OPENSTACK

Next Generation Edge: Edge Computing Architectures for Artificial Intelligence and Machine Learning Use Cases

Contents

1. Key Takeaways
2. Introduction
3. Purpose
4. Audience
5. OpenInfra Edge Computing Group
6. Definitions and Assumptions
7. Edge Computing Definition
8. AI/ML Considerations
9. Challenges
10. Data Considerations
11. Environmental Considerations
12. Edge Infrastructure
13. Hardware
14. Software
15. Section 15
16. Section 16
17. Networks
18. Lifecycle Management Tools
19. Section 19
20. Section 20
21. Section 21
22. Security and Privacy
23. Open Source Projects
24. Architectures
25. Edge Infra with Centralized Cloud
26. Section 26
27. Fully Distributed Infrastructure
28. Section 28
29. Use Cases
30. IoT Use Cases
31. Healthcare
32. Section 32
33. Section 33
34. Section 34

35. Consumer Scenarios
36. Section 36
37. Section 37
38. Intrusion Detection Systems
39. Supply Chain
40. Edge Robotics
41. Federated Learning (FL)
42. Section 42
43. Ultralight Autonomous Drones
44. Wearable Cognitive Assistance
45. Conclusions
46. Challenges and Risks
47. Future Directions
48. References
49. Authors

Key Takeaways

While it is still early in the Edge-AI revolution, the next twelve to eighteen months look to be an exciting time with new discoveries and understanding of how to best use AI to optimize edge use cases. Since the intersection of Edge Computing and AI is such a new area of research and development, taking a cross industry collaborative approach to identify needs and address requirements is crucial. Growing interest in AI and Edge is driving active investment in new and existing projects and working groups alike in the Open Source ecosystem as a way to ensure long-term success.

- To stay current it is recommended to follow relevant Open Source activities closely
- Consider joining or forming communities and projects to build that crucial shared knowledge base and Edge AI foundation

The ability for Edge Computing and Artificial Intelligence (AI) to be integrated is essential for many emerging and existing use cases. This requires a solid understanding of best practices and guidance because unlike other AI applications, Edge AI use cases have significant constraints that need to be taken into consideration for a successful project deployment.

- Pay close attention to the data and security requirements including the resources needed to support the use case
- Match the right architecture to the use case for optimal results
- Consider keeping the AI processing and data as close to where the data is generated to minimize latency, maximize data sovereignty and realize network efficiencies

The potential is there as infrastructure costs fall and as organizations become more comfortable with the concepts to build on the research and deployments that have already produced results. The use cases covered in this paper further demonstrate the great promise that Edge Computing applications combined with AI have for the future.

Introduction and Overview

Purpose

Artificial Intelligence and machine learning are hot topics, subjects of much recent investment, research and development efforts. The field is now so large and varied, that it would be impossible to cover it all in one paper. This document will focus on exploring and sharing information about cutting edge research related specifically to use cases where AI and Edge Computing intersect.

There are many possibilities and opportunities for using AI for Edge Computing use cases. However, they need to take into account an understanding of the specific constraints of Edge Computing, so the designs and implementations can successfully work in concert. This is an area of research and development that has been somewhat overlooked by the general AI community, so the OpenInfra Edge Computing Working Group, which has members with expertise with both AI and Edge Computing, is addressing the subject.

While there are several active Open Source projects and communities focused on both Edge Computing and AI, until now, there has been little activity that combines the two together. The intersection between these two areas should be considered a new field, where Open Source activities and solutions are forming. This is an opportunity to create a knowledge base of new challenges and requirements, for Open Source communities to start addressing this new problem space.

Audience

The audience is anyone who is interested in research, development or practical applications for incorporating Artificial Intelligence tools and technologies. Some of the groups include:

- **Open Source and Standards Communities**- The AI, Edge, networking and developer communities all need to work together to realize the full potential for building the next generation of systems. The Open Source and Standard communities can play a leading role in setting the reference architectures that are needed.
- **AI/ML Developers and Data Scientists**– Artificial Intelligence (AI) can be applied across a wide range of domains, with particularly compelling use cases emerging at the edge, where data is frequently generated and used in real-time. Does it make sense to always assume that all the data needs to be or can be transported to the core cloud for processing and management? Are there ways to optimize the processing and data manipulation to take advantage of the proximity to the edge?
- **Infrastructure Architects**– The edge has been somewhat overlooked by infrastructure architects, who have focused more on cloud and more centralized infrastructure. However, edge requirements have some unique characteristics that make them more challenging to incorporate AI applications. What are some of the changes that need to take place in order to make edge applications more efficient?
- **Telecommunications/Networking Community**– The telecommunications community, which by necessity is part of any edge architecture, has been exploring how implementing AI at the edge affects network traffic utilization and resources. What are some ways that network traffic can be optimized? Can AI at the edge help with the optimization process?
- **Academia and Researchers**– AI in general and AI at the edge are both areas that are in need of more research and development. Some areas to explore are how effective is AI in some research areas? Are there any new areas where AI can be used?
- **Governmental Policy Makers**– These groups support Edge and AI technologies by establishing progressive regulatory and funding frameworks that incentivize research and innovation.Their efforts aim to inform evidence-based policies on data governance, AI ethics, cybersecurity, and infrastructure investment, including recommendations for regulatory alignment for cross-border edge deployments. They also provide guidance on fostering sovereign AI capabilities to ensure legal and institutional frameworks evolve in step with technological advancements.

OpenInfra Edge Computing (Working) Group

The OpenInfra Edge Computing Group was formed in 2017 coming out of an increased interest in understanding the challenges and opportunities of use cases that require more computational power closer to the end users or devices. The group has already published several White Paperson topics of interest to the Edge Computing community.

The working group initially set out to explore the newly emerging landscape to better understand the demands and requirements that these applications put on the network and overall end to end infrastructure. By collecting relevant use cases, the working group members started to analyze the types of architecture models and to define infrastructure systems needed to support applications that are distributed over a broad geographic area, with potentially thousands of sites, and located as close as possible to discrete data sources, physical elements or end users.

The OpenInfra Edge Computing Group's main scope remains the software infrastructure stack. However, at the edge hardware features and network considerations play a larger role in defining and supporting the use cases, viable architecture options and tests for evaluating new and existing solutions. The OpenInfra Edge Computing Working Group addresses the needs for these solutions across different industries and global constituencies, to enable development activities for Open Infrastructure and other Open Source community projects.

Definitions and Assumptions

Edge Computing Definition

While Edge Computing has been a "hot" research topic for the past decade, many attempts to define exactly what Edge Computing is has been hard to pin down. Possibly the issue is the changing nature of the edge or just that the edge covers a broad area of IT infrastructure. Whatever the reason, this whitepaper builds on concepts covered in previous OpenInfra Edge Computing Group published whitepapers and uses the *Cloud Edge Computing*definition to serve as the definition that best fits for these purposes.

As a reminder, *Cloud Edge Computing*allows application developers and service providers cloud computing capabilities, as well as an IT service environment over a distributed WAN network. By extending the capabilities of traditional data centers out to the edge, increased computational power is available for current and emerging use cases. Note that there is some bias in this definition as it makes the assumption that some kind of core cloud is needed. In reality, it is quite possible, and in many use cases desirable, to minimize or eliminate the core cloud elements completely, allowing edge architectures to be a set of loosely or tightly connected remote nodes.

In the context of Artificial Intelligence, Edge Computing concepts and methods become even more important due to the resource-intensive demands of data processing and model training. These can benefit from being performed in proximity to the data sources in order to reduce latency and enhance efficiency.This approach ensures more efficient bandwidth consumption along with increased data sovereignty and security, as raw data is processed locally, and might not need to be transmitted through the network to remote central locations for processing.

Artificial Intelligence and Machine Learning (AI/ML) Considerations

The definitions of AI/ML have evolved significantly over the years as technologies and requirements have changed. From the early unfulfilled promise of neural networks to the spectacular success of using Large Language Models for creating tech support chatbots and solving some of the most difficult business challenges today, AI/ML as an area of opportunity and research is still very much in flex.

However, there are some core principles that have remained the same. For the purposes of this paper, it is limited to a definition that best matches Edge use cases and requirements. This means it is not strictly a telecom and networking centric model, but with the understanding that telco network support is needed to support AI and the edge. Thus the emphasis is on a more general definition that supports edge use cases, where AI will use data either generated at the edge or used at the edge.

Data and Environmental Challenges

The entire Artificial Intelligence revolution has been driven by access to vast amounts of data with volumes that surpass all previous levels. The Internet, the search engines, social media combined with the digitalization of millions of documents have created enormous data lakes, where for the first time in history we have the processing power combined with the data needed to be able to extract possibly useful and actionable information locally.

Data Considerations

What is most important in this context is that the data used for AI processing is often generated at the edge or over widely distributed systems and devices. This is really important to keep this in mind when developing architectures that support AI applications. There are a number of factors related to the data that need to be considered when designing an appropriate edge architecture.

- **Data Structure**- Data that is used for AI/ML purposes (for that matter, any purposes) comes in a number of different forms. For simplicity, this paper considers three main data categories:
    - **Structured data**– Examples include databases and network traffic. Each data point is carefully mapped following standards or documented definitions.
    - **Unstructured data**- Examples include images, sensor outputs and speech.
    - **Semi-structured data**- Examples include text and documents.

- **Data Location**- This concern addresses how the data is handled by the AI and where it is generated, where it is stored and where it is processed. Data collection and sharing should prioritize keeping costs low and data secure, increasing reusability of data across use cases. This is where it is critical to understand the parameters of the application and desired outcomes in choosing edge architectures that will support AI frameworks.
    - **Remotely generated data** – Examples include IoT, Telematics and fleet management, oil and gas well data and any other sensor data that is generated in the field.

    - **Centrally generated data**- Examples include centralized databases and data lakes, such as document repositories and customer databases.

- **Data Volume**-How much data is generated at each node? Does the data need to be stored locally and on what type of media? Each use case is unique on how much data is generated, but the overall volume and how much has to be stored, how long it has to persist and where it is stored are of concern, where network connectivity and resources can put severe limits on storage capacity and mobility. Use cases might include a large number of nodes that each generate relatively small volumes of data, but in aggregate can be significant, or alternatively, a few sites might create large volumes of data (video camera streaming is a good example of this type of node). Some of the questions to consider include:
    - Think about the balance between the cost of retaining the data locally for processing, versus the cost of transmitting it to a less costly centralized location. The decision will depend on the availability and cost of the connectivity and the amount of data that needs to be moved. How do you find the sweet spot?
    - Does the remote node support storage capabilities at all?

- **Data Lifecycle**– How long is the data needed and how long does it need to be retained? AI processing can affect these policies as it needs to use historical data that might not be active or retained for any other purpose. This can become an issue at the edge, where data storage often comes at a premium, or might not even be possible.

Environmental Considerations

Edge infrastructure is distinct in several critical ways from cloud or traditional data center systems with centralized infrastructure. Edge Computing assumes the infrastructure is installed in areas with limited administrator access, physical security and environmental controls. While AI use cases are expected to demand more advanced and expensive equipment, logistical and price constraints are still a major consideration for Edge operators. As AI applications become increasingly data-intensive and time-sensitive, traditional cloud-based AI architectures are struggling to meet modern demands. These systems rely heavily on centralized data processing, which introduces several critical challenges. Some of the constraints under consideration are as follows:

- **Compute Infrastructure Resources**- This is by far and away the most important consideration. Resources include not only computational / storage / scalability limitations, but also power, space and potentially the need for ruggedized equipment. Unlike other Edge applications, AI platforms are expected to be multi-system. It is important to design governance and control into the infrastructure, so operators must be prepared to manage clusters of systems. This adds a requirement for edge-local orchestration, automation and governance.
- **Local Access and Management**- A defining characteristic of edge is that there are very limited human administrative resources available to quickly address any problems or to perform maintenance. In some cases – such as satellites or deep ocean locations where access might even be physically impossible. Consequently, there is a high cost associated with maintenance and corrective actions.
- **Network Restrictions**- While the cost of WAN networking has come down in recent years, there are still restrictions related to latency and bandwidth cost or availability.

- **Privacy & Security**- Not only is there a need to physically protect the systems from public access due to their remote locations, but depending on the jurisdiction, there might be different privacy and data laws. Expensive components of AI systems need to be regularly audited to ensure that they have not been stolen or otherwise compromised. The network connections need to be properly secured.

- **Cost of infrastructure:**Considering the number of expected locations, edge infrastructure is often cost constrained compared to data center infrastructure, where they are less able to take advantage of economies of scale.
- **Hardware:**A critical challenge for Edge AI systems is related to the selection of suitable edge hardware and AI models that meet the specific needs of edge environments and the corresponding mapped applications. The modern trend of moving AI computation nearer to the origin of data sources has increased the demand for modern hardware solutions suitable for such environments.
- **Power Management:**Energy consumption includes the power required to transmit data and the communication cost induced by data transfers between application tasks. Inefficient design of edge resource management systems can significantly increase communication and energy consumption costs, which limits the lifetime of battery powered nodes and may result in power failure. Saving energy in data transfers between tasks remains an open problem which becomes even more critical for data intensive parallel AI applications, such as intelligent surveillance systems.
- **Storage:**Although many real-time processing platforms are available today, the amount of data generated is growing faster than the processing and storage capacity. To address future needs, real-time analytics functions need to be directly embedded at the level of processing units and automatically deployed, being scalable by design and use replicated distributed storage for certain services. To meet the stringent requirements of time-trigger IoT applications and to realize the deterministic networking in the Cloud-to-Edge continuum, it might be necessary to provide an IoT Gateway as a Service, which orchestrates different functionalities of the distributed gateway in the network, so as to implement replicated storage functions to provide stable storage for ad-hoc networks and effective bandwidth management, and distributed storage among nodes.
- **Rising Storage Costs:**Storing large volumes of raw data in the cloud—especially high-resolution video, sensor logs, and historical records—can become prohibitively expensive. This includes not only storage fees but also costs associated with data transfer, redundancy, and long-term retention.

- **Heterogeneous Computing Architectures:** An Edge Computing environment might include numerous heterogeneous devices, such as personal computers, mobile robots, smart cars, smartphones, and sensors. These devices differ in terms of system architecture, operating system, execution environment, and speed. A device may offer a single service or a multitude of services, such as data collection, task execution, data storage, and data caching. Such an environment introduces numerous challenges:
    - Uniform asset management and control of heterogeneous devices, including the efficient discovery, registration, and monitoring of a wide range of devices and services
    - Allocation of heterogeneous computing, sensing, and actuating resources to application tasks with a diverse quality of service requirements

    - Communication and collaboration of heterogeneous services regardless of application platforms, programming languages, operating systems, or system architecture
    - And finally the additional cost and overhead of managing a diverse environment

- **Heterogeneous and Multilayer Network Architectures:** A heterogeneous Edge Computing environment can have a mix of stationary and mobile devices. Devices equipped with multiple wireless communication technologies are common and will become more so with the spread of 5G and beyond into 6G networks. Wireless communication technologies have diverse features and differ in terms of bandwidth, latency, energy consumption, communication range, reliability, and network topology. Devices may communicate via infrastructure-based wireless LAN technologies, infrastructure-less wireless LAN technologies, or both. These characteristics enable the creation of a communication and network infrastructure with multiple and overlapping topologies, diverse source-to-destination links, and dynamic topologies and links due to the existence of mobile nodes. Given the characteristics, heterogeneous and multilayer communication and network infrastructures introduce several requirements:
    - Development of communication and energy consumption cost estimation models
    - Development of quality and link lifetime estimation models and their Integration for heterogeneous network communication technologies
    - Design of efficient discovery and monitoring protocol or set of protocols for a multilayer network infrastructure characterized by multiple and overlapping communication topologies, diverse source-to-destination links, and dynamic topologies and links
    - Development of routing and network management protocols capable of selecting routes and supporting data transmission services based on diverse quality of service requirements over a multilayer network infrastructure
    - Design of a network layer that provides a unified and easy-to-use interface to higher layers

- **Bridging the Gap: Cloud AI and Edge AI:** While cloud-based AI has been instrumental in scaling intelligent applications globally, its limitations in latency, cost, and privacy are becoming increasingly apparent in real-time, data-rich environments. Edge AI emerges not as a replacement, but as a complementary evolution—bringing intelligence closer to where data is generated. Together, cloud and edge architectures can form a hybrid AI ecosystem that balances centralized power with localized efficiency, unlocking new possibilities across industries. Importantly, the choice between cloud and edge AI often depends on the specific use case—whether it demands real-time responsiveness, data privacy, cost efficiency, or large-scale centralized processing.

Edge Infrastructure Components

The components needed to support AI use cases at the Edge include the following domains:

- Hardware
- Software
    - Operating systems and tools—this could include cloud native edge software and tools

    - Applications
- Networks
- Lifecycle management tools
- Security

Hardware

The choice of hardware plays a critical role in ensuring that the edge device can handle the computational demands of AI modelswhile adhering to strict resource constraints, such as limited processing power, memory, and energy. Key components include central processing units (CPUs), graphics processing units (GPUs), tensor processing units (TPUs), neural processing units (NPUs), and field-programmable gate arrays (FPGAs), each can be optimized for different AI workloads. Additionally, memory architectures such as random-access memory (RAM), video RAM (VRAM), high-bandwidth memory (HBM), and non-volatile storage solutions like solid-state drives (SSDs) and hard disk drives (HDDs) play a crucial role in managing data throughput and storage efficiency. Each of these hardware components have distinct advantages and limitations, impacting computational speed, energy efficiency, and scalability in AI applications.

Understanding the interplay between these hardware elements is essential for optimizing AI performance across diverse domains, including deep learning, data analytics, and real-time inference. For example, GPU vs CPU balance is important. In cases where AI/ML requires expensive GPU resources, it is also important to ensure that the adjacent CPU, network and storage components can sufficiently service the GPU capability. Selection of AI models is equally important for the overall performance of the system. AI models used in edge environments must be designed to be lightweight, fast, and resource-conscious, which often requires techniques like weight pruning and quantization to reduce model size and computational complexity while maintaining accuracy. Inefficient AI models can degrade the performance of the device, causing delays, excessive power consumption, or a reduction in the overall quality of AI-driven tasks. Thus, the AI models must be tailored to the specific edge device’s capabilities. Putting it all together properly means the hardware and AI models work in tandem to create a system that is not only capable of performing real-time, intelligent tasks but also operates reliably and efficiently within the constraints of edge environments.

Multiple research directions are currently exploring the efficient mapping of edge AI algorithms onto hardware platforms, aiming to establish best practices for performance, energy efficiency, and reliability. Notable hardware-centric AI solutions span a range of critical domains, including AI-driven biomedical systems, military applications, healthcare technologies, autonomous aerial and ground vehicles, security systems, and deep neural network (DNN)-based frameworks for applications, such as computer vision, data analytics, and robotics. Below are some links to related research that is mapping AI-based frameworks on edge nodes using specific hardware platforms.

Software

Infrastructure Services

As a multi-node system, AI clusters require services that are not present in single-node applications. For example, having network services such as NTP, DHCP, DNS, Secure Boot are critical for coordinating activities and creating valid certificates. Sites generally require at a minimum a VPN host and SSH gateway for secure management. Since the nodes might not always be connected to the WAN, edge control planes should be designed for independent operations, such as local monitoring, alerting and provisioning capabilities.

Edge sites often use the same container or VM platforms as cloud and enterprise systems. While these platforms may add some management overhead and operational costs, the added portability will likely make it easier to maintain applications across the whole ecosystem by reducing any specialization needed for edge sites. These requirements make Edge AI look closer to a "mini datacenter" than other edge installations, given the additional investment required for AI hardware and clusters. By embracing the additional management needs, architects can achieve additional control, resilience and security capabilities that may ultimately reduce cost and operational load.

Applications

Clearly the applications that run on the infrastructure stack are going to drive the use case. There are likely to be at least two sets – one for the base use case itself and the second set to support the AI. The specifics of the applications used at a high level will be covered in the individual use cases section, but details of what applications are part of the suite is outside of the scope of this document. Since the hardware resources are often limited at a given node, efficient code that takes advantage of containers, microcode and serverless platforms is considered best practice.

Networks

Given that Edge Computing is by its very nature distributed, the network is a critical component of any architecture. Unlike cloud-based architectures that typically use virtual networks within a data center, edge networks are always going to be wide area networks (WANs) of some form. They might be fully private LAN networks (campus wide for example) that would use a combination of wired (CATx) or wireless (802.x) ethernet, but more typically they will rely on network services from telecommunications service providers. These services might be anything from MPLS wireline services to satellite network connections and anything in between. Factors that need to be considered when designing the network includes:

- How much data traffic will be generated from each node and as an aggregate over the entire network? The amount of data can vary quite wildly depending on the use case. In some cases, it will likely need to be processed locally, if possible, to reduce the amount that needs to be transmitted over the network. Another possibility might be gateways where the data is further processed so that only a small amount of data would be sent to a central cloud location. An example of this might be video camera streams used for vehicular traffic prediction. Video streams can be network intensive, but if the data needed is only the number of vehicles passing through by time, the video can be

processed locally and only the small amount of data about the traffic patterns needs to be transmitted for further processing.

- Does the data need to be encrypted or otherwise secured using a Zero Trust connection or an SD WAN network? This includes how important it is to maintain a fully private network connection or can the use case support the use of secure sessions over a public connection. These decisions are going to rely on a combination of network availability, cost factors and the sensitivity of the data.
- What type of network connections will be part of the overall WAN network? This is probably the most consequential factor to consider as network connectivity availability is going to depend heavily on the locations of the distributed nodes and the amount of data that needs to be transmitted over them. For example, an oil and gas use case is likely to have many remote locations with limited access to high speed networks, while at the same time generating a substantial amount of data.

Lifecycle Management Tools

This section covers at a high level the suite of tools and processes needed to manage the systems once they have been deployed. DevOps Orchestration tools manage the systems to ensure they are correctly set up (Day 1) and kept up to date (Day 2). For edge, they should be highly automated, simplifying the management and reducing the potential for error. Additionally, IT governance is important to ensure that security and management practices are followed. These considerations drive the choice of tools and level of automation needed.

Orchestration, Automation and Provisioning

For any edge use case, and especially ones that incorporate AI, the orchestration and automation tools are critical to the project's success. Edge by its very nature requires a high degree of automation for it to be cost effective, and AI adds to that calculation by potentially adding significant overhead to the mix. The good news is that there are now a plethora of edge friendly orchestration tools and APIs to address the issue.

Orchestration means coordination of activities in the operational environment. Specifically, orchestration platforms use event driven workflows and state management to manage processes on the infrastructure towards a target state. This often requires the platform to work toward intermediate goals, often requiring feedback from the system while it is being updated; consequently, orchestration platforms typically have full-time access to the environment.

Automation describes specialized code, often known as DevOps scripts and playbooks, that transform the state of the infrastructure through a prescribed sequence of steps. While automation can be very sophisticated, it can be thought of as a code-like workflow designed to interact with the hardware and operating systems in the edge environment. Ideally, all automation used is version controlled, secured to prevent tampering, and immutable to ensure consistency between environments.

Finally, Provisioning complements Automation by provisioning the services and processes necessary to install operating systems in the environment. This is separate from Automation because it involves services that are external to the target environment. For highly automated edge environments, Provisioning and Automation will be tightly integrated to enable controls like immutable booting or image deployment.

Operators need to integrate automated processes to manage the edge hardware including onboard, initial setup, security compliance, updates, drift detection, audit and decommissioning. Orchestrating this automation is essential because edge sites often lack redundancy or excess capacity. These become even higher stakes for locations with limited physical access where a system recovery requires a truck roll.

Governance

Many developers and designers often overlook governance because it is seen as part of process and policy, not integral to the technical components and the design process. However, governance is critical to making sure that the project meets the requirements and KPIs that are set by the business. Without governance, it would be impossible to measure the value of the project or determine the return on investment. Project governance must be coupled with observability tools. These tools and processes are of course part of the management toolkit, so should be closely tied with the overall management approach.

Data processing

Data preprocessing is a fundamental component in AI development, particularly critical in Edge Computing environments where raw data must be efficiently transformed into machine learning-ready formats. Edge processing enables data handling at or near the source, reducing latency and bandwidth consumption while addressing privacy concerns.

Key preprocessing techniques such as feature selection, data normalization, and data augmentation need to be adapted to function efficiently on edge devices operating within strict computational and memory constraints. The preprocessing pipeline must dynamically adjust its operations based on available resources while maintaining essential functionality. This includes implementing intelligent data filtering and aggregation mechanisms to minimize costly data transfers.

A systematic approach, which combines local processing capabilities with adaptive resource management, ensures high-quality input data while balancing real-time processing requirements and resource limitations, is needed. The “Data Fabric”, an emerging architectural framework seamlessly integrates these distributed preprocessing workflows, providing consistent data access, governance, and transformation capabilities across the entire data lifecycle. This approach enables organizations to implement flexible preprocessing strategies that can adapt to varying computational resources, data volumes, and latency requirements while maintaining data quality and model performance across the entire edge-to-cloud continuum.

The main features / requirements that architects need to consider for implementing “Data Fabric” capabilities for enhanced Edge AI operations include:

- **Local Data Processing Capability:**Edge devices must be able to perform data preprocessing, feature extraction, and inference locally to minimize latency and reduce bandwidth consumption. This includes handling sensor data, image processing, or time-series analysis directly on the device.
- **Resource-Optimized Operations:**Data processing algorithms must be designed to operate within strict memory, storage, and computational constraints typical of edge devices. This includes efficient data compression, selective data sampling, and lightweight preprocessing techniques.
- **Adaptive Processing Pipeline:**The system should dynamically adjust its processing parameters based on available resources and current device conditions. This includes scaling down selected operations during high load, or battery constraints while maintaining essential functionality.
- **Minimal Data Transfer:**Processing pipelines should prioritize local data handling and only transmit essential, processed information to central systems. This requires intelligent data filtering and aggregation mechanisms at the edge.
- **Real-time Processing Support:**The edge system must support low-latency data processing for time-critical applications, ensuring that data can be processed and acted upon within required time constraints without relying on cloud connectivity.
- **Robust Error Handling:**Data processing must continue functioning effectively even under unstable network conditions or resource limitations, implementing appropriate fallback mechanisms and data buffering strategies.
- **Replicated Storages:**Replicated storage on the Gateway functionality can enable data storing at the edge location for low latency/high-performance networks (caching locally) that can be shared with other remote locations, and sized based on service requirements.

Security and Privacy

As with any IT system nowadays, security must be considered as part of the architecture and design from the start. With Edge Computing and AI there are the added complications of vulnerabilities introduced by the remote locations and the distributed nature of the nodes as well as the potential sensitivity of the data.

Machine Learning and Differential Privacy: If the data is sensitive (e.g., medical data), sharing it in plain form is undesirable. Homomorphic encryption enables the provider to process encrypted data, preserving confidentiality. However, risks like overfitting can still expose sensitive information. Differential privacy mitigates this by introducing small, controlled changes to the dataset, preventing the revelation of personal information. This concept extends to federated learning, a form of secure multi-party computation involving multiple service providers, where data and network training is decentralized.

Physical Security and Data Access: Given that edge architectures are often in exposed locations where physical access to the equipment is possible, any sensitive data must be encrypted both at rest and in transit.

Regulatory and Geolocation Considerations: Any use cases involving Personally Identifiable Information (PII) data that is subject to regulatory oversight must take that into consideration when choosing the appropriate architecture. For example, any use case that

involves gathering patient data (i.e. a remote heart monitoring application), would need to make sure that the data is not only protected from access by unauthorized entities, but it might need to be geographically limited by country specific regulations such as the US HIPAA regulations regarding patient records.

Relevant Open Source Projects

While this is far from a comprehensive list of the Open Source projects and activities that have some applicability to AI-supported Edge Computing use cases, it is clear that there are already many communities, working groups and projects that can be directly applied to building the solutions, both in the AI area and at the Edge. Of course, there are gaps, particularly in the intersection of the two technology areas, which opens opportunities to start new projects leveraging the frameworks that are already in place as foundational building blocks. The projects can be broken into different areas that map to the components covered above.

OpenStack– Under the auspices of the OpenInfra Foundation, OpenStack is a set of software components that provide common services for cloud infrastructure both at the core and the Edge. Its many tools and components are modularized so they can easily be used to support Edge and Edge AI applications.

StarlingX– StarlingX is a complete cloud infrastructure software stack designed for geographically distributed systems, including the Edge. It is used by the most demanding applications in industrial IOT, telecom, video delivery and other ultra-low latency use cases. With deterministic low latency required by edge applications, and tools that make distributed edge manageable, StarlingX provides a container-based infrastructure for edge implementations in scalable solutions ready for current and upcoming AI use cases.

Kata Containers– Kata Containers is an Open Source community project building a secure container runtime with lightweight virtual machines that feel and perform like containers. With the stronger workload isolation using hardware virtualization technology as a second layer of defense, Kata Containers delivers functionality required for a secure Edge.

LF AI & Data Foundation- An umbrella project under the auspices of the Linux Foundation, the LF AI & Data Foundation supports Open Source projects within the AI and data space. Some of the more notable projects include, the Generative AI Commonswhich has designed and developed the Model Openness Framework(MoF), a comprehensive system for evaluating and classifying the completeness and openness of machine learning models.

LF Edge- LF Edge is another umbrella organization that establishes an open, interoperable framework for Edge Computing independent of hardware, silicon, cloud, or operating system. LF Edge creates a common framework for hardware and software standards and best practices critical to sustaining current and future generations of IoT and edge devices.

EdgeLakeis a fully decentralized, self-managed, and cryptographically secured data platform where data is processed, managed, and accessed in-place. EdgeLake is a distributed, secure, and self-governing edge network that allows data to be processed where it's generated, governed by cryptography and blockchain, replicated for reliability, and exposed through standard interfaces. The backbone of the EdgeLake network is a cryptographically secured blockchain ledger that:

- Stores metadata and system policies,
- Manages access control and data ownership,
- Tracks provenance and node operations immutably.

Each node maintains a local copy of the ledger and verifies signatures before participating in any transaction or query. This ensures zero-trust security, verifiability, and traceability—essential for multi-tenant, multi-stakeholder environments such as energy trading platforms or inter-agency smart city deployments. EdgeLake ensures high availability (HA) by replicating data across multiple network nodes. Replication policies are automatically managed by the system based on application SLAs, node health, and network topology.

To promote transparency, innovation, and vendor neutrality, the open-source project is distributed under the Linux Foundation Edge initiative.

- It is fully aligned with LF Edge standards including Open Horizon and EVE.
- EdgeLake allows developers and organizations to build their own decentralized edge networks, tailored to their needs while staying compliant, cost-effective, and in control.

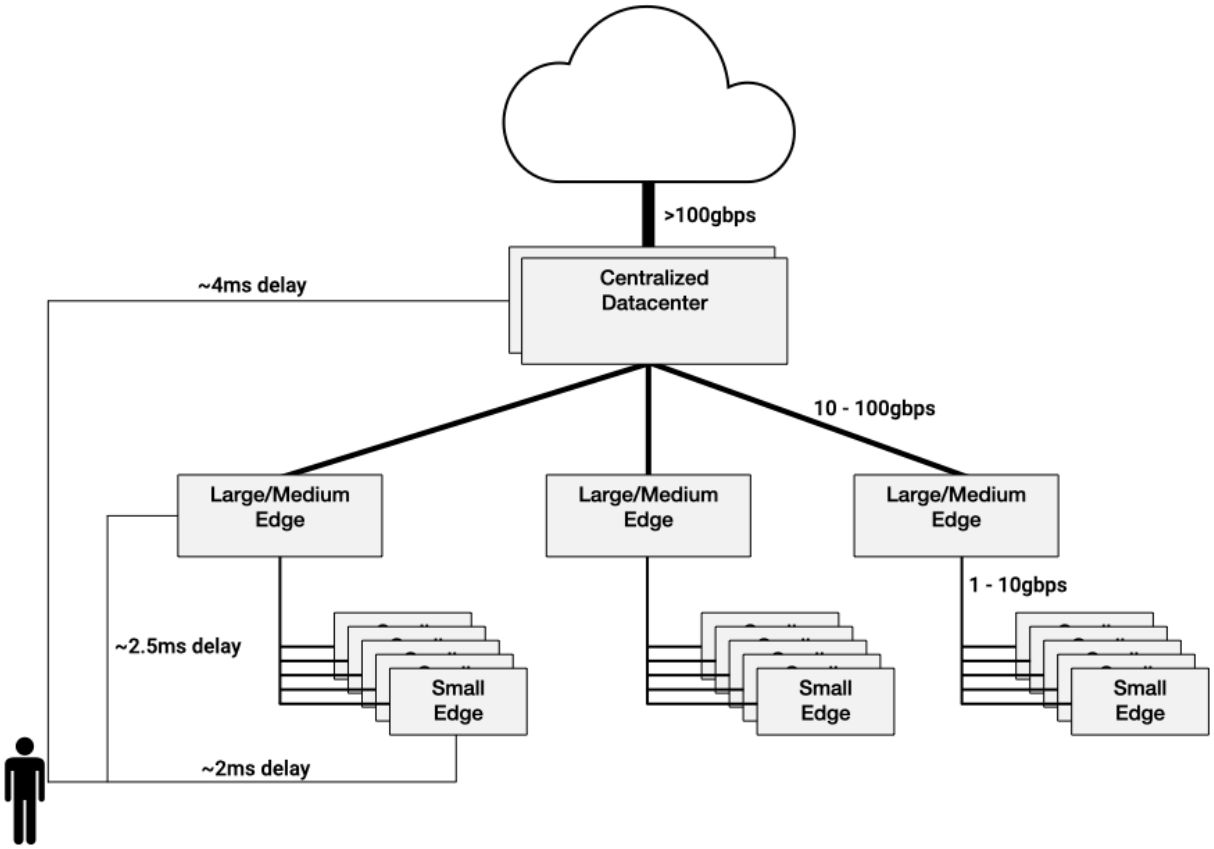
Architectures

At the high level, edge architectures fall into two basic models, with lots of variations. Within each model there are variations based on the size and mix of each of the nodes, network type, the number of nodes in the deployment and the nature of the use case. A general discussion of the challenges and risks of each approach can be found above.

- Edge infrastructure with a centralized cloud component
- Fully distributed infrastructure

Edge Infrastructure with Centralized Cloud

This architecture includes edge components as well as a centralized core.

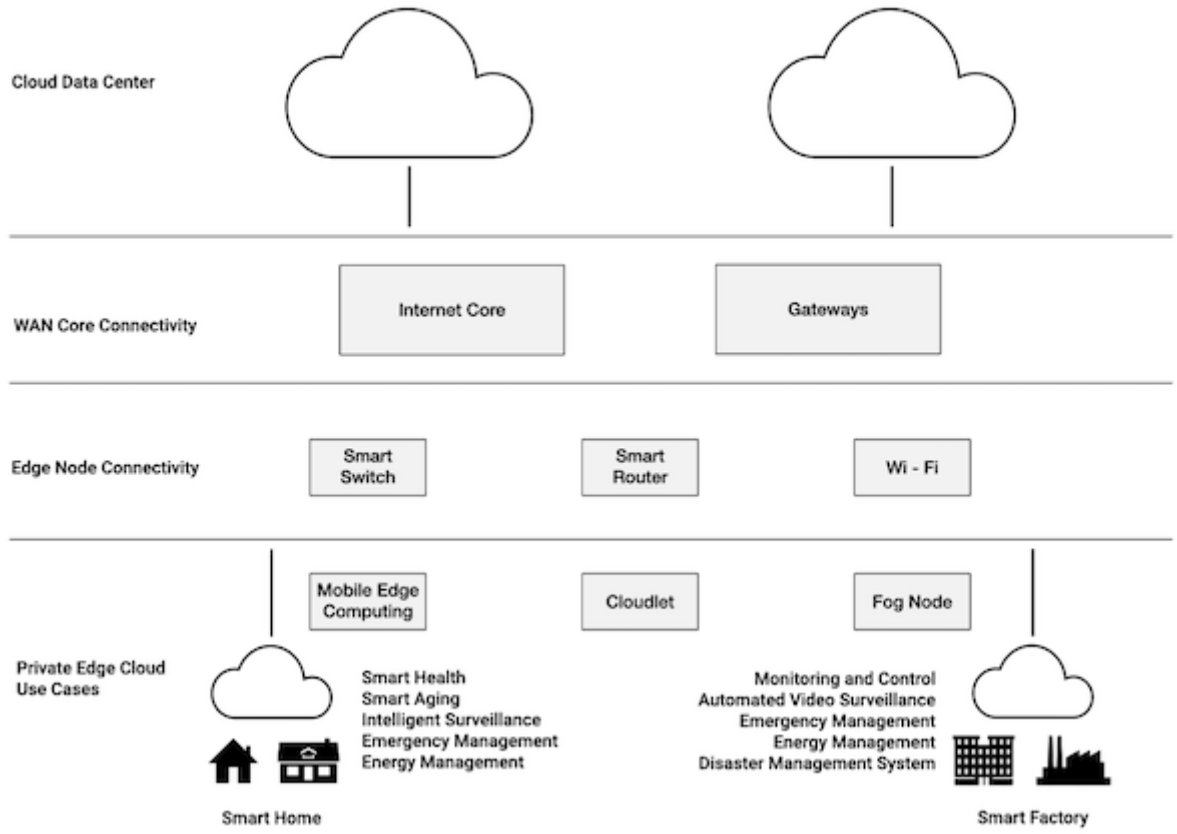


Edge Architecture with Centralized Cloud Component

Private Edge Cloud

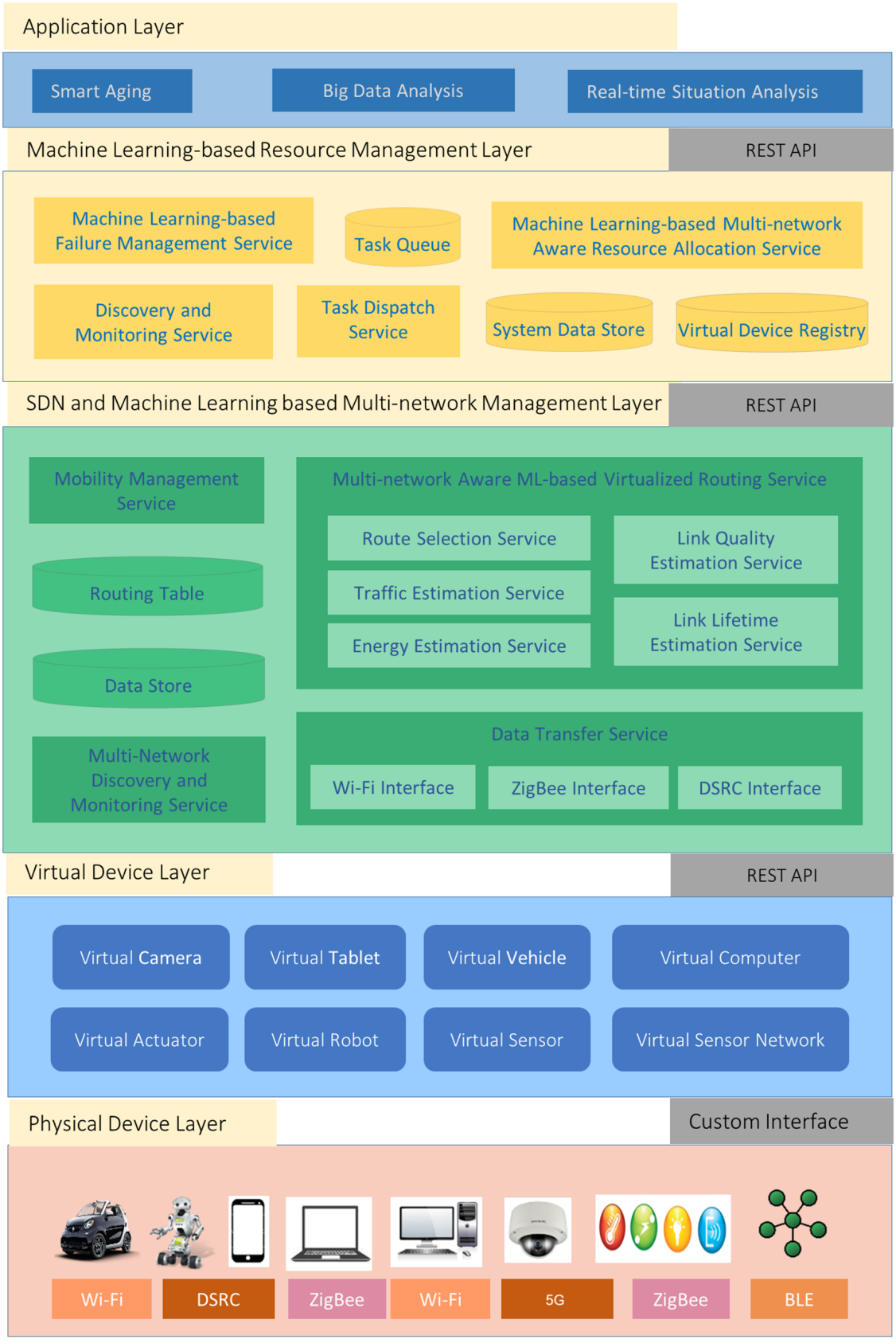
A private edge cloudis a small-scale cloud data center at a local physical area such as a home or an office. It consists of various stationary and mobile devices such as personal computers, mobile robots, smart phones and sensors, interconnected through a single or multiple infrastructure-based or infrastructure-less wireless local area networks.

The role of the private edge cloud in the ecosystem is depicted below. Various stationary and mobile devices such as personal computers, mobile robots, smart phones and sensors within a local area such as a home or factory are combined to create a small private cloud data center. This private cloud may provide computing, storage, data processing, data caching, and networking services. The private edge cloud can then be connected to a central virtualized cloud data center on the Internet.



Edge Cloud Network

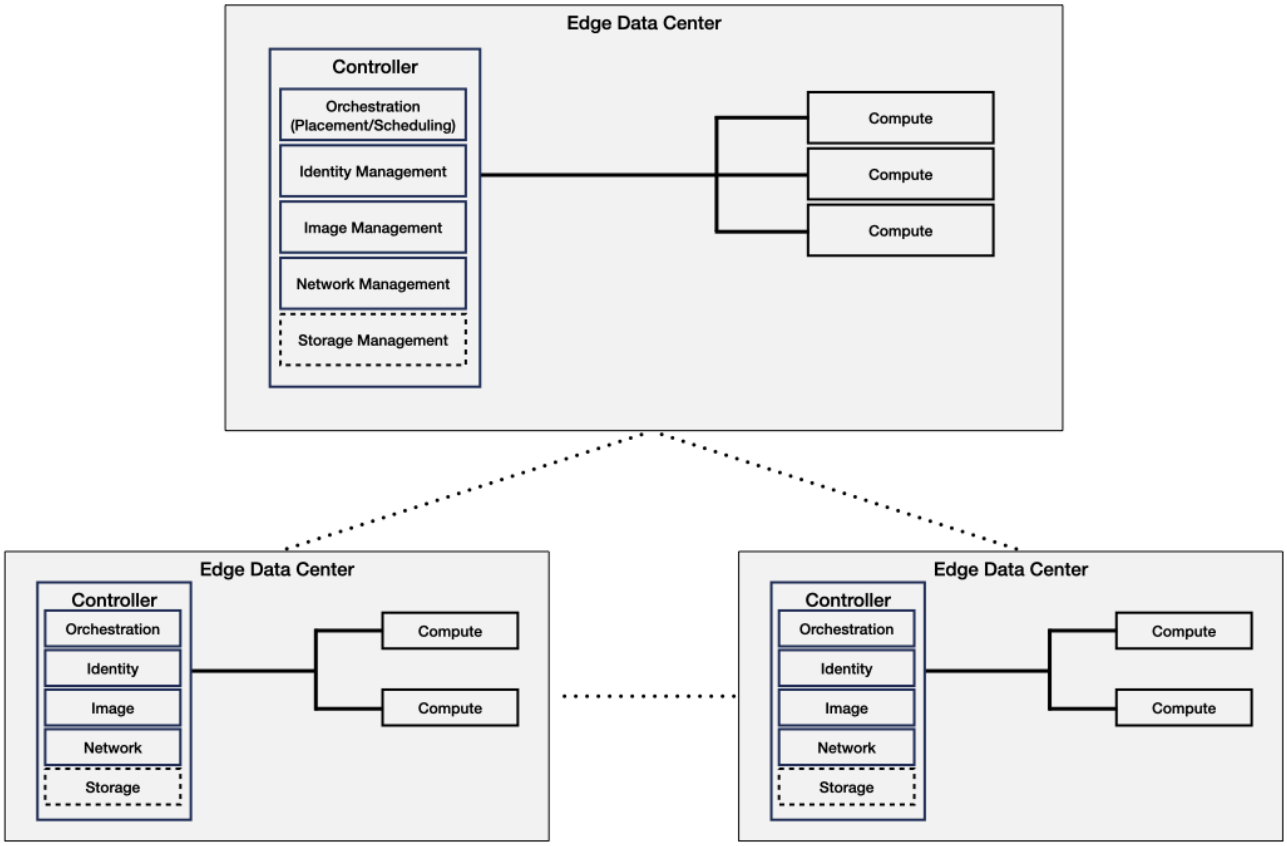
The diagram below goes into more details of the different components and which layer of the infrastructure stack they fit in.



Private Edge Cloud

Fully Distributed Infrastructure

This architecture includes only edge nodes and components that may or may not interact with each other. The following diagram outlines at a high level how a fully distributed architecture would look like.



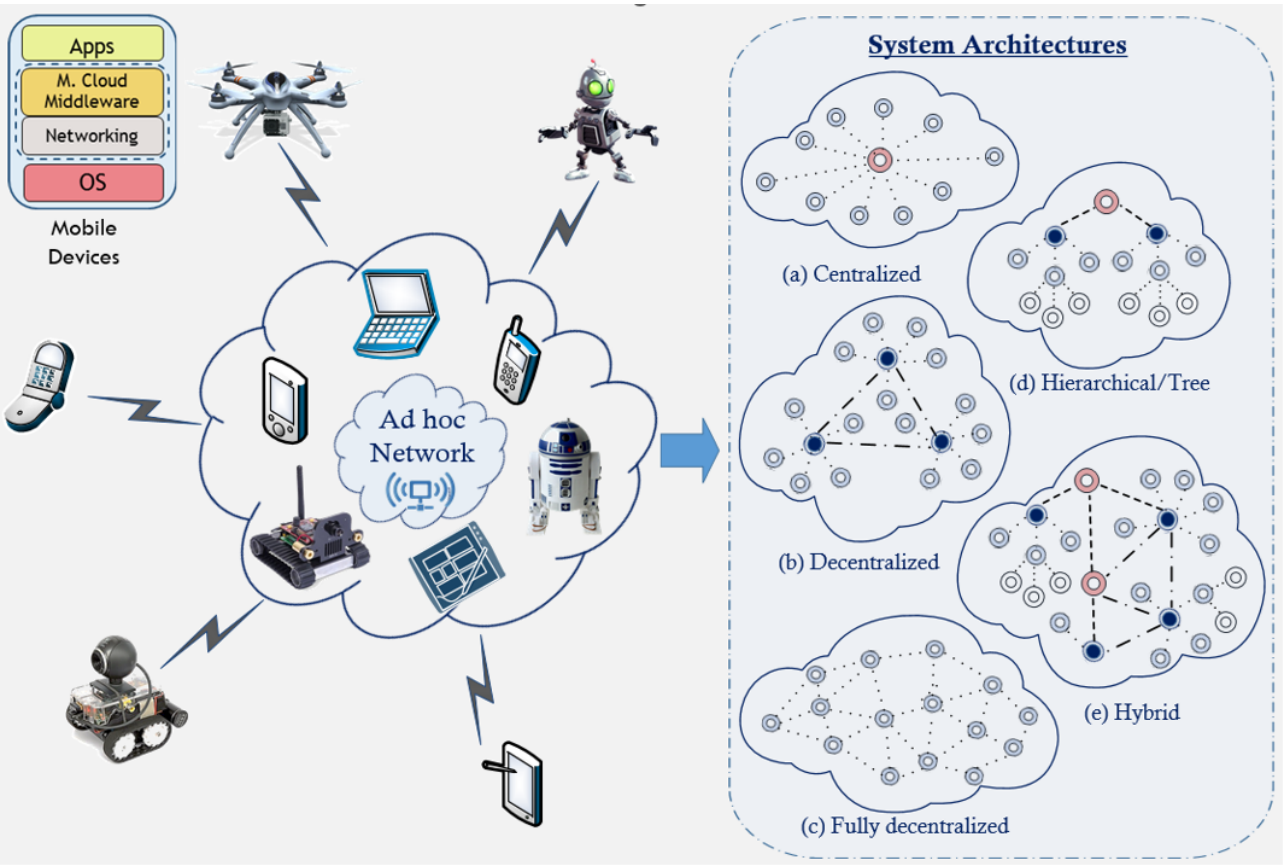
[Distributed Control Plane Scenario]

Mobile Ad Hoc Edge Cloud

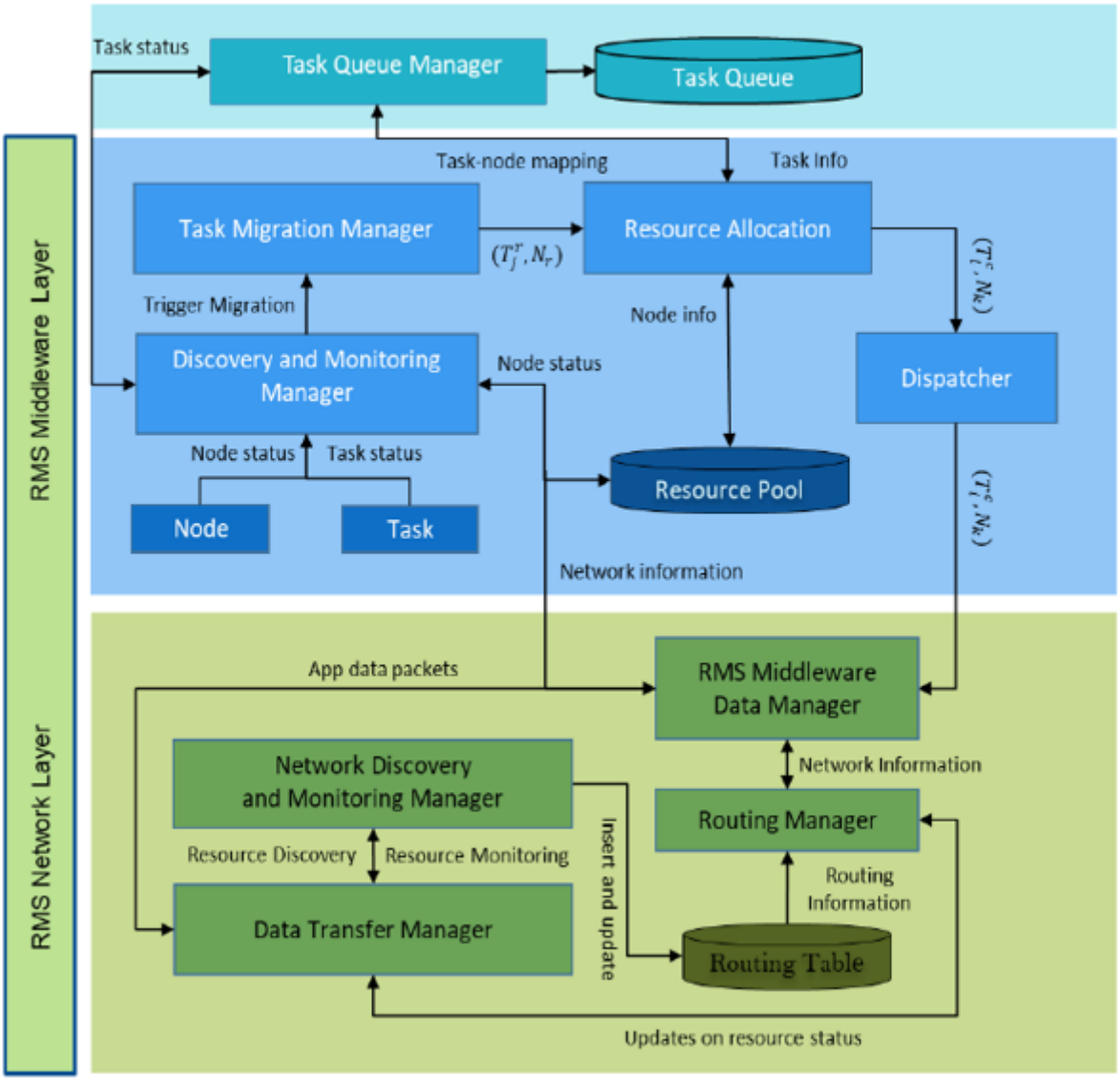
A mobile ad hoc edge cloud is a distributed computing infrastructure in which multiple mobile devices, interconnected through a mobile ad hoc network, are combined to create a virtual supercomputing node.

A block diagram and the architecture of a mobile ad hoc edge cloud are shown in the Figures below. Nodes in mobile ad hoc edge clouds are divided into three categories: service master nodes (SMNs), service provider nodes (SPNs), and service consumer nodes (SCNs). SMNs are responsible for allocating tasks submitted by the SCNs to the SPNs on the basis of the resource allocation policy. Nodes communicate with each other through a mobile ad hoc network, which provides several communication services, including routing and medium access. The resource management system is divided into two layers: a network layer and a

middleware layer. The network layer provides network and communication services and also collects network-level information, such as link quality and lifetime, which is used by the middleware layer during the resource allocation process.



Mobile Ad Hoc Edge Cloud



Mobile Ad Hoc Edge Cloud Architecture

Use Cases

The following section covers some typical and emerging use cases with recommendations for what types of architectures could best suit them. This is by no means a comprehensive set, but more of a sampling of use cases to help architects and developers working on developing edge implementations get started on their own design ideas.

Breaking down the high level architectures described above into their components and then mapping the typical requirements to the use cases can be a useful exercise. Some of the components can be quantified by ranges such as network latency and data rate. Despite there being a number of network connectivity options, ultimately they all come down to network traffic over a medium. Storage is also quantifiable as the total amount needed to support the application. The other components such as compute and security are actually made up of a

large number of smaller subcomponents, so, as such for the purposes of this document, they are defined as high, medium and low quantities. The following is a very high-level definition of what each of those terms mean.

Compute- This is defined as the underlying hardware and software to support the applications. Typically this includes CPU, memory and some form of operating system or container. It can include but not be limited to virtual machines, containers, microservices running on combinations of white, grey and black box hardware. Tradeoffs to consider include understanding that the more resources are installed at an edge node, the higher the total cost of ownership over time will be.

- **High**- Use cases and applications that need large amounts of processing, typically near-real time AI applications, high-volume, high-speed processing often associated with high volumes of volatile data. Examples include fleet management and local video processing applications.
- **Medium**– This applies to use cases that have some need for local processing, but there might be less data to process, or the application does not require near real-time results. The data change rate might be able to support less compute power to provide satisfactory results. Examples include oil and gas and some healthcare applications.
- **Low**– The lowest level of need for compute resources would be for use cases and applications where there might be a low volume of data, or the data is relatively static. Examples include facilities management and remote equipment maintenance applications or consumer smart home technology.

Storage– The amount of data storage required depends on both the volume generated at each device and how long it needs to be retained on site. The formula to determine the amount of storage needed is first finding the amount of data generated per day per device, then multiplying it by the number of days the data needs to be retained. This will give the base number for the raw data storage needed. Once that is determined, the amount of storage needed by the AI application itself also needs to be calculated.

The storage could be a combination of hard drives (less desirable, but cheaper) and SSDs. For extremely remote sites with a retention requirement, there can be a need for storage redundancy to reduce the risk of data loss.

- **High**– Examples of use cases with high data storage requirements include a high-resolution video camera application. It can generate as much as 1.5/TB per day of data and the application needs to retain the data for six months, which would require a minimum of 185/TB of storage per camera. This could quickly add up to significant costs.
- **Medium**– Covers a wide range of applications such as smart hospitals and patient digital twins.
- **Low**– Think of utility meters as an example, that only generate a small amount of data that does not need to be retained once it has been processed.

Security- Like compute, security covers many components at all layers of the infrastructure stack, ranging from physical security (sometimes problematic for edge locations), network security covering VPNs and Zero trust systems), to application security and finally most important data and application integrity and security. The CIA (Confidentiality, Integrity, and Availability) triad is applicable here, that is, who has the authorization to access the data, how is the authentication to access the systems done and finally, are the systems properly

protected so that it can be assured that the data and systems have not been compromised. Of course, the tradeoff is that the more security and encryption are added to the systems, the more compute and network resources are needed to support the use case. It would be wise to consider just how much security is needed for each use case under consideration. For example, a wildlife camera project would likely need little security, while a healthcare monitoring system would need enough security to be able to meet regulatory requirements.

- **High**– For some use cases, the entire suite of security systems will be required to support the use case. For example, a military monitoring system or another use case that has sensitive data or has geopolitical ramifications if the system was compromised.
- **Medium**– The amount and type of security for these use cases is going to be heavily dependent on the nature of that application, the sensitivity of the data and the risk that the data might be compromised.
- **Low**– An example of this is an edge application with no sensitive data (HVAC maintenance system) and little likelihood that it will be attacked by bad actors. Be aware, however, that even if you might consider the application to be at low risk, there is the potential for bad actors to access and use the system to jump into other networks, so there does need to be a minimum of security in place no matter what the use case is.

Use case	Network Latency	Network Data rate	Compute	Storage	Security
Factory Automation	0.25-10 ms	1 Mbps	Varies	Varies	High
IoT Sensor Analysis	50-100 ms	1 Mbps	Depends on data volume and complexity	Depends on number of nodes and data volume	Depends on sensitivity of the data
Process Automation	50-100 ms	0.1-100 Mbps	High	High	Medium
Intelligent Transport Systems and Supply Chain	10-100 ms	10-700 Mbps	High	High	Medium
Robotics and Telepresence	1 ms	100 Mbps	High	Medium	Medium
Virtual and Augmented Reality	1 ms	1 Gbps	High	Medium	Medium
Health Care	1-10 ms	100 Mbps	Varies	Medium /High (imaging)	High
Immersive Gaming	1 ms	1 Gbps	High	High	Medium
Smart Grid	1-20 ms	10-1500 Kbps	High	High	High
Education and Culture	5-10 ms	1 Gbps	Varies	Varies	Medium

Resource Requirements for Sample Use Cases

IoT - Real-time 3D Map Construction and Target Identification

The success of military or disaster relief operations depends on several critical factors, including a thorough understanding of the physical environment and the real-time detection and tracking of both stationary and mobile targets. To achieve situational awareness, mobile

robots and micro-drones equipped with audio, video, and environmental sensors are deployed to gather data. This data is then processed to construct a real-time, three-dimensional map of the environment.

The collected data is also used to detect and track both static and dynamic targets. These tasks rely on advanced image and video processing algorithms and AI models for analysis and decision-making. Given the complexity of these tasks, substantial computing power and storage capacity are required. To meet these requirements, a mobile ad hoc edge cloud is established, consisting of mobile robots, micro-drones, and wearable devices used by personnel involved in the operation, all interconnected through ad hoc networks.

The architecture consists of three main categories of devices, each serving a specific role in the system. **Sensing devices**, such as video sensors, are responsible for collecting data about objects and the environment. **Computing devices**, such as laptops, provide the necessary processing and storage services. Finally, **actuators**, such as autonomous tanks, perform physical actions based on the system's output.

These devices interconnected through ad hoc networking technologies are combined to create an ad hoc edge cloud. This cloud is utilized to execute computationally intensive tasks, such as 3D map construction and target tracking. See diagram in the Architecture section above.

Healthcare

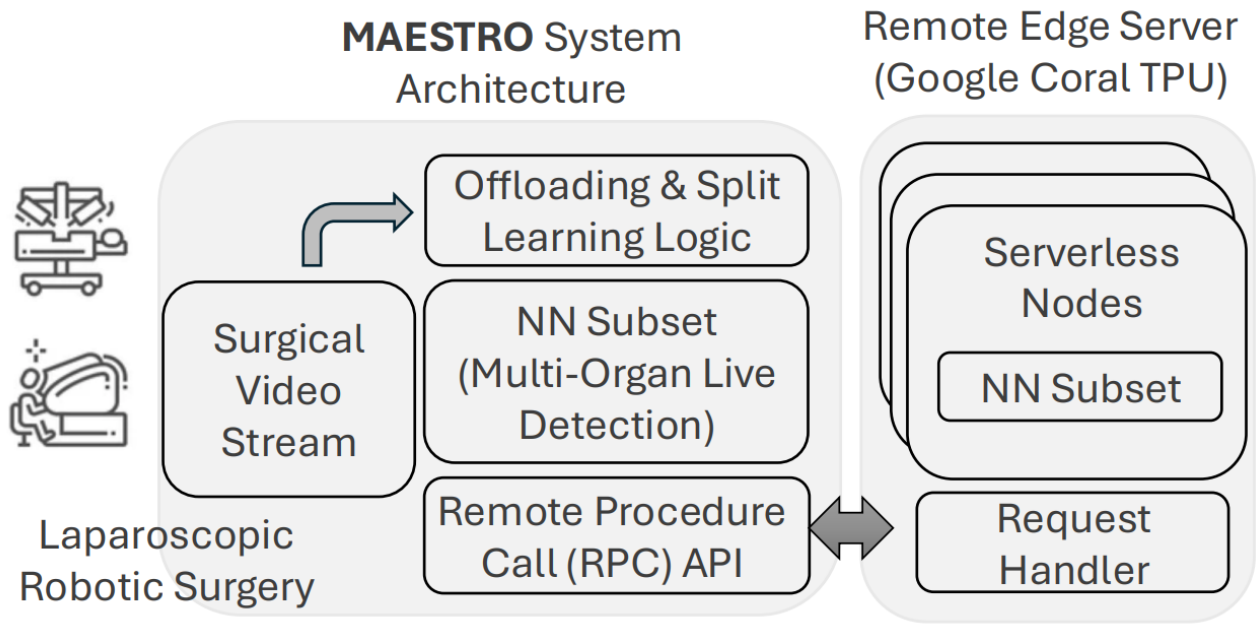
Healthcare is an area that has many interesting use cases where AI and Edge can be successfully integrated. Many of the use cases do require significant AI processing power and bandwidth, so balancing where the processing is done is critical for optimum performance and security of sensitive patient data.

Surgical Robotics

Surgical robotics is using a split learning approach that enables deep neural networks to be deployed across resource constrained edge devices combined with more powerful backends without ever exposing raw patient data. This concept has been demonstrated on a commercial robotic laparoscope by running the initial layers of a multi-organ segmentation network on the surgical console’s embedded accelerator and offloading only intermediate feature tensors to a serverless edge cluster.

The MAESTRO system sustains sub-100 ms segmentation updates—critical for surgery ergonomics and instant feedback. With an adaptive offloading algorithm, it also continuously profiles latency and on-device energy use, cutting on-device energy by over 50 percent compared to executing the full model locally.

Beyond its surgical application, this split-learning pattern is broadly applicable to any time-sensitive, data-sensitive imaging task, whether in endoscopy, ultrasound, or even nonmedical domains like industrial visual inspection, that can benefit from partitioning networks to support both strict latency requirements and privacy regulations such as HIPAA or GDPR. By combining lightweight on-device heads with elastic, serverless tails, practitioners gain a general-purpose blueprint for marrying state-of-the-art AI with the realities of Edge Computing.



System architecture overview

Pathology

The shift from monolithic cloud deployments to edge-proximate computing allows for real-time, collaborative digital pathology. In conventional telepathology, high-resolution virtual slides are either streamed from distant data centers, incurring multi-second latencies, or require costly proprietary appliances. Edge Computing reimagines this workflow by placing lightweight analysis and streaming services adjacent to the microscope, dramatically reducing latency while preserving throughput and privacy.

At its core, an edge-enabled telepathology systems have three layers:

1. **A local microscope integration software**, where a plugin or a service (e.g. µManager) captures live fields of view and exposes them via a minimal web interface.
2. **An edge processor**, in which containers or serverless functions run on a geographically co-located cluster (e.g., OpenStackat the network edge), handling both real-time video relay (ffmpeg + WebRTC) and on-demand analysis tasks such as stain normalization, nuclei detection, or tumor-margin measurement.
3. **A remote client**such as a browser-based viewer (e.g., OpenSeadragon) that receives and consume frames and interactive controls, enabling pathologists or other medical personnel to pan, zoom, and annotate with sub-second responsiveness.

The following areas are where Edge Computing research can drive forward telemedicine in general, and telepathology in particular. Together, they form a roadmap for developing modular, intelligent, secure, and network-aware systems that deliver practical, low-latency healthcare workflows at the network edge:

- Microservice Specialization
- Orchestration Intelligence
- Privacy-Aware Pipelines

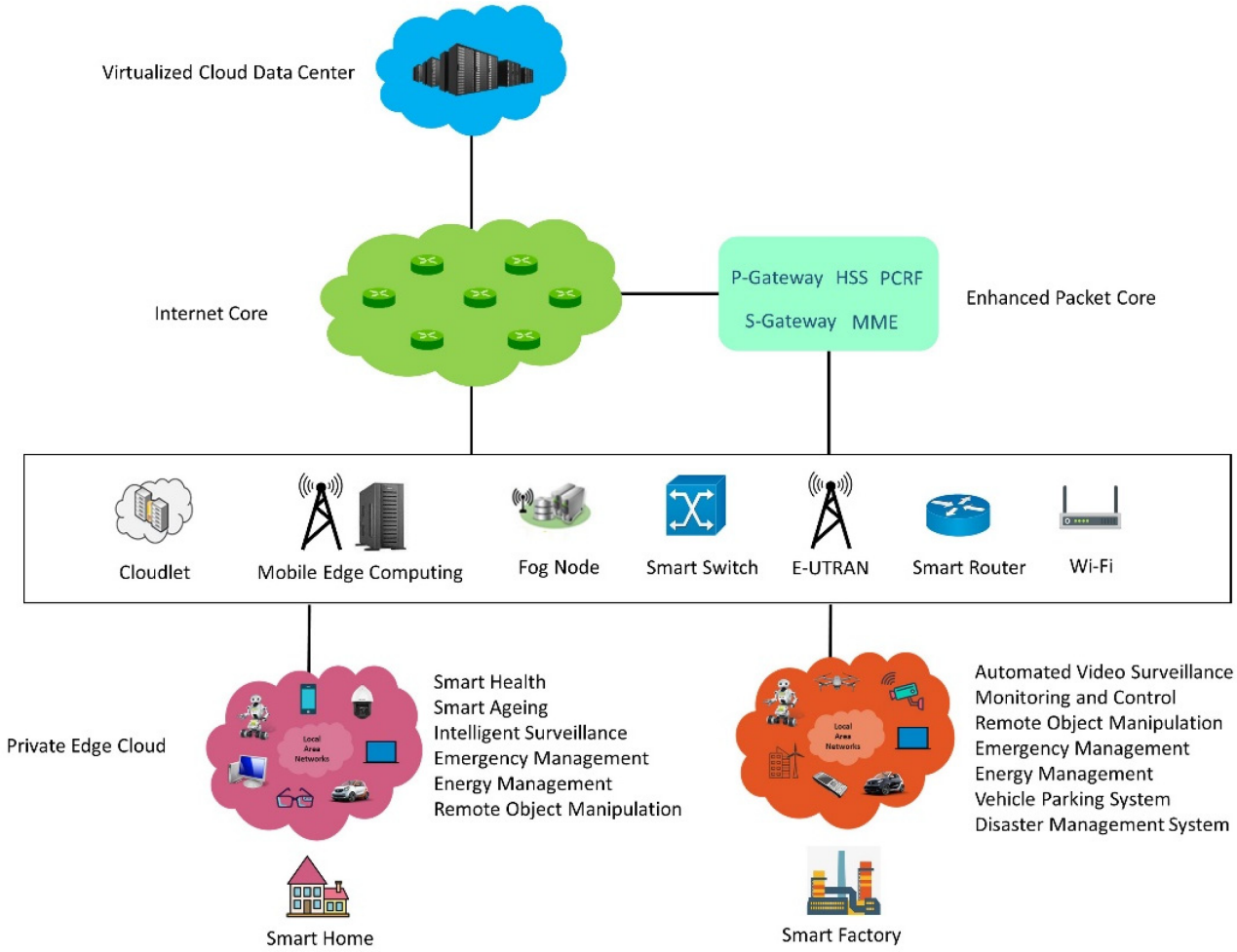
- Bandwidth-Adaptive Streaming

Smart Aging

Smart agingis a computationally intensive, real-time application designed to provide home and health management services. Key functions include health monitoring, disease management, surveillance and security, energy management, and disaster response. Many of these services need to perform real-time tasks such as situation recognition and control, which rely on data-intensive real-time audio-video processing algorithms, machine learning models, and a variety of sensing and actuating devices. These devices include surveillance cameras, audio sensors, smart grid sensors, and biosensors for data collection, as well as systems for taking actions such as calling emergency services or requesting assistance from mobile robots.

Executing these computationally demanding, real-time tasks require computing power that exceeds the capabilities of individual sensing and processing devices. To achieve this, a private Edge Computing node can be created by combining sensor nodes, wearable computing devices, smartphones, and other computing resources available within the LAN.

A variety of stationary and mobile devices, including personal computers, mobile robots, smartphones, and sensors available within the node environment, are combined to create a small private edge cloud. This private cloud delivers essential services such as computing, storage, data processing, sensing, and networking to support the smart aging application. The private mobile edge cloud is also connected to traditional Edge Computing systems and central virtualized cloud data centers over the Internet. These conventional systems are primarily used for processing non-real-time tasks and storing long-term data.



System Architecture

Consumer Scenarios

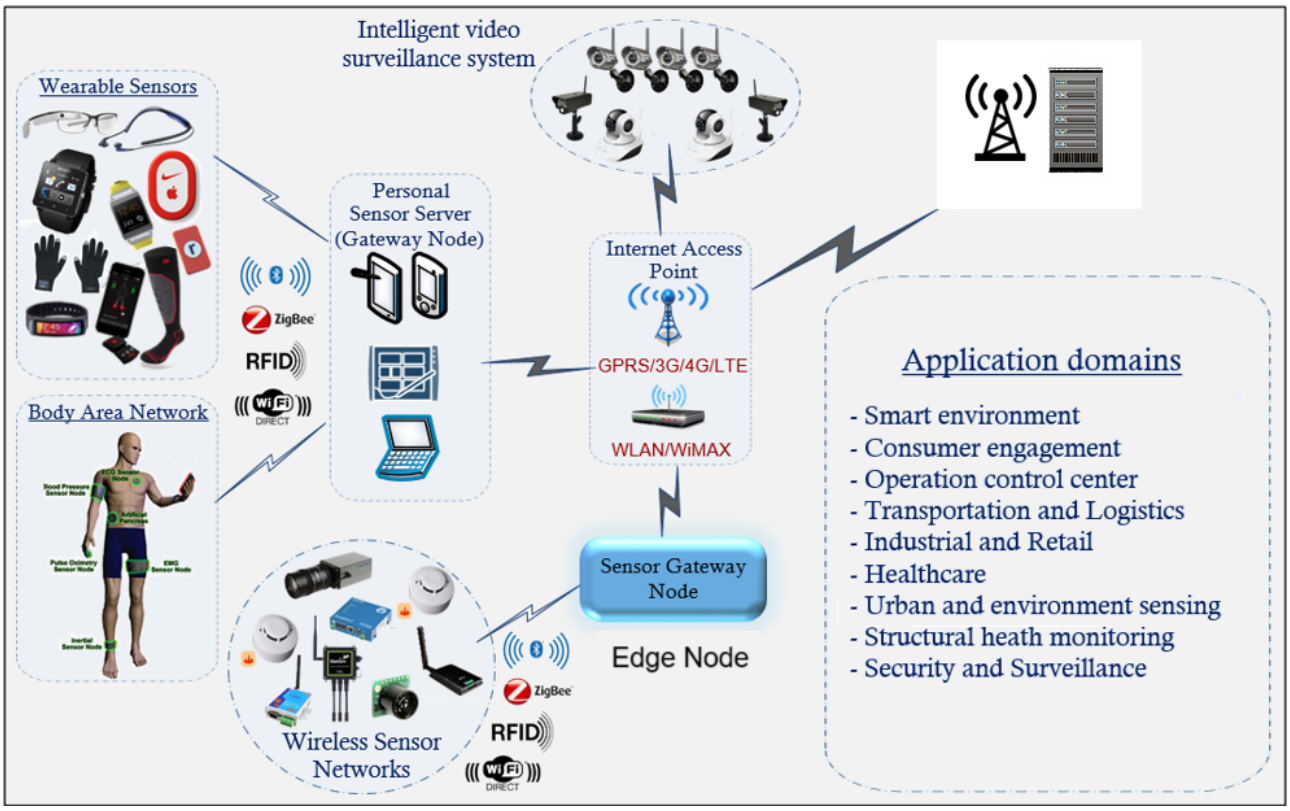
Smart Homes and Distributed Speech-based Interfaces

A smart home environment consists of various devices, such as voice assistants, security cameras, and a variety of sensors, which can utilize AI to learn user preferences, optimize energy consumption, enhance security features, and improve responsiveness in real time. Applying a distributed learning framework to this use case can allow for local training of AI models for data privacy and security.

The architecture is distributed as the devices communicate through a router, hub or gateway. The learning coordinator component can be in the home network or in a central, secure cloud service, and receives and aggregates model updates across each home, without ever transferring raw data. With the use of Edge Computing and a decentralized configuration this architecture ensures minimum bandwidth and data sovereignty, while being capable of operating with low latency and power real-time operations.

In the context of smart homes, distributed learning in Edge Computing enables AI models to be trained and refined locally across multiple devices while preserving data privacy and reducing latency. Instead of sending vast amounts of raw data to centralized cloud servers,

smart home devices—such as voice assistants, security cameras, and a variety of sensors—collaborate in a distributed learning framework. This allows them to learn user preferences, optimize energy consumption, enhance security features, and improve responsiveness in real time. By leveraging distributed AI at the edge, smart homes can deliver more personalized and adaptive experiences while minimizing bandwidth usage and ensuring data sovereignty.



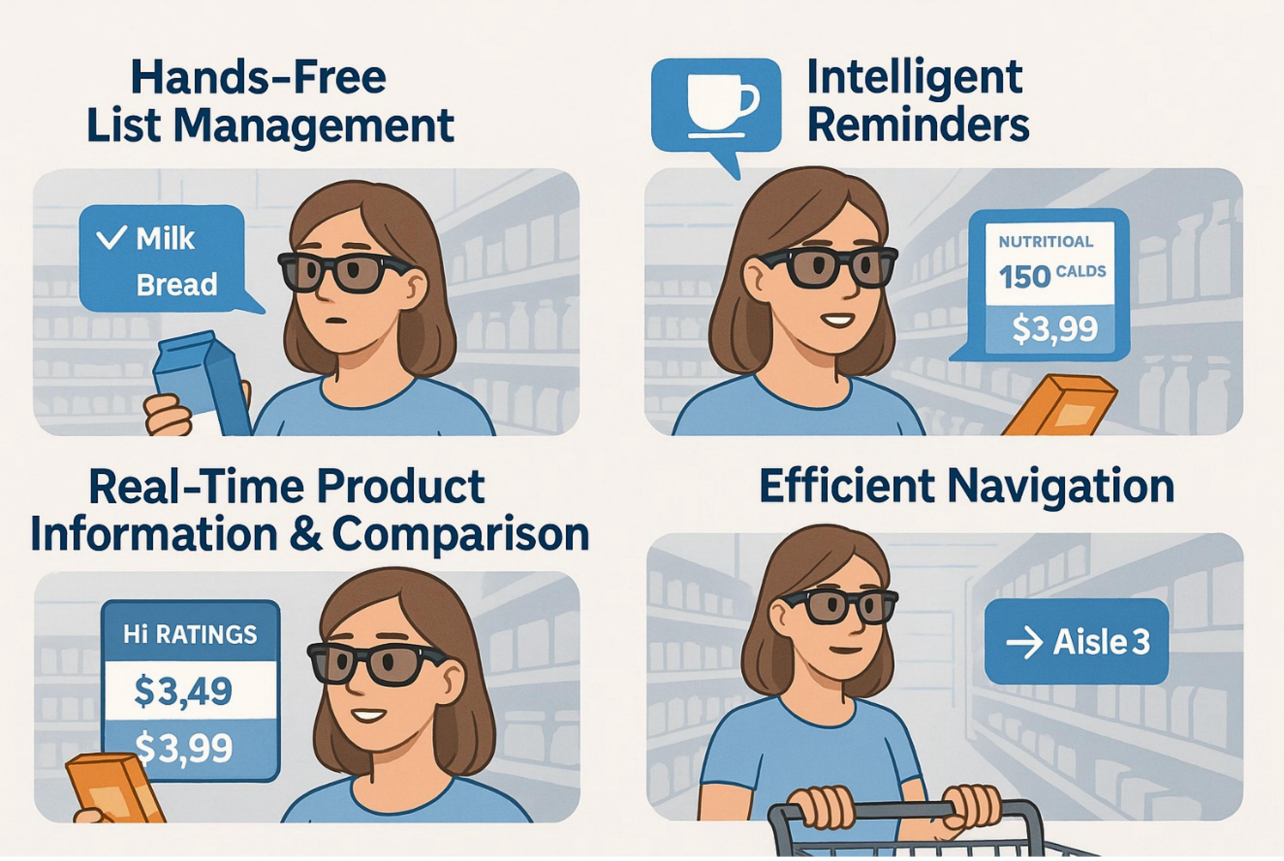
WiFi router with edge node

Personal Shopping Assistant Powered by AI and Dynamic Offloading

A personal shopping assistant integrated into augmented reality (AR) devices, such as smart glasses, represents a transformative AI application enabling advanced consumer technology. This use case explores a practical scenario where AI enhances the shopping experience by utilizing dynamic workload offloading across the device-edge-cloud continuum.

Conventional use of smartphones for managing shopping lists, price comparisons, and product information gets replaced by a hands-free, context-aware AI assistant embedded in AI-enabled smart glasses. This assistant leverages existing consumer behaviors and enhances them through real-time data processing and interaction.

The architecture for this use case is layered, due to resource constraints in the glasses and other parts of the system. The glasses only handle wake-word detection and display overlays with real-time AI processing. Nearby servers would handle more complex processing tasks like object recognition and regional cloud processing can be used to store user preferences, shopping patterns and more. Finally, long-term storage, model training, and complex analytics are managed centrally to leverage scalable resources.

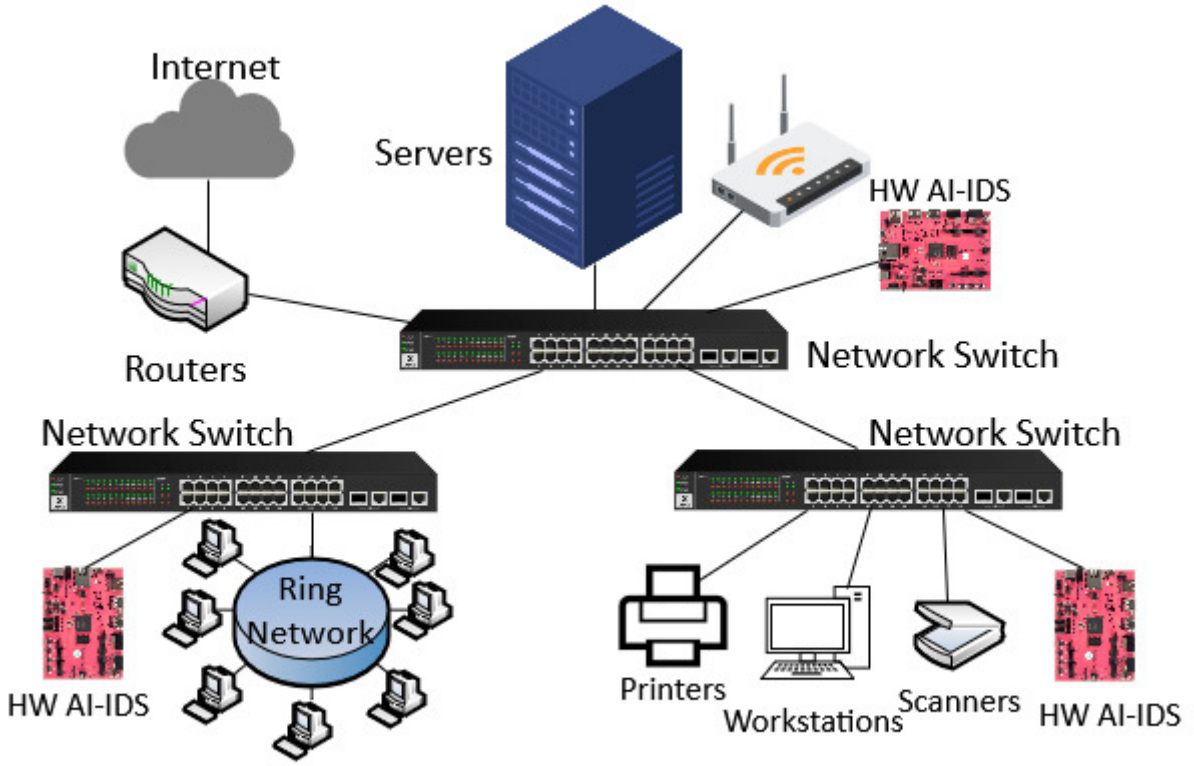


Illustration

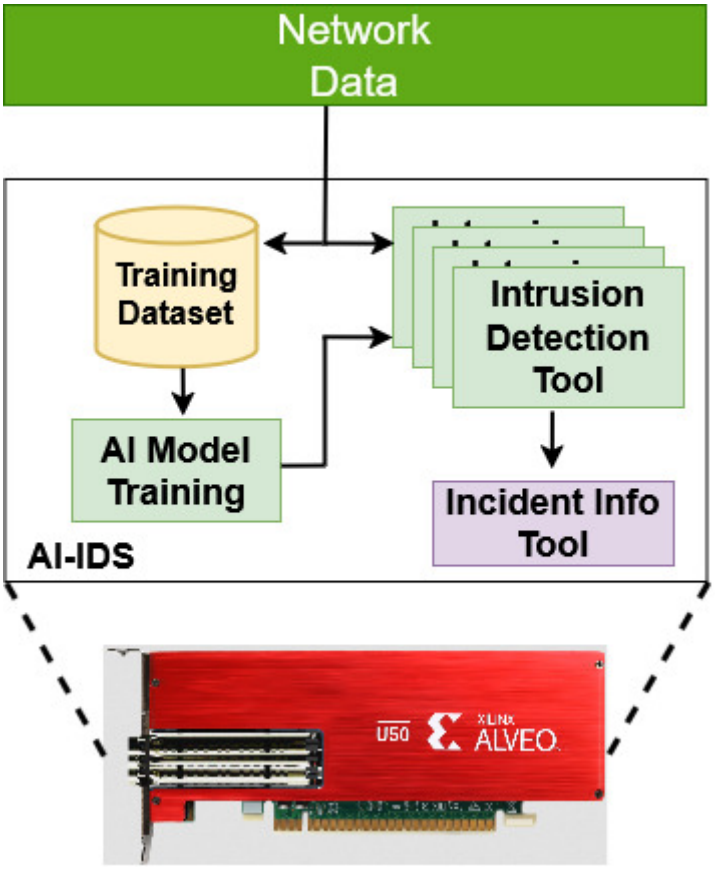
Security Hardware-based AI Intrusion Detection Systems

Hardware-based AI Intrusion Detection Systems (IDS) represent a specialized domain within cybersecurity, wherein AI algorithms are implemented on dedicated hardware platforms. These systems are designed for deployment in Edge Computing environments that are optimized for high-performance monitoring, detection, and prevention of unauthorized access or anomalous activities within the computing infrastructure. For example, in IoT and smart home contexts, these systems facilitate on-device detection of malware and network anomalies, enabling rapid and private threat response.

For instance, an FPGA-based AI IDS is particularly well-suited to be deployed in distributed edge environments as you can see in the figures below. A pre-trained machine learning model —such as a neural network or decision tree—is implemented directly on the FPGA fabric, where other components cover the pre-processing of the data, decision making based on the output of the AI model and generating alerts.



AI-IDS FPGA-based tool integration into Supply Chain network



AI-IDS FPGA-based tool

Supply Chain

A supply chain (SC) is a structured network of organizations, activities, information, and resources that facilitate the sourcing, the production, and the distribution of goods from suppliers to end customers. Supply chains can be turned into interconnected digital infrastructures, which can increase efficiency. Relying on IT infrastructures, however, introduces new forms of risk with no definitive method to safeguard these systems against the growing prevalence and sophistication of cyber threats, such as data breaches and corporate fraud.

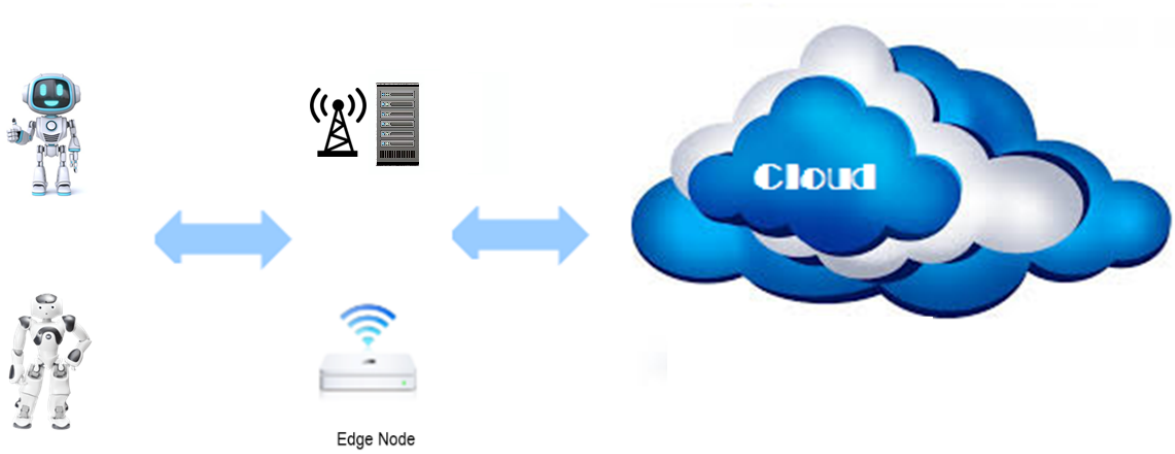
At present, no cybersecurity framework can fully prevent attacks on supply chains. However, by enabling sensitive data and AI insights to remain local at the Edge this can potentially reduce the attack surfaces and provide solutions so that the potential impact of cyber threats could be greatly reduced.



Supply Chain Illustration

Edge Robotics - Drones

Robotics is an area that can benefit from Edge Computing, as robots commonly have limited onboard resources. In a centralized architecture model, robots can rely on computational capacity in the edge node for executing sensitive tasks such as real-time situation analysis and large computational capacity in the cloud for tasks like batch video processing. This model offloads resource-intensive tasks from the robots, so they can have a more lightweight design, longer working life and simpler maintenance. Edge robotics also enables a shared object library that a network of robots can access and use to identify objects, navigation algorithms and other resources that require extensive storage capacity.



Edge Robotics Illustration

Federated Learning (FL)

Aviation companies collect vast amounts of operational data during every flight, yet much of this data remains underutilized for predictive maintenance purposes due to the limitations to building a shared dataset.

Federated Learning (FL) has the potential to allow for collaborative innovation through enabling secure, privacy-preserving collaboration among aviation companies, manufacturers, and regulatory bodies. Instead of transferring raw data across organizations, AI/ML models are trained locally at the airport facilities where data is generated, then model updates—not the raw data—are aggregated centrally, enabling the creation of robust predictive maintenance models while respecting confidentiality. This approach not only improves the accuracy of predictive maintenance but also fosters a new model of trusted collaboration across the aviation ecosystem.

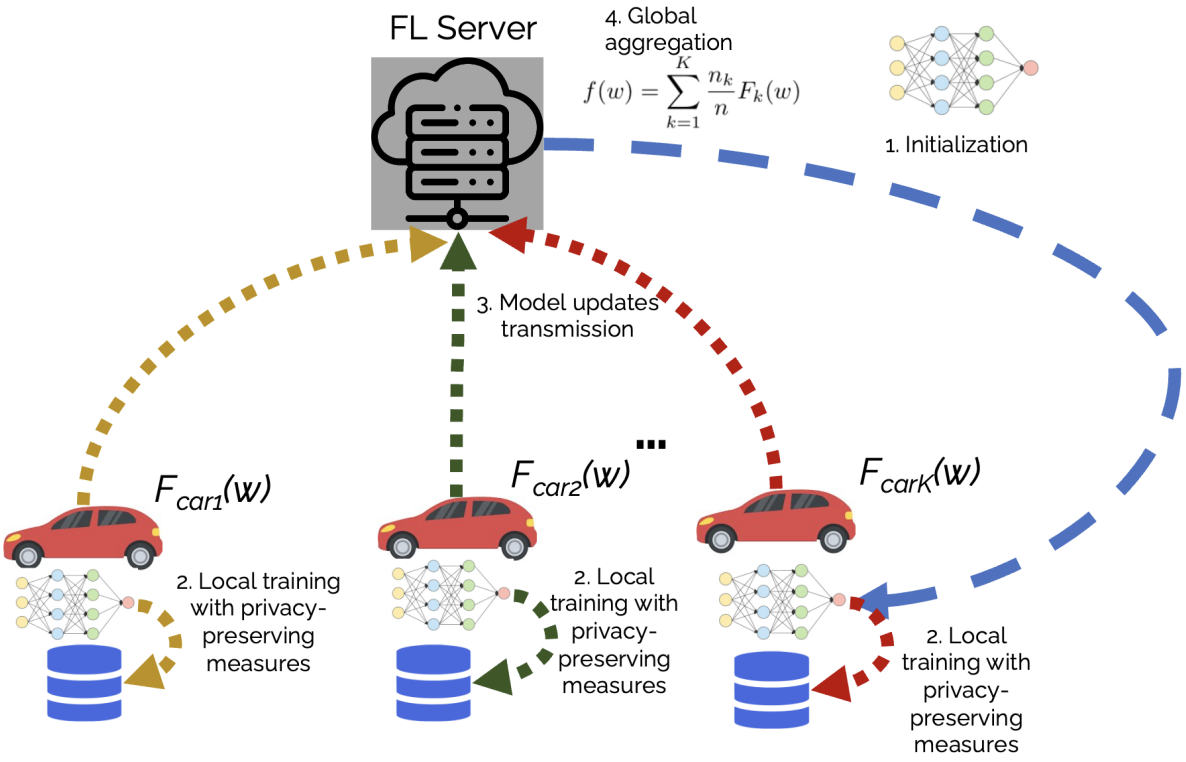
The architecture supporting this use case relies on a distributed Edge Computing topology designed for privacy, security, and efficiency. The edge nodes are placed in airport maintenance facilities, so that the AI models can be trained locally. A federated learning coordination layer orchestrates the model aggregation process, ensuring that collaboration among the various stakeholders remains seamless and secure. Confidential computing \techniques are combined with edge servers with AI acceleration capabilities, trusted execution environments (TEE), and secure storage to maximize security and data privacy.

Automotive/Telematics/Fleet Management (Federated Learning)

Connected vehicles (passenger or commercial fleet) would benefit from FL AI technology to collaboratively train a machine learning model that can be used for predicting real-time traffic conditions without sharing raw data.

The FL approach preserves data privacy by retaining raw data on individual vehicles, ensuring scalability across extensive networks, and enables real-time adaptability to dynamic traffic conditions. Collaborative model training can enhance traffic prediction and management capabilities providing more optimized traffic flow and improved efficiency and fleet management.

The architecture is built on a hierarchical Edge Computing environment combined with a centralized cloud architecture integrated with LTE/5G/6G connectivity. At the edge, vehicles serve as local data processors, training their respective model versions with the local data.



System Architecture

Ultralight Autonomous Drones for Urban Settings

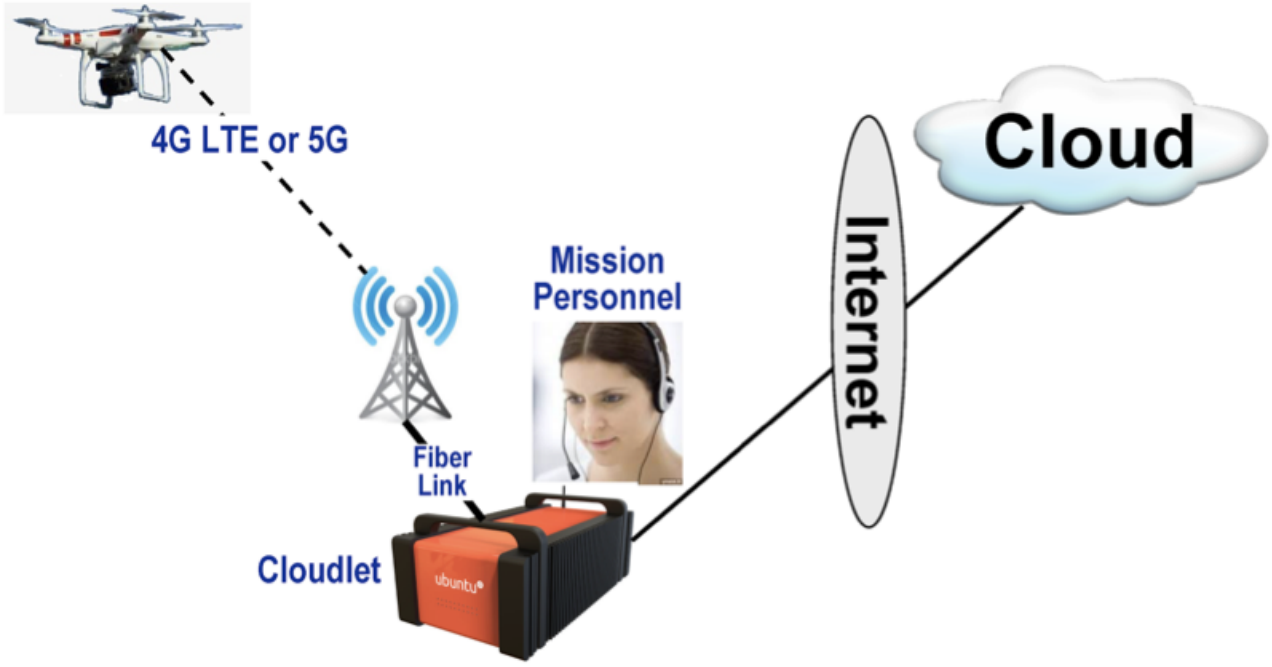
Autonomous drones in urban settings have the potential to make many labor intensive and expensive tasks safer and cheaper. Some examples include law enforcement, traffic monitoring, roof inspection, and bridge inspection. Drones need to have enough intelligence to be able to fly and navigate without a remote human pilot, and greater intelligence correlates with more powerful (and heavier) on-board computing and richer sensing. In dense urban settings, weight and limited flight time are fundamental impediments.

Edge computing enables an ultralight drone to offload compute intensive real-time operations over a low-latency, high-bandwidth wireless network to a ground-based cloudlet. The proof-of-concept system on the below diagram uses a COTS drone, which is programmable via API. The API gives full flight control and access to on-board sensors. Cellular connectivity is enabled by physically attaching a smartwatch with a custom harness (14g) that supports both Wi-Fi and

4G/5G as payload to the drone. On this platform, an Android app communicates with the drone over WiFi and controls it and also offloads compute-intensive tasks to a ground-based cloudlet over 4G LTE.



Proof-of-Concept Hardware Components



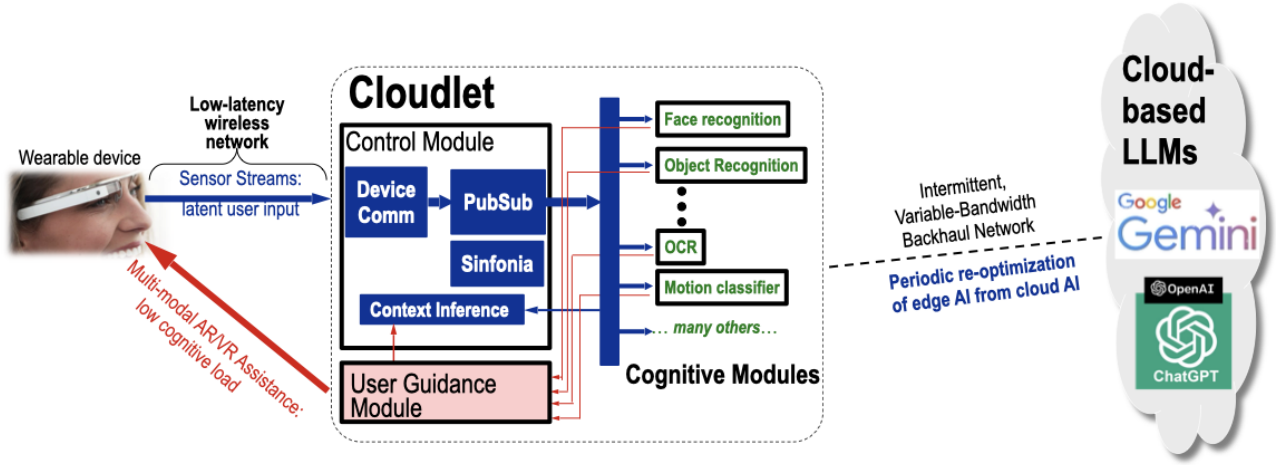
SteelEagle Architecture

Wearable Cognitive Assistance

Wearable cognitive assistance (WCA), an advanced form of Augmented Reality (AR), broadens the capabilities of GPS navigation systems by simplifying complex tasks for novice users by giving step-by-step instructions, and detecting and correcting errors in real-time. The solution combines wearable computers, Edge Computing, and AI algorithms (e.g., computer vision,

speech recognition, natural language understanding, and other ML-based capabilities). WCA apps use the compute-intensive AI algorithms for their time-critical execution, improving user productivity on a wide range of assembly, troubleshooting, and repair tasks. Since 2014, over 20 WCA apps have been built, thus offering substantial validation of the concept.

The below diagram shows a wearable device and provides a first-person viewpoint of a user's task. The AI operation typically has memory, CPU, and GPU demands that cannot be sustained on the wearable device; sensor streams from the device (such as video, audio, accelerometer, and gyroscope) are transmitted over a wireless network to a nearby cloudlet for task-specific processing. Guidance in visual, verbal, or tactile form is generated on the cloudlet, transmitted over the wireless network, and available to the user via the wearable device. Outside the time-critical paths of user-cloudlet interaction, fine tuning of the AI on the cloudlet can be performed by cloud-based AI such as LLMs.



Edge Computing Architecture for Wearable Cognitive Assistance

Conclusions and Future Directions

A key takeaway is that adding Artificial Intelligence and machine learning to Edge Computing is a complex undertaking. A solid understanding of all the components that go into both Edge Computing and Artificial Intelligence is essential before designing an architecture that can support these integrated use cases. Unlike other AI applications, Edge may have significant constraints that need to be taken into consideration for a successful design and implementation.

- The use cases demonstrate the great promise that AI enabled Edge Computing applications have for the future
- Pay close attention to the data requirements and resources needed to support the use case
- Match the right architecture to the use case for optimal results
- Consider keeping the AI processing and data as close to where the data is generated to minimize latency and realize network efficiencies

- The Open Source communities are well positioned to address the architectural requirements and tools needed for Edge Computing AI use cases. Contributions are encouraged from integrators, research institutions, and commercial adopters.

Challenges and Risks

Integrating AI into Edge Computing is no means an easy undertaking. There is significant risk that the return on investment will not be as expected. There is ongoing financial pressure to reduce the cost of edge devices and distributed infrastructure; however, this must be carefully balanced against the potential trade-off, as lower-cost edge devices may lack the computational capacity required to perform the level of data processing necessary for AI applications to generate actionable insights.

While a high degree of automation is needed to support AI, the applications themselves are not significantly different from other high CPU/data driven demanding architectures. Since AI is still so new it is unclear how much AI is just better machine learning and very smart search algorithms and how much is AI -- or if there is even a difference between these concepts.

Another factor that affects the outcomes is how it relates to incorporating networks and networking in the overall architecture. For the most part the network is the component that is most outside of the control of the developer, because the availability of the network connections is dependent on third party telecommunications providers.

Future Directions

While it is still early in the edge AI revolution, the next twelve to eighteen months look to be an exciting time with new discoveries and understanding of how to best use AI to optimize edge use cases. Right now, there is lots of research and testing to see how much is useful and how much is not.

For now, it is expensive to create the supporting infrastructure and data lakes, so how much ROI will come out of them, given the massive amounts of resources that are needed to create them is uncertain. However, the potential is there as the costs of the infrastructure falls, for the benefits to grow and as people and companies become more comfortable with the concepts and build on the research and deployments that have already produced results.

References

This is a list of a few interesting articles and further information related to Edge AI:

- Edge Computing: How decentralisation can reduce latency and support AI– Recent article on decentralized storage in an Edge Architecture
- (General Open RAN) White Paper- Advanced Management of Radio Access Networks using AI
- Fujitsu launches new 5G vRAN solution, contributing to the realization of flexible, open networks Feb 2023
- 10 Edge Computing use case example - From STL Partners, a UK analyst firm
- AI and Generative AI: transforming Europe’s electricity grid for a sustainable future - Report from the European Commission

- Forrester. Building the Future with AI at the Edge
- Edge AI vs Cloud AI: A Comparative Guide
- Edge AI: A Comprehensive Survey of Technologies, Applications, and Challenges
- Characterizing the Deployment of Deep Neural Networks on Commercial Edge Devices
- Analyzing Edge Computing Devices for the Deployment of Embedded AI

- A Review of Algorithm & Hardware Design for AI-Based Biomedical Applications
- Artificial Intelligence in the Military: An Overview of the Capabilities, Applications, and Challenges
- A systematic literature review on hardware implementation of artificial intelligence algorithms
- Hardware-Assisted Machine Learning in Resource-Constrained IoT Environments for Security: Review and Future Prospective
- A Survey on the Optimization of Neural Network Accelerators for Micro-AI On-Device Inference
- Hardware + AI-Based Biomedical Applications
- Hardware + AI-Based Military Applications
- Hardware implementation of AI algorithms, including healthcare, autonomous airplanes and vehicles, security, marketing customer profiling and other diverse areas
- Hardware + AI-based Security
- Hardware + AI-based DNN (computer vision, data analytics, robotics)

Authors

Contributors

- **Beth Cohen**- Telecom Product Strategy Consultant, Luth Computer
- **Buland Khan**- Chief Technologist & Senior Principal - DMTS, Wipro
- **David Solans Noguero** - Scientific Researcher, Telefónica Innovación Digital
- **Flavio Esposito**- Associate Professor of Computer Science, Saint Louis University, USA
- **Grigorios Chrysos**- Senior Researcher, Technical University of Crete
- **Ildiko Vancsa**- Director of Community, OpenInfra Foundation
- **Julian Thomas**- PhD candidate, Chair of Applied Cryptography, Computer Science Department, Friedrich-Alexander-Universität Erlangen-Nürnberg
- **Mahadev Satyanarayanan**- Jaime Carbonell University Professor of Computer Science, Carnegie Mellon University
- **Moshe Shadmon**- Founder and CEO, AnyLog
- **Nenad Gligoric**- CEO, Zentrix Lab
- **Rajat Kandoi**- Master Researcher, Ericsson Research
- **Rob Hirschfeld**- Founder and CEO, RackN
- **Sayed Chhattan Shah**- Southern Illinois University