Week 1 Lecture 1

45 mins

Overview of Cloud Computing

Dr. Rajiv Misra, Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Contents

- Overview of Cloud Computing
  - Brief introduction to Cloud Computing with focus on the new aspects in today's Clouds
  - Distinguish Cloud Computing from the previous generation of distributed systems
  - Brief History and the current demand of Cloud Data Centers
  - Features of cloud computing
  - Private and Public clouds

The Hype of Cloud: Forecasting

- Gartner in 2009 – Cloud computing revenue will soar faster than expected and will **exceed $150 billion** by 2013. It will represent 19% of IT spending by 2015.
- IDC in 2009 - Spending on IT cloud services will triple in the next 5 years, reaching **$42 billion**.
- Forrester in 2010 – Cloud computing will go from **$40.7 billion** in 2010 to **$241 billion** in 2020.
- Companies and even federal/state governments using cloud computing now: **fbo.gov**

Scalable Computing Over the Internet

- Evolutionary changes that have occurred in **distributed and cloud computing** over the past 30 years, **driven by applications with variable workloads and large data sets .**
- Evolutionary changes in machine architecture, operating system platform, network connectivity, and application workload.
- **Distributed computing** system uses multiple computers to solve large-scale problems over the Internet. Thus, **distributed computing becomes data-intensive and network-centric.**
- **The emergence of computing clouds** instead demands high-throughput computing (HTC) systems built with distributed computing technologies.
- **High-throughput computing (HTC)** appearing as computer clusters, service-oriented architecture, computational grids, peer-to-peer networks, Internet clouds, and the future Internet of Things.

Many Cloud Providers

- AWS: Amazon Web Services
  - EC2: Elastic Compute Cloud
  - S3: Simple Storage Service
  - EBS: Elastic Block Storage
- Microsoft Azure
- Google Compute Engine/AppEngine
- Rightscale, Salesforce, EMC, Gigaspaces, 10gen, Datastax, Oracle, VMWare, Yahoo, Cloudera
- And 100s more…

Customers Save: Time and Money

- "With AWS, a new server can be up and running in **three minutes** compared to **seven and a half weeks** to deploy a server internally and a **64-node Linux cluster** can be online in five minutes (compared with three months internally."
- "With Online Services, reduce the IT **operational costs** by roughly **30%** of spending"
- "A private cloud of virtual servers inside its datacenter has saved nearly **crores of rupees annually**, because the company can share computing power and storage resources across servers."
- 100s of startups can harness large computing resources without buying their own machines.

Virtualization and the Cloud

- **Virtual workspaces**:

  –An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols

  –Resource quota (e.g. CPU, memory share)

  –Software configuration (e.g. O/S, provided services)

- **Virtual Machines** (VMs):

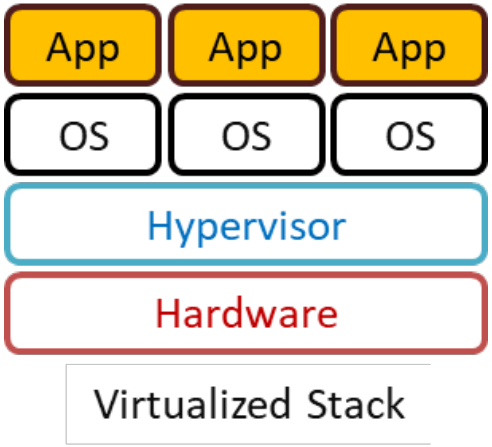
  –Abstraction of a physical host machine

  –Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs

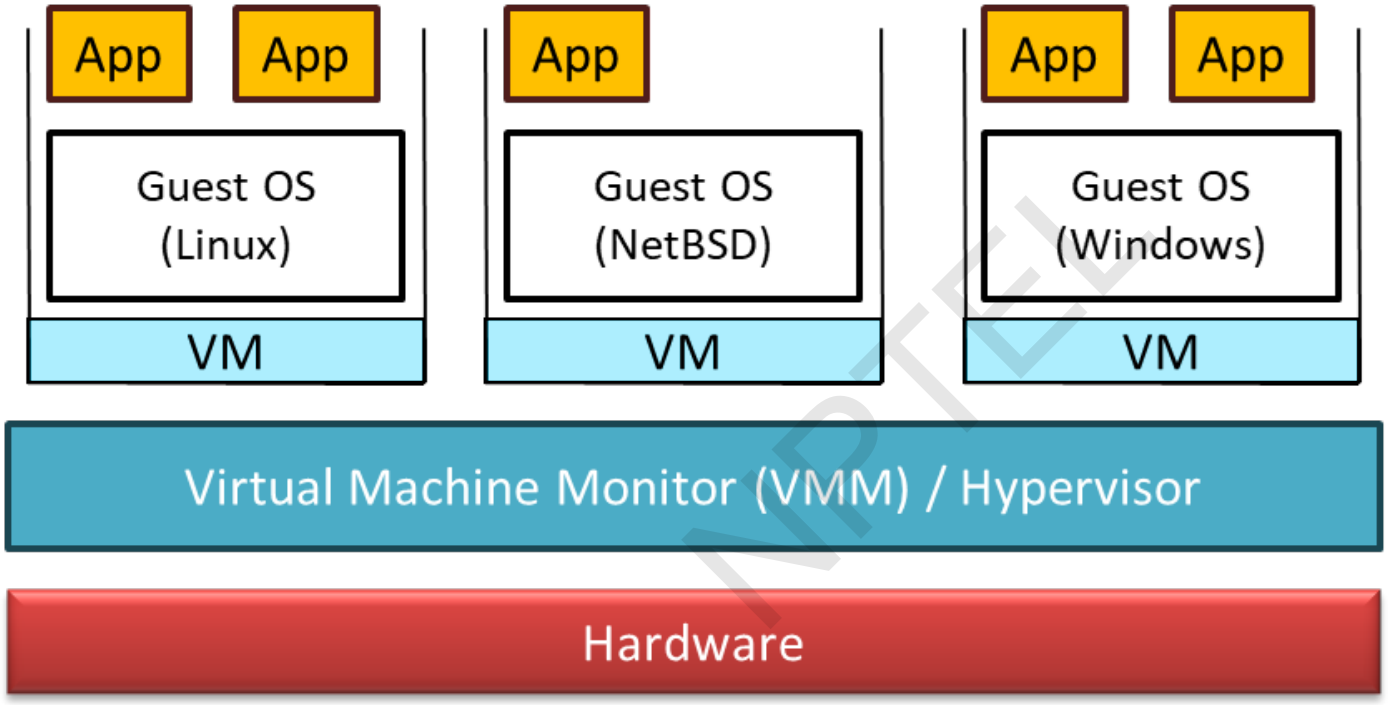
  –VMWare, Xen, etc.

- Provide infrastructure **API**:

  –Plug-ins to hardware/support structures



Virtualization and the Cloud



VM technology allows multiple virtual machines to run on a single physical machine.

Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

Virtualization and the Cloud

Advantages of virtual machines:

- Run operating systems where the physical hardware is unavailable
- Easier to create new machines, backup machines, etc.
- Software testing using “clean” installs of operating systems and software
- Emulate more machines than are physically available
- Timeshare lightly loaded systems on one host
- Debug problems (suspend and resume the problem machine)
- Easy migration of virtual machines (shutdown needed or not)
- Run legacy systems!

What is a Cloud?

- Advances in virtualization make it possible to see the growth of Internet clouds **as a new computing paradigm**.
- Dramatic differences between developing software for millions to use **as a service** versus distributing software to run on their PCs.
- **History:**
- **In 1984, John Gage** Sun Microsystems gave the slogan,

  "The network is the computer."
- **In 2008, David Patterson** UC Berkeley said,

  "The data center is the computer."
- Recently, **Rajkumar Buyya** of Melbourne University simply said:

  "The cloud is the computer."
- Some people view **clouds as grids** or **clusters** with changes through virtualization, since clouds are anticipated to process huge data sets generated by the traditional Internet, social networks, and the future IoT.

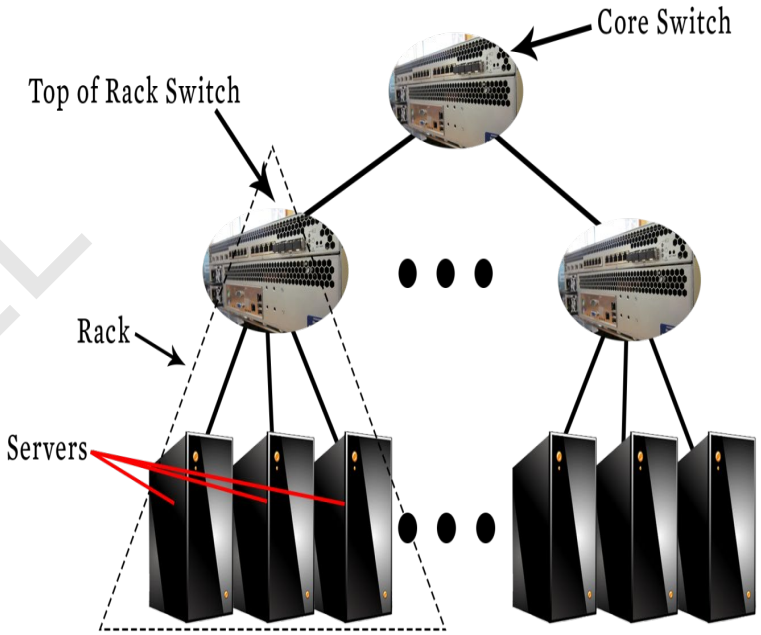
What is a Cloud?

A single-site cloud (as known as "Datacenter") consists of

- Compute nodes (grouped into racks)
- Switches, connecting the racks
- A network topology, e.g., hierarchical
- Storage (backend) nodes connected to the network
- Front-end for submitting jobs and receiving client requests
- (Often called "three-tier architecture")
- Software Services

A geographically distributed cloud consists of

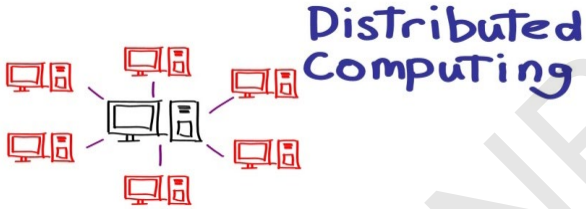
- Multiple such sites
- Each site perhaps with a different structure and services

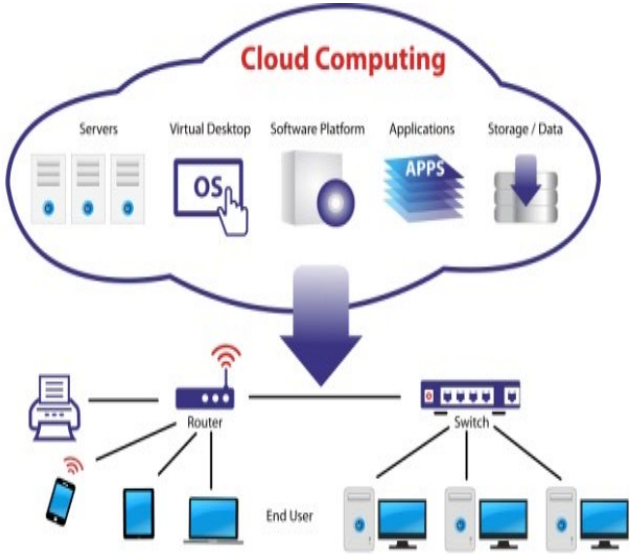


Computing Paradigm Distinctions

- Cloud computing overlaps with distributed computing.

- **Distributed computing:** A ***distributed system*** consists of multiple autonomous computers, having its own memory, communicating through ***message passing***.



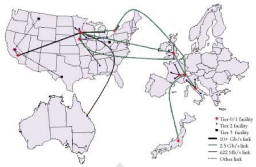


- **Cloud computing:** Clouds can be built with physical or virtualized resources over large data centers that are distributed systems. Cloud computing is also considered to be a form of ***utility computing or service computing.***

“A Cloudy History of Time”

First large data centers:
ENIAC, ORDVAC, ILLIAC
Many used vacuum tubes
and mechanical relays

Open Science Grid



Berkeley NOW Project
- Supercomputers
Server Farms (e.g., Oceano)

Cloud Datacenters

The first datacenters!

Time Sharing Companies & Data Processing Industry

Grids and Clusters

1940 1950 1960 1970 1980 1990 2000 2012

Data Processing Industry
(**1968**): $70 Milion
(**1978**): $3.15 Billion

Peer-to-Peer Systems (PCs - not distributed!)

BitTorrent

P2P Systems (90s-00s)
- Many Millions of users
- Many GB per day

IBM
xerox
Honeywell

Time Sharing Industry (1975):
- Market Share: Honeywell 34%, IBM 15%,
- Xerox 10%, CDC 10%, DEC 10%, UNIVAC 10%
- Honeywell 6000 & 635, IBM 370/168,
  Xerox 940 & Sigma 9, DEC PDP-10, UNIVAC 1108

Scalable Computing Trends: Technology

- **Doubling Periods** – storage: 12 months, bandwidth: 9 months, and CPU compute capacity: 18 months (what law is this?)
- **Moore's law** indicates that processor speed doubles every 18 months.
- **Gilder's law** indicates that network bandwidth has doubled each year in the past.
- Then and Now
  - Bandwidth
    - 1985: mostly 56Kbps links nationwide
    - 2015: Tbps links widespread
  - Disk capacity
    - Today's PCs have TBs, far more than a 1990 supercomputer

The Trend toward Utility Computing

- Aiming towards autonomic operations that can be self-organized to support dynamic discovery. Major computing paradigms are composable with ***QoS and SLAs (service-level agreements).***
- In 1965, MIT's Fernando Corbató of the Multics operating system envisioned a computer facility operating “like a power company or water company”.
- **Plug** your thin client into the computing Utility **and Play** Intensive Compute & Communicate Application
- **Utility computing** focuses on a business model in which customers receive computing resources from a paid service provider.
- All **grid/cloud platforms are regarded as utility service providers.**

Characteristics of Cloud Computing

Common Characteristics:

- Massive Scale
- Resilient Computing
- Homogeneity
- Geographic Distribution
- Virtualization
- Service Orientation
- Low Cost Software
- Advanced Security

Essential Characteristics:

- On Demand Self-Service
- Broad Network Access
- Rapid Elasticity
- Resource Pooling
- Measured Service

Features of Today's Clouds

I. **Massive scale:** Very large data centers, contain tens of thousands sometimes hundreds of thousands of servers and you can run your computation across as many servers as you want and as many servers as your application will scale.

II. **On-demand access:** Pay-as-you-go, no upfront commitment.
- And anyone can access it

III. **Data-intensive Nature:** What was MBs has now become TBs, PBs and XBs.
- Daily logs, forensics, Web data, etc.

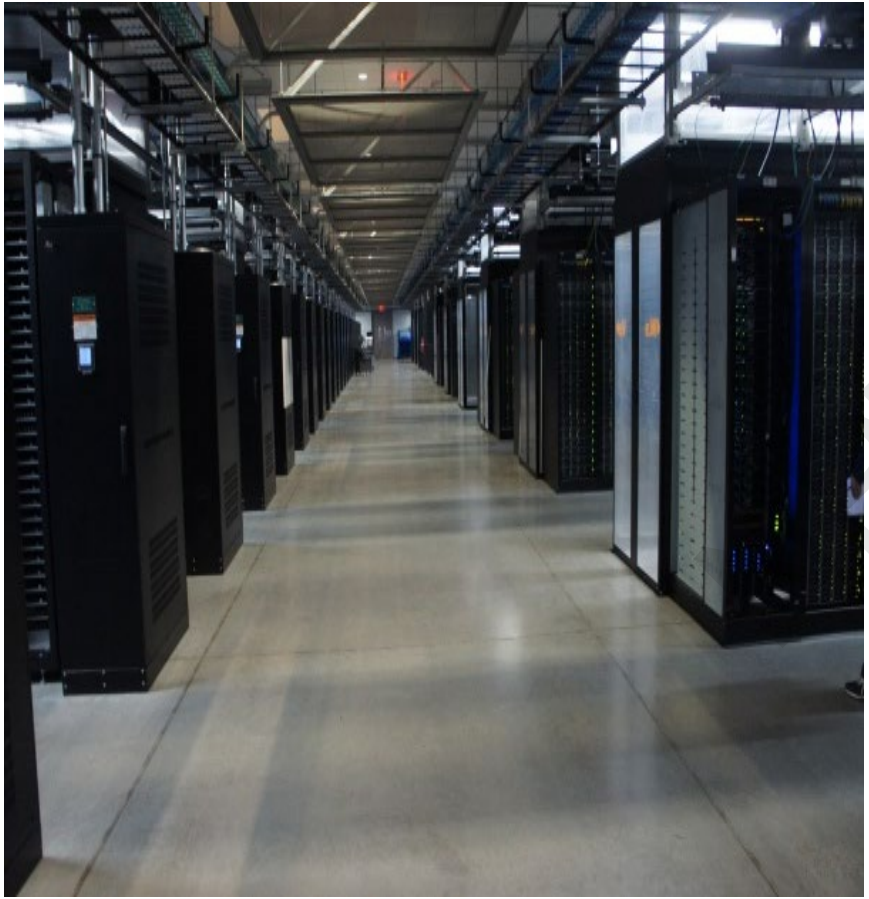
IV. **New Cloud Programming Paradigms:** MapReduce/Hadoop, NoSQL/Cassandra/MongoDB and many others.

- **Combination of one or more of these gives rise to novel and unsolved distributed computing problems in cloud computing.**

I. Massive Scale

- **Facebook [GigaOm, 2012]**
  - 30K in 2009 -> 60K in 2010 -> 180K in 2012
- **Microsoft [NYTimes, 2008]**
  - 150K machines
  - Growth rate of 10K per month
  - 80K total running Bing
  - In 2013, Microsoft Cosmos had 110K machines (4 sites)
- **Yahoo! [2009]:**
  - 100K
  - Split into clusters of 4000
- **AWS EC2 [Randy Bias, 2009]**
  - 40K machines
  - 8 cores/machine
- **eBay [2012]: 50K machines**
- **HP [2012]: 380K in 180 DCs**
- **Google: A lot**

What does a datacenter look like from inside?



Power and Energy

- WUE = Annual Water Usage / IT Equipment Energy (L/kWh)
  – low is good
- PUE = Total facility Power / IT Equipment Power
  – low is good (e.g., Google~1.11)

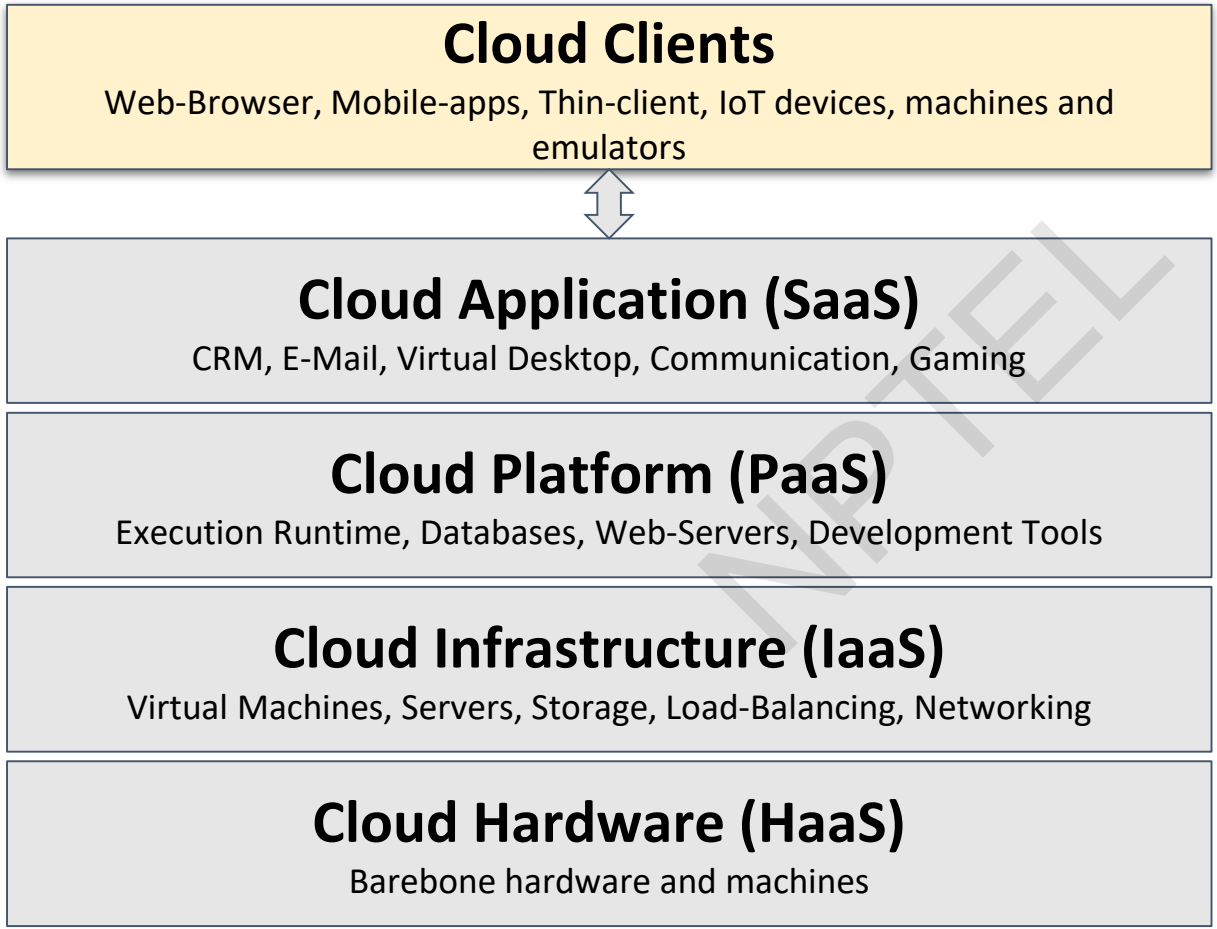
Off-site

On-site

Cooling

- Air sucked in
- Combined with purified water
- M oves cool air through system

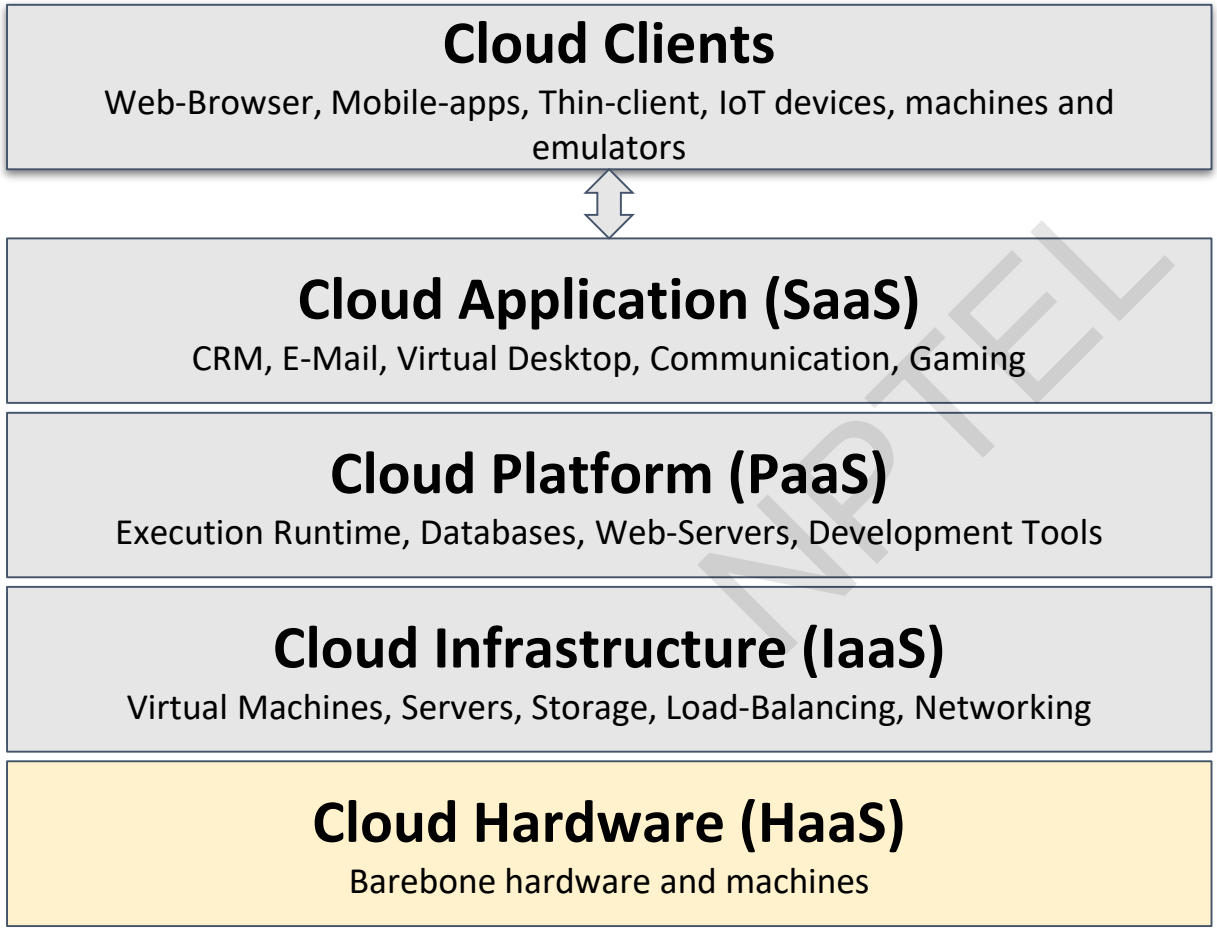
II. On-demand access: *AAS Classification



On-demand: renting vs. buying:

- AWS Elastic Compute Cloud (EC2): a few cents to a few $ per CPU hour
- AWS Simple Storage Service (S3): a few cents per GB-month

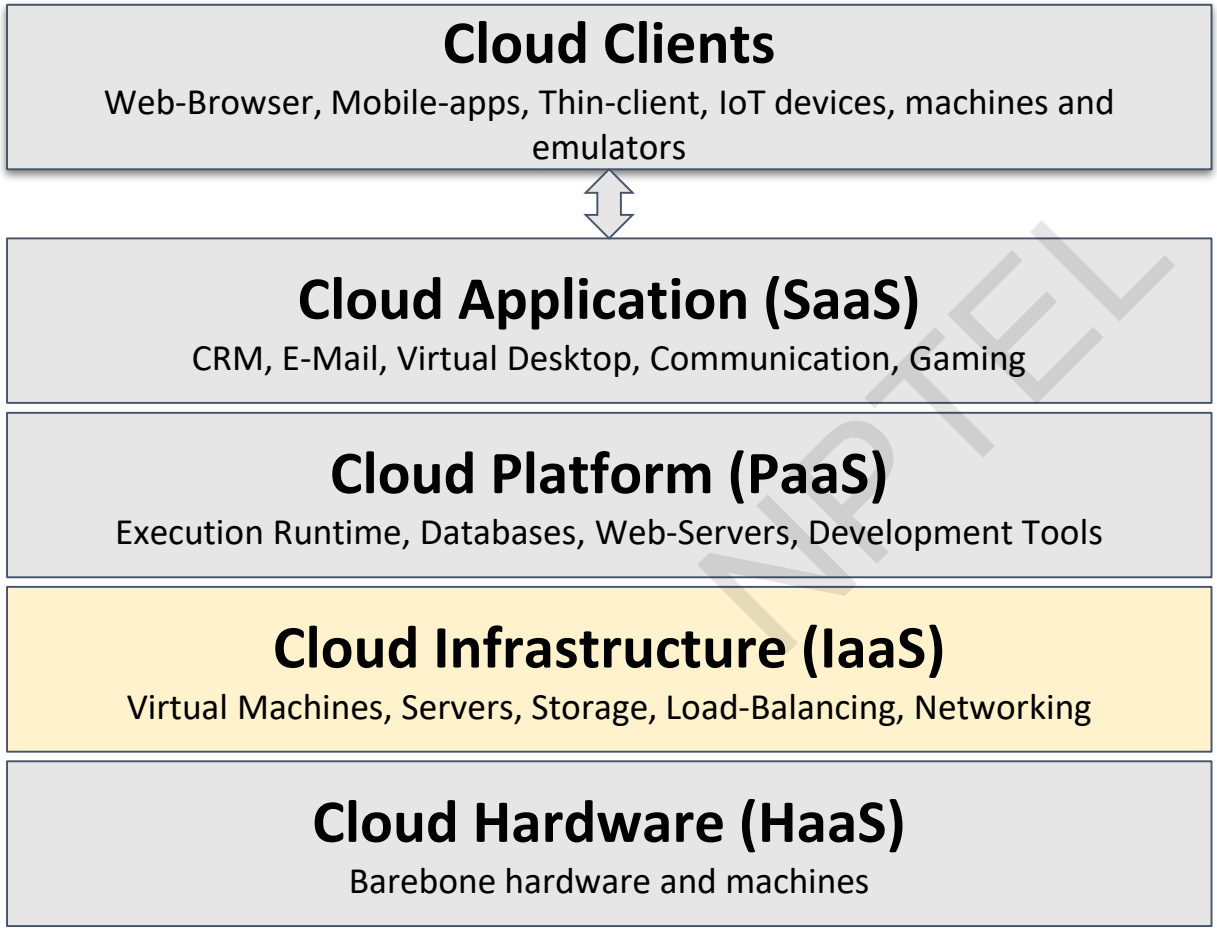
II. On-demand access: *AAS Classification



HaaS: Hardware as a Service

- Get access to barebones hardware machines, do whatever you want with them, Ex: Your own cluster
- Not always a good idea because of security risks

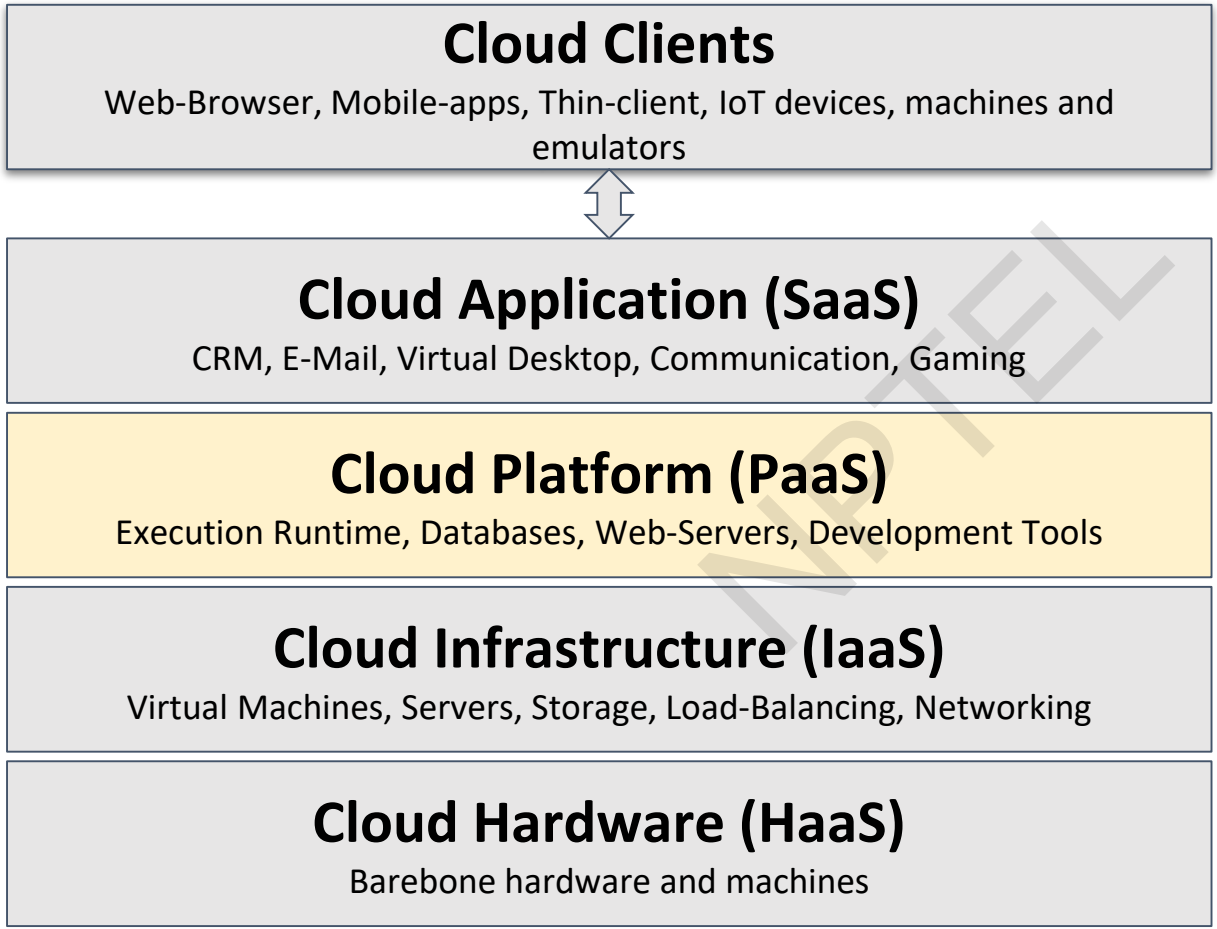
II. On-demand access: *AAS Classification



IaaS: Infrastructure as a Service

- Get access to flexible computing and storage infrastructure. **Virtualization** is one way of achieving this. subsume HaaS.
- Ex: Amazon Web Services (AWS: EC2 and S3), OpenStack, Eucalyptus, Rightscale, Microsoft Azure, Google Cloud.

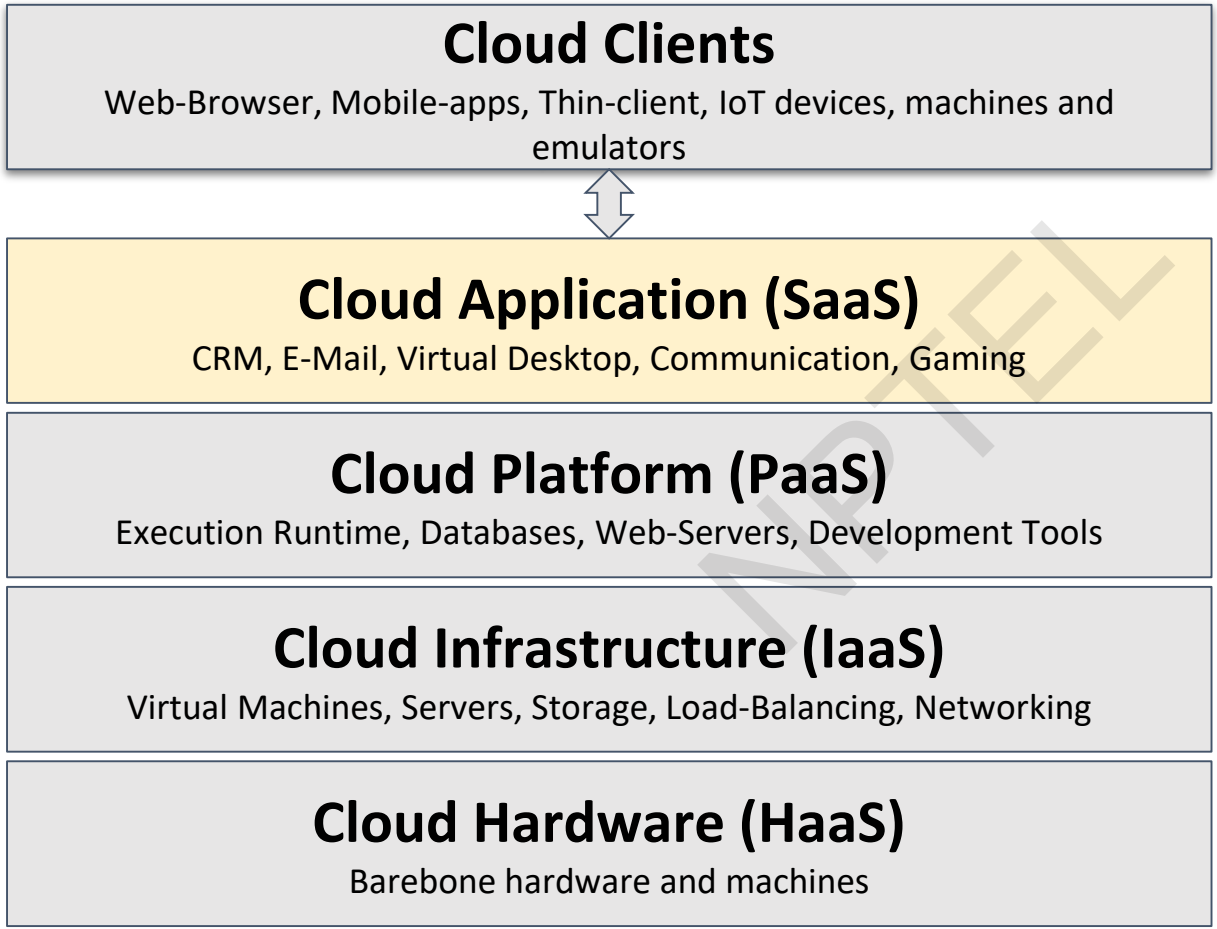
II. On-demand access: *AAS Classification



PaaS: Platform as a Service

- Get access to flexible computing and storage infrastructure, coupled with a software platform (often tightly coupled)
- Ex: Google's AppEngine (Python, Java, Go)

II. On-demand access: *AAS Classification



SaaS: Software as a Service

- Get access to software services, when you need them. subsume SOA (Service Oriented Architectures).
- Ex: Google docs, MS Office on demand

II. On-demand access: *AAS Classification

Services	Description
Services	Services – Complete business services such as PayPal, OpenID, OAuth, Google Maps, Alexa
Application	Application – Cloud based software that eliminates the need for local installation such as Google Apps, Microsoft Online
Development	Development – Software development platforms used to build custom cloud based applications (PAAS & SAAS) such as SalesForce
Platform	Platform – Cloud based platforms, typically provided using virtualization, such as Amazon ECC, Sun Grid
Storage	Storage – Data storage or cloud based NAS such as CTERA, iDisk, CloudNAS
Hosting	Hosting – Physical data centers such as those run by IBM, HP, NaviSite, etc.

Application Focused

Infrastructure Focused

III. Data-intensive Computing

- **Computation-Intensive Computing**
  - Example areas: MPI-based, High-performance computing, Grids
  - Typically run on supercomputers (e.g., NCSA Blue Waters)
- **Data-Intensive**
  - Typically store data at datacenters
  - Use compute nodes nearby
  - Compute nodes run computation services

- In data-intensive computing, the **focus shifts from computation to the data**
- CPU utilization no longer the most important resource metric, instead I/O is (disk and/or network)

IV. New Cloud Programming Paradigms

- Easy to write and run highly parallel programs in new cloud programming paradigms:
  - **Google:** MapReduce and Sawzall
  - **Amazon:** Elastic MapReduce service (pay-as-you-go)
  - Google (MapReduce)
    - Indexing: a chain of 24 MapReduce jobs
    - ~200K jobs processing 50PB/month (in 2006)
  - **Yahoo!** (Hadoop + Pig)
    - WebMap: a chain of several MapReduce jobs
    - 300 TB of data, 10K cores, many tens of hours (~2008)
  - **Facebook** (Hadoop + Hive)
    - ~300TB total, adding 2TB/day (in 2008)
    - 3K jobs processing 55TB/day
  - NoSQL: MySQL is an industry standard, but Cassandra is 2400 times faster

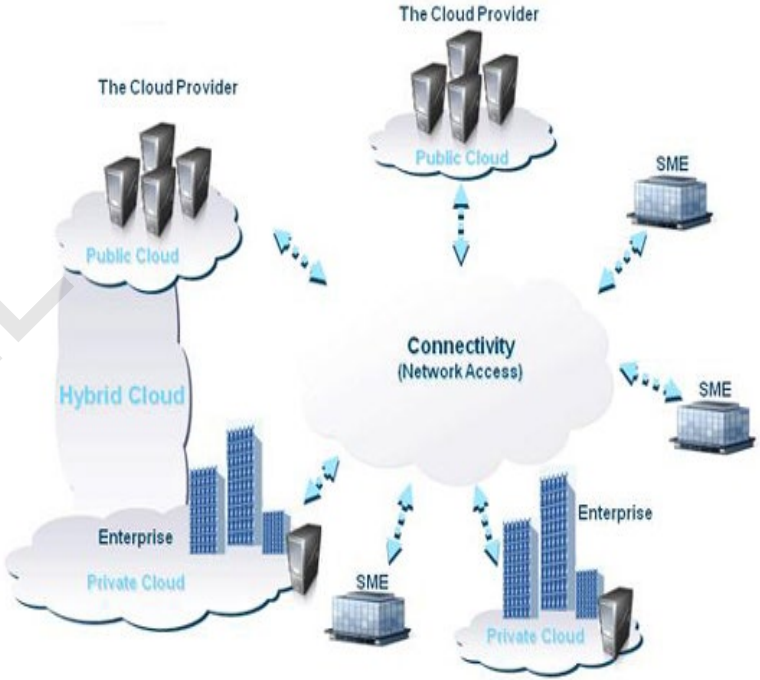
Two Categories of Clouds

- Can be either a (i) **public cloud**, or (ii) **private cloud**

- **Private clouds** are accessible only to company employees
- Example of popular vendors for creating private clouds are VMware, Microsoft Azure, Eucalyptus etc.

- **Public clouds** provide service to any paying customer
- Examples of large public cloud services include Amazon EC2, Google AppEngine, Gmail, Office365 and Dropbox etc.

- *You're starting a new service/company: should you use a public cloud or purchase your own private cloud?*



Single site Cloud: to Outsource or Own?

- Medium-sized organization: wishes to run a service for *M* months
  - Service requires 128 servers (1024 cores) and 524 TB
- **Outsource** (e.g., via AWS): *monthly* cost
  - S3 costs: $0.12 per GB month.
    EC2 costs: $0.10 per CPU hour (costs from 2009)
    Storage = $ 0.12 X 524 X 1000 ~ $62 K
    Total = Storage + CPUs = $62 K + $0.10 X 1024 X 24 X 30 ~ **$136 K**
- **Own**: monthly cost
  - Storage ~ $349 K / *M* Total ~ **$ 1555 K / *M* + 7.5 K** (includes 1 sysadmin / 100 nodes)
  - (using 0.45:0.4:0.15 split for hardware: power: network and 3 year lifetime of hardware)

Break Even analysis: **more preferable to own if:**

- $349 K / *M* < $62 K (storage)
- $ 1555 K / *M* + 7.5 K < $136 K (overall)

Breakeven points

***M* > 5.55 months (storage)**

***M* > 12 months (overall)**

-Startups use clouds a lot

-Cloud providers benefit monetarily most from storage

Conclusion

We covered the following topics,

- ❑ Overview of Cloud Computing
- ❑ Cloud computing v/s distributed computing
- ❑ Brief History and the current demand of Cloud Data Centers
- ❑ Various Features and Characteristics of Cloud computing
- ❑ On-Demand Cloud Computing with Private and Public clouds

Thank You!

Thank You!

References

No references were used.

Week 1 Lecture 2

30 mins

Cloud Computing and its Limitations to Support Low Latency and RTT

Dr. Rajiv Misra, Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Contents

On completion of this lecture you will get to know about the following:

- Understanding of today's cloud scenario
- Different objectives of cloud
- Current limitations of traditional cloud
- Why there is a need of Edge Computing?

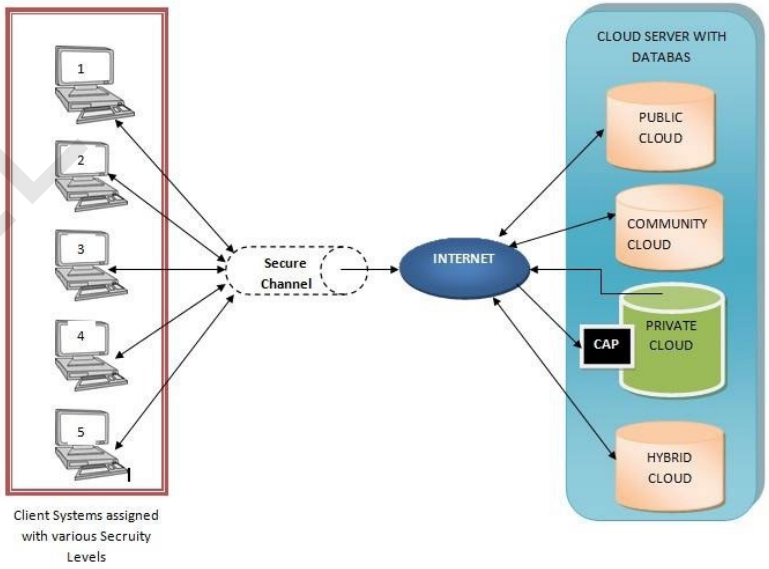
Current State of Today's Cloud

- Highly centralised set of resources
- Compute is going beyond VMs
- Storage is complemented by CDN
- Network stack is programmable
- The Web and Software-as-a-Service
- Infrastructure-as-a-Service
- High-Availability cloud

Current State of Today's Cloud:

Highly Centralized in Client-Server Architecture

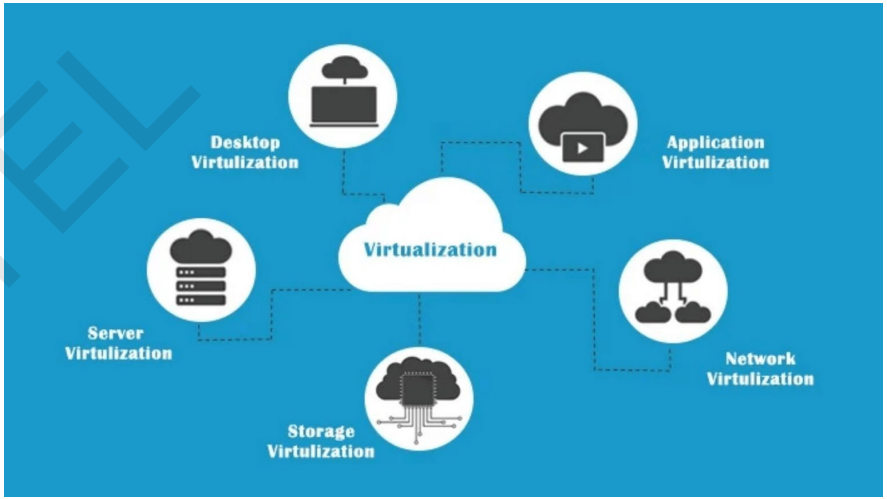
- Cloud computing started as all about virtual machines that were running in a remote data center (or storage).
- Highly centralized architecture closely resembles 90s client-server computing.
- For example cloud (the remote data center or the remote infrastructure) exposed by Amazon, Microsoft, Google, IBM and others is the server and the machine from which you are connecting to it and consuming the cloud resources is the client.



Current State of Today's Cloud:

Compute is going beyond VMs

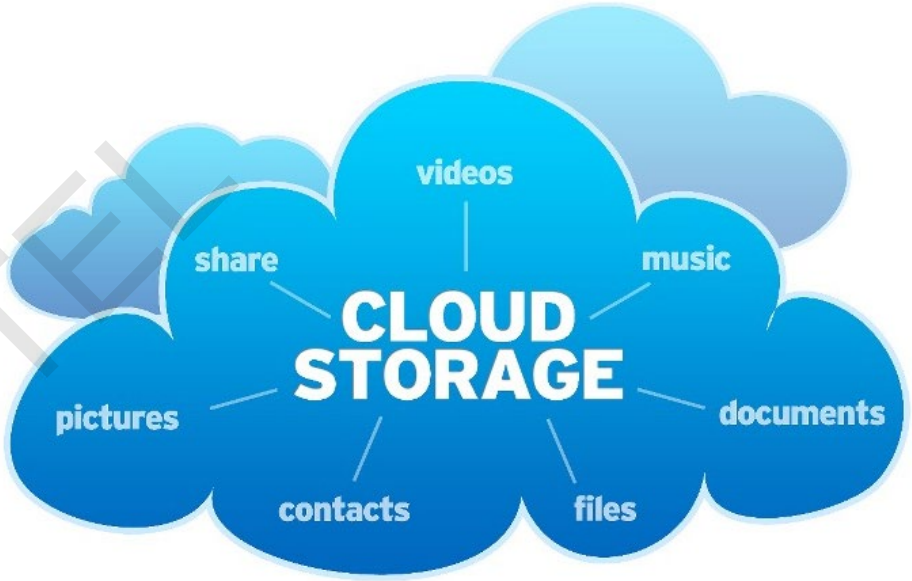
- Although cloud resembles the 90s client-server computing but at the same time compute has gone beyond VMs the first generation of cloud was all about VM virtual machines.
- Where you could programmatically launch a VM and you could SSH into it and take control of the Virtual Machine and install the software.
- But there is a dramatic shift in the compute where VMs are slowly getting replaced by containers.
- More and more workloads are moving towards containers.



Current State of Today's Cloud:

Storage is complemented by CDN

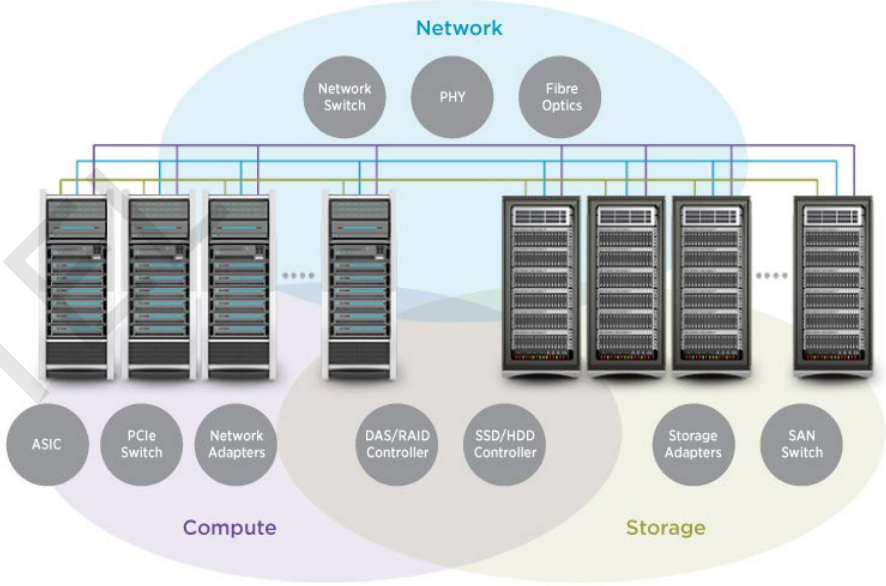
- Another important trend almost all the public cloud are in storage offerings.
- Object storage is complemented by a content delivery network today.
- Whenever you put an object in a bucket or a container of the public cloud storage you can click a check box to basically replicate and cache the data across multiple edge locations but this edge is not the edge that we are talking about this is the content delivery network where it caches the frequently accessed content in a set of pop or edge locations.



Current State of Today's Cloud:

Network stack is programmable

- Finally network has become extremely programmable today.
- If you look at the hybrid cloud, multi-cloud scenarios and how network traffic is getting routed and how load balancers firewalls
- and a variety of network components are configured it is through api's and programmability.
- The same capability of SDN is enabling hybrid scenarios particularly when we look at the combination of software-defined network with some of the emerging networking technologies.
- These mesh they are opening up additional avenues some of the very recent trends like Google's Anthos, IBM cloud private and some of the other container based hybrid cloud platforms are heavily relying on the programmable Network stack and also a combination of SDN with service mesh.
- This is the current state of the cloud and these trends represent how the cloud is currently being consumed or how it is delivered to customers but cloud is going through a huge transformation.



Multiple waves of innovation in Cloud: Pass to IOT

- Initially cloud was all about compute storage and network resources globally available highly centralized set of resources because cloud made compute and storage extremely cheap and affordable lot of industrial customers and enterprises started connecting devices to the cloud.
- The data that was not persisted or aggregated or acquired is now streamed to the cloud because it is extremely cheap to store data in the cloud.
- So a lot of companies and lot of industrial environments started to take advantage of the cloud by streaming the data coming from a variety of sensors and devices.
- Also use the cheaper compute power to process those data streams and make sense out of the raw data generated these sensors and devices and that was the next big shift in the cloud this was IOT pass.

Challenges for IOT-Pass

- If you look at azure IOT, Google Cloud IOT, AWS IOT core all of them essentially give you a mechanism a platform to connect devices and store data and process it in the cloud but it was not sufficient or it was not enough to address a lot of scenarios while cloud enabled capabilities like Big Data and IOT.
- Lot of customers were not ready to move the data to the cloud that is one challenge.
- The second one is the round trip from the devices to the cloud and back to the devices was too long and it was increasing the latency in a lot of mission-critical industrial IOT scenarios.
- Sending the data to the cloud and waiting for the cloud to process it and send the results back was just not feasible so there had to be a mechanism where data could be processed locally and compute comes much closer to the devices or the sources of data so that's how IOT led to edge computing and today almost every mainstream enterprise IOT platform has a complimentary edge offering and associated edge offering and more recently there has been a lot of focus on artificial intelligence.

Cloud for AI-ML

- Today's cloud has become the logical destination for training and running artificial intelligence and machine learning models.
- Due to accelerators like GPUs GPUs FPGAs it has become extremely cheap and also powerful to train very complex very sophisticated ML models and AI models
- But in most of the scenarios a model that is restrained in the cloud is going to be run in an offline environment.
- For example, you might have trained an artificial intelligence model that can identify the make and model of a car and automatically charge the tall fee for that vehicle when it passes through the toll gate now since the toll gates are on highways and freeways with very little connectivity and almost with no network access you need to run this model in offline scenario.
- So edge computing became the boundary for running these cloud trained AI models but running in an offline mode within the edge so that is how we are basically looking at the evolution of cloud and on the waves of innovation.
- So cloud are distributed or rather decentralized platform for aggregating storing and processing data with high performance computing IOT brought in all the devices to the cloud with IOT data at edge made cloud decentralized by bringing compute closer to the data source and now it is AI that is actually driving the next wave where cloud is becoming the de facto platform for training the models and edge is becoming the de facto platform for running the models so one is called the training the other one is called inferencing.

Limitations of current Cloud system

- AI use cases need **real-time** responses from the devices they are monitoring.
- Cloud-based inference cannot provide this real-time response due to inherent issues with **latency**.
- If edge devices have **connectivity issues** or no internet connection it can not perform well.
- **Sufficient bandwidth** required to transfer the relevant amount of data in a proper time frame can also be an issue.

Waves of Innovation: Cloud IoT Edge ML

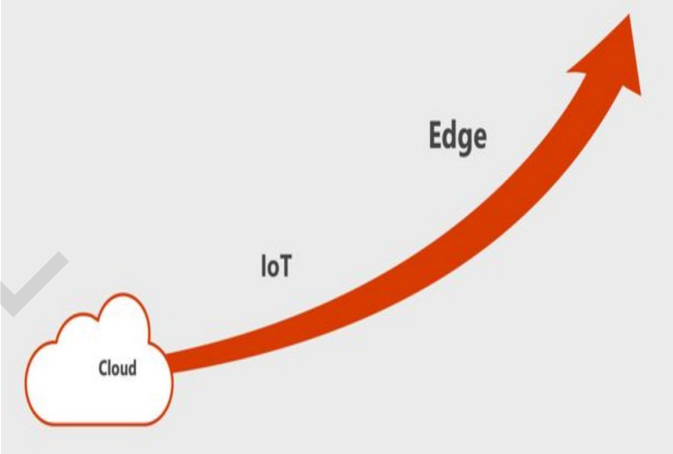
Cloud
(the waves of innovation started with cloud)
Globally available, unlimited compute resources

IoT
(IoT-as-SaaS platform is key drivers of public cloud)
Harnessing signals from sensors and devices, managed centrally by the cloud

Edge
(IoT realize not everything needs to be in the cloud)
Intelligence offloaded from the cloud to IoT devices

ML
(rise of AI, ML models are trained in cloud are deployed at the edge to make inferencing for predictive analytics)

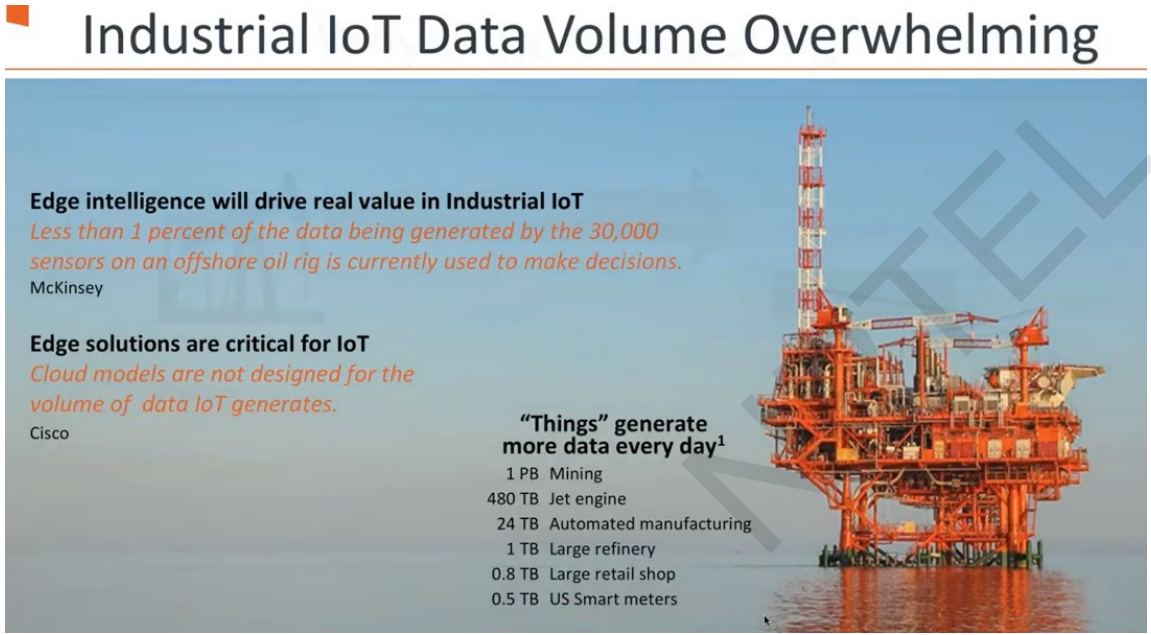
Breakthrough intelligence capabilities, in the cloud and on the edge



Current State of the Cloud

- Highly centralized set of resources,
- Resembles Client/Server computing
- Compute is going beyond VMs as Containers becoming mainstream
- Storage is complemented by CDN is replicated and cached at edge locations
- Network stack is programmable SDN enabling hybrid scenarios

Overview: Edge Computing use-case



Huge amounts of data are produced in the oil and gas industry by sensors like **temperature**, **pressure**, and **velocity**. The size of such data ranges from **terabytes** to **petabytes** (per second).

It is impractical to simply send all of that data to a cloud environment for processing due to a number of factors, including the high cost of connectivity into the cloud, the cost of bandwidth, the latency involved in moving all of the data to the cloud, and the security risks posed by cybersecurity attacks.

Figure Sources: https://www.norgren.com/en/process-control/energy

Edge Computing

- Edge computing makes the cloud truly distributed
- Moves core cloud services closer to the origin of data
- Edge Mimics public cloud platform capabilities
- Delivers storage, compute, and network services locally.
- Reduces the latency by avoiding the round trip to the cloud
- Brings in data sovereignty by keeping data where it actually belongs, savings on cloud and bandwidth usages

Functionality of Edge

- Data Ingestion and M2M Brokers
- Object Storage
- Functions as a Service
- Containers
- Distributed Computing
- NoSQL/Time-Series Database
- Stream Processing
- ML Models

Edge Computing: Overview

Edge computing operates on underlying principles such as Docker containers, Kubernetes, MQTT, Kafka, time and clock synchronization, and key-value stores at edge.

Recent advances of Machine Learning Inferencing at the Edge for use cases: predictive maintenance, image classifier, and self-driving cars for IoT applications such as Industry 4.0 etc.

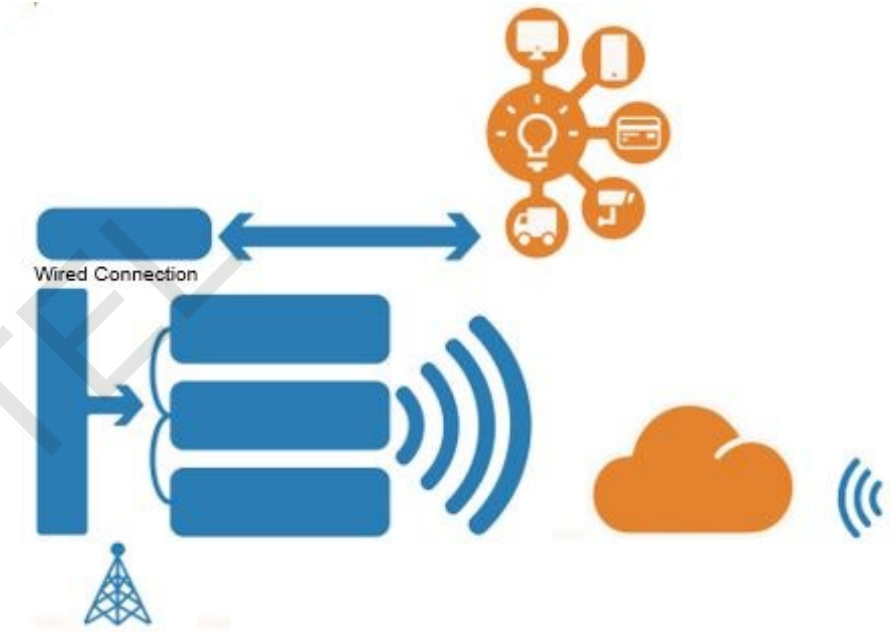
kubernetes

SELF-DRIVING CAR

Cloud Data Center: Current Demand

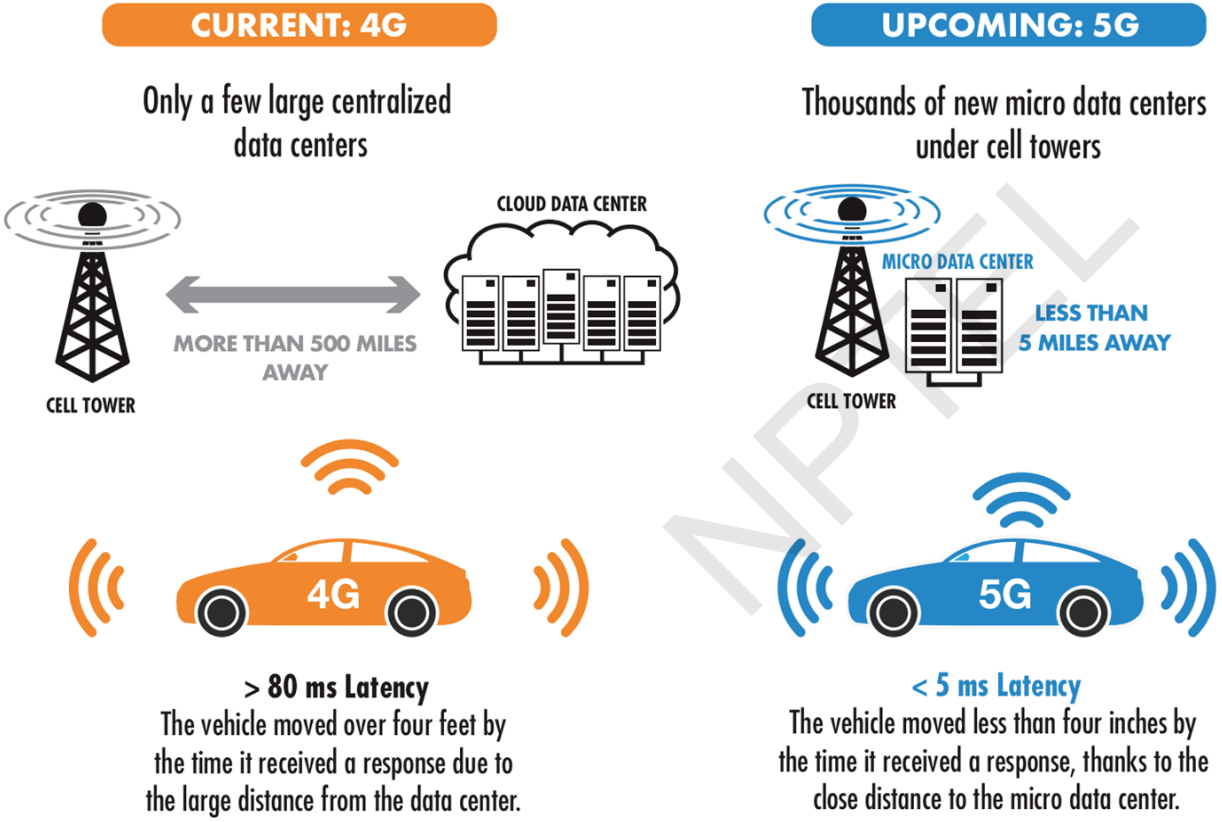
- In the next decade, we will continue to see skyrocketing growth in the number of IP-connected mobile and machine-to-machine (M2M) devices, which will handle significant amounts of IP traffic.

- Tomorrow's consumers will demand faster Wi-Fi service and application delivery from online providers. Also, some M2M devices, such as autonomous vehicles, will require real-time communications with local processing resources to guarantee safety.



- Today's IP networks cannot handle the high-speed data transmissions that tomorrow's connected devices will require. In a traditional IP architecture, data must often travel hundreds of miles over a network between end users or devices and cloud resources. This results in latency, or slow delivery of time-sensitive data.

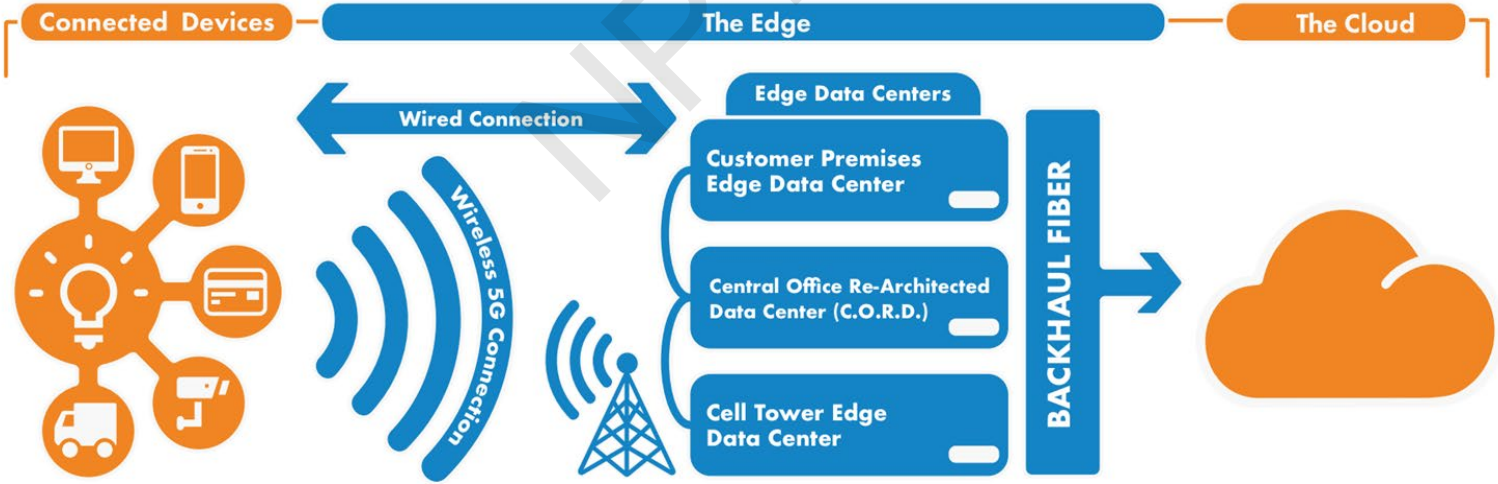
Cloud Data Center: Current Demand



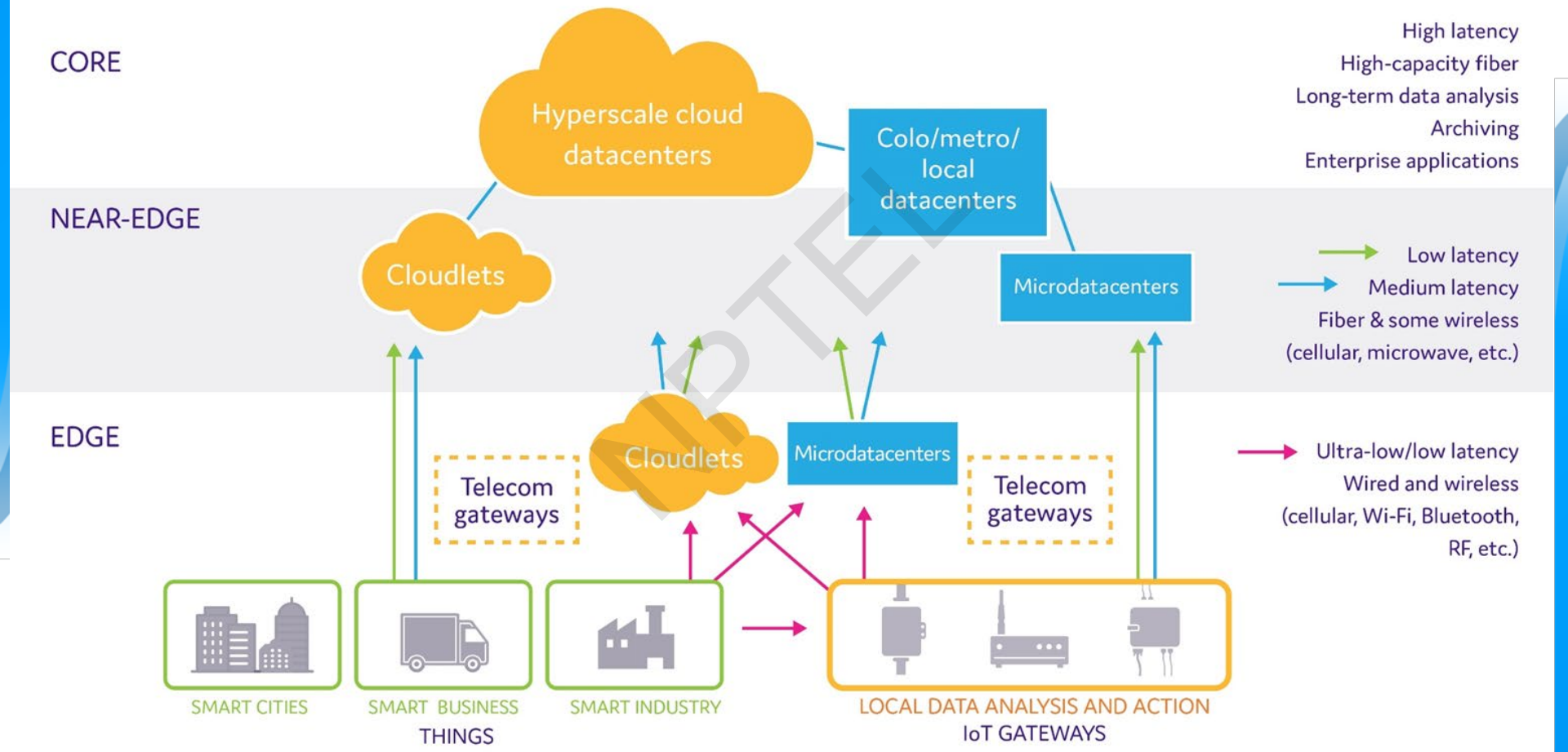
Edge Data Center: Solution

The solution to reducing latency lies in edge computing. By establishing IT deployments for cloud-based services in edge data centres in localized areas, we effectively bring IT resources closer to end users and devices. This helps us achieve efficient, high-speed delivery of applications and data. Edge data centres are typically located on the edge of a network, with connections back to a centralized cloud core.

Instead of bringing the users and devices to the data centre, we bring the power of the data centre to the users and devices. Edge computing relies on a distributed data centre architecture, in which IT cloud servers housed in edge data centres are deployed on the outer edges of a network. By bringing IT resources closer to the end users and/or devices they serve, we can achieve high-speed, low-latency processing of applications and data.



Edge Data Center: Solution



Why Move Cloud services to the Edge?

There are four main benefits of moving data centres to the edge,

1.Latency: edge data centres facilitate lower latency, meaning much faster response times. Locating compute and storage functions closer to end users reduces the physical distance that data packets need to traverse, as well as the number of network “hops” involved, which lowers the probability of hitting a transmission path where data flow is impaired

2.Bandwidth: edge data centres process data locally, reducing the volume of traffic flowing to and from central servers. In turn, greater bandwidth across the user’s broader network becomes available, which improves overall network performance

3.Operating Cost: because edge data centres reduce the volume of traffic flowing to and from central servers, they inherently reduce the cost of data transmission and routing, which is important for high-bandwidth applications. More specifically, edge data centres lessen the number of necessary high-cost circuits and interconnection hubs leading back to regional or cloud data centres, by moving compute and storage closer to end users

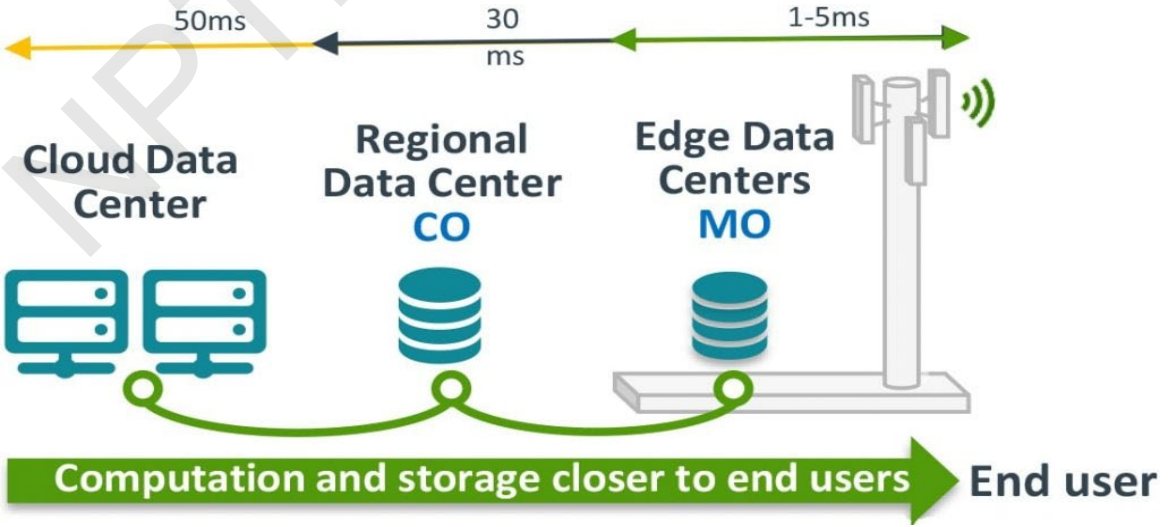
4.Security: edge data centres enhance security by: i) reducing the amount of sensitive data transmitted, ii) limiting the amount of data stored in any individual location, given their decentralized architecture, and iii) decreasing broader network vulnerabilities, because breaches can be ring-fenced to the portion of the network that they compromise.

Edge Data Center: Introduction

- The major benefit of an edge data center is the quick delivery of services with minimal latency, thanks to the use of edge caching. Latency may be a big issue for organizations that have to work with the internet of things (IoT), big data, cloud and streaming services.

- Edge data centers can be used to provide high performance with low levels of latency to end users, making for a better user experience. Typically, edge data centers will connect to a larger, central data center or multiple other edge data centers.

- Data is processed as close to the end user as possible, while less integral or time-centric data can be sent to a central data center for processing. This allows an organization to reduce latency.



Conclusion

We covered the following topics,

- ❑ Today's cloud scenario
- ❑ Objectives of cloud
- ❑ Current limitations of traditional cloud
- ❑ Need of Edge Computing
- ❑ Edge-Datacenter

Thank You!

Thank You!

References

No references were used.

Week 1 Lecture 3

30 mins

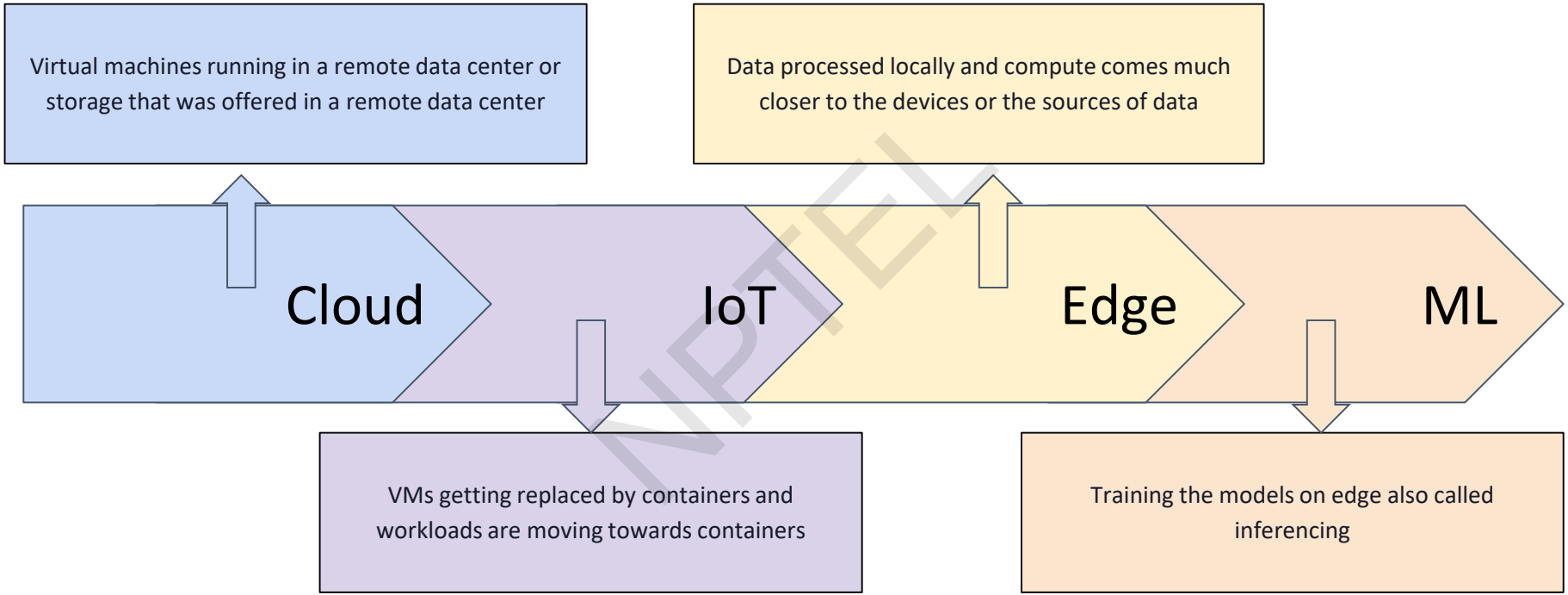
Introduction to Edge Computing

Dr. Rajiv Misra, Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Contents

- Introduction to Edge Computing
  - Edge Computing Architecture & building blocks
  - Edge Computing for IOT
  - Advantages of Edge Computing

Recapitulate: Evolution of Cloud

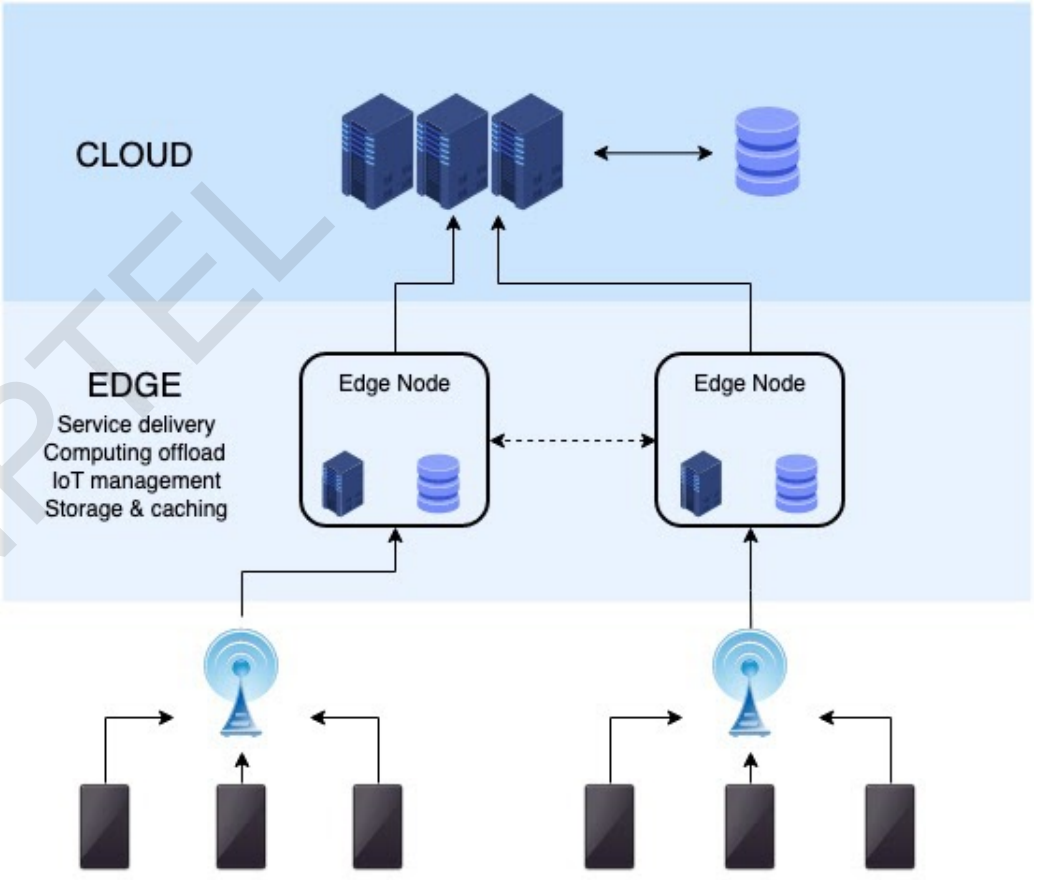


Introduction to Edge Computing

- Edge computing allows the cloud to be genuinely distributed.
- Don't need to rely on the cloud for all the processing and data aggregation collection processing and querying.
- Mimics the public cloud platform capabilities.
- Reduces the latency by avoiding the round-trip and brings in the data sovereignty by keeping data where it actually belongs.
- Delivers local storage, compute, and network services.

Edge Computing makes cloud distributed

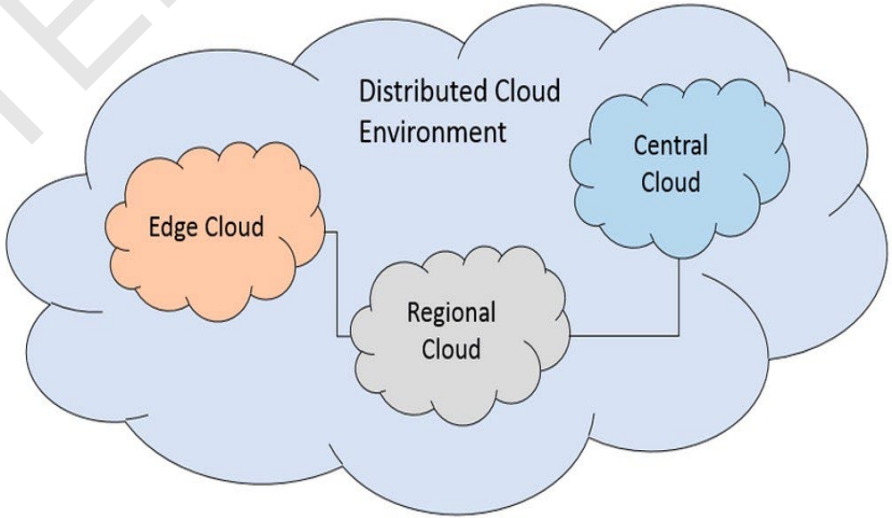
- Edge computing makes the cloud truly distributed. The current cloud or rather the previous generation of cloud was almost like a mainframe or like a client-server architecture where very little processing was done on the client side but all the heavy lifting was done by the cloud.
- With all the innovations in the hardware chips and with the affordable electronics and silicon it makes more sense to bring compute down to the last mile and actually keep the compute closer to the devices.
- So that's when edge computing becomes more and more viable where you don't need to rely on the cloud for all the processing and data aggregation, collection, processing and querying instead you could actually run a computing layer that is very close to the devices.



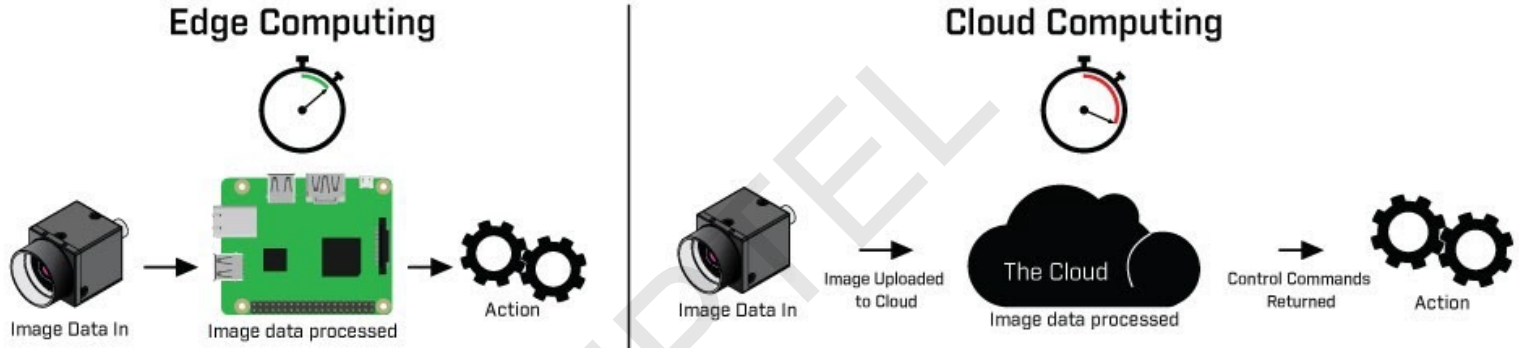
Edge Computing mimics the public cloud platform capabilities and Move cloud service closer to data-source

- The edge computing mimics the public cloud platform capabilities
  For example when you dissect an edge computing platform you would notice that it almost has all the capabilities of a typical public cloud

- IOT pass: it has device management, it has data ingestion, it has stream analytics and it can run machine learning models and it can run server less functions so all of those are capabilities that are predominantly available on the public cloud.
- But with edge computing they all come to the last mile delivery point and run very close to the source of the data which is sensors actuators and devices.

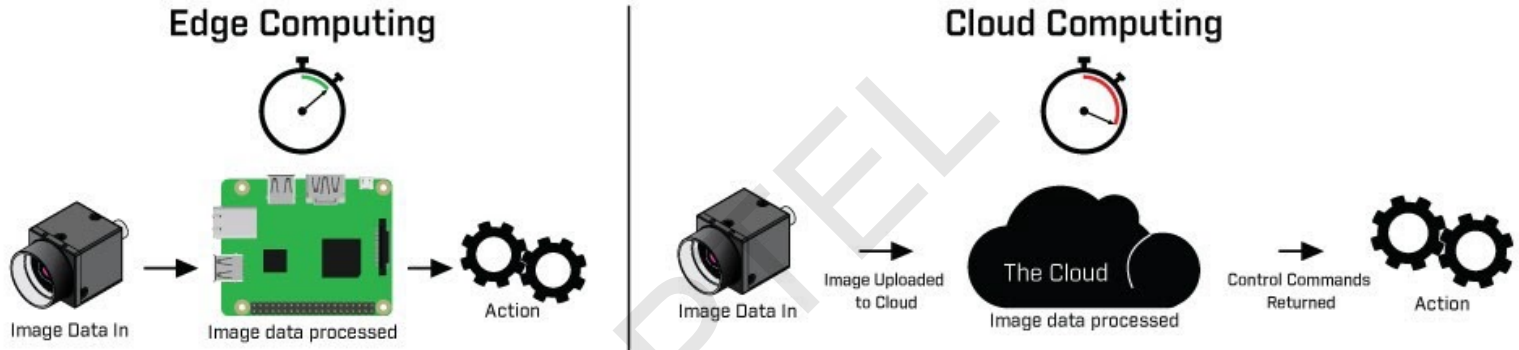


Edge Computing reduces the latency by avoiding the round-trip and brings in the data sovereignty



- The biggest advantage of deploying an edge computing layer is that it reduces the latency by avoiding the round-trip.
- It also brings in the data sovereignty by keeping data where it actually belongs to.

Edge Computing reduces the latency by avoiding the round-trip and brings in the data sovereignty



- **For example** in a healthcare scenario it may not be viable or it may not be compliant to actually stream sensitive patient data to the cloud where it is getting stored and processed instead the patient data should remain on-Prem within the hospital premises but it still needs to go through lot of processing and find out very useful insights so in that case the edge computing layer is going to stay close to the healthcare equipment with connectivity back to the cloud and the architects and the customer engineers will decide what data will stay within the edge boundary and what will actually cross that and move to the cloud may be anonymized data.

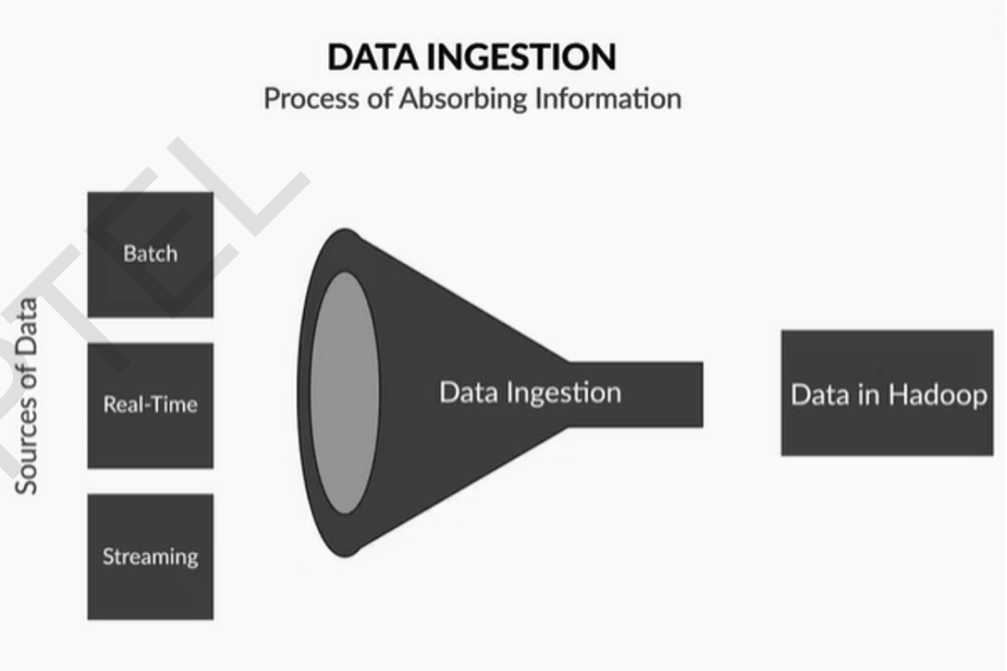
Building Blocks of Edge Computing

- Data Ingestion
- M2M Brokers
- Object Storage
- Function as a Service NoSQL/Time-Series Database
- Stream Processing
- ML Models

Building Blocks of Edge Computing

Data Ingestion

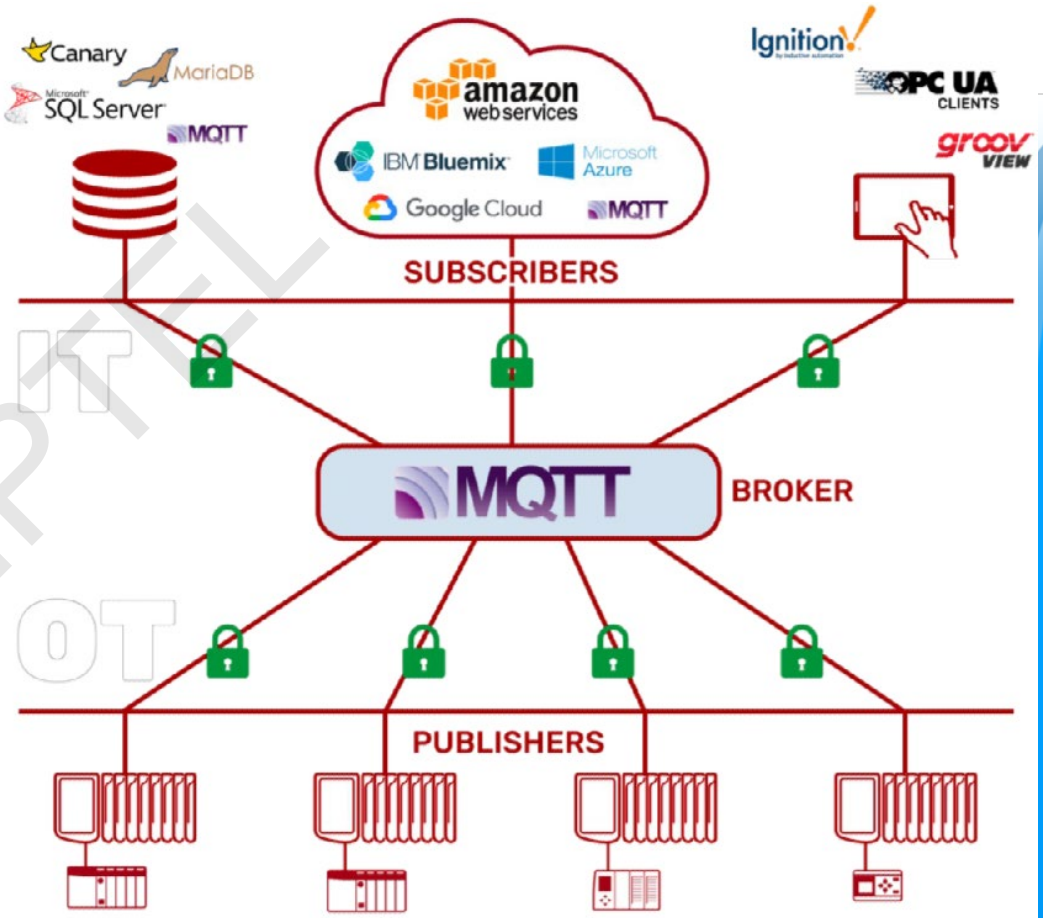
- This is the high velocity, high throughput data endpoint like the Kafka endpoint that is going to ingest the data.
- To ingest something is to take something in or absorb something. It is the process of obtaining and importing data for immediate use or storage in a database.
- Data can be streamed in real time or ingested in batches. In real-time data ingestion, each data item is imported as the source emits it



Building Blocks of Edge Computing

M2M Brokers

- Edge will also run message brokers that will orchestrate machine to machine communication.
- For example device one talks to device two via the M2M broker.

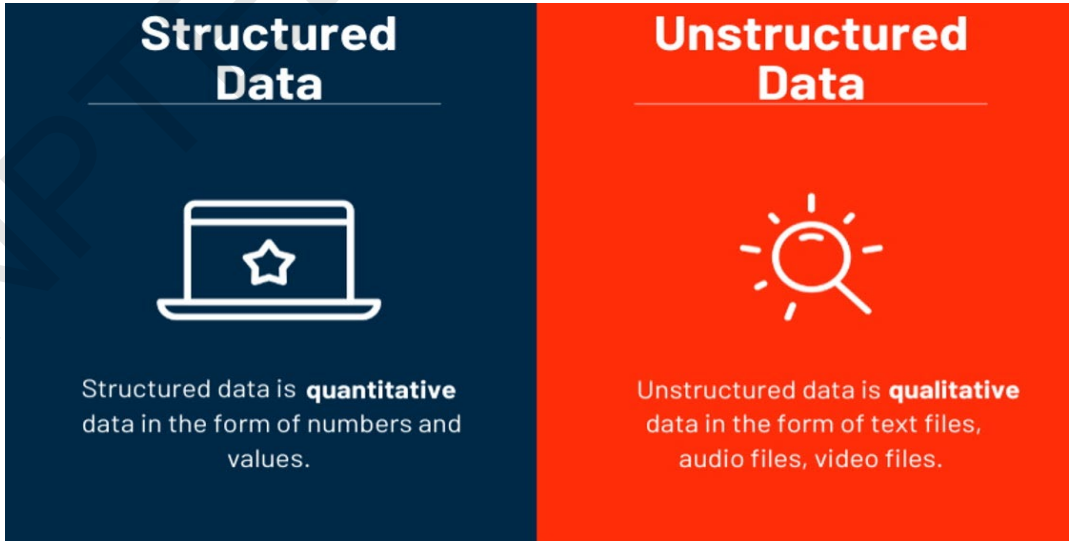


Building Blocks of Edge Computing

Storage

- **Object Storage**: there may be unstructured storage particularly to store the feed from video cameras and mics and anything that is unstructured will go into object storage.

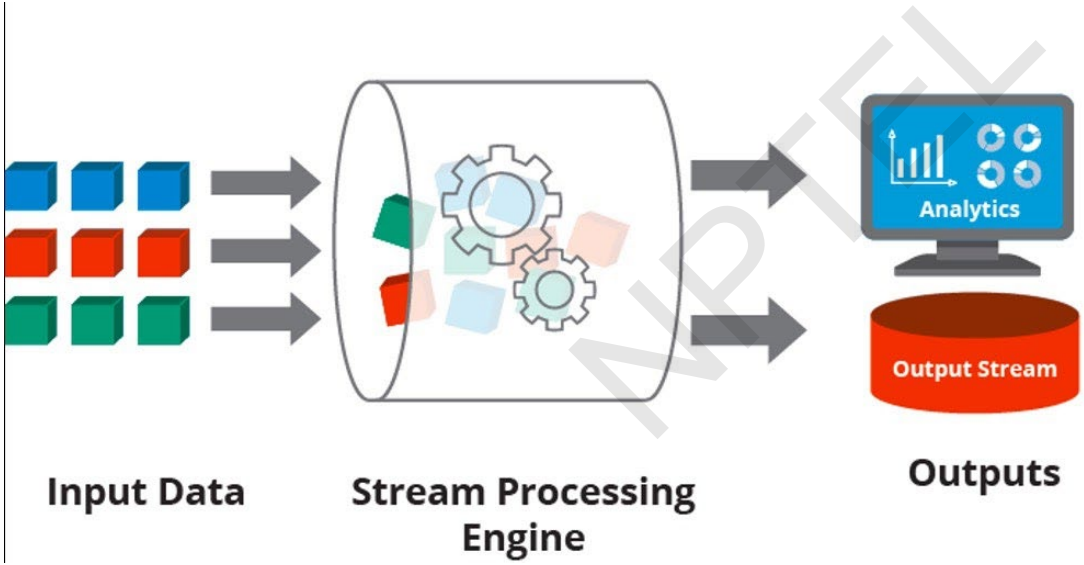
- **NoSQL/Time-Series Database**: More structured data goes into time series database and NoSQL database



Building Blocks of Edge Computing

Stream Processing

- It is a complex event processing engine that is enabling you to perform real-time queries and process the data as it comes.



- For example for every data point you want to convert Fahrenheit to Celsius or you want to convert the timestamp from one format to another, you could do it either in stream processing.

Building Blocks of Edge Computing

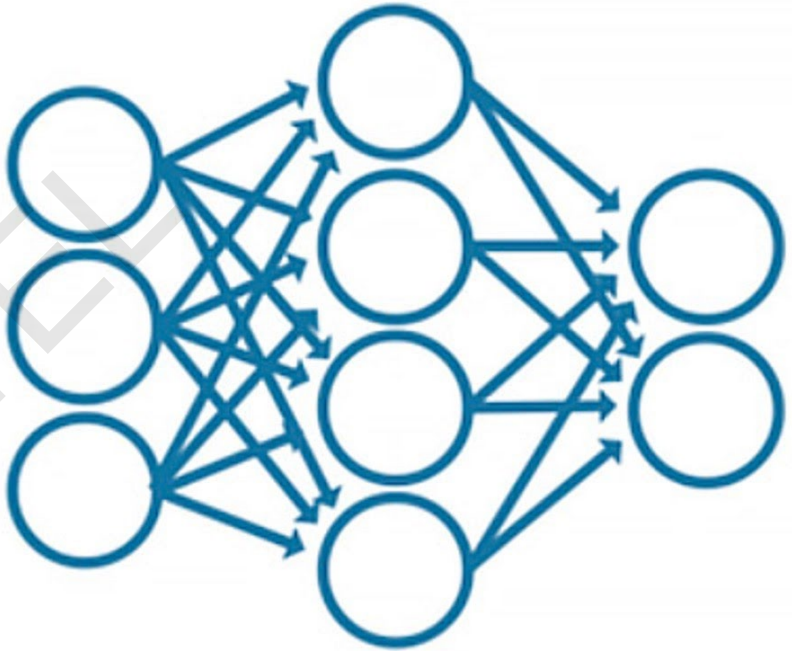
Function as a Service

- To add additional business logic there is a functions as a service which is actually responsible for running lightweight compute.
- If you need to do more sophisticated code you could actually move that to functions as a service.

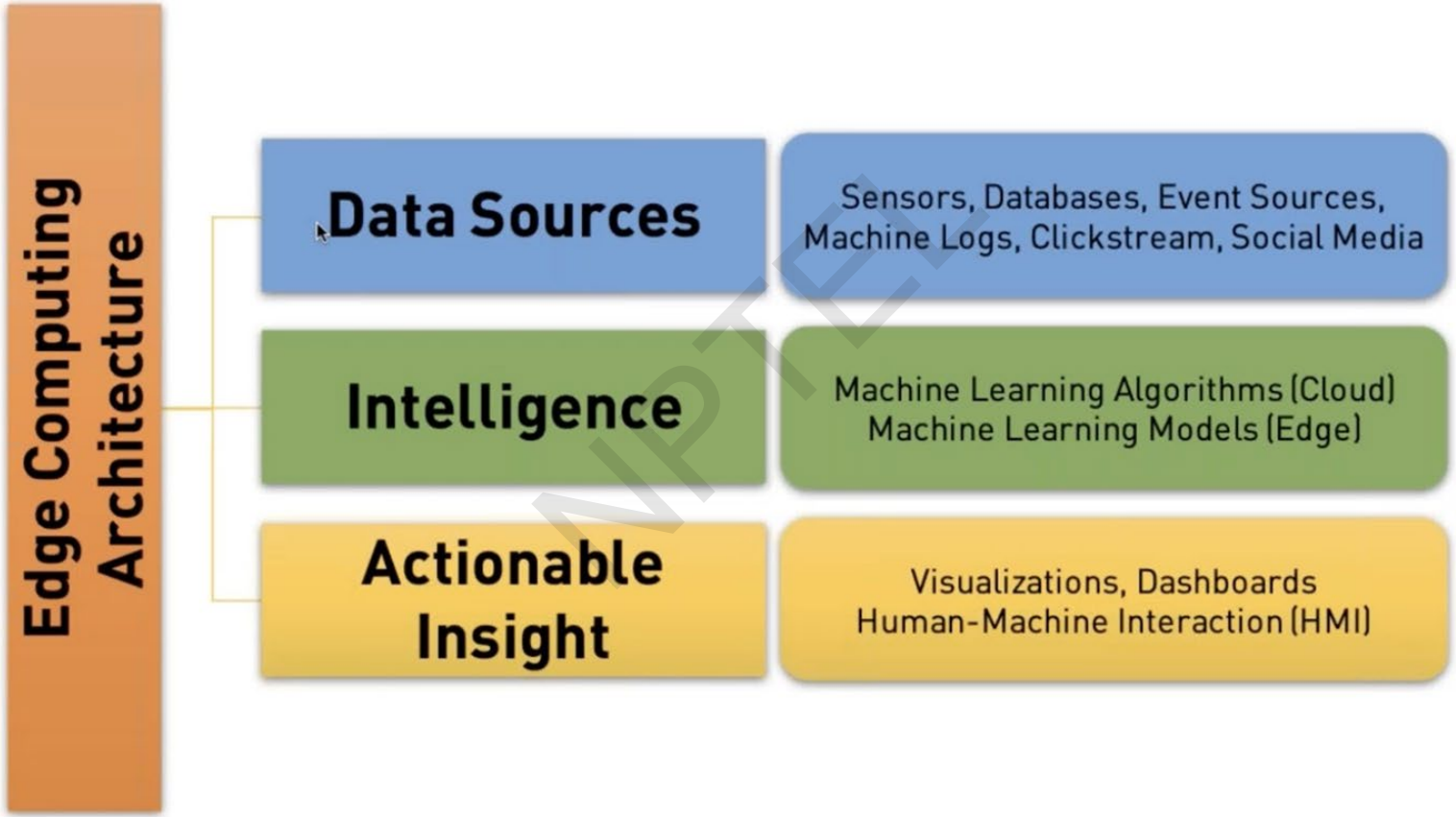
Building Blocks of Edge Computing

ML Models

- Lastely, there is an ML runtime for example most of the computing platforms are capable of running tensorflow light, cafe models and pytorch models, so you can actually process the data that comes in more intelligently and take preventive measures and perform predictive analytics.



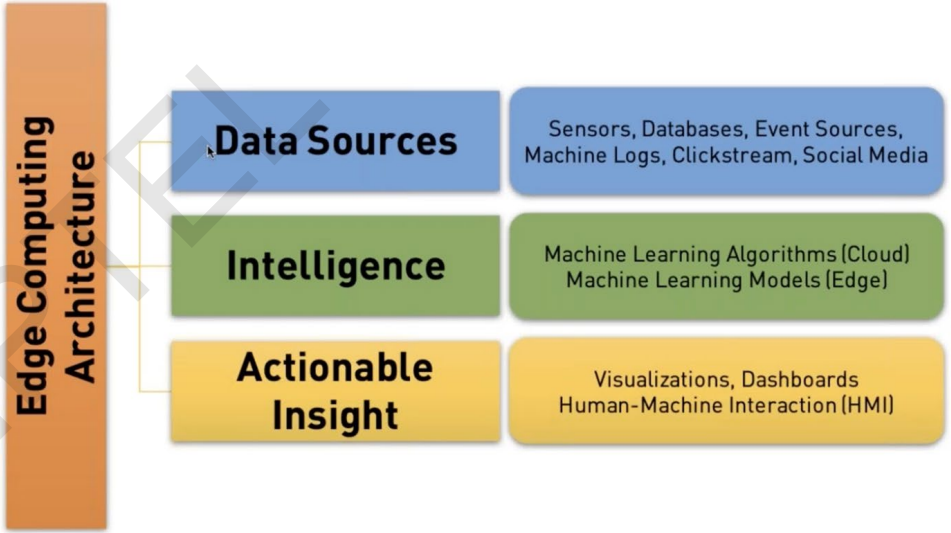
Edge Computing Architecture



Edge Computing Architecture

Three-tier Architecture

Now let's look at this from a different dimension.

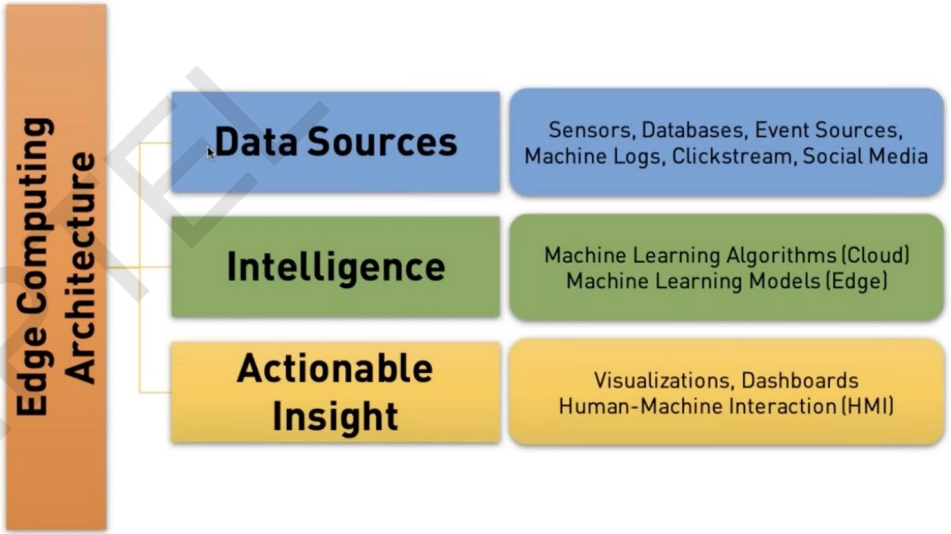


- There are data sources and by the way edge computing is not confined just to IOT, it could be even for non IOT use cases. Anything that generates data can be fed into an IOT like cameras, clickstream analysis, gaming, etc.
- A lot of use cases are relevant for the edge deployments so it's basically like a **three-tier architecture**.
- But this three-tier architecture is not the traditional three-tier that we are familiar of.
- There is no app server, no database, no middle layer, and there is no front end, so this is not a traditional three tier architecture.

Edge Computing Architecture

Data Source Tier

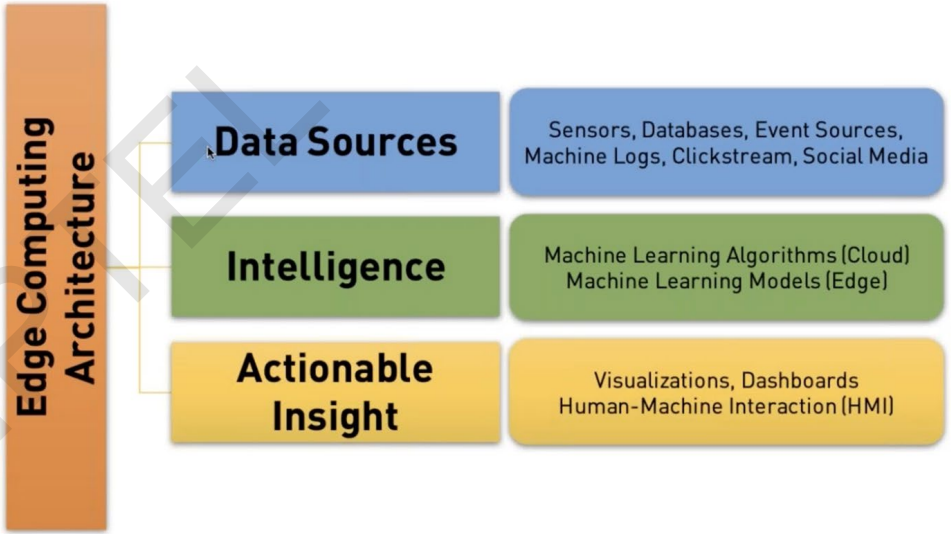
- The first tier is the Data Source Tier
- In industrial IOT environment, this could be a set of devices that are generating the data.
- These are nothing but original endpoint, from where the data is acquired or the origin of the data.



Edge Computing Architecture

Intelligence Tier

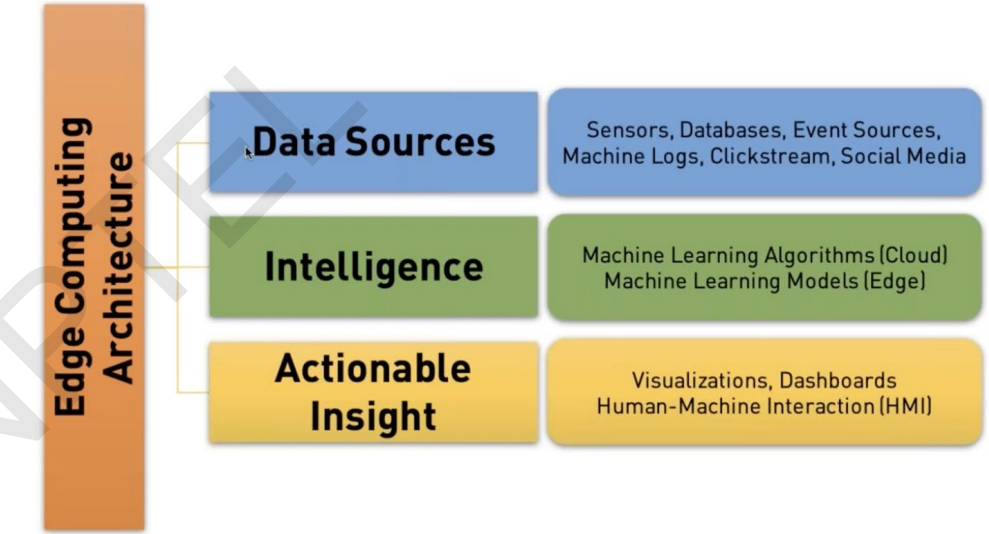
- Then there is an intelligent tier:
- Responsible for running the machine learning models.
- This intelligent tier cuts across the cloud and the edge so there is a very well-defined boundary between edge and cloud where the training takes place on the cloud and the inferencing is run on the edge.
- Collectively, this overlap between the cloud and the edge is this intelligence layer.



Edge Computing Architecture

Actionable Insight Tier

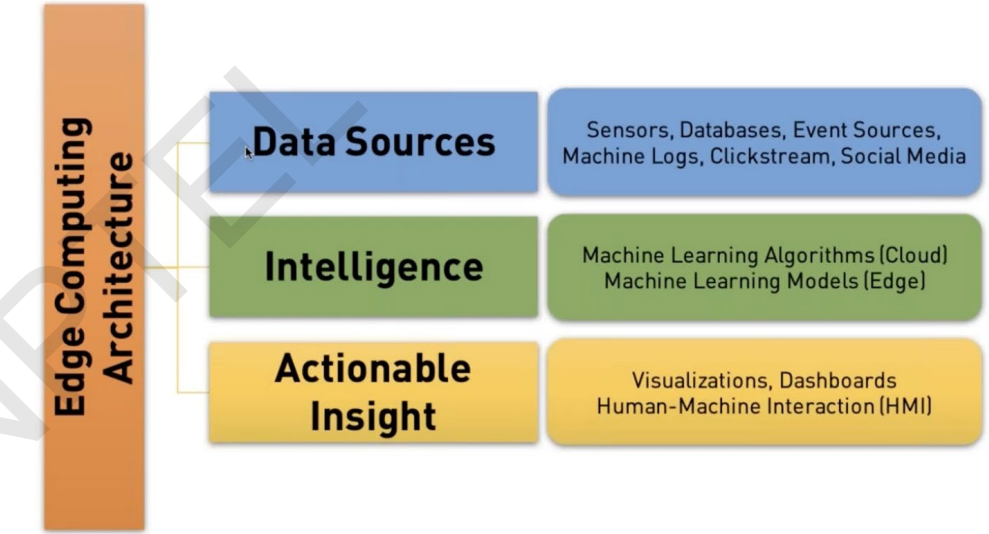
- Then there is an actionable insight layer:
- Responsible for sending an alert to the relevant stakeholders or populating the dashboards and showing some visualizations or even the edge taking an action to immediately shut down a faulty machine or controlling an actuator and again the actionable insight takes place on the edge so this is not the physical boundary.



Edge Computing Architecture

Summary

- In Summary, you logically look at the whole architecture so there is a data source which is the original endpoint from where the data is acquired or the origin of the data.
- Then there is an intelligence layer where the constant training and inferencing takes place.
- Then there is an insight layer where you actually visualize the outcome from the intelligence and also perform actions based on those insights so that is one way of visualizing edge computing.



Conclusion

We covered the following topics,

- In depth concepts of Edge Computing
  - Edge makes distributed cloud
  - Edge mimics the public cloud platform capabilities and Move cloud service closer to data-source
  - Edge reduces the latency by avoiding the round-trip and brings in the data sovereignty
- Building Blocks of Edge Computing
- Three tier architecture of Edge Computing
  - Data Source
  - Intelligence
  - Actionable Insight

Thank You!

Thank You!

References

No references were used.

Week 1 Lecture 4

45 mins

Edge Computing Paradigms

Dr. Rajiv Misra, Professor
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
rajivm@iitp.ac.in

Contents

- Types of Edge Computing - Thick-Edge, Thin-Edge and Micro-Edge
- Paradigms of Edge-Computing
  - Cloudlet and Micro Data Centers
  - Fog Computing, Mobile (Multi-Access)
  - Edge Computing (MEC)
  - Collaborative End-Edge-Cloud Computing.
- Latency Models for Cloud Edge System

Edge Computing Paradigms

Edge computing is a term used to describe intelligent computational resources located close to the source of data consumption or generation.

Edge computing has become an important solution to break the bottleneck of emerging technologies by virtue of its advantages of **reducing data transmission, improving service latency and easing cloud computing pressure.**

The edge computing architecture will become an important complement to the cloud, even replacing the role of the cloud in some scenarios.

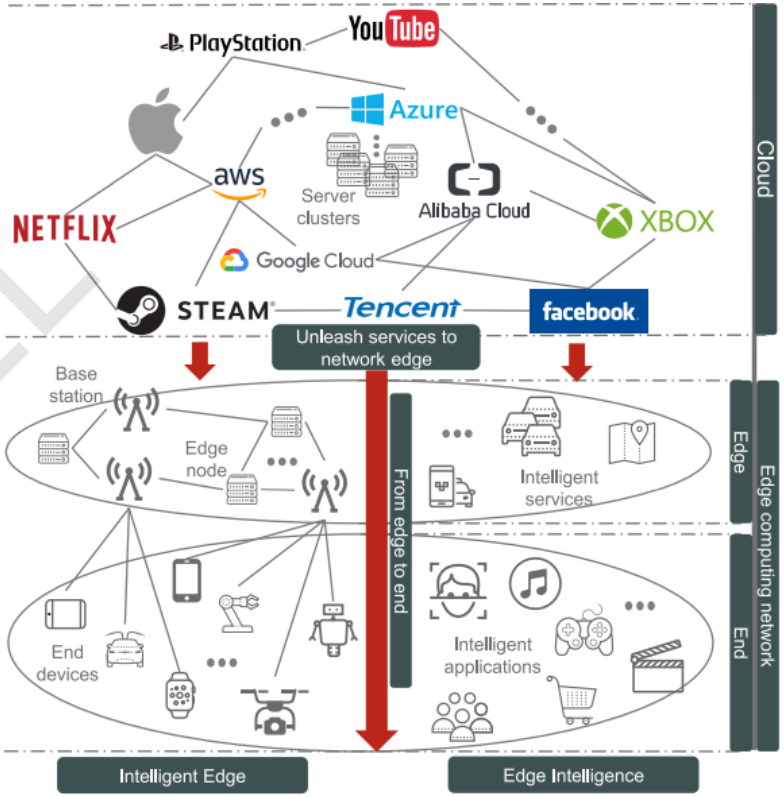


Figure Sources: Edge Intelligence: Edge Computing and Machine Learning (https://viso.ai/edge-ai/edge-intelligence-deep-learning-with-edge-computing/)

Edge Computing Paradigms

- With the proliferation of computing and storage devices, from server clusters in cloud data centres (the cloud) to personal computers and smartphones, we are now in an information-centric era in which computing is ubiquitous and computation services are overflowing from the cloud to the edge.
- 50 billion IoT devices will be connected to the Internet by 2020. On the other hand, Cisco estimates that nearly 850 Zettabytes (ZB) of data will be generated each year outside the cloud by 2021, while global data center traffic is only 20.6 ZB.
- This indicates that data sources for big data are also undergoing a transformation: from large-scale cloud data centers to an increasingly wide range of edge devices.
- However, existing cloud computing is gradually unable to manage these massively distributed computing power and analyze their data:
    - **a large number of computation tasks need to be delivered to the cloud for processing**, which undoubtedly poses serious challenges on network capacity and the computing power of cloud computing infrastructures;
    - **many new types of applications**, e.g., cooperative autonomous driving, have strict or tight delay requirements that the cloud would have difficulty meeting since it may be far away from the users.

Edge Computing Paradigms

Edge computing emerges as an attractive alternative, especially to host computation tasks as close as possible to the data sources and end users. Certainly, edge computing and cloud computing are not mutually exclusive. Instead, the edge complements and extends the cloud. Compared with cloud computing only, the main advantages of edge computing combined with cloud computing are three folds:

1) **backbone network alleviation**, distributed edge computing nodes can handle a large number of computation tasks without exchanging the corresponding data with the cloud, thus alleviating the traffic load of the network;

2) **agile service response**, services hosted at the edge can significantly reduce the delay of data transmissions and improve the response speed;

3) **powerful cloud backup**, the cloud can provide powerful processing capabilities and massive storage when the edge cannot afford.

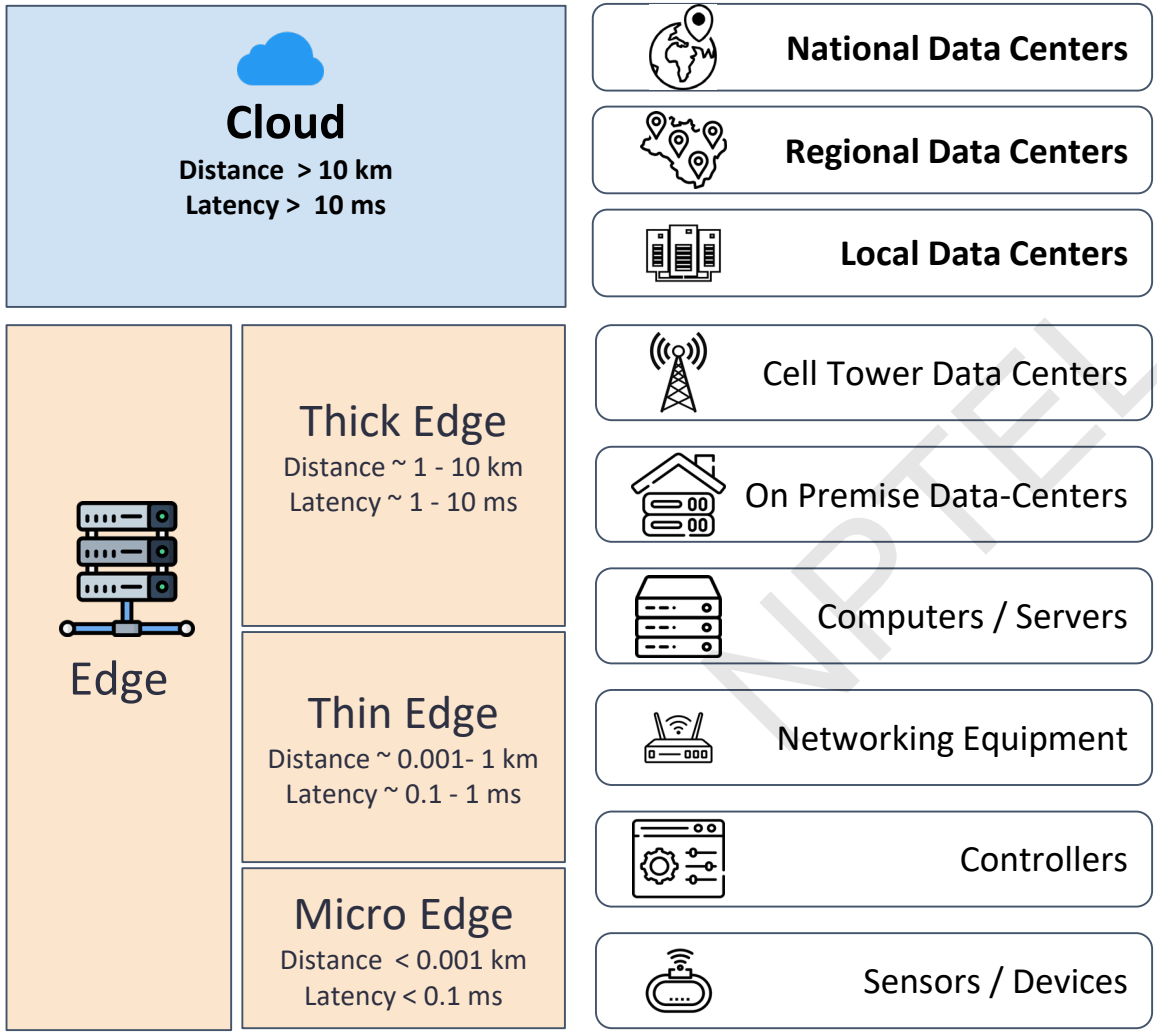
As a typical and more widely used new form of applications, various deep learning-based intelligent services and applications have changed many aspects of people's lives due to the great advantages of Deep Learning (DL) in the fields of Computer Vision (CV) and Natural Language Processing (NLP). These achievements are not only derived from the evolution of DL but also inextricably linked to increasing data and computing power.

Edge Computing Paradigms

Nevertheless, for a wider range of application scenarios, such as smart cities, Internet of Vehicles (IoVs), etc., there are only a limited number of intelligent services offered due to the following factors.

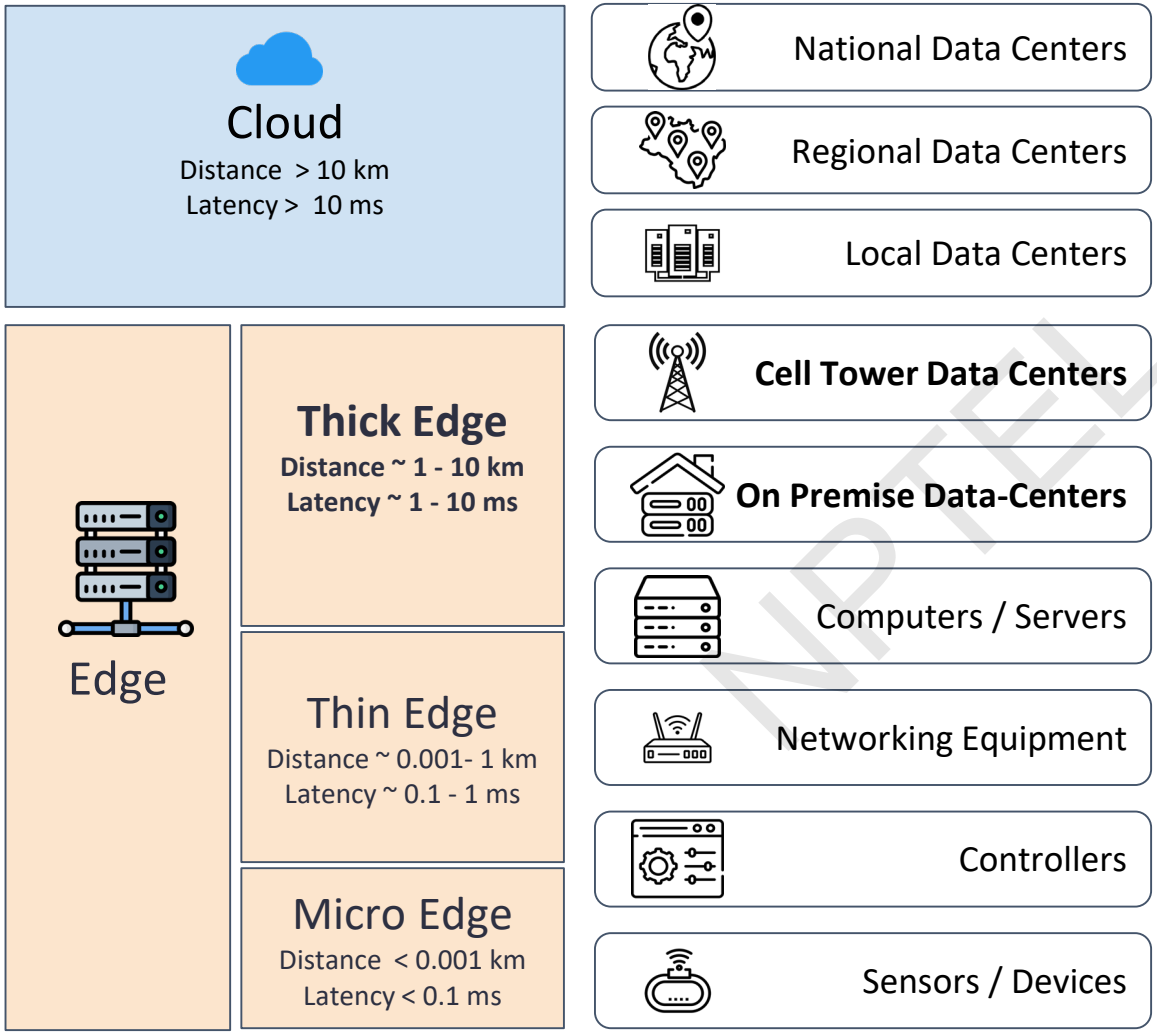
- **Cost**: training and inference of DL models in the cloud requires devices or users to transmit massive amounts of data to the cloud, thus consuming a large amount of network bandwidth;
- **Latency**: the delay to access cloud services is generally not guaranteed and might not be short enough to satisfy the requirements of many time-critical applications such as cooperative autonomous driving;
- **Reliability**: most cloud computing applications relies on wireless communications and backbone networks for connecting users to services, but for many industrial scenarios, intelligent services must be highly reliable, even when network connections are lost;
- **Privacy**: the data required for DL might carry a lot of private information, and privacy issues are critical to areas such as smart home and cities.

Types of Edge



We shall look at 3 types of Edge deployments - **Thick** edge, **Thin** edge and **Micro** edge.

Compared to the cloud, the edge has **lower distance from the data-source (<10 km)** and can support **low latency services (<10 ms)**.

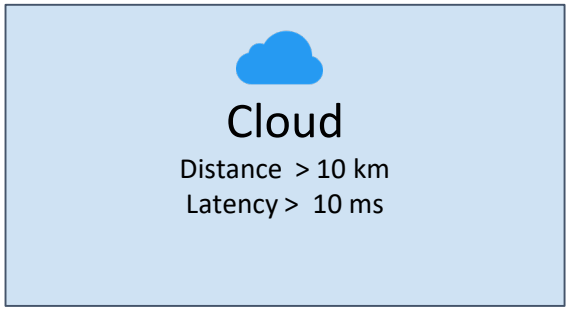


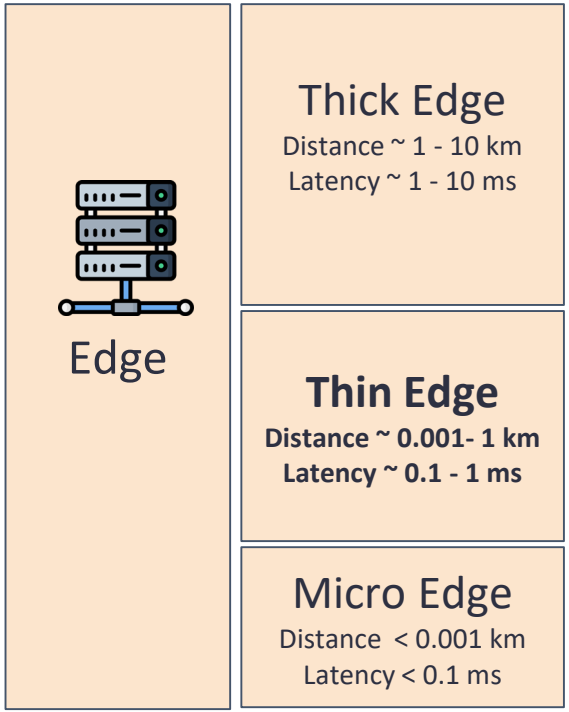
Types of Edge

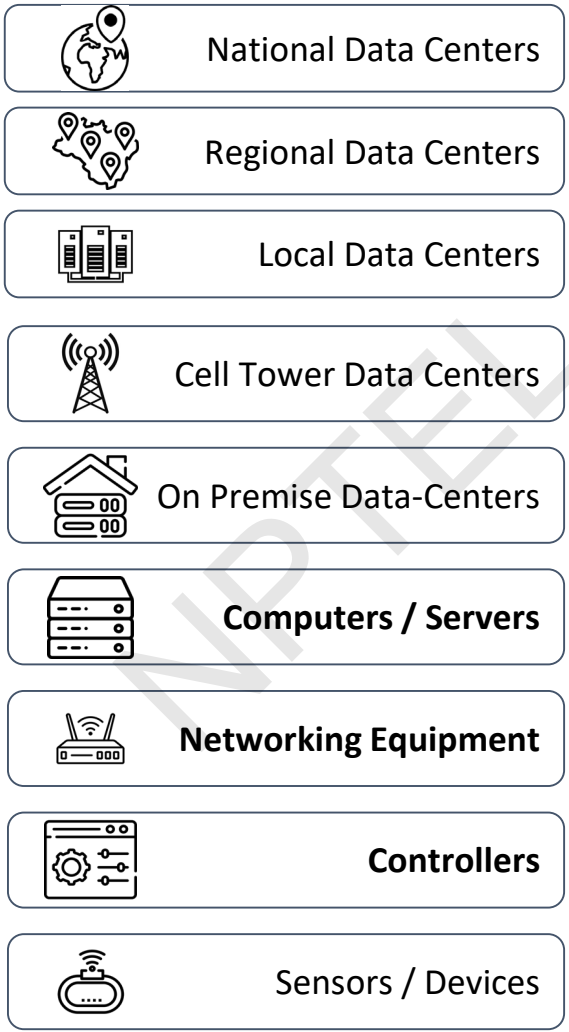
Thick edge describes compute resources (typically located within a data center) which are equipped with components designed to handle compute intensive tasks such as data storage and analysis. There are two types of compute resources located at the "thick" edge:

1. Cell tower data centers, which are rack based compute resources located at the base of cell towers

2. On prem data centers, which are rack-based compute resources located at the same physical location as the sensors generating the data







Types of Edge

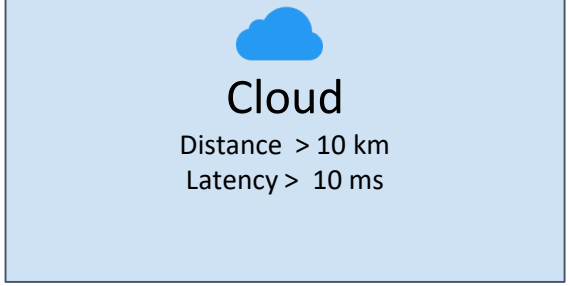
Thin edge describes the intelligent controllers, networking equipment and computers that aggregate data from the sensors / devices generating data. Resources are typically equipped with middle-tier processors and sometimes include AI components (GPUs or ASICs). The resources located at the "thin" edge:

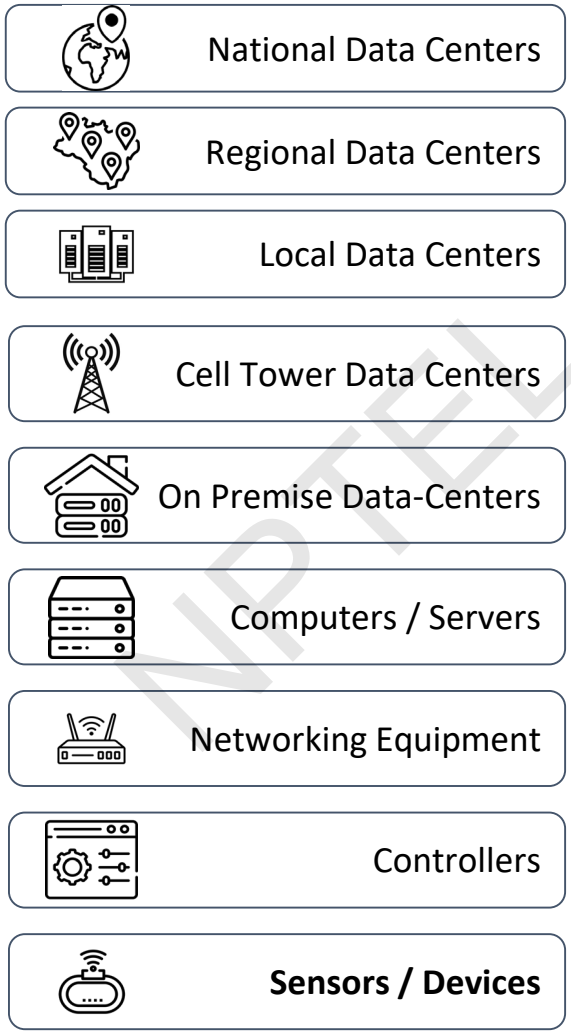
1. Computers, which are generic compute resources located outside of the data center (e.g., industrial PCs, Panel PCs)

2. Networking equipment, which are intelligent routers, switches, gateways and other communications hardware primarily used for connecting other types of compute resources.

3. Controllers, which are intelligent PLCs, RTUs, DCS and other related hardware primarily used for controlling processes.

Types of Edge



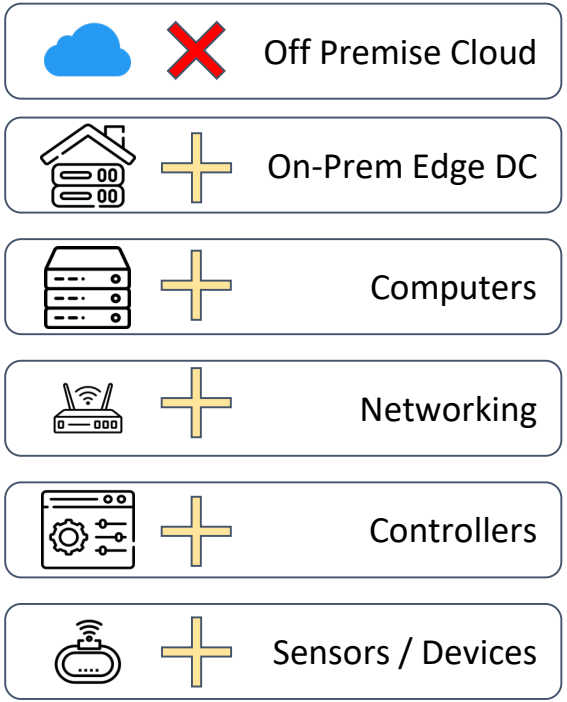


Micro edge describes the intelligent sensors / devices that generate data. "Micro edge" devices are typically equipped with low-end processors due to constraints related to cost and power consumption. Since compute resources located at the "micro edge" are the data generating devices themselves, the distance from the compute resource is essentially zero. One type of compute resource is found at the micro edge - **Sensors / Devices**, which are physical pieces of hardware that generate data and / or actuate physical objects. They are located at the very farthest edge in any architecture.

Types of Edge Architectures to realize Use-Cases

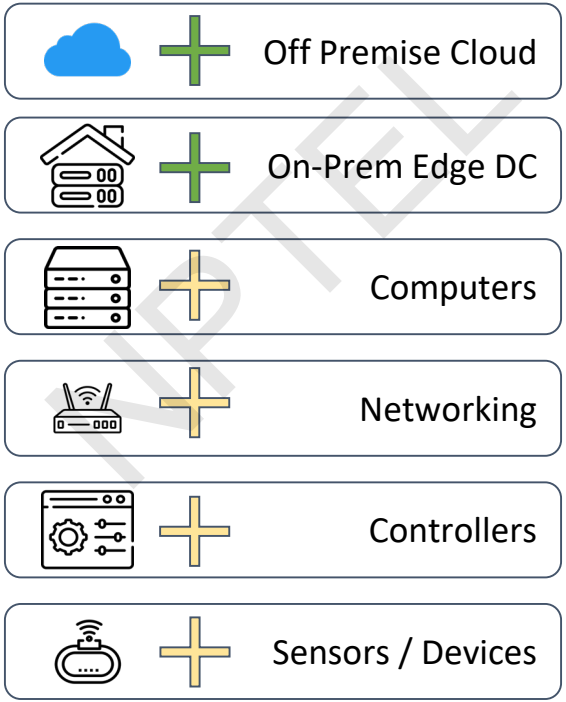
Edge-Only

Large on-prem systems (SCADA platforms, Data Lakes. etc.), no cloud



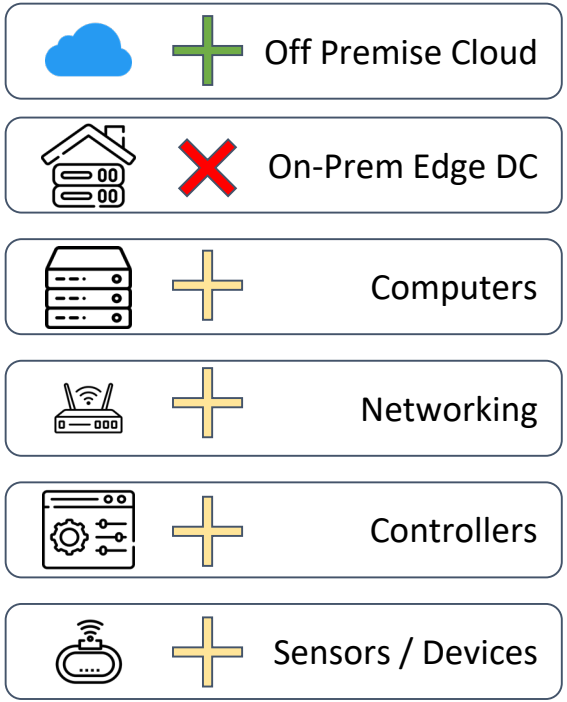
Thick Edge + Cloud

Large on-prem systems supplemented by cloud for expanded storage and analytics compute capacity



Thin/Micro Edge + Cloud

Small on-prem systems (gateways, industrial PC) that forward data to the cloud for storage and analytics



Resource always Included Resource may be Included Resource not Included

Comparison of Edge Architectures to realize Use-Cases

		Edge-Only	Thick Edge + Cloud	Thin/Micro Edge + Cloud
		✗ Off Premise Cloud + On-Prem Edge DC + Computers + Networking + Controllers + Sensors / Devices	+ Off Premise Cloud + On-Prem Edge DC + Computers + Networking + Controllers + Sensors / Devices	+ Off Premise Cloud ✗ On-Prem Edge DC + Computers + Networking + Controllers + Sensors / Devices
Use Cases	Maintenance / Downtime Reduction	**High**	**Low**	High
	Throughput / Process Optimization	High	**Low**	High
	Control / Labour Optimization	**Low**	Low	High

Paradigms of Edge Computing

In the development of edge computing, there have been various new technologies aimed at working at the edge of the network, with the same principles but different focuses, such as Cloudlet, Micro Data Centers (MDCs), Fog Computing and Mobile Edge Computing/Multi-access Edge Computing.

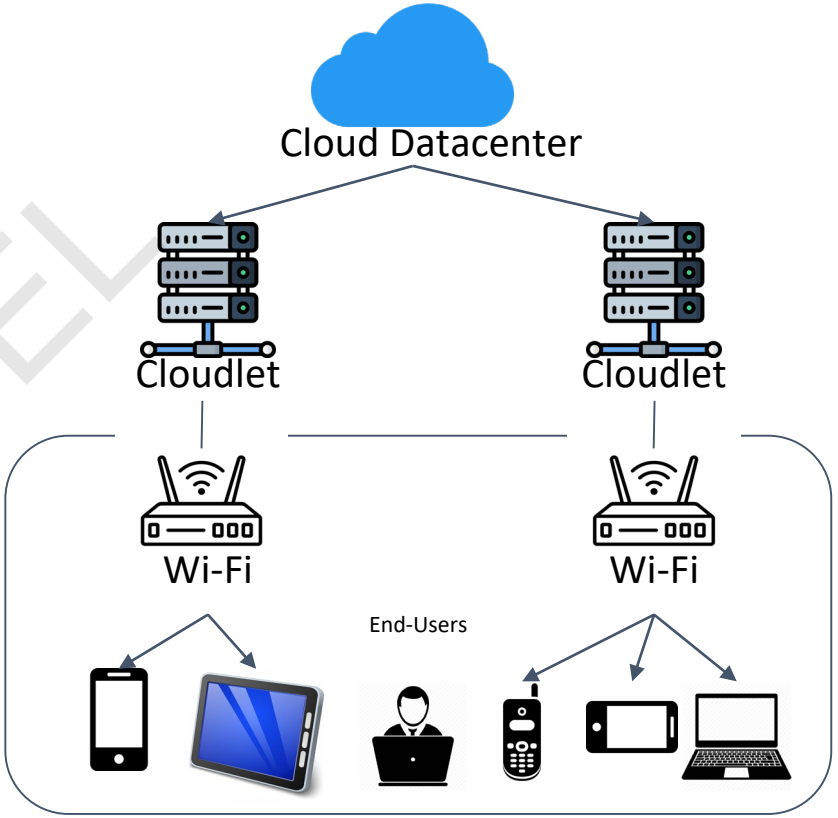
However, the edge computing community has not yet reached a consensus on the standardized definitions, architectures and protocols of edge computing. We use a common term “edge computing” for this set of emerging technologies.

The major paradigms in Edge Computing are as follows:

- **Cloudlet and Micro Data Centers**
- **Fog Computing**
- **Mobile (Multi-Access) Edge Computing (MEC)**
- **Collaborative End-Edge-Cloud Computing**

Cloudlets and Micro Data Centers

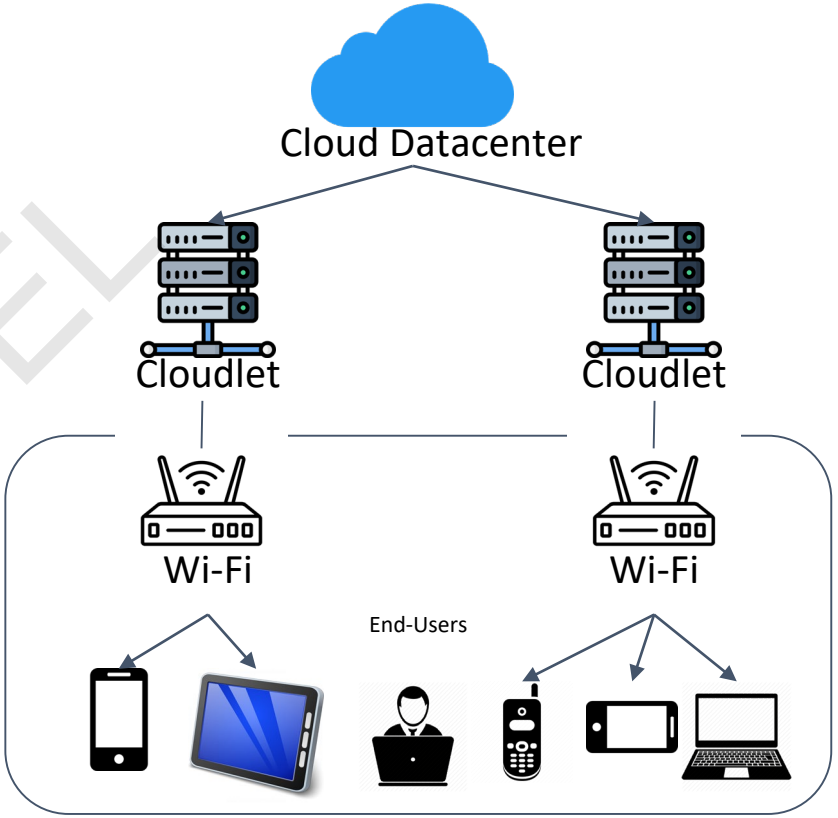
- Cloudlets can be used in environments where superior **situation awareness**, **decision making**, and **reliable connectivity** are required.
- Cloudlets can be integrated with sensors, image processing, pattern recognition, and video analytics to improve performance. Such technologies are **compute-intensive** and have stringent quality of service (QoS) requirements that can be achieved using cloudlets, which are located in users locality and can provide considerable processing capabilities and low communication latency.
- Cloudlets can **continually** provide services in the absence of an enterprise cloud and subsequently synchronize their activities with the cloud when it comes alive. In this manner, cloudlets can provide users with **persistent connectivity**.



Sources: Babar, Mohammad & Khan, Muhammad & Ali, Farman & Imran, Muhammad & Shoaib, Muhammad. (2021). **Cloudlet Computing: Recent Advances, Taxonomy, and Challenges**. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3059072.

Cloudlets and Micro Data Centers

- Cloudlets offer numerous utilities, such as lowering communication **latency** and improving **connectivity**.
- Cloudlets use Wi-Fi connections and extend the battery life of mobile devices by **offloading** compute-intensive tasks to a cloudlet for processing.
- Cloudlets, combined with an enterprise cloud, such as Google, are beneficial, as they provide applications and services to users.
- Moreover, cloudlets do not need to reach an enterprise cloud to **synchronize** themselves with others.
- However, when reaching an enterprise cloud is necessary, cloudlets process most of the data and send less traffic toward the cloud, resulting in low data storage requirements, low bandwidth, and reduced communication **latency**, **delays**, and **jitters**



Sources: Babar, Mohammad & Khan, Muhammad & Ali, Farman & Imran, Muhammad & Shoaib, Muhammad. (2021). **Cloudlet Computing: Recent Advances, Taxonomy, and Challenges**. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3059072.

Cloudlets and Micro Data Centers

- Similar to Cloudlets, **Micro Data Centers** are also designed to complement the cloud. The idea is to package all the computing, storage, and networking equipment needed to run customer applications in one enclosure, as a stand-alone secure computing environment, for applications that require lower latency or end devices with limited battery life or computing abilities.

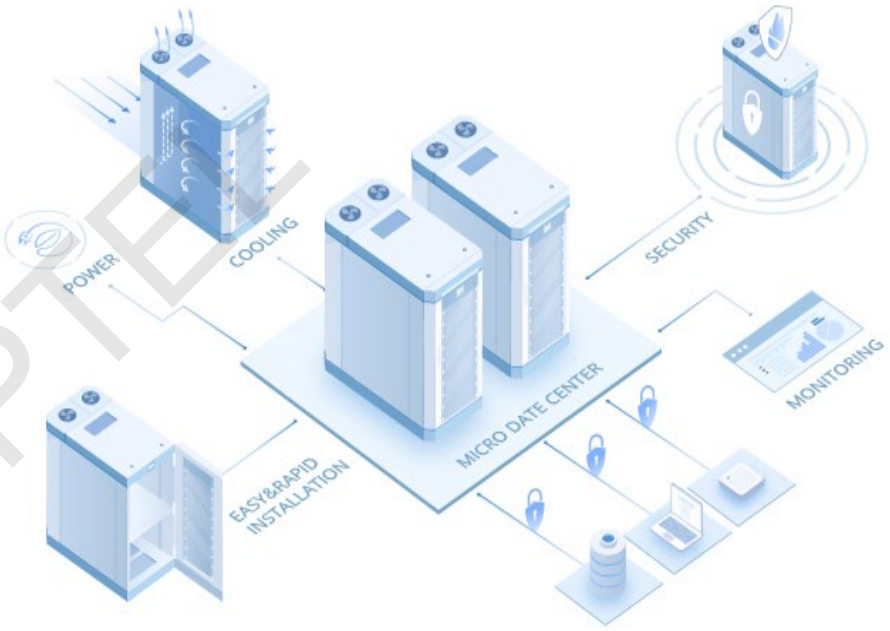


Figure Source: Micro Data Center – The Future Ready Choice for Modern Enterprise IT (https://community.fs.com/article/micro-data-center-and-edge-computing)

Fog Computing

- Fog Computing assumes a fully distributed multi-tier cloud computing architecture with billions of devices and large-scale cloud data centers.
- While cloud and fog paradigms share a similar set of services, such as computing, storage, and networking, the deployment of fog is targeted to specific geographic areas.
- In addition, fog is designed for applications that require real-time responding with less latency, such as interactive and IoT applications.
- Unlike Cloudlet, MDCs and MEC, fog computing is more focused on IoTs.

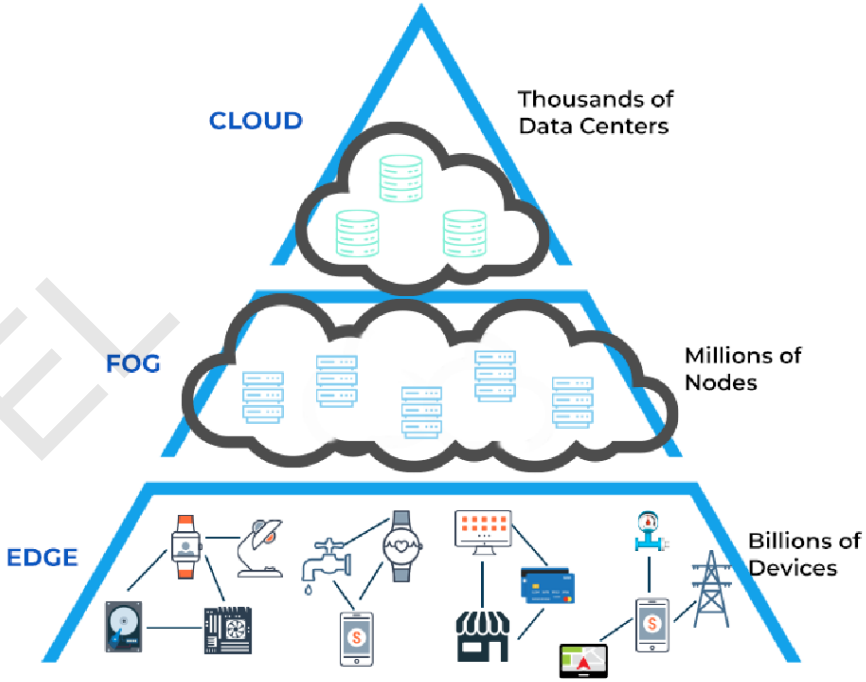


Figure Source: Edge Computing vs. Fog Computing: Key Comparisons (https://www.spiceworks.com/tech/cloud/articles/edge-vs-fog-computing/)

Comparison

Metric	Cloud	Fog	Edge
Deployment	Centralized	Decentralized	Decentralized
Distance from end user	Large	Small	Very Small
Computational Power	Pervasive	Limited	Limited
Efficiency	Low	High	Very High
Latency	High	Low	Ultra-Low
Processing location	Remote Datacenter	Remote Fog-Nodes	Local Edge Servers
Storage Capacity	High	Limited	Limited

Source: Daisy Nkele Molokomme, Adeiza James Onumanyi and Adnan M. Abu-Mahfouz, Edge Intelligence in Smart Grids: A Survey on Architectures, Offloading Models, Cyber Security Measures, and Challenges

Mobile (Multi-Access) Edge Computing (MEC)

- Mobile Edge Computing places computing capabilities and service environments at the edge of cellular networks.
- It is designed to provide lower latency, context and location awareness, and higher bandwidth.
- Deploying edge servers on cellular Base Stations (BSs) allows users to deploy new applications and services flexibly and quickly.
- The European Telecommunications Standards Institute (ETSI) further extends the terminology of MEC from Mobile Edge Computing to Multi-access Edge Computing by accommodating more wireless communication technologies, such as Wi-Fi.

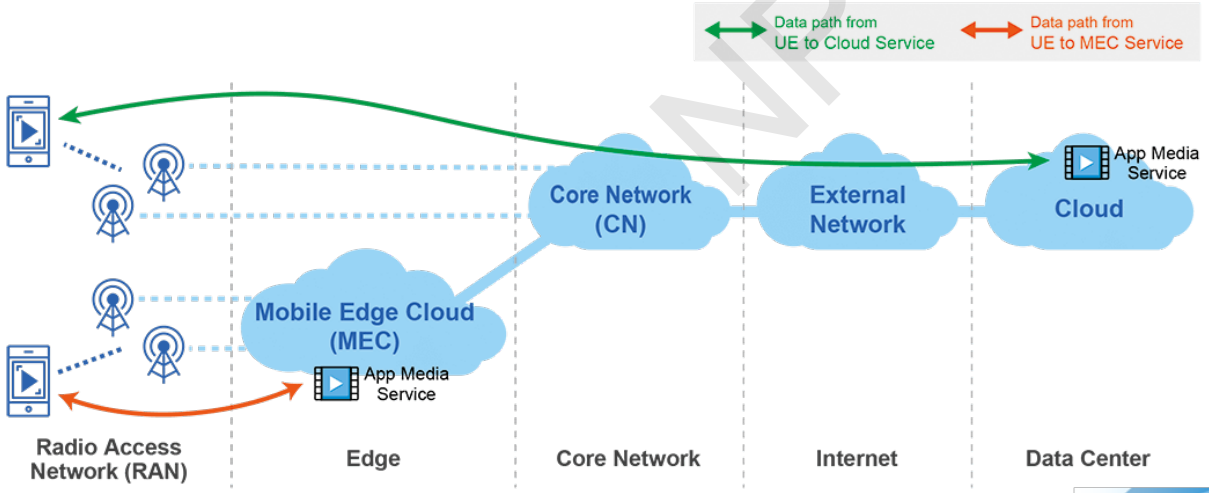


Figure Source: Multi-Access Edge Computing (https://devopedia.org/multi-access-edge-computing)

Collaborative End-Edge-Cloud Computing

- While cloud computing is created for processing computation-intensive tasks, it cannot guarantee the delay requirements. throughout the whole process from data generation to transmission to execution.
- Moreover, independent processing on the end or edge devices is limited by their computing capability, power consumption, and cost bottleneck. Therefore, **collaborative end-edge-cloud computing**, is emerging as an important trend.
- For a computation-intensive task, it will be reasonably segmented and dispatched separately to the end, edge and cloud for execution, reducing the execution delay of the task while ensuring the accuracy of the results.
- The focus of this collaborative paradigm is not only the successful completion of tasks but also achieving the optimal balance of energy consumption, server loads, transmission and execution delays.

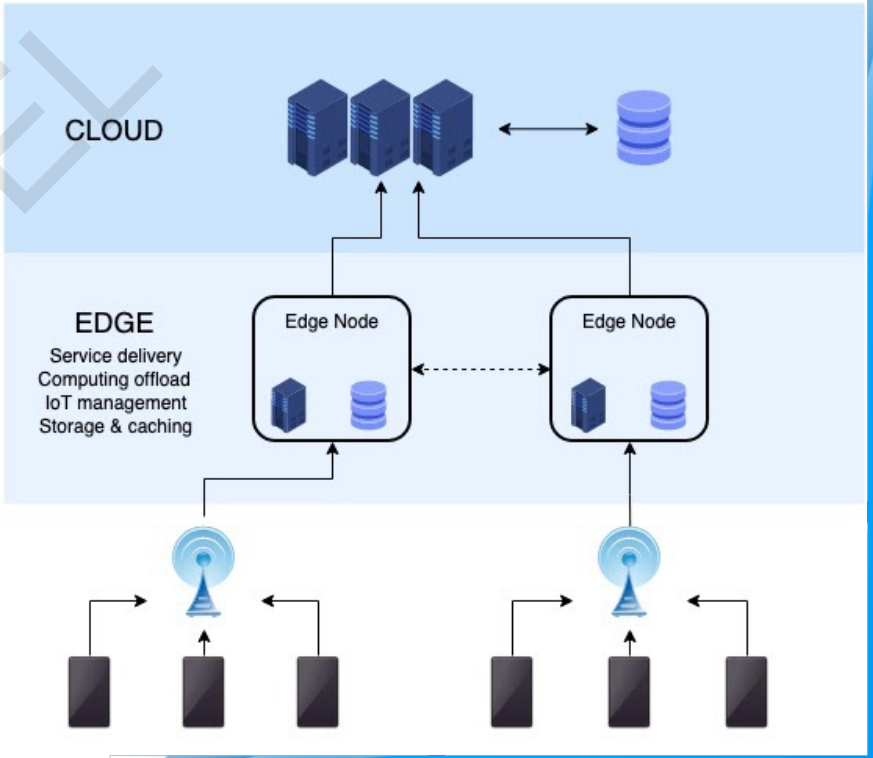


Figure Source: https://www.bmc.com/blogs/edge-artificial-intelligence/

Introduction to Edge-Cloud System Architecture

Introduction to Edge-Cloud System Architecture

- Number of Internet of Things (IoT) devices connected and the data produced by these devices has increased dramatically.

- This would require offloading tasks from the IoT devices to release heavy computation and storage to the resource-rich nodes such as Edge Computing and Cloud Computing.

- However, different offloading strategies require edge-cloud architecture which have a different impact on the service time performance of new set of applications such as Autonomous Vehicles, Augmented Reality
- (AR), online video games and Smart CCTV.

- An Edge-Cloud system architecture supports scheduling offloading tasks of IoT applications in order to minimize the enormous amount of transmitting data in the network. For time performance measurements in latency sensitive applications introduces the offloading latency models to investigate the delay of different offloading scenarios/schemes and explores the effect of computational and communication demand on each one.

Challenges in minimizing Computational and Communication for delay sensitive applications

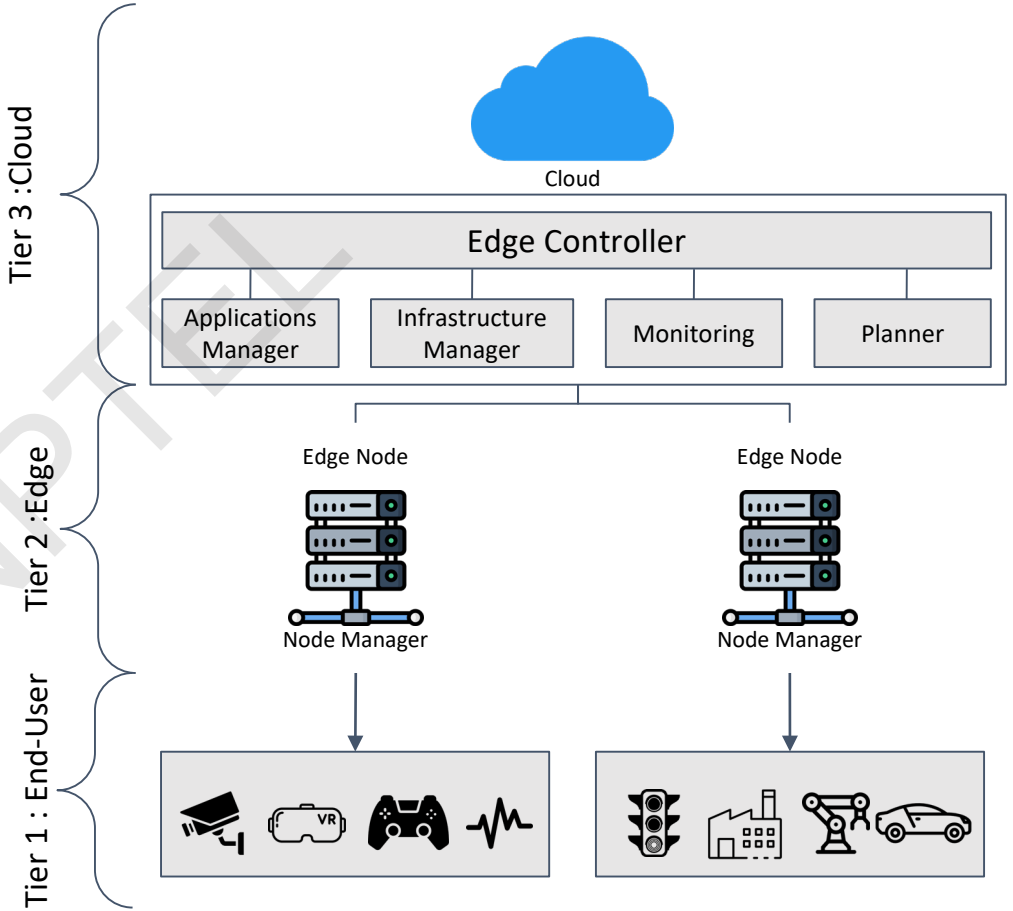
- For enhancing customer experience and accelerating job execution, IoT task offloading enables mobile end devices to release heavy computation and storage to the resource-rich nodes in collaborative Edges or Clouds.
- Resource management at the Edge-Cloud environment is challenging because it deals with several complex factors.
- Additionally, how different service architecture and offloading strategies quantitatively impact the end-to-end service time performance of IoT applications becomes complex due to dynamic and unpredictable assortment of interconnected virtual and physical devices. Latency-sensitive applications have various changing characteristics, such as computational demand and communication demand.
- Consequently, the latency depends on the scheduling policy of applications offloading tasks as well as where the jobs will be placed.
- Therefore, Edge-Cloud resource management requires to consider these characteristics in order to meet the requirements of latency-sensitive applications.

Challenges in minimizing Computational and Communication for delay sensitive applications

- We shall discuss delay model for latency-sensitive applications within the Edge-Cloud environment.
- Provide a detailed analysis of the main factors of service latency, considering both applications characteristics and the Edge-Cloud resources.
- The approach is to minimize the overall service time of latency-sensitive applications and enhance resource utilization in the Edge-Cloud system.
- We will discuss in summary, the following:
    - An Edge-Cloud system architecture that includes the required components to support scheduling offloading tasks of IoT applications.
    - An Edge-Cloud latency models that show the impact of different tasks' offloading schemes for time-sensitive applications in terms of end-to-end service times.

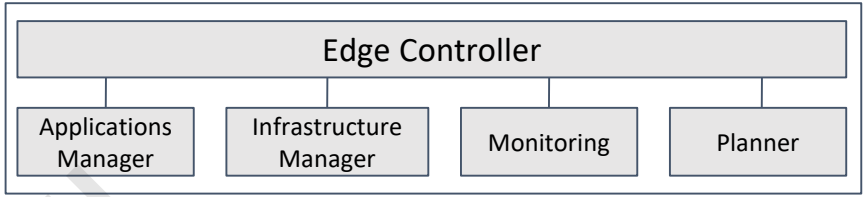
Edge-Cloud system architecture

- Edge-Cloud system architecture supports scheduling offloading tasks of IoT applications in order to minimize the enormous amount of transmitting data in the network.
- Latency models for the delay analysis of different offloading scenarios/schemes that explores the effect of computational and communication demand on each task.
- Different offloading decisions within the Edge-Cloud system can lead to various service times due to the computational resources and communications types and open for using AI/ML on task offloading issues in the Edge-Cloud environment.



Edge Controller/Orchestrator in Edge-Cloud System

- **Edge Controller/Orchestrator** (EC) is a centralized component responsible for planning, deploying and managing application services in the Edge-Cloud system.
- EC communicates with other components in the architecture to know the status of resources in the system (e.g., available and used), the number of IoT devices, their applications' tasks and where IoT tasks have been allocated (e.g., Edge or Cloud).
- The location of the Edge Controller can be deployed in any layer between Edge and Cloud.
- EC consists of the following components: **Application Manager, Infrastructure Manager, Monitoring and Planner.**



- For example, EC act as an independent entity in the edge layer that manages all the edge nodes in its control. It is also responsible for scheduling the offloading tasks in order to satisfy applications' users and Edge-Cloud System requirements. The EC is synchronizing its data with the centralized Cloud because if there is any failure, other edge nodes can take EC responsibility from the cloud .

Edge Controller/Orchestrator Components

Application manager: is responsible for managing applications running in the Edge-Cloud system. This includes requirements of application tasks, such as the amount of data to be transferred, the amount of computational requirement (e.g., required CPU), the latency constraints and the number of application users for each edge node.

Infrastructure Manager: The role of the infrastructure manager is to be in charge of the physical resources in the entire Edge-Cloud system. For instance, processors, networking and the connected IoT devices for all edge nodes. Edge-Cloud is a virtualized environment; thus, this component responsible for the VMs as well. In this context, this component provides the EC with the utilization level of the VMs.

Monitoring: The main responsibility of this component is to monitor application tasks (e.g., computational delay and communication delay) and computational resources (e.g., CPU utilization) during the execution of applications' tasks in the Edge-Cloud system. Furthermore, detecting the tasks' failures due to network issues or the shortage of computational resources.

Planner The main role of this component is to propose the scheduling policy of the offloading tasks in the Edge-Cloud system and the location where they will be placed (e.g., local edge, other edges or the cloud). This offloading tasks works on this component and passes its results to EC for execution.

Latency Sensitive Applications

Latency-sensitive applications have high sensitivity to any delays accrue in communication or computation during the interaction with the Edge-Cloud system.

- For instance, the IoT device sends data to the point that processing is complete at the edge node or the cloud in the back end of the network, and the subsequent communications are produced by the network in response to receive the results.
- Third, low-priority applications, which can be offloaded and not vital as high-priority applications (e.g., infotainment, multimedia, and speech processing).

- For example, self-driving cars consist of several services, classified these services in categories based on their latency-sensitivity, quality constraints and workload profile (required communication and computation).
- First, critical applications, which must be processed in the car's computational resources, for instance, autonomous driving and road safety applications.
- Second, high-priority applications, which can be offloaded but with minimum latency, such as image aided navigation, parking navigation system and traffic control.
- Third, low-priority applications, which can be offloaded and not vital as high-priority applications (e.g., infotainment, multimedia, and speech processing).

Latency-sensitive applications

Industry	Applications
Industrial automation	Industrial Control
	Robot Control
	Process Control
Healthcare Industry	Remote Diagnosis
	Emergency Response
	Remote Surgery
Entertainment Industry	Immersive Entertainment
	Online Gaming
Transport Industry	Driver Assistance Applications
	Autonomous Driving
	Traffic Management
Manufacturing Industry	Motion Control
	Remote Control
	AR and VR Applications

Latency Models for Edge-Cloud System

Latency consideration of the Edge-Cloud system is an essential step towards developing an effective scheduling policy and modelling the various offloading decisions for IoT tasks that can increase the Quality of Service (QoS).

With the increasing number of IoT devices, the amount of produced data, service time has been considered as one of the most important factors to be handled in Edge Computing. Additionally, using Edge Computing will enhance application performance in terms of overall service time comparing to the traditional Cloud system.

Firstly, task allocation in the Edge-Cloud system is not only two, but could be on any edge nodes. Moreover, edge nodes connected in a loosely coupled way on heterogeneous wireless networks, making the process of resource management and the offloading decision more sophisticated.

Secondly, given that task processing is allocated among multiple edge nodes working collectively and the cloud, it is challenging to make an optimal offloading decision. The latency models to investigate the delay of different offloading scenarios/schemes.

We discuss the following offloading schemes:

(1) offloading to the local edge,

(2) offloading to the local edge with the cloud and

(3) offloading to the local edge, other available edge nodes and the cloud.

Latency Models: Latency to Local Edge/Cloudlet

In one-level offloading system, is offloading to “Cloudlet” or “Local Edge” aims to provide a micro-data center that supports IoT devices within a specific area (such as a coffee shop, mall center and airport). In here, IoT devices can offload their tasks to be processed on the edge or cloud, as an example.

This offloading scheme provides ultra-low latency due to the avoidance of network backhaul delays.

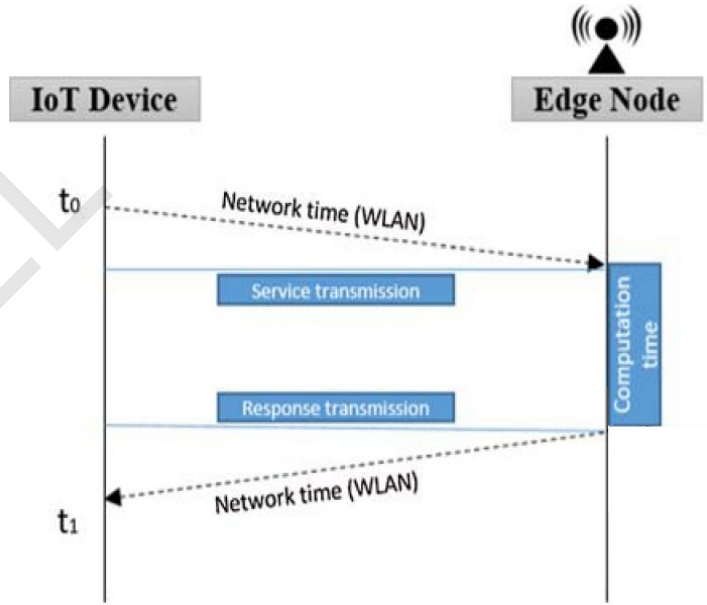
The end-to-end service time composed of two delays, network delay and computational delay.

The network delay consists of the time of sending the data to the edge and the time to receive the output from the edge to the IoT device.

The computation time is the time from arriving the task to the edge node until the processing has completed.

Therefore, the end-to-end service time latency is the sum of communication delay and computational delay, which can be calculated as follows:

$$L_{Local\_edge} = t_{te\_up} + t_{ce} + t_{te\_down}$$



Symbol	Meaning
t_{te_up}	Transmission Time between the IoT to the Edge node for uploading
t_{te_down}	Transmission Time between the IoT to the Edge node for Downloading
t_{ce}	Computation time in the Edge node

Latency Models: Latency to Local Edge with the Cloud

In this offloading scheme, rather than relying on only one Edge node, the IoT tasks can be processed collaboratively between the connected Edge node and the cloud servers. This has combined benefits of both Cloud and Edge Computing, where the cloud has a massive amount of computation resources, and the edge has lower communication time.

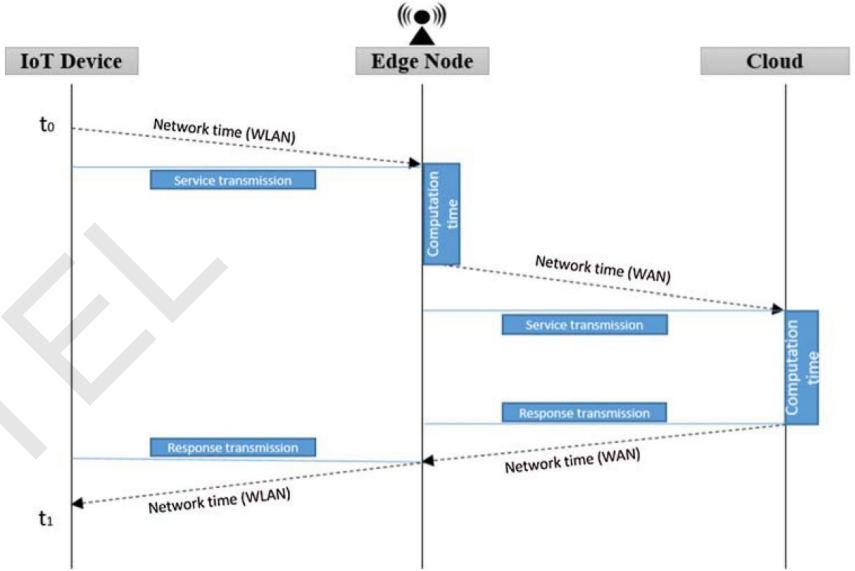
IoT devices send the computation tasks to the connected edge where a part of these tasks forwarded to the cloud. Once the cloud finishes the computation, it will send the result to the edge, and the edge will send it to the IoT devices.

The end-to-end service time composed of:

- communication time i.e., the time between the IoT device to the edge node and the time between edge nodes to the cloud.
- computation time i.e., processing time in the edge and processing time in the cloud.

The end-to-end service time can be calculated as:

$$L_{L\_C} = t_{te\_up} + t_{ce} + t_{tc\_up} + t_{cc} + t_{tc\_down} + t_{te\_down}$$



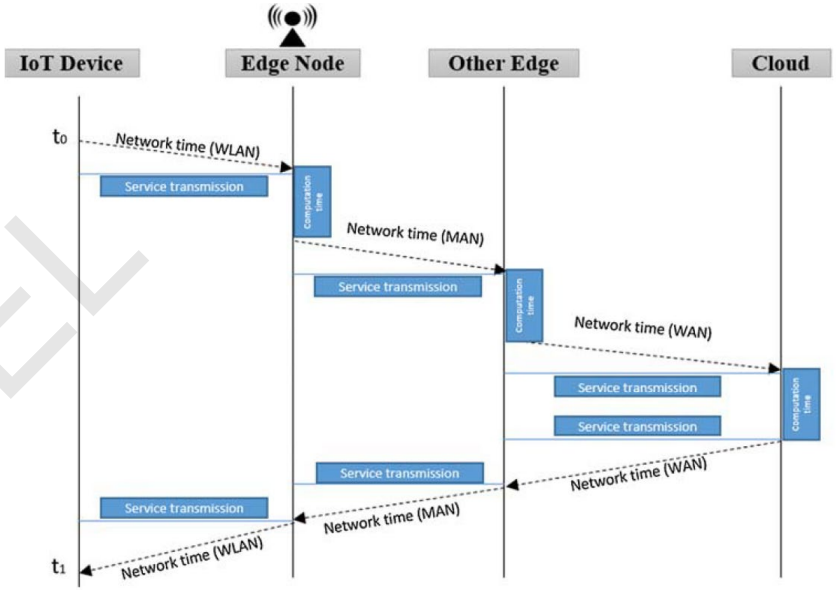
Symbol	Meaning
t_{te_up}	Transmission Time between the IoT to the Edge node for uploading
t_{te_down}	Transmission Time between the IoT to the Edge node for Downloading
t_{ce}	Computation time in the Edge node
t_{tc_up}	Transmission Time between the Edge node to the Cloud for uploading
t_{tc_down}	Transmission Time between the Edge node to the Cloud for Downloading

Latency Models: Latency to Multiple Edge Nodes with the Cloud

Three-level offloading scheme aims to utilize more resources at the edge layer and support the IoT devices in order to reduce the overall service time.
It adds another level by considering other available computation resources in the edge layer.

The end-to-end service time composed of:

- communication time i.e., the time between the IoT device to the edge node, the time between edge node to other collaborative edge node and the time between edge nodes to the cloud)
- computation time i.e., processing time in the edge, processing time in other collaborative edge node and processing time in the cloud.



Symbol	Meaning
t_{cc}	Computation time in the Cloud
t_{teo_up}	Transmission Time between the Edge node to other nearby Edge nodes for uploading
t_{teo_down}	Transmission Time between the Edge node to other nearby Edge nodes for Downloading
t_{ceo}	Computation time in the other nearby Edge node

Thus, the end-to-end service time can be calculated as follows:

$$L_{three-off} = [t_{te_{up}} + t_{ce} + t_{teo_{up}} + t_{ceo} + t_{tc_{up}} + t_{cc} + t_{tc_{down}} + t_{teo_{down}} + t_{te_{down}}]$$

Conclusion

We covered the following topics,

- Overview of Edge Intelligence and Intelligent services at the edge.
- The different Edges such as
  - Thick-Edge
  - Thin-Edge
  - Micro-Edge
- The different paradigms of Edge-Computing such as
  - Cloudlet and Micro Data Centers
  - Fog Computing
  - Mobile Edge Computing (MEC)
  - Collaborative End-Edge-Cloud Computing.
- An Edge-Cloud system architecture that includes the required components to support scheduling offloading tasks of IoT applications.
- An Edge-Cloud latency models that show the impact of different tasks' offloading scenarios/schemes for time-sensitive applications in terms of end-to-end service times.

Thank You!

Thank You!

References

- Babar, Mohammad & Khan, Muhammad & Ali, Farman & Imran, Muhammad & Shoaib, Muhammad. (2021). Cloudlet Computing: Recent Advances, Taxonomy, and Challenges. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3059072.
- Daisy Nkele Molokomme, Adeiza James Onumanyi and Adnan M. Abu-Mahfouz, Edge Intelligence in Smart Grids: A Survey on Architectures, Offloading Models, Cyber Security Measures, and Challenges