FUNDAMENTALS OF AI

Introduction to Artificial Intelligence (AI)

- **Definition:** AI is a branch of computer science that creates machines or programs capable of performing tasks that require human-like intelligence.
- **Key Points:**
    - It mimics human mental processes like reasoning, learning, and problem-solving.
    - AI doesn't strictly follow how humans think biologically; it focuses on achieving similar outcomes.
    - AI enables machines to act intelligently by processing data and making decisions.

What is AI?

1. AI involves creating software, robots, or systems that think like humans.
2. It is about solving problems that humans usually handle better.
3. AI systems store and process information efficiently, like associating details (e.g., identifying Amitabh Bachchan based on hints).
4. Definitions by Experts:
    - John McCarthy: *"AI is the science and engineering of making intelligent machines."*
    - Elaine Rich: *"AI studies programs at which people are currently better."*

Machine Learning vs. AI

- **AI:** The broader field focused on mimicking human intelligence.
- **Machine Learning:** A subset of AI that helps machines learn from data to make decisions and improve over time.
- **Examples:**
    - Virtual assistants (Siri, Alexa)
    - Self-driving cars
    - Recommendation systems

Applications of AI

1. **Daily Life Examples:**
    - Chatbots, navigation apps, facial recognition, and e-commerce recommendations.
2. **Industry-Specific Uses:**
    - **Healthcare:** Better diagnoses, personalized treatments.
    - **Cybersecurity:** Spam detection and data protection.
    - **Transportation:** Autonomous vehicles.

Advantages of AI

- **Efficiency:** Handles repetitive tasks quickly.
- **Accuracy:** Reduces errors in processes like medical diagnoses.
- **Personalization:** Improves customer experiences (e.g., recommendations).

Disadvantages of AI

- **Privacy Concerns:** Risks related to personal data usage.
- **Job Losses:** Automation can replace human jobs.
- **Bias:** Incorrect assumptions may lead to discrimination.
- **Dependence:** Over-reliance on AI can reduce human involvement.

Branches of AI

1. **Logical AI:** Uses logic to determine actions based on facts and goals.
2. **Search:** Explores possibilities, like choosing the best move in chess.
3. **Pattern Recognition:** Identifies patterns, such as faces in images.
4. **Common Sense Reasoning:** Tries to mimic human reasoning for everyday situations.
5. **Planning:** Develops strategies to achieve specific goals.
6. **Heuristics:** Uses shortcuts to solve problems faster.
7. **Genetic Programming:** Creates solutions by simulating evolution.

Knowledge Representation in AI

- Represents facts in structured ways to solve problems.
- Example: Answering questions by matching patterns with templates.

Responsible AI Use

- Combine human creativity with AI for the best results.
- Ensure fairness, privacy, and transparency in AI applications.
- Use explainable AI to build trust with users.

Two Classes of AI Problems

1. **First Class Problems**:
    - Use **search techniques** because no direct solution exists.
    - Require **knowledge about objects** and use **abstraction** to simplify the process.
    - Avoids data overload by pruning irrelevant parts.
    - Aims for **real-time solutions**.
2. **Second Class Problems**:
    - Focus on **simulating human behavior** to solve complex tasks.
    - Example: Simulating **paranoid behavior** (PARRY program).
    - Helps test human psychological theories and mimic reasoning, like answering questions from news articles.

Problem Solving and Search

1. **To build a system:**
    - **Define the problem** with clear goals and acceptable solutions.
    - **Analyze the problem** to focus on critical features.
    - **Use background knowledge** required to solve it.
    - Select **the best technique** for solving the problem.
2. **Problem Characteristics**:
    - A problem has **goals**, **objects**, and **actions** to achieve goals.
3. **Problem Space**:
    - Includes all possible states of the problem.
    - A **solution** is a series of actions that reach the goal.
    - A **search** explores the problem space to find solutions.

AI Search Types

1. **Uninformed Search** (Blind Search):
    - Don't use extra knowledge about the problem.
    - Examples:
        - **Breadth-First Search (BFS)**: Explore all nodes at the current level first.
        - **Depth-First Search (DFS)**: Go as deep as possible before backtracking.
        - **Uniform Cost Search**: Focuses on the least-cost solution.
2. **Informed Search** (Heuristic Search):
    - Uses extra knowledge to guide the search effectively.
    - Examples:
        - **Greedy Best-First Search**: Chooses the closest option to the goal.
        - **A***: Balances the cost of reaching a node and the estimated cost to the goal.

Production Systems

- **Structure for Search**:
    - **Rules**: Conditions and actions to follow.
    - **Database**: Stores facts during the process.
    - **Control Strategy**: Decides which rule to apply next.
    - **Rule Firing Module**: Executes the selected rule.

Heuristic Search

- **Heuristics** are problem-specific shortcuts that guide the search.
- A heuristic function estimates the best next move.
- Example: In finding a route between cities:
    - **State**: Current city.
    - **Operator**: Roads between cities.
    - **Heuristic**: Use straight-line distance to guide the search.

Search Algorithm Characteristics

1. **Completeness**: Finds a solution if one exists.
2. **Optimality**: Ensures the best solution.

3. **Time Complexity**: Time required to find a solution.
4. **Space Complexity**: Memory needed during the search.

General Problem Solvers (GPS)

- Solve symbolic problems like puzzles or chess.
- Example: **Towers of Hanoi**.
- **Limitations**: Can't handle complex, real-world problems.

Types of Search Problems

- **Special-Purpose Methods**: Designed for specific problems.
- **General-Purpose Methods**: Apply to various problems (e.g., **means-end analysis**, which reduces differences step by step).

Example Problems

1. **8-Puzzle Problem**: Slide tiles to arrange them in a goal state.
2. **Water Jug Problem**: Measure specific water amounts using two jugs.

Use of AI in IoT Systems to Solve Issues:
There’s no specific mention of how AI can be applied in IoT to address challenges such as:

- Real-time data processing.
- Resource optimization.
- Security and anomaly detection.
- Predictive maintenance and automation.

AI in IoT Systems:

- **Definition:** AI in IoT refers to the integration of machine learning, neural networks, and other AI techniques with IoT devices and systems.
- **Applications:**
  - *Predictive Analytics:* Analyzing IoT sensor data to predict failures or trends.

        - *Edge AI:* Performing computations on IoT devices rather than sending all data to the cloud.
        - *Security:* Identifying unusual activity to prevent cyberattacks.
        - *Automation:* Smart home systems, autonomous vehicles, and industrial automation.
    - **Challenges Solved:**
        - Managing massive IoT-generated data.
        - Enhancing decision-making in real-time.
        - Improving operational efficiency.

IoT COMMUNICATION PROTOCOLS

Connecting Smart Objects

Smart objects in the Internet of Things (IoT) rely on communication technologies to share data. These technologies can be wired (e.g., Ethernet) or wireless (e.g., Wi-Fi, Bluetooth). Wireless communication is preferred for IoT because:

- It simplifies setup (no cables).
- It allows devices to move while staying connected.

Key Communication Criteria

1. **Range**: The maximum distance between devices.
2. **Frequency Bands**: The part of the radio spectrum used for communication.
3. **Power Consumption**: How much energy devices need to communicate.
4. **Topology**: How devices are arranged in a network.
5. **Constrained Devices**: Devices with limited power, memory, or processing ability.
6. **Constrained-Node Networks**: Networks designed for low-power and resource-limited devices.

Communication Range

- **Short Range** (up to 10s of meters): Bluetooth, Visible Light Communications (VLC).
- **Medium Range** (10s to 100s of meters): Wi-Fi, ZigBee, LoRa.
- **Long Range** (more than 1 mile/1.6 km): Cellular (4G, NB-IoT), outdoor Wi-Fi.

NB-IoT (Narrow Band IoT)

NB-IoT is a low-power technology for IoT devices that ensures:

- Long battery life (10+ years).
- Improved coverage in hard-to-reach areas.
- Efficient use of spectrum for large-scale deployments.

Frequency Bands

- **Licensed Bands**: Require service subscriptions (e.g., cellular networks).

- **Unlicensed Bands**: Freely available but less protected (e.g., 2.4 GHz for Wi-Fi, ZigBee).

ISM Bands in India: Allow 865-867 MHz for IoT.

Power Consumption

1. **Powered Nodes**: Need a constant power source but are less mobile.
2. **Battery-Powered Nodes**: More flexible but require low-power communication protocols (e.g., Bluetooth, ZigBee).

Topologies

- **Star**: Central node connects all devices (common in Wi-Fi).
- **Mesh**: Devices communicate with each other, extending the network (used in ZigBee).
- **Peer-to-Peer**: Devices communicate directly without a central hub.

Constrained Devices

Devices are categorized based on resource limitations:

1. **Class 0**: Minimal resources (e.g., buttons).
2. **Class 1**: Slightly more capable (e.g., sensors).
3. **Class 2**: Can run full protocols (e.g., smart meters).

Constrained-Node Networks: Require efficient communication technologies to handle low power, data rate, and latency needs.

IoT Access Technologies

Different technologies are used for IoT based on their use cases:

1. **Bluetooth**: Short-range, low power.
2. **IEEE 802.15.4**: Low-cost, low-power networks like ZigBee.
3. **LoRaWAN**: Long-range, low power.
4. **Wi-Fi (IEEE 802.11)**: Higher data rates but more power consumption.

ZigBee

- Uses IEEE 802.15.4 for low-power, low-bandwidth communication.
- Common in smart homes and industrial automation.
- Employs AES encryption for secure data transfer.
- Utilizes a mesh network for better coverage.

IEEE 802.15.4 PHY and MAC Layers

- **PHY Layer**: Handles data transmission, offers various frequency bands (e.g., 2.4 GHz).
- **MAC Layer**: Manages how devices share channels, security, and network setup.

IEEE 802.15.4g and 802.15.4e

- **802.15.4g**: Designed for outdoor use (e.g., smart grids), offers better error correction and longer ranges.
- **802.15.4e**: Improves reliability and security, making it suitable for industrial use.

Wi-SUN

- Based on 802.15.4g, optimized for smart utility networks.
- Supports large payloads (up to 2047 bytes) and multiple data rates.

IEEE 802.11ah (Wi-Fi HaLow)

- A low-power Wi-Fi variant for IoT.
- Offers extended range but supports fewer devices.
- Uses a star topology and can relay signals for better coverage.

IoT Technologies

1. **Bluetooth**: Low power, short-range (IEEE 802.15.1).
2. **IEEE 802.15.4**: Standards for low-power wireless networks used in:
    - **ZigBee**: Automation and sensors.
    - **6LoWPAN**: IPv6 over low-power networks.
    - **WirelessHART** and other stacks.

ZigBee and ZigBee IP

- **ZigBee**:
  - Focuses on low bandwidth and low power.
  - Applications: Smart homes, industrial automation, and smart meters.
- **ZigBee IP**:
  - Integrates with open standards (e.g., IPv6, 6LoWPAN).
  - Supports protocols like ICMPv6 and RPL.

LoRa and LoRaWAN

- **LoRa**: PHY layer using **chirp spread spectrum** modulation for robust communication.
- **LoRaWAN**: Protocol stack with:
  - **Classes**:
    - A: Battery-efficient.
    - B: Extra receive windows.
    - C: Continuous listening for powered nodes.
  - **Gateways**: Central hubs in a star topology that relay data between devices and networks.

Security

- **Encryption**: Protocols like ZigBee and LoRaWAN use AES-128 for secure communication.
- **Integrity**: Validates data using Message Integrity Code (MIC)

6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks)

- **Purpose**: Allows low-power devices (like sensors) to use IPv6.
- **How it works**: Adds an **adaptation layer** between the network and link layers to compress headers and enable IPv6 on small devices.
- **Key Features**:
  - Reduces transmission overhead.
  - Supports fragmentation for large IPv6 packets.
  - Enables multi-hop communication over constrained networks.
- **Why it's important**: Makes it possible for IoT devices to connect directly to the internet using the IPv6 protocol.

CoAP (Constrained Application Protocol)

- **Purpose**: A lightweight protocol similar to HTTP but designed for IoT devices.
- **How it works**:
    - Uses a simple request-response model like HTTP (GET, POST, PUT, DELETE).
    - Operates over UDP (faster and lighter than TCP).
- **Key Features**:
    - Supports low-power and constrained devices.
    - Built-in mechanisms for discovery, resource observation, and event notifications.
- **Why it's important**: Enables IoT devices to communicate efficiently while saving energy and bandwidth.

MQTT (Message Queuing Telemetry Transport)

- **Purpose**: A messaging protocol used to transfer data between IoT devices.
- **How it works**:
    - Based on a **publish-subscribe model**:
        - Devices publish data to a central broker.
        - Other devices subscribe to topics to receive relevant data.
- **Key Features**:
    - Lightweight, uses minimal bandwidth.
    - Works on top of TCP for reliable delivery.
    - Designed for unreliable or low-bandwidth networks.
- **Why it's important**: Ideal for real-time communication between IoT devices, such as in smart homes or industrial systems.

XMPP (Extensible Messaging and Presence Protocol)

- **Purpose**: Originally a messaging protocol for instant messaging, now extended to support IoT applications.
- **How it works**:
    - Uses XML for exchanging structured messages.
    - Devices can communicate directly or through a central server.
- **Key Features**:
    - Supports secure and real-time communication.
    - Extensible, allowing it to adapt for IoT use cases.
    - Works well with multi-user communication.

- **Why it's important**: Provides a standardized way for IoT devices to send messages, useful for chat-like applications or device-to-device communication.

Li-Fi (Light Fidelity)

- **What it is**: A wireless communication technology using light (LEDs) to transmit data.
- **How it works**: Data is encoded into light signals that are received by a photodetector.
- **Advantages**:
    - Faster than Wi-Fi (in lab conditions).
    - Immune to radio frequency interference.
    - Works in environments sensitive to radio waves, like hospitals.
- **Limitations**:
    - Requires a line of sight between the transmitter (light) and receiver.
    - Cannot penetrate walls like Wi-Fi.

RFID (Radio Frequency Identification)

- **What it is**: A technology using electromagnetic fields to identify and track objects.
- **Components**:
    - **Tags**: Contain data (active or passive).
    - **Readers**: Scan tags using radio waves.
- **Uses**:
    - Supply chain management (e.g., tracking packages).
    - Inventory systems.
    - Access control (e.g., RFID cards).
- **Advantages**:
    - No need for line-of-sight scanning (unlike barcodes).
    - Efficient for tracking large numbers of items.

Cellular Technologies (GPRS, 4G, NB-IoT, 5G)

- **2G/3G (GPRS/Edge)**: Older networks primarily supporting IPv4.

    - **Use case**: Low-bandwidth IoT devices (e.g., basic smart meters).
- **4G (LTE)**: High-speed network supporting both IPv4 and IPv6.
    - **Use case**: Video streaming IoT applications, connected vehicles.
- **NB-IoT (Narrowband IoT)**: Specialized for low-power devices on 4G/5G networks.
    - **Use case**: Smart agriculture, health monitors.
- **5G**: Offers ultra-low latency, massive device support, and high-speed data.
    - **Use case**: Smart cities, industrial IoT, autonomous vehicles.

Z-Wave

- **What it is**: A wireless communication protocol designed for smart home devices.
- **How it works**: Operates in low-frequency bands (868 MHz in Europe, 915 MHz in the US) to avoid interference with Wi-Fi and Bluetooth.
- **Uses**:
    - Controlling smart lights, thermostats, locks, and alarms.
- **Advantages**:
    - Low power consumption.
    - Reliable mesh networking with strong signal penetration.
- **Limitations**:
    - Lower data transfer rates compared to Wi-Fi.
    - Limited range compared to LoRaWAN.

Summary Comparison of Technologies

Technology	Range	Data Rate	Use Case	Frequency	Power Consumption
Li-Fi	Room-specific	Up to 1 Gbps	High-speed environments	Light (Visible)	Medium
RFID	Few cm to meters	Low	Inventory, tracking	Radio waves	Low
Cellular	Kilometers	Varies by tech	Smart cities, vehicles	Licensed bands	Varies
Z-Wave	30-100 meters	100 Kbps	Smart homes	Sub-GHz	Very Low

FUNDAMENTALS OF IoT

1. Introduction to IoT

- The Internet of Things (IoT) connects everyday physical objects to the internet to exchange data.
- **Examples of IoT Devices**:
    - Smart home appliances like thermostats and lights.
    - Wearable devices such as fitness trackers.
    - Smart city solutions like traffic sensors.
- IoT makes devices “smart” by enabling them to sense, process, and communicate.

---

2. How Does IoT Work?

1. **Sensing**: Sensors in IoT devices capture data (e.g., temperature, motion).
2. **Communication**: Data is transmitted to other devices or servers using WiFi, Bluetooth, ZigBee, etc.
3. **Processing**: Data is analyzed locally (on the device), near the device (fog computing), or in the cloud.
4. **Action**: Based on analysis, devices can perform actions like turning off lights or sending alerts.

---

3. Features of IoT

- **Dynamic and Self-Adapting**: Devices adjust based on environmental changes (e.g., smart lights dim when the room is empty).
- **Interoperability**: Devices communicate seamlessly using standard protocols.
- **Self-Configuring**: Devices can automatically connect to networks and configure themselves.
- **Real-Time Operation**: IoT systems provide instant feedback and actions.
- **Unique Identification**: Every device has a unique ID for tracking and management.

---

4. Advantages and Disadvantages of IoT

- **Advantages**:

    - **Automation and Control**: Reduces human effort by automating tasks.
    - **Resource Efficiency**: Optimizes usage of resources (e.g., smart irrigation systems).
    - **Better Decision-Making**: Real-time data improves accuracy in decisions.
- **Disadvantages**:
    - **Security Risks**: Devices can be hacked.
    - **Privacy Issues**: Sensitive user data might be exposed.
    - **Compatibility Issues**: Devices from different manufacturers may not work together.

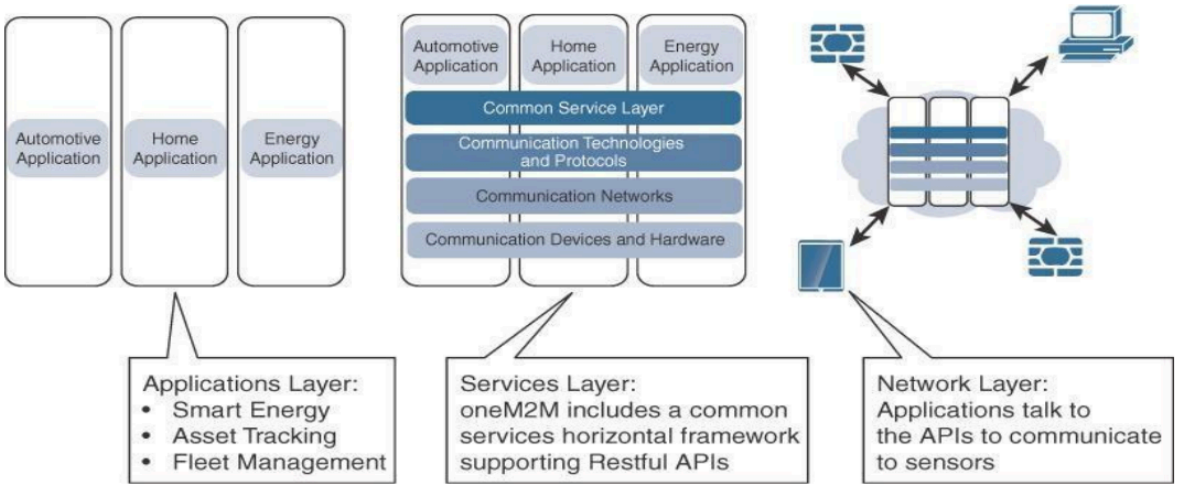
---

oneM2M Architecture

- **Purpose**: To provide a standardized framework for communication and interoperability in IoT systems.
- **Key Components**:
    - **Application Layer**: Includes software that interacts with users and devices.
    - **Service Layer**: Middleware that enables communication and management of devices across platforms.
    - **Network Layer**: Ensures reliable data transmission using protocols like WiFi, ZigBee, and LoRaWAN.
- **Features**:
    - **Interoperability**: Connects heterogeneous systems seamlessly.
    - **Scalability**: Supports millions of devices.
    - **Device Management**: Simplifies tasks like registration and data collection.

Example: Integrating a LoRaWAN-based sensor with a BACnet-based HVAC system using the oneM2M service layer.

The Main Elements of the oneM2M IoT Architecture



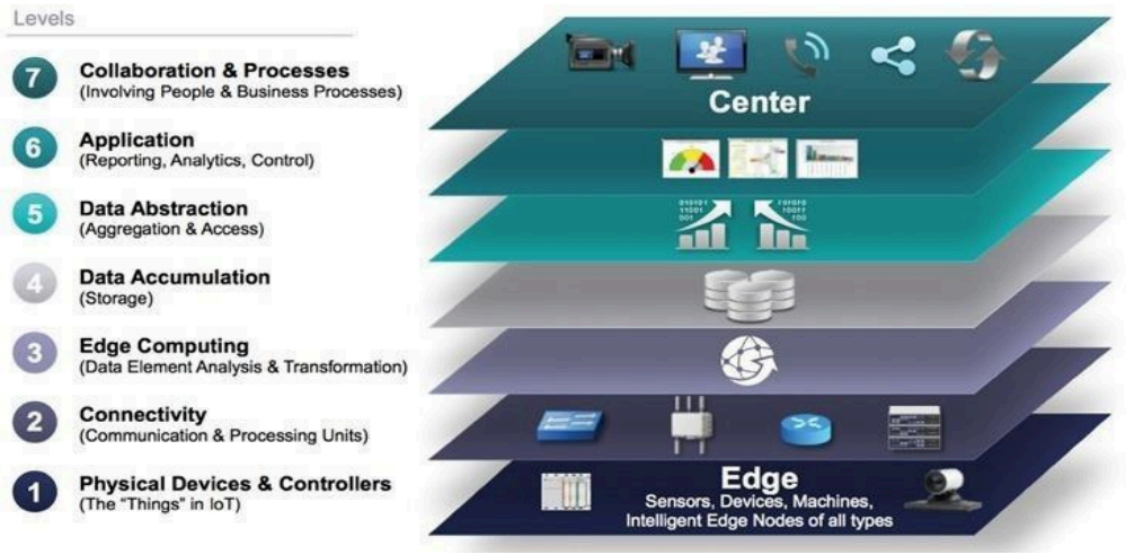
---

5. IoT World Forum (IoTWF) Standardized Architecture

A **7-layer model** defining IoT architecture:

1. **Physical Devices and Controllers**: Sensors and devices that gather data.
2. **Connectivity Layer**: Transmits data reliably and securely.
3. **Edge Computing Layer**: Processes and filters data close to the source.
4. **Data Accumulation Layer**: Collects raw data for further processing.
5. **Data Abstraction Layer**: Converts raw data into meaningful information.
6. **Application Layer**: Provides user interfaces and controls.
7. **Collaboration and Processes**: Integrates data for coordinated decisions.

IoT Reference Model Published by the IoT World Forum



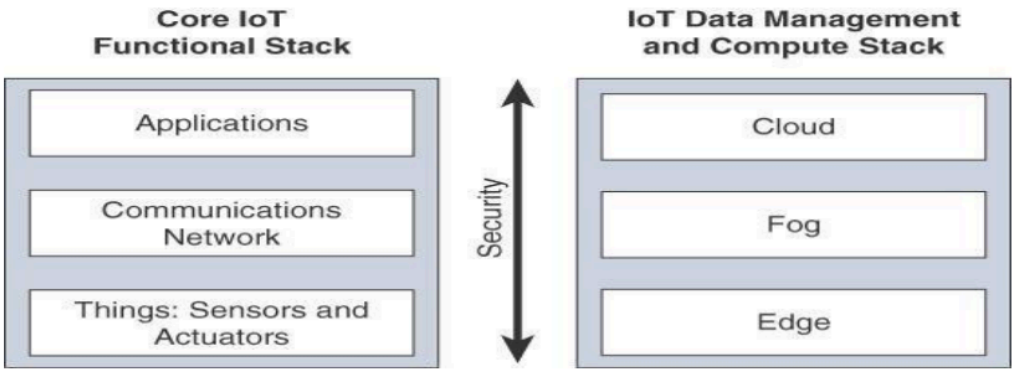
---

6. Simplified IoT Architecture

A practical framework that simplifies IoT into three layers:

1. **Edge Layer**: IoT devices (e.g., sensors) collect and process data locally.
2. **Fog Layer**: Intermediate systems analyze and filter data.
3. **Cloud Layer**: Central storage and complex data analysis.

Example: A smart factory where edge devices detect faults, fog computing handles regional analysis, and cloud systems manage historical trends.



---

7. Core IoT Functional Stack

1. **Things Layer**: Physical devices (e.g., sensors, actuators) that interact with the environment.
2. **Network Layer**: Connects devices and transmits data using protocols like WiFi or ZigBee.
3. **Application Layer**: Processes data, provides insights, and allows control through user interfaces.

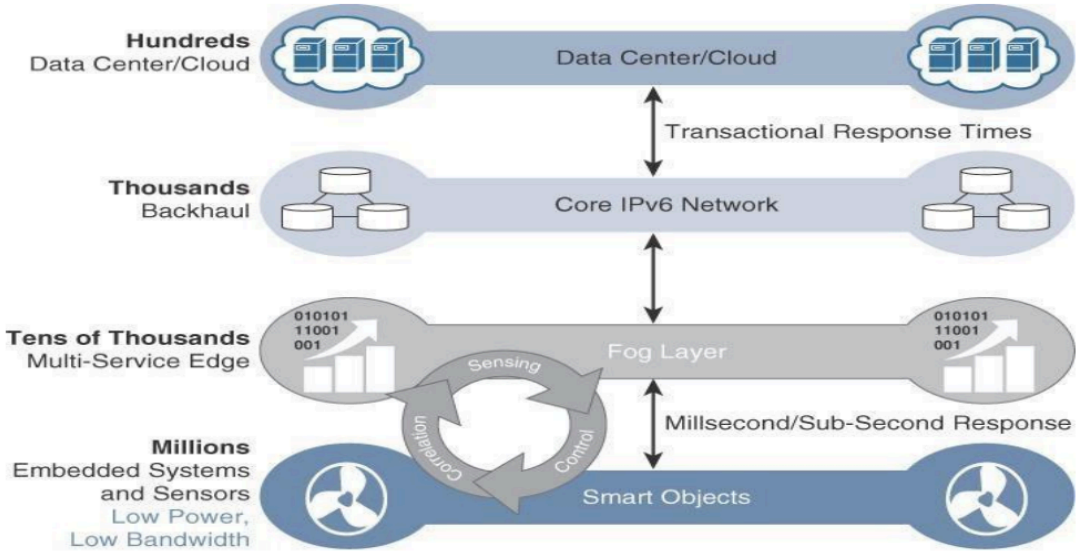
---

8. IoT Data Management and Compute Stack

- IoT generates vast amounts of data that need efficient handling:
    1. **Edge Computing**: Processes critical data on the device for quick action.
    2. **Fog Computing**: Analyzes and filters data locally to reduce latency.
    3. **Cloud Computing**: Performs deep analysis and long-term storage.

Challenges: Managing big data, ensuring data security, and reducing network congestion.

The IoT Data Management and Compute Stack with Fog Computing



---

9. Fog, Edge, and Cloud in IoT

1. **Edge Computing**: Basic processing happens directly on devices (e.g., a motion sensor sends alerts immediately).

2. **Fog Computing**: Data processing happens near the devices (e.g., traffic cameras analyzing congestion).
3. **Cloud Computing**: Centralized for long-term analysis and data storage (e.g., weather prediction systems).

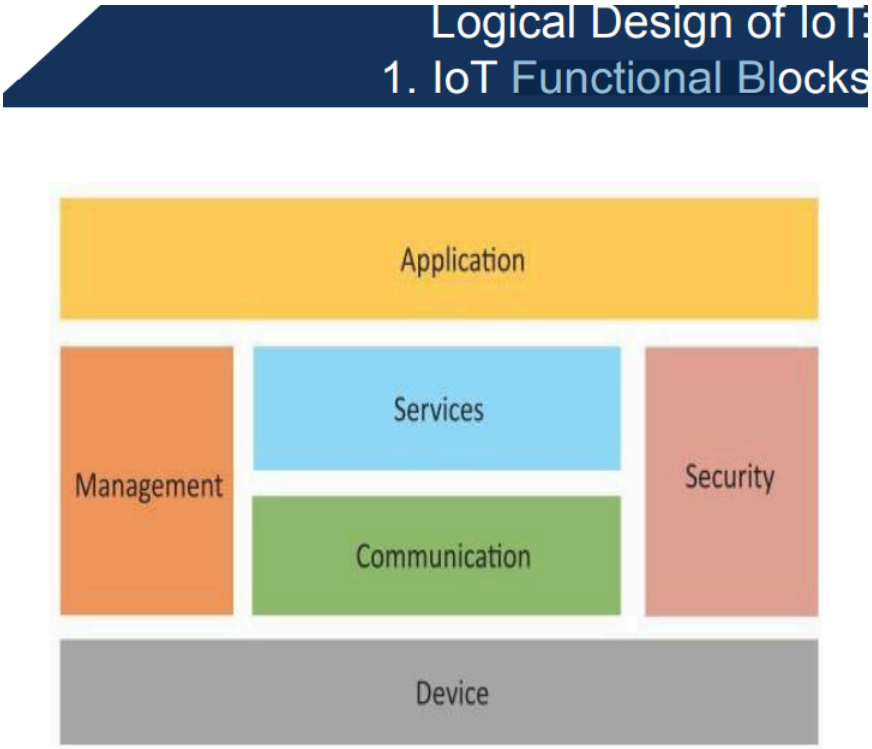
Comparison:

- Edge provides instant response.
- Fog balances local and cloud needs.
- Cloud handles large-scale computations.

---

10. Functional Blocks of an IoT Ecosystem

1. **Device**: Collects data (e.g., sensors, cameras).
2. **Communication**: Transmits data to other devices or servers.
3. **Service**: Offers functionalities like monitoring and control.
4. **Management**: Maintains performance and updates.
5. **Security**: Protects devices and data.



---

11. Sensors, Actuators, Smart Objects, and Connecting Smart Objects

- **Sensors**: Detect and measure physical changes (e.g., temperature, humidity).

- **Actuators**: Perform actions based on commands (e.g., opening a valve).
- **Smart Objects**: Combine sensors, actuators, and processors to perform tasks intelligently.
- **Connectivity**: Uses protocols like WiFi, ZigBee, or LoRaWAN for communication.

Example: A smart lock uses sensors to detect presence, connects to the internet, and locks/unlocks based on commands.

---

12. IoT Challenges

1. **Scalability**: Supporting millions of devices in a system.
2. **Security**: Protecting data from cyberattacks.
3. **Interoperability**: Ensuring devices from different manufacturers work together.
4. **Big Data**: Managing and analyzing vast amounts of data efficiently.
5. **Energy Efficiency**: Minimizing power consumption in IoT devices.

---

13. IoT Network Architecture and Design

- **Design Principles**:
    1. Scalability to support millions of devices.
    2. Security to safeguard data.
    3. Low latency for real-time applications.
    4. Efficient data flow to minimize network congestion.
- **Network Layers**:
    1. **Access Layer**: Connects devices to the network (e.g., WiFi, LoRa).
    2. **Transport Layer**: Uses protocols like IP for secure data transfer.
    3. **Application Layer**: Processes and presents data to users