EDGE COMPUTING

Jerin Biju
MDS202329
jerin.mds2023@cmi.ac.in

Aparna C
MDS202310
aparna.mds2023@cmi.ac.in

Nooh Ali
MDS202337
nooh.mds2023@cmi.ac.in

Hiba AP
MDS202326
hiba.mds2023@cmi.ac.in

What is edge computing?

Edge computing is a distributed computing paradigm that brings computation and data storage closer to the location where it is needed. Instead of relying solely on centralized data centers or cloud computing resources , edge computing extends computing capabilities to the edge of the network , closer to the source of data generation or consumption.

Why is edge computing useful

Low Latency : By processing data closer to where it is generated , edge computing reduces the time it take for data to travel between the source and a centralized data center or cloud. This results in enabling real-time or near-real-time applications

Bandwidth Optimization: Edge computing helps optimize network bandwidth by processing and filtering data locally before transmitting it to centralized servers or the cloud.This reduces the volume of data that needs to be transmitted over the network, saving bandwidth and reducing costs, especially in bandwidth constrained environments

Scalability: Edge computing enables distributed computing architectures that can scale horizontally by adding more edge devices or nodes as needed. This allows for greater scalability and flexibility compared to traditional centralized architectures.

Resilience: Edge computing architectures are inherently more resilient to network failures or disruptions since they can continue to operate autonomously even when disconnected from centralized resources. This ensures uninterrupted operation and improves reliability for critical applications in remote or harsh environments.

Data Privacy and Security: Edge computing can help address data privacy and security concerns by processing sensitive data locally, closer to its source, and reducing the need to transmit sensitive data over public networks. This enhances data privacy, protects sensitive information, and helps organizations comply with regulatory requirements.

Offline Operation: Edge computing allows applications to continue functioning even when connectivity to centralized resources is lost. This is particularly useful for applications in remote or offline environments, such as oil rigs, ships, or mining operations, where continuous operation is critical.

Real-time Insights: Edge computing enables real-time analysis and processing of data at the point of generation, allowing organizations to derive actionable insights and make faster decisions based on the latest data. This is valuable for applications such as predictive maintenance, anomaly detection, and situational awareness.

Cost Efficiency: Edge computing can help reduce costs associated with data transmission, storage, and processing by offloading computation and storage tasks to edge devices or local infrastructure. This can lead to cost savings, especially for applications with large volumes of data or high network bandwidth requirements.

Where is edge computing useful

Internet of Things (IoT): Edge computing is widely used in IoT deployments to process and analyze data generated by sensors, devices, and equipment at the edge of the network. This includes applications such as smart cities, industrial automation, smart homes, asset tracking, and environmental monitoring.

Industrial Automation and Manufacturing: Edge computing is used in industrial automation and manufacturing to enable real-time monitoring, control, and optimization of production processes. It supports applications such as predictive maintenance, quality control, process optimization, and remote asset management.

Healthcare and Telemedicine: Edge computing is deployed in healthcare settings to enable real-time monitoring, diagnosis, and treatment of patients. This includes applications such as remote patient monitoring, wearable health devices, medical imaging analysis, and telemedicine consultations.

Autonomous Vehicles and Transportation: Edge computing is used in autonomous vehicles and transportation systems to enable real-time decision-making and control. This includes applications such as autonomous driving, traffic management, vehicle-to-vehicle (V2V) communication

Edge AI and Video Analytics: Edge computing is used to deploy artificial intelligence (AI) and machine learning (ML) models at the edge of the network for real-time video analytics, object detection, facial recognition, anomaly detection, and predictive maintenance.

Architecture of edge computing

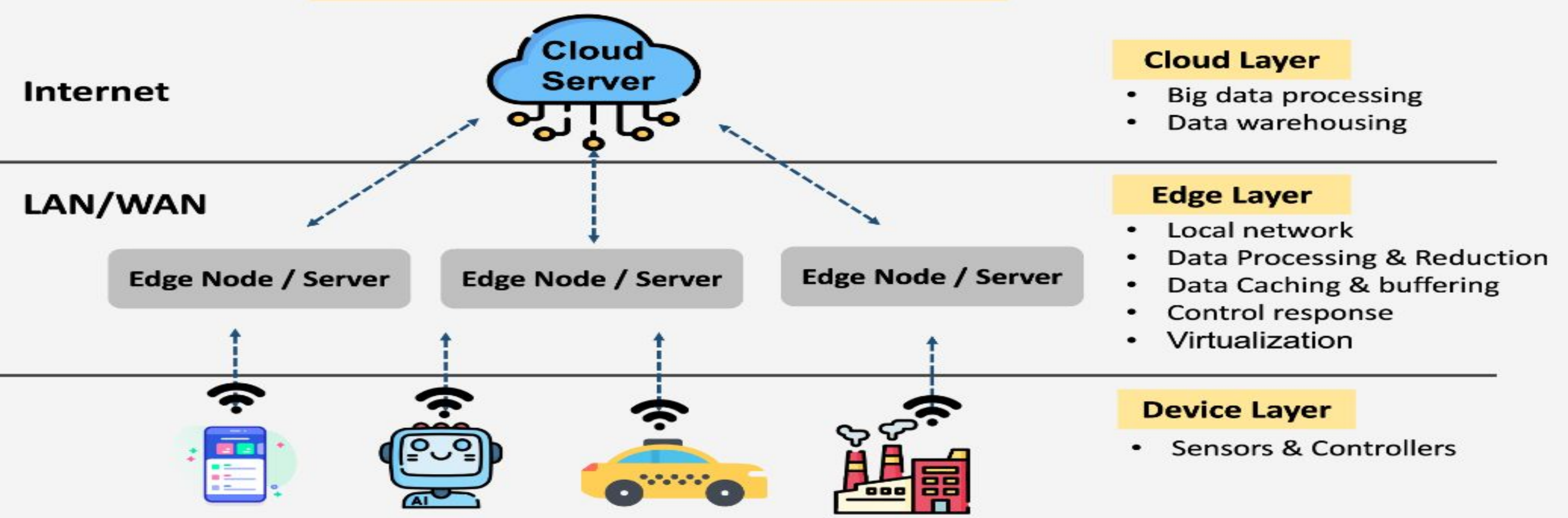


Figure ： Edge computing architecture overview
Source ： The research team

A typical edge computing architecture can be divided into three layers: The cloud layer, edge layer and device layer.

Device Layer

- Through their sensors, devices in the device layer collect and capture data
- Equipment in a hospital collecting vital signs of patients and autonomous vehicles capturing data of other nearby vehicles are all such examples.
- Although components in the cloud and edge layers possess better computing power, the devices in the device layer can still perform data analyses, processing and storage tasks which require negligible computing power, as well as process data closest to the data source in almost real-time.

Edge Layer

- core in the entire edge computing architecture.
- the edge layer contains edge servers that are larger in quantity and more vastly deployed.
- Through distributed edge computing, the edge layer can process data that is closer to the data source and address latency problems found in cloud computing.
- Data which cannot be processed in the edge layer can be sent to and analyzed in the cloud layer to ensure data integrity.

Cloud Layer

- Although edge computing was introduced to address network congestion and latency problems commonly found in cloud computing, cloud computing in fact still plays an important role in the entire edge computing architecture.
- If complex processing is needed, data will be sent to cloud layer from the edge layer.

  For example in the case of video processing, when movement is detected in a video feed at the edge, it may trigger sending relevant data to the cloud for further analysis or storage. This could involve identifying the type of movement, analysing patterns over time, or comparing against historical data for anomaly detection.

- Edge servers will also pass critical data to the cloud layer for long term storage and comprehensive analysis.

EdgeOS$_H$: A Home Operating System for Internet of Everything

SMART HOME

Our understanding of smart home is that it should be an automated and energy efficient domestic place where occupants could enjoy a healthier and more comfortable life. A smart home should come with

- Self-awareness
- Self-management
- Self-learning

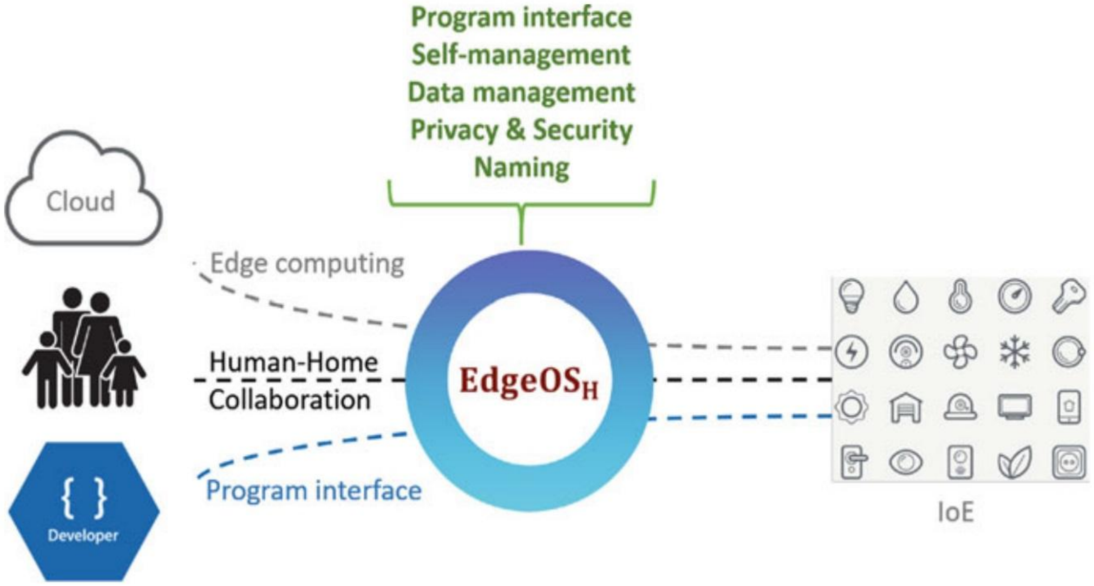
SMART HOME

- **PROBLEM** : Current systems often work in a silo-based manner. This makes it difficult to manage devices, data, and services.

- **SOLUTION** : EdgeOS$_H$ a proposed home operating system for the Internet of Everything.

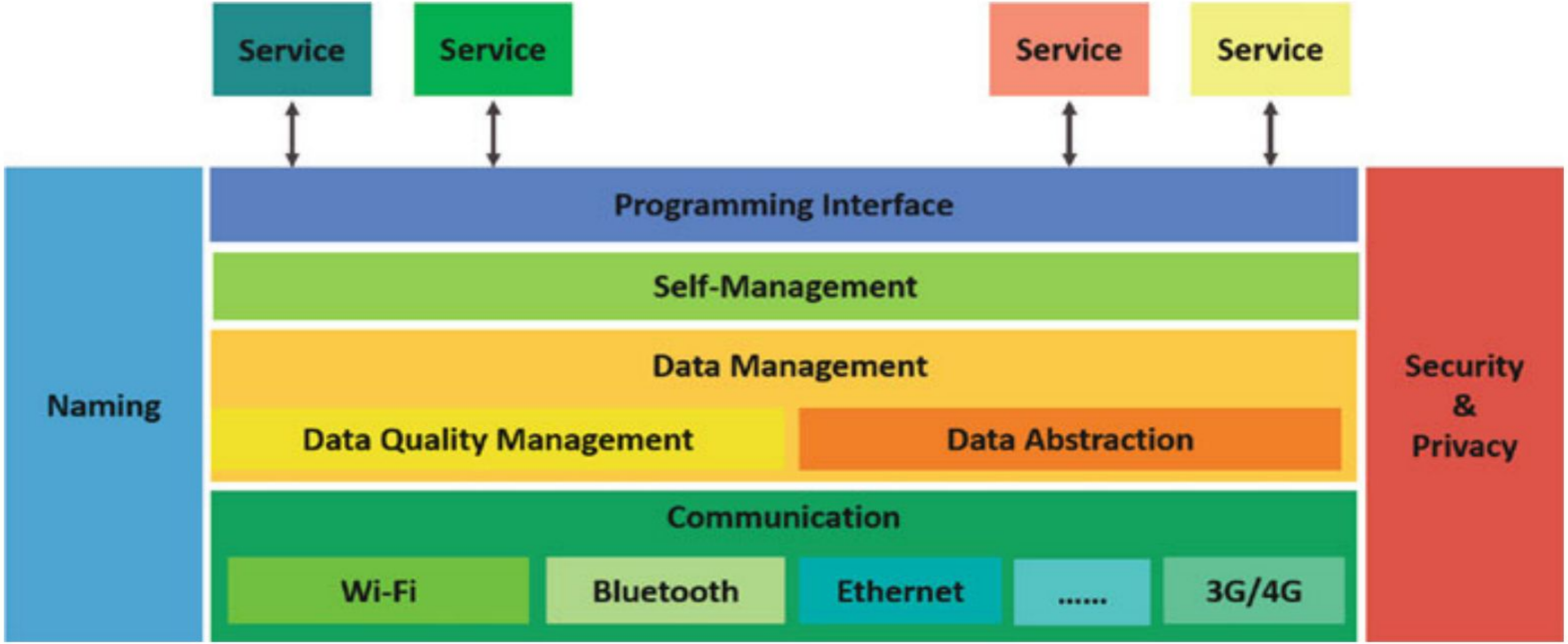
EdgeOS$_H$

1. **Data Flow in Smart Homes:**
   - Data is generated by various sensors and devices within a smart home.
   - The same data is consumed to serve the home occupants.
2. **Edge Computing for Smart Homes:**
   - The principle of “computing at the proximity of data sources” aligns with smart homes.
   - **Edge Computing** is the ideal paradigm for smart homes, as it enables computation close to where data is produced.



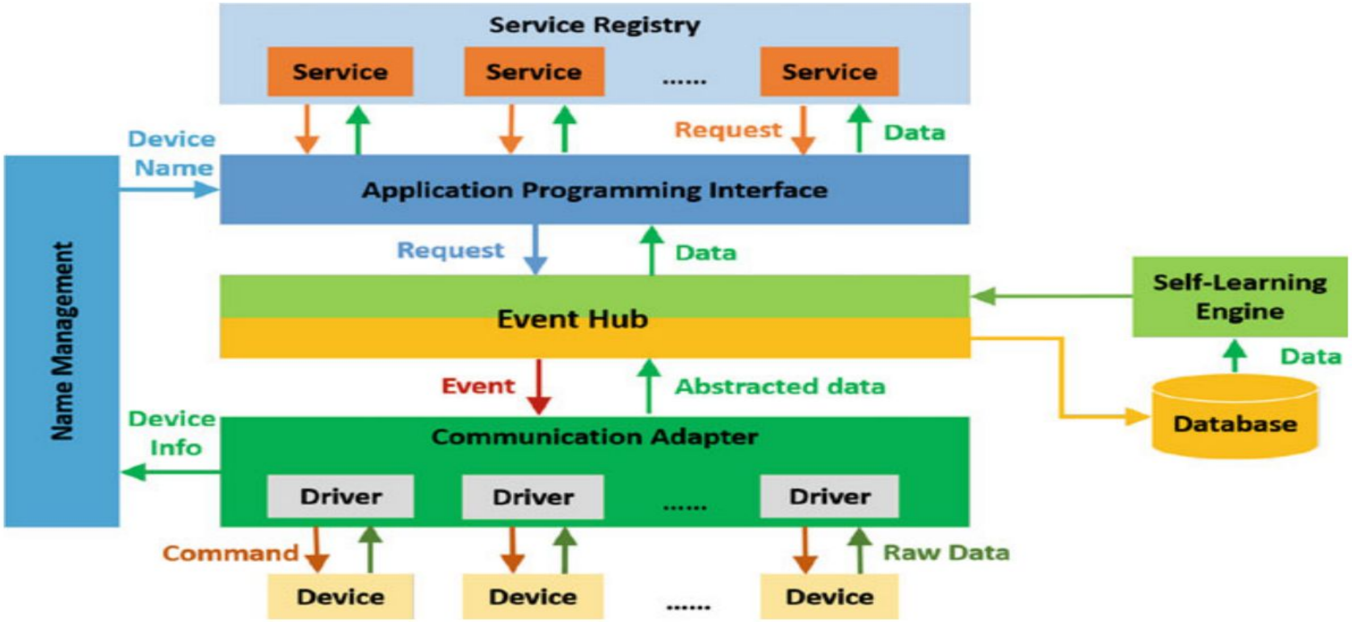
13. **Introducing EdgeOS$_H$:**
    - **EdgeOS$_H$** is a smart home operating system designed for the IoE.
    - It acts as a bridge connecting: Devices within the home, Cloud services, Home occupants, Developers.

Overview of EdgeOS$_H$

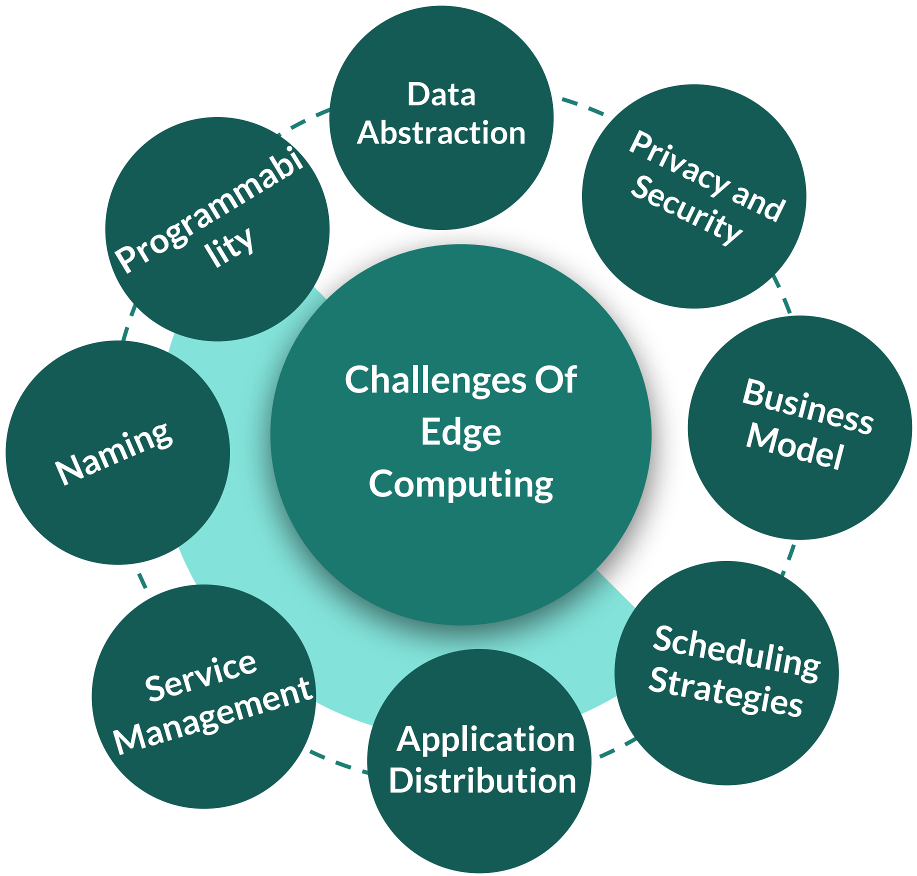


The Logical view of EdgeOS$_H$, which consists of four vertical layers: Communication, Data Management, Self-Management, and Programming Interface, as well as two extra components across all four layers, i.e., Naming and Security & Privacy.

Design of EdgeOSH



The design of EdgeOSH have seven components: Communication Adapter, Event Hub, Database, Self-Learning Engine, Application Programming Interface, Service Registry, as well as Name Management, which stretches across other components


Challenges Of Edge Computing
Data Abstraction
Privacy and Security
Business Model
Scheduling Strategies
Application Distribution
Service Management
Naming
Programmability

Programmability

- **Cloud Computing** : users write code and deploy it to the cloud without knowing where it runs
- **Edge Computing** : As tasks move closer to users in edge computing, dealing with different types of edge devices and their varying speeds makes it harder to develop and program applications.
- Uses **Computing streams** to enhance Programmability.
- Computing streams are sequences of functions or computations applied to data as it travels along its path, from its source to its destination.

Naming

- An efficient Naming mechanism for Edge Computing has not been build and standardized yet .
- **DNS** and **URI** naming mechanisms aren't adaptable to dynamic Edge networks, which often involve highly mobile and resource-constrained devices.
- New naming mechanisms
  - **Named Data Networking** [Hierarchically structured,Human friendly,Scalable but has security issues]
  - **MobilityFirst** [Better mobility support,But not Human friendly]

Data Abstraction

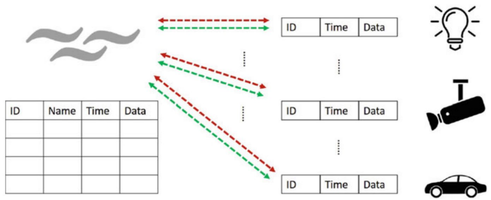


Fig. 5.2 Data abstraction issue for Edge computing

- Data reported from different things comes with various formats.
- Deciding the level of data abstraction can be tricky; filtering too much raw data may limit learning for certain applications or services.
- Data from Edge devices may be unreliable due to **low-precision sensors**, **hazardous environments**, and **unreliable wireless connections**.

Service Management

In edge network service management, a reliable system should support:

- **Differentiation**: Distinguishing services.
- **Extensibility**: Adapting to service changes.
- **Isolation**: Separating services.
- **Reliability (DEIR)**: Ensuring consistent performance.

Privacy and Security

- Balancing service provision with privacy protection poses a significant challenge.
    - For instance, in IoT-enabled homes, sensitive information can be derived from usage data, such as electricity or water consumption, raising concerns about privacy.
- Data collected from Edge devices should remain at the Edge to prioritize user ownership and privacy.
- Missing of efficient tools to protect data privacy and security at the Edge of the network

Application Distribution

- Current approaches include static distribution during compilation and dynamic distribution during runtime.
- But they need to adapt to the heterogeneous nature of Edge nodes and consider Cloud-Edge and Edge-Edge relationships
- The idea of program dependence graphs can help determine distribution based on resource, distance, and communication considerations, supporting both static and dynamic processes in Edge computing.

Scheduling Strategies

- Scheduling strategies in Edge computing aim to optimize resource usage, reduce response time, and enhance task processing efficiency.
- Graph theory serves as a basis for implementing scheduling strategies, where applications are represented as nodes and communication between locations as edges, facilitating efficient resource allocation and application mapping.

Business Model

- Users request data from owners, and processed results are provided by either the cloud center or Edge data owner.
- Transforms the model from a single center-user relationship to multilateral center/user-user relationships.

References

- Edge Computing: A Primer,Jie Cao, Quan Zhang, Weisong Shi
- Wikipedia contributors (14 April 2024).
- Fsp-group website

Thank You!