IoT Edge Working Group

EDGE NATIVE APPLICATION PRINCIPLES WHITEPAPER

AUTHORS

AMAR KAPADIA, AARNA NETWORKS; BRANDON WICK, AARNA NETWORKS; JOEL ROBERTS, CISCO; KATE GOLDENRING, FERMYON; DEJAN BOSANAC, RED HAT; TOMOYA FUJITA, SONY US LAB; RAVI CHUNDURU, VERIZON; NATALIE FISHER, VMWARE; STEVEN WONG, VMWARE.

REVIEWERS

FRÉDÉRIC DESBIENS, ECLIPSE FOUNDATION; PRAKASH RAMCHANDRAN, EOTF; MARK ABRAMS, SUSE.

CLOUD NATIVE COMPUTING FOUNDATION

PUBLISHED: Jan 17, 2023 (1st edition), October 24, 2022 (draft)

OBJECTIVE

The term "Edge Native" has been mentioned in many places including industry blogs, such as **Gartner**, **Macrometa**, and **FutureCIO**. Organizations including the **State of the Edge** and the **Linux Foundation** (LF) have also discussed edge native applications, but there has not been a focus on edge native principles.

This whitepaper focuses on edge native applications and how the principles are defined.

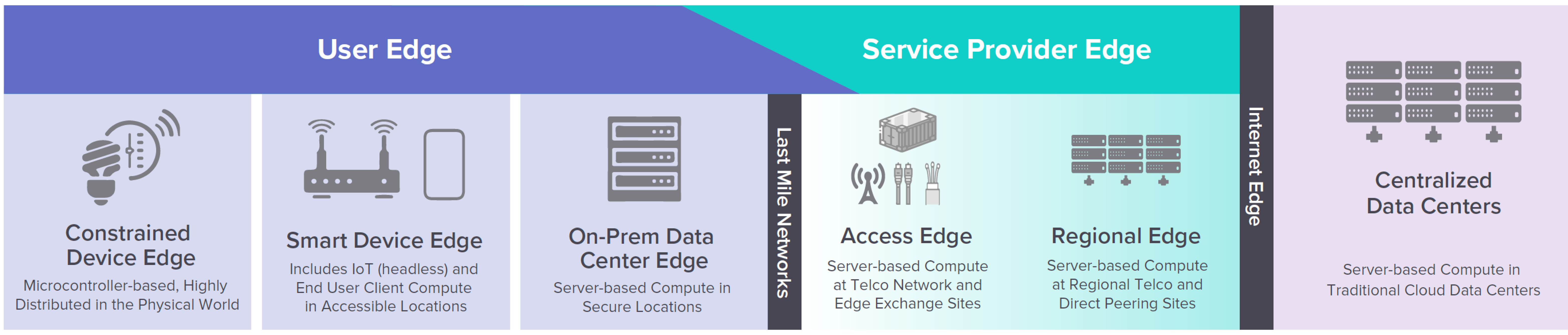
WHAT IS EDGE ?

Edge computing is a paradigm which brings data processing closer to the source, for example, robot control in a factory.

Over the next five years edge computing will become more ubiquitous as the industry is estimated to grow **38.9% from 2022 to 2030**. Companies are seeing the following benefits of having computing power at the edge:

- Reduced latency
- Bandwidth management
- Increased privacy for sensitive data
- Uninterrupted operations with unreliable networks

Various definitions of edge computing exist, but the one this paper will focus on is based on the geographical location of data processing resources. Geography-based edge is classified into multiple categories depending on the distance from the user. The diagram below shows the categories as defined by the **Linux Foundation Edge Whitepaper.**



Edge native principles share a number of similarities with cloud native principals; however, there are also some key differences.

CLOUD NATIVE VS EDGE NATIVE

Cloud native technologies, as defined by the Cloud Native Computing Foundation (CNCF), are:

> *"Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds. Containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this approach.*
>
> *These techniques enable loosely coupled systems that are resilient, manageable, and observable. Combined with robust automation, they allow engineers to make high-impact changes frequently and predictably with minimal toil."*

This broad mission remains applicable to edge applications as The Open Glossary of Edge Computing states that "edge native applications" leverage cloud native principles:

> *"An application built natively to leverage edge computing capabilities, which would be impractical or undesirable to operate in a centralized data center. Edge-native applications leverage cloud-native principles while taking into account the unique characteristics of the edge in areas such as resource constraints, security, latency and autonomy. Edge native applications are developed in ways that leverage the cloud and work in concert with upstream resources. Edge applications that don't comprehend centralized cloud compute resources, remote management, and orchestration or leverage CI/CD aren't truly "edge native", rather they more closely resemble traditional on-premises applications."*

As cloud native use cases venture out of traditional clouds to incorporate data and events at edge locations, new tools and techniques are evolving in keeping with the goal of delivering loosely coupled systems that are resilient, manageable, and observable, while also managing the unique characteristics of the edge.

SIMILARITIES

Many cloud native principles apply to edge native. This section describes these similarities.

ATTRIBUTES	CLOUD NATIVE & EDGE NATIVE
Apps and services portability	Applications and services abstract their coupling from the infrastructure. A well-written app can't tell where it is running and can be expected to be portable across platforms.
Observability	The platform is accompanied by a suite of well-documented interfaces and tool options to enable detection of issues and collection of metrics. This allows developers to build systems that are resilient and efficiently managed.
Manageability	Interfaces and tooling options are provided to manage apps and resources at scale. The platform also has a plug-in mechanism to provide baseline network connectivity, services, and management features.
Agnostic language and framework support	Apps and services can be implemented and hosted using a variety of popular languages and frameworks.

DIFFERENCES

The broad missions of edge native and cloud native share similarities, yet developers should be aware of the differences.

ATTRIBUTES	CLOUD NATIVE	EDGE NATIVE
App Model	Mostly microservice components that are built stateless for load balanced horizontal scaling.	While a service provider edge app may be quite similar, user edge apps may be singleton "monolithic"; in both cases, the state may be colocated with the app.
Data Model	Centralized model backing stateless components is common.	Caching, streaming, real-time, and distributed models are often utilized.
Elasticity	Rapid spin up and down; generally treating underlying resources as unlimited.	Limited elasticity due to constrained hardware resources at the edge; if available, scaling might be "vertical" up to a connected cloud.
Resilience	Resilience outsourced to cloud providers using redundant nodes spread across failure domains.	Often relies on hardened infrastructure, with recovery architectures for stateful components; in many cases, resilience may be lower than in the cloud.
Scale	Typically limited to a few locations, instances.	May span a large number of locations (up to 10,000s) with support for a large number of external devices (up to 100,000s).
Orchestration	Orchestration in large public or on prem cloud is designed to achieve efficiency and availability by running workloads on centrally pooled hosts, scheduled in a horizontal way.	Edge is decentralized with workloads deployed in a distributed way, often scheduled in a location-specific way.
Management	While both cloud native and edge native are manageable, the mechanism varies; cloud native relies on central management and automation.	Edge native requires a mix of remote and centralized management and zero touch provisioning of hardware and software. Staffing at the edge may be untrained, untrusted, minimal or even non-existent. "Brick" resistant "atomic" upgrades become desirable because of recovery challenges.
Networking	Apps can rely on high speed networks with rich capabilities.	Apps need to account for varied speeds (ranging from intermittent and poor, to excellent) and capabilities. Includes mobile and radio based, integrating data and events from non-IP protocol networks.
Security	Trusted fabric within secure facilities.	Zero trust in insecure environments.
Hardware awareness	Apps rarely have to concern themselves with hardware placement due to de facto uniformity of compute resources with flavors that cater to most apps.	Apps may have greater real time requirements making hardware platform, location, and/or security awareness a requirement. Developers need to be aware of a wider variety of hardware and interfaces.
Interacting with external resources	Applications rarely need to interact with local hardware resources.	Services deployed at the edge often need to interact with the local environment: cameras, sensors, actuators, users, and more.

EDGE NATIVE APPLICATIONS

Edge native applications are applications and services written for the edge. They are written in a way that accounts for the above similarities and differences. The core principles for these applications are listed below.

EDGE NATIVE PRINCIPLES

Edge native applications should follow the following principles to fulfill the mission of edge native mentioned earlier in the paper.

	PRINCIPLE	DESCRIPTION
RESOURCE & DEVICE AWARE	Hardware aware	Rather than having homogenous hardware platforms, developers need to be aware of a wider variety of hardware and interfaces.
	External device connectivity	Applications must know how to connect to the devices in their environment and be aware of the capability changes at runtime. For example, they must respond to dis/connecting sensors to the edge server after initial configuration or new devices entering the network. Capability is not static; rather it includes the environment, so orchestrators should be able to reconcile capability changes to application state.
	Aware of variable-connectivity	Applications must adapt to network connectivity that is unreliable or even unavailable (air gapped) by using mechanisms such as asynchronous communication, queueing and caches. "Pull" mechanisms may be required to overcome scale, network connectivity, and security issues where the edge pulls configuration from a central site.
AT SCALE MANAGEMENT	Centrally observable	While both edge and cloud native applications need to be centrally observable, edge native apps have unique considerations. Edge native app instances may be deployed at a massive scale of instances and/or face constrained staff or field support. For these reasons, techniques such as distributed data collection and centralized aggregation, a combination of open loop (human observable / actionable) and closed loop automation (machine actionable) are required. Observability covers metrics, logs, digital twins, alerting (events and alarms) and health monitoring.
	Infrastructure and platform management at scale	Infrastructure and platform management at scale is important for edge applications thus requiring declarative intent-based mechanisms. Furthermore, there might be unique requirements such as device onboarding, limitations on horizontal scaling, managing bare metal environments, and more. At a platform level, deploying or managing the Kubernetes or virtualization layers and various plugins is also a concern; all while keeping the platform layer as vendor neutral as possible to enable application portability.
	Application management at scale	The number of applications and the number of instances of these applications can be very large on the edge, requiring automation ranging from declarative placement and configuration intent, automated service assurance via closed loop automation, and aggregated management views across multiple application instances. Applications may also have realtime needs which means the linkage between applications and infrastructure/platform (e.g. use of GPUs, DPUs, FPGAs, CPU architectures, kernel optimizations, Kubernetes plugins) may be tighter than with cloud applications. In other words, application orchestration may trigger infrastructure/ platform orchestration underneath.
SPANNING		Applications do not exist at one location; tiers can straddle geographic, latency, and failure domain boundaries. In fact, edge apps may span public or centralized private clouds as well.
RESOURCE USAGE OPTIMIZATION		Since edge computing resources are constrained, an application has to optimize resource usage on a continuous basis. This may mean spinning applications up and down on-demand. It may mean migration and replication based on placement and availability intent. It may mean running different workloads at different times of the day.
APPLICATIONS ARE PORTABLE AND REUSABLE (WITH LIMITS)		Abstraction layers attempt to provide infrastructure and platform agnostic portability through vendor neutral PaaS. However, because of local resource, hardware platform, security, mobile network and other awareness, configuration options are required to adapt to local differences.

These nine principles can be grouped into a smaller set of five principles. Hardware awareness, external device connectivity, and awareness of variable network availability can all be considered under a broader principle of resource and device awareness. Similarly, the need for edge applications to be manageable at scale, centrally observable, and have manageable infrastructure and platforms can all be categorized under a principle of at scale management. The following are the broadened five principles: Spanning, Resource Usage Optimization, Portable and Reusable with Limits, Resource and Device Aware, and at Scale Management.

CONCLUSION & NEXT STEPS

This paper is a first edition and may experience revisions. Future papers related to sub-sections of this paper are anticipated.

HOW TO GET INVOLVED

The CNCF IoT Edge Working Group has regular meetings, a mailing list, and a Slack. See the **Communication section** of the working group GitHub page for the most up-to-date information. We welcome the reader to get involved by presenting Edge related projects, bringing an idea for an area of work for the group, or helping to revise this whitepaper and/or draft follow on paper.

PROJECTS AND INITIATIVES

As a part of this whitepaper, the CNCF IoT Edge Working Group is collecting a working list of open source projects that help application developers achieve the edge native application principles outlined in this paper.

The list can be found in **this spreadsheet** or through the QR code. To get edit access to add a project, join the **IoT Edge Working Group Google group**.

REFERENCES

Linux Foundation Edge Whitepaper:
https://www.lfedge.org/wp-content/uploads/2020/07/LFedge_Whitepaper.pdf

Open Glossary of Edge Computing [v2.1.0] State of the Edge:
https://github.com/State-of-the-Edge/glossary/blob/master/edge-glossary.md#edge-native-application

Cloud Native Computing Foundation (CNCF) Charter:
https://github.com/cncf/foundation/blob/main/charter.md

Gartner 'Cloud Native Isn't Edge Native':
https://blogs.gartner.com/thomas_bittman/2020/04/17/cloud-native-isnt-edge-native/

Macrometa 'Edge Native is not Cloud Native':
https://www.macrometa.com/blog/edge-native-is-not-cloud-native

Future CIO 'Cloud-Native versus Edge-Native: know the difference':
https://futurecio.tech/cloud-native-versus-edge-native-know-the-difference/

Edge Computing Market Size, Share & Trends Analysis Report By Component (Hardware, Software, Services, Edge-managed Platforms), By Application, By Industry Vertical, By Region, And Segment Forecasts, 2022 - 2030
https://www.grandviewresearch.com/industry-analysis/edge-computing-market