DELL Technologies

Blade Runner – Edge Architecture

Architecting ad-hoc composable infrastructure at the Edge

AUTHOR(S)

Eric Bruno, Dragan Savic, Joel Christner

Table of Contents

Abstract

Blade Runner is an architecture and vision for ad-hoc composable infrastructure at the Edge, to be used in wide range of edge application use cases. In addition to novel composability features, it includes features such as federated data storage and efficient real-time workload processing. The goal is to form an edge architecture for the near future, where an edge computing deployment of any size can be embodied independent of the cloud, to solve the needs of today's and tomorrow's edge applications. Although designed to stand alone, without requiring cloud resources, it will also work with public and private clouds, and enable multi-cloud application deployment as well. As a result, this edge architecture will include the high-level services provided by modern cloud providers, including container and VM workload definition, deployment, scheduling, and management; distributed and federated storage (with intelligence and guaranteed durability); dynamic node composability and cluster formation; and cluster management features specific to cloud application needs.

Background and Related Work

The following works are related:

1. OCTO-2795 – Define Infrastructure and Workload Attributes: Define attributes to describe and match both the capabilities of edge infrastructure and the requirements of workloads. This declarative method of describing edge equipment, together with the associated (and compatible) method of describing workload requirements leads to an edge solution that self-deploys and self-maintains the ideal system state.
2. OCTO-2469 – Overall MCCP Architecture - Design the architecture for the project Multi-Cloud Control Plane (MCCP) and implement it with limited working scenario. The study will provide some common scenarios for scheduling workload across multiple Kubernetes clusters, core services for analyzing workload and optimizing workload scheduling.
3. Project Amalfi – a virtual storage server that virtualizes underlying repositories (S3, Azure, ECS, disk) and exposes them to clients using the RESTful Amazon S3 API.
4. OCTO-2422 – Disaggregated Memory - Provide current status as well as a summary of the academia and industry research efforts in this area and explain why memory disaggregation is a difficult task.
5. Market-based Distributed Resource Allocation for IoT-EdgeCloud Systems - Ana Oliveira, et al
6. OCTO-2214 – Google Anthos on Dell Hardware – Characterize the low and high-end Anthos requirements and specifications to run Anthos on VSphere on a PowerFlex rack.
7. Multi-Cloud Microservices Based Architecture – an overview of a software framework intended to deploy and manage Applications which are constructed as a composition of stateless microservices that can reside on multiple clouds, either public or private.

Hypothesis

A need exists to define an edge-native architecture and environment that developers can target to build and deploy their edge applications, as opposed to adapting applications to be distributed across bespoke assemblies of data centers, cloud services, and MECs.

Goals

Define an ad-hoc composable infrastructure at the Edge for federated data storage and efficient real-time workload processing. There will be five initial components, all focused on edge computing, as part of a unified design:

- Federated Edge Storage
- Edge Composability
- Edge Workload Optimization and Load Balancing
- Advanced Edge Cluster Management
- Multi-cloud integration and enablement.

Key Insights

1. Novel storage enhancements for edge (i.e. federation, durability, distribution, etc.) are identified, initially designed, and proposed for embodiment in future project/PoC efforts, to best meet the needs of growing edge data demands.

2. A novel yet practical take on how multiple dimensions of composability can be implemented and made feasible at the edge to drive maximum efficiency of edge application processing has been specified and made ready for future PoC efforts.

3. Integrating edge workload orchestration with Kubernetes and other novel workload balancing techniques combine to form an edge vision of autonomous workload deployment and management, making it easier to deploy and manage edge applications.

Overview of Blade Runner

Blade Runner is the edge architecture for the near future, where an edge computing deployment of any size can be formed to stand alone and solve the needs of today's and tomorrow's edge data and application processing needs. Although designed to stand alone without requiring cloud resources, it will also work with public and private clouds, and enable multi-cloud application deployments as well. As a result, the architecture will include the high-level services provided by modern cloud providers, including container and VM workload definition, deployment, scheduling, and management; distributed and federated storage (with intelligence and guaranteed durability); dynamic node composability and cluster formation; and cluster management features specific to edge application needs.

Federated Edge Storage

Data is naturally distributed across the edge—it lives there! Instead of moving it all to a data center, colocation facility, or cloud, all of which are costly and latent, there needs to be an ability to query it in place, across the edges of an application. The reasons are two-fold: data often originates at the edge, and data needs to reside there for low-latency access to enable workloads that live in that edge. Often, data ends up in a private or public cloud for durability reasons. However, using erasure coding and advanced distribution techniques, data can be distributed amongst edge locations rather than all edge sites. Instead, it can be intelligently distributed and handled in an eventually consistent manner. However, 100% durability is still a goal. The edge becomes a widely distributed RAID system, where distributed data can be recovered even in the event of the loss of a set number of edge sites (see Figure 1).

Most edge applications include data capture (ingestion) and processing (including inferencing and analytics) at edge locations spread across geography. However, most redundancy in terms of primary storage is within the site or within centralized storage. These are not always feasible for continuous data operations in highly distributed edge deployments. This edge storage design aims to solve this while eliminating the need to replicate data everywhere, including both private cloud and expensive cloud storage. Within the proposed design, groups of edge nodes will work as a single RAID-like storage

system with 100% durability and eventual (yet approaching immediate) consistency. Instead of replicating all data across all edge locations, only state and metadata is shared (as with a RAID system) across remote edge nodes. Missing edge data sets can then be reconstructed and accessed at remote locations in the event a single edge server/location/storage device goes offline, becomes unavailable, is otherwise lost or destroyed, or it is determined that the data fails a checksum or other validity test. The detection and recovery processes will be automated, using optimal storage nodes for data recovery.

This design eliminates the loss of data, ensures continuous availability of critical data across remote edge locations in the event of failures (however permanent or transient they may be) without requiring the data be replicated across all edge locations or stored centrally as a whole. This edge storage design will ensure efficient data availability across locations in a more cost-effective way than it's done today (for example, complete data set replication, or cloud-based data movement). Additionally, this storage solution will be part of an intelligent edge storage solution, employing computational storage to move workloads to where the data resides, which is an important requirement for most edge applications.

Overall, this storage design will act as a *worldwide edge storage pool,* having distributed edge storage behave like a large RAID system. The front-end will initially be based on the RESTful S3 API (likely based on Project Amalfi) and will include the requisite components to enable a generic, global, replicated BLOB store that will internally shard data and replicate intelligently. A PoC will explore how to leverage lessons learned from erasure coding and other techniques to shard and replicate, and how conflict-free replicated data types (CRDTs) may help with global synchronization of changes across edge locations. In summary, it will have the following features:

- Shared storage across a distributed deployment
- A single consistent view of data at the organization level across applications
- 100% durability and eventual (yet approaching immediate) consistency
- No single point of failure
- No need to replicate all data across all locations
    - Only replicate enough to recreate missing data if a site failure occurs
- Utilizes computational storage to drive new levels of data processing performance, efficiency, and security, where data resides
- Edge storage state propagation across the storage federation
- Supports automated discovery of data loss and kick-off of the data recovery process based on the most efficient set of nodes to include

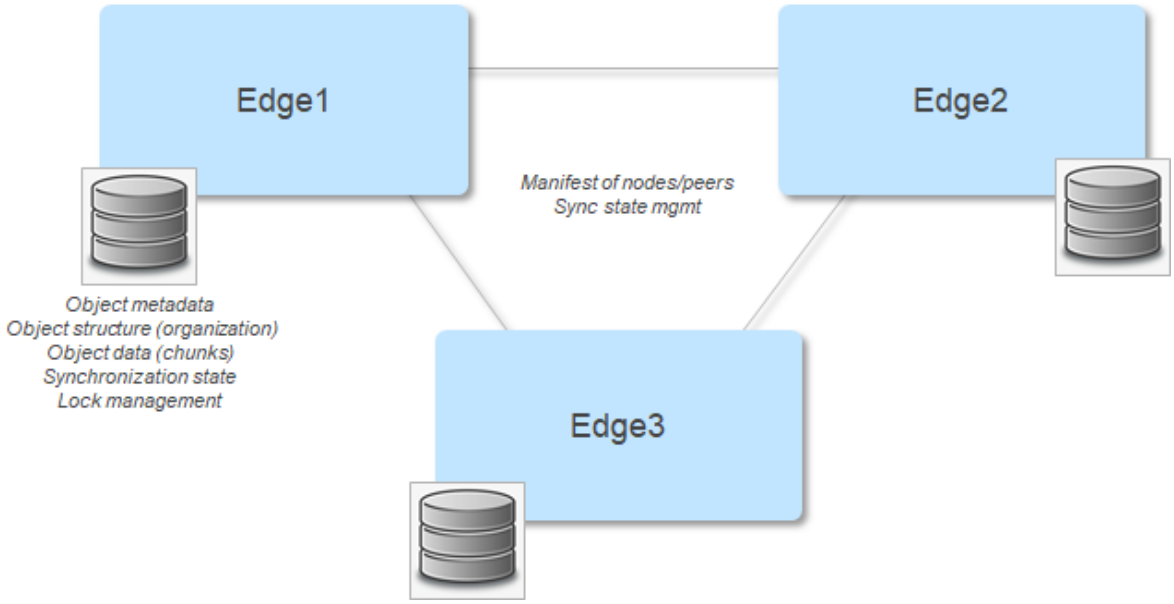

Figure 1 - Edge application storage can be spread across multiple edge locations

Replication across edge sites will be handed via lightweight communication mechanism such as WebSockets (or similar persistent implementation). Additionally, edge storage will support a front-end S3 interface and the underlying storage management components needed to intelligently synchronize metadata and application data across edge sites to ensure durability. This is called the Edge Storage Synchronization System, and is outlined in Figure 2.

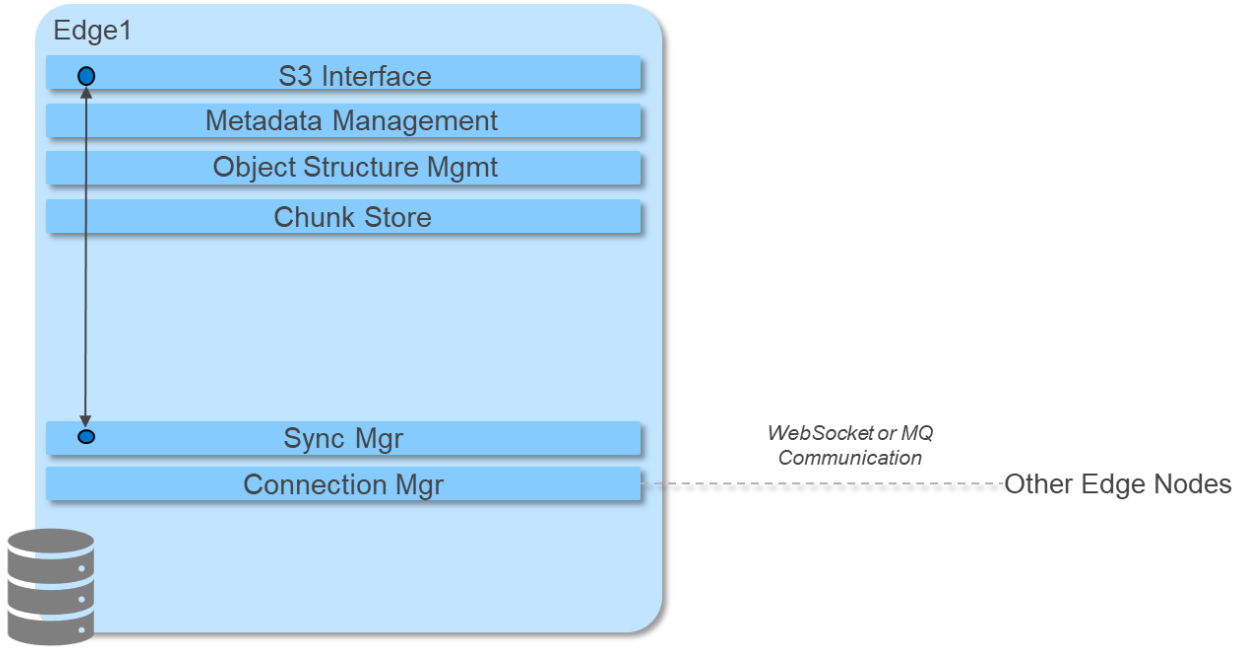

Figure 2 - Edge Storage Synchronization System

Assume an edge deployment involving multiple edge locations, and optionally cloud storage. As stated above, the goals are:

- Ensure data is readable/writeable from anywhere
- Ensure loss of an edge site will not result in loss of data
- Avoid replication (mirroring) across all edge locations

While the initial scope will be centered around object, future iterations will consider other types of storage including file and block. It would be ideal to have a horizontally scalable block storage platform, although this can be challenging due to the constraints and impact created by IO latency in underlying kernel-mode disk operations. Instead, work will begin by distributing object storage across the edge, and exploration around block will be considered after.

Metadata

Metadata will be maintained at each edge site for objects along with erasure coding (or similar) information related to the distribution of data shards. The assumption is that metadata represents a small subset of total data (~1%), providing a globally consistent view of contents, and can be used to reconstruct a lost edge site's data by understanding the physical location of underlying data shard replicas. For the first PoC phase, metadata will be limited to object storage and will later be enhanced to encapsulate a multitude of storage modalities (object, file, block).

Corner Cases

The following are corner cases that need to be resolved while proving out distributed edge object storage ideas:

1. Recreation of lost data due to device, site, or other failure
    a. How the object will be reconstructed
    b. Optimal mechanisms for shard retrieval from replicas, or
    c. Recreation of data from erasure coding
2. Loss of >n edge sites results in inaccessible data (likely temporary, but could be permanent)
3. Out of sync edge sites (eventual consistency)
    a. Reconnect and retrieve deltas
    b. Potential inaccessibility during a resynchronization
    c. Cluster split-brain handling

PoC First Phase

We'll start with a single, global, namespace, using generic BLOB storage with object storage as the first interface. At a later phase, we'll determine if it's feasible to solve federated file and block storage. Given that Blade Runner edge storage system implements an S3 interface, it can also work as a back-end to Project Amalfi for virtual storage support.

Edge Streaming Data Storage and Analytics

An additional plan, as a separate offering, includes a simple NoSQL/Object database built around JSON Schema, which will be used to validate data writes, and to create a relational table structure on top (see Figure 3).

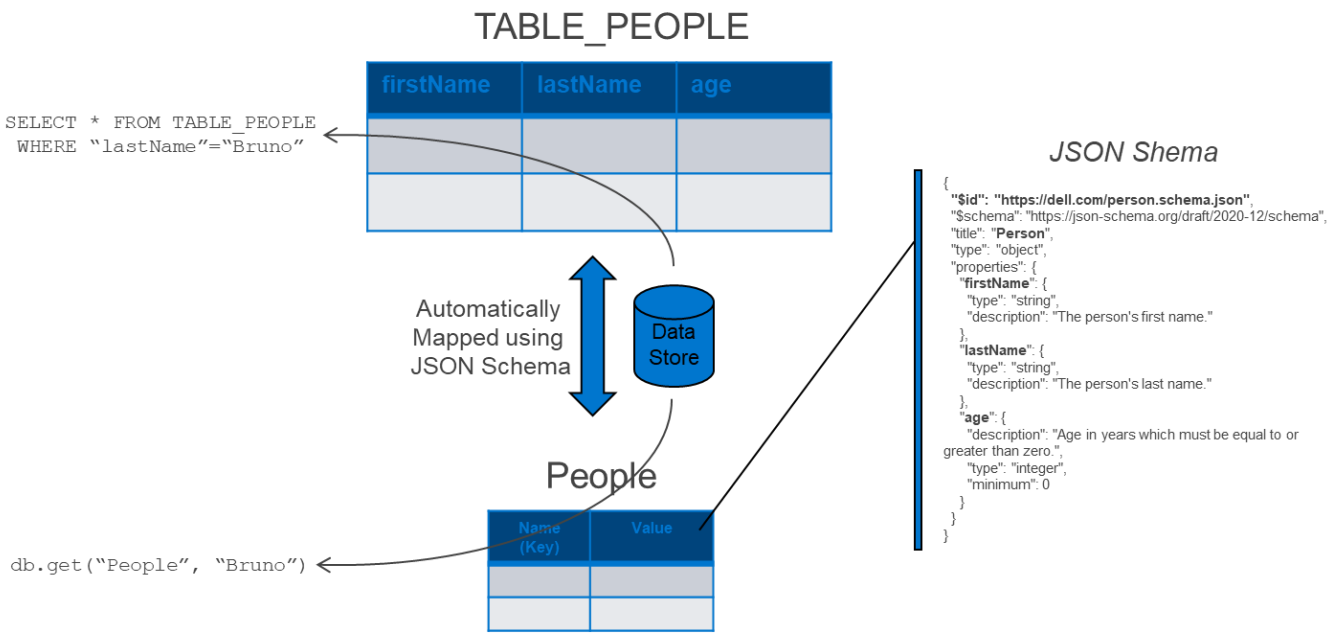

Figure 3 - Dual table and name-value data access via JSON Schema

A basic SQL parser will be supported on top, giving applications a choice of calling into the NoSQL API, or making SQL calls with possible ODBC/JDBC support (see Figure 4).

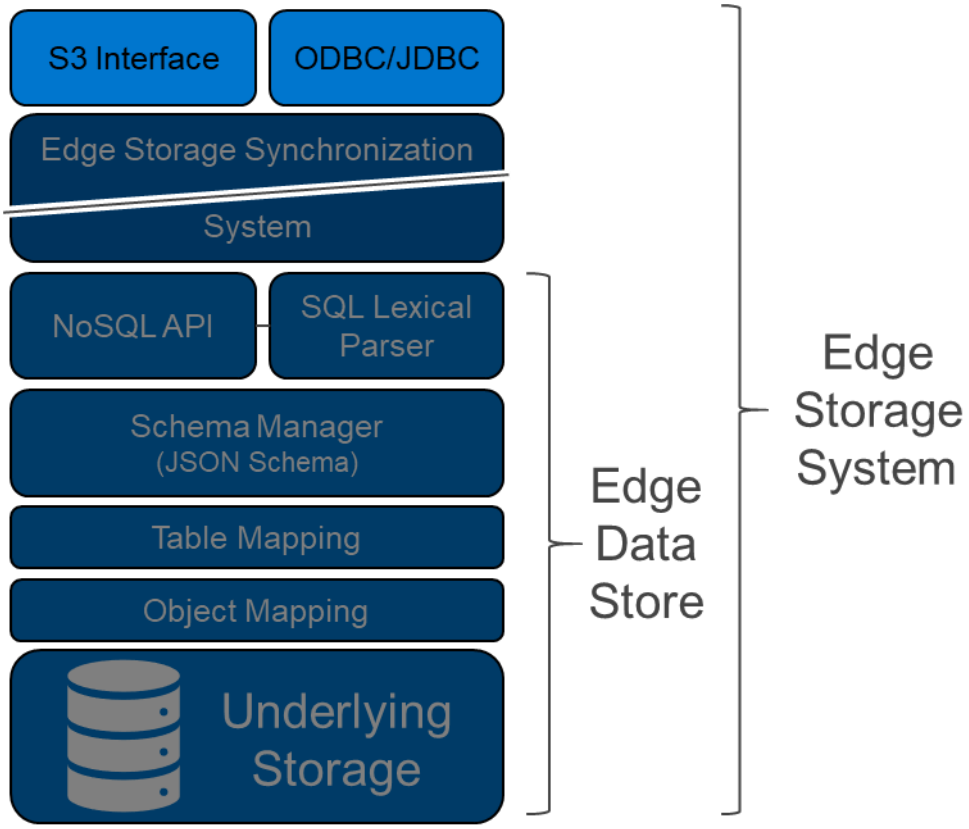

Figure 4 - High level design for the edge storage system PoC

In addition to the federated edge storage PoC described above, the following section explores using it as part of a real-time streaming analytics platform for the edge as a separate offering. To support streaming data analytics and storage, the Edge Storage PoC will integrate with Dell Storage Data Platform / Pravega.io.

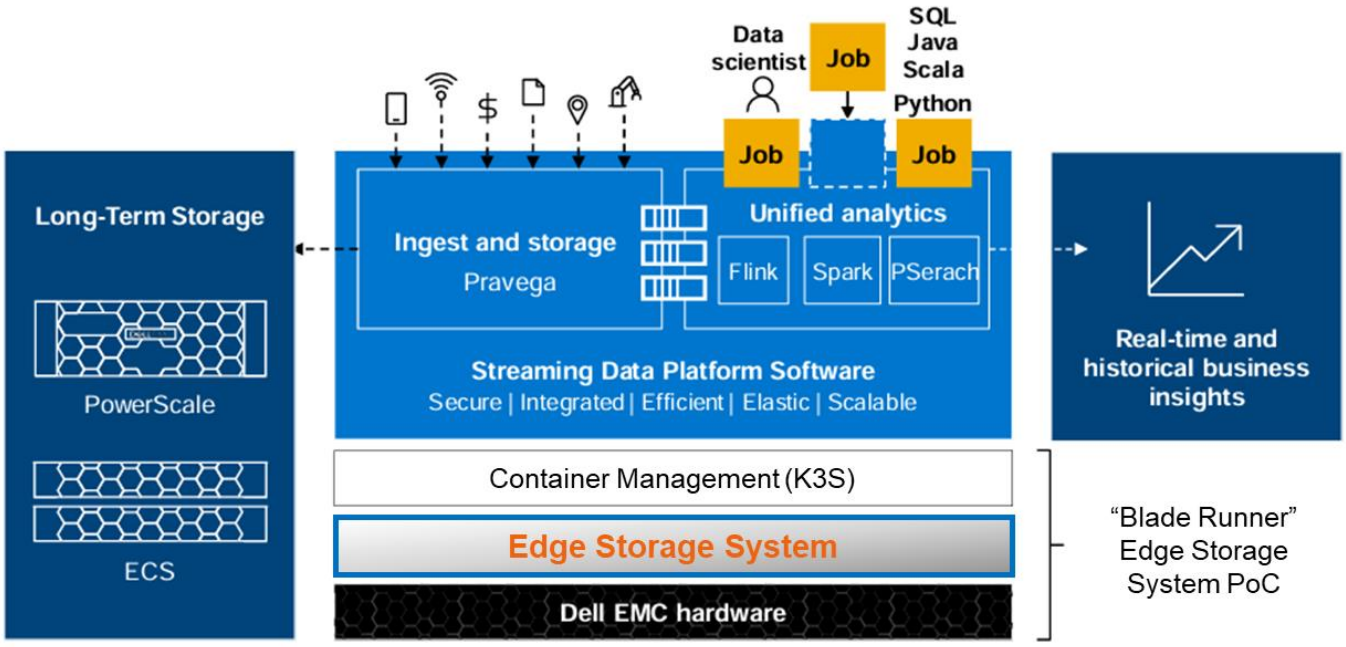

Figure 5 - Edge Storage System integrated with Dell Storage Data Platform

The goal will be to demonstrate seamless, and friction-free, integration of real-time streaming data ingestion with analytics and storage.

Security and Privacy

With computational storage, privacy-enhancing computation can be implemented to protect data while it's being used. This is achieved through a trusted environment for sensitive data, by performing processing and analytics in a decentralized manner, and encrypting data and algorithms before processing. This approach is designed specifically to meet the growing need and challenge to share data at the edge while maintaining privacy or security.

Tiered Storage

The PoC will explore the feasibility of implementing tiered storage for the edge, where data is automatically persisted and offloaded to the correct tier of storage according to data policy (see Figure 6 - Tiered storage for cloud, edge, and far-edge deployments). For instance, if the data policy for specific data types/sources specifies the highest levels of performance or lowest latency, data will be automatically stored in persistent memory if available, or the closest matching local storage otherwise. For data that needs long-term persistence or where ultra-low latency isn't required, the data will be offloaded to a different tier of storage, perhaps off-site. The decisions as to where to store the data to and load it from will be made by the Blade Runner edge platform, based on declarative attributes as part of the data policy.

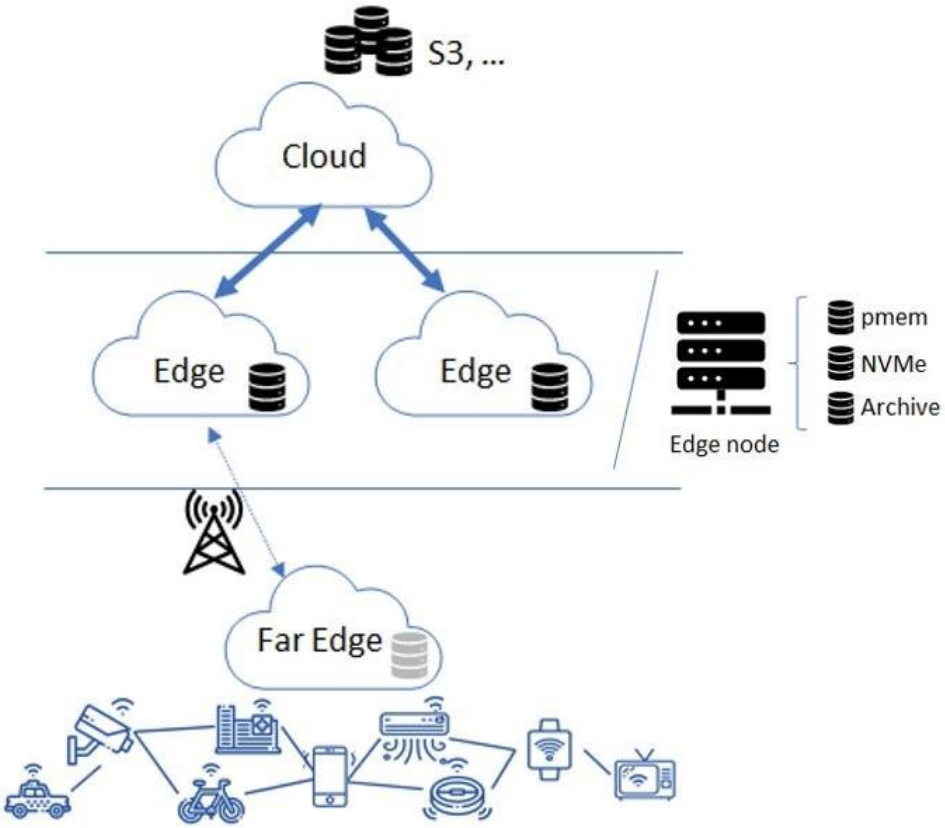

Figure 6 - Tiered storage for cloud, edge, and far-edge deployments

Expected Benefits

- 100% durability with eventual consistency
- Truly global namespace
- Read and write data from any edge location
- Manage capacity as a single system
- Privacy-enhancing computation
- Global data and service availability with near local performance
- Flexible access via NoSQL and SQL interfaces, *if feasible* due to challenges of layering on top of Object storage.
- Integration with streaming data processing frameworks and storage solutions
- Ability to automatically take advantage of tiers of storage according to performance, latency, and cost metrics
- The use of computational storage will be examined for optimization and security of edge workloads, but this will be explored in detail in the sections on composability and workload optimization in later phases.

Edge Composability

Blade Runner defines how edge nodes can be dynamically configured and reconfigured to serve edge applications and workloads optimally. For example, an edge node can take on the persona of computational storage, high-performance compute, GPU workload processing for AI, digital twin, and so on, SmartNIC, and others, according to application requirements (see Figure 7).

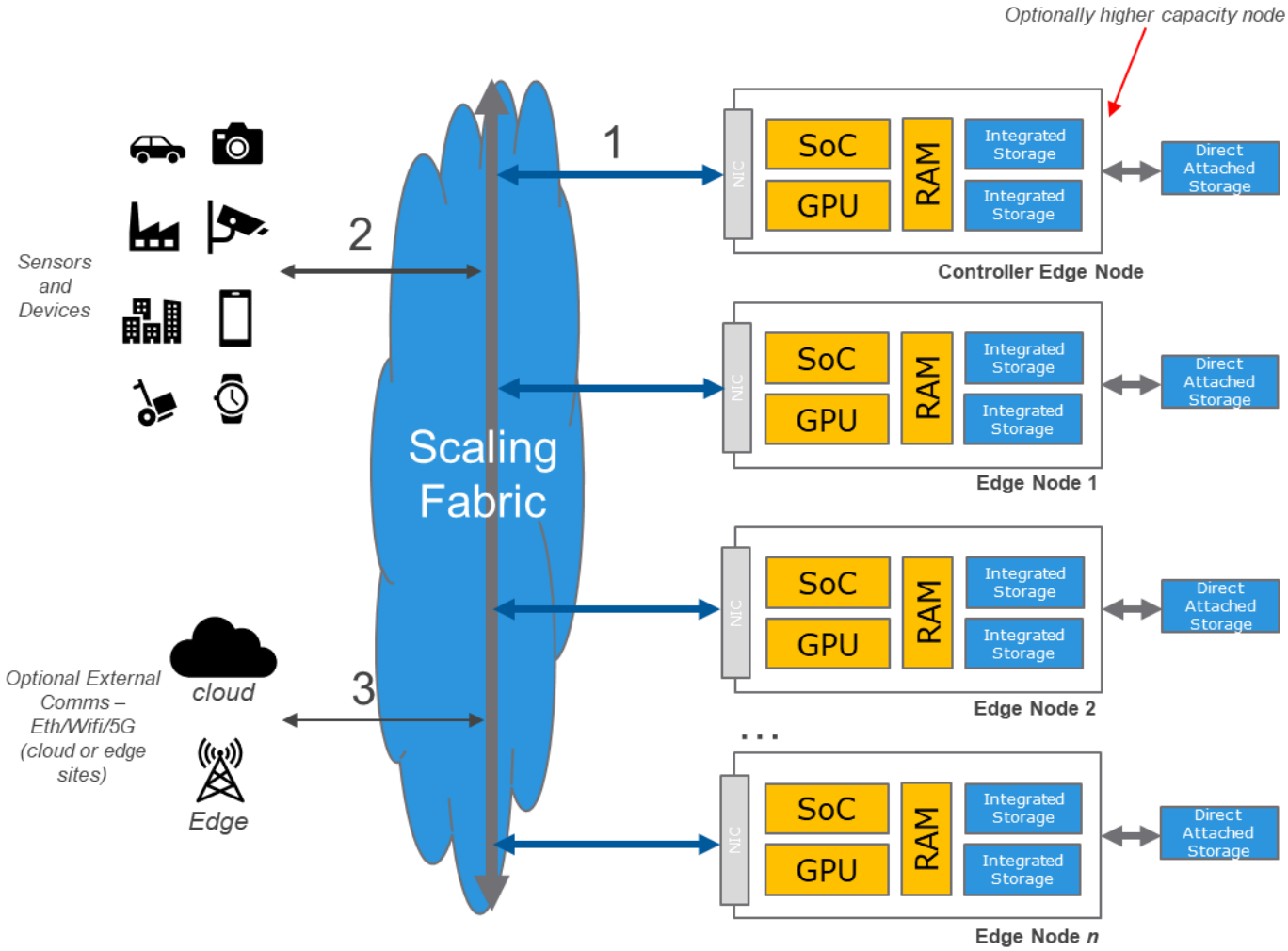

Figure 7 - Composable edge nodes, mostly equally configured

In this architecture, there are three classes of connectivity:

1. High-speed edge fabric communication
2. Low-latency sensor/device communication
3. Optional internet communication to other edge locations and/or the cloud

This connectivity takes place over a high-speed communication fabric that implements special features such as NVMe over fabric, low-latency communication protocols, and secure communication with other edge sights (i.e. MQTT, CoAP, and so on) as well as the cloud (i.e. REST over HTTPS).

Edge nodes can be combined dynamically to execute a workload in concert with other edge nodes via combined resources (i.e., additional RAM, GPU cores, and so on). These two vectors of composability (dynamic node personas, and dynamic node aggregation) lead to the ad-hoc nature of edge composability needed to process changing workloads in a dynamic edge deployment (i.e. the Telco Edge). In this sense, edge nodes can be dynamically configured and reconfigured as the following:

- SmartNIC
- Computational Storage node
- Compute node
- GPU node (AI, digital twin, GPGPU)
- Smart routing node

The hardware will switch off unused components to save on power and cooling costs. For example, if only a node's GPU is required, other nonessential parts should be turned off. Additionally, edge nodes can be dynamically configured into groups (peer relationship) or hierarchies (primary-secondary relationship):

- Two Compute Nodes combine to process a workload that requires extra processing power, storage, or RAM.
- One Compute Node uses another edge node as a SmartNIC, or as a high-speed smart storage node (see Figure 8).

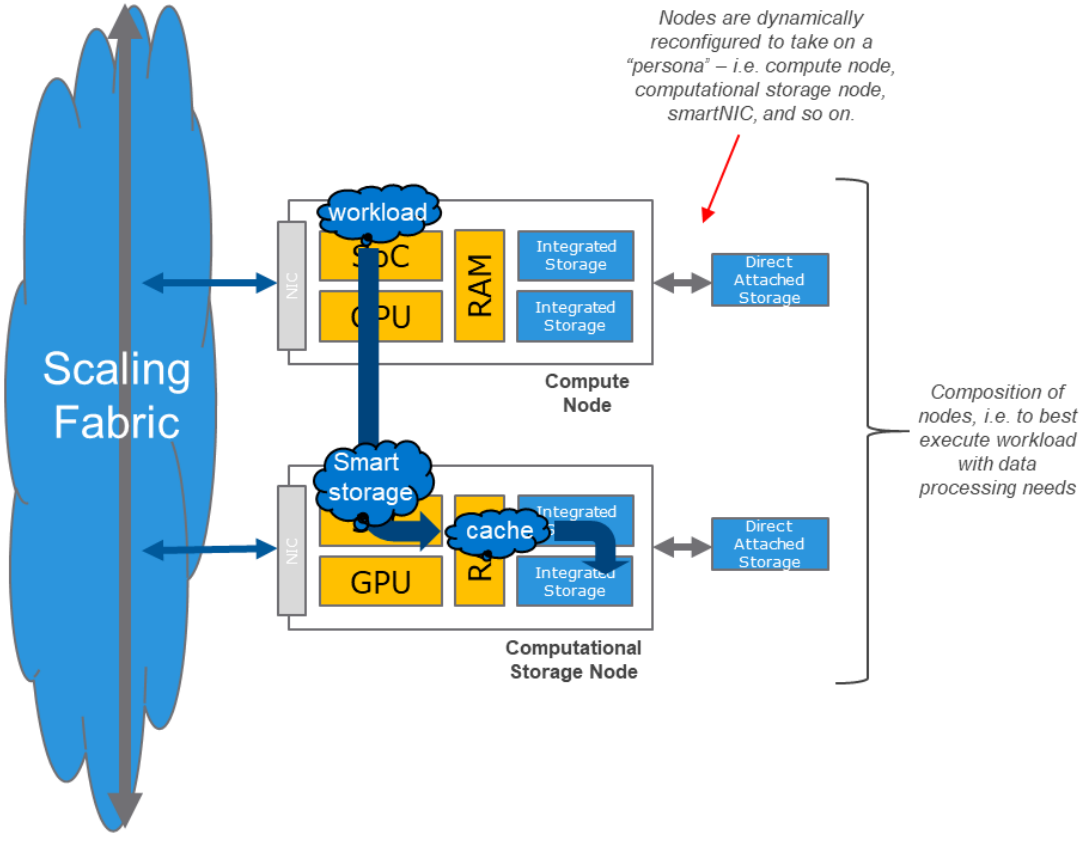

Figure 8 - Composable edge nodes combine to efficiently execute unique edge workloads

With this approach, not only is there an advantage of composability, but also those of parallelism and locality. For example, providing a way to perform smart storage with processing done where the data lives. Moving data consumes up to two orders of magnitude higher energy than processing it locally.

PoC for Novel Edge Composability

This PoC will look to implement the dimensions of edge composability from the following perspectives:

- The use of homogeneous edge hardware - each node equally equipped
- Edge nodes take on a "persona" based on workload requirements that drive dynamic configuration changes
    - For example, say all edge nodes at an edge site are equally equipped, one node will be dynamically configured to act as computational storage, where it handles data specific processing such as filtering and aggregation, another node as a SmartNIC, and a third node as a compute node for the main workload processing.
    - This edge composability framework will divvy up the work intelligently across these three nodes to be executed in parallel. Once that application or workload is completed, the nodes can be reconfigured for the next application.
- The composable part is based on how the nodes assume a context-aware persona, and then work together to process a workload
- The use of heterogeneous edge hardware
    - Use policies with well-defined and granular attributes that describe infrastructure so that this can be extended to a heterogeneous edge environment

Edge Workload Optimization and Load Balancing

Although work has been done regarding workload scheduling and orchestration at the edge, there is a need to explore it in context to data locality, continuous optimization, and composability. Additionally, workload stealing for balancing and efficient parallel processing can be expanded beyond what has been done to date to integrate Kubernetes and K3S, with prioritized scheduling based on unique workload requirements (i.e. latency).

Workloads will be scheduled at the edge by an orchestrator/algorithm (much like MCCP), but subject to continuous re-optimization based on changing edge node and network conditions. This will be expanded to utilize workload attributes to target candidate edge nodes for each workload.

With this algorithm, an underutilized node steals a request from another (busy) node that meets workload requirements (see Figure 9). Past work in this area will be expanded to consider rack-based node workload stealing versus cross-rack workload stealing, and even stealing (or failover) across edge locations. Research will be done to consider how work stealing algorithms can be integrated with K8S/K3S pod resource allocation.

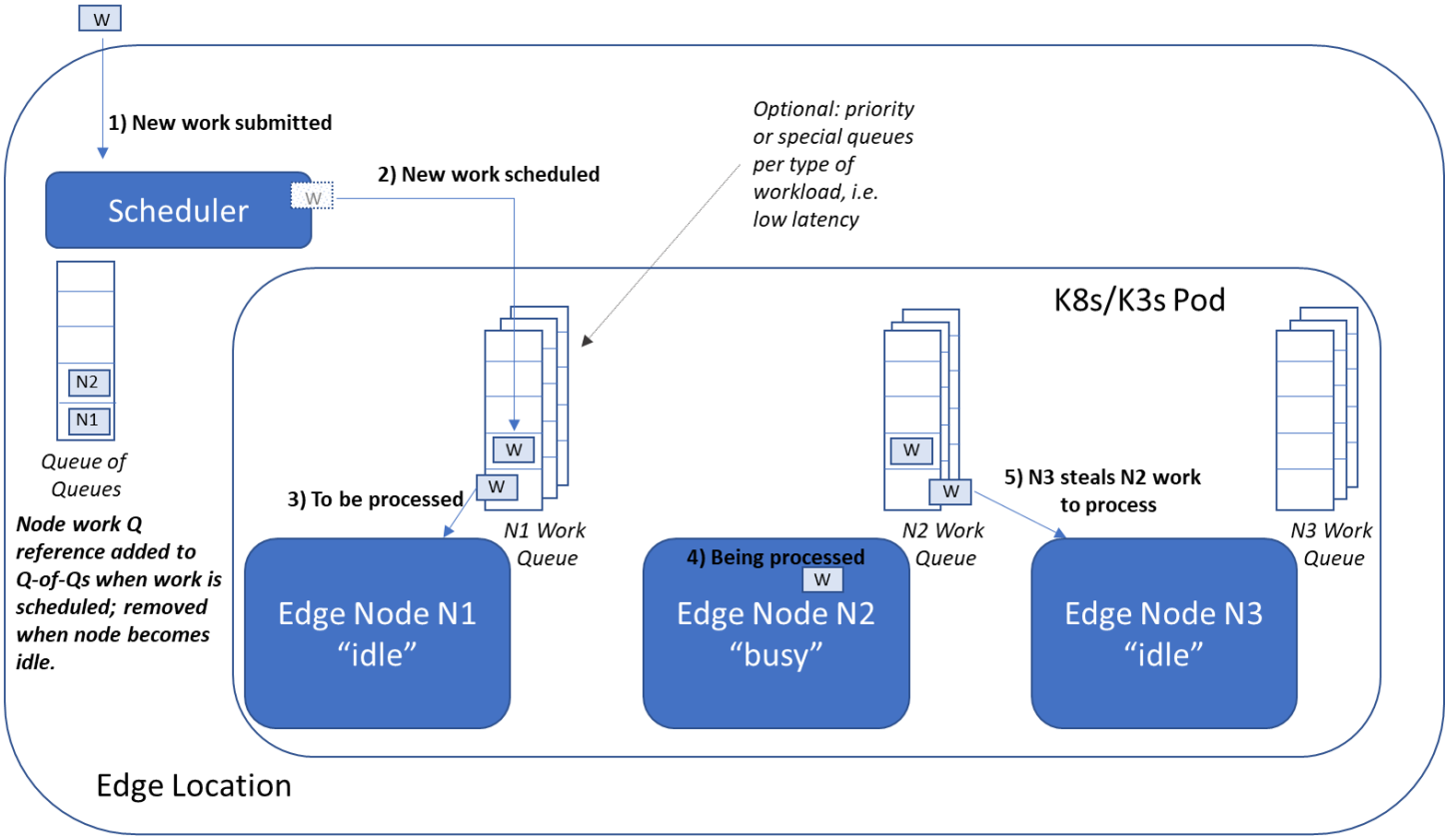

Figure 9 - Edge node workload stealing to efficiently self-regulate a balance processing

Like workload stealing, edge nodes can preemptively choose to move workloads to other edge node queues in what's referred to as workload deferral or delegation (see Figure 9).

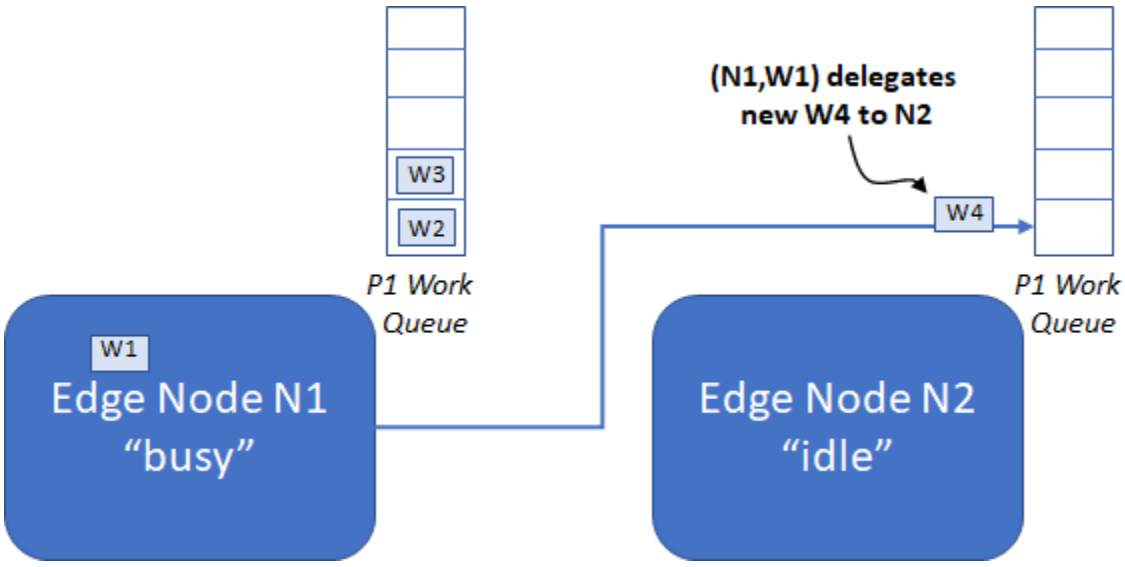

Figure 10 - Edge workload delegation (or deferral)

In this case, an overutilized node delegates a request to another node preemptively (one that meets workload requirements) without depending on or waiting for the workload to be stolen by another edge node. In this case, while processing workload W1, a new workload W4 is spawn as a result. Instead of enqueuing it within its own work queue, edge node N1 delegates the workload to node N2.

Advanced Edge Cluster Management

The above sections on edge architecture need to be integrated with modern orchestrators such as K8s and K3s to support container and VM-based workloads, small edge devices, and durable storage at the edge in multi-active implementations. Immutability will be explored here to help solve problems associated with edge configuration drift. This will be explored at the container and VM level, and possibly all the way down to the kernel or device firmware as well.

- Overall architectural integration with Kubernetes
- Immutable Edge Infrastructure: greatly reduces configuration issues related to drift by eliminating chance of changes
- Ability to dynamically assemble an edge cluster by allowing third parties to autonomously join and leave.

A project will be defined that uses the output study on edge attribute, along with available tools (K8s, Prometheus, and Admiralty) to embody these concepts and prove their applicability for workload placement and orchestration via a POC (see Figure 11). The attribute model will be extended to perform multi-cloud arbitrage for abstraction and workload placement.

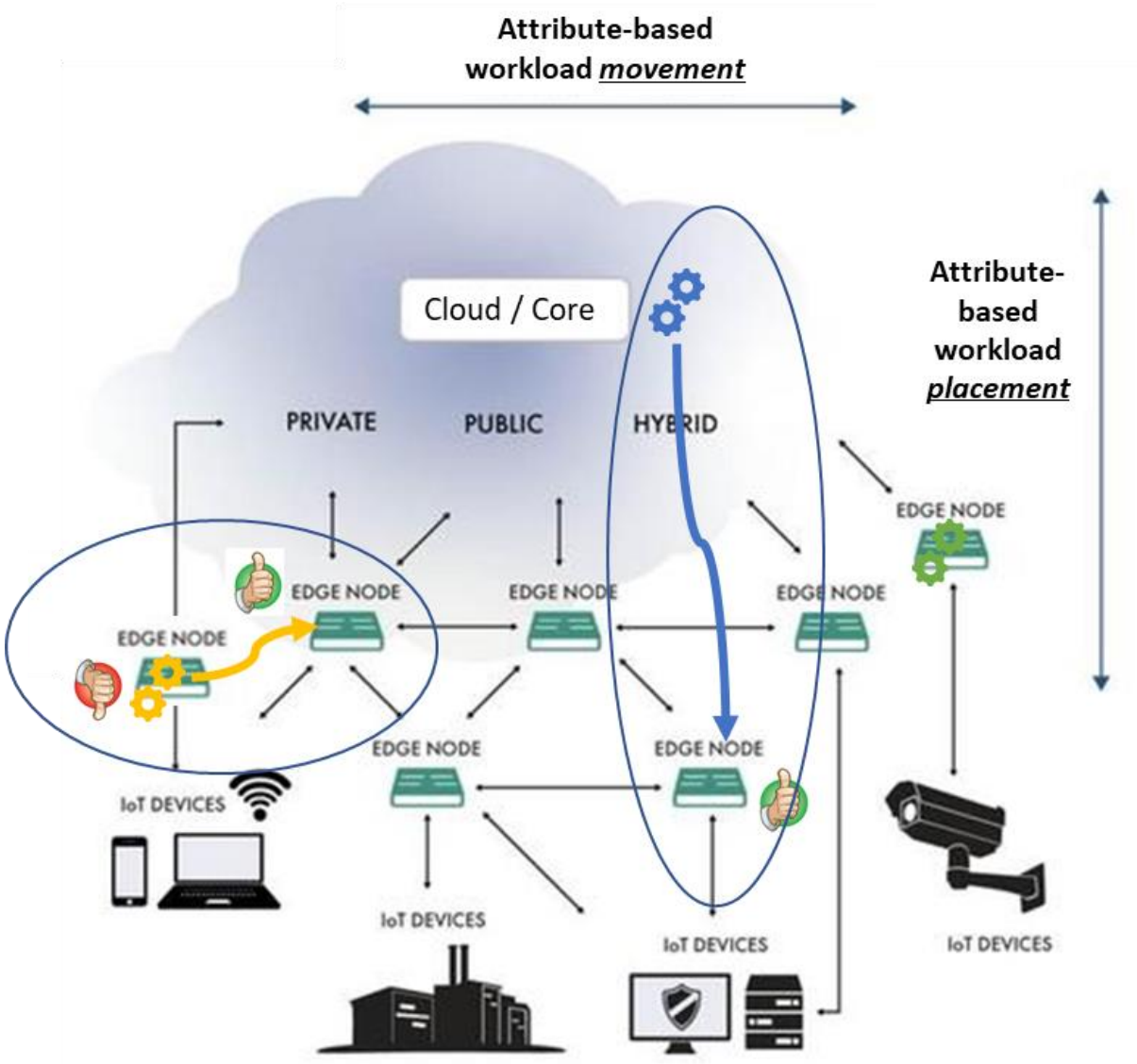

Figure 11 - Attributes/policies will be used to deploy and migrate workloads across the edge and multi-cloud

Multi-cloud integration and enablement

Although a goal of this effort is to define an edge architecture that doesn't depend on the cloud, there will be a need in many cases to integrate with the cloud. Although each of the main components (listed above) include cloud options specific to their area of focus, this component of the architecture focuses specifically on how cloud can enhance the overall edge architecture, and how this architecture will help drive a multi-cloud strategy. It's anticipated this component may be well-enough defined by the Anthos-as-a-Service effort, although integration with this effort will need to be defined here.

For example, a unified interface for serverless/FaaS development will explored, allowing developers to build functions and microservices that can be deployed across cloud providers and edge sites transparently, using a common API. This will help developers deploy to cloud providers of choice to avoid vendor lock-in and migrate across cloud providers more easily (see Figure 12). Additionally, functions and services can be automatically deployed to edge locations as per data locality and workload requirements

(i.e. low latency). Finally, the ability to build a network of services that use functions that span cloud providers will be explored. This work will utilize and extend research done as part of "Multi-cloud microservices-based framework", by Jaumir Valença da Silveira Junior.

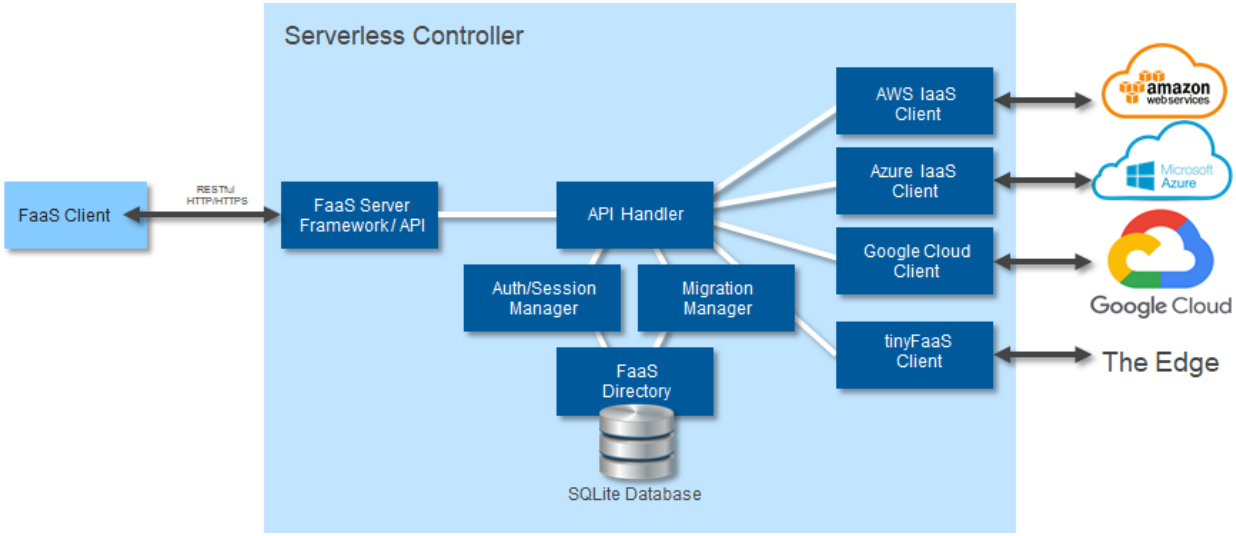

Figure 12 - Common FaaS Framework deployed and used across multi-cloud and edge

If this creates service dependencies (where the FaaS platform deployed within a provider's IaaS offering depends on other PaaS services with that provider) then similar abstractions can apply. For instance, if object storage is needed, Project Amalfi can be used to eliminate specific cloud storage implementations by abstracting around S3. This can further be used to broker the best cloud service for the use case considering performance, capabilities, and cost (see Figure 13).

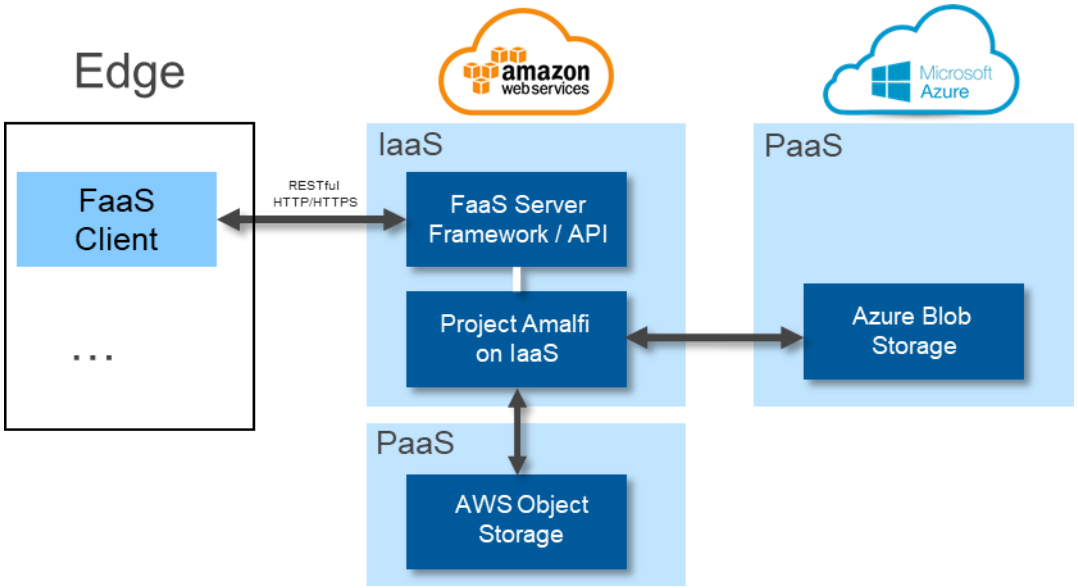

Figure 13 - FaaS abstraction within IaaS accessing abstracted multi-cloud storage

Integrated Blade Runner Vision

The Blade Runner components described in this study work together to form an end-to-end vision of services that enable stand-alone edge deployment, as well as edge-to-multi-cloud enablement. The layered approach to the architecture is shown in Figure 14 below.

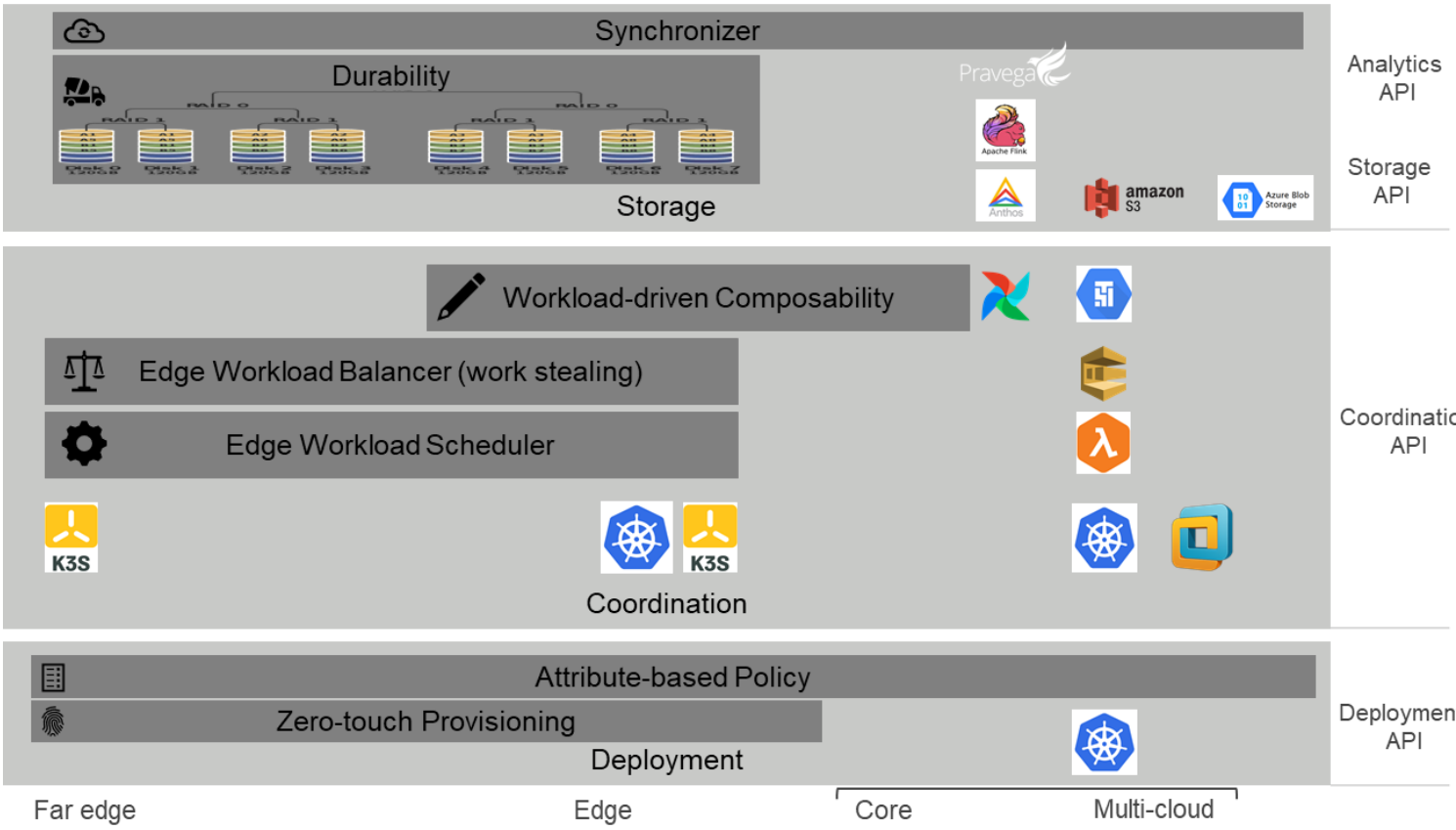

Figure 14 - Blade Runner for the Edge

APIs will be defined to support deployment, coordination of application workloads, the storage of data and real-time analytics performed on it. Various cloud technologies will be supported and, in some cases, deployed at the edge:

- Coordination
  - Edge: Kubernetes and K3S at the edge for orchestration
  - Edge: Apache Airflow at the edge for workload and data pipeline definition
  - Multi-cloud: Kubernetes, VM/Containers, FaaS, Cloud Composer for orchestration, functions, data pipelines
- Storage
  - Edge: Anthos, Pravega.io, Flink
  - Multi-cloud: AWS S3
- Data Analytics:
  - Edge: Anthos, Pravega.io, Flink

Impact to Dell

As Dell Edge and Telco business units are focused on delivering solutions for today's needs, the work defined in this study is meant to look out further on the horizon to meet the needs of a growing edge application ecosystem. The goal of this research is to help Dell be ready for future customer needs around Edge and deliver novel features and greater value to edge users. Specifically, the need for additional, newer storage at the edge is anticipated to grow much more quickly than otherwise thought.

Next Steps

The following studies and projects are currently planned, subject to change as insights and outcomes are gathered in the course of this work. These are listed below in the order they are anticipated to execute:

- **PoC:** Implement Edge workload stealing v2 with K8s integration
- **PoC**: Implement federated computational storage with 100% durability
- **PoC**: Composable edge using Edge node clusters
- **PoC**: Deploy KNative (or other FaaS platform) across multi-cloud and Edge
- **PoC**: Advanced Edge cluster management by extending K8s
- **Study**: Summarize the multi-cloud integration architecture at the Edge

References

- *Edge Processing to Grow 75% by 2025*, https://datacentremagazine.com/networking/edge-processing-will-grow-75-2025
- *CIO Guide to Distributed Edge*, https://www.gartner.com/smarterwithgartner/the-cios-guide-to-distributed-cloud/
- *Future of Cloud and Edge Infrastructure*, https://www.gartner.com/smarterwithgartner/gartner-predicts-the-future-of-cloud-and-edge-infrastructure/
- Dell Pravega.io, https://pravega.io/
- *Top Strategic Technology Trends of 2021*, https://www.gartner.com/smarterwithgartner/gartner-top-strategic-technology-trends-for-2021/