Edge Architecture & Infrastructure

Chapter 3

Outline

- ❑ Edge Infrastructure
  - Four construction goals
  - Three layers of edge
  - Four capabilities
  - Edge-native progress
- ❑ Computing Models
  - MEC vs cloudlet
  - KubeEdge, K3s, Greengrass
  - Collaboration: continuity
- ❑ Networking
  - Network development process
  - Four network architectures
  - Progress & trends
- ❑ Summary & Practice
  - Summary
  - Practice

Edge Infrastructure

Edge infrastructure is built to meet four goals that the centralized cloud cannot satisfy alone.

Low latency

Process data near the source for fast, real-time response which is essential for time-sensitive applications.

Bandwidth efficiency

Filter and preprocess locally so only useful data travels upstream, easing network load.

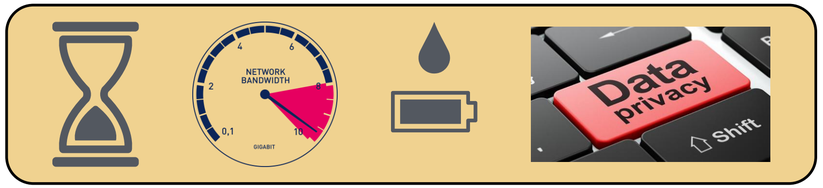
Reliability & resilience

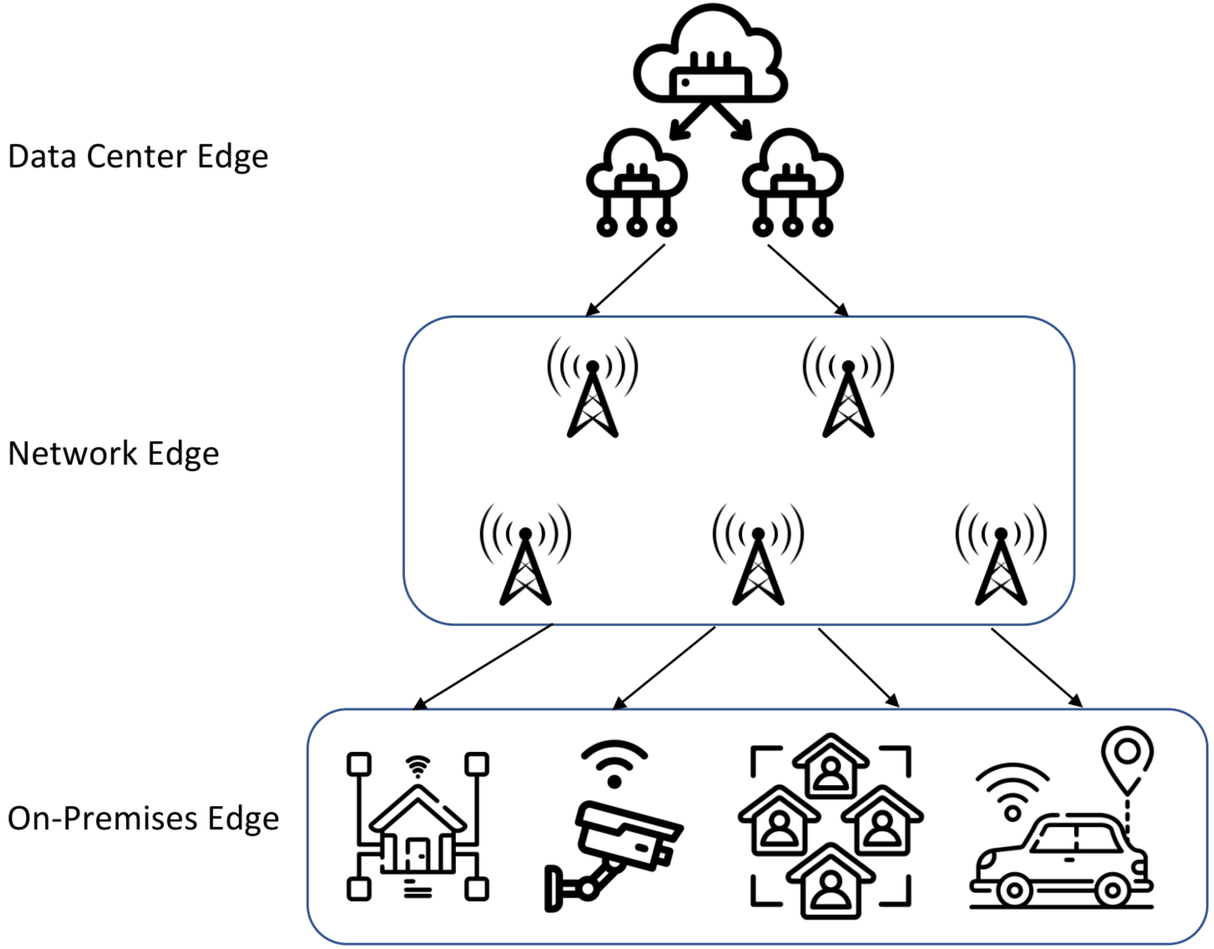
Local processing keeps operations running even when the connection to the cloud is disrupted.

Data privacy & security

Sensitive information is processed and stored locally, reducing exposure and breach risk.

Edge Infrastructure: Three Layers of Edge



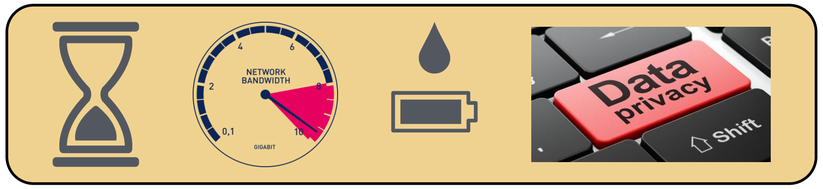


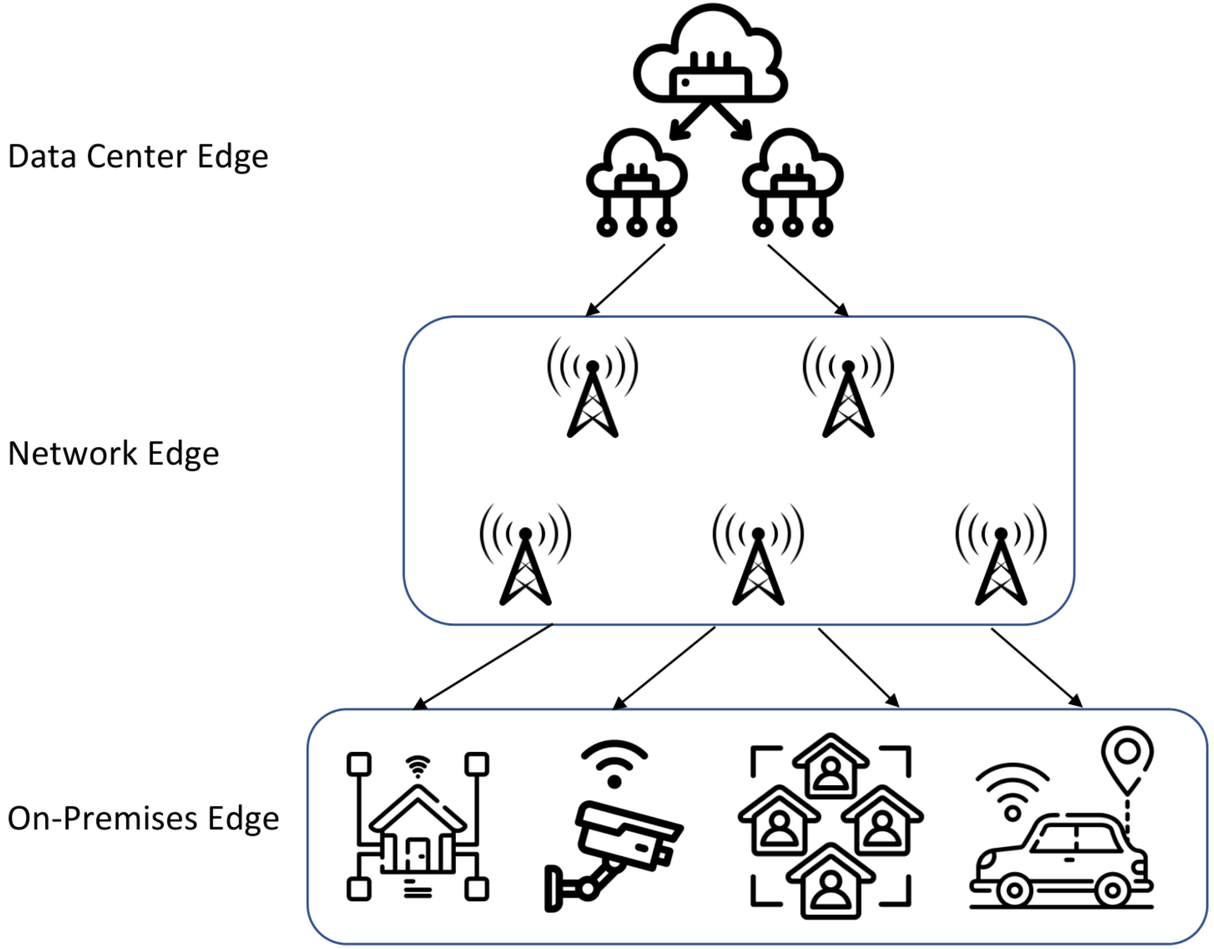
Three layers of edge

On-Premises Edge (Edge Device)

- Process at the physical location of the user/data source.
- Real-time processing and rapid response (e.g., collision prevention)

Edge Infrastructure: Three Layers of Edge



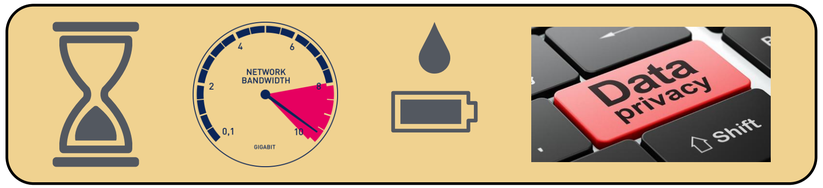


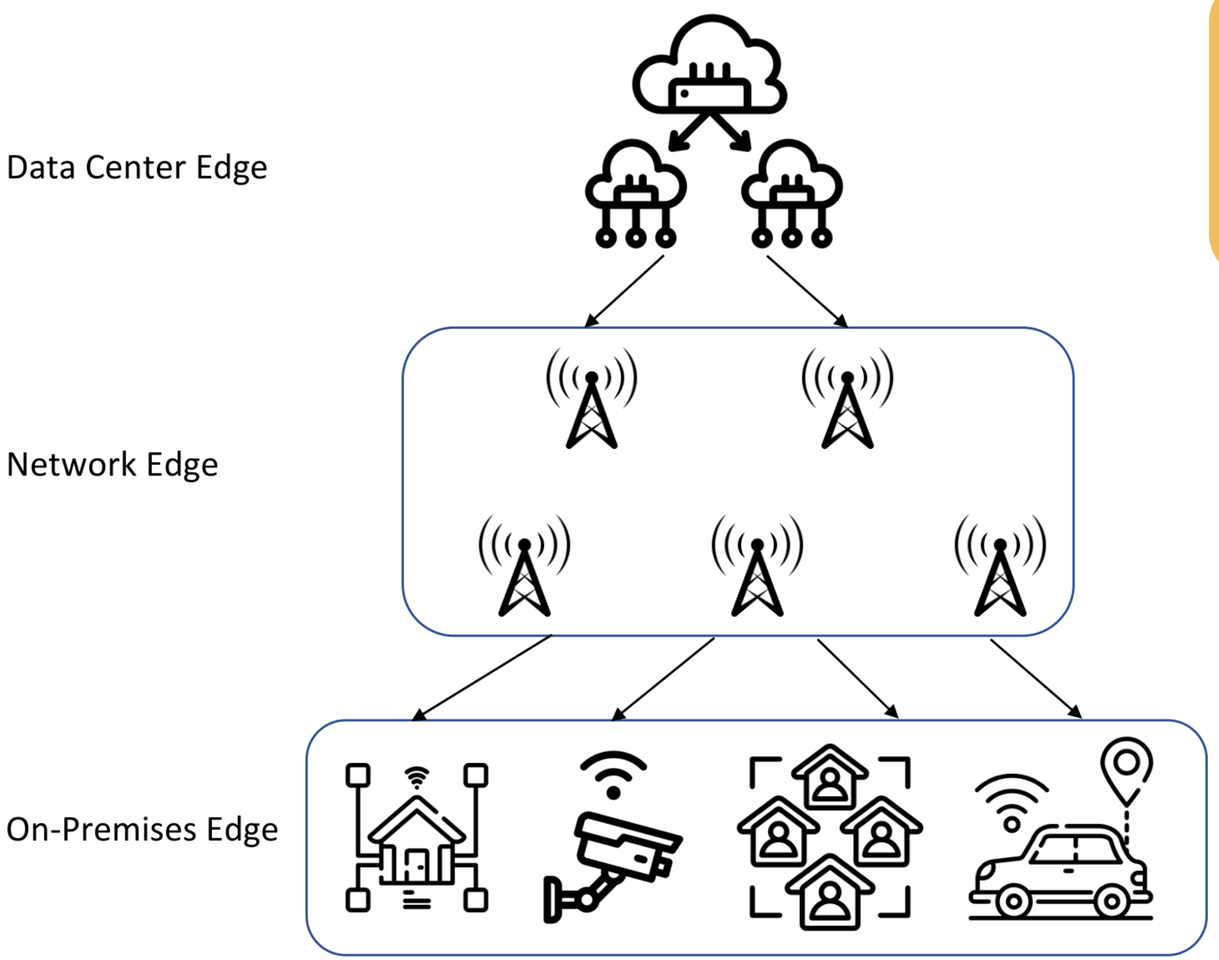
Three layers of edge

Network Edge (Edge Node)

- Expand the reach of computing by leveraging the infrastructure of telecommunication operators.
- Enable high-quality service delivery (video streaming, online gaming)
- Optimize mobile network and accelerate content delivery.
- Local servers, routers, on-site gateways, base station,…

Edge Infrastructure: Three Layers of Edge





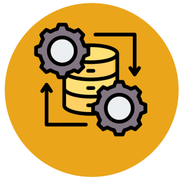
Three layers of edge

Data Center Edge

- Provide cloud-like computing power but are geographically distributed.
- Regional data centers for content delivery, gaming servers, smart city infrastructure, …

- The divisions between layers are **blurry and flexible**
  - **Overlapping** functions
    - e.g., data processing, storage, communication; different scales
  - **Dynamic** resource allocation
    - Based on current needs
  - **Distributed** nature
    - Functions can occur at multiple points along the continuum
  - **Context-dependent** design

Edge Infrastructure: Four Core Capabilities



Data process

Edge data processing involves collecting, filtering, cleansing, transforming, and aggregating data to enable real-time analytics and decision-making.

Cache and storage

Edge cache and storage provide fast, secure, and efficient data access through caching and long-term storage, helping reduce latency and support historical analysis.

Communication

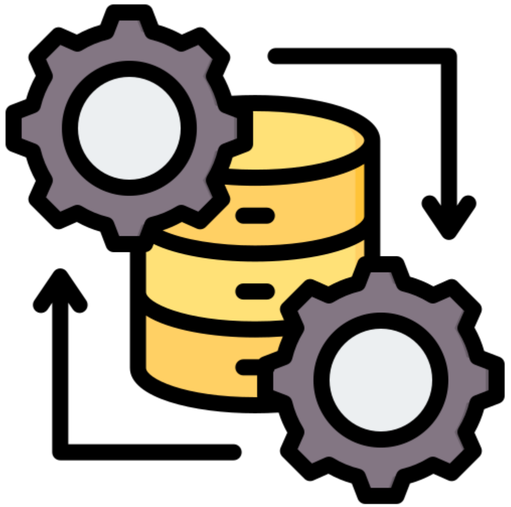
Edge communication enables fast and reliable data exchange between edge nodes and cloud servers, supporting low-latency and high-throughput applications.

Content delivery

CDNs improve content delivery speed and availability through edge replication, while supporting load balancing, routing, optimization, and security.

Together: data process + storage + networking + content delivery form a self-sufficient node that can operate independently of the cloud.

Edge Infrastructure: Four Core Capabilities

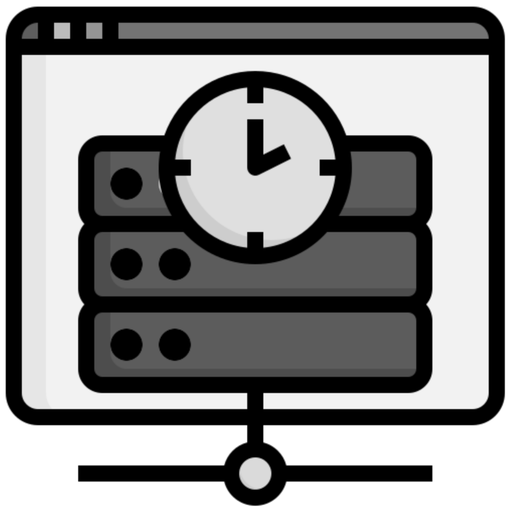


Data processing

1. **Collection** — Gather data from various sources
2. **Filtering** — Select relevant data points based on predefined criteria
3. **Cleansing** — Remove corrupt or irrelevant data to maintain data integrity
4. **Transformation** — Adjust data structure/format for downstream applications
5. **Aggregation** — Summarize data into a more manageable and insightful form for analysis

Edge Infrastructure: Four Core Capabilities

"Cache and storage is the foundational capability that supports the data lifecycle within edge infrastructure."



Cache and storage

- **Temporary storage (cache)**
  - Quick access and retrieval of frequently used data
- **Permanent storage**
  - Historical data is vital for trend analysis, long-term planning, or compliance with data retention policies
- **Design considerations**
  - Data durability, accessibility, security
  - Optimize distributed file systems, object storage, and databases for efficient data storage, management, and retrieval

Zhao, Y., Zhang, W., Zhou, L., & Cao, W. (2021). A survey on caching in mobile edge computing. Wireless Communications and Mobile Computing, 2021(1), 5565648.

Edge Infrastructure: Four Core Capabilities

Communication

- **Low-latency** communication by
  - Reduced travel distance
  - Technology support like 5G, Wi-Fi, and fiber networks
- **Improved performance** by
  - Enable local data processing
  - Improve bandwidth efficiency
  - Increase network scalability

Edge-Native: The Direction of Travel

Infrastructure is evolving from “cloud extended to the edge” toward edge-native systems designed for the edge from the ground up.

Cloud-native, pushed out

- Cloud tools adapted for the edge
- Heavier footprint, cloud assumptions
- Edge treated as a remote cloud site

→

Edge-native

- Designed for constraints & heterogeneity
- Lightweight, resilient, autonomous
- Aware of location, mobility & context

What’s driving it

Lightweight runtimes

Containers & unikernels sized for edge hardware

5G & beyond

Ultra-low latency unlocking new edge workloads

Edge AI

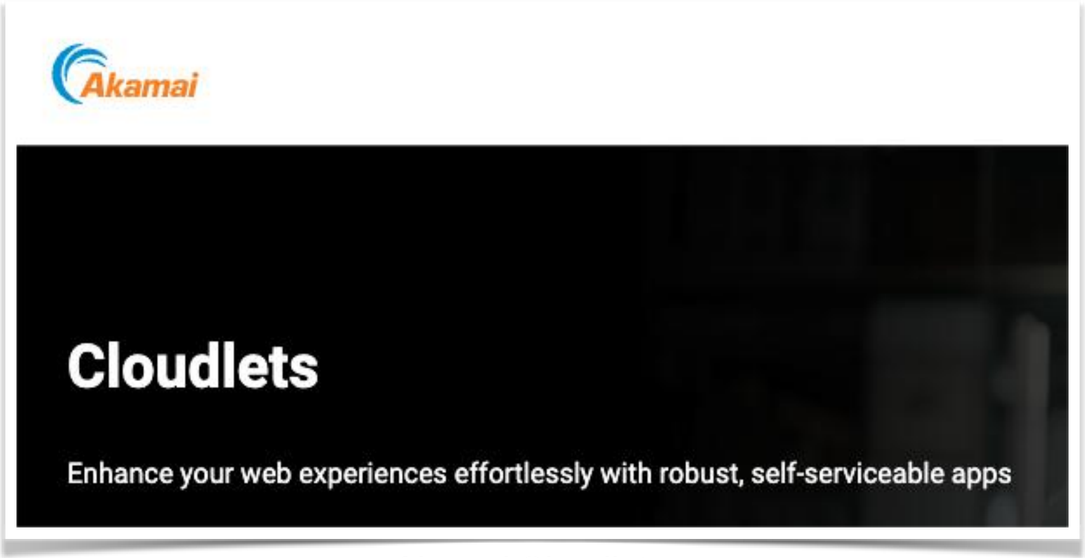
Models optimized to run inference on-device

Computing Models: Mobile/Multi-access Edge Computing (MEC)

- Mobile -> Multi-access Edge Computing

Computing Models: Cloudlet Computing

- Providing localized, powerful computing resources near mobile devices
- Serve as small-scale data centers
  - Offloading computation-intensive tasks from mobile devices to enhance performance and extend battery life



Akamai Cloudlet

Satyanarayanan, M. (2013, June). Cloudlets: at the leading edge of cloud-mobile convergence. In Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures (pp. 1-2).

Computing Models: Edge Platforms

KubeEdge

Extends Kubernetes to the edge. Syncs cloud & edge, supports offline autonomy of edge nodes, and manages devices via a device twin.

K3s

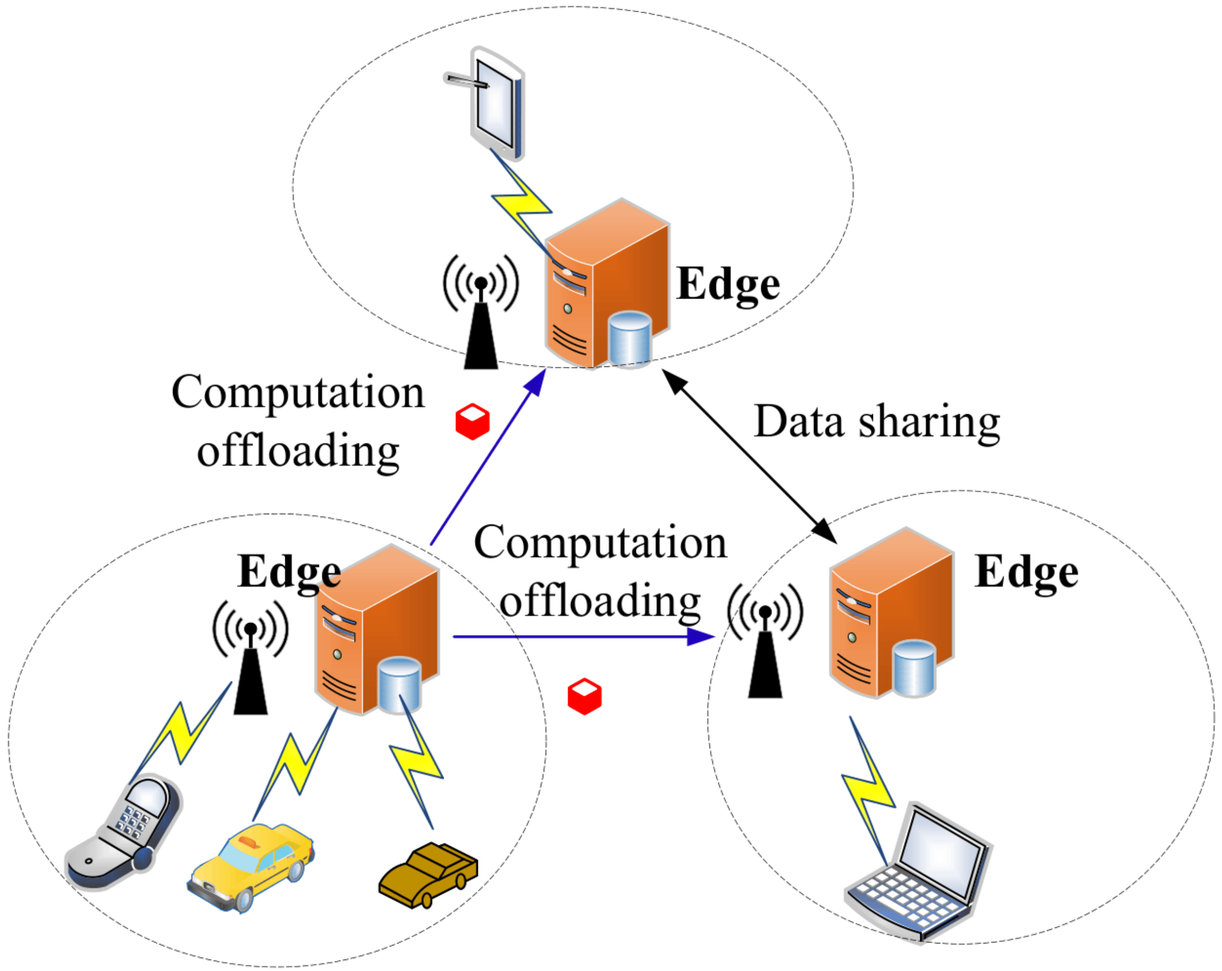
A lightweight, certified Kubernetes distribution (<100 MB) built for edge & IoT — low resource use, simple to install and operate.

AWS Greengrass

Extends AWS to edge devices — run Lambda functions, ML inference & messaging locally, syncing with the cloud when connected.

Bring familiar cloud-native tooling (containers, orchestration) to the edge, with offline autonomy for unreliable links.

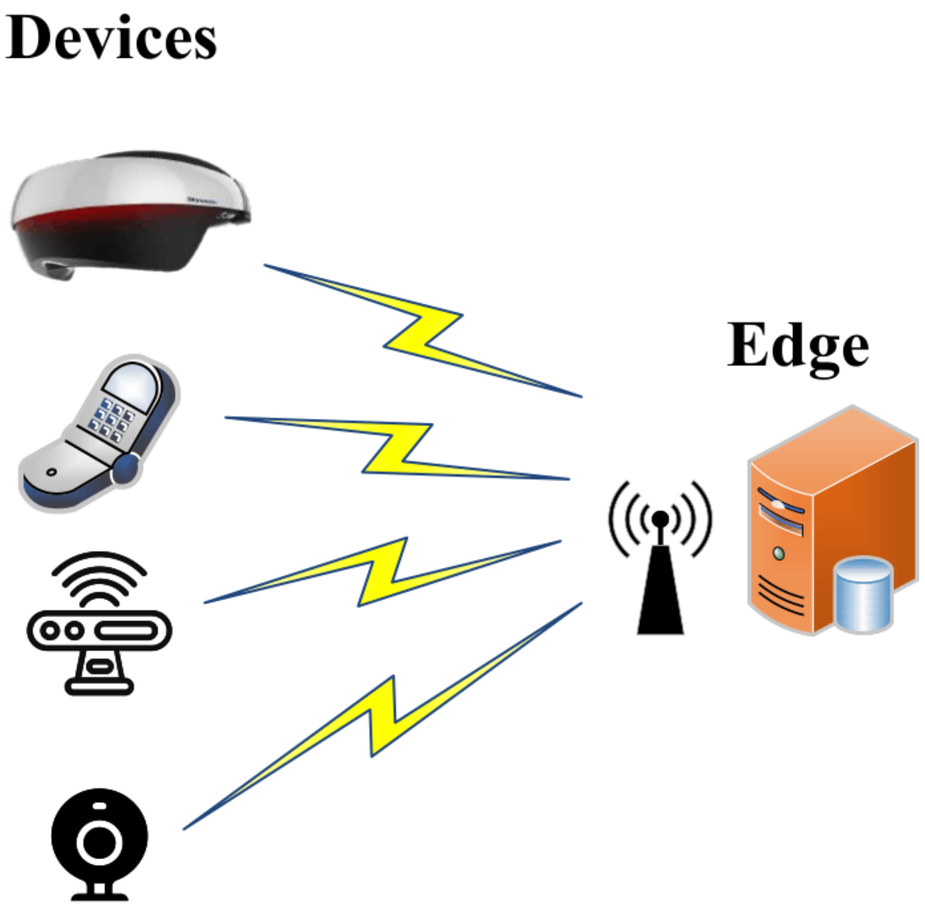
Computing Models: Edge-to-Edge



Edge-to-edge

- **Direct** interaction and coordination to enhance performance, reliability, and scalability.
  - Edge nodes communicate with each other to share data, balance workloads, and provide redundancy.
  - Efficient resource allocation and management enabled by **decentralized** protocols and frameworks.

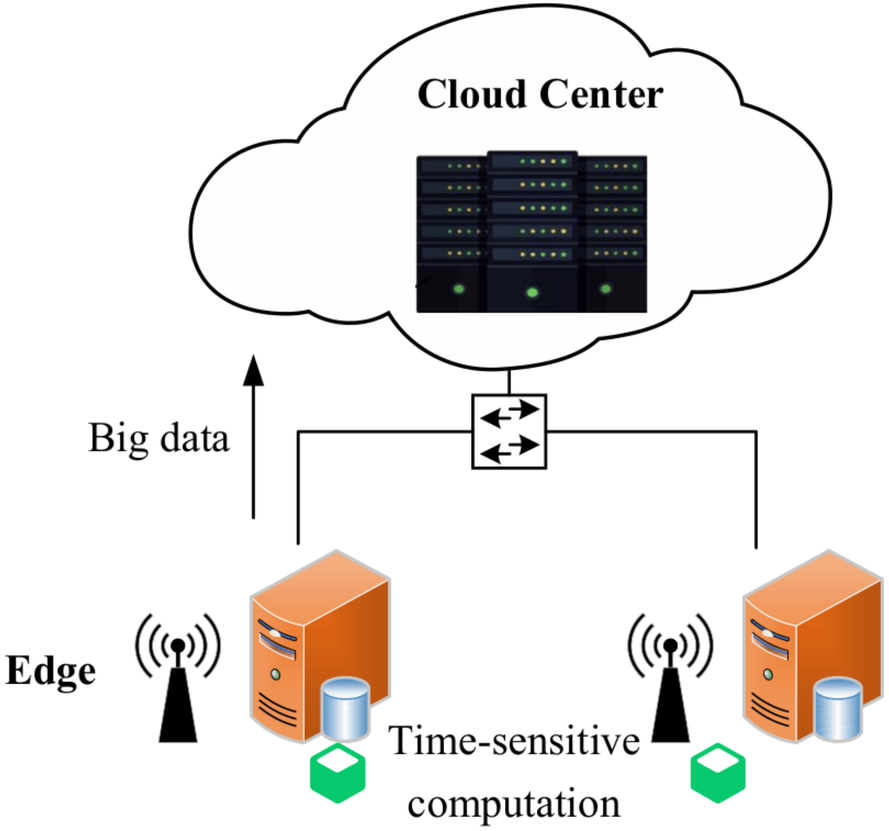
Computing Models: Edge-to-Device



Edge-to-device

- The pivotal way to improve device performance and **extend battery life**
  - Computation-intensive tasks are offloaded to nearby edge nodes.

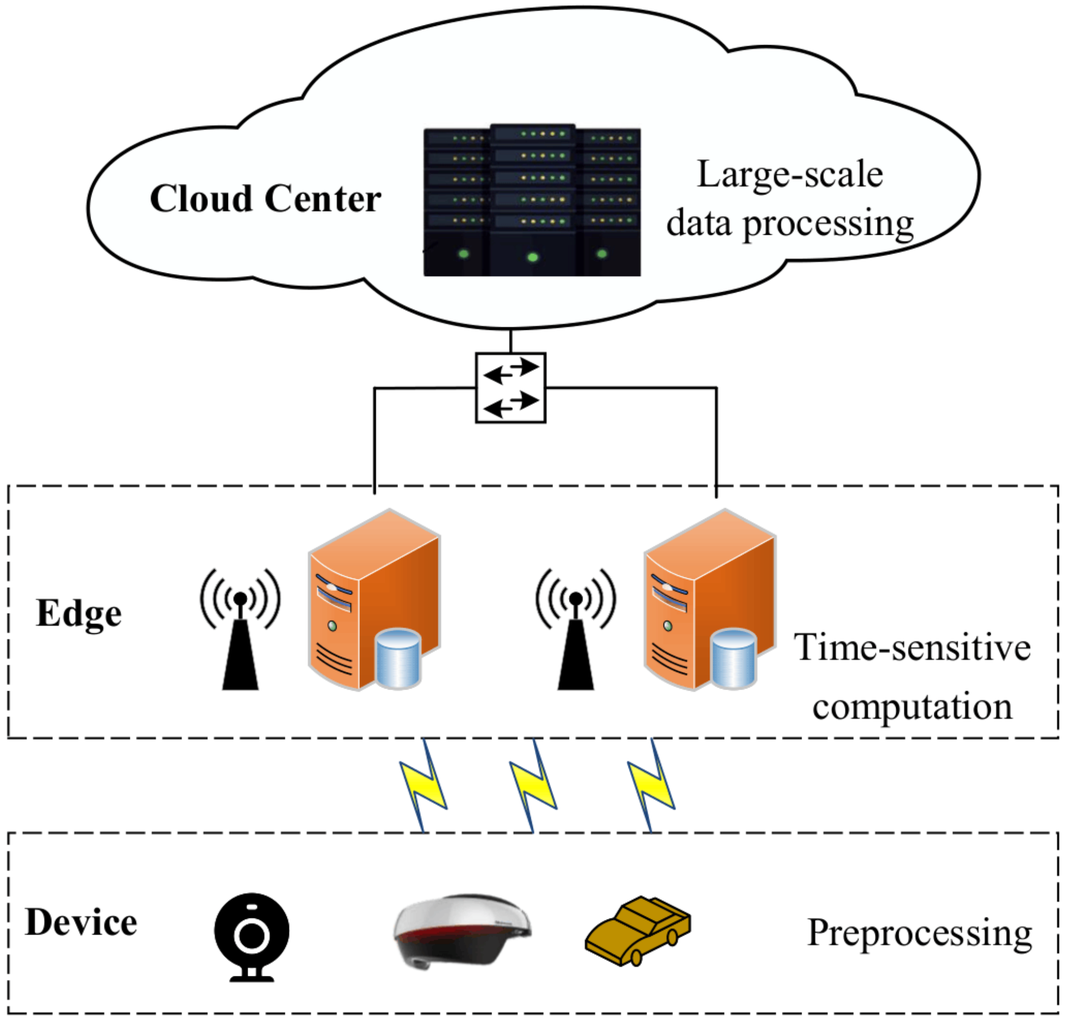
Computing Models: Edge-to-Cloud



Edge-to-cloud

- A balanced approach to data processing and storage.
  - Cloud resources can handle peak loads and extend data storage
  - Edge performs critical, time-sensitive computations

Computing Models: Cloud-Edge-Device



Cloud-Edge-Device

- A holistic approach where tasks are distributed across the cloud, edge nodes, and edge devices based on their computational requirements and latency sensitivity.
  - Cloud handles large-scale data processing and long-term storage
  - Edge nodes perform intermediate processing and provide low-latency response
  - Edge devices focus on data collection and immediate user interactions

Networking: Network Development Process

Networking and edge computing have converged through “network cloudification”

Integrate compute + network

Early work fused computing and network infrastructure inside data centers with high-speed interconnects.

NFV & virtualization

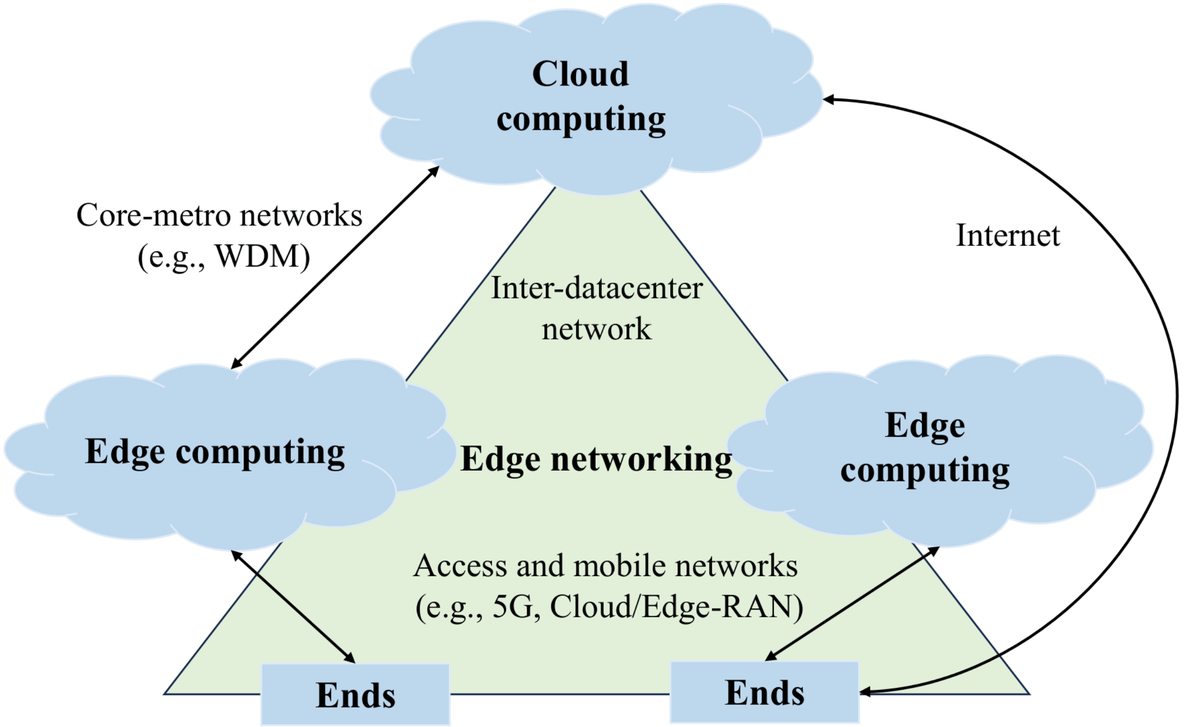
ETSI NFV abstracts network & compute over a common virtualization layer; MANO handles orchestration.

MEC + NFV integration

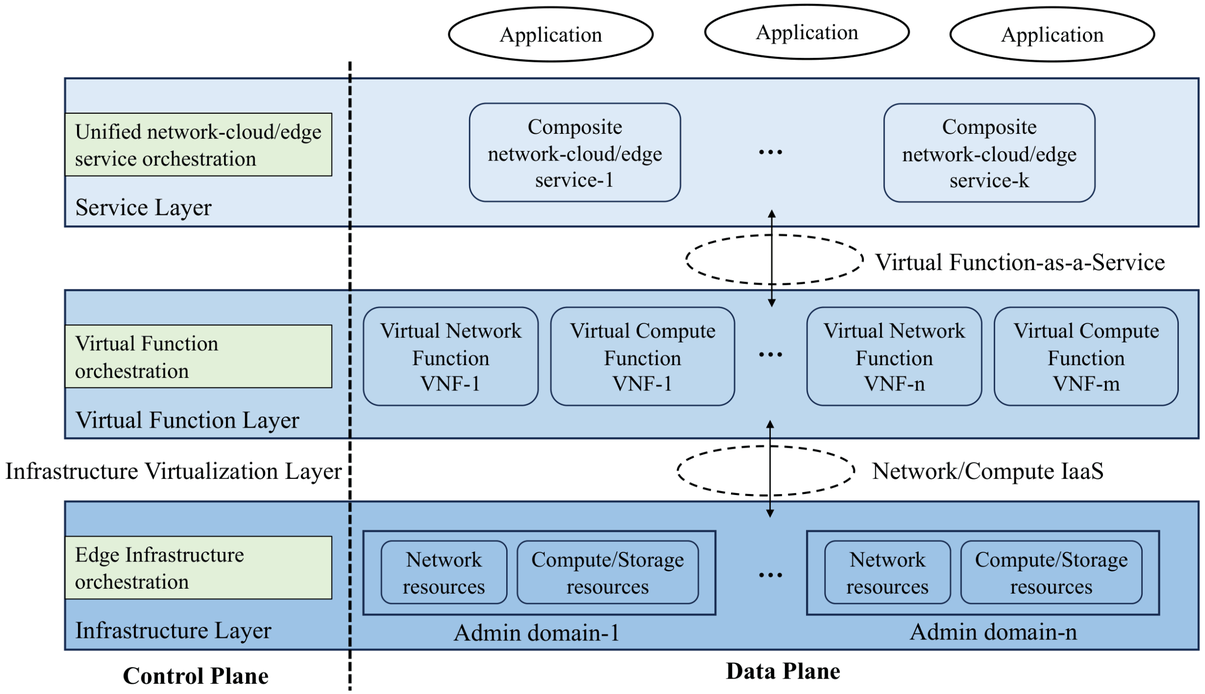
MEC apps and VNFs share the same virtual infrastructure; NFV MANO manages MEC.

3GPP/5G adds NFV + SDN + network slicing — multitenant virtual networks converging network & edge.

Networking: Edge Computing & Network Integration



Edge computing and networking



Architectural framework for edge-network integrations

Networking: Progress & Trends

Network cloudification

From dedicated appliances to commodity data-center-like systems running virtual network functions as software.

Network slicing

Multitenant virtual networks on shared infrastructure — a promising path to converge networking with cloud/edge.

Unified orchestration

Managing resources & functions across both network and edge, provisioning services that span both domains.

SDN + softwarization

Programmable, service-based architectures making the network flexible enough to co-host computing.

Summary

- Built for low latency, bandwidth efficiency, reliability & privacy — across three layers (data center, network, on-premises) with data process, cache and storage, networking & CDNs. **Bidirectional computation flow** between edge and cloud.
- MEC vs cloudlet; platforms like KubeEdge, K3s & Greengrass; and collaboration models (edge-edge, edge-device, edge-cloud, cloud-edge-device) that deliver continuity & unity.
- Networking is the backbone. Through NFV, MEC integration, 5G & network slicing, networks and edge computing are converging into unified, programmable platforms.

Practice

1. Discuss the different grades/layers of edge infrastructure. How do these layers contribute to the overall efficiency and scalability of an edge computing system?

2. Compare and contrast mobile edge computing (MEC) and multi-access edge computing (MEC).

3. What are collaborative edge computing models, and in what scenarios are they most beneficial?

4. What factors should be considered when choosing the right edge computing model for a specific application? Provide examples to support your explanation.

Project

Build a demo system for cloud-edge-device collaboration, with the functions of device side data collection, edge-side preprocessing data, and cloud batch data processing.

Device (Data collection)

Edge (Preprocessing)

Cloud (Batch processing)

The data flows one way: device collects → edge preprocesses → cloud batch-processes. Each layer does the job it is best suited for.

Project

Explore the impact of latency on application performance. Set up a cloud-based application and measure its performance, then replicate the application in an edge computing environment and compare the results. Utilize tools like https://prometheus.io/ for monitoring and https://grafana.com/ for visualization to measure and compare latency.

➢ Task for completing the project

- Device: script a sensor (or emulator) that emits raw readings
- Wire them together — device → edge → cloud messaging
- Edge: a Pi / VM cleans, filters & aggregates the stream
- Show it working end-to-end with sample data
- Cloud: batch-process the aggregated data (e.g. daily stats, model)
- Explain the result

References

- Zhao, Y., Zhang, W., Zhou, L., & Cao, W. (2021). A survey on caching in mobile edge computing. Wireless Communications and Mobile Computing, 2021(1), 5565648.
- Satyanarayanan, M. (2013, June). Cloudlets: at the leading edge of cloud-mobile convergence. In Proceedings of the 9th international ACM Sigsoft conference on Quality of software architectures (pp. 1-2).
- Mohammad Babar et al. “Cloudlet computing: Recent advances, taxonomy, and challenges”. In: IEEE Access 9 (2021), pp. 29609–29622.
- Qiang Duan and Shangguang Wang. “Network cloudification enabling network-cloud/fog service unification: State of the art and challenges”. In: 2019 IEEE World Congress on Services (SERVICES). Vol. 2642. IEEE. 2019, pp. 153–159.