2 The value of priority of the requirements for BEMS

Value	Rating	Description
1	Critical	This requirement is critical to the success of the project. The project will not be possible without this requirement
2	High	This requirement is a high priority, but the project can be implemented at a bare minimum without this requirement
3	Medium	This requirement is somewhat important, as it provides some value, but the project can proceed without it
4	Low	This is a low-priority requirement, or a "nice-to-have" feature if time and cost allow it
5	Future	This requirement is out of scope for this project and has been included here for a possible future release

- Within facility hierarchy (zones, buildings, floors, rooms)
- Within the organization hierarchy (departments) or
- Within a system (chillers, air handling units, solar thermal zones)

This will enable the analysis of data using advanced algorithms and analytics, which can be used by building managers, energy purchasers, agents, and technicians.

5. BEMS should have an integrated dashboard that can display iterative measurements to operations managers and experts. This will enable experts to improve energy efficiency and detect problems by visualizing building subsystems. A timely scenario can provide meaningful outcomes regardless of influencing factors (climate change, behavior changes, technological advancements).
6. BEMS should be based on open systems and industry standards so it can be integrated with other building systems. This integration will facilitate controlling and monitoring the building centrally. Scalability is also critical, as BEMS should be able to accommodate future expansions, additional sensors, and technological advancement.
7. Sustainability and energy efficiency are being increasingly emphasized by governments and regulatory bodies. BEMS architecture and design should be driven by codes, regulations, and energy efficiency standards. It should maintain compliance with these regulations, standards, and certification programs like LEED (Leadership in Energy and Environmental Design) [7] and BREEAM (Building Research Establishment Environmental Assessment Method) [8].

Once the guidelines are captured for the design of BEMS, one should focus on the drivers of BEMS as described earlier. Building energy performance is affected by occupants' interactive behavior in addition to weather conditions. For example, the occupants' decision to open windows, adjust the thermostat, or turn off the lights can have a significant effect on energy consumption. In this section, we will explain the design of BEMS considering occupants' energy-related behavior in a commercial/office building.

Based on the above discussions, by placing the building occupants in the focus while designing BEMS, we need to consider the following:

Table 8.3 Sample requirements for our use case BEMS

Req #	Requirement	Description	Priority	Frequency	Use case reference	Service category	Impacted stakeholders
Base requirements							
RQ-G-001	Heating control	Control heating/cooling according to room occupancy	1	Whenever room occupancy changes	Occupancy Use Case Link	Heating/cooling	Room occupants
RQ-G-002	Lighting control	Control lighting in the room as per the occupant's need	1	As requested by the occupant	Lighting Use Case Link	Lighting control	Room occupants
Security requirements							
RQ-S-001	Access control	User creation in the repository shall be limited to the system administrator	1	Whenever there is a change in building residents	Access Control Use Case Link	Access control	Operations staff, system administrator
RQ-S-002	Authorization	Interaction with the system should be restricted only to building residents	3	Whenever occupants interact with the system	Authorization Use Case Link	Access control	Building occupants
RQ-S-003	Privacy	The system should not share any individual occupant data with any other identity except when required by law	2	Whenever occupants interact with the system	Privacy Use Case Link	Privacy	Building occupants, operations staff
Reporting and analytics requirements							
RQ-R-001	Monitor energy	Energy consumption should be reported against factors influencing it, such as outside air temperature	2	Continuously for the duration of the operation of the system	Monitoring Use Case Link	Reporting	Building owner, operations manager
RQ-R-002	Identify energy inefficiency	The system should identify the sources/equipment that are consuming a lot of energy	1	Continuously for the duration of the operation of the system	Energy Inefficiency Use Case Link	Analytics	Operations manager

RQ-R-003	Forecast maintenance activities	The system should be able to forecast maintenance activities and end-of-life failures so that action can be taken	1	Continuously for the duration of the operation of the system	Forecast Use Case Link	Analytics	Operations manager, building owner
Usability requirements							
RQ-U-001	Multimode user interaction	The user should be able to interact with the system using a mobile application or voice command	2	Whenever occupants interact with the system	User Interface Use Case Link	User interface/ user experience	Building occupants
RQ-U-002	Identify the scope for improvement	The system shall identify processes that can be modified around operational requirements	4	On a monthly, quarterly, and annual basis	Improvement Use Case Link	User experience, analytics	Building occupants, operations manager, staff
Audit requirements							
RQ-A-001	Maintain an audit trail of anomalies	Keep track of any anomalies and trends in energy consumption	1	Continuously for the duration of the operation of the system	Audit Trail Use Case Link	System logging	Operations manager, audit team
RQ-A-002	Maintain equipment inventory	Maintain a record of all the equipment installed in the system or kept in store as spares	1	Whenever equipment is installed or replaced	Inventory Use Case Link	Inventory management	Operations staff, audit team

Table 8.4 Sample nonfunctional requirements for our use case BEMS

ID	Requirement
NFR-001	The system should be designed to keep the occupants comfortable and productive
NFR-002	The system should minimize environmental impact by operating optimally, removing energy inefficiency
NFR-003	The system should use new and innovative technologies and products
NFR-004	Human health and safety should be one of the top priorities of the system
NFR-005	After the installation of the system, the carbon footprint should reduce

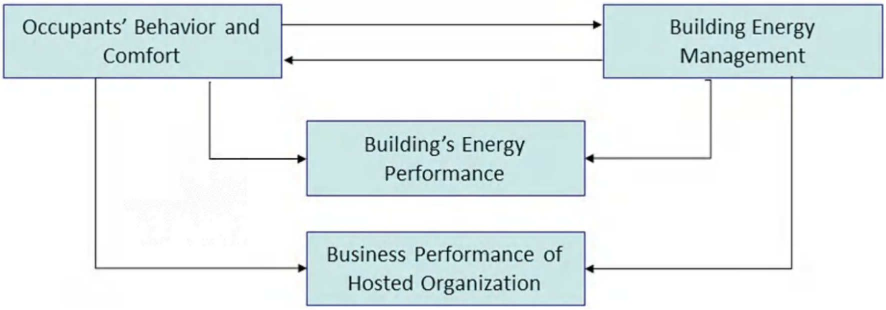

Occupants' Behavior and Comfort
Building Energy Management
Building's Energy Performance
Business Performance of Hosted Organization

Fig. 8.4 Interaction among occupant behavior, energy performance, business productivity, and energy management

- Occupants' behavior can significantly affect the energy efficiency of existing buildings [9].
- Employees content with the work environment significantly affects their productivity [10].
- Buildings ultimately exist to serve their occupants and keep them satisfied [11].

Figure 8.4 represents the interaction among the various factors discussed above on energy consumption. This influence is very important and should be taken care of while designing any BEMS. It can be easily inferred from the diagram that occupants' behavior affects the energy performance of the building, energy management, and business performance. On the other hand, building energy management also affects the behavior of occupants.

8.2.2 *Defining BEMS Architecture*

The architecture framework of BEMS has significantly improved from 1985 until today due to enhancements in technologies in sensors, actuators, data analytics, communication, and command and control systems. These are all critical components of BEMS.

1. *Environmental Quality and Occupancy Sensors*: Environmental quality sensors play a key role by providing data about environmental conditions such as indoor air quality. This helps in addressing the health and safety issues of occupants while supporting the management of energy consumption. BEMS must integrate these along with the metering and HVAC functions in traditional building systems.

Occupancy sensors are already a common way to regulate energy consumption based on how many people are using a space. An intelligent EMS can make real-time adjustments to building systems like HVAC and lighting in response to occupancy data.

BEMS also uses many other wireless sensors that gather data on several conditions, including temperature, humidity, energy usage, etc. It uses actuators to accept electrical input to act on what an analytics software recommends.

2. *Data Analytics*: While analytics within an energy management system has multiple purposes, a key function involves managing energy usage. As smart devices (sensors) gather more data, this data becomes more detailed, allowing analytics software to examine and resolve increasingly complex problems.

By including machine learning (ML), the analytics programs become more powerful to modify operations and increase energy efficiency. They can be used for forecasting energy consumption, detecting and predicting faults, seasonality, and pre-cooling or pre-heating modeling.

Data analytics and ML algorithms can also be used for optimizing the process of receiving and storing power from electricity generated onsite. It can decide when to share the excess electricity with a neighborhood microgrid to ensure optimal utilization of energy.

3. *Active Command and Control*: Active command and control of energy-consuming systems is an important task to meet the business demand for flexible workspaces. The system should allow control of equipment more effectively to help reduce energy consumption. This will enable reduced consumption during low traffic times while ensuring occupants have what they need when they need it.

The conceptual architecture of BEMS with different layers as mentioned above is shown in Fig. 8.5. The diagram also shows the integration of BEMS with a smart grid and a local renewable energy source. This increases energy efficiency by monitoring and controlling energy consumption while reducing energy costs and carbon emissions.

The functions of various layers in the architecture are:

1. *Physical Layer*: It uses sensors and meters for data collection regarding building energy consumption. The data include information on the consumption of electricity, gas, and water, as well as the temperature, humidity, and occupancy of the building.
2. *Automation Layer*: It uses electronic control systems and processors to ensure that HVAC, lighting, and other appliances are operated as efficiently as possible.

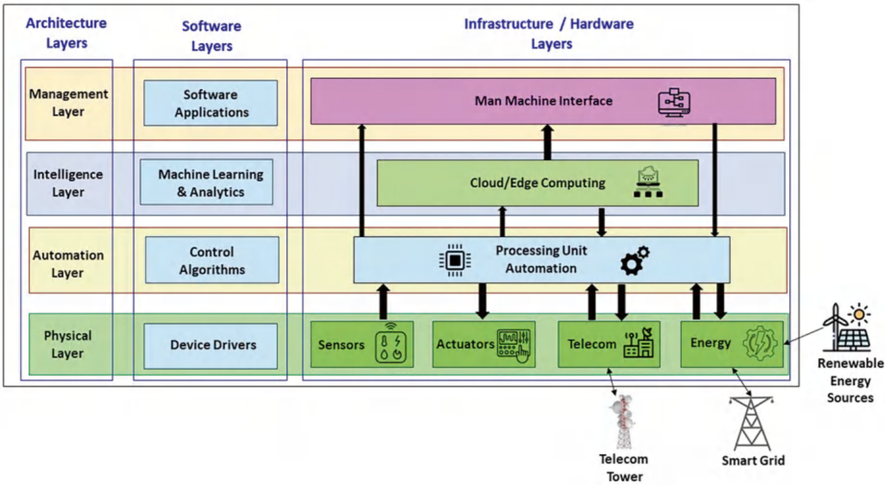


Fig. 8.5 Conceptual architecture of BEMS

The control of renewable energy sources such as batteries and solar panels can also be integrated with it (if available in the system).

3. *Intelligence Layer*: It performs data analysis on data that is collected from the sensors. It uses advanced machine learning algorithms and data analytics techniques to detect patterns in energy consumption, detecting areas with high energy consumption, and tracking energy consumption over time. It also optimizes energy efficiency and reduces waste by analyzing the data and adjusting the building's energy system through the automation layer. For example, adjusting the heating and cooling systems can maintain a comfortable temperature while minimizing energy consumption. This layer is typically implemented in the Cloud but may be available at the Edge if required.
4. *Management Layer*: This layer provides the user interface and reporting functionality to the users (covering both service providers and end users). The system can generate reports on energy usage, cost savings, and environmental impacts. It is possible to use these reports to track progress over time and to make data-driven decisions to further optimize the energy efficiency of the building.

8.2.3 System Design for Efficient Energy Management

While designing BEMS, we need to consider functional and nonfunctional requirements covering all aspects that we should be addressing using the system. To illustrate the design process, we consider occupants' comfort as the main driver, but we need to follow the same process for all the drivers as shown in Fig. 8.3.

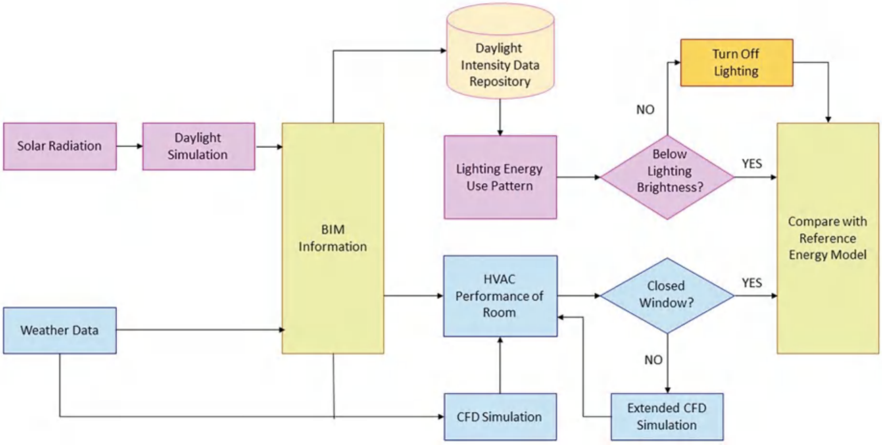


Fig. 8.6 BEMS conceptual workflow

We first need to identify and define the workflows to meet the requirements. The next step should be to define the control strategy to achieve the objectives. As a last step, we need to address communications between the users and the system. Once all these elements are in place, we can define the specifications of the equipment needed to implement the system as designed. We need to take care of the other requirements, like support for open standards and interoperability, to avoid falling into the trap of proprietary products and protocols. We will be following this methodology to design our BEMS, which is occupant-centric.

8.2.3.1 BEMS Workflows

Two major contributors to building energy consumption are HVAC and building lighting. These are impacted by the occupants' actions and weather conditions. The workflow for these factors is captured in Fig. 8.6. As can be seen from the diagram, there is a need to have building information modeling (BIM) information and computational fluid dynamics (CFD) analysis calculations. BIM contains information like location, orientation, and glazing properties, while CFD analysis involves fluid flow and heat conduction. These two can be implemented by the intelligent layer as proposed in the BEMS architecture either in the Cloud or at the Edge as per the requirements.

The interaction between BIM and CFD is shown in Fig. 8.7. It is important to understand and capture this interaction as the action of an occupant to open a window can change the flow of air within the room. It needs to be captured in the model that we build to control BEMS.

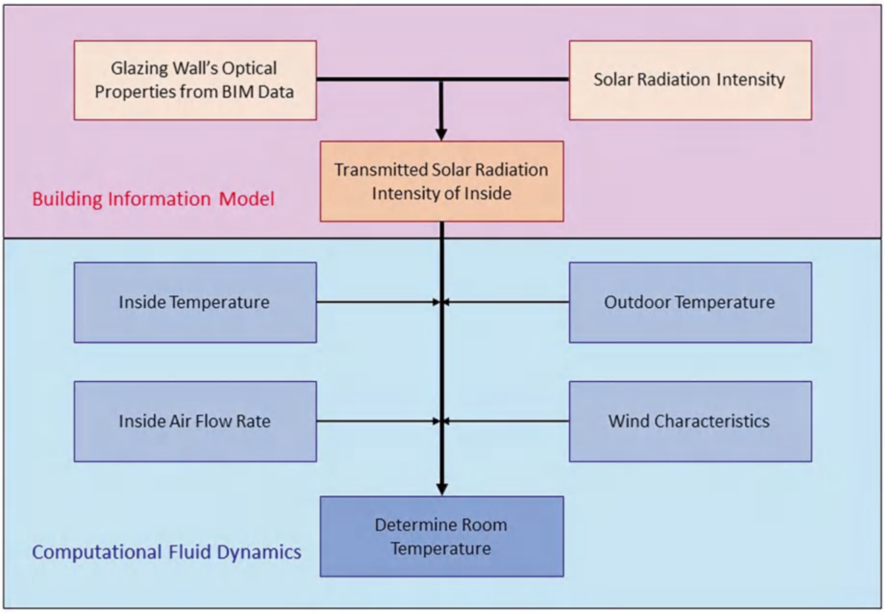

Glazing Wall's Optical Properties from BIM Data
Solar Radiation Intensity
Transmitted Solar Radiation Intensity of Inside
Building Information Model
Inside Temperature
Outdoor Temperature
Inside Air Flow Rate
Wind Characteristics
Determine Room Temperature
Computational Fluid Dynamics

Fig. 8.7 Schematic interaction workflow between BIM and CFD

8.2.3.2 BEMS Management Strategy

The BEMS model also needs to incorporate a management strategy that enables it to interact dynamically with the automation control unit of the BEMS control layer to improve energy efficiency. The management strategies can be classified into four types based on the approach. These are model predictive control, demand-side management, optimization, and fault detection and diagnosis [12]. These are shown in Fig. 8.8 and are explained in the following paragraphs:

1. *Model Predictive Control*: Model predictive control (MPC) can foresee building responses to control requests and can act sufficiently to accomplish the necessary operations. There are three methods for forecasting building energy usage, namely:

   (i) The white box model is a physics-based method that utilizes a straightforward procedure dependent on calculations to explain the energy performance of buildings. These are typically used for temperature control, forecast energy consumption, predictive whole building heat and moisture, and optimal control of HVAC.

   (ii) The black box model is a data-driven method that is dependent on statistical evaluations and artificial intelligence to evaluate and estimate building energy utilization. These are typically used for boilers, predictive control

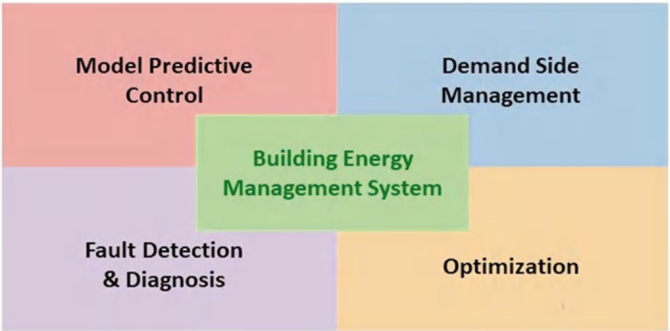

Model Predictive Control
Demand Side Management
Building Energy Management System
Fault Detection & Diagnosis
Optimization

Fig. 8.8 BEMS management strategies

for HVAC systems, peak load, thermal comfort, renewable energy storage, and sustainable power sources.

(iii) The grey box model is based on a hybrid method that is a combination of white box and black box approaches. These are typically used for optimizing the airflow volume and the air supply temperature setpoints, instant balance point temperature, and thermal building modeling.

2. *Demand-Side Management*: Demand-side management (DSM) is an arrangement of actions to enhance BEMS on the user side. It goes from enhancing energy efficiency by utilizing improved resources over intelligent energy rates with motivators for certain utilization arrangements. There are two approaches for DSM, namely demand response (DR) and energy efficiency (EE). The DR approach is very useful for nonresidential buildings for controlling HVAC systems, while the EE approach is more suitable for controlling appliance consumption and HVAC systems.
3. *Optimization*: It is a way to deal with optimization of the system and related issues. There are two approaches, namely atochastic optimization (SO) and robust optimization (RO).

(i) Stochastic optimization requires that the dissemination of information whether genuine or dubious, must be known or assessed. It is generally used for maximizing the comfort index utilizing minimum power consumption, effective policy measures, identifying energy consumption patterns, maximizing the general energy efficiency, load demand prediction of PV-integrated intelligent buildings, and energy savings. This is accomplished through analytics of actuators and information sources. It is useful for both residential and nonresidential buildings. It typically uses particle swarm optimization [13] and neural networks.

(ii) Robust optimization presumes hard constraints. For example, the building temperature cannot be more than 76 degrees. It is generally used for optimal planning of the components of the local energy system, supervising multi-HVAC systems, managing occupants' comfort and energy utilization, and coordinating the cooling system. It is mostly used for nonresidential buildings.

4. *Fault Detection and Diagnosis*: Fault detection and diagnosis (FDD) is a programmed procedure of detecting and separating flaws in BEMS to prevent any harm to the system. There are two techniques, namely data-driven and knowledge-driven. Both approaches are mainly used in nonresidential buildings.
   (i) Data-driven approaches use AI to resolve challenges. It requires adequate training information to build a robust model for fault detection. It is generally used for the detection of faults in the heating system and for recognizing irregular operation patterns.
   (ii) Knowledge-driven approaches depend on specialists to recognize and detect faults more viably and dependably, particularly in cases where analytic data is deficient and unsure. It is typically useful for analytic analysis of an air handling unit (AHU), recognizing potential reasons for inconsistencies for an AHU, distinguishing and assessing chosen faults in a cooling system, and distinguishing undetected flaws.

The designer of a BEMS needs to select a suitable management strategy to meet the business objectives. Typically, the most effective management strategy for BEMS is a combination of the abovementioned strategies. It can be implemented at the intelligence and automation layers of BEMS architecture, as was shown in Fig. 8.5. We recommend that while implementing the management strategy, a modular approach as described in Fig. 8.9 be followed.

The management system may consist of six modules: two core modules, namely learning and simulation, a sensing/data collection module, diagnostics and

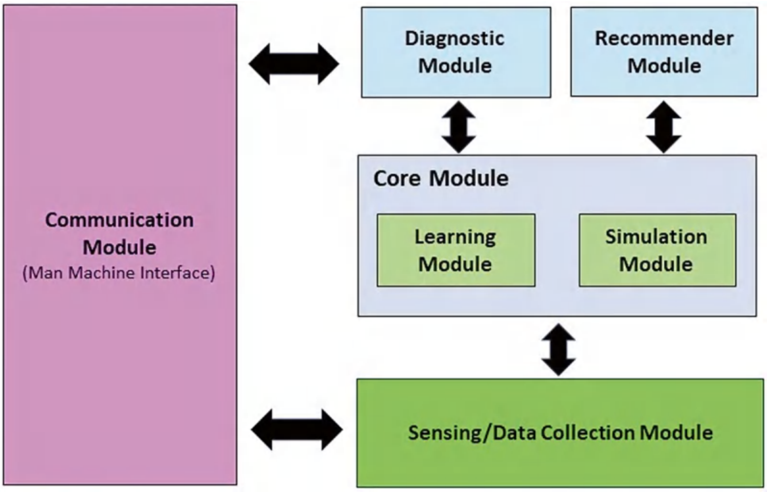


Fig. 8.9 High-level design for BEMS management

recommender modules that rely on the core modules and process their results to either diagnose or recommend [14], and finally, a module that is responsible for making building occupants involved and delivering the actions in a user-friendly and user-cooperative manner, i.e., the communication module. Each of the modules is described in the following paragraphs:

In this case, the BEMS management system can recommend the following types of actions to be taken:

- Lighting
- HVAC
- Maintenance
- Inform/explain

It is good practice to have the last action (inform/explain) implemented to ensure communications between the system, the building manager, and the occupants.

1. *Communication/Feedback Module*: The communication/feedback module is responsible for successful bidirectional communication with occupants. This module ensures that the system runs in coordination with the needs of occupants. It should incorporate various interaction options, like support for mobile phones, tablets, or laptops/desktops. It should have a user-friendly interface that is easy to navigate for the user with little or no training to operate computers. The module should also support voice commands and speech outputs. It should provide all the status and operational reports that may be needed by the building manager, administrator, or BEMS operators.

The module should allow the users to provide feedback to the system. The feedback can be used by the learning module to learn and build a model of how occupants perceive certain configurations. A configuration would consist of the occupant's description, heating/cooling level, outside temperature, time of the day, date in the calendar, etc. The communication/feedback module relies largely on the sensing/data collection module.

2. *Sensing/Data Collection Module*: The sensing/data collection module's task is to sense, collect, filter, and process data from buildings, occupants, and businesses. These would include movements (room entry/exit), window/door opening/closing, lighting, temperatures, etc. It would also signal obvious outliers and abnormal behavior. It generally works closely with the communication/feedback module. It can work independently, only in cases where it needs to automatically sense certain data without explicit communication with occupants. Otherwise, its actions will need to be coordinated by the communication/feedback module. It will also serve to store historical data that has been collected, which would further be utilized and processed by the core modules (learning and simulation modules).
3. *Diagnostics Module*: The diagnostics module is responsible for diagnosing issues and energy gaps when the energy systems do not behave as predicted by the simulation and learning modules. The learning module will support the diagnostics module by learning models from data that would be utilized to determine

the causes of deviations in the system's behavior. Input to the diagnostics module would be deviations and problems in the energy management system, and the output would be the list of possible causes, ordered by their probabilities. Depending on the nature of the cause, the recommender module could be communicated to calculate a list of possible remedial actions.

The diagnostics module can generate system alarms for equipment failure or violation of normal conditions. It can also identify both planned and unplanned maintenance requirements (e.g., systems can record the number of hours that motors have run or identify filters on air supply systems that have become blocked).

4. *Learning Module*: The learning module is responsible for learning from collected data and building models and classifiers that could further be used by the diagnostics module and the recommender module. The learning module would utilize machine learning and data mining algorithms, such as support vector machines or Bayes networks. To consider occupants' comfort, the module will also incorporate occupants' perceptions gathered through the communication/ feedback module and utilize them for the classification of conditions that achieve optimal energy efficiency with minimal occupants' discomfort.
5. *Recommender Module*: The recommender module processes the results obtained by the learning and simulation modules and suggests a set of recommended actions, ordered by priority. Each of the recommended actions would be assigned a value that would assess the level of improvement if implemented. The recommended actions are generated based on a set of predefined goals, which could be assigned different weights based on their relevance for the building management.

The building energy manager can either automate some of the processes and allow the system to proceed with the most favorable action or intervene by manually selecting one of the suggested remedial actions. Thus, the system can be made to operate in either a fully or semiautomated manner.

6. *Simulation Module*: The simulation module runs various what-if scenarios for further optimization of parameters. The simulation module is used either by the recommender module, the diagnostic module, or by the building manager directly to test and evaluate various scenarios. The simulation module gathers data from the data collection module and utilizes this to build prediction models for both optimization and diagnostics.

8.2.3.3 BEMS User Interface

Ensuring good user interfaces with BEMS is essential. A modern BEMS can be accessed in several ways using Web browsers via the Internet, through hand-held tablets and laptops, or palm devices and smart mobile phones. Providing multiple access methods allows building operators to use the BEMS in a way that fits their role and the way they work and encourages them to utilize the system as a building energy optimization tool. A poorly designed user interface discourages operators from using it, resulting in BEMS being ignored for improving energy usage. To optimize internal conditions and make ongoing savings, BEMS need to be regularly maintained. BEMS settings need to be checked at least every month and check that settings meet actual building requirements. The user interface should also support these functions, including any alerts as may be needed.

8.2.3.4 Selection of BEMS Components

As described in BEMS architecture, the system is made up of various components, such as sensors, actuators, controllers, and other automation subsystems. All these components must be able to communicate with each other easily. For instance, within the HVAC system, the thermostat must be able to communicate with the air handler unit and fan coil unit to cool a space properly. While selecting the components for building the BEMS, each piece of equipment must support open protocols like Bluetooth, Wi-Fi, and Zigbee to communicate with other components/devices. They should also be capable of connecting to different types of networks, like Ethernet, and cellular networks (such as 4G and 5G), depending on the communication requirement with the external world. The system should be able to support protocols such as HTTP, MQTT, and AMQP to facilitate transmission from one application/device to another and with the Internet. It is important to ensure that the need for reliability, security, and confidentiality of communications is not sacrificed while providing connectivity using different protocols.

It is equally important to ensure that the operating system and application runtime environment on the components/devices also support open standards. It is desirable to have a general-purpose operating system like Linux with a small footprint for the devices. Similarly, open-source runtime environments like Java, Python, or Node.js are preferred. They allow small application codes (applets) to run on the devices.

Integration capabilities are not only required at the device/component level but also at the application level. BEMS should be built upon a smart building integration platform that enables interoperability between systems. As part of this integration capability, it should command and control connected systems to help improve productivity and efficiency.

Scalability is an important aspect of designing BEMS. It should provide the ability to quickly onboard new assets (sensors, actuators, and other devices) so that they can be easily integrated into the system. There should be a single repository for all assets for building data, analysis, reporting, and control. It will enable the leveraging of machine learning and other features from a single, easily accessible point. We must ensure that extra hardware, especially, sensors is readily available so that they can be replaced when necessary.

8.2.3.4.1 Device Security

A prime concern for any connected device is security. While selecting the devices for BEMS, it is important to choose those devices that provide clear and concise documentation on cyber security measures, protocols, and practices.

It is preferable to select those devices that come with preinstalled applications that can control access to the devices and data. To ensure that the system is not compromised, devices should always send messages using encrypted data and automatically log out after a certain time. Some of the features to look for while selecting the devices are support for user authentication, encrypted traffic, JSON Web tokens, TLS authentication, and/or zero trust.

There are other things that must be done to improve the security of BEMS, such as:

- Keeping a detailed and up-to-date inventory of connected devices. The inventory should have information about device manufacturers, including models, serial numbers, firmware, hardware, and software.
- Maintaining information on configurations and operating systems for each device.
- Determining the risk profiles for each device and how it interacts with other devices within the network.
- Dividing the network into at least two subsections to limit vulnerable areas and reduce damage in the event of an attack.
- Using the most up-to-date firewall protocols and virtual local area networks (VLANs) to keep IT assets separate from smart devices.
- Monitoring and reporting the alerts in real time when managing risks.

8.3 Medical Data Sharing by Hospitals

The importance of data sharing in healthcare has become increasingly important recently, especially after the COVID-19 pandemic. During the pandemic, the ability to share patient information between healthcare providers, hospitals, and other healthcare organizations helped in the development and deployment of effective treatments. It helped to improve patient outcomes, reduce medical errors, and enhance the overall quality of care.

As the industry works toward finding treatments for chronic diseases impacting the world population, it must have access to recent and accurate clinical data. The major problems in sharing clinical data are ensuring data integrity and protecting patient privacy. The systems deployed by the healthcare providers must ensure that these problems are addressed. Any solutions should be able to differentiate between private data and clinical observations of a patient by de-identifying healthcare data [15]. This de-identified data can be used in clinical trials to discover new therapies and improve patient health outcomes. To ensure that any healthcare data is shared securely and efficiently, certain standards must be put in place, as described in the next section.

8.3.1 Best Practices for Medical Data Sharing

Medical data about a patient encompasses information collected by healthcare providers from a variety of sources, including the patient's diagnosis, test results, medications, treatment plans, and family medical history. This data is used to determine an approach to providing care to the patient. If required, it may have to be shared within and across networks of healthcare providers to augment existing therapies to improve patient outcomes. While sharing the data, it may be required to be de-identified to protect the privacy of the patient.

The healthcare provider should follow best practices and government regulations for the entire life cycle of patient data. It covers the process of how data is collected, stored, shared with other stakeholders, and archived.

As a first step, hospitals should improve patients' confidence in the use of their data. It can be achieved by ensuring that the data is accurate and secure. The second step to building the confidence of the patient is to ensure that hospital systems are transparent on how and where their data is shared, as well as the precautions being taken to keep it secured.

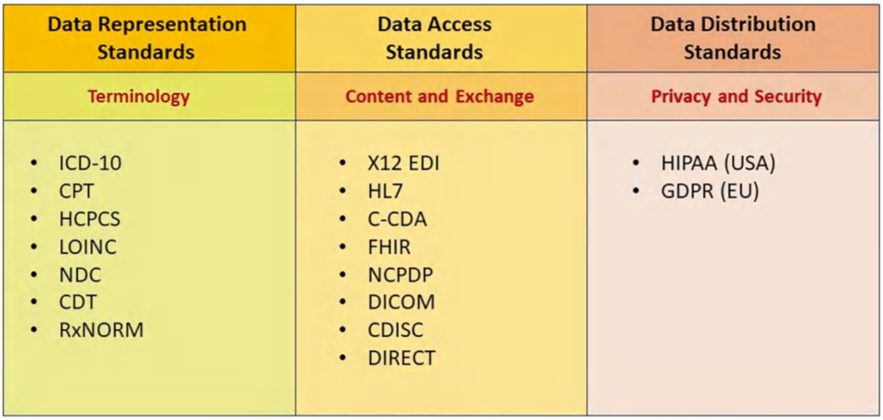
The third and most critical step is to invest in modern data management infrastructure that eliminates data silos among hospital departments, allowing for more easily transmissible and accurate data across networks.

In addition, healthcare providers should prioritize working with the right partners and hiring employees who understand and respect the criticality of the information they manage.

Finally, healthcare providers must have well-defined operating procedures to take care of the entire life cycle of patient data. The standard procedures of healthcare providers should cover how they collect and manage healthcare data, not only to build trust with patients but also to help them understand the true impact of that shared data.

Systems must be developed based on universally acceptable standards to enable all stakeholders to exchange patient information securely and seamlessly [16]. Data standards encompass representation, access, and distribution that define the approach and practices for developing, approving, and instituting compliance. Figure 8.10 summarizes these standards and brings them together for clarity. These requirements are fundamental to establishing control over the data layer for the efficient use and exchange of information.

- *Data Representation Standards*: These standards deal with business terms and definitions, allowed values, formats, logical and physical naming and abbreviation standards, model management standards, etc. These are also referred to as terminology standards [17] because they are used for identifying and exchanging common codes, for example, diagnosis codes, procedures, clinical codes, lab codes, and pharmacy codes. Some of the commonly used codes include ICD-10, CPT, HCPCS, SNOMED-CT, LOINC, NDC, CDT, and RxNORM.
- *Data Access Standards*: These standards deal with common data services, information exchange standards (e.g., XML), standard methods for bulk data



Data Representation Standards	Data Access Standards	Data Distribution Standards
Terminology	Content and Exchange	Privacy and Security
• ICD-10 • CPT • HCPCS • LOINC • NDC • CDT • RxNORM	• X12 EDI • HL7 • C-CDA • FHIR • NCPDP • DICOM • CDISC • DIRECT	• HIPAA (USA) • GDPR (EU)

Fig. 8.10 Healthcare data standards

movement and point-to-point interfaces, and data integration standards (e.g., extract, transform, load standards, etc.). They are also referred to as exchange or transport standards [17], as they typically help in "exchanging" or "transporting" data and define a framework that helps data exchange across systems. Some of the common standards are X12 EDI, HL7 v2.x, HL7 C-CDA, FHIR, NCPDP, DICOM, CDISC, and DIRECT.

- *Data Distribution Standards*: These standards deal with ownership and authority, requesting and approving access, internal and external data provisioning (e.g., portals, Web services, etc.), distribution controls and criteria-based access restrictions, distribution models (e.g., push, pull, publish, and subscribe, etc.), regulatory authority and audit, etc. Since these standards deal with the privacy and security of patient data, they are also referred to as privacy and security standards [17]. These standards help protect sensitive and confidential health information. Healthcare organizations should comply with these regulations to implement robust and secure health systems. Some of the most used standards are Health Insurance Portability and Accountability Act (HIPAA) in the USA and General Data Protection Regulation (GDPR) in the EU.

While designing medical data sharing systems, it is critical to document the guiding principles that should be considered while defining the architecture and design of the systems. It ensures that issues related to patient privacy, data security, interoperability, incorporating industry standards, and government-mandated compliances are addressed. Some of the important guidelines to be considered for improving data sharing among healthcare organizations are:

1. *Only Share Necessary Data*: We should first identify the purpose for which data is being shared. While sharing the data, it should be limited to the essential information required to achieve the desired outcome. For example, if the primary

mission is to deliver a specific care, then only data that will help provide that care should be shared.

2. *Limit Data Access to Those Who Need It*: There should be a very clear data governance policy. As part of the policy, it should be mentioned that data access is limited only to the individuals or departments that need access to that data. This helps in reducing the likelihood of data leakage by helping in tracking and monitoring access. This enables one to identify the cause in case something goes wrong. Reviewing and updating data access should be part of the data governance review plan. The hardware and software deployed should support the data governance policies.
3. *Invest in IT Infrastructure*: The software used for healthcare applications must be updated regularly. Outdated software working on obsolete hardware presents a significant risk to the integrity and security of healthcare data. This is especially true when sharing data over a network. Investment in new hardware and software, or managed IT services, is essential to update and maintain IT infrastructure. This ensures that IT infrastructure is updated and free of any weaknesses that can be exploited to launch cyberattacks. If public Cloud infrastructure is used, special care must be taken to ensure healthcare data security [15].
4. *Adopt Strong Data-Sharing Security Measures*: Once the data is shared with an organization, the organization sharing the data loses control over the data. It is important to share only the necessary data with proper access control and password protection. The data is most vulnerable when in transit, as the methods used to secure data in transit come with limitations. At the very least, HTTPS and TLS1.3 protocols should be used. Standard data-sharing security in healthcare includes tokenization and encryption.

When developing systems for healthcare applications that involve the sharing of data, we must capture the requirements, identifying the security and compliance needs. Table 8.5 provides a template to capture the system requirements. It is similar to Table 8.3 except that here we have captured the applicable standards (concerning privacy, security, distribution, and interoperability). The requirements should be categorized as:

- Base requirements
- Security and privacy requirements
- Reporting requirements
- Usability requirements
- Audit requirements

Base requirements cover the specific functionalities and features that software must provide to satisfy user needs. It typically covers requirements related to business rules, transaction corrections, adjustments, external interfaces, and administrative functions.

Security and privacy requirements cover user validation and authentication, access control, security measures to protect against data theft, personal privacy

Table 8.5 Template to capture requirements for medical data sharing applications

Req #	Requirement	Description	Priority	Compliance / Standards	Use Case Reference	Impacted Stakeholders
Requirement Serial Number	Name the Requirement	Description of the Requirement	Priority of the Requirement. Value can be taken from the Table 8.2	The applicable standard or government compliances to be met	Links to use case documentation	Users who will be impacted by this requirement
Base Requirements						
Security and Privacy Requirements						
Reporting Requirements						
Usability Requirements						
Audit Requirements						

issues related to data owners, and compliance with legal and government regulations.

Reporting requirements cover operational, administrative, management information systems (MIS), and reporting that may be mandated by government regulations.

Usability requirements cover user experience and how easily users can interact with the software systems. They include ease of use, intuitiveness, responsiveness, and accessibility.

Audit requirements cover transaction logging to cover data usage and sharing, historical data storage, archival, software license compliances, etc. These are important to support software audits by external parties or government agencies.

It is best to incorporate the functional and nonfunctional requirements separately. Nonfunctional requirements cover issues related to scalability, capacity, availability, reliability, recoverability, maintainability, serviceability, manageability, environment, interoperability, performance, etc.

To capture nonfunctional requirements, the format shown in Table 8.4 can be used. While designing the systems, the functional and nonfunctional requirements can be put into a traceability matrix that can be followed throughout the project.

Once the functional and nonfunctional requirements have been captured, there is a need to document the guiding principles. These should be considered while defining the architecture and design of the system. Guiding principles should address issues related to interoperability, industry standards, social responsibilities, transparency, privacy, etc.

The applications providing medical data-sharing functionalities shall follow the guiding principles as given below at the minimum.

1. The application should follow stringent data privacy and security guidelines. These should cover user authentication and allow access to those who require it. A secure, monitored environment should also be used to store data to prevent unauthorized access or disclosure.
2. Sensitive patient data should be safeguarded against unauthorized access by encrypting data in transit and at rest.
3. Use secure application programming interfaces (APIs) that adhere to industry standards, especially for encryption, etc. Use a protocol like Fast Healthcare Interoperability Resources (FHIR) to develop software that communicates with other systems for exchanging data.
4. Need to create an audit trail that tracks the access and use of personal healthcare data of an individual. It should record who accessed what data and when. This audit trail can help identify security breaches or violations.
5. It is important to conduct thorough testing and validation to ensure the software is compliant with regulations like HIPAA and GDPR. The testing should check for security flaws and validate compliance with industry standards.
6. The application should follow industry standards and others that may be mandated by government regulations for data storage and exchange and ensuring

privacy and security. It will ensure interoperability and systematic data exchange among different healthcare providers.

7. Compliance with healthcare regulations like HIPAA and HITECH to ensure patient privacy and data security.
8. The system should be designed to avoid noncompliance with regulations. Noncompliance can result in various repercussions, including legal liabilities, data breaches, a loss of patient confidence, and reputational harm. In addition, it may also result in financial penalties and the loss of business.

8.3.2 Defining Data-Sharing Architecture

The healthcare industry has been adopting new technologies such as the Internet of Things (IoT), artificial intelligence (AI), and big data to provide better healthcare outcomes. However, the amount of data generated from healthcare devices and applications brings a lot of challenges in medical data sharing. This gets exacerbated due to different healthcare standards, multiple protocols, system diversity, and incompatibility of data formats.

To overcome these challenges, multiple technical architectures have been proposed to meet the medical data-sharing requirements among healthcare providers through healthcare information exchanges (HIEs). These can be categorized broadly into three architectural types: centralized, federated, and hybrid [18]. They fall along a continuum from fully centralized on one end to fully decentralized on the other, with several hybrid permutations in between. There are distinctions within each model and variations in how a model may be implemented.

1. *Centralized Architecture*: In this model, all data that needs to be shared is normalized in a common format and terminology and is housed together in a central data repository. The stored data can be accessed and used by authorized users per defined policies and procedures. Figure 8.11 shows a centralized medical data-sharing architecture. More than one repository may exist for different kinds of data; for example, digitized radiographic images might be housed in a separate repository given their large size and specialized usage. The centralized approach may offer the best technical performance when measured by patient data availability and response time to user queries. It costs the most to set up and maintain because it requires a large upfront investment in technology in the form of large servers, which need to be monitored and stored in a secure location. This model also requires all participants to trust the central authority that stores and manages the data.
2. *Federated Architecture*: In this model, each participant organization maintains separate control of its data, typically in an Edge server at its location. The patient-specific data is shared with other users based on their requests. In a strictly decentralized model, every request for patient data must be made to every par-

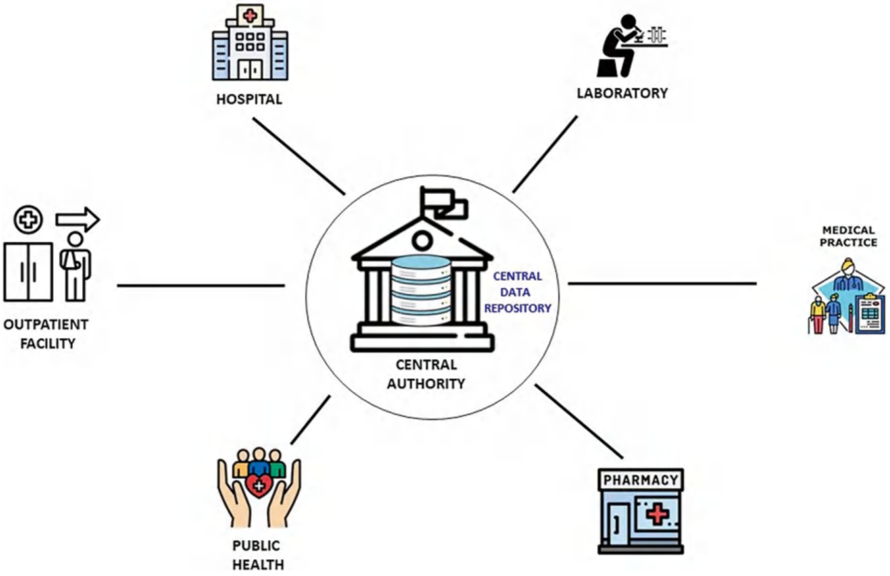


Fig. 8.11 Centralized medical data sharing architecture

ticipating data owner. This effectively limits it to relatively low-volume applications. The federated architecture is shown in Fig. 8.12.

To overcome this limitation, a centralized patient registry is maintained by one of the participants. To retrieve patient data, the requester sends query messages to the patient registry. The patient registry contains a virtual roadmap of where patient health records are located. It transmits the record's physical location back to the requester. The requester then requests the patient information from the data source where it is located. The data can be sent to the requester using any of the secure methods, like secure email, secure Web services, or through a VPN connection.

The federated model is considered less interoperable than the Centralized model because it does not allow a simple exchange of information between systems. It offers security but may lead to unacceptable performance limitations.

3. *Hybrid Federated Architecture*: This model builds on the decentralized model by adding a record locator service (RLS). RLS tracks where patients have received care and from where their source data can be requested. The hybrid-federated architecture is shown in Figure 8.13. Two forms of the hybrid-federated models are very common. In the first model, the participating organization manages data (copies of the original) in separate Edge servers at a central location, but without a shared central repository. This model is useful for clinical applications, in which healthcare providers access data for one individual patient at a time.

In the second model, the functionality of a centralized model is achieved for analytic purposes, either by layering a central repository of normalized shared data

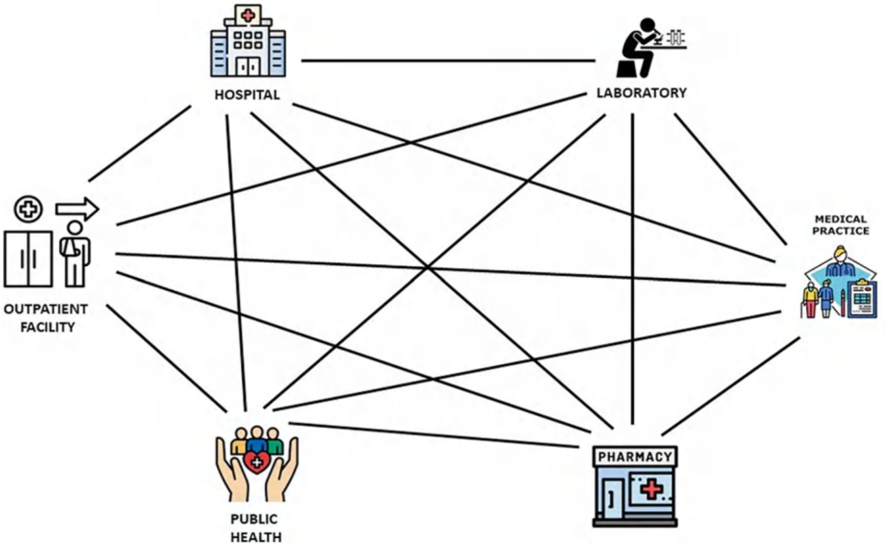


Fig. 8.12 Federated medical data sharing architecture

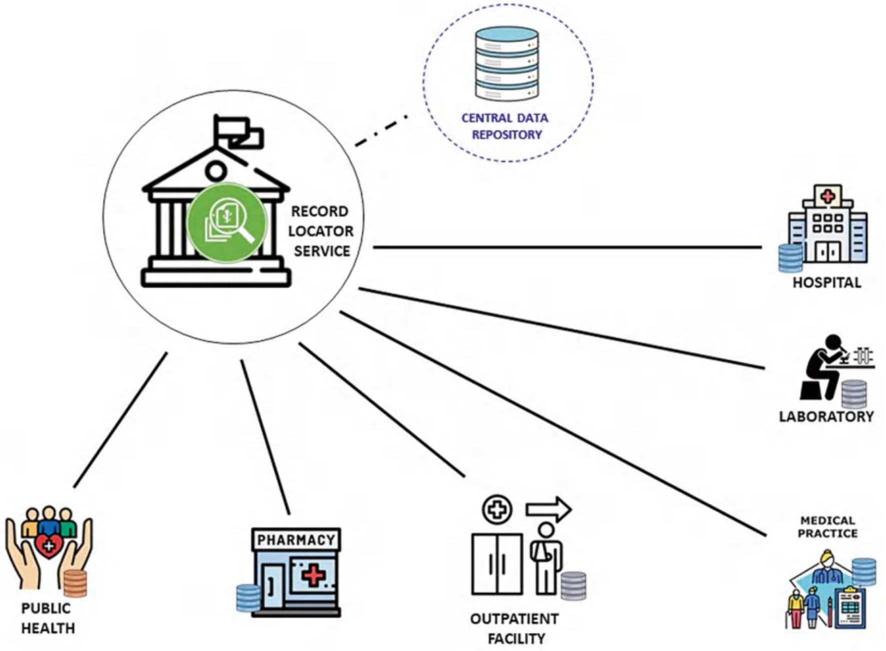


Fig. 8.13 Hybrid federated medical data sharing architecture

on top of the hybrid-federated architecture or by normalizing data in one computer where the data are partitioned by source. This model thus facilitates data use about multiple individuals by providers and other users. Another variation of hybrid-federated architecture is possible where data is partitioned and only de-identified data is shared centrally, whereas Personal Health Information (PHI) stays local.

Cloud, Blockchain, and AI technologies have been extensively used for developing Healthcare Information Exchanges (HIEs). This is irrespective of the architecture that is used for building the HIE. Each of these technologies has advantages and limitations while implementing medical data-sharing applications.

1. *Cloud Technology*: Records stored in the Cloud can be easily retrieved anywhere and anytime using the Internet. It is noted that their users do not always trust third-party Cloud providers. There is a probability of someone breaking into Cloud servers that contain sensitive information, such as patient medical records. One practical method to overcome this limitation is to encode sensitive information before storing it in the Cloud. This technique is called client-side encryption. The advantage of client-side encryption is that data is encrypted in the user's system. The encrypted data is then transferred to a Cloud server. Data encryption ensures a low probability that attackers can steal the data when sent to the Cloud server. The medical record is accessible to customers with a decryption key while remaining inaccessible to everyone else. However, data needs to be retrieved and decrypted before usage on the client side. If the usage is on the server side, then keys will need to be shared and Cloud operators will need to be trusted.

The main issues with the Cloud are data security, availability, integrity, information confidentiality, and network security. API, data encryption, and authentication are some security measures available to reduce security concerns for Cloud infrastructure. Cloud providers must ensure the confidentiality, integrity, and availability of data. Confidentiality helps to prevent intentional or unintentional access. Integrity helps reduce unauthorized modifications to the data by unauthorized or authorized users or processes. Availability helps users access medical records stored in the Cloud from anywhere using any authorized device connected to the Internet.

2. *Blockchain Technology*: Blockchain technology can be used for storing and recording an individual's electronic health record (EHR). A typical EHR is a compilation of health-related data that contains personal details (e.g., name, age, gender, weight, and billing information) and medical history, medications, and health problems (such as illnesses). Blockchain can play an important role in remote health, especially as it can certify trusted devices [19]. The decentralized feature of blockchain offers users anonymity and allows them to always keep their information safe. This ensures audit transparency and accountability, preserving medical data confidentiality and privacy, both at rest and in transition, thereby increasing trust in the data.

A typical solution based on blockchain technology may consist of creating and using structured contracts for data access, standardized audits, and cryptographic algorithms to maintain data security and integrity. The majority of EHR data remain unmodified once they are posted to the system. As a result, strongly protected EHRs saved using blockchain technology can be accessed with higher reliability by many collaborating medical institutions and individuals (such as doctors, hospitals, labs, and insurance companies).

Blockchain technology is useful for different kinds of healthcare applications. Some of the uses of the technology cover taking patient history data, storing it securely on a blockchain network, detecting fraudulent claims, or even tracking patient outcomes.

3. *Artificial Intelligence*: Artificial intelligence (AI) and machine learning (ML) have helped automate physician tasks, resulting in enhancements of clinical capabilities and access to care. Access to large, well-designed, well-labeled, diverse, and multi-institutional datasets is critical for model development and model deployment. The diverse datasets also mitigate racial and socioeconomic biases.

Blockchain technology can be used for storing and recording model parameters, training data, inputs, and outputs, thereby increasing audit transparency and accountability. By combining AI and blockchain technology, one can establish a safe connection to access external AI models and data. This allows multiple institutions to collaboratively train an ML model (i.e., federated learning), leading to enhanced efficacy.

The most popular applications of AI are clinical trial automation, fraud detection, and automated pharma dispensing. It has also been used for risk adjustment, predictive analytics, and automated monitoring.

Despite the broad adoption of AI, additional layers of security must be implemented to guarantee the confidentiality of sensitive personal information,
such as prescriptions for medications [20].

The Office of the National Coordinator for Health Information Technology (ONC) in the USA is working on standardizing a framework for the exchange of health information. It released Version 2.0 of the Common Agreement on April 22, 2024. Common Agreement Version 2.0 updates Common Agreement Version 1.1, published in November 2023, and includes enhancements and updates to require support for Health Level Seven (HL7) Fast Healthcare Interoperability Resources (FHIR)-based transactions. The Common Agreement sets forth the requirements each participant and sub-participant must agree to and comply with to participate in TEFCA. It incorporates all applicable standard operating procedures (SOPs) and the Qualified Health Information Network Technical Framework (QTF) [21].

The Trusted Exchange Framework and Common Agreement (TEFCA) has three goals:

(i) To establish a universal governance, policy, and technical floor for nationwide interoperability

(ii) To simplify connectivity for organizations to securely exchange information to improve patient care, enhance the welfare of populations, and generate health care value

(iii) To enable individuals to gather their health care information

8.3.3 System Design for Secure Data Sharing

While designing applications for secure data sharing, we need to consider both functional and nonfunctional requirements. In addition, we need to take care of the other requirements, such as support for open standards and interoperability, to avoid falling into the trap of proprietary products and protocols.

The proposed methodology for the development of applications for secure exchange of medical data consists of seven steps. The process is very similar to any other software development except that there is a need to consider compliance with the existing standards and follow protocols for secure data transfer across the service providers.

Step 1: Define Target User

There is a need to identify the target market and user needs. It is important because each geography has a different compliance requirement. For example, in the USA, the software must be compliant with HIPAA, whereas in the case of the EU, it needs to comply with GDPR, etc. Not complying with these regulations can result in significant penalties for the developer and users.

Step 2: Collect Requirements

Before developing the software, we need to consider both functional and nonfunctional requirements. In addition, we need to take care of the other requirements, like support for open standards and interoperability, to avoid falling into the trap of proprietary products and protocols. FHIR and HL7 [17] are two of the important standards and regulations; they establish how health data should be exchanged and shared.

Adopting HL7 and FHIR standards results in more efficient, accurate, and secure data sharing. This benefits the patients with improved care coordination and access to their records. Applications developed using these protocols are interoperable, providing greater productivity, reduced costs, and better outcomes overall.

Step 3: Design a Prototype

A prototype is an interactive working model of the application. The prototype should include a wireframe using a computer interface that allows users to interact with the product. Prototyping is important for the following reasons:

- It allows to test the user experience before fully creating the application.
- Before investing a significant number of resources in development, it is possible to collect feedback and make sure the final product has all the right user inputs.

While designing the prototype, we should also finalize the features that should be included and the technology (Cloud computing, Blockchain, AI/ML, Internet of Things) that will be used to develop the application.

Step 4: Develop the Final Prototype

Once the feedback from the users has been received using the initial prototype, the final prototype should be developed incorporating the feedback received from the users. This should be the basis for the development of the full application. It should be designed to keep up-to-date with changes in the underlying technologies.

Step 5: Write Code for the Application

The application should follow strict data privacy and security protocols. It should be able to authenticate users and only permit access to those who require it. A secure, monitored environment should also be used to store data to prevent unauthorized access or disclosure.

While developing the application, we can use secure application programming interfaces (APIs) and encryption. Sensitive patient data should be safeguarded against unauthorized access by encrypting data in transit and at rest. It should adhere to industry standards, such as Fast Healthcare Interoperability Resources (FHIR), to communicate with HIE networks.

It is essential to create an audit trail that tracks the access and use of EMR. This records who accessed what data and when, which can help identify security breaches or violations.

Step 6: Test and Validate for Compliance

It is crucial to conduct thorough testing and validation to ensure that software meets the HIE compliance criteria. The software testing should also cover security flaws and compliance with industry standards such as FHIR.

In addition, the software should also be tested and reviewed by users before being released. Any changes suggested by users should also be incorporated into the application.

Step 7: Deliver the Software

The last step of the process is making sure that the application is delivered in the best way possible. The application should be delivered with all needed updates, including documentation. It may need to be tested again after deployment in the target environment.

While developing any healthcare application, HIPAA compliance should be a priority. It can build the trust of patients and providers in the application. However, complying with HIPAA and HITECH regulations can be challenging, especially for smaller healthcare organizations and startups with limited resources. One should be aware that failing to adhere to HIE standards and regulations can result in various repercussions. These repercussions can be legal liabilities, data breaches, a loss of patient confidence, and reputational harm. Financial penalties and the loss of business possibilities are also further consequences of noncompliance.

8.4 Solar Energy Power Plant Management System

Solar power plants represent a key tool for developing a new long-term sustainable energy generation model that is completely eco-friendly. The Earth receives solar energy in the form of light and heat. Solar power plants convert these energy forms from the sun into electricity. Depending on how the solar plant harnesses sun energy, there are two main types of solar plants: solar thermal power plants and solar photovoltaic plants. The photovoltaic technology will directly convert the sunlight into electricity, while the solar thermal technology will capture the heat of the sun.

1. *Solar Thermal Power Plants*: Solar thermal plants use a conventional thermodynamic cycle to convert solar energy into electricity, unlike thermal power plants, which use fossil fuels. It concentrates the solar radiation on a small area by placing mirrors or lenses over a large area. Due to this, a huge amount of heat is generated at the focused area, which is used to heat a fluid until it is converted into steam. The steam is then fed to a turbine, where the thermal energy is converted to mechanical energy. This mechanical energy is used to run the alternator, which generates electricity. Once the thermodynamic cycle has been completed, the steam is returned to a condenser, where it recovers its liquid state, and the process is repeated.

This method is difficult and is not efficient in producing electrical power on a large scale. Figure 8.14 gives the schematic to show how a solar thermal power plant works.

There are two main types of solar thermal power plants:

(i) *Central Tower Solar Thermal Power Plant*: It consists of a tower of large mirrors called heliostats. These are capable of changing direction to capture the maximum solar radiation and concentrate the mirrors on a specific point. The heat so generated is then transmitted to a thermally conductive fluid,

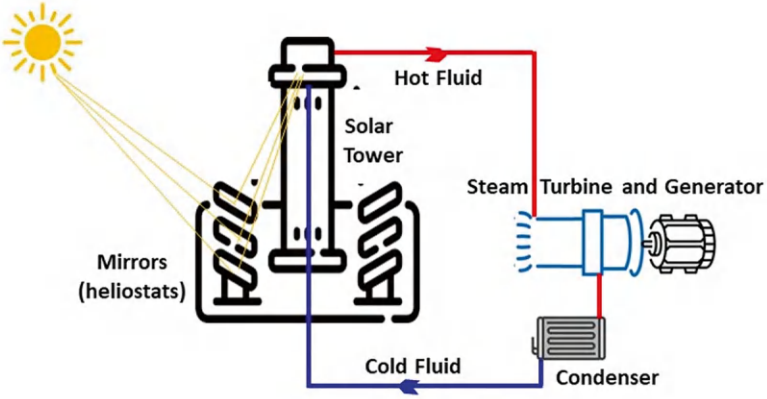


Fig. 8.14 Schematic diagram of a solar thermal power plant

which is converted to steam due to the rise in temperature. This starts a thermodynamic cycle to generate the electricity.

(ii) *Collector Solar Thermal Power Plant*: It uses high-temperature collectors to capture sunrays. The collectors are concave mirrors that are mounted on a structure that allows their position to be modified to increase the intensity of the solar radiation, reaching temperatures greater than 250 °C. It also then uses the conventional thermodynamic cycle to generate electricity.

2. *Solar Photovoltaic Power Plant*: A solar photovoltaic (PV) power plant is based on light energy from sun rays. It works on the principle of the photovoltaic effect, which produces direct current electricity. It uses panels consisting of photovoltaic solar cells made of silicon (monocrystalline or polycrystalline solar panels) or other materials with photovoltaic properties (amorphous solar panels). The solar panels are connected in parallel and are connected to a current inverter, where the direct current coming from the photovoltaic cells is transformed into alternating energy. The electricity is then directed to a transformer that changes the voltage so that it can be transported through the electrical grid lines to the consumption centers.

There are three types of photovoltaic systems according to their implementation.

(i) *PV Direct Systems*: These systems supply the load only when the Sun is shining. These systems do not have batteries and hence provide no storage for the power generated. An inverter may or may not be used depending on the type of load. If the load is AC, then an inverter is provided.

(ii) *Off-Grid Systems or Standalone Systems*: The off-grid system is an independent power plant. It is not connected to the grid. This type of system is used at locations where power from the grid is not available or not reliable, such as forests and hilly areas. It consists of solar panel arrays, storage batteries, and inverter circuits. Generally, this type of system is not used to generate electrical power in bulk amounts. This type of plant is used to sustain small loads.

The standalone system can be categorized as below, based on the configuration and the components used in the system.

- Direct-coupled standalone system
- Standalone system with battery storage
- Standalone system with batteries and charge controller
- Standalone system with AC and DC loads
- Hybrid standalone system

(iii) *Grid-Connected Systems*: These solar power systems are connected to a grid and are generally used for generating bulk power. They typically use a large number of solar panels. As they are connected to the grid, the output frequency and voltage must be matched with the grid's frequency and voltage for proper functioning. The energy generated from the plant is transmitted to the load using the grid.

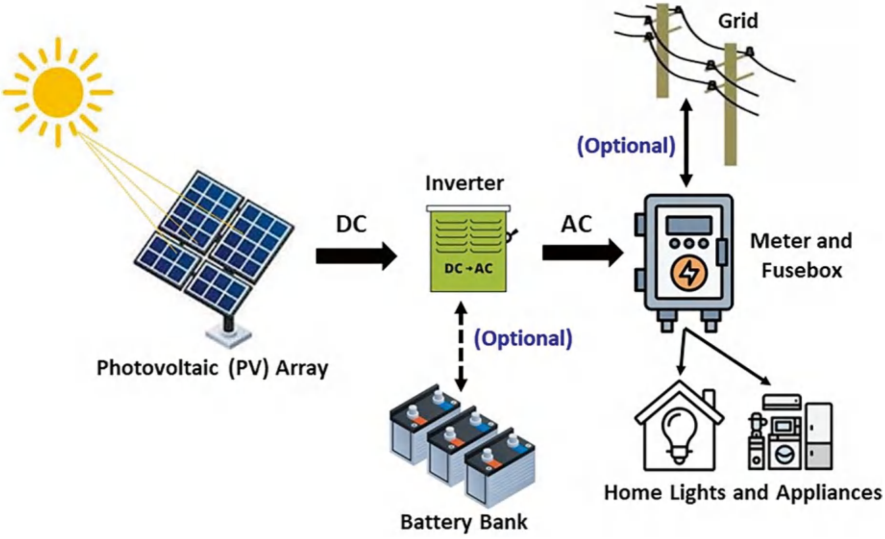


Fig. 8.15 Schematic diagram of a solar photovoltaic power plant

Figure 8.15 gives the generic schematic to show how a solar photovoltaic power plant works. The components that are optional based on the type of PV solar plant it is are marked in the diagram.

8.4.1 Best Practices for Solar Power Plant Management

Managing a solar power plant is a difficult task as it consists of a complex system of components that must work together seamlessly to ensure optimal performance. For our use case, we will consider a management system for a PV solar power plant connected to the grid. The components of a typical grid-connected solar power plant are shown in Fig. 8.16. They are a PV array, inverter, battery, transformer, and monitoring system (typically located remotely).

The array of solar panels is a key component of the system that needs to be managed. The solar panel is responsible for capturing sunlight and converting it into electricity. The efficiency of the panel is critical to the overall performance of the system. It is important to ensure that it is properly maintained and cleaned to maximize its output.

The inverter is responsible for converting the direct current (DC) energy generated by the solar panels into alternating current (AC) energy that can be used to power appliances and other electrical devices. It is a critical component for the operation of the power plant. If it fails, then the whole system will come down.

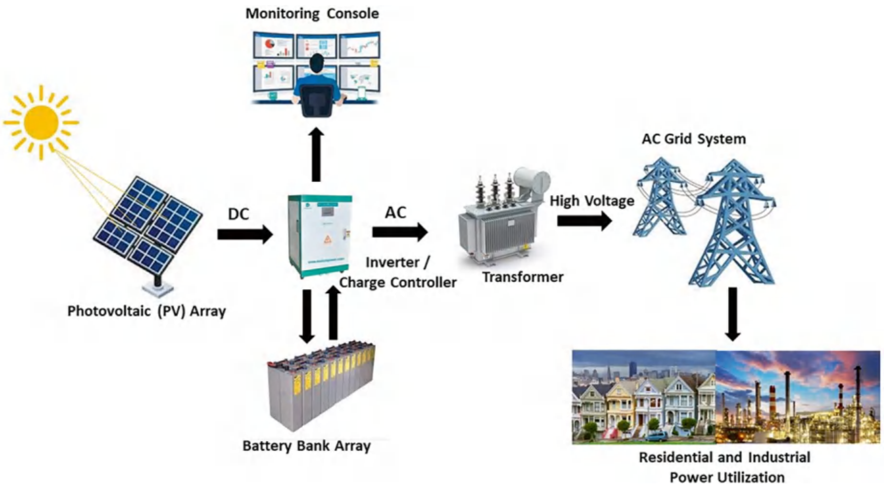


Fig. 8.16 Components of PV grid-connected solar power plant

Battery storage allows excess energy generated by the solar panels to be stored and used later. This is particularly useful during times when there is low sunlight or during power outages. Note that battery life is affected by the operating conditions and number of charging cycles. The charging capacity should be monitored, and batteries may need to be replaced over time.

The transformer is used to connect the power plant to the grid so that electricity generated by the plant can be transmitted to the end consumer of the energy.

Monitoring and control systems provide real-time data on the performance of the solar panels, inverter, and battery storage, allowing for proactive maintenance and troubleshooting. This helps to ensure that the system operates at optimal levels, maximizing energy output and minimizing downtime.

The goal of the PV solar power plant management system is to reduce the cost and improve the effectiveness of operations and maintenance (O&M) and energy storage systems. In addition, the system should also include asset management, monitoring, operations, preventive maintenance, corrective or condition-based maintenance (repair), and end-of-performance period (disposition).

For the management system to operate effectively, it should be able to collect data from the various components of the power plant. It should also be able to guide the field staff to carry out regular checks of the PV panels and associated components.

Some of the requirements of the system are:

(i) *PV Panel Checking/Monitoring*: Solar panels are exposed to open air, making them susceptible to airborne particles, fallen leaves, debris, bird dropping, snow, and buildups of ice. These accumulations make cleaning solar panels mandatory; they can reduce energy production by blocking the reception of sun rays.

(ii) *Inverter Monitoring*: The inverter also needs to be monitored regularly as it is a critical component of the plant, and its failure can lead to total shutdown of the plant.

(iii) *Battery Storage Monitoring*: The performance of the battery bank should be monitored constantly to ensure it is performing as expected. If there is any deviation, the system should be able to identify the fault and replace the faulty/failing battery.

(iv) *Transformer Monitoring*: The parameters of the transformer, such as operating temperature, oil temperature, winding temperature, and oil level, should be monitored to ensure that it is working properly.

Table 8.6 provides a template to capture the system requirements. The requirements should be categorized as follows:

- Base requirements
- Security requirements
- Reporting and analytics requirements
- Usability requirements
- Audit requirements

The nonfunctional requirements should be captured separately using the format given in Table 8.4. Using functional and nonfunctional requirements, a traceability matrix can be prepared. The matrix can be used throughout the project to ensure that development is as per the requirements.

There is a need to document the guiding principles for the design of the system. They can be used for defining the architecture and design of the system. Guiding principles should address issues related to interoperability, industry standards, security, social responsibilities, etc.

Some of the guiding principles for the solar power plant management system (SPPMS) are given below. It is not a complete and exhaustive list but includes guidelines that have maximum impact:

1. SPPMS should improve the efficiency of energy generation at the solar power plant.
2. SPPMS should incorporate the best practices for data presentation, quality of monitoring equipment, and transparency of measurement protocols and procedures.
3. SPPMS should have good reporting capability to obtain value from monitoring data. This will enable the analysis of data using advanced algorithms and analytics that can be used by plant operators.
4. The system should use the instrumentation, sensors, and actuators based on the performance measurement model used, the required accuracy, and other considerations to improve performance.
5. SPPMS should have an integrated dashboard that can display iterative measurements to operations managers and experts. This will enable experts to improve energy efficiency and detect problems by visualizing subsystems.

Table 8.6 Template to capture requirements for solar power plant management application

Req #	Requirement	Description	Priority	Service Category	Use Case Reference	Impacted Stakeholders
Requirement Serial Number	Name the Requirement	Description of the Requirement	Priority of the Requirement. Value can be taken from the Table 8.2	System functionality with which it is concerned	Links to use case documentation	Users who will be impacted by this requirement
Base Requirements						
Security Requirements						
Reporting and Analytics Requirements						
Usability Requirements						
Audit Requirements						

6. SPPMS should be based on open systems and industry standards for information and data communication. Scalability is also critical, and the system should be able to accommodate future expansions, additional sensors, and technological advancement.

8.4.2 Defining the Solar Power Plant Management System Architecture

As mentioned earlier, the objective of a PV solar power plant management system is to reduce the cost of energy generation and maintenance. The system should also be able to improve asset management, monitoring, and preventive maintenance to reduce system downtime.

For the management system to operate effectively, it should be able to collect data from the various components of the power plant, analyze it, and initiate actions based on the data analysis.

To meet these objectives, the architecture of the system should be modular and flexible to accommodate new technology and protocols. Figure 8.17 shows the conceptual architecture meeting these requirements. It consists of five layers each having a definitive function to perform. The functions of various layers in the architecture are:

1. *Physical Layer*: It uses sensors and meters for data collection regarding the atmospheric condition, PV solar panel performance, electricity generation, inverters, transformers, and electricity being transferred to the grid.

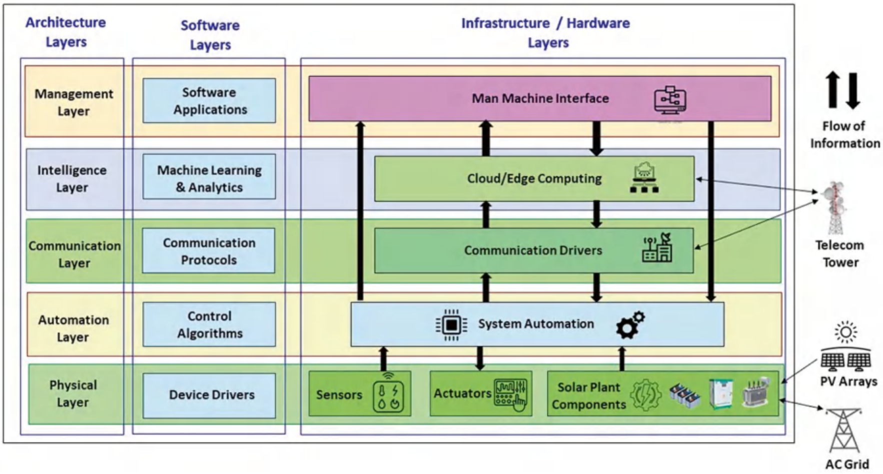


Fig. 8.17 Conceptual architecture of solar power plant management system

2. *Automation Layer*: It uses electronic control systems and processors to ensure that solar plant equipment is operating at optimal performance. It gives suitable commands to actuators to take necessary action for optimal operations.
3. *Communication Layer*: Solar power plants are generally located at remote locations and need to be connected to the central command and control center continuously. The communications ensure that there is always a fast, secure, and reliable connection maintained between the plant and the control center. It also handles the communication among various components of the solar power plant. The layer should support open standards and protocols for interoperability and ensure that all external communication from the system is encrypted.
4. *Intelligence Layer*: It performs data analysis on data collected from the sensors. It uses advanced machine learning algorithms and predictive analysis for plant maintenance. It helps to identify anomalies to prevent catastrophic failures or other costly issues before they occur. It also forecasts the remaining useful life of assets.

In addition to preventing asset failure, these AI-driven insights help operators to optimize energy generation, compare asset performance, and diagnose anomalies. It helps the power plants to stay in regulatory compliance and help meet contractual obligations by enabling better asset and process outcomes. This layer is typically implemented in the Cloud but may be available at the Edge if required.

5. *Management Layer*: This layer provides the user interface and reporting functionality to the users (covering both service providers and end users). It allows the operators at the control center to visualize data to increase operational awareness and collaboration across functional departments. The system generates reports on energy generated, cost savings, and environmental impacts. It is possible to use these reports to track progress over time and to make data-driven decisions to further optimize the operations of the solar plant.

8.4.3 System Design for Efficient Plant Management

The goal of a PV solar power plant management system is to increase the efficiency of energy generation while reducing the cost of operations and maintenance. It should cover all components of the solar plant, including energy storage systems. The system should also include asset management, monitoring, operations, preventive maintenance, corrective maintenance (repair), and disposition or replacement of the equipment.

The solar power plant management system should provide an integrated view of the operations. It is good practice for the system to provide the following functionalities at the minimum:

- Archive complete plant documentation for upkeep and reference.
- Monitor the dashboard with actual performance compared to expected for the day, month, or year.

- Customer/plant interaction tracking logs.
- Ticket or incident tracking.
- Mobile work-order flow management and documentation systems.
- Budget tracking.
- Workflow and decision support system.
- Asset Management.
- Integration and interoperability with the existing enterprise applications.

Standard data encryption techniques should be employed to protect the confidentiality and integrity of the data in transit over wide area networks. Onsite data storage is required to prevent data loss during communication network outages. It is recommended to have an onsite capacity for 6 months of data storage.

The system must standardize trouble-report code definitions, corrective actions taken, and results for auditing. This allows more definitive tracking of cause and effect, repetitive problems, and corrective action taken. This also leads to better operating efficiencies, better preventive techniques, and the identification of large-scale equipment problems.

The system should be designed based on the best practices for

- Monitoring
- Maintenance and measurement checklists
- Open standards
- Data presentation
- Machine learning and predictive analytics

8.4.3.1 Monitoring

Monitoring is an important aspect of solar plant operations. The objective of the monitoring should be to provide enough information to evaluate the number of solar resources available and the losses in each energy conversion process.

While designing the monitoring system, it is important to recognize that data is valuable and should be protected by appropriate measures. It should be established clearly who owns the monitoring data, who will access it, and for what purposes. In a monitoring system, one must take care that the system meets these basic requirements:

- Transparency of measurement protocols and procedures
- Ability to audit measurement protocols and procedures
- Ability to maintain hardware and software by a variety of service providers, including calibration and servicing requirements
- Ability of systems to share information with stakeholders securely
- Ability to ensure “operational continuity” (backup and restore)
- Support of third-party access for custom application development
- Security of software and applications

In addition, the system should have the capability to analyze data to evaluate solar power plant performance. It should be able to measure the following key performance indicators (KPIs) [22] at the minimum:

1. *Specific Yield*: Specific yield (kWh/kWp) is the energy (kWh) generated per kWp module capacity installed over a fixed period. For example,

$$\text{Daily specific yield}\left(kWh\,/\,kWp\right) = \text{Daily energy}\,/\,DC\ \text{capacity}$$

It indicates the number of full equivalent hours a plant produced energy during a specific time frame. The specific yield can be used for comparison of the production of a plant with different power plants or even different power production technologies. By comparing inverter-level-specific yields within a power plant, it is possible to detect which of the inverters is performing better than others.

2. *Capacity Utilization Factor*: Capacity utilization factor (CUF) is the output of the plant compared to the theoretical maximum output of the plant in a specific period. For example,

$$\text{Daily}\ CUF\left(\%\right) = \text{Daily energy}\left(kWh\right)/\left(\text{Plant capacity}\left(kWp\right)\times 24\right)$$

$$\text{Yearly}\ CUF\left(\%\right) = \text{Year energy}\left(kWh\right)/\left(\text{Plant capacity}\left(kWp\right)\times 24\times 365\right)$$

3. *Performance Ratio*: Performance ratio (PR) is indicated in percentage. It is the ratio between the actual and theoretical energy outputs of the PV plant. It indicates the proportion of the energy that is available after the deduction of energy loss (e.g., due to thermal losses and conduction losses).

8.4.3.2 Maintenance and Measurement Checklists

Maintenance of solar plants is critical for proper operations. Maintenance covers all components of the plant, including the cleaning of solar panels. If panels are not clean, then they impact the generation capacity of the plant. The impact on generation capacity can be reduced by periodic cleaning of the solar panels. The system should have provision to keep a complete record of the cleaning process, covering details like date, panels cleaned, and any problems that were identified during the cleaning process.

In addition, the system should maintain a standardized inspection checklist for the components. The system should record the details related to the inspection being carried out by the team. The checklist below for inspecting the components is not comprehensive but covers most of the items to be inspected. It can be customized to meet specific requirements of an installation.

1. Photovoltaic Modules (Solar Panels)

- Inspect modules for damage.
- Address array shading issues.
- Adjust array tilt for optimal sun exposure.

2. Mounting Systems

- Ensure proper functioning of the solar panel mounting system.
- Check for cracks or damage.

3. Solar Charge Controllers

- Check the functioning of solar charge controllers.
- Inspect for any visible damage.
- Verify the tightness of electric connections.

4. Energy Storage System

Inspection and testing of energy storage systems are required to ascertain the actual capacity over time. The checklist items include:

- Noninvasive tests such as impedance and voltage testing.
- Internal resistance/impedance.
- Voltage measurements.
  - Local-test batteries to assess performance.
  - Inspect the battery enclosure for any issues.
  - Inspect battery terminals and connections.

5. Inverter

- Inspect the interior cabinet for dust or debris.
- Verify the proper functioning of the inverter.
- Check for any signs of damage.

The system should maintain a history of the inspection of different components so that an analysis can be done to diagnose problems that require corrective maintenance or replacement of the components. The machine learning and analytics layer of the system carries out these analyses.

8.4.3.3 Open Standards

The selection of correct components and instruments is critical for the proper functioning of the plant. The component requirements may vary depending on the performance measurements, the model used, the required accuracy, and other considerations. The system should be developed using open standards for information and data communications.

The open standards must be used for the following four processes to ensure scalability and interoperability.

1. Device communication and plant sensor readings

2. Data collection and storage at the plant
3. Information transmission from the plant to the information database
4. Information access to the database from applications

Additionally, it is important to maintain transparency of procedures being implemented in the system.

8.4.3.4 Data Presentation

The data collected from the sensors should be of good quality. Data anomalies, such as missing data or highly inaccurate instrument readings for some time, should be highlighted immediately so that they do not skew the reported results.

A good data presentation of the collected data is equally important for the system. The reporting system should be powerful to extract meaningful insights into the functioning of the power plant. It should have accurate performance measurements, the ability to easily pinpoint issues, and prompt, cost-effective repair of any defects.

Reports should include the following information:

- Site name, location, size of PV plant, and other reference information.
- Insolation (onsite or satellite data, plane of array, kWh/m2), temperature (ambient, module).
- Real power and energy delivery (kW, kWh).
- Peak power delivery (kW).
- Other advanced meter data, such as reactive power (kVAR).
- Estimate of power that should have been produced and performance ratio.
- Time-based and energy-based availability.
- Inverter efficiency.
- Operations and maintenance plans should be built to notify actionable personnel on critical production or safety issues at the earliest.
- Fire alarms and intrusion detection alarms should be sent out immediately to the on-call personnel.

An Internet-accessible portal should be available with facilities to download raw data and a user-configurable dashboard with charts and tables to interpret this data. It increases operational awareness and collaboration across functional departments.

A data visualization tool lets users tap into a wide variety of data to create graphical displays or conduct ad hoc analyses to get insights into plant operations. It increases efficiency, resiliency, and sustainability by providing fast, easy, and secure access to real-time or historical data.

The system should provide reliable data backup and storage. The best practice is to store data from the logger for 6 months and then back up the data to Cloud storage.

8.4.3.5 Machine Learning and Predictive Analytics

Traditionally, the maintenance strategies are reactive and rely on visualization and alerting tools to identify a problem after it has occurred.

With advancements in machine learning, it is possible to identify anomalies by using deep learning tools to prevent catastrophic failures or other costly issues before they occur. The system should have deep learning tools to analyze the data collected from monitoring the components to forecast the remaining useful life of the assets. This will enable teams to manage maintenance plans based on urgency, criticality, and spare parts availability.

Machine learning algorithms and data analytics tools should provide teams with critical information based on analysis, such as (i) how to trade off costs versus risk and (ii) how to devise plans that maximize efficiency and profitability. The analytics should be able to determine how even subtle changes will influence asset performance based on user-defined operating criteria. The information also helps companies optimize energy use, compare asset performance, and diagnose anomalies.

These inputs are critical for power companies to stay in regulatory compliance and meet contractual obligations by enabling better asset and process outcomes. It also allows maintenance and engineering teams to work together proactively to evaluate assets before they fail, optimize maintenance schedules, and ensure that the best teams and resources are available to minimize downtime and disruptions.

8.5 Summary

Edge computing is becoming critical for developing scalable, efficient, and sustainable systems as it reduces latency, costs, and security risks. In this chapter, we have discussed three use cases for building energy management systems, medical data sharing among healthcare providers, and solar energy power plant management from a design perspective. For each use case, the best practices for designing the system, its architecture, and design have been described. These use cases should help any person who wishes to design an intelligent Edge computing application. It is important to follow these best practices and ensure that the system is designed based on industry standards and uses open protocols. The system should be designed for interoperability with other systems and devices. It should use design patterns to ensure it is easily implementable and compliant with regulatory requirements.

8.6 Points to Ponder

1. What are the obstacles in managing energy efficiently in intelligent buildings?
2. What data inputs does BEMS provide that lead to building operations insights?

3. A physician wants to quickly send patient data anonymously to an external specialist, e.g., a neurologist at a hospital, for consultation and advice. How it can be done?
4. A healthcare software provider is developing a machine learning model that needs to be trained on large, well-designed, well-labeled, diverse, and multi-institutional datasets to mitigate racial and socioeconomic biases. How it can be achieved?
5. Why is solar energy considered one of the best alternatives to fossil fuel-based energy? What are some of the popular use cases for solar energy?
6. What factors should be considered while designing a solar power plant management system to have maximum solar panel efficiency?

8.7 Answers

1. *What are the obstacles in managing energy efficiently in intelligent buildings?*

Three of the biggest obstacles in intelligent building energy management systems (BEMS) are:

- BEMSs need to provide a certain degree of comfort to building occupants, as every person has a different idea about what a comfortable environment feels like. Creating a comfortable environment and increasing energy efficiency are often seen as conflicting goals, especially in commercial high-rise buildings.
- Lack of service contracts is an obstacle. Without service contracts in place that can address small but necessary repairs, small issues can easily turn into much bigger problems.
- Competing interests between building owners and tenants, or even between different tenants, can make energy management more difficult. Generally, those paying the utility bills will seek energy-efficient solutions that lower costs, while occupant businesses prioritize the comfort of employees and clients.

2. *What data inputs does BEMS provide that lead to building operations insights?*

Some of the data inputs provided by BEMS that will lead to insights include:

- *Total energy consumption of systems and equipment connected to the electrical network*: Some of the systems are always operational, while other pieces of equipment and machinery may be connected only occasionally. BEMS provides both the total daily electrical consumption of the building and the role individual devices play in the overall energy usage.
- *Occupants' behavior*: Activity levels, behavior patterns, and comfort preferences of occupants are considered for all energy efficiency measures. BEMS provides this insight to formulate saving strategies.

- *Energy usage patterns*: BEMS provides the pattern of how and when the building uses energy. Reshaping these patterns is a key to cost reduction strategies.
- *Utility time of use charges*: BEMS provides insights into the timings of usage of various equipment. Shifting energy usage away from high-priced periods set by utility companies is a common way to generate savings.
- *Cyclical or seasonal factors*: Over time, BEMS captures information about the building's energy consumption patterns. The analytics module can consider these when generating proposed solutions.
- *Weather data*: Weather conditions can have a direct impact on energy use, specifically as it relates to HVAC systems. Collecting, compiling, and analyzing weather data enables BEMS to be proactive about HVAC energy consumption, especially on hot or cold days.

3. *A physician wants to quickly send patient data anonymously to an external specialist, e.g., a neurologist at a hospital, for consultation and advice. How it can be done?*

The physician can automatically anonymize identifiable personal information, such as name, medical number, or address, in medical reports, protocols, doctor's letters, and images and videos before sending them to third parties. It is a best practice to remove personal information before the data is shared.

For example, the personal medical data is:

Name: Mary Andrew
Address: 111 La Strada, Rome, Italy
Date of birth: 23 February 1986
Sex: Female
Medical conditions: Asthma

Medical data with personal information removed will look like this:

Case number: 1
Area of residence: Rome, Italy
Age group: 35–40 years
Sex: Female
Medical conditions: Asthma

4. *A healthcare software provider is developing a machine learning model that needs to be trained on large, well-designed, well-labeled, diverse, and multi-institutional datasets to mitigate racial and socioeconomic biases. How it can be achieved?*

It can be achieved by using the federated learning model. Federated learning is a paradigm for training ML models when decentralized data are used collaboratively under the orchestration of a central server [23]. The process is shown in Fig. 8.18.

In contrast to centralized training, where data from various locations is moved to a single server to train the model, federated learning allows for the data to remain

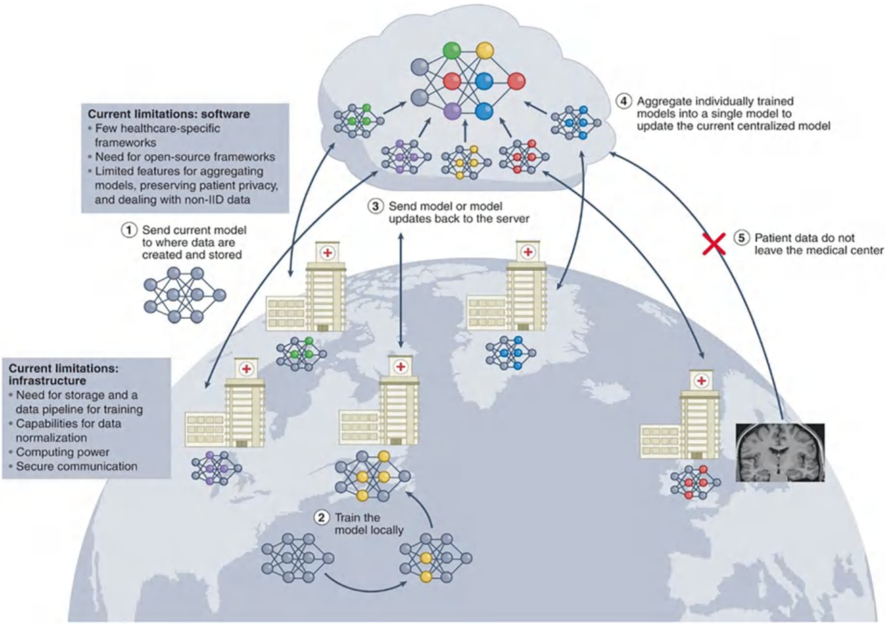

Current limitations: software
• Few healthcare-specific frameworks
• Need for open-source frameworks
• Limited features for aggregating models, preserving patient privacy, and dealing with non-IID data
① Send current model to where data are created and stored
③ Send model or model updates back to the server
④ Aggregate individually trained models into a single model to update the current centralized model
⑤ Patient data do not leave the medical center
Current limitations: infrastructure
• Need for storage and a data pipeline for training
• Capabilities for data normalization
• Computing power
• Secure communication
② Train the model locally

Fig. 8.18 Cross-silo federated learning for healthcare [24]

in place. At the start of each round of training, the current copy of the model is sent to each location where the training data are stored. Each copy of the model is then trained and updated using the data at each location. The updated models are then sent from each location back to the central server, where they are aggregated into a global model. The subsequent round of training follows, the newly updated global model is distributed again, and the process is repeated until the model converges or training is stopped.

The data never leaves a particular location or institution, and only individuals associated with an institution have direct access to its data. This mitigates concerns about privacy breaches, minimizes costs associated with data aggregation, and allows training datasets to quickly scale in size and diversity.

5. *Why is solar energy considered one of the best alternatives to fossil fuel-based energy? What are some of the popular use cases for solar energy?*

Solar energy is a clean and renewable energy source harnessing power from the sun without producing harmful pollutants or greenhouse gases. It is an almost inexhaustible source of energy. Solar energy has many advantages that make it popular as an alternate source of energy:

- Once installed, solar panels have relatively low operating and maintenance costs.
- Distributed solar power generation can enhance grid stability by reducing the need for centralized power plants and long-distance transmission lines.

- Solar energy systems are scalable and adaptable to various needs, from small installations to large utility-scale solar farms.
- Solar power allows individuals, businesses, and communities to generate their electricity, leading to reduced dependence on traditional utility grids.

Some of the popular applications of solar energy are:

- Solar water heating
- Solar distillation
- Solar heating of buildings
- Solar pumping
- Solar furnaces
- Solar greenhouses
- Solar cooking
- Solar electric power generation

However, solar energy requires the Sun to shine and be visible for energy generation.

6. *What factors should be considered while designing a solar power plant management system to have maximum solar panel efficiency?*

The management system should consider the following factors that affect the efficiency of solar cells:

- *Temperature*: Due to the intrinsic characteristic of the semiconductor material, solar cells' efficiency depends on the ambient temperature. The efficiency of solar cells is high at lower temperatures and reduces as the temperature increases.
- *Sun Intensity*: The intensity of the sun varies throughout the day. It is maximum in the afternoon and lower during evening and morning time. The efficiency of solar cells is maximum in the afternoon. The system should ensure the position of the panel in such a way that it gets the maximum sun throughout the day.
- *Solar Shading*: The efficiency of solar cells is highly dependent on solar shading. On a cloudy, rainy day, the solar cells do not generate energy at full capacity.
- *Reflection*: The solar cell works on the principle of photon energy. If the light is reflected away from the cell surface, the cell will not function optimally. To avoid this situation, an antireflection coating is used on the surface of the solar cells.
- *Cleaning of PV Panels*: The cleanliness of PV panels impacts the plant's generation capacity. The impact on generation capacity can be reduced by periodic cleaning of the solar panels. The system should have provisions to keep a complete record of the cleaning process.

References

1. Performance of energy management systems, Greg Wheeler, Oregon State University. https://www.aceee.org/files/proceedings/1994/data/papers/SS94_Panel5_Paper28.pdf
2. Mohamed, N., Lazarova-Molnar, S., & Al-Jaroodi, J. (2017). *CE-BEMS: A cloud-enabled building energy management system* (pp. 1–6). https://doi.org/10.1109/IC-BDSC.2016.7460393
3. Energy performance of buildings directive. https://energy.ec.europa.eu/topics/energy-efficiency/energy-efficient-buildings/energy-performance-buildings-directive_en. https://www.bpie.eu/wp-content/uploads/2021/09/Glossary-of-terms%E2%80%93Energy-efficiency-and-building-policies-in-the-EU_rev3.pdf
4. Dall'O', G. (Ed.). (2020). *Green planning for cities and communities*. Springer.
5. Yang, T., Clements-Croome, D., & Marson, M. (2017). Building energy management systems. *Encyclopedia of Sustainable Technologies.*, 291–309. https://doi.org/10.1016/B978-0-12-409548-9.10199-X
6. Hannan, M. A., Faisal, M., Ker, P. J., Mun, L. H., Parvin, K., Mahlia, T. M. I., & Blaabjerg, F. (2018). A review of internet of energy based building energy management systems: Issues and recommendations. *IEEE Access, 6*, 38997–39014. https://doi.org/10.1109/ACCESS.2018.2852811
7. LEED rating system: US Green Building Certification Standard. https://www.usgbc.org/leed
8. BREEAM sustainability rating scheme for green building certification. https://www.tuvsud.com/en-in/industries/real-estate/buildings/sustainability-rating-system/breeam-green-building-rating-system
9. Azar, E., & Menassa, C. C. (2011). Agent-based modeling of occupants and their impact on energy use in commercial buildings. *Journal of Computing in Civil Engineering, 26*, 506–518.
10. Ali, A. S., Chua, S. J. L., & Lim, M. E.-L. (2015). The effect of physical environment comfort on employees' performance in office buildings: A case study of three public universities in Malaysia. *Structural Survey, 33*, 294–308.
11. Andrews, C. J., Yi, D., Krogmann, U., Senick, J. A., & Wener, R. E. (2011). Designing buildings for real occupants: An agent-based approach. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, 41*, 1077–1091.
12. Mariano-Hernández, D., Hernández-Callejo, L., Zorita, A. L., Duque-Pérez, O., & García, F. . S. (2021). A review of strategies for building energy management system: Model predictive control, demand side management, optimization, and fault detect & diagnosis. *Journal of Building Engineering, 33*, 101692.
13. Tam, A. (2021, October). *A gentle introduction to particle swarm optimization in optimization*. https://machinelearningmastery.com/a-gentle-introduction-to-particle-swarm-optimization/
14. Lazarova-Molnar, S., Shaker, H. R., Mohamed, N., & Jørgensen, B. N. (2016). Fault detection and diagnosis for smart buildings: State of the art, trends and challenges. In *3rd MEC international conference on Big Data and Smart City 2016, Muscat, Oman*.
15. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2023). *Cloud computing with security and scalability. Concepts and practices*. Springer.
16. Patient Demographic Data Quality Framework, The Office of the National Coordinator for Health Information Technology. https://www.healthit.gov/playbook/pddq-framework/platform-and-standards/data-standards/
17. Yadav, K. S.. Healthcare data standards, LinkedIn Blog. https://www.linkedin.com/pulse/healthcare-data-standards-krishna-sayam-yadav-1f/
18. McCarthy, D. B., Propp, K., Cohen, A., Sabharwal, R., Schachter, A. A., & Rein, A. L. (2014). Learning from health information exchange technical architecture and implementation in seven beacon communities. EGEMS (Wash DC). 2014 May 5. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4371446/
19. Ahmad, R. W., Salah, K., Jayaraman, R., Yaqoob, I., Ellahham, S., & Omar, M. (2021). The role of blockchain technology in telehealth and telemedicine. *International Journal of Medical Informatics, 148*, 104399.

20. Zhang, Y., Li, S., Shi, Y., et al. (2023). Secure multi-party-join algorithms toward data federation. *International Journal of Software & Informatics, 13*(1), 117–137.
21. Trusted Exchange Framework and Common Agreement (TEFCA), Office of the National Coordinator for Health Information Technology. https://www.healthit.gov/topic/interoperability/policy/trusted-exchange-framework-and-common-agreement-tefca
22. Key Performance Indicators for Solar PV Plants. https://trackso.in/knowledge-base/key-performance-indicators-for-solar-pv-plants
23. McMahan, B., Moore, E., Ramage, D., Hampson, S., Aguera, Y., & Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In A. Singh & J. Zhu (Eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics* (pp. 1273–1282). ML Research Press.
24. Zhang, A., Xing, L., Zou, J., & Wu, J. C. (2022). Shifting machine learning for healthcare from development to deployment and from models to data. *Nature Biomedical Engineering, 6*, 1330–1345.

Chapter 9
Role of Regulatory

9.1 Introduction

The regulatory landscape for Edge AI, while still evolving, overlaps with the broader AI regulatory ecosystem but also presents some unique challenges. These challenges pertain to how existing regulations and emerging considerations apply to Edge AI. The role of existing regulatory bodies as applied to AI Edge computing is crucial to ensuring ethical, legal, and responsible development and deployment of AI technologies. As AI is increasingly integrated into Edge devices, regulatory frameworks become essential to address potential risks and ensure accountability. Here is a detailed exploration of the role of regulatory bodies in the context of AI Edge computing.

9.1.1 Early Stages

In the early stages of AI Edge computing, regulatory bodies began to recognize the potential of these technologies and the need for a regulatory framework to address their unique challenges.

9.1.2 Key Regulatory Considerations

Regulatory considerations span domains of data protection and privacy, security standards, interoperability and standards, ethical and responsible AI, and human rights and bias mitigation. We shall cover each of these in detail below.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_9

9.1.2.1 Data Protection and Privacy

Regulatory bodies, especially those focused on data protection and privacy, such as the FDA [1, 2], play a crucial role in defining guidelines for handling personal and sensitive data at the Edge. The General Data Protection Regulation (GDPR) [3, 4] in the European Union and the California Consumer Privacy Act (CCPA) [5] in the USA are examples of regulations that emphasize the rights of individuals in data processing.

9.1.2.2 Security Standards

Regulatory bodies are responsible for establishing security standards to safeguard AI Edge computing systems from cyber threats and unauthorized access. These standards help ensure the integrity, confidentiality, and availability of data processed at the Edge.

9.1.2.3 Interoperability and Standards

Regulatory efforts focus on promoting interoperability and setting standards for AI Edge computing systems. Standardization facilitates the seamless integration of diverse Edge devices, fostering a more cohesive and efficient ecosystem.

9.1.2.4 Ethical and Responsible AI

Regulatory bodies such as the FTC [6], FDA, and FCC [7] are increasingly interested in promoting ethical and responsible AI practices. Guidelines may include principles such as fairness, transparency, accountability, and inclusivity in the development and deployment of AI at the Edge.

9.1.2.5 Human Rights and Bias Mitigation

Regulators work toward ensuring that AI Edge computing technologies respect fundamental human rights. Efforts are made to address biases in AI algorithms, especially those that may lead to discriminatory outcomes.

9.1.3 Evolving Regulatory Landscape

This refers to the continuous changes and developments in laws, regulations, and policies such as security that govern various industries and sectors. This landscape is influenced by factors such as technological advancements, societal trends, economic conditions, geopolitical shifts, security, and emerging risks.

Security for some medical companies using Cloud computing, patient data collected in Europe contains Protected Health Information (PHI) [8], but it cannot be brought to servers in the USA for processing under existing laws. This leaves companies with two options. Either they move their intellectual property and crown jewel algorithms to Europe, thereby exposing them to potential leakage. Else, the company has to de-identify/anonymize patient data when bringing it to the USA for processing, and results are taken back to Europe for re-identification. This adds considerable overheads that include latency and data input-output (I/O) expense. There is an ongoing effort to evolve the laws so that healthcare businesses can continue to serve their patients.

9.1.3.1 Adaptability to Technological Advances

Regulatory bodies must maintain flexibility and adaptability to keep pace with the rapid evolution of AI Edge computing technologies. As new capabilities and use cases emerge, regulations need to be updated to address novel challenges.

9.1.3.2 International Collaboration

Given the global nature of AI Edge computing, regulatory bodies often engage in international collaboration. This involves harmonizing standards and regulations to create a cohesive framework that spans borders, fostering innovation while ensuring global compliance.

9.1.4 Enforcement and Compliance

This consists of Enforcement Mechanisms and Corporate Accountability as detailed below.

9.1.4.1 Enforcement Mechanisms

Regulatory bodies establish enforcement mechanisms to ensure compliance with AI Edge computing regulations. This may involve audits, inspections, and penalties for noncompliance to deter unethical or illegal practices.

9.1.4.2 Corporate Accountability

Regulators hold organizations accountable for the ethical use of AI at the Edge. Companies are encouraged to adopt transparent practices, conduct impact assessments, and implement measures to address potential risks associated with their AI systems. Companies are also encouraged to involve independent industry reviewers for better transparency.

9.1.5 Challenges and Future Directions

Regulations are evolving continuously as detailed below.

9.1.5.1 Regulatory Challenges

Regulatory bodies face challenges in keeping up with the rapid pace of technological advancements, the global nature of AI Edge computing, and the need to strike a balance between promoting innovation and safeguarding public interests.

9.1.5.2 Continued Evolution

The regulatory landscape for AI Edge computing will continue to evolve. Future regulations may focus on emerging issues such as explainability, accountability of autonomous systems, and the ethical use of AI in mission-critical sectors like healthcare and finance.

In conclusion, the role of regulatory bodies in AI Edge computing is multifaceted, encompassing data protection, security, ethical considerations, and international collaboration. A well-defined regulatory framework ensures that the deployment of AI at the Edge aligns with societal values, protects individual rights, and fosters innovation in a responsible and accountable manner.

9.2 Federal Trade Commission

The Federal Trade Commission (FTC) is an independent agency of the US government established in 1914 by the Federal Trade Commission Act [6]. The FTC serves as the primary federal agency responsible for protecting consumers and promoting competition in the marketplace. The FTC's mission is to prevent unfair or deceptive trade practices, maintain competition, and enforce antitrust laws to ensure a level playing field for businesses and consumers.

9.2.1 Historical Background

The FTC was established in response to concerns about unfair business practices, monopolistic behavior, and deceptive advertising prevalent in the early twentieth century. The Federal Trade Commission Act empowered the FTC to investigate and prosecute unfair methods of competition and unfair or deceptive acts or practices in commerce. Over the years, the FTC has evolved to address new challenges and emerging issues in the marketplace, including the rise of the digital economy, globalization of commerce, and technological advancements.

9.2.2 Key Functions and Responsibilities

Consumer Protection One of the primary functions of the FTC is to protect consumers from unfair, deceptive, or fraudulent business practices. The FTC investigates and takes enforcement actions against companies that engage in deceptive advertising, fraudulent marketing schemes, or other unfair trade practices. This includes false or misleading claims about products or services, deceptive pricing practices, and failure to disclose important information to consumers [9].

Competition Enforcement The FTC enforces antitrust laws to promote competition and prevent anticompetitive behavior in the marketplace. This includes investigating mergers and acquisitions that may harm competition, prosecuting anticompetitive conduct such as price-fixing or monopolization, and challenging anticompetitive practices in various industries. The FTC works to ensure that consumers have access to a wide range of choices and that businesses compete fairly to offer the best products and services at competitive prices.

Consumer Education and Outreach The FTC provides consumer education and outreach programs to empower consumers with information and resources to make informed decisions in the marketplace. This includes publishing consumer guides, educational materials, and online resources on topics such as identity theft, privacy protection, credit reporting, and online safety. The FTC also conducts public awareness campaigns and outreach events to raise awareness about consumer rights and responsibilities.

Privacy and Data Security The FTC plays a leading role in protecting consumer privacy and data security in the digital economy. It enforces laws and regulations related to the collection, use, and sharing of consumer data by businesses and online services. The FTC investigates data breaches, identity theft, and other cybersecurity incidents and takes enforcement actions against companies that fail to protect consumer information or engage in deceptive or unfair data practices.

Enforcement Authority The FTC has broad authority to investigate and prosecute violations of consumer protection and antitrust laws. It can issue subpoenas, conduct investigations, file administrative complaints, and seek civil penalties, injunctions, and other remedies against violators. The FTC also works closely with law enforcement agencies, regulatory bodies, and international partners to combat cross-border fraud, deceptive practices, and anticompetitive conduct.

Take the case of an enforcement action in July 2023 by the FTC related to illegal telemarketing calls to US consumers [10]. These were either personnel manually making the calls or devices at the Edge using Edge AI technologies including automated robocalls in humanlike voices. The intent was to falsely bait and dupe US consumers.

Here is another case of an enforcement action in April 2024 by the FTC related to employment [11]. FTC banned noncompete clauses from employment contracts even though it meant setting up a clash with businesses. Banning noncompete allows most employees to switch jobs between competing employers.

In summary, the FTC plays a vital role in safeguarding consumer interests, promoting competition, and enforcing laws to ensure a fair and transparent marketplace. As the economy and technology continue to evolve, the FTC remains committed to its mission of protecting consumers and promoting competition to benefit all Americans. As we live in a connected world, FTC actions indirectly benefit people living outside the USA too (e.g., scam calls in India).

9.3 Food and Drug Administration

The Food and Drug Administration (FDA), founded in 1930, is a regulatory agency of the US Department of Health and Human Services responsible for protecting and promoting public health. This is through the regulation and supervision of food safety, dietary supplements, prescription, and over-the-counter pharmaceutical drugs. FDA's purview includes medicines, vaccines, biopharmaceuticals, blood transfusions, medical devices, electromagnetic radiation-emitting devices (ERED), cosmetics, animal foods and feed, and veterinary products. The FDA also enforces other laws, notably Section 361 of the Public Health Service Act [12, 13] and associated regulations, many of which are not directly related to food or drugs. These include sanitation requirements on interstate travel and control of disease-carrying insects, rodents, and other pests. Additionally, the FDA is responsible for advancing public health by helping to speed innovations that make medicines more effective, safer, and more affordable. FDA also helps the public get accurate, science-based information they need to use medicines and foods to maintain and improve their health.

9.3.1 Historical Background

The origins of the FDA can be traced back to the late nineteenth and early twentieth centuries, a time when concerns about food and drug safety were growing in the USA. The passage of the Pure Food and Drug Act in 1906 marked the first federal regulation of food and drugs, leading to the establishment of the Bureau of Chemistry within the Department of Agriculture. Over the years, the agency underwent several reorganizations and expansions, culminating in the creation of the FDA in its current form.

Since then, the FDA has played a central role in shaping public health policy and protecting consumers from health risks associated with food, drugs, and other products. The agency has faced numerous challenges and controversies over the years, including drug safety scandals, foodborne illness outbreaks, and regulatory issues related to emerging technologies. Despite these challenges, the FDA has continued to adapt and evolve its regulatory framework to address new threats and promote public health.

In recent years, the FDA has focused on initiatives to modernize its regulatory processes, enhance transparency and communication with stakeholders, and promote innovation in healthcare. These efforts have included the implementation of new regulatory pathways for expedited drug approvals and the development of guidance documents for emerging technologies such as digital health and gene therapy. FDA efforts further include expansion of regulatory oversight to address global health threats such as antimicrobial resistance and pandemic preparedness. FDA was effective in accelerating emergency use approval of COVID-19 vaccines [14].

9.3.2 Key Functions and Responsibilities

Regulatory Oversight One of the primary functions of the FDA is to regulate various products to ensure their safety, efficacy, and quality. This includes setting standards, conducting inspections, and issuing approvals for products such as drugs, medical devices, and food items. The FDA regulates products throughout their life cycle, from development and testing to manufacturing and distribution.

Product Approvals The FDA reviews and approves new drugs, biologics, medical devices, and other products before they can be marketed and sold to the public. This process involves evaluating data from clinical trials and assessing the risks and benefits of the product. The FDA may also require post-market surveillance to monitor the safety and effectiveness of approved products.

Labelling and Advertising The FDA regulates the labeling and advertising of food, drugs, and other products to ensure that they are accurate, truthful, and not misleading. This includes requirements for product labeling, package inserts, and advertising materials. The FDA also monitors direct-to-consumer advertising and promotional activities to ensure compliance with regulations.

Inspections and Enforcement The FDA conducts inspections of facilities involved in the manufacturing, processing, and distribution of regulated products to ensure compliance with regulatory standards. When violations are identified, the FDA may take enforcement actions, such as issuing warning letters, seizures, injunctions, or product recalls, to protect public health.

Public Health Education The FDA provides information and resources to the public to promote health and safety. This includes consumer advisories, educational materials, and outreach campaigns on topics such as healthy eating, medication safety, and disease prevention. The FDA also collaborates with healthcare professionals, industry stakeholders, and other organizations to disseminate accurate and science-based information.

Research and Innovation The FDA conducts research and fosters innovation to support its regulatory mission. This includes efforts to develop new regulatory approaches, evaluate emerging technologies, and improve scientific methods for assessing product safety and effectiveness. The FDA also collaborates with academic institutions, industry partners, and other government agencies on research initiatives.

9.3.3 Regulatory Medical Device Classifications

The FDA categorizes medical devices into different classes based on the level of risk they pose to patients and the regulatory controls needed to provide reasonable assurance of safety and effectiveness. The FDA's medical device classifications consist of three main classes, namely class I, class II, and class III (Fig. 9.1). They are described in detail below [15, 16].

Class I Medical Devices

Class I devices are considered low risk and are subject to the least regulatory control. They are typically simpler in design, pose minimal potential harm to patients,

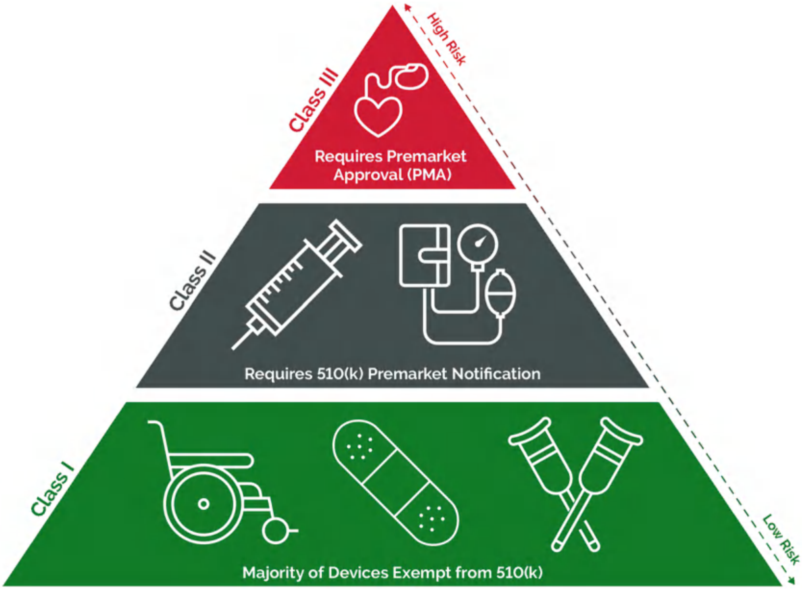

High Risk
Class III
Requires Premarket Approval (PMA)
Class II
Requires 510(k) Premarket Notification
Class I
Majority of Devices Exempt from 510(k)
Low Risk

Fig. 9.1 FDA medical device classifications [15]

and are often well understood. Examples include tongue depressors, bandages, non-powered wheelchairs, and elastic bandages. Most class I devices are exempt from the requirement of premarket notification, but they still need to comply with general controls, such as good manufacturing practices, labeling requirements, and registration with the FDA.

Class II Medical Devices

Class II devices are considered moderate risk and require greater regulatory control compared to class I devices. They may include devices that are more complex in design, have a higher potential for harm if used improperly, or are intended to support or sustain human life. Examples include infusion pumps, X-ray machines, certain types of catheters, and powered wheelchairs. Most class II devices require premarket notification through the 510(k) process. Manufacturers must demonstrate that the device is substantially equivalent to a legally marketed predicate device in terms of intended use, technological characteristics, and performance. Some class II devices may also require clinical data to support their safety and effectiveness.

Class III Medical Devices

Class III devices are considered high risk and are subject to the highest level of regulatory control. They are typically used to sustain or support life, are implanted into the body, or pose a significant risk of illness or injury. Examples include implantable pacemakers, heart valves, ventricular assist devices, and certain types of diagnostic imaging equipment. Class III devices require premarket approval (PMA) from the FDA before they can be marketed. This process involves a comprehensive review of scientific evidence, including clinical data, to ensure the device's safety and effectiveness. PMA is required for new devices or modifications to existing devices that raise new questions of safety or effectiveness.

Take the case of defective robotic procedure equipment being removed from the market due to the risks associated and harm to patients [17].

These classifications help ensure that medical devices are appropriately regulated based on their potential risk to patients, and they guide manufacturers through the regulatory pathways necessary to bring their devices to market. Additionally, the FDA may also classify certain medical devices as "combination products" when they involve a combination of a medical device with a drug or biological product.

9.3.4 Software as a Medical Device

Software as a Medical Device (SaMD) refers to software intended for medical purposes that performs its intended function without being part of a hardware medical device [18, 19]. SaMD plays a crucial role in modern healthcare, offering innovative solutions for diagnosis, treatment, monitoring, and management of various medical conditions. Unlike traditional medical devices that rely on physical components, SaMD operates solely through software algorithms, data processing, and user interfaces. This distinction introduces unique regulatory considerations, as SaMD presents challenges related to safety, efficacy, and data security.

Overall, the FDA plays a critical role in safeguarding public health and promoting the safety and effectiveness of food, drugs, and other products. Its regulatory authority extends across a wide range of industries and areas of public health, making it one of the most influential and important regulatory agencies in the USA.

9.4 Federal Communications Commission

The Federal Communications Commission (FCC) is an independent agency of the US government that regulates interstate and international communications by radio, television, wire, satellite, and cable. Established in 1934 by the Communications Act [20, 21], the FCC's primary mission is to ensure that communication services are accessible, reliable, and affordable for all Americans while promoting competition and innovation in the telecommunications industry.

9.4.1 Historical Background

The FCC traces its origins to the Federal Radio Commission (FRC), established in 1927 to regulate radio broadcasting in response to concerns about interference and chaos in the airwaves [22]. In 1934, Congress passed the Communications Act, which abolished the FRC and created the FCC as a successor agency with expanded authority to regulate all forms of interstate and international communication.

Since its creation, the FCC has played a central role in shaping the development of the communication industry and adapting to technological advancements and changes in the marketplace. It has overseen the transition from analog to digital broadcasting, the growth of cable and satellite television, the expansion of wireless and broadband Internet services, and the emergence of new communication technologies such as voice over IP (VoIP) and streaming media.

9.4.2 Key Functions and Responsibilities

Spectrum Allocation One of the FCC's most significant responsibilities is the allocation and management of radio frequency spectrum. The FCC assigns frequencies to various users, including broadcasters, wireless carriers, satellite operators, and government agencies, to prevent interference and ensure efficient use of the spectrum. It also licenses spectrum for commercial and noncommercial use through auctions and other mechanisms.

Licensing and Regulations The FCC issues licenses and regulates broadcasters, telecommunications providers, cable operators, satellite providers, and other entities involved in the communication industry. It sets technical standards, licensing requirements, and operating rules to ensure that communication services meet quality and reliability standards and comply with applicable laws and regulations.

Media Ownership Rules The FCC establishes rules and policies to promote diversity and competition in the media industry and prevent excessive consolidation of media ownership. It regulates ownership of broadcast stations, newspapers, and other media outlets to ensure that a diverse range of voices and viewpoints are available to the public.

Consumer Protection The FCC protects consumers by enforcing rules and regulations related to billing practices, service quality, privacy, and accessibility of communication services. It investigates consumer complaints, takes enforcement actions against companies that violate consumer protection laws, and educates consumers about their rights and responsibilities.

Public Safety and Homeland Security The FCC plays a critical role in ensuring the reliability and resilience of communication networks during emergencies and national security events. It works with industry stakeholders, government agencies, and public safety organizations to develop and implement strategies for maintaining communication services during disasters, terrorist attacks, and other emergencies.

Promotion of Innovation The FCC promotes innovation and investment in the communication industry by fostering competition, removing barriers to entry, and supporting research and development of new technologies. It encourages the deployment of advanced communication networks, such as 5G wireless and broadband Internet, to expand access to high-speed connectivity and drive economic growth.

In summary, the FCC plays a critical role in regulating and overseeing the communication industry to promote competition, protect consumers, and advance public policy goals such as universal access, public safety, and innovation. As technology continues to evolve and reshape the communication landscape, the FCC will face ongoing challenges and opportunities to adapt its regulatory framework and address emerging issues in the digital age.

9.5 General Data Protection Regulation

The General Data Protection Regulation (GDPR) is a comprehensive data privacy and protection law that was enacted by the European Union (EU) in May 2018 [3, 4]. It replaced the Data Protection Directive of 1995 and represents one of the most significant reforms in data protection regulation in recent years. The GDPR aims to harmonize data protection laws across EU member states, strengthen the rights of individuals regarding their personal data, and impose strict obligations on organizations that collect, process, or store personal data.

9.5.1 Key Principles of GDPR

GDPR has the following key principles [23]:

Lawfulness, Fairness, and Transparency Personal data must be processed lawfully, fairly, and transparently. Organizations must have a legal basis for processing personal data, and individuals must be informed about how their data is being used.

Purpose Limitation Personal data should be collected for specified, explicit, and legitimate purposes. Data should not be further processed in a manner that is incompatible with those purposes.

Data Minimization Organizations should only collect and process personal data that is adequate, relevant, and limited to what is necessary for the intended purpose.

Accuracy Personal data must be accurate and kept up-to-date. Organizations are required to take reasonable steps to ensure that inaccurate or incomplete data is corrected or erased.

Storage Limitation Personal data should be stored in a form that permits identification of individuals for no longer than is necessary for the purposes for which it is processed.

Integrity and Confidentiality Organizations must implement appropriate technical and organizational measures. This is to ensure the security of personal data and protect it against unauthorized or unlawful processing, accidental loss, destruction, or damage.

Accountability Organizations are responsible for complying with the principles of GDPR and must be able to demonstrate compliance through documentation, policies, and procedures.

9.5.2 Key Rights of Individuals Under GDPR

Right to Access Individuals have the right to obtain confirmation from organizations as to whether their personal data is being processed. If so, they also have the rights to access their personal data and obtain information about how it is being used.

Right to Rectification Individuals have the right to request the correction of inaccurate or incomplete personal data.

Right to Erasure (Right to Be Forgotten) Individuals have the right to request the deletion or removal of their personal data in certain circumstances. An example of this is when the data is no longer necessary for the purpose for which it was collected or when the individual withdraws their consent.

Right to Restriction of Processing Individuals have the right to request the restriction of processing of their personal data in certain situations. An example of this is when the accuracy of the data is contested or when the processing is unlawful.

Right to Data Portability Individuals have the right to receive their personal data in a structured, commonly used, and machine-readable format. An example of this is the ability to transmit that data to another controller without hindrance.

Right to Object Individuals have the right to object to the processing of their personal data for direct marketing purposes or on grounds relating to their particular situation. Organizations must stop processing the data unless they can demonstrate compelling legitimate grounds for the processing that override the interests, rights, and freedoms of the individual.

9.5.3 Key Obligations for Organizations Under GDPR

Data Protection Officer (DPO) Some organizations are required to appoint a Data Protection Officer responsible for overseeing GDPR compliance.

Data Protection Impact Assessments (DPIAs) Organizations must conduct DPIAs for processing activities that are likely to result in a high risk to the rights and freedoms of individuals.

Data Breach Notification Organizations must notify the relevant supervisory authority of data breaches within 72 h of becoming aware of them, unless the breach is unlikely to result in a risk to the rights and freedoms of individuals.

Privacy by Design and Default Organizations are required to implement technical and organizational measures to integrate data protection principles into the design of systems, products, and services. This is to ensure that only minimally necessary personal data is processed by organizations.

Data Transfers Organizations must ensure that any transfer of personal data to countries outside the European Economic Area (EEA) is done in compliance with GDPR requirements, such as by implementing appropriate safeguards or obtaining explicit consent from individuals.

Consent Organizations must obtain freely given, specific, informed, and unambiguous consent from individuals for the processing of their personal data. Consent must be easy to withdraw, and organizations must be able to demonstrate that consent was obtained.

9.5.4 Enforcement and Penalties

GDPR is enforced by National Data Protection Authorities (DPAs) in each EU member state, with the European Data Protection Board (EDPB) providing guidance and coordination at the EU level. DPAs have the power to investigate complaints, conduct audits, and impose administrative fines for violations of GDPR. Fines can be significant, with penalties of up to €20 million or 4% of the organization's global annual revenue, whichever is higher, for serious infringements.

9.5.5 Impact and Global Reach

Although GDPR is an EU regulation, its impact extends far beyond the borders of the EU. Many organizations around the world are subject to GDPR if they process personal data of individuals located in the EU, regardless of where the organization itself is based. GDPR has influenced data protection laws and regulations in other jurisdictions, and its principles have become widely recognized as best practices for data privacy and protection globally.

In summary, GDPR represents a comprehensive framework for protecting the privacy and rights of individuals in the digital age. By establishing clear principles, rights, and obligations for organizations, GDPR aims to create a more transparent, accountable, and trustworthy environment for the processing of personal data. Compliance with GDPR requires ongoing efforts by organizations to ensure that personal data is collected, processed, and protected in a lawful and ethical manner. GDPR prioritizes the rights and freedoms of individuals.

9.6 California Consumer Privacy Act

The California Consumer Privacy Act (CCPA) is a landmark data privacy law enacted by the state of California in 2018 [5]. It represents one of the most comprehensive privacy regulations in the USA and has had a significant impact on data protection practices for businesses operating in California.

9.6.1 Key Provisions of CCPA

CCPA has the following provisions:

Consumer Rights CCPA grants California consumers several rights regarding their personal information. These include the right to know what personal information is being collected, the right to access personal information, the right to request deletion of personal information, and the right to opt-out of the sale of personal information.

Notice and Disclosure Covered businesses must provide consumers with clear and conspicuous notice about data collection and processing practices. These include the categories of personal information collected, the purposes for which the information is used, and the categories of third parties with whom the information is shared.

Data Minimization CCPA requires covered businesses to limit the collection of personal information to what is necessary for the purposes disclosed to consumers and to avoid collecting unnecessary or excessive information.

Data Security Covered businesses must implement reasonable security measures to protect consumers' personal information from unauthorized access, disclosure, alteration, or destruction. This includes safeguards such as encryption, access controls, and regular security assessments.

Nondiscrimination CCPA prohibits covered businesses from discriminating against consumers who exercise their privacy rights, such as by denying them goods or services, charging them different prices, or providing them with a different level or quality of service.

Children's Privacy CCPA includes additional protections for minors under the age of 16, requiring opt-in consent for the sale of personal information of minors under 13 and providing the right to opt-out for minors aged 13–16.

9.6.2 Applicability and Enforcement

CCPA applies to for-profit businesses that meet certain criteria, including having annual gross revenues exceeding $25 million and processing the personal information of at least 50,000 California consumers, households, or devices annually. CCPA also applies to for-profit businesses deriving at least 50% of their annual revenues from selling California consumers' personal information.

The enforcement of CCPA is overseen by the California Attorney General's Office, which has the authority to investigate violations, issue warnings or fines, and bring enforcement actions against noncompliant businesses. CCPA provides for civil penalties of up to $2500 per violation, or up to $7500 for intentional violations, and allows consumers to bring private lawsuits against businesses for certain data breaches.

In summary, CCPA represents a significant step forward in consumer privacy rights and data protection in the USA, providing consumers with greater control over their personal information. CCPA also imposes new obligations on businesses to safeguard consumer data and respect privacy preferences. Compliance with CCPA requires ongoing attention and investment from businesses, but it also presents opportunities for organizations to build trust with consumers and differentiate themselves in the marketplace as privacy-conscious entities.

9.7 Health Insurance Portability and Accountability Act

The Health Insurance Portability and Accountability Act (HIPAA) [24] was enacted in 1996. It has two main purposes: to provide continuous health insurance coverage for workers who lose or change their jobs, and to ultimately reduce the cost of healthcare by standardizing the electronic transmission of administrative and financial transactions. Signed into law by President Bill Clinton on August 21, 1996, HIPAA has evolved significantly in response to the increasing prevalence of data breaches and cyber threats within the healthcare sector.

HIPAA comprises several key components, notably its **Titles** and **Rules**, which establish standards for the protection of health information. The act is divided into five main titles, with **Title I** focusing on health insurance reform and **Title II** addressing administrative simplification through privacy and security regulations for Protected Health Information (PHI).

9.7.1 Key Titles of HIPAA

Here are the five titles of HIPAA in detail.

9.7.1.1 Title I: Health Care Access, Portability, and Renewability

This title ensures that individuals who change or lose their jobs can maintain their health insurance coverage. It prohibits group health plans from denying coverage based on pre-existing conditions and mandates that insurers provide alternatives for individuals leaving group plans.

9.7.1.2 Title II: Administrative Simplification

Title II is crucial as it mandates national standards for electronic healthcare transactions and establishes privacy and security rules to protect PHI. This title includes several specific regulations:

- **Privacy Rule:** This rule sets national standards for the protection of PHI, limiting its use and disclosure without patient consent. It grants patients' rights regarding their health information, including access to their records.
- **Security Rule:** This rule outlines requirements for safeguarding electronic PHI (ePHI) through administrative, physical, and technical safeguards to prevent unauthorized access.
- **Breach Notification Rule:** Organizations must notify affected individuals and the Department of Health and Human Services (HHS) in the event of a breach involving unsecured PHI.

- **Enforcement Rule:** This rule establishes procedures for compliance investigations and penalties for violations of HIPAA regulations.

9.7.1.3 Title III: Tax-Related Health Provisions

This title includes provisions related to tax treatment of medical care and standardizes pre-tax medical expenditure accounts.

9.7.1.4 Title IV: Application and Enforcement of Group Health Plan Requirements

Title IV addresses requirements for group health plans, ensuring that they comply with HIPAA mandates while providing coverage.

9.7.1.5 Title V: Revenue Offsets

This title deals with revenue offsets related to health insurance coverage but is less frequently referenced compared to the other titles.

9.7.2 Protected Health Information Details

PHI refers to any individually identifiable health information held or transmitted by a covered entity or business associate. According to HIPAA, PHI encompasses a broad range of data that relates to an individual's past, present, or future health status [25]. Specifically, PHI includes:

- Demographic information such as names, addresses, birth dates, and Social Security numbers.
- Medical histories and records.
- Test results and laboratory findings.
- Insurance information related to healthcare services.

PHI does not include de-identified information that cannot be used to identify an individual. For instance, if an individual's health information is stripped of all identifiers, it is no longer considered PHI under HIPAA regulations.

9.7.3 HIPAA Compliance and Covered Entities

HIPAA compliance is mandatory for covered entities—healthcare providers, health plans, and healthcare clearinghouses that electronically transmit any health information. Additionally, business associates—entities that perform functions or activities on behalf of a covered entity involving the use or disclosure of PHI—are also subject to HIPAA regulations.

To ensure compliance with HIPAA's Privacy and Security Rules, covered entities must implement various safeguards:

- **Administrative Safeguards:** Policies designed to manage security measures protecting ePHI include training employees on compliance and limiting access to authorized personnel.
- **Physical Safeguards:** Measures such as securing physical locations where ePHI is stored and restricting access to sensitive areas.
- **Technical Safeguards:** Technology-based protections like encryption, firewalls, and secure user authentication processes aimed at preventing unauthorized access.

9.7.4 Impact of HIPAA

HIPAA has significantly impacted how healthcare providers manage patient information. By establishing national standards for privacy and security, it has enhanced patient trust in healthcare systems. Noncompliance can lead to severe penalties ranging from fines up to $50,000 per violation and possible jail time depending on severity.

In recent years, the rise in cyber threats targeting healthcare data has intensified HIPAA's relevance. Organizations must continuously adapt their practices to safeguard sensitive patient information against breaches while complying with evolving regulatory frameworks.

HIPAA remains a cornerstone of patient protection in the USA, balancing the need for privacy with efficient healthcare delivery. The ongoing challenge lies in maintaining compliance amid rapid technological advancements and increasing cyber threats while ensuring that PHI is handled securely across all platforms.

9.8 Regulatory Policies in Emerging Markets Case Study

Among a number of emerging markets, India has the highest growth rate [26].

Regulatory practices in India encompass a wide range of regulations, laws, and institutions that govern various sectors of the economy, society, and environment. India, being a diverse and populous country with a rapidly growing economy, requires robust regulatory frameworks to ensure fair competition, consumer protection, environmental sustainability, social welfare, and overall economic development. Here is an overview of regulatory practices in India across different sectors as relevant to Edge computing:

Environmental Regulation The Ministry of Environment, Forest and Climate Change (MoEFCC) formulates policies and regulations related to environmental protection and conservation. Additionally, the National Green Tribunal (NGT) has been established as a specialized body to handle cases related to environmental protection and enforcement of environmental laws [27]. For remote monitoring of environmental conditions such as pollution levels, it would require Edge devices and sensors. This may drive up market growth and new business opportunities, as described in a separate chapter of this book.

Food Safety Regulation The Food Safety and Standards Authority of India (FSSAI) is the regulatory body responsible for ensuring food safety and regulating the manufacture, storage, distribution, sale, and import of food products in India [28]. This would require remote sensors and monitoring of food products during transportation and storage.

Healthcare Regulation The Central Drugs Standard Control Organization (CDSCO) regulates the pharmaceutical industry in India. It ensures the quality, safety, and efficacy of drugs and medical devices through the enforcement of regulatory standards and guidelines [29]. This impacts data collection using remote medical devices.

These are just a few examples of regulatory practices in India, and the regulatory landscape is constantly evolving to address emerging challenges and opportunities in a rapidly changing economic and social environment. Effective regulation requires coordination among various government agencies, stakeholder engagement, transparency, and enforcement mechanisms to ensure compliance and accountability while using Edge computing.

9.9 Future Developments

Here are some likely future developments regarding AI regulation in the USA:

National AI Strategy The USA may develop a comprehensive national strategy to guide AI research, development, and deployment across various sectors. This strategy may prioritize investments in R&D, promote talent development, have ethical and societal implications, and foster international collaboration on governance.

AI Ethics and Transparency Guidelines There is increasing recognition of the need for ethical and transparent systems. The US government may work with industry stakeholders and research organizations to develop ethics guidelines and standards that promote fairness, accountability, transparency, and privacy in development and deployment.

Regulatory Framework for AI The US government may introduce a regulatory framework specifically tailored to technologies to address concerns related to safety, bias, discrimination, privacy, and security. This framework may involve updating existing laws and regulations, such as the Fair Credit Reporting Act and the Civil Rights Act, to address AI-related challenges.

Data Governance and Privacy Data governance and privacy regulations may be strengthened to address the increasing use of systems that rely on vast amounts of data. This may include updates to data protection laws, such as the HIPAA [30] and the CCPA, to address new challenges posed by AI technologies.

AI Regulatory Agencies or Task Forces The US government may establish specialized regulatory agencies or task forces to oversee governance and regulation. These agencies could be responsible for monitoring developments, conducting risk assessments, issuing guidelines, enforcing regulations, and promoting public awareness and engagement on AI-related issues.

Sector-Specific Regulations The US government may introduce sector-specific regulations for AI applications in critical sectors such as healthcare, finance, transportation, and defense. These regulations may address sector-specific risks and requirements, such as clinical validation for medical systems, algorithmic trading regulations for financial systems, and safety standards for autonomous vehicles.

International Collaboration The USA may engage in international collaboration and cooperation on AI regulation to exchange best practices, harmonize standards, and address cross-border challenges. This may involve participation in international forums, such as the Global Partnership on Artificial Intelligence (GPAI) and the Organisation for Economic Co-operation and Development (OECD) AI Policy Observatory [31].

Stakeholder Engagement The US government may adopt a multi-stakeholder approach to AI governance and regulation, involving government agencies, industry stakeholders, academia, civil society organizations, and the public. This may include establishing platforms for dialogue, consultation, and collaboration to ensure that AI regulation reflects diverse perspectives and interests.

Innovative Products and Services By establishing clear guidelines for Edge AI, the US government can assist in the creation of a trustworthy environment that encourages innovation. For instance, stringent safety standards for autonomous vehicles and medical devices ensure these technologies can be deployed confidently, leading to advancements in transportation and healthcare. Additionally, interoperability standards promote seamless integration of various Edge AI systems, driving the creation of smart cities and homes. As regulations evolve to keep pace with technological advancements, they enable the responsible and widespread adoption of innovative Edge AI solutions, transforming industries and improving daily life.

Overall, the emerging future developments in AI regulation in the USA are likely to be driven by the need to balance innovation and economic competitiveness with ethical considerations, risk mitigation, and societal values. By proactively addressing AI regulation, the USA can harness the transformative potential of AI while minimizing potential risks and maximizing societal welfare.

9.10 Summary

As AI continues to pervade all aspects of our lives, incidents of intentional or unintentional misuses may also continue to rise. In response, societies will demand new regulations, and governments are expected to respond commensurately. Businesses using Edge computing need to be cognizant of evolving regulatory environments and remain compliant. This will represent additional costs of doing business. However, in a number of cases, this will also offer opportunities for innovative products and services.

9.11 Points to Ponder

1. What's the role of regulations in the medical industry?
2. In which cases one may treat software as a separate medical device vs. an adjunct to a medical device?
3. Why FDA has reservations in using AI/ML-based techniques for medical diagnosis?
4. What is the medical value of data de-identification?
5. Is it sufficient to remove a patient's information for broad data sharing?
6. Should a patient be given access to his own medical data?
7. What are similarities and differences between GDPR and CCPA?
8. Under what conditions does the FTC take legal action against companies and M&A?
9. Do regulatory agencies in developed nations overreach in their actions?

9.12 Answers

1. What's the role of regulations in the medical industry?

The role of medical regulation is to protect the interests of medical service consumers and patients.

2. In which cases one may treat software as a separate medical device vs. an adjunct to a medical device?

   If a software application is used to collect data from another medical device and display it along with historical values, then the software is not treated as a separate medical device. However, if the software application is used to identify abnormal values or give interpretations of the data, then it is treated as a separate medical device and FDA approval is required.

3. Why FDA has reservations in using AI/ML-based techniques for medical diagnosis?

   FDA is not opposed to the use of AI/ML techniques being used for medical diagnosis but does require the presence of a human element in the final decision-making. This is no different from the role of software in self-driving cars. In case of an injury or accident, the question of liability arises. Invariably, the software providers or car manufacturers may not be willing to accept that kind of liability. It is for this reason they currently make it a requirement for human drivers to be always behind the wheel, even in the autopilot mode. For the same reason, the FDA requires a medical doctor to certify reports and results for the final diagnosis. However, doctors can use the software to assist and speed up their decision-making process. Consider the example of examining X-ray or MRI results of patients with bone injuries. The orthopedic analysis support software can highlight current as well as potential locations with displacement, gaps, or cracks. However, in the end, a doctor must examine and agree with the suggestions before a treatment can begin.

4. What is the medical value of data de-identification?

   Medical science advances by building upon knowledge of how patients react to the treatment and drawing generalized inferences for applying this knowledge to other patients with similar symptoms. However, due to human biases, sometimes a patient may feel discrimination in their careers or society if the news of their diseases is spread beyond a strict need-to-know basis. Another example is a future employer not wanting to hire a heart patient, thinking it will increase future health care costs for the company. Similarly, health information of political leaders is often a highly guarded secret. For instance, Pakistan's founder, Mr. Jinnah's, deteriorating health condition was kept highly secret. Even today, some people argue that if this information were known publicly, then the partition of India may not have happened. Thus, medical data needs to be de-identified before being shared across the medical community, as not sharing any data is detrimental to the progress of medical sciences.

5. Is it sufficient to remove a patient's information for broad data sharing?

   While it is important to remove a patient's identifiable information, it is also important to replace it with a uniquely hashed token. This will enable future data and results for the same patient to be correlated, so doctors can observe the progression of a disease and the efficacy of a treatment.

6. Should a patient be given access to his/her own medical data?

Current practice is to share medical data with the patient on a need-to-know basis. Also, since a patient may not be medically savvy, there is a tendency for doctors to up-level or obfuscate the details. This becomes a hindrance if a patient needs to seek health care away from home base. Hence, there is a need to give full access to a patient for his own medical records. This is fully supported by HIPPA.

7. What are similarities and differences between GDPR and CCPA?

   Similarities are that both require data to be stored in local geoservers and both protect the privacy of individuals.

   The difference is that GDPR is for the entire EU, but CCPA is California-focused. Also, CCPA is less restrictive than GDPR.

8. Under what conditions does the FTC (and in general regulatory agencies) take legal actions against companies, and does it overreach in its actions sometimes?

   FTC takes legal action in various situations.

   For example, FTC investigated Amazon, alleging anti-competitive practices, alleging the company used anticompetitive practices to keep its competitors from getting a foothold in the digital retail space.

   As another example, FTC got involved when Nvidia announced a $40 billion acquisition of Arm Holdings, a chipmaker whose designs are used in over 95% of the world's mobile devices. However, the deal was terminated in 2022 after facing significant regulatory opposition from around the world, including China. Regulators were concerned that the acquisition would give Nvidia an unfair advantage over its competitors and that Nvidia could use Arm's technology to harm them.

   But also, "with great power comes great responsibility." Regulatory agencies in developed nations are active in protecting consumer interests and preventing monopolies from forming, and in doing so, they can overreach in their actions.

   For example, FTC also took legal action against the acquisition of Activision by Microsoft. It was allowed to happen after Microsoft agreed to some concessions to make sure it did not impede competition through vertical integration.

   As another example, consider the breakup of AT&T in the 1980s. In 1984, AT&T's local telephone service was broken up into seven Baby Bells. The breakup gave consumers access to more choices and lower prices for long-distance service and phones. The breakup may have delayed the availability of high-speed Internet service for many consumers.

References

1. https://www.usa.gov/agencies/food-and-drug-administration
2. https://www.fda.gov/
3. https://commission.europa.eu/law/law-topic/data-protection
4. https://gdpr.eu/
5. https://oag.ca.gov/privacy/ccpa
6. https://www.ftc.gov

7. https://www.fcc.gov
8. https://www.hhs.gov/answers/hipaa/what-is-phi/index.html
9. https://www.ftc.gov/about-ftc/bureaus-offices/bureau-consumer-protection
10. https://www.ftc.gov/news-events/news/press-releases/2023/07/ftc-law-enforcers-nationwide-announce-enforcement-sweep-stem-tide-illegal-telemarketing-calls-us
11. https://www.ftc.gov/news-events/news/press-releases/2024/04/ftc-announces-rule-banning-noncompetes
12. https://en.wikipedia.org/wiki/Public_Health_Service_Act
13. https://www.ssa.gov/policy/docs/ssb/v7n8/v7n8p15.pdf
14. https://www.fda.gov/emergency-preparedness-and-response/coronavirus-disease-2019-covid-19/covid-19-vaccines
15. https://www.fda.gov/medical-devices/overview-device-regulation/classify-your-medical-device
16. https://www.arenasolutions.com/resources/articles/how-to-classify-your-medical-device-for-fda-approval/
17. https://www.tampabay.com/news/health/2024/02/12/da-vinci-surgical-robot-intuitive-surgical-inc-palm-beach-county/
18. https://www.fda.gov/medical-devices/digital-health-center-excellence/software-medical-device-samd
19. Kumar, S. N., & Bhatt Pramod Chandra, P. (2024). *Project management in cloud applications*. Springer. https://link.springer.com/book/10.1007/978-3-031-53890-2
20. https://transition.fcc.gov/Reports/1934new.pdf
21. https://bja.ojp.gov/program/it/privacy-civil-liberties/authorities/statutes/1288
22. https://en.wikipedia.org/wiki/Federal_Radio_Commission
23. https://www.cyberpilot.io/cyberpilot-blog/data-protection-principles-the-7-principles-of-gdpr-explained/
24. https://en.wikipedia.org/wiki/Health_Insurance_Portability_and_Accountability_Act
25. https://www.medsafe.com/hipaa-compliance/what-is-considered-phi-under-hipaa/
26. https://www.forbes.com/sites/afshinmolavi/2024/03/06/india-rises-as-emerging-markets-darling
27. https://www.greentribunal.gov.in/
28. https://www.fssai.gov.in/
29. https://cdsco.gov.in/
30. https://www.cdc.gov/phlp/publications/topic/hipaa.html
31. https://oecd.ai/en/

Chapter 10
Future of Edge AI

10.1 Introduction

The rapid adoption of artificial intelligence (AI) applications by various industries across the globe has risen in the last few years. The need for real-time responsiveness, minimal latency, and stringent privacy drives this escalating demand for AI applications. To meet these requirements, a solution is needed to enable data processing on the local devices to reduce dependence on the central servers for faster response times, better reliability, and higher security. These requirements are addressed elegantly by Edge computing. As the name suggests, Edge computing is a distributed computing paradigm that brings computing resources closer to the data sources. This convergence of AI with Edge computing has given rise to Edge AI.

With time, Edge AI will continue to mature, allowing for more robust real-time analytics and decision-making at the Edge. With advancements in communication networks, hardware and software developments, and artificial intelligence breakthroughs, Edge AI will become increasingly precise and effective.

Chapter 8 of this book shows how Intelligence at the Edge has fundamentally altered numerous industries by facilitating immediate decision-making, enhancing safety protocols, and optimizing operational effectiveness through different use cases. However, the progress and adoption of Edge AI are not without challenges. One of the biggest challenges facing Edge AI today is resource-constrained environments that are holding back Edge AI in terms of data processing and AI modeling.

In this chapter, we will explore the latest innovations in Edge AI and how they are poised to influence its adoption in the future, especially in sectors such as healthcare, manufacturing, transportation, retail, and entertainment. We also discuss the new and upcoming developments in Edge AI.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5_10

10.2 Evolution of AI at the Edge

Edge AI is not a new concept and has been used since the beginning of this century. Akamai Technologies pioneered the Edge computing concept in 1998 with a revolutionary content delivery network (CDN) approach to data processing. CDN brought the computing power closer to data sources rather than relying entirely on centralized data centers [1]. Initially, the focus was on shifting computing from centralized servers to local devices to reduce latency. Over the past decade, with advancements in communication and hardware technologies, Edge AI adoption has grown exponentially due to other advantages such as reduced bandwidth usage and enhanced privacy. Network bandwidth usage is reduced due to Edge AI because all the data is not transmitted to a central server, and some of it gets processed locally. Hardware capabilities have increased further to support AI and machine learning models at the Edge. This has enabled sophisticated decision-making capabilities on the Edge devices, introducing a new era in computing and data analysis where decisions can be made faster and more efficiently.

Edge AI is evolving to collect and process data using a pre-trained AI, machine learning, or deep learning model to make predictions and decisions. In some cases, Edge AI can use real-time data for ongoing performance improvements of existing trained models without continuous reliance on Cloud computing and central storage.

10.2.1 Key Drivers of Edge AI Growth

Several factors are driving the growth of Edge AI. These factors are best represented by BLERP [2], which stands for Bandwidth, Latency, Economics, Reliability, and Privacy. These have been major drivers for the adoption of Edge AI in industries like healthcare and manufacturing.

1. **Bandwidth:** Traditional Cloud-based AI systems require all new data (or images, in the case of vision) to be collected and uploaded to the Cloud for processing and analysis. This requires a large bandwidth to function normally. Since Edge AI keeps data processing local to the device, there is a minimal requirement to transfer the data to the Cloud platform. Therefore, the bandwidth usage for Edge AI-powered devices is not as high as those using traditional Cloud AI.
2. **Latency:** Edge AI uses local processing for inferencing and results. It eliminates the delays in data transfer (due to network traffic), provisioning, and execution on a centralized Cloud server. Depending on the Edge devices and their processing capabilities, the latency in delivering inference results can be measured in milliseconds instead of seconds.
3. **Economics:** Bandwidth usage of Edge AI devices is not high. The reduction in high-bandwidth network communication provides significant potential cost savings. Another cost of Cloud-based AI systems is large-scale data storage at the

central Cloud servers. For example, a camera-based Edge AI device does not require that all images be sent to the Cloud as these are processed locally.

Edge AI is more scalable than the Cloud in terms of local deployments. The Edge AI-enabled devices are relatively inexpensive and easy to deploy for many applications. Cloud-based AI systems expanding the Cloud data processing and storage services are not linearly scalable, often time-consuming, and costly.

4. **Reliability:** One of the most important characteristics of Edge AI is the increased independence it provides to all endpoint Edge devices by moving processes from the Cloud to the Edge. This allows the devices to perform uninterruptedly even if there are temporary disruptions in the network services. It improves the availability and reliability of the system as the processing is not halted due to the loss of network. The performance of the Edge devices is more reliable, with minimal lag and no single point of failure.
5. **Privacy:** The traditional AI systems transfer all the data to the central Cloud server for processing, which can be a major cause of data loss. Edge AI, on the other hand, processes the data locally, thereby reducing the chances of a breach or leakage of data. In addition, users can easily firewall the native data for external connections and define rules as to who can access the data stored within Edge devices. The entire system can easily be protected as an on-premise system. These measures minimize the chances of a data breach resulting in better system security while addressing user concerns regarding data privacy.

10.2.2 Key Technologies Behind Edge AI

Edge AI-enabled systems are complex and require support at multiple fronts to deliver the promised benefits. Listed below are some of the key technologies that are required to power the various functions and capabilities of Edge AI-enabled systems.

1. **Reliable Edge Computing Infrastructure:** The Edge infrastructure should be deployed using devices and gateways that can process and analyze data locally. These devices should have sufficient computational power, storage capacity, and connectivity to manage the data generated by IoT devices. The devices should support protocols that can provide secure data transfer from Edge to the Cloud. Edge computing infrastructure should also be able to support hybrid architectures.
2. **AI-Capable Edge Devices:** Edge devices should be based on low-power GPUs and TPUs that are capable of processing machine learning algorithms and neural networks. These devices can execute AI models to process data in real time, enabling intelligent decision-making at the Edge without the need to send data to centralized servers. The advancement in GPUs and TPUs has reduced the size and cost of Edge devices, making them accessible for a wide range of applications.
3. **AI Algorithms and Models:** The advancements in AI Algorithms and Models enable them to run on Edge devices with limited computing resources [3]. These

models are optimized for efficiency and performance, enabling real-time analytics and decision-making.

4. **Distributed Intelligence:** Edge AI is based on the premise that processing occurs at the Edge, but Cloud platforms remain crucial for tasks like model training and updating with global insights. A typical AI-enabled Edge system relies on distributed intelligence, where decision-making is not solely centralized but shared between Edge devices and Cloud platforms. It is important to have intelligent algorithms that can collaborate and adapt to changing conditions. A constructive interaction between Edge and Cloud is vital for optimal performance of the system.
5. **Data Preprocessing:** IoT devices generate a lot of data based on the applications. This data may be too voluminous or noisy, so it may not be possible to process it entirely at the Edge. It is important to have an effective mechanism to preprocess the data to extract relevant information before transmitting it to Cloud servers.
6. **Energy Efficiency:** Most of the Edge devices are located at remote locations, and it is not always possible to power them through a stable and reliable power supply. These Edge devices are mostly battery- or solar-powered. Therefore, these should be energy efficient. Optimizing algorithms and resource usage can extend the lifespan of Edge devices by reducing energy consumption.
7. **Security and Privacy:** Edge devices must have strong security measures in place to protect against cyber threats and unauthorized access to data and algorithms. Data privacy is equally important, especially when dealing with sensitive information. For example, medical data can be deidentified at the Edge before being sent to the central server for processing [4].
8. **Industry Standards Compliance:** The Edge systems must adhere to data governance regulations and industry standards to ensure ethical and legal use of data.

10.3 Distributed Artificial Intelligence

Distributed artificial intelligence (DAI) is a subfield of artificial intelligence research dedicated to the development of distributed solutions for problems. It emerged as a subfield of artificial intelligence in 1975, mainly dealing with interactions of intelligent agents [5]. DAI systems were conceived as a group of intelligent entities, called agents, that interacted through cooperation, coexistence, or competition. Multi-agent systems and distributed problem solving are the two main DAI approaches. In multi-agent systems, the focus is on how agents coordinate their knowledge and activities. For distributed problem solving, the major focus is how the problem is decomposed and the solutions are synthesized.

DAI takes advantage of large-scale computation and spatial distribution of computing resources. DAI systems do not require all the relevant data to be aggregated in a single location, in contrast to monolithic or centralized AI systems, which have tightly coupled and geographically close processing nodes. Therefore, DAI systems often operate on sub-samples or hashed impressions of very large datasets. In addition, the source dataset may change or be updated during the execution of a DAI system.

These properties allow it to solve problems that require the processing of very large data sets. DAI systems consist of autonomous learning processing nodes (agents) that are distributed, often at a very large scale. DAI nodes can act independently, and partial solutions are integrated by communication between nodes, often asynchronously [6, 7]. By their scale, DAI systems are robust, elastic, and loosely coupled. Furthermore, DAI systems are built to be adaptive to changes in the problem definition or underlying data sets due to the scale and difficulty in redeployment.

10.3.1 Evolution of Distributed Artificial Intelligence

There are minimum requirements that need to be met before an approach can be considered a distributed artificial intelligence (DAI). The three main characteristics of DAI are [8]:

- Distribution of tasks between agents.
- Distribution of powers.
- Method of communication of the agents.

These can be further detailed to differentiate DAI systems [9]. Some of the aspects that can be considered to differentiate DAI systems are:

- **Granularity of Agents:** The agents can be either acting at a task-level or a statement-level problem decomposition.
- **Agent's Knowledge**: This could be either redundant or specialized, but is generally heterogeneous.
- **Control Distribution in the system:** The control is distributed to ensure convergence of the problem to a solution. There are multiple classifications for control systems, such as benevolent or competitive; team or hierarchical; and static or shifting roles.
- **Communication Method:** The communication can happen through a blackboard model or a message model, and either at low or high-level content.

From a DAI system perspective, based on the approach adopted while designing the system, one can classify it as a distributed problem solving (DPS) system or multi-agent system (MAS). In the case of DPS, several branches work together to achieve a common goal. In the case of MAS, multiple independent agents work together, and their interactions lead to a solution and look for an emerging solution from their interactions.

10.3.1.1 Distributed Problem Solving

In the case of a distributed problem solving (DPS) system, multiple agents work together to solve a specific problem [10]. Cooperation among the agents is the key to solving the problems since no individual agent has sufficient information, knowledge, and capabilities to solve the whole problem. A challenge for DPS is to make sure that information and capabilities are correctly allocated in such a way that agents complement rather than conflict with each other.

The DPS approach is well suited to solve problems in the areas of distributed planning and control, interpretation, cooperating expert systems, cognitive models of cooperation, and human cooperation backed by digital tools. A typical approach adopted for DPS systems is to reduce a larger problem into interdependent subtasks—spatial, temporal, or functional. The partial solutions are then integrated to fit into an overall solution. Figure 10.1 shows the process adopted for distributed problem solving.

There are multiple advantages of DPS:

1. It is relatively cheap to connect multiple devices from a hardware standpoint, even cheaper than having a centralized processor.
2. Many applications are distributed by nature and design. The ability to modularize the problem into subproblems is a great advantage, as modules are easier to check, debug, and maintain.

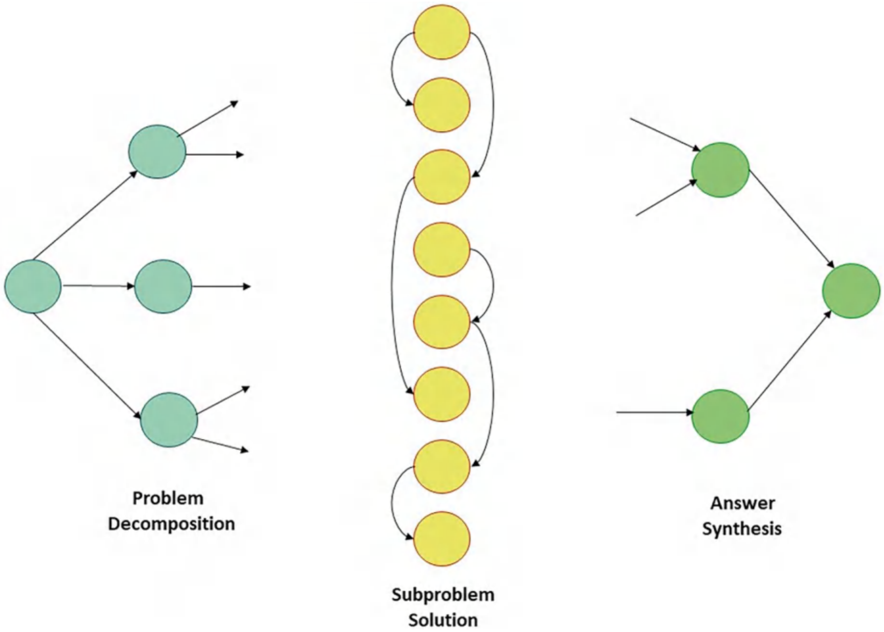


Fig. 10.1 Distributed problem-solving process

3. It facilitates the integration of AI with human intervention.

DPS is not without its challenges. The major challenge is cooperation among multiple agents, which increases the complexity of the system exponentially.

10.3.1.2 Multi-agent Systems

Multi-agent systems (MAS) are based on the principle that individual agents interact with each other based on predetermined rules/constraints and, because of this interaction, they can come to an acceptable solution to the problem.

The interactions cover two aspects: (i) between agents and (ii) between agents and the environment. An individual agent does not know the full problem space and can only partially solve the problem. Therefore, it must discover the solution by learning. Typically, multi-agent systems use reinforcement learning, deep learning, or deep convolutional networks to know about the environment.

Machine learning can use agent-based models (ABM) as an environment and a reward generator while ABM can use machine learning to refine the internal models of the agents. The learning can be improved by integrating it with Mean-Field Games [11]. This facilitates not only the interaction between individual agents but also tracks the decision-making process in huge groups of agents. This approach enables us to understand how a single agent acts in response to a group (and vice versa).

ABMs are used in several applications, from urban planning to epidemiology, from economics to transportation. Regardless of the application, multi-agent systems have similar characteristics, as shown in Fig. 10.2:

- Many agents (whether “intelligent” or not) at various scales.
- An environment where they operate.
- A set of learning rules and decision-making heuristics to regulate the exchanges with other agents.
- A map of what interactions are possible and how.

10.3.1.3 Swarm Intelligence

The inspiration for swarm intelligence (SI) comes from Mother Nature. Swarm or collective intelligence consists of multiple agents (autonomous entities performing the task) that are decentralized and capable of self-organizing. The term was first coined by Bloom [12] in 1995 while researching complex adaptive systems.

Swarm intelligence uses this indirect coordination mechanism, called *stigmergy* [13]. When a task is performed in the environment, a trace is left intentionally by the agent. The leftover trace then triggers another event. In this manner, the whole series of tasks are performed until the defined goal is achieved.

A typical swarm system has the following properties:

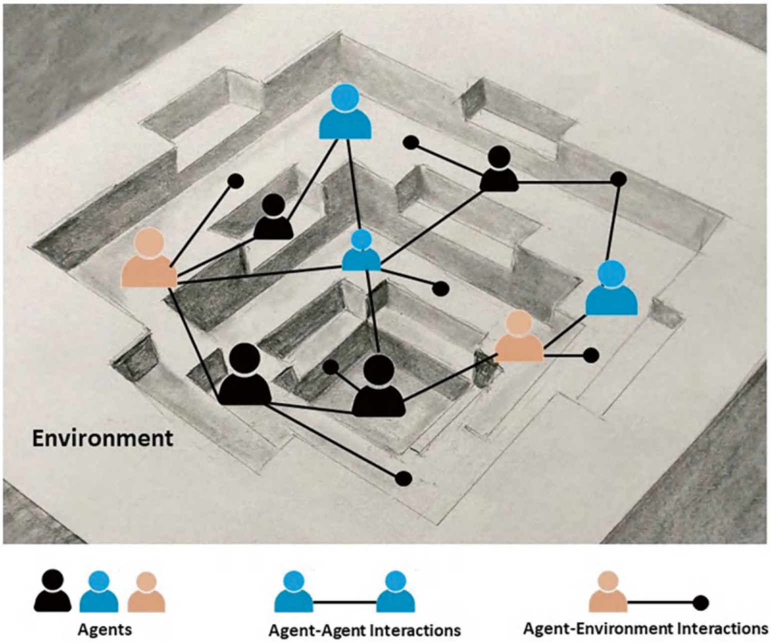


Fig. 10.2 Multi-agent systems

- It has many agents, which are homogeneous (either identical or belonging to few typologies).
- Agents interact according to basic rules that only exploit local information exchanged directly with another agent or via the environment.
- The group of agents eventually self-organizes and results emerge from the system's overall behavior.

The individual behaviors can be described in probabilistic terms, i.e., each agent stochastically acts based on his local perception of the neighborhood. The stochastic behavior of the agents and the above properties ensure that the system can be scaled, parallelized, and made fault-tolerant (they keep working even when parts malfunction), as well as completely decentralized and unsupervised.

The difference between the multi-agent system and the swarm intelligence is that MAS has heterogeneous agents whereas SI has homogeneous agents.

10.3.2 Research in Distributed Artificial Intelligence

Research in distributed AI is gaining attention due to the need for efficient resource management and reliable operations. The prime focus is on *improving efficiency, scalability, and decision-making capabilities* through the integration of AI techniques and distributed architectures.

The large amount of data produced by IoT-driven applications requires data-driven solutions for resource management. Multiagent systems distribute problem-solving tasks among autonomous processing nodes to handle this large amount of data. The research is focused on designing algorithms to optimize such large-scale distributed AI systems. This section will discuss optimizing algorithms for various DAI approaches, from swarm intelligence to federated learning.

10.3.2.1 Swarm Intelligence Algorithms

Swarm intelligence (SI) algorithms have been designed by looking at the natural swarm agents' behaviors. Swarm intelligence is used in controlling robots and unmanned vehicles, predicting social behaviors, optimizing telecommunication and computer networks, etc. Some of the common SI algorithms are described below:

1. **Particle Swarm Optimization:** Particle swarm optimization (PSO) [14] is inspired by the social flocking behavior of birds and the schooling behavior of fish. The algorithm makes use of all agents to locate the optima in a multidimensional space. The initial optimum is assigned with any random position and velocity in space. With the passage of time and continuous exploration and exploitation, the optima may be found. This algorithm has been used for dimensionality reduction in machine learning and hyperparameter tuning in deep learning.
2. **Ant Colony System:** The ant colony system (ACS) [15] is inspired by the communication of the ants, which is done by using a harmonic chemical known as a pheromone. The ant's probability of choosing the path is a function of the chemical intensity and the distance between the locations. The algorithm uses historical information and constructs the solution for the individual agent using a probabilistic step-wise approach. The probability of selecting any component for constructing a solution depends on that component's heuristic contribution to the overall cost function. Once the cost function is calculated, the history related to that path is also updated. This approach is used to achieve the desired goal by the collective behavior of simple robotic agents.
3. **Artificial Bees Colony:** Artificial bees colony (ABC) [16] is inspired by the natural communication and distribution within the beehive. The scout bees are sent from the hive to locate the nectar. They return to the hive and give information about the hive's location, fitness, and distance of food using a waggle dance. The algorithm can be used as an alternative optimization function to traditional gradient descent algorithms. It can also be used for solving clustering problems as an alternative to traditional clustering approaches.

10.3.2.2 Multi-agent System

Multi-agent systems have been in existence since the start of distributed artificial intelligence. The initial research in MAS was confined mainly to the game theory and negotiation theory. The advancement of machine learning and deep learning has enabled multi-agent systems to explore newer technologies, such as reinforcement learning.

Reinforcement learning [17] is concerned with how an intelligent agent should act in a dynamic environment to maximize the cumulative reward. It works by learning a *policy or* a function that maps an *observation* obtained from its environment to an *action.* To maximize the reward, the focus is on finding a balance between exploration (of uncharted territory) and exploitation (of current knowledge)

Multi-agent reinforcement learning (MARL) is about developing multiple reinforcement learning agents. These agents learn by dynamically interacting with the environment in which they are present. MARL makes it possible for multiple agents to interact with the environment and one another to collaborate, coordinate, compete, or collectively learn to accomplish a particular task. It can be further broken down into three broad categories:

- **Cooperative**: All agents working toward a common goal. The agents get the same rewards, and therefore they are playing with each other.
- **Competitive**: Agents competing with one another to accomplish a goal. The agents' rewards are exactly opposite to each other, and therefore they are playing against each other.
- **Mixed Mode:** It combines elements of both cooperation and competition.

MARL is a new field of research and has many promising use cases to which it can be applied. For example:

1. Online resource allocation in a computing network.
2. For cellular network optimization, it can guide base stations to maximize mobile service quality.
3. Smart traffic control systems can control traffic lights to minimize wait time for each car in a city. This makes traffic lights more adaptable based on estimates of expected wait time.

10.3.2.3 Federated Learning

We learned about federated learning basics in both Chaps. 6 and 7 of this book. In traditional machine learning approaches, the machine learning models are trained on data that is aggregated from several Edge devices like smartphones, laptops, etc. and brought together to a centralized server. The learning process happens in this centralized data store, where machine learning algorithms like neural networks train themselves on the aggregated data and make predictions on new data.

The major concern while training machine learning models centrally is the risk of compromising data security or privacy. Federated learning is a way to develop and validate accurate, generalizable AI models from diverse data sources while mitigating the risk of compromising data security or privacy. It enables AI models to be built from a vast range of data located at different sites without the data ever leaving individual sites. It is very useful in the case of healthcare, where privacy is a major concern.

Federated learning is simply the decentralized form of machine learning. The learning methods are distributed across the Edge devices themselves. At the start of each round of training, the current copy of the model is sent to each location where the training data are stored. Each copy of the model is then trained and updated using the data at each location. The updated models are then sent from each location back to the central server, where they are aggregated into a global model. The subsequent round of training follows, the newly updated global model is distributed again, and the process is repeated until the model converges or training is stopped.

This decentralized machine learning approach has several advantages, including improved data privacy, and reduced bandwidth requirements compared to sending raw data. Due to its various advantages, federated learning is gaining traction beyond healthcare and is moving into financial services, cybersecurity, transportation, high-performance computing, energy, and drug discovery.

Federated learning is still in its infancy, and research is required to address the issues related to:

1. **Heterogeneity:** The heterogeneity comes from three aspects: data, model, and system, as different organizations are collaborating to train the model.
2. **Limited Visibility of Data:** As data is distributed across organizations, there is limited visibility of the training data and a potential lack of trust among participants training a single model.
3. **Potential Privacy Inferences:** The parameters of the uploaded models may also be exploited by attackers to infer user privacy information. There is a need to have rigorous encryption or obfuscation methods to ensure privacy.
4. **Limited or Unreliable Connectivity:** The participants in federated learning come from various parties with different data resources that may have limited or unreliable connectivity. There is a need to have federated models that go beyond average accuracy to take care of this unreliability.
5. **Security and Performance:** Federated learning requires participating parties to run a common binary code on each dataset and trust the incoming program. This can potentially create a security hazard with malicious code. Another issue is the training run time due to multiple hops between different dataset locations. There is a need to come up with models that balance the trade-offs between higher performance and better security.

10.4 AI as a Service

Artificial Intelligence as a Service (AIaaS) is a service offered by third-party vendors that allows businesses to incorporate AI-powered tools and capabilities in their systems. It is a low-risk and cost-effective model for enterprises to experiment with AI and check its suitability for their business processes.

AIaaS provides out-of-the-box platforms and is easy to set up, making it simple to test out various platforms, services, and machine learning algorithms. Users can access application programming interfaces (APIs) and tools without having to write any complex code to harness the capabilities of AI. If successful, enterprises can invest in resources to build and implement AI systems from scratch.

Different AI provider platforms offer various services that enterprises can leverage differently depending on operational needs. Like the Software as a Service (SaaS) business model, companies can subscribe to AIaaS services. Given below are some of the common types of AIaaS and their use cases.

- **Bots and Virtual Assistants:** Bots and virtual assistants are types of conversational AI that use machine learning algorithms and natural language processing (NLP) to learn from human interactions. These are widely employed across all industries. They are generally used in customer self-service, such as troubleshooting common issues or providing relevant answers to customers' most frequent queries. Some of the common examples are ChatGPT, Alexa, Siri, or Google Assistant.
- **Machine Learning Frameworks:** Machine learning frameworks (MLF) are Cloud-based software libraries and tools that allow developers to build custom AI models. Enterprises can use them to build customized models to investigate and identify trends in their data and make predictions. MLF comes in a variety of options, from pre-trained models to models designed for a particular use case. Some of the common examples are Google Cloud AI and Microsoft Azure machine learning.
- **Application Programming Interfaces:** Application programming interfaces (APIs) are software bridges that enable communication between two applications, especially from third-party service providers. They can be used by enterprises to integrate bots and voice assistants with their live chat software or Web site without code. Other common uses for APIs include machine vision and NLP applications such as urgency detection or sentiment analysis. Some of the common examples are IBM Watson Natural Language Understanding API and Amazon Rekognition API.
- **AI of Things:** Artificial Intelligence of Things (AIoT) embeds AI technology and machine learning capabilities into IoT. It helps in analyzing data to identify patterns, gather operational insights, and detect and fix problems. Generally, AIoT providers offer forecasting services that enable IoT devices to predict when a machine and equipment may need maintenance, helping businesses avoid expensive interruptions. Some of the common examples are AWS Cloud IoT services, Google Cloud IoT Core, and Microsoft Azure IoT.

10.4.1 Key Architectural Components

Delivering AIaaS requires a well-defined architecture supported by underlying infrastructure, tools, and processes like any other service. AIaaS architecture has four basic components: AI infrastructure, AI services, APIs, and AI tools, as shown in Fig. 10.3.

1. **AI Infrastructure:** AI infrastructure supports underlying AI and ML models. Data and compute resources constitute the AI infrastructure required to support ML models.

   - **AI Data:** ML models rely heavily on input data that can be sourced from multiple sources. These models are built to learn from patterns in the existing data. The accuracy of the predictions of these models depends on the volume and diversity of data. This data can come from relational databases, unstructured data (binary objects), stored annotations in NoSQL databases, and a pool of raw data in a data lake [18].
   - **AI Compute:** Advanced ML techniques require combining central processing units (CPUs) and graphic processing units (GPUs). Cloud providers offer clusters of CPU-GPU combination-backed virtual machines (VMs). AIaaS requires computing resources like VMs, serverless computing, and batch processing. These computing methods are used to enhance parallel processing and automate ML tasks.

2. **AI Services:** The AIaaS vendors provide services that are readily available and do not need custom ML models for their consumption. The common AI services offered are

   - **Cognitive Computing:** Cognitive computing services include speech, text analytics, voice translation, and search. Developers access these services as REST [19] endpoints and integrate with the applications with API calls.

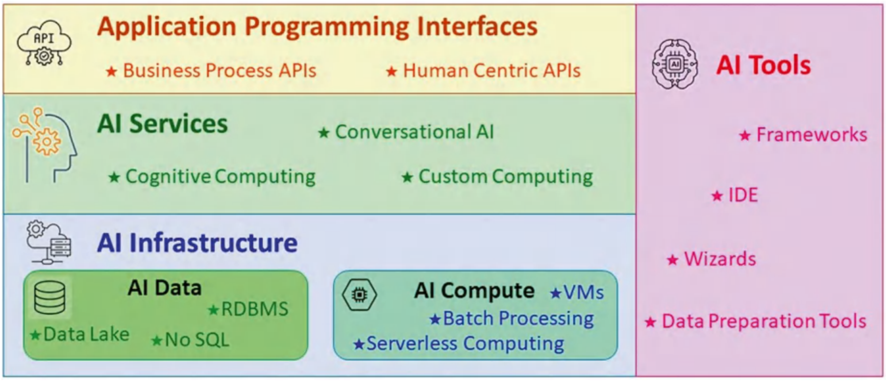


Fig. 10.3 Key architectural components of AIaaS

- **Custom Computing:** Although cognitive computing serves the generic cases, there is a shift toward custom computing, enabling users to experience cognitive computing using custom datasets. The user employs his data to train cognitive services. Some examples of custom computing services are virtual assistants, chatbots, and automated email response services.
- **Conversational AI:** The AIaaS providers are helping developers integrate bots (voice, text) across platforms by leveraging bot services. Using this service, Web and mobile developers can add digital assistants to their applications.

3. **Application Programming Interface (API):** The AIaaS vendors provide readily available APIs. These APIs use the underlying infrastructure. The common APIs offered are:

    - Business Process APIs: These APIs allow the developers to integrate AI services with enterprise applications to complete the business workflows.
    - Human-Centric APIs: These APIs are used by developers to integrate voice, text, or images in the applications developed by them without doing much of the coding.

4. **AI Tools:** These tools promote the usage of VMs, storage, and databases as tools are built in sync with the data and compute platforms.

    - **Wizards:** Wizards reduce the complexity of training ML models. At the backend, these tools in totality offer a multi-tenant development environment.
    - **Integrated Development Environment (IDE):** IDEs and notebooks (browser-based) help in easy ML model testing and management, thereby enabling users to build smart applications with ease.
    - **Data Preparation Tools:** The performance of ML models heavily depends on the quality of data. To ensure the quality of data, the service providers are providing data preparation tools that can perform the extract, transform, load (ETL) jobs. The output of these ETL jobs is then fed into the ML pipeline for training and evaluation purposes.
    - **Frameworks:** The AIaaS providers offer ready-to-go VM templates with frameworks such as TensorFlow, Apache MXNet, and PyTorch. Such VMs train complex neural networks and ML models as VMs are GPU-supported entities.

10.4.2 AI as a Service at the Edge

AIaaS can accelerate the deployment of data-intensive and computation-intensive AI applications with the support of an Edge computing environment. In this distributed computing paradigm, data and its processing are pushed close to the Edge. AI services at the Edge are critical in a wide range of industries, where collecting and analyzing data in real time to make instant local decisions is important. Some industries that can benefit are security and surveillance, transportation, agriculture, medical care, etc. Given below are some of the common AI services that are offered at the Edge by service providers.

1. **Edge Development and IoT Ready Platforms**
    - **Edge Hardware Library:** It allows one to choose the right hardware from a wide variety of choices. These include best-in-class, proven production-ready solutions with assured support and guarantees.
    - **Edge Software/Firmware Stack:** The functionality provided includes device and service management, AI modules, and security features. It helps to develop specific applications faster and with less code.
    - **Custom Software Application Development:** Custom application development requires support for workflow automation, monitoring, diagnostics, application integration, and AI algorithms. These are offered as application development stack services at the Edge.
2. **Communication Services**
    - **Connectivity Options:** Modular interfaces supporting multiple protocols for local and Cloud connectivity. They can be deployed in various environments.
    - **Support for Multiple Protocols:** Software drivers for G5, LTE, GSM, and Wi-Fi for local and Cloud connectivity.
    - **Gateway Compatibility:** Software drivers and modules are available as plug-and-play to support connectivity to customer gateways at the Edge. This enables the developers to connect and communicate with multiple software stacks.
3. **Platform Integration**
    - **IoT Platform Integration:** Seamless integration with Cloud platforms such as Azure IoT Edge, Google IoT, or AWS IoT services. This reduces the effort of creating Edge instances for customers.
    - **Integration of Third-Party API/Applications:** The access to APIs from third-party vendors' applications ensures a seamless experience and business continuity.
    - **Continuous Integration and Continuous Delivery/Deployment of CI/CD Environments:** To support large deployments that are distributed, the user needs to be able to test and deploy changes across the entire system.
4. **Custom Algorithm Development, Training, and Testing**
    - **AI Models:** Customized ML models are developed especially for the Edge that can read patterns from existing data and use learning to make future predictions. These models are designed such that they do not need big data to operate or work. As a result, these are suitable for all types of deployments, i.e., at the Edge or in a central data center.

- **Embedded AI Testing:** AI components require extensive testing and validation before they can be deployed. Companies can use testing services to test their AI setups. This will considerably reduce capital expenditure on robotics, skilled staff, and embedded systems.

10.4.3 Future of AI as a Service

AIaaS will grow exponentially as more and more businesses go for digital transformation with AI. AIaaS provides access to tools and capabilities in a flexible and scalable Cloud environment. It will help businesses harness technologies like natural language processing, machine learning, or deep learning capabilities. Some of the common applications that will drive AIaaS growth are:

1. **Natural, Human-Like Conversational Experiences**

   AI-powered bots use data from the customer's knowledge base to generate accurate, conversational replies. Large enterprises can automate repetitive tasks by buying ready-made and pre-built bots and customizing them to create a unique chatbot persona to match the voice and tone of the brand.

   As chatbots employ natural language processing (NLP) algorithms to identify language patterns from human conversations, they can learn from each interaction and provide answers based on the identified patterns. With AI, the conversations will only get better and provide an enhanced customer experience.
2. **Better Collaboration and Reduced Data Silos**

   AIaaS provides technology that makes it easy to consolidate fragmented data in one place and collaborate more efficiently. This will enable businesses to merge teams and responsibilities, thereby nurturing cross-functional collaboration to increase operational efficiency.
3. **Develop More Computing APIs**

   APIs are built to add functionalities to any kind of application. The developers can identify the features that need to be offered as AIaaS and build the respective APIs. Smaller updates or patches can be made as and when the need arises. Common API services include voice recognition, emotion detection, NLP, language translation, and computer vision.
4. **Build In-house Foundational Capabilities**

   AIaaS calls for systematic coordination between AI service providers and users to prevent sensitive data from being compromised. Businesses using AIaaS are expected to train their employees who work with sensitive systems to keep them cyber-safe. Over the years, it will become essential for all working staff to know, understand, and engage in security practices to collaborate with AIaaS seamlessly.
5. **AI Test Setups**

   Before AI components or applications can be deployed, there is a need to test and validate them extensively. The developers can use AIaaS to test their AI setups. This will considerably reduce capital expenditure on robotics, skilled staff, and embedded systems.

10.5 Price, Performance, and Security Considerations

The adoption of Edge AI offers significant advantages, as discussed in the earlier section of this chapter, but it is not without its challenges. These challenges contribute to complexities in Edge AI implementation, resulting in hurdles for deployment in practical scenarios. Some of the common challenges faced are data management, data privacy, security, cost, performance, and scalability.

In the following paragraphs, we describe the impact that these factors have on Edge AI implementation and suggest some of the possible ways the challenges can be overcome.

1. **Data Management:** Data management needs to be considered at three levels, namely data movement, data storage, and data governance. Each of these impacts the performance and security of the Edge system.

    - It is important to reduce data movement to minimize latency and maximize real-time decision-making with Edge AI systems. To achieve this objective, one of the approaches is distributed intelligence. It employs federated learning, which leverages distributed data across multiple Edge devices to train AI models and enhance data quality, privacy, and diversity. In this case, the data in its original form always stays on the device and is never gathered in one central location.
    - Limited storage capacity on the Edge devices necessitates the employment of data compression techniques to optimize memory usage and accommodate larger data volumes.
    - Data governance requires that the data at the Edge complies with regulations. Data governance requires dedicated frameworks especially designed for this purpose, based on state-of-the-art technology like blockchain to ensure Edge devices are secure, efficient, and reliable.

2. **Security and Privacy:** Security is a primary concern for Edge AI, given the sensitive nature of data processed by Edge devices. Ensuring data privacy and protection against cyber threats is a significant challenge in Edge AI deployment.

    When designing a secure Edge AI solution, organizations must assume that a malicious person could get physical access to a machine. In other words, some unauthorized person could steal the machine and take it off-site to extract sensitive data, maliciously patch the operating system, or even change system drivers. To counter these physical threats, software techniques using hardware security features [4], such as a secure and measured boot, remote attestation, and drive encryption, are used. In addition, the stored data at the Edge is encrypted, and the hard drive is partitioned so that the boot partition is made immutable and cannot be easily rewritten or changed.

    To secure communication between nodes at the Edge and the Cloud, authenticated and encrypted communication between these systems is needed. In addition, anomaly detection methods can identify and mitigate attacks targeting Edge AI systems, thereby ensuring the entire system's security.

3. **Performance:** For Edge AI systems to be effective in making predictions and decisions, there is a need to train ML models over the vast data contributed by multiple Edge devices. There are two approaches to achieving this goal: centralized or federated learning. Each approach has some risks associated with it. Centralized learning has risks of a single point of failure, security, and privacy issues as data must travel to the central location and data providers need to trust the central authority. In the case of federated learning, there is a risk of security with training codes traveling to remote locations, as each distributed site needs to trust the incoming code. Note that in this case, data stays on the Edge with the participants. However, due to latency and multiple training iterations, federated learning has performance concerns. There is a need to balance the security versus performance trade-off. One of the approaches is to use a collaborative federated learning (CFL) [20] solution that combines the advantages of centralized and decentralized machine learning schemes, without compromising security.
4. **Cost:** The lack of standardization, especially in hardware and varying computing capabilities across different Edge devices, makes it difficult for developers to create universally compatible Edge AI applications. This tends to increase the cost of Edge AI applications. There is a need to establish industry-wide hardware standards for Edge devices across diverse environments.

   In addition to hardware cost, there is a cost associated with training and operations of models across widely distributed Edge devices. There is a need to work on different approaches to reduce this cost.
5. **Scalability:** Edge AI systems face scalability challenges in three key areas: computational, data, and system scalability. These are described below:
   - Computational scalability is the ability of a system to handle increasing data volumes without exceeding device capacity without compromising accuracy and responsiveness.
   - Data scalability is the ability of a system to manage large data volumes without performance impact.
   - System scalability is the ability to manage the growing number of Edge devices and users.

   In addition to the above scalability issues, another challenge is to integrate Edge AI into existing systems. This requires ensuring consistent performance across different platforms and technologies. As mentioned earlier, a lack of standardization creates problems with the seamless integration and scalability of Edge AI solutions. Initiatives such as Open Neural Network Exchange (ONNX) [21] offer promising pathways to address these concerns.

Many groups are working on solving these issues, but still a lot of work needs to be done before Edge AI systems can be deployed at scale by enterprises.

10.6 Emerging Trends at the Edge

In this section, we will look at some of the emerging trends that will shape the future of Edge AI. The technological advancements in the design and development of (i) low-cost Edge devices; (ii) energy-efficient hardware; (iii) compact machine learning models; and (iv) robust, easy-to-use, and easy-to-deploy software and tools will fuel the rapid adoption of Edge AI and related applications. This will result in more developers, applications, and use cases leading to more adopters of Edge AI in all verticals, especially industrial IoT, healthcare/wellness, consumer electronics, and smart everything.

1. **Hybrid Implementation of Edge AI:** More powerful and energy-efficient Edge devices will be used for data ingest, preparation, model development, and training. A hybrid approach may emerge where external Cloud computing platforms will be used for model development. These models will then automatically or manually be deployed to the Edge AI devices for inference, decision-making, and implementation.

   This type of hybrid implementation of Edge and Cloud may provide a more seamless architecture for model maintenance, management, and continuous learning. This approach will result in a significant reduction in the amount of data that must be sent to the Cloud offering a balance between scalability and low-latency processing. For example, Microsoft Azure provides seamless integration between Cloud services and Edge, facilitating the deployment of AI applications across distributed Edge networks [22].
2. **Development of Energy-Efficient Edge Devices:** To reduce the cost and for reasons of sustainability, there is a need to develop energy-efficient Edge devices. A lot of work is being done on the development of low-power, high-performance computing, such as neuromorphic computing [23] and data-efficient AI. Neuromorphic chips consist of numerous artificial neurons and synapses, mirroring the behavior of brain spikes [24]. These chips offer significant advantages for scaling Edge AI applications. They consume less power, offer faster processing speeds, and equip Edge AI systems with human-like reasoning capabilities. It enables the development of AI-enabled Edge devices capable of real-time learning and adaptation. These are highly beneficial for various applications like obstacle avoidance and robust acoustic perception.

   BrainChip, IBM, and Intel are some of the key players in the development of neuromorphic chips [25]. BrainChip's Akida™ neural processor supports on-device learning, allowing for personalization and customization without Cloud connectivity. With the addition of temporal event-based neural nets (TENNs), Akida is capable of processing complex time-series data applications, enhancing efficiency while maintaining accuracy. IBM's TrueNorth chip embodies neuromorphic principles, exhibiting remarkable energy efficiency while delivering cognitive capabilities suitable for Edge AI applications. In the future, the devices will become smarter with on-device learning capabilities at the individual and network levels with the help of approaches such as collaborated federated learning.

3. **Edge AI for Automation:** The industrial Internet of Things (IIoT) is similar to IoT but focused on industrial automation devices. IIoT devices often perform critical tasks and handle high amounts of data used to make real-time decisions in a demanding, high-security environment that requires reliability. IIoT is the backbone for smart manufacturing and Industry 4.0, making machines and processes better [26].

   By using AI and ML in an IIoT environment, the devices can be trained based on the massive amounts of available data in the industrial environment. ML algorithms running on these devices are used for predicting maintenance, production tracking, and even energy optimization that reduce costs and increase productivity. While it is still evolving, AIoT and Edge AI are the future of industrial automation.

4. **Automating the Edge Operations with AI Assistants:** Managing large Edge deployments is a complex task and takes a lot of effort. Research is being done to explore how GenAI and co-pilot-based Edge automation development tools can significantly reduce the development process complexity. This automation of code generation can help meet the stringent requirements of Edge operations workloads, such as low latency and high security.

   Next-generation Edge platforms will include AI-based policy-driven deployments. These policies will include dynamic workload migration and resource optimization algorithms to ensure seamless workload distribution and efficient task execution.

5. **Generative AI at the Edge**: Generative AI (GenAI) can create new content, images, and videos that are at the near-human creativity level. The integration of generative AI at the Edge can deliver richer and more immersive user experiences while preserving data privacy and minimizing reliance on centralized Cloud infrastructure.

   This requires large language models' (LLMs) convergence with Edge devices. To deploy LLMs on the Edge devices, their model sizes need to be reduced. There is a need to work on techniques such as pruning, quantization, and distillation to produce smaller models while maintaining high performance. Many companies are working in this area; TinyChat, the NVidia IGX Orin Developer Kit, and Qualcomm's Stable Diffusion are all examples of serious efforts being made to bring generative AI to the Edge [27].

   Advancements in computer vision at the Edge have enabled real-time analysis of visual data, enabling applications to have contextual awareness and situational understanding. This has enabled the development of applications for smart surveillance and industrial automation.

   GenAI at the Edge has enabled natural interfaces such as voice commands, gestures, and facial expressions. These interfaces allow users to interact with devices intuitively and efficiently without the need for traditional input methods like keyboards or touchscreens. The ability of Edge AI systems to accurately interpret and respond to various natural inputs enhances the user experience.

6. **Micro AI for Edge:** The current LLMs are too computationally demanding to be usable for inferencing at the Edge devices. Efforts are being made to develop lightweight, hyper-efficient AI models designed specifically for resource-constrained Edge devices. These ML models will yield smaller, more sophisticated, multi-modal models, including transformers, without sacrificing accuracy.

   Efforts are underway to develop lightweight models in the following categories:

   - **Domain-Specific and Task-Focused Models:** These models are trained on machine-driven translation, text summarization, or question-answering tasks. Similarly, domain-specific Models are trained on data from respective domains like healthcare, finance, legal documents, etc. They have a smaller footprint but offer a deeper understanding of the targeted domain so can deliver more accurate and relevant outputs.
   - **Smaller Models:** These models are advantageous for deployment on Edge devices with limited resources or for minimizing computational overhead. This allows resource-sensitive model development for the Edge. TensorFlow Lite is an example of this approach.

7. **Data-Efficient AI:** This enables AI algorithms to operate effectively with minimal data requirements. Ongoing research on data-efficient AI explores many techniques, ranging from augmenting and using pre-trained models with domain knowledge to paradigms that engage humans in the data-labeling processes. These methods eliminate the need for extensive data collection, thereby reducing computing power on the Edge devices. Other data-efficient techniques, such as model pruning, enable models to fit on the Edge devices without compromising performance. Moreover, emerging approaches like one-shot learning and few-shot learning inherently enable models to learn from minimal data samples [27].
8. **Incremental Learning:** In the case of incremental learning, machine models process new information over time, maintaining and building upon previous knowledge. Incremental learning has the potential to become one of the popular methods of training ML models due to the advantages offered by it [28]. The main driving forces for the adoption of this methodology are:

   - **Efficient Use of Resources:** It requires storing less data at a time, which leads to a significant reduction in memory space at the Edge devices. For example, a fraud detection system in a bank can update its model with each transaction, rather than storing all transactions to process them later.
   - **Real-Time Adaptation:** Using incremental learning methodology, the models can adapt to changes in real time. For example, a news recommendation system can learn a user's changing preferences over time and recommend articles based on their most recent interests.
   - **Efficient Learning:** Breaking a task into smaller parts can enhance the machine learning model's ability to learn new tasks quickly and effectively.

- **Learning from Nonstationary Data:** Incremental learning models are highly valuable where data evolves rapidly. For example, a weather prediction model can continuously adapt its forecasts based on the most recent climate data.

 Many groups are working on adapting incremental learning methodology for training deep learning models, especially recurrent neural networks (RNNs) and certain types of convolutional neural networks (CNNs). These models learn from new data by updating their weights incrementally, allowing them to handle streaming data or environments that change over time.

 Incremental learning has its challenges, such as models forgetting old information, difficulty in handling concept drift, and the risk of overfitting. While implementing incremental learning, these concerns should be addressed.

9. **Distributed AI:** Distributed AI leverages decentralized learning and collaborative intelligence to enhance Edge computing capabilities. NVIDIA's Federated Learning Toolkit [29] empowers Edge devices to collaboratively train AI models without sharing raw data, preserving data privacy while improving model accuracy. This distributed approach enables Edge AI systems to adapt dynamically to changing environments and diverse user needs, paving the way for more resilient and responsive Edge applications.

   Federated learning and swarm learning are extensively used for training Edge devices. While federated learning allows Edge devices to collaboratively train a shared machine learning model without sharing raw data, swarm learning fosters decentralized and self-organizing AI systems. These decentralized learning approaches promise scalable, efficient, and privacy-preserving solutions for future Edge AI deployments.

10. **Digital Twins:** A digital twin is a digital representation of a physical object, person, or process that can be used to simulate its behavior to understand better how it works in real life [30]. It spans the object's life cycle and is updated from real-time data, machine learning, and reasoning to help make decisions. Digital twins have the potential to deliver more agile and resilient operations. Many large enterprises are already exploring and investing in digital twins. This investment will grow in the future with advancements in the underlying IoT and ML technologies. There are different types of digital twins depending on the area of application. Some of these are described below:
    - **Product Twins:** A product twin is a representation of a product and includes products at various life cycle stages. It gets live, real-time data on a product to represent its current state.
    - **System Twins:** A system twin represents how different products come together to form an entire functioning system. System twins provide visibility regarding the interaction of different products and may suggest performance enhancements.
    - **Process Twins:** A process twin reveals how systems work together to create an entire production facility. Process twins can help determine the precise timing of events that ultimately influence overall productivity and effectiveness.

- **Production Plant Twins:** A production plant twin represents an entire manufacturing facility. It may consist of many different systems and process twins that work together to form an entire functioning production plant. Production plant twins provide visibility regarding the interaction of different systems and processes, which can be used to suggest improvements in performance and cost reduction.
- **Infrastructure Twins:** Infrastructure twins are similar to production plan twins as these represent complete physical infrastructure such as a highway, a building, or even a stadium.

Digital twins are being used by enterprises across the world for implementing large projects because of the potential advantages. The use of digital twins enables more effective research and design of products, with an abundance of data created about likely performance outcomes. It helps companies make required product refinements before starting production, thereby reducing cost and improving quality. Even after a new product has gone into production, digital twins can help mirror and monitor production systems to achieve and maintain peak efficiency throughout the entire manufacturing process. Digital twins enable what-if simulation of a complex system, such as an aircraft or a large ship, during operation without risking the actual system.

Some of the use cases where digital twins are beneficial are:

- **Physically large projects:** Buildings, bridges, and other complex structures.
- **Mechanically complex projects:** Jet turbines, automobiles, and aircraft.
- **Power equipment:** It covers both power generation and transmission.
- **Healthcare services:** To track various health indicators and generate key insights. Efforts are underway to mimic digital twins of a patient for personal healthcare [31].

10.7 Summary

The demand for AI applications at the Edge is driven by requirements of real-time responses, minimal latency, less bandwidth consumption, and privacy—best represented by BLERP factors. To deliver these solutions, there is a need to have reliable Edge computing infrastructure supported by AI-capable Edge devices that can execute AI algorithms and models. As the Edge computing environment is inherently distributed, we also discussed the concepts of distributed artificial intelligence (DAI), such as distributed problem solving, multi-agent systems, and Swarm Intelligence. We also described how DAI takes advantage of large-scale computation and spatial distribution of computing resources.

This chapter also described the concept of AI as a Service (AIaaS) offered by third-party vendors that allows businesses to incorporate AI-powered tools and capabilities in their systems. We explained the reasons for the rapid adoption of AIaaS in sectors such as healthcare, manufacturing, transportation, retail, and entertainment. We reviewed major challenges prohibiting the large-scale adoption of Edge AI, such as price, performance, and security. We concluded the chapter with new and upcoming developments such as energy-efficient Edge devices, data-efficient micro AI, automation, digital twins, and generative AI at the Edge.

10.8 Points to Ponder

1. What are the main drivers for the Edge AI? Which industries have most benefitted from Edge AI? List some use cases from these industries.
2. Explain how smart traffic control systems can use multi-agent reinforcement learning.
3. What are the benefits and challenges of Artificial Intelligence as a Service (AIaaS)?
4. What are the limitations of Swarm Intelligence?
5. Suggest some methods to improve the performance of Edge AI.

10.9 Answers

1. *What are the main drivers for the Edge AI? Which industries have most benefitted from Edge AI? List some use cases from these industries.*

   The main drivers for Edge AI's exponential growth are collectively referred to as BLERP, which stands for Bandwidth, Latency, Economics, Reliability, and Privacy.

   Healthcare, Utilities, Retail, Manufacturing, and Automobile are some of the industries that have most benefited from Edge AI. These are also the main growth engines for Edge AI adoption.

   Use cases for some of the industries listed above are given below:

   Healthcare:

   1. Patient Monitoring: Using local AI models and wearable devices, heart rate, blood pressure, glucose levels, and breathing can be monitored, and patient condition can be assessed for proactive alerts.
   2. Maintaining Privacy: Edge AI patient monitoring processes data locally to maintain privacy while enabling timely notifications.

   Utilities:

   1. Edge AI energy management: Enterprises can manage their energy consumption better by using Edge computing and smart grids.
   2. Remote asset monitoring: In the oil and gas industry, assets can be monitored remotely using Edge AI capabilities of real-time analytics by processing data much closer to the assets.

   Retail:

   1. Edge AI at checkouts: Edge AI cashier-less services such as Amazon Go automatically counts items placed into a shopper's bag without a separate checkout process.
   2. Speech recognition: Speech recognition algorithms are used on local devices to provide a better customer experience, such as Apple Siri or Amazon Alexa.

   Manufacturing:

   1. Robotic arms: Robot arms gradually learn better ways to grasp packages on the production lines, thereby improving efficiency and productivity.
   2. Fault detection: Edge AI helps manufacturers analyze and detect changes in the production lines before a failure occurs.

   Automobiles:

   1. Self-driving cars: Edge AI is used in autonomous vehicles to improve safety, enhance efficiency, reduce accidents, and decrease traffic congestion.
   2. Autonomous truck convoys: Autonomous vehicles can facilitate automated platooning of truck convoys, removing drivers from all trucks except the one at the front.

2. *Explain how smart traffic control systems can use multi-agent reinforcement learning.*

   The problem can be formulated as a discounted cost Markov Decision Process (MDP) [32]. Each traffic signal junction can be modeled as an independent agent. The agent decides the signal duration of its phases in a round-robin manner using multi-agent Q-learning with ε-greedy or double Q-learning method-based exploration strategies [33]. This eliminates the overestimation problem during exploration. Mean field approximation can be used to model interactions among the agents. It helps in making agents learn a better cooperative strategy. The agent can update its Q-factors based on the cost feedback signal received from its neighboring agents. To improve the stability and robustness of the learning process, a reward allocation mechanism and a local state-sharing method should be introduced. This approach is effective in minimizing the average delay of vehicles in the network.
3. *What are the benefits and challenges of Artificial Intelligence as a Service (AIaaS)?*

   AIaaS is an off-the-shelf AI service offering that helps organizations implement AI tools and technologies without the complexities of developing AI solutions in-house. It offers the following benefits to the organization:

   - **Advanced infrastructure at a fraction of the cost:** Businesses use plug-and-play AI functionality to keep up with evolving business needs. They need to pay for the actual usage and AI functionality without making an upfront investment.
   - **Quick deployment:** AIaaS is the quickest way to introduce AI within the organization as it is very easy to install and setup.
   - **Scalability:** AIaaS allows businesses of all sizes to deploy and scale AI based on business needs.

- **Low- to no-code skills required:** AIaaS can be used even if a company lacks an in-house AI developer or programmer.
- **Usability.** AIaaS is a ready-out-of-the-box solution that can be deployed by even non-technical people.
- **Boost team productivity and efficiency:** AIaaS allows the use of Generative AI, and sentiment analysis to streamline workflows to improve team productivity without additional headcount.
- **Enhance the customer experience:** AIaaS allows businesses to implement AI faster to deliver personalized, conversational support, thereby increasing customer experience.

AIaaS implementation is not without its challenges. Some of the common challenges faced while implementing AIaaS are given below:

- **Concern about data privacy and security:** AIaaS vendors will have access to business data so that AI and ML algorithms can work as per the business requirements. This increases the risk of exposing data during transit or stored on servers in case of a security breach.
- **Risk of biased or unreliable data:** The results and decision-making could be inaccurate if the AIaaS providers train an AI model on unreliable, biased, or unethical data.
- **Reduced transparency:** Since only inputs and outputs are known, there is no understanding of the inner workings of AI systems, like which algorithms are being used, whether the algorithms are updated, and which versions apply to which data.
- **Compliance with regulatory standards:** Regulations governing the use of AI may vary across industries or locations. The AIaaS vendor shall meet the compliance standards as required. The vendor should be transparent and proactive in sharing any compliance changes.
- **Long-term costs:** AIaaS costs can quickly spiral as is the case with any service offering. To control the cost there may be a need to hire and train staff with AI specific experience.

4. *What are the limitations of Swarm Intelligence?*

   Swarm Intelligence has generated the interest of AI scientists who are working to improve its performance. The major challenges faced by Swarm Intelligence are:

   - **Behavior Coordination:** It is very difficult to predict the behavior of the group. Coordination of many agents in a group is complex and may require sophisticated communication and control mechanisms.
   - **Limited Individual Knowledge and Capabilities:** Individual agents may have limited cognitive abilities. It is important to know the functioning of an individual agent before the functions of the group can be understood.
   - **Lack of Predictability:** The group behavior of the agents can change even with a small change in the simplest of the rules. This makes it difficult to understand how the system will behave in different scenarios.

- **Action Scalability:** The action of the group is driven by a stochastic process and therefore the action of an agent looks like a noise. Managing many agents can lead to computational and organizational complexities.

5. *Suggest some methods to improve the performance of Edge AI.*

   There are multiple methods to improve the performance of Edge AI systems and applications. They can be broadly classified under the following categories.

   - **Hardware Optimization:** Selecting the right-sized compute engines to meet or exceed the required performance levels. For an AI application, these compute engines must perform the functions of the entire cycle. For this purpose, a dedicated AI accelerator may be required. Performance scaling can be achieved by using AI accelerators in module format or with additional AI accelerator chips. Application Specific Integrated Circuit (ASIC) that are designed for specific purposes can also be used. ASICs are more energy efficient and smaller in size. There are different types of ASICs available, for example, Tensor Processing Units, Vision Processing Units, and Neural Processing Units, etc.
   - **Model Optimization Methods:** There are three major techniques for optimizing AI models:

     1. **Quantization:** It reduces the precision of weights, parameters, biases, and activations so that they occupy less memory. This reduces the size of the model so that it can fit easily on memory-constrained Edge devices. For example, quantizing to 8 bits from 32 bits reduces the model size by 4×.
     2. **Pruning:** It is a process of identifying and eliminating connections or neurons (parameters) that are redundant or unimportant. It produces models having a smaller size for inference. With reduced size, the model becomes both memory and energy-efficient, thereby faster at inference with minimal loss while maintaining similar accuracy as before.
     3. **Knowledge Distillation:** It is the process of training a smaller model with the help of a large, trained AI model. This results in model size reduction often many times smaller than the original model while maintaining similar accuracy.

   - **Federated Learning:** It is a distributed collaborative learning method that allows different Edge devices with different datasets to work together to train a global model. It is useful to meet the needs of modern IoT-based applications from a privacy protection perspective. It has the potential to be the basis for next-generation artificial intelligence learning.
   - **Hyperparameter Tuning:** Hyperparameters are used to get optimal performance of the model. In complex models, many hyperparameters are required to be tuned, whereas in a lightweight model, each parameter must be tuned strictly to a particular range.
   - **Energy Efficiency Optimization Techniques:** Edge devices are battery-operated devices that store the power in a battery. The computing capacity of

a device is defined as the maximum number of inferences per watt. There is a need to optimize this computational capacity. There are many techniques to optimize energy consumption per inference. Some of the techniques employed are Neural Architecture Search (NAS) and Hardware-Aware NAS, Algorithm-Accelerator Co-Design Method, Memory Optimization, and Energy-Efficient Communication Protocols, etc.

References

1. Dilley, J., Maggs, B., Parikh, J., Prokop, H., Sitaraman, R., & Weihl, B. (2002, November). Globally distributed content delivery. *IEEE Internet Computing, 6*(5), 50.
2. Situnayake, D., & Plunkett, J. (2023, January). *AI at the Edge: Solving real world problems with embedded machine learning*. O'Reilly Media, Inc.
3. Li, X., & Zhang, Z. J. (2019). Research and analysis for real-time streaming big data based on controllable clustering and edge computing algorithm. *IEEE Access, 7*, 171621–171632.
4. Sehgal, N. K., Bhatt, P. C. P., & Acken, J. M. (2023). *Cloud computing with security and scalability. Concepts and practices*. Springer.
5. Chaib-Draa, B., Moulin, R., Mandiau, P., & Millot, P. (1992). Trends in distributed artificial intelligence. *Artificial Intelligence Review, 6*, 35–66.
6. Cammarata, S., & McArthur, D. (1983). Strategies of cooperation in distributed problem solving. In *Proceedings of international joint conference of artificial intelligence*, pp. 767–770.
7. Saxena, M. K., Biswas, K. K., & Bhatt, P. C. P. (1990). DISPROS—A distributed Blackboard Architecture. In *Proceedings of IEA/AIE*.
8. Ponomarev, S., & Voronkov, A. E. (2017). *Multi-agent systems and decentralized artificial superintelligence*. https://arxiv.org/abs/1702.08529
9. Dumke, R., Mencke, S., & Wille, C. (2009). *Quality assurance of agent-based and self-managed systems*. CRC Press.
10. Durfee, E. H. (2001). Distributed problem solving and planning. In M. Luck, V. Mařík, O. Štěpánková, & R. Trappl (Eds.), *Multi-agent systems and applications. ACAI* (Lecture notes in computer science) (Vol. 2086). Springer.
11. Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., & Wang, J. (2018). *Mean field multi-agent reinforcement learning*. https://ar5iv.labs.arxiv.org/html/1802.05438
12. Bloom, H. (1995). *The Lucifer principle: A scientific expedition into the forces of history*. Atlantic Monthly Press.
13. Heylighen, F. (2016). Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research, 38*, 4–13.
14. Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948). IEEE.
15. Colorni, A., Dorigo, M., & Maniezzo, V. (1991). Distributed optimization by ant colonies. In F. Vaerla & P. Bourgine (Eds.), *Proceedings of the European conference on artificial life* (p. 134). Elsevier Publishing.
16. Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization.* Technical Report 06. Erciyes University
17. Koelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research, 4*, 237–285.
18. *What is a data lake?* AWS Cloud Services. https://aws.amazon.com/what-is/data-lake/
19. *What is REST API?* IBM. https://www.ibm.com/topics/rest-apis

20. Saha, P. P., Sehgal, N. K., & Faezipour M. (2024). *Collaborative federated learning cloud based system*. The 2024 world congress in computer science, computer engineering, & applied computing (CSCE'24).
21. Open Neural Network Exchange. https://en.wikipedia.org/wiki/Open_Neural_Network_Exchange
22. Edula, V. (2023, July). *Introduction to Azure Edge computing and its application*. https://www.geeksforgeeks.org/introduction-to-azure-edge-computing-and-its-application/
23. Schuman, C. D., Kulkarni, S. R., Parsa, M., et al. (2022). Opportunities for neuromorphic computing algorithms and applications. *Nature Computing Science Newsletter, 2*, 10–19. https://doi.org/10.1038/s43588-021-00184-y
24. *Neuromorphic computing and engineering, next wave of AI capabilities*. https://www.intel.com/content/www/us/en/research/neuromorphic-computing.html
25. Neuromorphic chip market size & share analysis –Growth trends & forecasts (2024–2029). *Mordor Intelligence* (2024). https://www.mordorintelligence.com/industry-reports/neuromorphic-chip-market
26. Soori, M., Arezoo, B., & Dastres, R. (2023). Internet of Things for smart factories in Industry 4.0. A review. *Internet of Things and Cyber-Physical Systems, 3*, 192–204. ISSN 2667-3452. https://www.sciencedirect.com/science/article/pii/S2667345223000275
27. *2024 state of Edge AI report*. https://www.wevolver.com/article/2024-state-of-edge-ai-report/the-future-of-edge-ai
28. Rao, S. (2023, September 16). Machine learning, illustrated: incremental learning. *Towards Data Science*. https://towardsdatascience.com/machine-learning-illustrated-incremental-machine-learning-4d73747dc60c
29. Wen, Y., Li, W., Roth, H., & Dogra, P. (2019). *Federated learning powered by NVIDIA Clara.* https://developer.nvidia.com/blog/federated-learning-clara/.
30. McKinsey & Company. (2023, July). *What is digital-twin technology?* https://www.mckinsey.com/featured-insights/mckinsey-explainers/what-is-digital-twin-technology
31. Katsoulakis, E., Wang, Q., Wu, H., et al. (2024). Digital twins for health: A scoping review. *npj Digital Medicine, 7*, Article 77. https://doi.org/10.1038/s41746-024-01073-0
32. Prabuchandran, K. J., Hemanth Kumar, A. N., & Bhatnagar, S. (2014). Multi-agent reinforcement learning for traffic signal control. In *17th international IEEE conference on intelligent transportation systems (ITSC)*, Qingdao, China, pp. 2529–2534.
33. Wang, X., Ke, L., Qiao, Z., & Chai, X. (2021, January). Large-scale traffic signal control using a novel multiagent reinforcement learning. *IEEE Transactions on Cybernetics, 51*(1), 174–187.

Index

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2025

N. K. Sehgal et al., *AI on the Edge with Security*,
https://doi.org/10.1007/978-3-031-78272-5