CLOUD NATIVE
COMPUTING
FOUNDATION

Storage Working Group

# SWG Intro
# KubeCon EU 2018

Clint Kitson - @clintkitson - VMware

Deck prepared by Alex Chircop - @chira001 - StorageOS

# Panel Today

Ardalan Kangarlou @8ardalan8 (NTAP)

Quinton Hoole @quintonhoole (Huawei)

Saad Ali @the_saad_ali (Google)

Steve Wong @cantbewong (VMware)

# Storage WG

The Storage Working Group meets on the 2nd and 4th Wednesday of every month at 8AM PT (USA Pacific) (join: https://zoom.us/my/cncfstoragewg)

Further details :

https://github.com/cncf/wg-storage

Meeting minutes: https://goo.gl/wRqerO

Mail list: https://groups.google.com/forum/#!forum/cncf-wg-storage

Over 30 members from many companies (and individuals) including:

Datera, Dell, Diamanti, Docker, DriveScale, Google, HPE, Huawei, IBM, Iguazio, Infinidat, Mesosphere, NetApp, OpenEBS, OpenSDS, PortWorx, Pure, Quantum, RedHat, StorageOS, Upbound, VMware

# Why is storage critical?

1. There’s no such thing as a stateless architecture, applications store state somewhere.
2. Cloud native is about supporting patterns such as portability. Containers on their own do not enable portability.
3. Interoperating with storage increases cloud native’s relevance and leads to better applications.

# Goal of storage in CNCF

- In order to drive ubiquity of cloud native computing, the CNCF intends to enable a thriving **storage eco-system** that is vendor and platform neutral and **interoperable for applications**

# Storage WG Mandate

Primary Priority:

- Clarify terminology and landscape
- How components are used in clouds
- Compare and contrast with regards to properties, ie.: availability, durability, performance, scalability, API

# Projects

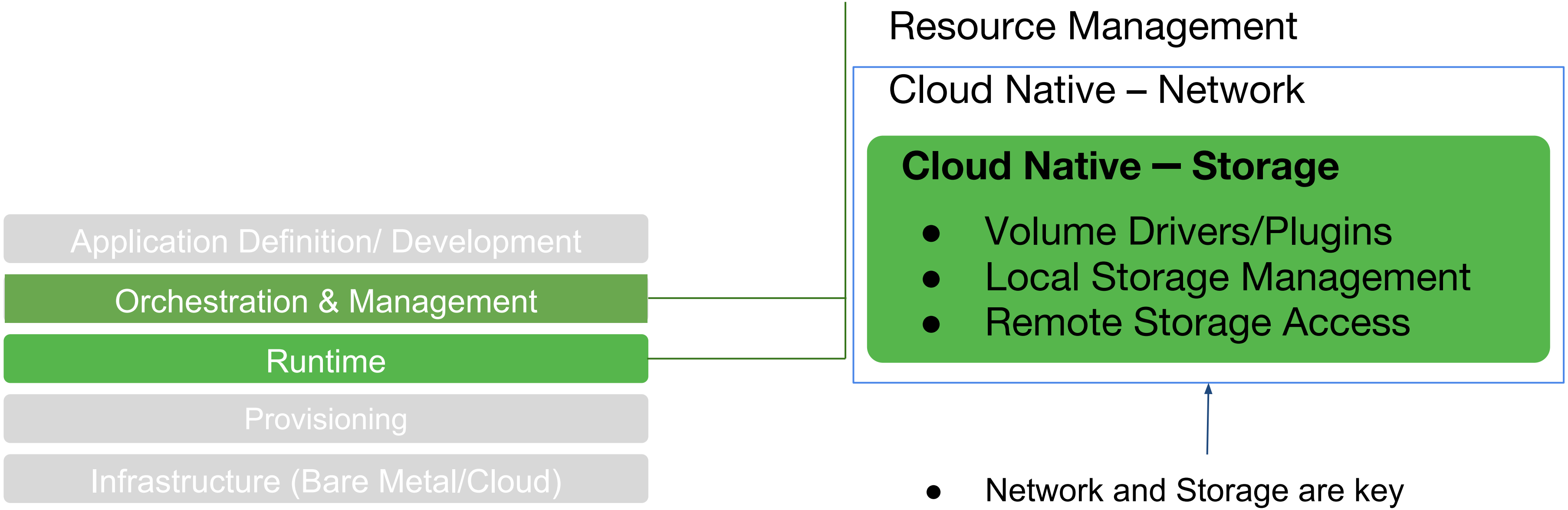
Over the last few months, several projects have presented at the WG and collated feedback including :

CSI, Rook, REX-Ray, TiKV, PingCAP, Dotmesh, Yugabyte, OpenEBS, Open Services Broker, Vitess, Minio

Storage projects accepted by the CNCF TOC:

- Rook accepted as a Sandbox Project
- Vitess accepted as an Incubation project

# Storage in CNCF Reference Architecture



Application Definition/ Development

Orchestration & Management

Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

Resource Management

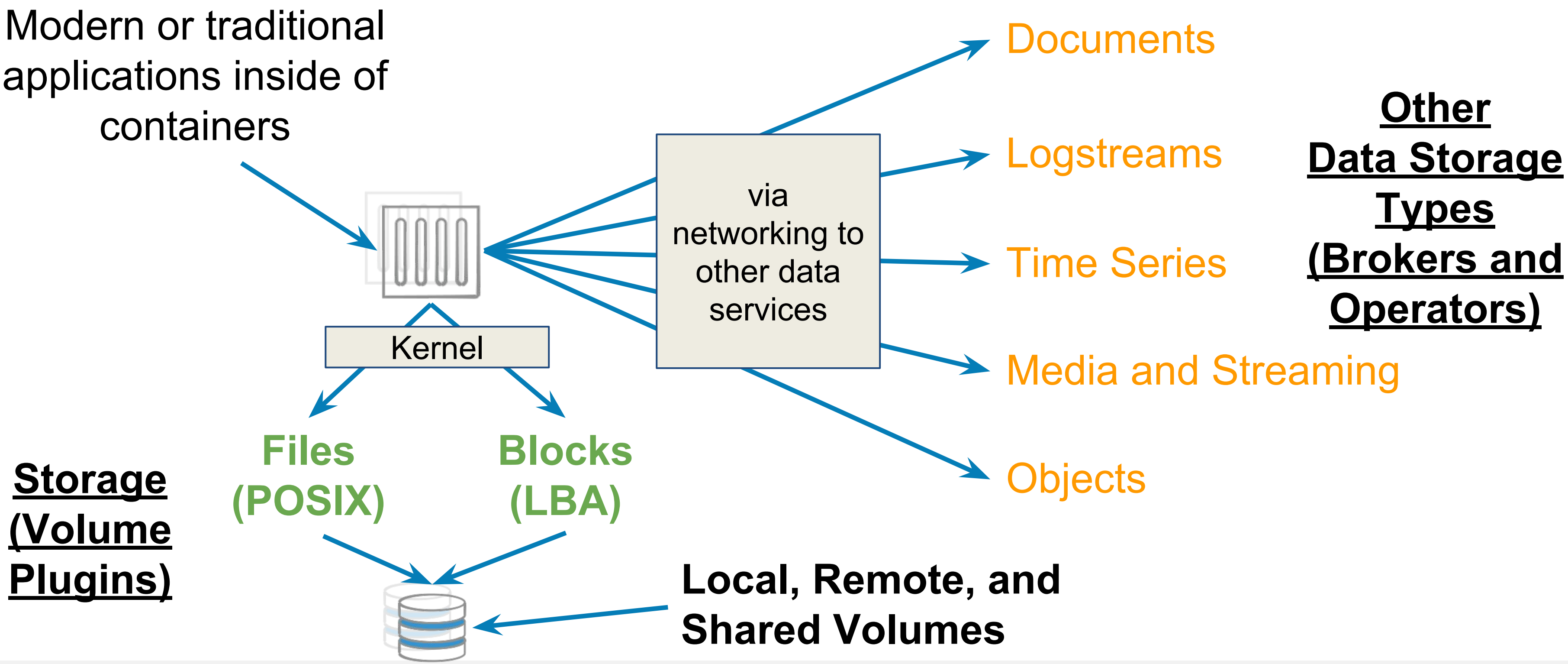
Cloud Native – Network

**Cloud Native – Storage**

- Volume Drivers/Plugins
- Local Storage Management
- Remote Storage Access

- Network and Storage are key infrastructure components
- Focus on interoperability

# Storage and data services



Modern or traditional applications inside of containers

Kernel

via networking to other data services

Files (POSIX)

Blocks (LBA)

**Storage (Volume Plugins)**

**Local, Remote, and Shared Volumes**

Documents

Logstreams

Time Series

Media and Streaming

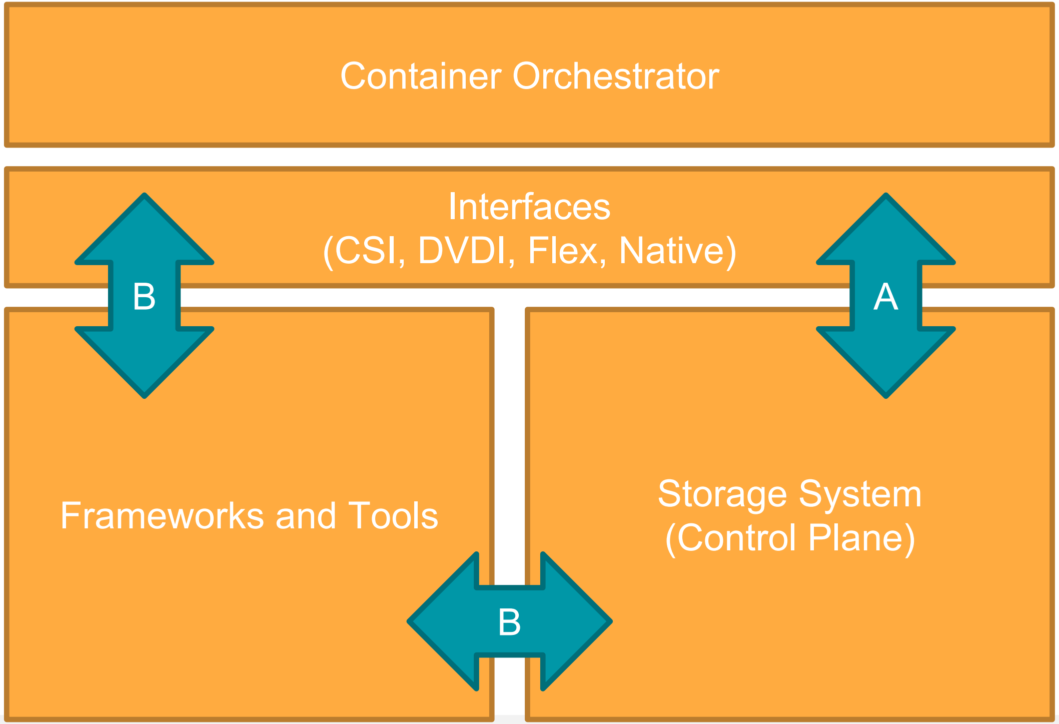
Objects

**Other Data Storage Types (Brokers and Operators)**

# Storage Patterns

- **Interoperable** – applies where a CO can consume storage from Storage Systems
- **Self Service** – applied where the Storage System has an API and Interface that supports dynamic provisioning of volumes through a composable CO interface
- **Interface** – a standards based API supported by the CO
- **API Framework / Tools** – components which act as an intermediary layer between the CO and the Storage System to facilitate the Interface between the CO and the Storage System
- Storage System can be constituted of **(a) an External Service** (e.g. an external storage array or a cloud provided service) or could be **(b) an Orchestrated Storage Platform** where the components of the Storage System are containerized, deployed and orchestrated on a Cloud Native Platform
- **Storage Access Method** – the way that the storage is consumed by an application

# Consumption and Provisioning



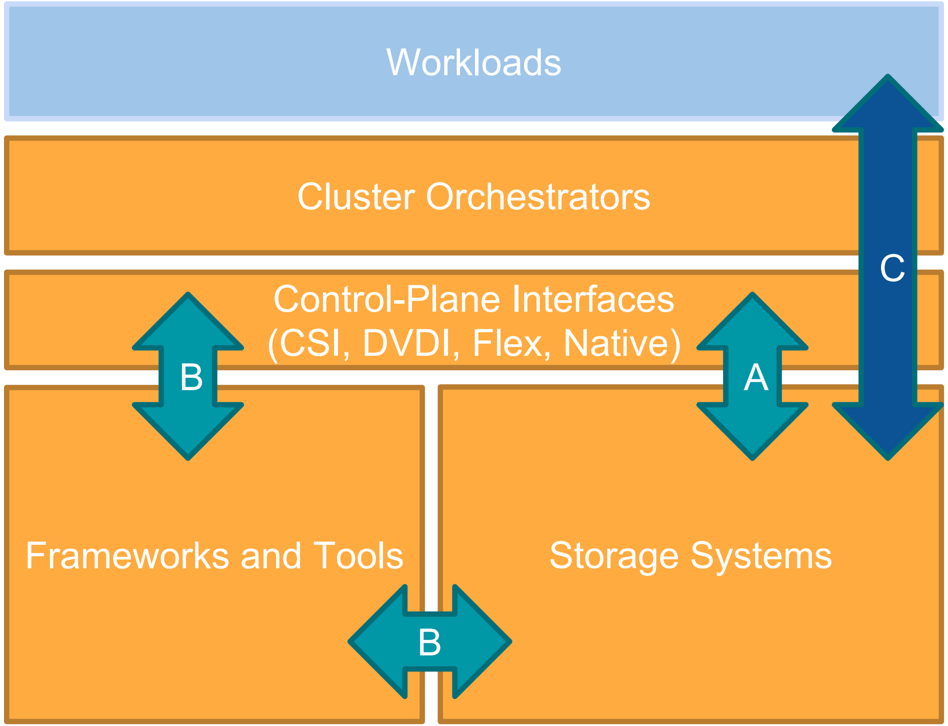
- CO supports one or more **Interfaces** to interact with the Storage System

- Storage System can **(A)** support interface API directly and interact directly with the CO or can **(B)** interact with the CO via an **API framework layer** or other **Tools**.

- Storage system must support the ability to consume volumes through the CO composable interface to be considered **Interoperable**

- Storage system must support the ability to dynamically provision volumes through the CO composable interface in order to be considered **Self Service**

# CNS Volumes Landscape



- CO supports one or more **Interfaces** to interact with the Storage System

- Storage System can **(A)** support control-plane interface API directly and interact directly with the CO or can **(B)** interact with the CO via an **API framework layer** or other **Tools**.

- Storage system must support the ability to provision and consume (C) volumes through a standard interface to be considered **Interoperable**

- Workloads interact (C) with storage systems over various data-plane methods

[Landscape Sheet]

# Next Steps

Advanced session to discuss WG mandate and priorities from the TOC

Opportunity for sub-groups to focus on specific storage technologies e.g. Interfaces (CSI), Volumes, KV stores, object stores, distributed databases

# Other sessions while you are at Kubecon

- SIG Storage Intro - Saad Ali
- Kubernetes Storage Lingo 101 - Saad Ali
- Using Kubernetes Local Storage for Scale-Out Storage Services in Production - Michelle Au
- Container Storage Interface: Present and Future - Jie Yu
- Policy-Based Volume Snapshots Management in Kubernetes - Jing Xu

# Technical Panel

Questions?

CLOUD NATIVE
COMPUTING
FOUNDATION

# THANK YOU