



# StarlingX Project Overview

Learn, Try, Get Involved!

---

# Let Me Introduce StarlingX

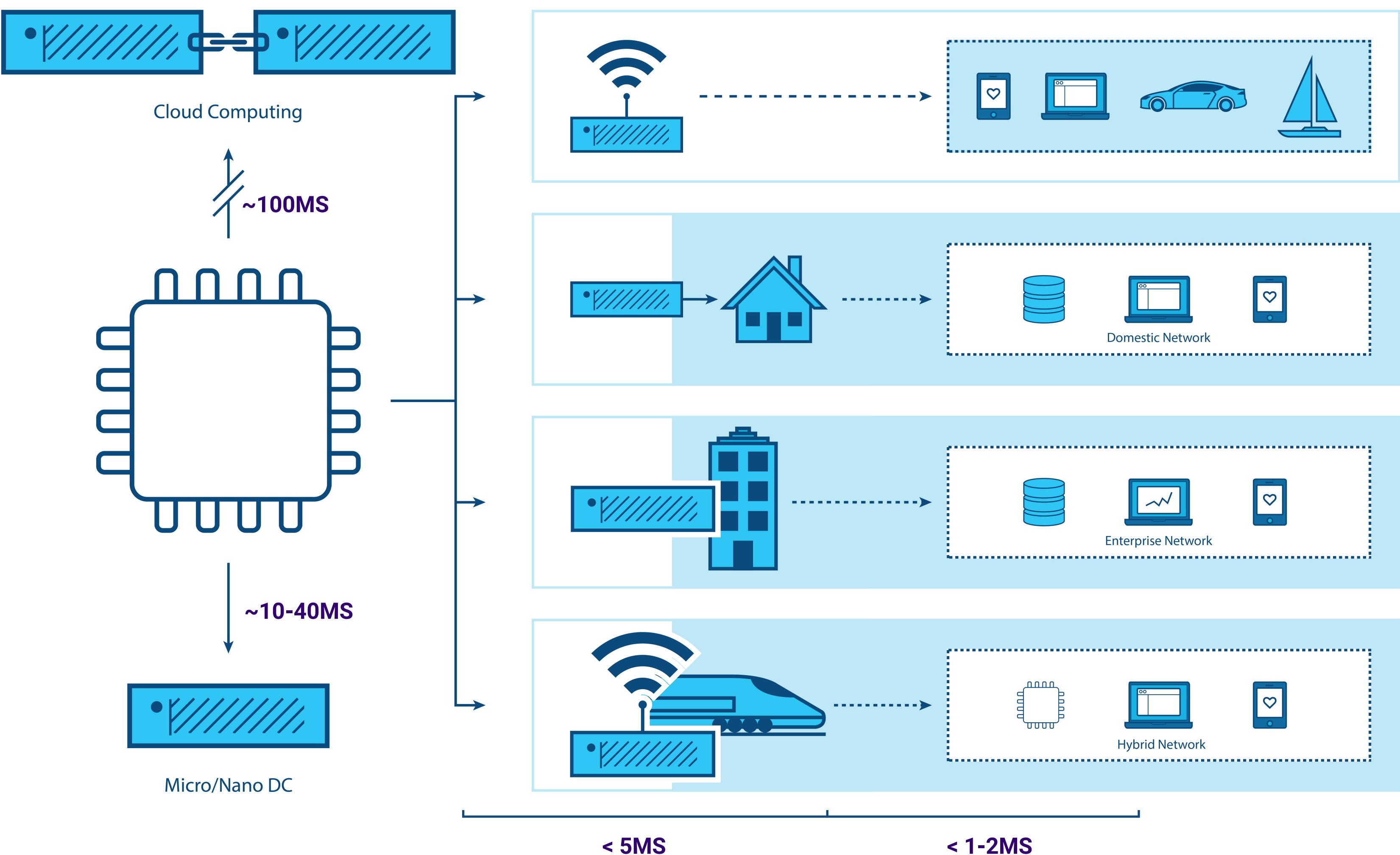
- Top-level OpenStack Foundation project
- Software stack providing high performance, low latency, and high availability for Edge Cloud applications
- Frequent releases
  - <https://git.starlingx.io/>
  - <http://mirror.starlingx.cengn.ca/mirror/starlingx/release/>
- Growing community
  - Inviting users, operators and developers to try out the software and participate in the community

# Project Overview

# What Is Driving Edge Computing?

- A. Latency
- B. Bandwidth
- C. Security
- D. Connectivity

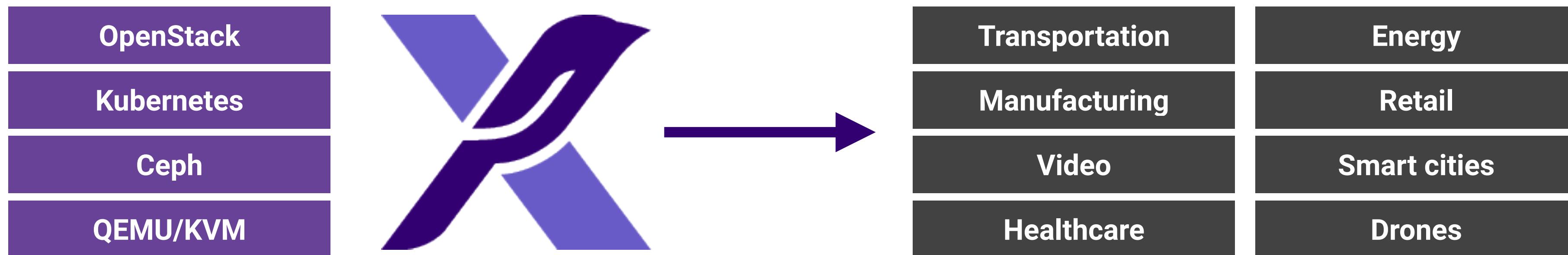
**“WHERE” MATTERS**



# Intent of the StarlingX Project

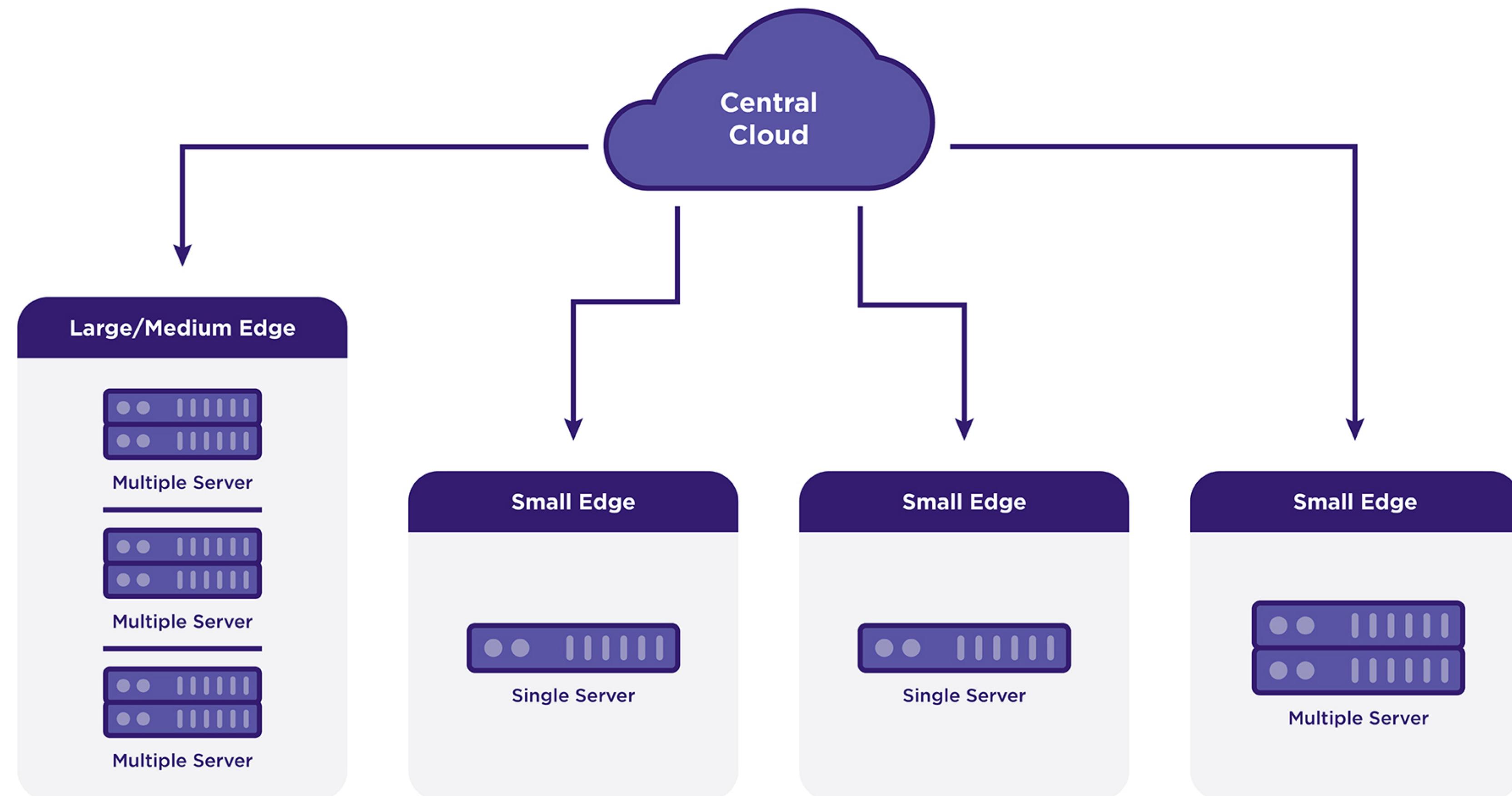
## Re-Configure Proven Cloud Technologies for Edge Compute

- Orchestrate system-wide
  - Deploy and manage Edge clouds, share configurations
- Simplify deployment to geographically dispersed, remote Edge regions



# StarlingX Edge Deployments

- Geographically distributed multi-region deployment,
- Central Datacenter providing Orchestration and Synchronization Services,
- Geographically distributed Edge Sites of various sizes



# StarlingX Edge Use Cases

- On-premises clouds for
  - Industrial Automation
  - Hospitals
  - Transportation and Rail
- Central Office clouds for
  - vRAN
  - Local data centers
  - uCPE
  - Multi-Access Edge Computing (MEC)
- IIoT and VCS (Vehicle Cloud Services) clouds for
  - Smart cities
  - Autonomous vehicles with V2X (Vehicle-to-Any) communication models



# StarlingX Technology

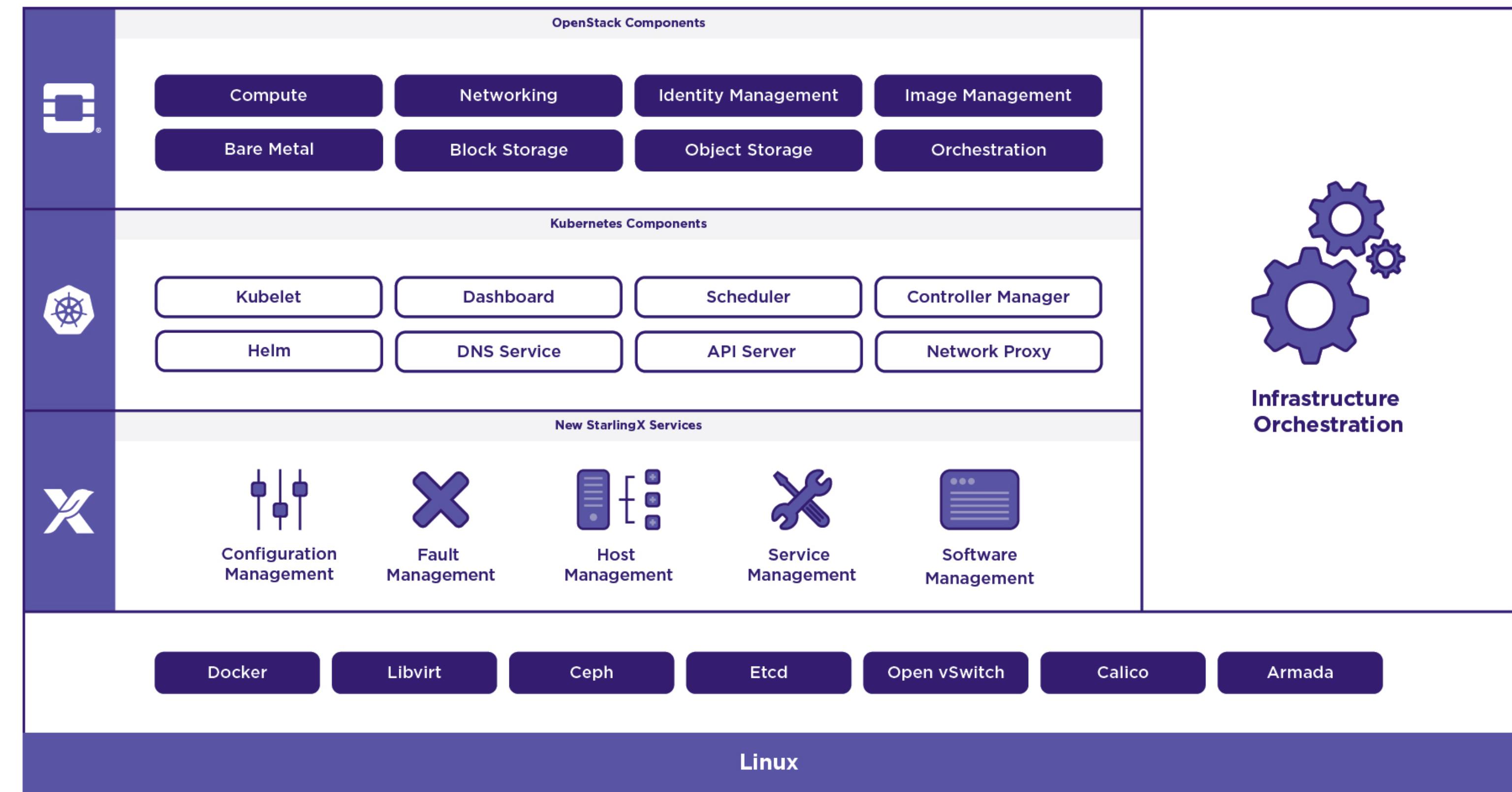
# StarlingX - Edge Virtualization Platform

StarlingX provides a **deployment-ready, scalable, highly reliable** Edge infrastructure software platform

Services from the StarlingX virtualization platform focus on

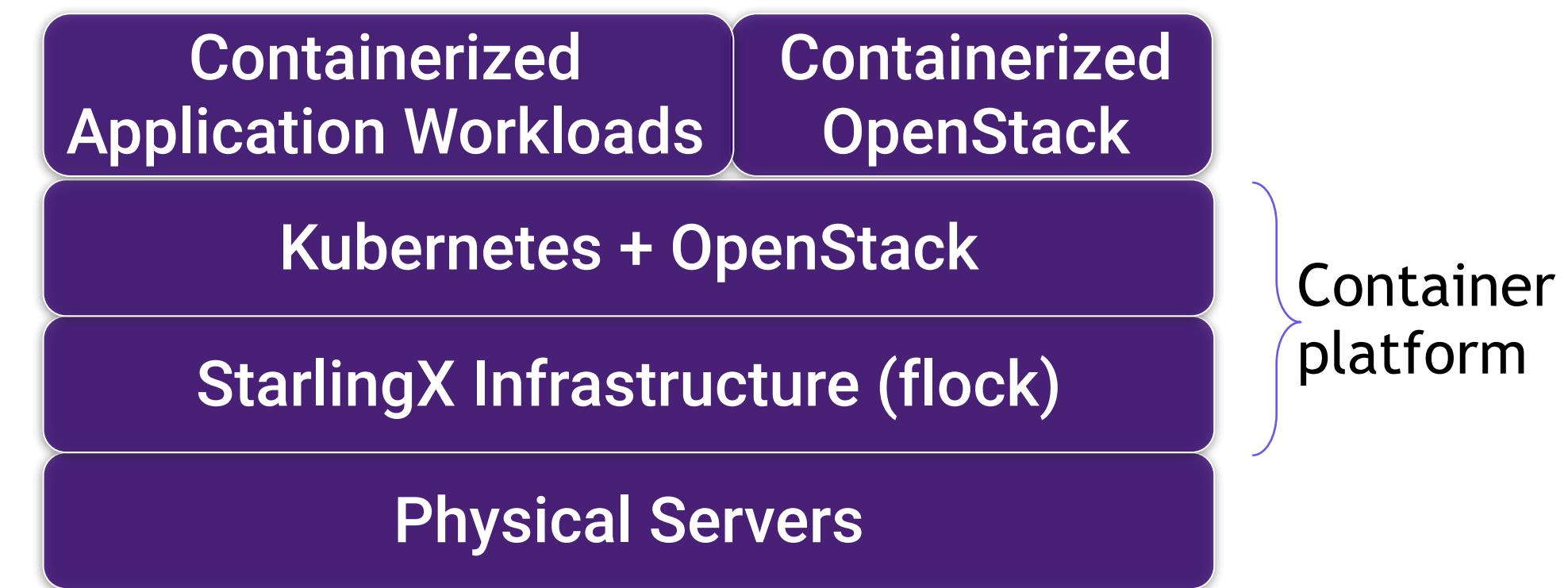
- **Easy deployment**
- **Low touch manageability**
- **Rapid response to events**
- **Fast recovery**

A complete edge orchestration platform for bare metal, VM and container workloads

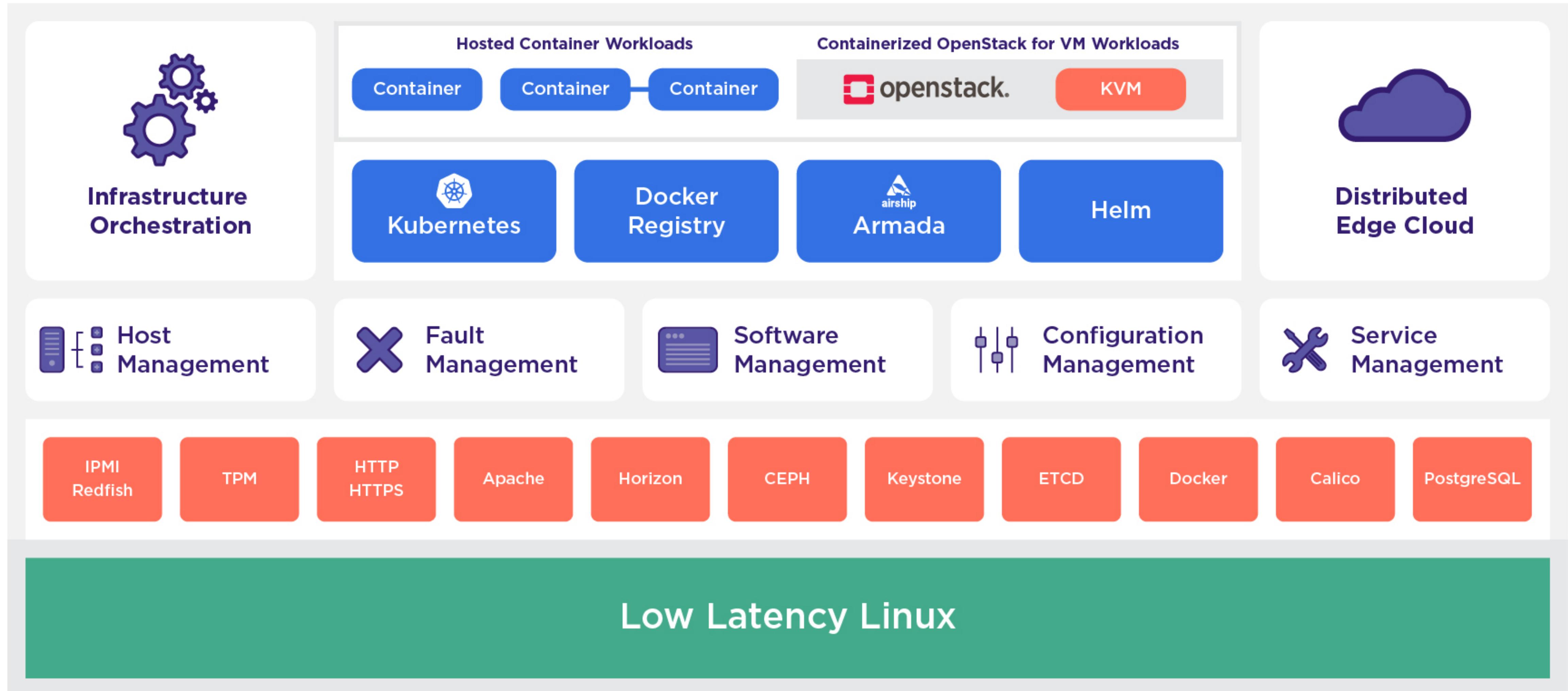


# StarlingX Evolution

- A hardened cloud-native platform integrating OpenStack and Kubernetes on dedicated physical servers
- Containerized OpenStack services based on the latest release
- Closely aligned with the current OpenStack code base
  - The StarlingX and OpenStack communities are working together on Edge related enhancements
- Kubernetes-based edge sites for containerized workloads



# Distributed Edge Cloud Native Platform



# StarlingX – Deployment Models

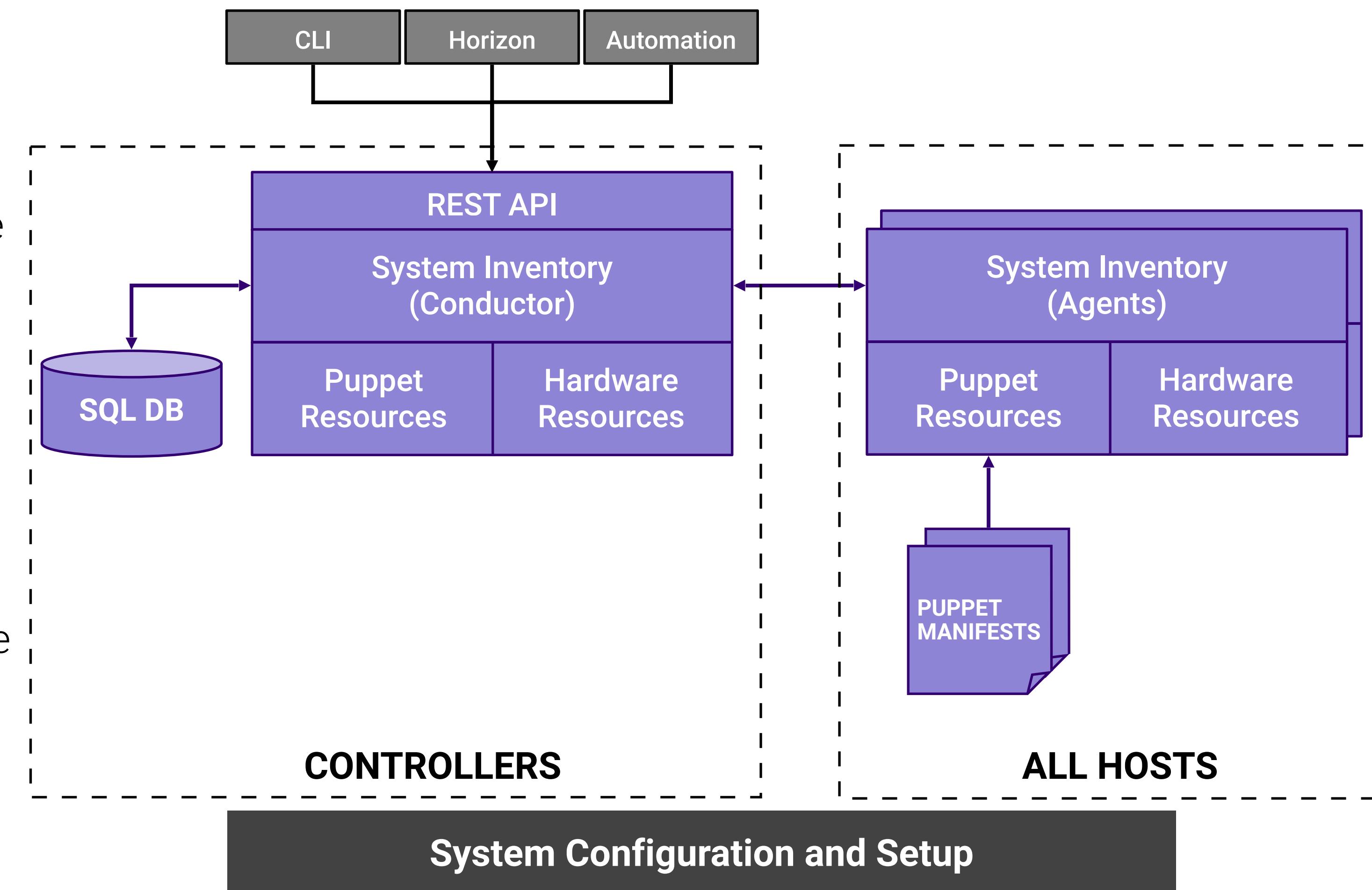
- Scalable deployment models from 1-100 servers addressing the wide range of edge use cases
- Focus on minimizing the infrastructure footprint
  - One & two server solution overhead
  - 2 cores/server
  - Frame level solution
  - 2 server master implementation
  - 1 core overhead/worker node
- Storage
  - Integrated CEPH for one & two node solutions
  - Co-located CEPH on master nodes for small frame level deployments
  - Dedicated CEPH storage nodes for larger configurations

# Infrastructure Management

## “The Flock”

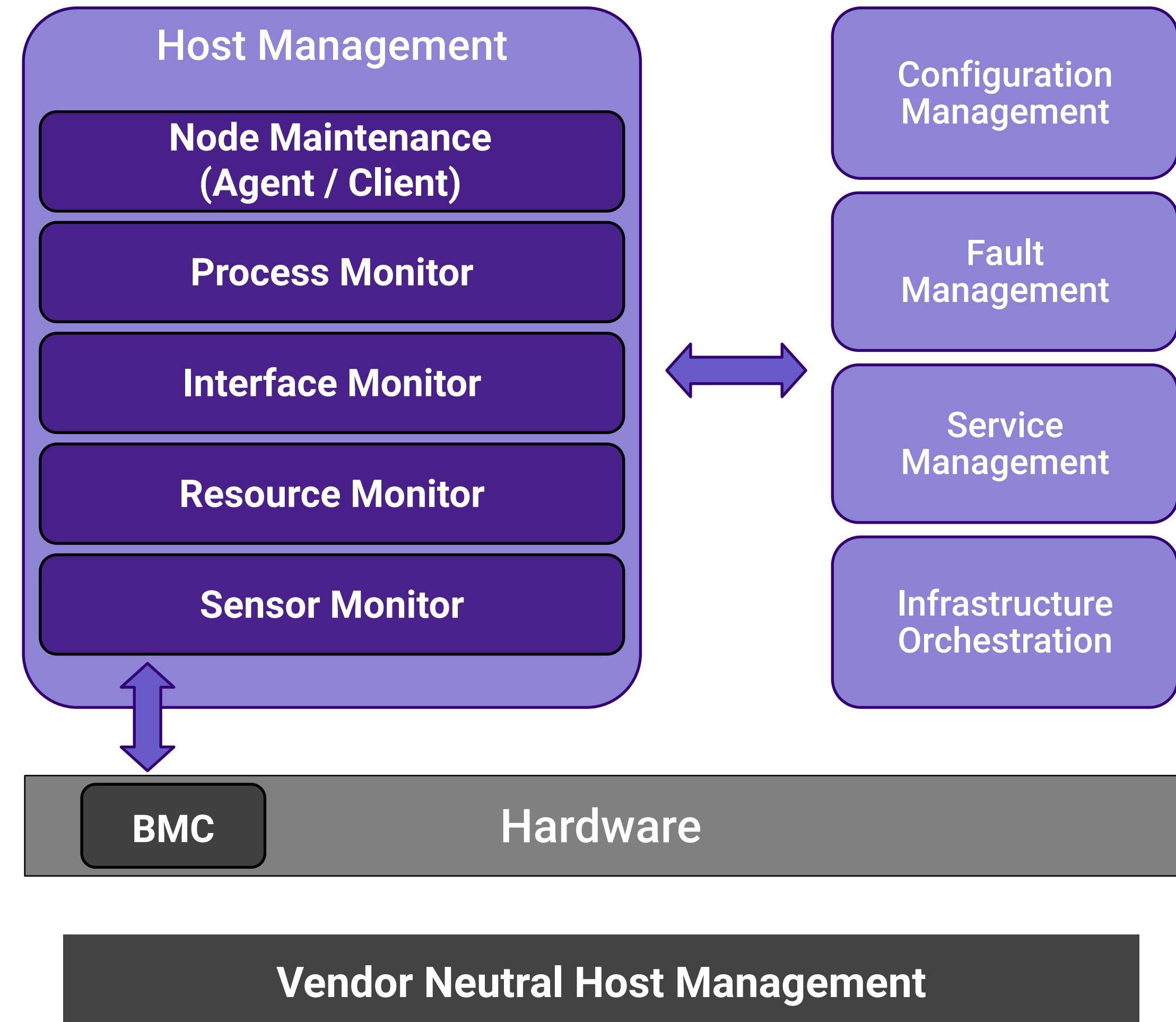
# Configuration Management

- Manages installation
  - Auto-discover new nodes
  - Manage installation parameters (i.e. console, root disks)
  - Bulk provisioning of nodes through XML file
- Nodal Configuration
  - Node role, role profiles
  - Core, memory (including huge page) assignments
  - Network Interfaces and storage assignments
- Inventory Discovery
  - CPU/cores, SMT, processors, memory, huge pages
  - Storage, ports
  - GPUs, storage, Crypto/compression H/W



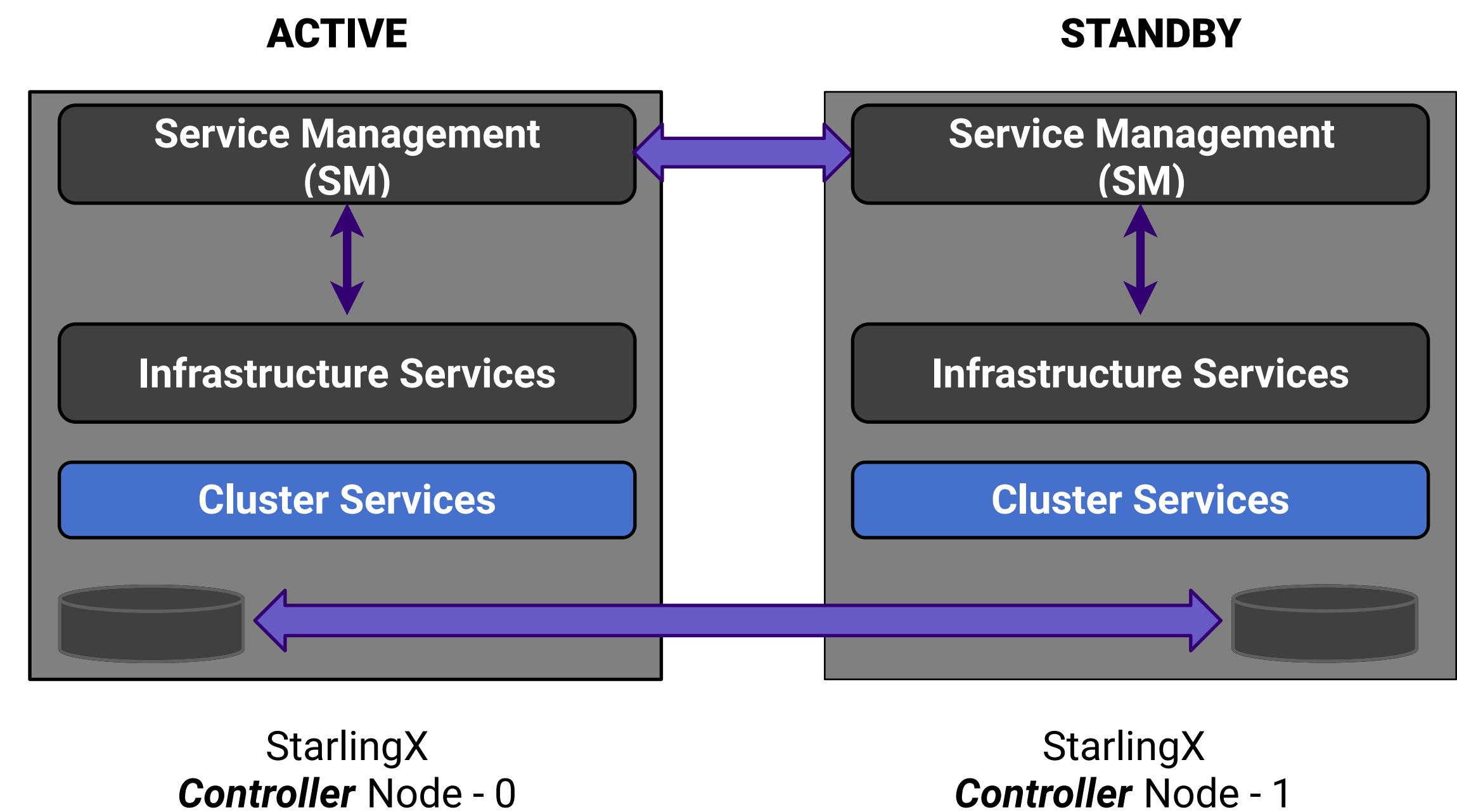
# Host Management

- Full life-cycle management of the host
- Detects and automatically handles host failures and initiates recovery
- Monitoring and fault reporting for
  - Cluster connectivity, critical process failures
  - Resource utilization thresholds, interface states
  - H/W fault / sensors, host watchdog
  - Activity progress reporting
- Interfaces with board management (BMC)
  - For out of band reset
  - Power-on/off
  - H/W sensor monitoring



# Service Management

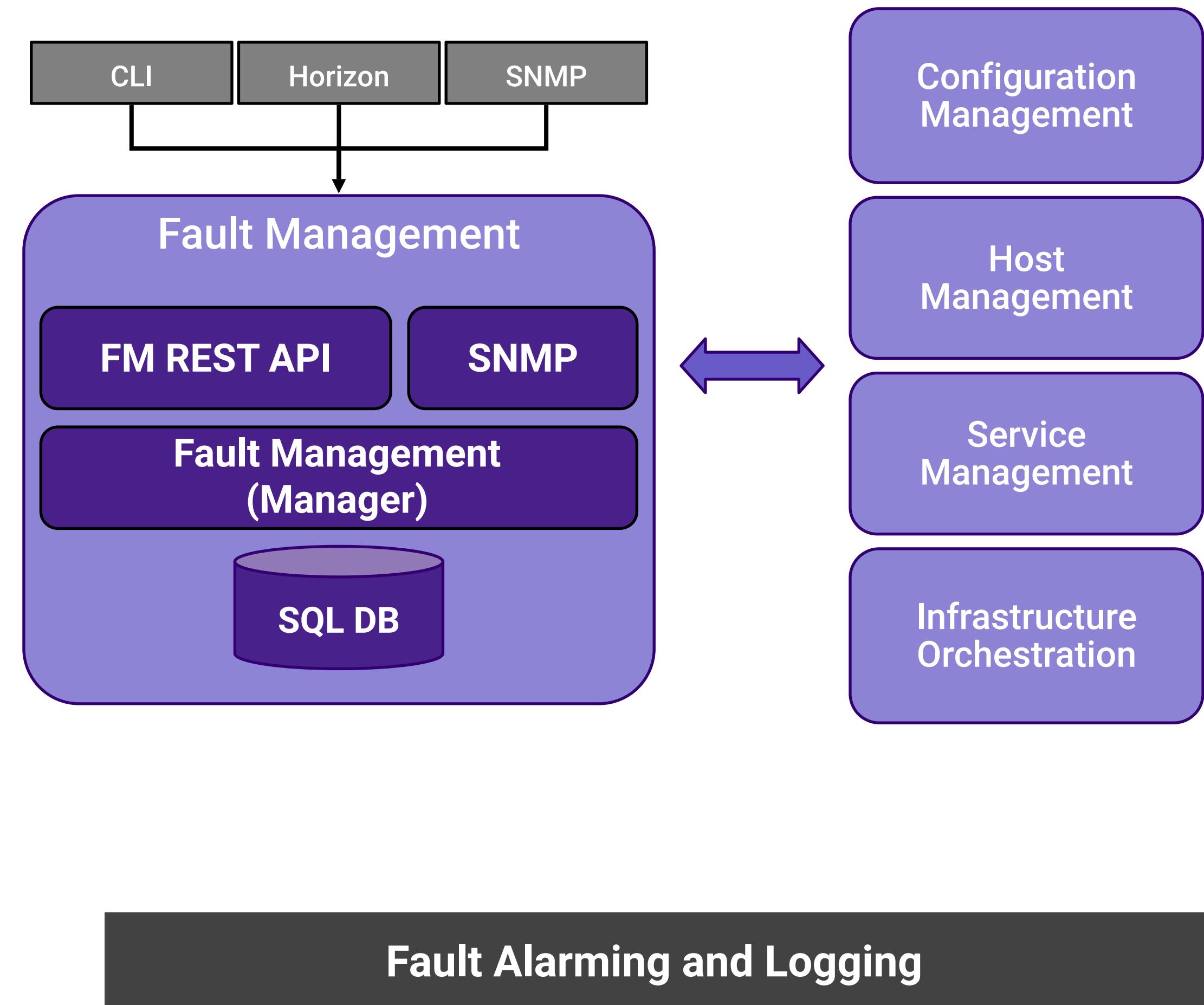
- High availability manager
  - Redundancy model can be N+M or N across multiple nodes
  - Currently 1+1 HA Controller Cluster
- Uses multiple messaging paths to avoid split-brain communication failures
  - Up to 3 independent communication paths
  - LAG can also be configured for multi-link protection of each path
  - Messages are authenticated using HMAC SHA-512 if configured / enabled on an interface-by-interface basis
- Active or passive monitoring of services
- Allows for specifying the impact of a service failure



**High Availability for Critical Infrastructure**

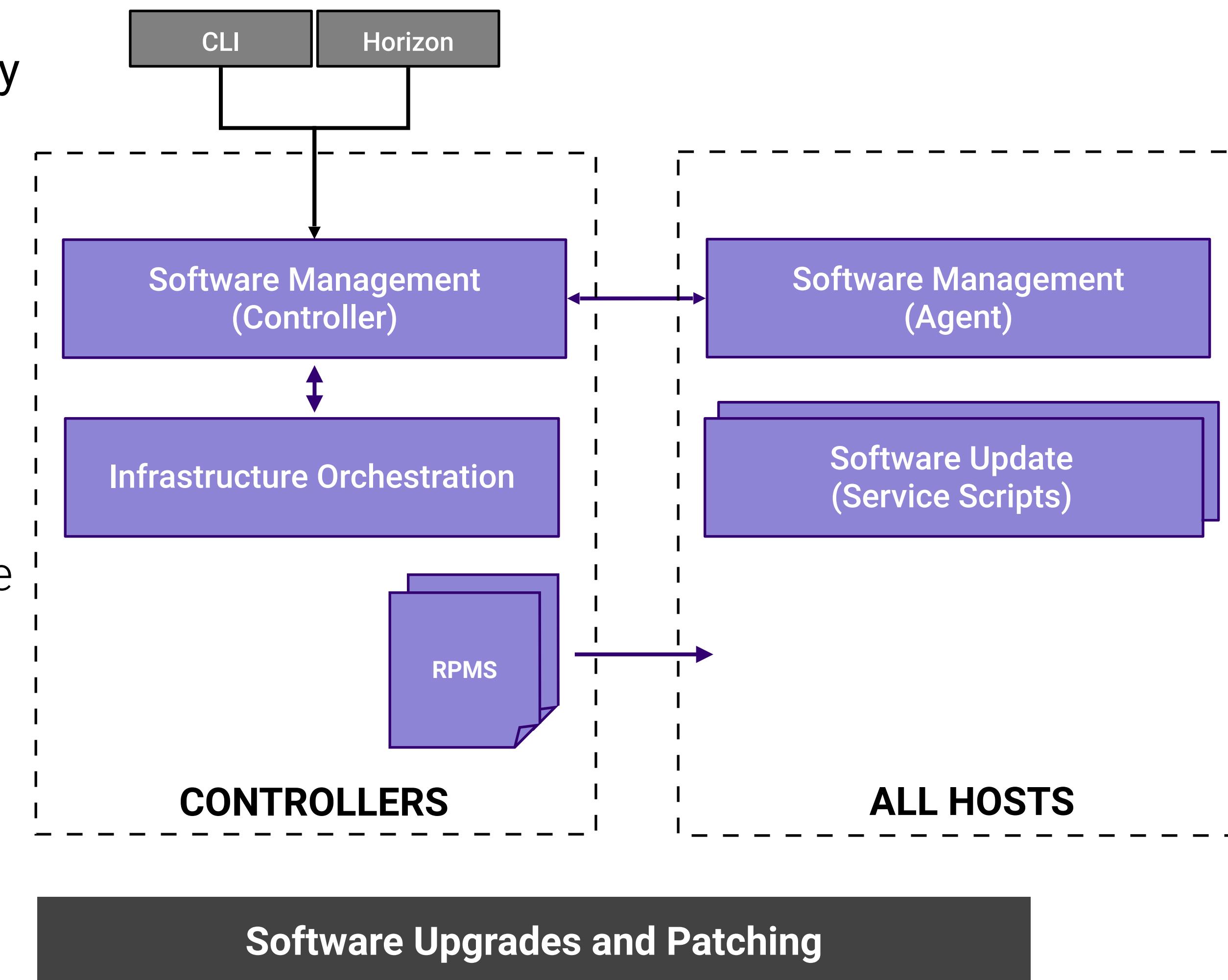
# Fault Management

- Framework for infrastructure services via API
  - Set, clear and query customer alarms
  - Generate customer logs for significant events
- Maintains an Active Alarm List
- Provides REST API to query alarms and events
- Support for alarm suppression
- Operator alarms
  - On platform nodes and resources
  - On hosted virtual resources
- Operator logs - Event List
  - Logging of set/clear of alarms
  - Related to platform nodes and resources
  - Related to hosted virtual resources



# Software Management - in progress

- Automated deploy of software updates for security and/or new functionality
- Integrated end-to-end rolling upgrade solution
  - Automated, low number of steps
  - No additional hardware required for upgrade
  - Rolling upgrade across nodes
- In-service and reboot required patches supported
  - Reboot required for kernel replacement etc.
  - VM live migration is used for patches that require reboot
- Manages upgrades of all software
  - Host OS changes
  - New / upgraded StarlingX service software
  - New / upgraded OpenStack software

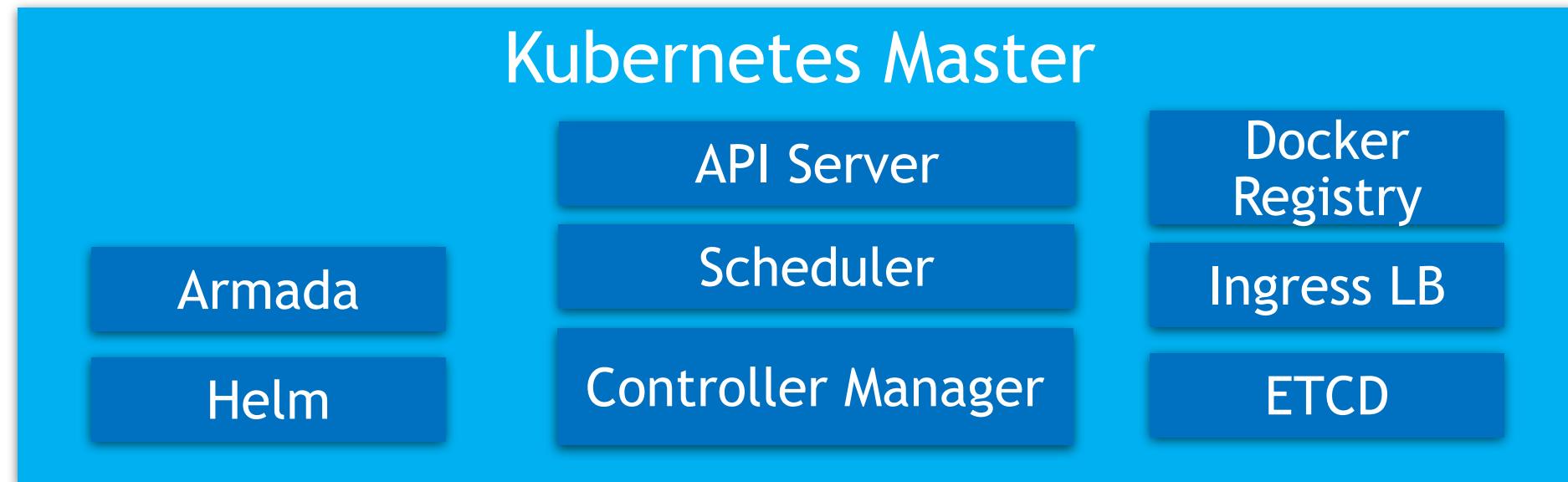




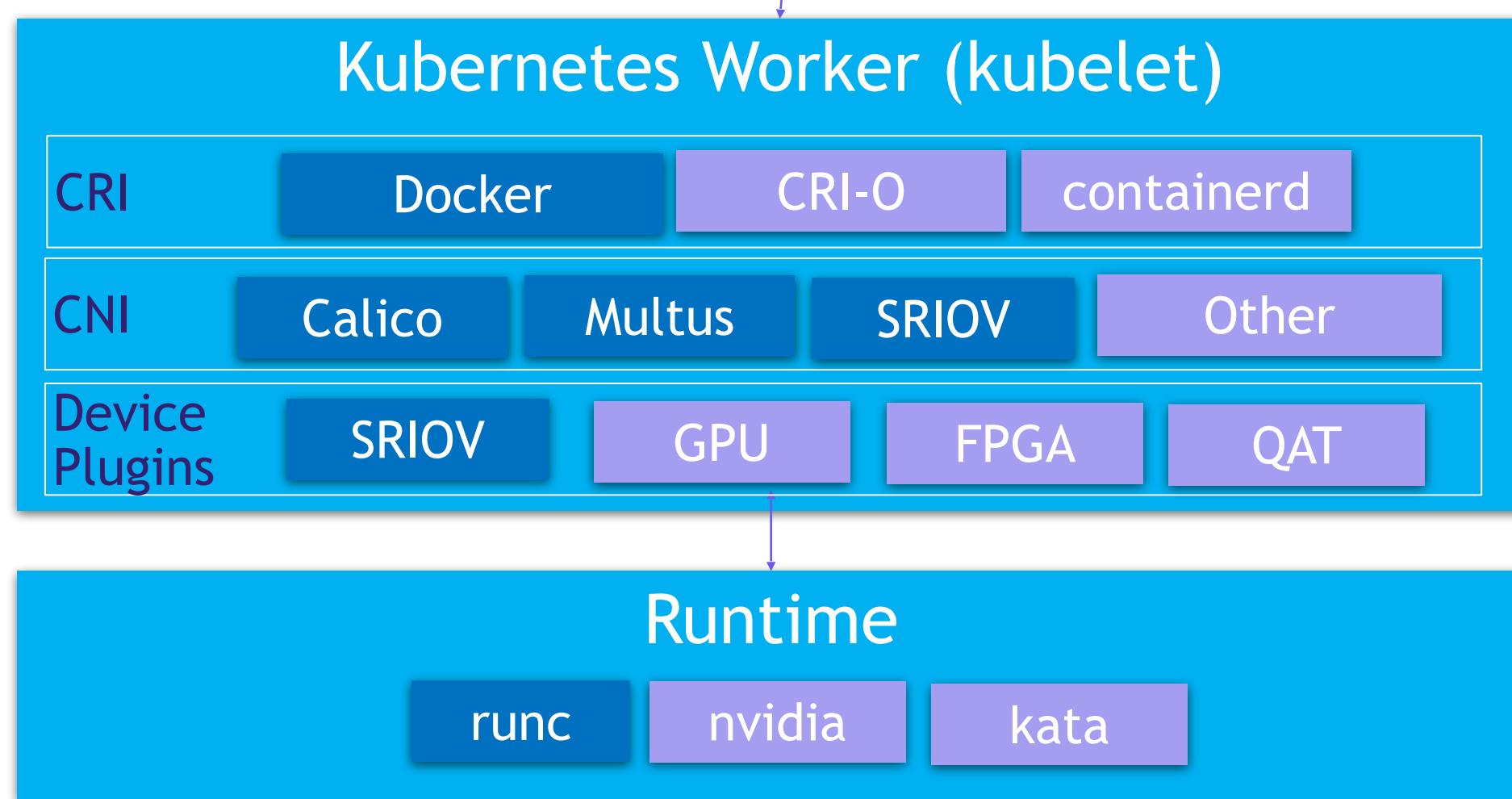
# Container Platform

# Kubernetes Cluster Software Components

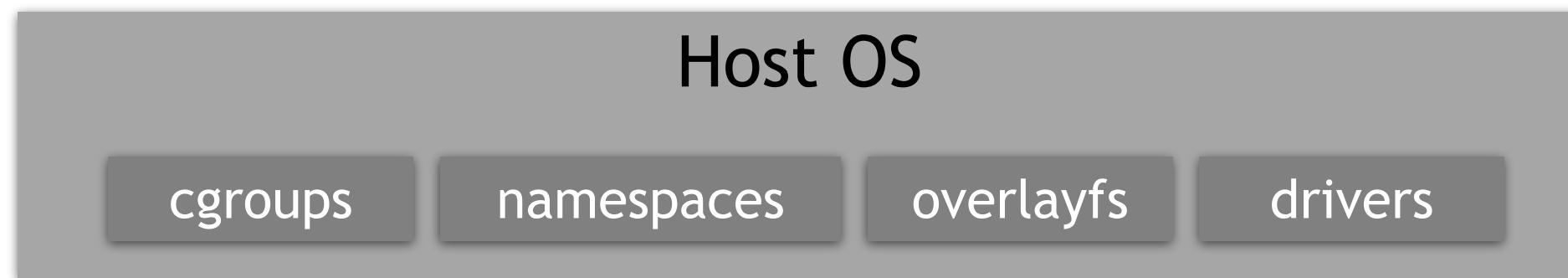
## Control Plane



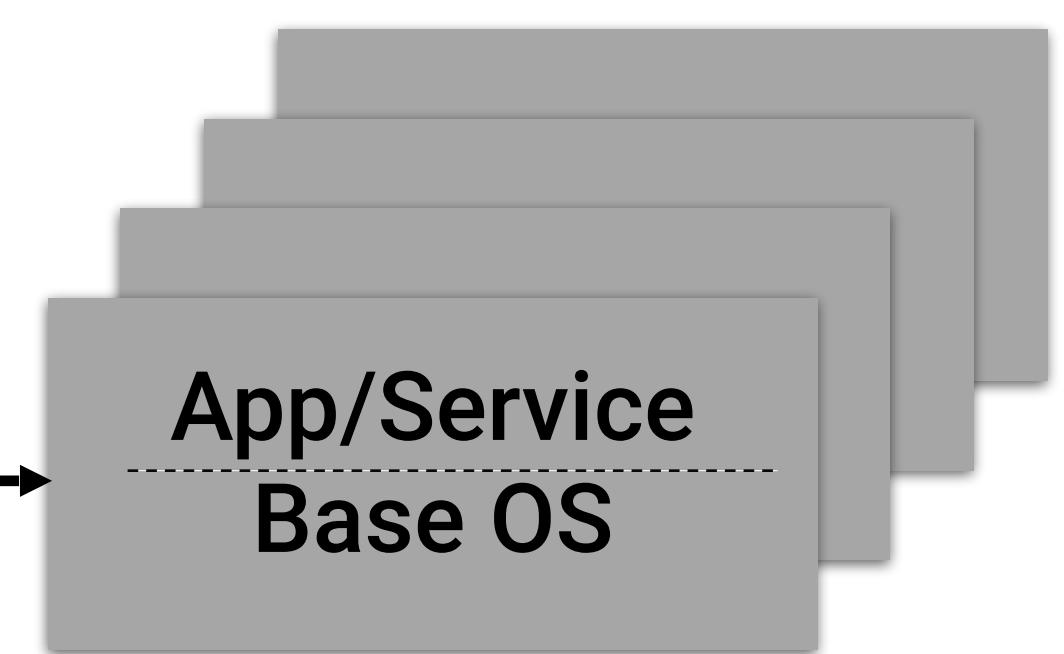
## Worker Nodes



## Host Abstraction



## Application Containers



**App/Service  
Base OS**

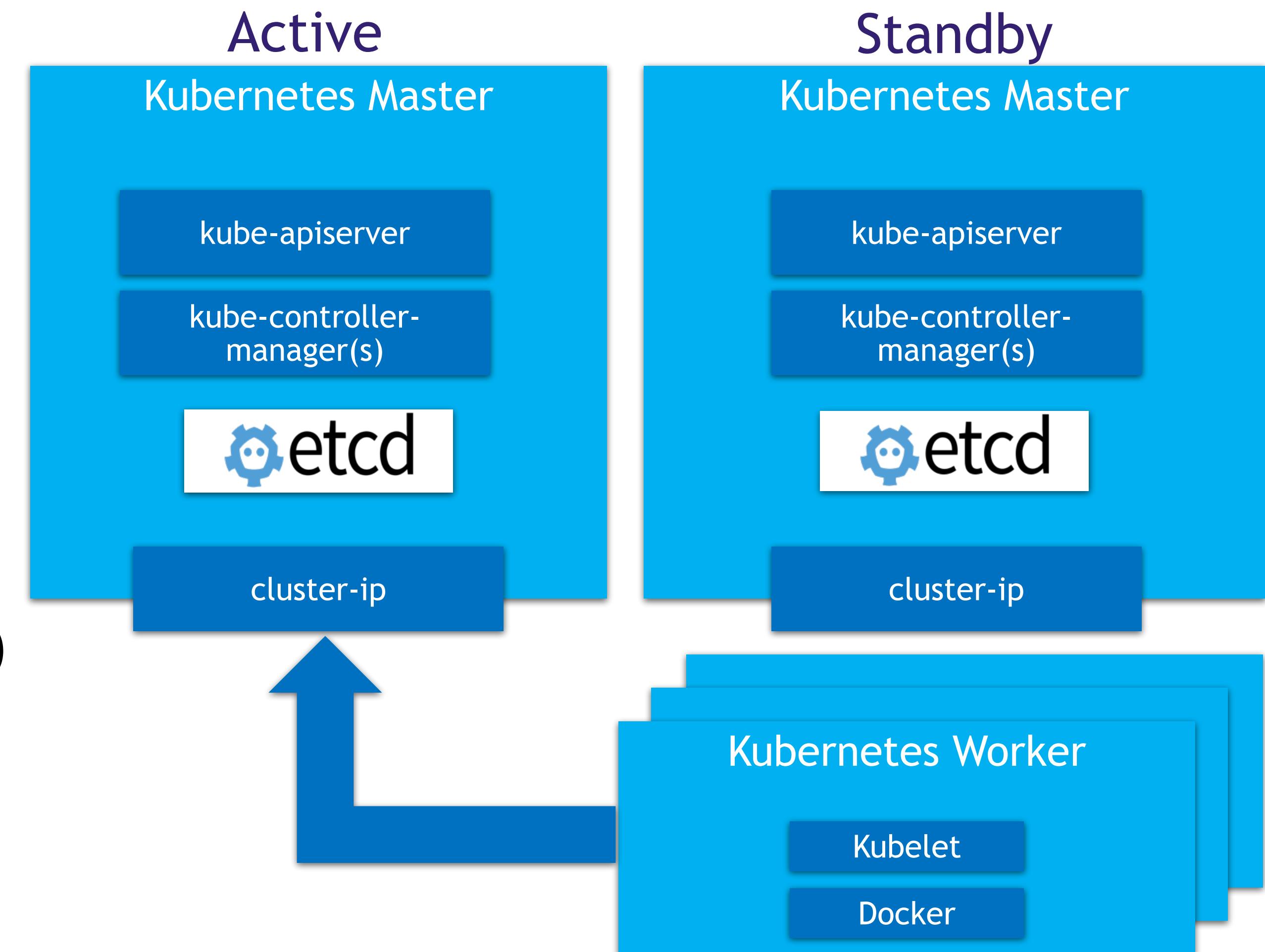
- Full infrastructure and cluster orchestration
- Security and Policy

- Flexible ecosystem of base OS
- Evolution to other container runtimes
- Kernel and Accelerated networking

Current  
Future

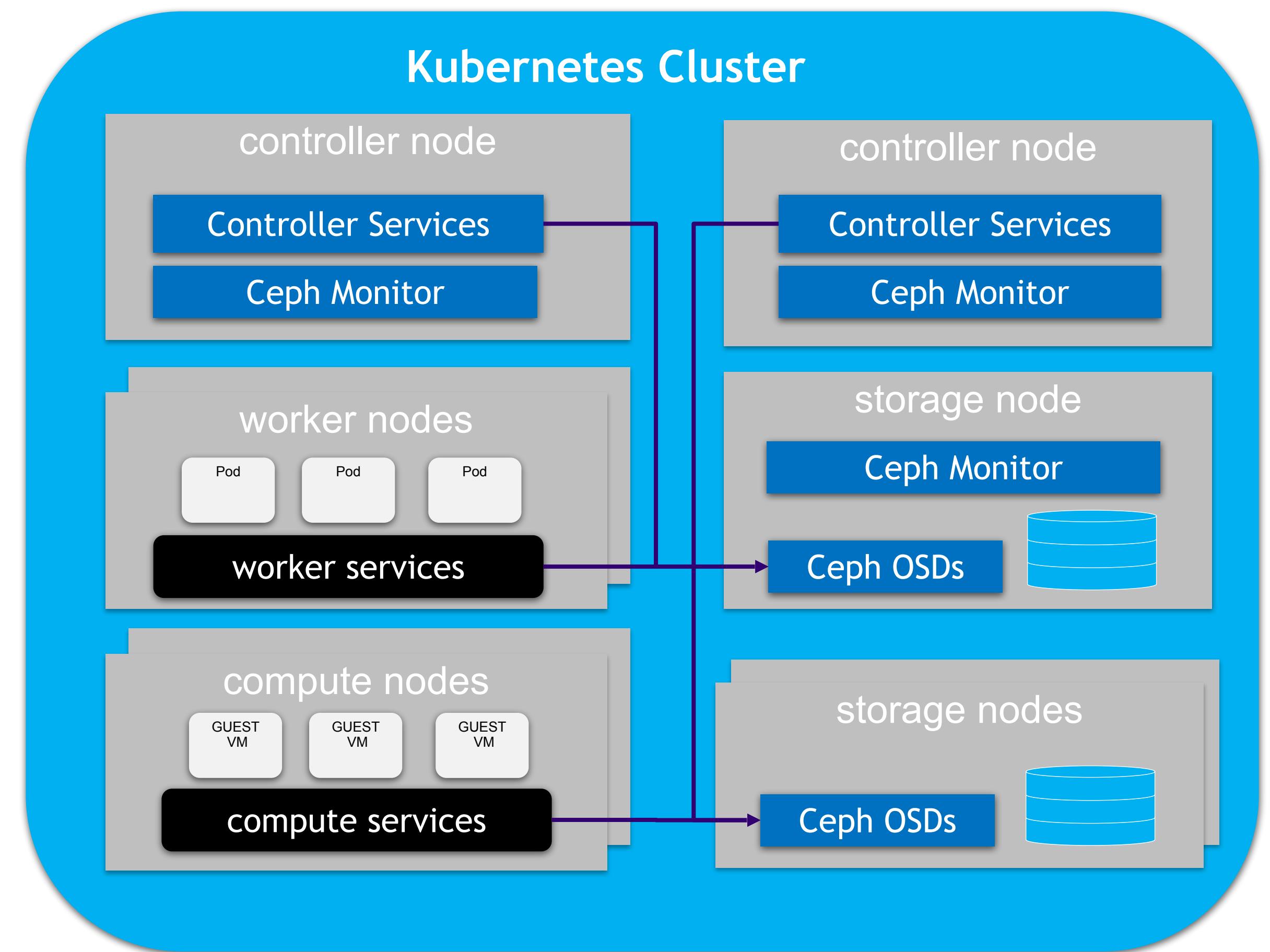
# Kubernetes Deployment Architecture

- Kubernetes deployed in a highly available configuration
  - Deployed in a 1:1 service model
  - All-in-One Simplex/Duplex deployments supported using same service management
- Requests directed to active instances via cluster floating IP address
- DRBD backed file system for redundant persistent storage
- Service availability and activity managed by Service Management (SM)
  - Handles HA sparing of individual services
  - Actively monitors host, service and network availability
  - Mitigates split-brain scenarios



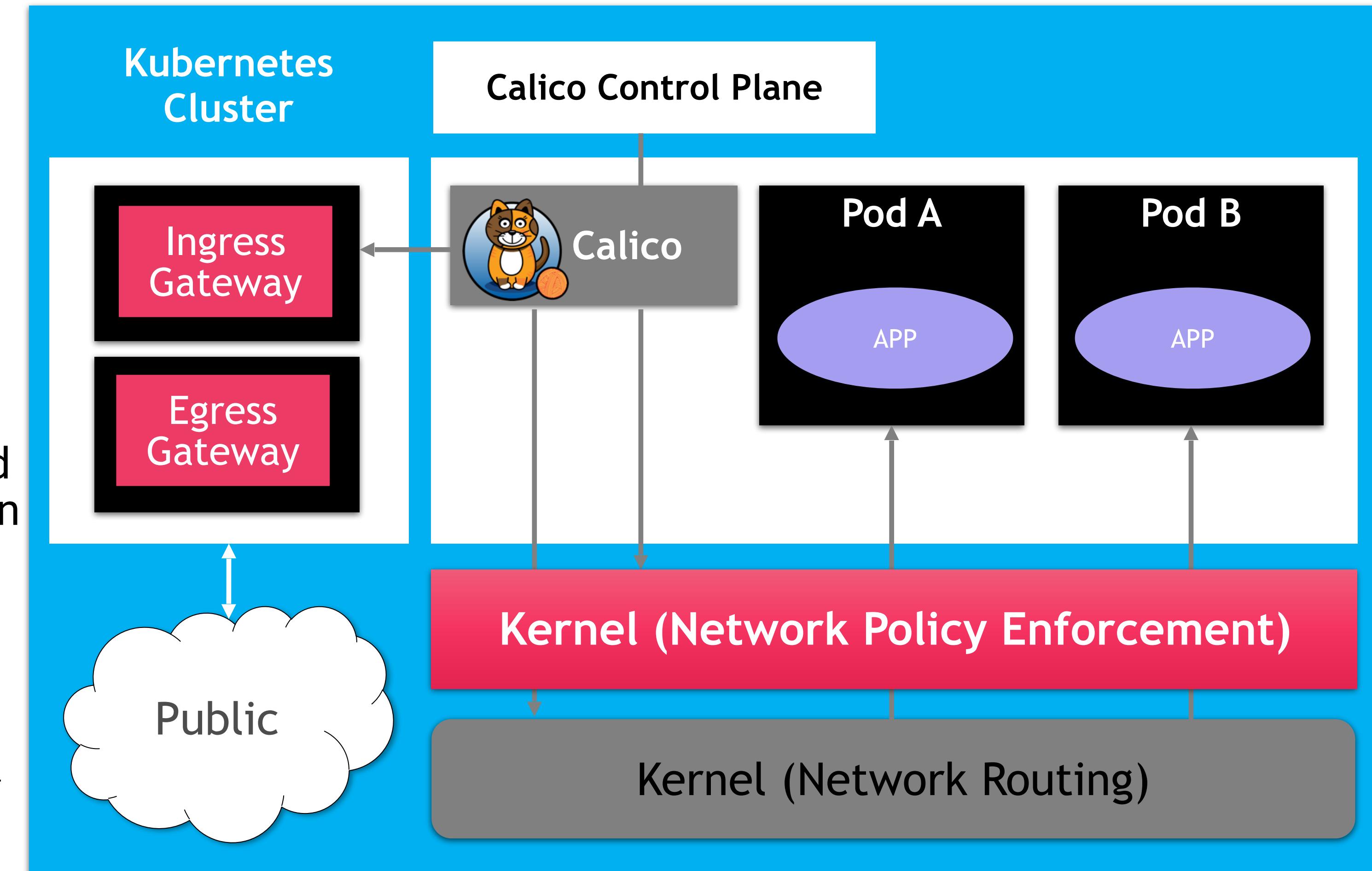
# Cluster Persistent Storage

- Ceph uniquely delivers object, block, and file storage in one unified system
- Highly scalable and highly available deployment with distributed Ceph monitors and Object Storage Devices (OSD) for data replication
  - Automatic cluster storage deployment and replication
  - Unified storage solution for all deployments: AIO-SX/DX, Standard, Multi-cloud
  - Fully managed Ceph Cluster Map (hyper-scale)
- Kubernetes persistent storage provided by Ceph's RADOS Block Devices (RBD) provisioner
  - Persistent Volumes (PVs) and Claims (PVCs)
  - Default Storage Class
- OpenStack backend storage solution for services: Glance, Cinder, Swift, Nova



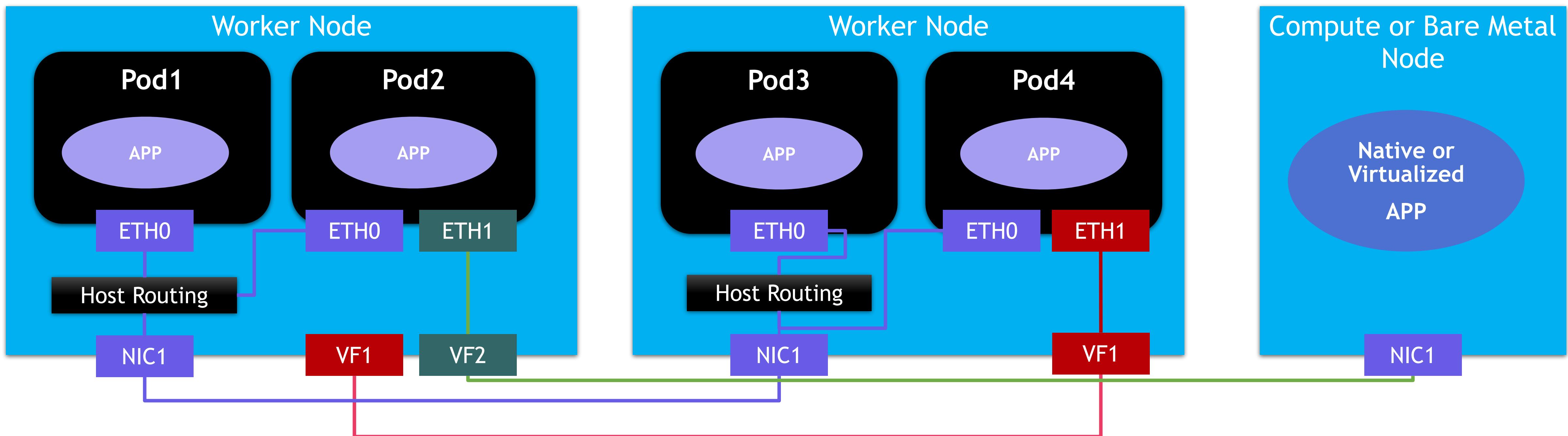
# Kubernetes Cluster Networking

- Calico provides a pure L3 fabric solution for interconnecting containers
- Calico leverages the Linux kernel for routing and policy enforcement
- Calico leverages Border Gateway Protocol (BGP) for control plane
- Calico leverages Open Standards and is a full Open Source network solution
- Calico is highly scalable, and is operator and policy friendly:
  - No overlay, no tunnelling, no VRF tables (no overhead) – pure routing
  - Access Controls enforced through L3/L4 security policies



# Kubernetes Accelerated Networking

- Multus, SR-IOV, DPDK
- Kubernetes managed accelerated network devices (via Device and CNI plugins)
- Containers bind the driver to the Virtual Function (VF) or DPDK devices directly
- No host routing or switching is involved for SRIOV and provides the best direct IO

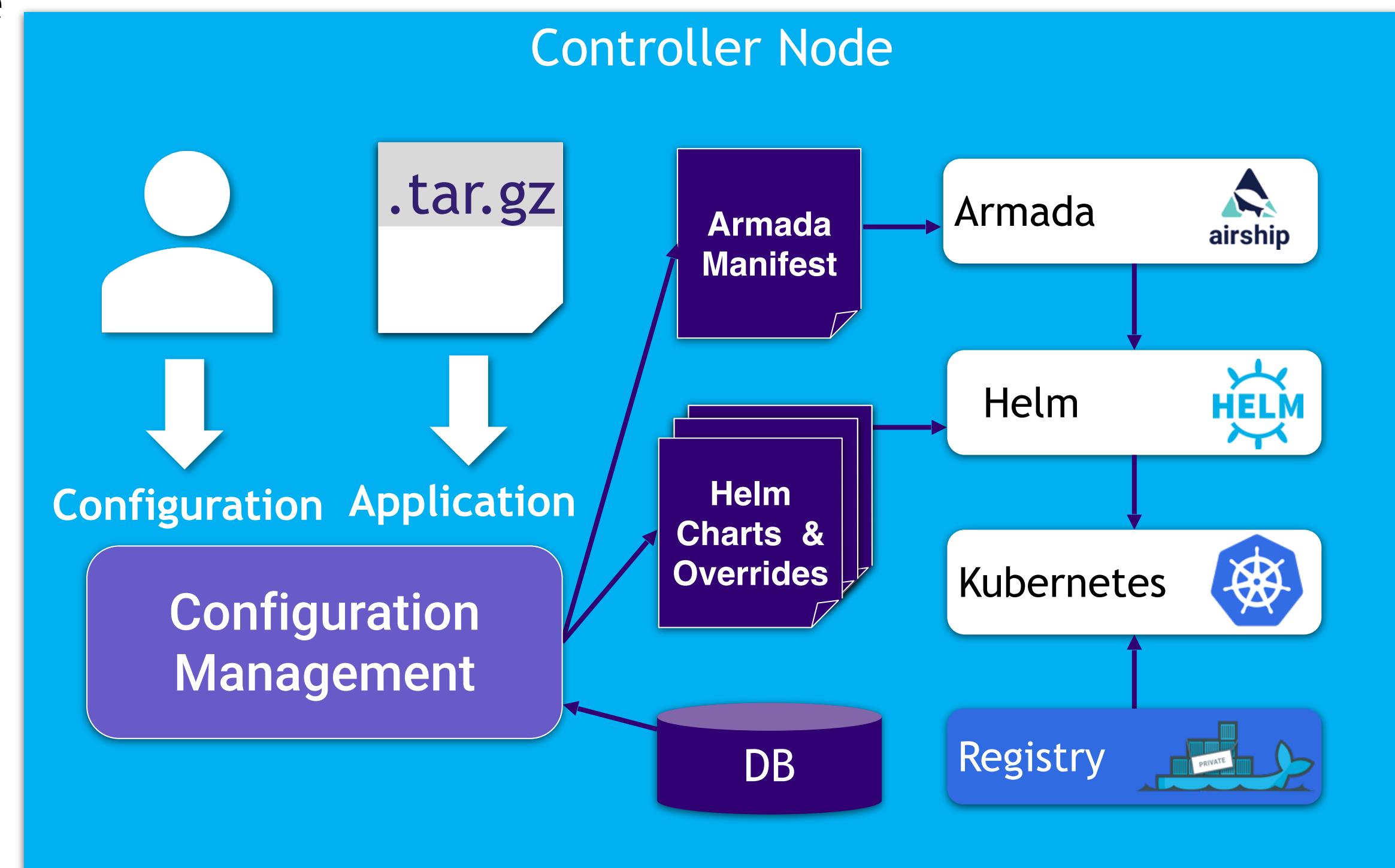


# Additional Capabilities

- Local replicated docker image registry
- Integration with openstack keystone
  - Local docker image registry authentication
  - Authentication/authorization of k8s API
- Huge Page support
  - Enables pods to allocate and consume huge pages from pre-allocated host pool
- Kubernetes CPU manager static policy
  - Enables applications to reserve exclusive CPUs in their pod spec

# Application Management

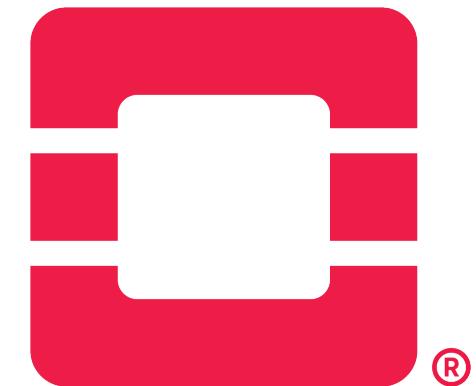
- **Helm**
  - Helm Charts help you define, install, and upgrade even the most complex Kubernetes application.
  - Template based approach to Kubernetes configuration
  - System and User overrides combined to provide final deployment configuration
- **Armada**
  - Manages the dependencies between multiple Helm Charts and expression of those relationship
  - Static and default configuration attributes
- **Application**
  - Curated software bundle of Armada manifest and Helm Charts
  - Application lifecycle coordinated by Configuration Management
  - User uploads, applies / removes application with single command operations



# OpenStack

# OpenStack Deployment

- OpenStack is deployed as a containerized Kubernetes application
  - OpenStack control plane running in pods
  - OpenStack virtual machines running on host
  - Leverages Kubernetes' strengths to manage, scale and update the OpenStack services
- Deployed using Helm (using OpenStack-Helm charts) and Armada (orchestrator for deploying Helm charts from OpenStack Airship).
- StarlingX provides application APIs to install and configure the containerized OpenStack application
  - Application tarball contains helm charts and armada manifest for StarlingX
  - Automatic generation of helm configuration values based on system configuration
  - User can easily customize helm configuration of OpenStack Services



# Supported OpenStack Services

- Configuration optimized and system validated within StarlingX
  - Keystone, Nova, Neutron, Glance, Cinder, Horizon, Heat, Barbican, Ironic
  - Telemetry
    - Ceilometer, Gnocchi, Panko, Aodh

# Day 2 Configuration Changes

- Configuration changes can be applied after the application has been deployed
- Update the helm chart overrides
  - system helm-override-update ...
- Reapply the application
  - system application-apply ...
- Only charts impacted by the configuration change will be updated

# Distributed Cloud

# Distributed Cloud Overview

- Introduced in StarlingX 3.0
- Heterogeneous Distribution of Kubernetes and OpenStack Clouds
- Central Cloud (System Controller)
  - Hosting shared services
  - System-wide infrastructure orchestration functions
- Remote, geographically dispersed edge clouds
  - Communication with the System Controller node through REST APIs/L3
  - Running a control plane for autonomous operation
- In line with the [Distributed Control Plane](#) reference architecture model defined by the OSF Edge Computing Group

# Distributed Cloud - System Controller

- Centralized deployment of container platform on sub-clouds
  - Automated and declarative configuration
- Sub-cloud health monitoring and management
- Synchronized User Authentication & Authorization with Keystone
- Centralized Docker registry for infrastructure and applications
- Centralize Horizon dashboard - single pane of glass
- Configuration portal for shared platform data
  - DNS, NTP/PTP, API Firewall, SNMP, Remote Logging, ...
- Glance image caching in edge clouds

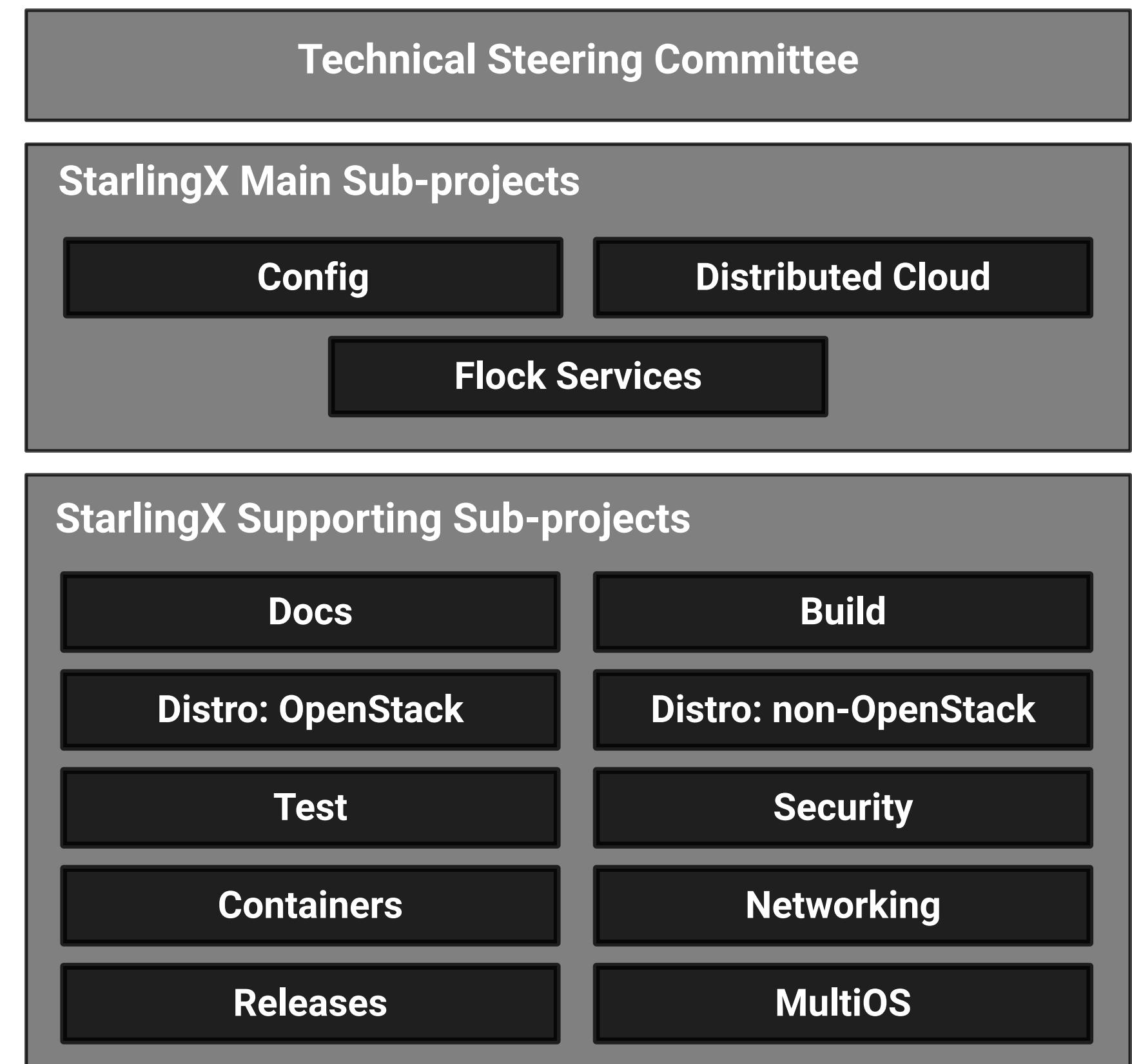
# Community and Contributing

# Principles

- The StarlingX project follows the “four opens,”
  - Open Collaboration
  - Open Design
  - Open Development
  - Open Source
- Technical decisions will be made by technical contributors and a representative Technical Steering Committee.
- The community is committed to diversity, openness, encouraging new contributors and leaders to rise up.

# Sub-project Structure

- Main sub-projects
  - New functionality and services
- Supporting sub-projects
  - Supporting services, test and infrastructure
- Sub-project team structure
  - 1 Team Lead
  - 1 Project Lead
  - Core Reviewers
  - Contributors



# Governance Roles

- **Contributor**
  - Someone who made a contribution in the past 12 months
    - Code, test or documentation
    - Serving in a leadership role
  - Can run and vote for elected positions
- **Core Reviewer**
  - Active contributors to a sub-project, appointed by fellow core reviewers
  - Responsible for reviewing changes and specifications
  - Can approve code and documentation changes

# Governance Roles

- Technical Lead
  - Per sub-project
  - Core Reviewer with additional duties
  - Helps guiding the technical direction of a sub-project
- Project Lead
  - Sub-project level coordination work
    - Tracks and communicates progress and priorities
  - Sub-project ambassador

# Governance Bodies

- Technical Steering Committee (TSC)
  - Responsible for overall project architectural decisions
  - Managing the sub-project life-cycle
  - Making final decisions if sub-project Core Reviewers, Technical Leads or Project Leads disagree
  - The TSC members are selected by the community through an election process

# Get Involved

- Code and documentation are available through git
  - [git.starlingx.io](https://git.starlingx.io)
- Apache 2 license
- IRC: #starlingx@Freenode
- Mailing List for daily discussions
  - <http://lists.starlingx.io/cgi-bin/mailman/listinfo/starlingx-discuss>

# Where to Contribute?

- Bugs are tracked in Launchpad
  - <https://bugs.launchpad.net/starlingx>
- New ideas are introduced in the specs repository
  - <https://git.openstack.org/cgit/openstack/stx-specs/>
- Design and implementation work is tracked in StoryBoard
  - [https://Storyboard.openstack.org/#!/project\\_group/86](https://Storyboard.openstack.org/#!/project_group/86)
- Further information about sub-teams and processes
  - <https://wiki.openstack.org/wiki/StarlingX>

# Community

- You do not need to be an Individual Member of the OpenStack Foundation in order to contribute, but if you want to vote in the annual OpenStack Foundation Board of Directors election, you may join: [openstack.org/join](https://openstack.org/join)
- If you are contributing on behalf of an employer, they will need to sign a corporate contributor license agreement, which now covers all projects hosted by the OpenStack Foundation (same model such as Apache and CNCF)

# Communication

- #starlingx@Freenode, IRC channel for online discussions
- Mailing Lists: [lists.starlingx.io](https://lists.starlingx.io)
- Email: [info@starlingx.io](mailto:info@starlingx.io)
- Weekly meetings:
  - Zoom calls
  - <https://wiki.openstack.org/wiki/Starlingx/Meetings>
- Twitter handle: @StarlingX

# Thank You!