容器行业存储标准CSI与Apache Mesos

by 宋子豪

MESOSPHERE

Who am I

- Apache Mesos PMC, Committer
- Tech Lead @Mesosphere
- Leading developments on Containerization in Mesos and DC/OS
- M.S. of Computer Engineering from University of California, Santa Barbara

Overview

- State of storage in Container Orchestrator today
- Benefits of standardization – CSI
  - User perspective
  - Orchestrator perspective
  - Storage Provider perspective
- Overview of CSI
- Mesos overview
- Adopting container standards
- Highlighted new features
- Future roadmap

Background – user demand

Over the past 2 years there has been a huge shift involving *stateful* applications becoming a mainstream feature used by most container users.

redis

MySQL

mongoDB.

PostgreSQL

elastic

Apache CASSANDRA™

Background – container orchestrators

Popular container orchestrators have *independently* evolved storage interfaces

CLOUD FOUNDRY

DC/OS

kubernetes

Background – storage providers

Selected open source and commercial vendors have solutions – sometimes usable across orchestrator platforms

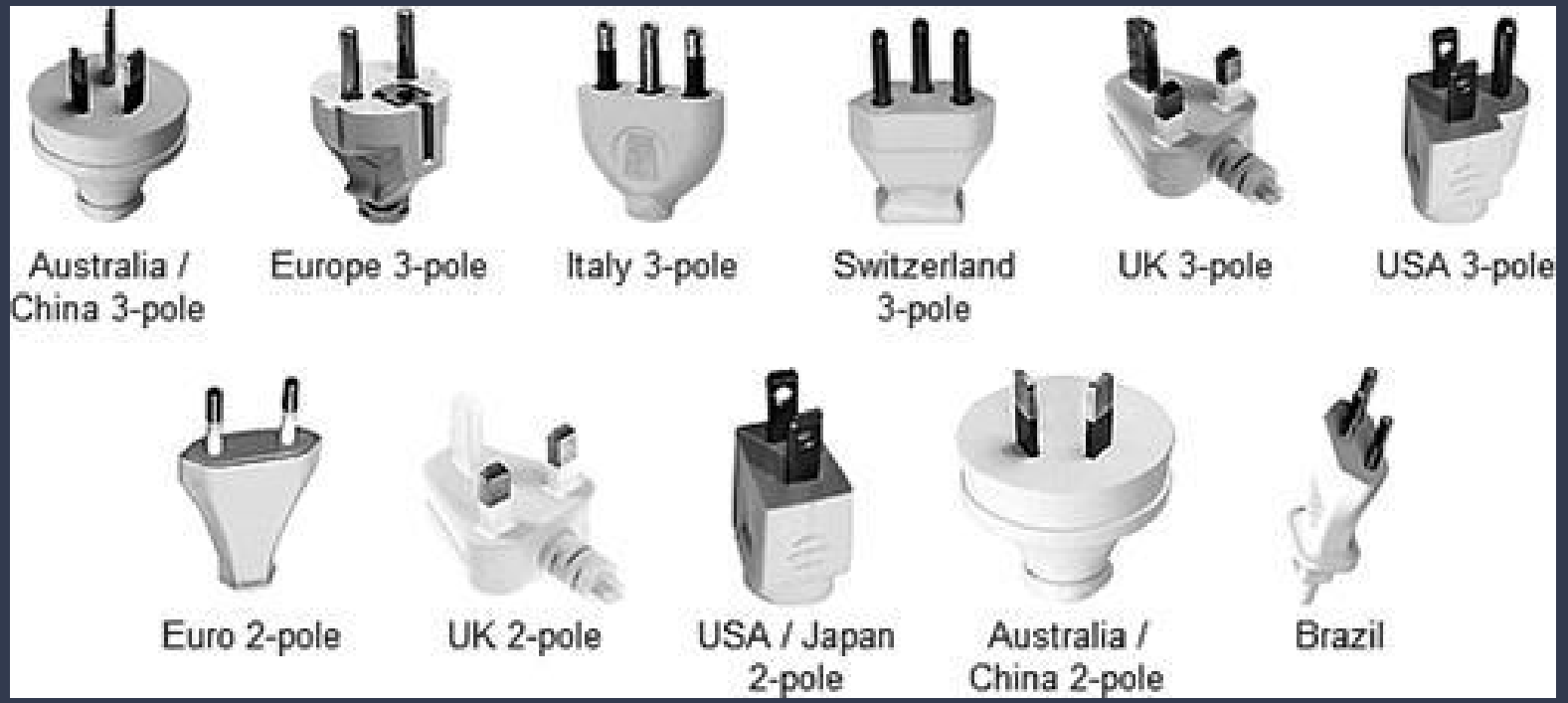
State of the world today

	A	B	C	D	E
1	Project Name	Containerize	Northbound Interface	Framework	Sponsor
2	AWS EBS	Yes	DVDI	REX-Ray	{code} by Dell EMC
3	AWS EBS	No	DVDI	Convoy	Rancher
4	AWS EBS	No	DVDI	OpenStorage	Portworx
5	AWS EBS	No	K8s In-Tree	K8s Volume Plugin	
6	AWS EBS	Yes	K8s Flexv	REX-Ray	{code} by Dell EMC
8	AWS EFS	Yes	K8s Flexv	REX-Ray	{code} by Dell EMC
9	AWS EFS	No	DVDI		
10	Azure Disk	No	DVDI	REX-Ray	{code} by Dell EMC
11	Azure Disk	No	K8s In-Tree	K8s Volume Plugin	
12	Azure File	No	K8s In-Tree	K8s Volume Plugin	
13	BeeGFS	No	DVDI		
14	Block Bridge	Yes	DVDI		Block Bridge
15	BTRFS	No	DVDI	OpenStorage	Portworx
16	Buse	No	DVDI	OpenStorage	Portworx
17	Ceph	No	DVDI	Contiv	Rancher
18	CephFS	No	K8s In-Tree	K8s Volume Plugin	
19	CephRBD	Yes	DVDI	REX-Ray	{code} by Dell EMC
20	CephRBD	No	K8s In-Tree	K8s Volume Plugin	
21	CephRBD	No	K8s Flexv	REX-Ray	{code} by Dell EMC
22	CIFS	No	DVDI		
23	CIFS		K8s Flexv		
24	Cinder	Yes	DVDI	REX-Ray	{code} by Dell EMC
25	Cinder	No	DVDI		
26	Cinder	No	K8s In-Tree	K8s Volume Plugin	
27	Cinder	No	K8s Flexv	REX-Ray	{code} by Dell EMC
28	CoprHD	No	DVDI	OpenStorage	Portworx
29	Device Mapper	No	DVDI	Contiv	Rancher
30	Diamanti		K8s Flexv		

5 plugins for AWS EBS being maintained

Variations of storage interface: Is this good for the community?

Users
Container Orchestrators
Storage Providers



CSI: Goals

The Container Storage Interface (CSI) is modeled on the successful OCI and CNCF sponsored CNI interoperability initiatives in the container and network space respectively.

Its goal is to provide a *vendor neutral*, curated specification that allows standardized storage plugins to be published and utilized across multiple container orchestrators, including Mesos and DC/OS.

CSI: Overview

- Control plane interface
  - CSI “steps aside” after wiring volume to container– not a bottleneck in the data IO plane
  - Flexible deployment
- Focus on volume lifecycle
  - Create
  - Publish/Unpublish (to nodes, to containers)
  - Destroy
- Service-oriented
  - Long running
  - gRPC; CO is a client of plugin services

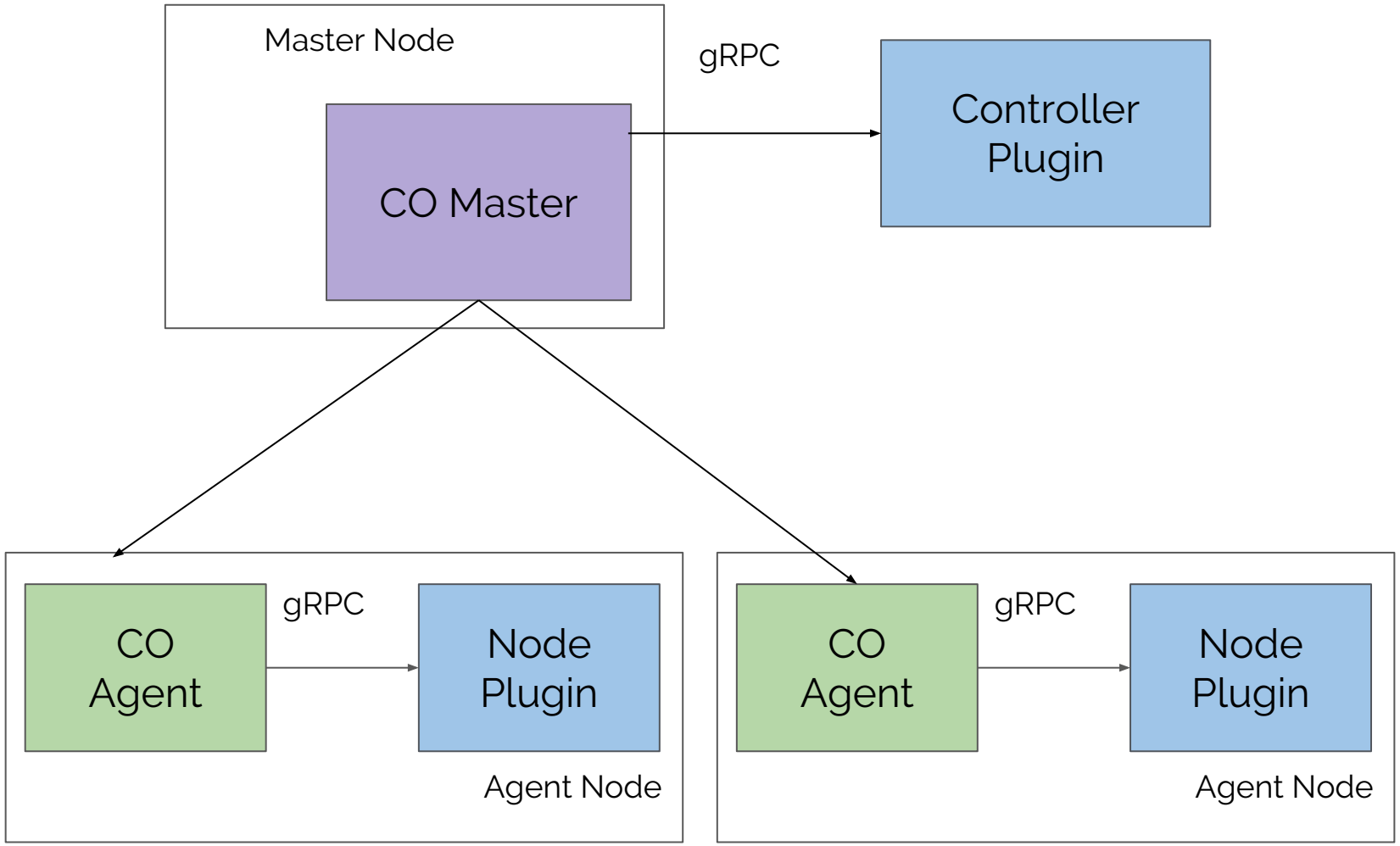
CSI: Configuration / Operation

- CSI spec focuses on *protocol* over operational concerns
- Minimal deployment requirements
  - gRPC endpoint as UNIX socket*
  - location via CSI_ENDPOINT envvar
- Packaging guidelines / recommendations (optional)
  - vendor implementations packaged as “plugins”
  - plugins should expect to be supervised
  - plugins should expect to be isolated

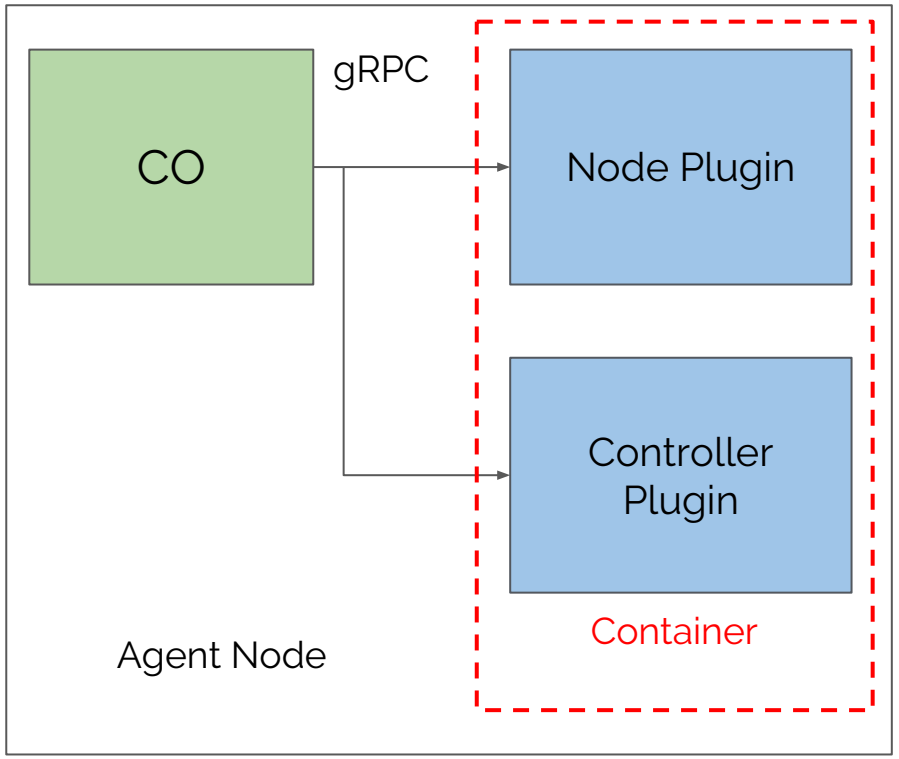
CSI: Plugin Composition

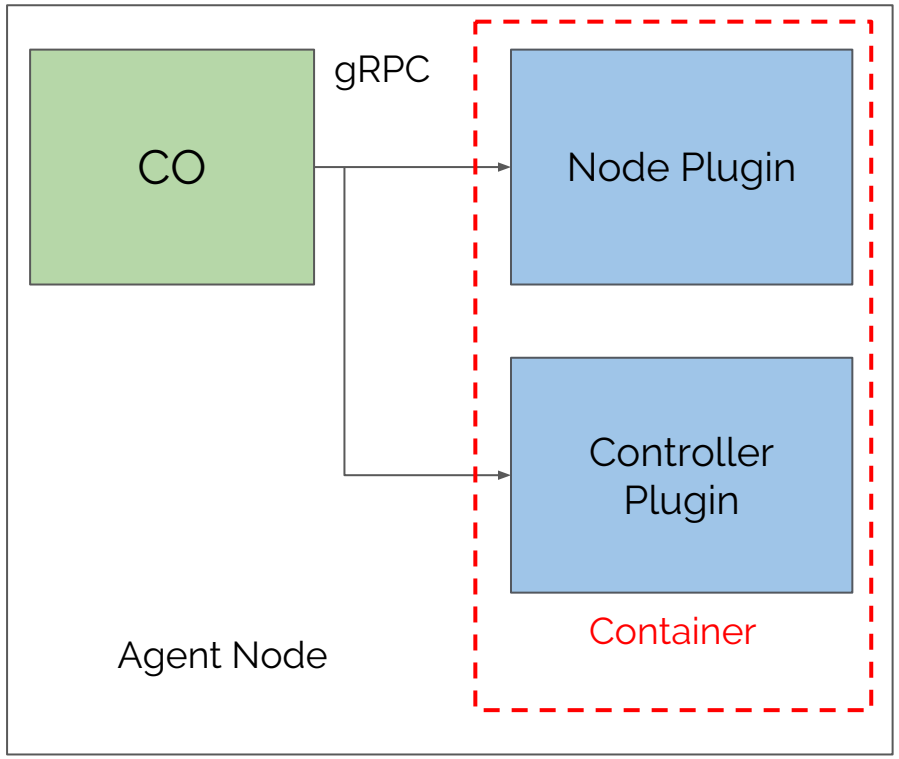
- 3 core gRPC services
  - Identity
  - Controller
  - Node
- Flexible composition
  - Identity+Controller+Node (headless)
  - Identity+Controller
  - Identity+Node

CSI integration: option #1



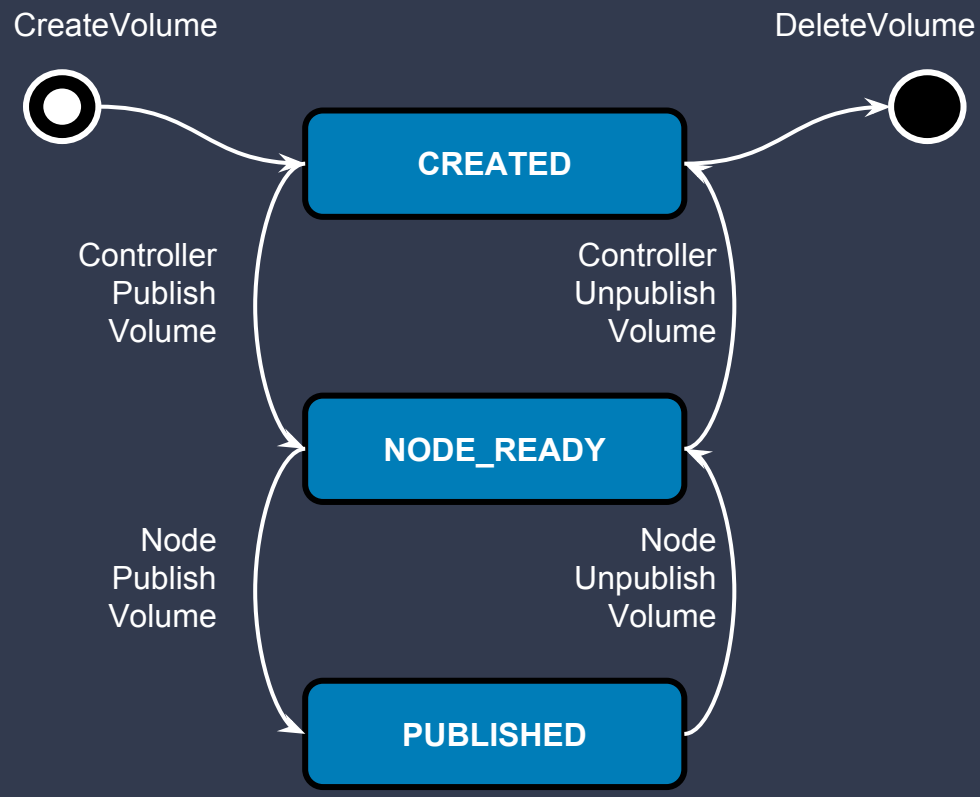
CSI integration: option #2





CSI: Volume Lifecycle

- CO provisions volumes
  - → CSI “attach to node”
  - → CSI “mount vol in CT”
- Plugins advertise support for lifecycle ops via *Capabilities
  - Create/Delete Volume
  - Controller Publish/Unpublish



CSI: Identity Service

- GetSupportedVersions
- GetPluginInfo

CSI: Controller Service

- ControllerGetCapabilities

- CreateVolume, DeleteVolume
- Controller { PublishVolume, UnpublishVolume }
- ListVolumes
- ValidateVolumeCapabilities
- GetCapacity

CSI: Node Service

- ProbeNode
- Node { PublishVolume, UnpublishVolume }
- GetNodeID
- NodeGetCapabilities

Mesos Integration with CSI

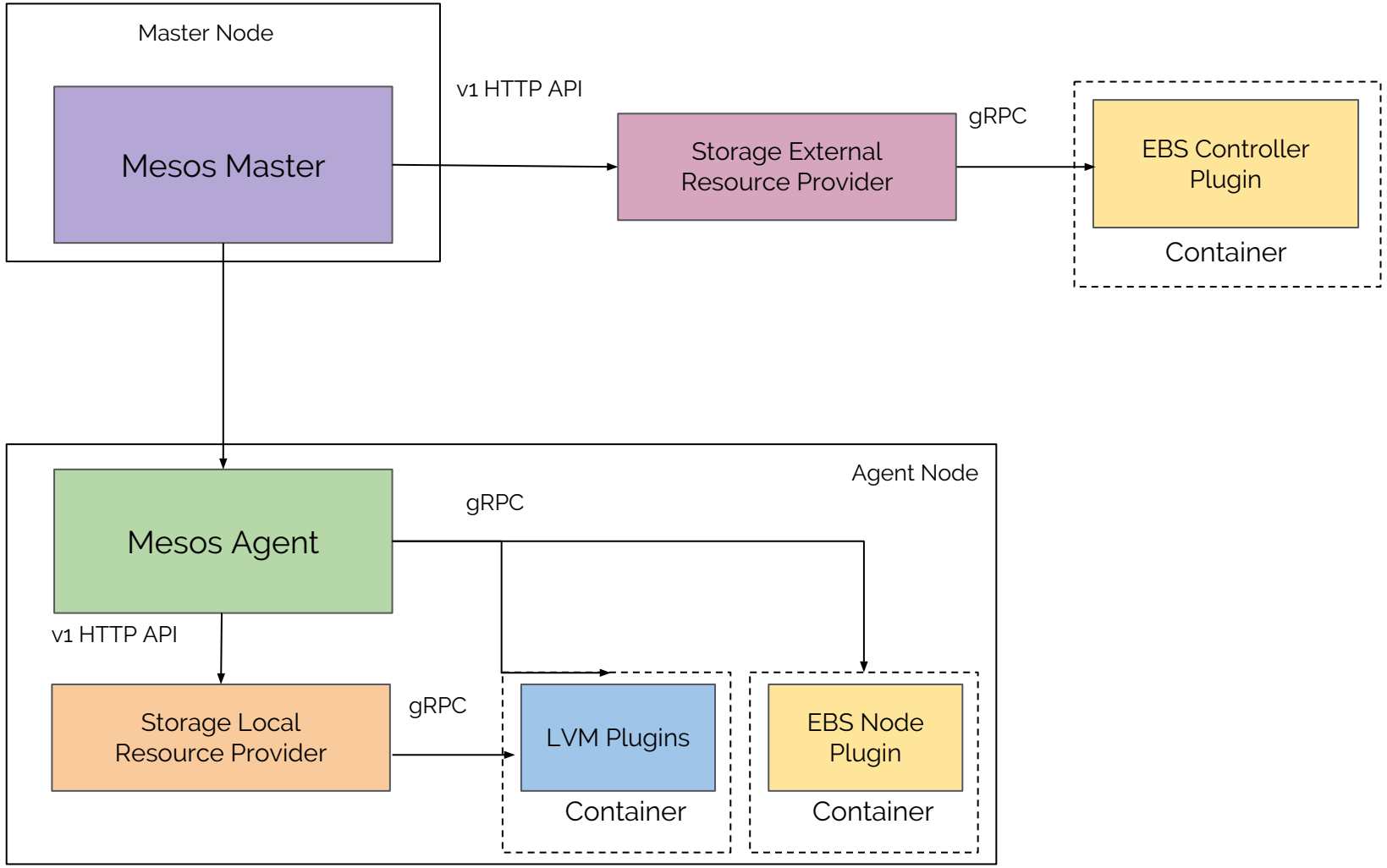
- New Concept: **Resource Provider (RP)**
  - An interface for providing resources to Mesos
  - Can be both *Local* and *External*
  - Agent can be viewed as a *Local RP*
- Why introduce RP?
  - Allow customization and extension on Resources
  - Support external resources (not tied to an agent)

Storage Resource Provider

- Introduce a first class Storage Resource Provider
  - Talk to CSI plugins
  - Expose “disk” resources
  - Handle operations (e.g., volume provisioning)

- Goal
  - Storage vendors just need to give Mesos the CSI plugin Docker image name, and Mesos will handle the rest.

Mesos CSI Integration (Mesos 1.5 & 1.6)



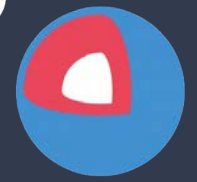
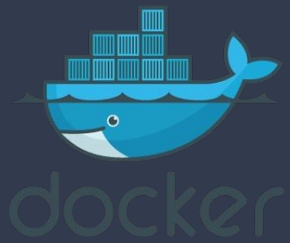
Mesos Roadmap on Storage Support

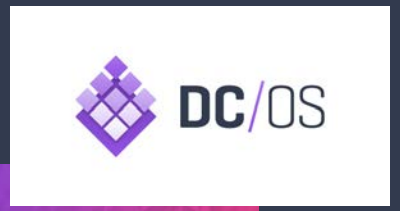
- Local Resource Provider (LRP) integration
- Storage LRP w/ CSI integration
- External Resource Provider (ERP) integration
- Storage ERP w/ CSI integration

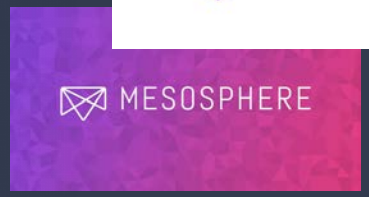
- Epic: https://issues.apache.org/jira/browse/MESOS-7235
- LRP support is targeted for Mesos 1.5
- ERP support is targeted for Mesos 1.6

Community: Who is involved with CSI








CSI Roadmap: Beyond intro release

CONTAINER STORAGE INTERFACE

Considering these - priority tbd, your feedback encouraged:

- Snapshot support
- Volume resizing
- Quota
- Windows OS/container support
- User ID & credential passthrough to storage provider

This is deemed out of scope - up to orchestrator platform to implement, differentiate

- Storage class (aka profiles)

Community: How to get involved

github: spec, sample code, issue tracking

- https://github.com/container-storage-interface

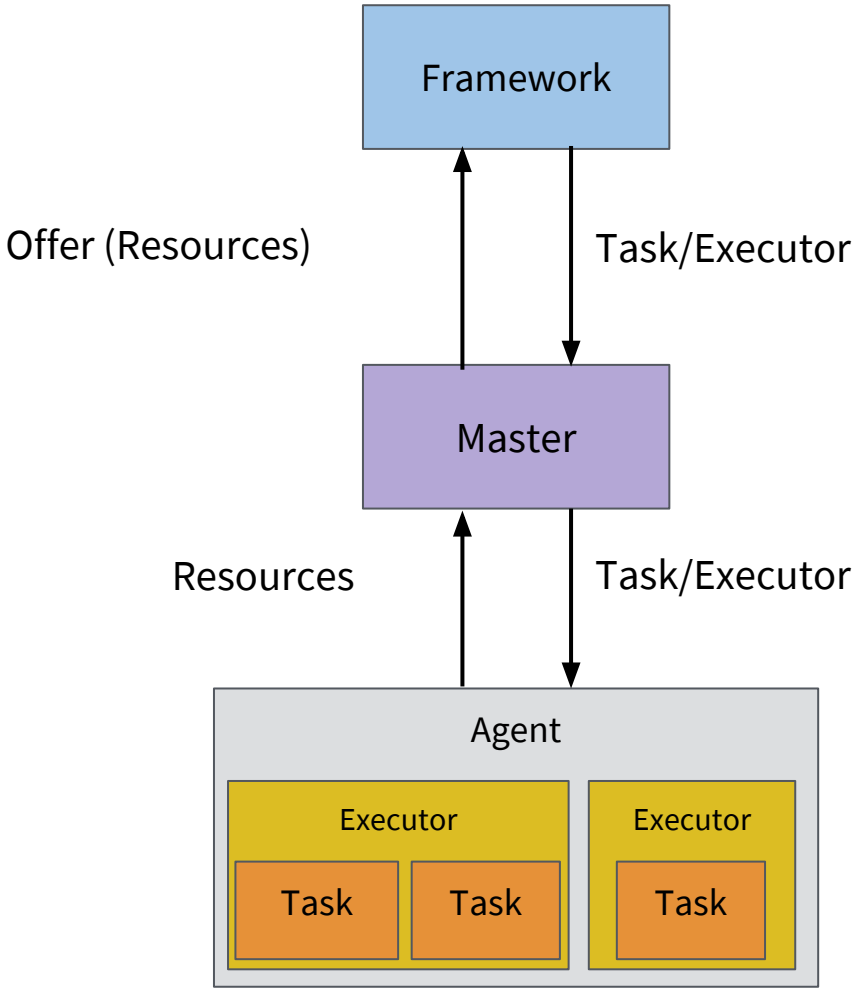
online 1 hour meeting every 2 weeks

- https://zoom.us/j/790748945
- notes: https://docs.google.com/document/d/1-oiNg5V_GtS_JBAEViVBhZ3BYVFlbSz70hreyaD7c5Y/edit#heading=h.h3flg2md1zg
- recorded, see notes for link

google+ group for mailing list communication

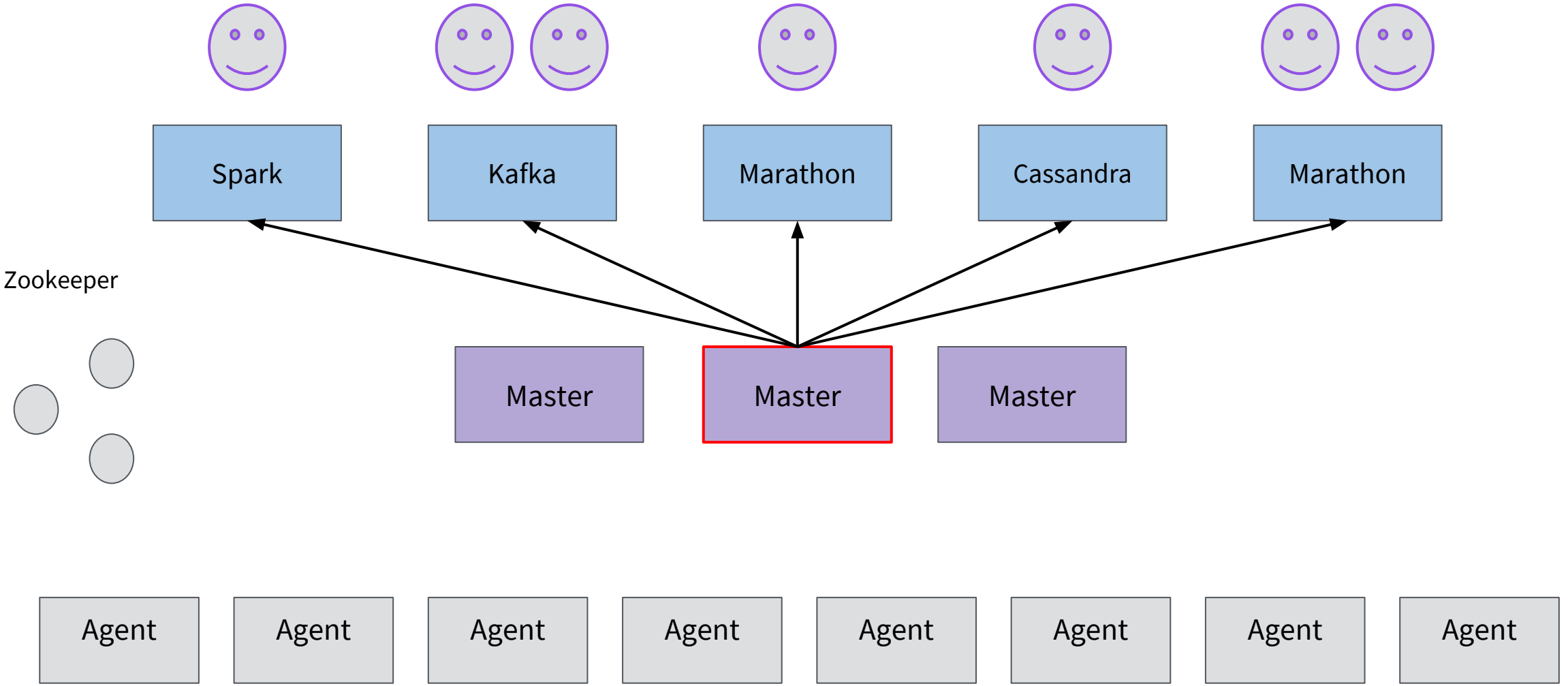
- container-storage-interface-community

Mesos programming abstraction

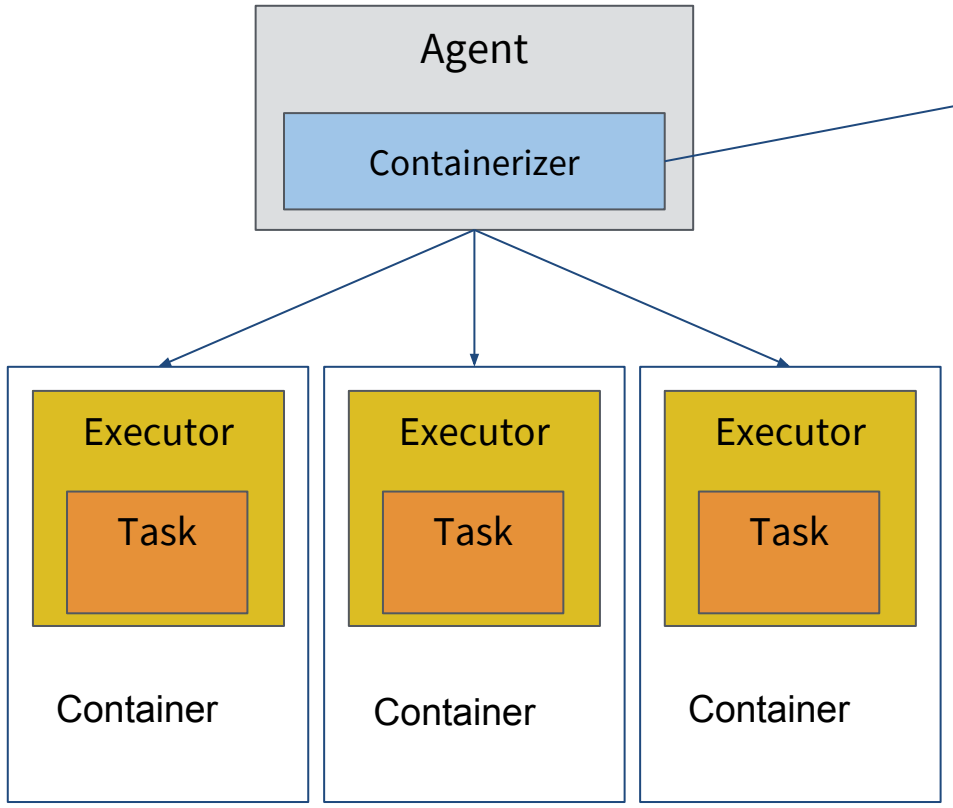


- Framework
- Resource/Offer
- Task
- Executor

A typical Mesos framework



Containerizer and isolators (0.18, 2014)



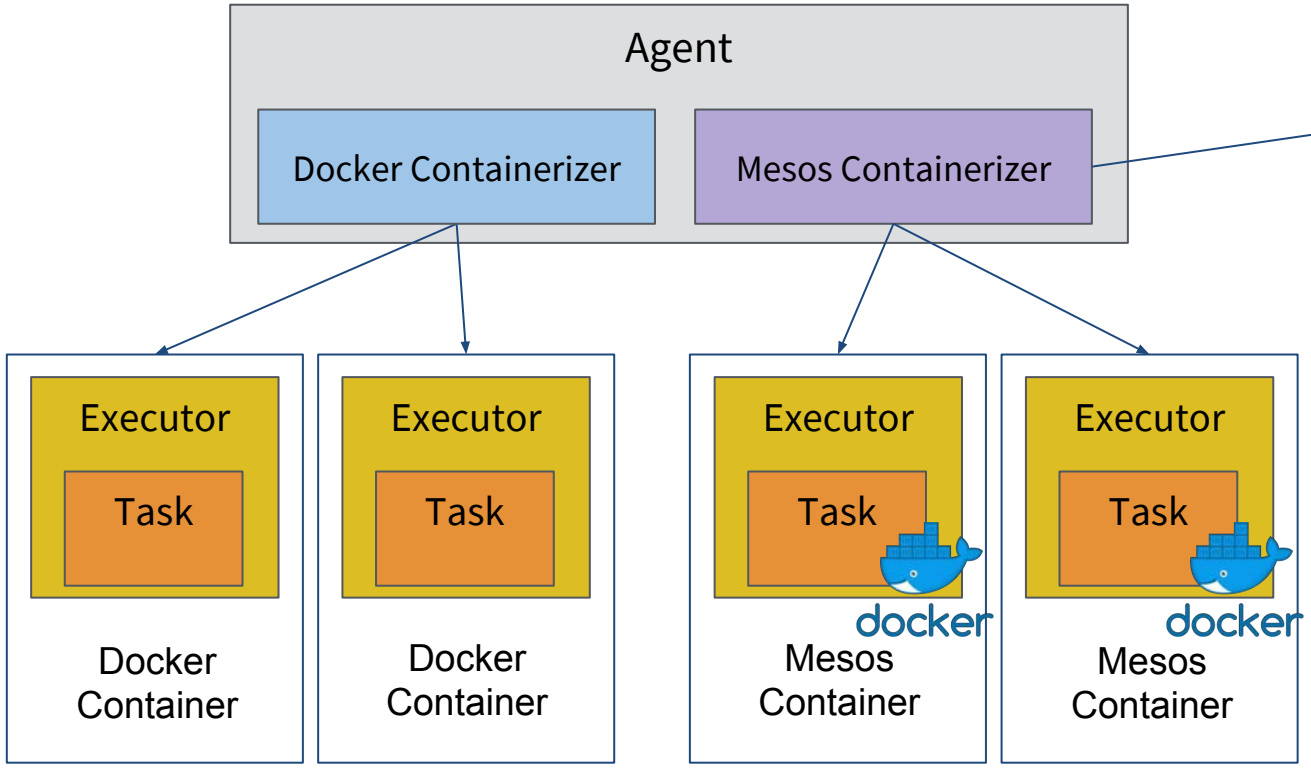
- **Pluggable architecture**
- **Isolators** (lifecycle hooks)
  - cgroups/cpu
  - cgroups/mem
  - ...
- **Launchers** (process mgmt)
  - linux (cgroups & ns)
  - posix
  - windows

Current list of isolators

- environment_secret
- appc/runtime
- cgroups/blkio
- cgroups/cpu
- cgroups/cpuset
- cgroups/devices
- cgroups/hugetlb
- cgroups/mem
- cgroups/net_cls
- cgroups/net_prio
- cgroups/perf_event
- cgroups/pids
- disk/du
- disk/xfs
- docker/runtime
- docker/volume
- filesystem/linux
- filesystem/posix
- filesystem/shared
- filesystem/windows
- gpu/nvidia
- linux/capabilities
- namespaces/ipc
- namespaces/pid
- network/cni
- network/port_mapping
- posix/cpu
- posix/mem
- posix/rlimits
- volume/host_path
- volume/image
- volume/sandbox_path
- volume/secret

https://github.com/apache/mesos/blob/master/docs/mesos-containerizer.md

Native Docker image support (0.28, 2016)



- **Isolators**
  - ...
  - volume/host_path
  - linux/capabilities
  - posix/rlimits
  - docker/runtime
  - ...
- **Launchers**
- **Provisioners**
  - Docker image provisioner
  - Appc image provisioner

docker

MESOS

Adopting container standards

- Container images
  - Docker
  - AppC
  - OCI image spec
- Container network
  - CNI
- Container storage
  - DVDI
  - CSI

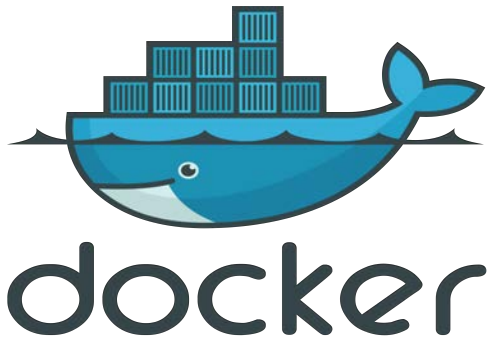
MESOS

Supported through pluggable interfaces in MesosContainerizer

De facto container standard

Google Cloud Platform

Registry API



Volume Plugin (DVDI)

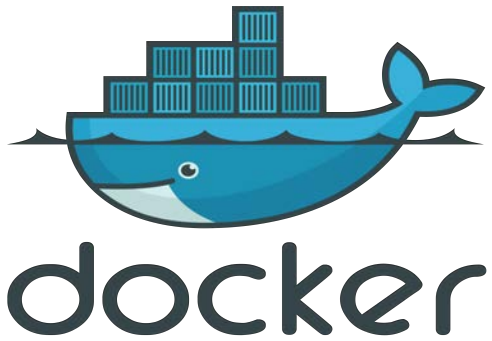
Network Plugin (libnetwork)

We need true container standards!

- Stable interfaces
- Backward compatibility
- Multiple implementations
- Vendor neutral
- Interoperability

Ideal world

Registry API

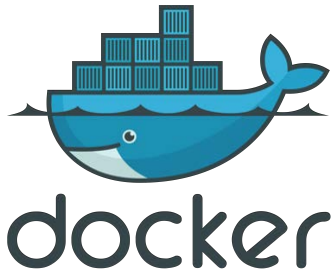


Volume Plugin (DVDI)

Network Plugin (libnetwork)

Ideal world

Registry API

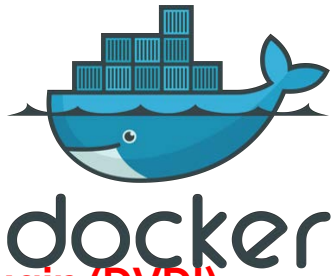


Volume Plugin (DVDI)

Network Plugin (libnetwork)

Ideal world

~~Registry API~~ → Container Image Spec



~~Volume Plugin (DVDI)~~
→ Container Storage Spec

~~Network Plugin (libnetwork)~~
→ Container Network Spec

Standards we need for containers

- Image
- Networking
- Storage
- Runtime
- Metrics
- ...

Standards we need for containers

- Image
- Networking
- Storage
- Runtime
- Metrics
- ...

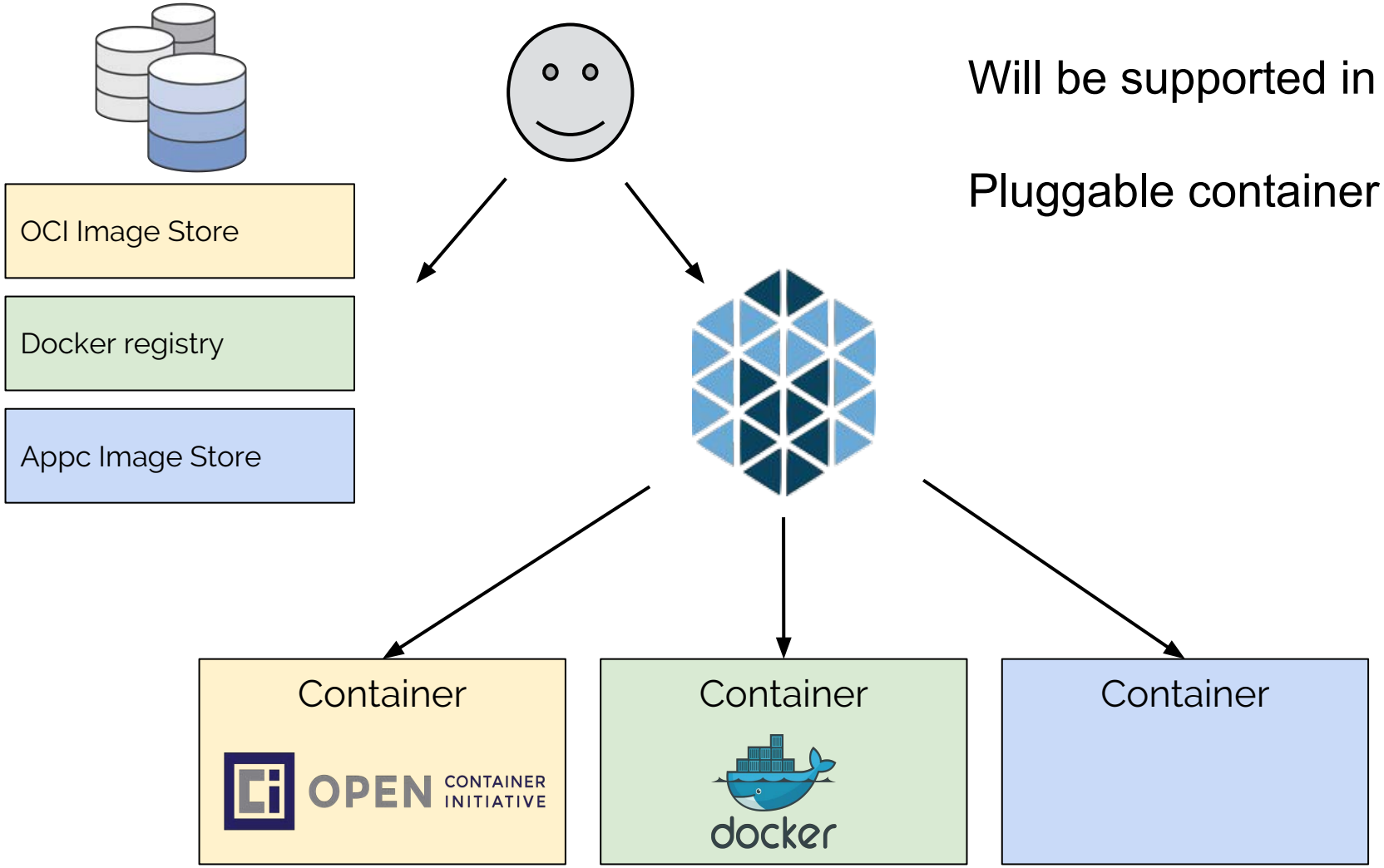
Container image spec

- Scope
  - How to package application bits into images
  - How to package application configs into images
  - How to store and transfer images
  - How to unpack images to get application bits and configs

OCI: Open Container Initiative

- OCI image spec
  - https://github.com/opencontainers/image-spec

Mesos will support OCI image spec (soon)



Will be supported in **MesosContainerizer**

Pluggable container image format

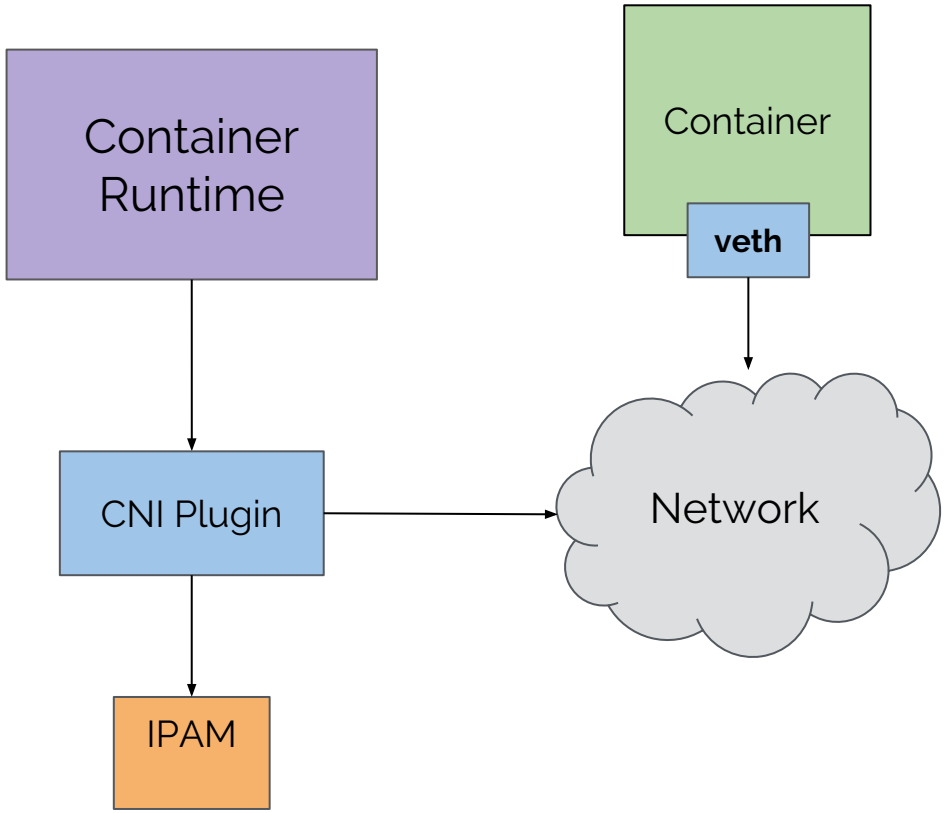
Container networking spec

- Scope
  - How to connect containers
  - How to allocate IP Addresses
  - How to enforce security policies
  - How to isolate performance
  - How to provide quality of service
  - How to balance network traffic

CNI: Container Networking Interface

- A simple CLI based interface
- Container orchestrator should invoke the CLI commands
  - Before container starts
  - After container terminates
- Adopted by major container orchestrators and network vendors
  - Recently joined CNCF
  - https://github.com/containernetworking/cni

CNI: Container Networking Interface



- Each plugin implements two CLI commands:
  - **ADD**: Attach network to the network namespace
  - **DEL**: Detach network from the network namespace
  - Pass config using arguments and environment variables

Mesos supports CNI

via an Isolator in MesosContainerizer:

--isolation=network/cni,...

CNI

PROJECT CALICO

weave

CISCO

cilium

JUNIPER NETWORKS

.....

Container storage spec

- Scope
  - How to Create/Destroy volumes
  - How to Attach/Detach volumes
  - How to Mount/Unmount volumes
  - How to create snapshots
  - How to restore snapshots

CSI: Container Storage Interface

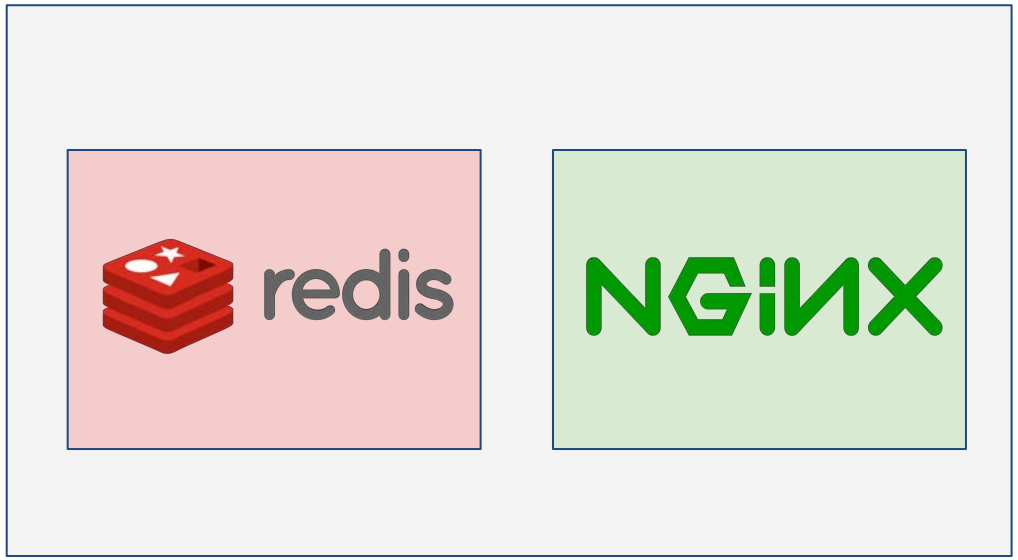
- Joint work between major container orchestrators
  - Mesos, Kubernetes, Docker, Cloud Foundry
  - https://github.com/container-storage-interface

- The goal of CSI in v1.0
  - One storage plugin works for all COs
  - Support dynamic provisioning
  - Support both local and remote storage
  - Support Mount and Block volumes

Highlighted new features

- General nesting support
- Remote debugging support

Why nested container?

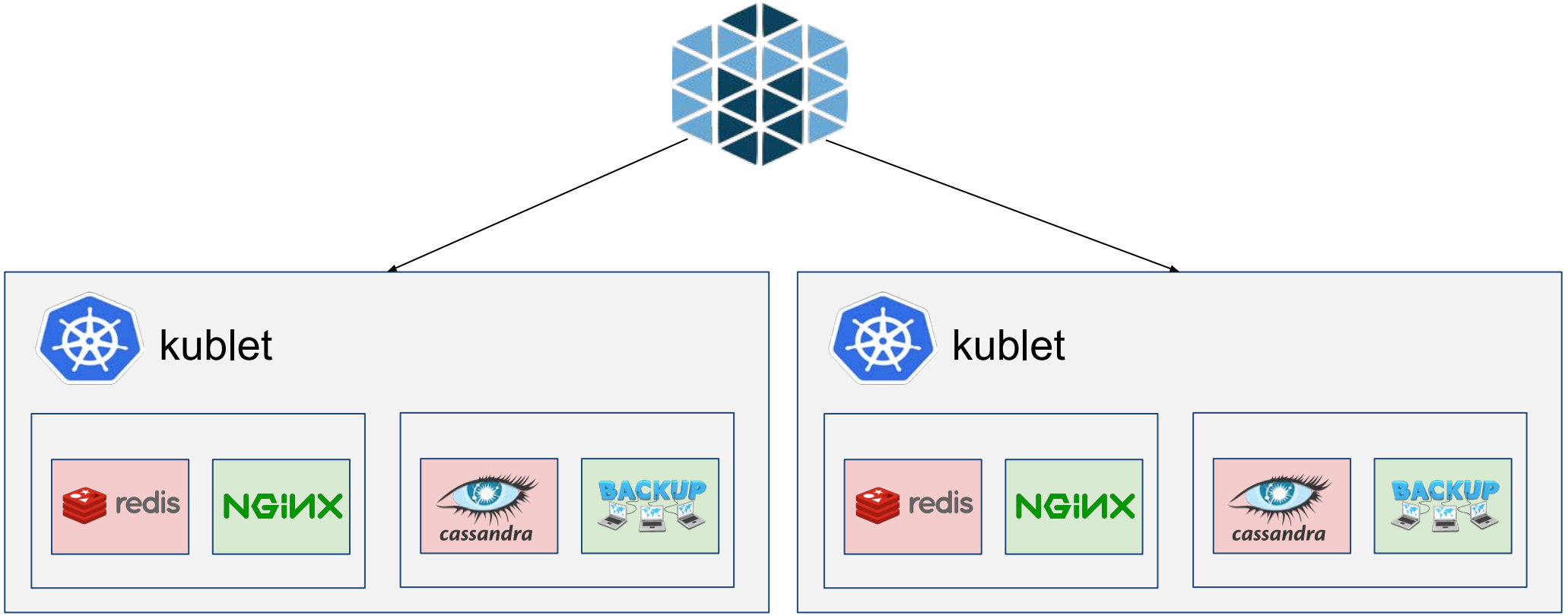


Sidecar pattern

Why nested container?

Transient Container

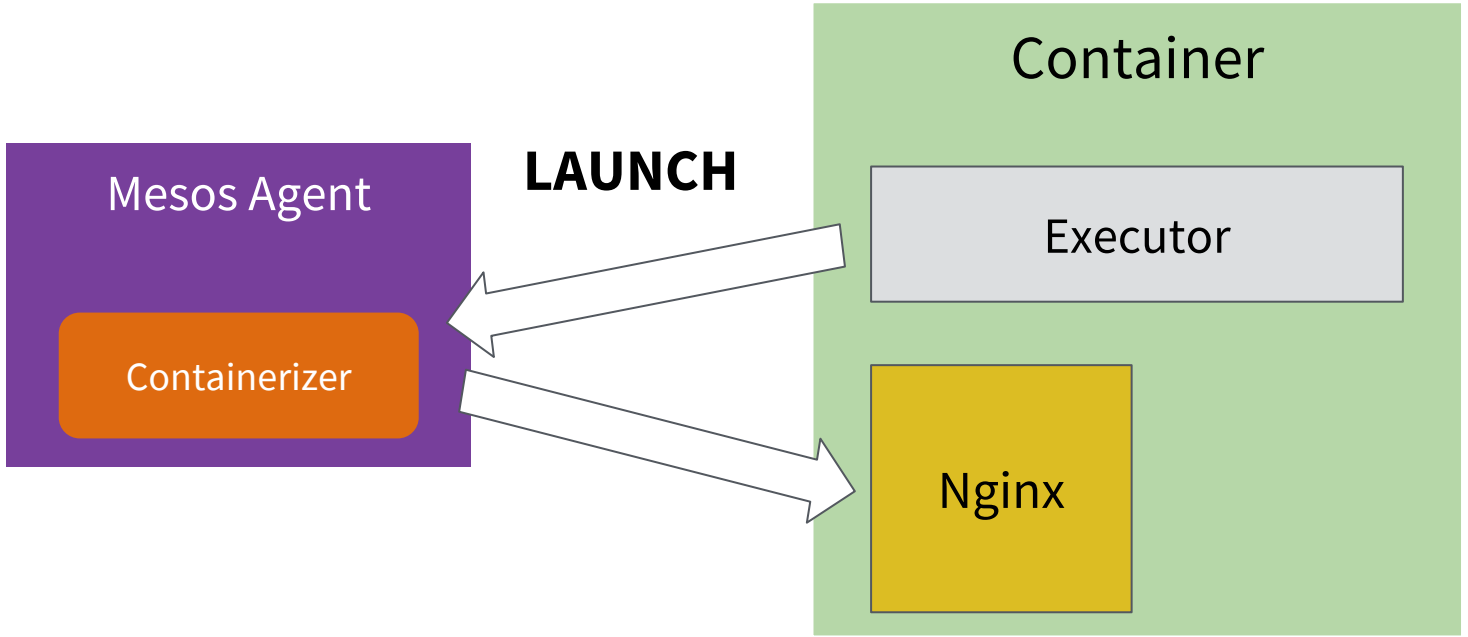
Why nested container?



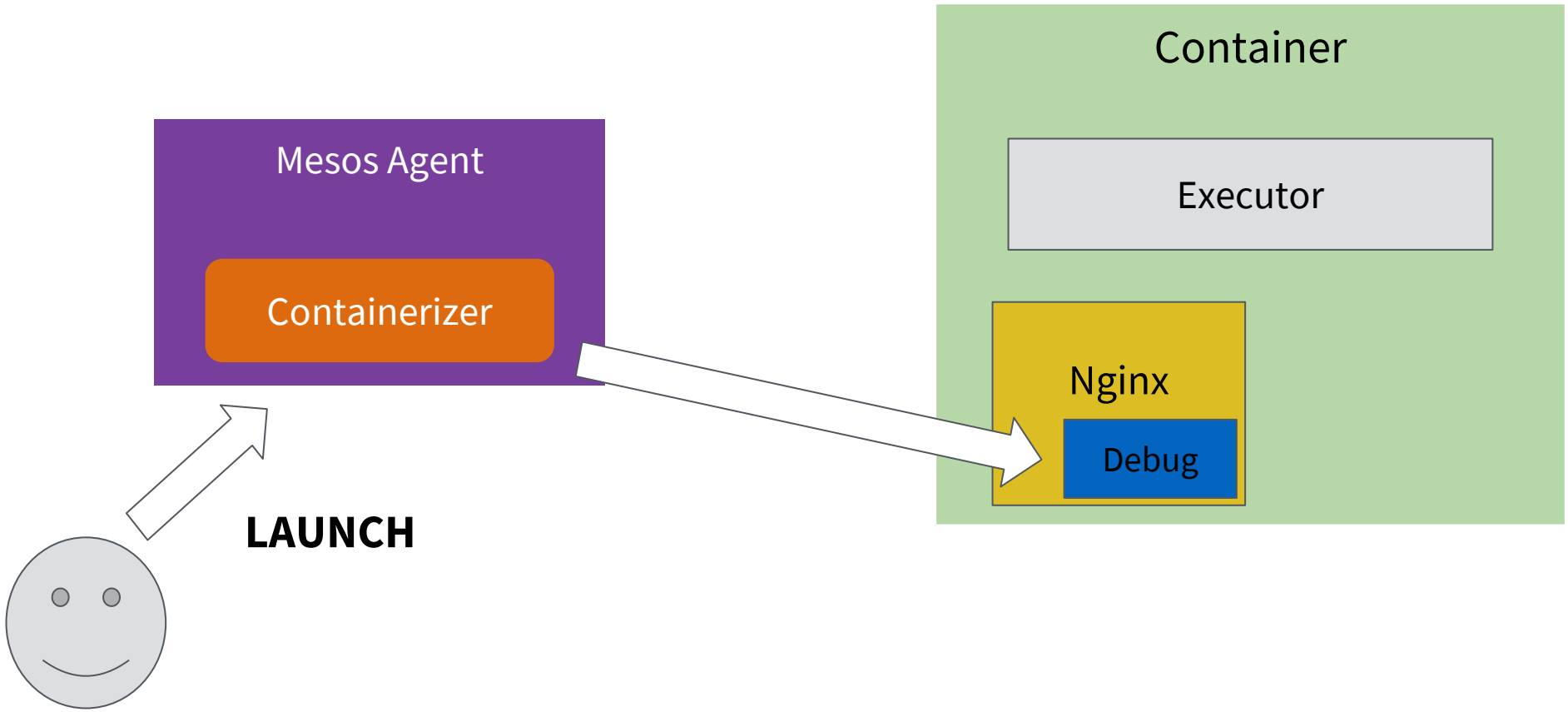
Hierarchical Container

MesosContainerizer supports nesting

- Depth > 2!
- Volume sharing with siblings
- Fully compatible with other features



Use nesting to support debugging!



Remote debugging support

- Similar to `docker exec` and `docker attach`, but can be done remotely
- Fully integrated with Mesos authn/authz
- Leverage nested container support

Future Roadmap

- Standalone mode
- Host port isolation
- PAM module support
- Unified artifacts store
- Seccomp and SELinux
- LXC support
- VM support
- User namespace
- ...

Summary

- Containerization in Mesos
  - Stable, in production for years
  - Option to not rely on Docker daemon
  - Pluggable and extensible
  - Embracing container standards