StarlingX Project Overview

Learn, Try, Get Involved!

Let Me Introduce StarlingX

- New, top-level OpenStack Foundation pilot project
- Software stack providing high performance, low latency, and high availability for Edge Cloud applications
- The first release came out in October, 2018
  - https://git.starlingx.io/
- Growing community
  - Inviting users, operators and developers to try out the software and participate in the community

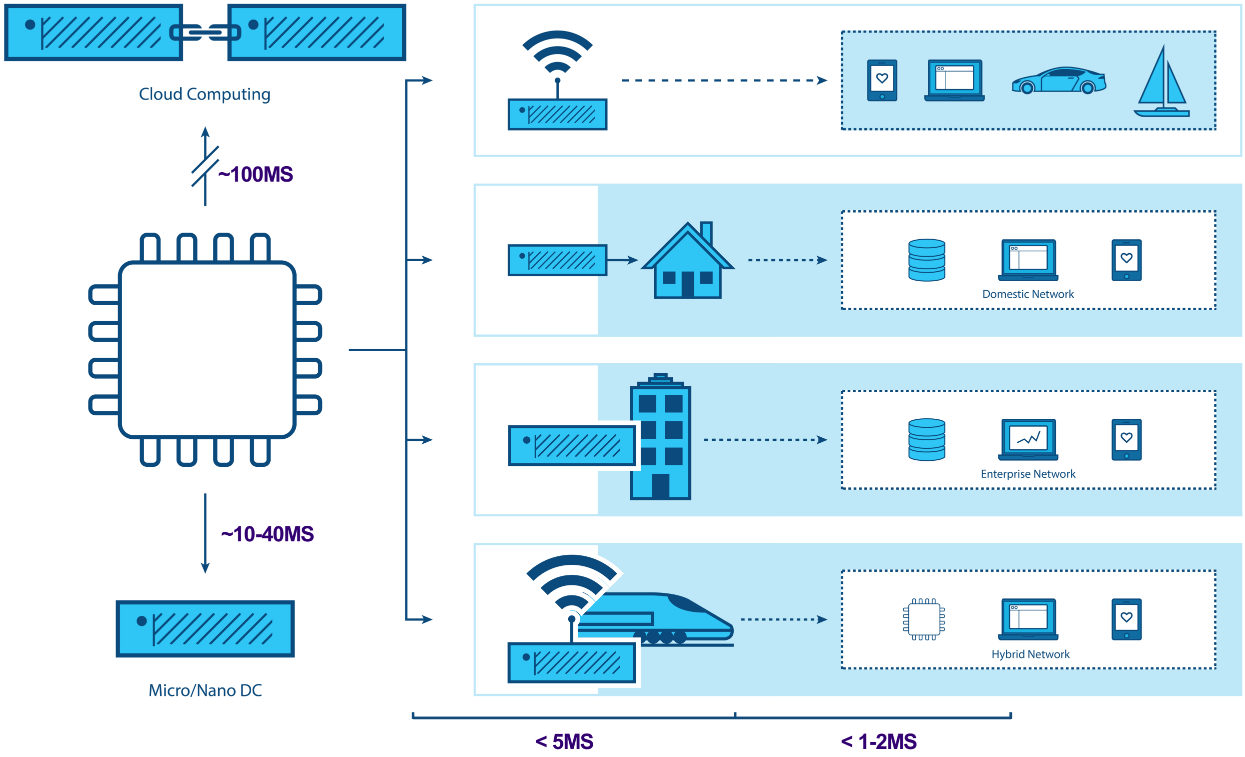
Project Overview

STARLINGX

What Is Driving Edge Computing?

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

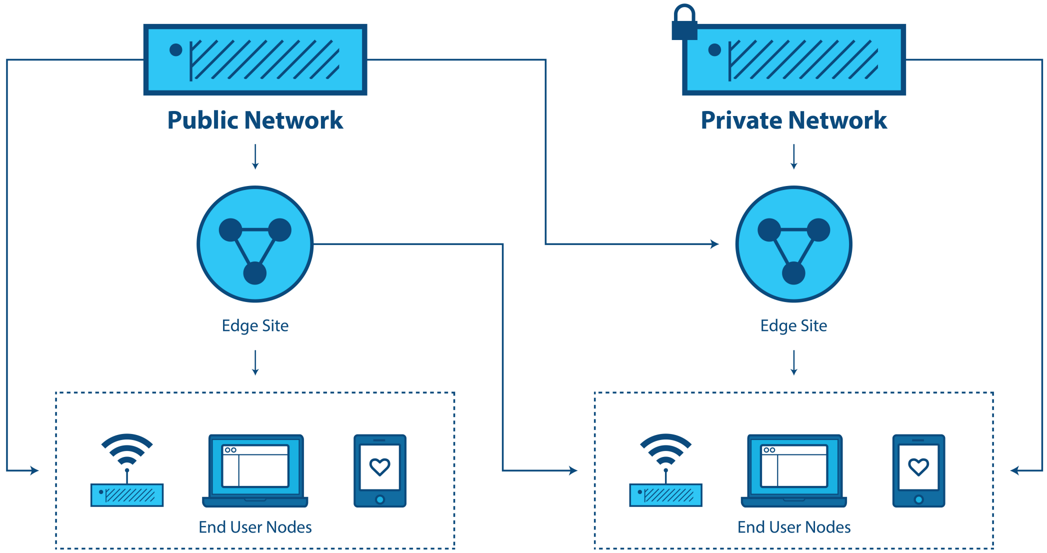
“WHERE” MATTERS



What Problems Is StarlingX Solving?

Massive data growth

Network needs to be smarter

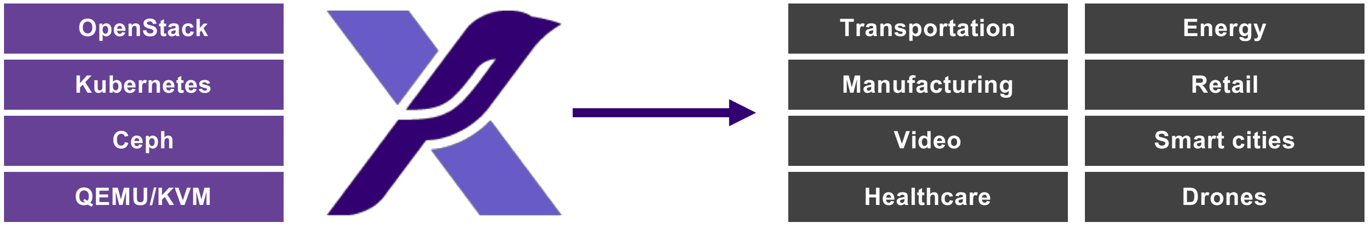


1. Distributed infrastructure demands a different architecture
2. The maturity and robustness of Cloud is required everywhere
3. Managing a massively distributed compute environment is hard

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



OpenStack
Kubernetes
Ceph
QEMU/KVM

Transportation
Manufacturing
Video
Healthcare

Energy
Retail
Smart cities
Drones

*Other names and brands may be claimed as the property of others

StarlingX Technology

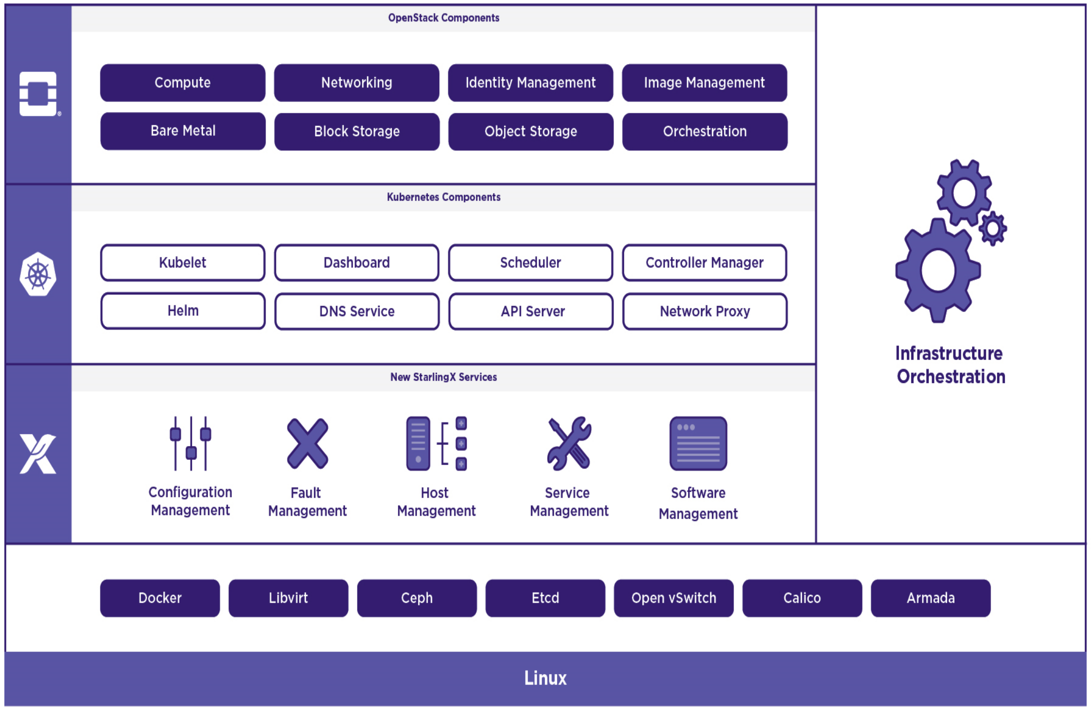
StarlingX –Edge Virtualization Platform

STARLINGX

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



OpenStack Components

Compute
Networking
Identity Management
Image Management
Bare Metal
Block Storage
Object Storage
Orchestration

Kubernetes Components

Kubelet
Dashboard
Scheduler
Controller Manager
Helm
DNS Service
API Server
Network Proxy

New StarlingX Services

Configuration Management
Fault Management
Host Management
Service Management
Software Management

Infrastructure Orchestration

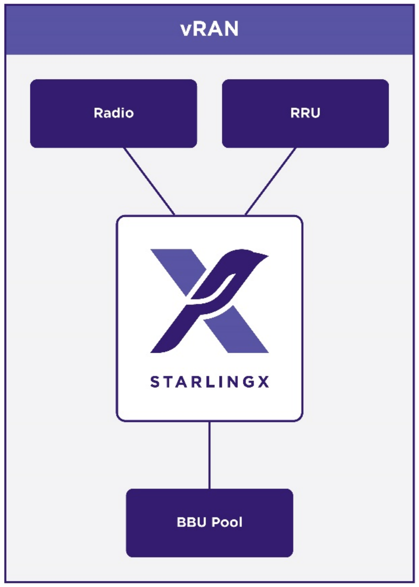
Docker
Libvirt
Ceph
Etcd
Open vSwitch
Calico
Armada

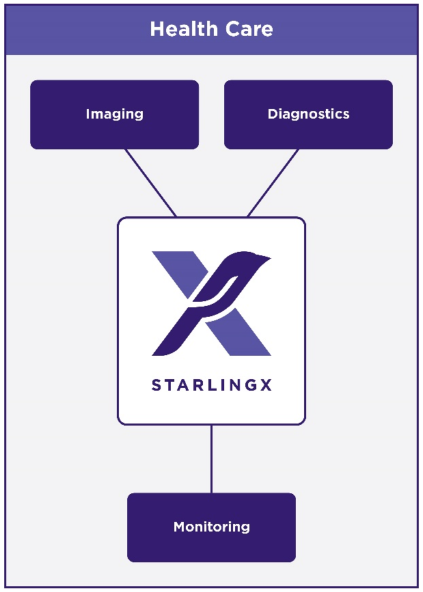
Linux

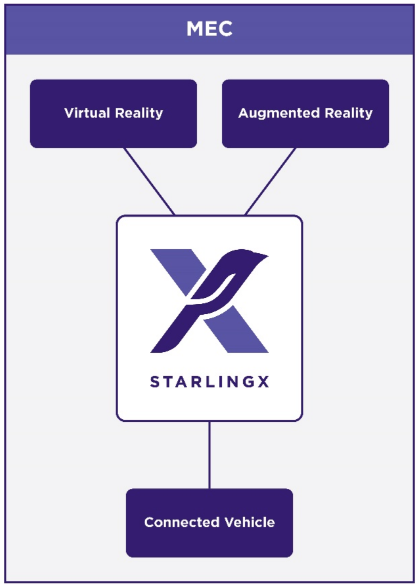
There are more OpenStack and Kubernetes components used than represented in this diagram.

* Other names and brands may be claimed as the property of others

Edge Computing Use Cases

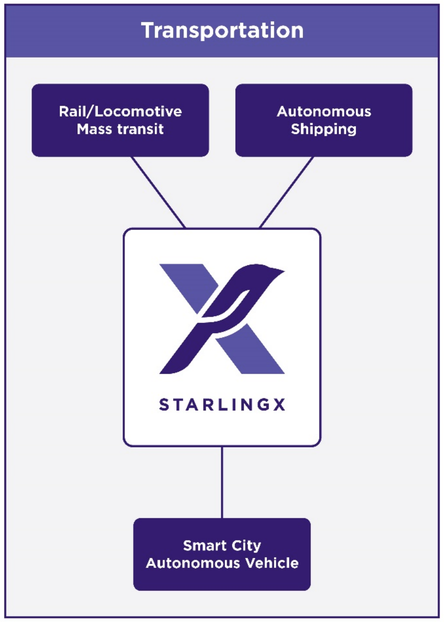


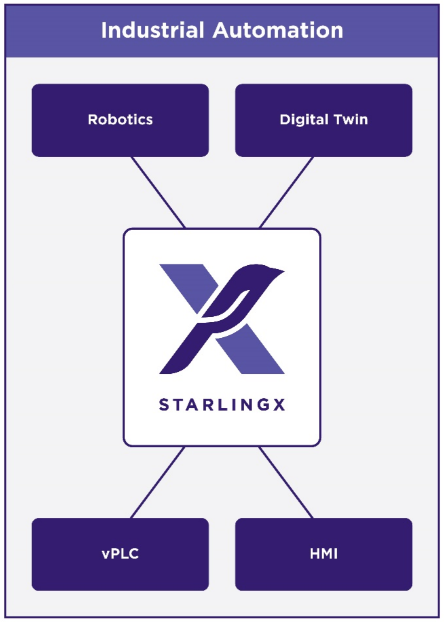


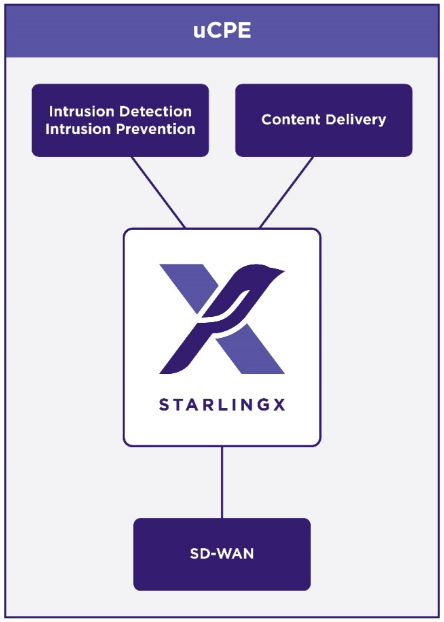


MEC == Multi-access Edge Computing

Edge Computing Use Cases

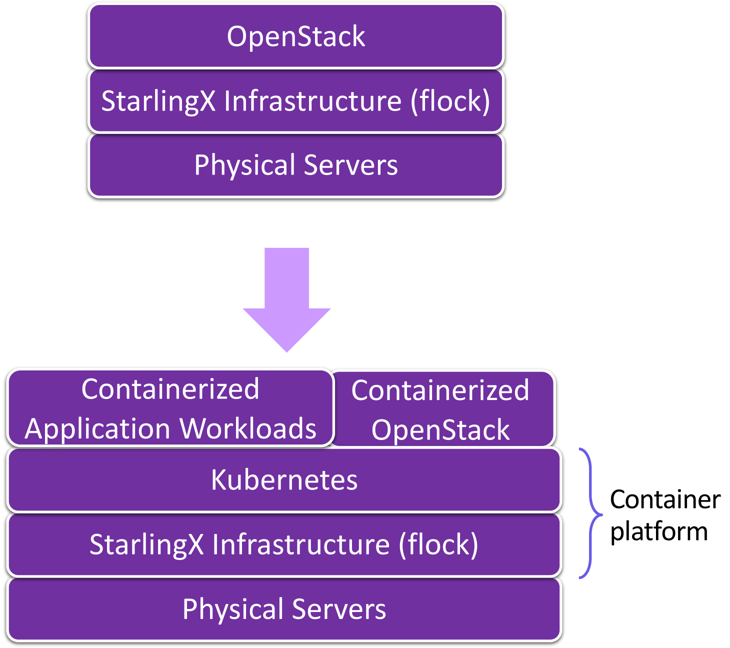






StarlingX –Evolution

- StarlingX Release 1.0 provided
  - A hardened OpenStack platform on dedicated physical servers
- StarlingX Release 2.0 will provide
  - A hardened cloud-native kubernetes platform on dedicated physical servers
  - Containerized Openstack
  - Ability to deploy kubernetes only Edge sites for containerized workloads



OpenStack

StarlingX Infrastructure (flock)

Physical Servers

Containerized Application Workloads

Containerized OpenStack

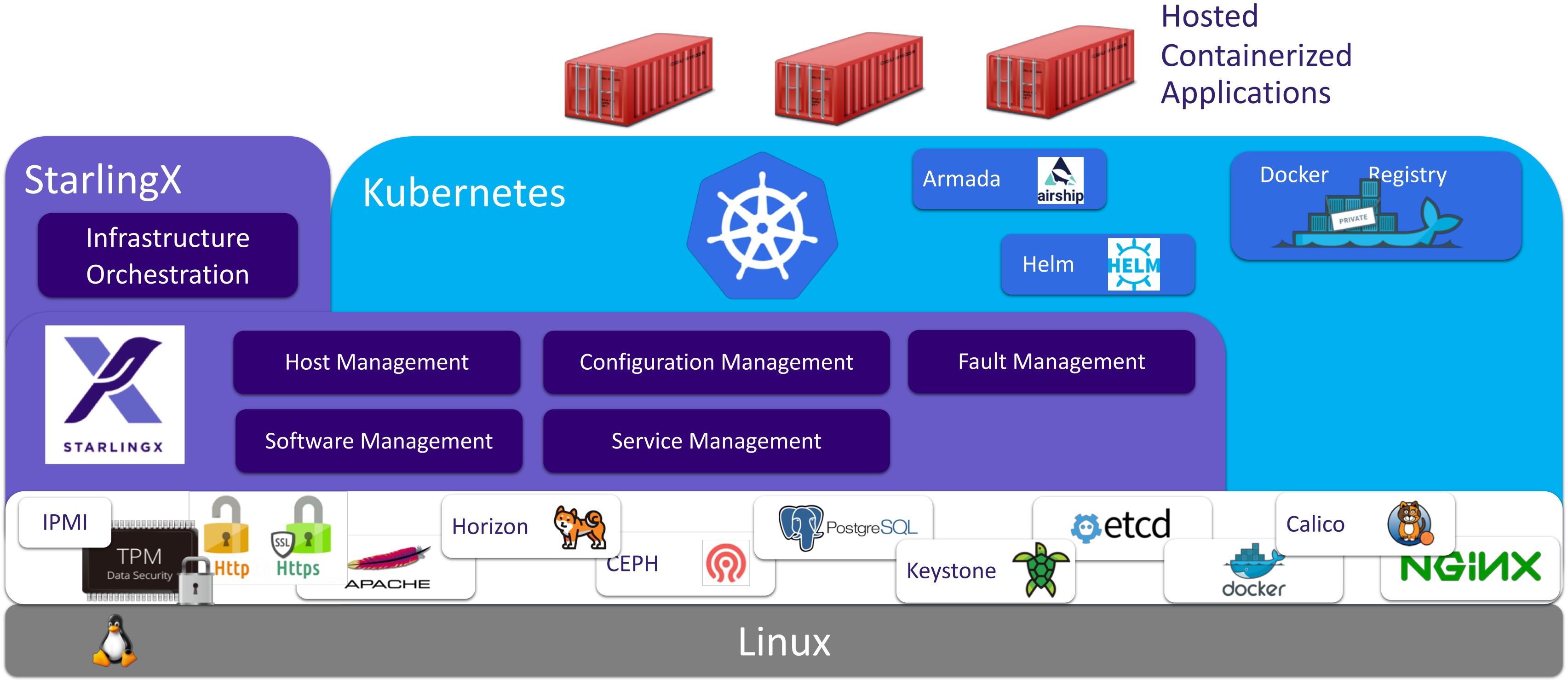
Kubernetes

StarlingX Infrastructure (flock)

Physical Servers

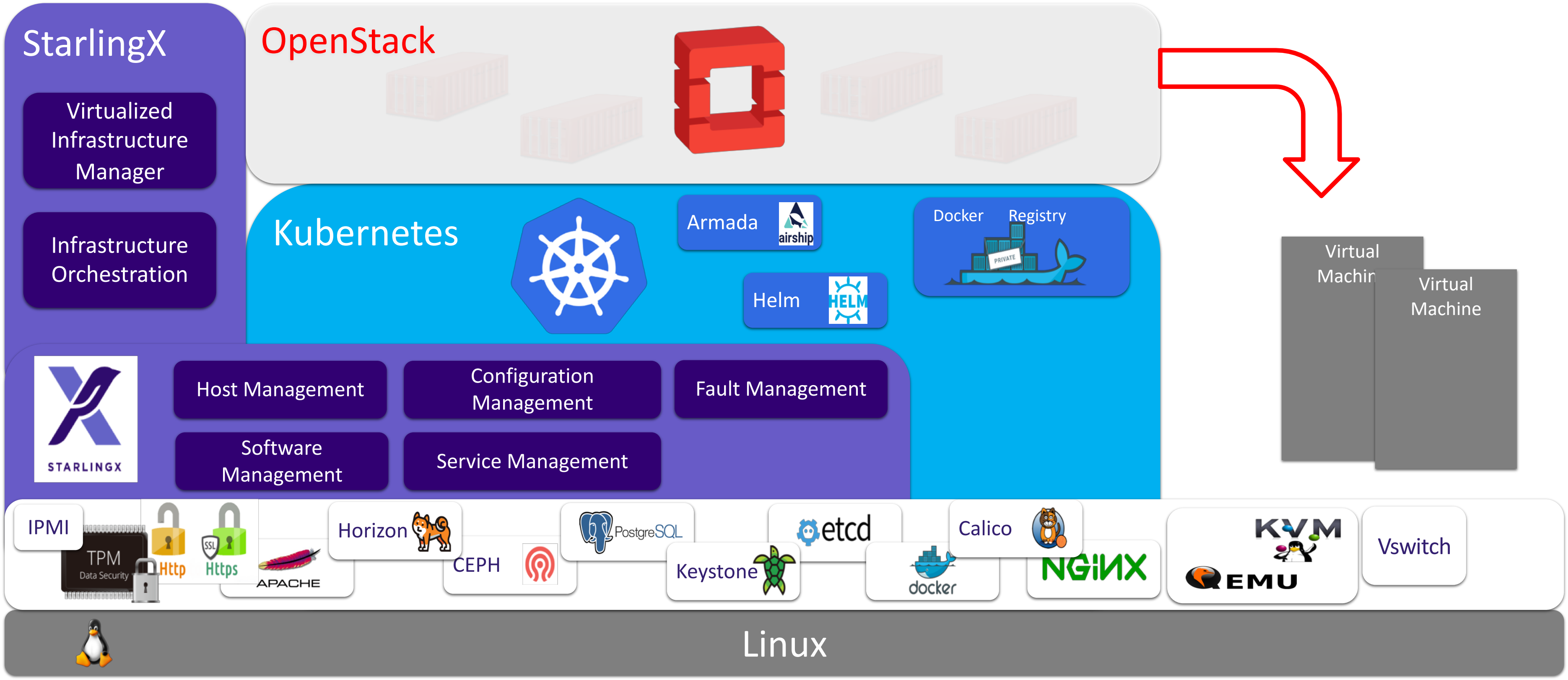
Container platform

StarlingX Software Architecture



STARLINGX

StarlingX Software Architecture



StarlingX

Virtualized Infrastructure Manager

Infrastructure Orchestration

OpenStack

Kubernetes

Armada

airship

Helm

HELM

Docker Registry

PRIVATE

Virtual Machin

Virtual Machine

STARLINGX

Host Management

Configuration Management

Fault Management

Software Management

Service Management

IPMI

TPM

Data Security

Http

Https

APACHE

Horizon

CEPH

PostgreSQL

Keystone

etcd

docker

Calico

NGINX

KVM

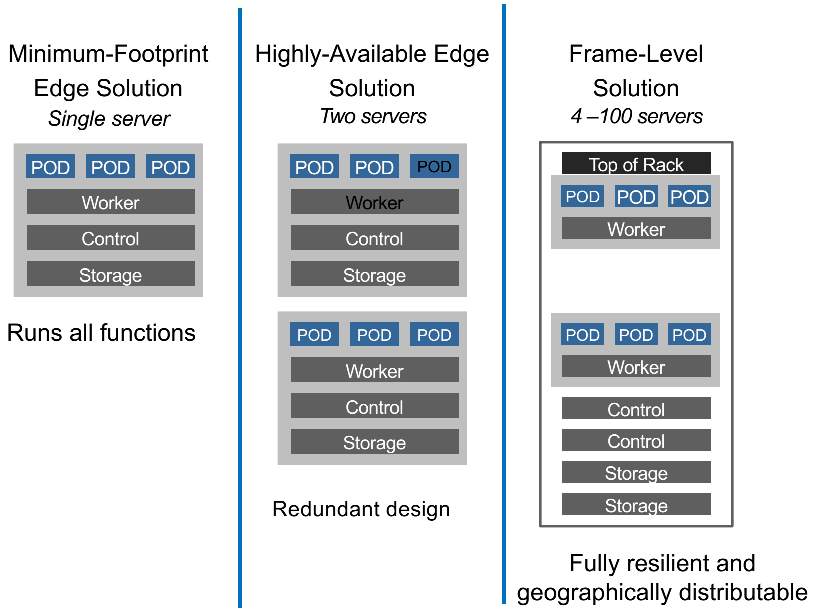
QEMU

Vswitch

Linux

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



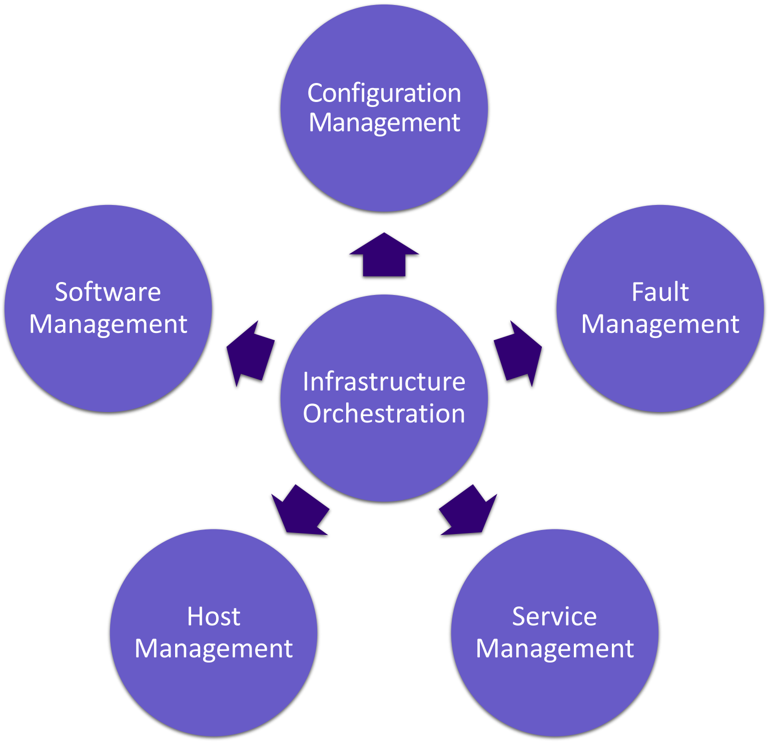
STARLINGX

Infrastructure Management
“The Flock”

Infrastructure Management

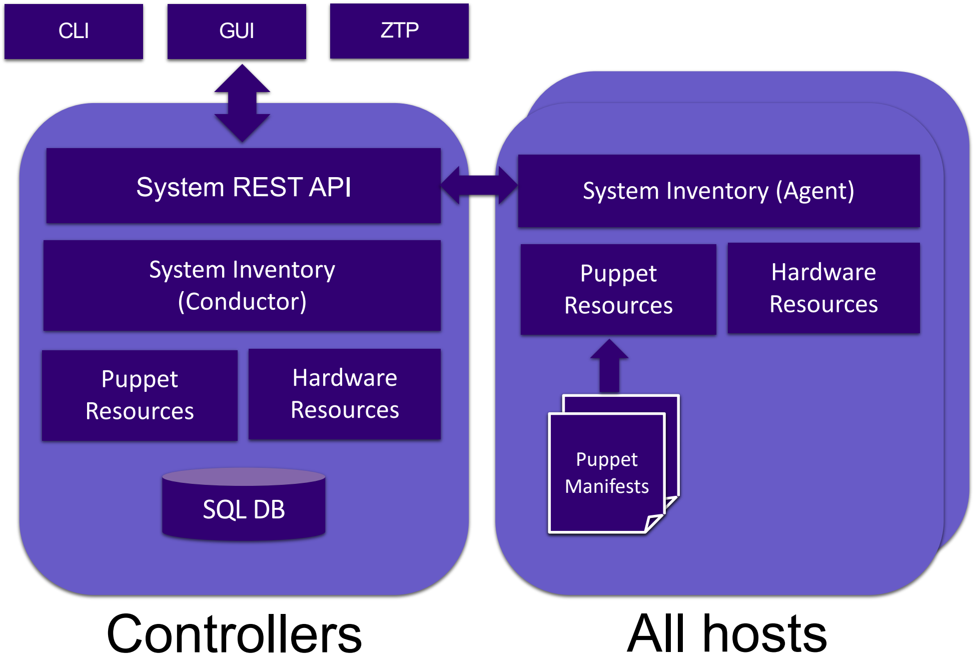
https://www.starlingx.io/

- StarlingX provides a deployment-ready, scalable, highly reliable infrastructure software platform
- Services from the StarlingX container and virtualization platform focus on:
  - Easy deployment
  - Low touch manageability
  - Rapid response to events
  - Fast recovery



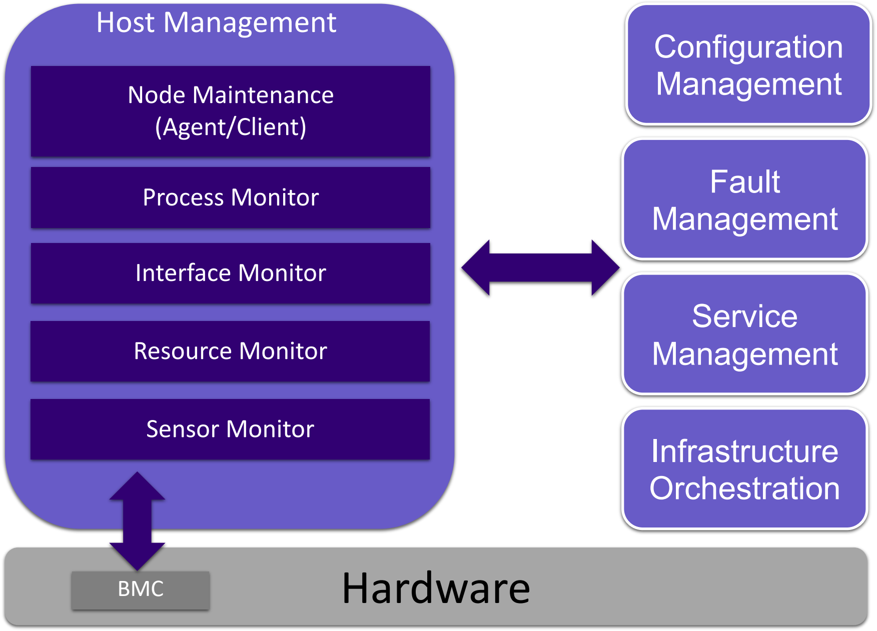
Configuration Management

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



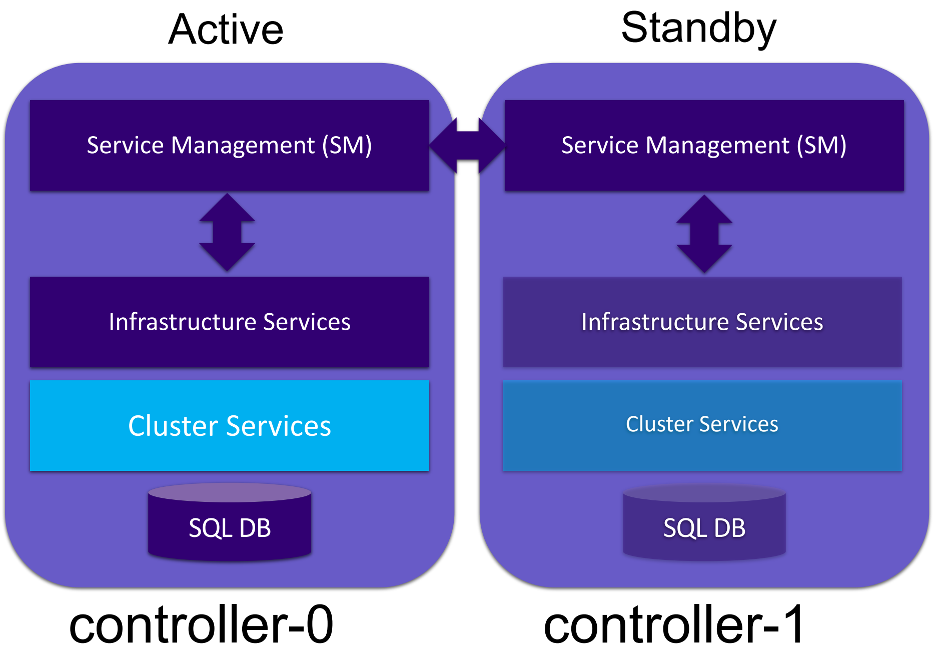
Host Management

- Full life-cycle and availability management of the physical hosts
- 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



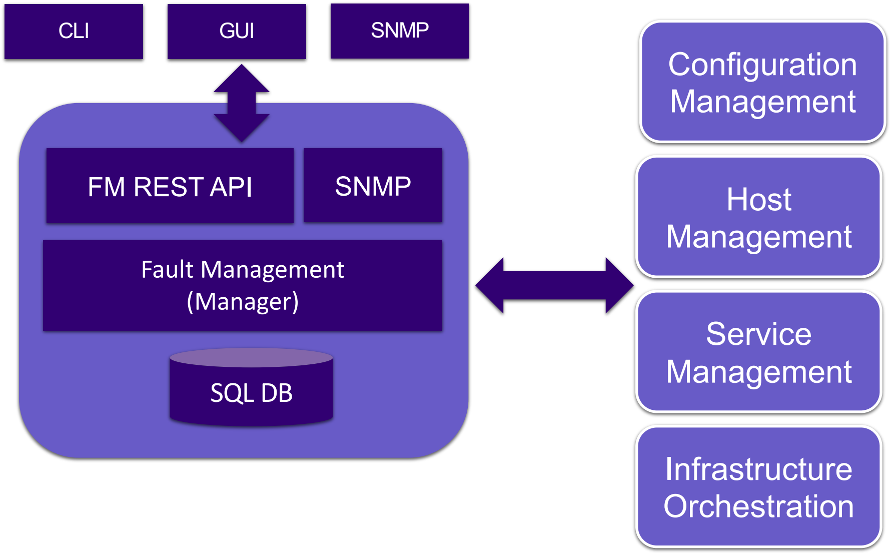
Service Management

- Manages high availability of critical infrastructure and cluster services
  - Supports many redundancy models: 1:1 N+1 N+M
  - Active or passive monitoring of services
  - Allows for specifying the impact of a service failure and escalation policy
  - Automatically recovers failed services
- 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



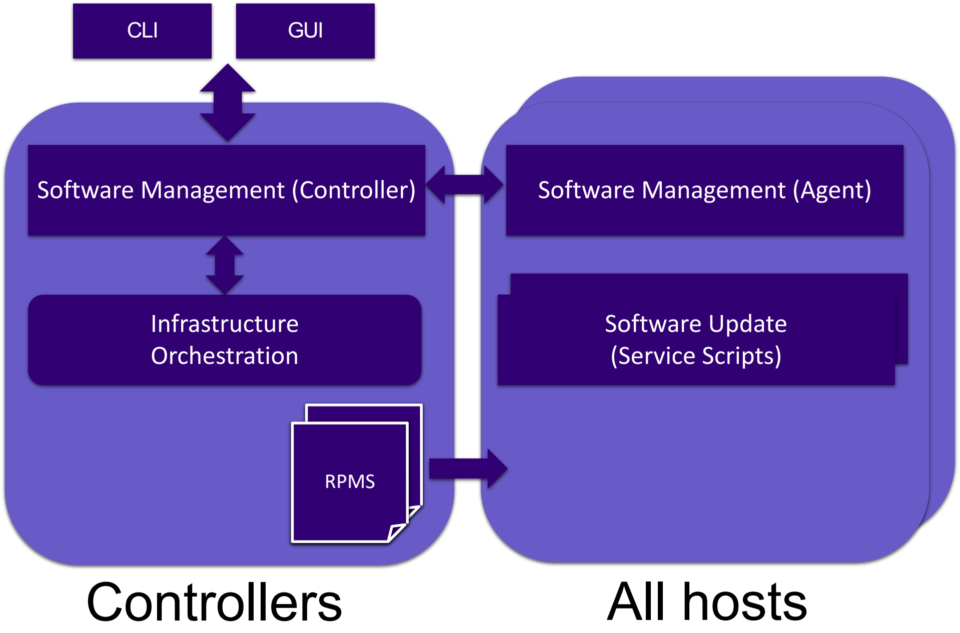
Fault Management

- Framework for infrastructure services to raise and persist alarm and event data
  - 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/Clears of alarms
  - Related to platform nodes and resources
  - Related to hosted virtual resources



Software Management

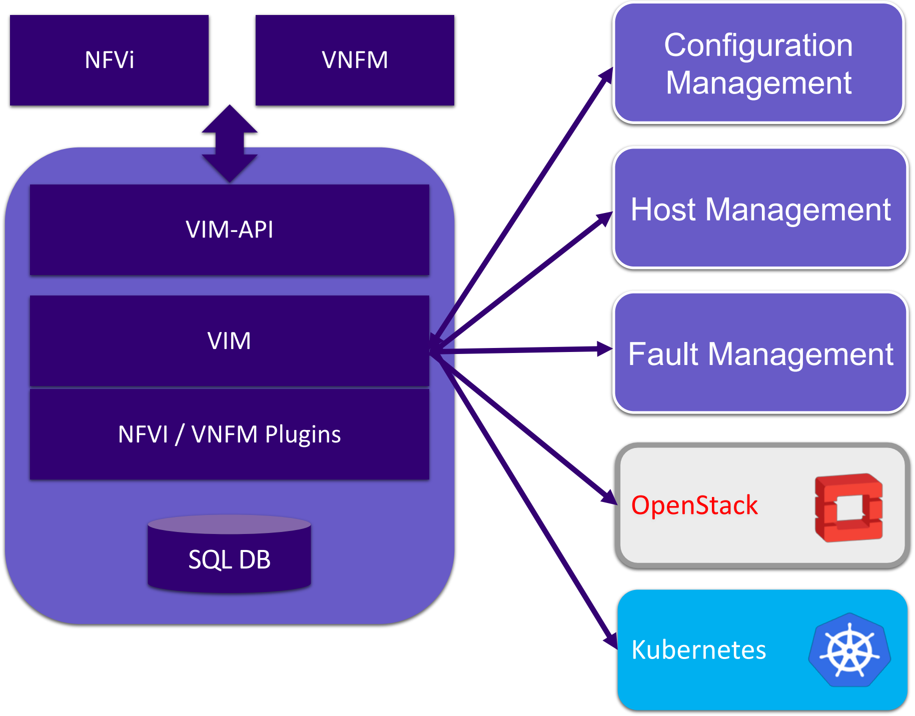
- Automated deploy of software updates for security and/or new functionality
- Integrated end-to-end rolling upgrade solution
  - Automated, low number of steps
  - No additional hardware required for upgrade
  - Rolling updates across nodes
  - Automatically moves Kubernetes Containers and migrates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


CLI
GUI
Software Management (Controller)
Software Management (Agent)
Infrastructure Orchestration
Software Update (Service Scripts)
RPMS
Controllers
All hosts

Infrastructure Orchestration

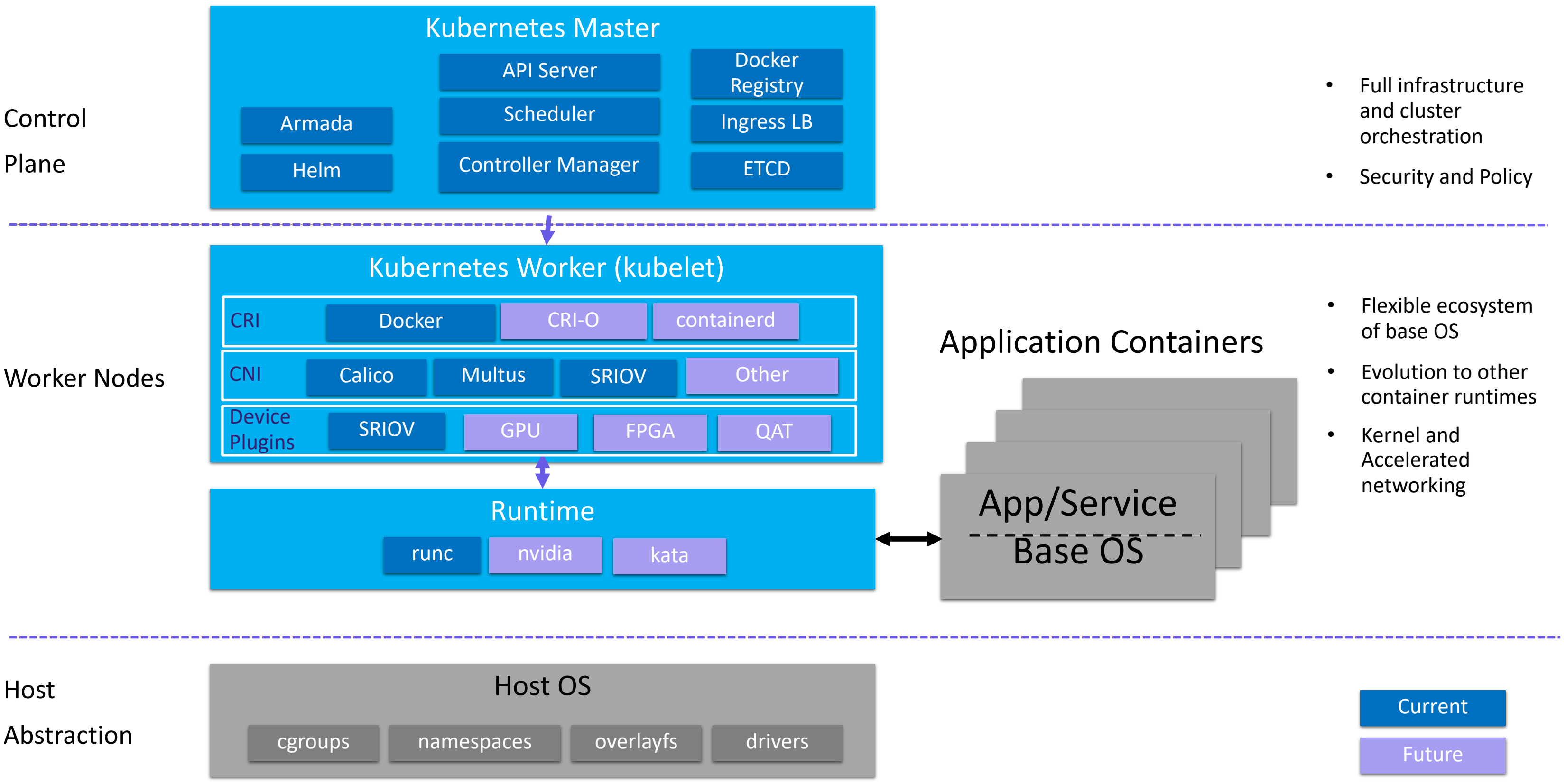
Virtual Infrastructure Manager (VIM)

- Orchestration of infrastructure service operations
  - Host maintenance (lock, unlock)
  - NFV MANO compliant API
  - NFVi plugins for lifecycle management
  - K8S node maintenance and Pod lifecycle
  - OpenStack L3/DHCP service scheduling and placement
- Orchestration of software management operations
  - Software update application of cluster
  - Software upgrade application of cluster
  - Coordinated multi-node update
  - Maintains service availability during operations



Container Platform

Kubernetes Cluster Software Components



Control Plane

Kubernetes Master

Armada

Helm

API Server

Scheduler

Controller Manager

Docker Registry

Ingress LB

ETCD

- Full infrastructure and cluster orchestration
- Security and Policy

Worker Nodes

Kubernetes Worker (kubelet)

CRI: Docker, CRI-O, containerd

CNI: Calico, Multus, SRIOV, Other

Device Plugins: SRIOV, GPU, FPGA, QAT

Runtime

runc, nvidia, kata

Application Containers

App/Service

Base OS

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

Host Abstraction

Host OS

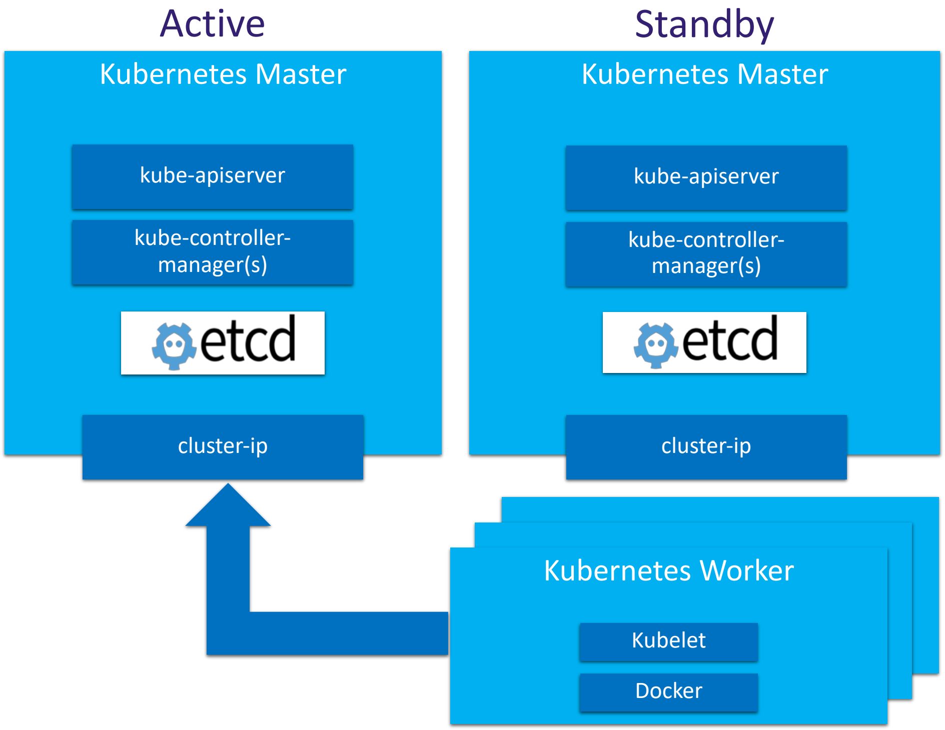
cgroups, namespaces, overlayfs, drivers

Current

Future

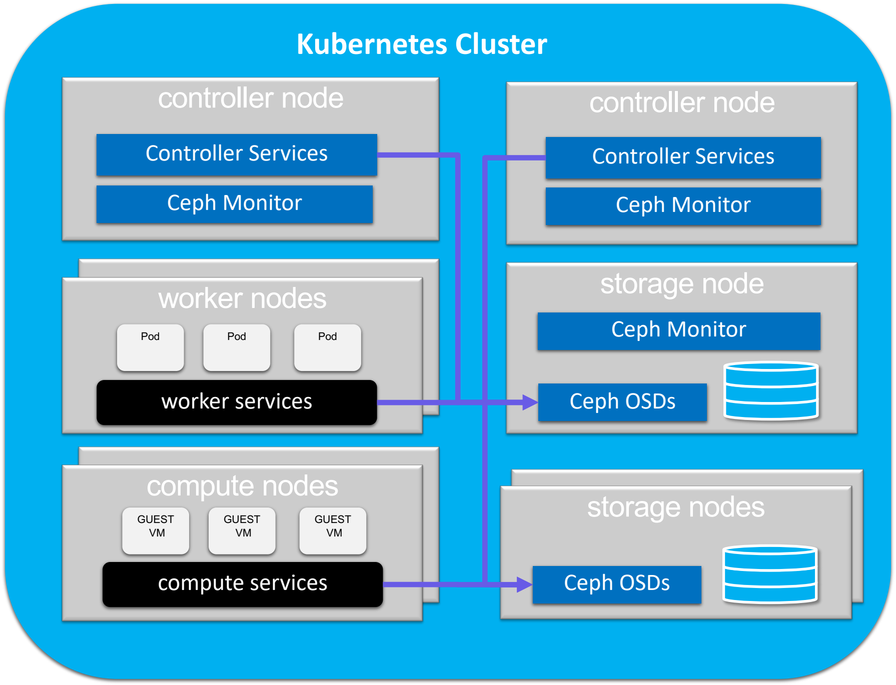
Kubernetes Deployment Architecture

- 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 actives 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



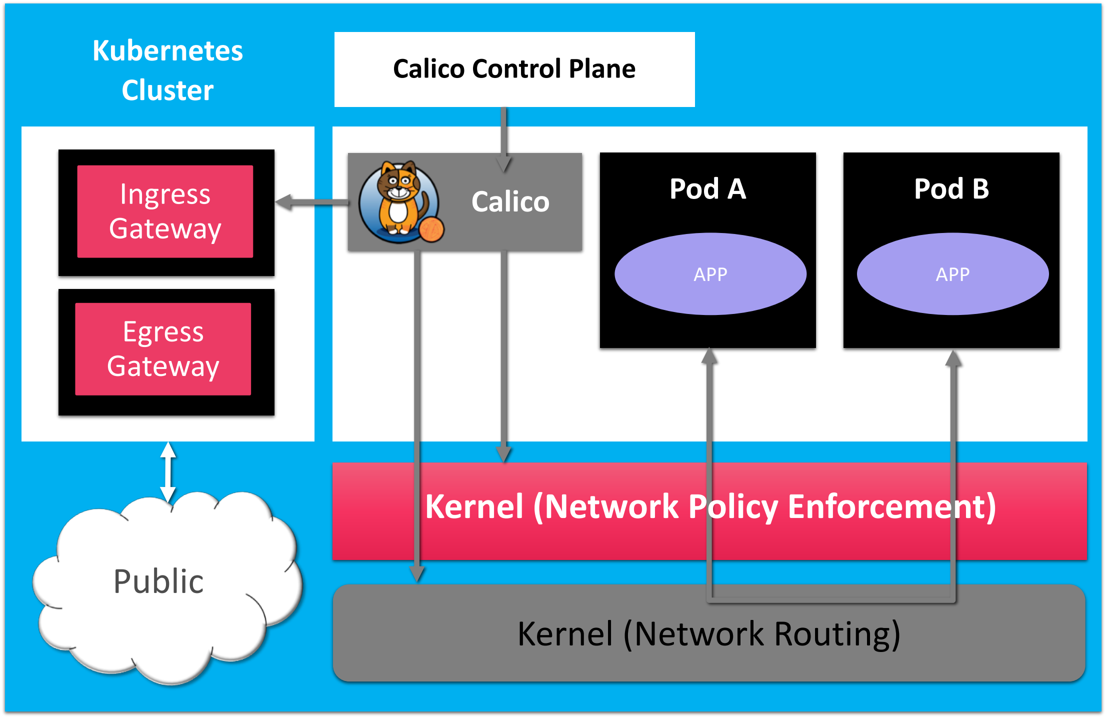
Cluster Persistent Storage

- 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

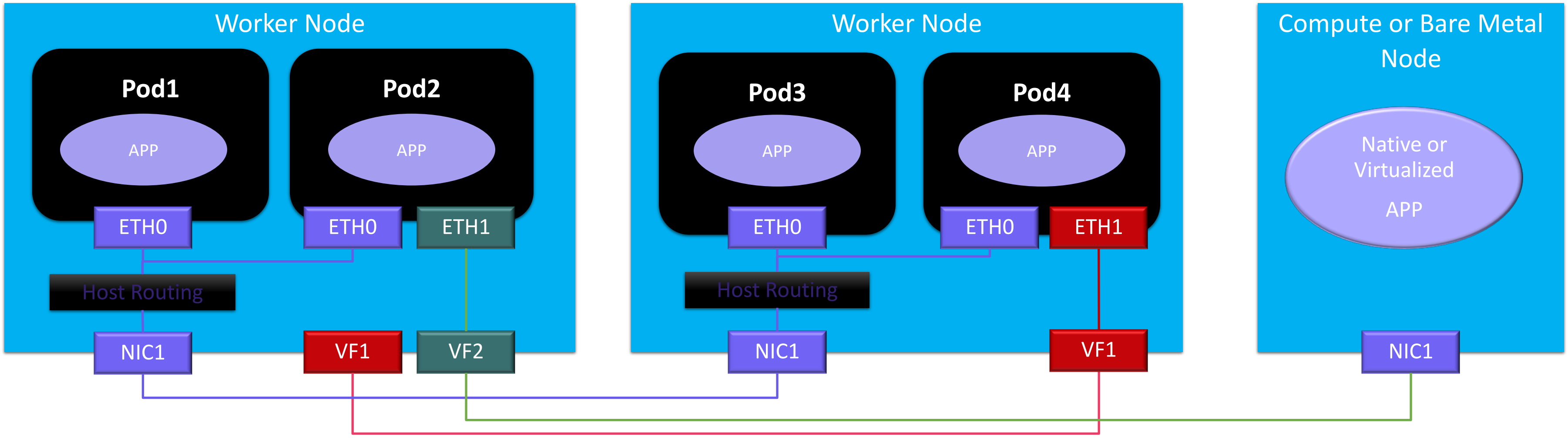
- Calico provides a pure L3 fabric solution for interconnecting containers
- Calico leverages the Linux kernel for routing and policy enforcement
- Calico leverages Border Gateway Protocol (BGP) for control plane
- Calico leverages Open Standards and is a full Open Source network solution
- Calico is 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, SRIOV, 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

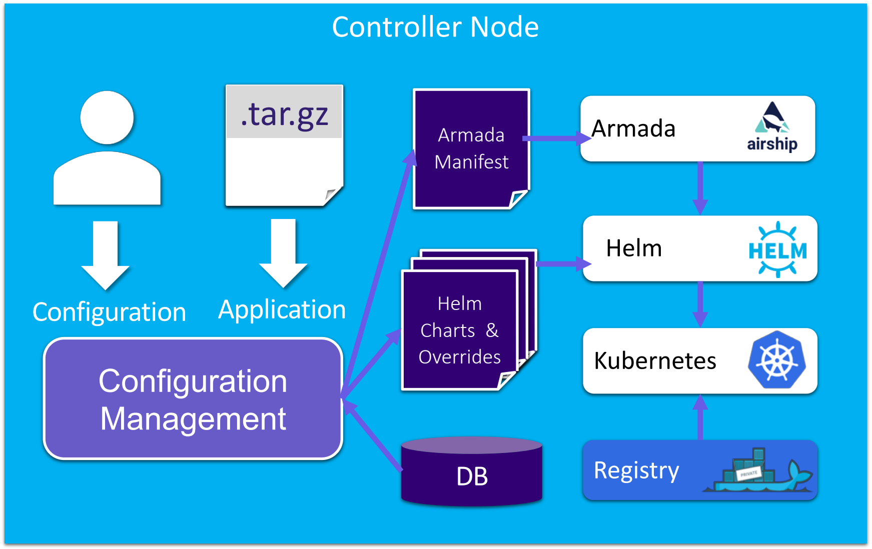


Additional Capabilities

- 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

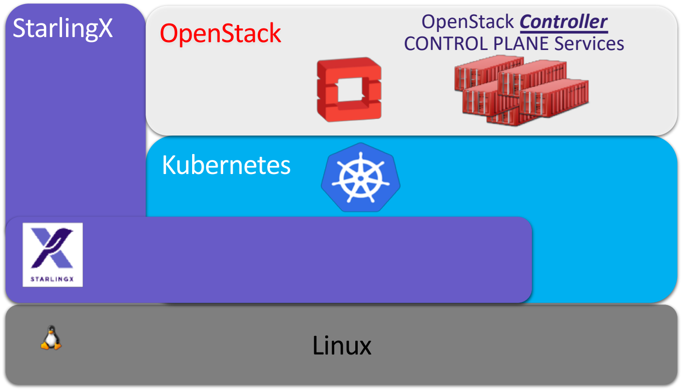
- OpenStack is deployed as a containerized Kubernetes application
  - OpenStack control plane running in pods
  - OpenStack virtual machines running on host
  - Leverages Kubernetes' strengths to manage, scale and update the OpenStack services
- Deployed using Helm (using OpenStack-Helm charts) and Armada (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

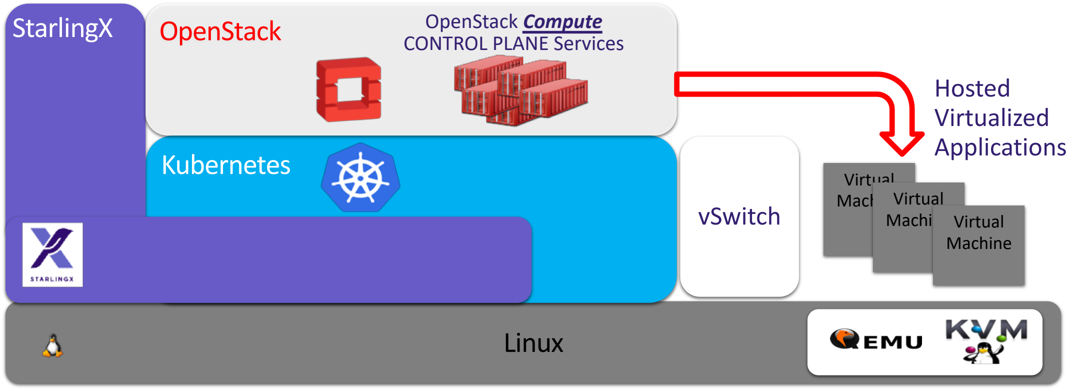
Configuration optimized and system validated within StarlingX:

- Keystone
- Nova
- Neutron
- Glance
- Cinder
- Horizon
- Heat
- Barbican
- Telemetry
  - Ceilometer
  - Panko
  - Gnocchi
  - Aodh
- Ironic

Kubernetes MASTER Nodes



Kubernetes WORKER Nodes



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

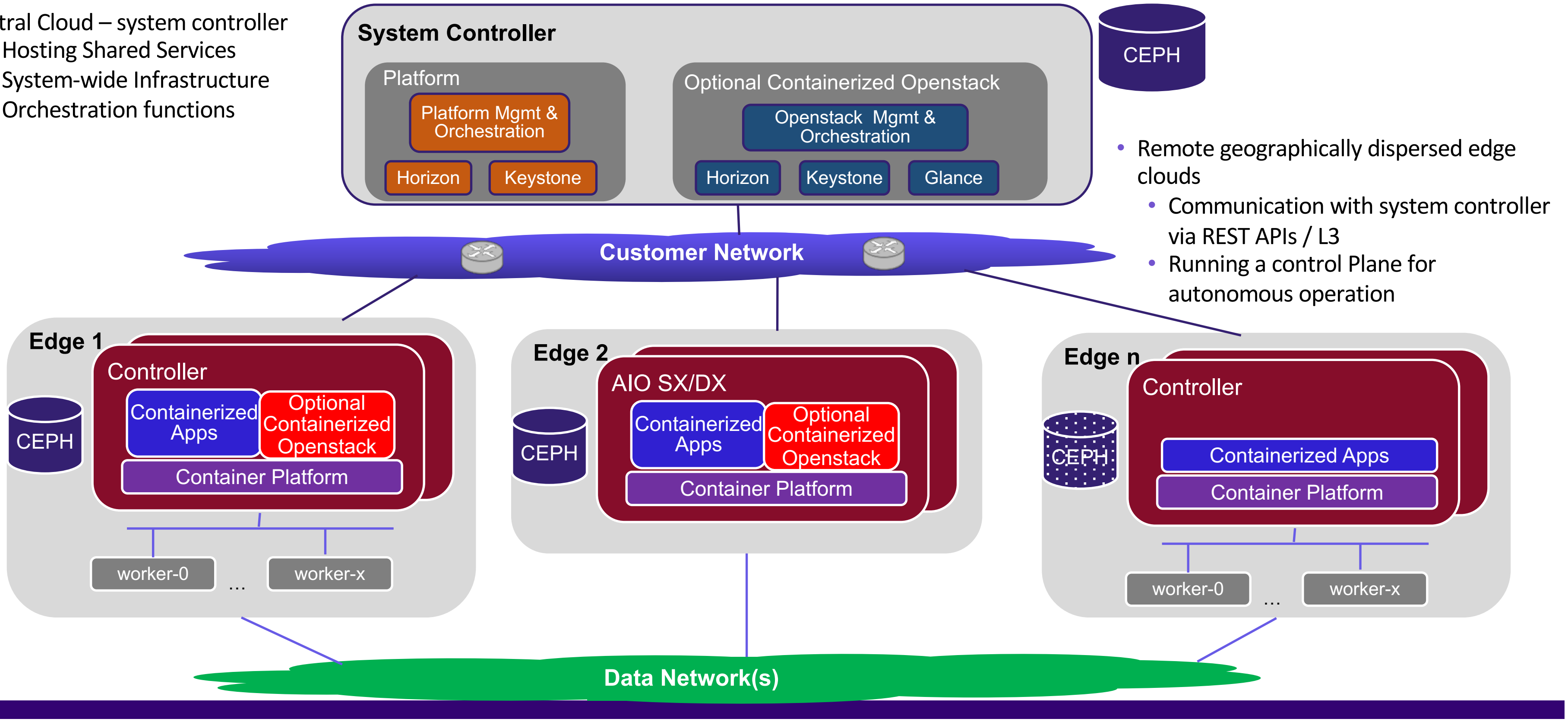
Distributed Cloud

Distributed Cloud Architecture

- 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 system controller via REST APIs / L3
  - Running a control Plane for autonomous operation



System Controller- Container Platform Features

- Deployment of container platform on sub-clouds from the system controller
  - Declarative and automated configuration
- Management and monitoring of sub-cloud health
- Synchronized User Authentication & Authorization leveraging keystone as the back end
  - System controller is the provisioning portal for keystone
- A secure centralized docker registry for infrastructure and applications
  - Could be integrated with infra and application CI/CD frameworks
- System wide software update orchestration
- System wide fault aggregation
- Configuration portal for shared platform data
  - ex. DNS, NTP/PTP, API Firewall, SNMP, Remote Logging
- Centralized Horizon dashboard for platform – single pane of glass

System Orchestration of distributed Kubernetes Edge Clouds

System Controller- Openstack Features

- Synchronized User Authentication & Authorization leveraging keystone as the back end
  - System controller is the provisioning portal for keystone
- Glance image caching in edge clouds
- Centralized Horizon dashboard for openstack services
  - Single pane of glass
- Configuration portal for shared openstack data

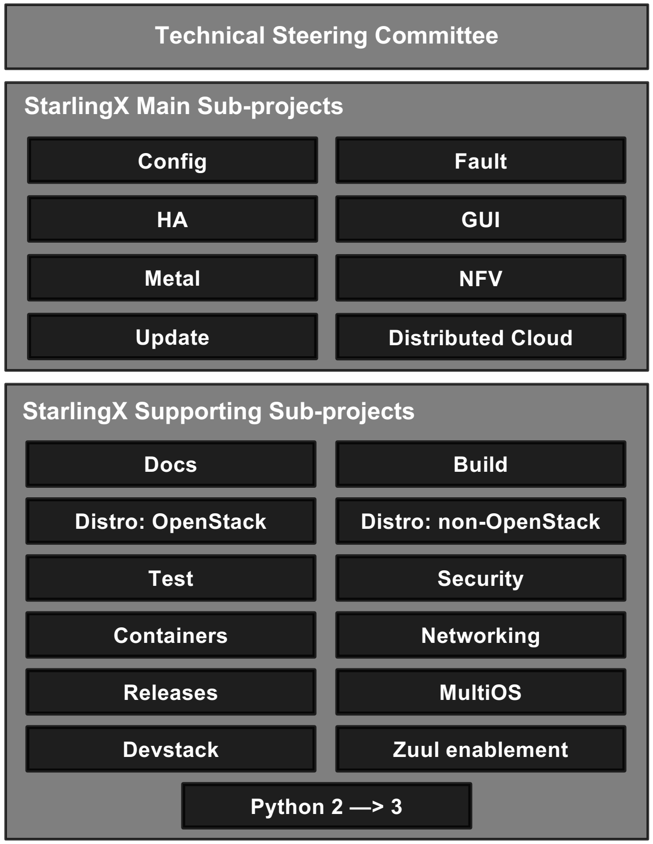
Community and Contributing

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



Technical Steering Committee

StarlingX Main Sub-projects

Config	Fault
HA	GUI
Metal	NFV
Update	Distributed Cloud

StarlingX Supporting Sub-projects

Docs	Build
Distro: OpenStack	Distro: non-OpenStack
Test	Security
Containers	Networking
Releases	MultiOS
Devstack	Zuul enablement
Python 2 —> 3	

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 merge 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
  - It will be comprised of 7 people, where the initial group will be appointed; the project will move to an election based system within the first year
  - The initial TSC members are Brent Rowsell (Wind River), Ian Jolliffe (Wind River), Dean Troyer (Intel) and Saul Wold (Intel)

Get Involved

- Code and documentation are available through git
  - 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
- 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
- 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
- Email: info@starlingx.io
- Weekly meetings:
  - Zoom calls
  - https://wiki.openstack.org/wiki/Starlingx/Meetings
- Twitter handle: @StarlingX

STARLINGX

Thank You!