LF EDGE

WHITE PAPER

Sharpening the Edge II: Diving Deeper into the LF Edge Taxonomy and Projects

JUNE 2022

Table of Contents

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.

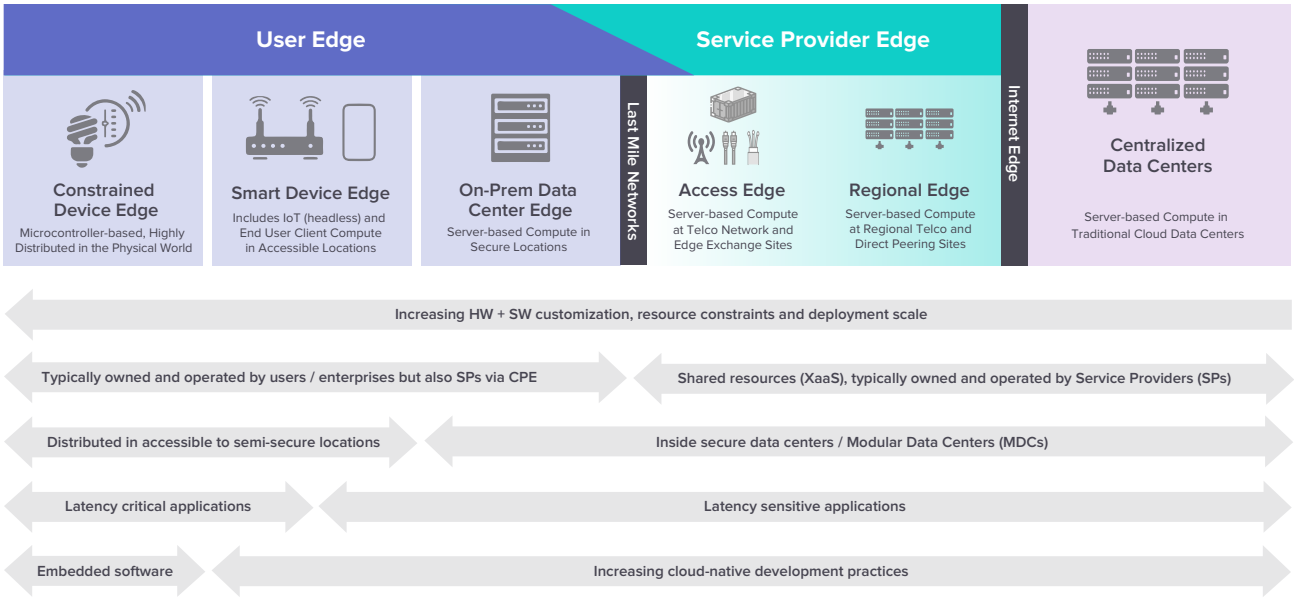


Figure 1. LF Edge Taxonomy

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.

LF Edge: What and Why

The Linux Foundation's LF Edge (LF Edge) 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.

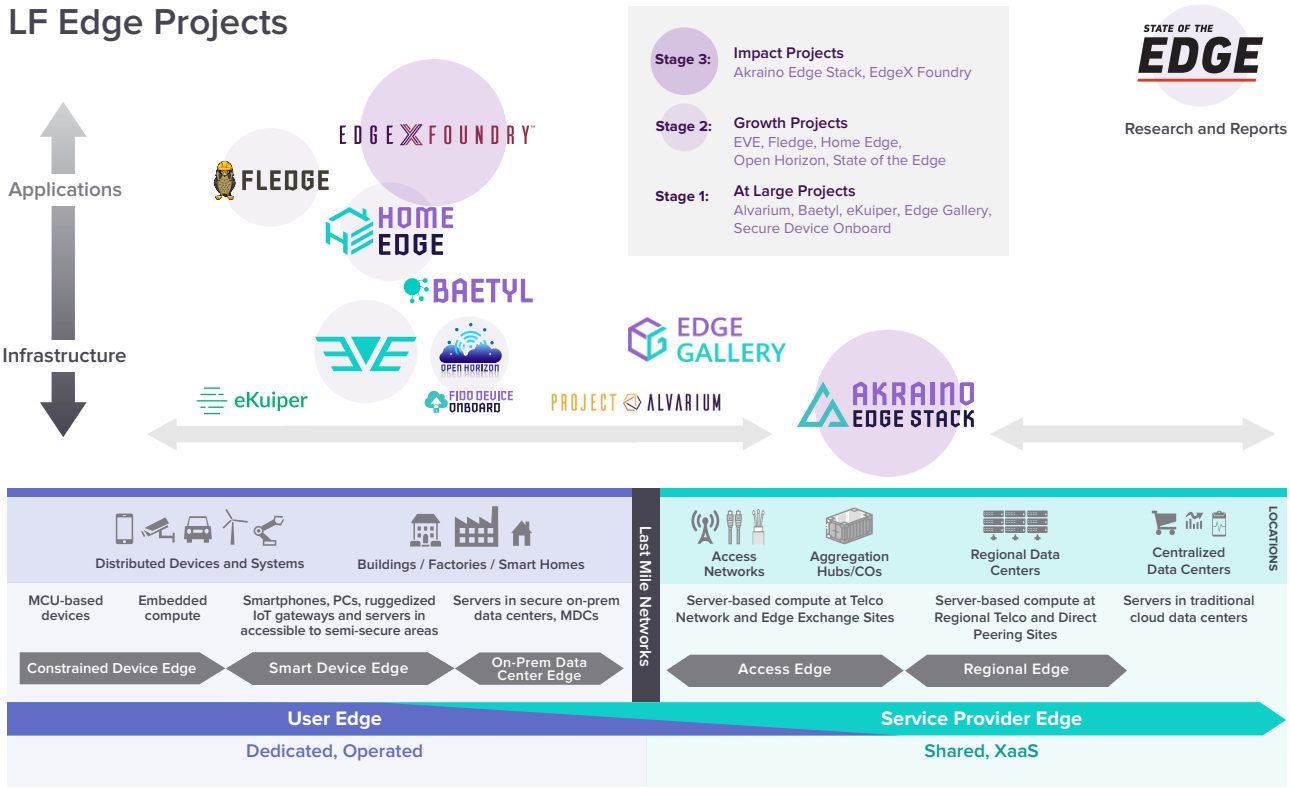


Figure 2. LF Edge Project Landscape

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 to 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.

"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 most elements enabling a use case run entirely on-prem, or the solution heavily 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 leverage 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.

"Whether most elements enabling a use case run entirely on-prem, or the solution heavily takes advantage of the cloud for computing scale depends on a combination of use case and risk profile."

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 certain functions such as 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."

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, integrating heterogeneous components, and more. Important to note is that edge native architectures do not mean that the cloud is not comprehended; rather, the solution is architected to prioritize on-premise 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, collaboration on standardization and interoperability is 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 spread the coding of components across different developers, and to 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.

"Organizations tend to approach the edge continuum either by expanding up from traditional OT systems 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/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 systems 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. Internet of Things (IoT) solutions are enabling the connection of formerly isolated operations for increased visibility and prescriptive analytics that yield improvements such as increased efficiency, quality and safety. Despite providing clear ROI, IoT solutions must comprehend the unique challenges of OT environments. Example challenges include lack of electrical power, low/unreliable connectivity bandwidth, dirt, heat, humidity, equipment with 20-30 year lifespans, regulations concerning safety and security, and proprietary systems from hundred year-old suppliers. OT users also 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 or computer scientists who are comfortable with the latest data center tools and cloud-native application development.

On the other hand, IT administrators usually are not vertical industry subject matter experts. They typically do not know the operational, scientific and business aspects of industrial operations, even though they may be 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 infrastructure and applications (e.g. 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.

Today, the "OT Up" approach to edge deployments often leverages lighter "gateway" class hardware and sensors deployed immediately above traditional industrial control systems. This approach modernizes edge infrastructure with IoT technology to normalize various IP and non-IP communication protocols into a standard IP-based format (e.g. MQTT, OPC-UA) so that data can be sent over a network. These edge devices may also buffer data in a local database for data persistence regardless of backend connection status, and they may perform light local analytics via a rules engine or an AI inference model. Software running on these edge nodes may or may not be built with cloud-native principles in mind.

Meanwhile, an "IT Down" approach to edge computing involves extending cloud and 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 while providing maximum agility

"Both OT operations and IT infrastructure 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."

and flexibility at the edge and making sure to protect critical industrial operations and systems.

Using similar logic, Gartner has coined the terms "Edge In" and "Cloud Out". "Edge In" is similar to the "OT Up" concept in that 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. "Cloud Out" aligns with the "IT Down" approach in terms of making data center-based services (or at least components of them) closer to where the consumers of these services are.

The two approaches to edge computing must not be treated as mutually exclusive. Both OT operations and IT infrastructure 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 lacks context. Imagine the MLOps challenges when this data situation is your starting point.

"OSS software is not about free, rather freedom and time to market."

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, rather freedom and time to market. Like TCP/IP became networking's foundational technology not a differentiator, OSS does 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 often 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 distribut-

ed 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, for example, 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 about 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.

> "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 Four Main Paradigms for Edge Management and Orchestration

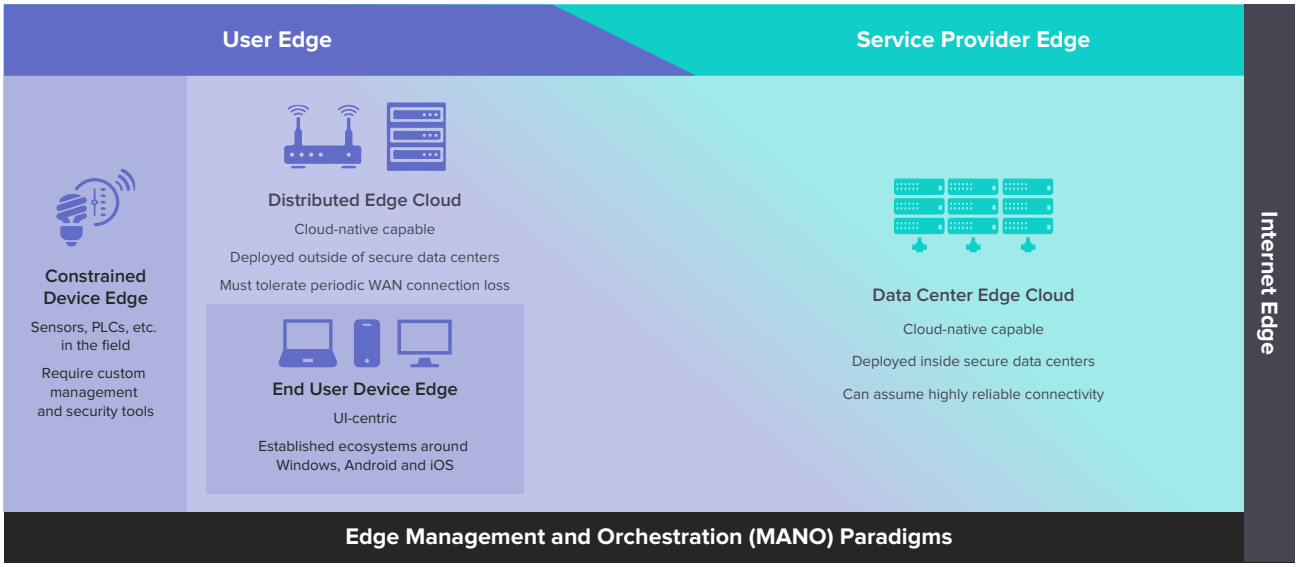


Figure 3. The Four Main Paradigms for Edge Management and Orchestration (placeholder graphic)

Spanning the edge continuum from the User to Service Provider Edge, there are four main paradigms for Edge Management and Orchestration (MANO): the Constrained Device Edge, End User Device Edge, Distributed Edge Cloud, and Data Center Edge Cloud (Figure 3). While each of these general paradigms may share similar principles, they necessarily require different tools for management and orchestration, security, connectivity and analytics due to inherent technology constraints and differences in ecosystem maturity. Driving factors include hardware resource constraints, whether a use case is UI- 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, whether the use case is latency-critical or latency-sensitive, and how reliable the connection is between the edge node(s) and centralized infrastructure.

DATA CENTER EDGE CLOUD

The Data Center Edge Cloud MANO paradigm involves edge resources deployed in Regional and Access Edges and the On-Prem Data Center Edge (as defined in the LF Edge taxonomy), and borrows heavily from traditional data center tools and practices for manageability, orchestration and security. Administrators can count on computing resources in these environments being physically secure, having a well-defined network perimeter (e.g. protected by firewall), and a constant connection (typically fiber) to their orchestration console. However, we are seeing some evolution in management, orchestration and security tool sets with the adoption of cloud-native principles, proliferation of Kubernetes, increasing scale of distributed data centers, adoption of software-defined networking, and so forth.

DISTRIBUTED EDGE CLOUD

The Distributed Edge Cloud MANO paradigm involves telemetry-centric edge computing resources deployed at the Smart Device Edge that are capable of supporting cloud-native architecture but located 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, to the fringes of the On Prem Data Center Edge. These edge nodes 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 cloud to the Distributed Edge Cloud MANO 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.

"The goal for Distributed Edge Cloud MANO is to extend the public-cloud experience to highly distributed edge locations while taking into account the unique needs in areas such as zero trust security and zero touch provisioning."

The goal for Distributed Edge Cloud MANO is to extend the public-cloud experience to highly distributed edge locations while taking into account the unique needs in areas such as zero trust security and zero touch provisioning. These tools serve to abstract hardware complexity using virtualization and containerization technologies to simplify the developer experience by exposing virtual resources (e.g. CPU, memory, storage, networking) for edge applications. While the operating model is similar to the data center, including enabling CI/CD, Distributed Edge Cloud MANO and security solutions need to take into account that the computing resources are typically often more CPU and memory constrained (for example, having as low as 1GB of available system memory), that they are physically-accessible to hackers, they are often deployed on untrusted networks, and they are likely to periodically lose connectivity to centralized resources. As such, MANO solutions optimized for the traditional data center are typically not suitable for all Distributed Edge Cloud use cases.

END USER DEVICE EDGE

Also at the LF Edge taxonomy's Smart Device Edge are end user devices (e.g. PCs, Smartphones, Tablets) that are UI-centric. The End User Device Edge MANO 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, compared to 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 account is hacked.

CONSTRAINED DEVICE EDGE

On the far extreme of the continuum is the Constrained Device Edge MANO paradigm, characterized by lightweight devices and sensors in the physical world. These devices leverage microcontrollers and have kilobytes to megabytes of memory, rendering them only 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 device management and security tools due to the resource constraints. Software and firmware updates tend to be monolithic in nature.

Immutable vs. Mutable Edge Resources

"Spanning the public cloud to the Distributed Edge Cloud, computing resources are almost always considered immutable, general-purpose and organized as a pool of resources defined by a configuration."

When moving towards the Constrained Device Edge, the resources to be managed are not necessarily immutable. Spanning the public cloud to the Distributed Edge Cloud, computing resources are almost always considered immutable, 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 resources as immutable (for example creating a super computer based of Raspberry Pi's) but the resources can also be mutable, 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 the physical world is the difference between the macro and micro level of a given edge resource. For example, a vehicle 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 and data.

"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 and data."

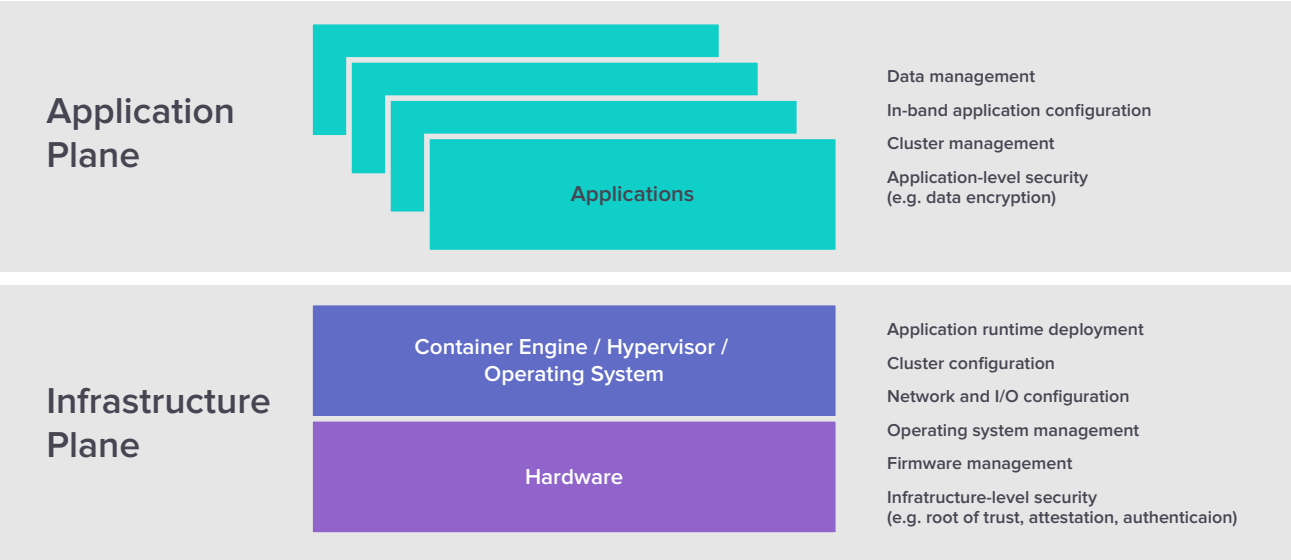


Figure 4. The Infrastructure and Application Planes

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

When dealing with end user and 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.

"It will be most likely that edge solutions will rely on an "orchestrator of orchestrators" model that aggregates underlying management and security tools that are tailored for the needs of each edge MANO paradigm."

We will see increasing coordination between tool sets for deploying, managing and securing edge infrastructure and applications spanning the edge continuum over time, however due to the inherent tradeoffs across the four main edge MANO 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 for admins and developers is to have a "single pane of glass" that streamlines the workflow for all of their operations, it will be most likely that edge solutions will rely on an "orchestrator of orchestrators" model that aggregates underlying management and security tools that are tailored for the needs of each edge MANO paradigm.

Project Contributions for Edge Management and Orchestration

The following is an overview of LF Edge project efforts in the functional area of edge MANO. 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.

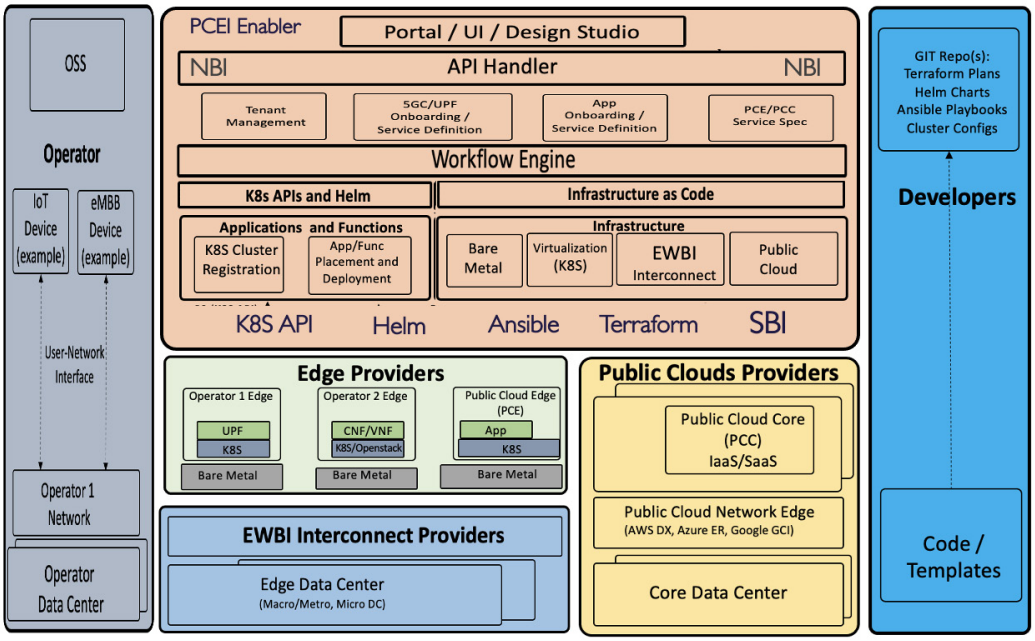
AKRAINO

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.

"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."

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.



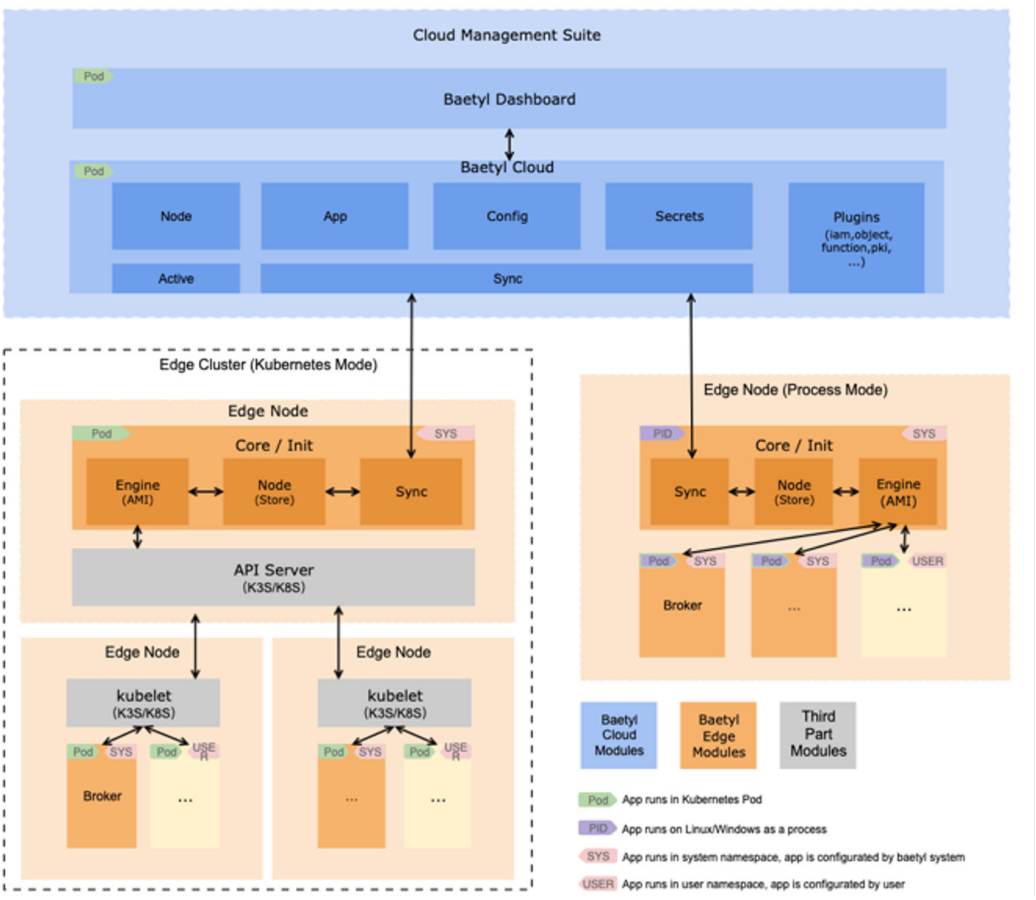
Akraino PCEI Blueprint Architecture

BAETYL

The Baetyl project aims to seamlessly scale applications and data from the Public Cloud to the Distributed Edge Cloud, 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.

"In Baetyl, applications are configured and managed by users in the cloud and then run at the edge."



Baetyl Architecture

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.

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,

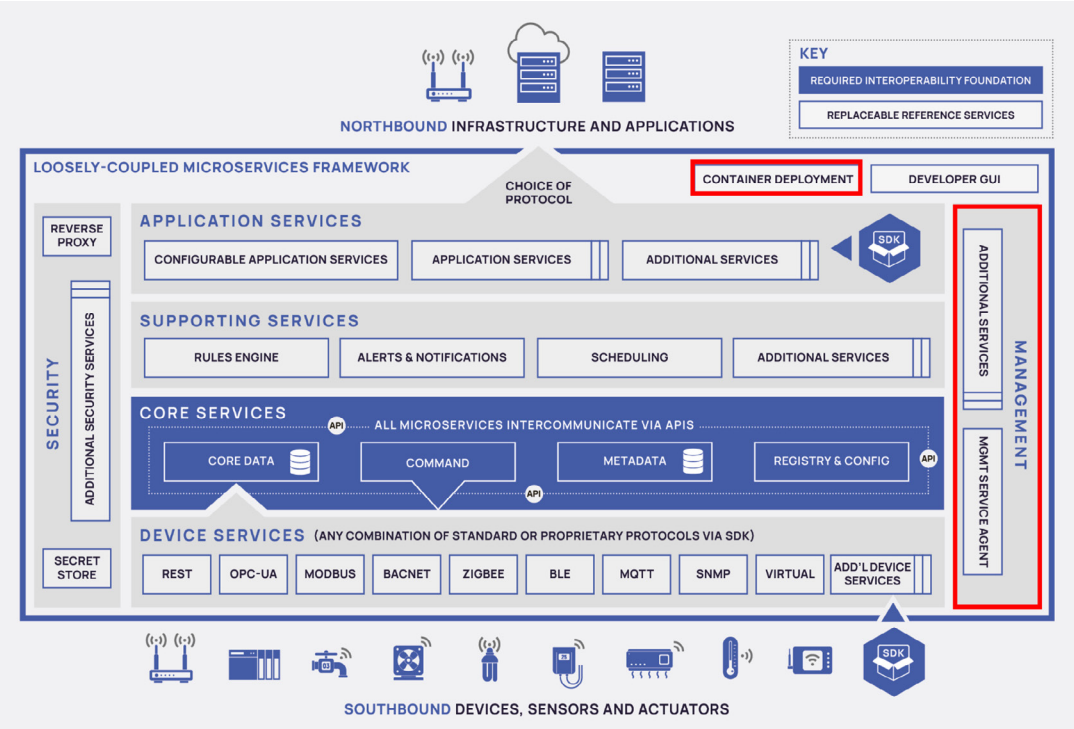
confidential information such as passwords and certificates, and policies for data storage. Applications are specified exactly which edge instances they need to be deployed to in the form of tag matching and are eventually translated into resources such as Pod/Service/ConfigMap for Kubernetes to run 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.

EDGE X 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 System Management Services

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 IoT use cases including different connectivity protocols, security and system management for widely-distributed compute nodes and scaling down to more constrained devices at the Distributed Edge Cloud. 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.

"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."

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).

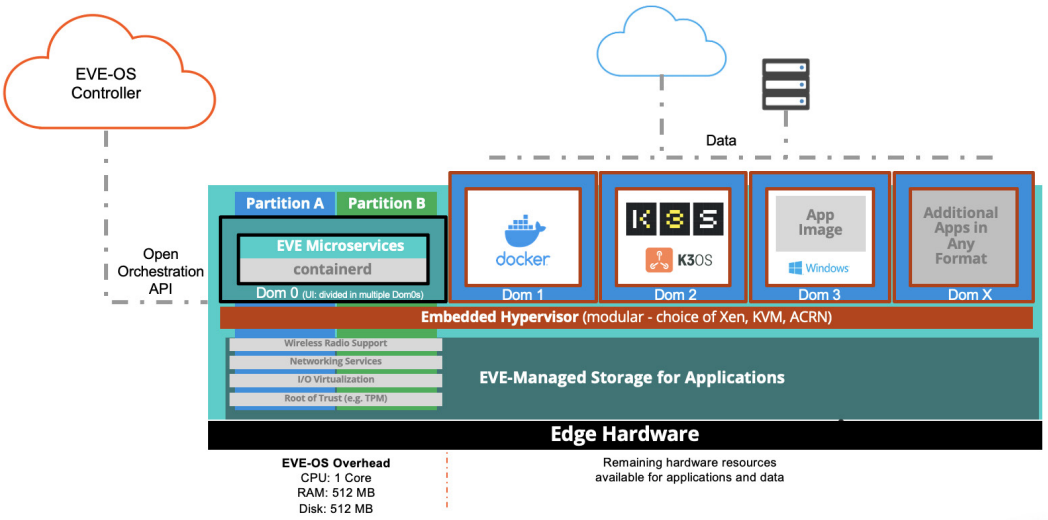
There are two services that provide the EdgeX Foundry system management capability.

- 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.

Project EVE is building EVE-OS which is a universal, bare metal operating system optimized for Distributed Edge Cloud MANO, blending principles from the data center with specific capabilities in areas such as zero trust security and zero touch provisioning. 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 outside of traditional data centers. 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 Regional Cloud Edge MANO paradigm above.



EVE-OS Architecture

"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 and presenting these resources to applications."

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.

The operating system 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 OT and IT networks.

EVE-OS is also designed to expect that distributed 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 rolls 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, formerly Secure Device Onboard) is a device onboarding scheme from the FIDO Alliance, sometimes called "device provisioning". It is most beneficial for IoT devices and distributed edge computing nodes due to the scale and diversity of these resources but it is applicable to all of the edge MANO paradigms.

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, allowing for a more efficient supply chain."

A unique feature of FDO is that the device owner can select the IoT platform at a late stage in the device life cycle, allowing for a more efficient supply chain. 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.

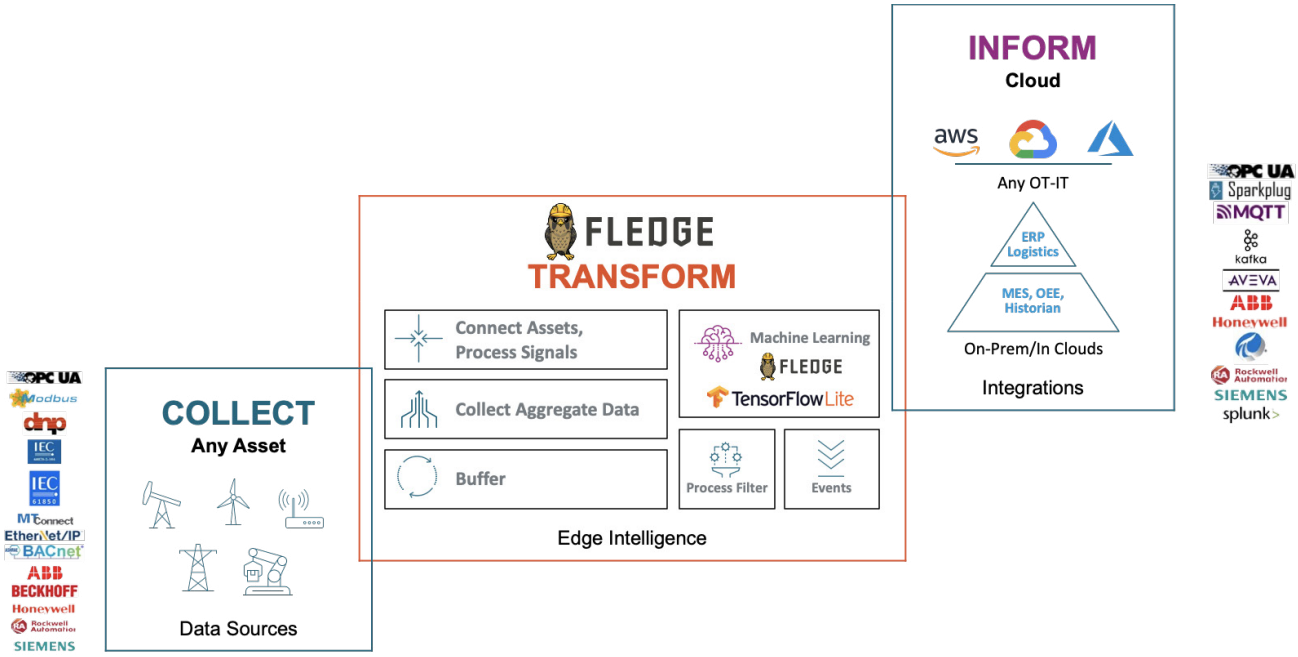
Project Fledge is building an open community-driven IIoT platform focused on the data, data pipeline and application layers unique to industrial use cases. Fledge's pluggable microservice architecture is designed to collect and aggregate data from any machine, sensor or protocol, filter/transform data 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.

> "From a user perspective the ability to reuse, configure and deploy intelligent pipeline logic can be done without writing code from the Fledge UI."

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.



Fledge Solution Architecture

HOME EDGE

The Home Edge Project, seeded by code from Samsung Electronics, concentrates on driving and enabling a robust, reliable, and intelligent edge computing frame-work, platform and ecosystem running on a variety of home devices. 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.

> "The Home Edge Project concentrates on driving and enabling a robust, reliable, and intelligent edge computing frame-work, platform and ecosystem running on a variety of home devices."

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.

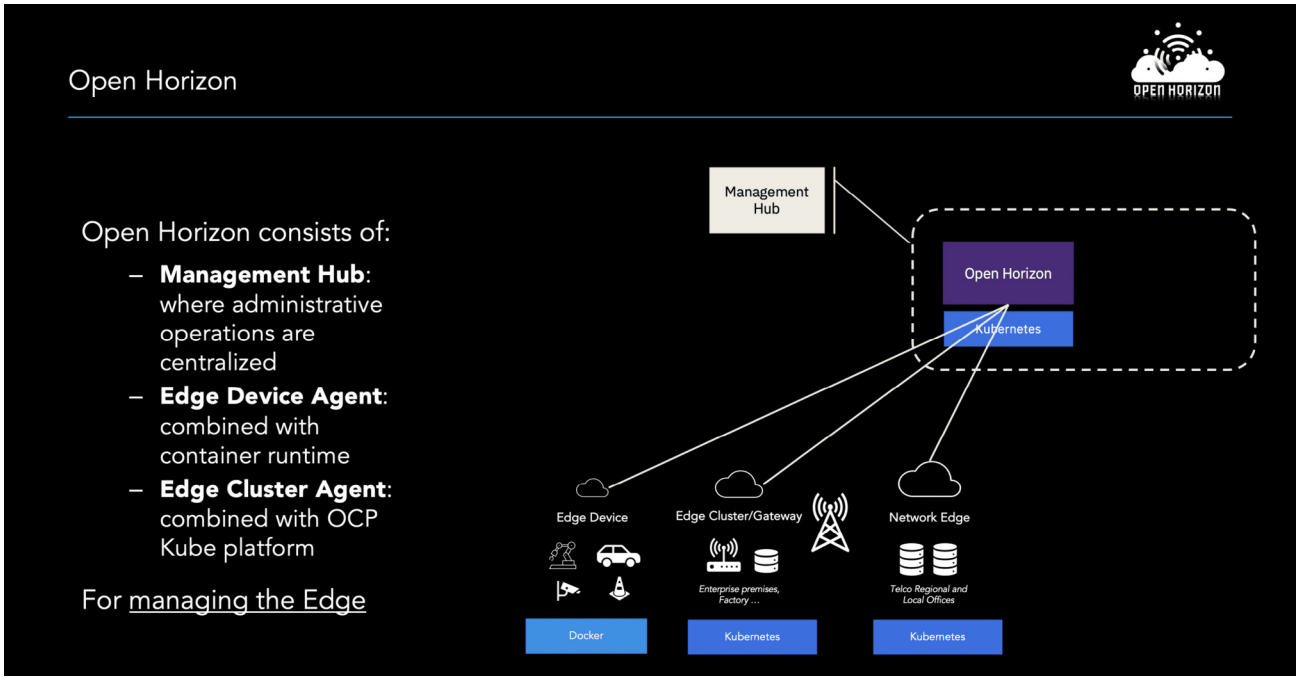
Service orchestration, data storage/retrieval, cloud synchronization based use cases can be developed using the Home Edge. Docker based container releases are being done in the Docker hub frequently for ease of use.

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 Open Horizon agent supports use cases that span all four main edge MANO paradigms."

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 main edge MANO paradigms: Data Center Edge Cloud, Distributed Edge Cloud, End User Device Edge, and Constrained Device Edge. 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. FIDO Device 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.

"A common misconception is that open source code is less secure than proprietary software, however this simply isn't the case."

Unique Security Challenges at the Edge

"Nodes at the User Edge require careful attention to unique security challenges because they are typically deployed outside of secure data centers.

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. in the Data Center Edge Cloud MANO paradigm) tend to be quite similar to traditional data centers. Meanwhile, nodes at the User Edge (e.g. spanning the Distributed Edge Cloud to the Constrained Device Edge MANO paradigms) require careful attention to unique security challenges because they are typically deployed outside of secure data centers in locations such as the factory floor, retail stores, inside wind turbines, on trucks and within rooftop HVAC systems, to name a few.

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

SCALE

Part of the value of edge computing and IoT stems from having numerous edge nodes 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.

"Security practices for edge solutions that bridge OT and IT systems need to strike a balance between these different priorities."

LACK OF PHYSICAL AND NETWORK PERIMETERS

Edge computing solutions distributed in the field often have 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 it. 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, distributed edge computing 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 and data scientists) 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. As such, security practices for edge solutions that bridge OT and IT systems need to strike a balance between each organizations' different priorities.

UNATTENDED OPERATION

Unlike end user devices, 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 leveraging zero trust architecture, it is advisable 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 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

"All released blueprints have passed a vulnerability scanning process implemented by Akraino's security subcommittee."

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. The Akraino 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.

PROJECT 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."

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.

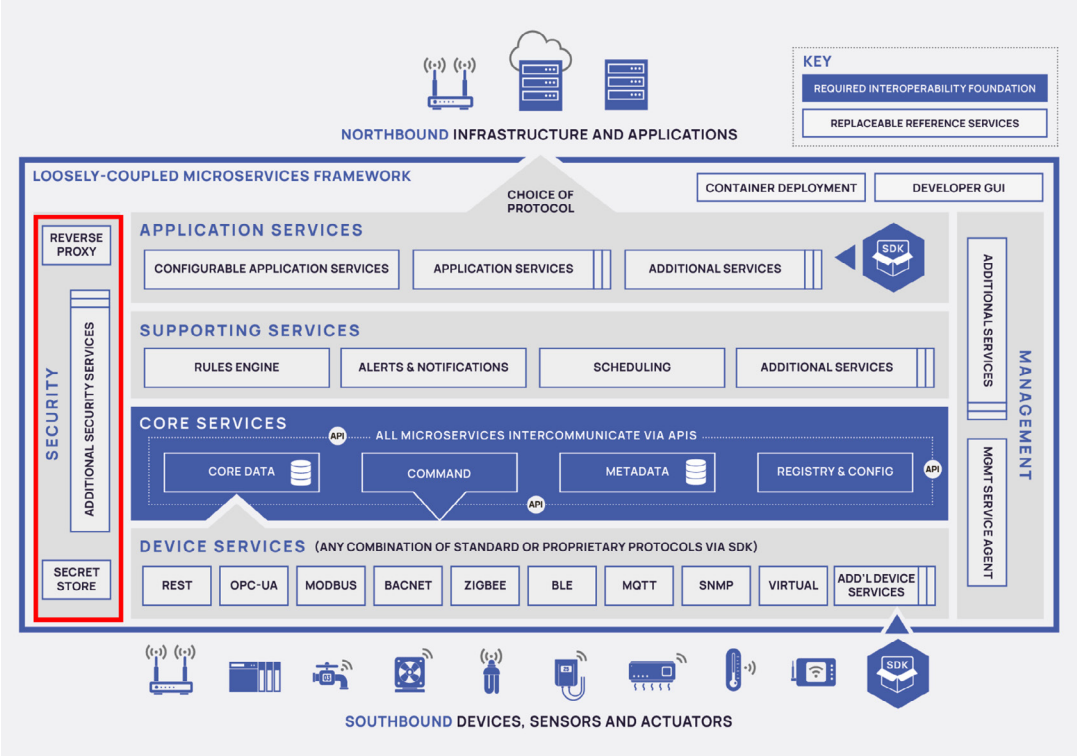
"Security is one of the most important parts of Baetyl, and the whole security system consists of three parts: connection security, service security, and device security."

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.

EDGE X FOUNDRY™



EdgeX Foundry Security Services

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 application framework security features are also built on a foundation of open interfaces and pluggable, replaceable modules.

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)

"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."

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-OS features a state-of-the-art zero trust security architecture optimized for the Distributed Edge Cloud MANO paradigm in that it assumes deployed nodes are physically-accessible and do not have a defined or trusted network perimeter. Features include:

"EVE-OS features a state-of-the-art zero trust security architecture optimized for the Distributed Edge Cloud MANO paradigm."

- **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 application stacks deployed above). 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 incidents 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.

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, Home Edge received the OpenSSF Best Practices "Gold" badge through the Scorecard system, indicating a high level of security development.

"Home Edge received the OpenSSF Best Practices "Gold" badge through the Scorecard system, indicating 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 currently implements zero-touch deployments through an embedded instance of FDO."

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 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, 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 in the data center but there is an increasing mix of non-IP transports to contend with as you approach the Constrained Device Edge. In the IoT

"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. "

and Industrial worlds, many systems communicate over legacy local area networks such as 4-20mA current loops, serial and CAN Bus, as well as modern 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, with examples including Modbus, BACnet, PROFINET and Ether-CAT. 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.

Project Contributions for Edge Connectivity

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

AKRAINO

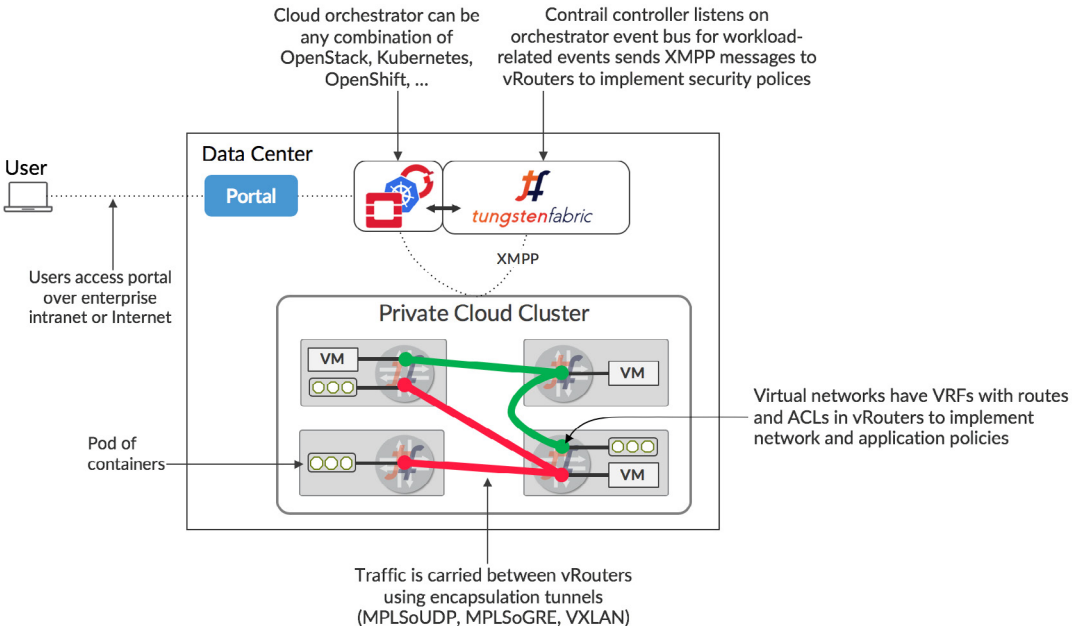
Akraino blueprints can provide an end to end EdgeStack to support Virtualized Network Elements (NFVI) per Open-RAN (O-RAN) requirements. The Akranio project advocated collaborating with O-RAN Alliance's specification workgroup 6 (six) responsible for cloud specifications, to align with and publish multiple 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, the Network 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.

"The Network 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."

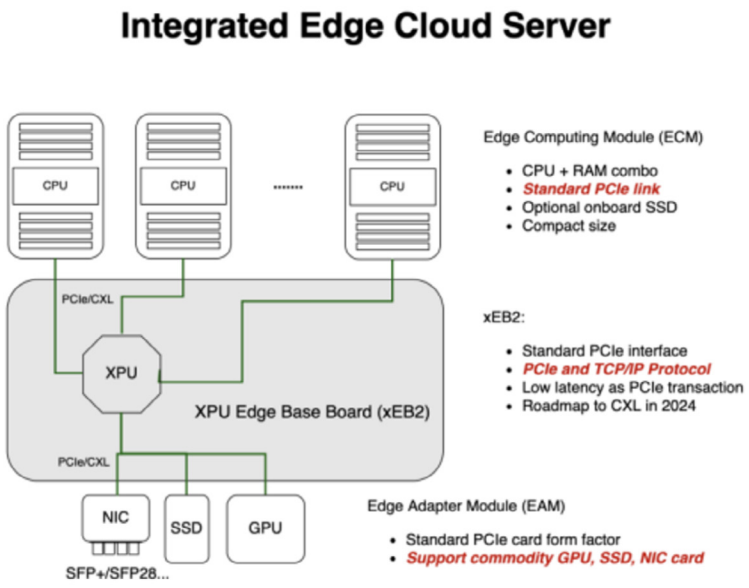
Another example of an innovative Akraino blueprint is the Integrated Edge Cloud (IEC) for compact edge with PCIe networking and Cloud-on-Board (CoB). In contrast to centralized data centers, the networking in the compact integrated edge cloud needs to be reconsidered, as the networking challenge shifts from solving for the scale-out to a massive number of connected servers to enabling the scale-in for connecting a small number of servers in an edge location. The time-honored methods of increasing the port density in a switch or utilizing the high throughput NIC won't work in a small scale cluster with less than 32 servers. Hence a novel way to rebuild the networking architecture for the integrated

edge cloud is needed not only in terms of cost but also, and even more importantly, due to energy constraints, given the expected large number of deployments of the integrated edge cloud sites.



Akraino Network Cloud with Tungsten Fabric (TF) Blueprint

Due to advances in PCIe networking, the system-on-board (SoB) connectivity and the cloud clustering technologies can be unified into one single and simple solution, which we named the Cloud-on-Board (CoB) Architecture. In the CoB architecture, we can connect CPUs directly without additional adapters. The CoB has three major components: 1) Edge Computing Module (ECM): CPU+RAM+OS SSD combo, 2) Edge Base Board (EB2): for PCIe Data Fabric, 3) Edge Adapter Module (EAM): PCIe-compatible Device, such as GPU, NIC, SSD etc.



Akraino Integrated Edge Cloud (IEC) for Compact Edge with PCIe networking and Cloud-on-Board (CoB)

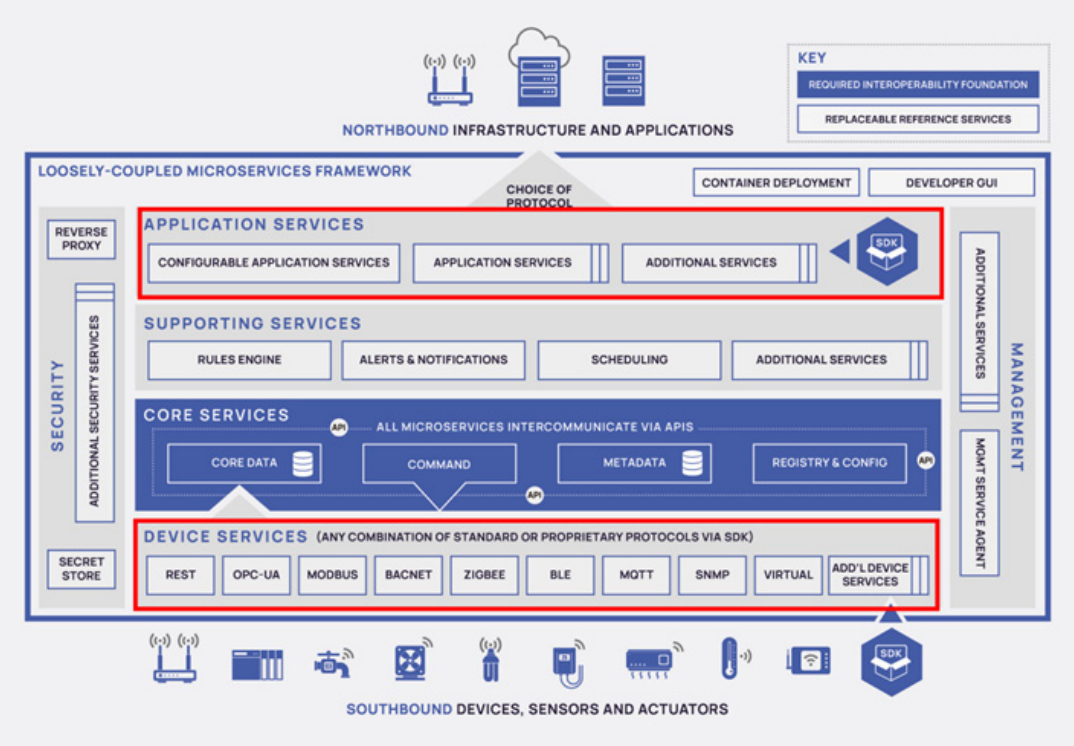
EDGE X FOUNDRY™

The inherent heterogeneity and complexity at the edge is best illustrated in relation to connectivity and interoperability requirements, both southbound and northbound.

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

- **Southbound:** EdgeX Foundry provides reference implementations for key IoT and legacy (brownfield) OT 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 (e.g. AWS IoT, Azure IoT, Google IoT Core) 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 OT, enterprise and IT environments.

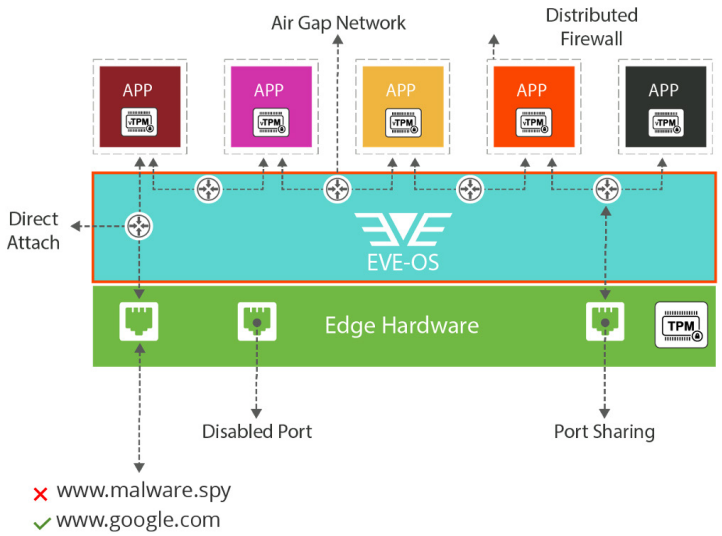


EdgeX Foundry Connectivity Service

EVE provides extensive networking functionality for Distributed Edge Cloud MANO 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 workloads 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 EVE API provides admins with the ability to granularly assign I/O to applications, both dedicated to specific workloads and shared among two or more workloads."

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 an IIoT platform. Its primary function is intelligent pipelines that move data from sources to destinations for OT."

Fledge is Architected for OT Data

Fledge is architected to unify the diversity of OT users, applications, and data sources, types, formats and destinations at the edge to support the challenges of physical distribution of sensors, machines, processes and factories that cannot run in the cloud. It enables OT applications and data pipelines from any machine edge to any backend. The beginning and ending of flows is usually determined by physical location, latency, data volumes, application processing requirements and cost.

Fledge standardizes industrial data pipeline management (collection, transformation and integration) from sensors to clouds and uses a common open API for scalability, manageability, supportability and security. It supports most industrial protocols, data types and data transformations. If a protocol does not exist in the existing library, it generally takes two weeks to enable it. New data/registry mappings take less than a day. Automated schema translations typically take 1-2 days. Edge MLOps workflows 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

A drawback and potential bottleneck for deploying applications at 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 explicit-ly-defined individual depency. A service does not otherwise have a shared network connection with all other services in a deployed application.

"Open Horizon secures and simplifies the internal networking between dependent 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.

"Where analytics 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."

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 almost always be driven proximal to the user or process in the field, for example deploying a vehicle's airbag 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 analytics 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 MANO 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 MANO 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 spanning regional edge data centers to 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 for end user devices.

"The LF Edge projects provide stakeholders from IT and OT with robust remote MANO 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."

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

"The Akraino Blueprint "Robot Basic Architecture Based on SSES" 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."

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.

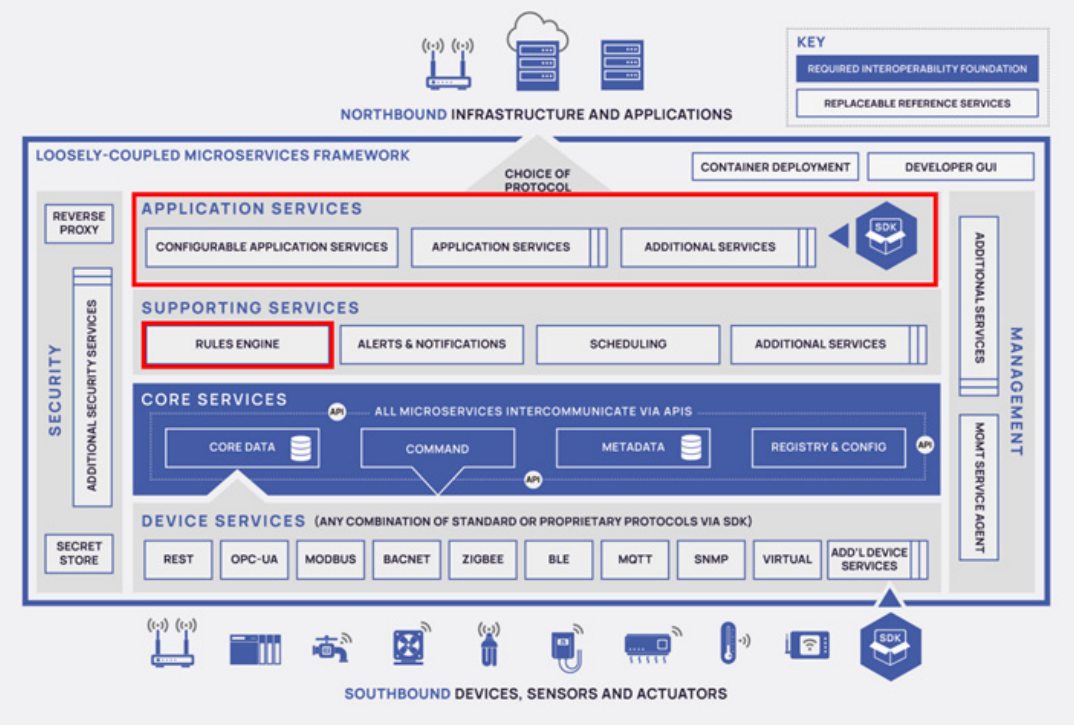
PROJECT ALVARIUM

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.

"Alvarium aims to be a key enabler for driving value from edge data by introducing confidence in that data."

EDGE X FOUNDRY™

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.



"The EdgeX message bus architecture allows sensor data to easily be streamed to any analytics package."

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. With eKuiper, users can realize fast data processing on the edge and write rules in SQL.

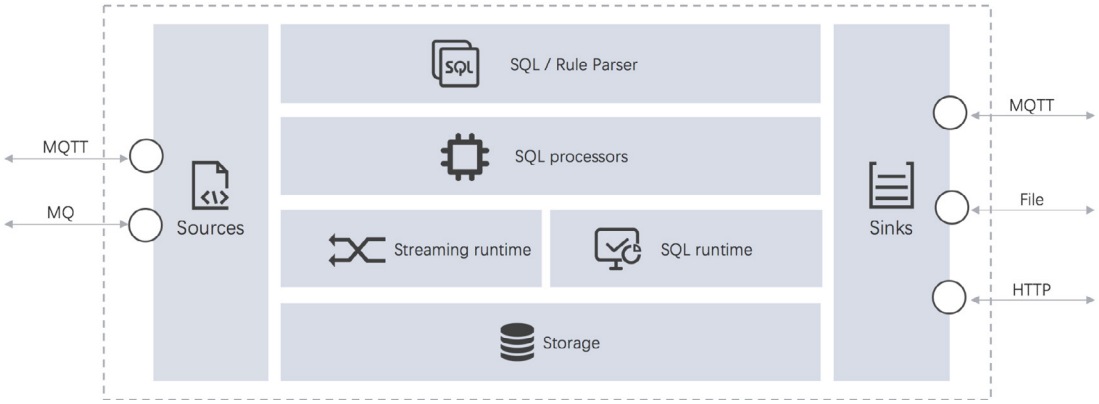
More broadly, an analytics service could be some simple logic built into an application service, a rules engine package, or an agent of some artificial intelligence/machine learning system.

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 optimized for many types of resource-constrained edge devices. A key goal of eKuiper is to migrate the cloud streaming software frameworks (such as Apache Spark, Apache Storm and Apache Flink) to the edge. eKuiper references these cloud streaming frameworks, and also considers special requirements of edge analytics, and introduces a rules engine, which is based on Source, SQL (business logic) and Sink that can be used for developing streaming edge applications. It can be leveraged in various IoT edge use cases, such as real-time processing of production line data, analyzing Connected Vehicle data in real real time, and real-time analysis of urban facility data in smart city scenarios. eKuiper edge processing can reduce system response latency, save network bandwidth and storage costs, and improve system security.

"eKuiper is an IoT data analytics or stream processing engine optimized for many types of resource-constrained edge devices."



eKuiper Architecture

eKuiper is lightweight and highly efficient, optimized for resource-constrained devices with high throughput processing. It is platform agnostic , deployable on x86, ARM and PPC CPU architectures with 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 destination. 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 and it supports data extract, transform and filter through SQL syntax. eKuiper also supports streaming concepts like time windows and 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 provides flexibility for deploying analytic applications. Text-based rules are used for business logic implementation and deployment through a REST-API.

"A major benefit of EVE-OS' 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 on edge hardware."

EVE-OS is not in the data path, rather it is a highly secure operating environment to deploy any kind of distributed edge 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 on 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.

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.

"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."

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 features a unique component called Model Manager composed 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.

"Open Horizon's Model Manager component enables bi-directional synchronization of machine learning assets and related files separate from any containerized applications that may consume those analytics."

State of the Edge

"State of the Edge 2022 addresses three aspects shaping the development of edge computing: connectivity, application infrastructure, and location."

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.

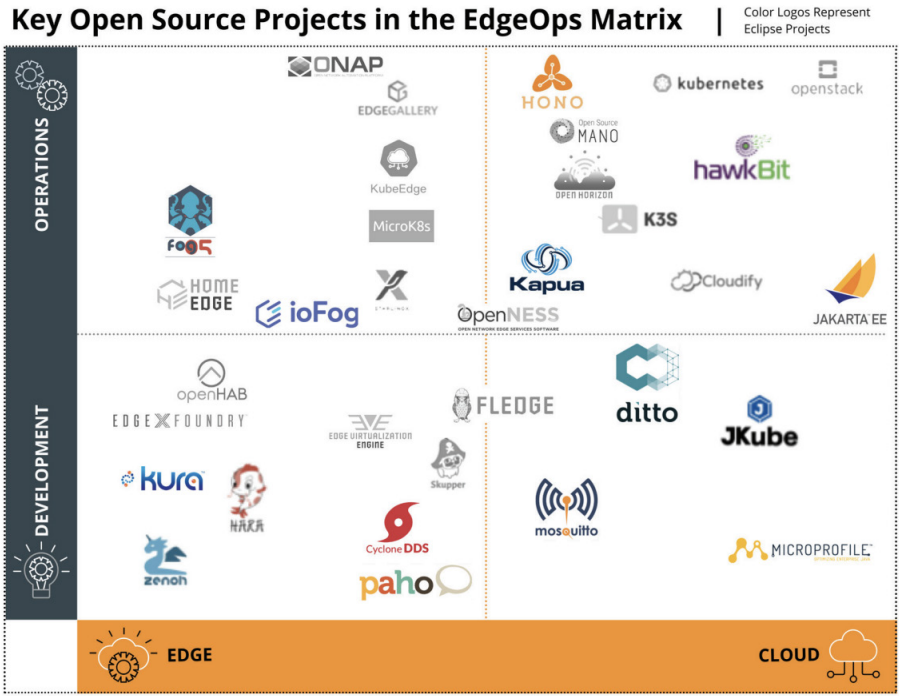
Industry Collaboration

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 an 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.

The LF Edge community is also prioritizing collaboration 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.

"Edge computing takes a village and membership and participation in LF Edge provides access to collaborations within the broader edge computing landscape."



Source: 2021 Eclipse Foundation EdgeOps Whitepaper

Edge computing takes a village and membership 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.