STORAGEOS

Everything you need to know about Storage for Kubernetes

Webinar

Alex Chircop

About me

- **Founder & CEO of StorageOS, building a software defined, cloud native storage platform**
- **Co-chair CNCF Storage SIG**
- 25 years of engineering infrastructure platforms before embarking on the startup adventure
- Previously Goldman Sachs and Nomura
- Twitter: **@chira001**

CNCF SIG Storage - Get involved!

The **CNCF Storage SIG** meets on the

2nd and 4th Wednesday of every month at 8AM PT (USA Pacific)

Call	https://zoom.us/my/cncfstoragewg
SIG rep	https://github.com/cncf/sig-storage
Meeting Minutes	http://bit.ly/cncf-storage-sig-minutes
Mail List	https://lists.cncf.io/g/cncf-sig-storage

STORAGEOS

Why is storage so important?

Why Storage?

There is no such thing as a stateless architecture!

All applications store state somewhere...

Challenge: No Storage Pets!

Cloud Native Storage should be:

- **Declarative & Composable**
- **API Driven**

Challenge: Data needs to follow!

Cloud Native Storage should be:

- **Application Centric**
- **Platform Agnostic**
- **Agile**

Challenge: Humans are fallible - automate everything!

Cloud Native Storage should be:

- **Consistently Available**
- **Performant**
- **Natively Secure**

STORAGEOS

CNCF Storage Landscape Whitepaper

http://bit.ly/cncf-storage-whitepaper

- How is a storage system **deployed**
- Definition of the **attributes** of a storage system
- Definition of the **layers** in a storage solution with a focus on terminology and how they impact the attributes
- Definition of the **data access interfaces** in terms of volumes and application APIs
- Definition of the **management interfaces**

Instantiation and Deployment

Instantiation	Description
Hardware	**Deployed as hardware solution in a datacenter.** This limits the portability of the application and generally means that such systems cannot be deployed in a public cloud environment
Software	**Deployed as software components** on commodity hardware, appliances or cloud instances. Software solutions tend to be more **platform agnostic** and can be installed both on-premises as well as cloud environments. Some software defined storage systems can also be **deployed as a container** and **deployment can be automated by an orchestrator**.
Cloud Services	**Consumed from public cloud providers**. Cloud services provide storage services in cloud environments.

Storage Attributes

Availability	Scalability	Performance	Consistency	Durability
Failover	Clients	Latency	Delay to access correct data after a commit	Data protection
Moving access between nodes	Operations	Operations		Redundancy
Redundancy	Throughput	Throughput	Delay between commit and data being committed to non-volatile store	Bit-Rot
Data protection	Components			

Storage Layers

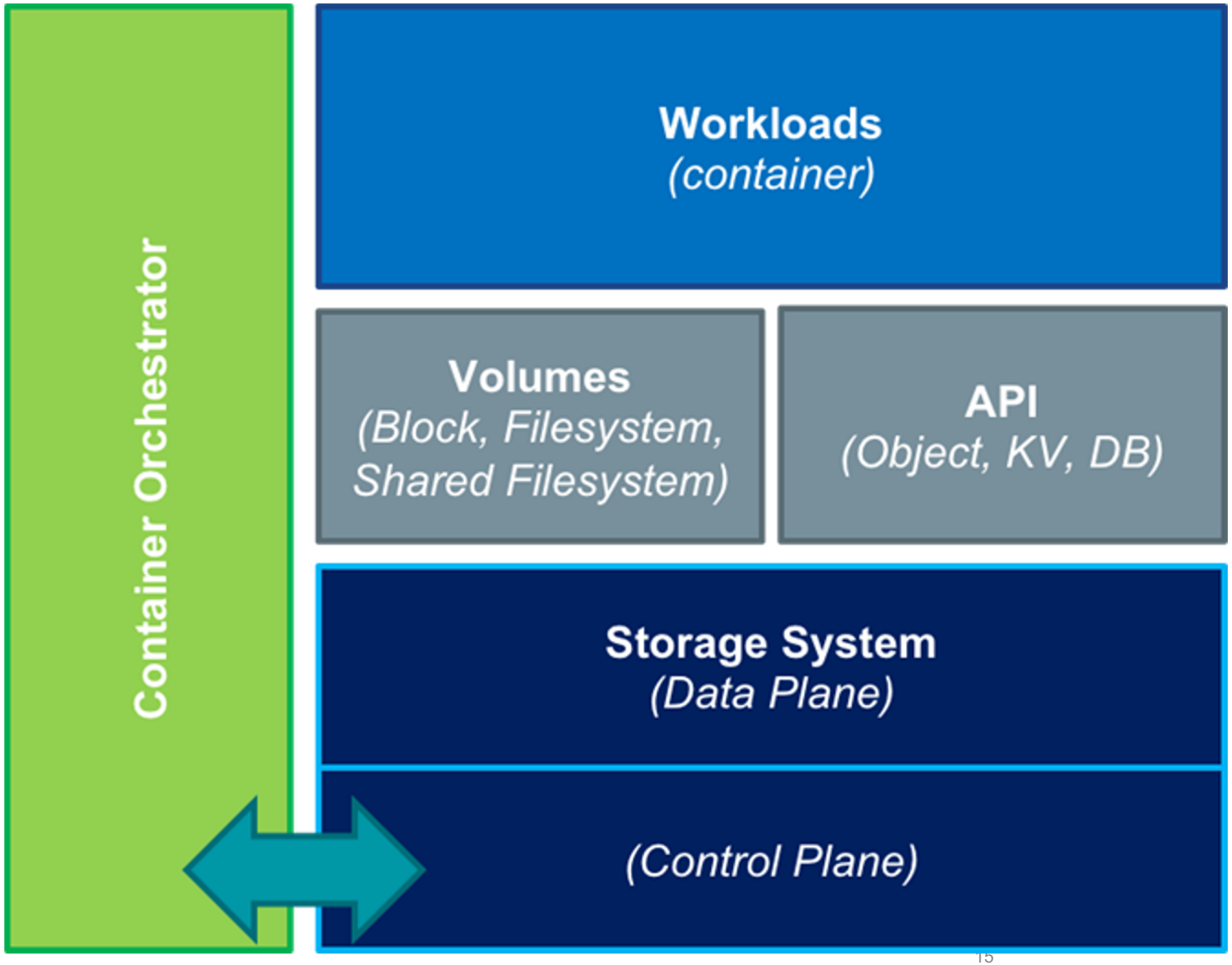
Orchestrator, Host and Operating System
Storage Topology (centralized, distributed, sharded, hyperconverged)
Data Protection (RAID, Erasure coding, Replicas)
Data Services (Replication, Snapshots, Clones, etc.)
Physical, Non-Volatile Layer

Storage Topology Comparison

	Local	Remote	Distributed
Availability	**Limited by failure of components locally and ability to failover.** If a node fails, the local storage is isolated to the local node.	**May be limited by single points of failure.** Workloads can move to another node and reconnect to the remote storage.	**Clients may access numerous nodes, and any storage node failures can be mitigated.** The additional complexity may also add operational complexity which may affect availability or the ability to recover errors.
Scalability	**Limited by local architecture** (1 node; typically TB)	**Limited by monolithic architecture** (2-16 nodes; typically 10s-100s of TB)	**Scale by adding additional systems.** (3-1000s nodes; often supports PB)
Consistency	**Yes** (storage system implementation is easy)	**Yes** (storage system implementation is harder with more nodes)	**Yes** (storage system implementation is hardest)
Durability	**Limited by local components (less)**	**Limited by monolithic architecture (more)**	**Scaling out to additional systems increases durability (most)**
Performance	**Limited by local components, can benefit low-latency applications (100us-5ms, GB/sec)**	**Similar to local, but additional overhead in network transport (500us-5ms, GB/sec)**	**Scaling out to additional systems increases performance (500us-5ms, GB/sec)**

*** The information in this table are generally accepted attributes and measurements among local, remote, and distributed storage systems.*

Data Access Interfaces



Storage can be accessed via **Data Access Interfaces**:

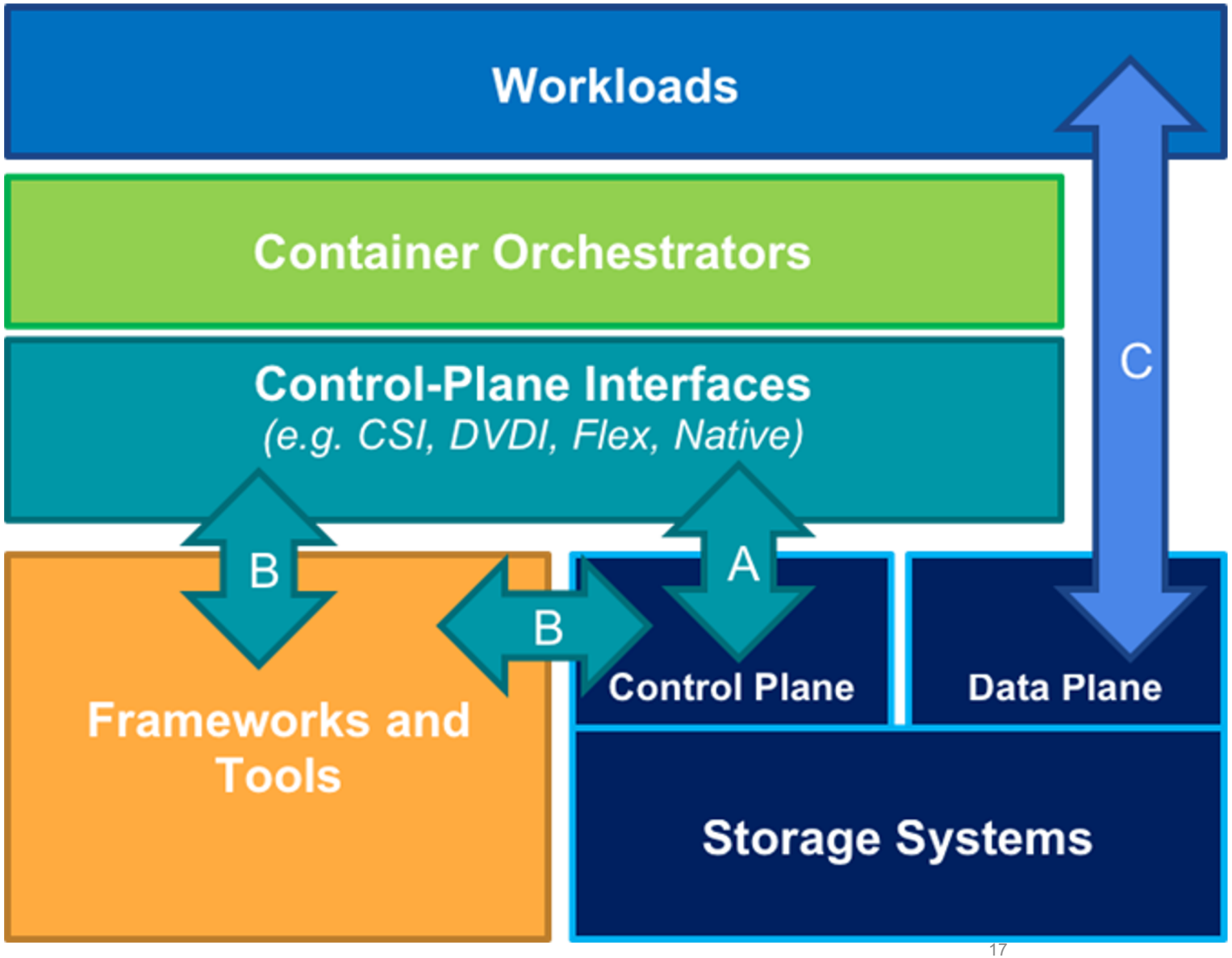
- **Volumes** – accessed through a more traditional file interface in a **block** or **filesystem** interface
- **API** – other ways to persist data such as **object stores**, **KV stores** or **databases**

Data Access Interface Comparison

Data Access Interface	Most suited	Least suited
Block	• Availability • Low latency performance • Good throughput performance for individual workloads	• Capacity scaling • Sharing data with multiple workloads simultaneously
Filesystem	• Sharing data with multiple workloads simultaneously • Optimised throughput for aggregated workloads	• Strong file locking integrity when filesystems are shared
Object Store	• Availability • Large capacities (PB scale) • Durability • Sharing data with multiple workloads simultaneously • Optimised throughput for parallelised workloads	• Low Latency performance

*** The information in this table are generally accepted attributes and measurements for data access interfaces.*

Orchestration and Management Interfaces



Container Orchestration system (**CO**) uses an interface to interact with a storage system

The storage system can:
(A) support control-plane API directly
(B) interact via an API Framework layer or other Tools

Workloads consume **(C)** storage via a data access interface

Control Plane Interfaces

Container Storage Interface (CSI)

CSI v1.0.0 was released in November 2018. Standard for Kubernetes.
https://github.com/container-storage-interface/spec

K8S Native Drivers

Native drivers which are “in-tree” to K8S and provide the interfaces for Persistent Volumes (PV) and Persistent Volume Claims (PVC)
https://kubernetes.io/docs/concepts/storage/

Docker Volume Driver Interface

Provides a mechanism for storage vendors to write a volume driver so that storage systems can be used to provide volumes for a docker container.
https://docs.docker.com/storage/

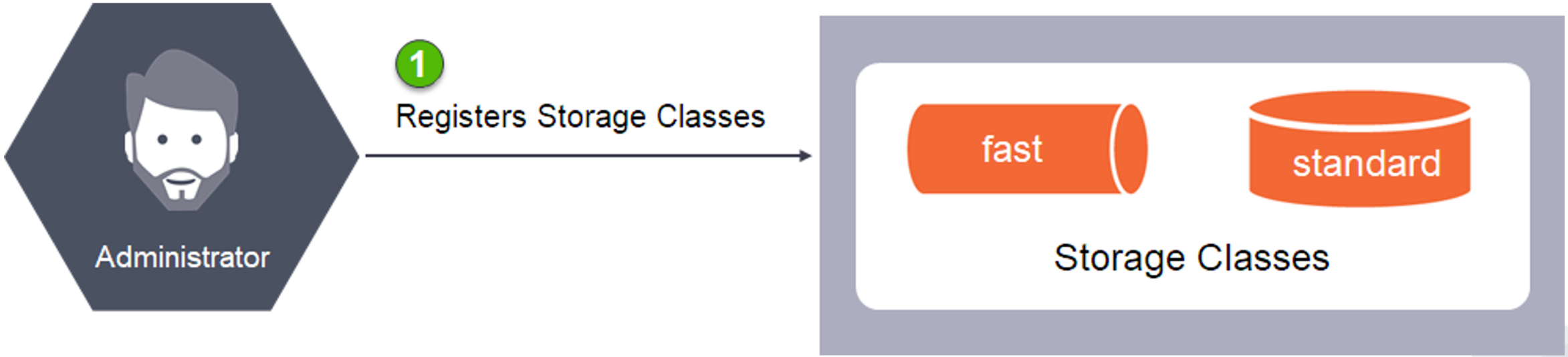
K8S Flex Volume

“out of tree” external volume driver for K8S. Preference is now for CSI.
https://github.com/kubernetes/community/blob/master/contributors/devel/flexvolume.md

STORAGEOS

How is storage configured in Kubernetes?

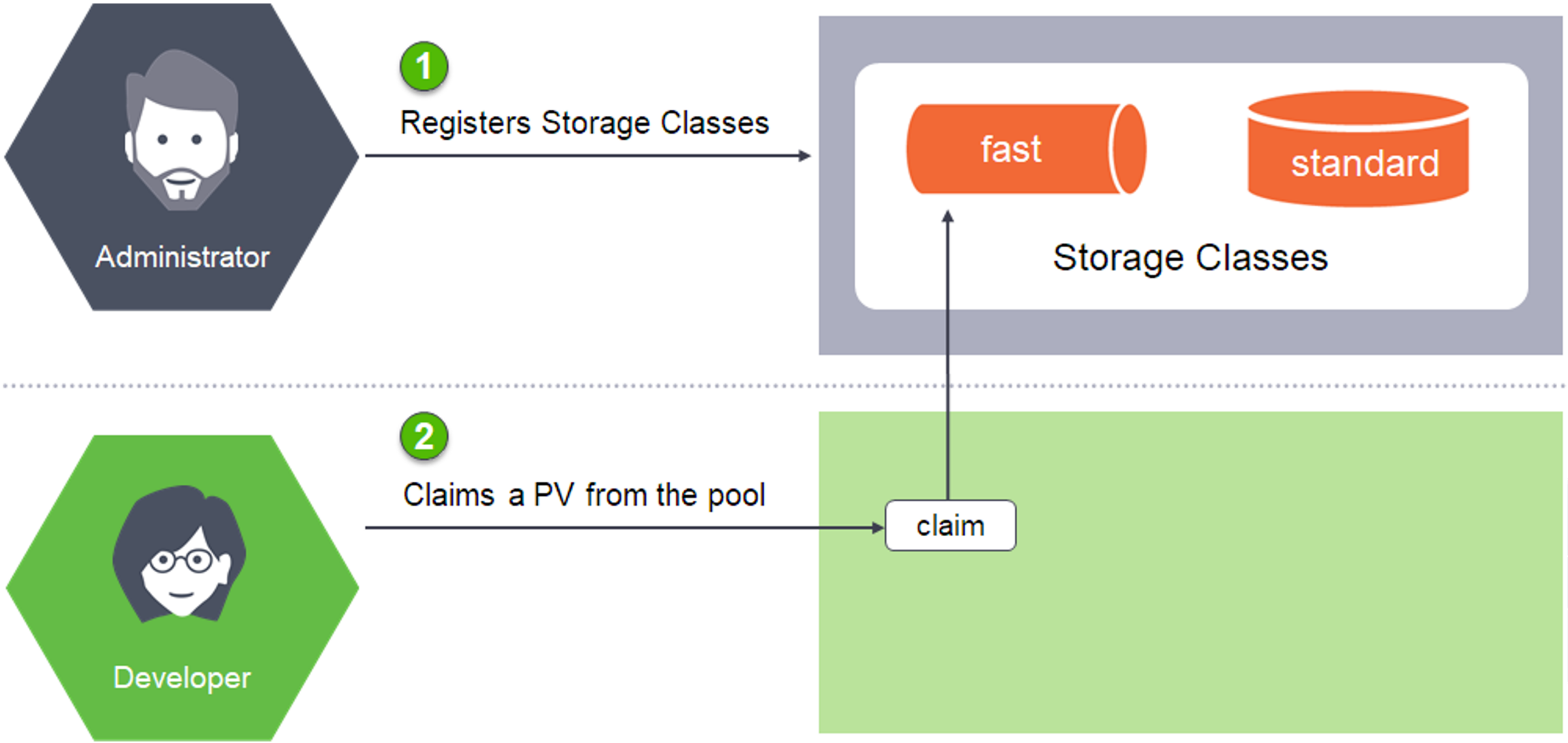
Dynamic Provisioning in Kubernetes: StorageClass



An Example StorageClass

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
      name: fast
      parameters: fsType: ext4
      pool: default      storageos.com/replicas: ”2"
provisioner: storageos
```

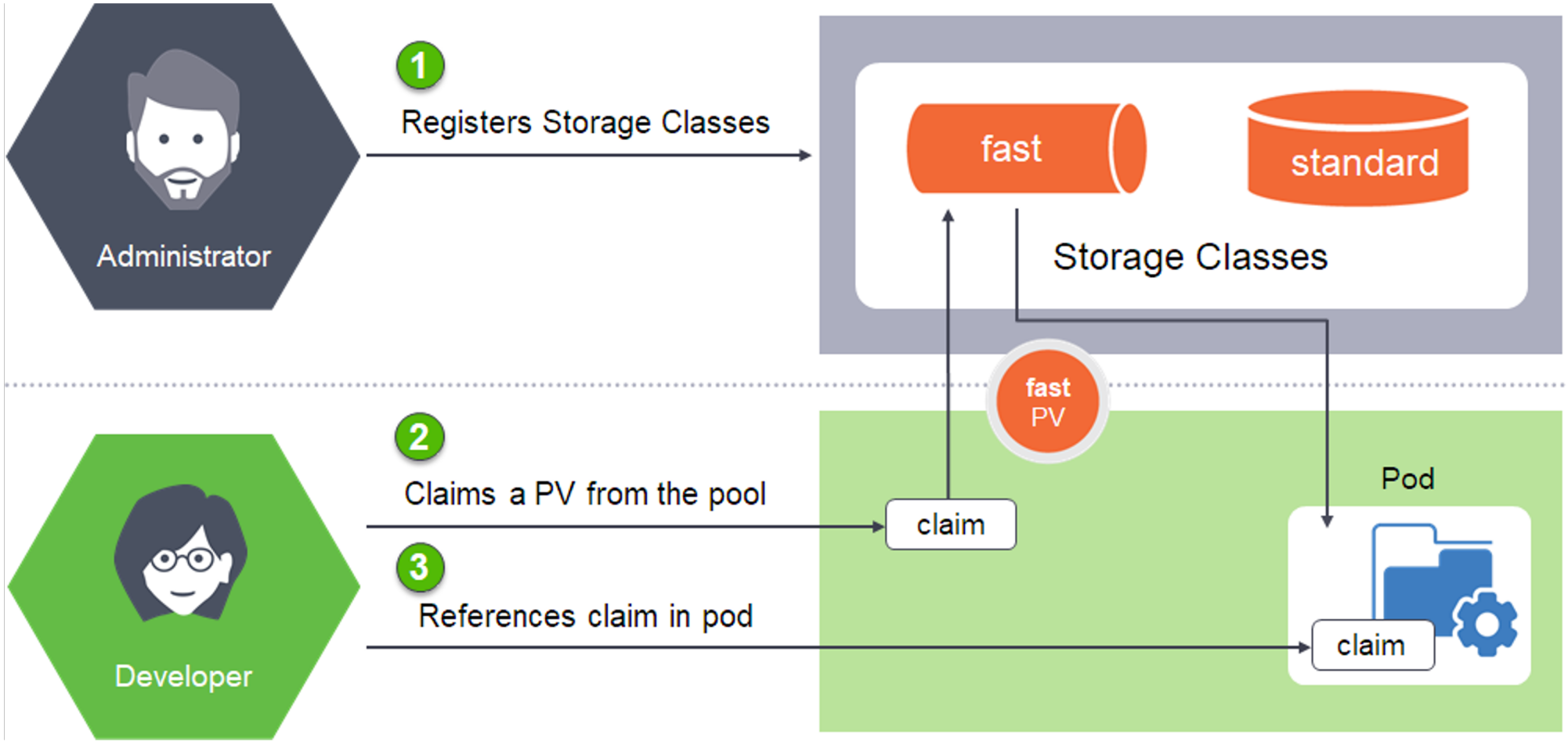
Dynamic Provisioning in Kubernetes: PersistentVolumeClaim



An example PersistentVolumeClaim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: database-vol1
  annotations:
    volume.beta.kubernetes.io/storage-class: fast
  spec:
    accessModes:
      - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
```

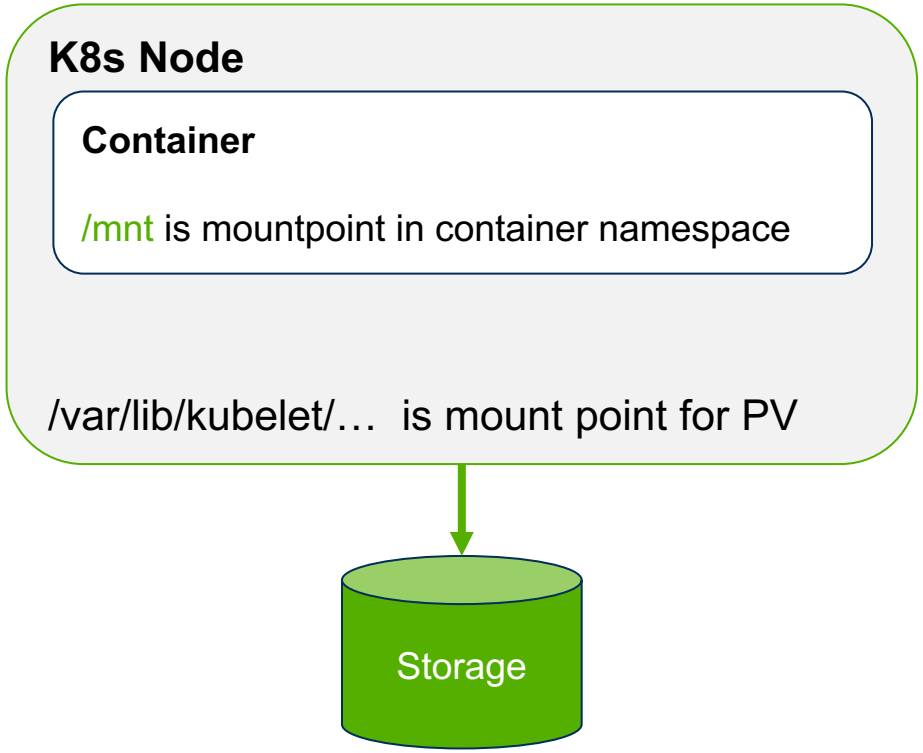
Dynamic Provisioning in Kubernetes: PersistentVolume



Using a volume in a pod

```
apiVersion: v1
kind: Pod
metadata:
 name: test
spec:
 containers:
   - name: debian
     image: debian:9-slim
     command: ["/bin/sleep"]
     args: [ "3600" ]
     volumeMounts:
       - mountPath: /mnt
         name: v1
 volumes:
   - name: v1
     persistentVolumeClaim:
       claimName: pvc-1
```

1. PVC is requested
2. Storage system creates volume (PV)
3. PV is attached to Node and mounted
4. Bind mount of the volume into specified path in container



Volume Access Modes

Access Mode	Description
ReadWriteOnce "RWO"	Volume is mounted and accessed exclusively by only one node.
ReadWriteMany "RWX"	Volume can be mounted on multiple nodes at the same time.

STORAGEOS

Example of provisioning a stateful workload

STORAGEOS

Questions?

STORAGEOS

Thank You

Email: alex.chircop@storageos.com
Twitter: @chira001
Slack: slack.storageos.com

www.storageos.com