边缘计算
Edge Computing
CONSORTIUM

开源边缘计算平台

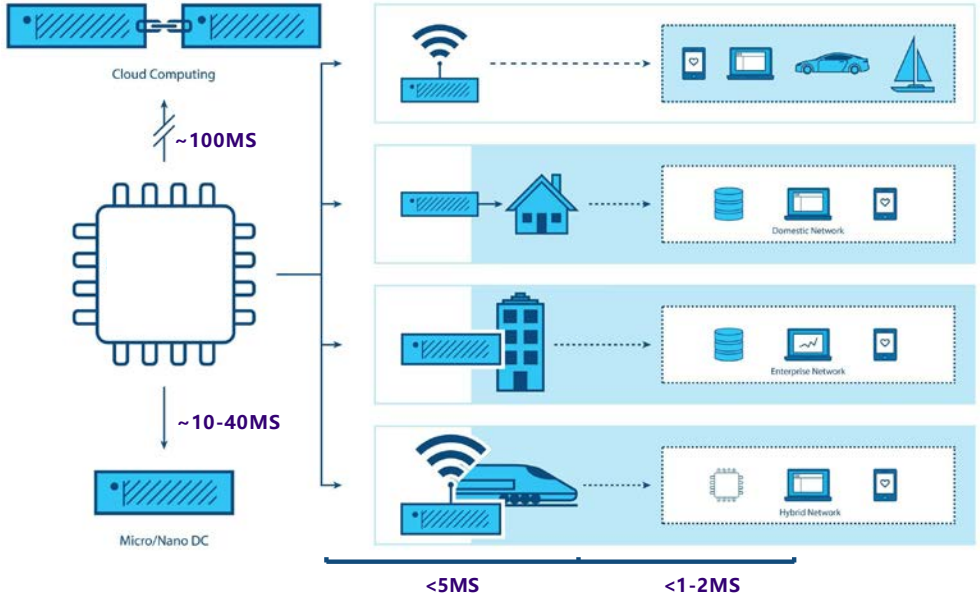
Introduce The Open Source Edge Computing Platform - StarlingX

汇报人：九州云信息科技有限公司
汇报日期： 2018年11月

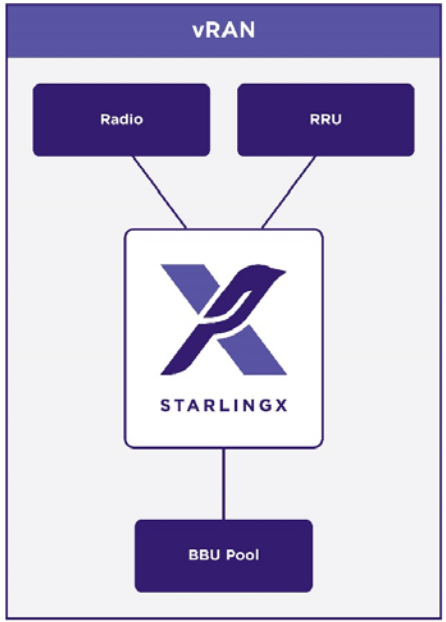
What Is Driving Edge Computing?

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

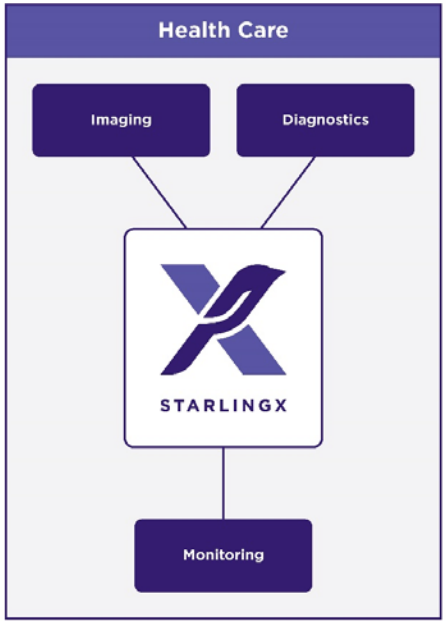
“WHERE” MATTERS

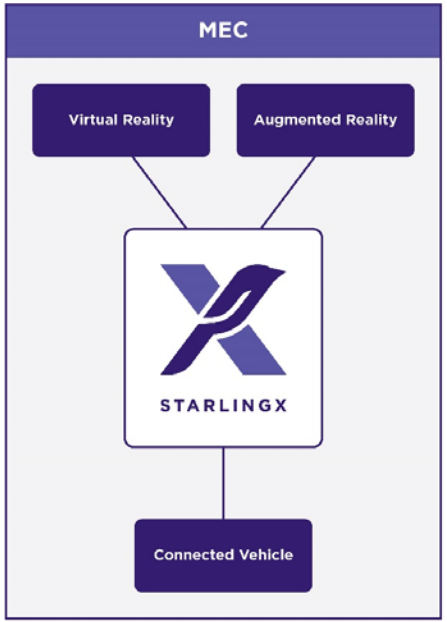


Edge Computing Use Cases



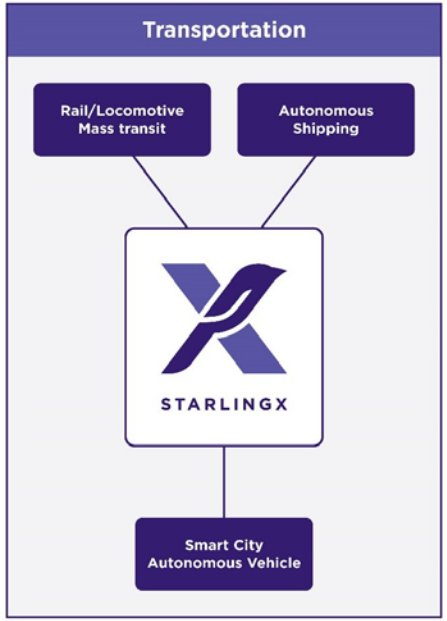
vRAN == virtual Radio Access Network

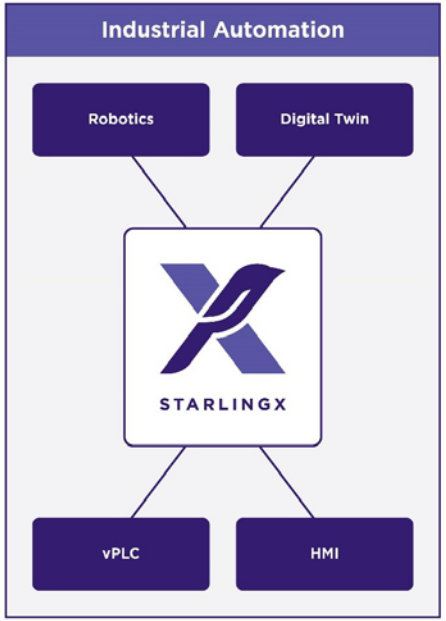


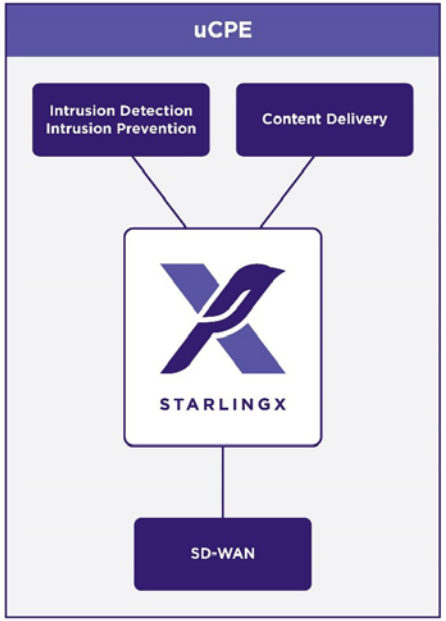


MEC == Multi-access Edge Computing

Edge Computing Use Cases







uCPE == universalCustomer Premises Equipment

Solving The Challenges: Edge Initiatives

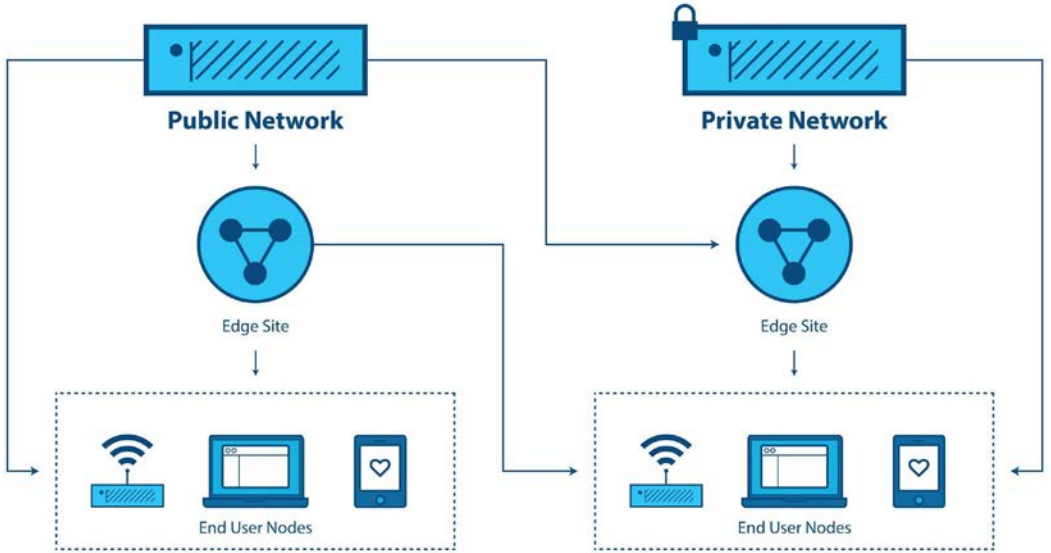
Edge Workgroup

Edge Workgroup

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

StarlingX Technology

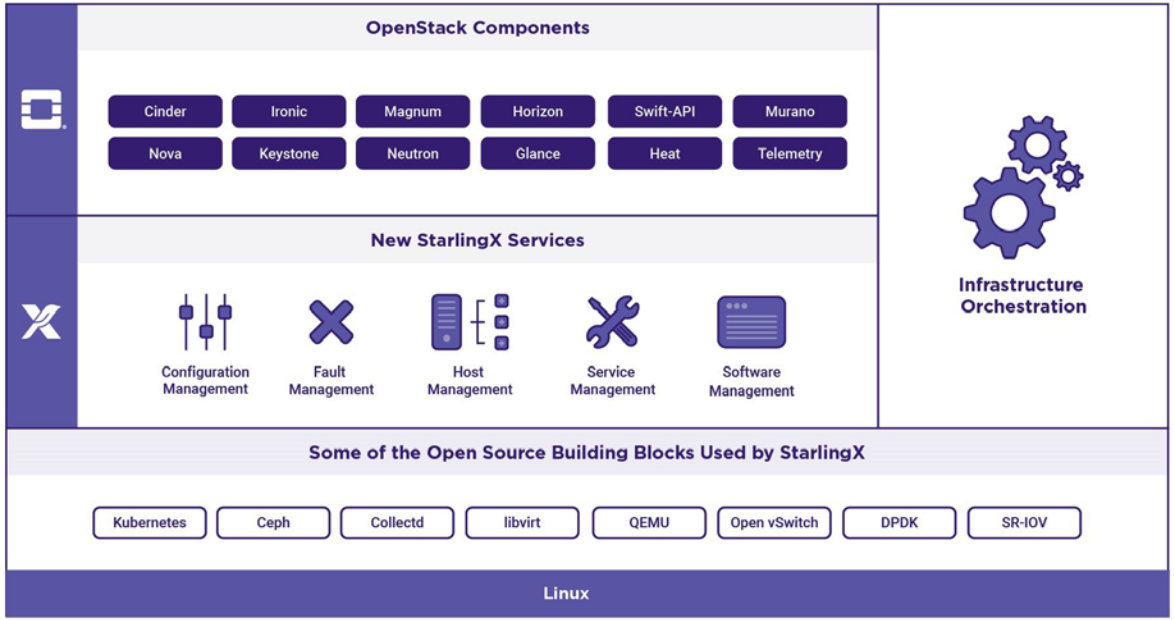
StarlingX – Edge Virtualization Platform

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**

Think control at the Edge, control between IoT and Cloud, control over your virtual machines.



Scalability from Small to Large

Single Server

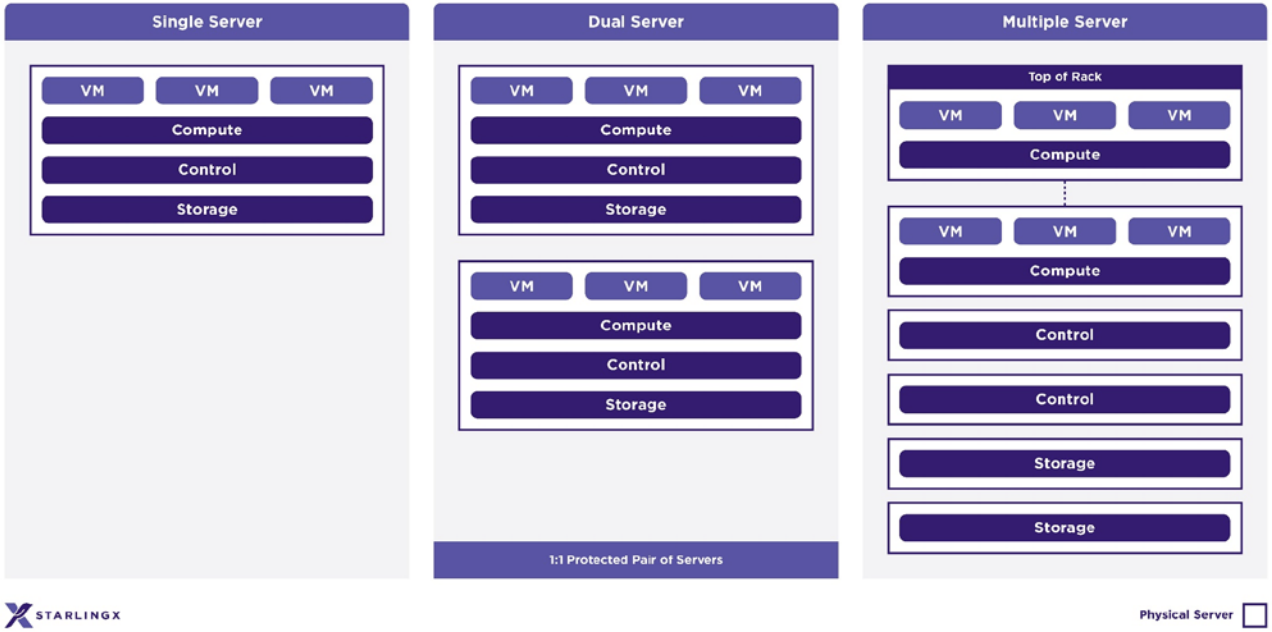
- Runs all functions

Dual Server

- Redundant design

Multiple Server

- Fully resilient and geographically distributable



Configuration Management

Manages installation

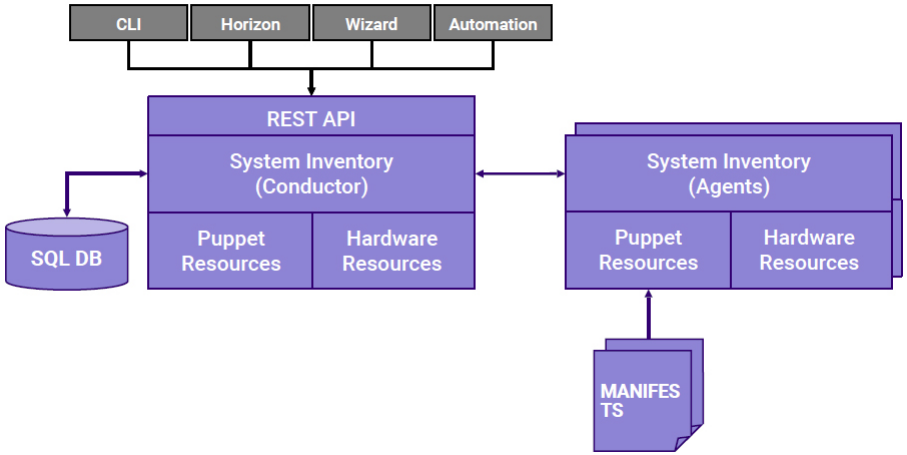
- Auto-discover new nodes
- Manage installation parameters (i.e. console, root disks)
- Bulk provisioning of nodes through XML file

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



System Configuration and Setup

Host Management

Full life-cycle management of the host

Detects and automatically handles host failures and initiates recovery

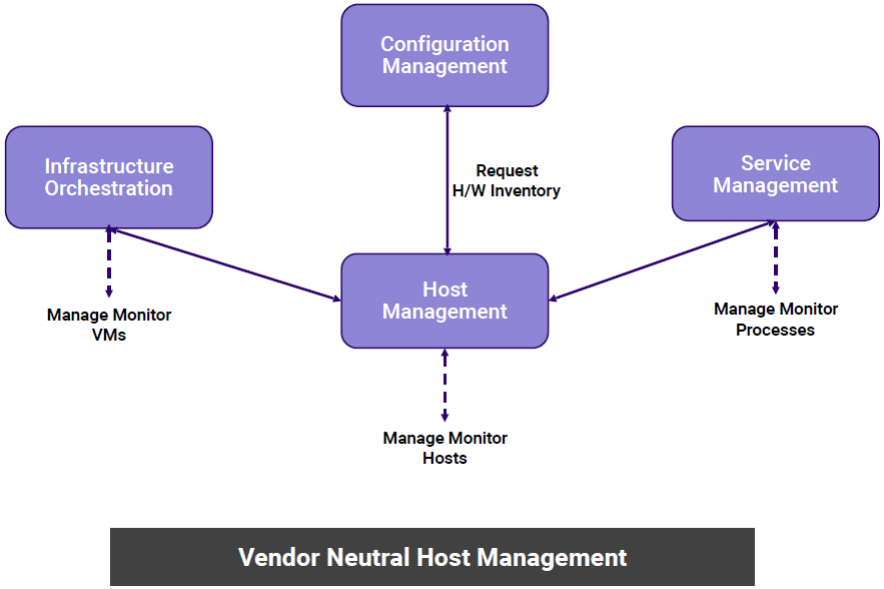
Monitoring and alarms for

- 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

Manage the host via RESTAPI



Configuration Management

High availability manager

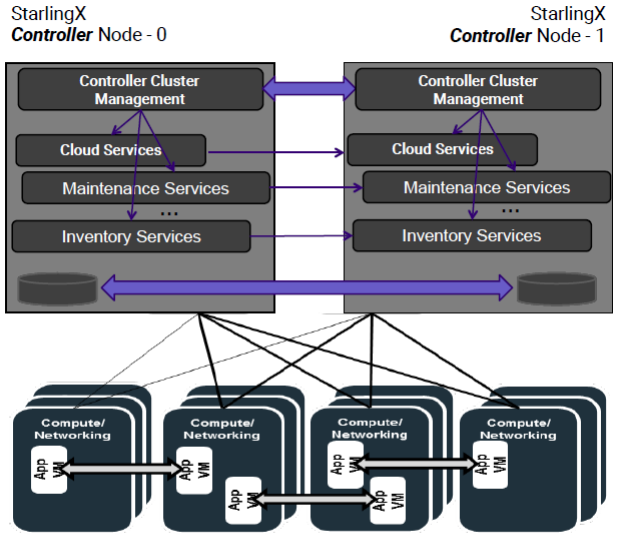
- Redundancy model can be N+M or N across multiple nodes
- Currently 1+1 HA Controller Cluster

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

Software Management

Automated deploy of software updates for security and/or new functionality

Integrated end-to-end rolling upgrade solution

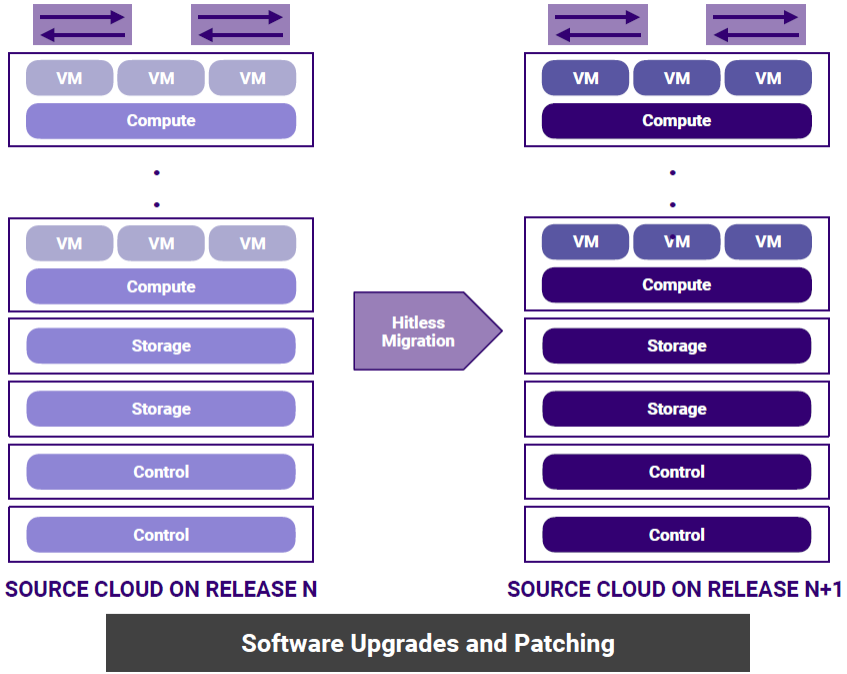
- Automated, low number of steps
- No additional hardware required for upgrade
- Rolling upgrade across nodes

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



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

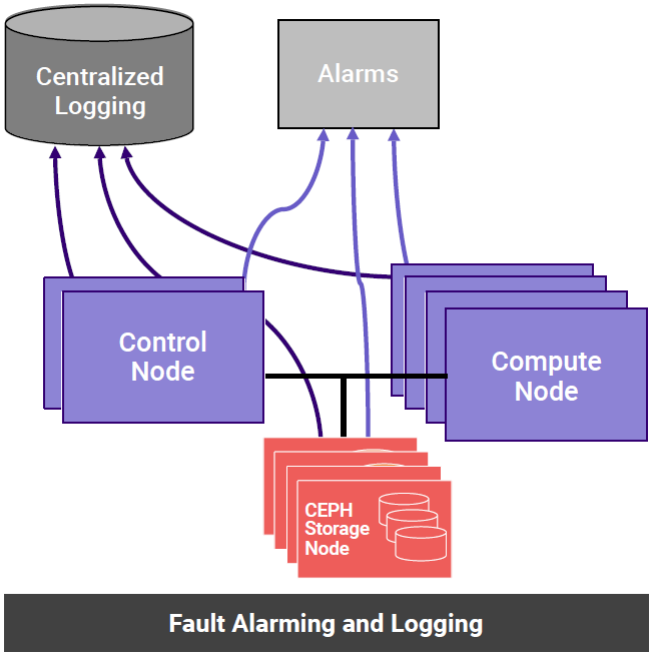
Support for alarm suppression

Operator alarms

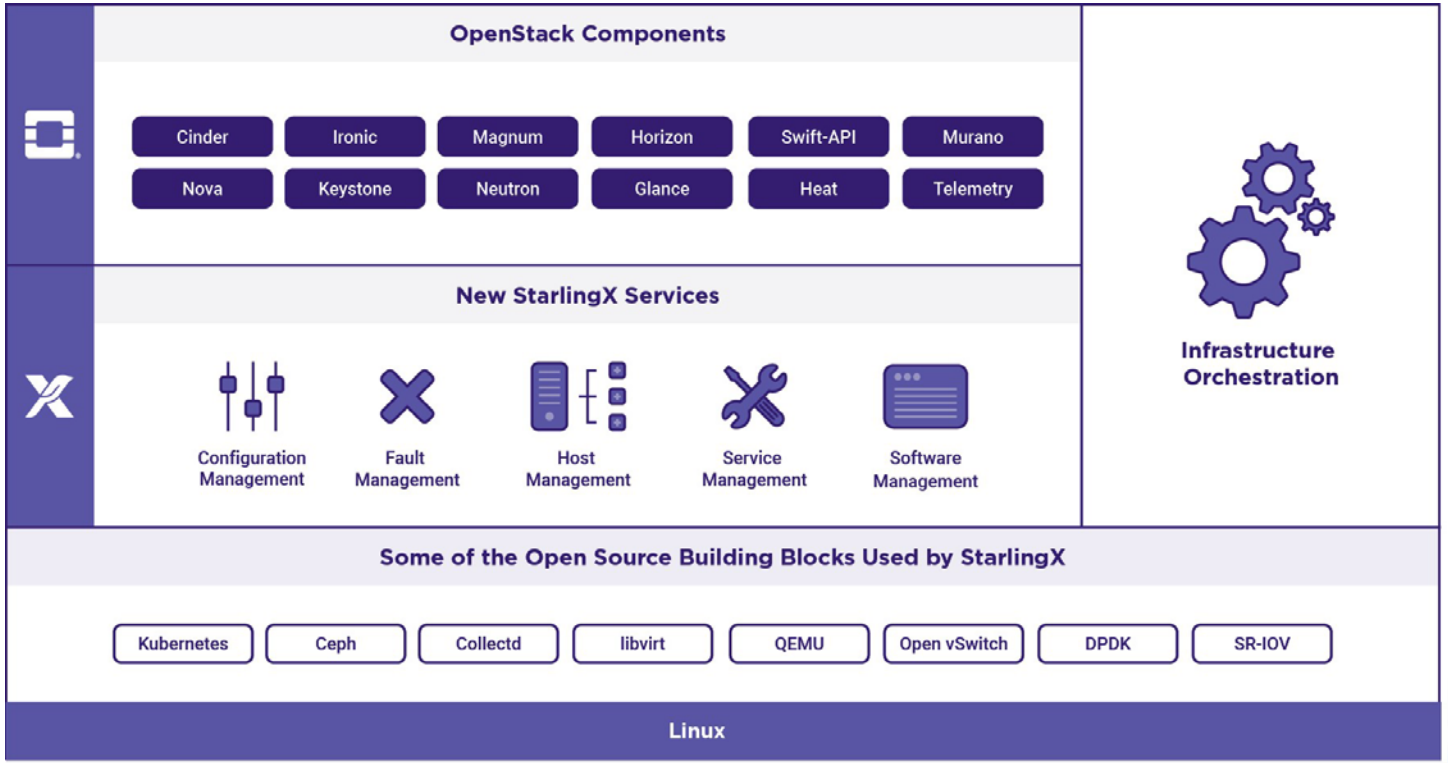
- On platform nodes and resources
- On hosted virtual resources

Operator logs - Event List

- Logging of set/Clears of alarms
- Related to platform nodes and resources
- Related to hosted virtual resources



Current Architecture



Next Generation Container Architecture

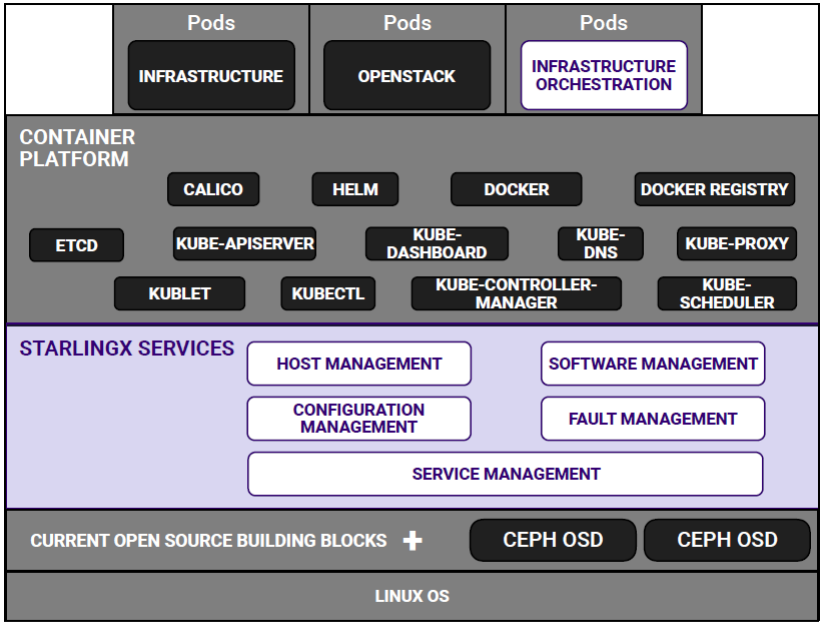
StarlingX is evolving to

- Running OpenStack containerized
- On top of a bare metal Kubernetes cluster
- With OpenStack Helm managing the lifecycle of the OpenStack cluster

With Kubernetes Cluster initially supporting

- Docker runtime
- Calico CNI plugin
- CEPH as persistent storage backend
- HELM as the package manager
- Local Docker Image Registry

Along with Kubernetes cluster available for non-OpenStack end user applications



Full Support for VMs and Containers

The Road to the Edge

Build it yourself from open source components

- Building blocks need refinement
- Time consuming
- Gaps to fill

Use StarlingX

- New services provide improved manageability for the platform and high availability for your applications to meet Edge Cloud requirements
- Tested and available as a complete stack
- Mission-ready for your applications

StarlingX Community & Activities

- StarlingX had the first release on October 24. The release includes 1,329 commits from 84 contributors, including developers representing Wind River, Intel, 99Cloud, China Unionpay, Suse, Fujitsu, and NEC, among others.

- After the first release the community is focusing on new functionality as well as making their tools and processes more efficient. As part of this work StarlingX is working on containerization it’ s infrastructure and improving the build and deployment experience both for users and developers.

- The first StarlingX Technical Steering Committee (TSC) has been formed with 8 members from 6 companies: Ana Cunha (Ericsson), Brent Rowsell (Wind River), Curtis Collicutt (INTERdynamix), Dean Troyer (Intel), Ian Jolliffe (Wind River), Miguel Lavalle (Huawei), Saul Wold (Intel), Shuquan Huang (99cloud)

Sub-project Structure

Main sub-projects

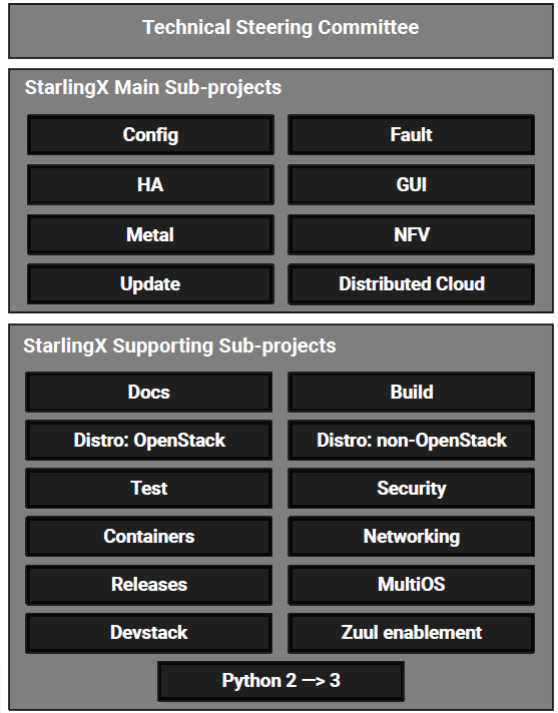
- New functionality and services

Supporting sub-projects

- Supporting services, test and infrastructure

Sub-project team structure

- 1 Team Lead
- 1 Project Lead
- Core Reviewers
- Contributors



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

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

边缘计算
Edge Computing
CONSORTIUM
THANKS!