22AIE305

22AIE305: CLOUD COMPUTING

Dr. C. Rajan,
Office: N110 (West cabin)
WhatsApp: 8113053359
rajancv@am.amrita.edu

Cloud computing

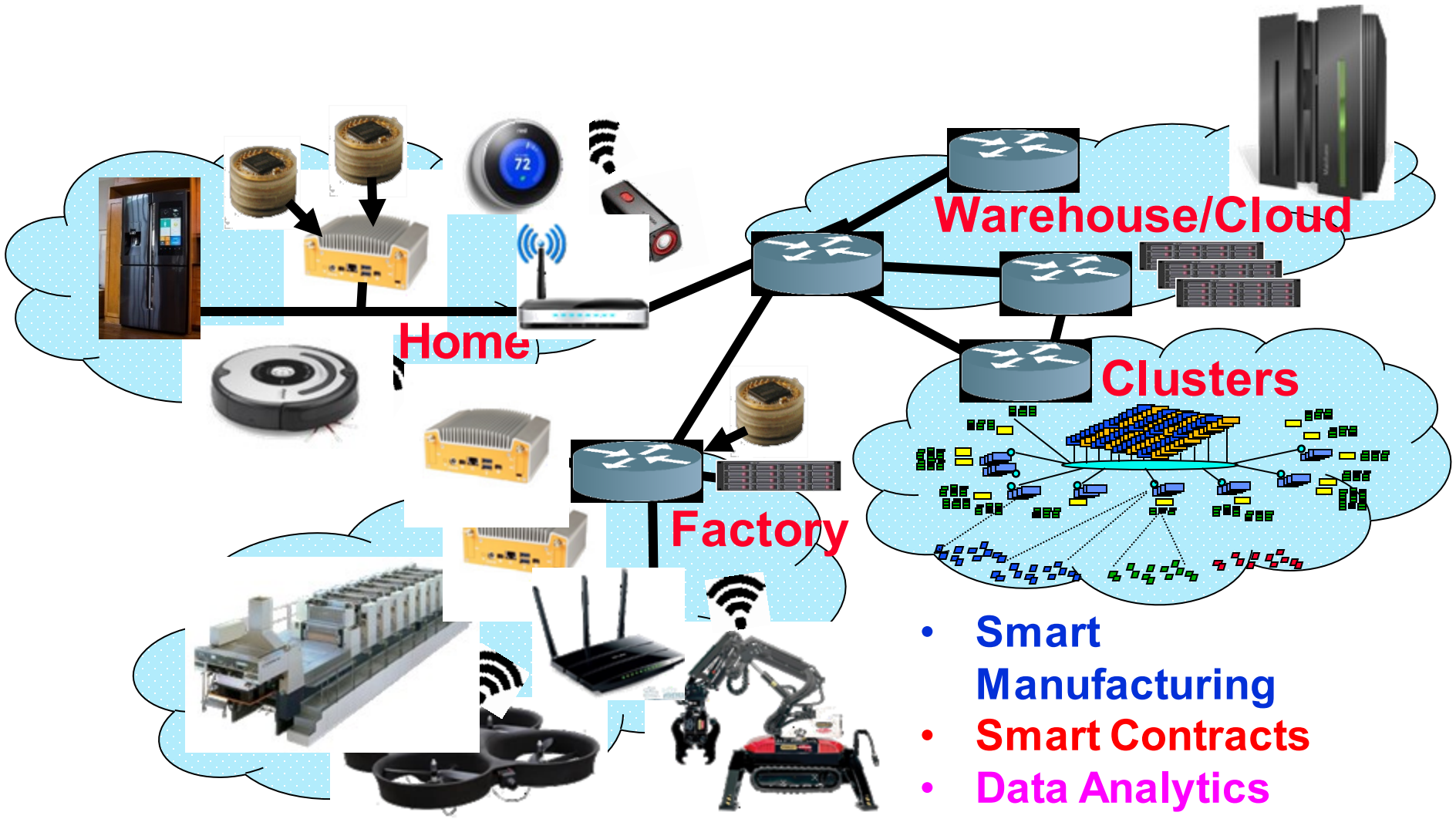
- In Cloud Computing, collected data are streamed to a central server and data is processed on a powerful central server which is usually far away from the data source.
- Cloud Computing or simply “the Cloud”, is a means to host and deliver on-demand computing resources or services via

  a public infrastructure (e.g. the Internet).
- The services vary from elementary services like computation and data storage to more advanced services like machine learning.

Cloud computing provides

- highly scalable and almost unlimited storage for business solutions.
- really high processing power when compared to the Fog/Edge computing.
- High latency, possible downtime, and security
- Resource sharing 24x7 basis
- High redundancy

A Physical View



- Smart Manufacturing
- Smart Contracts
- Data Analytics
- Robotic Factories

Mobile Cloud Computing (MCC)

- smartphones do not have sufficient processing power, memory, and storage capabilities as a typical and traditional cloud does.
- There are research works initiated in order to bring in an appropriate partition of software applications into a set of easily manageable modules.
- The specific and smaller modules are kept within smartphones, whereas the common and reusable service modules are being taken to cloud environments.

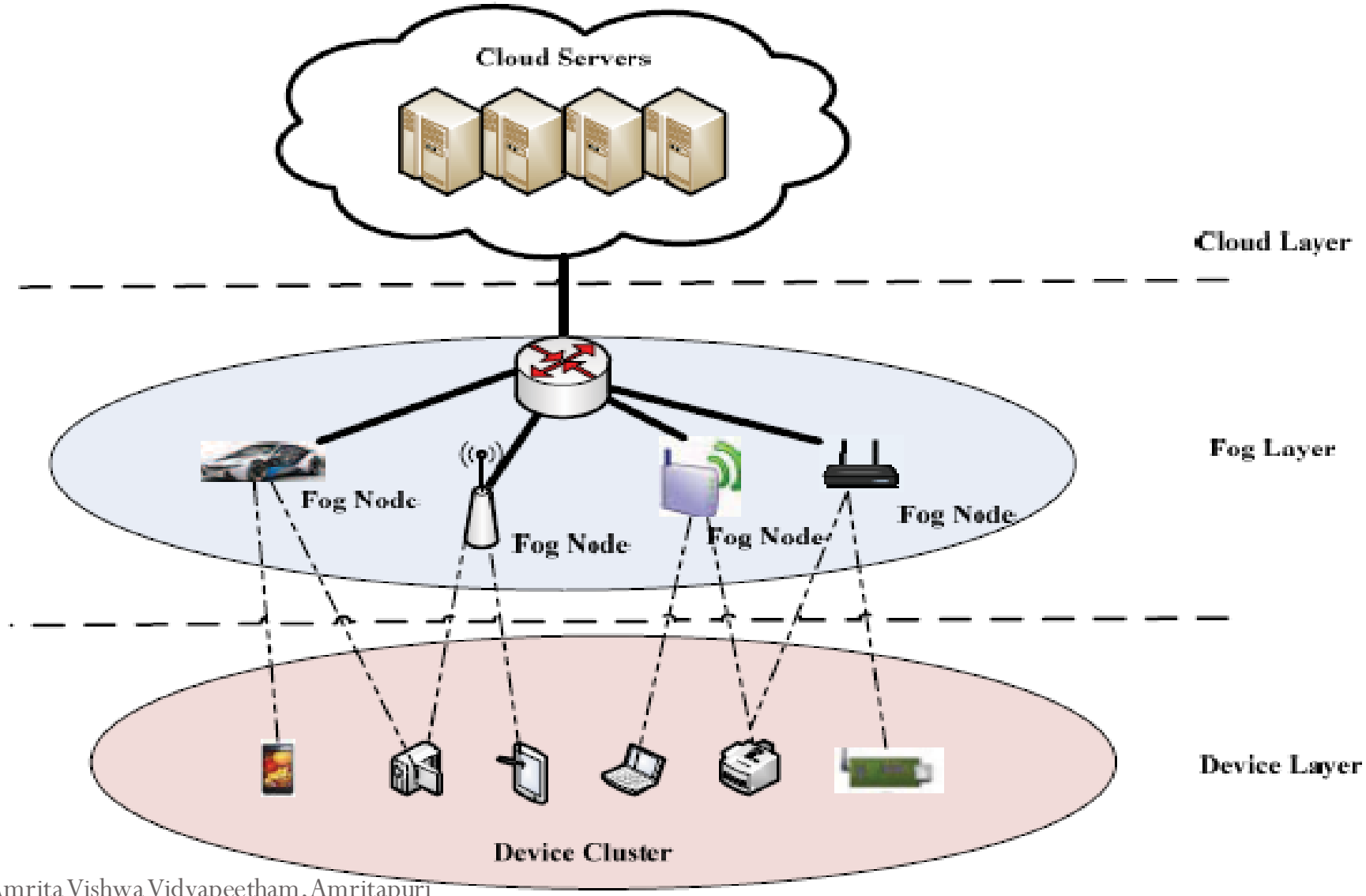
Fog computing

- Fog computing is a distributed architecture that provides computing, storage, and communication services near IoT/Arduino/edge devices, overcoming the limitations of cloud computing such as response time delay, bandwidth utilization, connectivity, and security.
- It supports mobility, scalability, performance, and heterogeneity, enabling agility, proactiveness, and logistical data for artificial or machine learning devices.
- Incorporating fog computing in the digital twin process makes it more robust.

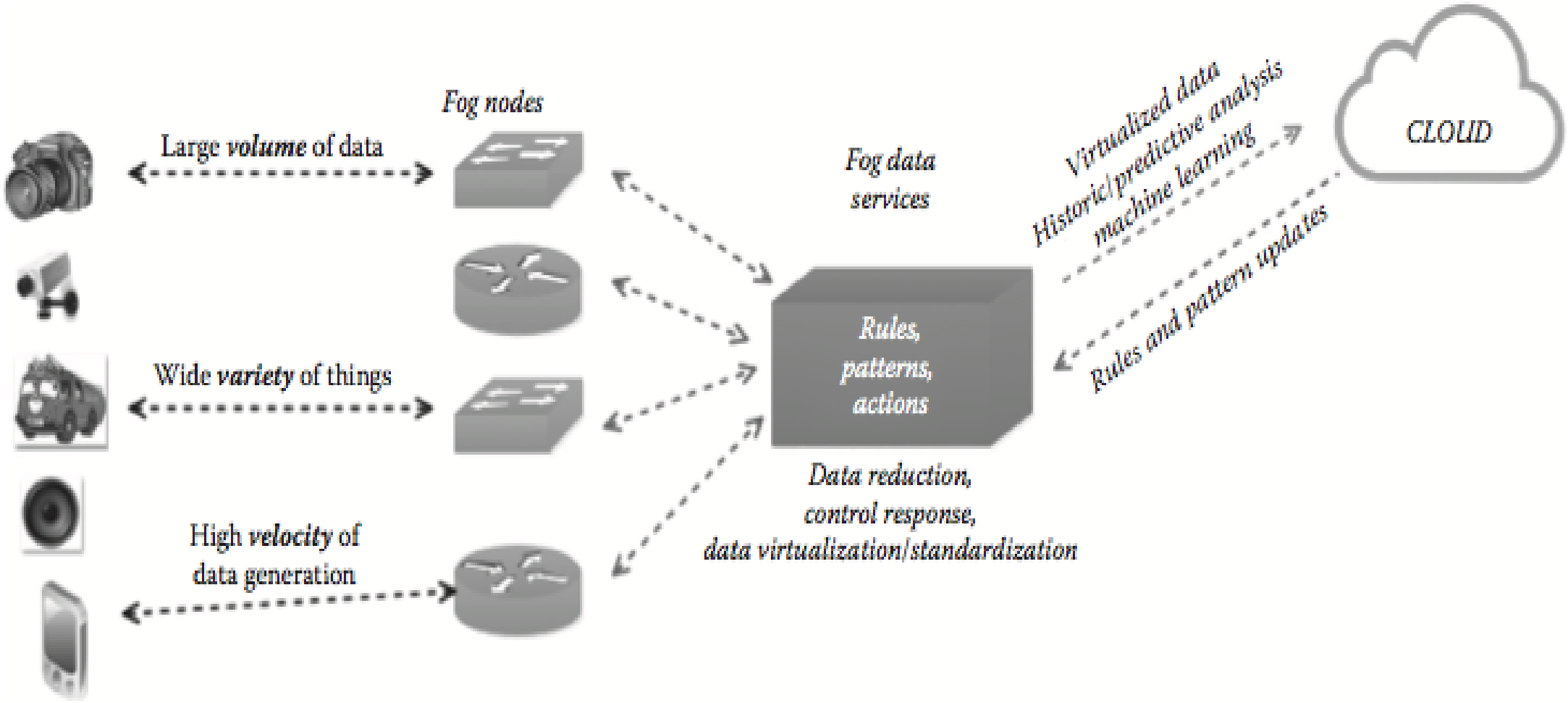
Limitations of cloud computing

- 1. Response time delay.
- 2. Bandwidth utilization.
- 3. Connectivity.
- 4. Security.
- Fog computing addresses these challenges by placing the required services near the edge-device
- Prevalence of ubiquitously connected smart-devices relying on cloud services create bandwidth bottlenecks

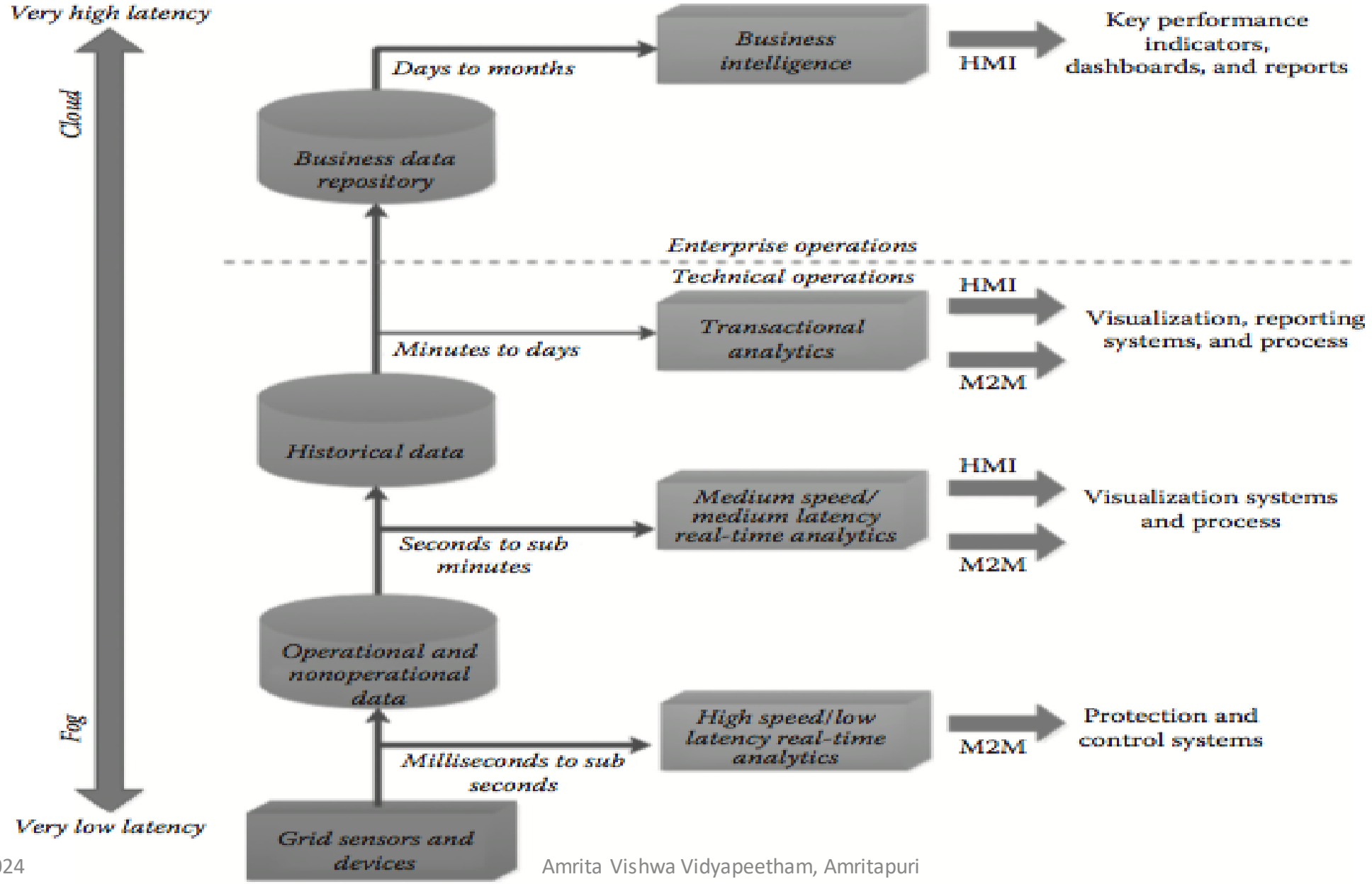
Fog Computing Layers



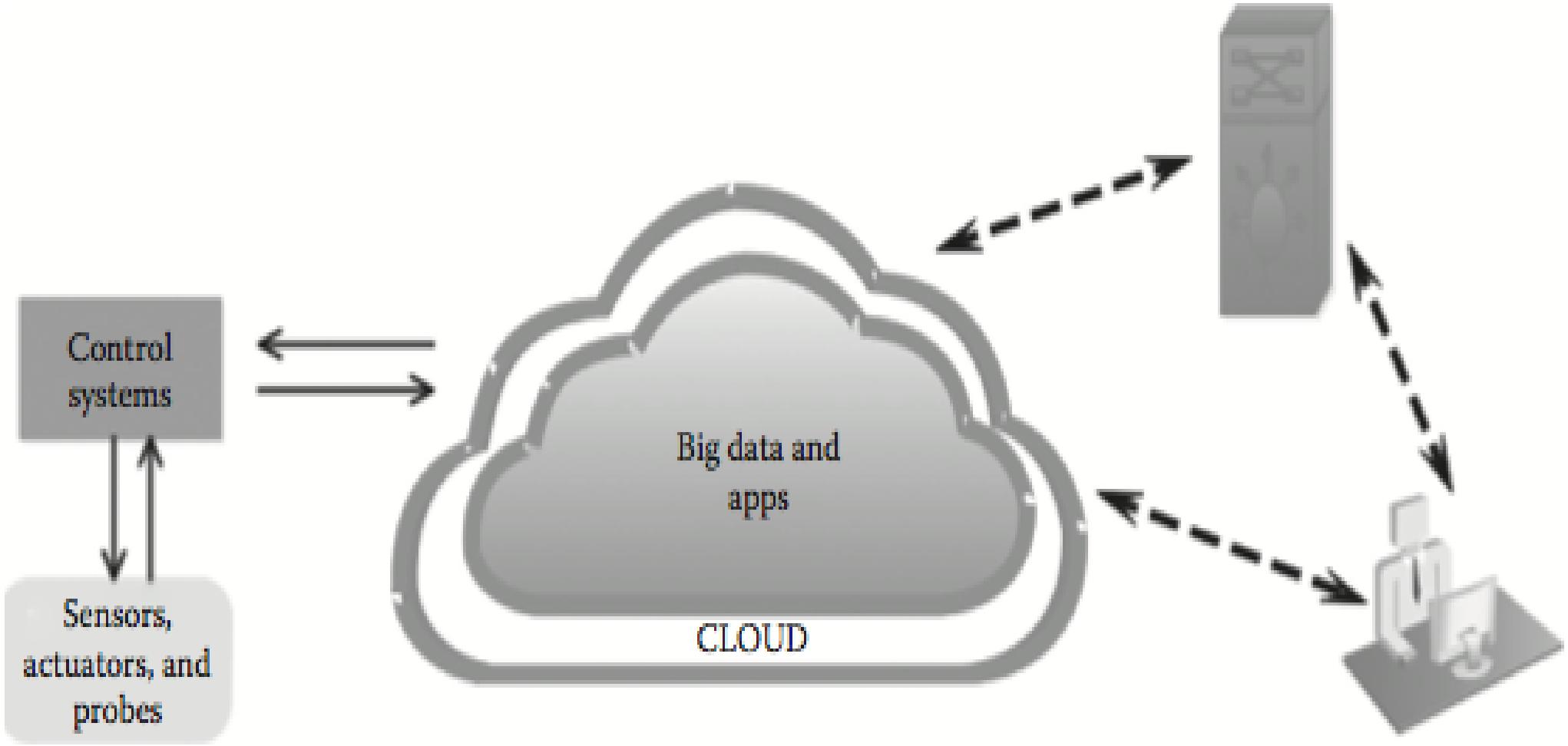
Depicting the control flow for the fog device data analytics



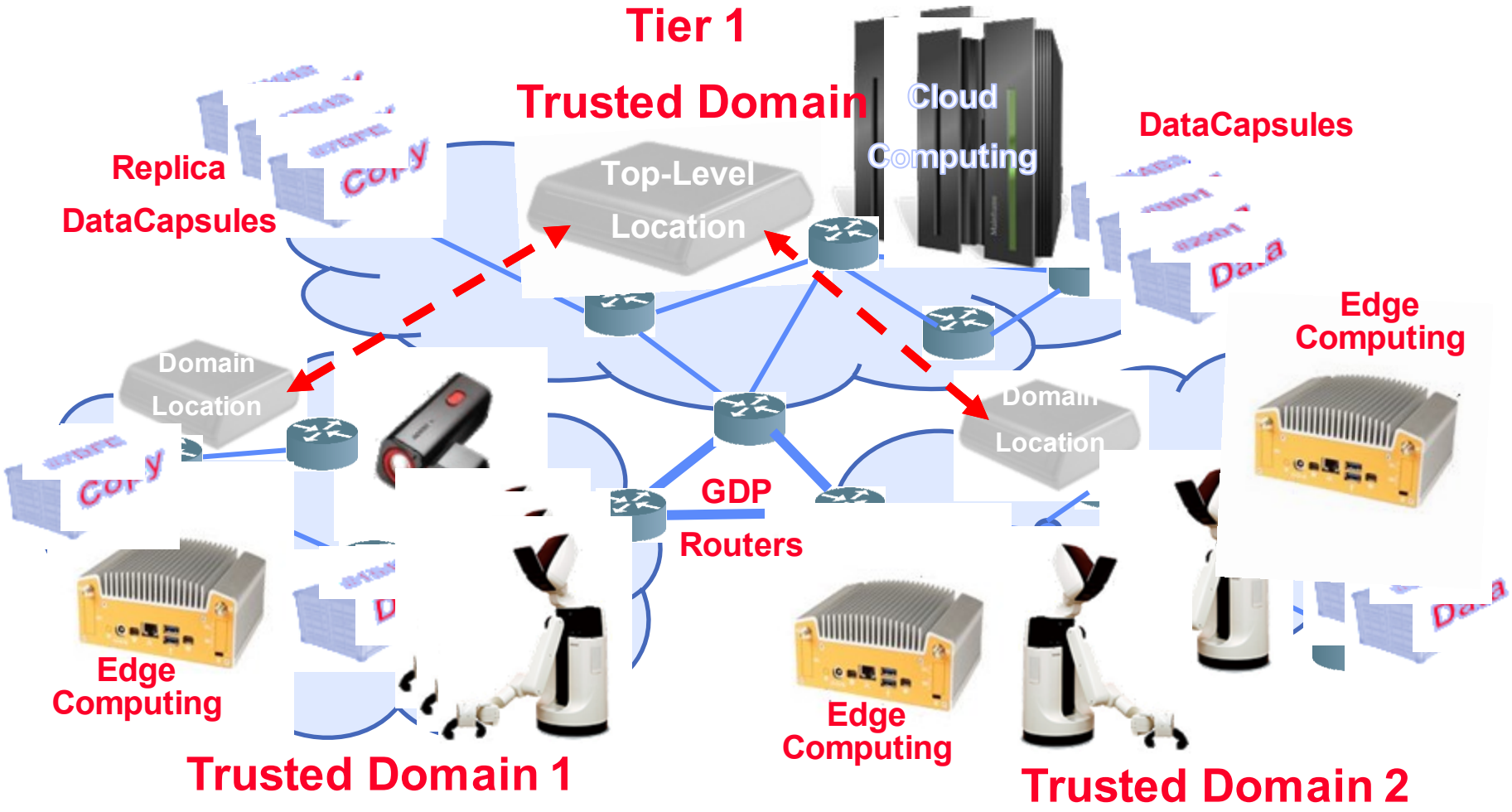
Describing fog device-based and cloud-based data analytics



The end-to-end fog—cloud integration architecture.



Fog Robotics on the Global Data Plane (GDP)



Need for Edge computing

- Smart sensors and actuators are being increasingly deployed in many environments such as homes, hospitals, offices, factories and hotels in order to minutely monitor, precisely measure, and insightfully manage the various parameters of the environments.
- Powerful sensors are embedded and etched on different physical, mechanical, electrical, and electronics systems in our everyday environments in order to empower them to join in the mainstream computing.
- All these generate voluminous amount of data

The Introduction of Fog/Edge Computing

- Traditional networks, which feed data from devices or transactions to a central storage hub (data warehouses and datamarts), cannot keep up with the data volume and velocity created by IoT devices.
- Nor can the datawarehouse model meet the low latency response times that users demand.
- This is how Fog and Edge computing became popular

Fog vs Edge Computing

- The prime objectives of edge computing and fog computing are similar.
- Both of them bring cloud computing-like capabilities to the edge of a network.
- They enable the computation and storage capacities within the close proximity of end users to reduce service latency and save network bandwidth for delay sensitive applications.

Fog vs Edge Comparison

	Edge computing	Fog computing
Architecture	Hierarchical, decentralized, distributed	Hierarchical, decentralized, distributed
Proximity to end devices	Located in end devices	Near (single network hop or few network hops)
Latency	Low	Low
Bandwidth costs	Low	Low
Resource	More limited	Limited
Computation and storage capabilities	More limited	Limited
Mobility	Supported	Supported
Scalability	High	High
Service	Virtualization	Virtualization

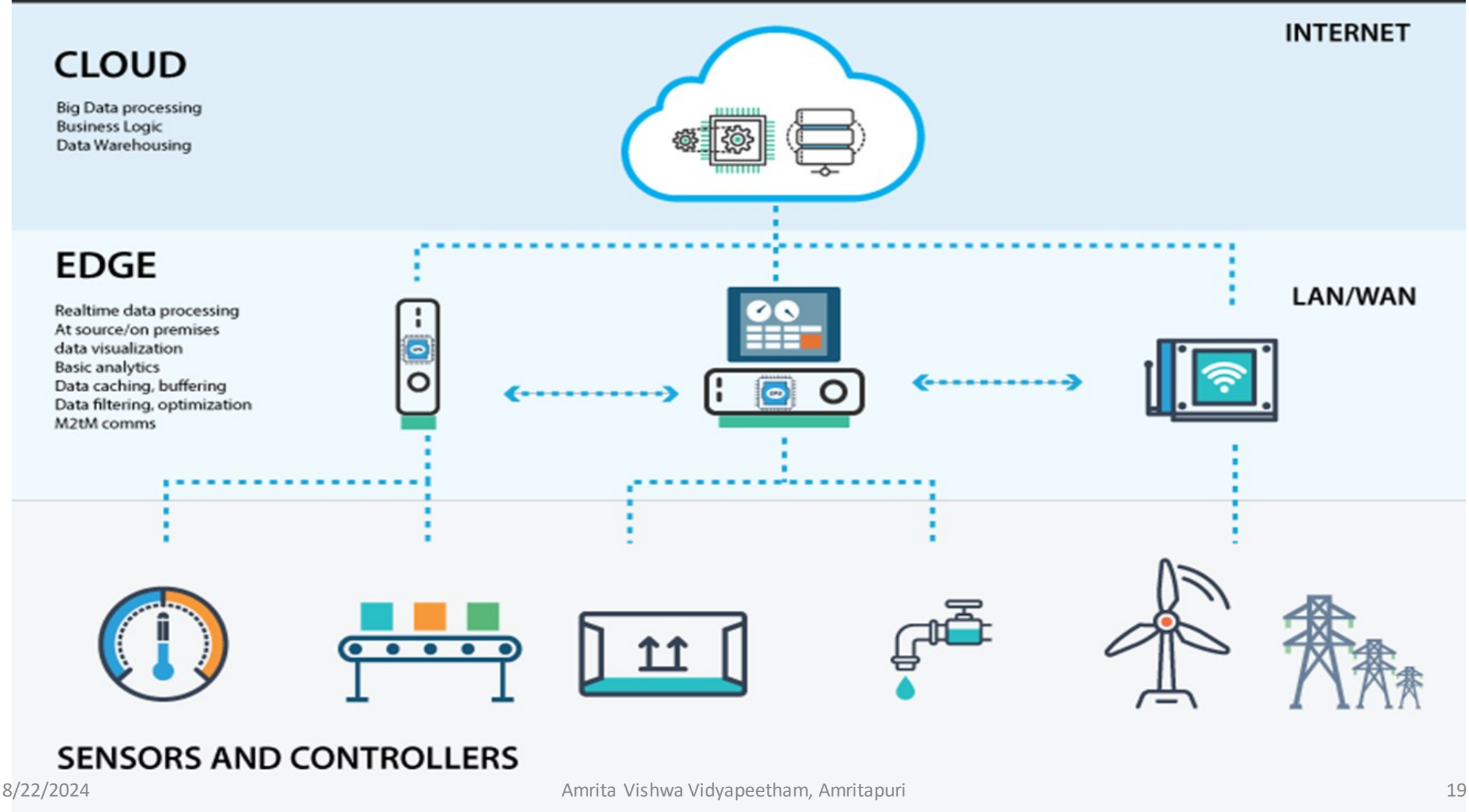
Edge computing

- Edge computing is a computing model that extends cloud service to the edge devices.
- It refers to the enabling technologies which allow computation and storage to be performed on edge devices so that computing and storing happens in the proximity of data sources.
- The edge nodes and devices with computing capacity perform a large number of computing tasks (e.g., data processing, temporarily storing, devices management, decision making, and privacy protection) to reduce the network latency and traffic between end devices and cloud.
- These edge nodes can be composed of smart sensors, smart phones, and smart vehicles, even a special edge servers.

Edge nodes

- They can interconnect and intercommunicate in the local network to form an edge network.
- Moreover, edge devices connect with cloud data center by core network.
- Edge computing provides edge intelligence services nearby to meet the critical demands of the digital industry in agile connection, real-time services, data optimization, application intelligence, security and privacy protection.

Cloud and Edge Computing - Complementary Technologies powering IIoT

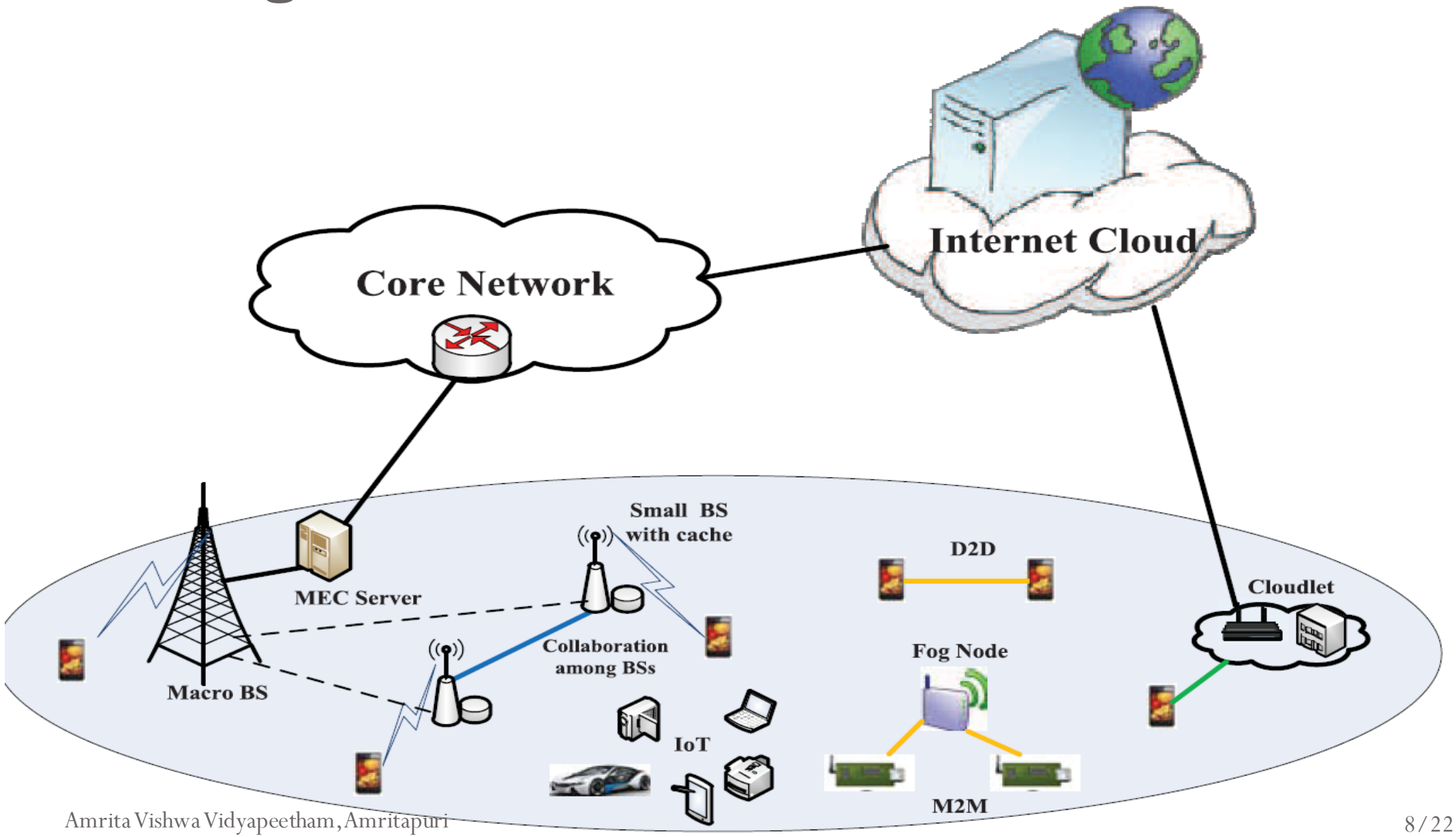


	Cloud computing	Fog computing
Latency	High	Low
Real time interactions	Supported	Supported
Mobility	Limited	Supported
Location awareness	Partially supported	Supported
Number of server nodes	Few	Large
Geographical distribution	Centralized	Decentralized and distributed
Distance to end devices	Far (multiple network hops)	Near (single network hop or few network hops)
Location of service	Within the Internet	At the edge of the local network
Working environment	Specific data center building with air conditioning systems	Outdoor (streets,base stations, etc.) or indoor (houses, cafes, etc.)
Communication mode	IP network	Wireless communication: WLAN, WiFi, 3G, 4G, ZigBee, etc. or wired communication (part of the IP networks)
Dependence on the quality of core network	Strong	Weak
Bandwidth costs	High	Low
Computation and storage capabilities	Strong	Weak
Energy consumption	High (especially the energy consumption of data center coolant system)	Low

Fog versus Edge Computing

- This is the time to clarify the perceptible differences between fog and edge computing models.
- With the faster integration of the IoT devices, instruments, monitors, controllers, sensors, actuators, robots, machines, and so on in our daily environments, there is a need to perform a portion of computing at IoT devices or around their proximity.
- However due to the constrained resources of these IoT devices, the overwhelming usage of microcontrollers as the IoT device gateway for remote connectivity (IoT device-to-enterprise/cloud integration) has dawned upon us.

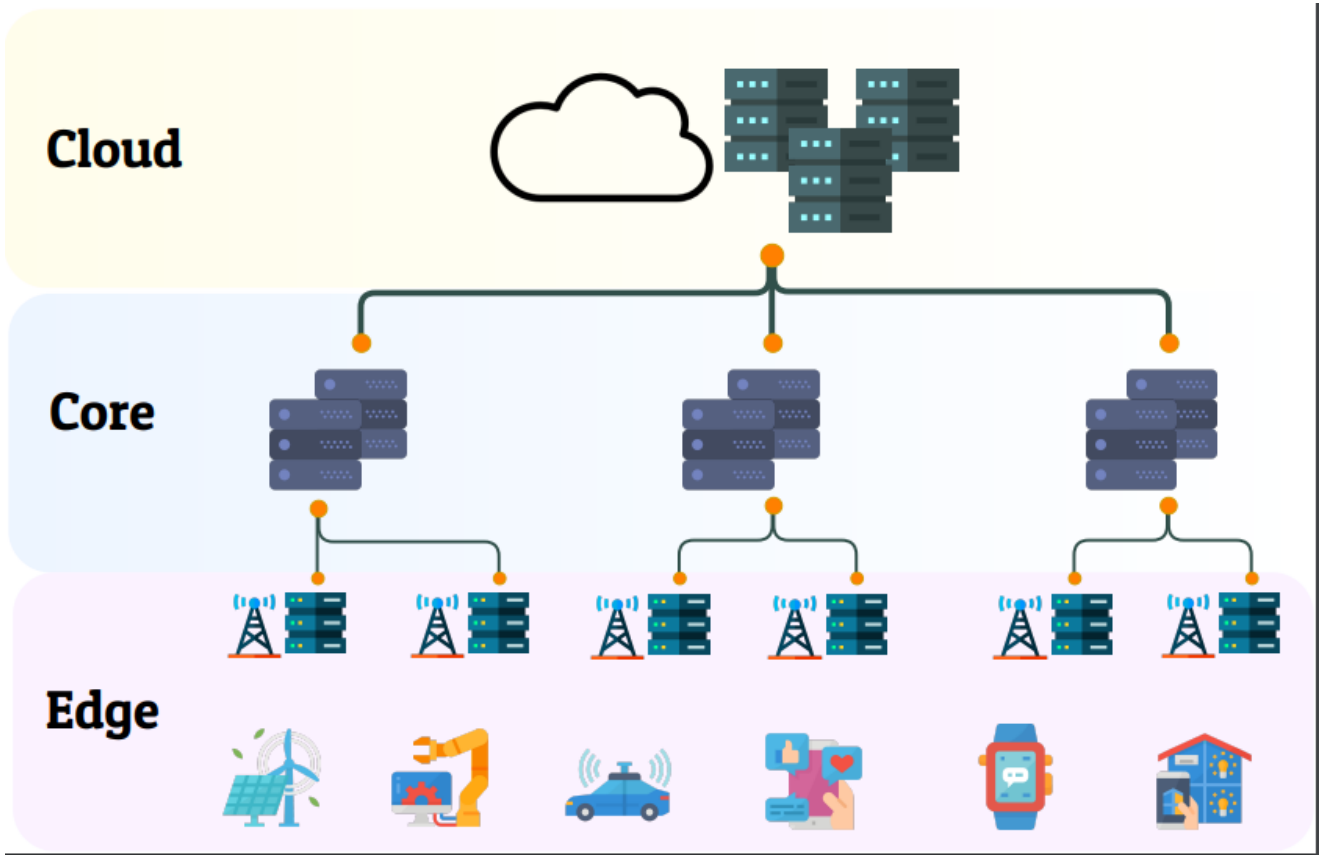
General Edge Networks



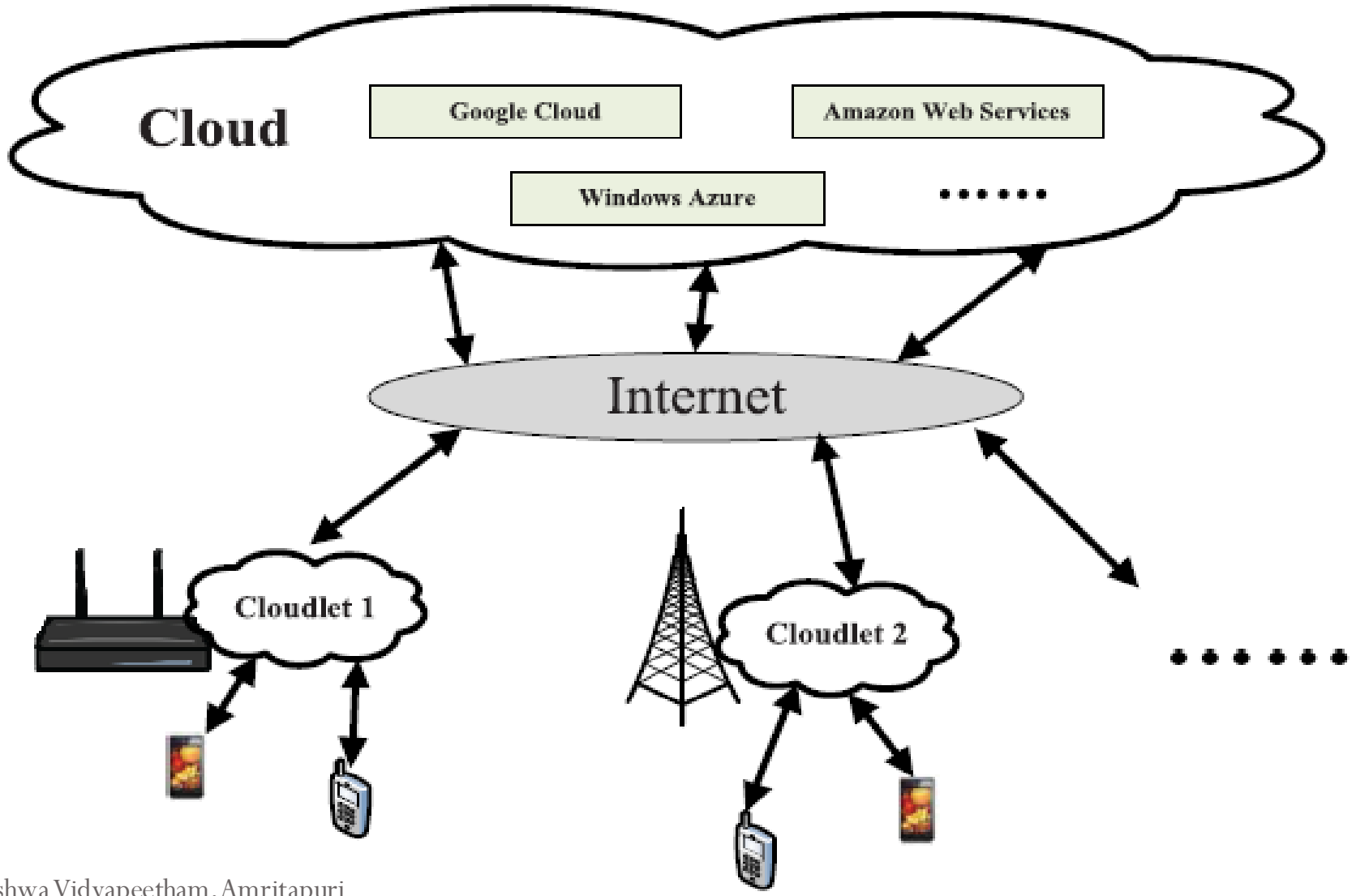
Edge Computing

Bring the cloud closer to the users (Mobile Edge Computing, Cloudlets, Fog Computing)

- Computation and Storage units are placed near the user
- On the “Edge” of the network



Cloudlets


Cloud
Google Cloud
Amazon Web Services
Windows Azure
Internet
Cloudlet 1
Cloudlet 2

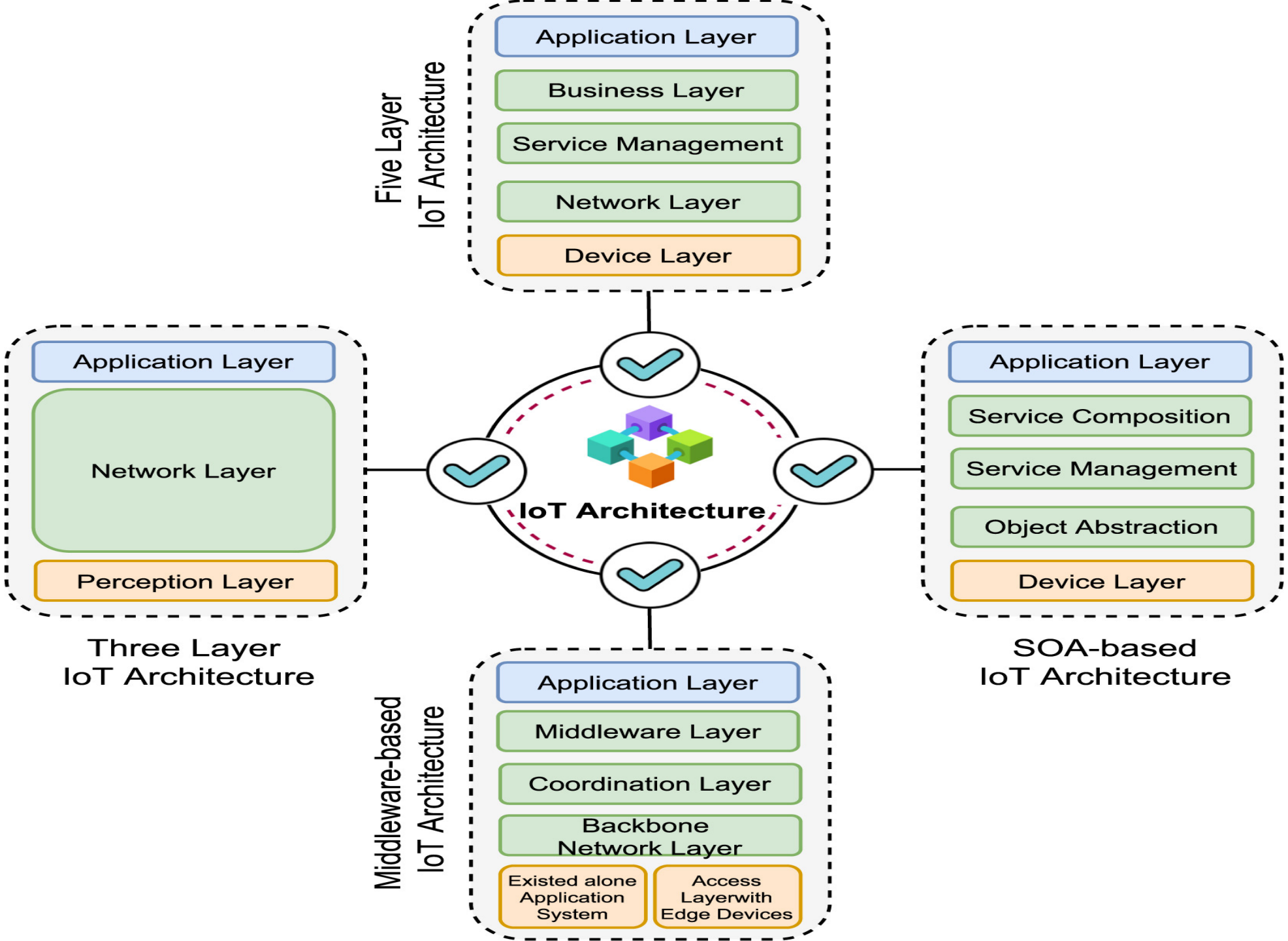
Advantages of Edge Computing

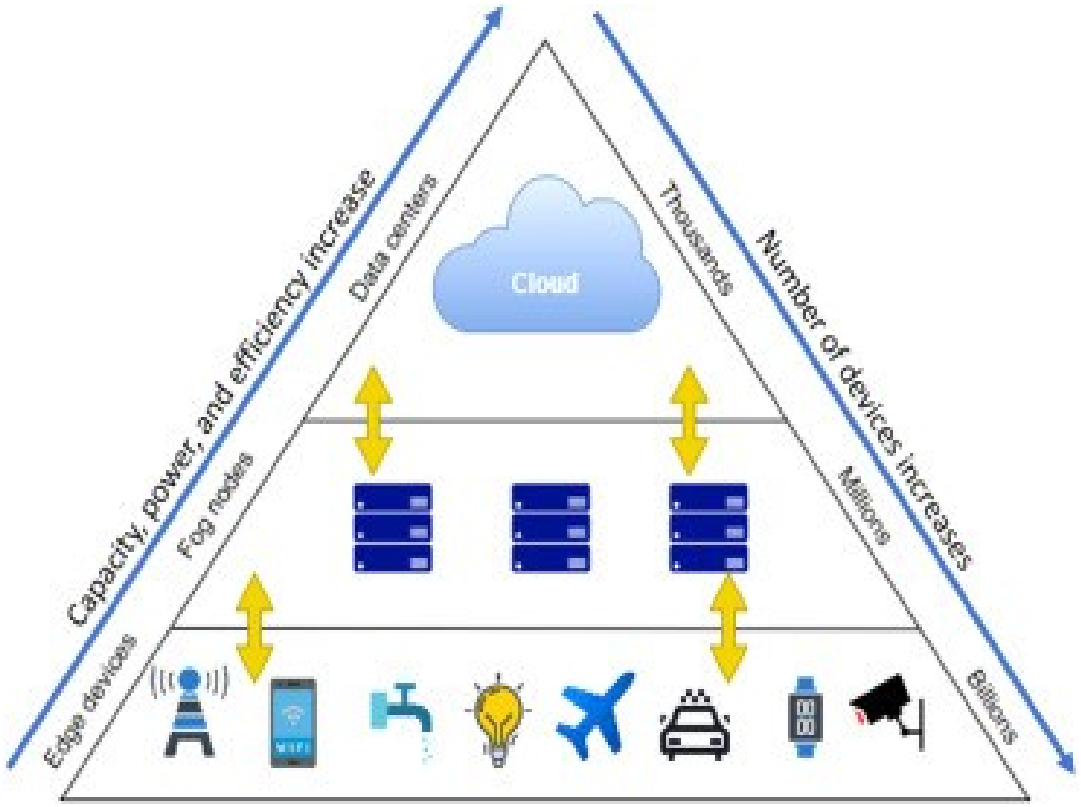
- Reduced Latency
  - Augmented reality
  - Cognitive assistance
  - 51% better response time
- Bandwidth Reduction
  - Up to 67% for BW hungry applications
- Security
- Isolation
  - Disaster resilience
- Energy Efficiency
  - Up to 42 % Reduction
  - Active time to Idle time
  - Type of apps
- Proximity Services
  - Device 2 Device
- Context Information
  - Location based
- Reliability
- Data sovereignty
  - Geofencing

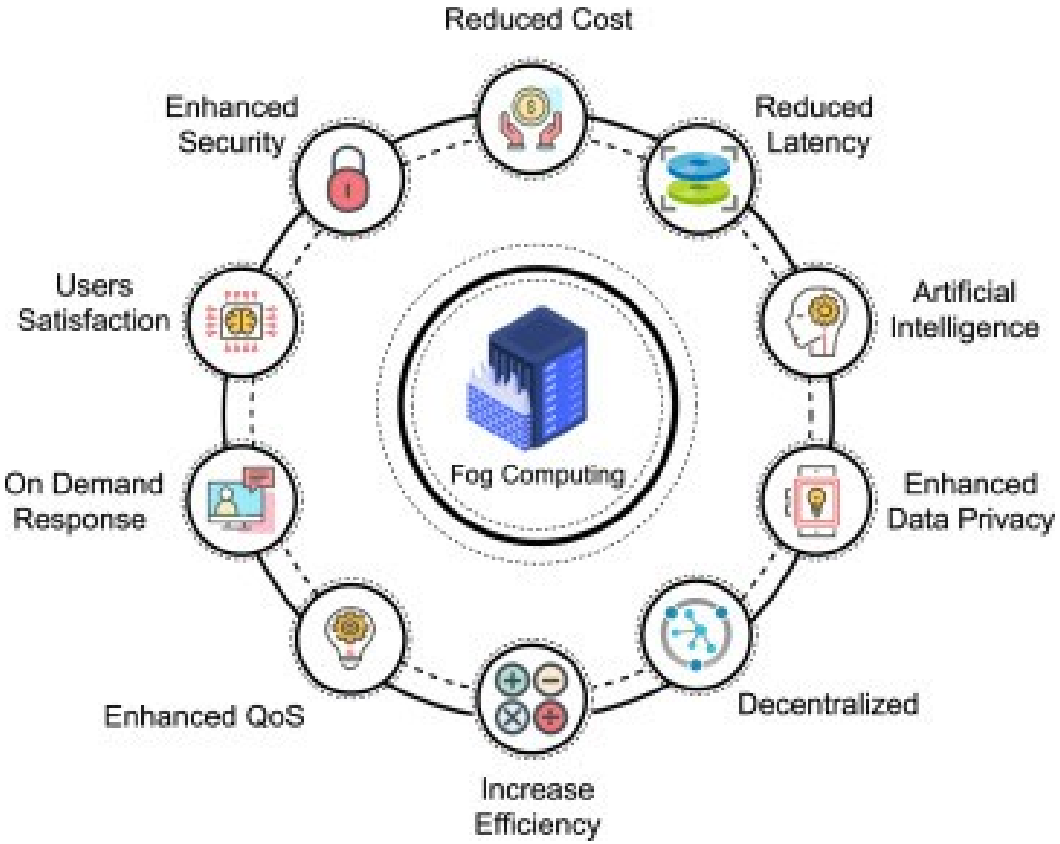
Comparing Mobile Computing Architectures

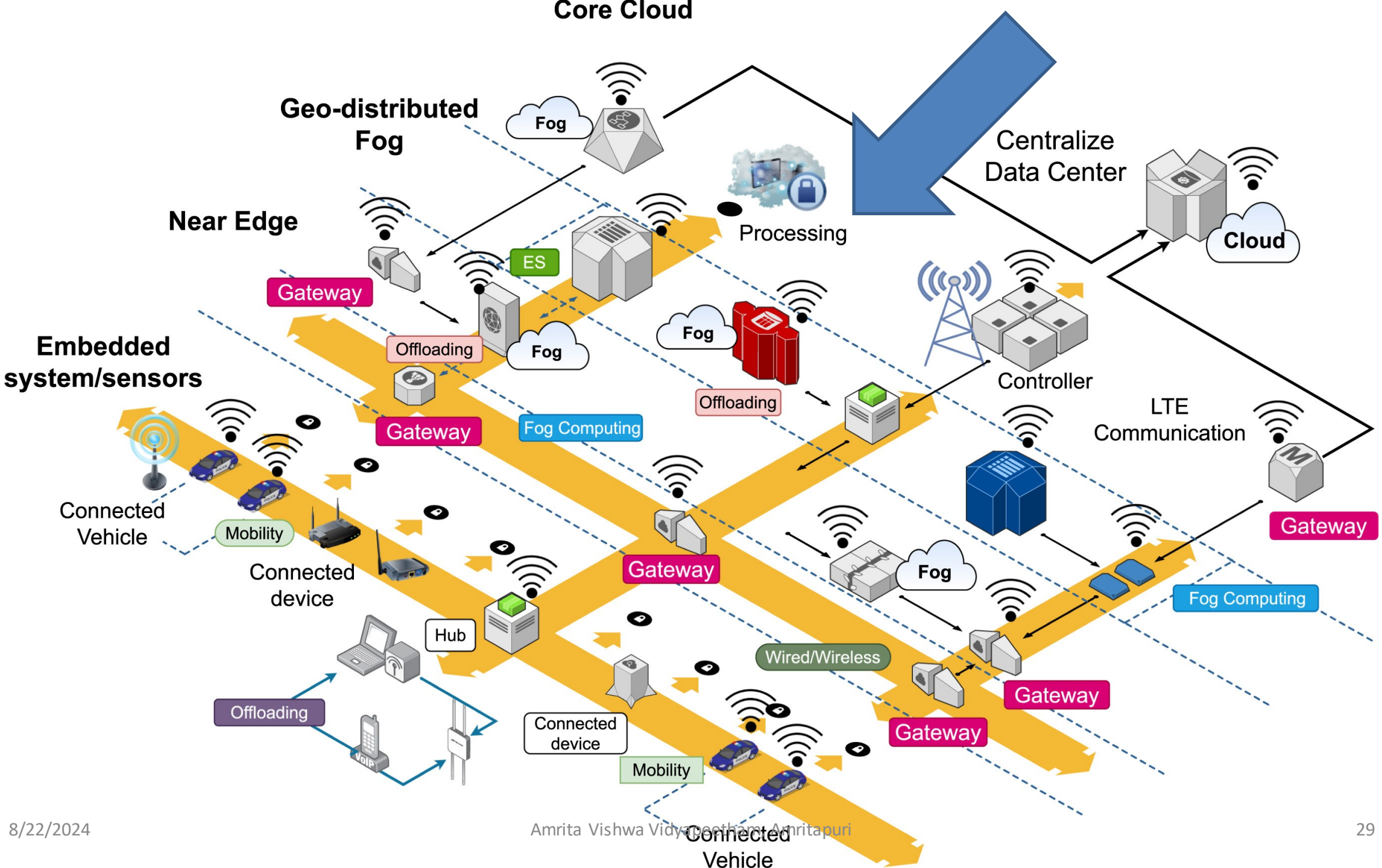
	MCC	MEC	FOG	CLOUDLET
Hierarchy	2 tiers	3 tiers	3 or more tiers	3 tiers
Latency	High	Low	Low	Low
Ownership	Cloud providers	Mobile operators	Fog node owners	Home or Local Business
Population	Large	Medium	Small	Small
Location	Data centers	RAN	Between Device and Data Center	Device or between Device and Data Center
Context awareness	No	Yes	Yes	Maybe
Cooperation	No	No	Yes	No

Courtesy: Wang et al, 2017


Five Layer
IoT Architecture
Application Layer
Business Layer
Service Management
Network Layer
Device Layer
Application Layer
Network Layer
Perception Layer
Three Layer
IoT Architecture
IoT Architecture
Application Layer
Service Composition
Service Management
Object Abstraction
Device Layer
SOA-based
IoT Architecture
Middleware-based
IoT Architecture
Application Layer
Middleware Layer
Coordination Layer
Backbone
Network Layer
Existed alone
Application
System
Access
Layerwith
Edge Devices

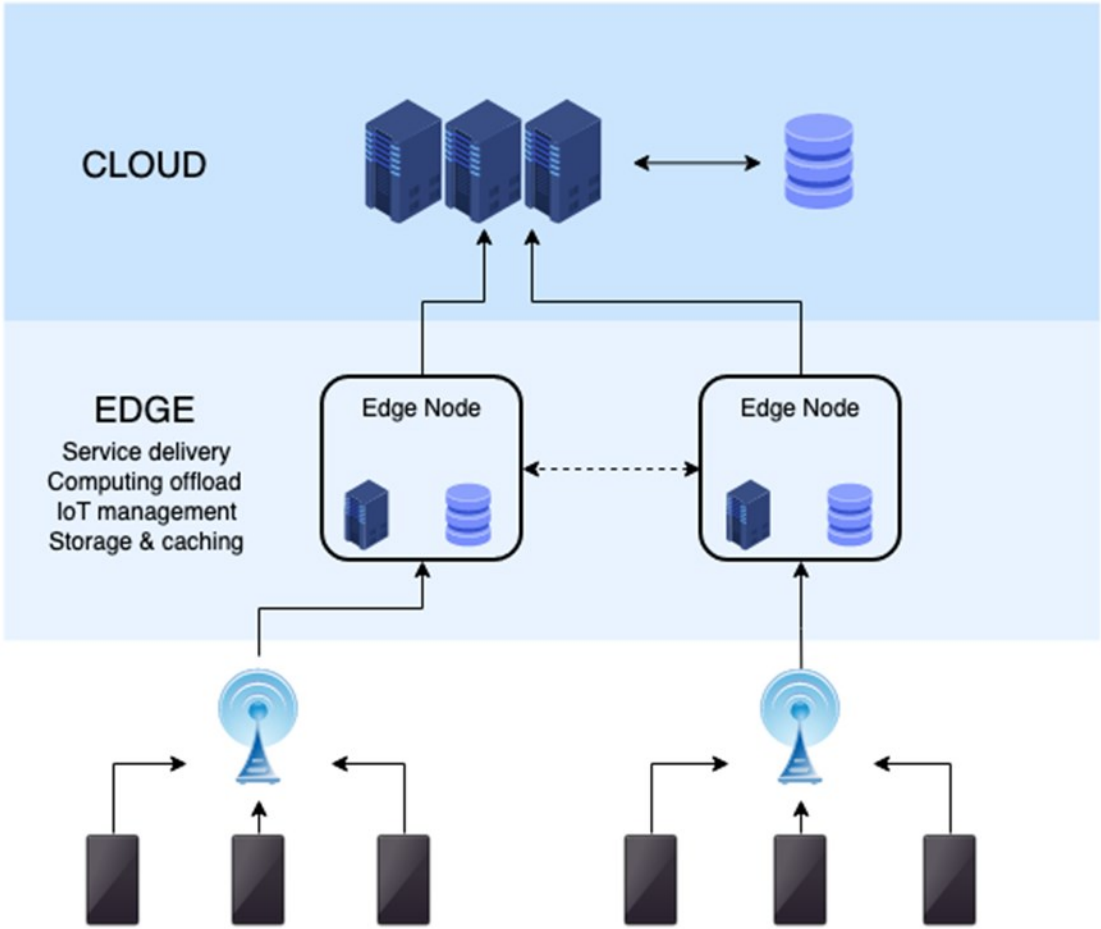

Capacity, power, and efficiency increase
Data centers
Fog nodes
Edge devices
Cloud
Thousands
Millions
Billions
Number of devices increases


Reduced Cost
Enhanced Security
Reduced Latency
Users Satisfaction
Artificial Intelligence
Fog Computing
On Demand Response
Enhanced Data Privacy
Enhanced QoS
Decentralized
Increase Efficiency

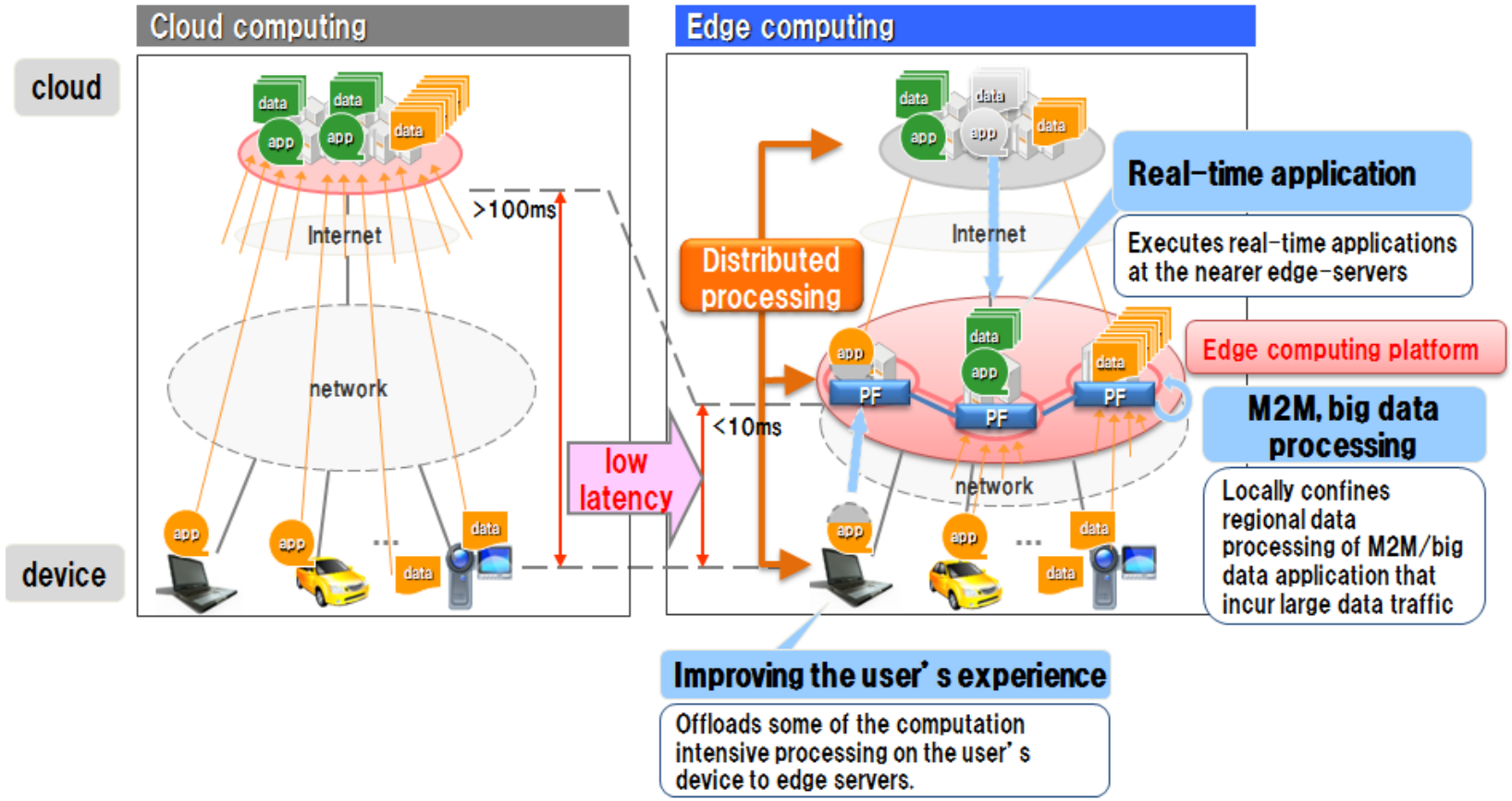

Core Cloud
Geo-distributed Fog
Fog
Centralize Data Center
Cloud
Near Edge
Processing
ES
Gateway
Offloading
Fog
Embedded system/sensors
Fog
Offloading
Controller
Gateway
Fog Computing
LTE Communication
Gateway
Connected Vehicle
Mobility
Connected device
Gateway
Fog
Fog Computing
Hub
Wired/Wireless
Gateway
Gateway
Offloading
Connected device
Mobility
Connected Vehicle

Edge computing

- Edge Computing brings processing power closer to where the data is being generated.
- It does not send generated data to a central location directly.
- The data can be processed on the device where the sensors are connected or a gateway device in close proximity.
- **Fog/Edge** Computing and storage systems are located closer to the edge to eliminate latency.
- Fog/Edge computing can be integrated with cloud computing if needed.
- They can work as an extension of cloud computing

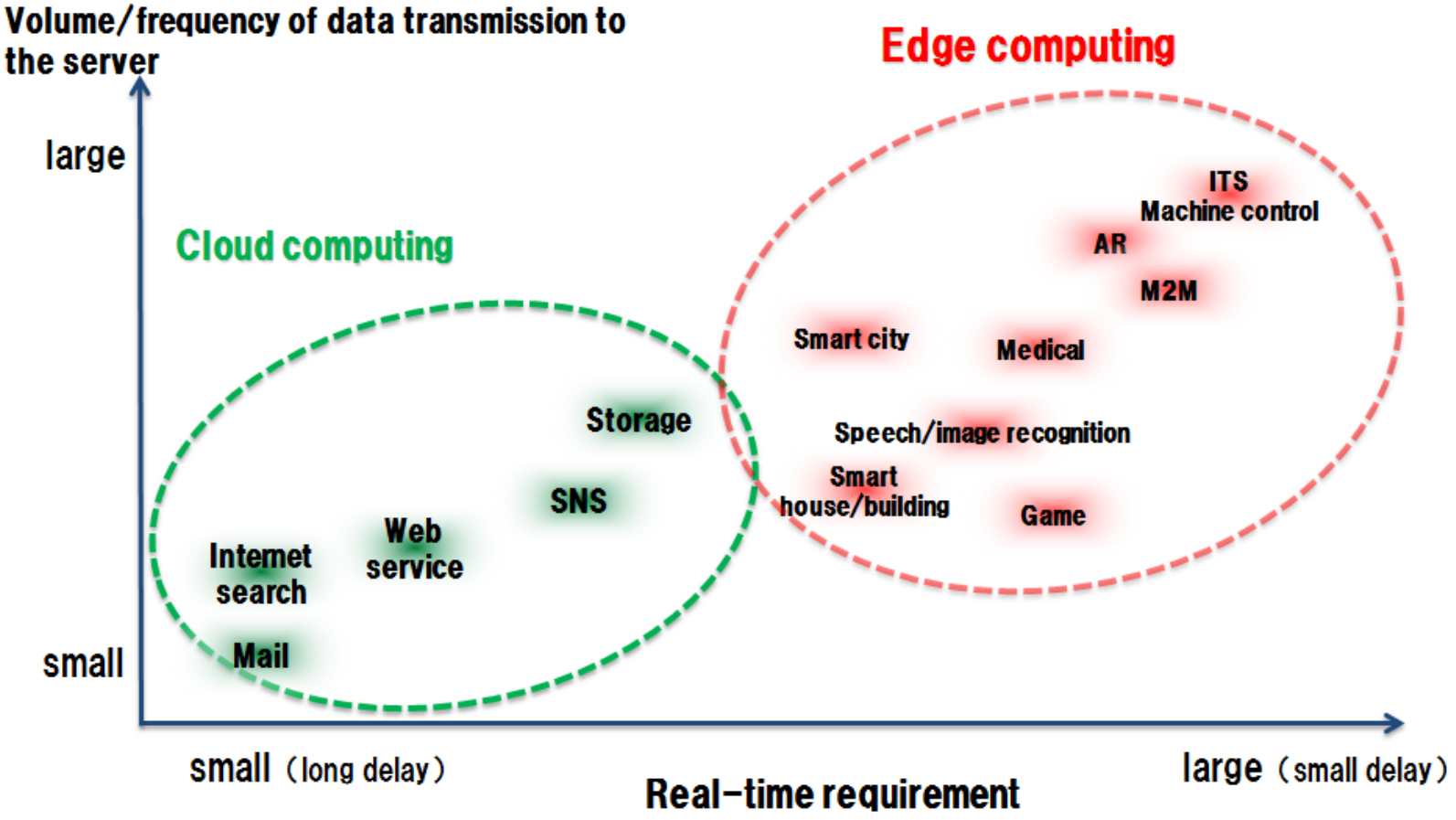

CLOUD
EDGE
Service delivery
Computing offload
IoT management
Storage & caching
Edge Node
Edge Node

Cloud Vs Edge



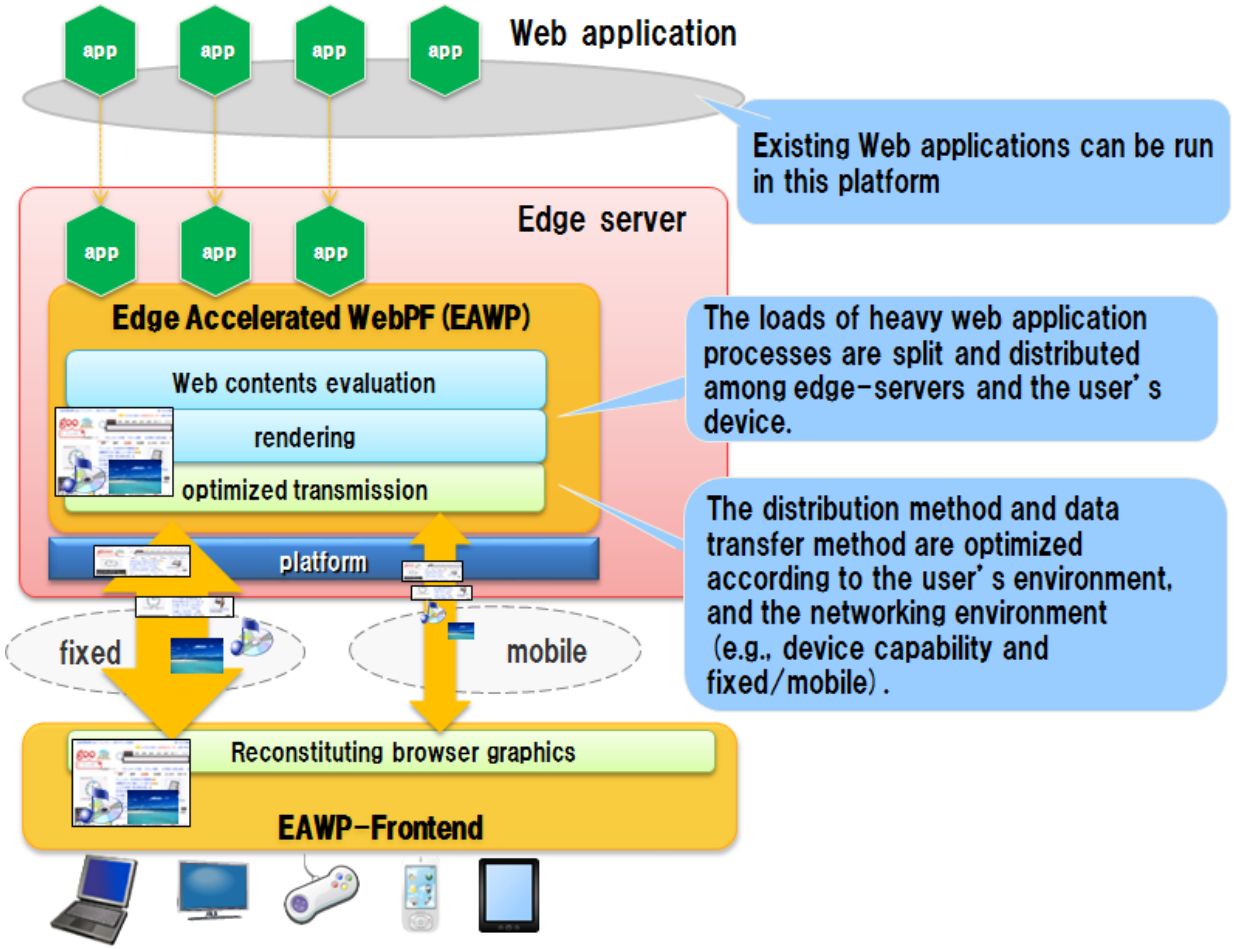
Courtesy: NTT

Edge Computing Applications



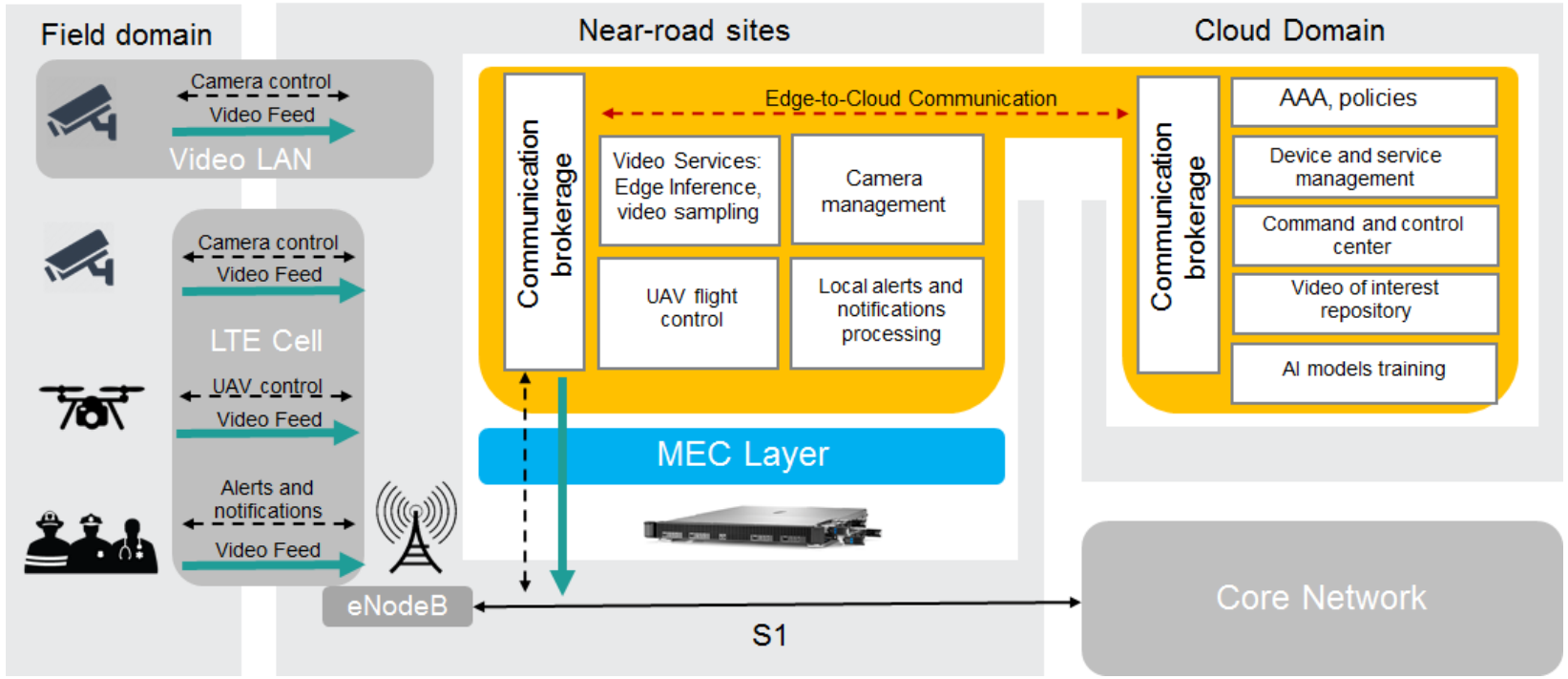
Courtesy: NTT

Edge Accelerated Web Platform



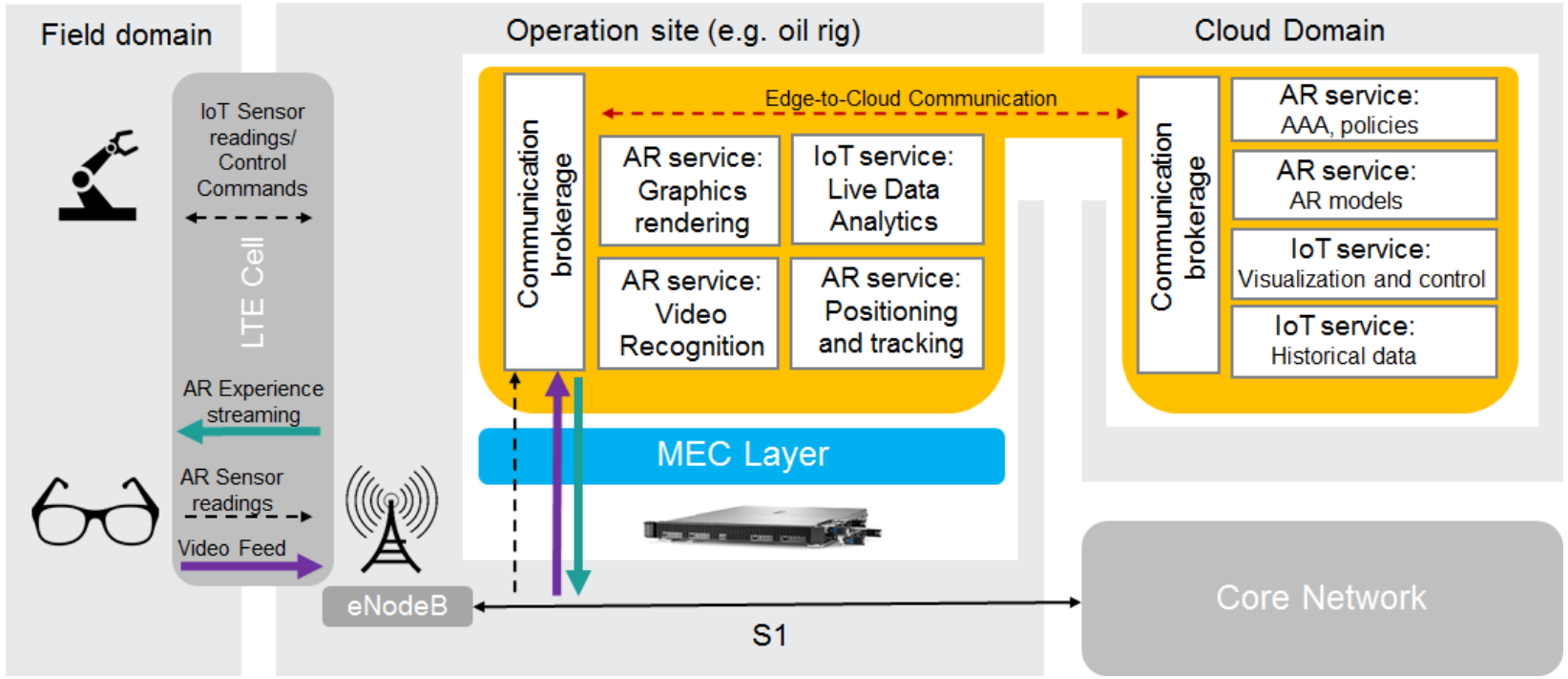
Courtesy: NTT

Smart City Surveillance



Courtesy: AWS

Augmented Reality



Courtesy: AWS

Mobile Edge Computing (MEC)

- We have mobile base-stations in many locations in order to fail-safe relay mobile communication.
- These facilities are being recommended for doing the edge data processing and analytics.

Resource-Constrained Sensors Behind Fog Devices

- Many sensors do not have enough compute, memory, and storage power in order to have their own IPv6 address.
- Hence, they can hide behind an edge device which has the power to have its own IPv6 address.
- It is, therefore, pertinent to configure ground-level and resource-constrained sensors and actuators behind the edge device.

The Use Cases of Fog/Edge Computing

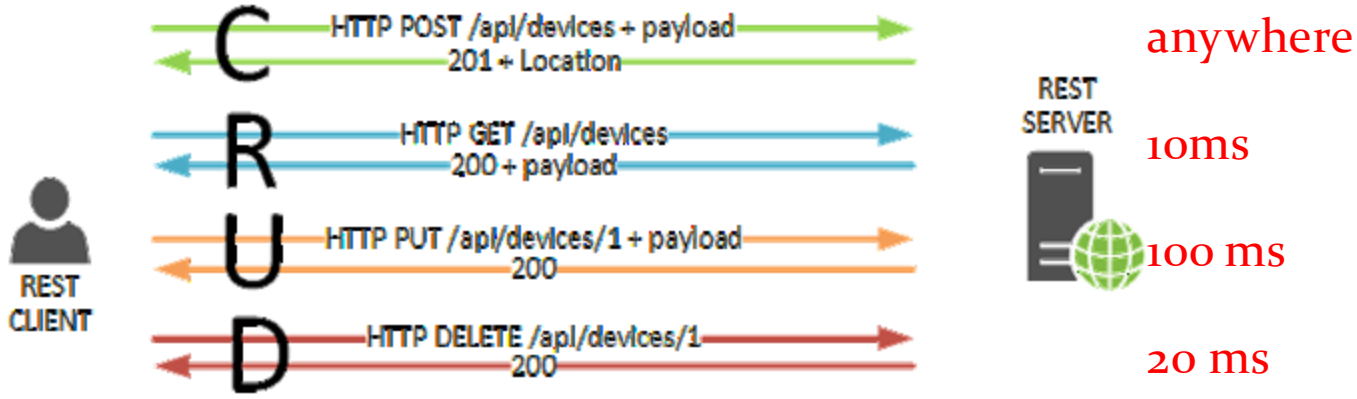
- The rapid growth of personal, social, and professional devices at our daily environments has seeded this inimitable computing style.
- The communication becomes wireless, sensors and devices are heterogeneous and large in number, geo-distribution becomes the new normal, the interconnectivity and interactions among various participants emit a lot of data, and so on.
- The amount of data getting generated and gathered at the edge of the network is really massive in volumes.

Examples

- Fog/Edge paradigm is used in commercial airlines since the data is processed on aircraft.
- It is used in spacestations as data processing is local
- It is used in ships, submarines, mines, remote locations
- Unmanned Aerial Vehicles (UAV), and orbital satellites use cloud computing (and not Fog/Edge) since they are operated from a central remote location based on the streamed sensor data from sensors or cameras.

Web Applications on the cloud

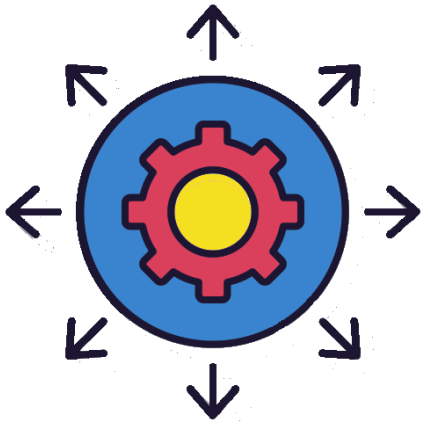
- Collection of independent stateless handlers
  - REST (flexibility)
  - Scalability
  - Fault tolerance



- What Enables this model?
  - **Shared storage layer**

CloudPath

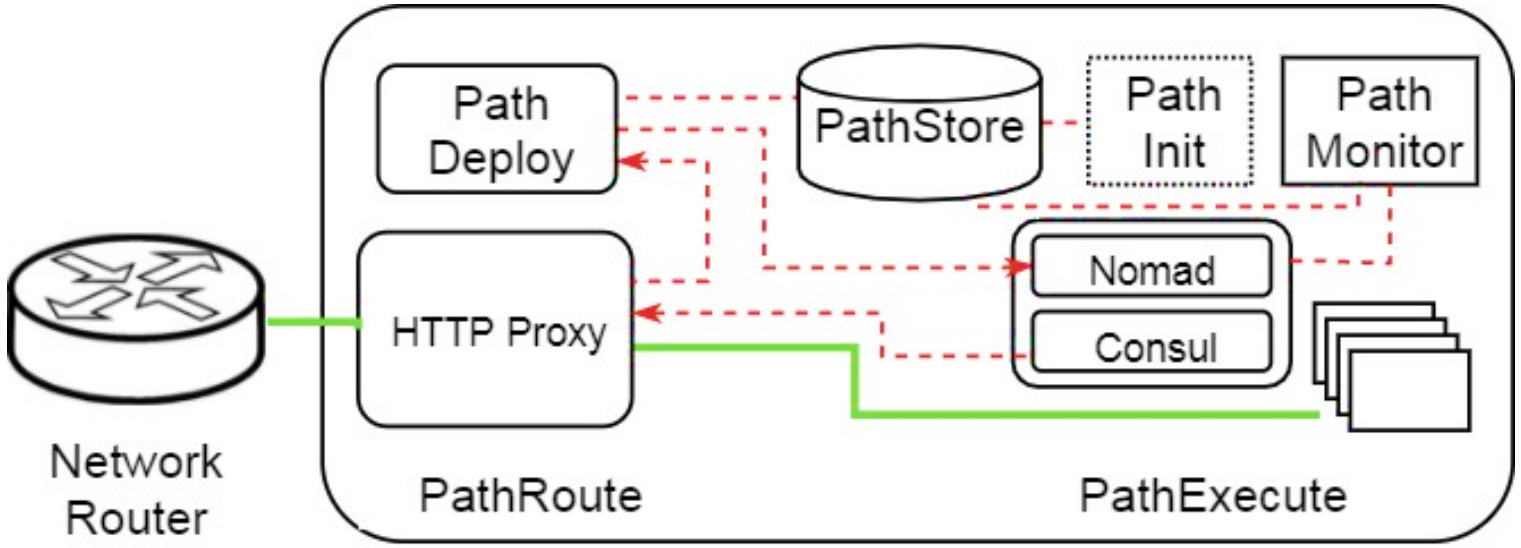
- A platform that enables the execution of 3rd party applications on a progression of datacenters deployed along the geographical span of the network



- Separation between application code and data
  - Developers: organize applications as a collection of stateless functions
  - CloudPath: on-demand replication of code and data

- Provides a common runtime on all cloud nodes

CloudPath



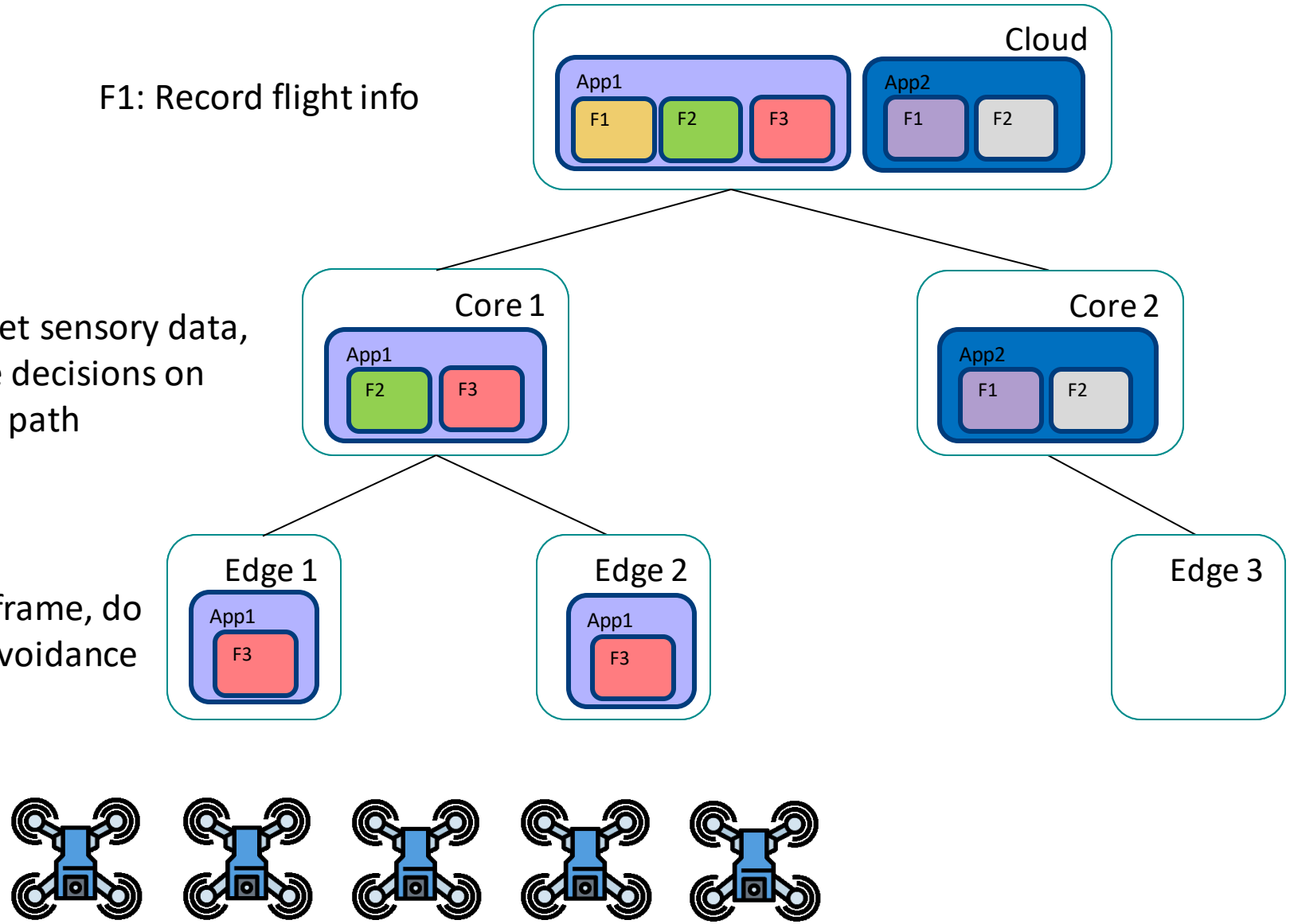
- Each node consists of:
  - Common execution environment --> **PathExecute**
  - On-Demand deployment --> **PathDeploy**
  - Routing --> **PathRoute**
  - Common storage layer --> **PathStore**
  - Monitoring --> **PathMonitor**

Example

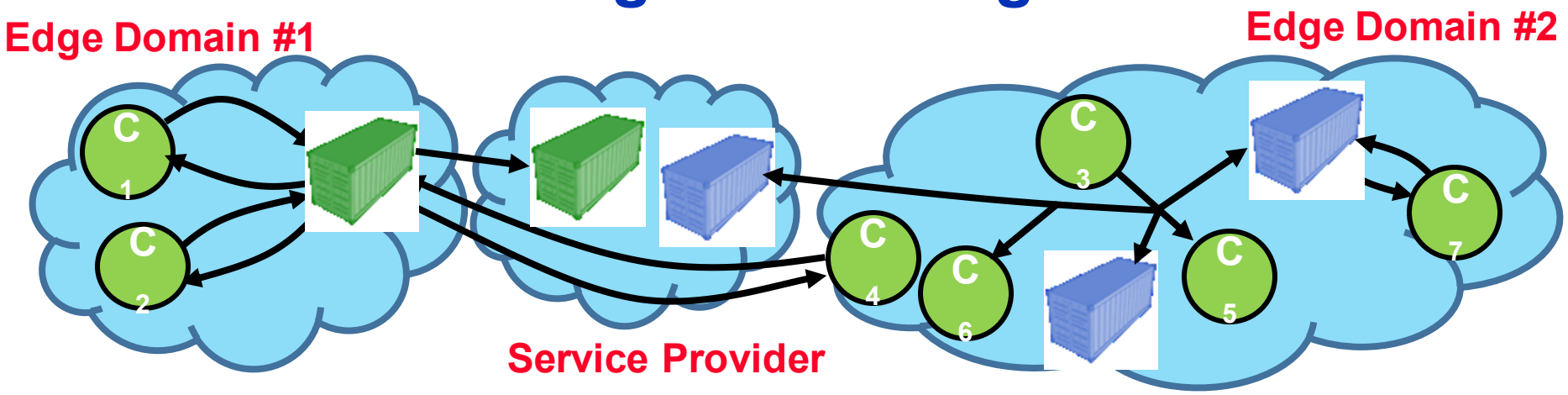
F1: Record flight info

F2: Get sensory data, make decisions on flight path

F3: Get frame, do object avoidance

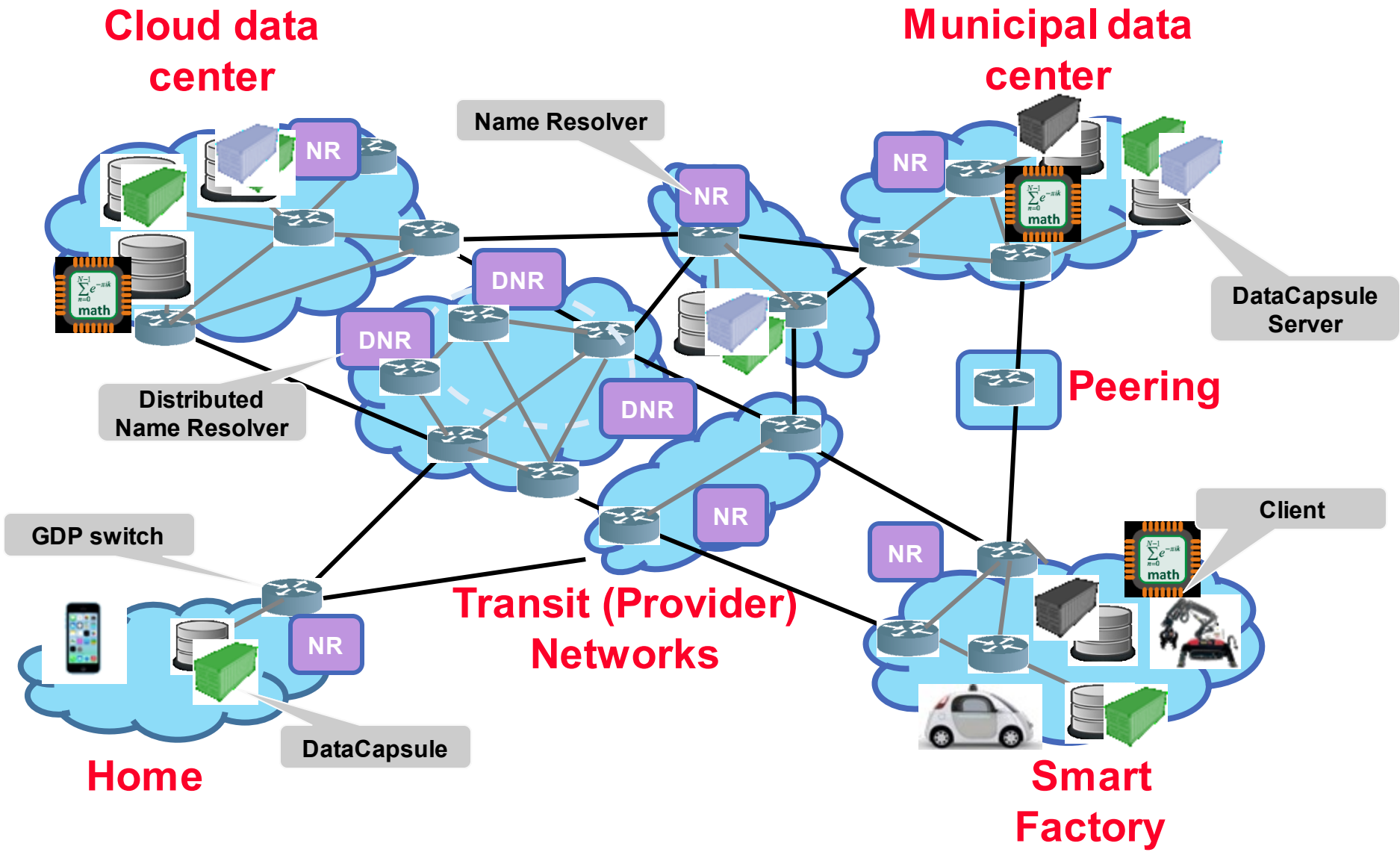


Global Data Plane (GDP) and the Secure Datagram Routing Protocol



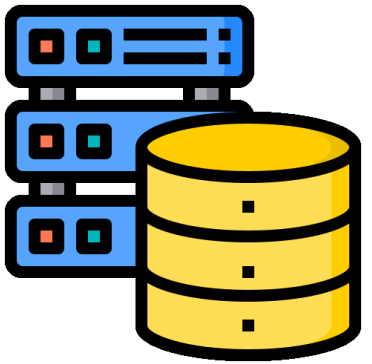
- Flat Address Space Routing
  - Route queries to DCs by names, independent of location (e.g. no IP)
  - DCs move, network deals with it
  - Short-term Channels ("μ-SSL channels")
- Black Hole Elimination: Delegation of Names
  - Only servers authorized by owner of DC may advertise DC service
- Routing only through domains you trust!
  - Secure Delegated Flat Address Routing

- Secure Multicast Protocol
  - Only clients/DC storage servers with proper (delegation) certificates may join
- Queries (messages) are *Fibers*
  - Self-verifying chunks of DataCapsules
  - Writes include appropriate credentials
  - Reads include proofs of membership
- Incremental deployment as an overlay
  - Prototype tunneling protocol ("GDPinUDP")
  - Federated infrastructure w/routing certificates


Cloud data center
Municipal data center
Name Resolver
NR
DNR
DataCapsule Server
Distributed Name Resolver
Peering
GDP switch
Client
Transit (Provider) Networks
DataCapsule
Home
Smart Factory

Pathstore: A Distributed Storage Layer For The Edge

- PathStore is a data storage layer for a multi-tier cloud architecture

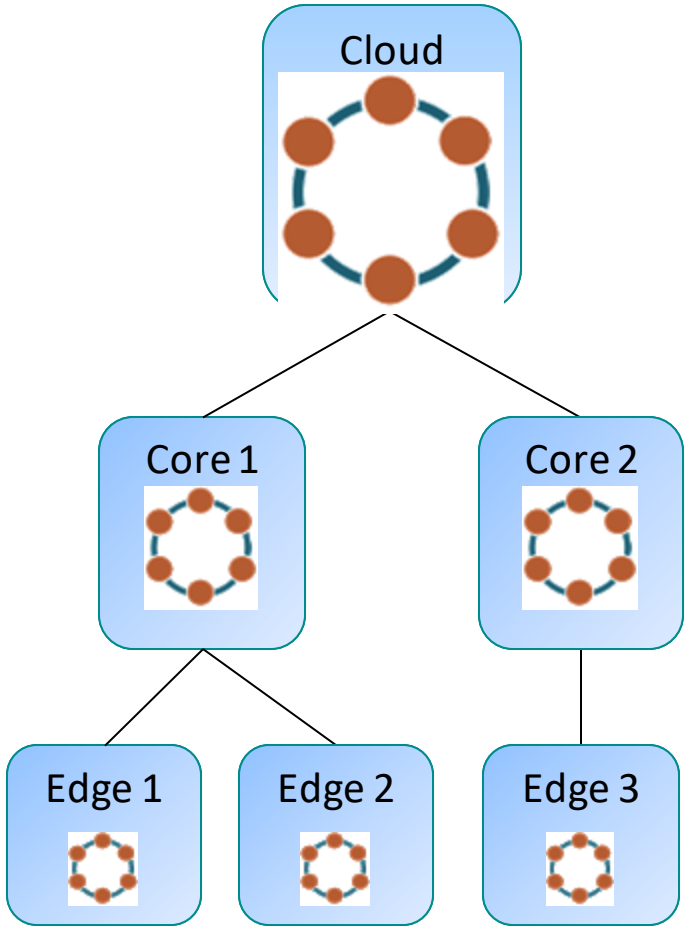


- Hierarchy of data stores
  - Each PathStore node runs an independent Cassandra ring and PathStore copies data between rings

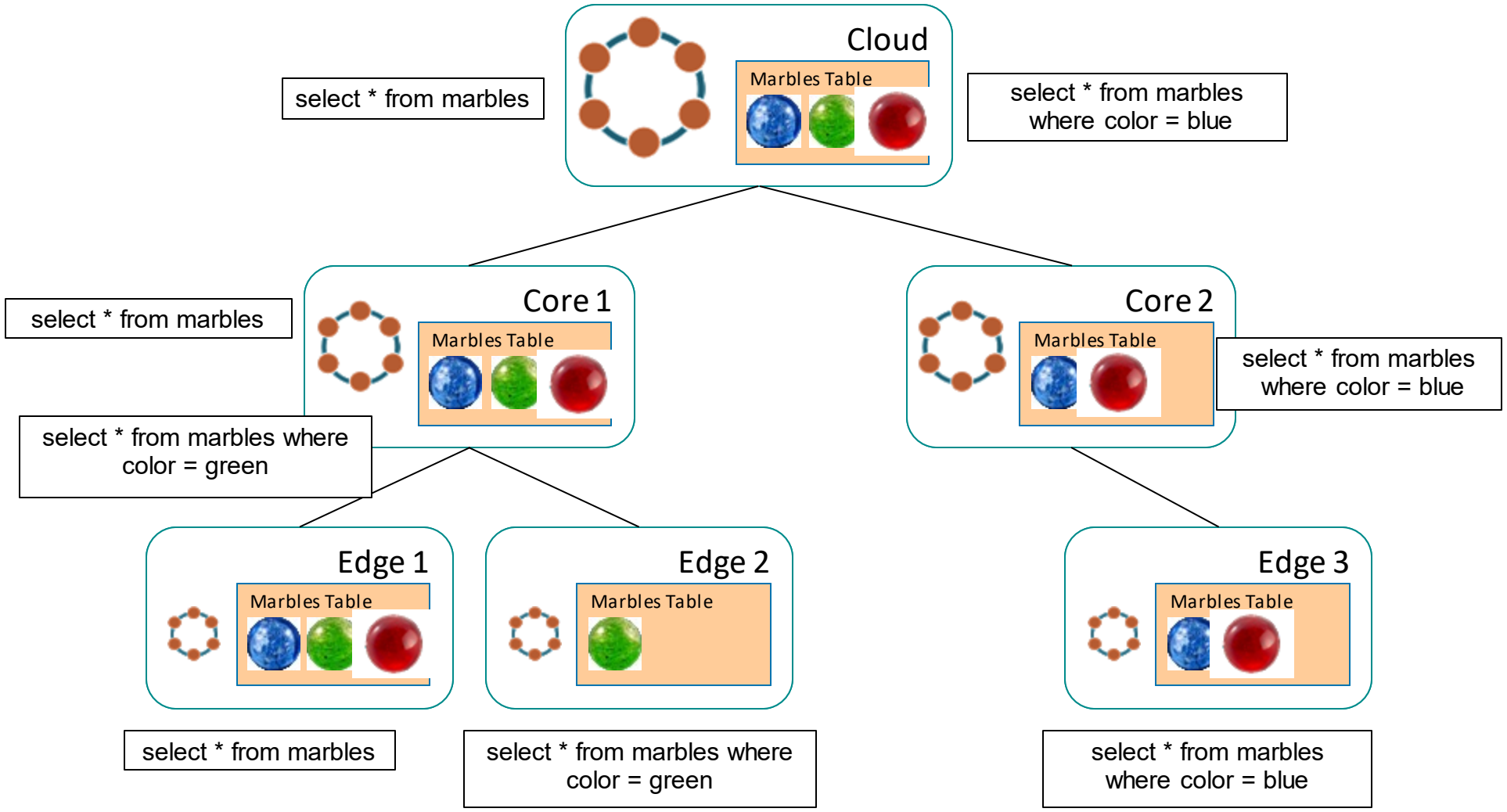
- Support Eventual consistency
  - Extension called SessionStore supports session consistency

PathStore Design

- Data store at the root of the hierarchy is persistent
- Other act as temporary partial replicas
  - Data replicated on-demand
  - All reads and writes executed against local replica
- CQL based interface
- Row-level replication



Example


Cloud
Marbles Table
select * from marbles
select * from marbles where color = blue
Core 1
Marbles Table
select * from marbles
select * from marbles where color = green
Core 2
Marbles Table
select * from marbles where color = blue
Edge 1
Marbles Table
select * from marbles
Edge 2
Marbles Table
select * from marbles where color = green
Edge 3
Marbles Table
select * from marbles where color = blue

SessionStore

- SessionStore ensures session consistency on a top of otherwise eventually consistent replicas.
- Enforces session consistency by grouping related data accesses into a session, and using a session-aware reconciliation algorithm to reconcile only the data that is relevant to a session when switching between data centers.
  - Only replicates relevant data (row level)
  - Only replicates to the relevant destination

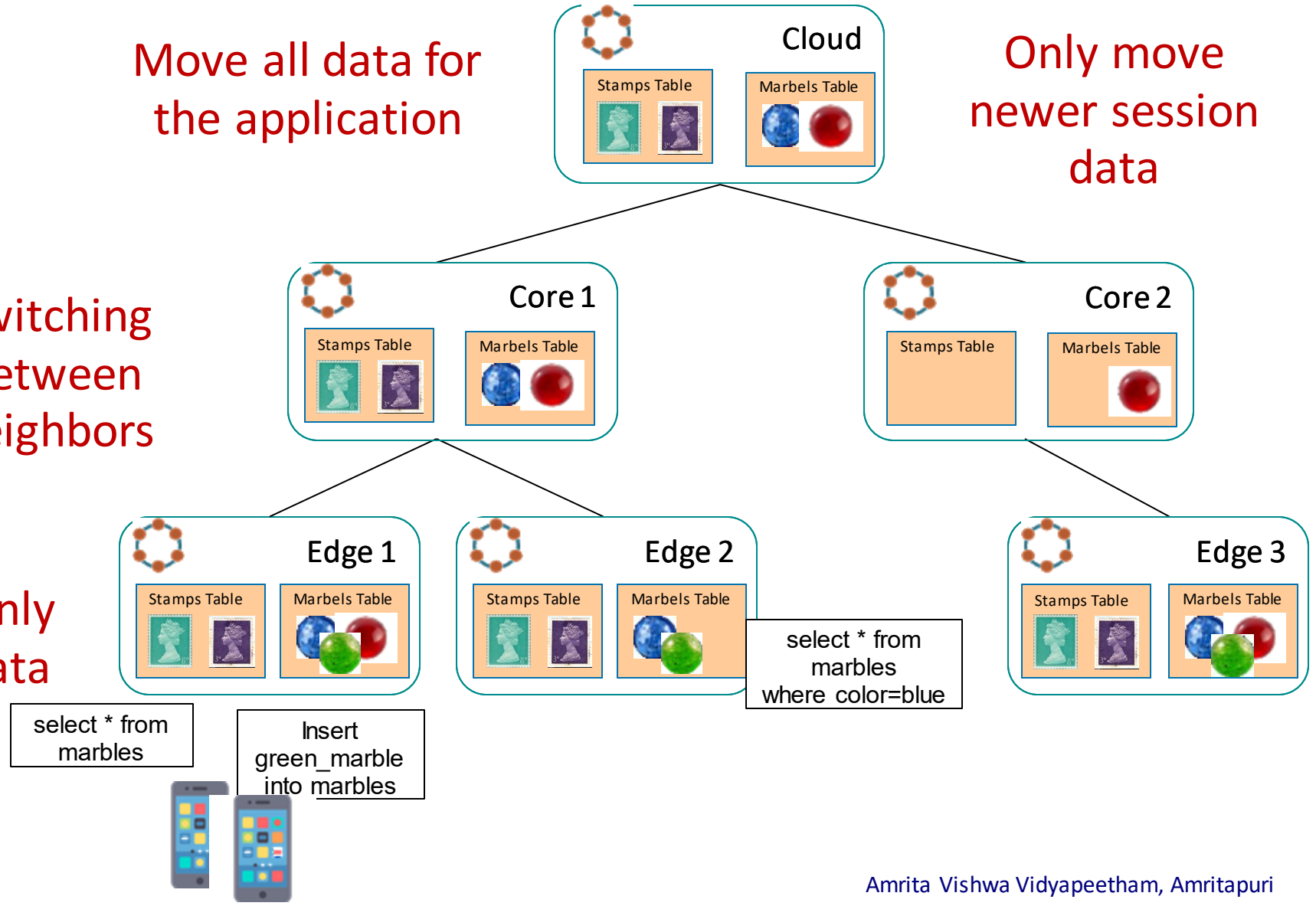
SessionStore

Move all data for the application

Only move newer session data

Switching between neighbors

Transfer only session data



Session Tracking

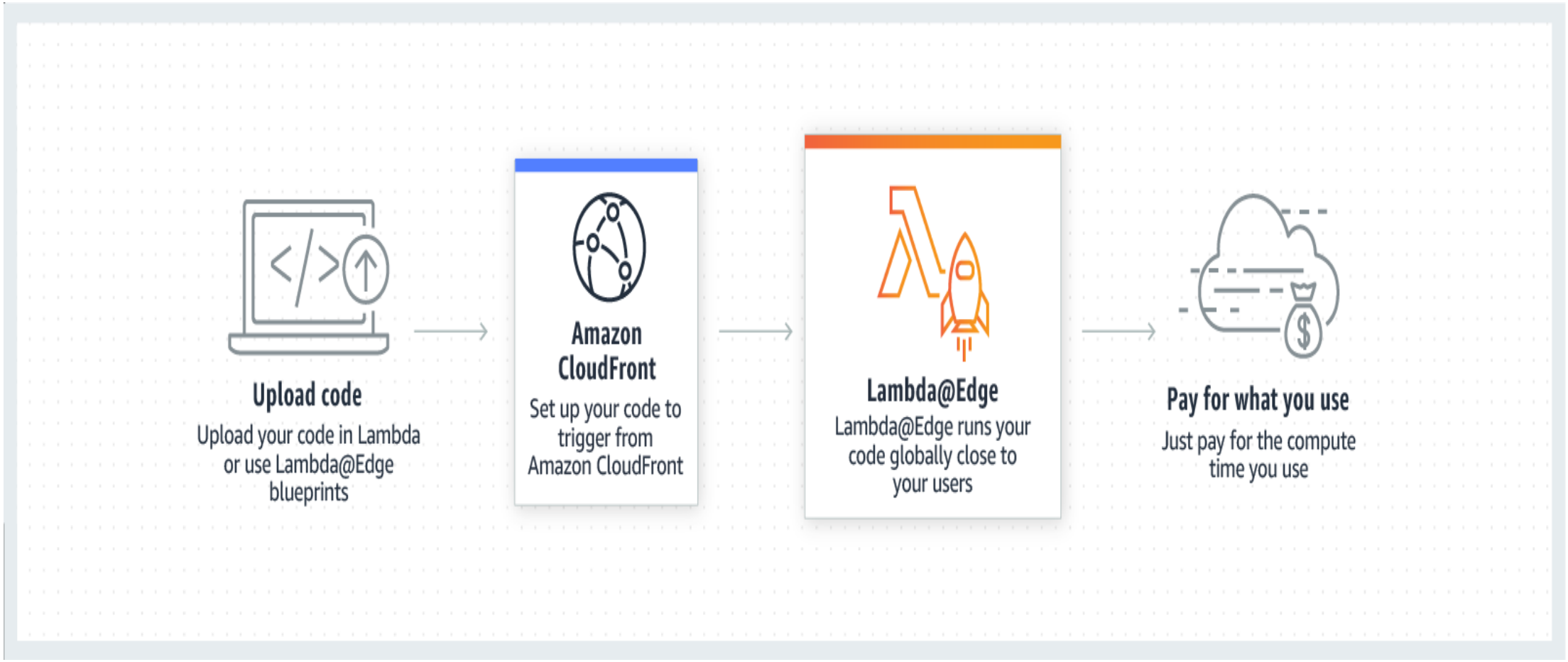
- We identify each session using a Session Token, or **stoken**.
- **Stoken** consists of a four fields:
  - A unique session id (SID)
  - timestamp
  - current replica
  - status
- The stoken is encrypted and signed to prevent forging and misrepresentation.

- To keep track of data related to a session, a **command-Cache** is added to each PathStore replica that stores all the *CQL SELECT* statements that were run on behalf of a session

Network Requirements

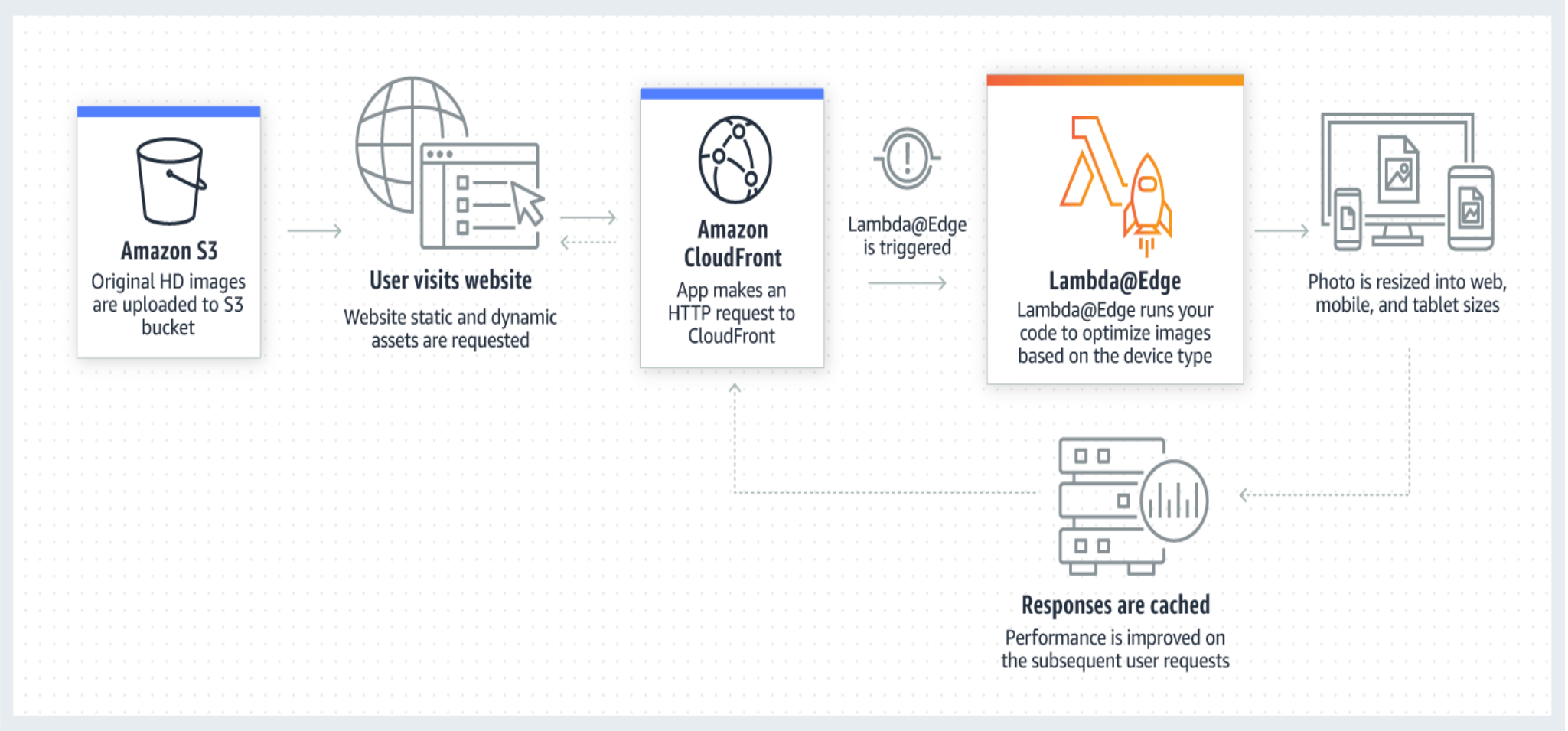
- Routing requests for applications based on functions
- Synchronizing clocks between Datacenters
- Providing locking service for data
  - Better guarantees for data

AWS Lambda @ Edge



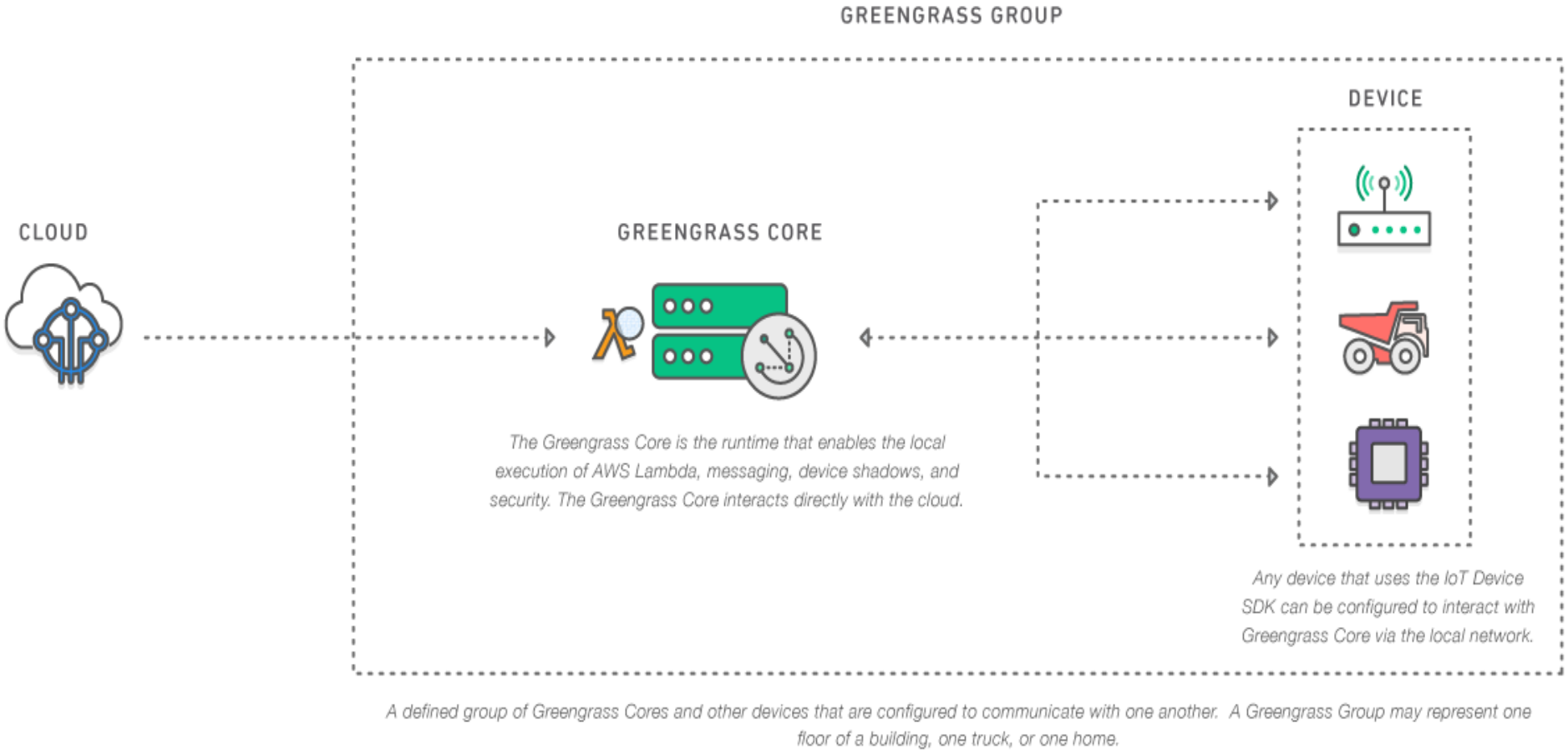
Courtesy: AWS

Image Processing with Lambda @ Edge



Courtesy: AWS

AWS IoT Greenglass



Courtesy: AWS

Table 1.2 Comparison of cloud, fog and edge computing concepts [13]

Feature	Cloud computing	Fog/Edge computing
Access	Wired or wireless	Wireless
Access to the service	Through server	At the edge device
Availability	Mostly available	Mostly volatile
Bandwidth usage	High	Low
Capacity—Computing	Higher	Lower
Capacity—Storage	Higher	Lower
Connectivity	Internet	Many protocols (Fig 1.5)
Content distributed to	Edge device	Anywhere
Content generator	Man made	Sensor made
Content generation at	Central server	Edge device
Control	Centralized	Distributed
Data analysis	Long term	Instant/Short term
Data processing	Far from data source	Closer to data source
Latency	High	Minor
Location of resources (i.e. processing and storage)	Center	Edge
Scalability	High	Low
Security	Weaker	Stronger
Mobility	Limited	Supported
Number of users	Millions	Billions
Virtual infrastructure location	Enterprise server	Enterprise/User devices

	Edge computing	Fog computing
Location of data collection, processing, storage	Network edge, edge devices	Near-edge and core networking, network edge devices and core networking devices
Handling multiple IoT applications	Unsupported	Supported
Resource contention	Serious	Slight
Focus	Things level	Infrastructures level