LF Edge Architecture Whitepaper V2 (2022) - Working Draft

Timeline: Final Draft by the end of June
Delivery at Edge Computing World / Open Source Summit (week of June 20)

Milestones:

M0	Confirm outline	COMPLETE
M1	Share working draft for initial committee input	COMPLETE
M2	First round of committee input due	April 19, 2022
M3	**First full round of content due**	**May 17, 2022**
M4	**Content lock**	**May 24, 2022**
M5	Proofread,refine and polish	May 24- 28, 2022
M6	Delivery to Creative Services	May 31, 2022
M6	Final review / approval of layout	June 14, 2022
M7	Publishing `	Week of June 20

Work towards OSS Summit and Edge Computing week of June 20

Table of Contents

- Introduction
    - Recap of LF Edge: What and Why
    - Recap taxonomy from last paper, reference link
    - Stress the goal including enabling tech providers and end users to focus on value add
    - OSS isn't about giving your IP away
    - PoV on ideal edge tech stack
        i. Key tenets and architectural concepts
        ii. Why open really matters, e.g. trusted data
- OT and IT convergence
    i. Why OSS and Linux in general is critical across the board
    ii. Highlight differences between OT and IT
    - Overall trends, e.g. software PLCs, physical to virtual separation of concerns
- Scaling deployments in the real world
    - Key principles
        i. What's different at the edge + related tradeoffs
    - Difference between application and infrastructure management
        i. Importance of separating these two planes in architecture
    - Four main paradigms for edge management
        i. Data center (metro/regional)
        ii. Distributed edge cloud
        iii. Client edge
        iv. Constrained edge
    - Related contributions from each project (make sure in each to highlight differences between application and infrastructure management)
        i. EVE - bottoms up approach, extending cloud-native to lightest hardware possible
        ii. Open Horizon - overall approach and bleeding into mobile and constrained devices
        iii. SDO

        iv. etc…
    ○
- Security
    - OSS and security, how it works, overall trends (stats are ideal)
        i. What's different at the edge
        ii. Key threat vectors, put into context with real-world breaches
        iii.
        iv. Considerations at data, network, compute, and code levels
        v. Vision for data trust vs. just security - e.g. Alvarium
        vi. Related contributions from each project
- Edge networking
    - Overall trends and tradeoffs in edge networking, from constrained devices to regional edges
    - Detailed considerations on WANs, especially private 5G (Akraino, Baetyl, Edge Gallery Focus)
    - Considerations for local area networking / distributed devices (e.g. "fog"... All projects but focus on IoT frameworks)
    - Related contributions from each project
- IoT
    - TBD (EdgeX and Fledge focus)
- Edge analytics
    - Inference vs training
    - Federated learning
    - TinyML (eKuiper input)

    - General refresh on projects
        i. 2021 project milestones - overall summary
        ii. 2022 focus areas - by project
        iii. Examples of market adoption
        iv. Examples of cross-project collaboration

Start of New Paper

Insert TOC

Introduction

This white paper is a follow-up to the LF Edge community's original, collaborative 2020 paper titled "*Sharpening the Edge: Overview of the LF Edge Taxonomy and Framework*" which details the LF Edge taxonomy, high level considerations for developing edge solutions, key use cases and provides an introduction to LF Edge.

As defined in the *Sharpening the Edge* paper, edge computing is the delivery of computing capabilities to the logical extremes of a network in order to improve the performance, security, operating cost and reliability of applications and services. Edge computing mitigates the latency and bandwidth constraints of having to pass raw data to the cloud for processing. By shortening the distance between devices and the computational resources that serve them, the edge can usher in new classes of applications. In practical terms, this means distributing new resources and software stacks along the path between today's centralized data centers and the increasingly large number of deployed nodes in the field, on both the service provider and user sides of the last mile network. In essence, edge computing is distributed cloud computing, comprising multiple application components interconnected by a network.

The goal of the LF Edge taxonomy (Figure 1) is to clarify market confusion by breaking the continuum down based on inherent technical and logistical tradeoffs rather than using ambiguous terms. The taxonomy also comprehends a balance of interests spanning the cloud, telco, IT, OT, IoT, mobile and consumer markets. For more details on the taxonomy, reference the 2020 paper.

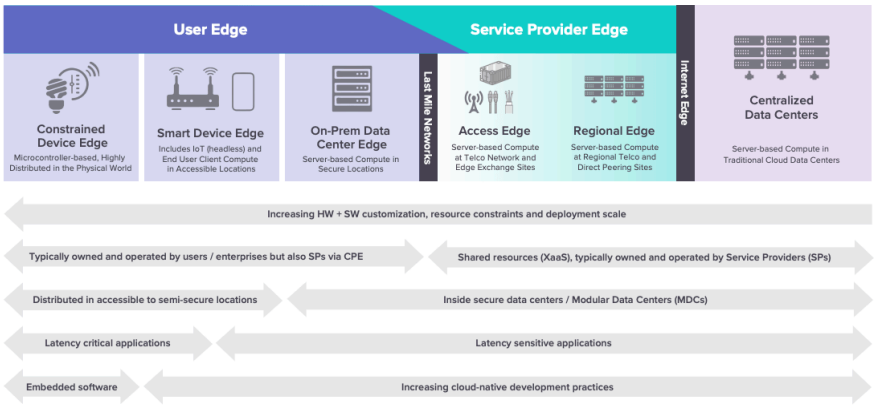


Figure 1. LF Edge Taxonomy

As two years have passed, much has changed in the edge ecosystem and the LF Edge community has grown considerably and made great progress towards building an open, modular framework for edge computing. This publication builds on the 2020 paper by diving deeper into key areas of edge manageability, security, connectivity and analytics and highlights how each project is addressing these areas.

Recap of LF Edge: What and Why

The Linux Foundation's LF Edge (LFE) was founded in 2019 as an umbrella organization to establish an open, interoperable framework for edge computing that is independent of hardware, silicon, cloud or operating system. The project offers structured, vendor neutral governance and has the following mission:

- Foster cross-industry collaboration across IoT, Telecom, Enterprise and Cloud ecosystems
- Enable organizations to accelerate adoption and the pace of innovation for edge computing
- Deliver value to end users by providing a neutral platform to capture and distribute requirements across the umbrella
- Seek to facilitate harmonization across Edge projects

As with other LF umbrella projects, LF Edge is a technical meritocracy and has a Technical Advisory Council (TAC) that helps align project efforts and encourages structured growth and advancement by following the Project Lifecycle Document (PLD) process. All new projects enter as Stage 1 "At Large" projects which are projects that the TAC believes are, or have the potential to be, important to the ecosystem of Top-Level Projects, or the Edge ecosystem as a whole. The second "Growth Stage" is for projects that are actively developing their community of contributors, governance, project documentation, and other variables, and have identified a growth plan for doing so. Finally, the third "Impact Stage" is for projects that have reached their growth goals and are now on a self-sustaining cycle of development, maintenance, and long-term support.

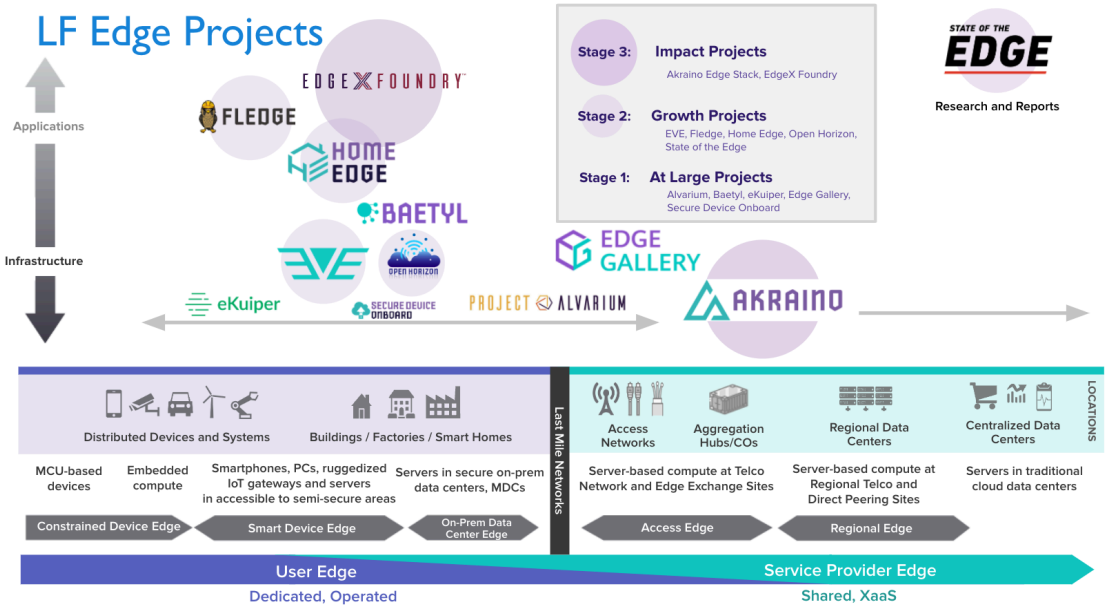


Figure 2. LF Edge Project Landscape

Macro Trends at the Edge

Edge computing is a hot topic today, a trend being driven by the exponential growth of both devices and data on networks. The sheer amount of data on networks is forcing a shift to a more distributed model. This is not surprising as throughout the history of computing we have seen the pendulum swing between centralized and decentralized computing models every 10-15 years. We went from mainframes to PCs, then with the internet came the rise of mobile and the resulting ubiquitous connectivity drove investments in the cloud.

IoT use cases are a big driver of the edge trend because networks have historically been designed for download-centric use cases but IoT solutions are inherently upload-centric. 5G is also accelerating the need for edge computing. 5G networks rely on hyper-local connections between nearby antenna base stations to offer extremely high bandwidth and low latency connections to data consumers. This connectivity is faster than the upload land-based connection, hence despite the ultra fast local connection to the base station there's now a bottleneck upstream. This requires edge computing to pre-process data locally to both benefit from the high bandwidth connection downstream to users and offload the network bottleneck upstream.

The net is that edge computing is driven by needs spanning latency, bandwidth savings, autonomy, security and privacy. Use cases include IoT, Edge AI, 5G (especially private 5G) and security, spanning all market verticals. It is important to note that the cloud isn't going away with the rise of the edge. We are simply going to see a broader distribution of computing resources.

Cloud-native Expanding to the Edge

Over the past 10 years, software architectures have increasingly adopted cloud-native principles, meaning platform independence, loosely-coupled microservice-based architecture and continuous integration and delivery (CI/CD). Cloud-native architecture provides maximum flexibility for continuously delivering new software innovations across the spectrum of distributed compute resources. The edge is a continuum and a key goal is to extend the public cloud experience as far into the physical world as possible until technology constraints mandate embedded software. Whether all elements enabling a use case run entirely on prem, or the solution also takes advantage of the cloud for computing scale depends on a combination of use case and risk profile. While a building automation solution may safely take advantage of cloud resources, this is not necessarily the case for a nuclear power plant. Still, we will see more and more solutions that take advantage of both edge and cloud resources.

Edge computing solutions inherently comprehend the cloud in some form, otherwise the conversation is just about legacy on-prem applications. There are a variety of ways the cloud and the edge can work together, for example:

- Cloud-centric. The cloud is used as a centralized data store and for near-infinitely scalable computing capabilities, working in concert with edge applications that collect, normalize and pre-process data from disparate data sources.
- Cloud Support. The cloud is used for training AI/ML models that are subsequently deployed at the edge, with all data being retained on-prem.
- Edge-centric. Customer data remains entirely on-prem with only remote orchestration of hardware and applications being performed from the cloud. This model benefits from centralized infrastructure management of fleets of edge computing resources but addresses concerns over security and IP protection, and/or requirements for data privacy and sovereignty.

The term "edge native" has also emerged as a way to describe architectures that leverage cloud-native principles but with edge-centric operation in mind. This includes specific accommodations for location, latency, bringing together various

software components and more. Important to note is that edge native architectures do not mean that the cloud is not comprehend, rather the solution is architected to prioritize on-prem needs.

Open Source Driving Standards

Despite the benefits of edge computing, developing and deploying edge solutions requires a complex mix of hardware, software and skill sets. Given the inherent complexity of bridging the physical and digital worlds at the edge, standardization and interoperability are especially important.

Standards Development Organizations (standards bodies or SDOs) are still critical for driving standards, however in recent years open source collaboration has emerged as the modern way to drive standardization. This trend is being accelerated by the transition to cloud-native software architectures based on microservices that are bound together with APIs. In addition to the shared technology investment helping developers focus on value creation instead of reinvention, modern cloud-native software architectures make it easier to divide resources and create separation of concerns between open source and proprietary interests. This in turn enables developers to more rapidly innovate and mix and match proprietary and open source components and develop ecosystems built around a common foundation.

OT/IT Convergence

OT/IT Convergence has been a hot topic over the past several years. The *Sharpening the Edge* paper touches on the unique needs of Operational Technology (OT) and Information Technology (IT) organizations, and what has become clear since is that organizations tend to approach the edge continuum either by expanding up from traditional OT in the physical world or down from the traditional IT data center and cloud. This can be viewed as “OT Up” vs. “IT Down” and each trajectory brings its own set of considerations.

OT is rooted in industrial processes within factories, refineries, buildings and beyond, and connecting their formerly isolated operations for increased visibility. Prescriptive analytics, that can be accessed remotely, is a key driver for Internet of Things (IoT) solutions. OT environments and constraints are unique; lack of electrical power, low/unreliable bandwidth, dirt, heat, humidity, equipment with 20-30 year life-spans, regulations concerning safety/security and proprietary systems from hundred year-old suppliers. OT users have diverse backgrounds – typically they are mechanical, electrical, chemical, and biological engineers, plus scientists, mechanics, and even operators without college degrees. They are not typically IT administrators, software developers, nor computer scientists who are comfortable with the latest cloud tools and cloud-native application development.

On the other hand, IT administrators are not industry subject matter experts. They typically don’t know the business and scientific sides of technology operations, even though they are in charge of data and device security and often need to maintain deployed operational technology. To save cost, and manage operations from anywhere, IT staff prefer centralized applications (e.g., running in data centers or public clouds) where they have fewer touch points to manage, and where they feel more confident about the physical security of the servers running the applications.

An “IT Down” approach involves extending data center practices toward the physical world while still maintaining centralized control. This includes technologies and practices such as software-defined infrastructure, cloud-native software applications and architecture, CI/CD pipelines, and more. A key goal for the IT Down approach is to extend these principles as far into the edge continuum as possible, without compromising the unique needs of OT. IT’s goal is to retain centralized and remote management at scale that provides maximum agility to rapidly software-define new use cases to stay competitive at the edge, while also making sure to protect critical industrial operations, devices, and processes.

Today, “OT Up” edge deployments often leverage lighter “gateway” class hardware that can run the cloud-native style applications. The OT Up approach therefore modernizes edge infrastructure to not only collect and process data locally,

but also to normalize various non-IP protocols into a modern standard so that data can be communicated over a network (e.g. MQTT, OPC-UA). Edge devices can buffer data in a local database for data persistence regardless of backend connection status, and they can perform light local analytics such as spanning a rules engine to a machine learning inference model.

Using similar logic, Gartner has coined the terms “Cloud Out” and “Edge In”. In the Cloud Out approach, the idea is to make data center-based services (or at least components of them) closer to where the consumers of these services are. Meanwhile, in the Edge In approach, the idea is to perform software logic (business processes, abstraction, data cleanup, etc.) as quickly as possible to allow local actions to be taken without the need to go to the cloud.

The two approaches to edge must not be treated as mutually exclusive. Both IT infrastructure and OT applications must strike an appropriate balance as they move toward each other. As these two trajectories intersect, the lines between OT and IT will continue to blur and collaboration will be essential. This blurring of the lines will result in less physical separation between OT and IT resources (e.g., PLCs vs. servers) and more collaborative management of the same physical resources. For example, software-defined PLCs are increasingly being consolidated onto edge infrastructure that can also perform data analytics. It is critical for developers and solution providers to jointly focus on core needs in areas such as performance, uptime, safety, and security, and not remain too narrowly focused on legacy assumptions for who they believe is responsible for a given role in the field.

Linux in the Industrial World

The fourth industrial revolution requires its own version of the LAMP stack (Linux, Apache, MySQL, PHP). There are well over a hundred proprietary industrial protocols in use today due to industrial solution vendors creating their own to lock customers into their systems. Even when two PLCs use the same protocol, the data models for like machines are unique. Adding to that data chaos, the schemas in various OEE, historian, MES, and SCADA databases, that must work in concert, are inevitably different. For these reasons and more, OT data has historically remained siloed and often lacking context. Imagine the MLOps challenges when this data situation is your starting point.

Henry Ford proved the value and efficiency of the assembly line, shared components and benefits of ordered processes. Industrial 4.0 software requires the same consistency that OT taught the world a hundred years ago. It needs the elimination of data silos, consistent APIs and methods, multiple data type support (e.g. time-series, image, vibration, radiometric, transactional), shared orchestration and security. Open source software (OSS) has become the virtual equivalent of Henry Ford’s insights. When the foundation needs to be common to enable the next leaps in value creation for all, OSS is today’s proven choice.

As has been written many times, OSS software is not about free but freedom and time to market. Like TCP/IP became networking’s foundational technology not a differentiator, OSSdoes the same for OT. By sharing services like data infrastructure, protocol translations, data pipeline management, security and orchestration resources can be applied higher up the value chain while ensuring interoperability. Like Linux itself, commercial support is available for those benefiting from the code.

Data Trust

All of the current technology trends are creating a network effect that requires more software-defined intelligence at all points in networks. We’re also seeing a rapid rise in Artificial Intelligence (AI) solutions which in turn is driving more data through automation. This represents not only a massive opportunity but also a real risk with the creation of more fake data (e.g. deepfakes), which in turn necessitates more measures for security and data trust.

Where we’re headed in the next 5-10 years is ambient computing, meaning compute capability pervasively embedded in the physical world, effectively making fixed compute the new mobile. In order to take advantage of all of these distributed compute resources to drive new business models and customer experiences through interconnected ecosystems, it’s critical to build systems that enable data to flow through heterogeneous networks with measurable confidence.

The transparency of open source collaboration inherently drives a degree of trust. As such, each of the projects within LF Edge are contributing to this goal. The mission of Project Alvarium is to build out the concept of trust fabrics that take a system level approach to ensuring data confidence by binding together various trust insertion technologies with a standardized SDK and algorithms that drive measurable data confidence. While a longer-term vision, the only way to get to this future state is by architecting today with an open approach that drives trust and transparency between different internal and external stakeholders.

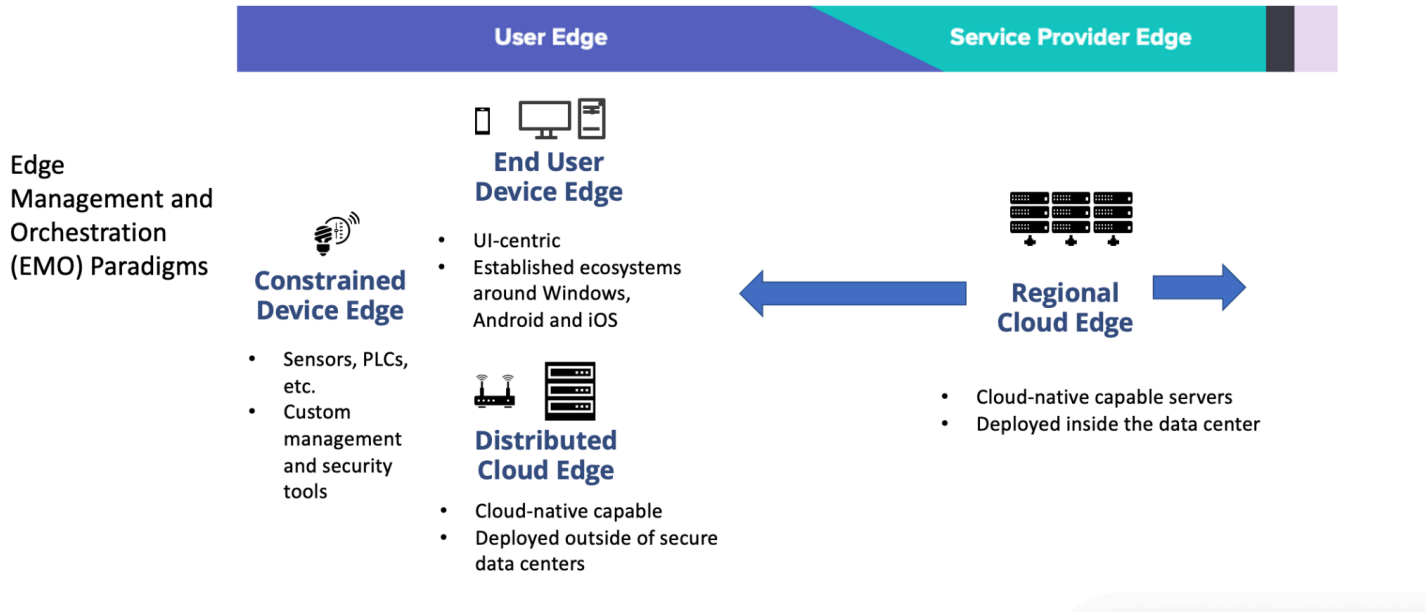
Scaling Edge Deployments in the Real World

There has been much discussion on edge computing in recent years but there is also still quite a bit of confusion. The 2020 LF Edge taxonomy aimed to reduce this confusion with a high level overview of the edge continuum and use cases. In the two years since, we have seen increasing investment for edge processing to enhance established workloads such as caching video content to reduce latency and upstream bandwidth consumption when streaming to consumers. We've also seen increasing proof-of-concepts (PoCs) and pilot deployments of distributed edge computing solutions in both the Industrial and Enterprise spaces.

As with any solution, edge computing starts with a use case, then a focus on applications and hardware for an initial PoC. Once business value is proven in a PoC, the challenges of deploying and managing distributed edge computing solutions at scale in the real world become readily apparent. The following sections further break down the edge continuum and dive deeper into areas such as management, security, connectivity, and analytics, including related LF Edge project contributions focused on simplifying deploying edge computing solutions in the real world.

The Four Main Paradigms for Edge Management

Spanning the edge continuum from the User to Service Provider Edge, there are four main paradigms for securing and managing deployments - the Constrained Device Edge, End User Device Edge, Distributed Cloud Edge and Regional Cloud Edge. Each of these general paradigms have similar principles for security and management, however they necessarily require different tools for management, security, connectivity and analytics due to inherent technology constraints and differences in ecosystem maturity. Driving factors include hardware resource constraints, whether the use case is user- or telemetry-centric, whether the edge nodes are physically-accessible with no defined network perimeter or they are protected within a highly-secure data center, and how reliable the connection is between the edge node and centralized infrastructure.



The Four Main Edge Management and Orchestration Paradigms (placeholder graphic)

Regional Cloud Edge

The Regional Cloud Edge paradigm involves edge resources deployed in regional and access edges and traditional on-premise data centers and borrows heavily from traditional data center tools and practices for manageability, orchestration and security. Administrators can count on computing resources being physically secure, having a well-defined network perimeter (e.g. protected by firewall), and a constant connection (typically fiber) to their orchestration console. We are seeing some evolution in management, orchestration and security tool sets with the adoption of cloud-native principles, proliferation of Kubernetes, new solutions to manage distributed data centers in growing scale, adoption of software-defined networking, and so forth. These data centers are primarily located within the Service Provider Edge as defined in the LF Edge Taxonomy but also bleed into traditional on-prem data centers at the User Edge.

Distributed Cloud Edge

The Distributed Cloud Edge management paradigm involves telemetry–centric edge nodes that are capable of supporting cloud-native architecture but deployed outside of traditional data centers. This can span from a gateway device in a truck to a cluster of servers on a factory floor or in a retail store, bleeding into the fringes of the data center. These edge nodes primarily sit on the User Edge and are a form of “cloud” resources in that they provide server-like services to end user devices, other proximal edge nodes and constrained sensors and devices.

As you progress from the data center to the Distributed Cloud Edge paradigm there is increasing diversity in hardware types and application needs due to varying environmental conditions, regulatory requirements and domain-specific use cases. However, a key attribute of Distributed Edge Cloud resources is that they are capable of running Linux and supporting cloud-native principles including platform independence, hardware abstraction through containerization and

virtualization and continuous delivery (e.g. CI/CD) of applications. In contrast, resources at the Constrained Device Edge are even more diverse and inherently require embedded software and custom management and security tools to align with the capabilities of the hardware.

Distributed Cloud Edge nodes benefit from purpose-built edge orchestration solutions that extend the public-cloud experience to on-prem and field locations.. Abstraction of applications from hardware using virtualization and containerization technologies simplifies the experience for developers by exposing virtual resources (e.g. CPU, memory, storage, networking) to enable software-defined functionality through continuous delivery of virtual machines and/or containers. While the operating model is similar to resources in the centralized cloud and the Regional Cloud Edge, Distributed Cloud Edge management and orchestration solutions need to take into account that the computing resources are typically more CPU and memory constrained (for example, as low as 1GB of available system memory), physically-accessible to hackers, often deployed on untrusted networks, and are likely to periodically lose connectivity to centralized resources. As such, management and security solutions optimized for the data center are typically not suitable for Distributed Cloud Edge use cases.

End User Device Edge

Also on the User Edge, end user devices (e.g. PCs, Smartphones, Tablets) are dedicated to specific users and are UI-centric. The End User Edge Device management paradigm benefits from well-established ecosystems built around operating systems like Windows, iOS and Android. End user devices have the advantage of applications that users interact with dynamically and can interpret language differences, unlike IoT devices that must be designed to work together so data models are interoperable. Unlike telemetry-centric Distributed Edge Cloud nodes, end user devices also benefit from users being present to notice potential security issues, for example if your email is hacked.

Constrained Device Edge

On the far extreme of the continuum is the Constrained Device Edge management paradigm, characterized by devices and sensors in the physical world These devices leverage microcontrollers and have kilobytes to megabytes of memory, rendering them suitable for performing basic functions. Unlike resources upstream, these devices do not have the resources (e.g. CPU power, memory) to support an abstraction layer that enables cloud-native principles like containerization and they inherently require highly customized management and security tools due to the resource constraints. Software and firmware updates tend to be monolithic in nature.

Immutable vs. Mutable Edge Resources

When moving towards the End User Device Edge and Constrained Device Edge, the resources to be managed are not necessarily immutable. In the cloud, Metro and Regional Cloud Edge and Distributed Cloud Edge, computing resources are almost always considered immutable and general-purpose and organized as a pool of resources defined by a configuration. At the more constrained lower extreme of the edge continuum, it is also possible to treat the resource as immutable (for example creating a super computer based of Raspberry Pi's) but the resources can also be mutable and categorized (e.g. by ownership, geolocation, and function) and managed as ad-hoc clusters that are defined by discovery.

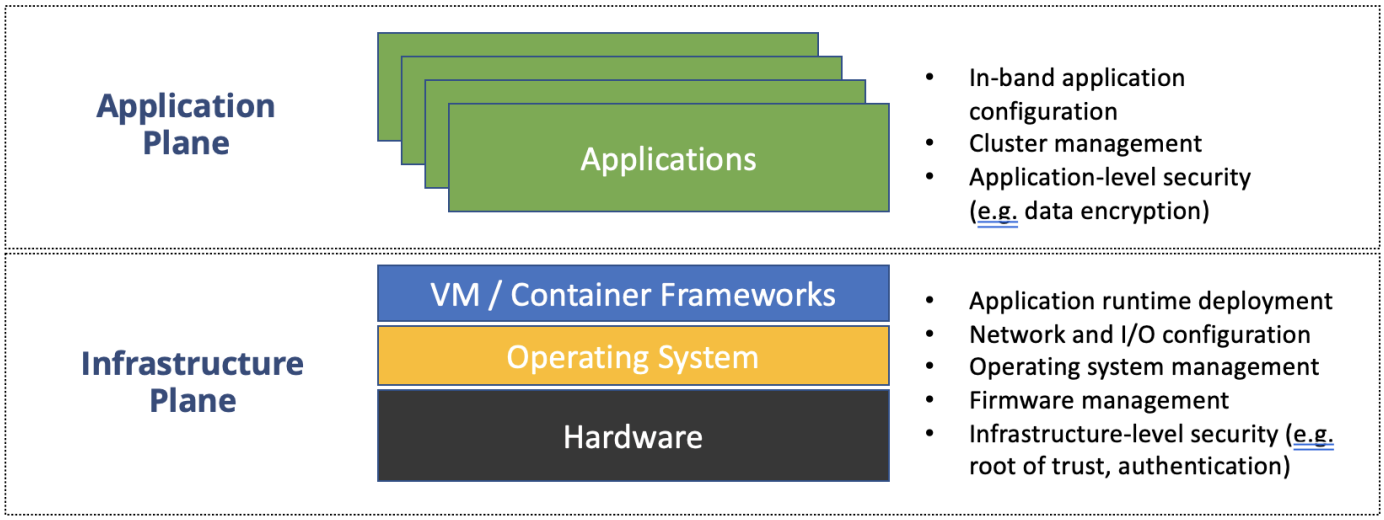
Further, the clusters are not about grouping resources as pools but more as contexts that are useful for the microservices that run in these resources, For example, two smartphones owned by two different users may run the same service (contact list for example) but while it is important to know that they are in the same context, it does not make sense to load balance between the two services. In another example, a LIDAR sensor in the front left corner of a car is inherently different from the LIDAR at the front right corner of the car, even if it is important to have them in the same context.

Another aspect when moving towards End User Device Edge is the difference between the macro and micro level of the resource. A car for example can be either considered as a mutable resource when treated as a single unit at the macro level, but can also be partially treated as a mini cloud on wheels at the micro level.

Infrastructure vs. Application Management

The edge is highly fragmented with a wide variety of hardware, software, networks, and skill sets and this complexity increases exponentially the closer you get to people and devices in the physical world. Over time, we have seen a dizzying array of Industrial and IoT platforms mixing various degrees of functions for data ingestion, normalization, analytics, management and security. However, rarely will one company do all of these functions well, plus this vertically-integrated model creates vendor lock-in. This has been the norm in the OT world for many years.

Meanwhile, a very common practice in the IT world is to separate out the infrastructure and application planes. This provides maximum flexibility for application deployment while retaining a consistent infrastructure foundation for management and security. That said, it is important to differentiate between infrastructure management and application management. Infrastructure management focuses on the underlying hardware including processing, networking and storage resources, the operating system, any virtualization and container technologies, and the deployment of any application runtimes on top of these resources. In contrast, application management involves the direct configuration of application runtimes. A key delineator is that infrastructure management is performed out-of-band of applications and data, whereas application management is performed in-band with applications.



The Infrastructure and Application Planes

A key goal for projects within LF Edge is to maintain this separation between these two planes, in addition to architecting modularity into each plane for maximum flexibility. Developers are then able to pick and choose their preference of OSS and proprietary ingredients to integrate into differentiated commercial offerings within the broader edge ecosystem.

Of note is that when dealing with End User Devices or Constrained Devices, it is important to understand that the difference between Infrastructure and Application management may not be possible due to multiple factors like the restriction of the operating system and memory constraints. For example, on a sensor where only a microcontroller is available, the only way to include this device as part of the edge is to have a library that makes it look like a discoverable service.

We will see increasing coordination between tool sets for deploying and managing edge applications at each edge management paradigm over time, however due to the inherent tradeoffs across the four core management paradigms it is unlikely for there to be a single, universal engine that is capable of addressing all use cases across the continuum. So, while a common goal is to have a “single pane of glass” that streamlines workflows for admins and developers, it will be most likely that edge solutions will rely on an “orchestrator of orchestrators” that aggregates underlying management infrastructure tailored for the needs of each edge paradigm.

Project Contributions for Edge Management and Orchestration

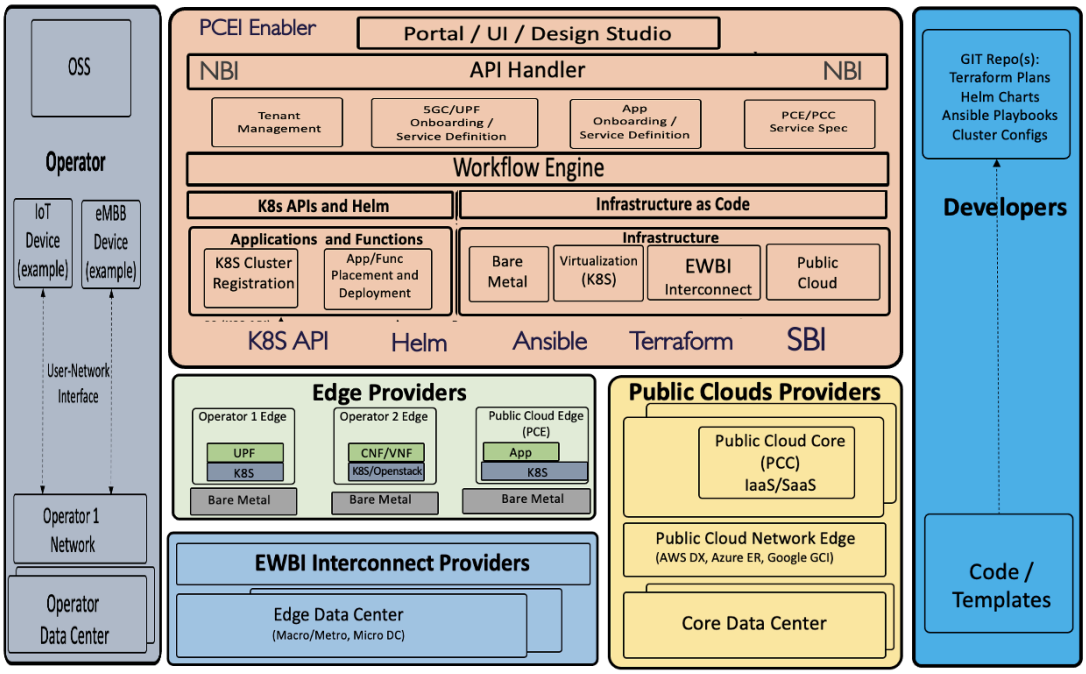
The following is an overview of LF Edge project efforts in the functional area of edge manageability and orchestration. While some projects focus exclusively on infrastructure or application management, others integrate tools in both planes. A given project’s focus on the infrastructure versus application plane is roughly indicated by logo placement in the Y-axis in Figure 2.

Akraino

Currently, there are over 20 Akraino Blueprints that are tested and validated in real hardware labs supported by users and community members. Akraino’s blueprint approach is unique in that it focuses on the overall solution stack, as such projects tend to address both the infrastructure and application planes.

Akraino blueprints solve unique problems for different and ever-evolving manifestations of the edge spanning the User Edge to the Service Provider Edge, reflecting the reality that there is not a “one size fits all” approach to edge computing. As such, Akraino blueprints provide developers with a collection of solutions that do not rely on a single and common stack for all functional areas of the edge addressed in this paper. Rather, the select blueprints are used as representative models for a given problem space. It is however important to note that Akraino’s blueprint diversity can be used to construct strategic “super blueprints” that can address many critical aspects of the edge, from infrastructure to interconnection, to applications.

An example of a specific Akraino blueprint that focuses on a blend of infrastructure and application management is Public Cloud Edge Interface (PCEI). PCEI enables infrastructure orchestration and cloud native application deployment across public clouds (core and edge), edge clouds, interconnection providers and network operators. The notable innovations in PCEI are the integration of Terraform as a microservice to enable DevOps driven Infrastructure-as-Code provisioning of edge cloud resources (bare metal servers, operating systems, networking) public cloud IaaS/SaaS resources, private and public interconnection between edge cloud and public cloud, integration of Ansible as a microservice to enable automation of configuration of infrastructure (e.g., servers) and deployment of Kubernetes and its critical components (e.g., CNIs) on the edge cloud, as well as the introduction of a workflow engine to manage the stages and parameter exchange for infrastructure orchestration and application deployment as part of a composable workflow. PCEI helps simplify the process of multi-domain infrastructure orchestration by enabling a uniform representation of diverse services, features, attributes, and APIs used in individual domains as resources and data in the code that can be written by developers and executed by the orchestrator, effectively making the infrastructure orchestration across multiple domains DevOps-driven.



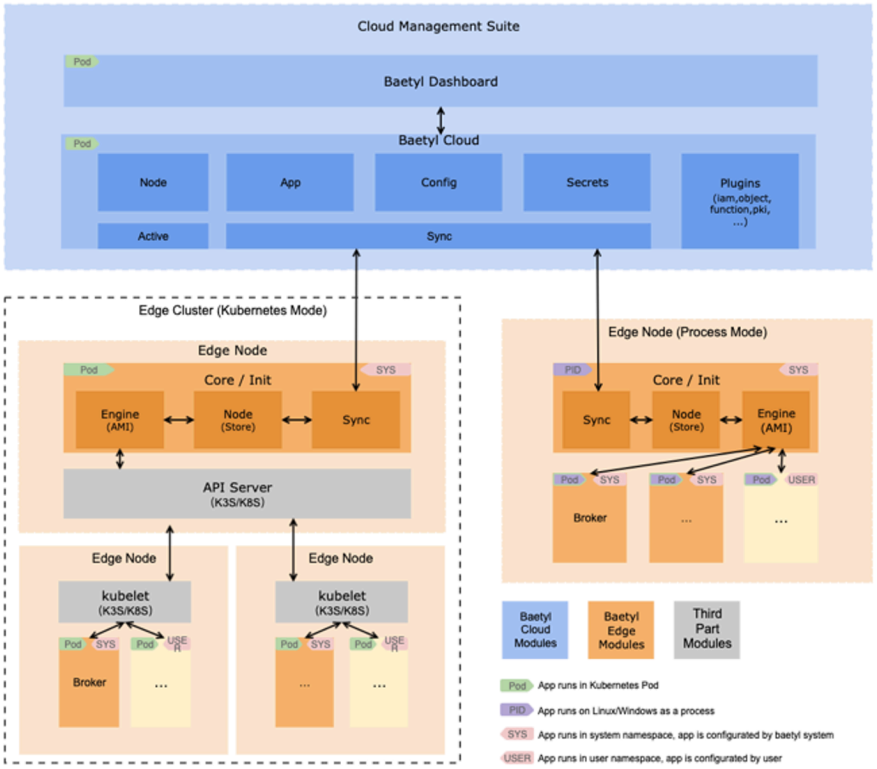
Akraino PCEI Blueprint Architecture

Baetyl

The Baetyl project aims to seamlessly scale applications and data from the centralized cloud to the Cloud Edge, enabling the convergence of edge and cloud computing. As with Akraino, Baetyl is focused on providing a full reference stack that includes elements of both infrastructure and application management.

Baetyl consists of baetyl-core, which runs on the edge, and baetyl-cloud, which runs in the cloud. The former continuously receives commands and data from the cloud, manipulates Kubernetes at the edge to run the corresponding application, and provides feedback to the cloud on the operational status of the application. The latter provides management APIs for users, sends commands and data to the edge, and receives and processes reporting information from the edge. With baetyl-core and baetyl-cloud, users can easily control hundreds of edge instances to perform different tasks such as data collection, endpoint control and video recognition.

Unlike many similar projects, Baetyl treats each edge instance as an independent Kubernetes cluster, rather than a working node in a larger cluster, and is not dependent on any particular modified Kubernetes distribution. In Baetyl, edge instances can be either single machines or highly available multi-machine clusters capable of load balancing and failover, which provides great flexibility for different application scenarios.



Baetyl Architecture

In Baetyl, applications are configured and managed by users in the cloud and then run at the edge. Users can define multiple different applications on the cloud, each of which can contain multiple containers, configuration files, data files, confidential information such as passwords and certificates, and policies for data storage. Applications are specified exactly which edge instances they need to be deployed to in the form of tag matching and are eventually translated into resources such as Pod/Service/ConfigMap for Kubernetes to run on edge devices.

At the edge, applications are also automatically injected with security certificates so that they can access many system services provided by Baetyl. One of these services is MQTT Broker, which will allow endpoint IoT devices to communicate with each other over the local network, and applications can use the MQTT protocol to collect information or control devices. Another service is Function Computing, which allows simple and fast processing of local data. Another service is remote connectivity, which allows secure uploading of local data to third-party services.

Baetyl can support a wide variety of different applications, for example, EdgeX Foundry can be configured in the cloud using Baetyl and then sent down to the edge to be measured and run.

EdgeX Foundry

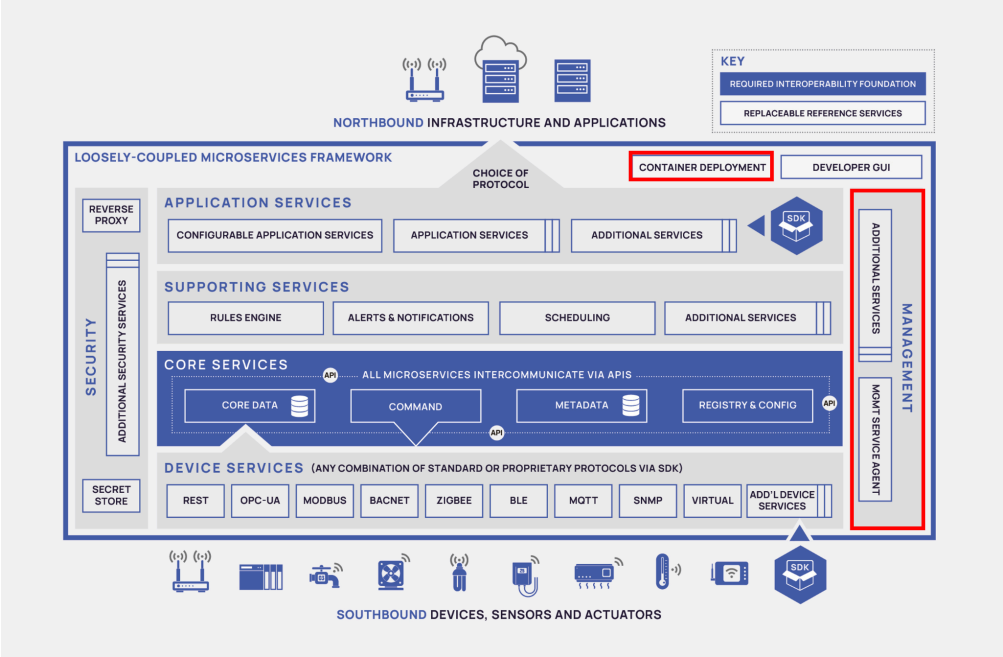
EdgeX Foundry is a vendor-neutral, loosely-coupled microservices framework that enables flexible, plug-and-play deployments leveraging a growing ecosystem of available third-party offerings, including proprietary innovations. At the heart of the project is an interoperability framework hosted within a full hardware- and OS-agnostic reference software platform. The reference platform helps enable the ecosystem of plug-and-play components that unifies the marketplace and accelerates the deployment of IoT solutions. EdgeX Foundry is an open platform for developers to build custom IoT solutions, either by feeding data into it from their own devices and sensors, or consuming and processing data coming out.

EdgeX Foundry focus is to exploit the benefits of edge compute by leveraging cloud-native principles, loosely-coupled microservices, platform-independence, and by enabling an architecture that meets specific needs of the IoT edge including different connectivity protocols, security and system management for widely distributed compute nodes and scaling down to constrained devices at the Distributed Cloud Edge. The sweet spot for EdgeX Foundry is enabling use cases where local decisions are at/or near real time and when automation and actuation is supported by multiple sources of data. EdgeX addresses critical interoperability challenges for edge nodes and data normalization in a distributed edge computing architecture.

While EdgeX adopters can deploy and orchestrate the services in a native environment of their choosing, the project also provides a set of Docker containers and Ubuntu Snaps for added convenience. Additionally, the project provides Docker Compose files and Helm Chart examples to help facilitate orchestration and deployment in containerized and other cloud-native environments. As an application framework, EdgeX Foundry management tools focus primarily in the application plane through APIs that expose in-band configuration of an EdgeX deployment but telemetry on infrastructure utilization is also made available.

In a large solution deployment, there could be many instances of EdgeX each managing and controlling a subset of the “things” in the overall deployment. In this case, a centralized management system will manage the fleet of edge systems and resources of the overall deployment. The EdgeX Foundry system management capability helps facilitate a larger edge management solution. When a management system wants to start or stop the entire deployment, EdgeX Foundry system management capability is there to receive the command and start or stop the EdgeX Foundry platform and associated infrastructure of the EdgeX Foundry instance that it is aware of.

Likewise, when the centralized edge management system needs service metrics or configuration from EdgeX Foundry, it can call on the EdgeX Foundry system management services to provide the information it needs (thereby avoiding communications with each individual service).



EdgeX Foundry System Management Services

There are two services that provide the EdgeX Foundry system management capability.

- System Management Agent: the microservice that other EdgeX systems or services communicate with and make their management request (to start/stop/restart, get the configuration, get the status/health, or get metrics of the service.
- System Management Executor: the executable that performs the start, stop and restart of the EdgeX services as well as gathering metrics from these services.

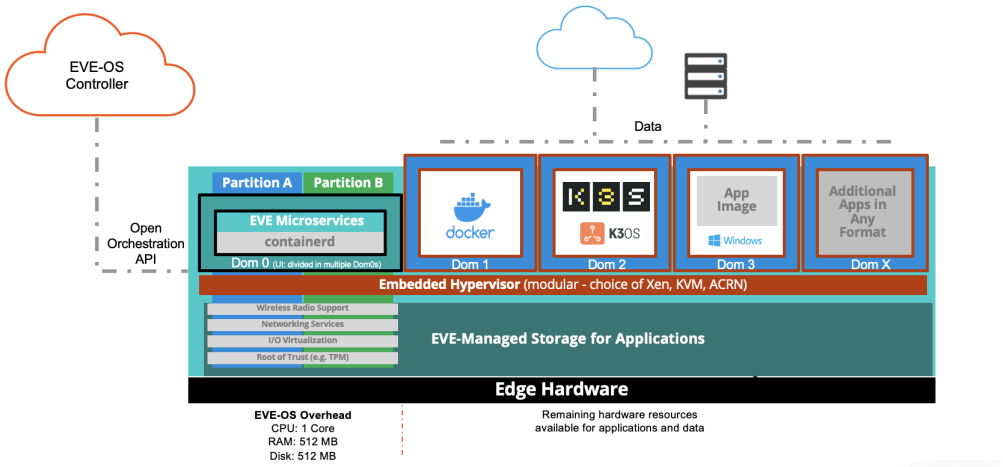
eKuiper

As an edge stream processing application, eKuiper can be deployed as a microservice application that is agnostic to choice of infrastructure and application management. EdgeX Foundry uses eKuiper as its reference rule engine.

EVE

Project EVE is building EVE-OS which is a universal, bare metal operating system optimized for Distributed Cloud Edge computing, blending cloud-native principles from the data center with the specific capabilities in areas such as zero trust security and zero touch provisioning required for deployments in the field. EVE-OS is focused on the infrastructure plane and is architected to simplify how developers deploy, manage and secure choice of both edge hardware and applications at the Distributed Cloud Edge. It extends data center principles as far into the edge as possible until hardware constraints mandate embedded software that is tailored to the capability of the device. Based on Linux and including a modular Type 1 hypervisor (e.g. KVM, Xen), EVE-OS has a footprint of just 512MB of memory and disk and is designed to scale from a single box such as a gateway, hub or router (immediately upstream of the Constrained Device Edge) to clusters of servers at the fringes of the traditional data center at the Metro and Regional Data Center Edge above.

EVE-OS mimics the public cloud experience for developers in the sense that it abstracts away the complexity of the hardware below by virtualizing all resources (e.g. processing, memory, storage, networking, I/O) and presenting these resources to applications deployed in any combination of virtual machines (with choice of guest OS spanning Linux to Windows), containers and clusters. This provides developers with maximum flexibility as they build solutions with myriad different components.



EVE-OS is designed with the assumption that distributed edge nodes are physically-accessible on untrusted networks. As such, it always initiates communication to its central orchestrator so this communication can traverse any type of network proxy (e.g., man in the middle, inspect/intercept, HTTP, HTTPS, SOCKS) on segmented IT and OT networks.

EVE-OS is also designed to expect that Distributed Cloud Edge nodes under management will periodically lose connectivity to their central controller. Any node that loses connection to its controller will continue to run in its current operating state until that connection is restored. The operating system leverages an eventual consistency model in which the desired operating state is set in the controller and whenever an edge node is able to connect it downloads any delta in software configuration and works to update the system in a separate OS partition. If successful, it switches over to the new software image. If not, it continues to run as it was previously.. Unlike agent-based management solutions, EVE-OS is a bare metal solution with tight coupling to the hardware so it is impossible to “brick” a device in the field during updates. This is critical to prevent unnecessary truck roles to the field.

Once application runtimes are deployed and secured by EVE-OS, admins would leverage the in-band management tools for these applications. Examples include the web consoles for LF Edge application frameworks such as EdgeX Foundry and Fledge, Azure IoT for Azure IoT Edge, AWS IoT for Greengrass and SUSE Rancher for K3S.

For security purposes, EVE-OS does not allow users to directly SSH into an edge node using a username/password. When explicitly allowed by the administrator through its central orchestrator, a SSH session is enabled by using SSH keys exchanged between the controller and the edge node (which is authenticated by its crypto-based ID).

EVE-OS continually collects logs for use in debugging the OS itself along with any deployed application, even when connectivity to its centralized controller is lost. Policy for log retention and when to upload this data to the central controller is fully configurable in order to conserve WAN bandwidth as necessary.

FIDO Device Onboard (FDO)

FIDO Device Onboard (FDO, formerly Secure Device Onboard) is a device onboarding scheme from the FIDO Alliance, sometimes called "device provisioning". FDO has authors from Intel, Qualcomm, ARM, Amazon, Microsoft, and Google. This is an important industry step, where a single standard is agreed upon by multiple chip manufacturers and cloud providers. It was originally based on the Intel Secure Device Onboard (SDO) protocol. FDO 1.0 has functional equivalence to SDO, and FDO 1.1, released in April 2022, adds certain additional features that are not available in SDO.

FDO is an automatic onboarding mechanism, meaning that it is invoked autonomously and performs only limited, specific, interactions with its environment to completely onboard the device to its intended IoT platform. It leverages IETF encoding, attestation and encryption standards: CBOR encoding, COSE encryption, Entity Attestation Token (EAT). The adoption of IETF standards allows FDO to adopt new techniques as they are adopted by the underlying IETF standards. It uses cryptographic identification for all device-initiated operations, so it fits nicely into a Zero Trust network.

A unique feature of FDO is that the device owner can select the IoT platform at a late stage in the device life cycle. The device owner may also create or choose the secrets or configuration data at this late stage. This feature is called "late binding". Late binding allows for a more efficient supply chain, where a single device meets the need for a single function, even though the device must interface into disparate management environments. FDO permits the device to adapt to the customer management environment during onboarding, by downloading data, scripting, and using software (as needed).

FDO also works in Internet, Intranet (corporate networks), and closed networks, so that it allows a given device SKU to satisfy the widest possible set of environments.

In FDO, when a device is first "unboxed" and installed, the operating system invokes FDO before invoking any other service. The FDO protocol allows the device to identify and connect to a prospective IoT platform over a TCP/IP network. FDO implements its own attestation and protocol security, so that it can run directly on top of TCP. FDO can also run under TLS, so that it is compatible with certificate-based cloud security.

Due to late binding, a new device running FDO does not yet know the prospective IoT platform to which it must connect. For this reason, the IoT platform shares its network address with a “Rendezvous Server.” The device connects to one or more Rendezvous Servers until it determines how to connect to the prospective IoT platform. Then the device connects to the IoT platform directly for attestation and onboarding.

The device is configured with instructions to query Rendezvous Servers. These instructions allow the device to query local network Rendezvous Servers before querying Internet-based Rendezvous Servers. In this way, the devices' determination of the IoT platform can occur within a closed network.

FDO is designed such that the device initiates connections to the Rendezvous Server and to the prospective IoT platform, and not in the reverse. This is a common industry practice for devices connected over the Internet.

LF Edge Implementation of FDO

The FDO project within LF Edge implements a full suite of FDO protocols including:

All-in-one testing environment for learning about the protocol

- FDO Manufacturing tools that create a manufacturing workstation to initialize FDO-based devices
- FDO Owner Server to convert an IoT platform to support onboarding of FDO-based devices
- FDO Device implementations, portable C code to implement the FDO protocols. This code must be integrated into the device base software, operating system and networking code. Devices with Trusted Platform Module (TPM) can use TPM key storage for FDO. Devices with custom key storage can be accommodated through modification of the source code. In addition to the C implementation, a Java-based device is provided for testing.
- FDO supply chain tools for manipulating and extending the Ownership Voucher

Since FDO is heavily based on encryption, the server-based tools are implemented in Java, a language where cryptography is part of the basic library interface. Device-based tools are implemented in C for compatibility with the widest possible set of devices.

FDO 1.1 Release

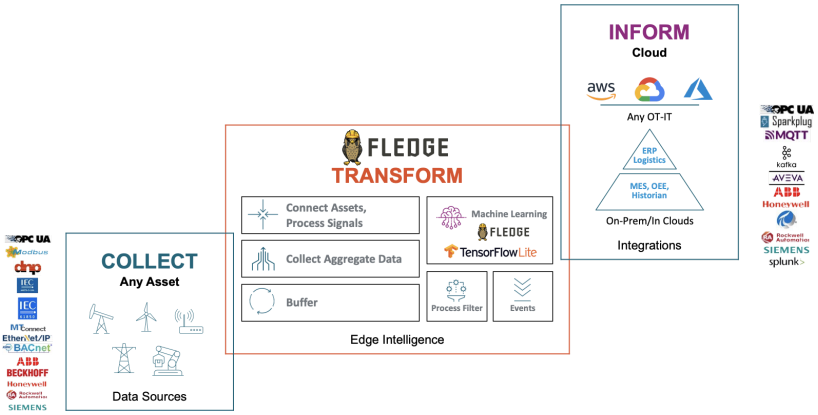
FDO 1.1 was released in April 2022. This version fixes issues that were found when two independent FDO implementations were first tested against each other. FDO 1.1 based interoperation has already been proved between LF-Edge’s and RedHat’s implementation to verify the correctness of the changes in FDO 1.1. These changes make FDO 1.1 ready for scaling.

In addition, based on the inputs from the user community, FDO 1.1 provides the following functionality improvements over FDO 1.0:

- Option to authenticate supply chain partners within the Ownership Voucher using certificate trust (“X5CHAIN” mechanism)
- Option for supply chain partners to embed data, such as a token, into the Ownership Voucher, which the device can verify during onboarding. This can be used to allow the supply chain partner to provide its own onboarding software and data to the onboarding device, rather than requiring the partner to power on the device and install the software (“OVEExtra” mechanism)
- Rendezvous Server forwarding added to the specification to make it possible for Rendezvous Server federation in the future.

Fledge

Project Fledge is an open community-driven IIoT platform focused on the data, data pipeline and application layers unique to industrial use cases. Fledge's pluggable microservice architecture is designed to collect and aggregate data from any machine, sensor or protocol, filter/transform data ingress and egress, process data on the edge and tightly integrate data with any destination service or system (clouds, data science/ML systems, OEE, MES, historian, ERP, logistics, etc.). Supporting most data types (time-series, array, radiometric, vibration, image) Fledge is ideal for managing simple to complex data pipelines and operating machine edge applications including ML inference.



Fledge Solution Architecture

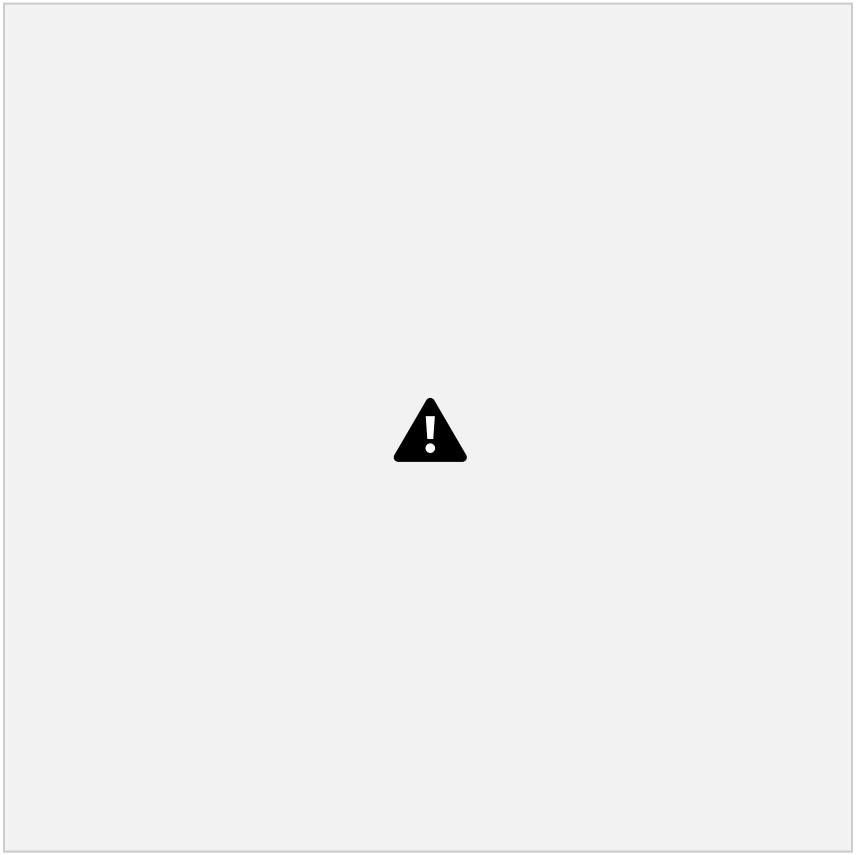
The users of OT data systems are diverse. They include the no-code, low-code and source code-capable. To address this diversity Fledge has an extensible UI, a concept called "plugins" and an extensive open API. Plugins are used for adding protocols, filters, processes, events and notifications, ML operations, control logic and integrations. From a developer perspective the modularity makes for rapid agile development simplifies testing and unifies the community. From a user perspective the ability to reuse, configure and deploy intelligent pipeline logic can be done without writing code from the Fledge UI. Configurations can be done from the UI or from the RESTful API. Logs and configurations are then stored in Fledge.

Critical to the project is a modern microservice-based architecture supporting deployments on raw hardware and within virtual machines or containers. Packages are available for any Linux OS and ARM, Intel, nVidia or Google CPU/GPU systems. Using a Restful API to configure, update and monitor Fledge also allows for maximum management flexibility.

In terms of infrastructure management, the Fledge project supports the EVE, Open Horizon and Akraino projects.

Home Edge

The Home Edge Project, seeded by code from Samsung Electronics, concentrates on driving and enabling a robust, reliable, and intelligent home edge computing open source framework, platform and ecosystem running on a variety of devices at daily home lives. To accelerate the deployment of the edge computing services ecosystem successfully, the Home Edge Project will provide users with an interoperable, flexible, and scalable edge computing services platform with a set of APIs that can also run with libraries and runtimes. Home Edge has focus in both the application and infrastructure planes and the project collaborates with various other LF Edge communities like EdgeX Foundry for IoT interoperability.



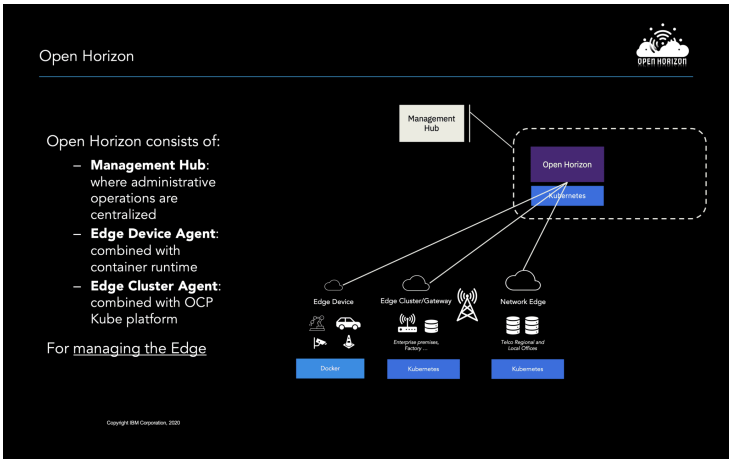
Home Edge is made up of multiple modules each for specific functionality. The Edge Orchestration Module handles Edge (device) Discovery, Service Offloading (load balancing between devices); Edge Setup, and Service Management and Monitoring. The Data Storage Module provides persistent storage (Core Data) and Metadata to identify the node. The DS Module also consists of the I/O Agent that, via APIs, allows for the accessing of the data. The Cloud Synchronization Module provides MQTT based data transfer to the cloud. The MQTT client acts as an agent to send and receive data from the cloud interface.

The Home Edge project has been exclusively targeting the home environment with many smart devices. Home Edge currently leverages REST (IP) based devices connected to the same network. Typically a user scenario would be when a third party application wants a service which is not present in the same device. The application can get the details of the devices in the same network where the service is available and orchestrate.

Service orchestration, data storage/retrieval, Cloud synchronization based use cases can be developed using the Home Edge. Docker based container releases are being done in the Docker hub frequently for ease of use.

Open Horizon

The Open Horizon project has created a solution that allows a single administrator to deploy and manage applications on a fleet of up to 40,000 edge computing nodes (bare Linux hosts or Kubernetes clusters). These nodes, through the Open Horizon agent software, autonomously manage the service software lifecycle of containerized workloads and related machine learning assets on the device or cluster, even with unreliable network connectivity or frequent disconnections..



The agent component takes up less than 30MB memory and is designed to run on a device or in a Kubernetes cluster with modest hardware requirements of 512MB RAM and 20GB of storage (including OS), and. Edge nodes should be running a Linux distribution or macOS. The agent currently supports armhf, arm64, risc-v, ppc64le, and x86 micro-architectures, several of which were contributed by the community over the last year.

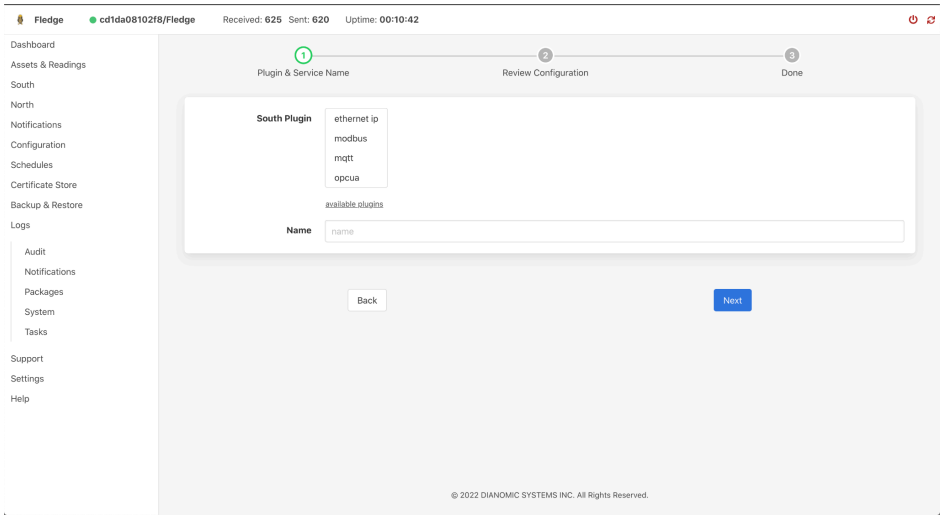
The Open Horizon agent supports use cases that span all four edge management paradigms: Metro and Regional Data Center, Distributed Edge Cloud, End User Device, and Constrained Device edges. The Management Hub components (e.g. control plane) can be located at the edge or in the cloud, or even offered as-a-service.

The key distinctives of Open Horizon include:

- Inversion of control - autonomous agents do all of the work, and initiate connections to the Management Hub
- Out-of-band update of analytics - ML models do not need to be containerized, and can be delivered separately from the applications that utilize them
- Perfect forward secrecy - each message sent, from agent to hub or vice versa, is encrypted with a new key

Open Horizon implements run-time dependency management for deployed services, and has separate application strategies for stateful vs stateless services. The project is also working individually with both Samsung and mimik Technology to extend management of edge computing services to non-traditional hosts, including mobile devices, robots, and vehicles.

The project works closely with other LF Edge projects in several ways. FIDODevice Onboard (FDO) is embedded into Open Horizon to enable zero-touch onboarding of the Open Horizon agent. EdgeX Foundry and Fledge are delivered by Open Horizon (see the ORRA project and SmartAg foundation for examples). And other projects have plans to either embed Open Horizon within their solution or integrate with the APIs.



Edge Security

A common misconception is that open source code is less secure than proprietary software, however this simply isn't the case. It is rare for any single company to have the same breadth of security knowledge as an entire OSS community and the transparency of open source collaboration ensures that more eyes are on the code to ensure robust design and to fix any potential vulnerabilities.

In addition to focusing on security within individual projects, the Linux Foundation has numerous efforts underway to drive Software Bill of Materials (SBOM) and overall secure software supply chain. The LF and the Open Source Security Foundation (OpenSSF) also recently joined a summit at the White House to talk through related issues.

Unique Security Challenges at the Edge

Over time, software-defined edge computing is only expected to become more sophisticated and we will begin processing more and more critical information in distributed locations. Many edge computing systems host their own web servers for remote maintenance and logins, making them a prime target as attack surfaces, especially for bad actors who could input or extract data and disrupt an entire ecosystem from a single unsecured system. Users need solutions to deliver new applications to the edge that drive efficient business outcomes while also maintaining an appropriate security posture.

Not all edge locations are created equally when it comes to security. Practices for securing deployments at the Service Provider Edge (e.g. the Metro and Regional Cloud Edge management paradigm) tend to be quite similar to traditional data centers. Meanwhile, nodes at the User Edge (e.g. spanning the Distributed Cloud Edge to the Constrained Device Edge management paradigms), are deployed outside of traditional data centers in locations such as the factory floor, retail stores, inside wind turbines, on trucks, or within rooftop HVAC systems, to name a few. These deployments require careful attention to unique security challenges. .

The following are some key areas that make securing edge solutions unique.

Scale
Part of the value of edge computing and IoT stems from having numerous devices connected in order to understand the holistic picture of your operations. Over time, we will see distributed edge deployments scale to the trillions, which is numerous orders of magnitude larger than the volume of deployments in centralized locations. This translates into an unwieldy number of distributed edge assets that an organization must secure and manage. Solutions oriented towards securing and managing data center infrastructure typically aren't set up for this kind of scale.

Lack of Physical and Network Perimeters
Another key challenge for securing distributed edge computing solutions is that there are often no physical or network perimeter. As such, a robust zero trust security model is essential and developers must assume that someone can walk up to an edge node and try to start hacking on itl. It is also very common to have to rely on a backhaul network and parameters (such as NATs and proxies) that are owned or managed by someone else when not practical to create your own network (e.g., cellular backhaul). In general, distrubted edge solutions should not rely on having an owned, trusted network or firewall to protect them.

Diverse Technology and Skill Sets
The edge is inherently heterogeneous, comprising a variety of technologies including sensors, communication protocols, hardware types, operating systems, control systems, networks, and so forth. Skill sets spanning OT and IT (e.g., network and security admins, DevOps, production, quality and maintenance engineers, data scientists, etc.) are necessary to realize edge computing as a convergence of the physical and digital worlds. Security solutions for edge deployments outside of traditional data centers need to accommodate a wide variety of technologies and skill sets in order to be effective.

Varying Priorities
In the IT world, it is typically acceptable to immediately shut down access to the network to isolate an affected system in the event of a security breach. Meanwhile, the impact due to information loss (e.g., credit card data or IP) plays out over a long period of time. In contrast, in the OT world, a security compromise can lead to immediate loss of production and risk to safety, so any issues need to be addressed gracefully. Security practices for edge solutions that bridge OT and IT systems need to strike a balance between these different priorities.

Unattended Operation
Unlike devices at the End User Device Edge, distributed telemetry-centric edge nodes typically do not have a user directly associated with them and operate unattended on a daily basis. This makes security especially important. Users can readily tell if their email or social account has been hacked but an unattended edge device that has been compromised could wreak havoc on business operations for a significant period of time before the issue is detected. The scale of a breach can also be many orders of magnitude greater due to the connected nature. In addition to zero trust architecture, it is especially important to consider building intelligence into unattended systems to continuously monitor and report on any anomalies.

Constrained Devices and Legacy Systems
Many IoT sensors and devices are too constrained resource-wise to employ security measures such as encryption. The same goes for legacy systems that were never intended to be connected to broader networks, let alone the internet. In

order to protect these devices, we must rely on more capable compute immediately upstream (e.g. at the Distributed Cloud Edge) to serve as the first line of defense, providing functions such as root of trust and encryption.

As we seek to reap the benefits of edge computing, we must realize the nuances it requires of our security approach. It can't be the same as what we're used to in data centers; instead, we must consider the edge's characteristics to bolster a distinct approach.

The LF Edge projects have a goal to harmonize the security approach into an overall reference architecture. The following are examples of current work of the individual projects.

Project Contributions for Edge Security

Akraino

Akraino Releases 4 and 5 made available in 2021 included K8s ready blueprints and multi-cloud deployments such as Public Cloud Edge Interface, AI Edge, 5G MEC System, Integrated Edge Cloud, Integrated Cloud Native blueprint families, Automotive, IoT, Metaverse Areas, and more. All released blueprints have passed a vulnerability scanning process implemented by Akraino's security subcommittee.

The Akraino security subcommittee is responsible for the security architecture, functional security requirements and implementation of recommendations for Akraino, encompassing both platform and network security. The subcommittee developed new automated security vulnerability identification features and increased efficiency in the blueprint security certification process. Akrino project prioritizes security testing as part of its CI/CD workflow, implementing BluVal and Lynis based testing. For 2022, the security sub-committee plans to further enhance its automated security scripts with predefined test scripts inline with the BlueVal framework.

The security subcommittee is constantly reviewing existing security requirements and updating them according to new security vulnerabilities found in the applications and libraries used by Akraino blueprints or vulnerabilities that are found in the host OS. The security team has developed several levels of security requirements based on the maturity of the project. These requirements are reviewed every 6 months and approved by TSC before being released to the Akraino community.

Akraino blueprints that are in the incubation or mature state use the latest TSC approved security requirements that were approved at least 6 months prior to the Akraino blueprints' release day.

In addition to Akraino blueprint security, the Akraino security subcommittee works on defining Akraino platform security. The platform security requirements are defined by a security questionnaire. The Akraino platform security questionnaire provides a set of questions about platform hardware, firmware, and host software security. These questions are used to assess the level of security implemented by the platform vendor and should be agnostic to the platform architecture. Akraino blueprint owners may add the questionnaire to the blueprint specifications as an additional security requirement for platforms that will execute this blueprint.

In addition, Akraino has enhanced its API map, integrated upstream components, explored downstream labs, and approved new incubation blueprints including buffer at the edge, smart data transition for CPS, and CPS robotics. Two blueprints entered maturity stage in 2021: Connected Vehicle Blueprint and IEC Type 4: AR/VR oriented Edge Stack for Integrated Edge Cloud (IEC) Blueprint Family.

Alvarium

Project Alvarium is building a framework and SDK for system-level trust fabrics spanning silicon to cloud that deliver data from devices to applications with measurable confidence. A trust fabric is a system-level approach by layered various trust insertion technologies spanning silicon-based root of trust, authentication, trusted operating systems and application frameworks, confidential computing, immutable storage, distributed ledger and so forth, bound together by the Alvarium framework.

Alvarium aims to provide an additional level of security to edge stacks along with a mechanism to protect privacy and IP based on policies set by data owners. By enabling data to traverse heterogeneous networks with measurable confidence, trust fabrics will enable an entire new era of business models and customer experiences driven by interconnected ecosystems. They will also help maintain privacy, identify fake data (e.g. AI-generated deepfakes) and address increasing data compliance requirements (e.g. GDPR). Finally, they will enable heterogenous stakeholders to consolidate workloads on common infrastructure. In effect, trust fabrics will help turn security from a cost center to a profit center.

Baetyl

Security is one of the most important parts of Baetyl, and the whole security system consists of three parts: connection security, service security, and device security.

Connection security means that the communication between the edge and the cloud is secure. Each edge instance with Baetyl installed is automatically assigned a unique certificate by the system. baetyl-core located at the edge uses this certificate to establish a TLS secure link with the cloud, called OTA channel, and all interactions with the cloud are reached through the OTA channel. At the same time, the baetyl-cloud in the cloud will not accept connections without the certificate, which ensures secure communication.

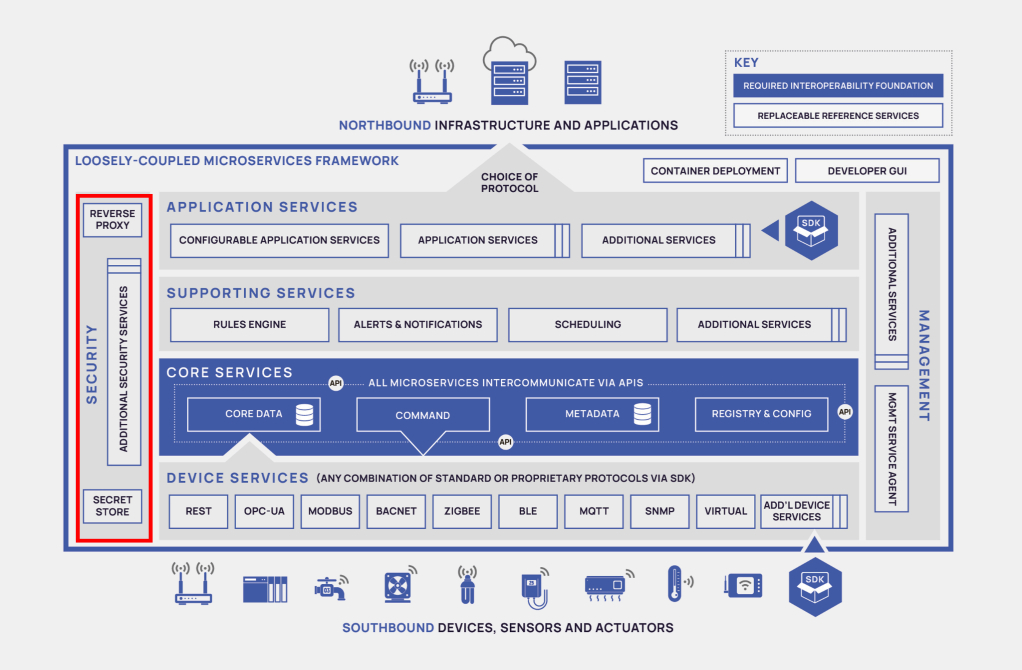
Service security means that all Baetyl system services are protected. Applications that want to access system services must use Baetyl-injected security certificates, which ensures that all communications are audited. For external connections, Baetyl's System Services also prefers to use TLS connections, but for compatibility purposes also supports normal TCP/UDP connections to support the ultra-lightweight sensors.

Device security means that devices with Baetyl installed are silent by default. No console is required to use Baetyl, no SSH service is required, and no keyboard or monitor needs to be connected. The only way to control Baetyl is through the OTA channel, which is always initiated by the edge to connect to the cloud, so edge devices do not require a public IP address and can be placed in the local network and behind a strict firewall.

EdgeX Foundry

The EdgeX application framework can be used to easily firewall and filter critical data and devices, processing locally on premise, and only expose selected data and applications to the cloud.

Security elements, both inside and outside of EdgeX Foundry, protect the data and control of devices, sensors, and other IoT objects managed by EdgeX Foundry. Based on the fact that EdgeX Foundry is a "vendor-neutral, open source software platform at the edge of the network", the EdgeX Foundry security features are also built on a foundation of open interfaces and pluggable, replaceable modules.



EdgeX Foundry Security Services

There are two major EdgeX Foundry security components. The first is a security store, which is used to provide a secure place to keep the EdgeX Foundry secrets. The second is an API gateway, which is used as a reverse proxy to restrict access to EdgeX Foundry APIs and platform controls. EdgeX Foundry security components provide the following capabilities:

- Secret creation, store and retrieve (password, cert, access key etc.)
- API gateway for other existing EdgeX Foundry microservice REST APIs
- User account creation with optional either OAuth2 or JWT authentication
- User account with arbitrary Access Control List groups (ACL)

eKuiper

eKuiper can be deployed vastly from resource constrained device to edge gateway, many of these environments have no internet connection for security reasons. eKuiper works well in such environments to keep the data local and safe. For edge/cloud communication scenarios, eKuiper can help users compute at the edge to filter critical data before sending outside.

eKuiper is managed using a REST API which can use JWT based authentication with RSA256 encryption. eKuiper can connect to external systems to consume or publish data. eKuiper supports the security connection to many of those systems. For example, eKuiper allows configuring the secure connection to the HTTP service, MQTT broker and EdgeX foundry.

EVE

EVE-OS features a state-of-the-art zero trust security architecture that assumes that Distributed Cloud Edge nodes are physically-accessible, in addition to not having a defined network perimeter. Features include:

- Hardware Root of Trust: EVE-OS leverages the cryptographic identity created in the factory or supply chain in the form of a private key generated in a hardware security model (e.g., TPM chip). This identity never leaves that chip and the root of trust is also used to store additional keys (e.g., for an application stack such as Azure IoT Edge). In turn, the public key is stored in its central orchestration console.

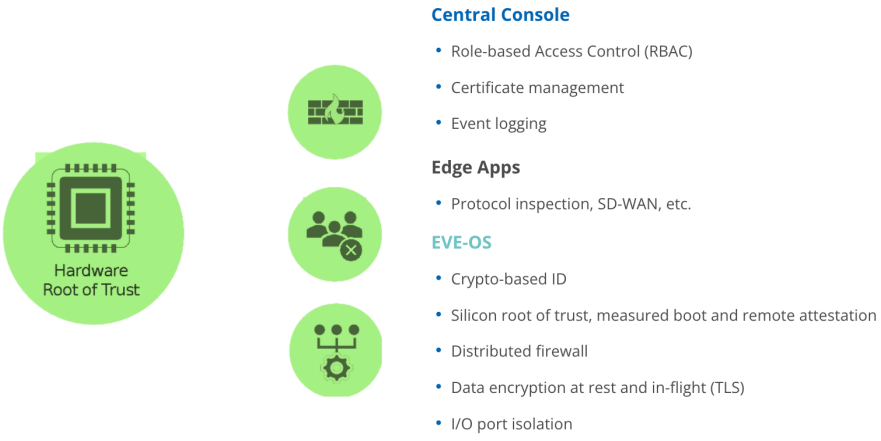
- No Usernames and Passwords: An edge compute node running EVE-OS leverages its silicon-based trust anchor (e.g., TPM) for identity and communicates directly with its remote console to verify itself. This eliminates having a username and password for each edge device in the field, instead all access is governed through role-based access control (RBAC) in a centralized console. Hackers with physical access to an edge computing node have no way of logging into the device locally. This crypto-based ID is critical for addressing attacks such as the Mirai and Verkada DDoS attacks that were a result of compromised user credentials.

- Measured Boot: EVE-OS performs measured boot of all the components including the operating system itself, along with the ability to enter a maintenance mode when the measured boot does not match the expected PCR measurements. It stores the cryptographic keys in a manner that ensures that even if a hacker accesses the storage disk, they should not be able to get them.

- Distributed Firewall: EVE-OS has granular, software-defined networking controls built in, enabling admins to govern traffic between applications, compute resources, and other network resources based on policy. The distributed firewall can be used to govern communication between applications on an edge node and on-prem and cloud systems, and detect any abnormal patterns in network traffic. These rules can be defined by IP address, TCP/UDP ports, hostnames, source IP subnets, and so forth.

- I/O Port Blocking: As a bare metal solution, EVE-OS also provides admins with the ability to remotely block unused I/O ports on edge devices such as USB, Ethernet and serial. Combined with no local login credentials, this provides an effective measure against insider attacks such as Stuxnet that leverage USB sticks to side-load malware.

- Centralized Management: All features within EVE-OS are exposed through an open, vendor-neutral API that is accessed remotely through the user's orchestration console of choice. Edge nodes block unsolicited inbound instruction, instead reaching out to their centralized management console at scheduled intervals and establishing a secure connection before implementing any updates.



EVE-OS Full Stack Security Approach - People, Process and Technology

All security features are implemented in a curated, layered fashion to establish defense in depth with considerations for people, process, and technology. Edge computing nodes running EVE-OS can be deployed at various points in a network for segmentation and these nodes can host additional security applications for protocol inspection, SD-WAN, etc.. In the

area of security and data trust, the EVE and Alvarium communities are collaborating to leverage EVE-OS as a trusted operating system for the Alvarium reference stack. While EVE-OS currently serves the authentication and ownership functions of FDO today, the community is evaluating adopting this technology as it becomes more ubiquitous because it provides the benefit of tracking device ownership throughout a heterogeneous supply chain.

Fledge

Fledge is managed using a RESTful API. Administrator and user rights are supported using HTTPs and certificates for encryption and authentication. Fledge accommodates the security methods and roots of trust defined by the source and destination of each pipeline. A common configuration of Fledge may include splitting a data pipeline to multiple destinations. In these cases Fledge will manage the unique credentials per connection.

Update, deletes, and rollbacks of Fledge applications and configurations support both a push or pull mechanism. In cases where Fledge is deployed deep behind the DMZ or on the other side of a data diode, scalable management can be securely performed. Fledge updates, bug fixes and security patches can be managed using private repos with authorized and signed code.

Home Edge

The Home Edge development team has paid special attention to improving security with the Coconut release. The security subcommittee enhanced the architecture based on a threat tree and incorporated new security functionality including two-way authentication (PKI), identification, authorization of access to resources (RBAC), Docker container verification, and Cloud Sync (MQTT security based on PKI). Introduction of automatic code scanning by vulnerability search systems (CodeQL, LGTM, LFXSecurity, Dependabot) made it possible to detect and fix all vulnerabilities found. Increasing the importance of testing and the inclusion of CI/CD in the project has increased the quality of the code and the search for errors that can lead to sensitive data leakage. As a result, the project received OpenSSF Best Practices "*Gold*" badge and the OpenSSF ScoreCard system displays a high level of security development.

The security subcommittee plans to continue to increase the number of security and quality analysis systems, code test coverage, monitor new CVE, CWE found by the security community, and also create security use-cases."

Open Horizon

Open Horizon currently implements zero-touch deployments through an embedded instance of FDO. By having edge agent software initiate all northbound communications, the project avoids the need to open ports in firewalls. The Management Hub (control plane) thus does not know the hostname or IP address of any of the edge nodes that it communicates with. All communication is encrypted and sent over TLS and only the sender and intended recipient can read messages. The code further employs Perfect Forward Secrecy to ensure that if any communication is somehow decrypted, no other messages will be vulnerable.

Open Horizon is working with AccuKnox on integrations of AppArmor and KubeArmor, and to ensure observability both within and outside of containerized applications.

Open Horizon is also strategizing with Project Alvarium on securing the software supply chain. This will necessarily include both consuming and utilizing SBOM-based information, and potentially allowing application deployment policies to make decisions based on related SBOM metadata.

Edge Connectivity

Connectivity needs vary widely across the edge continuum and require considerations at both the network transport and application levels. As detailed in the 2020 Sharpening the Edge paper, a key delineator between the Service Provider and User Edges is that the Service Provider Edge is almost always on a WAN relative to devices and end users whereas the User Edge is typically on a LAN relative to devices and end users. The exception for this is if a service provider deploys a Customer Premise Equipment (CPE) on-prem, or if a user owns their own data center upstream of the WAN.

When it comes to edge networking transport, a key goal is to get traffic into IP protocols as quickly as possible so traffic can freely traverse networks over transports spanning wired and wireless options. IP networking is the norm at the Service Provider Edge but from there is an increasing mix of non-IP transports at the User Edge. In the IoT and Industrial worlds, many systems communicate over legacy local area networks such as 4-20mA current loops, serial, CAN Bus and low-power wireless technologies such as Bluetooth and LoRa. IoT gateways serve the function of converting these transports into IP traffic.

In terms of application-level protocols, there is a wide array of choices to contend with when developing edge solutions. While there are tens that matter in the IT world (e.g. REST, MQTT), there are literally hundreds if not thousands in the OT/Industrial world when proprietary protocols are considered. Edge solutions often have to comprehend a blend of these application-level protocols and LF Edge projects like EdgeX Foundry and Fledge are focused on simplifying data flow in heterogeneous environments.

Due to the dynamic nature of edge deployments, it is critical that networks are able to adapt to current operational context in order to optimize performance and uptime. This can include dynamically switching between available connections. LF Edge projects like Akraino, Baetyl and EVE make specific considerations for enabling optimation at the transport level.

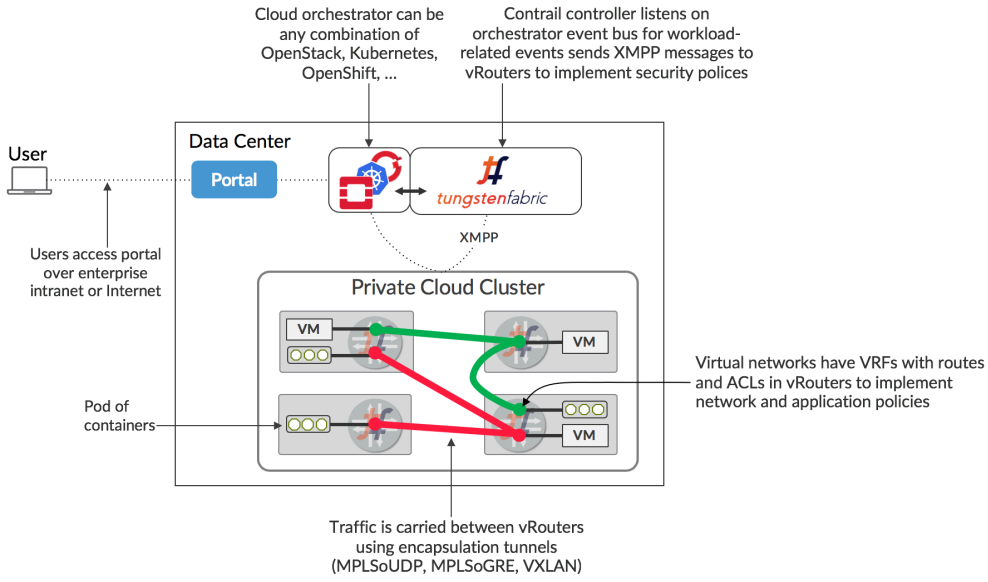
Project Contributions for Edge Connectivity

The LF Edge projects are addressing connectivity needs both at the transport level in terms of network virtualization and optimization and application level for protocol normalization to facilitate IoT interoperability. The following are examples of each project's focus in the area of connectivity.

Akraino

Akraino blueprints can provide an end to end Edge Stack to support Virtualized Network Elements (NFVI) per Open-RAN (O-RAN) requirements. The Akranio project advocated collaborating with O-RAN Alliance's specification workgroup 6 (six) responsible for cloud specifications, to align with and publish multiple blueprints to support various RAN use cases for Radio Edge cloud, including ORAN-Software Community's Near-RT RIC software, Network Cloud with RS-IOV or OVS-DPDK, Integrated Cloud Native, Kubernetes Native Infrastructure provider Access edge and more.

There are a number of blueprints that provide interesting examples for edge connectivity that can be applied in different parts of the edge ecosystem. As an example, theNetwork Cloud with Tungsten Fabric (TF) blueprint provides a fully distributed networking stack based on a microservices architecture, implementing a distributed networking framework for Edge computing. TF SDN Controller provides seamless and full integration between different types of workloads VNFs, CNFs and PNFs using a common networking stack integrated with different orchestration platforms like OpenStack and Kubernetes. The TF SDN Controller works as single entity running at the core, distributed core or edge sites, or public cloud (AWS, Azure, GCP or Equinix Metal) and fully integrated with OpenStack Neutron Plugin, Kubernetes CNI, for all types of Edge computing workloads. The solution provides the Tungsten Fabric Kernel vRouter, DPDK vRouter, and support for SR-IOV and SmartNIC.



Akraino Network Cloud with Tungsten Fabric (TF) Blueprint

EdgeX Foundry

A major difference between the edge and cloud is inherent heterogeneity and complexity at the edge. This is best illustrated in relation to connectivity and interoperability requirements, south and northbound:

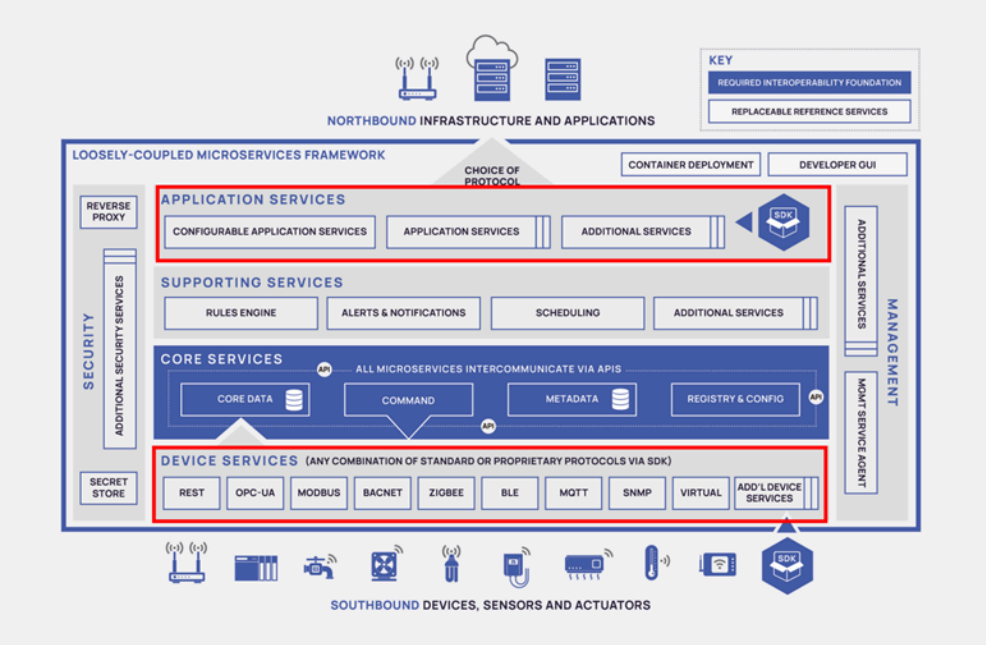
- Southbound: The edge is where the IT computer meets the OT 'thing' and there is a multitude of 'things' with which we will want to communicate, using a range of different 'connectivity' protocols at or close to real time. Many of these 'things' are legacy devices deployed with some old systems (brownfield).

  EdgeX Foundry provides reference implementations for key IoT protocols (e.g. MQTT, REST, Modbus, BACnet, SNMP, etc.) along with SDKs to allow users to add new ones. All complemented with connectors from a commercial ecosystem, making OT connectivity a configuration and not a programming task.

- Northbound: Just as EdgeX provides a variety of connectors to the world of OT 'things', it also provides flexibility of choice with regard to making edge data and operations accessible to the enterprise and cloud IT environments. EdgeX has services that prepare (transform, enrich, filter, etc.) and groom (format, compress, encrypt, etc.) edge data before being sent to an endpoint of choice. These services can publish data via HTTP, MQTT or nearly any IT related protocol to any enterprise or cloud (Amazon Web Services, Google IoT Core, Azure IoT etc.) endpoint. As with the south side, custom northbound services can be created using an SDK to connect EdgeX to any existing system or functionality. In fact, many organizations today use multi-cloud approaches to manage risk,

take advantage of technology advances, avoid obsolescence, obtain leverage over cloud price increases, and support organizational and supply-chain integration.

EdgeX Foundry is cloud agnostic and provides flexible connectivity to and from all different enterprise and IT environments.



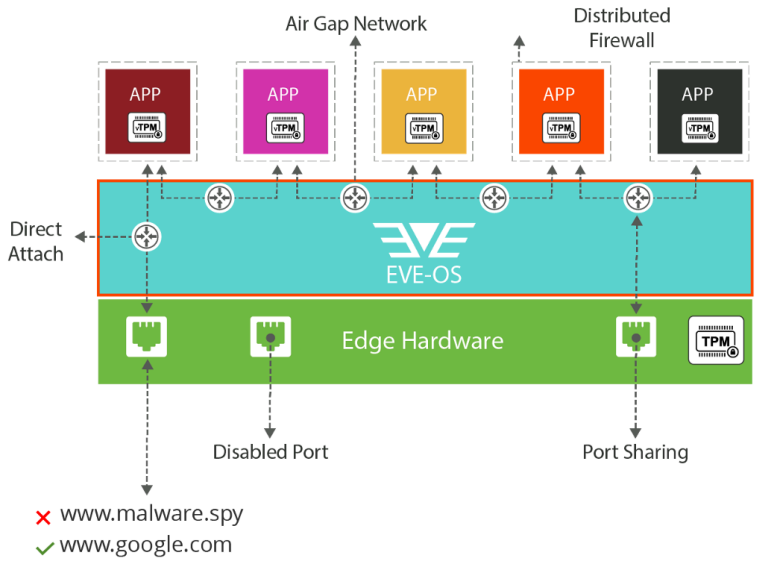
EdgeX Foundry Connectivity Services

EVE

EVE provides extensive networking functionality for Distributed Cloud Edge nodes by virtualizing all available I/O (e.g. Ethernet, WiFi, cellular, serial) and exposing these resources to applications based on policy. This results in highly efficient utilization of hardware resources including remote control of CPU, memory, networking and I/O.

The EVE API provides admins with the ability to granularly assign I/O to applications, both dedicated to specific applications and shared among two or more workloads. Connectivity to one or more backends (cloud or on premises) can be fully automated and policy can be set to prioritize backup interfaces for internet traffic to ensure continuity. For example, an admin can establish Ethernet 1 as primary, and Ethernet 2, WiFi and LTE as sequenced backup networks. Conversely, as part of its robust zero trust security model, EVE-OS provides the ability to disable unused I/O ports to prevent physical tampering.

Advanced networking capabilities include exposing DNS, DHCP, and NAT switching and routing functionality to the applications behind ethernet interfaces, including the ability to assign ethernet interfaces from an air-gapped network to the applications running on the edge hardware. Since it virtualizes physical I/O, EVE-OS can also pass a physical interface directly to an application that has the necessary I/O drivers. Admins can establish vLANs on the physical ethernet interfaces and distribute these vLANs to applications running on the device. Finally, EVE-OS supports network technologies like SR-IOV for performance optimization.



EVE-OS Distributed Firewall

The foundational networking features built into EVE-OS are complementary to LF Edge application frameworks like EdgeX Foundry and Fledge, along with any third-party edge application deployed in a virtual machine or container.

Fledge

Fledge is an IIoT platform. Its primary function is intelligent pipelines that move data from sources to destinations for OT. Along the way, data is transformed, optimized, buffered and analyzed, arriving in a state that can be used by multiple destinations. The pipeline is the steps involved in aggregating, organizing, filtering, processing and moving data.

Fledge is Architected for OT Data

The physical distribution of sensors, machines, processes and factories can not run in a cloud. The sources, types, formats and destinations of OT data are many. The applications and users of OT data are diverse. The physical environments from sources to destinations can be challenging. OT applications that require data pipelines must be able to operate from any machine edge to clouds. The beginning and ending of flows is usually determined by physical source/destination locations, latency, data volumes, application processing requirements and costs. Unlike typical cloud-based pipelines, intelligent OT-data pipelines enable edge application development and operation too.

OT automation requires standardizing data pipeline management from sensors to clouds (collection, transformation and integration). Fledge uses a common open API for scalability, manageability, supportability and security. Fledge supports most industrial protocols, types of data and data transformations. If a protocol does not exist it generally takes two weeks to support it. New data/registry mappings take less than a day. Automated schema translations 1-2 days. Edge MLOps work flows are automated.

Fledge Data Type Support

- Time Series
- Image
- Vibration and Acoustic
- Array
- Radiometric
- Transactional

Fledge Transformations

- Signal Processing
- Machine Learning
- Computer Vision
- Mappings
- Conversions
- Meta Data
- Any mathematical expression
- Time synchronizations
- Conditional forwarding

Data Integrations

- **Clouds**
    - AVEVA
    - AWS
    - Azure
    - Google
        - IoT Core
        - Pub/Sub
- **Systems Egress**
    - Graphite
    - HarperDB
    - InfluxDB
    - OSIsoft PI
        - OMF
        - PI WebAPI
        - OMF Hint (auto AF translator)
    - Splunk
    - Thingspeak (Matlab)
- **Protocols Egress**
    - HTTP/s
    - HTTP-c
    - IEC104
    - KAFKA
    - KAFKA-Python
    - MQTT
    - OPC-UA
    - REST
- **OT Protocols Ingress** (Over 100, see documentation)
    - From ABB

To Yokogawa

Open Horizon

A drawback and potential bottleneck for deploying applications on the edge is creating secure connections between an application and any external resources. Open Horizon is exploring ways to simplify that task through partnering with solutions that allow application-directed networking.

Open Horizon secures and simplifies the internal networking between dependent services in a deployed application by only connecting a service with each explicitly-defined individual depency. A service does not otherwise have a shared network connection with all other services in a deployed application.

Edge Analytics

In a perfect world we'd just run the bulk of analytics workloads in the cloud where compute is centralized and readily scalable, however the benefits of centralization must be balanced out with factors that drive decentralization. The explosion of devices and data is driving a need for more processing at the edge, with reasons including reducing latency and network bandwidth consumption and ensuring autonomy, security and privacy. Edge computing means that we're simply moving some aspects of the data aggregation and analytics out of centralized data centers, closer to where the data originates and where decisions are made in the physical world.

As illustrated in the LF Edge taxonomy, the "edge" is not one location, rather a continuum spanning billions of constrained devices in the field to thousands of regional data centers located just downstream of centralized cloud resources. Use cases that are latency-critical or require a high degree of security and privacy will always be driven proximal to the user or process in the field, for example deploying a vehicle's airbag from the cloud when milliseconds matter or stripping PII from interactions with consumers. The same goes for use cases that are inherently upload-intensive, such as computer vision (e.g. extracting information from streaming video) or analyzing high bandwidth vibration data in an industrial use case. Meanwhile latency-sensitive applications such as streaming video content, AR/VR and cloud gaming will typically take advantage of upstream edge tiers (e.g. offered by telcos and service providers) or the cloud because of the scale factor spanning many end users.

Enterprise robotics also relies heavily on edge analytics. Use cases in manufacturing, production, agriculture, and retail are emerging rapidly due to macro economic pressures, including cost of labor, manpower shortages, and legal/liability issues. In these use cases, analytics functionality is most important, followed by reduced SWaP (size, weight, and power consumption), employee safety, data privacy, and cloud independence – all of which are characteristic of edge computing. To achieve these objectives requires progress in key areas of edge analytics technology:

- Fusion of sensor touch and tactile data, combined with AI to allow robotic handling of objects of various shapes and friction coefficients, and in variable circumstances

- Computer vision. In addition to detecting and recognizing people, enterprise robots also must identify dangerous situations, for example leaning or unstable objects (such as a leaning pallet in a warehouse), incorrect lighting, slippery floors, foreign objects on a conveyor belt, etc.

- Speech recognition. First and foremost, enterprise robots need to recognize "immediate and urgent" voice commands in order to prioritize human safety; for example if someone shouts "Stop Now" the robot must stop - regardless of who is the speaker, level of background noise, or other circumstance. Second, enterprise robots need to accept verbal instructions, rather than programming interfaces (e.g. keyboard, app) inconvenient for rugged, wet, and fast-paced environments

- Data privacy. Enterprise operations do not trust public clouds with video and audio that may contain sensitive and/or proprietary information. Deep learning training must be handled on-premise or otherwise trusted manner

Today the edge component of AI typically involves deploying inferencing models local to the data source but even that will evolve over time to include more training and even federated learning at the edge. Where workloads are best deployed across the edge-to-cloud continuum is ultimately driven by a balance of performance, cost (e.g. bandwidth consumption, labor), security, privacy and autonomy. Increasingly, considerations for energy density and efficiency are also coming into play.

Deploying edge analytics introduces additional technical and logistical challenges due to increasing complexity of the hardware, software and required domain knowledge the closer you get to the physical world. As a result, to date many edge AI solutions have been lab experiments or limited field trials, not yet deployed and tested at scale. It's important to consider that many of the general considerations for deploying AI in the cloud carry over to the edge. For instance, results must be validated in the real world — just because a particular model works in a pilot environment doesn't guarantee that the success will be replicated when it's deployed in practice at scale.

In addition to dealing with real-world challenges with technology fragmentation and diverse skills sets in the field, developers need ways to accommodate model drift over time due to changing context (e.g. camera angle and lighting in the case of computer vision), retrain and update these models continuously, and manage and secure the underlying infrastructure. These underlying management tasks are especially difficult for widely distributed nodes compared to centralized data center resources. A key factor for success at the edge is having the right orchestration and security tools for each part of the edge continuum.

The LF Edge projects provide stakeholders from IT and OT administrators to developers and data scientists with robust remote orchestration tools to not only be able to initially deploy and manage edge infrastructure and AI models at scale in the field, but also continue to monitor and assess the overall health of the system. The projects comprehensively enable edge AI from the Metro and Regional Data Center Edge to the Distributed Edge Cloud and even the Constrained Device Edge with the addition of eKuiper in 2021. As an agent-based solution, Open Horizon is also exploring the enablement of AI at the End User Device Edge.

Finally, it's important to note that by design, the LF Edge community is primarily focused on infrastructure and application frameworks that facilitate edge analytics with better management, security and connectivity capabilities, not analytics themselves. The latter is seen as a key point of differentiation for developers and end users and also requires specific domain knowledge for a given vertical use case. The community does encourage end users to join their vertical working groups to bring their use cases and challenges so the community can best address these needs in the underlying foundation.

Project Contributions for Edge Analytics

Akraino

Akraino is enabling edge analytics through a number of the blueprints. One example is Fujitsu's Akraino blueprint "Robot Basic Architecture Based on SSES" that creates a framework for fusion of robot sensor data necessary for food preparation and production robotics use cases, including tactile, touch, surface friction, and more. The framework combines sensor data collection, machine and deep learning models for analysis, and feedback for mechanical robot control. A multi sensor module (MSM) has been prototyped for PoC and demo purposes.

Signalogic is contributing to the blueprint automated speech recognition (ASR) functionality. A 20,000 word real-time vocabulary is being implemented on a pico ITX Atom board (quad-core, 3.5" x 3.5", 10 W) suitable for Fujitsu's food prep and production use cases, as well as a range of robotics use cases in manufacturing, production, agriculture, and retail. The implementation includes robust audio noise processing to deal with background and robot mechanical noise.

Alvarium

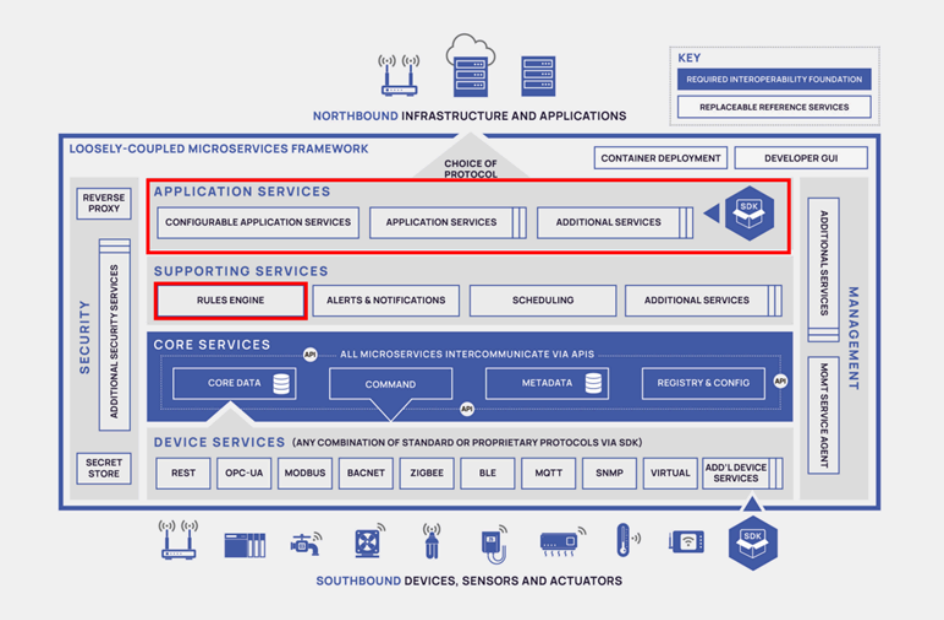
While not specifically focused on edge analytics, Alvarium aims to be a key enabler for driving value from edge data by introducing confidence in that data.

Baetyl supports AI inferencing models inferencing at the edge and specifies in the cloud whether the model can be accelerated using the neural network chip at the edge.

EdgeX Foundry

For a variety of reasons (latency, costs to ship and store data, etc.), more intelligence is moving closer to the edge where the data originates. EdgeX Foundry supports an open, plug and play approach to edge analytics. The EdgeX model is to get edge data to the adopters' choice of analytics package so that it can be used at the edge to act quickly. EdgeX uses and provides integration with eKuiper – an open-source rules engine package and fellow LF Edge project – as its default, reference implementation, analytics package. eKuiper is a lightweight package for IoT edge analytics and stream processing implemented in Golang, which can run on various resource constrained edge devices. Users can realize fast data processing on the edge and write rules in SQL.

More broadley, an analytics service could be some simple logic built into an application service, a rules engine package, or an agent of some artificial intelligence/machine learning system.

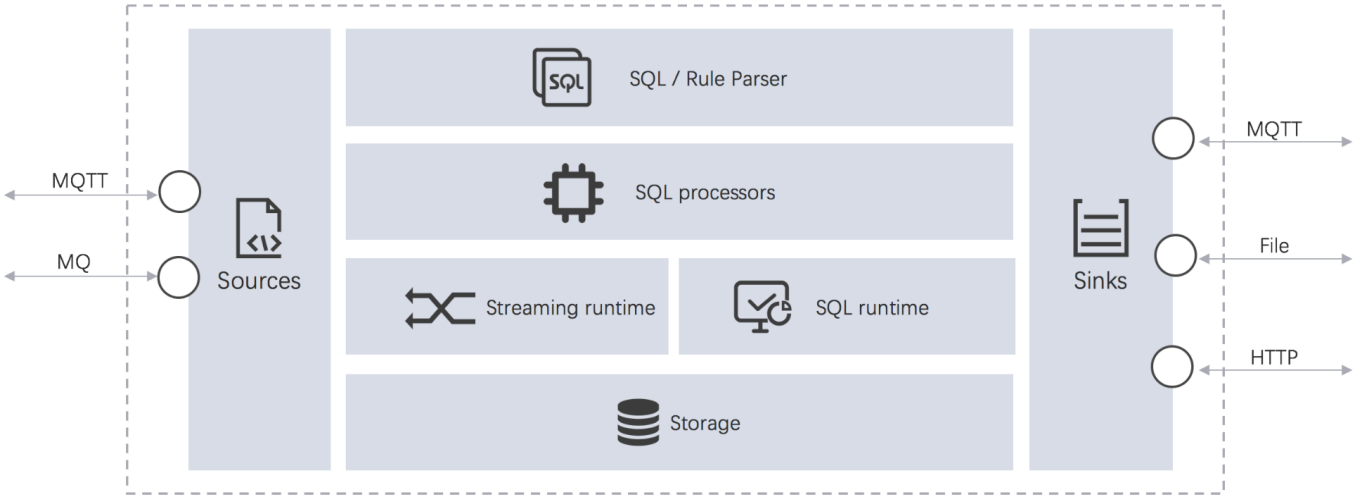


EdgeX Foundry provides an abstract message bus interface, and implements the ZeroMQ and MQTT protocols respectively. The message bus architecture allows sensor data to easily be streamed to **any** analytics package.

eKuiper

eKuiper is an IoT data analytics or stream processing engine running at resource constraint edge devices. It can be run at all kinds of resource constrained edge devices. One goal of eKuiper is to migrate the cloud streaming software

frameworks (such as Apache Spark, Apache Storm and Apache Flink) to the edge side. eKuiper references these cloud streaming frameworks, and also considered special requirements of edge analytics, and introduced rule engine, which is based on Source, SQL (business logic) and Sink, rule engine is used for developing streaming applications at edge side. It can be run at various IoT edge use scenarios, such as real-time processing of production line data in the IIoT; Gateway of Connected Vehicle analyze the data from data-bus in real time; Real-time analysis of urban facility data in smart city scenarios. eKuiper processing at the edge can reduce system response latency, save network bandwidth and storage costs, and improve system security.



eKuiper Architecture

eKuiper is lightweight and highly efficient, optimized for resource constraint devices with high throughput processing. It is cross-platform, which can be deployed cross CPU and OS support, including X86, ARM and PPC CPU arch; various Linux distributions, OpenWRT, MacOS and Docker. It can connect to different message brokers, databases and files to analyze the data produced by various sources and sink the analyzed result to any destinations. It is highly extensible so that it can be extended to connect to new or private data producers/consumers to integrate with any data intensive ecosystems.

The data analyzation logic is presented through SQL. It supports data extract, transform and filter through SQL syntax. It also supports streaming concepts like time windows, stream join through SQL syntax. It has more than 60 built-in functions, including mathematical, string, aggregate and hash, and more. Moreover, UDF is supported to extend the analytic ability. For example, it can extend UDF to work with machine learning algorithms and run against streaming data. Lastly, it is flexible to deploy the analytic applications. Text-based rules are used for business logic implementation and deployment through REST-API.

Thus, eKuiper is a lightweight yet powerful streaming engine for the edge.

EVE

EVE-OS is not in the data path, rather a highly secure operating environment to deploy any kind of application. EVE specifically addresses the needs for deploying edge AI outside of physically-secure data centers with focus on zero trust security and zero touch deployment capability to accommodate non-IT skill sets. A major benefit of concurrent support for both VMs and containers is that users can consolidate legacy apps (e.g. Windows-based SCADA, Historian, VMS, PoS) alongside new containerized AI models son edge hardware

By virtualizing all of the hardware resources, EVE-OS also provides a mechanism for developers to assign analytics workloads to specific resources, for example one application to a specific set of CPU cores and another to a GPU. The

EVE API also exposes health and utilization metrics or the hardware below and this data can be used to further optimize analytics workload performance.

Fledge

Most OT “machine” edge applications are focused on pipeline management, OEE, safety, maintenance and logistics services. Latency, data volume, reliability, security/policy and cost are the main drivers for operating these applications on the edge vs the cloud.

The concept behind Fledge’s filters is to create a set of small, useful pieces of functionality that can be inserted into the data flow from the south data ingress side to the north data egress side. By making these elements small and dedicated to a single task it increases the re-usability of the filters and greatly improves the chances when a new requirement is encountered that it can be satisfied by creating a filter pipeline from existing components or by augmenting existing components with the addition of any incremental processing required. The ultimate aim being to be able to create new applications within Fledge by merely configuring filters from the existing pool of available filters into a suitable pipeline without the need to write any new code.

Data processing is done via plugins that are known as filters in Fledge, therefore it is not possible to give a definitive list of all the different processing that can occur, the design intent is that it is expandable by the user. The general types of things that can be done are;

- ML inference
- Modify a value in a reading
- Modify asset or datapoint names
- Add a new calculated value
- Add metadata to an asset
- Compress data
- Conditionally forward data
- Write back to an asset (setpoint control)
- Create an event and make a notification
- Data conditioning

Open Horizon

Open Horizon features a unique component called Model Manager comprised of two parts: Cloud Sync Service running in the Management Hub (control plane) and Edge Sync Service embedded in the agent running on edge nodes. The component enables bi-directional synchronization of machine learning assets and related files separate from any containerized applications that may consume those analytics. This allows edge-based analytics services to have separate deployments of a single application and yet custom analytics per edge node. Likewise, separate applications could all consume identical analytics. This separation of concerns allows applications to be deployed at their own natural cadence while allowing analytics to be updated more frequently, and without application restarts.

State of the Edge

State of the Edge is a unique project within the LF Edge community in that it does not produce code, rather it provides an annual research report covering the latest important developments in edge computing

infrastructure. The content is free and shareable, aiming to crowdsource a common vocabulary for the broad and varied world of edge computing and pinpoint and describe the major trends.

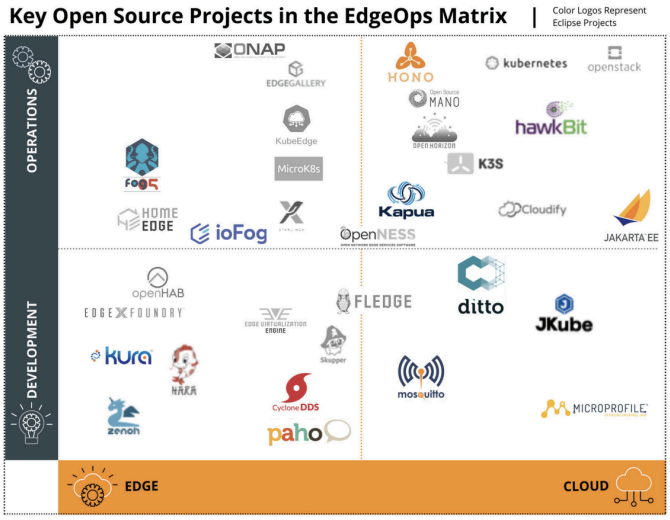
State of the Edge 2022, out in June 2022, addresses three aspects shaping the development of edge computing: connectivity, application infrastructure, and location. It takes a close look at the broadband access gap in the US, an issue that's core to the future of some of the most promising edge computing use cases; it examines the ins and outs of translating cloud native principles of application development and infrastructure management to deploying and running software at the edge; and explores new physical locations where compute infrastructure is being deployed to answer the need for ever more distributed platforms, including both on the ground and in Earth's orbit.

-

Industry Collaboration

An important highlight is that LF Edge makes a special point to collaborate with other industry forums, including other OSS consortia. Examples include Akraino's close collaboration with organizations spanning ETSI MEC to CNCF, SDO's implementation of the FIDO specification, and alliance with the Digital Twin Consortium (DTC) led by the EdgeX Foundry community.

In 2021, LF Edge and the Eclipse Foundation's IoT and Edge communities formed an alliance to further collaboration in the edge space. The Edge Native working group at the Eclipse Foundation aims to deliver a unified vision and platforms for the seamless development and operation of edge native applications suited for heterogeneous environments. Figure X is a graphic of various Eclipse projects focused on edge operations, along with rough placement of their relationship to LF Edge efforts.



Source: 2021 Eclipse Foundation EdgeOps Whitepaper

The LF Edge community is also increasingly working with vertical industry forums. An example is the collaboration between the Fledge and EVE communities and The OSDU Forum, part of the Open Group and focused on developing an open, standards-based foundation to accelerate innovation in the energy space. These communities are assisting in building a proof-of-concept for OSDU's edge computing reference architecture, with the goal of integrating more

open-source efforts over time. In another example, Fledge has also been collaborating with LF Energy and has created an energy-focused variant called Fledge Power. Collaborating with these vertical industry forums is a key focus for the LF Edge community.

Edge computing takes a village and participation in LF Edge provides access to collaborations within the broader edge computing landscape. Our collective goal is to enable the deployment and management of differentiated, interoperable edge and IoT solutions that drive new business outcomes and customer experiences while also ensuring security, privacy and safety. We encourage you to engage today to help us on this mission.You can learn more and get involved with any of the projects by visiting www.lfedge.org.

######

Version 1 content for reference only

Linux Foundation Edge: Taxonomy and Framework

Contents

1	Executive Summary	3
2	Introduction	3
2.1	Introducing the Edge Continuum	3
2.2	Extending Cloud Native Principles to the Edge	5
2.3	Considerations for the Service Provider Edge	6
2.3.1	Architectural Trends at the Service Provider Edge	7
2.3.2	Edge Application Deployment at the Service Provider Edge	8
2.3.3	Design Strategy for Backend Application Mobility Workloads at the Service Provider Edge	9
2.3.4	Design Strategy for Edge Application Mobility at the Service Provider Edge	9
2.3.5	Design Strategy for User Device Mobility at the Service Provider Edge	9

2.3.6 Identifying the Optimum Edge Location to Serve a User 10
2.4 Considerations for the User Edge 10
2.4.1 Securing and Managing Distributed Devices 11
2.4.2 Accommodating both Legacy and Modern Applications 12
2.4.3 Addressing Protocol Fragmentation in IoT Use Cases 12
2.4.4 Latency-Critical Applications 12
2.4.5 Separation of Concerns in IT and OT Environments 12
2.5 Edge Deployment Patterns 13
2.6 Trends for Edge AI 14
2.7 Edge Computing Use Cases 14
2.7.1 Industrial IoT (IIoT) 15
2.7.2 Computer Vision 15
2.7.3 Augmented Reality (AR) 16
2.7.4 Retail 16
2.7.5 Gaming 17
2.7.6 Assisted Driving 17
2.7.7 Summary of the Edge Continuum 18
3 LF Edge Project Portfolio 19
3.1 LF Edge Project Summaries 20
3.1.1 Stage 3: Impact Projects 20
3.1.2 Stage 2: Growth Projects 20
3.1.3 Stage 1: At Large Projects 21
3.2 Project Focus Across the Edge Continuum 21
3.3 For more Information on LFE Projects 22
4 Summary 22

1 Executive Summary

Companies in a wide range of vertical markets are aggressively exploring new commercial opportunities that are enabled by extending cloud computing to the edge of the network. The concept of edge computing promises exciting new revenue opportunities resulting from the delivery of new types of services to new types of customers, in both consumer and enterprise segments.

Intended for readers interested in both the technical and business aspects of edge computing, this white paper introduces a set of open-source software projects hosted by the Linux Foundation (LF) and its subsidiary organization LF Edge (LFE). It describes opportunities for companies to participate in and benefit from these projects, accelerating the development, deployment and monetization of edge compute applications.

The paper includes references to online resources associated with each project, providing developers with access to a wealth of technical information as well as the open-source software itself.

2 Introduction

Edge computing represents a new paradigm in which compute and storage are located at the edge of the network, as close as both necessary and feasible to the location where data is generated and consumed, and where actions are taken in the physical world. The optimal location of these compute resources is determined by the inherent tradeoffs between the benefits of centralization and decentralization.

This white paper introduces the key concepts of edge computing and highlights emerging use cases in telecom, industrial, enterprise and consumer markets.

The paper also provides details of eight open source edge projects hosted by The Linux Foundation (LF) and its subsidiary LF Edge (LFE) umbrella organization. The LF is a non-profit technology consortium founded in 2000 to standardize Linux, support its growth and promote its commercial adoption. The LF and its projects have more than 1,500 corporate members from over 40 countries. The LF also benefits from over 30,000 individual contributors supporting more than 200 open source projects.

Founded in 2019, the mission of LF Edge is to establish an open, interoperable framework for edge computing independent of hardware, silicon, cloud or operating system.

2.1 Introducing the Edge Continuum

As defined in the Linux Foundation's Open Glossary of Edge Computing, edge computing is the delivery of computing capabilities to the logical extremes of a network in order to improve the performance, security, operating cost and reliability of applications and services. By shortening the distance between devices and the cloud resources that serve them, and also reducing the number of network hops, edge computing mitigates the latency and bandwidth constraints of today's internet, ushering in new classes of applications. In practical terms, this means distributing new resources and software stacks along the path between today's centralized data centers and the increasingly large number of deployed nodes in the field, on both the service provider and user sides of the last mile network.

In essence, edge computing is distributed cloud computing, comprising multiple application components interconnected by a network. Many of today's applications are already distributed, such as (1) a smartphone application with a cloud backend, (2) a consumer device, such as thermostat or a voice control system that connects directly to the cloud, (3) a smartwatch or sensor connected to a smartphone and then to the cloud and, (4) an industrial IoT (IIoT) system connected to an edge gateway and then to an on-premise system and/or the cloud. In addition, many LTE and 5G network functions will be distributed to the edge, enabling new business models and use cases, including those for dedicated private

networks, fixed wireless access, SD-WAN and network slicing, thereby catering to the needs of many enterprises and vertical industries.

It helps to visualize edge computing through the continuum of physical infrastructure that comprises the internet, from centralized data centers to devices. By locating services at key points along this continuum, developers can better satisfy the latency requirements of their applications. Figure 1 illustrates the edge computing continuum, spanning from individual devices (shown on the left) to centralized data centers (shown on the right). Historically, cloud providers and Content Delivery Networks (CDNs) have reduced overall end-to-end latency by moving some services (such as the ability to cache data) out of centralized data centers and into distributed Points of Presence (POPs) closer to the devices being served. This has created a “cloud edge” or “internet edge” capable of improving the performance of traditional applications, such as streaming video and rich web content, but has not been enough to address many emerging applications, especially those that require a more sophisticated distribution of resources along the edge continuum for reasons of latency, bandwidth, security, privacy and autonomy.

This paper focuses on the two main edge tiers that straddle the last mile networks, the “**Service Provider Edge**” and the “**User Edge**”, with each being further broken down into subcategories. Figure 1 summarizes the edge continuum, along with key trends that define the boundaries of each category, including the increasingly complex design tradeoffs architects need to make the closer compute resources get to the physical world.

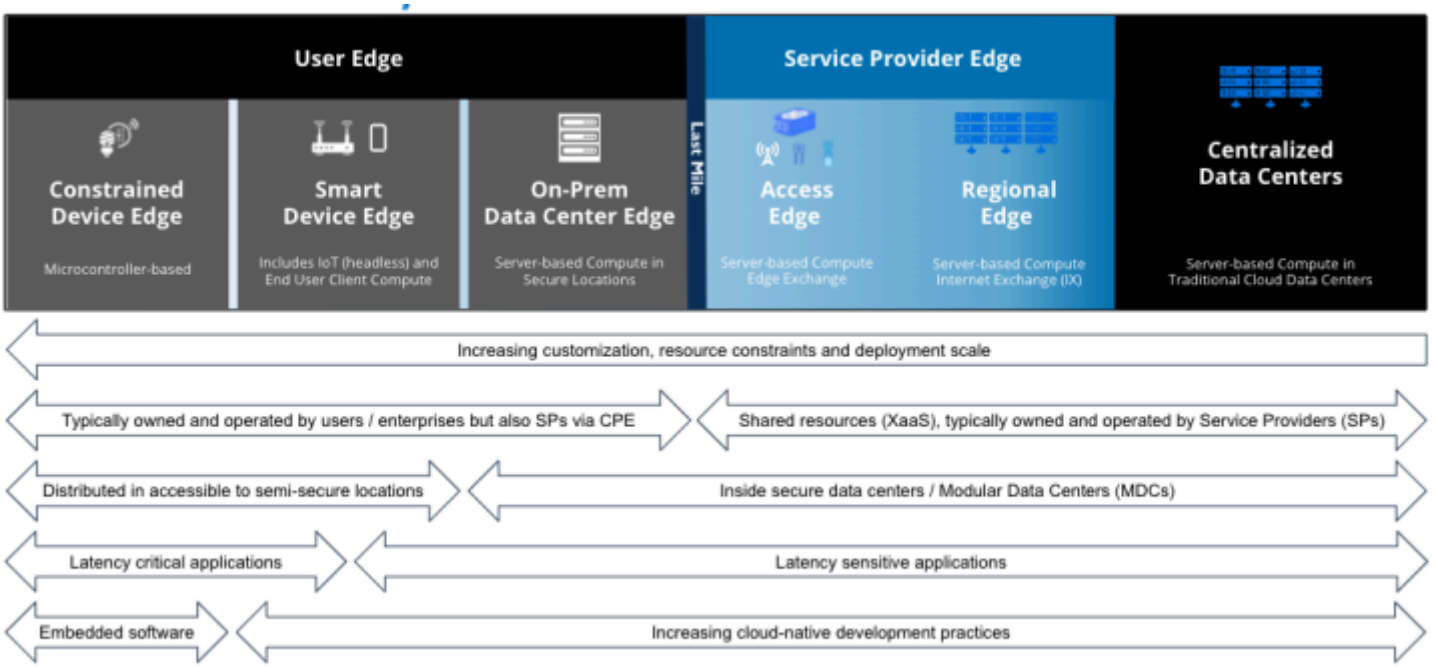


*Figure SEQ Figure * ARABIC 1: Summary of edge continuum.*

The far right of the diagram shows centralized data centers, which represent mostly cloud-based compute. These facilities offer economies of scale and flexibility that are not possible or appropriate on a device. Centralized cloud resources are practically unlimited, whereas device resources are inherently constrained. A centralized cloud can oversee the collective behavior of a large number of devices, for example configuring, tracking and managing them, but it’s limited by the centralized location of the data centers and the fact that the resources are shared.

Moving along the continuum from centralized data centers toward devices, the first main edge tier is the **Service Provider (SP) Edge**, providing services delivered over the global fixed/mobile networking infrastructure. Like the public cloud, infrastructure (compute, storage and networking) at the Service Provider Edge is often consumed as a service. Solutions at the Service Provider Edge can provide more security and privacy than the public cloud because of differences between the public internet and the private networks, including mobile cellular systems, operated by service providers. It leverages the existing trillion-dollar investments by Communications Service Providers (CSPs), who will have their own commodity servers in place at the network edge and will also cross-connect with cloud providers and bare-metal operators in nearby

locations. Infrastructure at the Service Provider Edge is generally more standardized than infrastructure at the User Edge but there are still unique requirements for regulatory compliance and ruggedization, depending on where it is deployed.

The Service Provider Edge is distributed and brings edge computing resources much closer to end users. For example, CSPs can leverage their fixed and mobile networks at the edge and provide a platform for many edge applications, thus creating new business models and use cases as part of a network's evolution, such as when a wireless provider upgrades their network to 5G. Also, in the case of fixed networks, CSPs terminate their networks within enterprise buildings and homes in the form of Customer Premise Equipment (CPE) and these resources can be leveraged further to deliver various edge services.

The second top-level edge tier is the **User Edge** which is delineated from the Service Provider Edge by being on the other side of the last mile network. Sometimes it is a necessity to use on-premise and highly distributed compute resources that are closer to end-users and processes in the physical world in order to further reduce latency. However, the most common reason for placing compute at the User Edge is to conserve broadband network bandwidth, reducing the need to unnecessarily backhaul data across the last mile network, whether to compute and storage at the Service Provider Edge or all the way back to centralized data centers. Additional reasons for placing compute at the User Edge include autonomy, increased security and privacy, and lower overall cost, if the available resources match the need of the application workload. Compared to the Service Provider Edge, the User Edge represents a highly diverse mix of resources. As a general rule, the closer that edge compute resources get to the physical world, the more constrained and specialized they become.

As indicated in Figure 1, a key difference between the edge tiers is who owns the computing assets. Resources at the Service Provider Edge and within the public cloud are typically not owned by the end user but are, instead, shared across many users. In contrast, resources at the User Edge are typically dedicated and customer-owned and -operated. Applications that only use resources on the User Edge result in a business model based on CAPEX rather than OPEX, with the infrastructure and technology acquisition, operational complexity and scaling being the responsibility of the user rather than delivered as a managed service. Increasingly, though, service providers (and cloud providers) are building managed service offerings that support and even include on-premise compute and networking infrastructure, making it possible to deliver applications that combine resources at both the User Edge and Service Provider Edge. Examples include a provider operating private cellular base stations for connectivity across a remote mining site or an analytics company providing Artificial Intelligence (AI) analysis and decision-support from the Service Provider Edge, supporting devices on the User Edge.

The edge computing taxonomy and associated terminology presented in this document were developed with careful consideration, seeking to balance various market lenses (e.g. telecom, cable, IT, OT/industrial, consumer) while also creating high-level categories according to key technical and logistical tradeoffs. These tradeoffs include whether a compute resource is capable of supporting application abstraction (e.g. through containers and/or virtual machines), whether it is in a physically-secure data center or accessible, and whether it is on a LAN or a WAN relative to the process/user it serves. This document seeks to provide a holistic point of view without using edge terminology that may mean something to one entity but can be confusing to another. For example, the terms "near" and "far" edge are commonly used by telecom providers to distinguish between infrastructure closer to users/subscribers (far edge) versus infrastructure further upstream (near edge). This can be confusing because relative location is viewed through the eyes of the service provider instead of the user. In another example, the terms "thin" and "thick" have been used in some circles to characterize degrees of on-premise edge compute capability, however these terms do not delineate between resources at the User Edge that are physically secured in a data center versus distributed in an accessible location.

2.2 Extending Cloud Native Principles to the Edge

With the introduction of containerization and Kubernetes, a rapidly increasing number of organizations are moving to cloud-native software development based on platform-independent, microservice-based architecture and Continuous Integration / Continuous Delivery (CI/CD) practices for software enhancements. The same benefits of cloud-native

development in the data center apply at the edge, enabling applications to be composed on the fly from best-in-class components, scaling up and out in a distributed fashion and evolving over time as developers continue to innovate.

In a perfect world, developers would have a universal foundation that enables them to deploy containerized workloads anywhere along the device to cloud data center continuum as needed, in order to balance the benefits of distributed and centralized computing depending on the use case and context. However, this isn't universally possible due to inherent technical and logistical tradeoffs, including the need to accommodate legacy investments and protect safety-critical systems.

As defined by the Open Glossary of Edge Computing, an "edge-native application" is one which is impractical or undesirable to operate entirely in a centralized data center. Edge-native applications leverage cloud-native principles while taking into account the unique characteristics of the edge in areas such as resource constraints, security, latency and autonomy. It is important to note that the term "edge-native" does not mean that an application isn't developed with the cloud in mind, rather that edge-native applications are designed to work in concert with upstream resources. An edge-optimized application that doesn't comprehend centralized cloud compute resources, remote management and orchestration, or leverage CI/CD isn't truly "edge native," rather it is a more traditional on-premise application. An example is a traditional Supervisory Control and Data Acquisition (SCADA) application within a nuclear power plant that has no connection to the cloud for security purposes. Ultimately, developers need a foundation that extends cloud-native principles as far down the edge continuum as feasible while accounting for inherent tradeoffs.

2.3 Considerations for the Service Provider Edge

The Service Provider Edge consists of infrastructure on the other side of the last mile network from the User Edge. It consists of two subcategories, the Regional Edge and the Access Edge, with the former traditionally being associated with backhaul networks and the latter with front- and mid-haul networks. Figure 2 illustrates the correlation between the Regional and Access Edges to these terms.

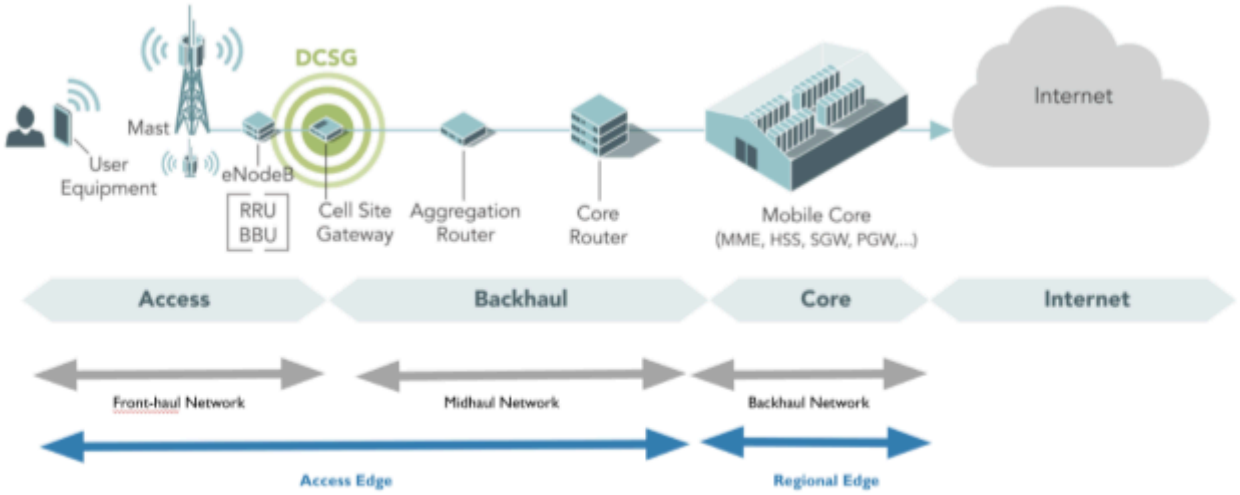


*Figure SEQ Figure * ARABIC 2. Regional and Access Edges mapped to various industry terms.*

To understand how the Service Provider Edge and its subcategories relate to each other and the rest of the internet, it helps to review how traffic routes to and from centralized data centers. Centralized data centers, such as those in Amazon's US West and US East regions, exist in specific locations that are far from most major metropolitan areas. These public cloud data centers connect to edge resources over internet backbones, which fan out across the continents and terminate in regional Internet Exchange Points (IXPs). IXPs exist in major cities and are the primary bridge between the access networks and the rest of the internet. For many reasons, centralized data centers are not well-suited for time-sensitive workloads, mainly because traffic from edge locations would need to travel relatively long distances and

traverse multiple network hops, both of which add latency and jitter. An emerging trend, however, is for public cloud operators to create regional caches to address this issue.

As a result, providers are increasingly locating compute resources in data centers at the Regional Edge to reduce network hops while still retaining moderate scalability benefits compared to resources located at the User Edge. These edge sites are sometimes owned by telco network operators but equally common are the Multi-Tenant Colocation (MTCO) facilities owned by companies like Equinix and Digital Realty. These MTCO companies have built large regional data centers adjacent to the IXPs, often in the same building, and lease space for servers and other IT equipment to multiple tenants, including the major public clouds. A rich confluence of data passes through these locations. There is an emerging trend for non-telco providers to build direct peering sites that bridge compute resources in regional data centers to centralized cloud data centers through the Internet Exchange, for example a provider like Equinix peering with a public cloud. Further, CDN operators are evolving to enable customers to run custom applications at IXP sites. As a general rule, Regional Edge data centers are capable of supporting edge workloads that can tolerate latencies in the 30ms - 100ms range.

Also within the Service Provider Edge is the Access Edge which spans the “middle mile” between regional data centers and the actual last mile network. Access Edge sites include front- and mid-haul infrastructure spanning cell towers, cable head-ends, aggregation and pre-aggregation hubs and central offices, and other facilities which house network access equipment such as cellular radio base stations, as well as xDSL and xPON equipment. Service providers and edge colocation companies are repurposing existing facilities and deploying small-to-medium-scale micro data centers at or near these access site locations to provide “one-hop” proximity to the last mile network. These data center facilities support low-latency workloads, including those that require a predictable connection to the last mile network with latencies below one millisecond.

As with the Regional Edge, there are many companies deploying IT equipment at the Access Edge, including telcos at their network access sites, but there is also an emerging trend for new business models operated by edge co-location companies. Compute resources on the Access Edge and the Regional Edge can work in concert to balance trade-offs between scalability, cost, complexity and latency.

2.3.1 Architectural Trends at the Service Provider Edge

Many webscale design principles can be applied to implement cloud-like compute capabilities at the Service Provider Edge. Over the last few years, orchestration technologies like Kubernetes have made it possible to run cloud-native workloads in on-premise, hybrid or multi-cloud environments. Most applications offloaded to the Service Provider Edge will not require significant changes in their design or code and will retain continuous delivery pipelines that can deploy specific workloads at Service Provider Edge sites, such as those which have low latency, high bandwidth, or strict privacy needs. In addition, workloads may interact with networks in complex ways, such as to prioritize Quality of Service (QoS) for specific applications based on needs such as giving priority to life safety applications.

Major content owners like Netflix, Apple and YouTube are expected to retain their cache-based distribution models, which entail storing states in the centralized public cloud along with Authentication and Authorization (AA) functions, while redirecting the delivery of content from the “best” cache as determined by Quality of Experience (QoE) at the client device, where “best” doesn’t always means the nearest cache. This approach will be retained for other distributed workloads utilizing edge acceleration like Augmented Reality (AR), Virtual Reality (VR), Massively Multiplayer Gaming (MMPG) etc.

Content delivery networks such as Akamai and Cloudflare will maintain their existing distribution models but will also likely increase the number of PoPs in their network as well as extend their networks farther out in the Service Provider Edge as they look to enhance their content caching capabilities while expanding their other lines of business, such as their security and distributed workload products, which benefit from being farther out in the network and closer to the devices.

Cloud providers, including Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure, are also expected to increase the number of PoPs in their network as well as extend their networks farther out to the Service Provider Edge. Each cloud provider will likely seek to differentiate their edge offerings in unique ways: some will focus on

AI workloads, others will look to simply expand the number of regions in which users can provision resources, and others will look to build edge capabilities into their IoT toolchains.

According to the design principles mentioned above, the Service Provider Edge will need to ensure a deterministic method of measuring and enforcing QoE based on key application needs such as latency and bandwidth. As most internet traffic is encrypted, these guarantees will likely be based on the transport layer, leading to the evolution of congestion control algorithms which determine the rate of delivery. A similar design principle will evolve for geographical data isolation policies for stores and workloads, beyond just complying with global data protection regulations.

Figure 3 shows an example deployment of highly-available edge applications at the Service Provider Edge, which could be federated across multiple service provider networks at peering sites while also cooperating with public cloud workloads.

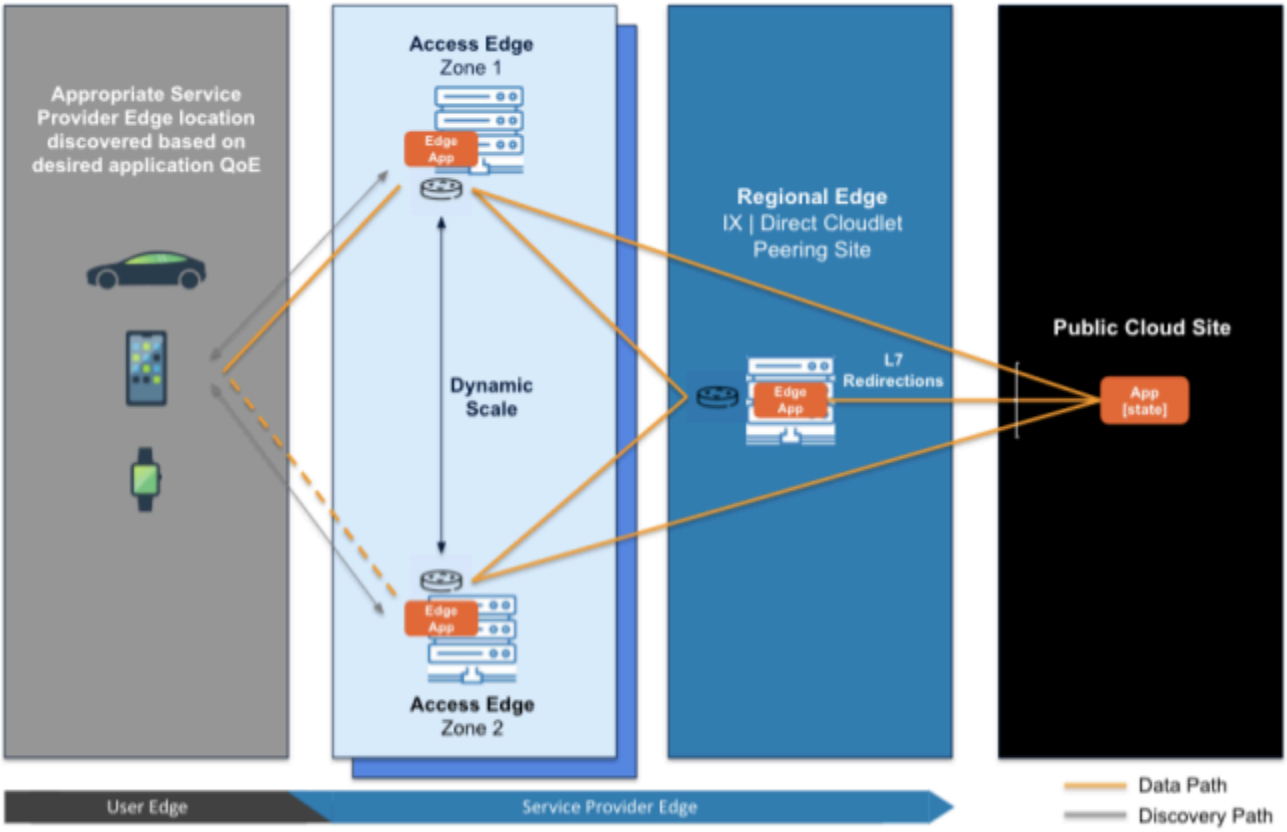


*Figure SEQ Figure * ARABIC 3. Example Service Provider Edge Application Deployment.*

2.3.2 Edge Application Deployment at the Service Provider Edge

Developers can study the geographical consumption patterns of their customers, as well as determine the optimal latencies and QOS requirements of their applications. Using Machine Learning (ML) algorithms, they can even predict how these patterns might change over time for advanced planning purposes. Orchestration services (such as custom Kubernetes schedulers) will emerge, and these will allow developers to specify their workload requirements in order to provide automated placement.

The deployment of application backends can be independent of network mobility or specific device attachment. Backend services deployment can be based on a number of different strategies to enable mobility of edge applications, including:

- **Static,** whereby the developer chooses the specific edge sites and the specific services for each site.

- **Dynamic,** whereby the developer submits criteria to an orchestration service and the orchestration service makes best-effort decisions about workload placement on behalf of the developer. One implementation of this would have developers choose a region in which they yield control to a system operator's or cloud operator's orchestration system in order to determine the optimum placement of workloads based on the number of requested compute instances, the number of users and any specialized resource policies.

The Akraino project is working on blueprints for the lifecycle management of edge applications based on the following workflow for deployment:

1. Create the cluster, deploying microservices as a set of containers or Virtual Machines (VMs);
2. Create the application manifest, defining an application mobility strategy that includes QoE, geographical store and privacy policies;
3. Create the application instance, launching the Edge Application and autoscaling.

For more information on this topic, please visit the developer section for the Akraino Edge Stack project.

2.3.3 Design Strategy for Backend Application Mobility Workloads at the Service Provider Edge

Workloads at the Service Provider Edge should instantiate and migrate based on demand and resource availability. For example, a backend for stateless applications might need to move across zones based on compute capacity, specialized resources and/or Service Level Agreement (SLA) boundaries. Stateful workloads can synchronize states from centralized servers and redirect them at layer 7 to edge applications, operating consistently, regardless of the orchestration system employed. The orchestration platform may offer periodic QoE hints to centralized servers to assist with the redirection process, but they can also operate independently.

2.3.4 Design Strategy for Edge Application Mobility at the Service Provider Edge

Application mobility is based on resource awareness, a backend for stateless applications that can move across zones based on compute capacity, specialized resources and/or Service Level Agreement (SLA) boundaries. Stateful edge applications synchronize states from centralized servers to the edge and redirect them at Layer 7 to edge applications, operating consistently regardless of an individual CSP's orchestration system. The CSP's platform may offer periodic QoE hints to centralized servers to assist with the above redirection process.

2.3.5 Design Strategy for User Device Mobility at the Service Provider Edge

Since device mobility is based on route awareness, it's important to review how data moves across mobile networks before explaining the design principles of device mobility.

A mobile device connected to a wireless network attaches to the nearest tower, then tunnels all application data to the nearest gateway which is further tunneled[1] to the regional gateway, which is then transferred over the internet exchange to a public cloud and back. Regional gateways called packet gateways (PGWs) can be viewed as anchors, which CSPs utilize for enforcing centralized subscriber control, like policies, billing and management. Routing data in this way, however, is sub-optimal and cannot enforce the latency, bandwidth and privacy guarantees which edge applications require. Application backends at the Service Provider Edge are then challenged to follow individual consuming devices as they move from one region to another.

[1] Tunnels are GTP-U encapsulations over IP. For more information, see this Wikipedia entry.

An easier solution is provided by local breakout[2], which allows service providers to place their anchors at edge sites, near the location of devices. Control and User Plane Separation (CUPS) for these packet gateways is a key step, deploying lightweight cost-effective distributed user plane functions (UPFs) at each edge site. Obtaining the GPS location for the UPF, if exposed from a centralized control plane, assists in identifying the nearest application backend. Another approach is for devices to attach to a geographically co-located anchor based on the physical location of the device, in which case local breakout works seamlessly with the edge cloud orchestration scheme behind these anchors.

Recent trends in 5G CUPS allows for local breakout and anchor redistribution, which is being deployed today. Network appliance vendors have started to virtualize their network functions, disaggregating their hardware from the software, and running network functions in virtual machines or containers. These are called Virtual Network Functions (VNFs) and Container Network Functions (CNFs). Network operators may use a common orchestration plane, such as that provided by Kubernetes, enabling CNF lifecycle management with a continuous delivery pipeline. The life cycle management techniques can be extended not only to anchors but to virtualized radio heads at the access edge.

The control plane separation will also allow Software Defined Networking (SDN)-like programming of tunnels to redirect traffic from devices to distributed endpoints. These tunnels can carry user application traffic as Packet Data User (PDU) sessions. Organizations like 3GPP are working on standards for redirecting edge application IP flows within PDU sessions so that they may be routed to the nearest anchors. The PDU sessions with embedded tunnel IDs as transport state present state synchronization issues, thus existing 3GPP session continuity procedures are not viable because they expect that a device will maintain PDU sessions across thousands of distributed anchors. Fortunately, the anchored routing structure can be changed by leveraging container mobility techniques used by web scale companies, but that requires not just virtualizing the compute (VNF/CNF) but also virtualizing the networks such that underlying IP routing is based on the identity of application and location of device. Identifier Locator Addressing is a means to implement network overlays without the use of encapsulation can help achieve anchorless device mobility.

2.3.6 Identifying the Optimum Edge Location to Serve a User

The nearest edge location is not always the best. Instead, clients must be steered to application backends based on the most recently recorded QoE for the application at each geographically-located edge site. The network may provide QoS mapping to improve QoE.

Based on this design, an application discovery engine could be embedded across multiple CSPs which records the health of the application backend and the QoE for each application, across all edge sites within a region, exposing a control API to identify the best location. This API can also be used to tune the rate of content delivery for the best experience. For example, content services like Netflix and YouTube maintain dozens of different bitrate encodings for the same movie or TV show, so that the optimal resolution can be delivered based on device characteristics, network congestion and other factors. A discovery engine can be employed that would return a ranked list of Uniform Resource Identifiers (URIs), identifying the optimum sites nearby, based on selection criteria that include:

- Edge application instances in sites geo-located based on the client's location;
- URI rank based on recent Layer 4 QoE measurements (latency and bitrate).

The LF Edge Akraino Edge Stack project has defined such an Application Discovery engine. Please visit the Find Cloudlet section for the Akraino Edge Stack project for more information and the definition of a control API implementation.

Later in the document there is a comprehensive list of use cases and workload attributes which individual CSPs can use today to serve enterprise- and privacy-centric edge use cases. However, to truly unlock the next generation of applications, developers must be able to deploy applications across multiple operator networks. One solution to this problem involves operators linking their edge cloud resources using a smart federation scheme at the last mile. Traditionally CSPs have federated to provide us global coverage, where sometimes they adopted the sub optimal

[2] Local Breakout enables the Mobile Network Operator (MNO) to break out internet sessions into the Home network, to provide inbound roamers with an ability to order data, which is provided directly by the visited network.

approach of rerouting traffic to anchors in the home network. A more efficient strategy is for CSPs to federate directly via a peering exchange as described above, or even across last mile Radio Access Networks (RANs).

2.4 Considerations for the User Edge

The User Edge consists of a diverse mix of compute form factors and capabilities that get increasingly unique as deployments get closer and closer to the physical world. Technical tradeoffs at the User Edge include varying degrees of compute capability (especially system memory) as well as the need for specific I/O capabilities to support both legacy and modern data sources. User Edge compute resources also require various degrees of ruggedization, including extreme temperature support with fanless design for reliability, specialized certifications (e.g. Class 1 / Division 2 for explosion proof), highly specific form factors, unique needs for Management and Orchestration (MANO) and security, and so forth. Moreover, while consumer-oriented devices tend to have a typical lifespan of 12-18 months, enterprise and industrial edge computing assets in the field need to support a long service life of 7+ years. Given all of these inherent tradeoffs, it is helpful to break the User Edge down further into several subcategories.

At the far-right end of the User Edge tier is the **On-Premise Data Center Edge** subcategory which can be considered server-class infrastructure located within traditional, physically-secure data centers and Modular Data Centers (MDCs), both inside and close to buildings like offices and factories. These resources tend to be owned and operated by a given enterprise and are moderately scalable within the confines of available real estate, power and cooling. Tools for security and MANO are similar to those used in a centralized cloud data center, however there is some evolution required to support coordination across multiple locations, such as with Kubernetes clusters.

In the middle of the User Edge is the **Smart Device Edge,** which consists of hardware located outside of physically-secure data centers but still capable of supporting virtualization and/or containerization to support cloud-native software development. These resources span consumer-grade mobile devices and PCs to hardened, headless gateways and servers that are deployed for IoT use cases in challenging environments such as factory floors, building equipment rooms, farms and weatherproof enclosures distributed within a city. While capable of general-purpose compute, these devices are performance-constrained for various reasons including cost, battery life, form factor and ruggedization (both thermal and physical) and therefore have a practical limit to processing expandability when compared to resources in an upstream data center. There is an increasing trend for these systems to feature coprocessing in the form of GPUs or FPGAs to accelerate analytics, with the added benefit of distributing thermal dissipation which is beneficial in extreme environments. Resources at the Smart Device Edge can be deployed and used standalone (e.g. a smartphone, IoT gateway on a factory floor) or embedded into distributed, self-contained systems such as connected/autonomous vehicles, kiosks, oil wells and wind turbines.

At the farthest extreme of the User Edge tier is the **Constrained Device Edge** subcategory, represented by microcontroller-based devices that are highly distributed in the physical world. These devices range from simple, fixed-function sensors and actuators that perform little-to-no localized compute to more capable devices such as Programmable-Logic Controllers (PLCs), Remote Terminal Units (RTUs) and Engine Control Units (ECUs) addressing time- and safety-critical applications. Devices at this tier leverage embedded software and have the most unique form factors to conform to highly specific environments and user experiences.

The Smart Device Edge includes both headless compute resources targeted at IoT use cases (e.g. gateways, embedded PCs, routers, ruggedized servers) and client devices that have a user interface (e.g., smartphones, tablets, PCs, gaming consoles, smart TVs). Together, constrained and headless Smart Devices represent the “things” in IoT solutions, with Smart Devices providing localized general-purpose compute capability. The spectrum of compute devices targeting IoT workloads is often referred to as the “IoT Edge”.

As a general trend in the area of networking, IoT use cases tend to be constrained by the upload of data collected from the physical world, whereas end user client use cases tend to be constrained by content download. This results in different considerations for applications, storage, network topologies and so forth, depending on the use case and available resources.

2.4.1 Securing and Managing Distributed Devices

Resources at the Constrained and Smart Device Edges are typically deployed and used in semi-secure to easily accessible locations in the field. As such, it is important to adopt a zero-trust security model and not pre-suppose a device is behind a network firewall. In all cases, distributed computing resources need a remote software update capability to avoid costly truck rolls, and in the case of on-premise data centers and smart devices, evolve their capability over time through modular, software-defined architecture. However, MANO and security solutions optimized for the data center are not suitable for the Constrained and Smart Device Edges due to the available compute footprint, deployment scale factor, potentially intermittent connectivity and typical lack of physical and network security. Solutions should also leverage techniques like Zero-Touch Provisioning (ZTP) to avoid requiring IT skill sets for secure deployment in the field.

IoT and client-centric compute resources at the Smart Device Edge are capable of leveraging MANO tools that support abstraction through containerization and virtualization and have headroom for security features like data encryption. Meanwhile, constrained devices leverage embedded software images that are typically tailored to the host hardware and may need to rely on a more capable device immediately upstream for added security measures. As a result, MANO tools for the constrained device edge often provide target support that is specific by device and manufacturer. Meanwhile smart devices can afford the necessary abstraction to make MANO tools more standardized and platform independent, such as through the use of Linux with containers working across both x86 and Arm-based IoT gateways or through a mobile operating system such as Android supporting applications on a variety of manufacturers' smartphones. Whenever possible, all key security functions (e.g. authentication, boot, encryption) should be enabled by a hardware-based root of trust, such as Trusted Platform Module (TPM) or Arm TrustZone, but this is not always an option for highly-constrained devices.

2.4.2 Accommodating both Legacy and Modern Applications

As with centralized cloud data centers, many user edge compute resources need to accommodate legacy applications in parallel with modern, cloud-native workloads. This is relatively straightforward in an on-premise data center through the use of well-established enterprise virtualization software together with Kubernetes, however it is not feasible to leverage these same tools on more constrained hardware deployed in the field. Special consideration must be made for an abstraction layer that is optimized for resource-constrained hardware and comprehends the unique security needs for devices distributed outside a secure data center. The ability to abstract virtualized and/or containerized workloads on a given compute node is typically limited by available memory, with the practical lower limit being roughly 256MB: just enough to host an abstraction layer together with a workload. This memory constraint is the primary delineator between the Smart and Constrained Device Edges and is generally the limit for extending cloud-native software development practices closer to the source of data. Below this memory capacity, software needs to be embedded with tight coupling to hardware which limits flexibility and reduces the scope for expandability through abstracted, modular applications.

2.4.3 Addressing Protocol Fragmentation in IoT Use Cases

Compared to the entirely IP-based data flow spanning the Cloud, Service Provider and On-Premise Data Center Edges, resources for IoT workloads serving constrained and smart devices must comprehend a diverse mix of legacy and modern connectivity protocols, spanning wired and wireless transport as well as both standard and proprietary formats. This is especially the case in the IIoT space where there are hundreds of legacy protocol formats to comprehend. Rather than expecting one connectivity standard to dominate, it is important to have edge software frameworks that can normalize a variety of IoT data sources into desired IP-based formats for further processing upstream. Openness here enables users to retain control over their data by not getting locked into any particular backend service.

2.4.4 Latency-Critical Applications

Safety- and latency-critical applications that require “hard” real time operation for deterministic response comprise another key driver for running workloads at the User Edge. Resources like PLCs, RTUs and ECUs have been used in industrial process control, machinery, aircraft, vehicles and drones for many years, requiring a Real-Time Operating System (RTOS) and specialized, fixed-function logic. Time- and safety-critical processes such as controlling a machine, applying a vehicle’s brakes or deploying an airbag are universally operated locally because they can't rely on control over a last-mile network, regardless of the speed and reliability of that connection. This scenario is contrasted with latency-sensitive applications such as video streaming that operate in “soft” real time and are often delivered by the Service Provider Edge for scalability. With latency-sensitive applications a networking issue can result in a poor user experience but will not cause a critical, potentially life-threatening failure.

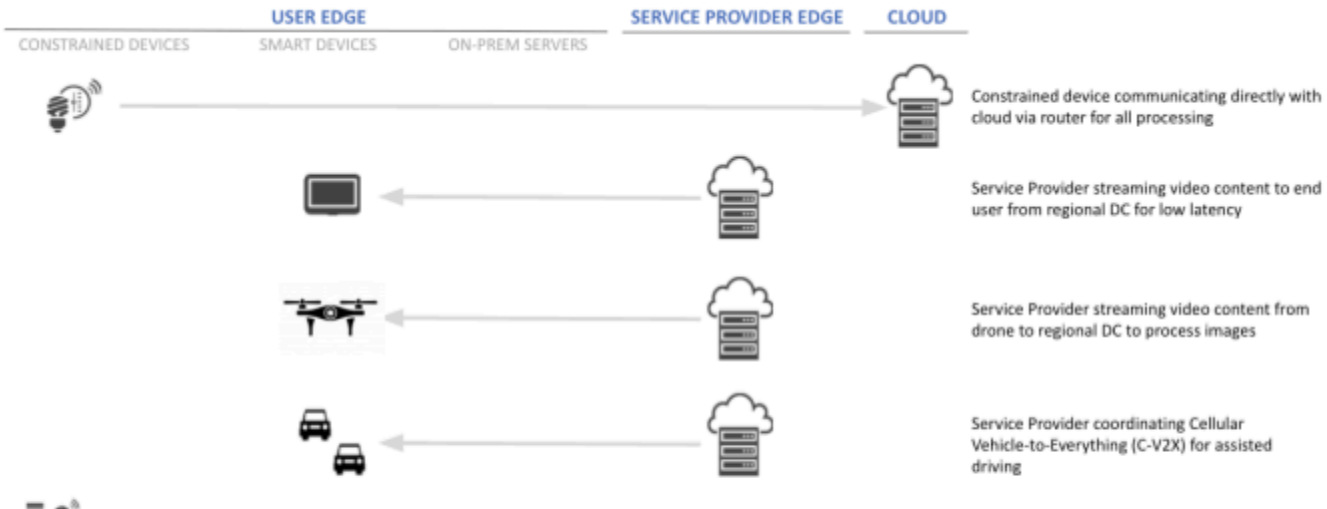
2.4.5 Separation of Concerns in IT and OT Environments

OT organizations have historically isolated their layered industrial control infrastructure (e.g. PLCs, SCADA and MES systems) from broader networks, to ensure security and uptime. However, a key aspect of IIoT involves connecting these assets and the associated processes to networked intelligence to drive new outcomes. In order to create a separation of concerns from control systems with no risk of disrupting existing processes, industrial operators rely on network segmentation, typically installing a secondary “overlay network” that taps into data from existing control systems, in addition to new sensors installed throughout their environment to enable analytics workloads. Meanwhile, there is a trend for consolidation of mixed-criticality workloads on common infrastructure, for example with a virtualized “soft PLC” providing control functionality while additional virtualized and/or containerized data management, security and analytics applications run in parallel and interact with higher edge tiers. This consolidation requires specific considerations in the abstraction layer to ensure separation of concerns between these mixed-criticality workloads.

In summary, developers need flexible tools at the user edge that enable them to run legacy, safety- and latency-critical, modern containerized workloads concurrently while protecting their operations from undue risk, all while taking advantage of the scale benefits of working together with the Service Provider Edge and the Cloud. Table 1 provides a detailed summary of attributes by edge tier.

2.5 Edge Deployment Patterns

The sub-categories under the User Edge work with the Service Provider Edge and Cloud as part of a tiered compute continuum, but not necessary in series. Constrained and smart devices distributed in the physical world (such as smart thermostats, smartphones and connected vehicles) often communicate directly with the Service Provider Edge and Cloud, bypassing all On-Premise Data Center infrastructure. Devices can also be deployed on-premise and interact with more capable local edge compute, which in turn interacts with the Service Provider Edge and Cloud. The continuum is a complex matrix of locality, capability, form factor and ownership. Figure 4 illustrates examples of various edge deployment patterns.



Note that this diagram is simplified in the sense that it does not take into account that resources will be communicating “north, south, east and west” with multiple peers across the continuum, depending on use case. This is often referred to as “fog computing”.

2.6 Trends for Edge AI

Regarding Artificial Intelligence and Machine Learning (AI/ML) at the edge, the general trend is for deep learning and model training to occur where resources are plentiful, as in the centralized cloud, with models subsequently being pushed to more constrained resources at the Service Provider and User Edges for performing inferencing on data locally. The location of model execution along the edge continuum depends on a variety of factors, including addressing latency issues, ensuring autonomy, reducing network bandwidth consumption, improving end user privacy and meeting requirements for data sovereignty.

There is an emerging trend for running federated learning and even training models at the edge to address privacy and data sovereignty issues, although the potential for regional bias then needs to be considered. Another emerging trend at the Constrained Device Edge is deploying ML inferencing models in microcontroller-based resources. An example is a ML model that enables a smart speaker to recognize a wake word (e.g. “Hey Google” or “Hey Alexa”) locally before subsequent voice interactions are powered by servers further up the compute continuum. Dubbed “Tiny ML”, this requires highly specialized toolsets to accommodate the available processing resources and is outside of the scope of LF Edge at the time of this writing.

2.7 Edge Computing Use Cases

Enterprises in numerous market segments are deploying edge-hosted applications in order to capitalize on new business opportunities that are enabled by provisioning local compute as an extension of centralized cloud architectures. Figure 5 provides some examples of the wide number of use cases that benefit from edge computing and related enabling technologies.

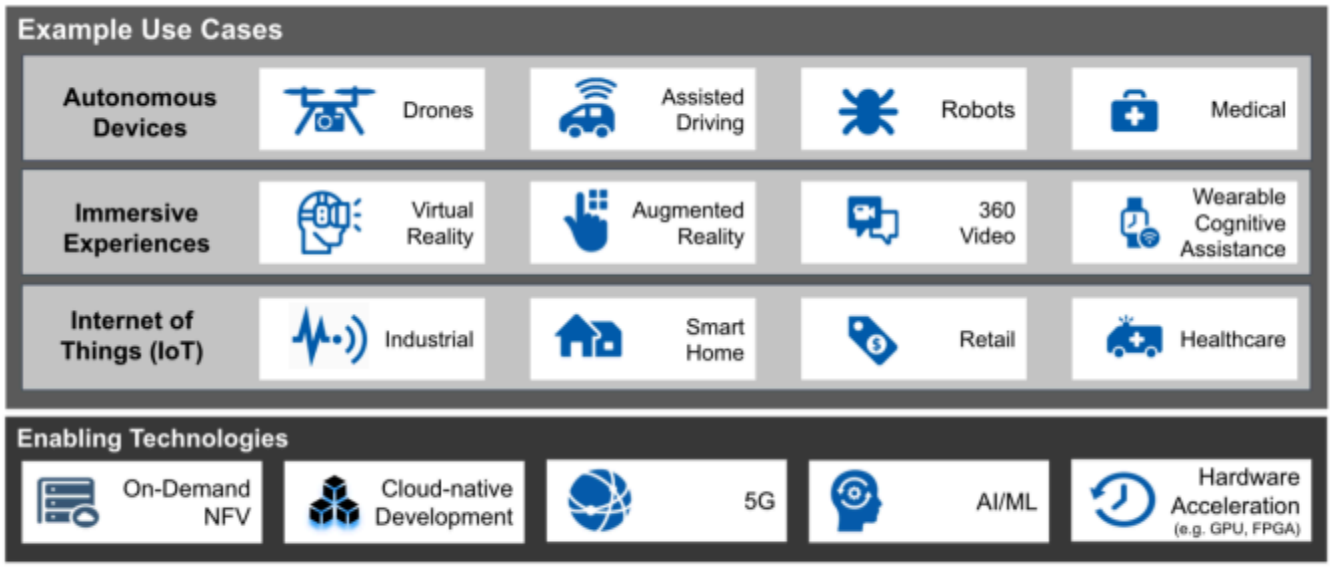


*Figure SEQ Figure * ARABIC 5. Example edge computing use cases and enabling technologies.*

This section discusses a variety of use cases to highlight key considerations and benefits:

- Industrial IoT;

- Computer Vision;
- Augmented Reality
- Retail;
- Gaming;
- Assisted Driving.

2.7.1 Industrial IoT (IIoT)

Edge compute delivers a number of important benefits for IIoT use cases in markets such as manufacturing, utilities, oil and gas, agriculture and mining.

With edge computing, industrial operators can perform time-critical analytics close to their sensors, machines and robots, reducing the latency for operational decision-making. This makes their processes more agile and responsive to changes. To maximize efficiency and minimize OPEX, functions necessary for real-time operation are hosted on-premise while those that are less time-critical may run in the public, private or hybrid cloud.

An edge compute architecture can ensure that high-value, proprietary information never leaves a factory. This can minimize the security threats associated with transmitting data to the cloud over public networks that are vulnerable to hacking, as well as help organizations meet data sovereignty requirements.

Remote operations such as mines or oil rigs typically have intermittent connectivity to the cloud and must be able to function autonomously. In these scenarios, on-premise device edge compute enables real-time operational decisions based on the local analysis of sensor data. Data required for long-term process optimization or multi-site aggregation is sent to the cloud whenever connectivity is available, which in certain locations might only be over a satellite link available at irregular intervals.

A significant amount of data is “perishable”, meaning it is only valuable if acted on in the moment. The cost of connectivity through the service provider edge is minimized by processing data from sensors locally and sending only relevant information to the cloud, instead of raw streams of data. This is critical for high-bandwidth vibration data used for predictive maintenance use cases or devices like smart meters used in agriculture or utilities that connect to the cloud via low-bandwidth Narrowband IoT (NB-IoT) networks. In these cases, additional data might periodically be centralized in the cloud for the training of AI models that are then pushed close to operations at the edge for inferencing.

2.7.2 Computer Vision

Computer vision technology is widely used in video surveillance for law enforcement and building security, as well as monitoring industrial processes. Modern high-resolution IP cameras, however, generate significant volumes of data, for example 4 Mbps per device for an array of 4-megapixel (MP) cameras. While cameras can be configured to minimize bandwidth requirements by transmitting only when motion is detected, this is of no help in environments such as a city street or factory production line where the surveillance system network connection must be provisioned to cope with constant motion. Analyzing video in the cloud therefore requires a high-bandwidth network connection to transmit a continuous, high-resolution stream. If network bandwidth is constrained, the accuracy of the analysis is limited by the lower resolution of compressed video.

With edge compute, however, high-resolution video data is processed either within the smart camera itself as the edge node, or on a nearby edge server. High-end IP cameras have sufficient processing power to run algorithms such as facial recognition, leveraging analytics based on AI and/or deep learning technologies. Only selected events and/or video sequences that are flagged as important are transmitted to the cloud, for example an individual of interest, a vehicle with a

specific license plate or a defective component. This significantly reduces the required network bandwidth while ensuring high quality, high accuracy analytics.

Edge compute also reduces latency, which is important for any time-critical vision-based detection scenarios such as factory automation or facial recognition. Continuous process control, for example, leverages the low latency associated with edge compute to ensure the near-real time detection of process deviations or manufacturing flaws, enabling production lines to be stopped or control parameters to be adjusted quickly enough to minimize wastage.

Drones used in applications such as surveying, package delivery and surveillance will leverage low-latency computer vision systems that perform object recognition for navigation within edge compute nodes on the ground rather than in heavy, power-hungry systems on the drone itself. This reduces the cost of the drones while also minimizing their power consumption, thereby maximizing both battery life and flight time.

Processing video at the user edge also mitigates privacy concerns, especially in surveillance applications that are subject to regulatory constraints or in commercial applications where process information is valuable intellectual property.

2.7.3 Augmented Reality (AR)

Enterprises are increasingly adopting Augmented Reality (AR) in order to improve the efficiency of their operations, leveraging technology familiar to consumers through applications like Pokémon Go and its more sophisticated successors.

In industrial environments, AR can guide lesser-skilled workers through maintenance tasks without having an expert engineer on site at all times. This can either be done with a pre-scripted instruction overlay in the worker's field of view, or with a remote expert talking the on-site worker through performing a complicated task through their eyes. Similarly, in aerospace AR provides technicians with maintenance and diagnostic information via smart goggles, eliminating the need to physically reference bulky, complex and possibly outdated manuals in hard-to-reach locations inside a wing, fuselage or engine cowling.

AR applications typically analyze the output from a device's camera to supplement the user's experience. The application is aware of the user's position and the direction they are looking in, with this information provided via the camera view and/or positioning techniques. The application is then able to offer information in real-time to the user, but as soon as the user moves that information must be refreshed. Additionally, for many use cases it is valuable to update critical real time data from sensors in the user's field of view, for example the temperature and pressure of a tank while an operator is working through a maintenance procedure.

Edge compute improves the efficiency of enterprise AR by reducing the dizziness associated with high latency and slow frame refresh rates, that can otherwise lead to an experience that is frustrating, potentially nausea inducing and ultimately disorienting. Edge-hosted systems ensure predictable latency, resulting in a consistent experience for users instead of the constantly-changing delays that result from cloud-hosted implementations. Moving compute power into edge servers located close to the user allows an AR application to eliminate the need for high processing bandwidth on goggles that therefore become expensive, power-hungry and too heavy for comfortable use over an extended period.

In another example, edge computing and AR are poised to deliver truly immersive media experiences for sports fans while at the game. Sports such as baseball, cricket, football and soccer have already held successful trials in "smart stadia" enabling spectators to stream video from unique, custom camera angles, including drones and spider-cams. "Virtual cameras" present views from within the field of play, giving spectators the opportunity to experience the action from the perspective of the players themselves. All these use cases require edge compute in order to guarantee the responsiveness that spectators expect while eliminating the need to backhaul prohibitive amounts of data to the cloud.

2.7.4 Retail

For brick-and-mortar retailers, almost 90% of global retail sales occur in physical stores so most retailers are investing in computing infrastructure located closer to the buyer, with edge computing as an extension of their centralized cloud environments. In-store edge environments focus on the digital experience of the customer, through edge applications supporting local devices such as smart signage, AR-based mirrors, kiosks and advanced self-checkout.

Retailers can deliver personalized coupons when shoppers walk into stores as WiFi, beaconing and computer vision systems recognize customers who previously signed up to connect while in-store. Smart fitting rooms equipped with AR mirrors can show shoppers in different clothing without the requirement to physically try them on. Meanwhile, infrared beacon and computer vision technology can generate heat maps that provide retailers with insights on in-store traffic patterns, allowing them to better configure their space and optimize their revenue-per-square-foot.

Infusing self-checkout systems with computer vision capability and integrating them with RFID and Point-of-Sale (PoS) systems gives them the ability to confirm that the item scanned by a customer matches what's in their bag, improving loss prevention. Vision algorithms can also be used to enable facial recognition to authenticate payments and gesture recognition for touchless commands, as well as the delivery of personalized offers at the point of sale.

Through the use of edge compute, retailers are able to ensure improved security for sensitive customer information. When data is transferred from devices to the cloud, security and compliance risks increase, but edge compute applications can filter information locally and only transfer data to the cloud that is required for strategic operational planning.

Edge compute provides a lean, highly reliable IT infrastructure for retailers that can run multiple applications while supporting the control and flexibility of cloud-based services. High-resiliency in-store micro data centers have become the solution of choice, managed and orchestrated remotely so that IT staff aren't required on-site. A chain of retail stores can be treated as an entire ecosystem rather than just a collection of individual locations.

Most large retailers have tremendous investments in both cloud-native and mobile applications, for the benefit of their customers, their associates and their employees. The edge continuum provides them with the opportunity to use the same software development tools for both environments, as well as the same deployment tools for deploying applications to the data center, the cloud, or elsewhere along the continuum to the on-premise user edge. If retailers fail to leverage the edge continuum this way and continue to manage their on-premise investments as traditional enterprise IT assets, they will deprive themselves of the flexible, responsive and dynamic attributes that their cloud and mobile teams already enjoy.

2.7.5 Gaming

Massively Multiplayer Games (MMPGs) served up by the cloud typically involve players controlling their avatars, with any movement of an avatar needing to be communicated as quickly as possible to all players who have that avatar in their field of view. Latency has a major impact on the overall user experience, to the point where perceptible delays can render a game effectively unplayable. A video game must appear to respond instantaneously to keystrokes and controller movements, implying that any commands issued must complete a round-trip over the network and be processed fast enough by the data center for the player to feel like the game is responding in real time. For the best multiplayer experience, the latency must be consistent across all players, otherwise those with the lowest latency have the opportunity to react faster than their competitors.

Edge compute improves the experience of cloud-enabled gaming by significantly reducing latency and providing the necessary storage and processing power in edge data centers. With processing centers for a game running at the edge of the network, for example in each metro area, the ultra-low latency results in reduced lag-time. This enables a more interactive and fully immersive experience than if the game is hosted in a remote cloud data center.

Edge compute is expected to trigger new subscription-based MMPG business models along with reduced hardware costs for end-users: with edge processing enabling high-quality experiences, less processing power is required in the users'

hardware itself. The gaming industry hopes that this reduction in hardware costs will spur greater user investment in new subscriptions, driving overall growth in this segment.

2.7.6 Assisted Driving

While edge compute will be a critical enabler for the holy grail of fully-autonomous driving, that vision is many years away from being realized, for a host of reasons beyond the scope of this paper. Assisted driving technologies, however, are being deployed today and edge compute is key to their viability.

The number of sensors in a vehicle grows with each model year, along with each introduction of new capabilities in safety, performance, efficiency, comfort and infotainment. Although most of the sensor data is processed in the vehicle itself due to autonomy and the safety considerations of latency critical applications, some capabilities, such as alerting in the event of deviations from the norm, require data to be moved to the cloud for analysis and follow-up. Edge computing helps to limit the amount of data that is sent to the cloud, reducing the data transmission cost and minimizing the amount of sensitive data such as Personally Identifiable Information (PII) leaving the vehicle.

The infotainment system in a vehicle is the most prominent user interface besides the driving controls. To learn what functions and applications users are really using and where the design of interactions should be optimized, ML algorithms represent an important tool for uncovering relevant insights within the vast amount of available data. Edge compute brings ML models, which were trained in the cloud, to the vehicle itself, so that the available behavioral and sensor data can be used locally for predictions that improve overall user interaction.

Efficient battery monitoring and predictive maintenance are key to the long-term customer experience for vehicle owners and operators. Edge compute addresses these challenges through the ability to aggregate data and perform the real-time evaluation of relevant battery parameters and sensor values. Appropriate information can be automatically uploaded to backend operational systems in the cloud, enabling dealers or fleet operators to automatically schedule preventative maintenance at a time and place that balances convenience for the user against the severity of problems that have been detected.

Edge compute technologies can enable secure, frictionless entry to a vehicle based on multi-factor authentication, for example using a camera for face recognition, an infrared camera for spoofing detection and a Bluetooth sensor to detect the proximity of the driver’s smartphone.

Finally, once the proportion of smart vehicles reaches a critical threshold within a certain geography, smart traffic management will become feasible, enabled by roadside edge compute. In one example, if a road intersection has an edge node deployed to which the majority of vehicles can communicate while coming towards the intersection, the edge node can aggregate the location and speed data from nearby vehicles, optimize traffic light timing for efficient traffic flow and notify the smart vehicles in advance about the situation at the intersection. Widespread deployments of such edge nodes will enable Cellular Vehicle-to-Everything (C-V2X) applications that optimize traffic flows not only for individual intersections, but over wider areas thanks to the cloud-based analysis of edge data and the centralized orchestration of the individual intersections.

2.7.7 Summary of the Edge Continuum

Each edge tier represents unique tradeoffs between scalability, reliability, latency, cost, security and autonomy. In general, compute at the user edge reflects dedicated, operated resources on a wired or wireless local area network (LAN) relative to the users and processes they serve. Meanwhile, the Service Provider Edge and Public Cloud are shared resources (XaaS) on a wide area network relative to users and processes. Table 1 summarizes key attributes of each edge.

In many applications, User Edge workloads will run in concert with Service Provider Edge workloads. Workloads on the User Edge will be optimized for bandwidth savings, latency criticality, safety and security, whereas workloads on the Service Provide Edge will be optimized for scale. For example, an AI/ML model might be trained in a centralized cloud

data center or on the Service Provider Edge but pushed down to the User Edge for execution. Table 1 summarizes key attributes of each edge.

Attribute	User Edge			LAST MILE NETWORK	Service Provider Edge		Centralized Cloud Data Centers
	Constrained Device Edge	Smart Device Edge	On-prem Data Center Edge		Access Edge	Regional Edge	
Hardware Class	Constrained microcontroller-based embedded devices (e.g. voice control speakers, thermostats, light switches, sensors, actuators, controllers). KBs to low MBs of available memory.	Arm and x86-based gateways, embedded PCs, hubs, routers, servers, small clusters. >256MB of available memory but still constrained. Accelerators (e.g. GPU, FPGA, TPU) depending on need.	Standard servers and networking with accelerators		Standard servers and networking with accelerators, telco radio infrastructure	Standard servers and networking with accelerators	Standard servers and networking with accelerators
Deployment Locations	Highly distributed in the physical world, embedded in discrete products and systems	Distributed in field, outside of secure data centers (e.g. factory floor, equipment closet, smart home) or embedded within distributed systems (e.g. connected vehicle, wind turbine, streetlight in public R.O.W.)	Secure, on-premise data-centers and micro-data centers (MDCs), e.g. located within an office building or factory. Typically owned and operated by enterprises.		O, RO, Satellite DCs, owned and operated by service providers (e.g. ISPs, CSPs). Resources can also be located at User Edge in the case of CPE owned and managed by a service provider	CO, RO, Satellite DCs, owned and operated by service providers (e.g. ISPs, CSPs).	Centralized DCs, Zones, Regions owned and operated by CSPs. Compute in DCs located near key network Points of Presence (PoP) is at the "Cloud Edge" or "Internet Edge".
Global Node Footprint	Trillions	Billions	Millions		Hundreds of Thousands	Tens of Thousands	Hundreds
Role/ Function	Fixed to limited function applications, rely on higher-classes of compute for advanced processing. Emerging simple ML capability via TinyML.	Hyperlocal general compute for apps and services. Dynamic, SW-defined configuration with limited scalability. Includes IoT Compute Edge (headless systems) and End User devices (e.g. smartphones, PCs, gaming consoles).	Local general compute for applications and services with moderate scalability. Dedicated to a specific enterprise.		Providing last miles access to the internet for users/enterprises. High availability, pubic and private, general and special. Broad scalability. Shared resources for IaaS, PaaS, SaaS, SDN (XaaS).	High availability, pubic and private, general and special. Broad scalability. Shared resources for IaaS, PaaS, SaaS, SDN (XaaS).	Hyperscale or webscale, public, general purpose. Public cloud involved shared resources for IaaS, PaaS, SaaS, SDN (XaaS).
Software Architecture	Embedded software/ firmware, Real-time Operating Systems (RTOS) for time-critical applications.	Bare metal to containerized/ virtualized depending on capability and use case. Linux, Windows and mobile OS'es (e.g. Android, iOS).	Virtualized, containerized and clustered compute. Linux and Windows.		Virtualized, containerized and clustered compute. VNF, CNF, managed services, networking. Linux and Windows.	Virtualized, containerized and clustered compute. VNF, CNF, managed services, networking. Linux and Windows.	Bare metal, VMs, Clusters, Containers, all architectures, all services. Linux and Windows.
Security, M&O	Specialized OTA M&O tools, often custom by device/manufacturer. May rely on higher-class compute for security.	Require specific security and M&O tools due to resource constraints, unique functionality, accessibility and limited field technical expertise. Often unable to rely on a network firewall.	Evolution of cloud data center security and M&O tools to support distributed Kubernetes clusters. Benefits from physical and network security of purpose-built data centers.		Evolution of cloud data center security and M&O tools to support distributed Kubernetes clusters in regional locations	Evolution of cloud data center security and M&O tools to support distributed Kubernetes clusters in regional locations	Traditional cloud data-center security and M&O tools
Physical Attributes	Highly-specific form factors for every device	Diverse mix of specialized form-factors with unique I/O, industrial ruggedization, regulatory certifications, etc. based on use case	General purpose server-class infrastructure with some ruggedization and regulatory considerations (e.g. for MDCs)		Purpose-built radio infrastructure. General purpose server and networking hardware. Power, thermal, ruggedization and regulatory considerations for localized resources.	General purpose server and networking infrastructure with power, thermal, ruggedization and regulatory considerations for localized resources	General purpose server infrastructure

*Table SEQ Table * ARABIC 1: Summary of Edge Attributes.*

The boundaries between edge tiers are not rigid. As mentioned previously, the Service Provider Edge can blend into the User Edge when CPE resources are deployed on-premise in order to provide a user with connectivity and compute as a managed service. Meanwhile, the User Edge can also extend to the other side of the last mile network, as in the case of enterprise-owned private cloud data centers. While the edge boundaries are not rigid, they are instructive: certain technical and logistical limitations will always dictate where workloads are best run across the continuum based on any given context.

Regardless of the definitions of various edge tiers, the ultimate goal is to provide developers with maximum flexibility, enabling them to extend cloud-native development practices as far down the continuum as possible, while recognizing the practical limitations. The following sections dive deeper into LF Edge and how each project within the umbrella is working to realize this goal.

3 LF Edge Project Portfolio

The Linux Foundation's LF Edge (LFE) was founded in 2019 as an umbrella organization to establish an open, interoperable framework for edge computing independent of hardware, silicon, cloud or operating system. The project offers structured, vendor neutral governance and has the following mission:

- Foster cross-industry collaboration across IoT, Telecom, Enterprise and Cloud ecosystems;
- Enable organizations to accelerate adoption and the pace of innovation for edge computing;
- Deliver value to end users by providing a neutral platform to capture and distribute requirements across the umbrella;
- Seek to facilitate harmonization across edge projects.

As with other LF umbrella projects, LF Edge is a technical meritocracy and has a Technical Advisory Committee (TAC) that helps align project efforts and encourages structured growth and advancement by following the Project Lifecycle Document (PLD) process. All new projects enter as Stage 1 "At Large" projects which are projects that the TAC believes are, or have the potential to be, important to the ecosystem of Top-Level Projects, or the edge ecosystem as a whole. The second "Growth Stage" is for projects that are interested in reaching the Impact Stage, and have identified a growth plan for doing so. Finally, the third "Impact Stage" is for projects that have reached their growth goals and are now on a self-sustaining cycle of development, maintenance, and long-term support.

3.1 LF Edge Project Summaries

LFE comprises the following open source projects, explained in more detail in the online resources:

3.1.1 Stage 3: Impact Projects

- **Akraino Edge Stack** is a software stack that supports high-availability cloud services optimized for edge computing systems and applications. It offers users new levels of flexibility to scale edge cloud services quickly, to maximize the applications and functions supported at the edge and to help ensure the reliability of systems that must be completely functional at all times. Akraino Edge Stack delivers a deployable and fully-functional edge stack for edge use cases including IIoT, telco 5G core, virtual Radio Access Network (vRAN), Universal Customer Premises Equipment (uCPE), Software-Defined Wide Area Networking (SD-WAN) and edge media processing. It creates a framework for defining and standardizing APIs across stacks, via upstream/downstream collaboration. Akraino Edge Stack is currently composed of multiple blueprint families that include specific blueprints under development. The community tests and validates the blueprints on real hardware labs supported by users and community members.
- **EdgeX Foundry** is a vendor-neutral, loosely-coupled microservices framework that enables flexible, plug-and-play deployments that leverage a growing ecosystem of available third-party offerings or to include proprietary innovations. At the heart of the project is an interoperability framework hosted within a full hardware- and OS-agnostic reference software platform. The reference platform helps enable the ecosystem of plug-and-play components that unifies the marketplace and accelerates the deployment of IoT solutions. EdgeX Foundry is an open platform for developers to build custom IoT solutions, either by feeding data into it from their own devices and sensors, or consuming and processing data coming out.

3.1.2 Stage 2: Growth Projects

- **EVE** is an edge computing engine that enables the development, orchestration and security of cloud-native and legacy applications on distributed edge compute nodes. Supporting containers, clusters, VMs and unikernels, it provides a flexible foundation for IoT edge deployments with a choice of hardware, applications and clouds.
- **Home Edge** is a robust, reliable and intelligent home edge computing open source framework, platform and ecosystem. It provides an interoperable, flexible and scalable edge computing services platform with APIs that can also be used with libraries and runtimes.

- **State of the Edge** is a vendor-neutral platform for open research on edge computing dedicated to accelerating innovation by publishing free, shareable research and analysis on edge computing. The project publishes the yearly State of the Edge reports, maintains the Open Glossary of Edge Computing and oversees the LF Edge Interactive Landscape.

3.1.3 Stage 1: At Large Projects

- **Baetyl** (pronounced “Beetle”) is a general-purpose platform for edge computing that manipulates different types of hardware facilities and device capabilities into a standardized container runtime environment and API, enabling the efficient management of application, service and data flow through a remote console both in the cloud and on-premise.
- **Fledge** is a proven software framework for the industrial edge focused on critical operations, predictive maintenance, situational awareness and safety. Fledge has been deployed in industrial use cases since early 2018. Fledge is architected to integrate IIoT, sensors, machines, ML/AI tools-processes-workloads and clouds with industrial production process (Level 0), sensing and manipulating (Level 1), monitoring and supervising (Level 2), manufacturing operations management (Level 3) and business planning logistics (level 4), as per ISA95.
- **Open Horizon** is a platform for managing the service software lifecycle of containerized workloads and related machine learning assets. It enables management of applications deployed to distributed webscale fleets of edge computing nodes and devices without requiring on-premise administrators.

3.2 Project Focus Across the Edge Continuum

The general focus area for each project along the edge continuum is depicted in Figure 6, though the scope of each project tends to span further across the spectrum as it integrates with various upstream and downstream efforts. This includes extending up and down the compute continuum and offering varying degrees of application- vs. infrastructure-centric benefits.

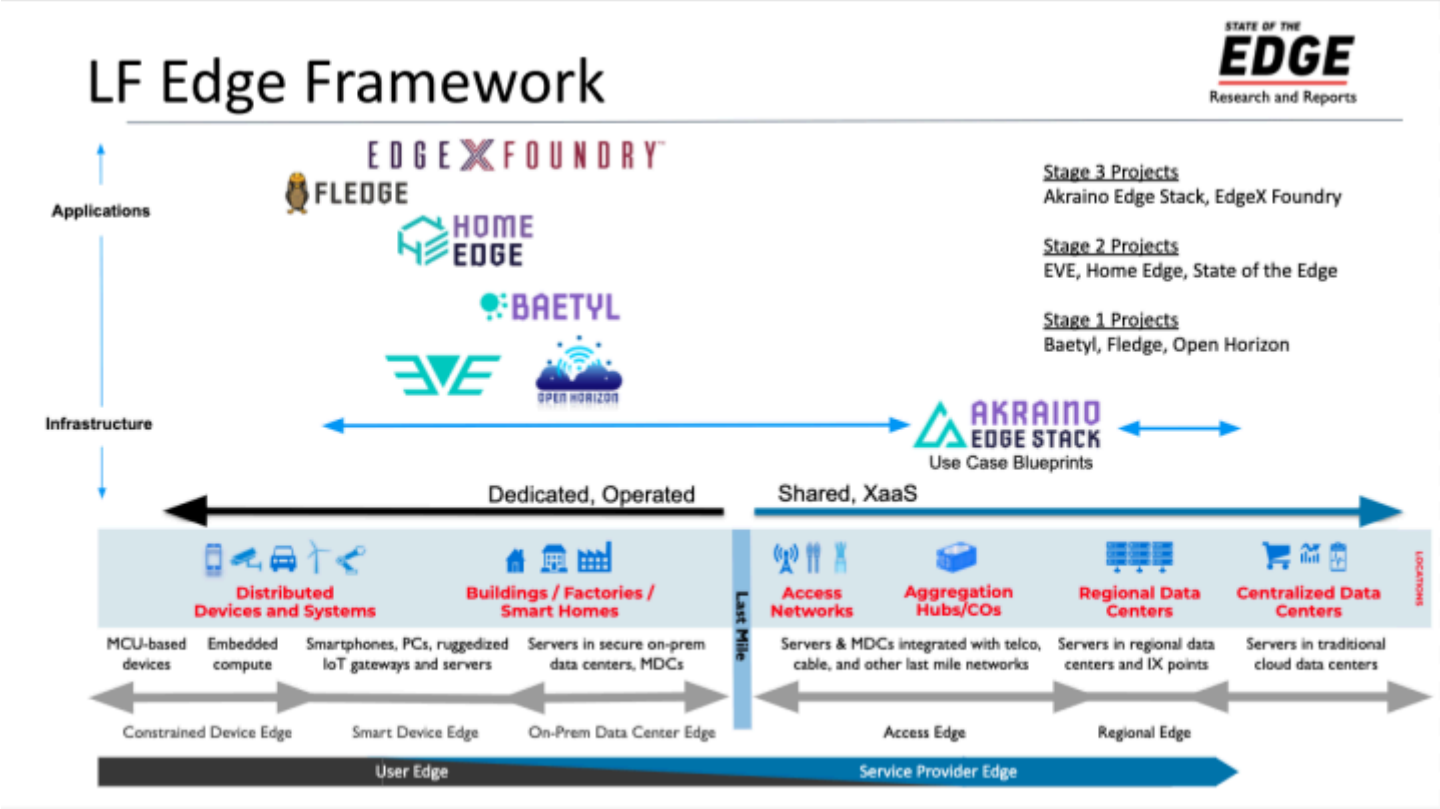


*Figure SEQ Figure * ARABIC 6: LF Edge project framework.*

In terms of general project focus, Akraino addresses the unique infrastructure needs of the Service Provider Edge through holistic blueprints, with reach into the various subcategories of the User Edge.

The mission of Project EVE is to create a universal orchestration foundation for enterprise and IIoT edge computing use cases at the Smart Device Edge, like Android has provided for smartphones. EVE addresses the need to accommodate both legacy and modern applications on constrained IoT edge compute resources while meeting the unique security and scale requirements for devices deployed outside of the data center.

Baetyl and Open Horizon are focused on enabling the delivery of containerized workloads to resources distributed across the Smart Device Edge but also have a footprint that extends through the Service Provider Edge to the Cloud. The Open Horizon controller can be deployed centrally in the cloud, regionally at the Service Provider Edge or locally at the On-Premise Data Center Edge.

EdgeX Foundry and Fledge serve as application frameworks for IoT use cases at the Smart Device Edge to address the fragmentation in the market stemming from diverse technology choices spanning hardware, operating system and connectivity protocols. These frameworks provide an open foundation for deploying analytics and other value-added services, with each taking a slightly different architectural approach that balances tradeoffs between flexibility, portability, footprint and performance. Their efforts bridge to the Constrained Device Edge, facilitate local data processing and in turn relay data to and from higher edge tiers.

Home Edge is focused at the Smart Device Edge for consumer use cases in the home.

The State of the Edge project spans the entire edge computing continuum, conducting research and producing free reports on edge computing and related topics. The project also oversees the Open Glossary of Edge Computing, which seeks to be an industry-wide lexicon for edge computing as well as a tool to align terminology across all LF Edge projects. Finally, the project maintains the LF Edge Interactive Landscape, which is a database-driven taxonomical landscape of edge-related vendors, organizations, projects, standards and technologies.

LFE will add more projects over time with a philosophy of being inclusive but also offering structure and promoting increasing harmonization. Per the project mission, the community aims to develop common best practices and eventual unification of APIs as appropriate. The result will be an open ecosystem for edge computing with infrastructure that can be context-aware of the needs of workloads running above, regardless of who wrote them. As an example, imagine a world where infrastructure could prioritize QoS for a healthcare app running right next to one that delivers entertainment content.

3.3 For more Information on LFE Projects

For more information on LFE projects, refer to their respective websites:

- Akraino Edge Stack.
- Baetyl.
- EdgeX Foundry.
- EVE.
- Fledge.
- Home Edge.
- Open Horizon.
- State of the Edge.

4 Summary

The concept of edge computing promises exciting new revenue opportunities resulting from the delivery of new types of services to new types of customers, in both consumer and enterprise segments. Compelling use cases include applications such as industrial IoT, computer vision, augmented reality, retail, gaming and assisted driving.

The Linux Foundation (LF) and its subsidiary organization LF Edge (LFE) have initiated a range of open-source software projects that enable companies of all types to collaborate around solutions for developing, deploying and monetizing edge applications and services. Recognizing the compelling business potential that results from extending cloud computing to the edge of the network, hundreds of developers from industry-leading organizations worldwide are participating in these projects that result in edge-optimized solutions for orchestration, management cloud services, frameworks and more.

This white paper has provided an overview of the architectures, use cases and LFE projects associated with edge compute. In-depth technical information is available via the individual projects' websites and interested developers are encouraged to participate in the LFE community.

The Join page on the LFE website provides information on joining LFE, explaining the processes for both existing LF members and non-members. There's also a link to an Inquiry page where interested parties can ask specific questions and obtain additional information.