LF Edge Architecture Whitepaper V2 (2022) - Working Draft

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

Milestones:

M0	Confirm outline	COMPLETE
M1	Share working draft for initial committee input	COMPLETE
M2	First round of committee input due	April 19, 2022
M3	Second round of input due	April 29, 2022
M4	Proofread and reorg	May 10, 2022
M5	Proofread and refine and polishing	May 24- 28, 2022
M6	Delivery to Creative Services	May 31, 2022
M6	Final review / approval of layout	June 14, 2022
M7	Publishing `	Week of June 20

Work towards OSS Summit and Edge Computing week of June 20

Table of Contents

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

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

Start of New Paper

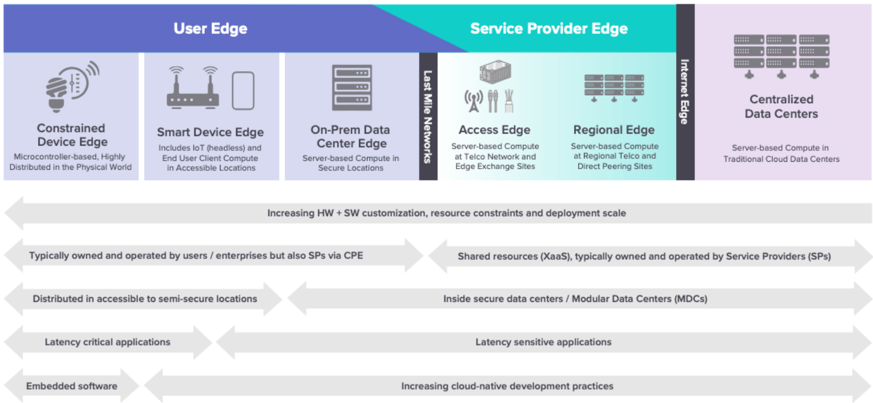
Insert TOC

Introduction

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

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

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.


User Edge
Service Provider Edge
Constrained Device Edge
Microcontroller-based, Highly Distributed in the Physical World
Smart Device Edge
Includes IoT (headless) and End User Client Compute in Accessible Locations
On-Prem Data Center Edge
Server-based Compute in Secure Locations
Last Mile Networks
Access Edge
Server-based Compute at Telco Network and Edge Exchange Sites
Regional Edge
Server-based Compute at Regional Telco and Direct Peering Sites
Internet Edge
Centralized Data Centers
Server-based Compute in Traditional Cloud Data Centers
Increasing HW + SW customization, resource constraints and deployment scale
Typically owned and operated by users / enterprises but also SPs via CPE
Shared resources (XaaS), typically owned and operated by Service Providers (SPs)
Distributed in accessible to semi-secure locations
Inside secure data centers / Modular Data Centers (MDCs)
Latency critical applications
Latency sensitive applications
Embedded software
Increasing cloud-native development practices

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, networking, IoT and analytics and highlights how each project is addressing these areas.

Recap of LF Edge: What and Why

The Linux Foundation's LF Edge (LFE) was founded in 2019 as an umbrella organization to establish an open, interoperable framework for edge computing 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~~Committee~~ (TAC) that helps align project efforts and encourages structured growth and advancement by following the Project Lifecycle Document (PLD) process. All new projects enter as Stage 1 “At Large” projects which are projects that the TAC believes are, or have the potential to be, important to the ecosystem of Top-Level Projects, or the Edge ecosystem as a whole. The second “Growth Stage” is for projects that are actively developing their community of contributors, governance, project documentation, and other variable~~interested in reaching the Impact Stage~~, and have identified a growth plan for doing so. Finally, the third “Impact Stage” is for projects that have reached their growth goals and are now on a self-sustaining cycle of development, maintenance, and long-term support.

[below copy is from v1]

Macro Trends

Intro

Open Source Driving Standards

OSS collaboration driving interop through APIs, especially critical at the edge.   Application level,

Cloud-native Architecture

- PoV on ideal edge tech stack
- Key tenets and architectural concepts, e.g. Loosely coupled, microservice, decouple edge and cloud resources.
-

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 IT organizations and what has become clear since is that organizations tend to either approach the edge continuum by expanding up from traditional (OT) in the physical world or down from the centralized IT data center. This can be viewed as “OT Up” vs. “IT Down” and each trajectory brings its own set of considerations.

OT is rooted in industrial processes within factories, refineries, buildings and beyond, and connecting their formerly isolated operations for increased visibility and prescriptive analytics is a key driver for IoT. OT environments and constraints are unique; lack of power, low/unreliable bandwidth, dirt, heat, humidity, equipment with 20-30 year life-spans, regulations concerning safety/security and proprietary systems from 100 year old suppliers.   OT users are mechanical, electrical, chemical, biological engineers and scientists, mechanics or operators without college degrees.   They are not computer scientists comfortable with the latest cloud tools or coding.

Meanwhile, an “IT Down” approach involves extending data center practices further in the physical world while recognizing the unique OT needs. This includes technologies and practices such as software-defined infrastructure, cloud-native software architecture, continuous integration and delivery (CI/CD) and more. A key goal for the IT Down approach is to extend cloud-native development principles as far into the continuum as possible, while also recognizing that there are inherent tradeoffs. This provides maximum agility to rapidly software-define new use cases to stay competitive.

large suppliers dominating vertical Today, “OT Up” edge deployments often leverage lighter “gateway” class hardware that modernizes existing infrastructure by performing functions such as normalizing various connectivity protocols into a modern standard (e.g. MQTT, OPC-UA), buffering data in a local database for data persistence regardless of backend connection status, and light local analytics spanning a rules engine to a machine learning inference model.

As these two trajectories intersect and blur we will Some shadow IT still exists but we’re also seeing more collaboration. Etc…

Cloud-Out versus Edge-=IIn

In the cloud-out approach, the idea is to make data center-based services (or at least components of them) closer to where the consumer of these services are.

In the edge-in approach, the idea is to perform software logic (business processes, abstraction, data cleanup, etc..) as quickly as possible to allow local actions to be taken without the need to go to the cloud.

The two approaches must not be treated as mutually exclusive.

These trends are expanded upon in the specific sections within this paper.

Linux in the Industrial World

- Why OSS and Linux in general is critical across the board
- Key considerations for OT/industrial, why Linux and OSS is so beneficial
- OSS isn’t about giving your IP away
- Etc…

Data Confidence

Ultimate goal is trusted data, attach confidence to data itself
Requires open infrastructure

- WHAT OTHER TRENDS TO CALL OUT?

Scaling Edge Deployments in the Real World

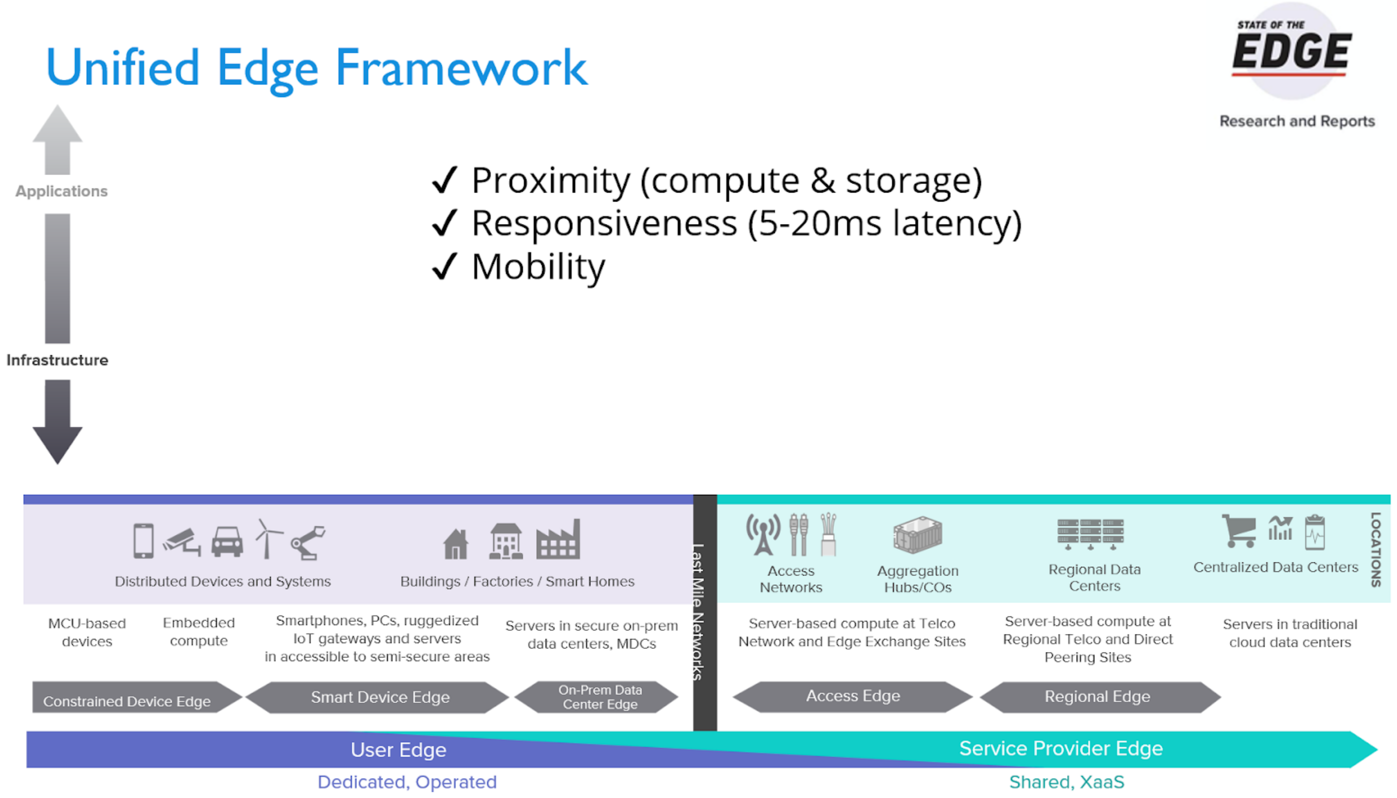
There has been much discussion on edge computing in recent years but there is also still quite a bit of confusion. The LF Edge taxonomy aims to reduce this confusion but…

Many PoCs. Starts with a use case, then applications and hardware. Once past a few PoCs the need for management and security becomes apparent.

The Four Main Paradigms for Edge Management

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

Insert graphic on the four paradigms



Unified Edge Framework

STATE OF THE EDGE

Research and Reports

✓ Proximity (compute & storage)
✓ Responsiveness (5-20ms latency)
✓ Mobility

Applications
Infrastructure
Distributed Devices and Systems
Buildings / Factories / Smart Homes
Last Mile Networks
Access Networks
Aggregation Hubs/COs
Regional Data Centers
Centralized Data Centers
LOCATIONS
MCU-based devices
Embedded compute
Smartphones, PCs, ruggedized IoT gateways and servers in accessible to semi-secure areas
Servers in secure on-prem data centers, MDCs
Server-based compute at Telco Network and Edge Exchange Sites
Server-based compute at Regional Telco and Direct Peering Sites
Servers in traditional cloud data centers
Constrained Device Edge
Smart Device Edge
On-Prem Data Center Edge
Access Edge
Regional Edge
User Edge
Service Provider Edge
Dedicated, Operated
Shared, XaaS

Metro and Regional Data Center Edge

The Metro and Regional Data Center paradigm borrows heavily from traditional data center tools and practices for manageability and security. However, we are seeing some evolution with the adoption of cloud-native principles, proliferation of Kubernetes and new tools to manage distributed data centers in growing scale. These data centers are entirely located within the Service Provider Edge as defined in the LF Edge Taxonomy.

Distributed Edge

The Distributed Edge involves telemetry–centric hardware that can span from a gateway device in a truck to a cluster of servers on a factory floor or retail store, bleeding into the fringes of the traditional data center. These edge nodes sit on the User Edge and are a form of “cloud” resources in that they provide services to end users, other proximal edge nodes and constrained sensors.

Similar to embedded devices, there is a wide array of hardware types and application needs due to varying environmental conditions, regulatory requirements and domain-specific use cases. However, a key difference between Distributed Edge hardware and Constrained Devices is that the former is capable of running Linux and supporting cloud-native principles including platform independence, hardware abstraction through containerization and virtualization, continuous delivery (e.g. CI/CD) and so forth.

This edge hardware requires more sophisticated edge orchestration to software define functionality over time through continuous delivery.

Client Device Edge

Also on the User Edge, Client Devices (e.g. PCs, Smartphones, Tablets) are typically dedicated to specific users and are UI-centric. Today there are well-established ecosystems around the likes of Windows, iOS and Android but there is a trend for cross-over between these devices and telemetry-centric edge computing. Different because they run applications that users interact with dynamically and can interpret language differences, unlike IoT devices that must be hard coded to work together. These devices also benefit from users being around to notice potential security issues, for example if your email is hacked.

Constrained Device Edge

On the far extreme of the continuum are constrained devices in the physical world with kilobytes to megabytes of memory that perform fixed functions. These devices don't have the resources (e.g. CPU, memory) to support an abstraction layer that enables cloud-native principles like containerization and inherently require highly customized management and security tools due to the resource constraints. Software and firmware updates tend to be monolithic.

When moving towards Client Device Edge and Constraint Device Edge the resources to be managed are not necessarily immutable. In the cloud, Metro and Regional Data Center Edge and Distributed Edge the resources are always considered immutable and generic and organized as a pool of resources defined by a configuration. For Client Device Edge and Constraint Device Edge, it is also possible to treat the resource as immutable (for eample creating a super computer based of Raspberry Pies) but the resources can also be mutable and typed (by ownership, geolocation, and function) and managed as ad-hoc clusters that are defined by discovery. And 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 phones 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.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 it is important to have them in the same context.

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

Infrastructure vs. Application Management

The edge is highly fragmented with a wide array of hardware, software, networks, and skill sets and this complexity increases 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 management 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 is of the underlying hardware and related compoute, networking and storage resources, the operating system, any virtualization and container technologies, and the deployment of any runtimes on top of the operating system. In contrast, application management involves the direct configuration of application runtimes. Infrastructure management is out-of-band of applications and data, whereas application management is in-band.

Graphic of the two planes

A key goal for projects within LF Edge is to maintain this separation in addition to architecture modularity into each plane to maximize flexibility. End users are then able to pick and choose their enabling ingredients to integrate into differentiated commercial offers within the broader edge ecosystem.

However when dealing with Client Device or Contraint Device, 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. On a sensor where only a micro controller 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.

Project Contributions for Infrastructure Management

Akraino

Project EVE

Project EVE is building EVE-OS which aims to be a universal bare metal foundation operating system for Distributed Edge computing. EVE-OS provides a foundation to extend the public cloud experience as far into edge as possible while meeting the unique needs of distributed edge deployments. It mimics the public cloud experience by abstracting away the complexity of the hardware by virtualizing all of the resources (e.g. processing, memory, storage, networking) and supporting any combination of virtual machines, containers and clusters. This enables developers to deploy any combination of legacy and new innovations and assign virtual resources to each application as they would in the cloud.

With a footprint of just 512MB of memory and disk, EVE-OS is architected to scale from a single box such as a gateway, hub or router to clusters of servers at the fringes of the data center. Below EVE-OS is the Constrained Device Edge which is inherently custom today, and it bleeds into the Metro and Regional Data Center Edge above.

EVE-OS Architecture Graphic

Compared to data center orchestration solutions that assume constant connectivity between a controller that instructs the servers what to run, EVE-OS assumes that the edge node will lose connectivity to its central controller at some point. It is also designed to communicate through firewalls and NATs on segmented IT and OT networks. It 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 the delta to works to update in a separate partition. If successful it switches over to the new software image, if not it continues to run as it was.

The next section describes how EVE-OS also provides a robust zero trust security model that is anchored in silicon-based root of trust, with layers of defense in depth to protect critical edge assets.

Open Horizon

Talk to core effort and bleeding into Client and Constrained Device Edges

SDO

Project Contributions for Application Management

EdgeX Foundry

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

Fledge

etc…

Edge Security

OSS and Security

how it works, overall trends (stats are ideal)

Can pull from recent LF sources:
State of SBOM and Cybersecurity Readiness
OpenSSF and Linux Foundation Address Software Supply Chain Security Challenges at White House Summit
Open Source Foundations Must Work Together to Prevent the Next Log4j Scramble
Linux Foundation: Defending the Global Software Supply Chain from Cyberattacks in 2021
How LF Communities Enable Security Measures Required by the US Executive Order

Unique Challenges at the Edge

Compared to
The security challenges along the edge continuum vary, as does the node's ability to address them.

Outside of the data center have to assume someone can walk up to a box and hack on it
Client devices have users associated with them whereas constrained and distributed cloud edge doesn't
ConstraIned devices lack resources to perform functions such as encryption
Wide mix of skills outside of the data center
Etc.

Differences between OT and IT

CIA vs. AIC
OT really cares about uptime and safety, immediate loss of production or even human life

Real World Threat Vectors

Over the past several years we have seen an increasing number of hacks at the edge. Mirai and Verkada - change passwords (or none at all), analytics or anomalies
Stuxnet - prevent local tampering
Target breach - change passwords
Etc.

-

Project Contributions for Edge Security

Project EVE

EVE-OS features a state-of-the-art and zero trust security architecture that assumes that edge nodes distributed in the field are physically accessible, in addition to not having a defined network perimeter. Features include support for silicon-based root of trust, measured boot, remote attestation, crypto-based ID (eliminating local device login), full disk encryption, remote port blocking, distributed firewall and more. Distributed firewall capability enables secure routing of data between edge applications and both on-prem and cloud resources based on network-wide policies.

Explain how EVE-OS could have prevented real-world hacks…

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. 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.

Insert copy for other project contributions above in alphabetical order…

  ○ Akraino

Project Akraino

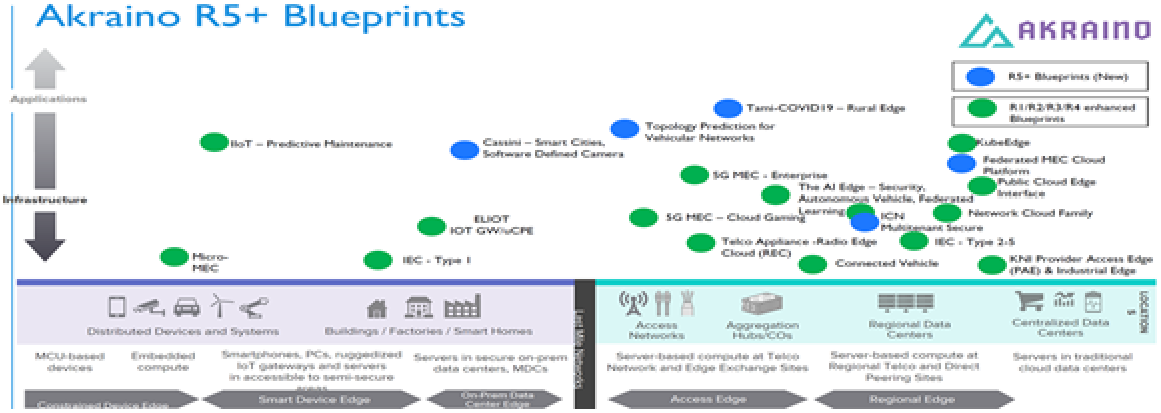
LF Edge Akraino project is an AT&T Lab Research Open Source Project spin-off that AT&T started with the intention to create an Open Source Software (SW) stack supporting high-availability Cloud Services optimized for Edge Computing Systems and Applications (Ref: https://about.att.com/sites/labs_research/open_source). In 2018, along with several other projects, AT&T decided to bring its Akraino project to Linux Foundation (https://www.linuxfoundation.org/press-release/akraino-edge-stack-new-linux-foundation-project-aims-drive-alignment-around-high-availability-cloud-services-network-edge/). Since then, as part of Linux Foundation Edge, Akraino project has evolved substantially, and currently is an umbrella project hosting more than thirty (30+) Integration and Feature projects (internally denoted as Blueprints) spanning a broad variety of Use Cases across a broad array of Technology Frameworks, including 5G, AI/ML, ICN, Edge IaaS/PaaS, IoT, K8, ICN/IEC etc. for both Provider and Enterprise Edge domains. These Blueprints have been created by the Akraino community and focus exclusively on the Edge in all of its different forms. What unites these Blueprints is that they have been tested by the community and are ready for adoption as-is, or used as a starting point for customizing a new edge blueprint.

The applied Design principles by Akraino Blueprints follow a holistic design focused on Availability, Capacity, Security, and Continuity as well as:

- Finite set of configurations - in order to reduce complexity, the design will follow a finite set of configurations.
- Support Multiple workloads types such as VMs, Containers, micro services, etc.,
- Security – The design needs to validate the security of the blueprint.
- Autonomous, turn-key solution for service enablement
- Platform, VNF and application assessment and gating – assess whether the application is fit to run at the edge (e.g. Latency, Sensitiveness, Code Quality).

Currently, Akraino project is at Release 5 (R5), approved in October 2021, and delivering a fully functional Open-source Edge Stack that enables a diversity of Edge Platforms across the globe. Akraino new Release life cycle management is scoped to 6 months and the coming new Release 6 (R6) is tentatively scheduled for approval by April 2022. The Akraino Blueprints work and preparation for R6 is on-going.

Akraino's 5th (fifth) Release brings three (3) new Blueprints (from a total of 30+ Blueprints and Feature Projects) and enables additional support for various deployments of Kubernetes across the Edge, including Smart Cities, Cloud Native Automotive, and Multi-tenant use cases.

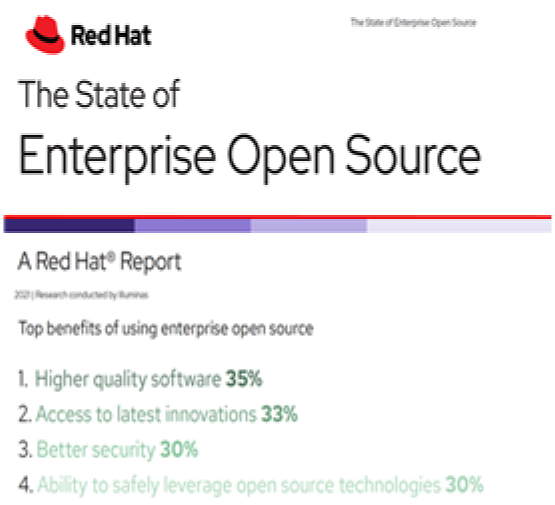

Akraino R5+ Blueprints
AKRAINO
R5+ Blueprints (New)
R1/R2/R3/R4 enhanced Blueprints
Applications
Infrastructure
IIoT – Predictive Maintenance
Cassini – Smart Cities, Software Defined Camera
Topology Prediction for Vehicular Networks
Tami-COVID19 – Rural Edge
KubeEdge
Federated MEC Cloud Platform
Public Cloud Edge Interface
5G MEC - Enterprise
The AI Edge – Security, Autonomous Vehicle, Federated Learning
5G MEC – Cloud Gaming
ICN Multitenant Secure
Network Cloud Family
ELIOT IOT GW/uCPE
Telco Appliance -Radio Edge Cloud (REC)
IEC - Type 2-5
Micro-MEC
IEC - Type 1
Connected Vehicle
KNI Provider Access Edge (PAE) & Industrial Edge
Distributed Devices and Systems
Buildings / Factories / Smart Homes
Last Mile Networks
Access Networks
Aggregation Hubs/COs
Regional Data Centers
Centralized Data Centers
LOCATION
MCU-based devices
Embedded compute
Smartphones, PCs, ruggedized IoT gateways and servers in accessible to semi-secure areas
Servers in secure on-prem data centers, MDCs
Server-based compute at Telco Network and Edge Exchange Sites
Server-based compute at Regional Telco and Direct Peering Sites
Servers in traditional cloud data centers
Constrained Device Edge
Smart Device Edge
On-Prem Data Center Edge
Access Edge
Regional Edge

New use cases combined with updated and existing Blueprints provide an edge stack for Industrial Edge, Public Cloud Edge Interface (PCEI), Federated ML, KubeEdge, Private LTE/5G, Smart Device Edge, Connected Vehicle, AR/VR, AI at the Edge, Android Cloud Native, Smart NICs, Telco Core and Open-RAN, NFV, IoT, SD-WAN, SDN, ETSI MEC, and more.

All of the Akraino Blueprints have been tested and validated on real Hardware (HW) Labs supported by Users and Community members - the Akraino Community has established a full-stack, automated testing with strict community standards to ensure high-quality blueprints.

LF Edge Akraino project is a good example reflecting the findings in the Red Hat's *The State of Enterprise Open-Source* reports from 2021 and 2022 (see below) indicating the importance of Open Source lower TCO (Total Cost of Ownership) shift to drop down to 6th place (from top position indicated 2 years (2019) ago) and preference, choice and assigned priority by Enterprises to select an Open -source Project due to the following top 3 (three) benefits of using an Open Source Project, namely:

1. Higher Quality Software (SW)
2. Access to latest Innovations
3. Better Security



Red Hat

The State of Enterprise Open Source

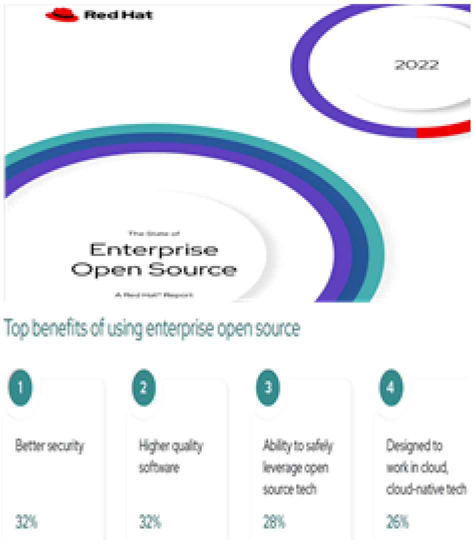
The State of
Enterprise Open Source

A Red Hat® Report

2021 | Research conducted by Illuminas

Top benefits of using enterprise open source

1. Higher quality software 35%
2. Access to latest innovations 33%
3. Better security 30%
4. Ability to safely leverage open source technologies 30%



Red Hat

2022

The State of
Enterprise
Open Source
A Red Hat® Report

Top benefits of using enterprise open source

1	2	3	4
Better security	Higher quality software	Ability to safely leverage open source tech	Designed to work in cloud, cloud-native tech
32%	32%	28%	26%

Related to “higher quality of SW”, all Akraino Blueprint projects are subject to follow “BluVal” (Blueprint SW Validation) Framework procedures and fulfil different requirements, depending on the level of Blueprint maturity life-cycle (Proposal, Incubation, Mature, Core, Archived). The purpose of the Blueprint Validation (BluVal) Framework project is to define a Standard Set of Tools and Tests to evaluate Akraino Blueprints in order to determine if the Blueprints are “Akraino ready/ validated”. BluVal Framework is a Diagnostic Toolset Framework that validates different Layers in the Akraino Infrastructure developed and used in Akraino edge stack. BluVal integrates different test cases, its development employs a declarative approach that is version controlled in LF Gerrit, integrated in CI/CD tool chain where peer Jenkins jobs can run the test cases and the results are reported in LF Repo (Nexus). The test cases cover all Blueprint layers in the cluster. BluVal Test Toolset Framework common (high-level listed here) requirements are:

- Support Kubernetes,
- Integrate with LF Gerrit,
- Run in an Akraino validation lab,
- Store test results in a Database.

In order to further facilitate the flow, commissioning and dependencies on supply of SW from various vendors, Akraino TSC (Technical Steering Committee) decided in 2021 to merge its two (2) Sub-committees, namely, “Upstream” and “Downstream” Sub-committees into one Sub-committee “Upstream and Downstream” committee, in order to facilitate to have collection of coordinators liaison to SW upstream and downstream and provide a single point of liaison and reporting/requests towards Akraino TSC.

Related to latest Innovations, Akraino project has been closely interacting with representatives (Chair) of ETSI MEC ISG since its early days in 2018. Each year, at bi-annual Akraino Technical meeting taking place in Spring and Fall each year, ETSI MEC representatives had participated and shared the latest evolvements in the standardization and development related to Multi-access Edge Computing. Besides, in 2020, a join hackathon was conducted. This interaction evolved to close co-operation in 2021, when LF Edge Akraino TSC and ETSI MEC ISG decided to institutionalize this co-operation. Each organization selected a representative that, together, prepared an Action Plan for the evolved LF Edge Akraino Project and ETSI MEC Co-operation.

This decision for co-operation between LF Edge Project and ETSI MEC was not coincidental, but in line with GSMA OPG (Operator Platform Group) decision in 2021 for developing an Architecture that shall enables E2E (end-to-end) Services on the Edge through the alignment of 3GPP EDGEAPP (Architecture for enabling Edge Applications, TS 23 558) and ETSI MEC (Multi-access Edge Computing) Architecture (please see below).

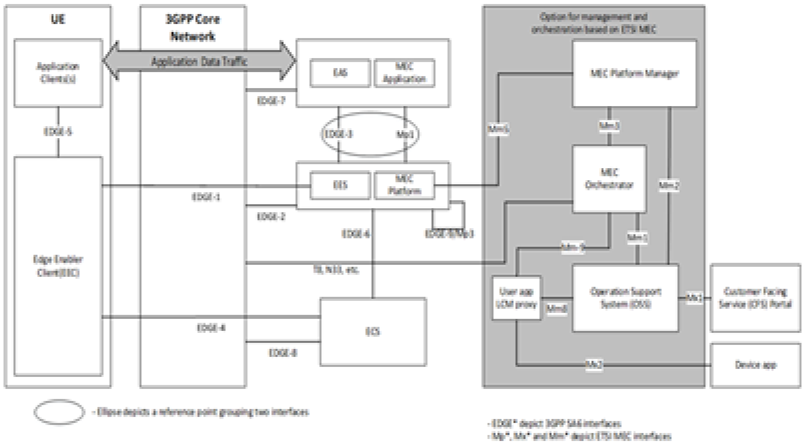
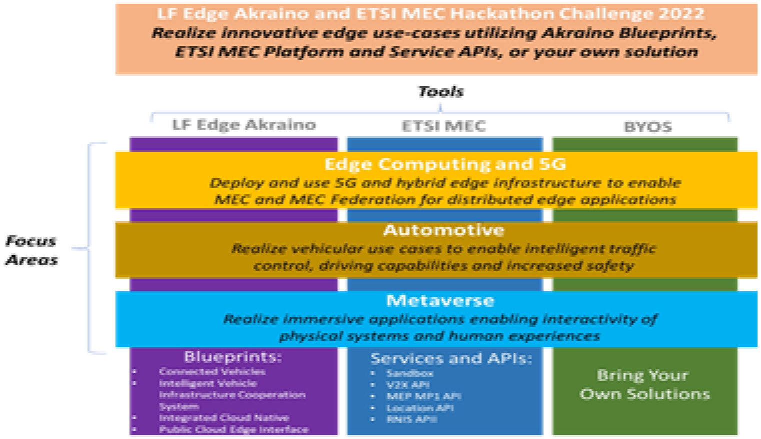


Fig.: Relationship in 3GPP EDGEAPP Architecture and ETSI MEC Architecture

Currently, LF Edge Akraino project and ETSI MEC is in process of planning their joint 2nd Hackathon, preliminary scheduled to take place in June 2022. The preliminary scope and challenge outlined for the Hackathon can be seen at next diagram:


LF Edge Akraino and ETSI MEC Hackathon Challenge 2022

Realize innovative edge use-cases utilizing Akraino Blueprints, ETSI MEC Platform and Service APIs, or your own solution

Tools

LF Edge Akraino | ETSI MEC | BYOS

Focus Areas

Edge Computing and 5G

Deploy and use 5G and hybrid edge infrastructure to enable MEC and MEC Federation for distributed edge applications

Automotive

Realize vehicular use cases to enable intelligent traffic control, driving capabilities and increased safety

Metaverse

Realize immersive applications enabling interactivity of physical systems and human experiences

Blueprints:
- Connected Vehicles
- Intelligent Vehicle Infrastructure Cooperation System
- Integrated Cloud Native
- Public Cloud Edge Interface

Services and APIs:
- Sandbox
- V2X API
- MEP MP1 API
- Location API
- RNIS APII

Bring Your Own Solutions

As it is seen from the LF Edge Akraino – ETSI MEC Hackathon 2022 planning diagram above, selected Akraino Blueprints providing Applications in **the Automotive and Metaverse (AR/VR/XR) Use Cases (UCs**) shall participate and share their accumulated knowledge and expertise from building and running Cloud-native Applications to contribute to evolvement of ETSI MEC and the effort to develop Functions and Features enabling Capabilities related to V2X and AR/VR Use Cases (UCs) in the converged EDGEAPP and ETSI MEC Architectures for enabling end-to-end Edge Services.

More specifically, as illustrated in the Figs 1 & 2 below related to Edge Services in V2X and AR/VR Use Cases (as part of the ongoing discussions for enhancements in EDGEAPP Architecture foreseen in "5G Advanced" release (Ref. 3GPP, Rel. 18, March 2022), an Edge Service or an EAS **(Edge Application Server, e.g. V2X server)** can be provided via different EDNs (Edge Data Networks) deployed by different EES (Edge Enabling Server) ECSPs (Edge Computing Service Providers).

Each ECSP may not have the required Infrastructure to install the EAS in every EDN due to financial, regulatory and operation constraints. A user can access the same Edge Service served by different EASs, which are registered to different EESs (Edge Enabling Servers) and deployed by different ECSPs, which have a Service Level Agreement (SLA) to share

Edge Services. These ECSPs can deploy EESs to serve different Mobile Networks (PLMNs) or different coverages of the same Mobile Network (PLMN).

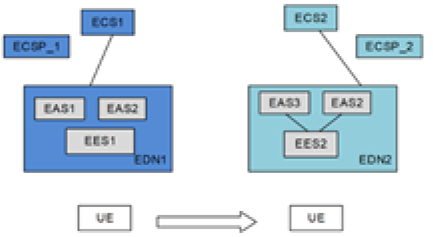


Fig 1.: EAS (Edge Application Server) deployed by different ECSPs (Edge Computing Service Providers)

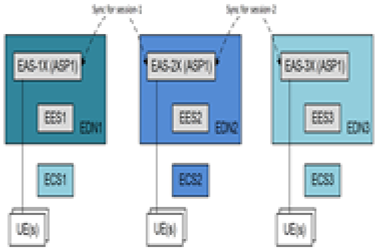


Fig. 2: EAS (Edge Application Server) with same Service deployed by same ASP (Application Service Provider) in different EDNs (Edge Data Networks)

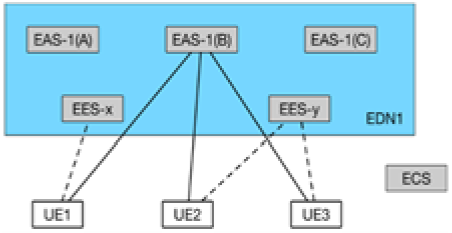


Fig. 3.: Several EAS (Edge Application Server) with same Service deployed in different locations in the same EDN (Edge Data Network)

For certain Use Cases (UCs), as indicated in Fig.3 above, involving Real-Time Communication in a Multi-User Session, both between User's (UEs) AC (Application Client via EEC/EDC (Edge Enabler Client/Edge DNS Client) and EAS and between different ACs via the EAS, it may be necessary or beneficial to use Services from a Single Common EAS to meet the strict Latency Requirements and to avoid the need for inter-EAS synchronization. The UCs may include, for example, a Team of Robots co-ordinating together on a Manufacturing floor, a Team of Surgeons using **VR Headsets and Robotic Surgery equipment** to operate together on a Patient, or a Group of Trucks using **V2X for Platooning.**

Dependent on the use case, the EEL (Edge Enabling Layer) may apply different additional criteria to determine this **common EAS.** E.g., it could be desirable to determine the EAS so that the Latency for all the ACs in the session is approximately the same or that the Latency for a specific AC is minimized. There is further utilization of capabilities related to EEL (Edge Enabling Layer) & AEF (API Exposing Function) and 5G NDL (Network Data Layer) specified and stored NF's Application Context (ACR/ACT, Application Context Relocation/Application Context Transfer) for assuring Service Continuity between S-EAS and T-EAS) as well as Data Traffic split rendering between EASs and CAS (Cloud Application Server). For evolved different roles related to Communication Services and Infrastructure, please see in the below figure:

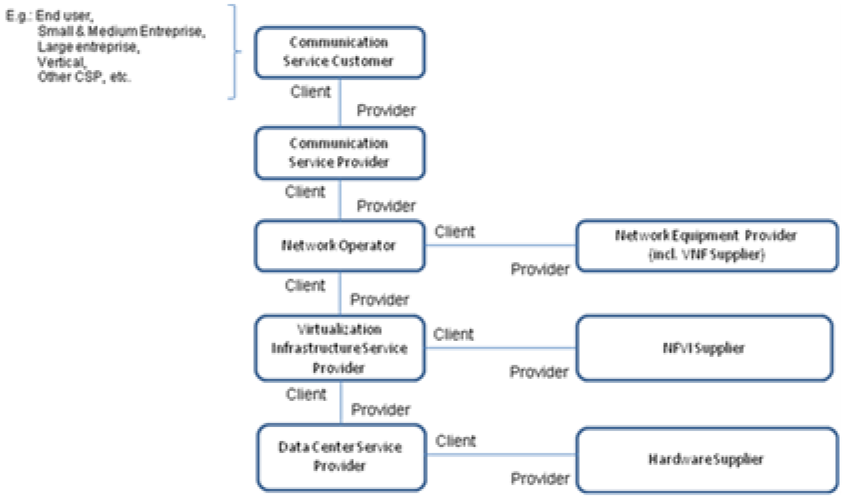
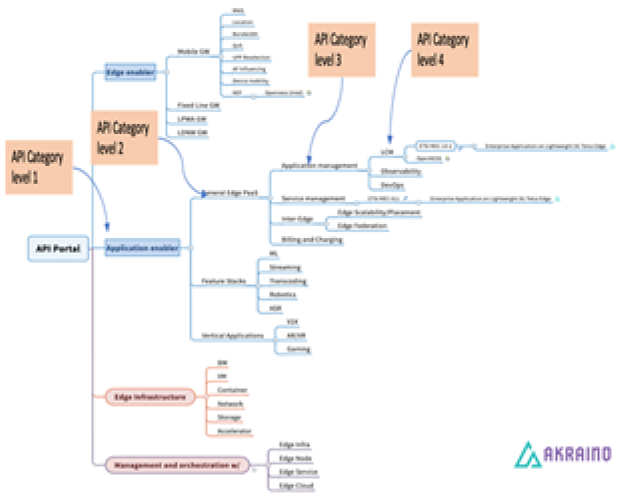
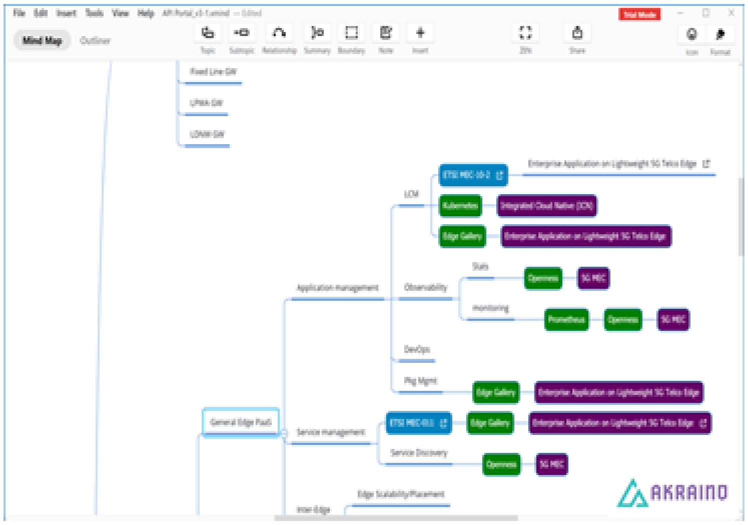

E.g.: End user,
Small & Medium Entreprise,
Large entreprise,
Vertical,
Other CSP, etc.
Communication Service Customer
Client
Provider
Communication Service Provider
Client
Provider
Network Operator
Client
Provider
Network Equipment Provider (incl. VNF Supplier)
Client
Provider
Virtualization Infrastructure Service Provider
Client
Provider
NFVI Supplier
Client
Provider
Data Center Service Provider
Client
Provider
Hardware Supplier

Fig.: High-level model of roles

Besides, within Akraino API Sub-committee, there is an ongoing study to map all the Blueprints' APIs in order to facilitate interoperability among the Akraino Blueprints as well as enable a quick overview and access to supporting documentation related to the APIs implemented by the Akraino Blueprints.

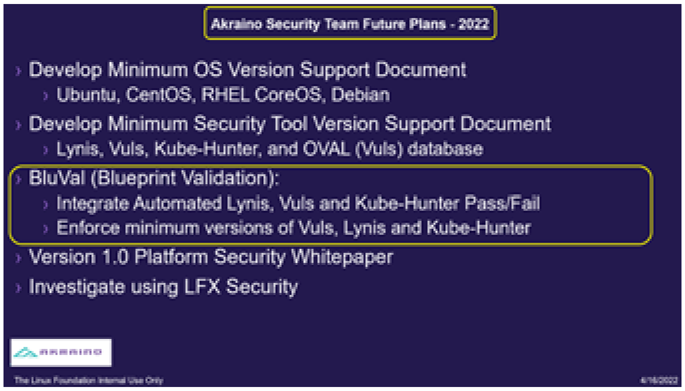




Akraino API Sub-committee uses Mind-map SW program to visualize the used APIs in a diagram with active links that can instantly direct the use and provide access to used API documentation. Please see below a snap-shot of the Akraino API Sub-committee API Portal use of Mind Map to visualize the Blueprints' APIs.

Related to Security, currently, the Akraino Security Sub-committee has implemented a pre-integrated with Akraino BluVal semi-automated validation as well as enables Security Platform for both x86 and Arm. Depending on the Blueprint maturity level (Proposal, Incubation, Mature, Core, Archived), there are formalized and documented different Lynis Tests.

There is Automated Lynis, Vuls and Kube-Hunter and Log Output Pass/Fail Analysis are provided. For 2022, Akraino Security Sub-committee plans to fully automate its security run scripts with BluVal Framework so that any Blueprint that initiates to validate its new SW script through BluVal will also automatically trigger running of pre-defined Security Test scripts as indicated below:



Akraino Security Team Future Plans - 2022

- Develop Minimum OS Version Support Document
  - Ubuntu, CentOS, RHEL CoreOS, Debian
- Develop Minimum Security Tool Version Support Document
  - Lynis, Vuls, Kube-Hunter, and OVAL (Vuls) database
- BluVal (Blueprint Validation):
  - Integrate Automated Lynis, Vuls and Kube-Hunter Pass/Fail
  - Enforce minimum versions of Vuls, Lynis and Kube-Hunter
- Version 1.0 Platform Security Whitepaper
- Investigate using LFX Security

The Linux Foundation Internal Use Only

4/16/2022

Additional and more detailed information about LF Edge Akraino project can be obtained from LF Edge web page with a link: https://www.lfedge.org/projects/akraino/

Insert copy for other project contributions above in alphabetical order…

- ○ Baetyl
- ○ Edge Gallery
- ○ EdgeX Foundry

    - eKuiper
    -
    - Fledge
    - Home Edge
    - Open Horizon
    -
    - Secure Device Onboard
    - State of the Edge

Edge Networking

- Edge networking
    - Overall trends and tradeoffs in edge networking, from constrained devices to regional edges
    - Detailed considerations on WANs, especially private 5G
    - Considerations for local area networking / distributed devices (e.g. “fog”)
    - Related contributions from each project

Two key paradigms. User Edge LAN, Service Provider Edge WAN. Bleed with private networks.

Considerations within the Service Provider Edge

Considerations within the User Edge

Contributions by Project

Edge Analytics

    - Inference vs training
    - Federated learning
    - TinyML
    -

LF Edge Project Update

- 2021 project milestones [pull from 2021 Annual Report, beginning on page 9]
- 2022 focus areas
- Market adoption
- Cross-project collaboration

    - Akraino
    - Baetyl
    - Edge Gallery
    - EdgeX Foundry