



From DevOps to EdgeOps: A Vision for Edge Computing

White Paper



The Edge as We See It



Depending on the point of view, the edge of a network can mean any number of things – from mobile or remote data centers with rugged servers to gateways, actuators, or the tiniest sensors interfacing with the physical world. Each of these perspectives encompasses myriad technology domains, sub-categories, and use cases. And each use case may require unique combinations of hardware, software, certifications, and other aspects.

Regardless of perspectives or requirements, the edge is wherever local compute capacity and storage is needed. That means, despite their many differences, every edge deployment comes with similar challenges.

EDGE | NATIVE

The Eclipse Edge Native Working Group

Our mission is to deliver a unified vision and platform for seamless development and operation of edge native applications in heterogeneous environments wherein computational power and data storage are physically distributed wherever needed along the edge-to-cloud continuum.

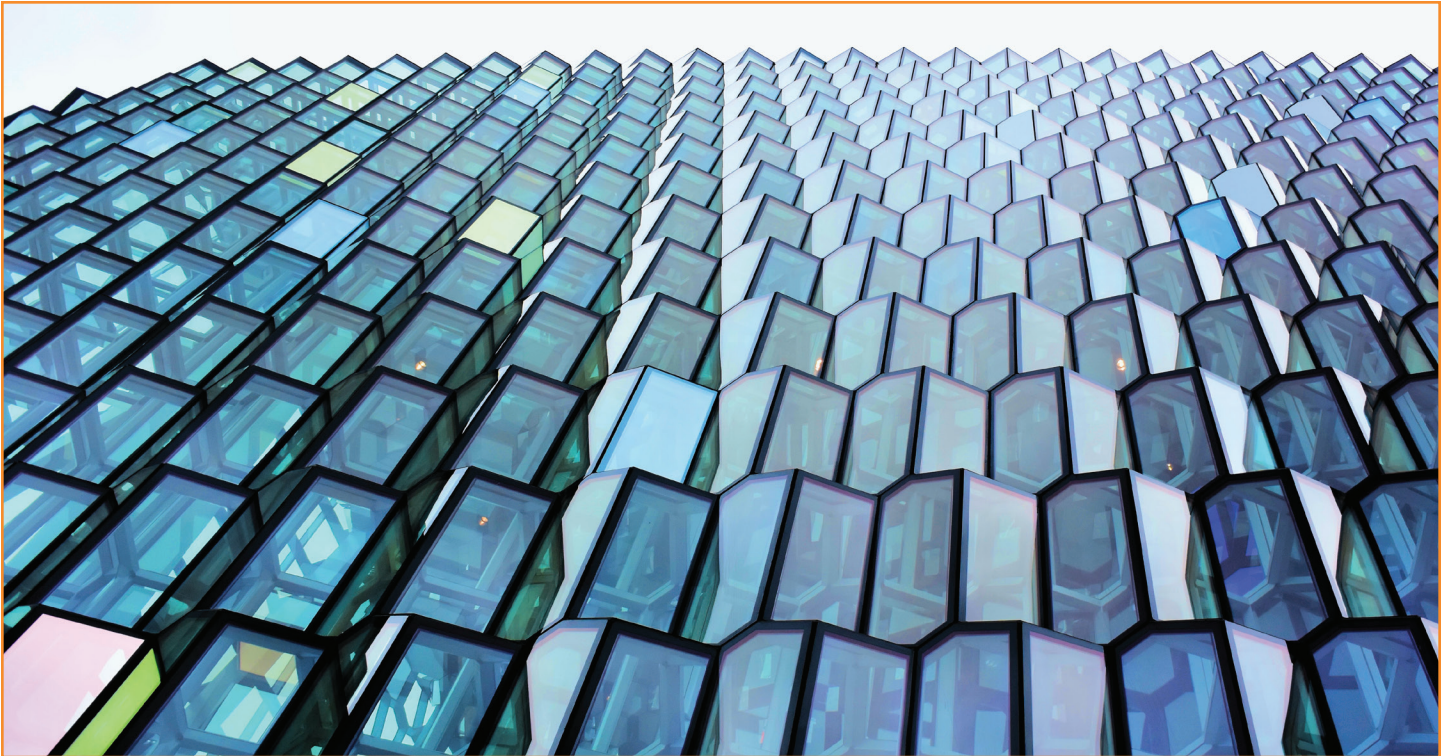
The Eclipse Edge Native Working Group focuses on making sense of the diversity and challenges in edge deployments. Our objective is to simplify the complexity at the edge by working together to create a common foundational software layer for a data-driven world. Working group members realize this goal by delivering edge computing platforms and tools that embody their vision for EdgeOps – DevOps at the Edge.

Before we explore EdgeOps and the concepts associated with it in more detail, it's important to understand the various challenges with edge deployments and what those challenges mean for developers and enterprises that design and deploy edge computing solutions.

Our objective is to simplify the complexity at the edge by working together to create a common foundational software layer for a data-driven world.



Edge Computing Brings New Challenges



Edge computing challenges are different from those in traditional computing environments since they compound the challenges of operational technology (OT), information technology (IT), and cloud computing environments. In the context of this paper, OT refers to the hardware and software an organization uses to control and automate devices, processes, and events within the organization, including the management of the resulting data. On the other hand, IT refers to the business-level systems, workflows, processes and data the organization relies on to support its mission.

From an OT perspective, deploying, securing, and managing stationary industrial automation equipment on a manufacturing floor differs substantially from deploying remote assets in the energy and utilities industries and from the mobile applications used in the transportation, rail, mining, and agriculture industries. Each industry has its own standards and certifications to ensure domain knowledge, best practices, and safety are preserved and applied. These

differing requirements make it challenging to broadly standardize edge computing hardware.

For the most demanding edge applications, there are often trade-offs between price, performance, and overall functionality that are driven by environmental and operational requirements. Factors such as operating temperature, shock and vibration, protection against contaminants, safety, as well as size, weight,

and power (SWaP) all have an impact on hardware designs. Other edge applications may be driven by price, performance, or specific functional requirements. Most often, an application- or use-case-specific piece of hardware results.

While this scenario is nothing new for established manufacturers of embedded boards and systems, the relatively new edge computing software paradigm still needs to mature to fully cope with the variety and diversity of edge computing devices.

In addition to these high-level computational hardware and software design challenges, there are a number of application-specific considerations when deploying at the edge.



Communications and Connectivity

For connecting remote systems and devices, there are a broad range of industrial field communication technologies and protocols such as Modbus, Controller Area Network (CAN bus), BACnet, LoRaWAN, and OPC Unified Architecture (OPC UA), as well as many other proprietary, legacy, and emerging options. Depending on the application, low power protocols such as NB-IOT, ZigBee, and Z-Wave may also be part of the equation.

The relatively new edge computing software paradigm still needs to mature to fully cope with the variety and diversity of edge computing devices.

Moving toward the cloud side of the equation, any number of fixed and wireless communications technologies may be required. Each of these technologies offers different features, performance, and behaviors related to usable bandwidth, latency, and reliability. Applications using cellular connections may also require local certification, increasing both effort and cost.

Power Requirements and Sources

With edge devices, power consumption is always an issue. The more functionality built into the device, the more power required. When considering power

needs, it's also important to factor in unreliable and dirty power sources, as well as the fact that some edge devices are not always on. With this in mind, some applications need to rely on low and alternative power sources, such as batteries, solar panels, and energy harvesting.

Security, Safety, and Reliability

Uptime requirements, and the need to protect against security threats and data loss play an important role in edge computing. Security, safety, and reliability

concerns sometimes require software architectures and solutions that are designed according to specific safety integration levels (SILs), or that conform to OT-centric or even vertical application-specific standards, such as [EN 61508](#), [ISO 13849](#), or [IEC 62443](#).

At the edge, there are physical security vulnerabilities that don't exist in data centers. In the field, physical protection of hardware is often difficult, requiring tamper detection and anti-tampering measures. Higher compliance requirements, along with concerns about data integrity and sovereignty, increasingly become an important aspect of software for edge OT infrastructure.

Management and Maintenance

Efficient and secure field servicing of remote devices requires appropriate infrastructure, policies, and processes that are supported at the software level. This includes full life cycle management, rollbacks, updates, and provisioning, all provided over the air (OTA) for individual and groups of devices at scale. Out-of-band management may also be a consideration.

Integration of Diverse Technologies

With edge computing, it's imperative to factor in the diversity of computing solutions being deployed. Devices at the "far edge" are typically application-optimized to efficiently interface with other equipment and assets to meet the needs of a specific industry or business requirement. It is these differing requirements that result in such significant technology diversity that edge computing solutions must address.



Making Sense of the Edge-To-Cloud Continuum



With the diversity of deployment scenarios and challenges, it's clear there is no one-size-fits-all solution for the edge. Varying options for data collection, compute, storage, and communications are available along a continuum from the very edge to the cloud, and spanning different geographical locations, network tiers, and administrative domains (Figure 1).

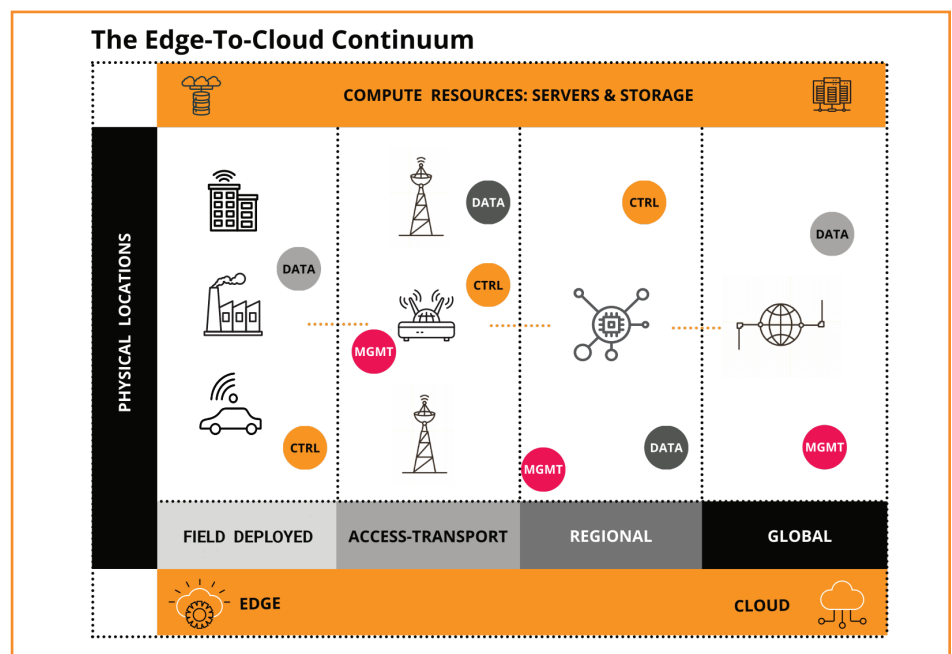


Figure 1: The Edge-To-Cloud Continuum. Physical locations, network tiers, and system planes (data, control, management).

Edge infrastructure typically encompasses a heterogeneous mix of resources that can be extremely diverse in terms of capabilities and operating conditions. For example, devices at the very edge may include sensors, on-board units, mobile phones, and even robots. Devices may be battery powered, connected using wireless links, have varying levels of mobility, and operate in non-physically secured environments, such as roadways, public buildings, and remote locations. In contrast, cloud infrastructure is stationary and highly homogeneous – interconnected through reliable and fast connections and operating in a physically secured environment.

It would be naive to consider the cloud and the edge as two discrete locations. Rather, they are the two

extremities of a continuum inside which compute, storage, and networking resources can be deployed at any point, in any topology (see Figure 1). In turn, this means the various software components of a solution are also susceptible to be distributed throughout the continuum.

Harmonizing Data, Control, and Management Resources

The availability of distributed and heterogeneous resources is simply the first step toward creating a fully functional edge-to-cloud solution. These resources must then be harmonized and made to work together. The traditional systems approach is to encompass different planes for data, control, and management (Figure 2).

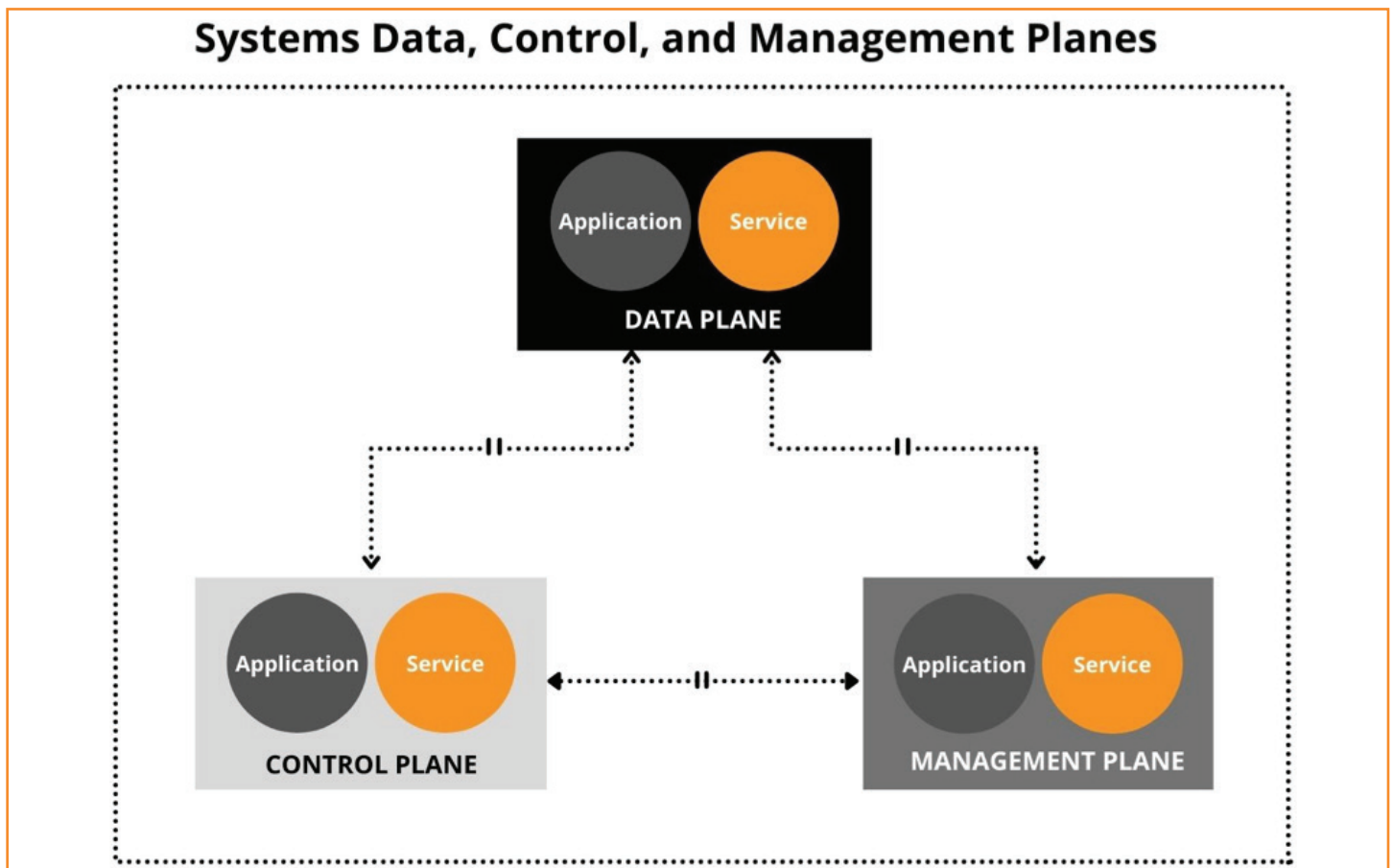


Figure 2: Systems Data, Control, and Management Planes. Adapted from RFC 7426.

Each plane has its own unique purpose within the system:

- The **data plane** includes network, compute, and storage for managing application data according to the instructions received from the control plane.
- The **control plane** makes and enforces runtime decisions about how the data plane processes and acts on data as it is received.
- The **management plane** monitors, configures, and maintains the state of each resource.



For more information about these planes, refer to [IETF RFC 7426](#), Software-Defined Networking (SDN): Layers and Architecture Terminology.

While this three plane concept similarly applies to the edge-to-cloud continuum, there are some differences. Components used within the continuum (system) are inherently distributed and need to operate according to varying traffic patterns based on bandwidth, latency, and timescale. These types of operations differ from a pure cloud system wherein each resource produces and consumes data in a similar manner.

Considering the above, edge devices should not be forced to operate using mechanisms designed for other purposes, such as cloud data centers. Many Edge and IoT solutions are focused on data acquisition through sensors, device control through actuators, local data processing, or a combination of the above. On the other hand, most IoT cloud platforms provide data analytics and machine learning capabilities at a larger scale for use cases that can tolerate higher latencies. Edge applications impose diverse requirements on the traditional three plane model and must be accounted for to optimize development and operation of the edge-to-cloud continuum.

The Edge-To-Cloud Continuum and Open Source



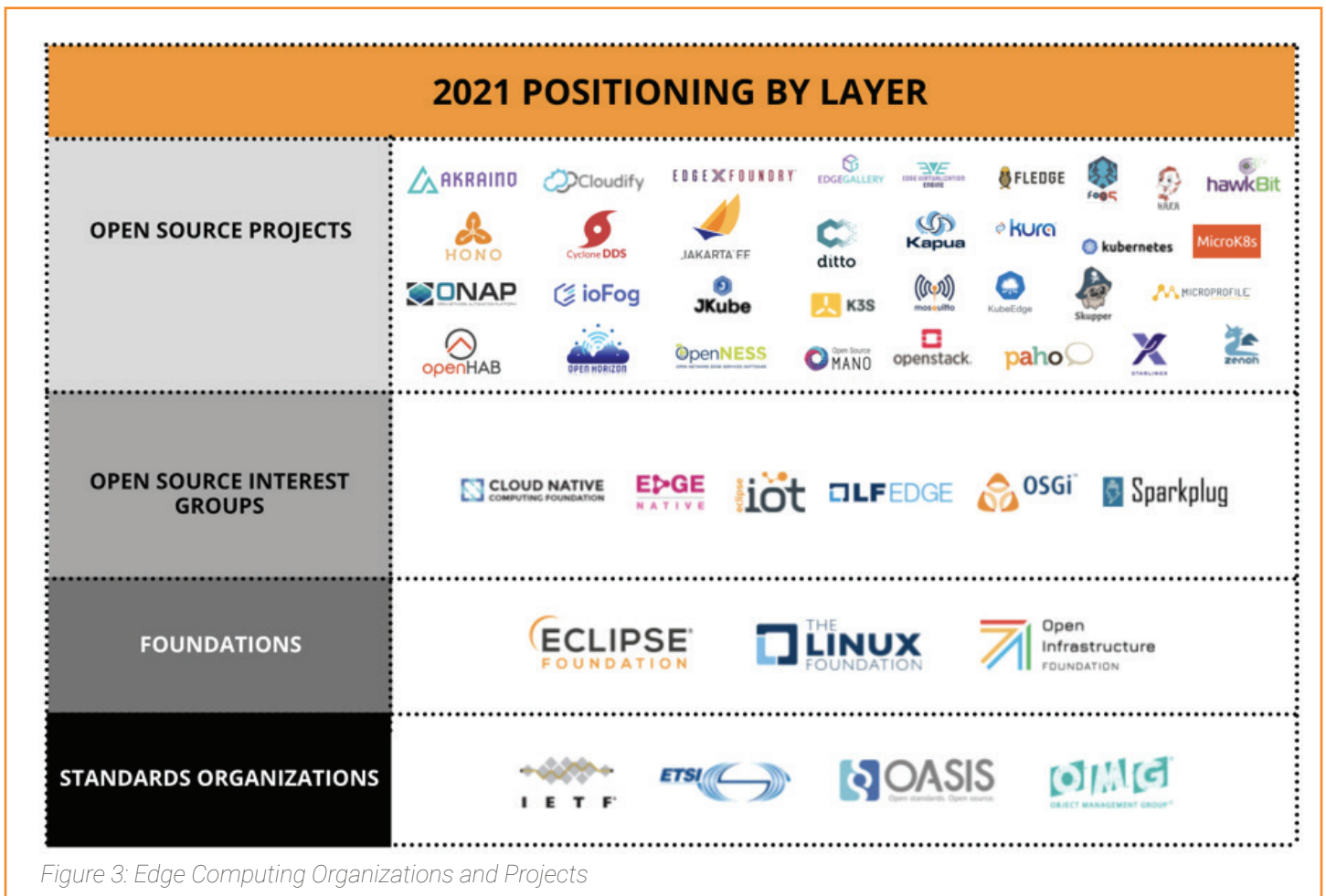
The “edge-to-cloud” continuum is a broad and diverse space. Within this space, there are a number of standards and open source organizations/projects focused on edge computing technologies (Figure 3 on the following page).

Starting at the bottom of Figure 3 and working our way up, we can take a closer look at how each of these groups contribute to the edge-to-cloud continuum.

Standards Organizations provide stable building blocks upon which others can base their innovations and are particularly influential in highly regulated markets such as telecommunications, transportation, utilities or healthcare.

Open Source Foundations are member-driven and vendor-neutral, providing a perfect environment for collaboration, innovation, and ecosystem building. Innovation flourishes in an open and transparent

Within the “edge-to-cloud” space, there are a number of standards and open source organizations/projects focused on edge computing technologies.



environment with a level playing field. Open source foundations are also increasingly involved in standardization efforts. For example, in 2019, the Eclipse Foundation introduced the [Eclipse Foundation Specification Process \(EFSP\)](#), which defines a general framework for developing open source specifications.

Open Source Interest Groups operate within open source foundations. Each community has its own edge computing scope and mission. For example, the Cloud Native Computing Foundation (CNCF), home to the ubiquitous Kubernetes container orchestration platform, is involved because Kubernetes

is often deployed on edge compute nodes to manage workloads. Additionally, groups such as LF Edge and the [Eclipse Edge Native Working Group](#) have a tight focus on edge computing platforms and tools.

Open Source Projects are the keystones of the edge computing ecosystem. Later in this paper, we will explore how and where some of the most widely used projects fit within the edge-to-cloud continuum and their focus (i.e., development vs. operations).

But first, let's further explore the concept and importance of EdgeOps.

EdgeOps: DevOps for the Edge



So, what is EdgeOps? Traditionally, DevOps is the combination of best practices in software development and IT operations to shorten a system's development life cycle (SDLC), enabling continuous delivery without impacting quality. Simply put, EdgeOps applies these same DevOps principles to the edge-to-cloud continuum.

While this may seem relatively simple on the surface, as previously discussed, there are some differences between DevOps and EdgeOps that create unique edge development requirements along with the unique capabilities needed to address those requirements.

Edge Developers Have Specialized IDE Needs

When it comes to coding for the edge, the languages and IDEs are the same, but the tools, and libraries developers rely on must be tailored for the application. In some cases, runtime settings will need to be tweaked

There are some differences between DevOps and EdgeOps that create unique edge development requirements along with the unique capabilities needed to address those requirements.

as well. For example, a web application developer may use a local database instance or a sandbox version of a REST API to write code. An edge developer, on the other hand, might need the equivalent “local environment” for devices (such as motor controllers), which are found only at the edge and communicate specific field protocols and field bus technologies.

Build Processes Must Support More Processor Types

Traditional build processes are mostly concerned with single CPU architecture environments. It’s rare to come across a cloud server that doesn’t belong in the x86/AMD64 category.

Edge environments, on the other hand, are a mix of 32-bit and 64-bit ARM devices and x86/AMD64 devices. Frequently, 8-bit and 16-bit microcontrollers are also involved. Build processes for edge computing must produce outputs that are ready for deployment on all target devices. This includes field programmable gate array (FPGA), vision processing unit (VPU), tensor

processing unit (TPU), and graphics processing unit (GPU) technology.

New Testing Approaches Are Needed

Given the heterogeneity of edge environments, and the importance of specific hardware configurations, software testing at the edge cannot rely exclusively on virtualization or simulated devices. Some level of integration testing with actual hardware is required. This said, it can be challenging to virtualize an edge computer with specific sensors attached to General Purpose Input/Output (GPIO) pins. Moreover, there is currently no readily available library or tool that allows developers to validate software across the wide variety of SOCs, supporting chips and boards.

Furthermore, with edge computing, tools may require testing when hardware is idle. This may include installing a new operating system image on the hardware each time a test is executed to verify that a newly deployed edge computer is functioning properly.



Software Packaging Must Support More Technologies and Environments

The edge-to-cloud continuum includes everything from microcontroller binaries, to containers, to virtual machines. With the variety of technologies available, software packaging for edge environments introduces new complexities.

One aim of edge computing is to keep properly functioning hardware components in place, while also expanding their usefulness. Software artifact repositories are no exception. Let's consider container image repositories as an example, keeping in mind that containers are just one software packaging technology option. Essentially, the packaging and hosting of software for the edge is the same as for any other repository. However, the container images produced in the build phase must now be available to all edge geographical locations while using the smallest amount of network bandwidth possible.

One aim of edge computing is to keep properly functioning hardware components in place, while also expanding their usefulness. Software artifact repositories are no exception.



Let's use a geographically dispersed mining operation as an example, assuming each edge geography has backhaul connectivity over a satellite link. In such a scenario, the container image repositories must be distributed so that each one can serve the edge computing nodes on its local network and prevent hundreds of duplicate pulls of the same container image over the limited and expensive backhaul link.

With these kinds of considerations and challenges, additional features, security measures, and configurations will likely be needed for technologies that were previously primarily centralized.

Software Updates Must Be Carefully Staged

Managing software releases has always been an important consideration. Edge computing adds an OT aspect that further increases that importance. Systems that rely on edge applications may experience outages, downtime, or physical damage due to new software releases. Unplanned or failed software releases can produce outcomes for which there is no repair.

To put this into perspective, we can take a backup snapshot of a database before upgrading to a new release, but we cannot take a backup snapshot of 200 meters of sheet aluminum before it is improperly cut.

What this means to edge computing is that we need a staged approach to release preparation, approval, and management of edge applications. And, the approach must include verification processes before transitioning to the next step in the upgrade sequence.

EdgeOps Professionals Must Define Edge-Specific Capabilities

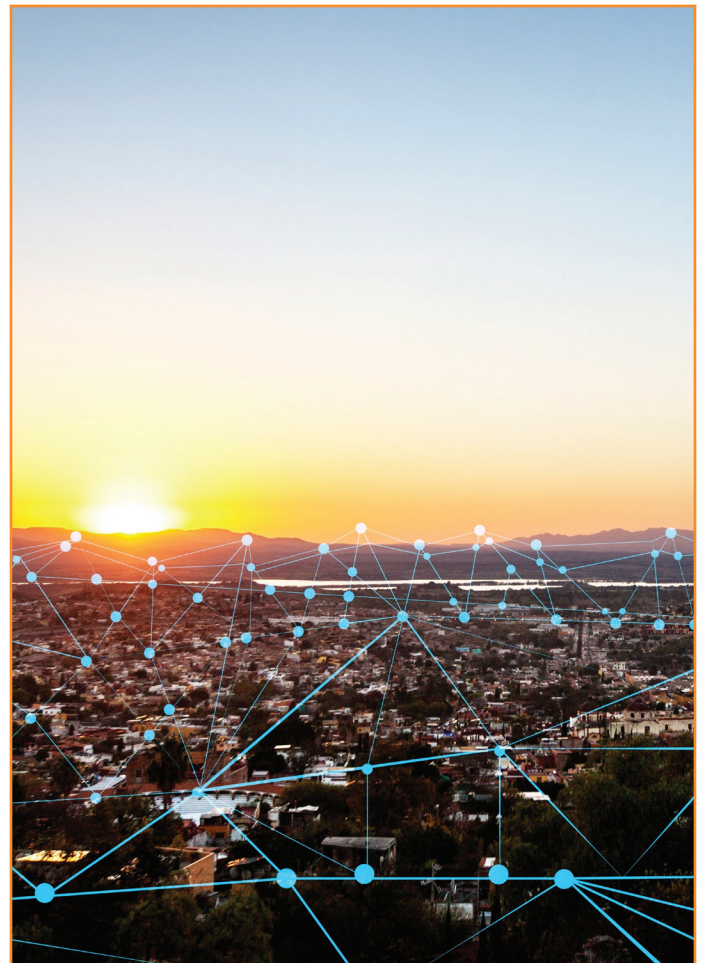
In addition to defining compute, storage, and network infrastructure, EdgeOps professionals must also define edge-specific aspects, such as video camera rotation, maximum tolerable network latency, and compatible processor architectures. Extensible models of edge infrastructure are emerging to allow significant flexibility, while also providing enough standardization to make edge environments manageable.

Secure Remote Monitoring and Management Is Essential

In an edge application, the entire environment in which it operates must be securely monitored, managed,

and optimized remotely. As an example, if a work instruction is deployed to a factory workbench targeted for a specific technician, that same application must be automatically transferred to a different workbench when the technician completes the first task and moves elsewhere (regardless of geography), without creating a security or operational breach.

EdgeOps monitoring tools should be able to retrieve any necessary data from a private LAN, while also protecting system, device, and end-user security. The connectivity may be varied for any given edge compute node, which means monitoring must be adjusted to accommodate limited bandwidth or infrequent connections. Ideally, application monitoring for the edge is flexible enough to handle a wide range of connectivity scenarios.



The Eclipse Foundation Enables EdgeOps



With the many and varied requirements described in the previous section, it's clear that seamless EdgeOps cannot currently be delivered through a single platform.

This said, there are a number of open source projects that play a critical role in building a world-class EdgeOps ecosystem. Figure 4, on the following page, shows an EdgeOps Matrix of these projects. The horizontal axis represents where a project is positioned on the edge-to-cloud continuum, while the vertical axis is meant to demonstrate where a project fits on the development tools vs. operational platform spectrum. While many projects cover a range on both axes, we have attempted to pinpoint positions according to typical roles in an edge computing architecture. Projects hosted at the Eclipse Foundation are featured with a black line and red dot.

This matrix demonstrates the breadth and completeness of Eclipse Foundation projects that enable the Edge Computing ecosystem, including:

- Frameworks for building cloud native Java enterprise applications: [Jakarta EE](#) and [Eclipse MicroProfile](#)
- Libraries that support popular protocols, such as Message Queuing Telemetry Transport (MQTT): [Eclipse Paho](#) and [Eclipse Mosquitto](#)
- Platforms to implement high-throughput message routing: [Eclipse Hono](#)

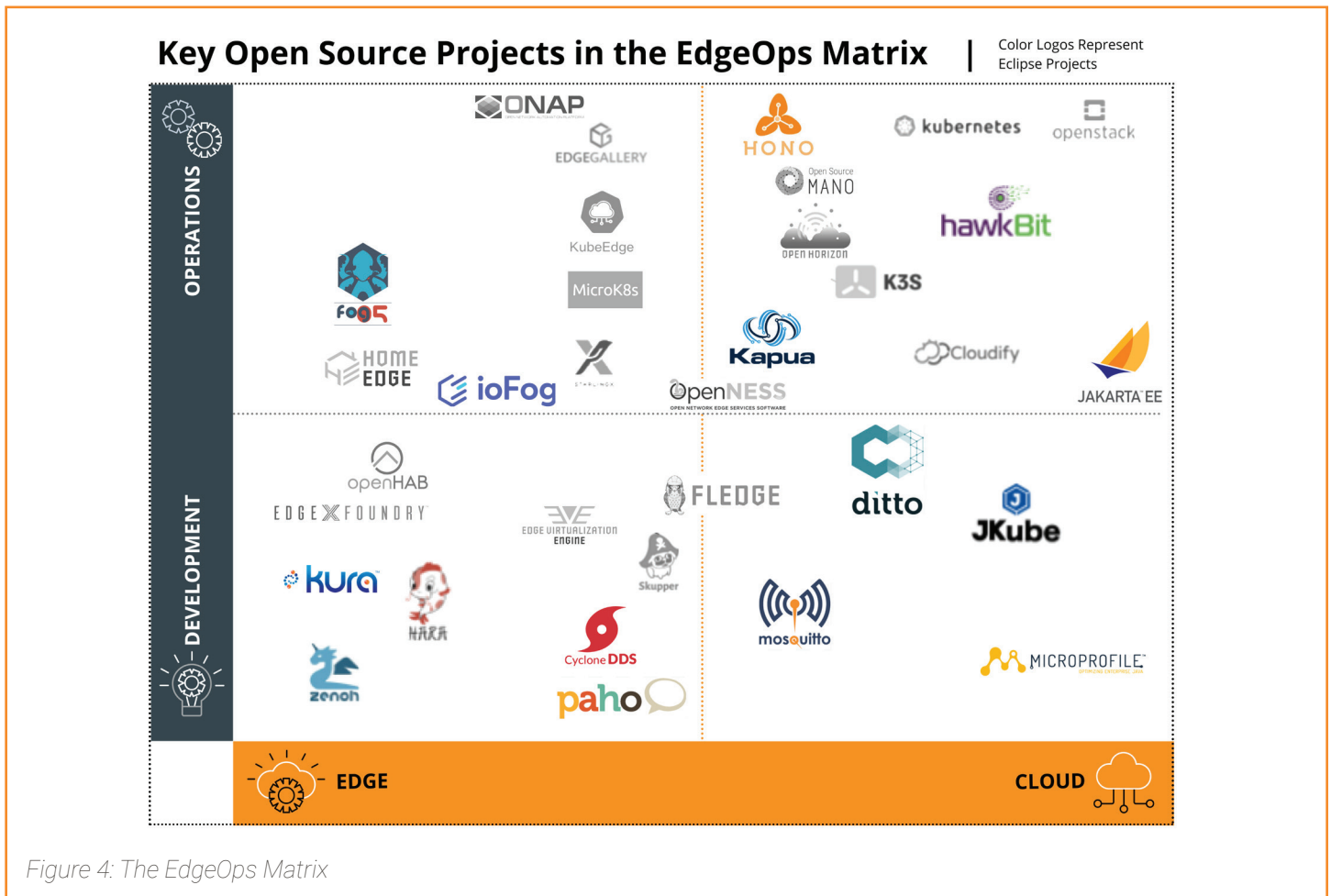


Figure 4: The EdgeOps Matrix

- Technology for device software updates: [Eclipse hawkBit](#)
- Technology for digital twins: [Eclipse Ditto](#)
- IoT edge device middleware: [Eclipse Kura](#)
- IoT integration platforms: [Eclipse Kapua](#)

The Eclipse Foundation is also a member of the Cloud Native Computing Foundation (CNCF) and strongly supports industry standards, such as Kubernetes, for container orchestration.

In addition to the projects listed above, the Eclipse Foundation is home to three of the most innovative projects within the EdgeOps ecosystem: [Eclipse ioFog](#), [Eclipse zenoh](#), and [Eclipse fog05](#). Each of these

projects is production-quality, commercial-grade, and licensed under commercial-friendly terms. All three are hosted by the [Eclipse Edge Native Working Group](#).

Eclipse ioFog Is a Complete Edge Computing Platform

Eclipse ioFog unifies the various components needed for edge computing to provide the data, control, and management planes in one technology layer. The project introduces the concept of an Edge Compute Network (ECN), which is a collection of edge compute nodes that can span the entire edge-to-cloud continuum and any geography (see Figure 5).

A software overlay network enables edge service mesh, peer-to-peer data transmissions, and dynamic

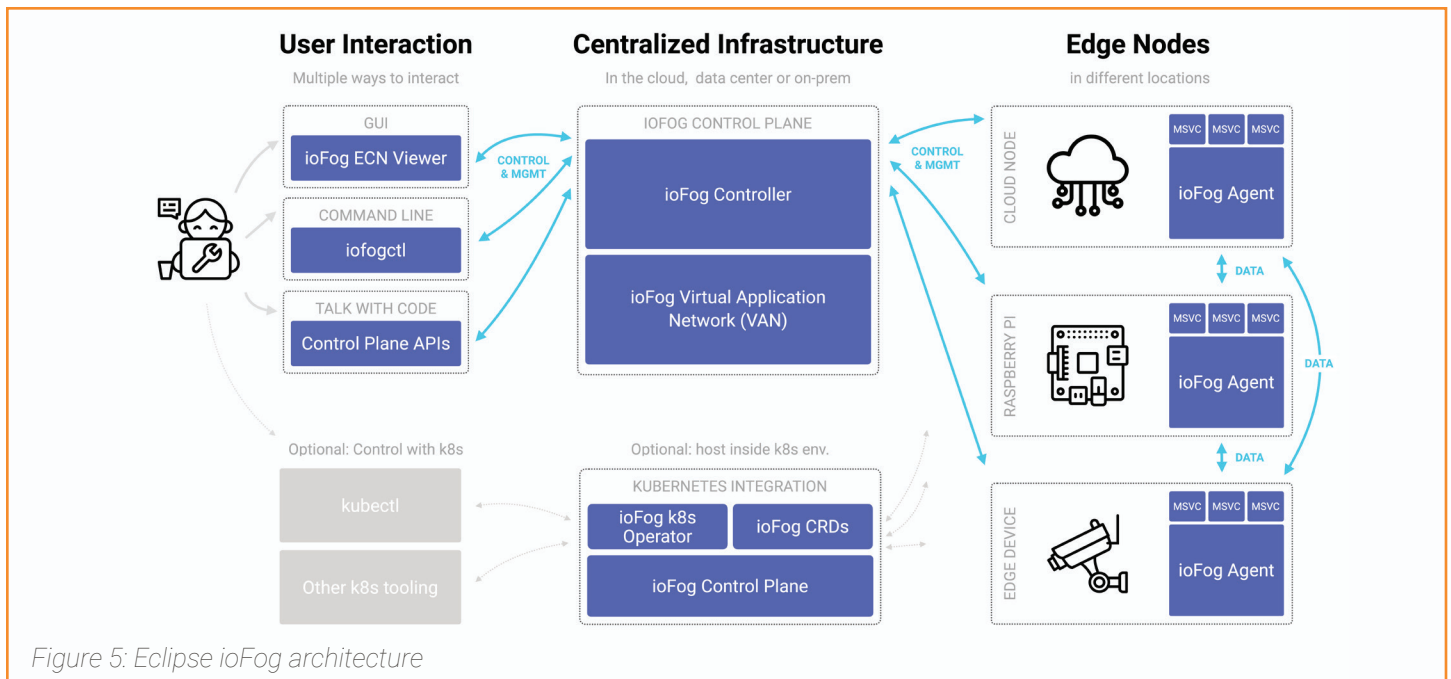


Figure 5: Eclipse ioFog architecture

interconnection of multiple edges and multiple clouds. Existing cloud applications can be deployed to the edge with minimal effort. Applications can open network communications as occurs in the cloud, or they can take advantage of the managed data plane with flexible routing for tight control over how microservices connect.

The user interface (ECN Viewer) provides an easy way to view and manage ECNs right out of the box. Edge compute nodes can leverage any processor architecture and run any Linux distribution. Eclipse ioFog also provides additional security layers on all edge nodes to protect the ECN from the edge in.

Eclipse ioFog is completely focused on delivering EdgeOps capabilities. Users can:

- Monitor, deploy, start, and stop applications composed of any number of microservices across any number of edge compute nodes.
- Tune traffic control and management for each node independently.

- Remotely upgrade or roll back the ioFog infrastructure. All elements are represented in YAML for edge infrastructure as code.
- Use namespaces to manage all ECNs in one place and easily switch between ECNs to perform operations as needed.
- Script EdgeOps and make edge testing or deployments part of continuous integration and continuous deployment (CI/CD) systems.

The Eclipse ioFog technology fits at the center of the edge stack and is meant to be extended. A set of APIs is used to coordinate EdgeOps tasks or build an ECN management interface. With these capabilities, EdgeOps professionals can:

- Use existing hardware abstractions to simplify edge software development challenges or add use-case-specific interfaces, such as handling proprietary protocols.

- Use edge resource definitions to define the most important aspects of edge environments so it can become part of ECN management.
- Collaborate with team members and manage applications at a larger scale using application templates.
- Set policies for automated artifact pruning and resource consumption limits that protect edge compute nodes from running out of disk space, CPU, or RAM.



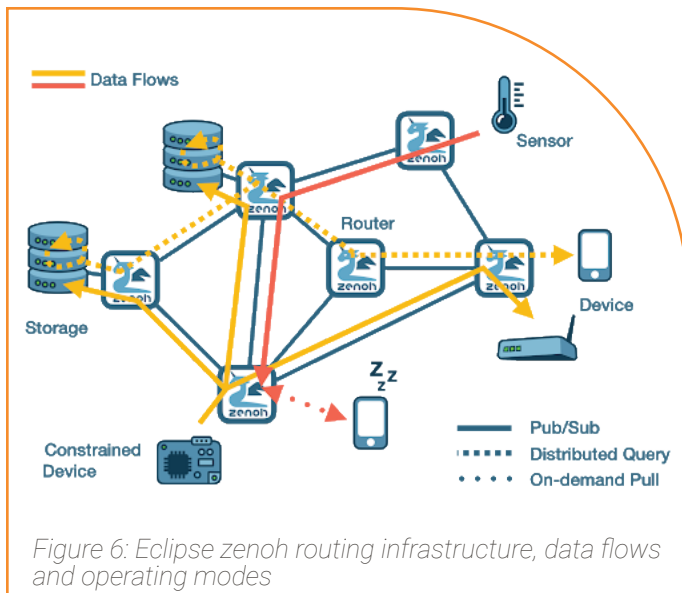
The Eclipse ioFog technology fits at the center of the edge stack and is meant to be extended. A set of APIs is used to coordinate EdgeOps tasks or build an Edge Compute Network (ECN) management interface.

- Add container registries and mix/match registries to take advantage of public images while keeping private images secure.
- Start with Eclipse ioFog to deploy edge microservices immediately and then build upon it to reach the required scale.

Eclipse zenoh Unifies Data and Computations

Eclipse zenoh provides a unified paradigm for data-in-motion, data-in-use, data-at-rest, and computations in the edge-to-cloud continuum. It does this by carefully interconnecting heterogeneous sets of resources and applications that operate in diverse environments and networks. (see Figure 6)

Eclipse zenoh blends traditional publish/subscribe technologies with geo-distributed storage, queries, and computations in a consolidated and location-transparent API. This approach allows developers to focus on how to use the data, rather than where data is located. To achieve this, Eclipse zenoh uses time



and space efficiency to manage various symmetric and asymmetric systems, as well as different communications patterns, such as support for sleeping resources that may be present in the edge-to-cloud continuum.

The technology's simplified approach to operating data, control, and management planes from the edge to the cloud, brings significant benefits to many edge technology use cases:

- Edge robotics
- IoT sensors
- Industrial automation
- Infrastructure management
- Connected vehicles
- Content delivery networks

To simplify development of distributed applications at any scale, Eclipse zenoh provides first-class abstractions for publish/subscribe, storage, query, and evals:

- Efficient publish/subscribe primitives support multiple levels of reliability, dynamic discovery, fragmentation, and wire-level batching.
- Primitives for defining geo-distributed storage.
- Well-defined semantics for querying and aggregating tasks.

In addition, Eclipse zenoh allows applications to register computations that are triggered by queries. This simple mechanism allows many patterns, including RPC and map-reduce, to be implemented.



The Eclipse zenoh protocol enables applications to efficiently scale up and scale down. A scalable routing infrastructure allows different application components, such as microservices, to seamlessly reach internet scale.

Eclipse zenoh has a very simple API that makes it extremely fast to get started and productive. APIs are available for the most popular programming languages and more are added regularly.

The Eclipse zenoh protocol enables applications to efficiently scale up and scale down. A scalable routing infrastructure allows different application components, such as microservices, to seamlessly reach internet scale. Eclipse zenoh implementations can effectively exploit the parallel processing in modern CPUs, achieving high throughput in sustained conditions.

At the same time, the Eclipse zenoh design enables efficient scale-down capabilities. A minimal wire overhead of 4 bytes enables support of constrained transport mechanisms, such as Low-Power WANs (LPWANs), Low-Power Wireless Personal Area Networks (LoWPANs), and Bluetooth Low Energy (BLE), as well as extremely resource-constrained devices, such as 8-bit microcontrollers.

Finally, Eclipse zenoh offers extremely low latency, making it appropriate for applications that require exceptionally low overhead and reaction times.

Eclipse fog05 Provides a Comprehensive Fog Computing Platform

Eclipse fog05 provides a decentralized infrastructure for provisioning and managing compute, storage, communications, and I/O resources that are available anywhere along the edge-to-cloud continuum (see Figure 7).

The technology addresses highly distributed and heterogeneous systems, even those with extremely resource-constrained nodes, by offering unified life cycle management for applications. This enables end-to-end operation of composite applications, spanning from cloud and edge data centers to terminal devices. Eclipse fog05 enables end-to-end infrastructure control and management for cooperative robotics, on-board infotainment, assembly line coordination, and other applications.

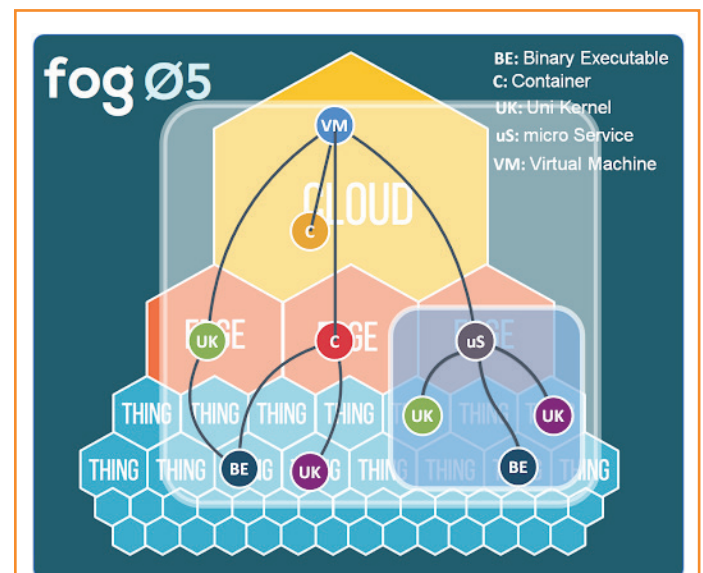


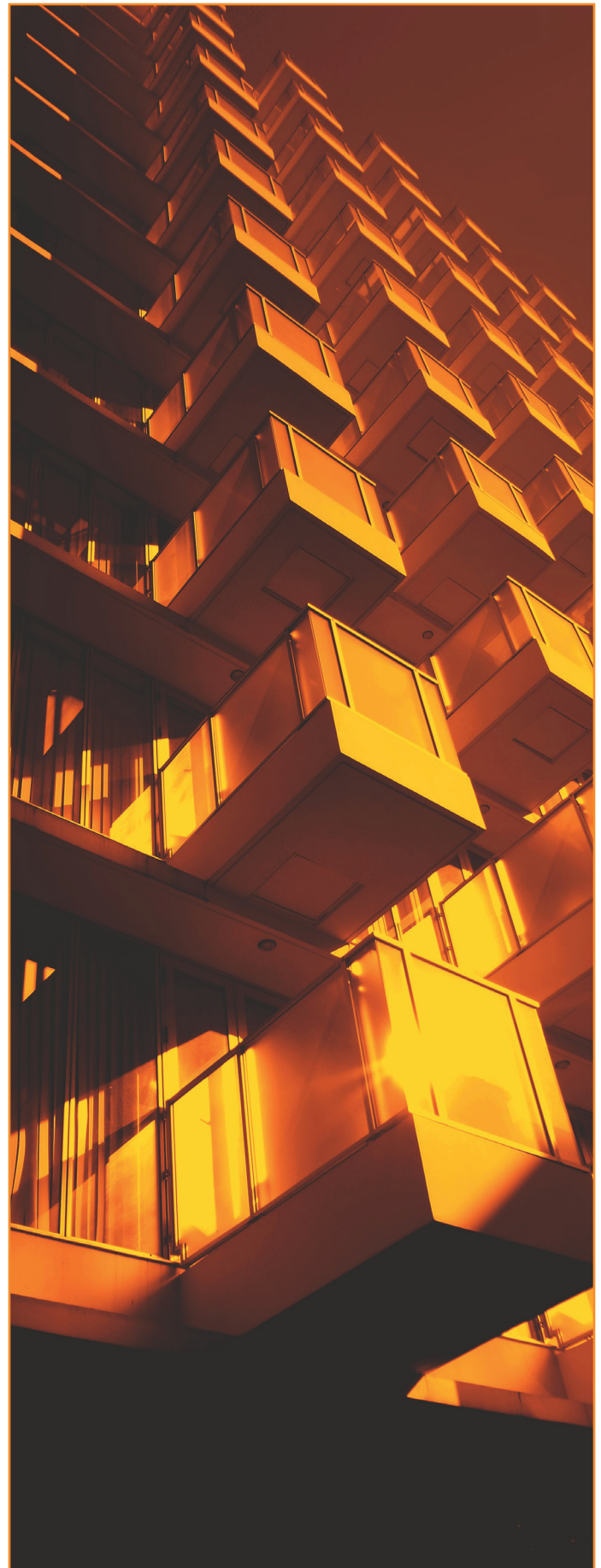
Figure 7: The Eclipse fog05 deployment model

Eclipse fog05 supports heterogeneous runtimes, hypervisors, and networking technologies to enable deployment of applications composed of one or more virtual machines, containers, Robot Operating System 2 (ROS2) applications, native applications, or a mix of these, and other components. This wide-ranging support is possible because the modular design supports an open-ended set of easy-to-write and easy-to-extend plugins that target relevant technologies, such as operating systems, network fabrics, hypervisors, and containers, among others.

Eclipse fog05 control and management planes are powered by Eclipse zenoh, enabling local distribution of the infrastructure state along the edge-to-cloud continuum while maintaining a global view. This results in greater fault tolerance when operating in an environment that does not offer the homogeneity and stability of cloud data centers.

Eclipse fog05 also offers a common API and information model for application management. Applications are defined using a single descriptor that characterizes the application components, their interdependence, and their requirements for computing, storage, and networking. The descriptor is provided to Eclipse fog05, which matches the requirements to the underlying infrastructure. This allows application developers to focus on their applications, instead of how to deploy them on a distributed infrastructure.

Finally, Eclipse fog05 also offers a unified API for managing virtualization infrastructure and harmonizing its underlying heterogeneity and complexity.



Making Edge Computing and EdgeOps a Reality With Open Source

Considering the complexity and technical challenges associated with edge computing, it's very likely that most successful platforms will be completely open source or built around an open source core. An open source approach enables commercial players to achieve a meaningful level of differentiation by investing in value-added features and services rather than the core platform.

At the same time, it's important to point out that all open source governance and licensing models are not created equal. The ideal open source project leverages commercial-friendly licenses, such as the Apache License or the [Eclipse Public License](#), and a vendor-neutral governance model.

A world as diverse as the edge requires an ecosystem approach. And, such ecosystems can only exist as open source communities.

The Eclipse Foundation provides a mature, scalable, and business-friendly environment for open source software collaboration and innovation. Activities are structured around two complementary concepts: projects and working groups. The Eclipse Edge Native Working Group is one of more than 15 working groups currently hosted at the Eclipse Foundation.

The Eclipse Edge Native Working Group promotes innovation and enables delivery of edge computing platforms and tools that embody the vision for EdgeOps described in this paper. The Eclipse ioFog, Eclipse zenoh, and Eclipse fog05 projects are representative of the commercial and community success to which we aspire with all of our projects. They are the cornerstones of our vision for the future of edge computing, and we offer an open invitation to use and contribute to the improvement of these technologies.



Get Involved Today

There are many ways for individuals and organizations to benefit from, and get involved with, the Eclipse Edge Native Working Group and the more than [380 projects hosted at the Eclipse Foundation](#). Options range from anonymously downloading and using the software to contributing enhancements and helping to drive strategic direction with an Eclipse Foundation membership.

Benefits of membership in the Eclipse Foundation include:

- Proven processes and best practices for open source software development
- Intellectual property management and licensing services that enable entrepreneurial collaboration and commercialization of open source software
- Mentorship, guidance, and expertise in creating and managing open source projects and in community building and support
- Marketing services that increase project visibility

Learn More

Discover the benefits of getting involved in the Eclipse Foundation:

- To learn more about the Eclipse Edge Native Working Group, visit the [website](#).
- For more information about the different ways to get involved in the Eclipse Foundation, follow the links on [this page](#).
- To learn more about the benefits of becoming a member of the Eclipse Foundation, visit our [membership page](#).

Authors

Luca Cominardi



Dr. [Luca Cominardi](#) is a Senior Technologist at [ADLINK Technology](#), working on Edge Computing and Distributed Systems technologies. He is an active contributor to the Eclipse Edge Native Working Group and Committer to Eclipse zenoh. Luca holds degrees in Computer Science and Telematics Engineering from University of Brescia, Italy, and the University Carlos III of Madrid (UC3M), Spain.



Robert Andres



Robert Andres is the Chief Strategy Officer at [Eurotech Group](#), where he is responsible for driving Edge Computing M2M, and IIoT initiatives within a wide range of markets and applications. Robert is an active contributor to the Eclipse IoT and Edge Native Working Groups at the Eclipse Foundation.



Kilton Hopkins



Kilton Hopkins is the co-founder and CTO of [Edgeworx](#), a startup that provides enterprise-scale edge computing. He is the creator of the Eclipse ioFog open source edge computing technology. Kilton is a former IoT advisor to the City of San Francisco and recently served as IoT program director for the Level program at Northeastern University.



Frédéric Desbiens



Frédéric Desbiens is the program manager for IoT and Edge Computing at the [Eclipse Foundation](#) where he helps drive community and open source innovation. Previously, he worked as a product manager, solutions architect, and developer for companies as diverse as Pivotal, Cisco, and Oracle. Frédéric holds an MBA in electronic commerce, a BAsc in Computer Science, and a BEd, all from Université Laval.



EDGE | NATIVE



BOSCH



HUAWEI





About the Eclipse Foundation

The Eclipse Foundation provides its global community of individuals and organizations a mature, scalable, and business-friendly environment for open source software collaboration and innovation.

The Foundation is home to the Eclipse IDE, Jakarta EE, and more than 375 open source projects, including runtimes, tools, and frameworks for cloud and edge applications, IoT, AI, automotive, systems engineering, distributed ledger technologies, open processor designs, and many others.

The Eclipse Foundation is a not-for-profit organization supported by more than 300 members, including industry leaders who value open source as a key enabler for their business strategies. To learn more, follow us on [Twitter](#) and [LinkedIn](#), or visit eclipse.org.