



StarlingX Introduction

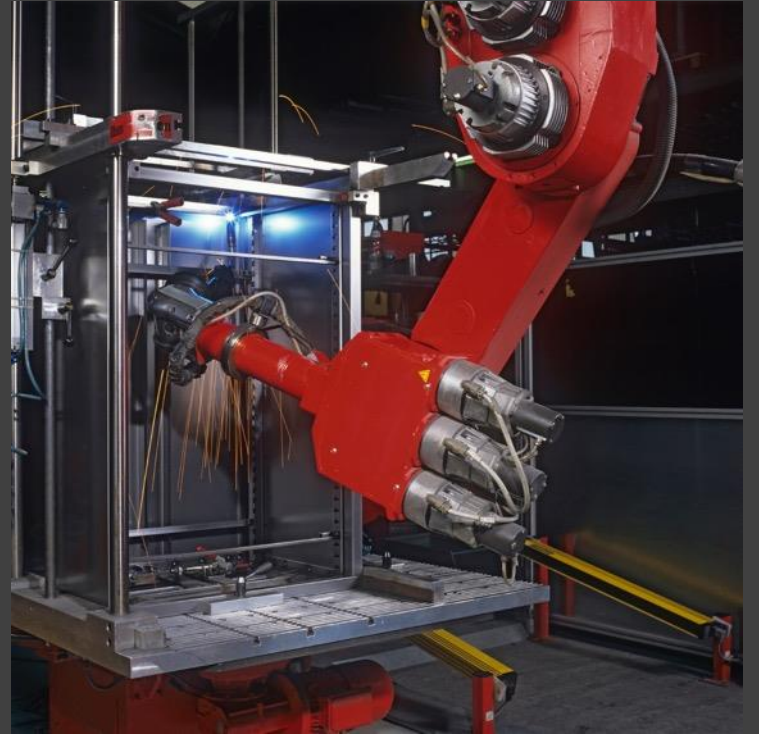
Building community momentum

Ian Jolliffe, Product Architect, Wind River

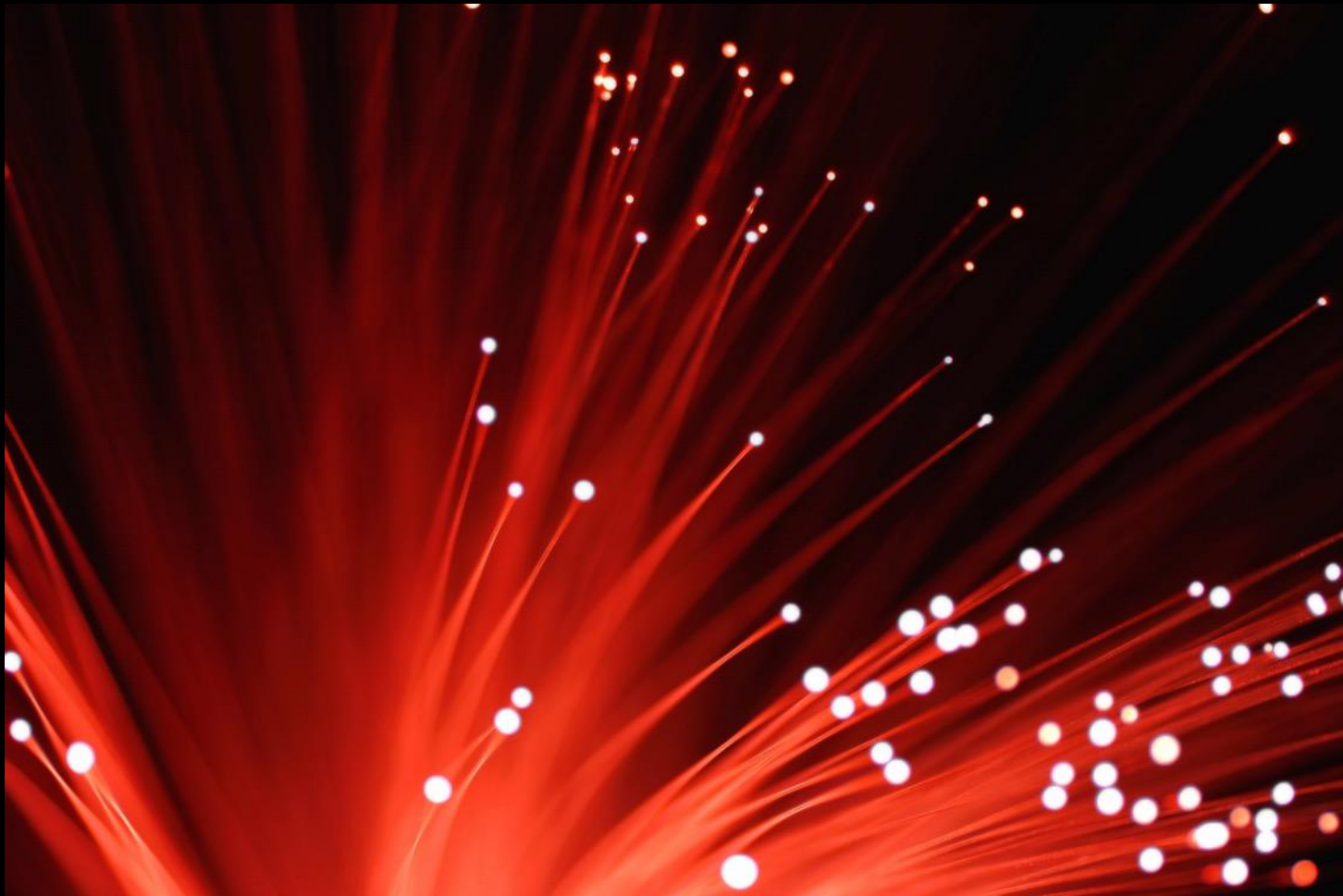
Brent Rowsell, Principal Technologist, Wind River

We believe in the power of Cloud technologies to transform the safety and security of critical infrastructure









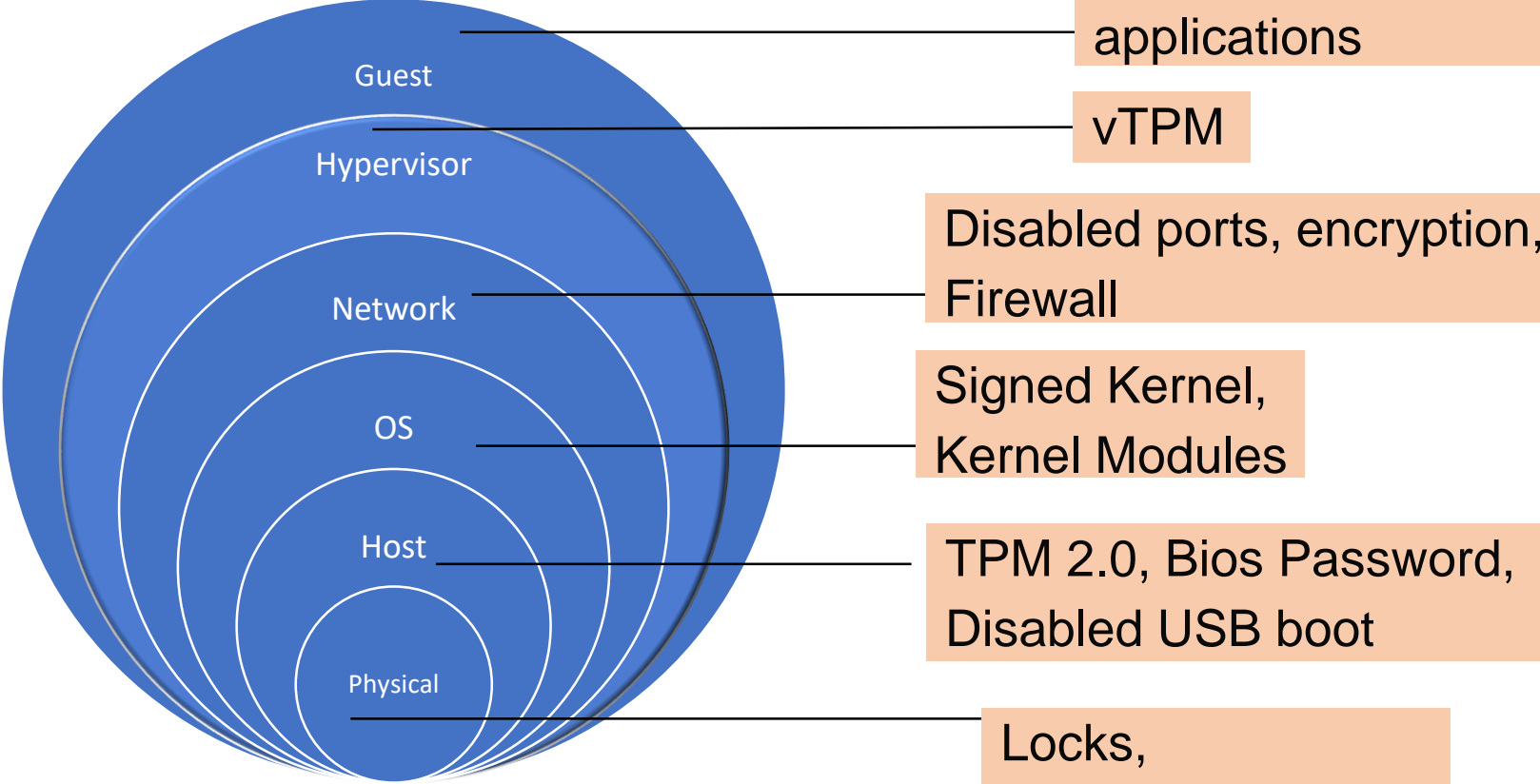
Introducing StarlingX

- StarlingX Launched in May at the Vancouver OpenStack summit
- Formed with seed code from the Wind River Titanium Cloud portfolio
- Provides an integrated platform with focus on high availability, quality of service, performance, security and low latency needed at the edge
- K8s and OpenStack are first class citizens
- Aligned with the OpenStack Foundation Edge Working Group

Where are we now?

- Early days - Forming and Norming
- 100 active developers
- Working build recipe
- Repos hosted by openstack foundation <https://git.openstack.org/cgit/openstack/stx-xxxxx>
- Starting to attract new developers
- New features being added

Security at the Edge



Propose Security Break Out at Next Summit

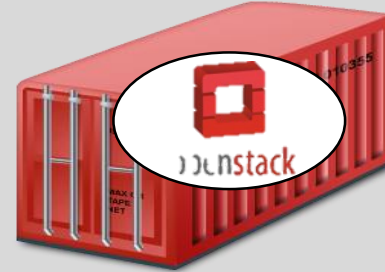
Control Plane

StarlingX Services

HA, Config, Mtce, EPA, Security, Patching



kubernetes



StarlingX / Linux

Storage



StarlingX / Linux

Compute



StarlingX / Linux

Compute



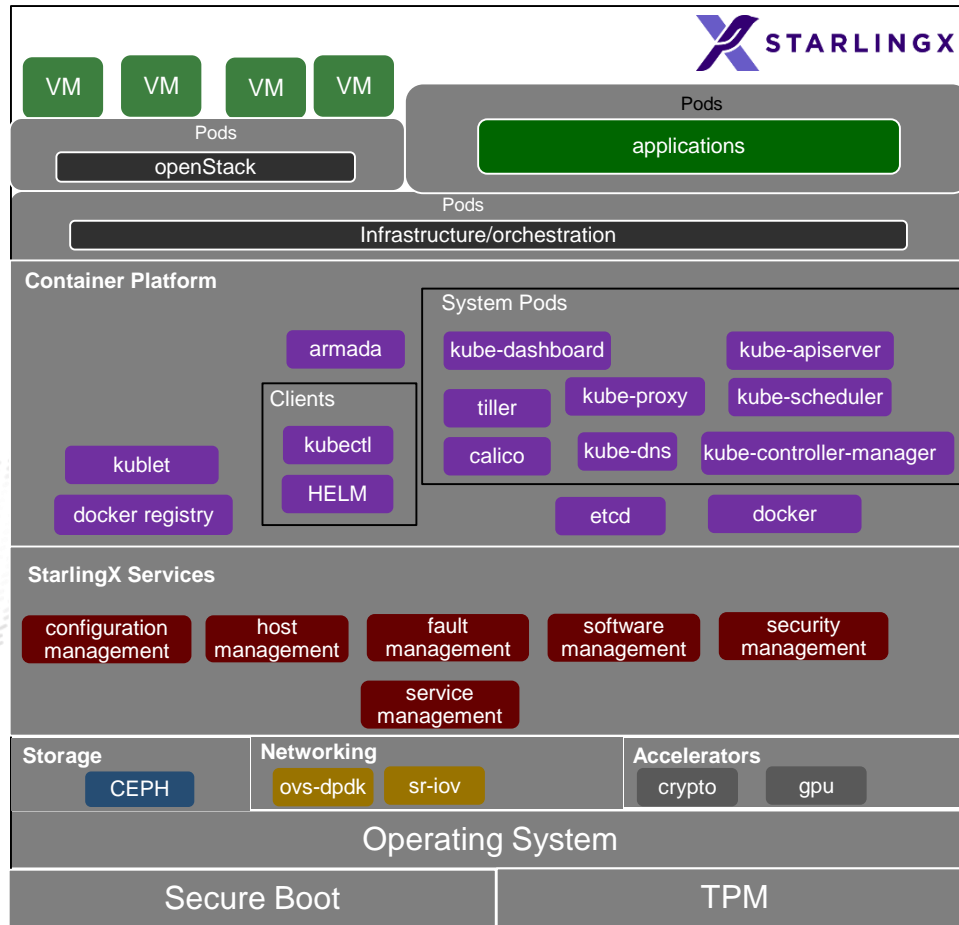
StarlingX / Linux

Network

CNI / OVS

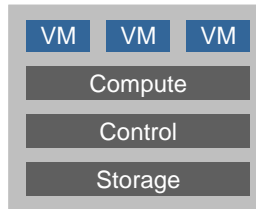


StarlingX / Linux

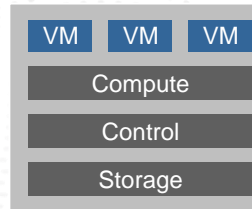


Scalable Solutions for Edge Use Cases

Grouse
Single server

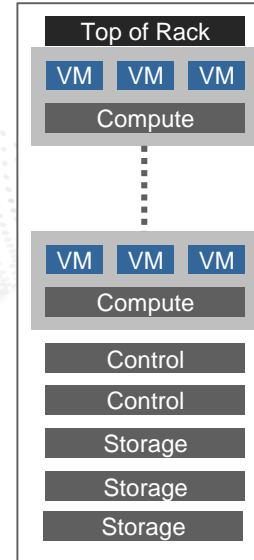


Whistler
Two servers



1:1 protected pair
of servers

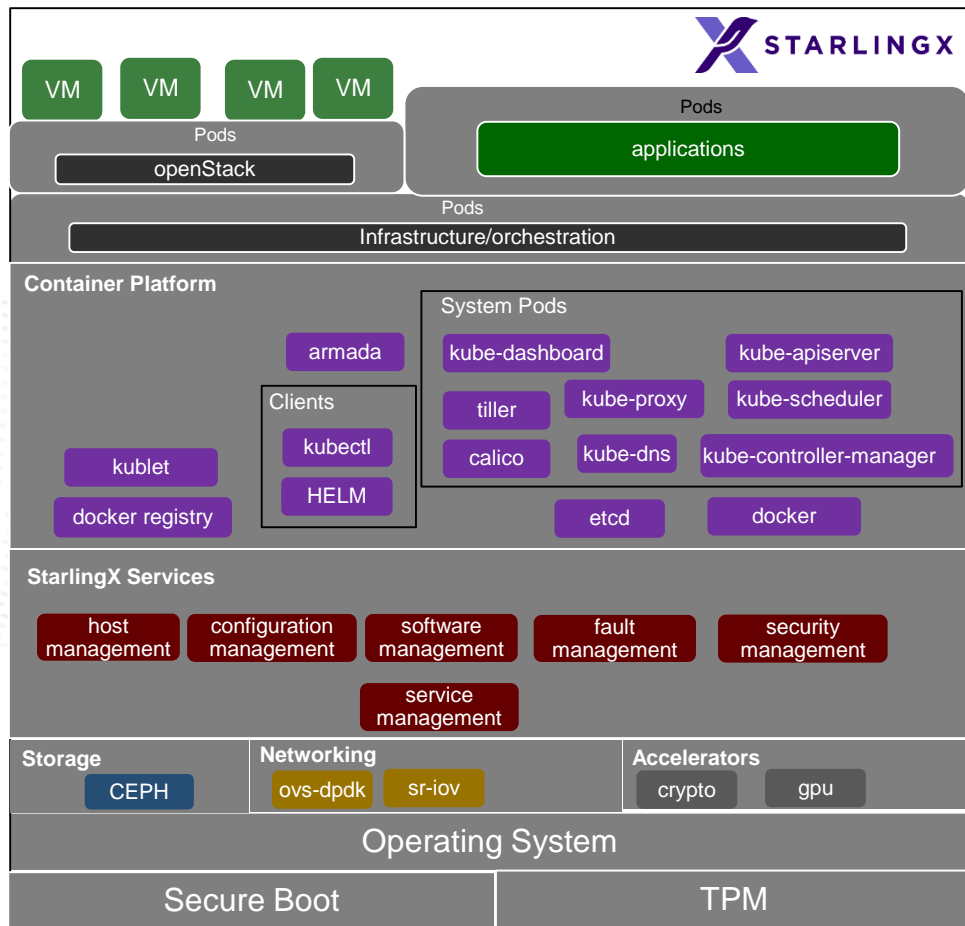
Robson
4–100 servers



Flexible Deployment Models for the Edge

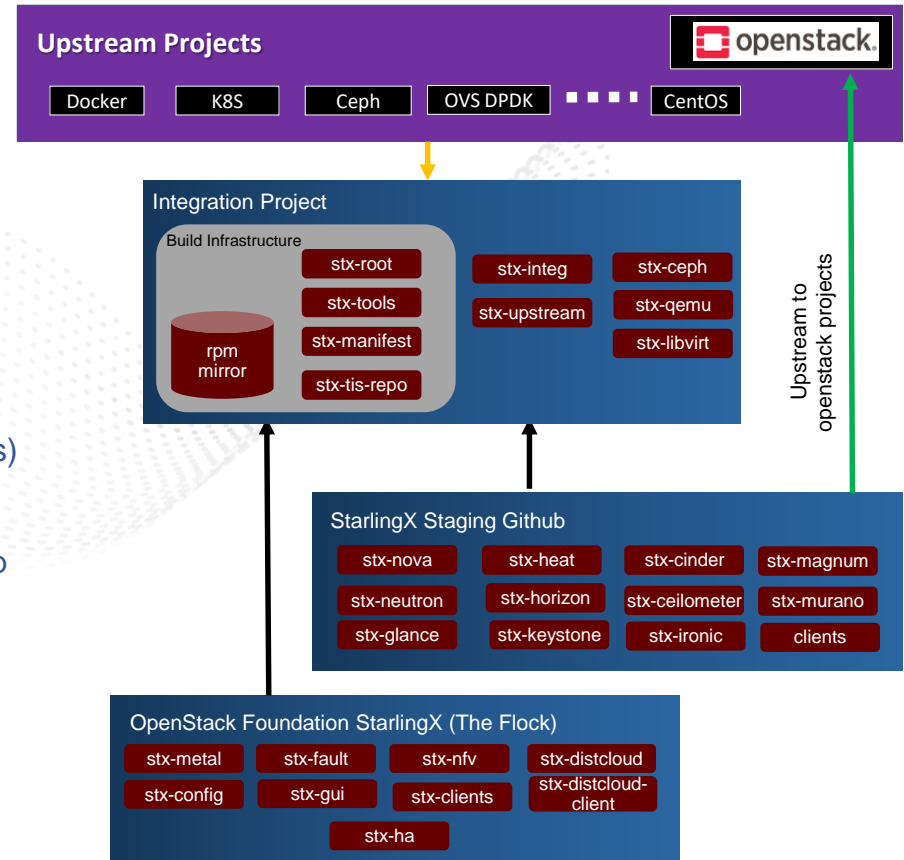
Software Stack

- OS is based on Centos with multiple kernel configurations for performance and security
 - Extensions/fixes to open source packages, upstream to applicable projects
- CEPH as the default storage backend
- OVS-DPDK and SR-IOV support for openstack networking
- Support for crypto and gpu accelerators
- StarlingX Services for system deployment and management
- Container platform - k8s with support for HELM and armada
- Containerized infrastructure orchestration
- Containerized openstack based on Pike with a significant number of value added extensions for hardening, new features and performance enablement
 - Upstreaming in progress– building momentum
- K8s cluster available for end user applications (control plane apps)



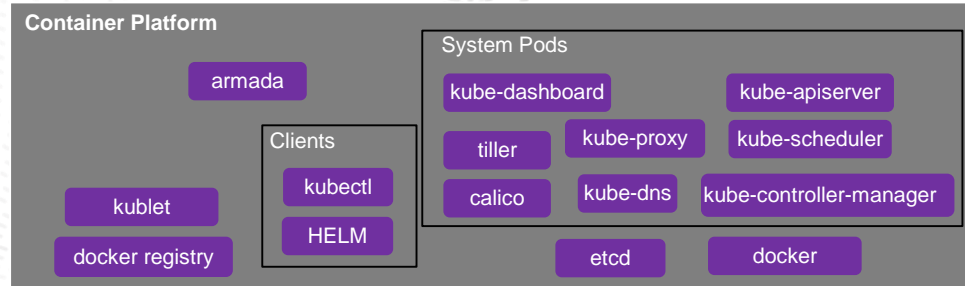
Project Structure

- Openstack Foundation StarlingX – “The Flock”
 - stx-metal - Bare Metal Management
 - stx-config - Configuration Management
 - stx-ha - High Availability/Service Mgmt
 - stx-nfv - NFVI Orchestration
 - stx-fault - Fault Management
 - stx-update - Installation/Update/Patching
 - stx-clients - Client libraries
 - stx-gui - Horizon plugins for StarlingX services
 - stx-distcloud – Distributed Cloud (in progress)
 - stx-distcloud-client – Distributed Cloud Client (in progress)
- StarlingX Staging Github
 - Holding repo for StarlingX openstack extensions being upstreamed
- Integration Project
 - Pulls in components required to build edge cloud infrastructure distributions
 - Build infrastructure and tooling
 - Distro build, patches, packaging
 - Working to drive patches to zero over time



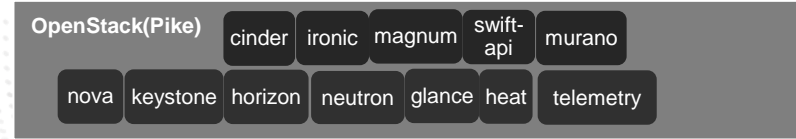
Container Platform

- High availability k8s
- Calico CNI plugin
- CEPH as persistent storage backend
- Authentication/authorization of Kubernetes API
- K8s master components runs on StarlingX controller nodes
- Docker runtime
- Local docker image registry
 - HELM as the package manager
 - Armada for managing multiple Helm charts (ex. openstack)



Openstack Support

- Optional set of services currently based on the Pike release
- Openstack services are containerized
- Deployment and life cycle managed by StarlingX leveraging k8s/helm/armada
 - Includes openstack dependencies (db, rabbit etc.)
- Significant number of changes for hardening, new features and performance enablement
 - Upstreaming in progress– building momentum



Working with Akraino

- Far Far Edge problem
- Enable 5G use cases at the Edge vRAN
- Work with EdgeX and NEV SDK
- Blueprint proposal ready
- Looking for community feedback

