StarlingX Project Overview

BRUCE E. JONES, INTEL

IAN JOLLIFFE, WIND RIVER

GREG WAINES, WIND RIVER

STARLINGX
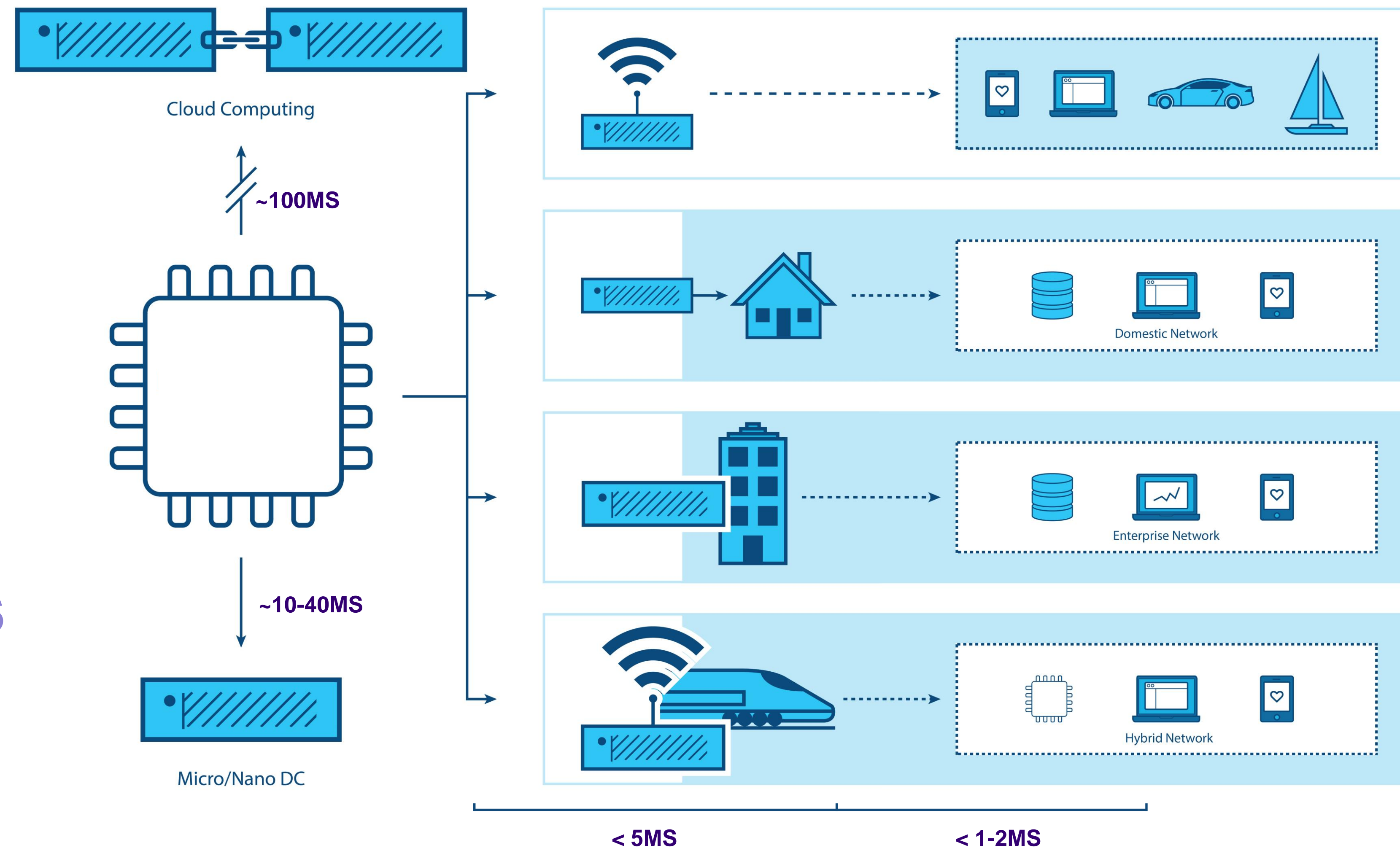
Let Me Introduce StarlingX

- 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, with the second release planned for May/2019,
- Growing community
  - Inviting users, operators and developers to try out the software and participate in the community
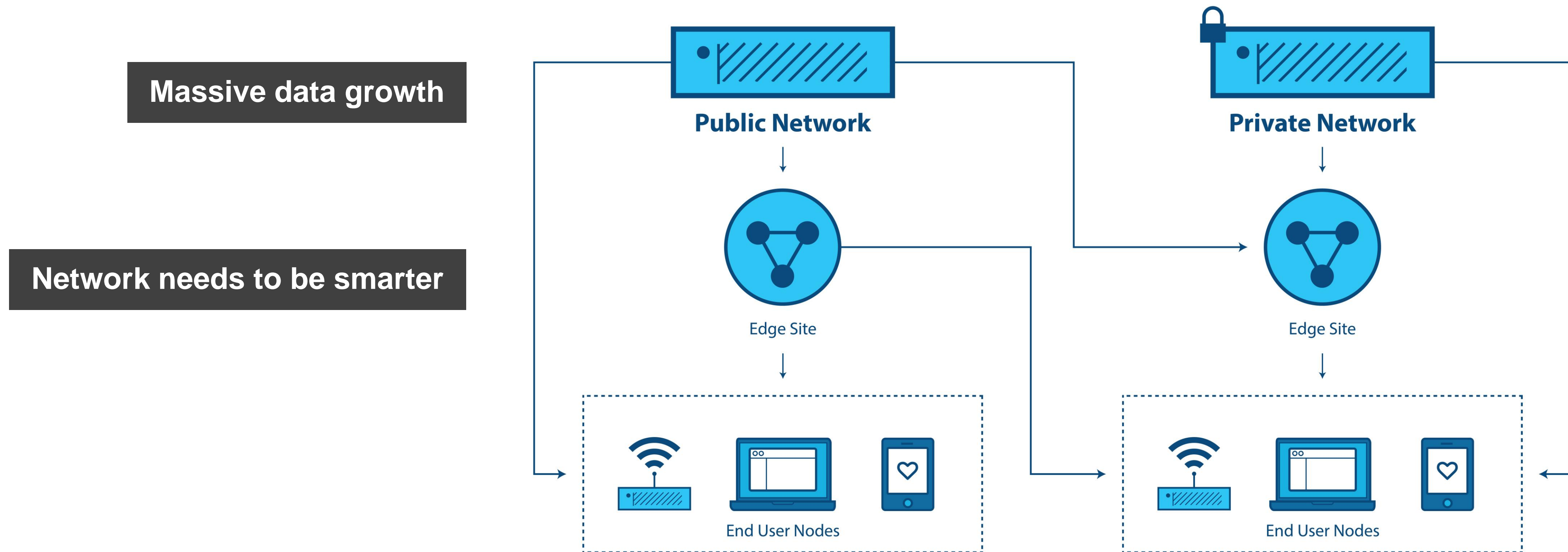
Project Overview

What Problems Is StarlingX Solving?

Massive data growth

Network needs to be smarter

Public Network

Private Network

Edge Site

Edge Site

End User Nodes

End User Nodes

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

Goals of the StarlingX Project

Re-Configure Proven Cloud Technologies for Edge Compute

- Orchestrate system-wide for bare metal, VMs and Container workloads
  - Deploy and manage Edge clouds, share configurations
- Simplify deployment to geographically dispersed, remote Edge regions

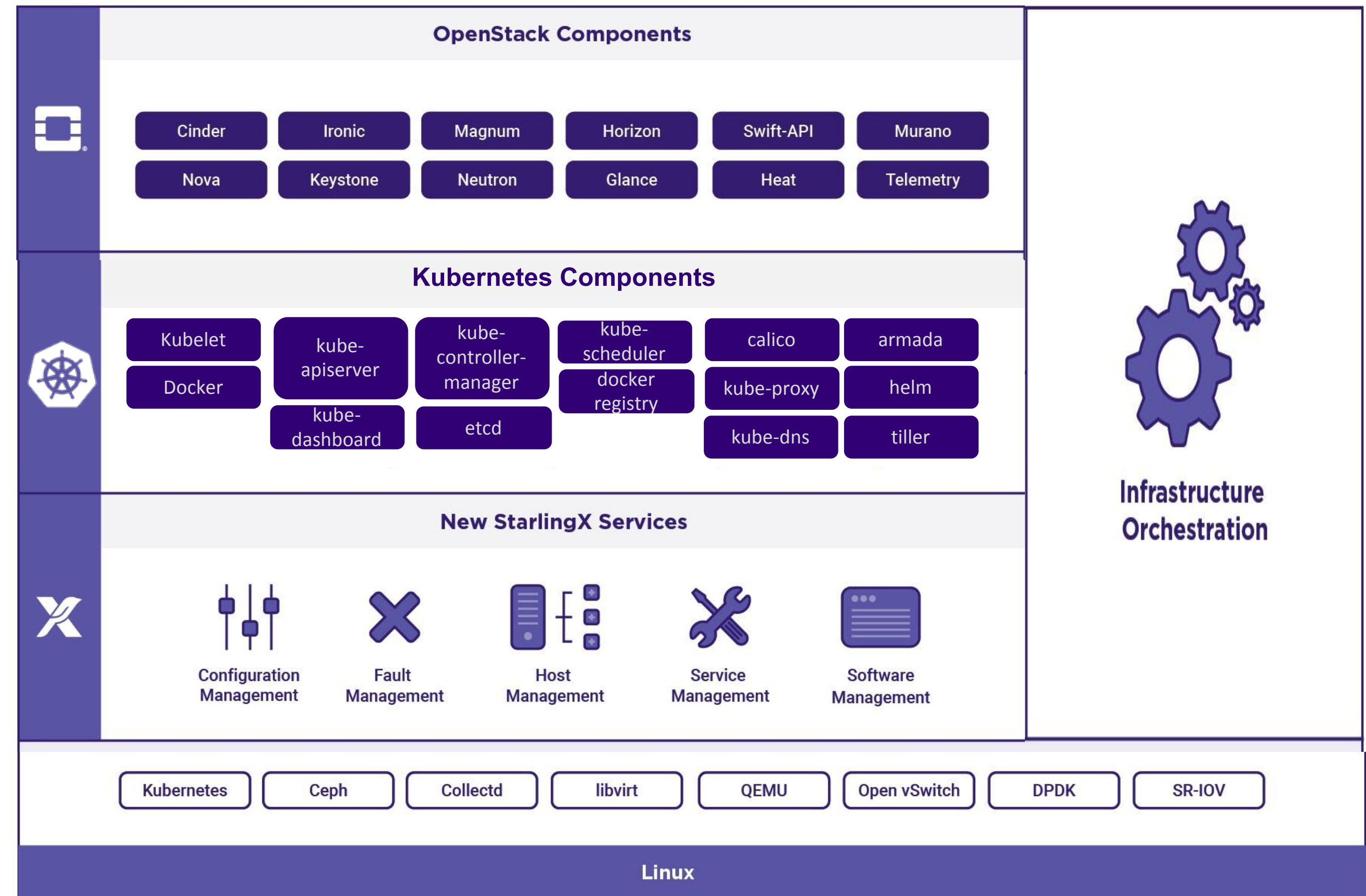| Transportation	Energy
Manufacturing	Retail
Video	Smart cities
Healthcare	Drones
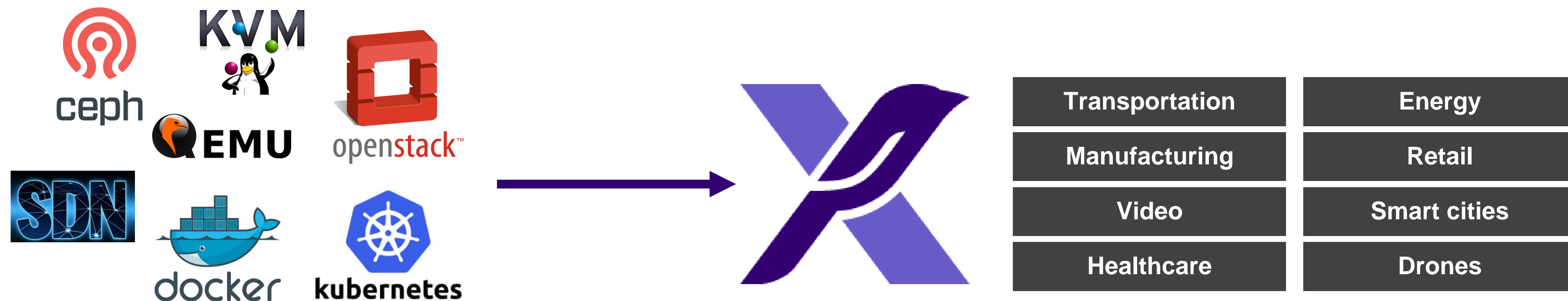
StarlingX Technology

StarlingX - Edge Virtualization Platform

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

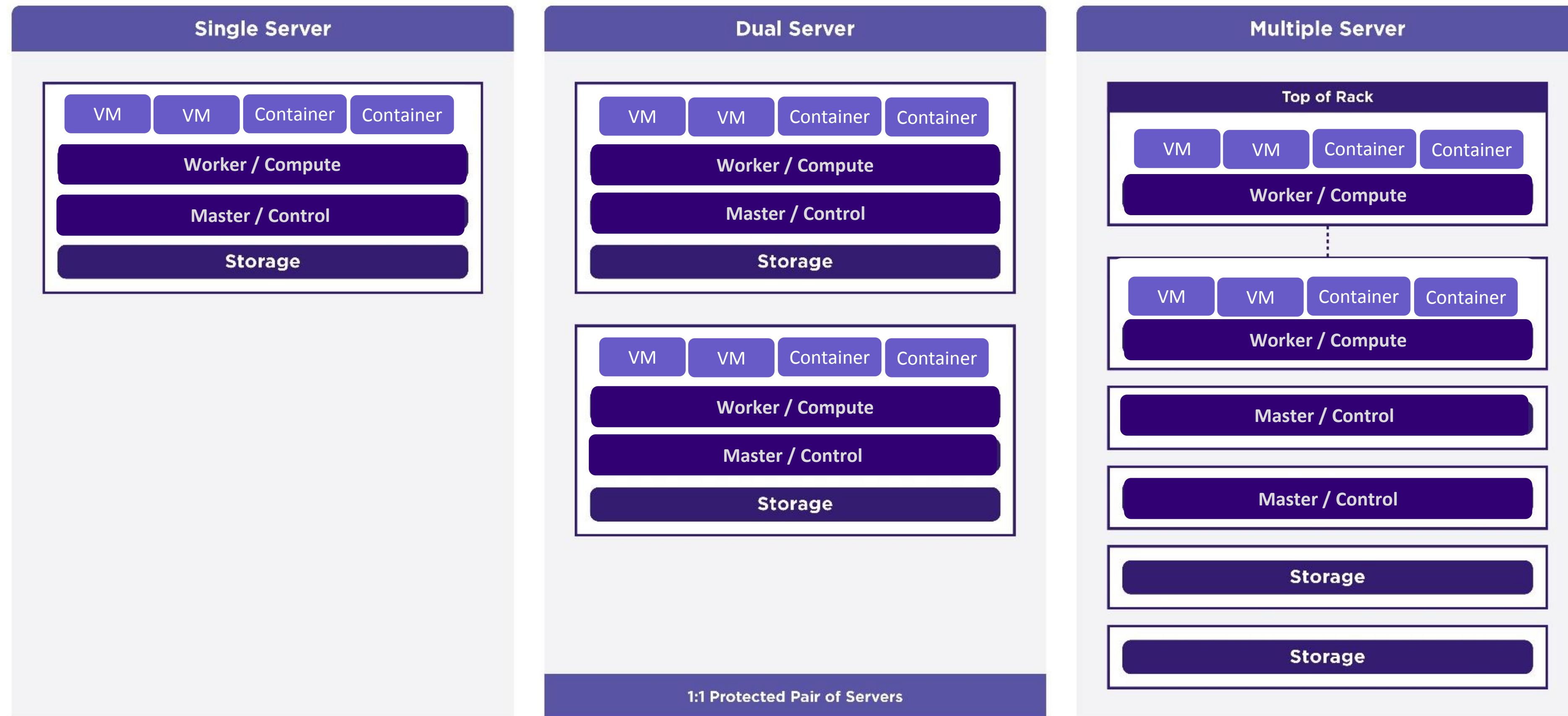Services from the StarlingX virtualization platform focus on
- **Easy deployment**
- **Low touch manageability**
- **Rapid response to events**

- **Fast recovery**

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


OpenStack Components

Cinder	Ironic	Magnum	Horizon	Swift-API	Murano
Nova	Keystone	Neutron	Glance	Heat	Telemetry

Kubernetes Components

Kubelet, kube-apiserver, kube-controller-manager, kube-scheduler, calico, armada, Docker, docker registry, kube-proxy, helm, kube-dashboard, etcd, kube-dns, tiller

New StarlingX Services

Configuration Management, Fault Management, Host Management, Service Management, Software Management

Infrastructure Orchestration

Kubernetes, Ceph, Collectd, libvirt, QEMU, Open vSwitch, DPDK, SR-IOV

Linux

Scalability from Small to Large

- ## Single Server
  - Runs all functions

- ## Dual Server
  - Redundant design

- ## Multiple Server
  - Fully resilient and geographically distributable

Single Server

| VM | VM | Container | Container |
| Worker / Compute |
| Master / Control |
| Storage |

Dual Server

| VM | VM | Container | Container |
| Worker / Compute |
| Master / Control |
| Storage |

| VM | VM | Container | Container |
| Worker / Compute |
| Master / Control |
| Storage |

1:1 Protected Pair of Servers

Multiple Server

Top of Rack

| VM | VM | Container | Container |
| Worker / Compute |

| VM | VM | Container | Container |
| Worker / Compute |

Master / Control

Master / Control
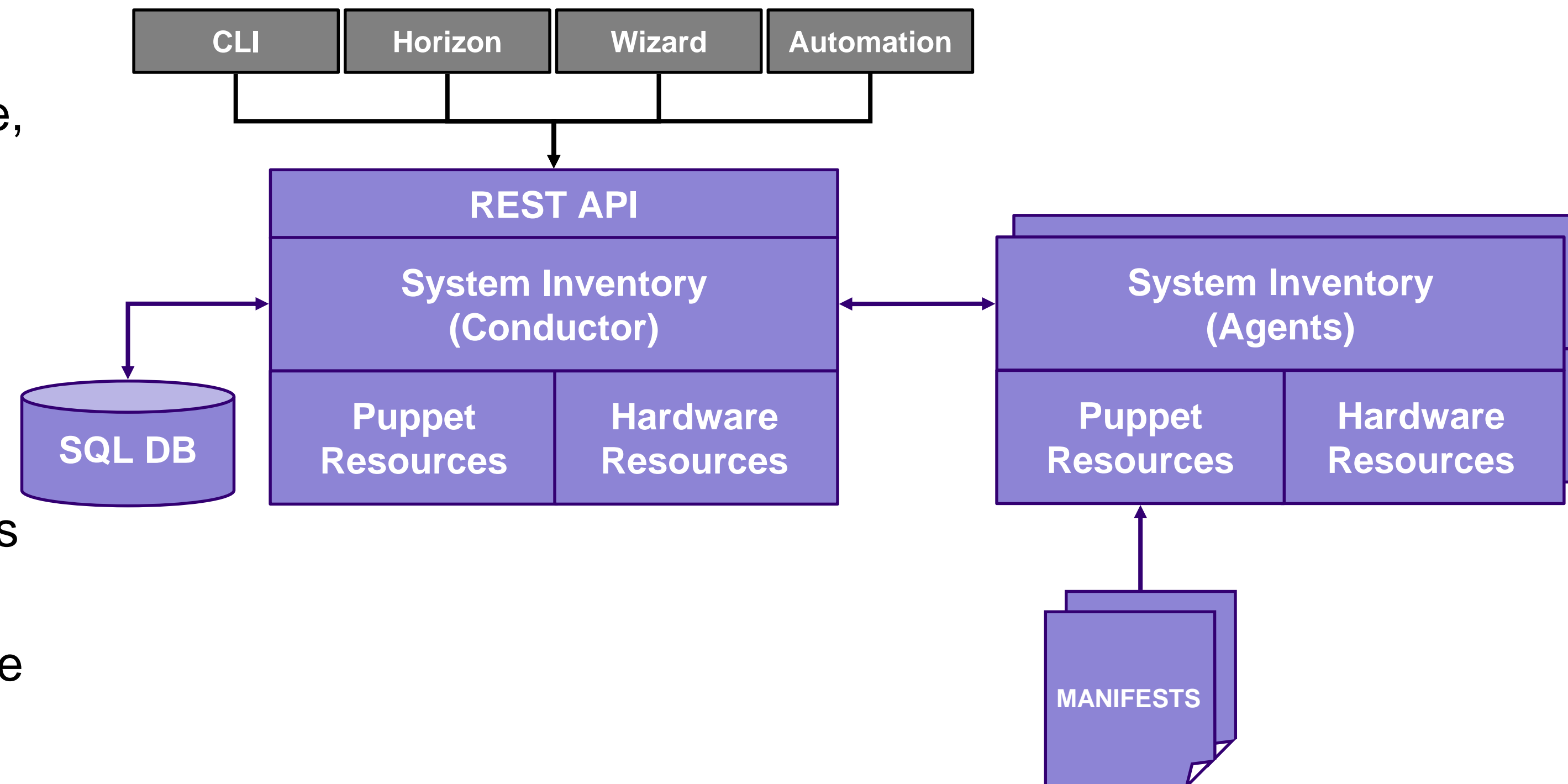
Storage

Storage

Physical Server

Scaling from the Core to the Edge

- Geographically distributed multi-region deployment
- Orchestration and Synchronization Services provided by Centralized Cloud
  - 'Single Pane of Glass' Horizon at Central Cloud provides access to GUI Panels from all clouds,
  - Users synchronized across all clouds,
  - Images synchronized across all clouds,
  - SW Updates can be applied across all clouds from Central Cloud.
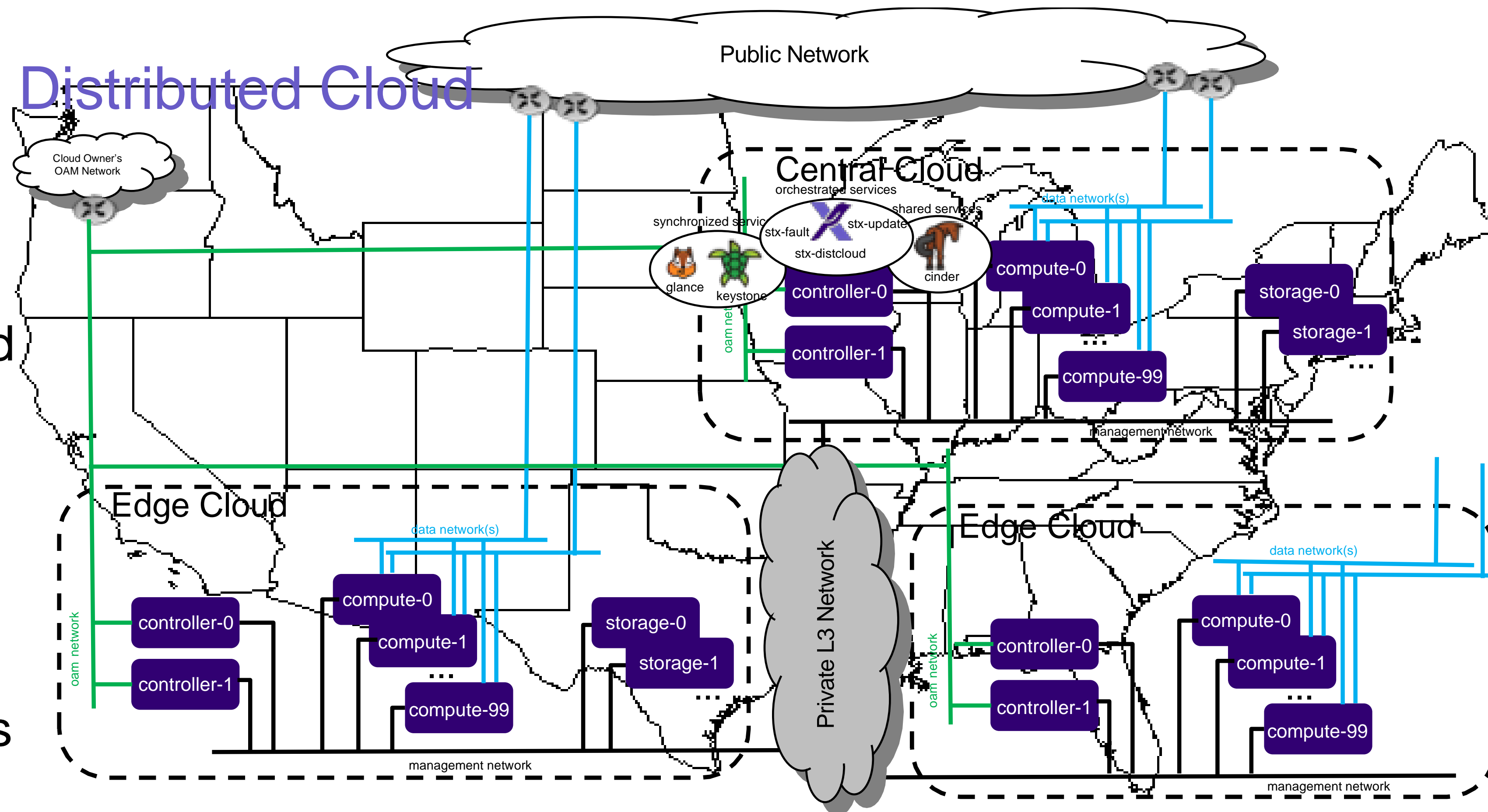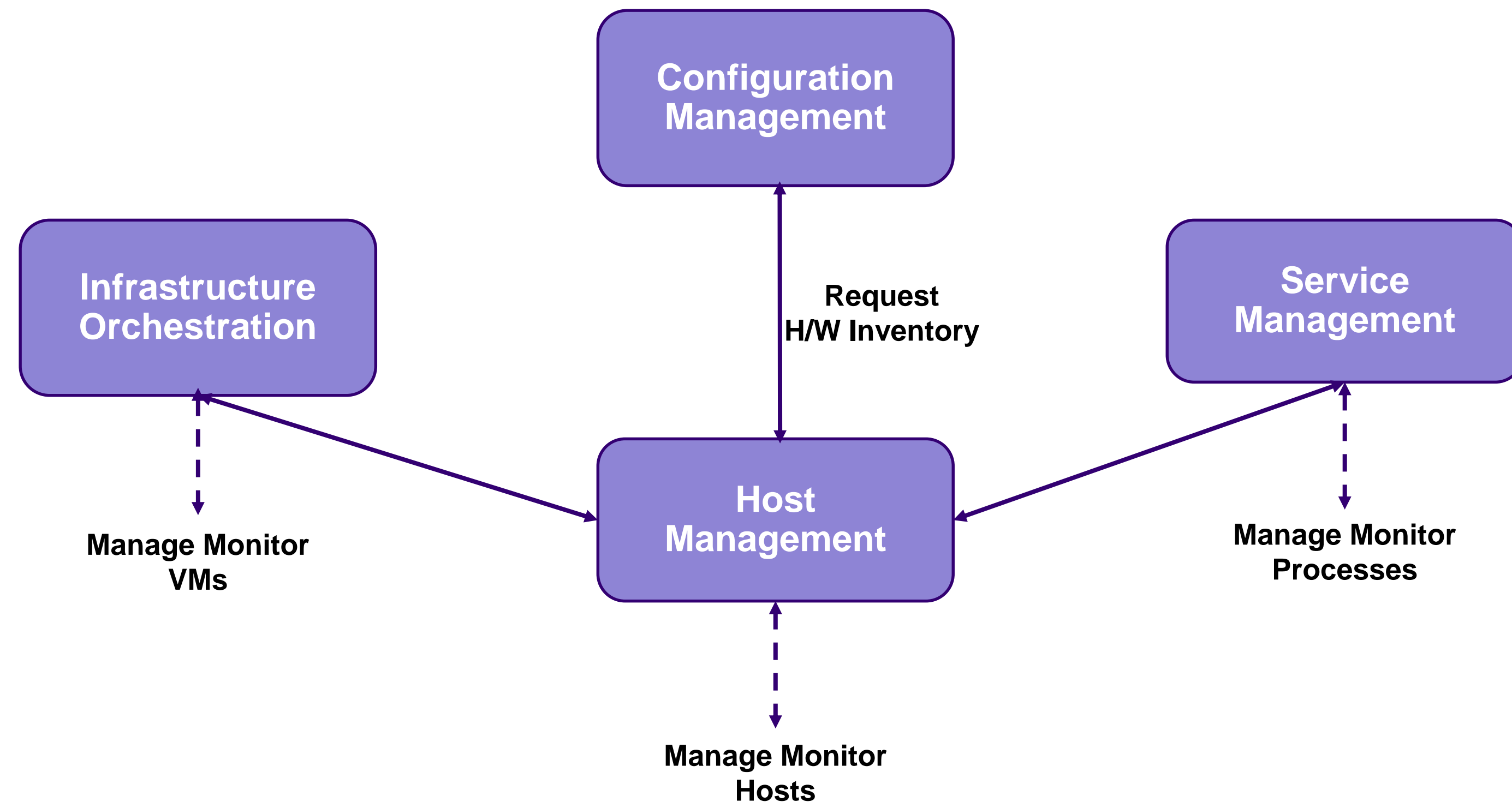
Configuration Management

- 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

CLI	Horizon	Wizard	Automation

REST API

System Inventory (Conductor)

Puppet Resources	**Hardware Resources**

SQL DB

System Inventory (Agents)

Puppet Resources	**Hardware Resources**

MANIFESTS

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 REST API

Configuration Management

Infrastructure Orchestration

Service Management

Request H/W Inventory

Host Management

Manage Monitor VMs

Manage Monitor Processes

Manage Monitor Hosts

**Vendor Neutral Host Management**

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 Kubernetes software
  - New / upgraded OpenStack software

SOURCE CLOUD ON RELEASE N

SOURCE CLOUD ON RELEASE N+1
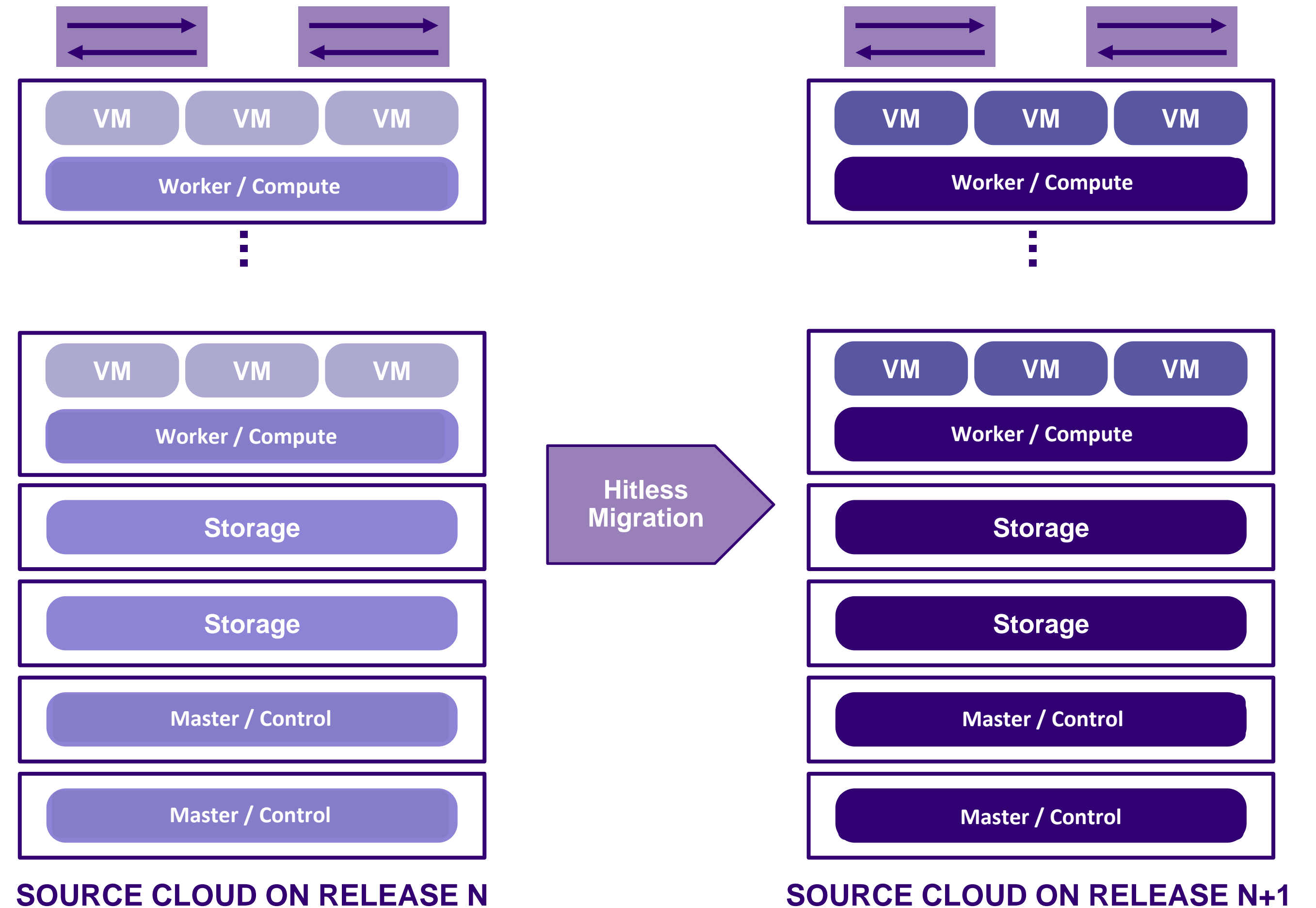
Software Upgrades and Patching

Architecture
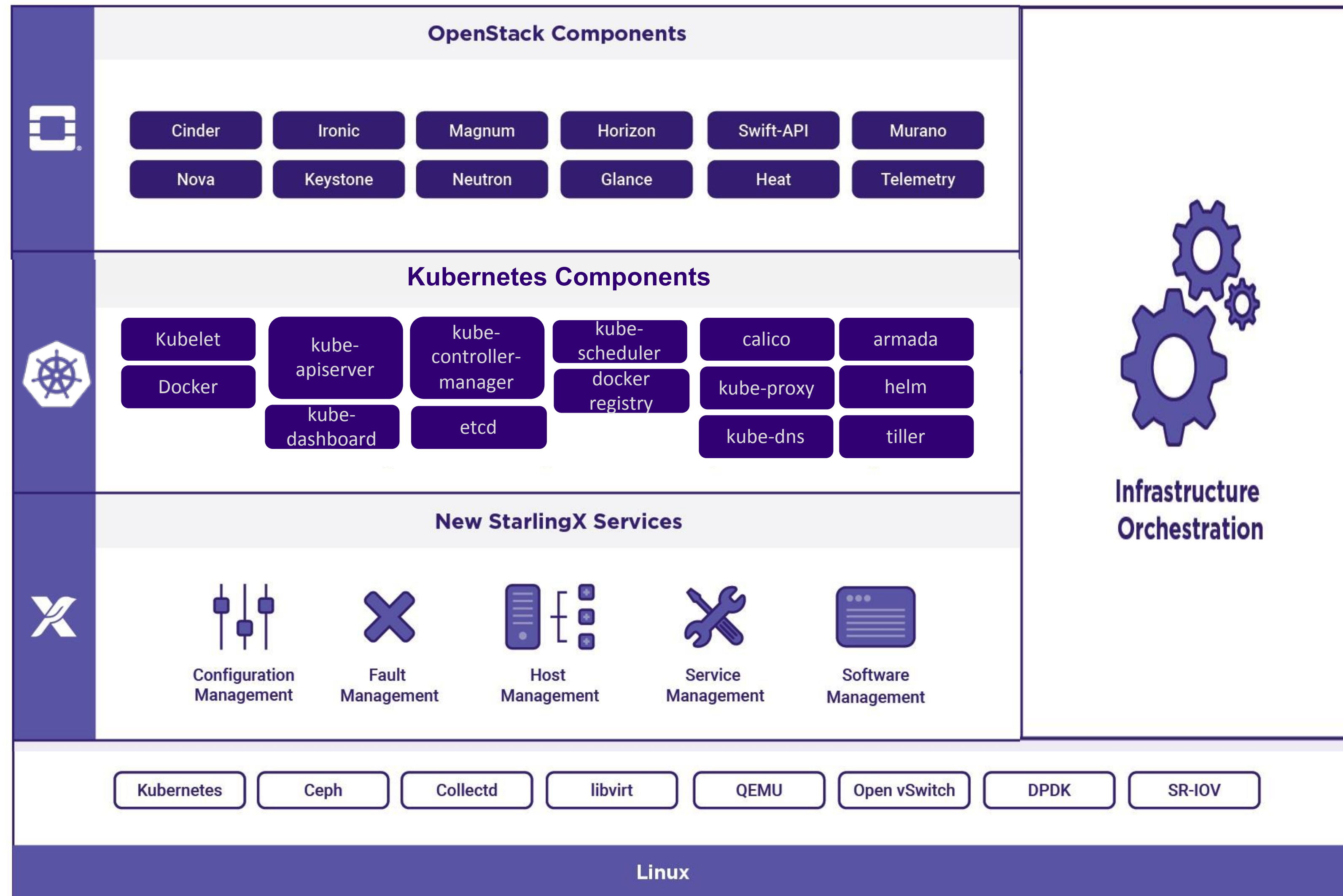
The Road to the Edge

- 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**
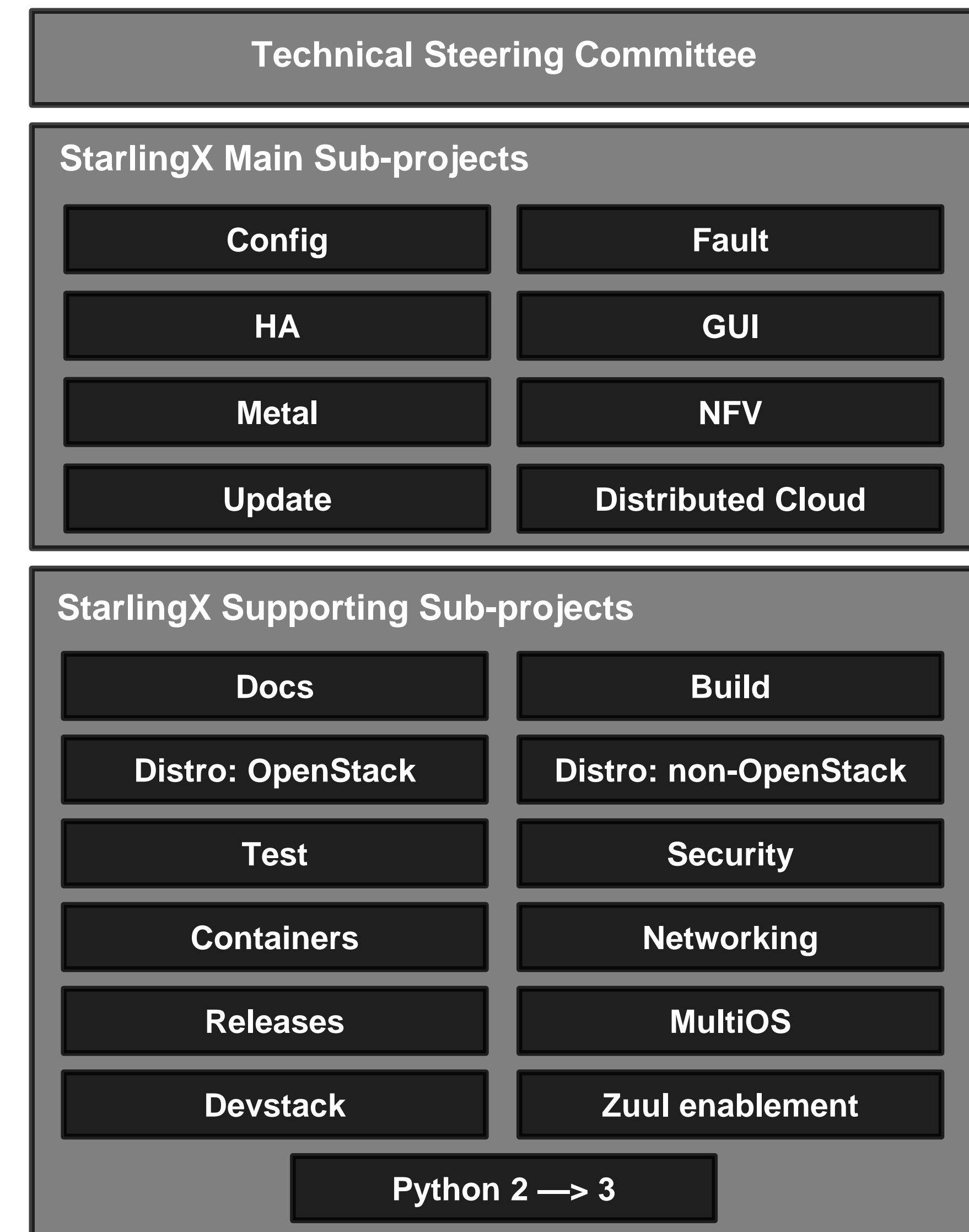 - **Mission-ready** for your applications

Community and Contributing

Principles

- The StarlingX project follows the "four opens,"
  - Open Collaboration
  - Open Design
  - Open Development
  - Open Source
- Technical decisions are 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 9 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), Saul Wold (Intel), Curtis Collicutt (Interdynamics), Ada Cunha (Ericsson), Shuquan Huang (99Cloud) and Miguel Lavalle (Verizon/OATH)

Contributions

- Code and formal documentation are available through git / gerrit
  - git.starlingx.io
- Informal documentation is also on our wiki:
  - https://wiki.openstack.org/wiki/StarlingX
- 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

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)
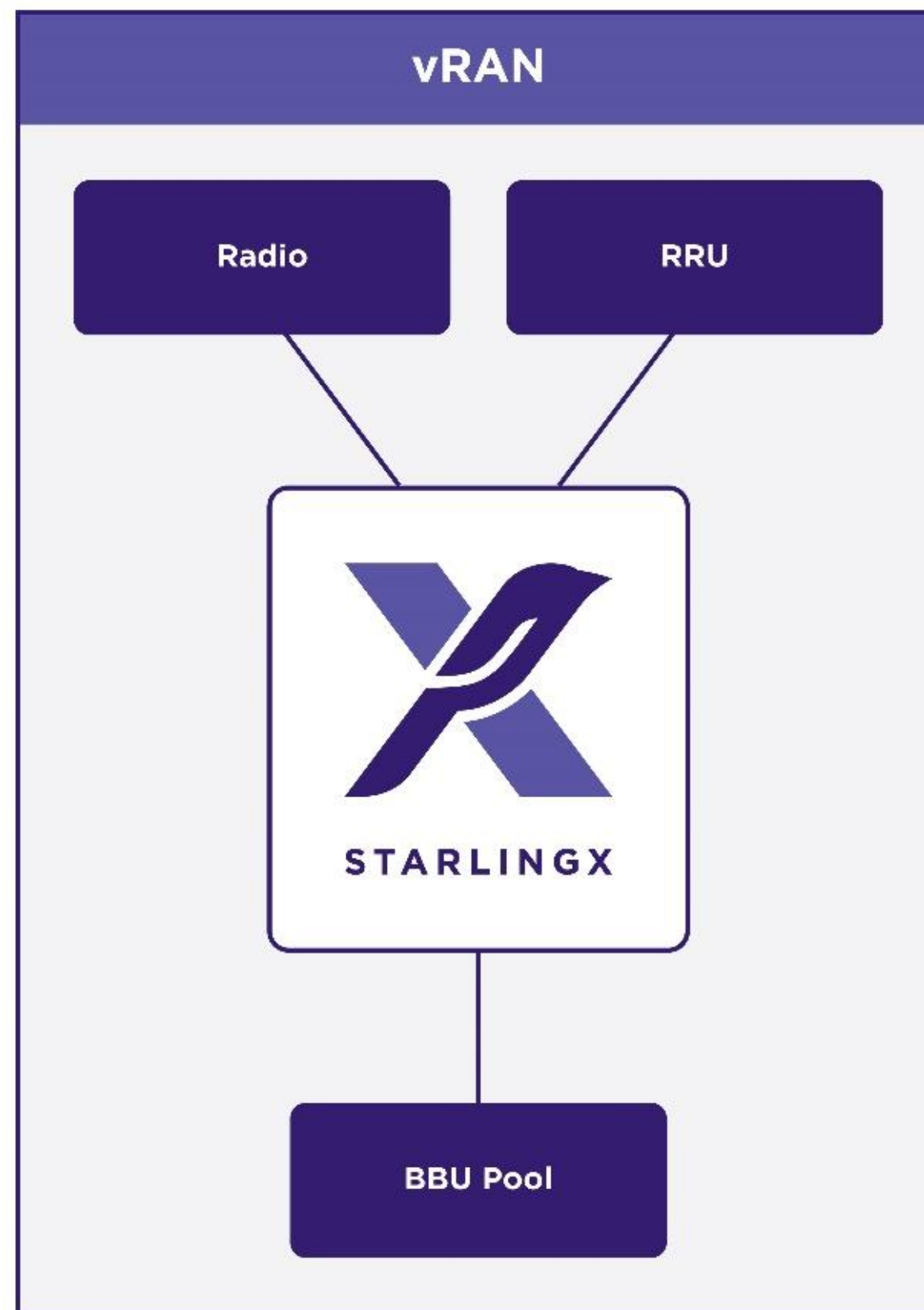
Communication

- #starlingx@Freenode
- Mailing Lists:
  - lists.starlingx.io
- Email:
  - http://lists.starlingx.io/cgi-bin/mailman/listinfo/starlingx-discuss
- Weekly meetings:
  - Zoom calls
  - https://wiki.openstack.org/wiki/Starlingx/Meetings
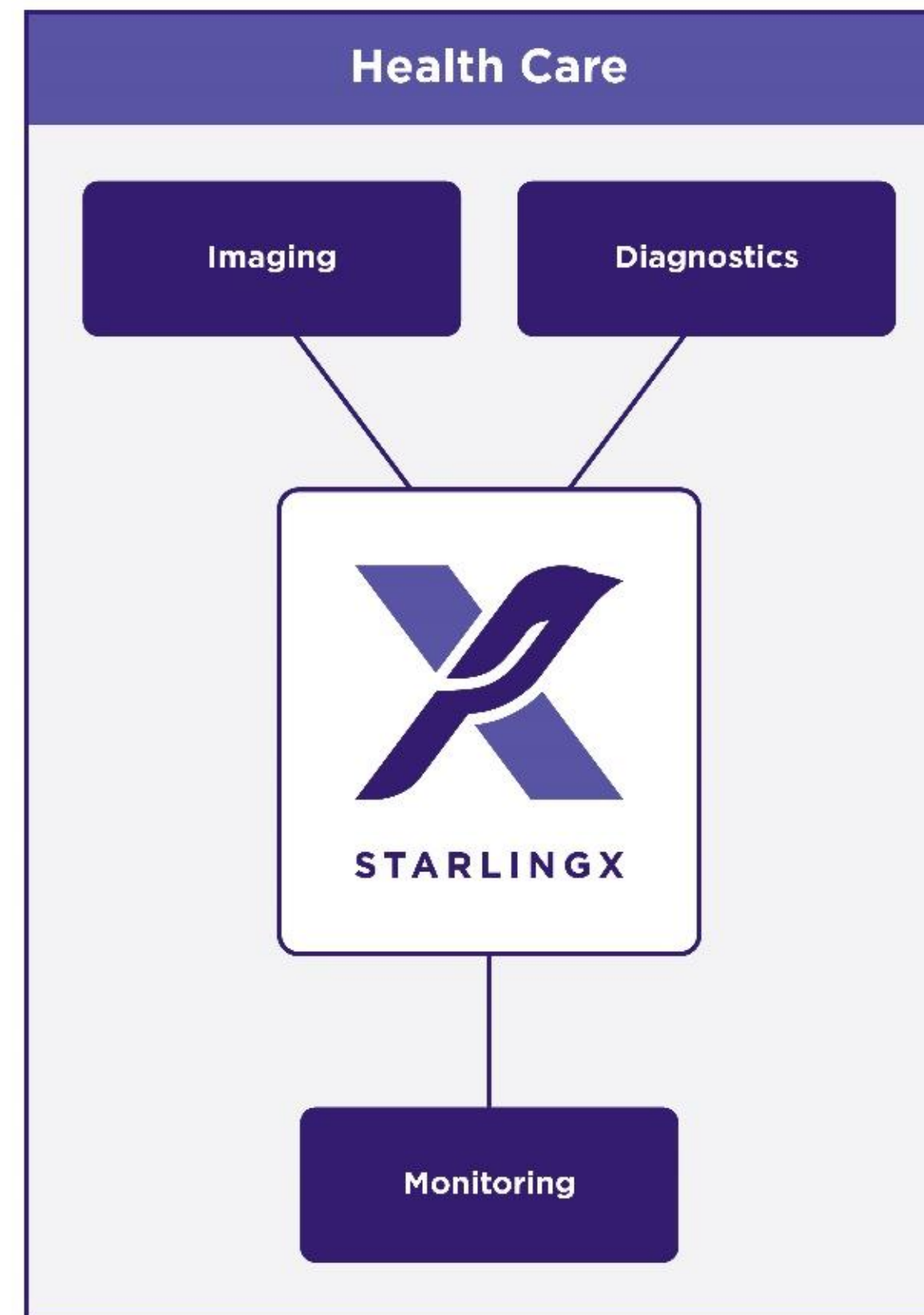
Thank You!
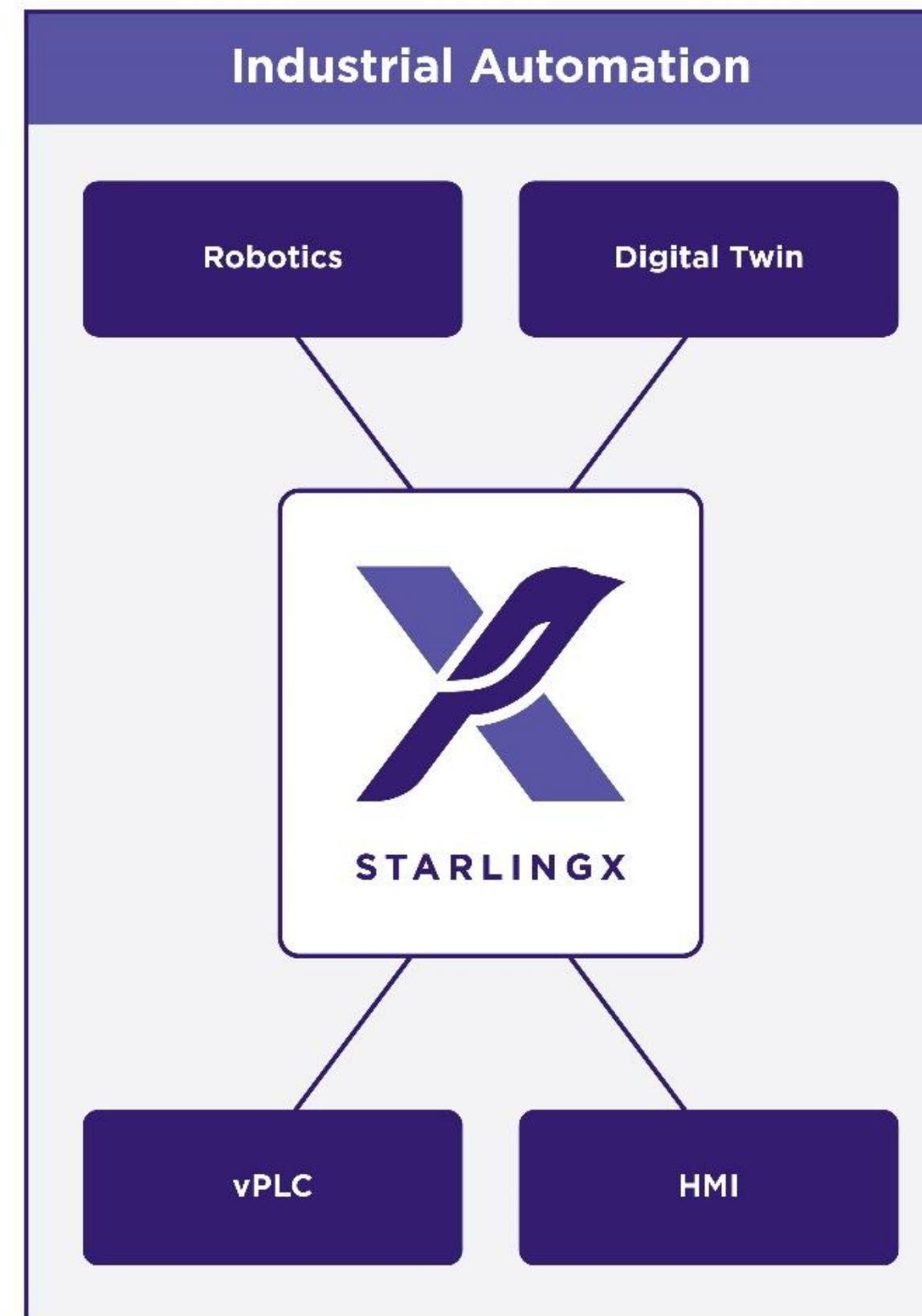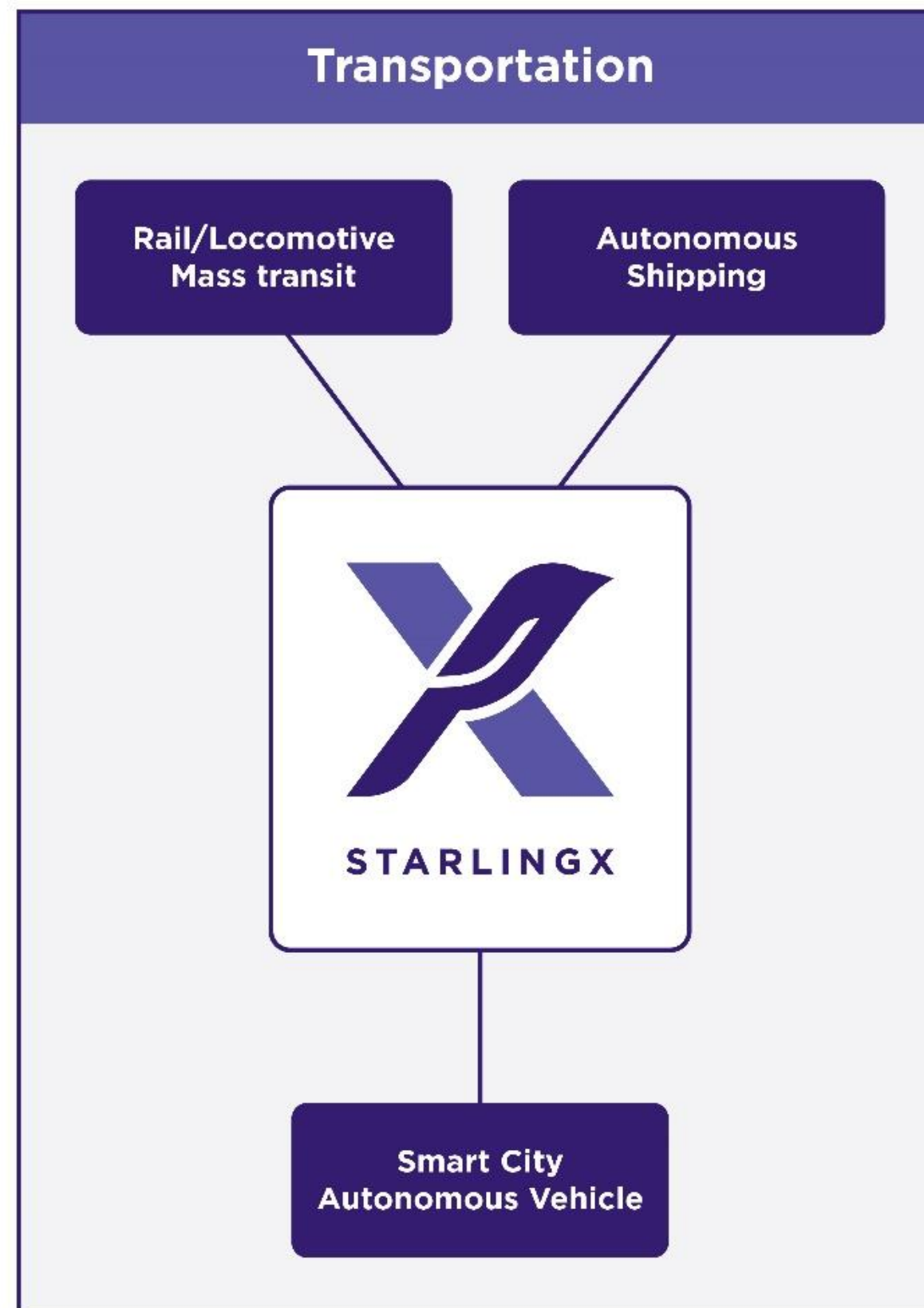
Q&A

Appendix

Edge Computing Use Cases

vRAN == virtual Radio Access Network
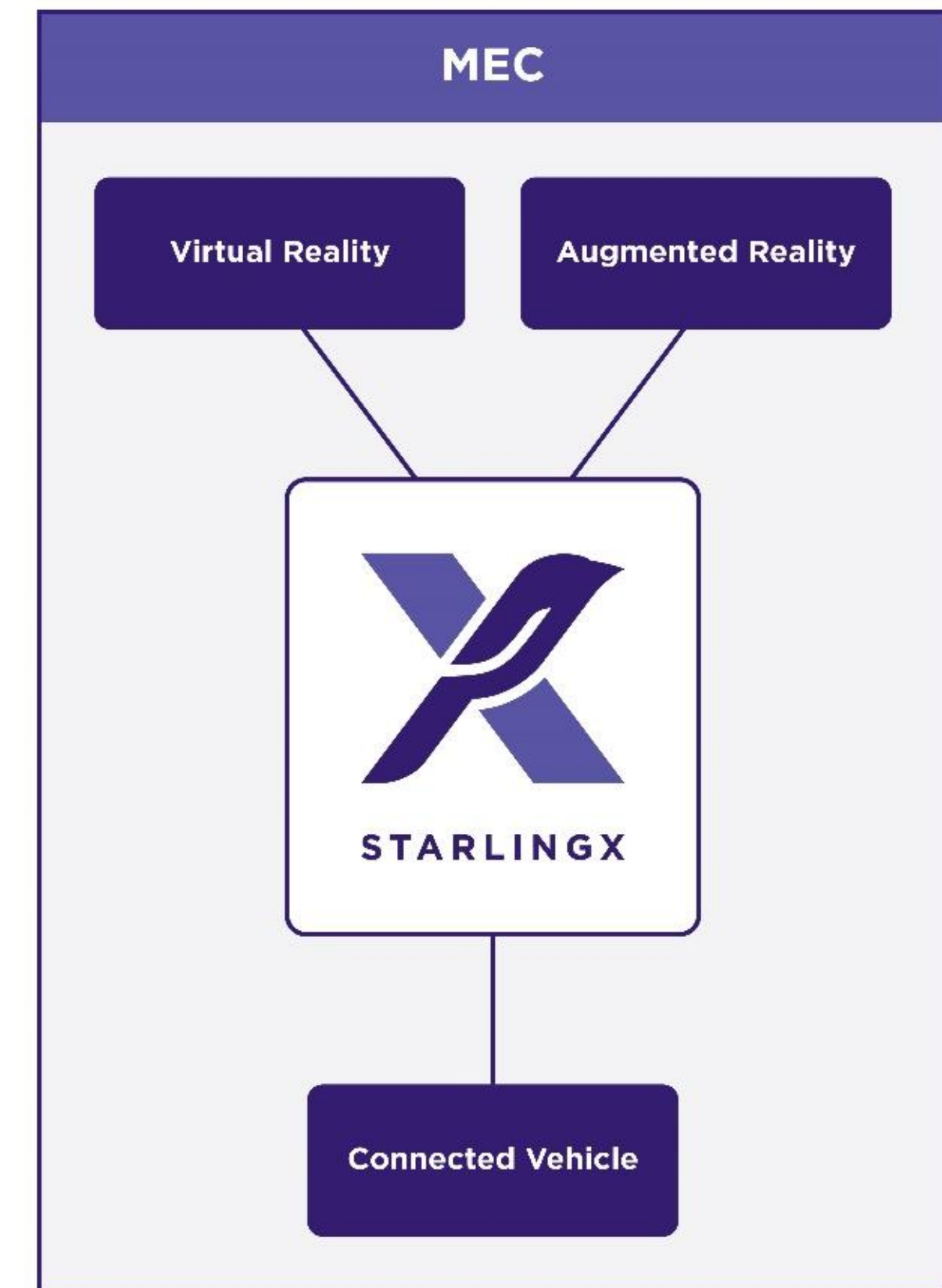
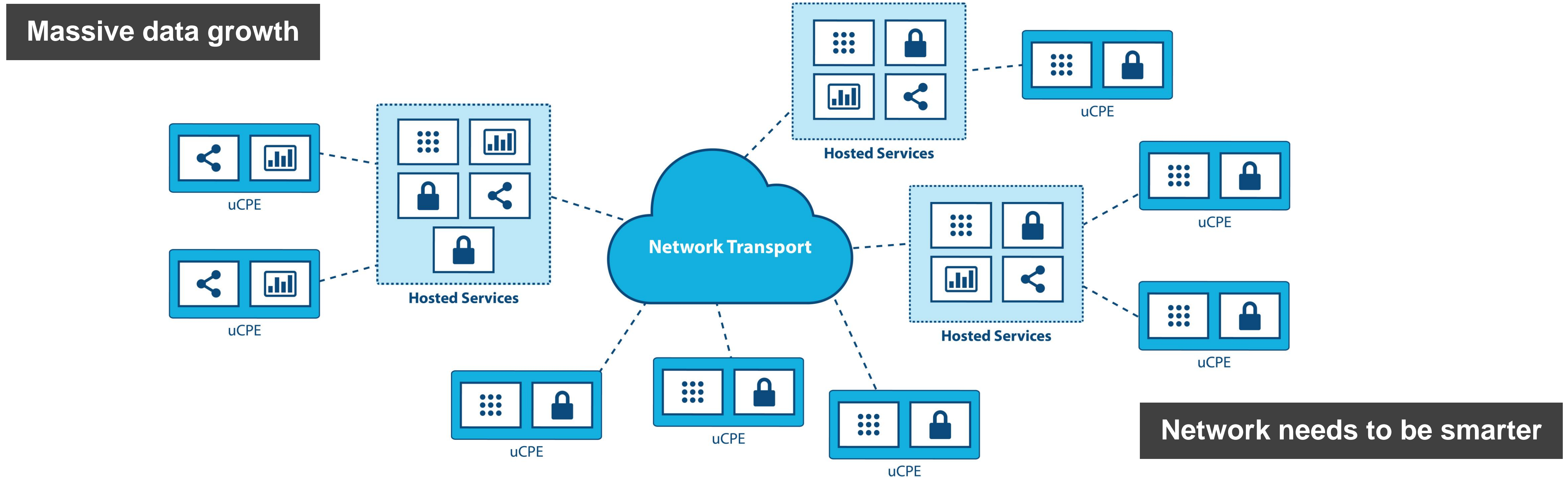MEC == Multi-access Edge Computing

Edge Computing Use Cases
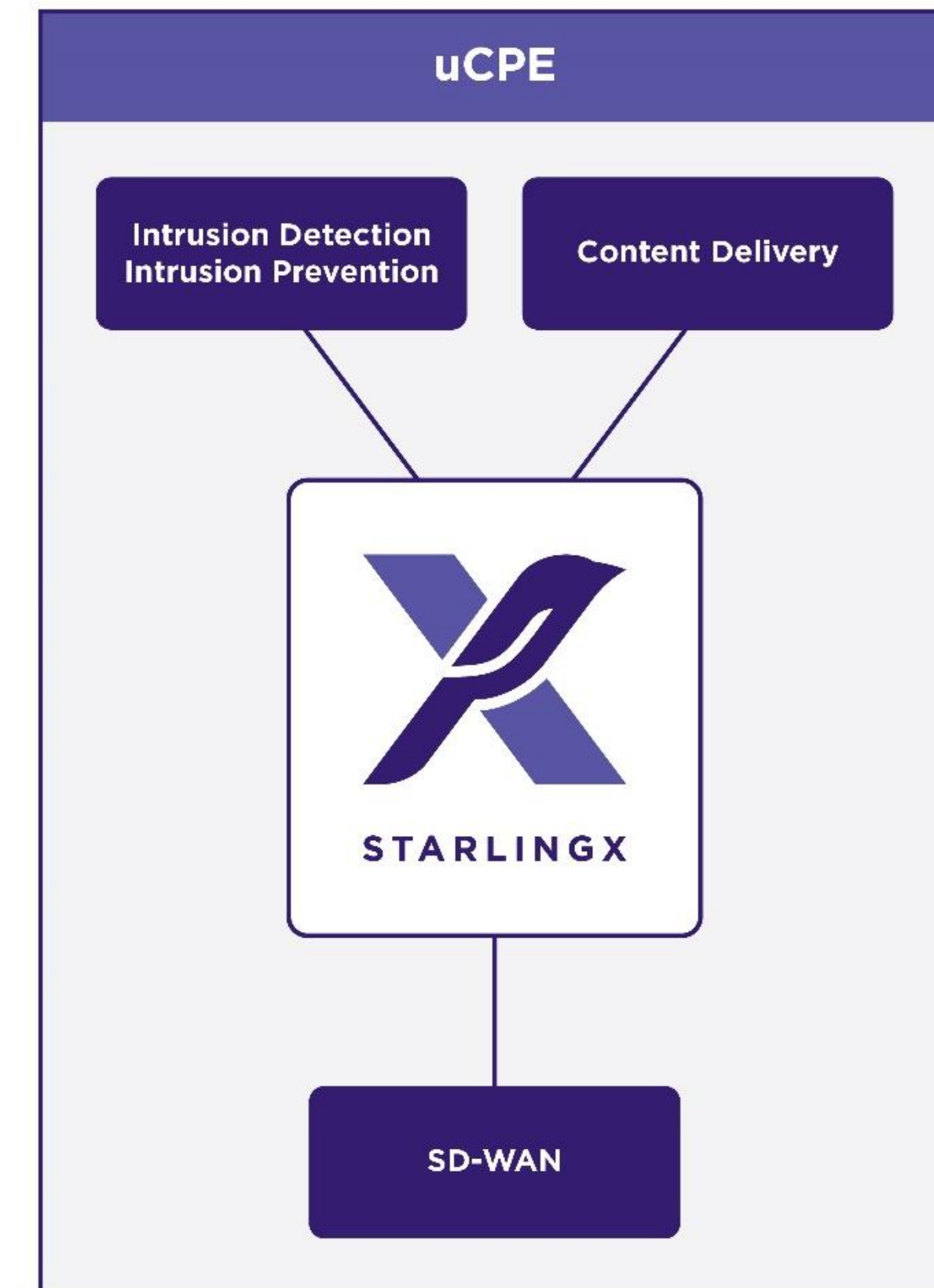
uCPE == universal Customer Premises Equipment

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

Standard Configuration