CHINA OpenInfra Days

IT大咖说 知识共享平台

intel

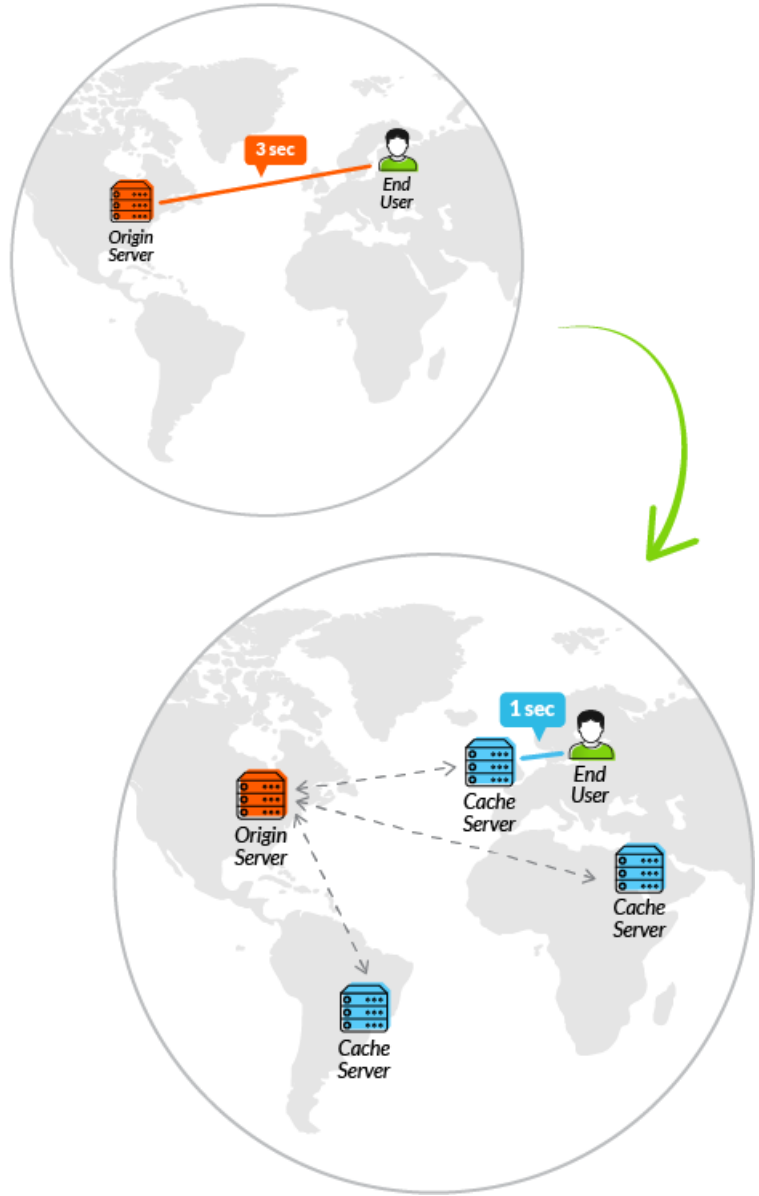
Stor4NFV: Exploration of Cloud-native Storage in OPNFV

NFV Cloud

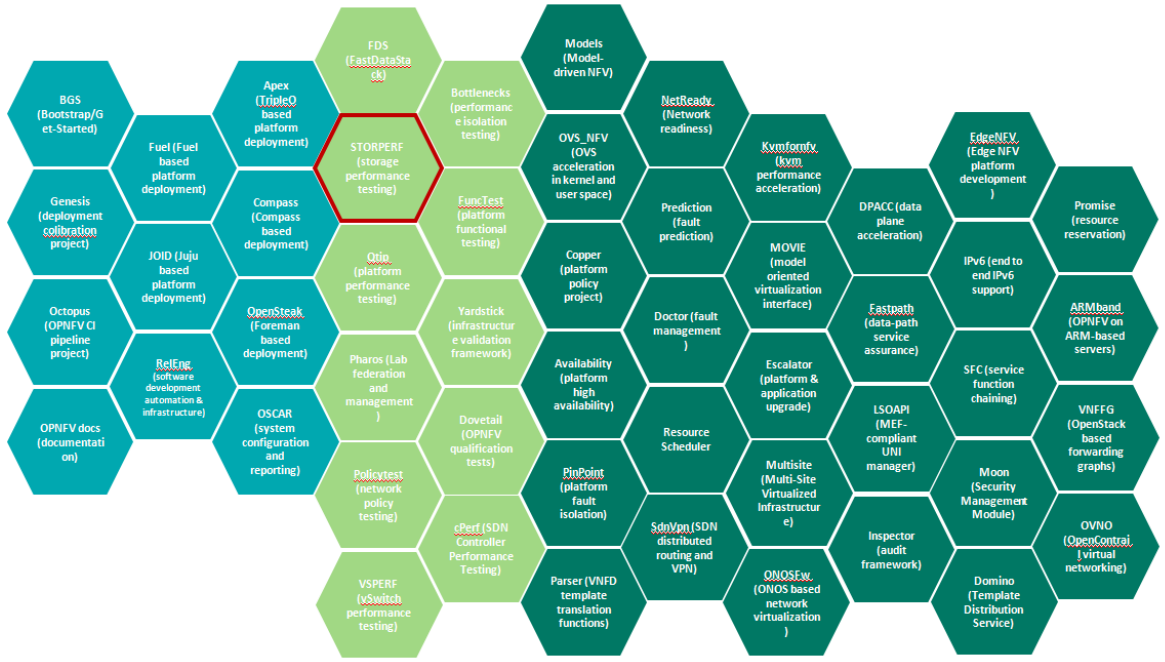
A Network Functions Virtualization (NFV) cloud is a datacenter and network built to host, deploy, and service virtual network functions (VNFs) using a cloud network.

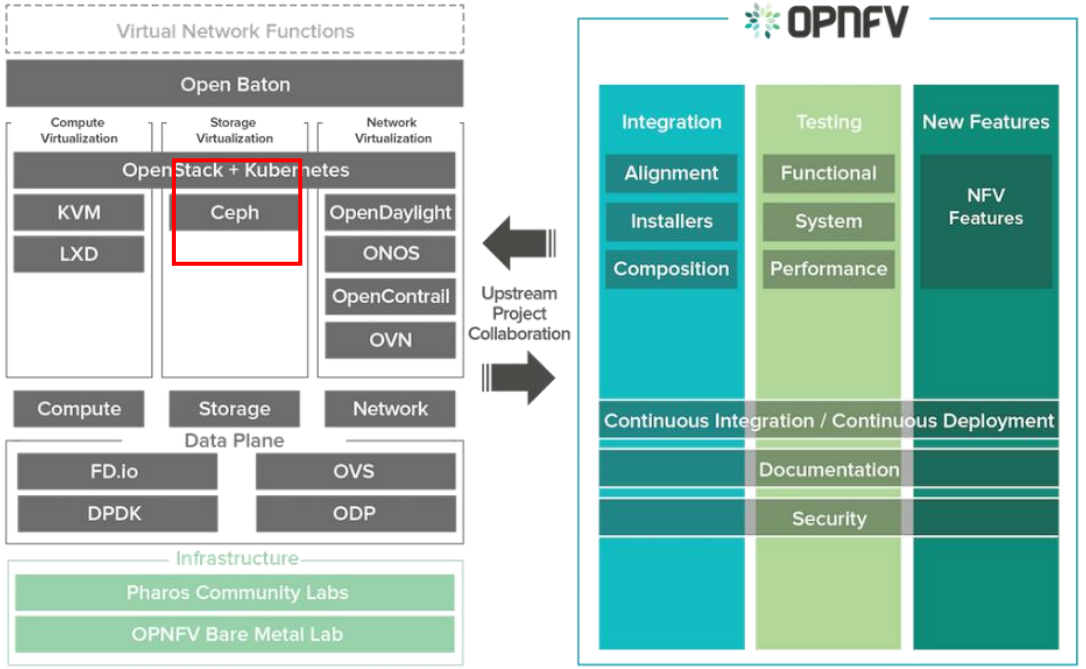
OPNFV

- ❖ accelerate NFV transformation
- ❖ a reference NFV platform
- ❖ an integrated open source platform
- ❖ a large range of use cases
- hi-definition video streaming for use with virtual CDN



OPNFV Storage Project Landscape





- There is only one official OPNFV project that is solely focusing on storage – storperf, which is providing the benchmarking for storage performance. There is no storage functionality focused project.
- Ceph has been part of the official release architecture since Arno, however it is only used by the installers and there is no project covering how to use it in a functional view

Stor4NFV

Stor4NFV provides a storage solution based on Ceph and OpenSDS, and focuses on the optimization for storage intensive use cases of NFV, like I/O performance improvements.

Stor4NFV: Status

☐ Became one official OPNFV project in Sept 2017

☐ F release

- Integrate Ceph with OpenSDS
- Build installers of Stor4NFV, including Compass4NFV
- Support K8s scenario

☐ Goals of G release

- Integration with Apex installer
- Support OpenStack scenario
- Integration with storperf project
- Client cache performance

Stor4NFV: Target

I/O performance improvements will be one initial target, but we also need to consider scaling and stability factors as well. Ultimately storage will need to progress to be a key part of the entire OPNFV architecture.

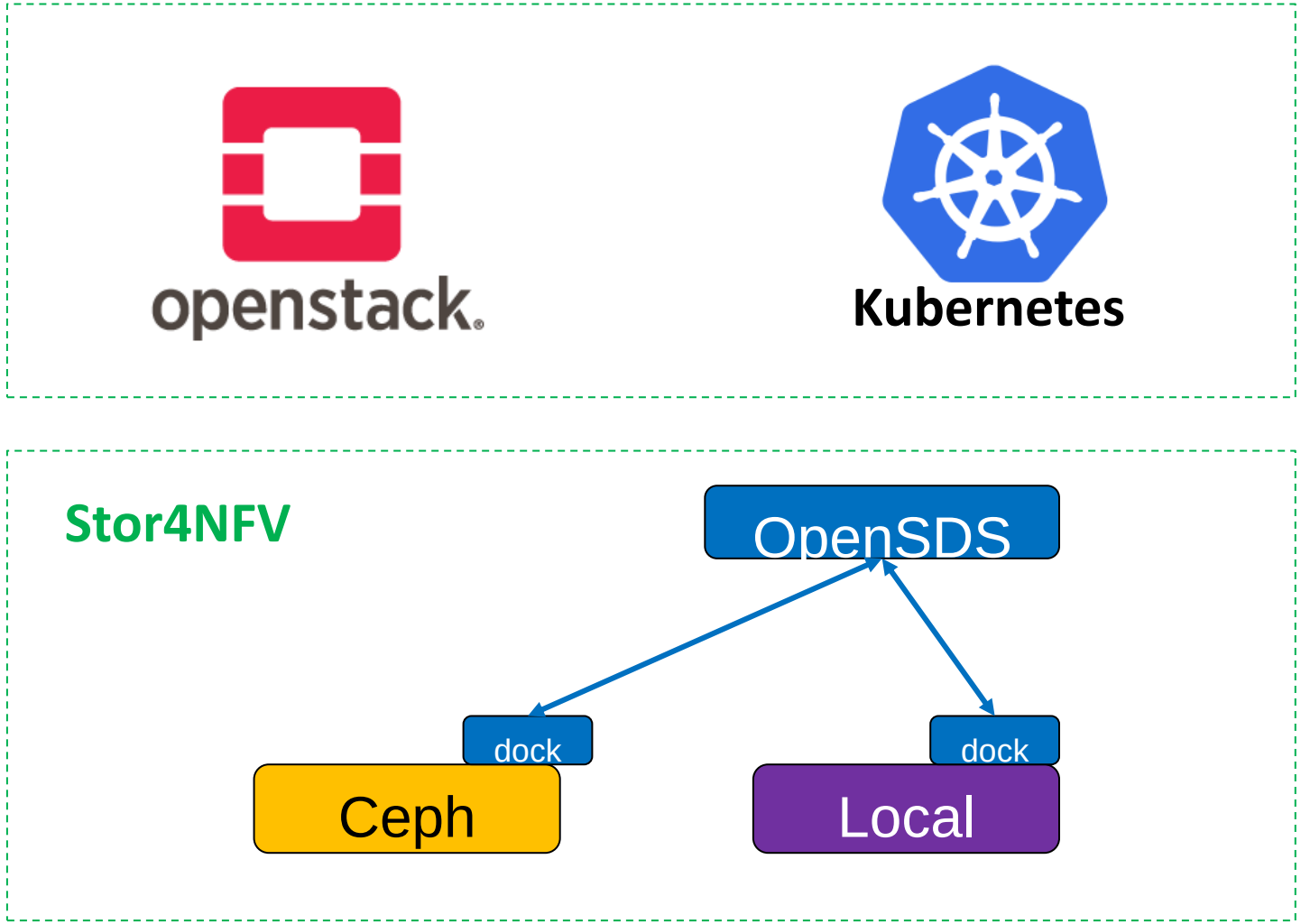
- Client RDB cache to accelerate Ceph I/O read and write
- High throughput and low latency solution based on all flash storage media
- Customized optimization approaches for different sorts of data, such as small data and large data
- …

Stor4NFV Architecture

Standalone

with OpenStack

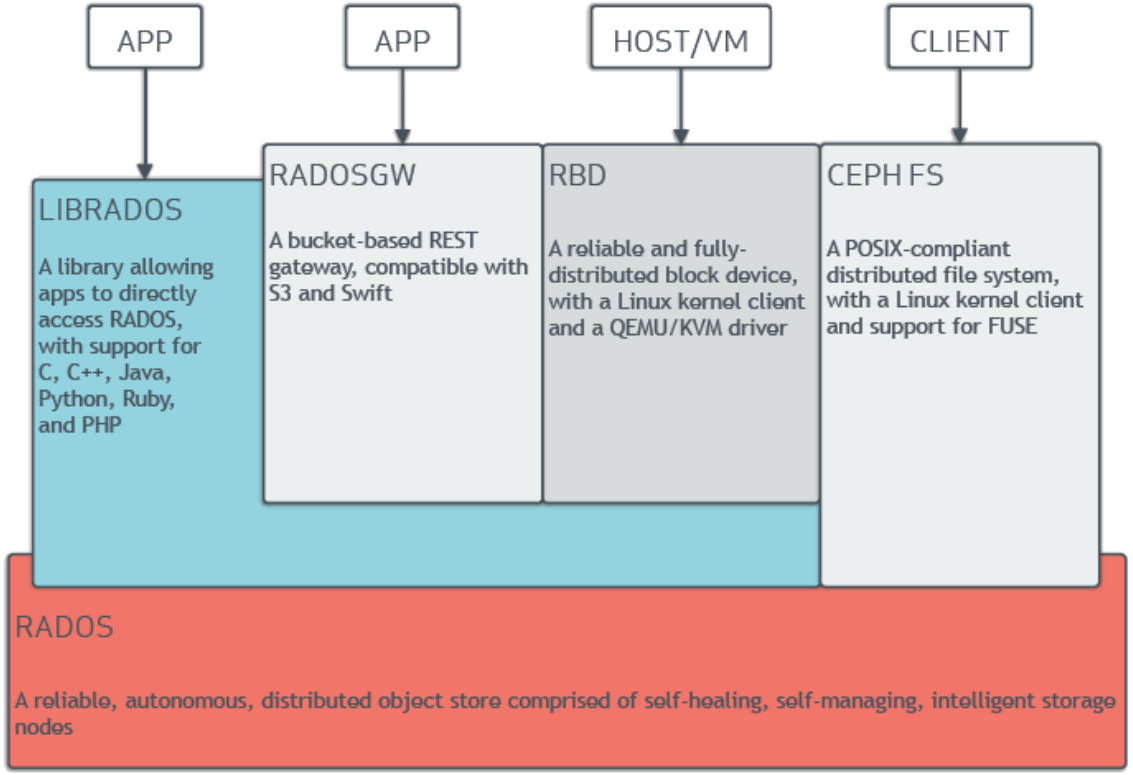
with K8S



Ceph

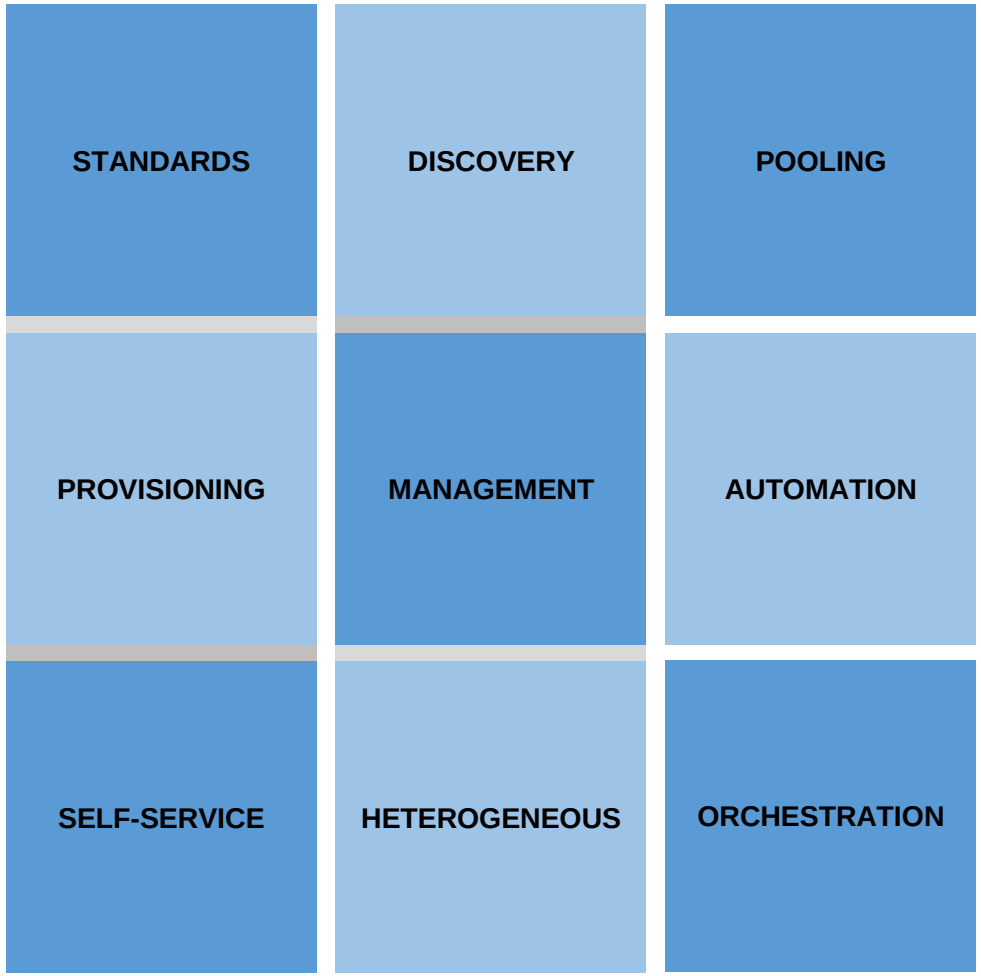
Ceph is by default recommended in the reference design since A release.

For Stor4NFV project, Ceph is the backend driver of OpenSDS.



OpenSDS

OpenSDS is software-defined storage control for traditional and cloud native environments with enterprise, commodity and cloud storage



OpenSDS: Community

Technical Steering Committee

Steven Tan, Chairman
Huawei, VP & CTO Cloud Storage Solution

Rakesh Jain, Vice-Chair
IBM, Research Engineer and Architect

Allen Samuels
Western Digital, R&D Engineering Fellow

Anjaneya "Reddy" Chagam
Intel, Chief SDS Architect

Jay Bryant
Lenovo, Cloud Storage Lead

End-User Advisory Committee

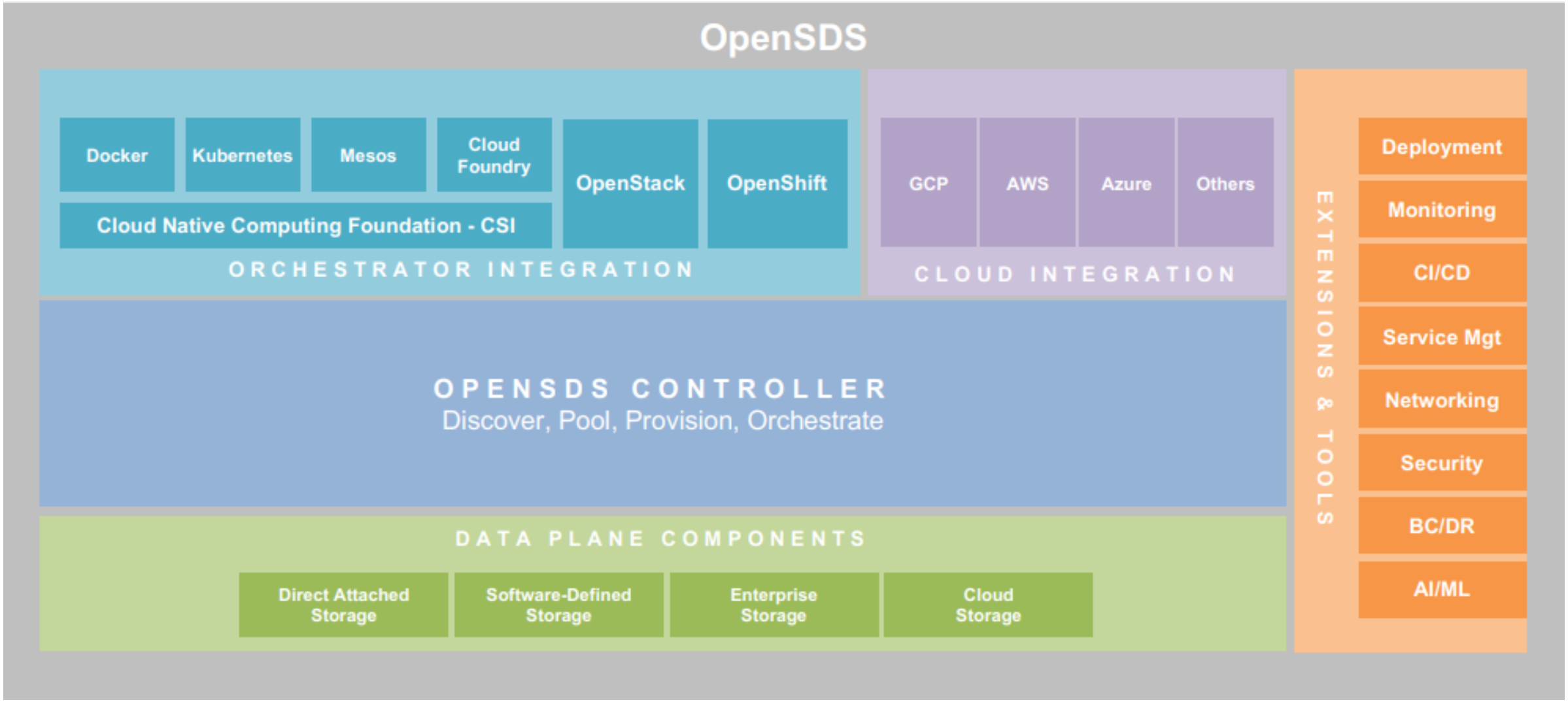
Cosimo Rossetti
Vodafone, Lead Storage Architect

Yusuke Sato
Yahoo Japan, Infrastructure Lead

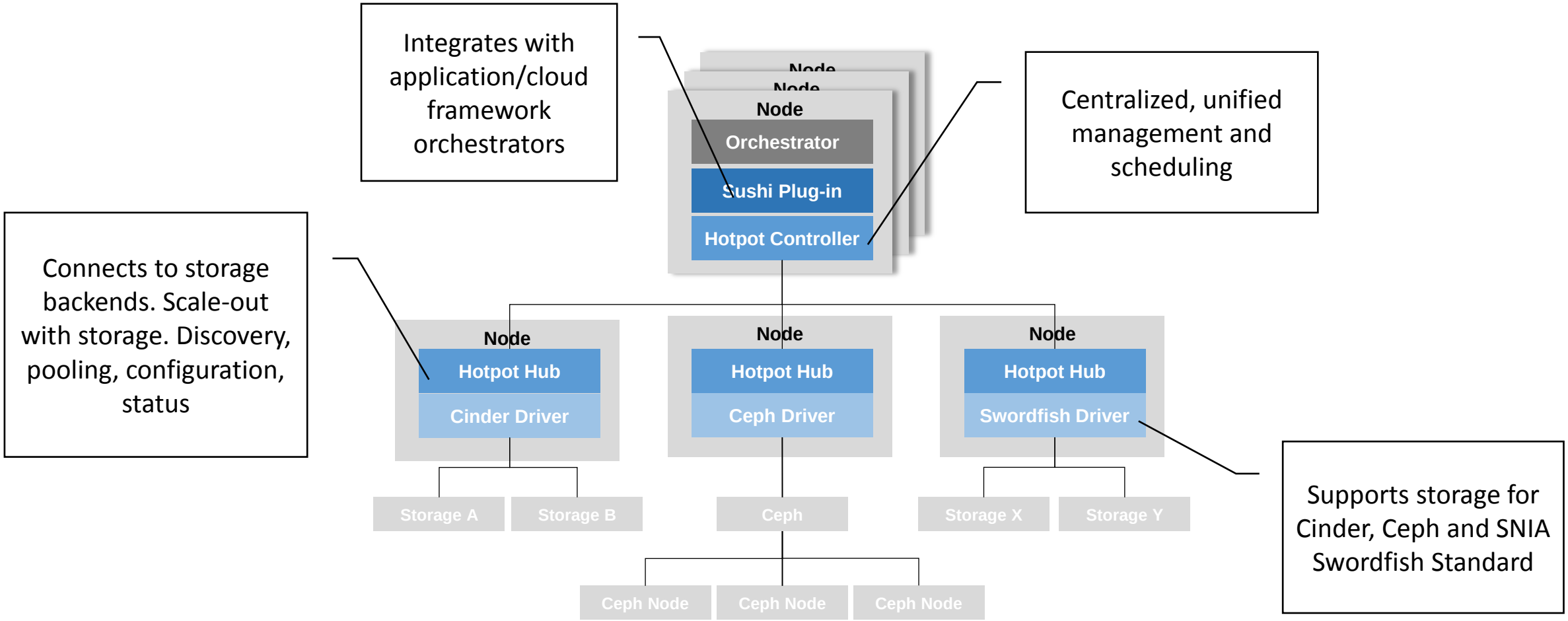
Kei Kusunoki
NTT Communications, Storage Architect

Yuji Yazawa
Toyota ITC, Group Lead

OpenSDS: Framework



OpenSDS: Architecture



OpenSDS: Key Value to Ceph

Policy-Driven Orchestrator

Storage Provisioning and Data Management

STORAGE PROFILE

- **Profile Name**
- **Regions {list of regions storage can be provisioned from}**
- **Protocols {list of data transfer protocols}**

- **Profile Policies (configured by administrator)**
  - **Max Request Size** {max vol/share/object size}
  - **Performance** {QoS, latency, throughput, IOPs}
  - **Availability** {HA mirror|replicas|EC{m,n}, geo-distribute}
  - **Optimization** {thin|compress|dedupe}
  - **Protection** { {snapshot|backup {pool}}}
  - **Lifecycle**{event,{migrate|replicate|compress|archive|delete|erase}}}
  - **Tiering** {list of tiers and conditions}
  - **Networking** {VPN …}
  - **Security** { ACL, encryption, compliance, …}
  - **Sharing** { none | read write | read only } {list of tenants to share}

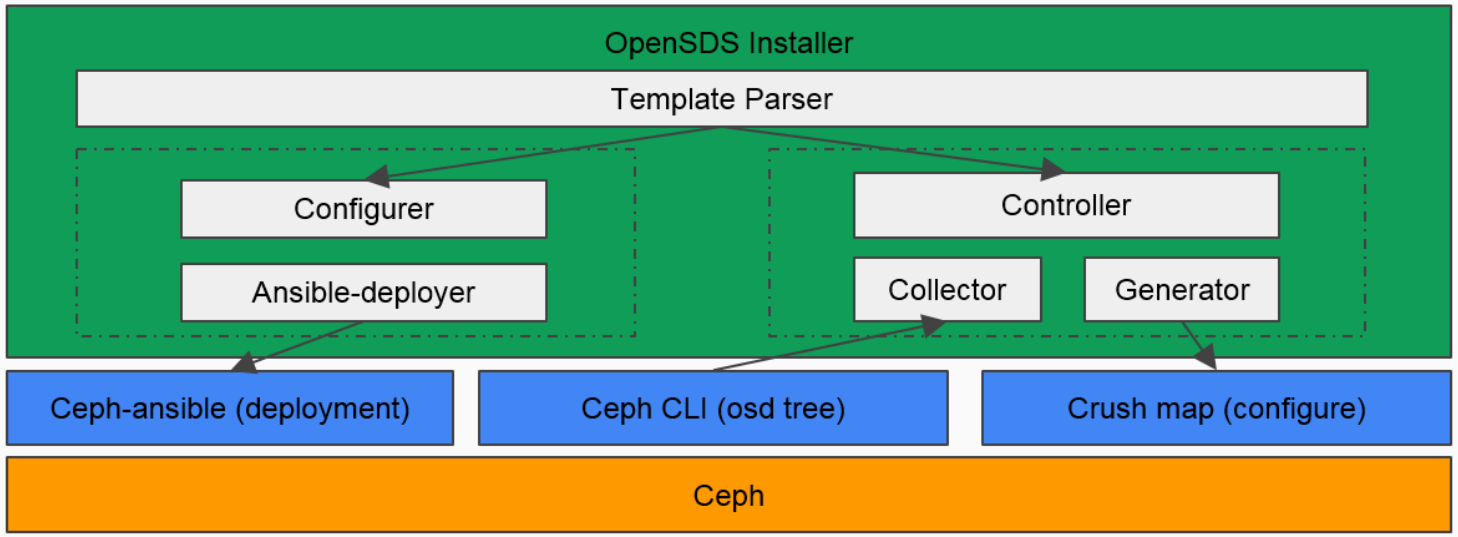
OpenSDS: Key Value to Ceph

Dedicated Differentiator

Enable advanced features (config, crushmap, …)

template.yml

```
configFile: /etc/ceph/ceph.conf
pool:
  ssd:
    storageType: block
    recoveryTimeObjective: 0
    provisioningPolicy:
    - fixed
    accessProtocol: rbd
    maxIOPS: 1000
    disks:
    - hostname: test
      path: /dev/loop0
```



OpenSDS: OpenStack Scenarios

☐ Keystone Integration

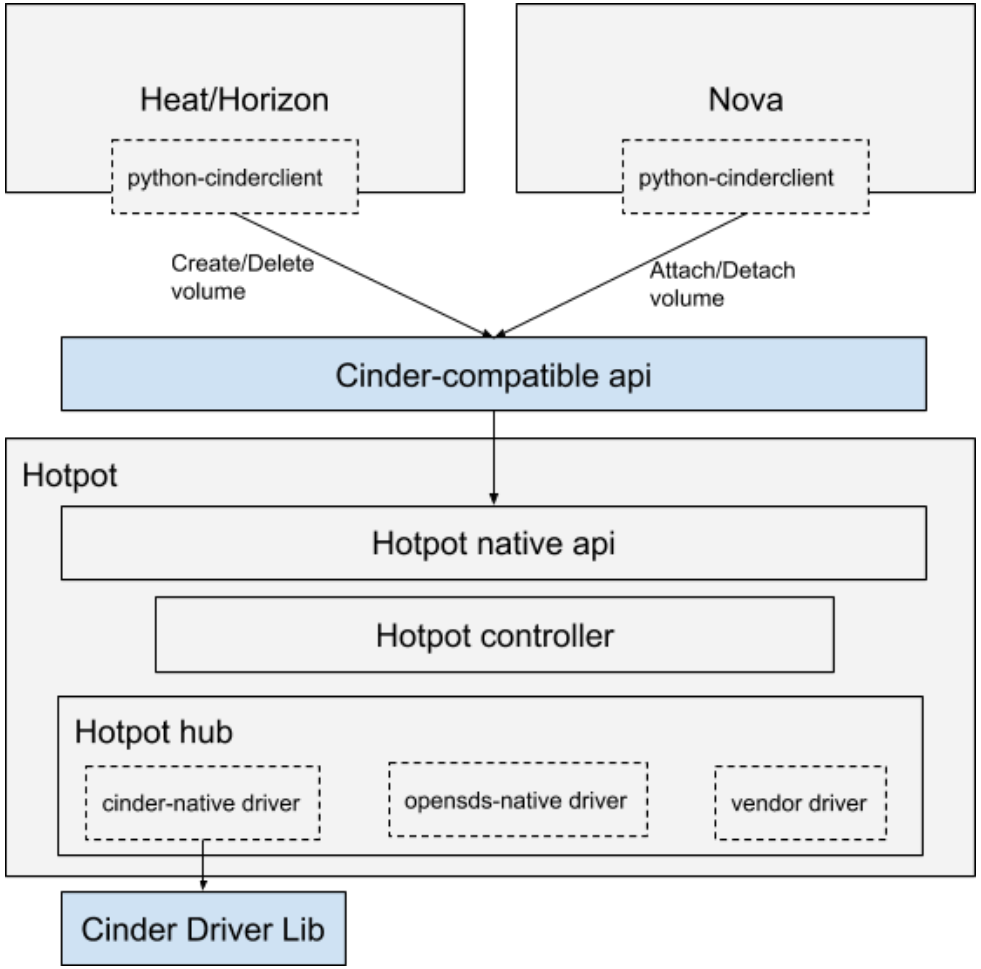
- OpenSDS should recognize tenants (projects in OpenStack) and users created in OpenStack Keystone.
- OpenSDS should provide authentication and multi-tenant authorization through Keystone's Identity APIs.

☐ Glance Integration

- OpenSDS also needs to integrate with Glance and work with its image stores so that hotpot can upload volume to image stores and create volume from image.

☐ Cinder Driver Lib Integration

- There is a POC implementation of Cinder driver lib by a Red Hat engineer: https://github.com/Akrog/cinderlib. It is a Python library that allows volume drivers to be used outside of Cinder. We can write a golang-python sdk of southbound driver that uses this driver lib.



OpenSDS: Kubernetes Scenarios

☐ Container-Storage-Interface (CSI)

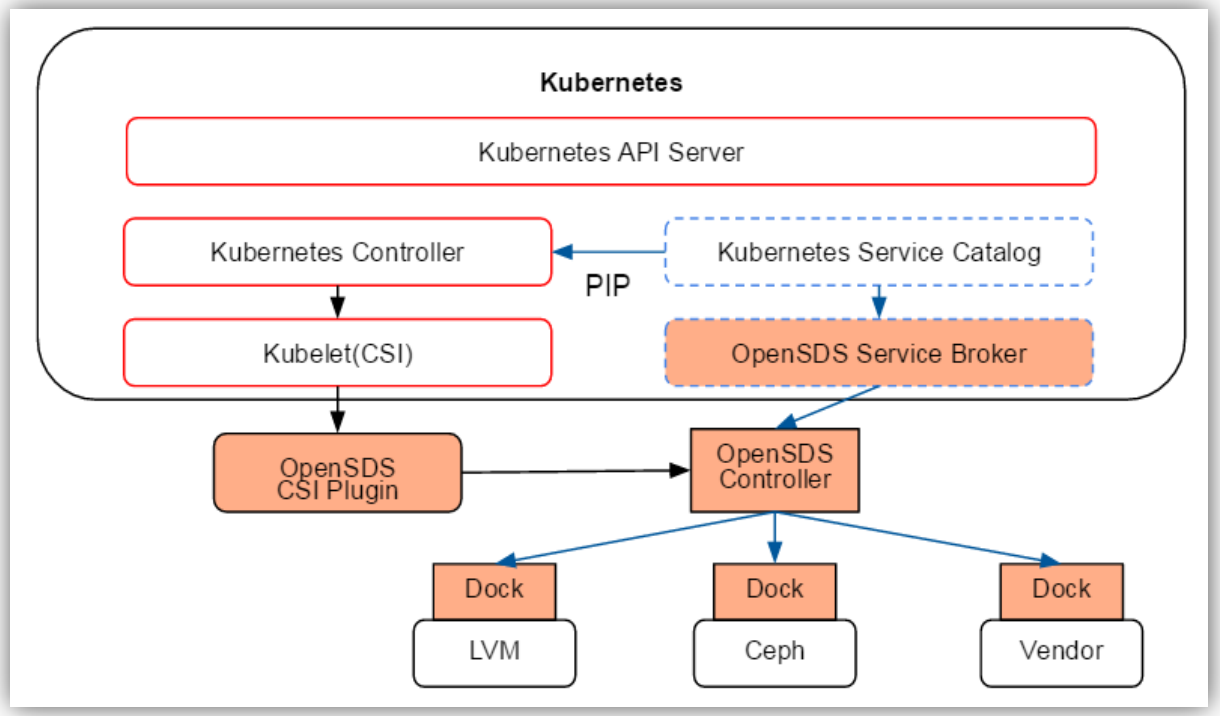
- Enable storage vendors (SP) to develop a plugin once and have it work across a number of container orchestration (CO) systems.
- OpenSDS is the first batch of storage controllers to support CSI in k8s 1.9

☐ Kubernetes Service Catalog

- Integration between Kubernetes and brokers implementing the OSB API.
- 4 resources: Broker, ServiceClass, Instance, Binding.

☐ OpenSDS Service Broker

- Responsible for advertising a catalog of service offerings and service plans to Service Catalog, and acting on requests from Service Catalog for provisioning, binding, unbinding, and deprovisioning.
- Expose OpenSDS advanced features (replication, migration, data protection and so on) to Kubernetes without changing a line of code.



https://github.com/opensds/nbp

THANK YOU