The SMACK Stack on Mesosphere DC/OS

Using Cloud Infrastructure

#OSCON

MESOSPHERE

2018

Kaitlin Carter

- ➔ Instructor & Content Developer at Mesosphere
- ➔ Develop Technical Trainings
- ➔ Instructional Designer

John Dohoney, Jr.

➔ Solution Architect at Mesosphere

➔ 10+ years in Digital Transformation Technologies

➔ 20+ years in Linux systems architecture

Agenda

DC/OS

Workshop Goals

Learn and understand:

- How to install, configure, and maintain SMACK Stack technologies on DC/OS.
- Benefits of using SMACK on DC/OS for data pipelines.

Gain hands on experience:

- Installing DC/OS with Ansible.
- Deploying a SMACK Stack.
- Deploying a application that uses the SMACK Stack.

Lab Environment

Your **lab environment** consists of **7 nodes**:

- **Bootstrap Node**: DC/OS CLI and Bastion host.
- **Master Node**: Controls the cluster.
- **Public Agent Node**: Facilitates communication from outside the cluster to the services running in the cluster.
- **Private Agent Nodes x4**: The nodes where our deployed services will run.

Lab Instructions:

- https://github.com/mesosphere/oscon-smack-stack

Raffle!

To participate:

- Email us confirming at education@mesosphere.com

Raffle Rules:

- There is a 1st and 2nd place.
- You can only enter once.
- Winners announced at the end of today’s session - must be present.

DC/OS

Raffle

1st Prize:

- Star Wars Legos
- Swag bag

2nd Prize:

- Predator 3 Drone
- Swag bag

Intro to SMACK Stack:

- History of Big Data, Slow Data, and Fast Data
- Motivation & Problems Solved
- Intro to SMACK

Fast Data: Historical Context

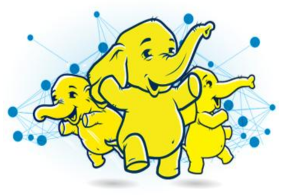
Days *Hours* *Minutes* *Seconds* *Microseconds*

Batch → **Micro-Batch** → **Event Processing**

Reports what has happened using descriptive analytics

Solves problems using predictive and prescriptive analytics

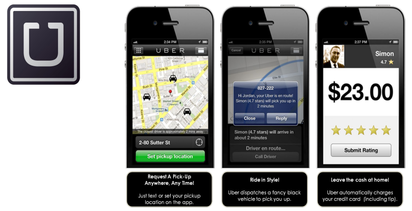
Billing, Chargeback



Product recommendations

Customers Who Bought This Item Also Bought
Black Friday! Loctek Curved Panel UHD HD TV Wall Mount Bracket Articulating Arm Swivel &
AmazonBasics High-Speed HDMI Cable - 6.5 Feet (2 Meters) Supports Ethernet, 3D, 4K and Audio Return
Cheetah Mounts APTMM2B TV Wall Mount for 32-65" TVs (Many from 20-75") Specifically fits up

Real-time Pricing and Routing

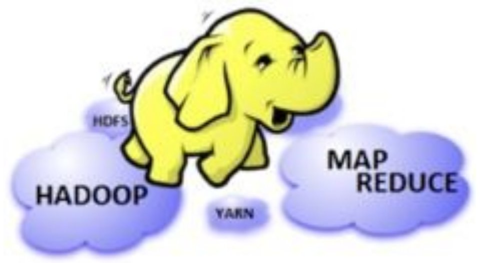


Real-time Advertising

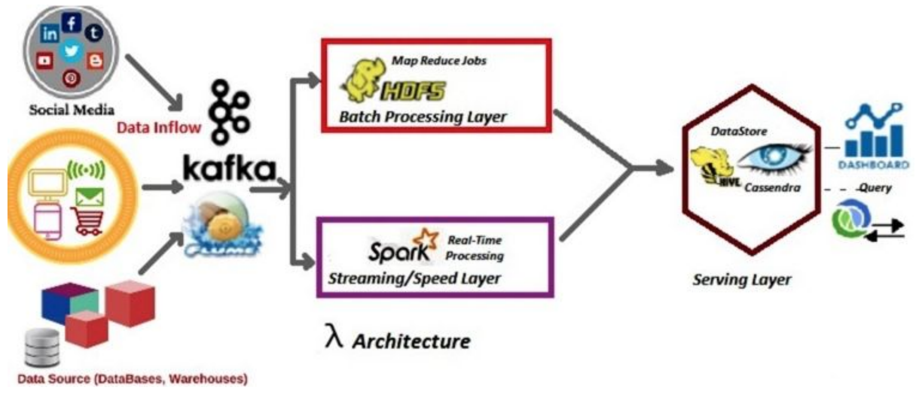
Predictive User Interface

DC/OS

Recent Data Architectures



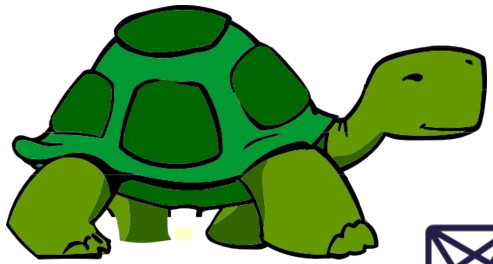
- Architectures affecting Digital Transformation
- Hadoop Map-Reduce
  - Slow Data Pattern
- Lambda Architecture – *SMACK Stack application*
  - Bridge Between
    - Slow Data
    - Fast Data
- FAST Data Architecture – *SMACK Stack application*



DC/OS

What is "Slow Data"

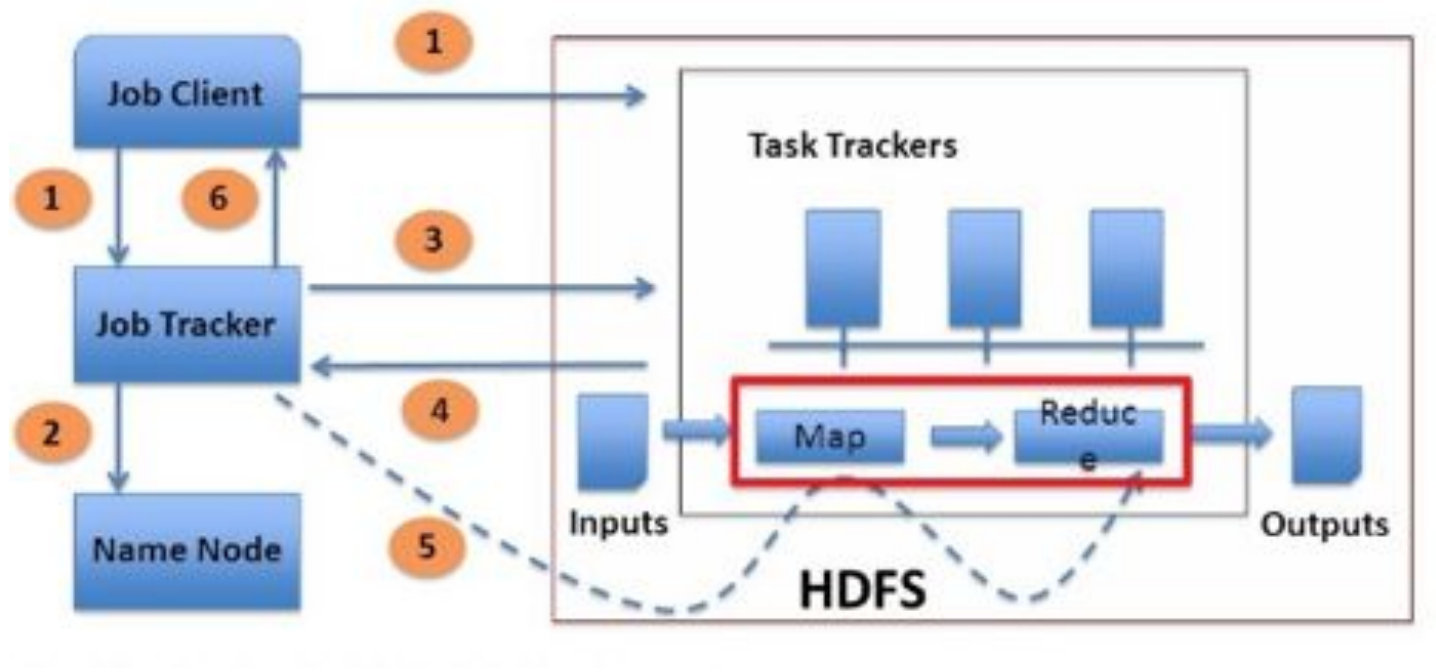
- Slow Data is captured as part of a business process with no intention of its usage, intrinsic value for trends, and in some cases its presence is only a status symbol with no corporate value.
- Can not be enriched, can not be combined, and usually not de-normalized – think about it…
- Lives/Resides in "glaciers", "lakes", and "warehouses" and in most case if lost or deleted there is little consequence – perhaps with the exception of compliance retention
- Not capable of streaming – the delta is not that interesting, the rate of change, nor the patterns of change



DC/OS

Hadoop MapReduce

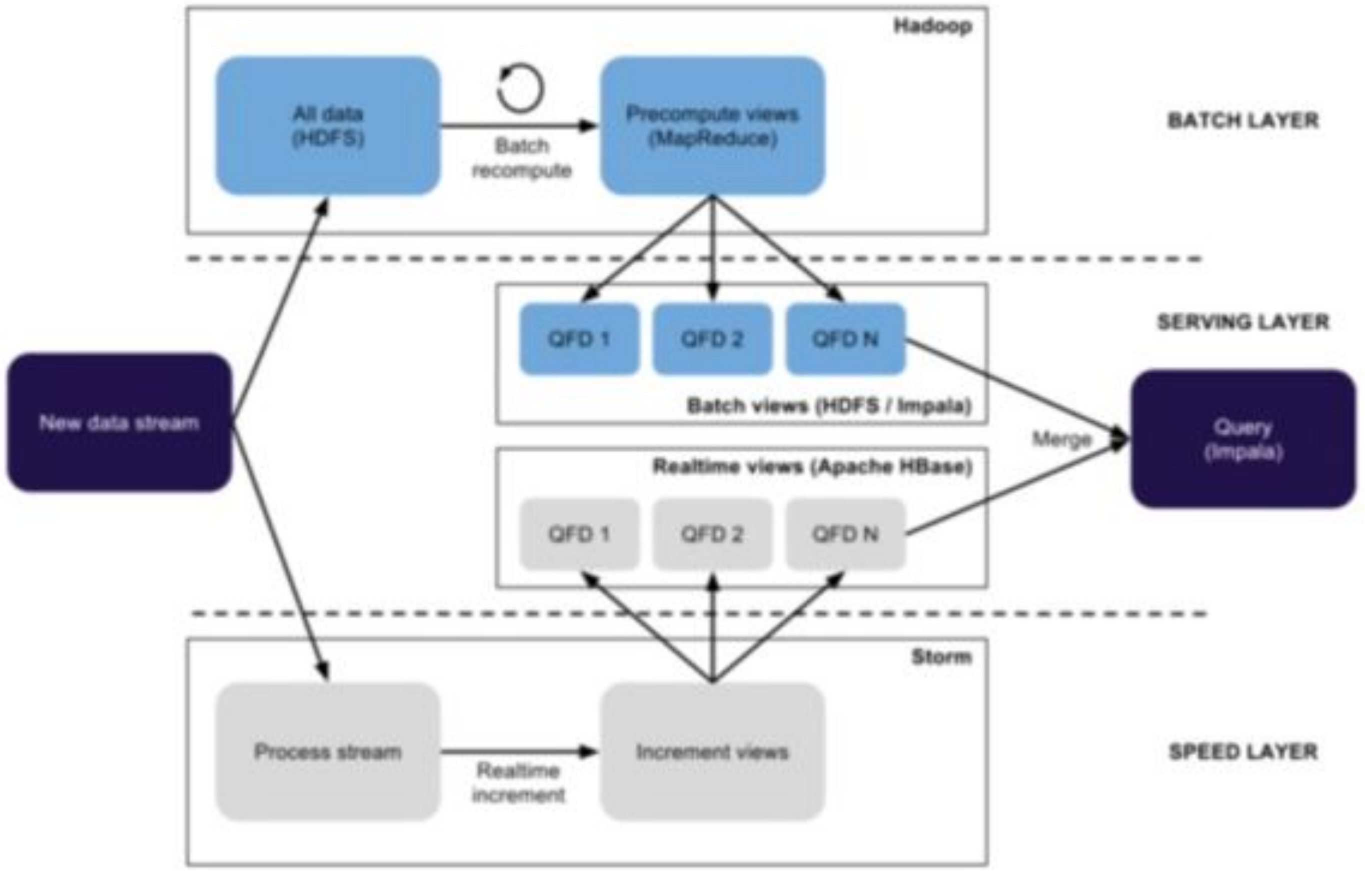
1. Job Submitted
2. Job queries HDFS Name-Node(s) to find data
3. Job Tracker creates execution plan and submits to Task Trackers
4. Task trackers perform task and report status to Job Tracker
5. Job Tracker manages task phases
6. Job Tracker finished task and updates status



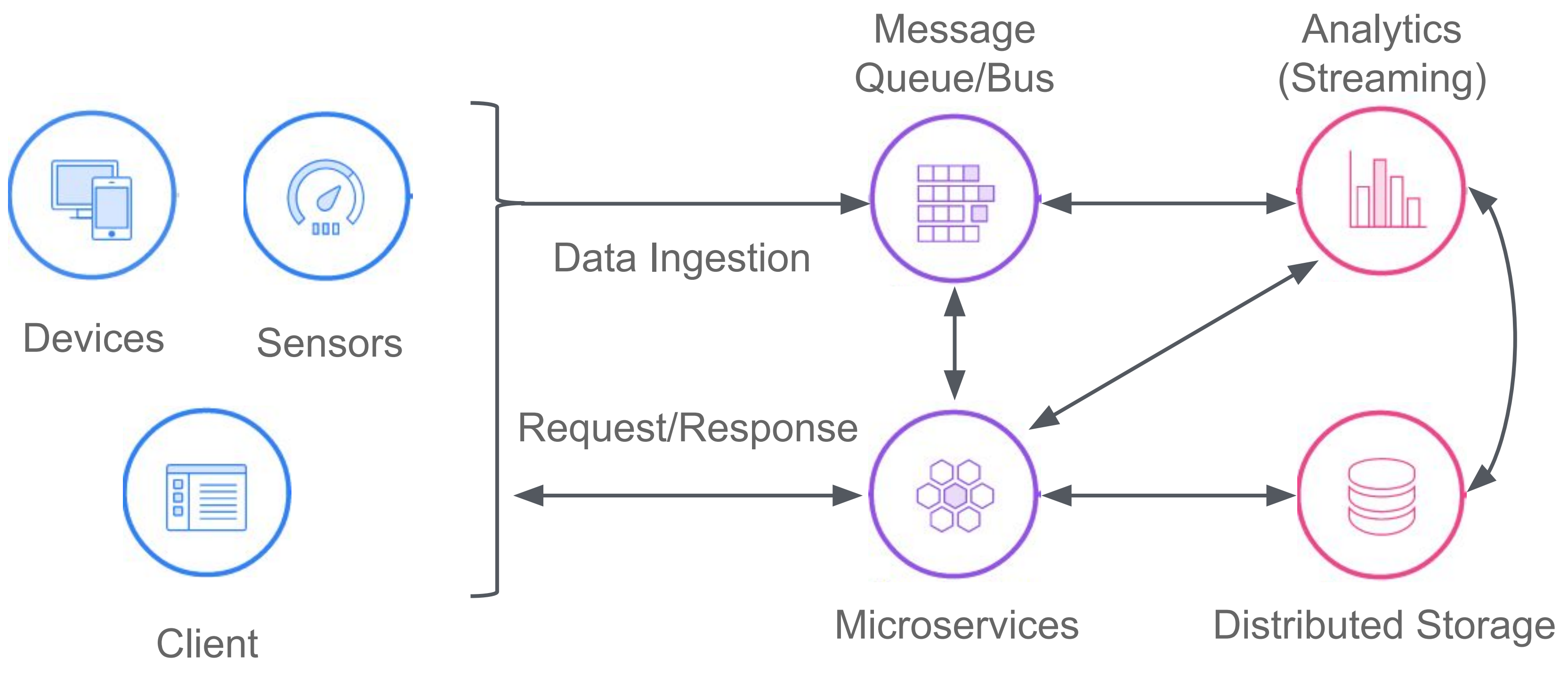
DC/OS

Architecture

- Transitional Architecture in many cases
- Used in an enterprise where Slow and Fast data exist
- SMACK, or “SMACK-Like” Stack used to implement system



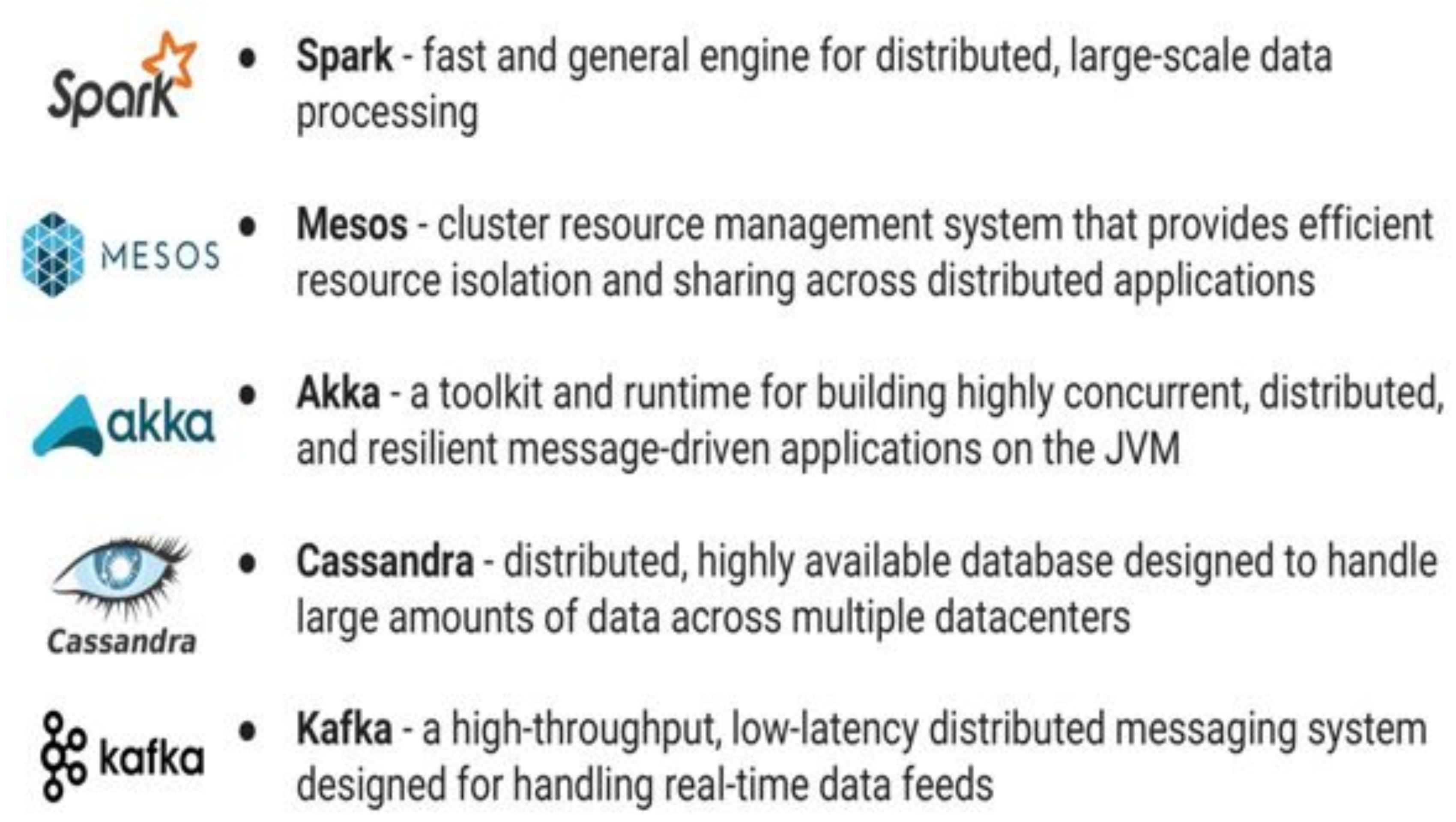
Modern Application -> Fast Data Built-in



Use Cases:

- Anomaly detection
- Personalization
- IoT Applications
- Predictive Analytics

The SMACK Stack is based on...



- **Spark** - fast and general engine for distributed, large-scale data processing
- **Mesos** - cluster resource management system that provides efficient resource isolation and sharing across distributed applications
- **Akka** - a toolkit and runtime for building highly concurrent, distributed, and resilient message-driven applications on the JVM
- **Cassandra** - distributed, highly available database designed to handle large amounts of data across multiple datacenters
- **Kafka** - a high-throughput, low-latency distributed messaging system designed for handling real-time data feeds

Why SMACK Stack...

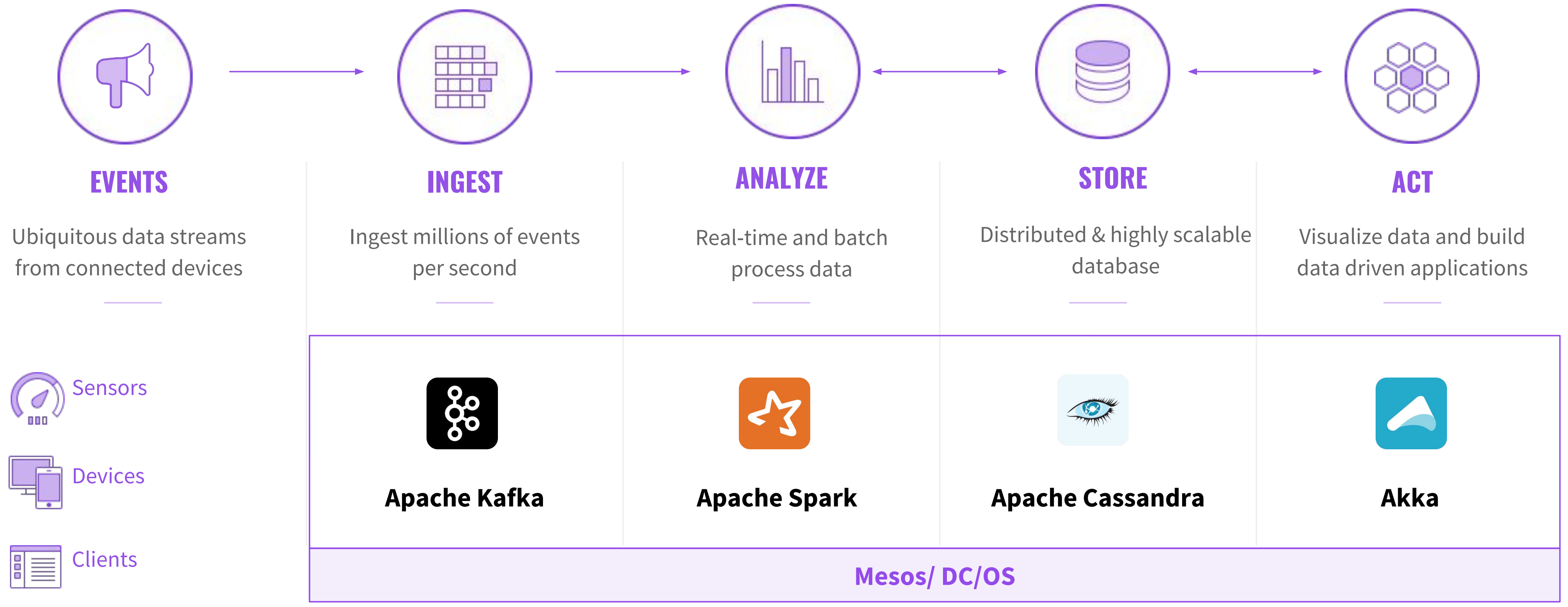
- It is a toolbox for many data processing architectures
- It has been “Battle-Tested” and used in many industry verticals
- Probably the shortest path to Minimum Viable Product (MVP)
- Proven to easily be scalable and highly elastic
- SMACK is a single platform for many kinds of applications
- Is well suited for deployment as a unified cluster management for a diversity of workloads

Success Model

Spark
MESOS
akka
Cassandra
kafka

- Shortest path to Minimum Viable Product (MVP)
- Battle-Tested, Scalable and already designed for Cloud Native

In review, the SMACK Stack is ...



EVENTS: Ubiquitous data streams from connected devices

INGEST: Ingest millions of events per second

ANALYZE: Real-time and batch process data

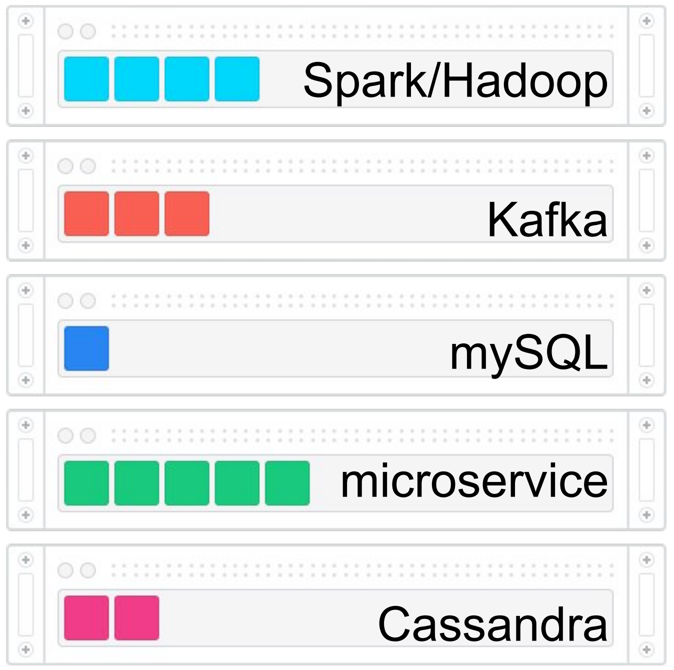
STORE: Distributed & highly scalable database

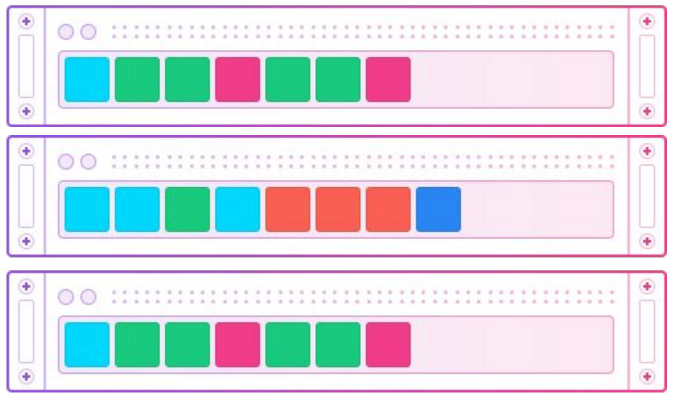
ACT: Visualize data and build data driven applications

Intro to DC/OS:

- Core Concepts
- DC/OS Architecture
    - Containers & Container Orchestration
    - Interacting with DC/OS & the DC/OS Catalog
    - Mesos

Multiplexing of Data, Services, Users, Environments



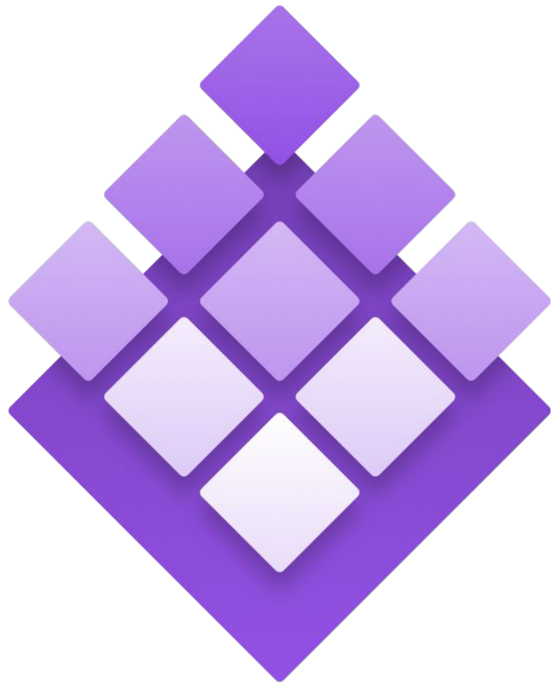


Typical Datacenter
siloed, over-provisioned servers,
low utilization

Apache Mesos
automated schedulers, workload multiplexing onto the
same machines

DC/OS is...

- 100% open source (ASL2.0)
  - + A big, diverse community
- An umbrella for ~30 OSS repos
  - + Roadmap and designs
  - + Documentation and tutorials
- Familiar, with more features
  - + Networking, Security, CLI, UI, Service Discovery, Load Balancing, Packages, ...



DC/OS

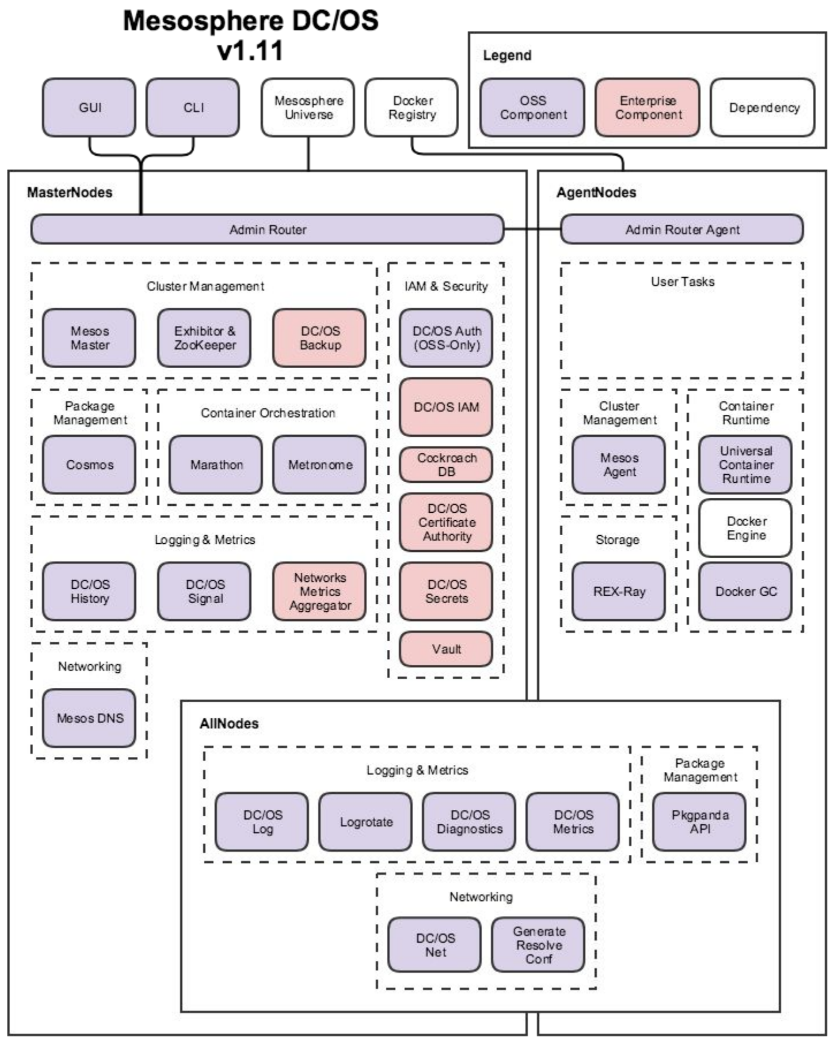
Quick Knowledge Check

Is the mesos component in DC/OS also the foundational technology in the SMACK stack?

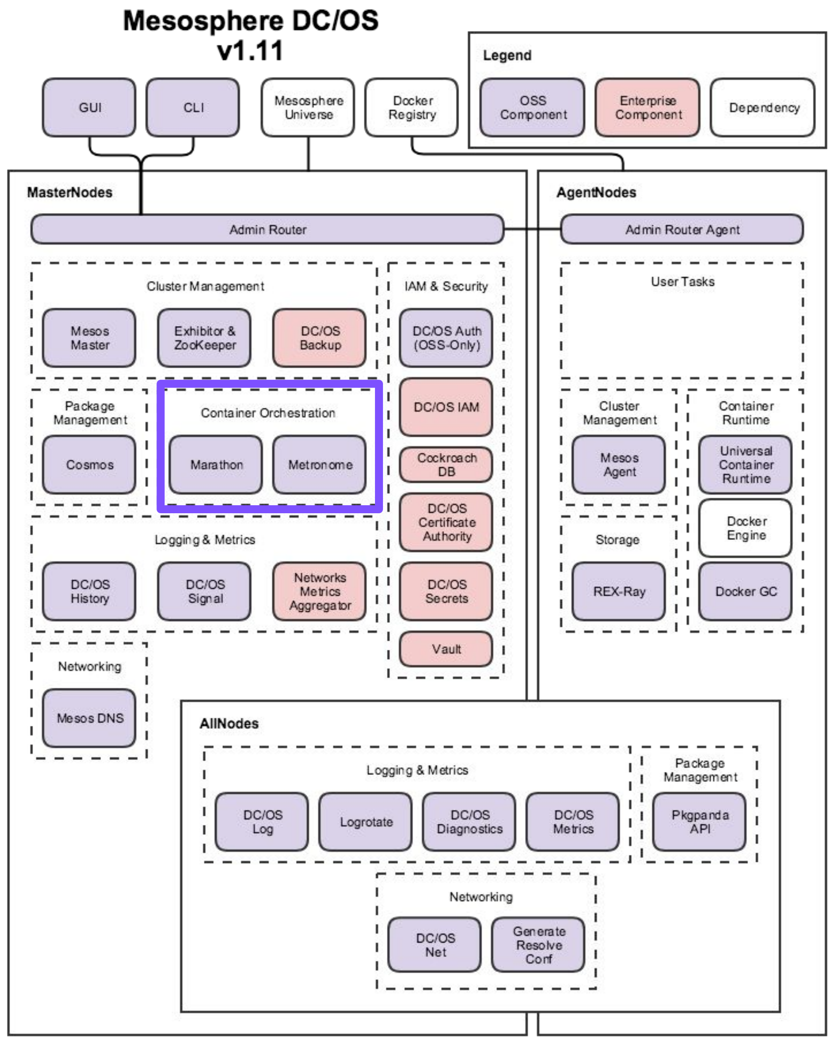
DC/OS Brings it All Together

- Resource management
- Task scheduling
- Container orchestration
- Logging and metrics
- Network management
- “Universe” catalog of pre-configured apps
- And much more https://dcos.io/

DC/OS Architecture Overview: DC/OS Components



DC/OS Architecture Overview



Containers: Docker

- Rapid deployment
- Some service isolation
- Dependency handling
- Container image repository

DC/OS

Containers: Runtime

Docker Engine

- Docker images only
- Must be installed on all cluster nodes.

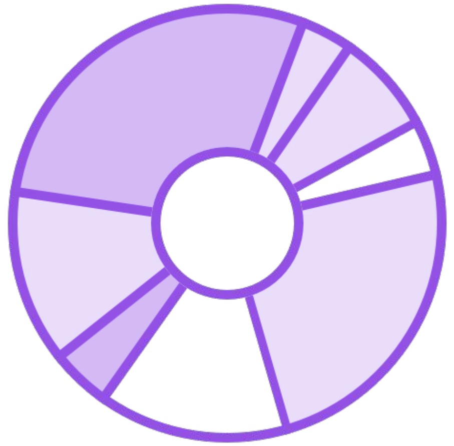
UCR

- Docker images
- Mesos containers
- GPU & CNI support
- Installs with DC/OS

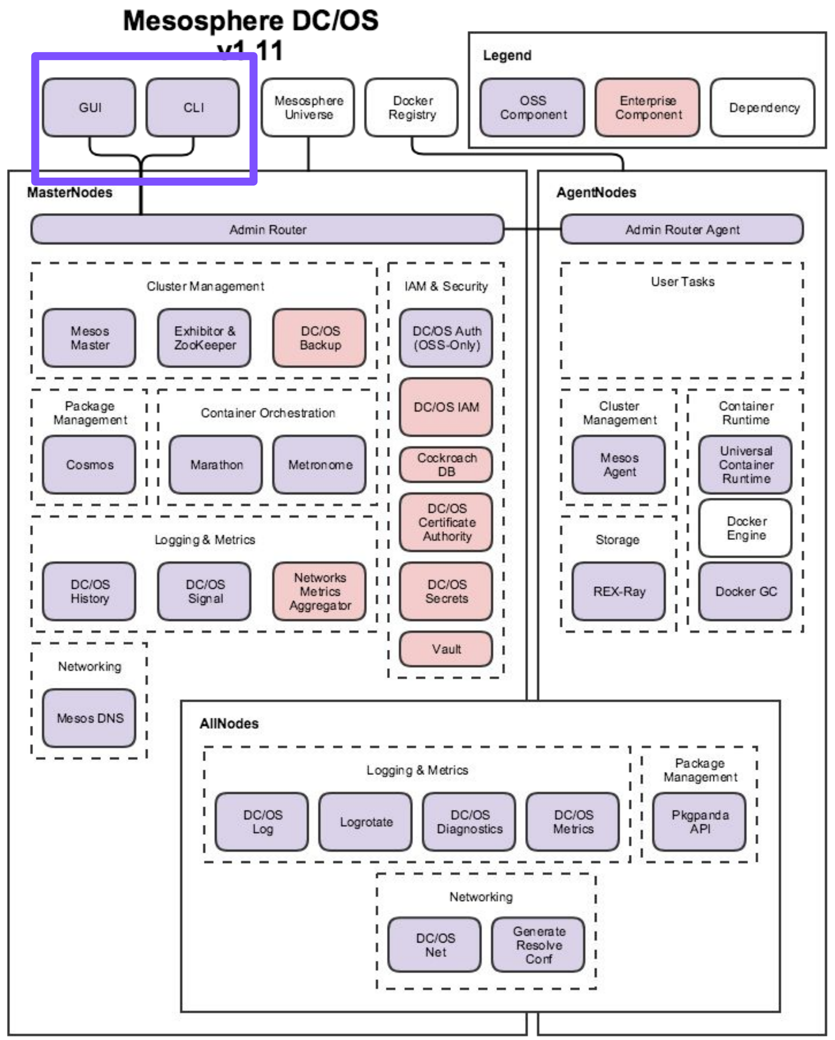
DC/OS

Containers Orchestration: Marathon

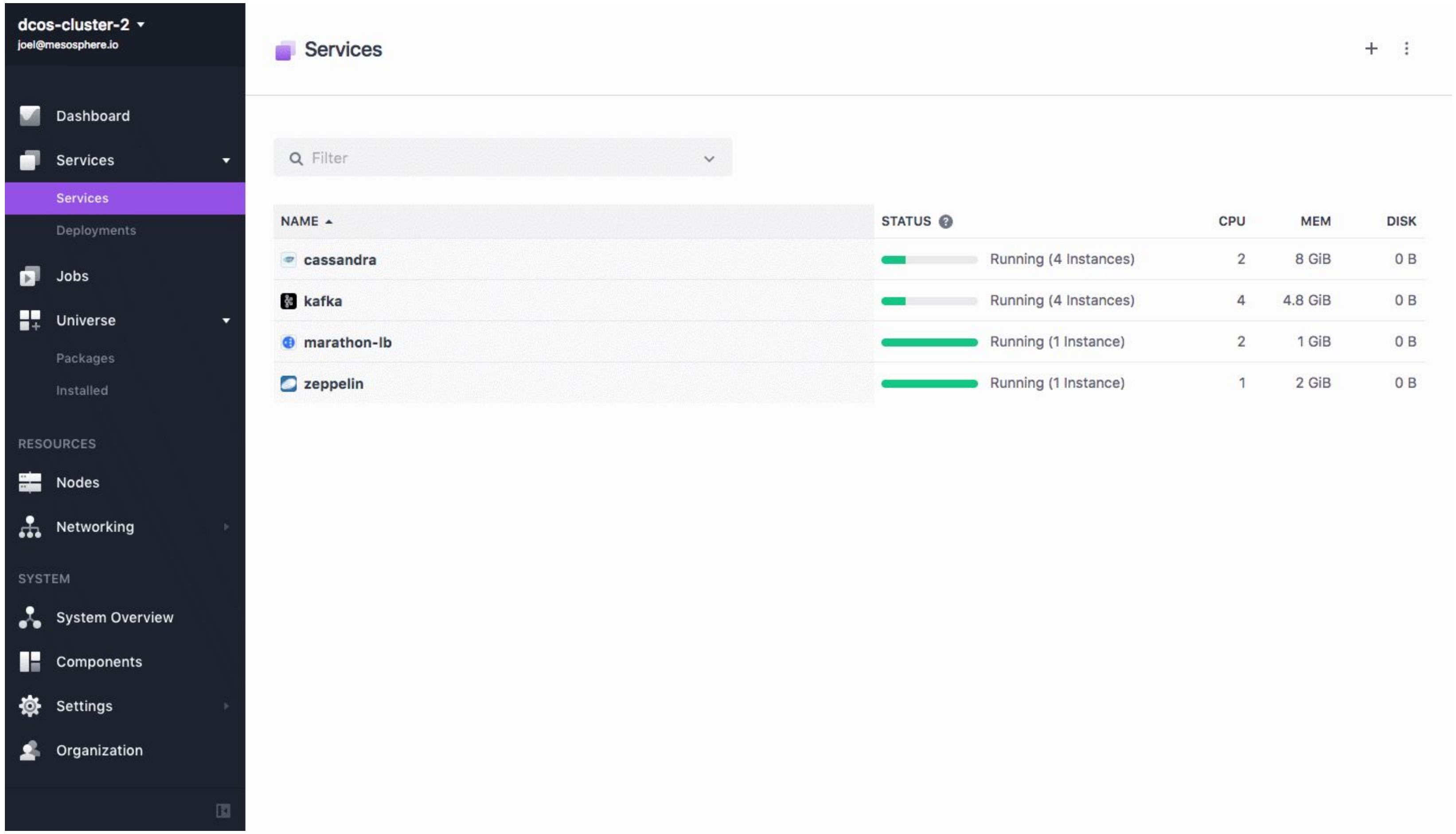
- Built-in scheduler for long-running services and Mesos frameworks.
  - Starts and keeps applications running.
  - Similar to a distributed init system.
- A Mesos framework is a distributed system that has a scheduler.
- Mesos mechanics are fair and HA.



DC/OS Architecture Overview

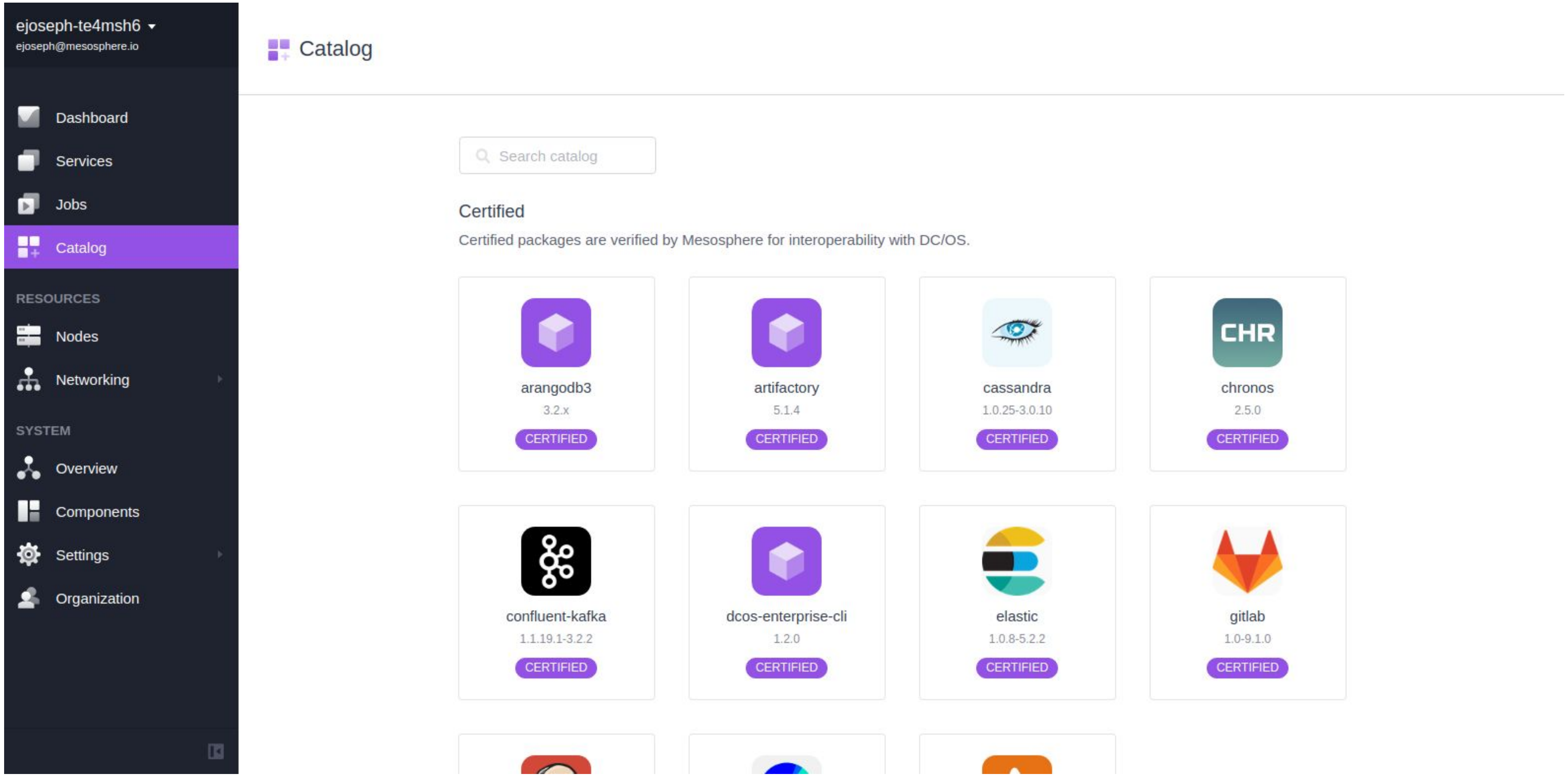


Interact with DC/OS: DC/OS UI



NAME	STATUS	CPU	MEM	DISK
cassandra	Running (4 Instances)	2	8 GiB	0 B
kafka	Running (4 Instances)	4	4.8 GiB	0 B
marathon-lb	Running (1 Instance)	2	1 GiB	0 B
zeppelin	Running (1 Instance)	1	2 GiB	0 B

Interacting with DC/OS: Installing Catalog Packages



ejoseph-te4msh6
ejoseph@mesosphere.io

Dashboard
Services
Jobs
Catalog

RESOURCES
Nodes
Networking

SYSTEM
Overview
Components
Settings
Organization

Catalog

Search catalog

Certified

Certified packages are verified by Mesosphere for interoperability with DC/OS.

arangodb3 3.2.x CERTIFIED	artifactory 5.1.4 CERTIFIED	cassandra 1.0.25-3.0.10 CERTIFIED	chronos 2.5.0 CERTIFIED
confluent-kafka 1.1.19.1-3.2.2 CERTIFIED	dcos-enterprise-cli 1.2.0 CERTIFIED	elastic 1.0.8-5.2.2 CERTIFIED	gitlab 1.0-9.1.0 CERTIFIED

Interact with DC/OS: DC/OS CLI

DC/OS CLI for Node & Cluster Management.

- dcos config
- dcos node
- dcos cluster

DC/OS CLI for App Management.

- dcos package
- dcos job
- dcos marathon
- dcos task

Interacting with DC/OS: Installing Catalog Packages

```
{
  "service": {
    "name": "kafka",
    "user": "nobody",
    "virtual_network_enabled": false,
    "virtual_network_name": "dcos",
    "virtual_network_plugin_labels": "",
    "placement_constraint": "[[\"hostname\", \"MAX_PER\", \"1\"]]",
    "deploy_strategy": "serial"
}
```

Tour DC/OS & Demo

- DC/OS UI and CLI walk through
  - Nodes page
  - Dashboard
  - Catalog: smack packages and k8s package.
  - Services page: marathon apps
  - Jobs page: metronome

Advanced Installation

1. **Prerequisites**:

- Docker
- OS packages
- NTP enabled
- Overlay for Docker
- DC/OS Package
- /genconf
    - IP Detect
    - Config file

2. **Install Process:**

- Generate installer
- Serve install files
- Install master
- Install agents

```
$ sudo bash dcos_install.sh master
```

DC/OS

Installing DC/OS Lab

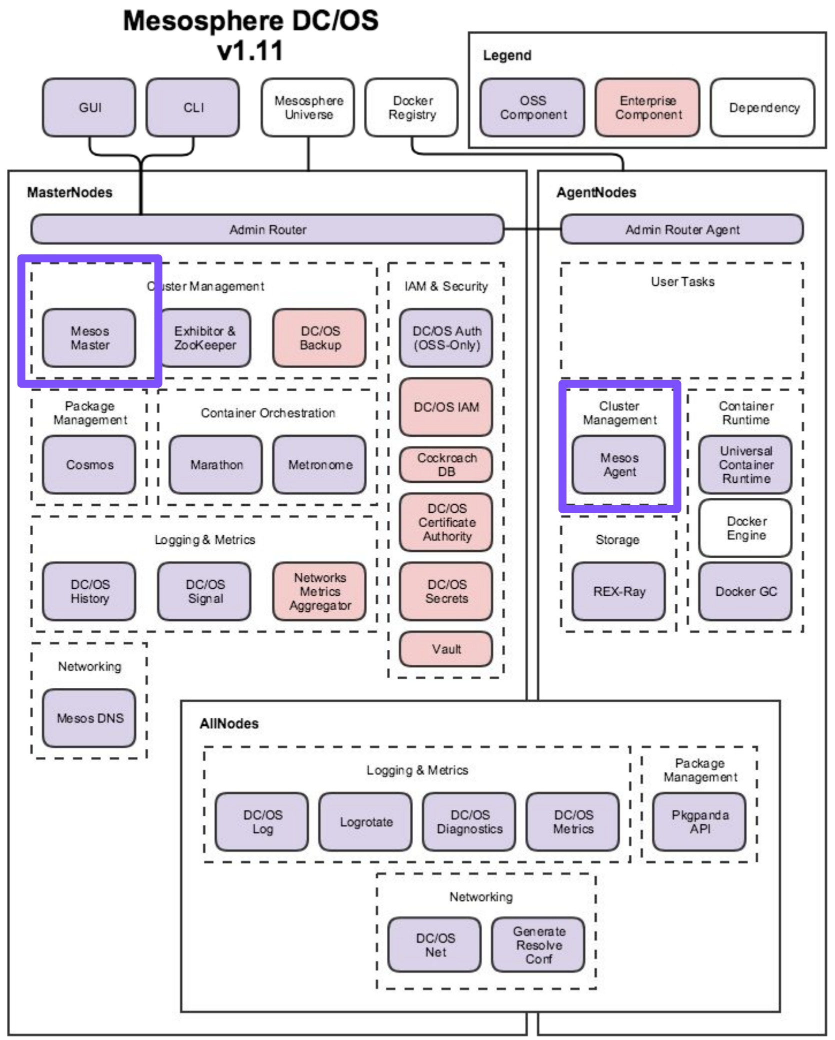
Server Assignments:

- https://tinyurl.com/y9uq9pa6

In this lab you will:

- Install a cluster of DC/OS nodes with Ansible.
- Explore the DC/OS UI.
- Install the DC/OS CLI on the bootstrap node.
- Try out the the DC/OS CLI.

DC/OS Architecture Overview

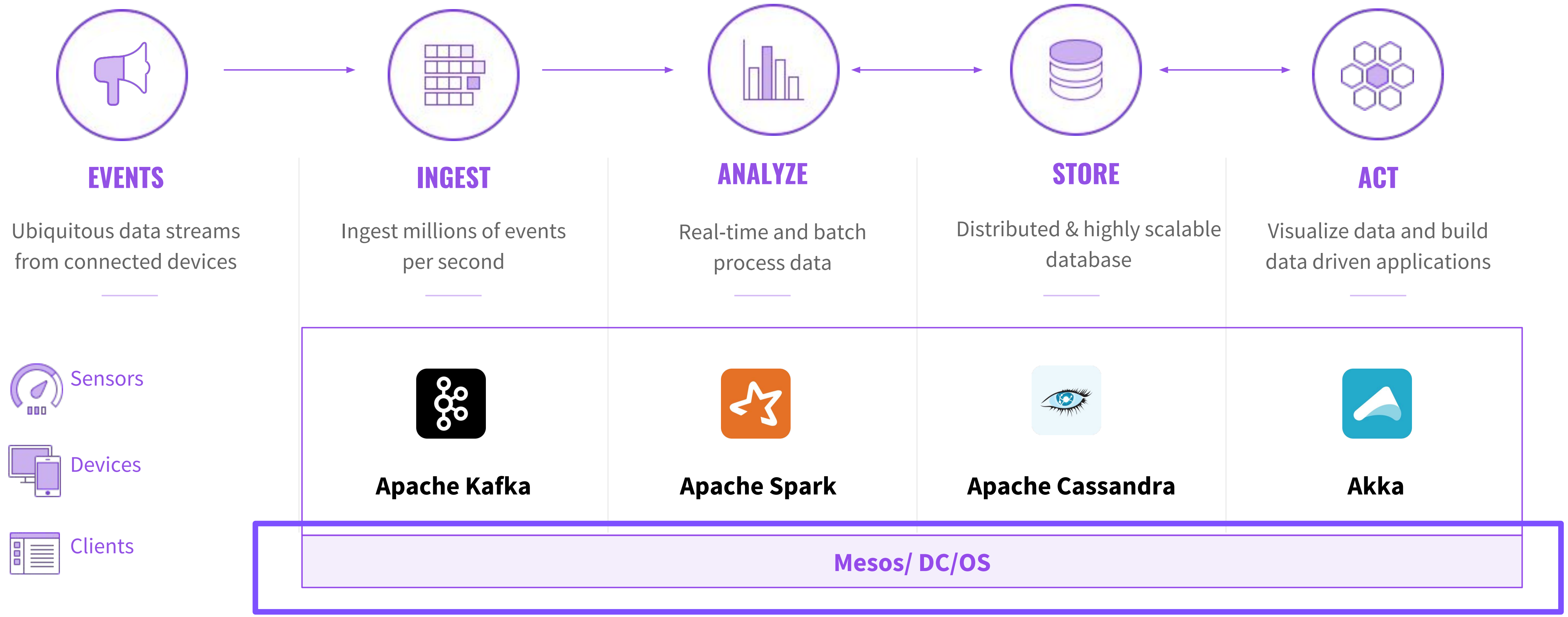


SMACK stack

E
S
O
S

- History & Context
- Intro to Mesos
- Architecture

SMACK Stack



EVENTS

Ubiquitous data streams from connected devices

INGEST

Ingest millions of events per second

ANALYZE

Real-time and batch process data

STORE

Distributed & highly scalable database

ACT

Visualize data and build data driven applications

Build Block of Modern Internet

- A cluster resource negotiator
- A top-level Apache project
- Scalable to 10,000s of nodes
- Fault-tolerant, battle-tested
- An SDK for distributed apps
- Native Docker support

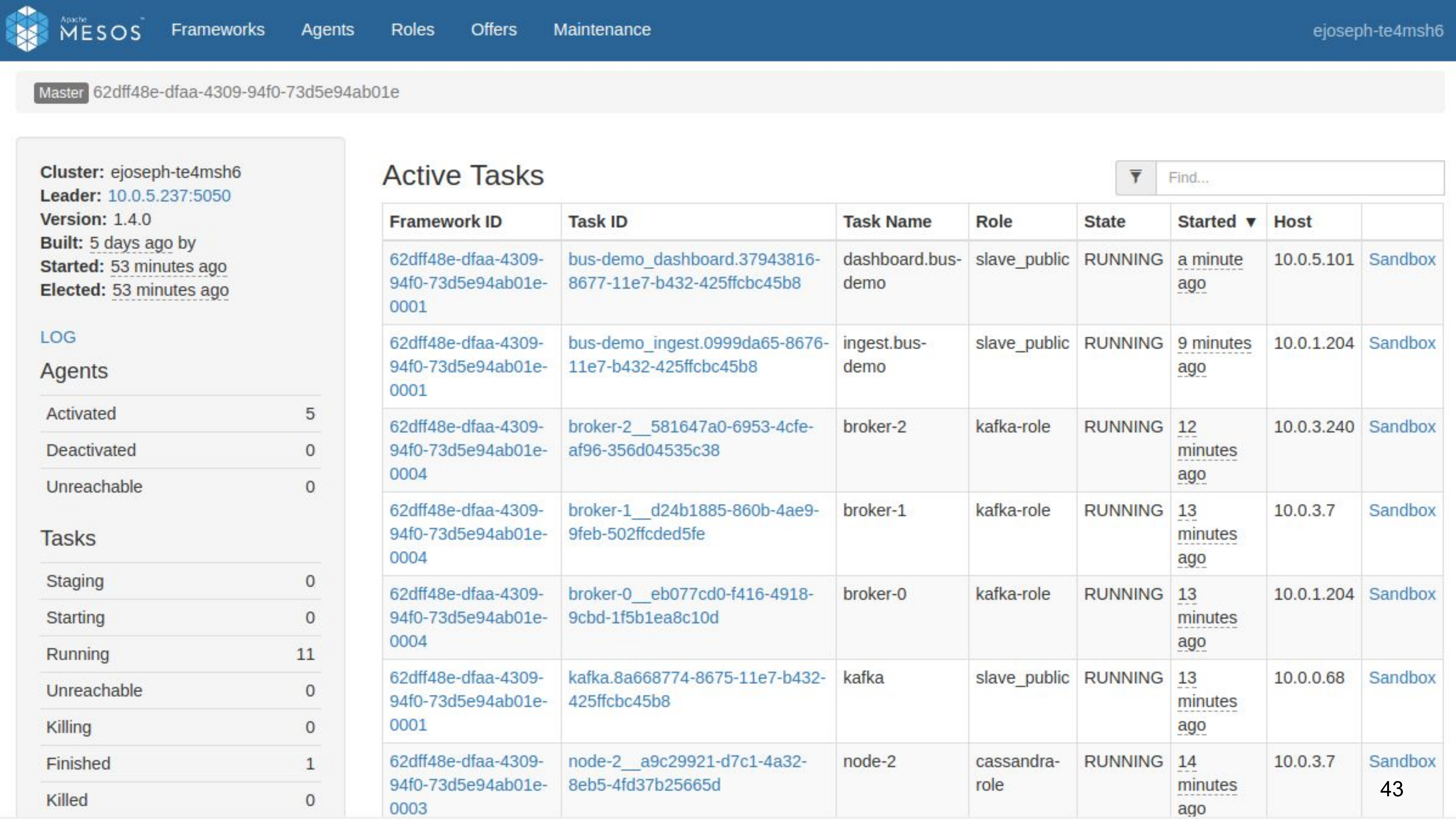
UBER verizon airbnb ebay

DC/OS

Mesos: Datacenter Kernel

- Opens source Apache project.
- Resource manager.
- Pools resources from set of servers to create “one giant computer”.
- Mesos master orchestrates agent tasks.
- Mesos agents provide resources.

DC/OS



Master 62dff48e-dfaa-4309-94f0-73d5e94ab01e

Cluster: ejoseph-te4msh6
Leader: 10.0.5.237:5050
Version: 1.4.0
Built: 5 days ago by
Started: 53 minutes ago
Elected: 53 minutes ago

LOG

Agents

Activated	5
Deactivated	0
Unreachable	0

Tasks

Staging	0
Starting	0
Running	11
Unreachable	0
Killing	0
Finished	1
Killed	0

Active Tasks

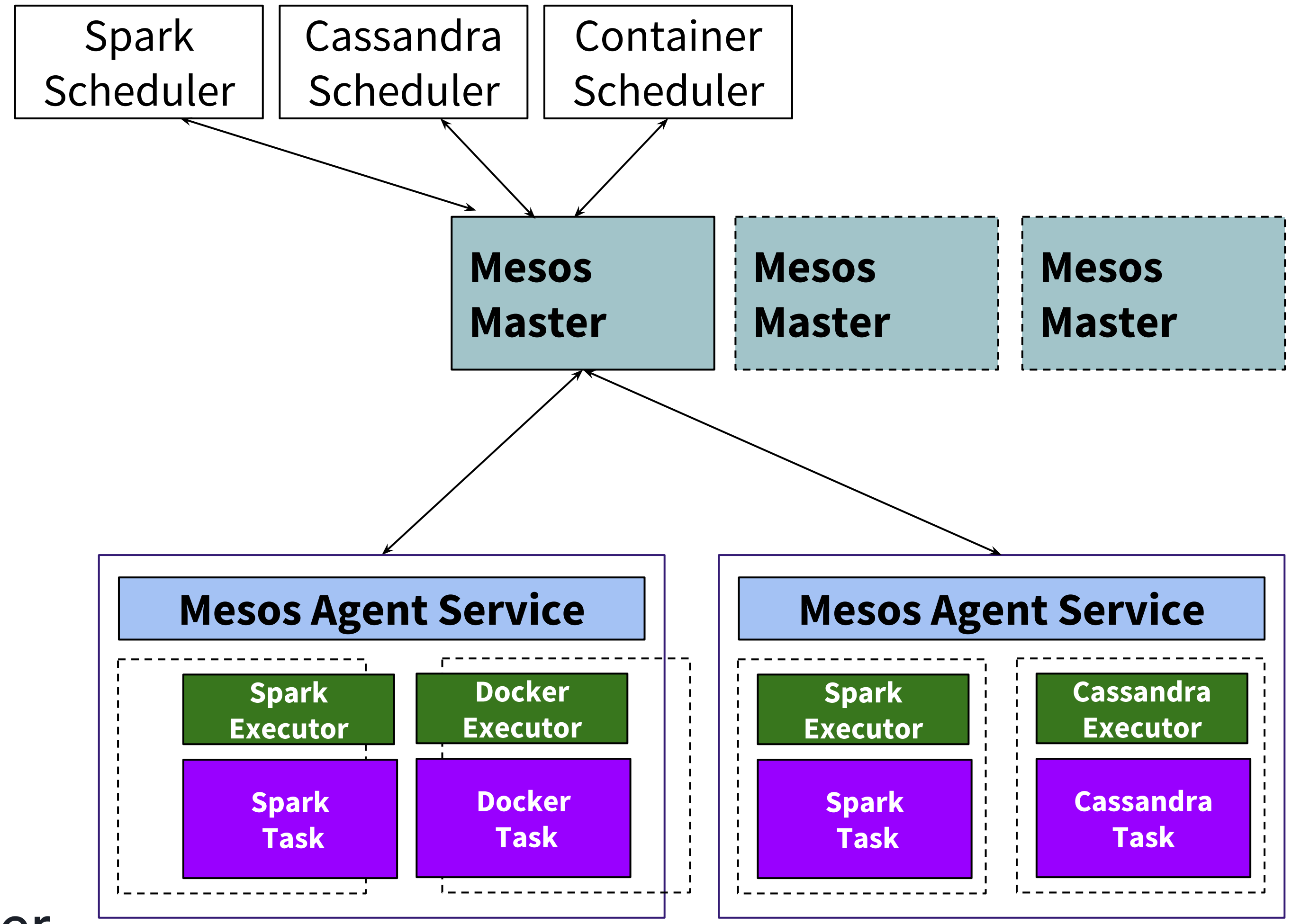
Find...

Framework ID	Task ID	Task Name	Role	State	Started ▼	Host	
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0001	bus-demo_dashboard.37943816-8677-11e7-b432-425ffcbc45b8	dashboard.bus-demo	slave_public	RUNNING	a minute ago	10.0.5.101	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0001	bus-demo_ingest.0999da65-8676-11e7-b432-425ffcbc45b8	ingest.bus-demo	slave_public	RUNNING	9 minutes ago	10.0.1.204	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0004	broker-2__581647a0-6953-4cfe-af96-356d04535c38	broker-2	kafka-role	RUNNING	12 minutes ago	10.0.3.240	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0004	broker-1__d24b1885-860b-4ae9-9feb-502ffcded5fe	broker-1	kafka-role	RUNNING	13 minutes ago	10.0.3.7	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0004	broker-0__eb077cd0-f416-4918-9cbd-1f5b1ea8c10d	broker-0	kafka-role	RUNNING	13 minutes ago	10.0.1.204	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0001	kafka.8a668774-8675-11e7-b432-425ffcbc45b8	kafka	slave_public	RUNNING	13 minutes ago	10.0.0.68	Sandbox
62dff48e-dfaa-4309-94f0-73d5e94ab01e-0003	node-2__a9c29921-d7c1-4a32-8eb5-4fd37b25665d	node-2	cassandra-role	RUNNING	14 minutes ago	10.0.3.7	Sandbox

Mesos Architecture

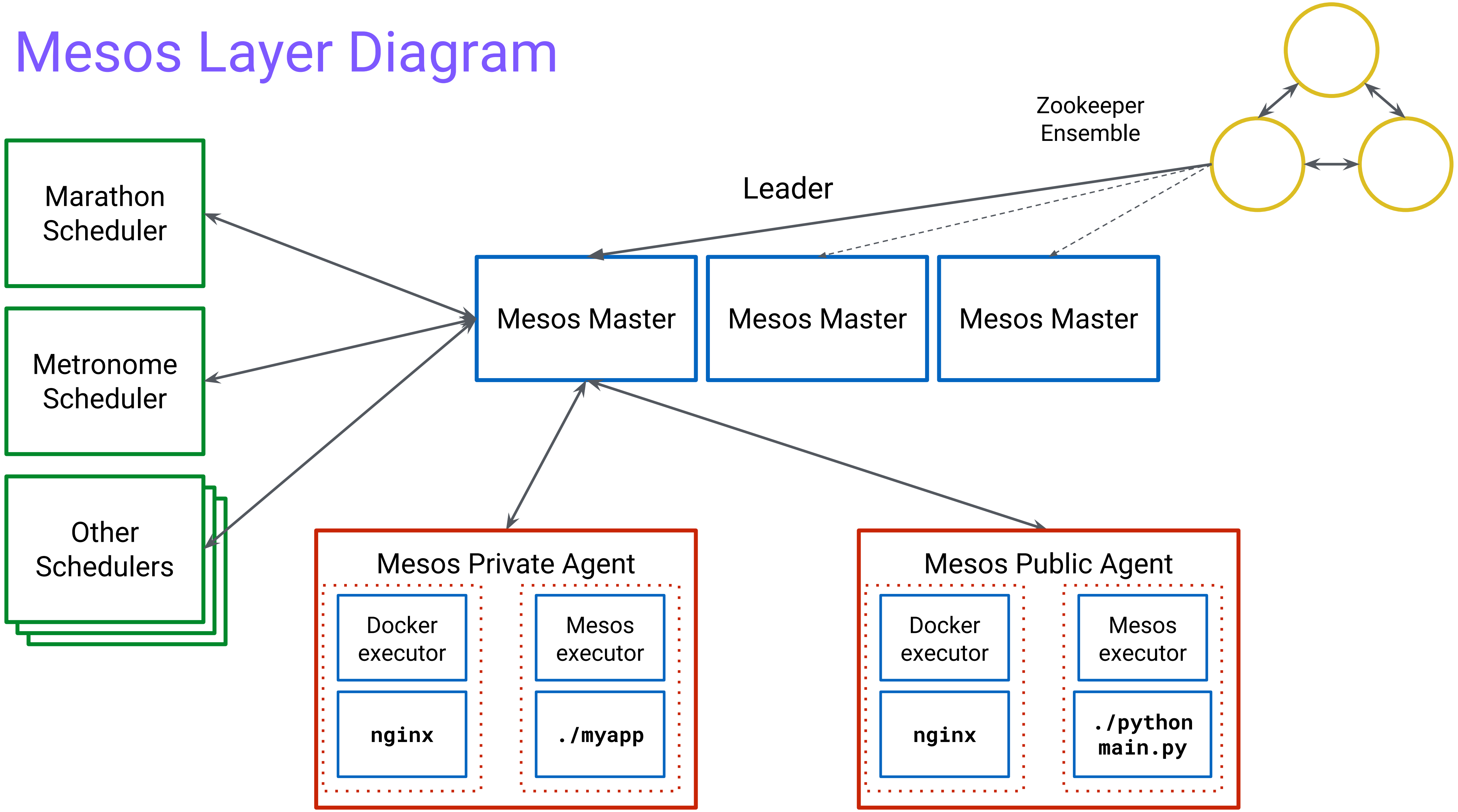
Two-level Scheduling

1. Agents advertise resources to Master
2. Master offers resources to Framework
3. Framework rejects or uses resources
4. Agent reports task status to Master

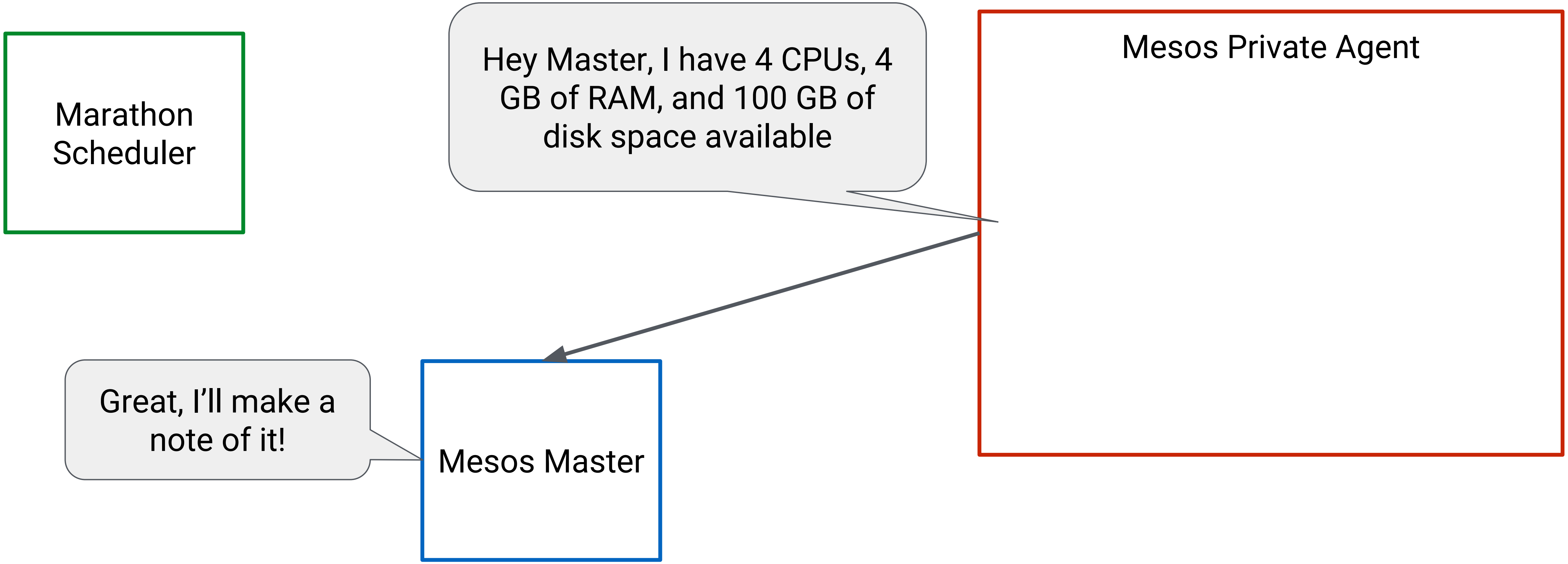


DC/OS

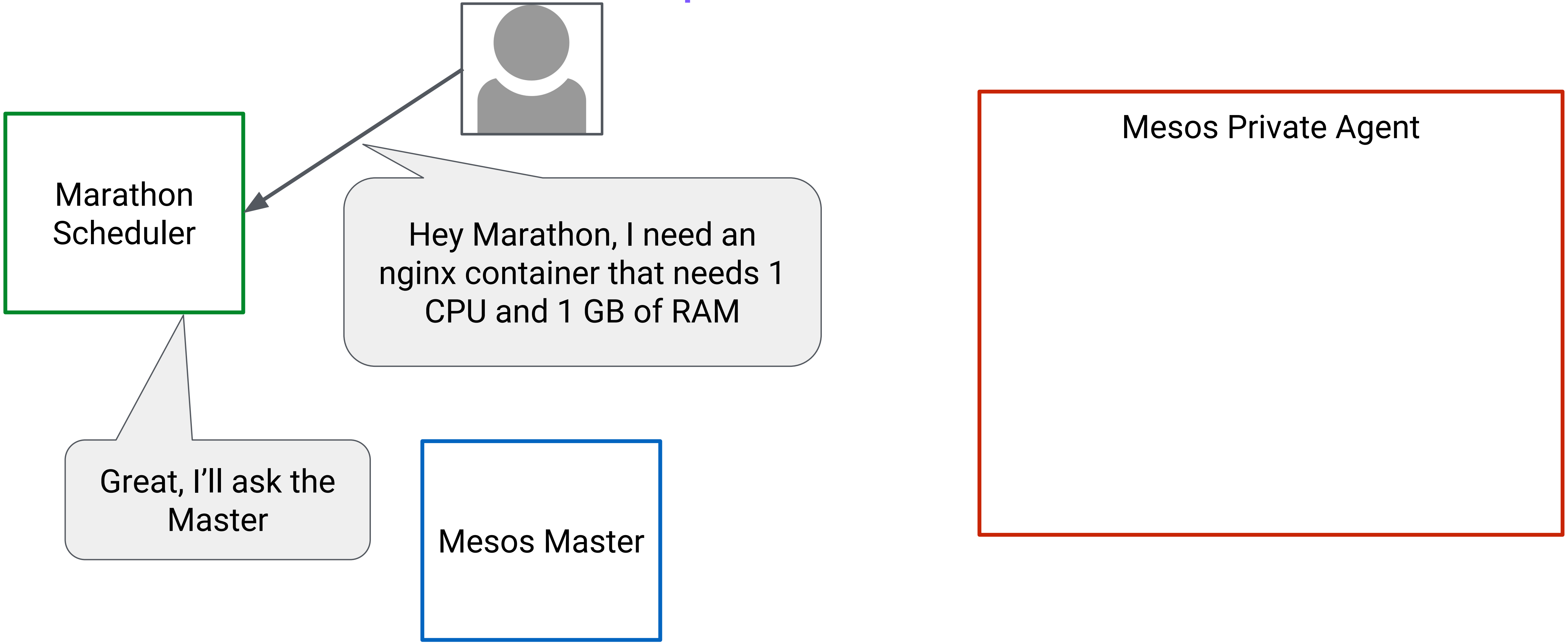
Mesos Layer Diagram



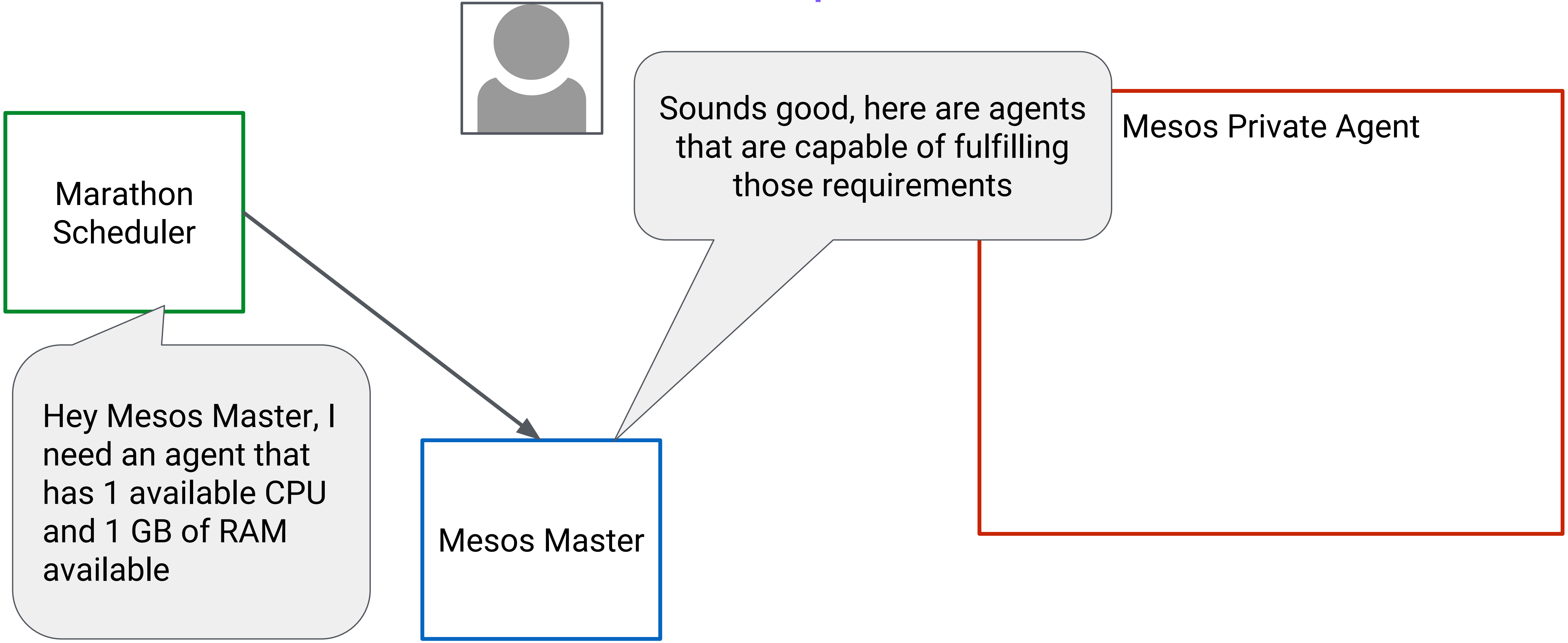
Mesos in Action - Resource Offer


Marathon Scheduler
Hey Master, I have 4 CPUs, 4 GB of RAM, and 100 GB of disk space available
Mesos Private Agent
Great, I'll make a note of it!
Mesos Master

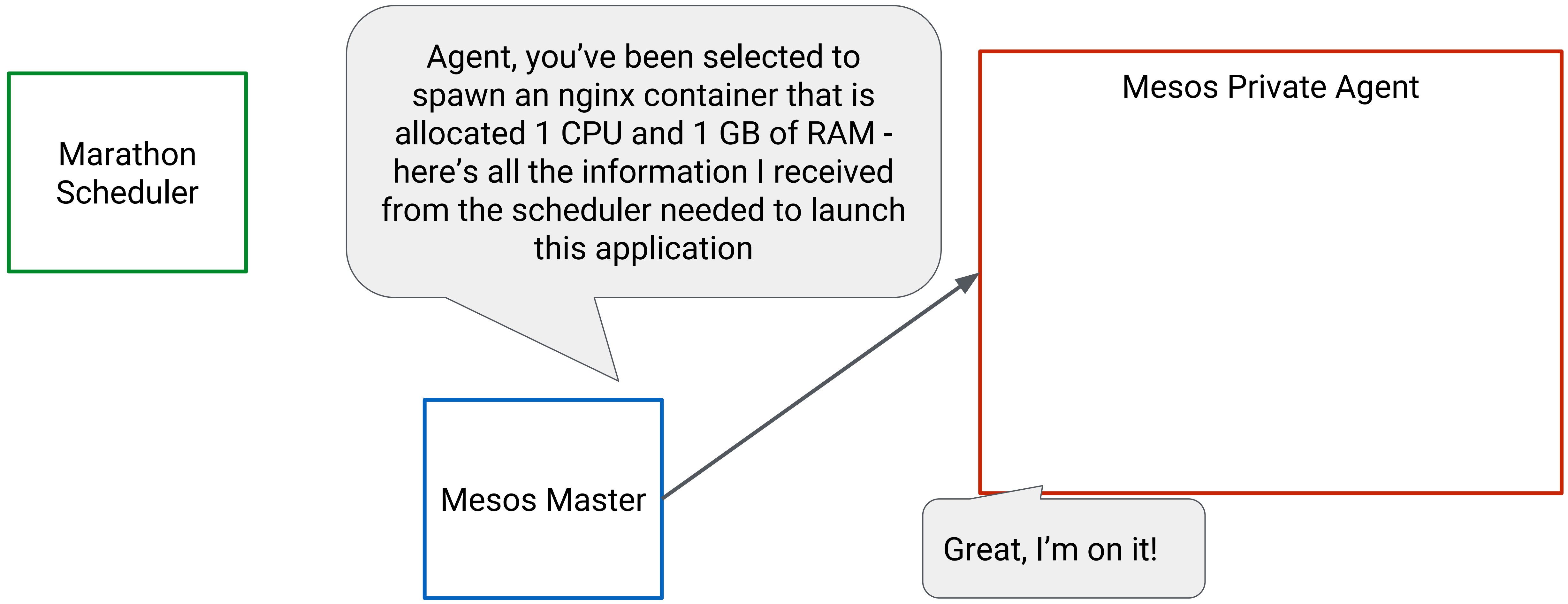
Mesos in Action - User Request


Marathon Scheduler
Hey Marathon, I need an nginx container that needs 1 CPU and 1 GB of RAM
Great, I'll ask the Master
Mesos Master
Mesos Private Agent

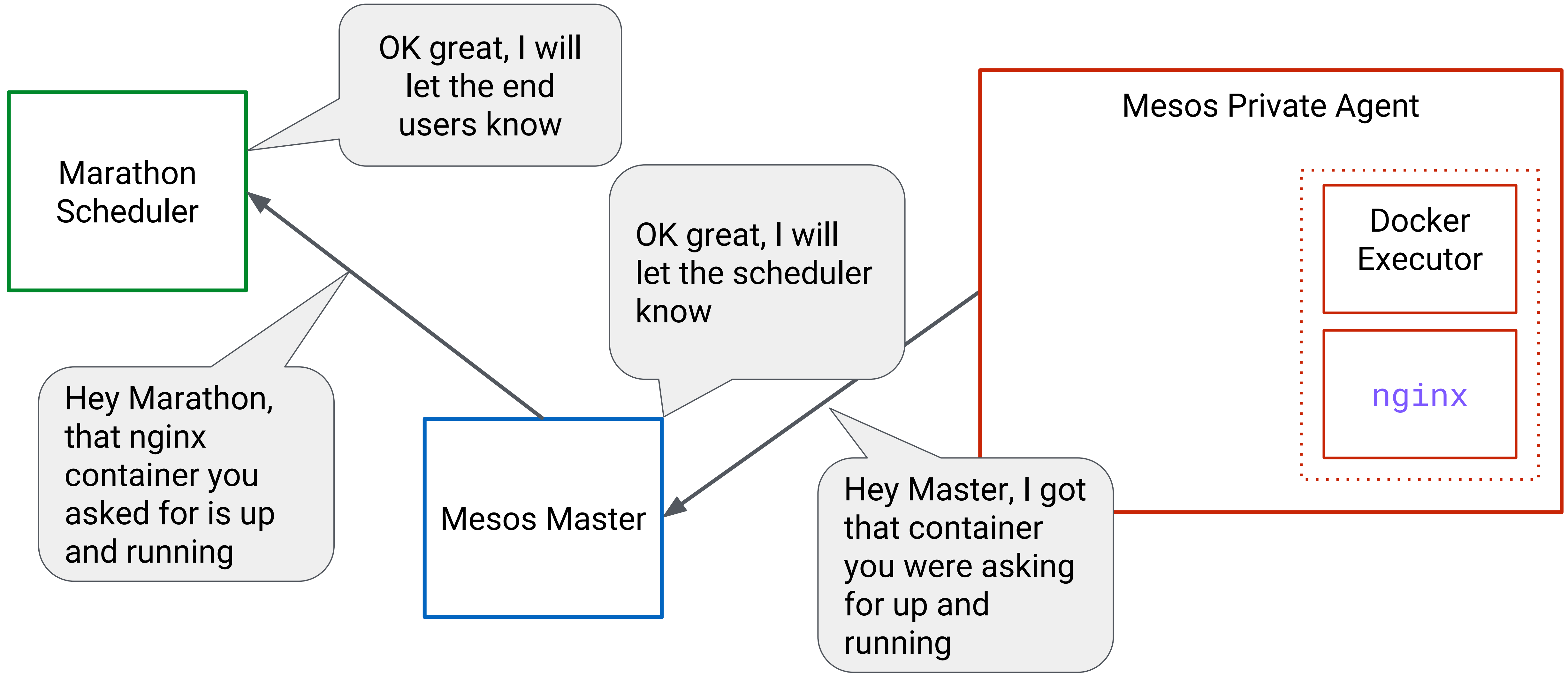
Mesos in Action - Scheduler Request


Marathon Scheduler
Hey Mesos Master, I need an agent that has 1 available CPU and 1 GB of RAM available
Mesos Master
Sounds good, here are agents that are capable of fulfilling those requirements
Mesos Private Agent

Mesos in Action - Container Launch


Marathon Scheduler
Agent, you've been selected to spawn an nginx container that is allocated 1 CPU and 1 GB of RAM - here's all the information I received from the scheduler needed to launch this application
Mesos Private Agent
Mesos Master
Great, I'm on it!

Mesos in Action - Container Running


Marathon Scheduler
OK great, I will let the end users know
Hey Marathon, that nginx container you asked for is up and running
Mesos Master
OK great, I will let the scheduler know
Hey Master, I got that container you were asking for up and running
Mesos Private Agent
Docker Executor
nginx

Quick Knowledge Check

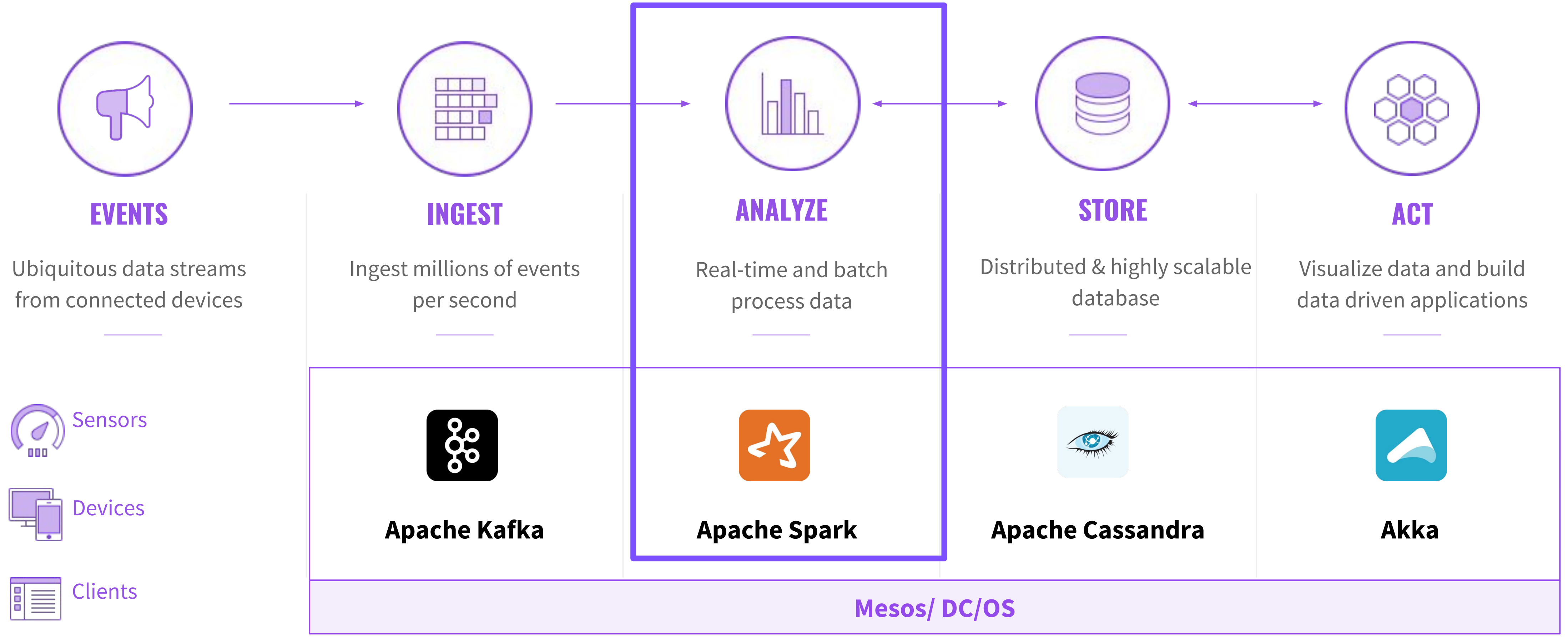
How many leading Mesos masters can you have in a DC/OS cluster?

- 1
- 3
- 5

SMACK stack
P
A
R
K

- Context
- Intro to Spark
- Installing, Configuring, & Managing

SMACK Stack


EVENTS
Ubiquitous data streams from connected devices
INGEST
Ingest millions of events per second
ANALYZE
Real-time and batch process data
STORE
Distributed & highly scalable database
ACT
Visualize data and build data driven applications
Sensors
Devices
Clients
Apache Kafka
Apache Spark
Apache Cassandra
Akka
Mesos/ DC/OS

Streaming Analytics

Micro-batching

- Apache Spark (Streaming)

Native Streaming

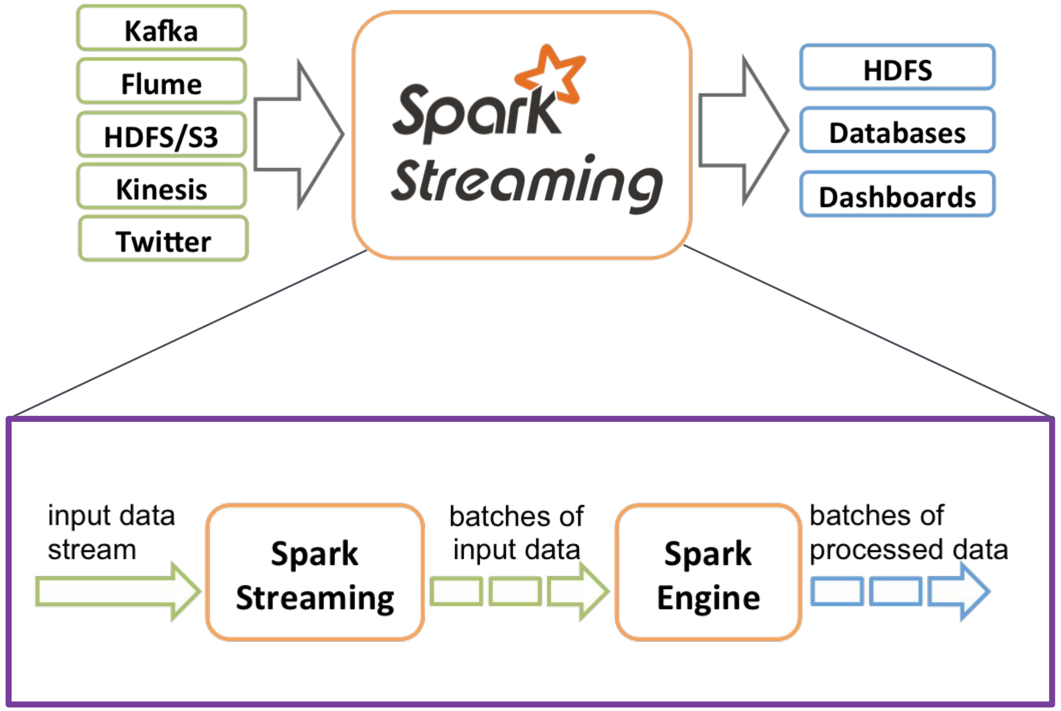
- Apache Flink
- Apache Storm/Heron
- Apache Apex
- Apache Samza

Spark: Streaming Analytics

Typical Use: distributed, large-scale data processing; micro-batching

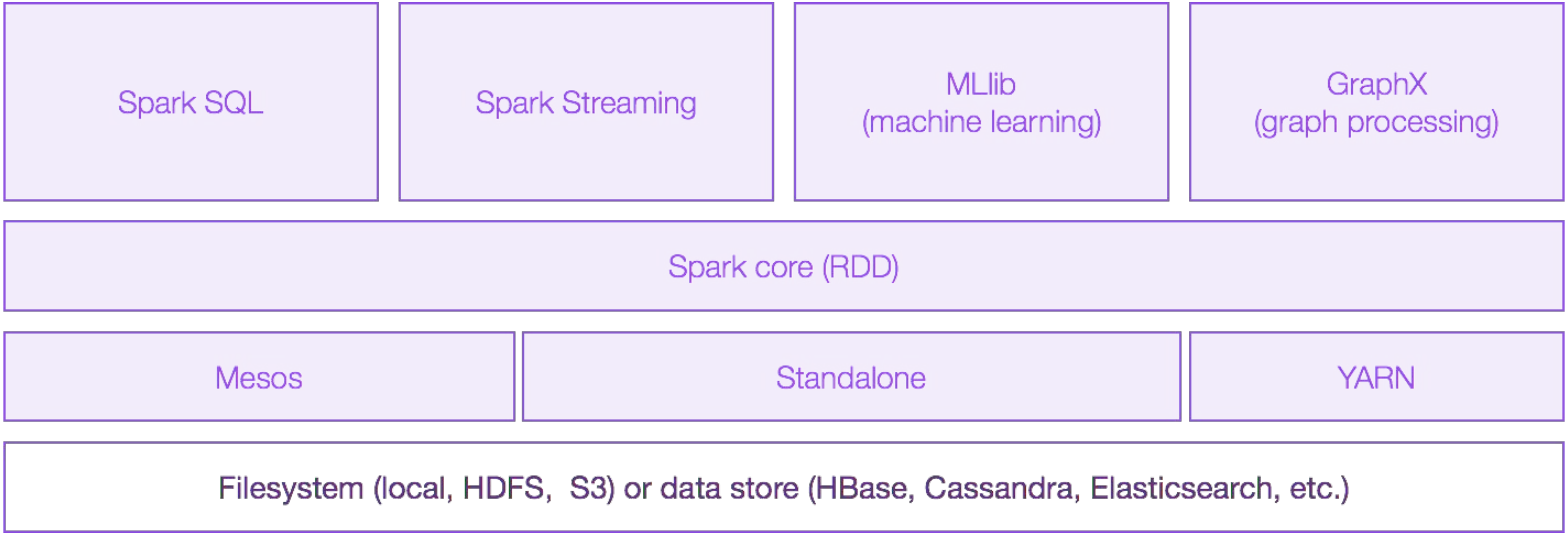
Why Spark Streaming?

- Micro-batching creates very low latency, which can be faster
- Well defined role means it fits in well with other pieces of the pipeline



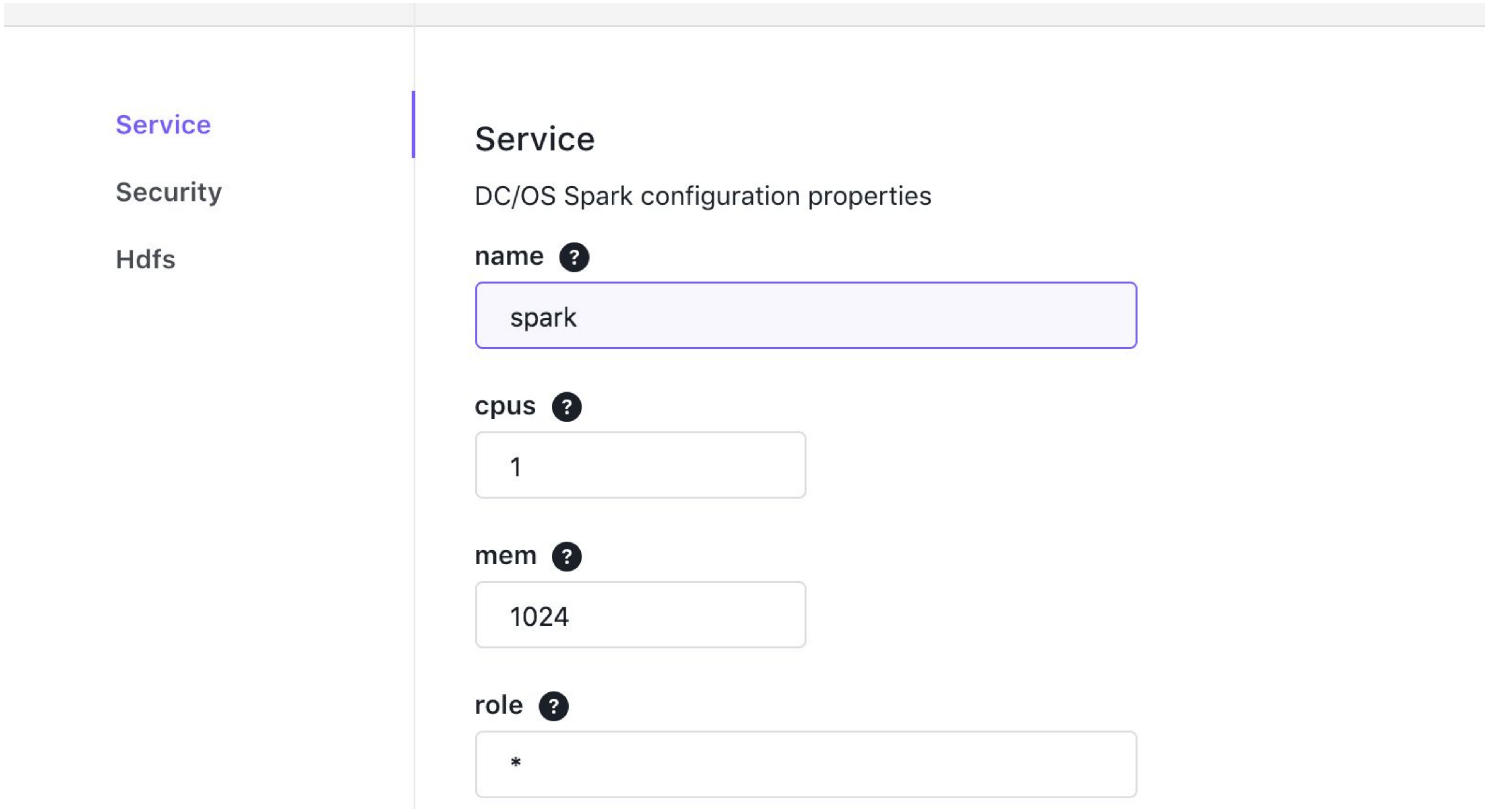
DC/OS

Spark: Architecture


Spark SQL
Spark Streaming
MLlib (machine learning)
GraphX (graph processing)
Spark core (RDD)
Mesos
Standalone
YARN
Filesystem (local, HDFS, S3) or data store (HBase, Cassandra, Elasticsearch, etc.)

DC/OS

DC/OS Spark Package



Service

DC/OS Spark configuration properties

name: spark

cpus: 1

mem: 1024

role: *

DC/OS Spark Package Parameters

Service

- Name
- CPU
- Mem
- User
- Role for Spark Dispatcher
- “Quota” parameter - restricts resource usage.

HDFS

- HDFS configuration file location

Security

- Kerberos
- Kerberos configuration

DC/OS Spark Package Default Parameters

Service

- 1 CPU
- 1 GB Memory
- Root user for executor
- Role for Spark Dispatcher is “*

HDFS

- DC/OS HDFS default configuration

Security

- Kerberos is disabled

Interacting with Spark

Spark UI

- Monitor Jobs

DC/OS CLI Subcommands

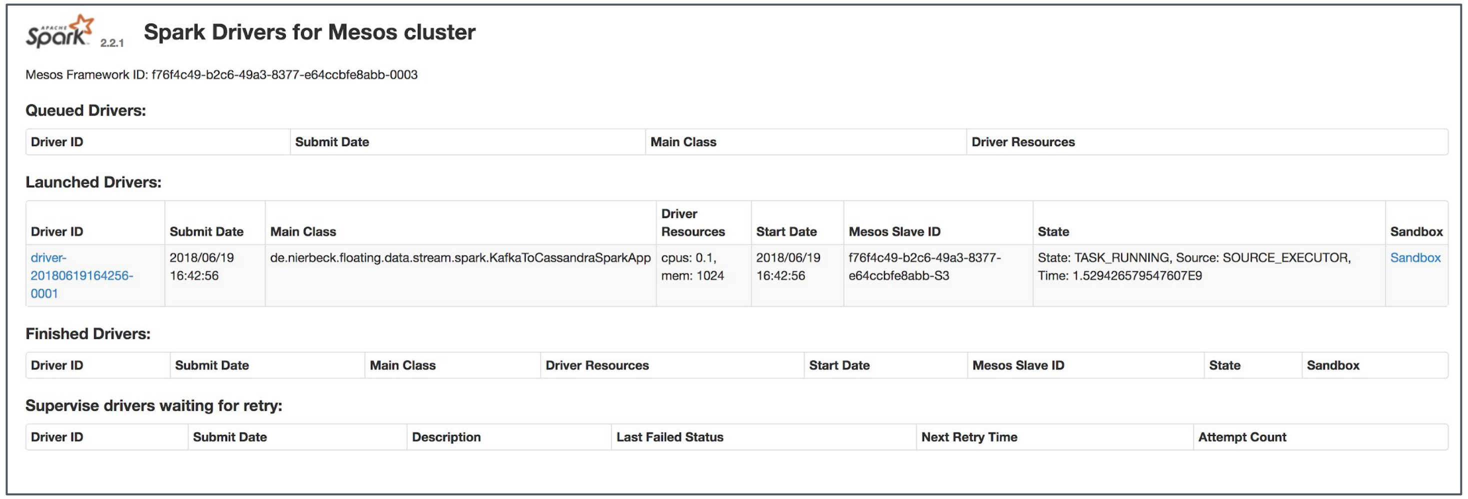
- Submit & Monitor jobs

DC/OS CLI

- dcos task exec -it

Connection Information from UI

- Dispatcher and dispatcher proxy LB info.


2.2.1 **Spark Drivers for Mesos cluster**

Mesos Framework ID: f76f4c49-b2c6-49a3-8377-e64ccbfe8abb-0003

Queued Drivers:

Driver ID	Submit Date	Main Class	Driver Resources

Launched Drivers:

Driver ID	Submit Date	Main Class	Driver Resources	Start Date	Mesos Slave ID	State	Sandbox
driver-20180619164256-0001	2018/06/19 16:42:56	de.nierbeck.floating.data.stream.spark.KafkaToCassandraSparkApp	cpus: 0.1, mem: 1024	2018/06/19 16:42:56	f76f4c49-b2c6-49a3-8377-e64ccbfe8abb-S3	State: TASK_RUNNING, Source: SOURCE_EXECUTOR, Time: 1.529426579547607E9	Sandbox

Finished Drivers:

Driver ID	Submit Date	Main Class	Driver Resources	Start Date	Mesos Slave ID	State	Sandbox

Supervise drivers waiting for retry:

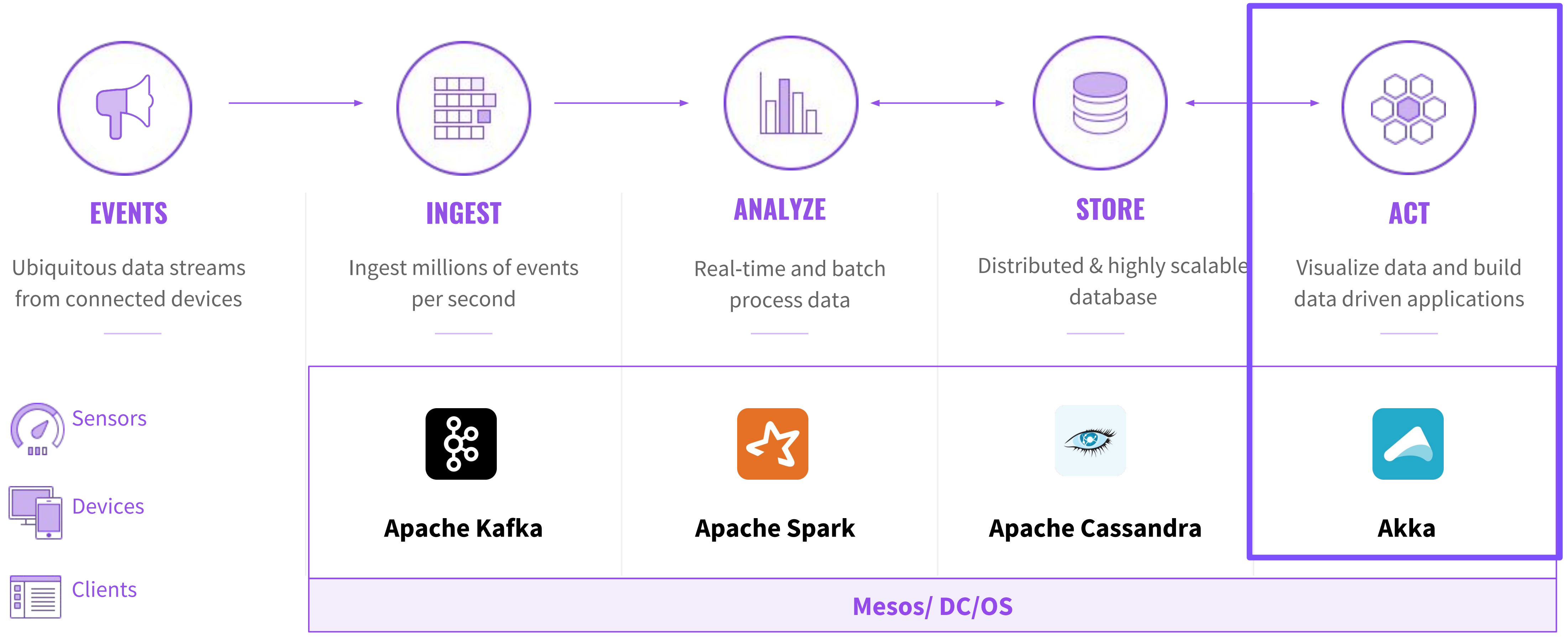
Driver ID	Submit Date	Description	Last Failed Status	Next Retry Time	Attempt Count

SMACK stack

K
K
A

- Intro to Akka
- Configuring

SMACK Stack


EVENTS
Ubiquitous data streams from connected devices
INGEST
Ingest millions of events per second
ANALYZE
Real-time and batch process data
STORE
Distributed & highly scalable database
ACT
Visualize data and build data driven applications
Sensors
Devices
Clients
Apache Kafka
Apache Spark
Apache Cassandra
Akka
Mesos/ DC/OS

DC/OS

Akka Driven Applications

Akka is a toolkit for building highly concurrent, distributed, and resilient message-driven applications for Java and Scala.

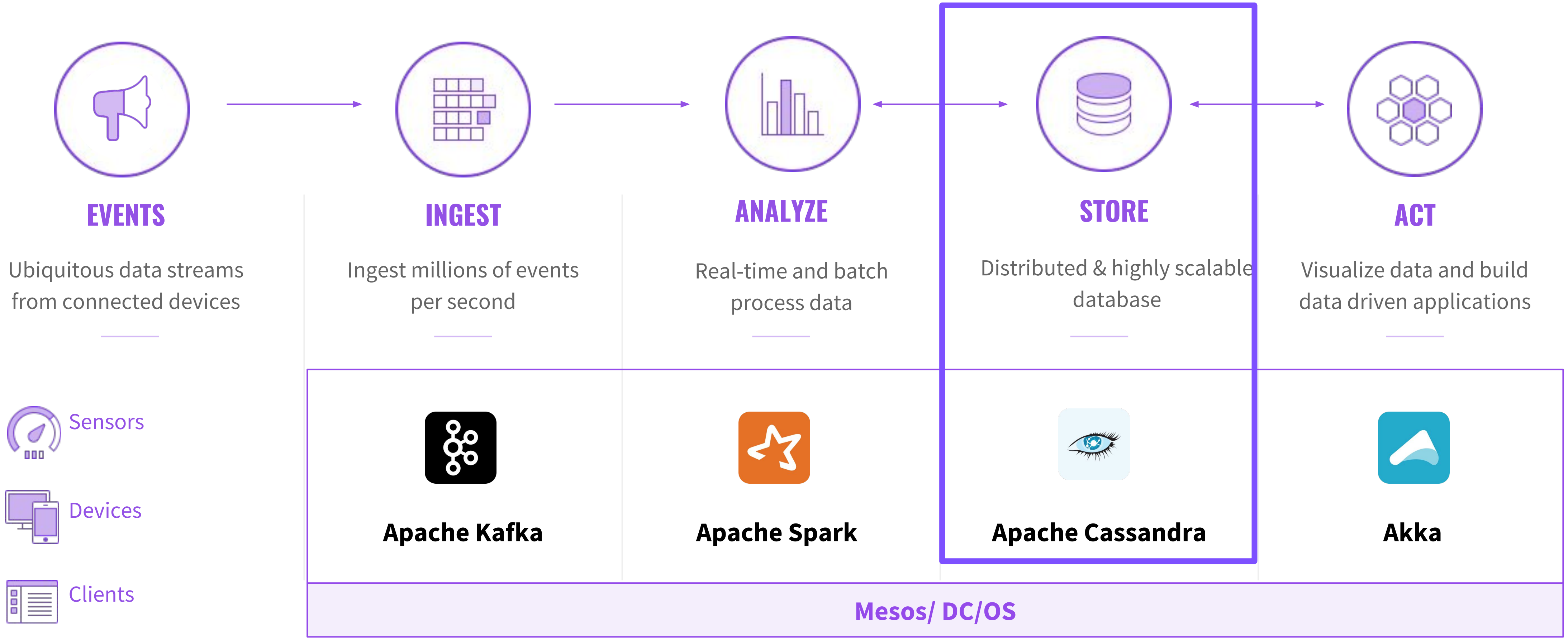
- Simple
- Highly Performant
- Elastic
- Reactive

SMACK stack

A
S
S
A
N
D
R
A

- History & Context
- Intro to Cassandra
- Installing, Configuring, & Managing

SMACK Stack


EVENTS
Ubiquitous data streams from connected devices
INGEST
Ingest millions of events per second
ANALYZE
Real-time and batch process data
STORE
Distributed & highly scalable database
ACT
Visualize data and build data driven applications
Sensors
Devices
Clients
Apache Kafka
Apache Spark
Apache Cassandra
Akka
Mesos/ DC/OS

DC/OS

History of Distributed Storage

NoSQL

- ArangoDB
- MongoDB
- Apache Cassandra
- Apache HBase

Filesystems

- Quobyte
- HDFS

Time-Series Datastores

- InfluxDB
- OpenTSDB
- KairosDB
- Prometheus

SQL

- MemSQL

DC/OS

Cassandra

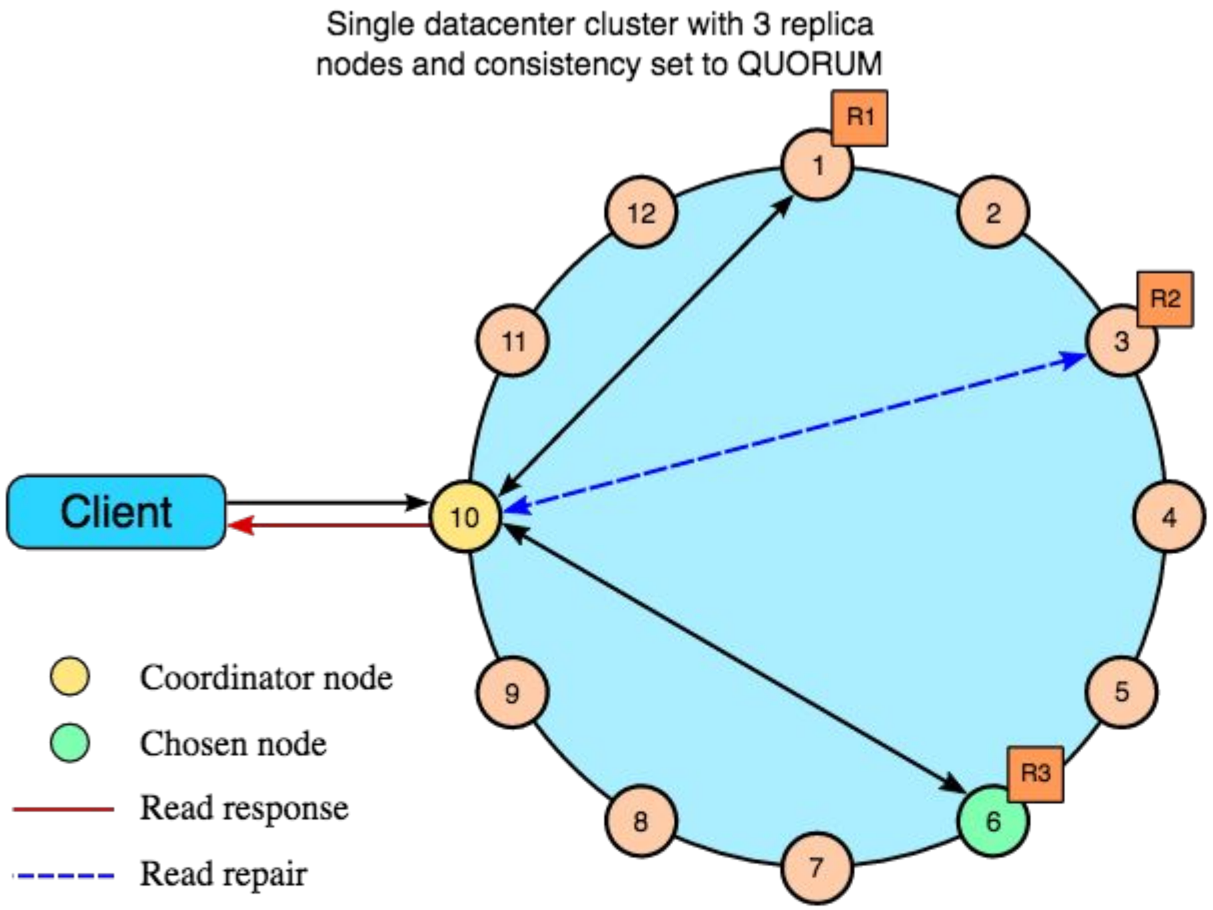
Typical Use: No-dependency, time series database

Why Cassandra?

- A top level Apache project born at Facebook and built on Amazon’s Dynamo and Google’s BigTable
- Offers continuous availability, linear scale performance, operational simplicity and easy data distribution

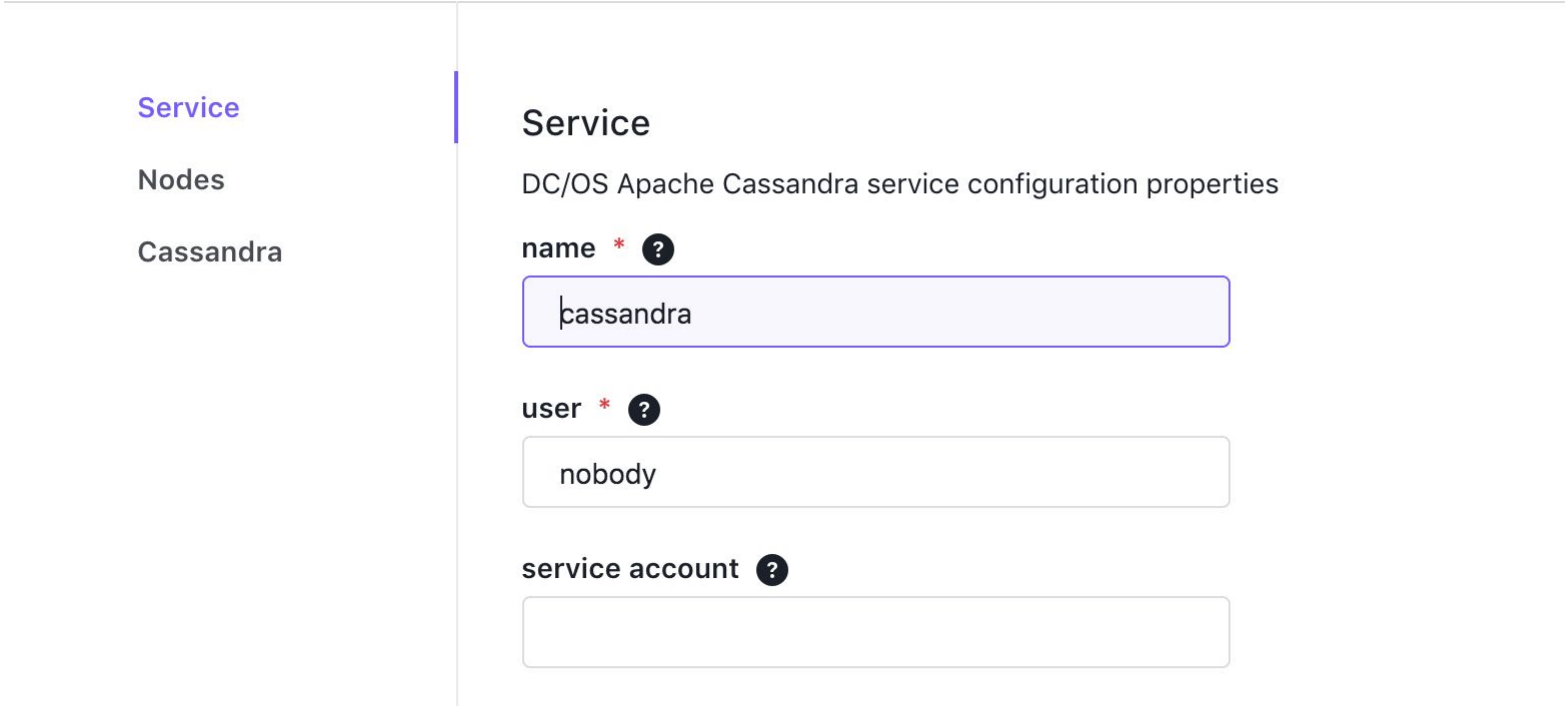
Cassandra Architecture

- Cassandra is eventually consistent
- Multiple parameter to tweak read/write consistency
  - Write Strategies:
    - Any, One, Quorum, All, ..
  - Read Strategies:
    - One, Quorum, ALL
- Granularity: single row/key



DC/OS

DC/OS Package Definition



Service

DC/OS Apache Cassandra service configuration properties

name * : cassandra

user * : nobody

service account :

DC/OS Cassandra Package Parameters

Service

- Cluster name
- Data Center
- Region

Nodes

- Number of nodes
- Placement constraints
- Racks
- Resources*

Cassandra:

- Practitioner
- Hinted handoff
- Concurrent reads and writes
- tombstone*

DC/OS Cassandra Package Default Parameters

Node

- 3 nodes
- Placement constraint: 1 Cassandra node per DC/OS private agent.
- .5 CPU
- 10 GB Diskspace
- 4 GB RAM

Cassandra

- Hinted handoff enabled
- Partitioner is Murmur3partitioner
- Concurrent Reads 16
- Concurrent Writes 32

Interacting with Cassandra

Connection information from UI or CLI

- Node address and port
- DNS for service

DC/OS CLI: dcos task exec

- Connect to a task

Cqlsh

- Connect to the cluster data store.

Backup & Restore with DC/OS CLI

- Backup to AWS or Azure
- Restore

API

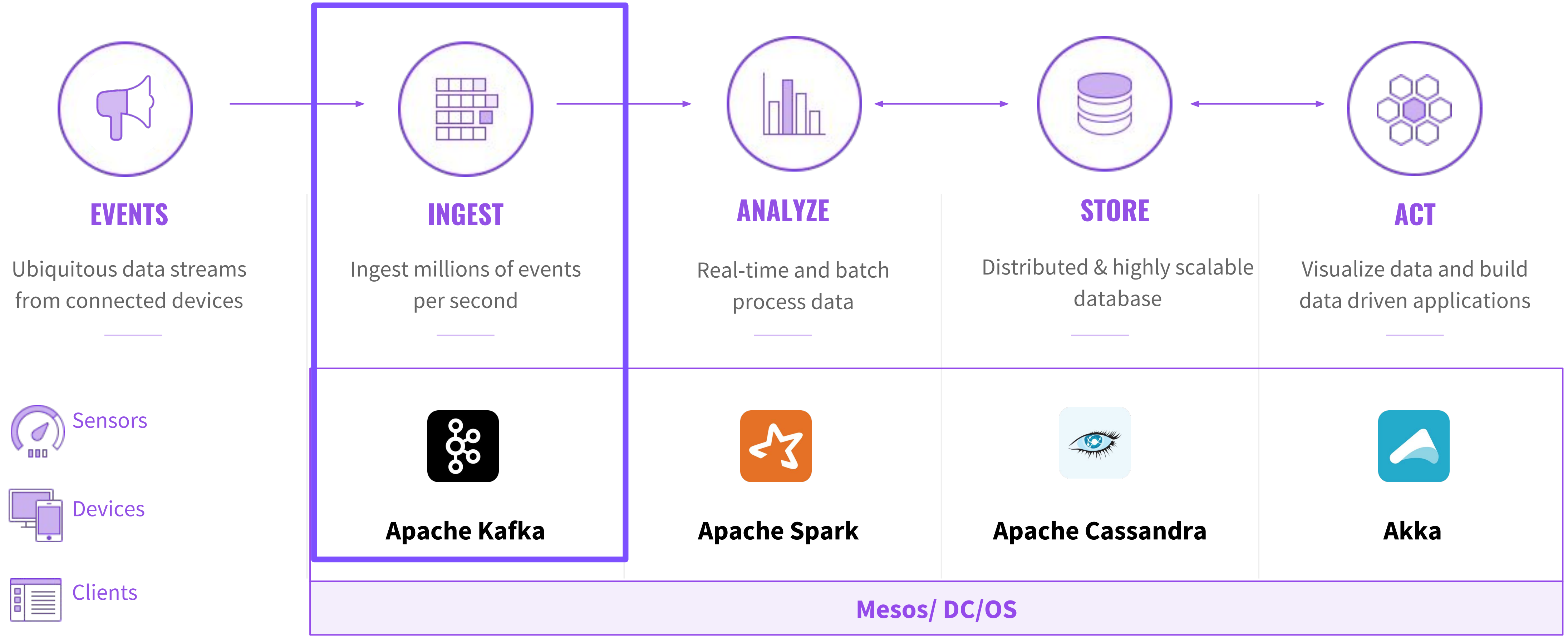
- Replace a node
- Restart a node
- Pause a node

SMACK stack

A
F
K
A

- Messaging Queues
- Intro to Kafka
- Installing, Configuring, & Managing

SMACK Stack



EVENTS

Ubiquitous data streams from connected devices

INGEST

Ingest millions of events per second

ANALYZE

Real-time and batch process data

STORE

Distributed & highly scalable database

ACT

Visualize data and build data driven applications

Sensors

Devices

Clients

Apache Kafka

Apache Spark

Apache Cassandra

Akka

Mesos/ DC/OS

Messaging Queues

Message Brokers

- Apache Kafka
- ØMQ, RabbitMQ, Disque

Log-based Queues

- fluentd, Logstash, Flume

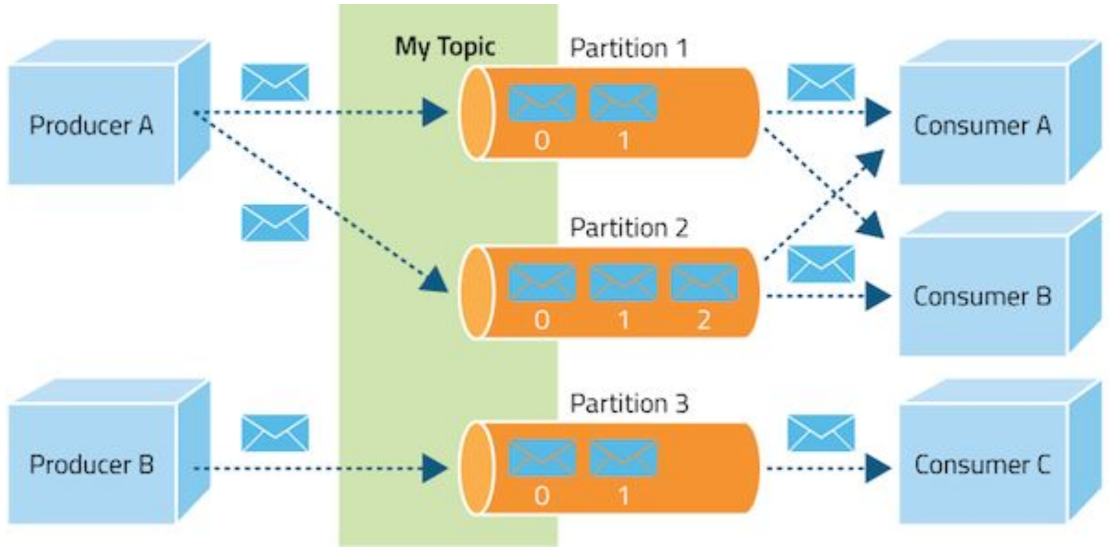
see also queues.io

Kafka

Typical Use: A reliable buffer for stream processing

Why Kafka?

- High-throughput, distributed, persistent publish-subscribe messaging system
- Created by LinkedIn; used in production by 100+ web-scale companies [1]



Kafka: Delivery Guarantees

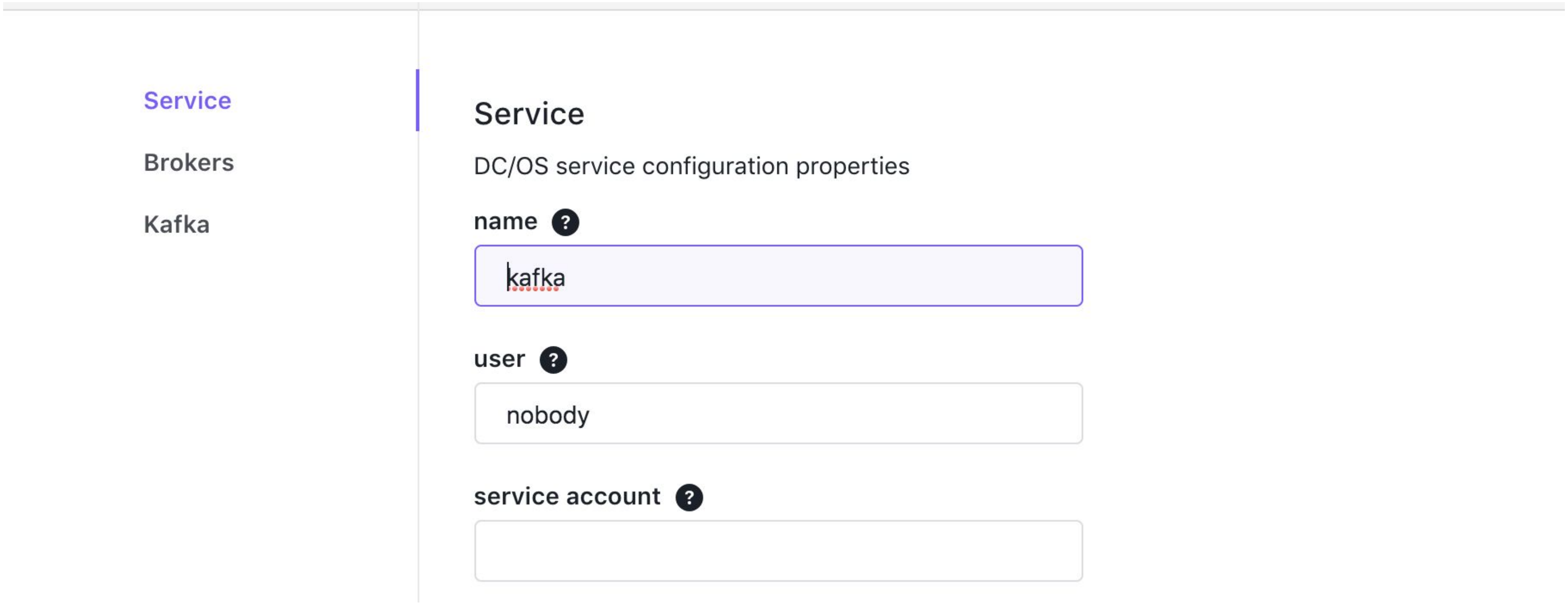
- At most once—Messages may be lost but are never re-delivered
- At least once—Messages are never lost but may be redelivered (Kafka)
- Exactly once—Messages are delivered once and only once (this is what everyone actually wants, but it's tricky)

Murphy's Law of Distributed Systems:

Anything that can go wrong, will go wrong … partially!

DC/OS

DC/OS Kafka Package



Service

DC/OS service configuration properties

name

kafka

user

nobody

service account

DC/OS Kafka Package Parameters

Sevice

- Service name
- Placement contraints
- Region
- Deploy strategy

Brokers

- Resources*
- Number of brokers

Kafka

- Topic management
- Logging

DC/OS Kafka Package Defaults

Sevice

- Service name: Kafka
- Placement constraints: 1 Kafka broker per DC/OS private agent.
- Region: unselected.
- Deploy strategy: Serial

Brokers

- Resources*
- Number of brokers: 3

Kafka

- Topic management*
- Logging*

Interacting with Kafka

Connection information from UI or CLI

- VIP load balancing
- Node address and port
- DNS for service

DC/OS CLI: dcos task exec

- Connect to a task

Kafka API

- Manage nodes
- Manage topics

DC/OS CLI Subcommands

- Manage topics

SMACK Stack Lab 2

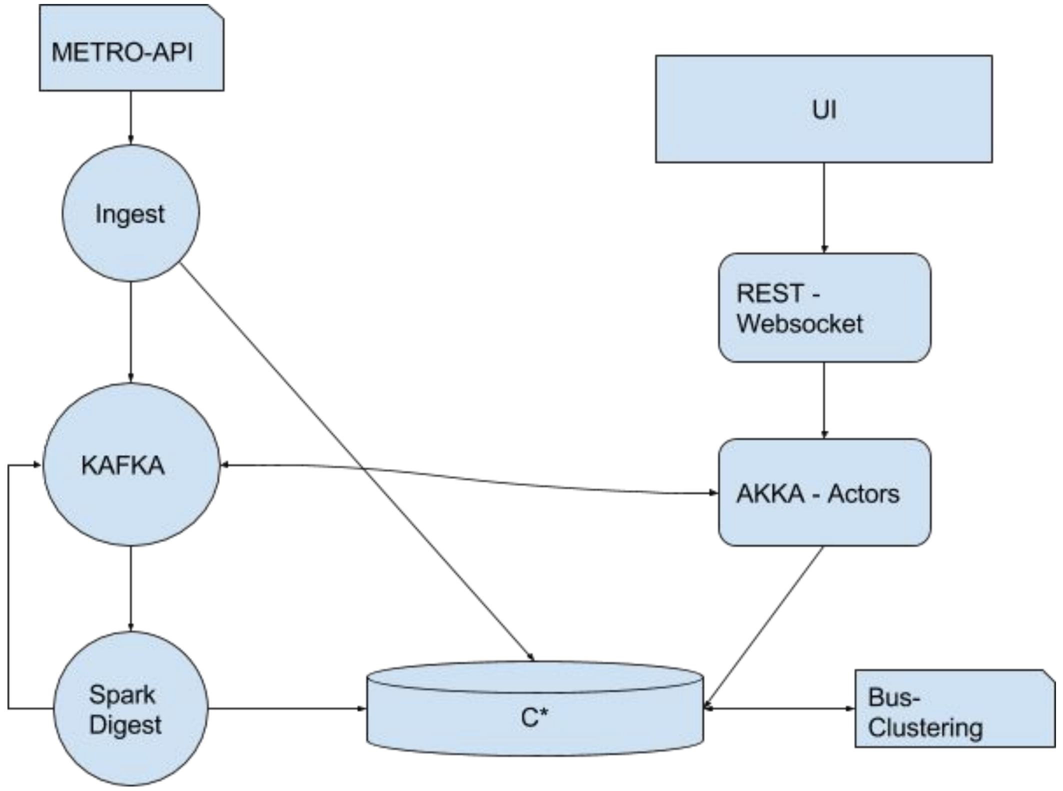
In this lab you will use a script to install:

- Spark
- Cassandra
- Kafka

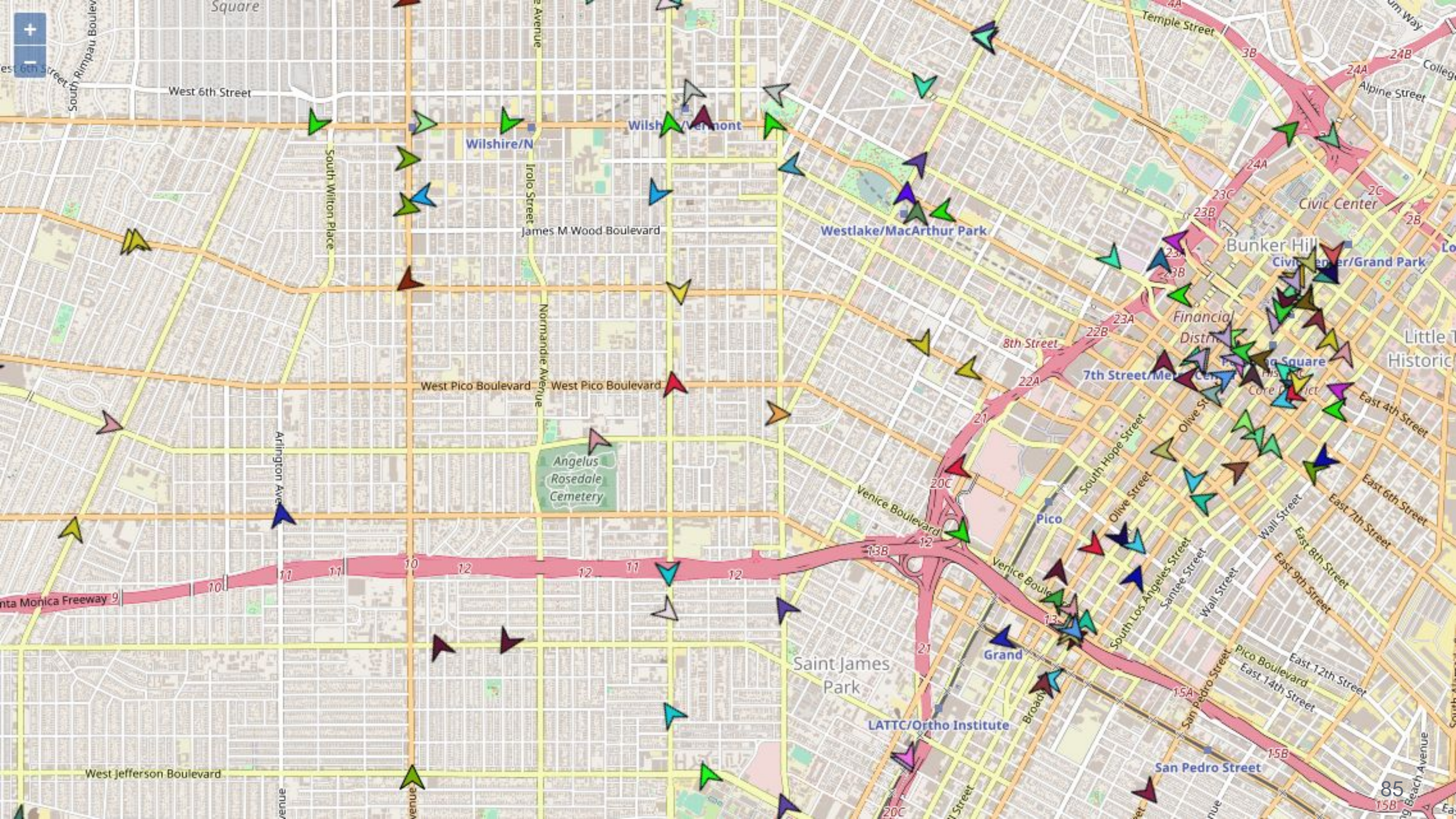
Case Study & Demo:

- Los Angeles Metro
- Final Lab

SMACK Stack Demo: Los Angeles Metro



Available for you to try at: https://github.com/mesosphere/oscon-smack-stack


West 6th Street
Wilshire/Normandie
Wilshire/Vermont
Westlake/MacArthur Park
James M Wood Boulevard
West Pico Boulevard
Angelus Rosedale Cemetery
Santa Monica Freeway
Saint James Park
LATTC/Ortho Institute
Grand
Pico
7th Street/Metro Center
Pershing Square
Civic Center/Grand Park
Bunker Hill
Civic Center
Financial District
Historic Core
Little Tokyo Historic
San Pedro Street
West Jefferson Boulevard
Temple Street
Venice Boulevard
8th Street
East 4th Street
East 6th Street
East 7th Street
East 8th Street
East 9th Street
East 12th Street
East 14th Street
Olive Street
South Hope Street
South Los Angeles Street
Santee Street
Wall Street
San Pedro Street
Normandie Avenue
Irolo Street
South Wilton Place
Arlington Ave
Alpine Street
College

SMACK Stack Lab 3

In this lab you will:

- Generating data
- Using Akka
- Monitoring the pipeline

Next Steps:

- Community
- Get Help
- Raffle Winners

Community

Join the Community: dcos.io/community

Get Help

- Mailing List
- Slack
- StackOverflow

Get Involved

- JIRA
- GitHub
- Working Groups

Get Updates

- Twitter @dcos
- YouTube
- Meetup

Self-Service: Documentation

DC/OS Documentation: https://docs.mesosphere.com

- Versioned
- Release Notes
- Component

Service Docs: https://docs.mesosphere.com/service-docs/

- Specific to Certified Packages
- Versioned
- Release Notes

Raffle!

DC/OS

Questions?

@dcos

chat.dcos.io

users@dcos.io

/dcos
/dcos/examples
/dcos/demos