Philip Norman & Sunil Shah

SCALING LIKE TWITTER WITH APACHE MESOS

MESOSPHERE

MODERN INFRASTRUCTURE

Dan the Datacenter Operator

- Doesn’t sleep very well
- Loves automation
- Wants to control what runs in his datacenter

Alice the Application Developer

- Finds setting up infrastructure tedious
- Wants her application to be deployed as quickly as possible

Modern Infrastructure

3 TENETS

Clean separation of responsibilities

No more 3am wake ups

Easy programmatic deployment

Modern Infrastructure

CLEAN SEPARATION

Before

- Dan cares about his hardware and Alice’s software that runs on it
- Alice cares about her software and what hardware Dan provides

Now

- With Mesos, all the nodes are provisioned exactly the same (but may have heterogenous hardware).
- Dan doesn’t care what software is deployed since applications are well encapsulated.
- Alice doesn’t care where her software is deployed because it’s easy enough to scale up and down.

NO MORE 3AM WAKE UPS

Before

- Dan had to react every time an application or machine went down.

Now

- Mesos and Marathon monitor running tasks.
- If a task fails or is lost (due to a machine going offline), Mesos communicates that to Marathon.
- Marathon restarts the application.
- Dan gets to sleep peacefully!

Modern Infrastructure

EASY PROGRAMMATIC DEPLOYMENT

Before

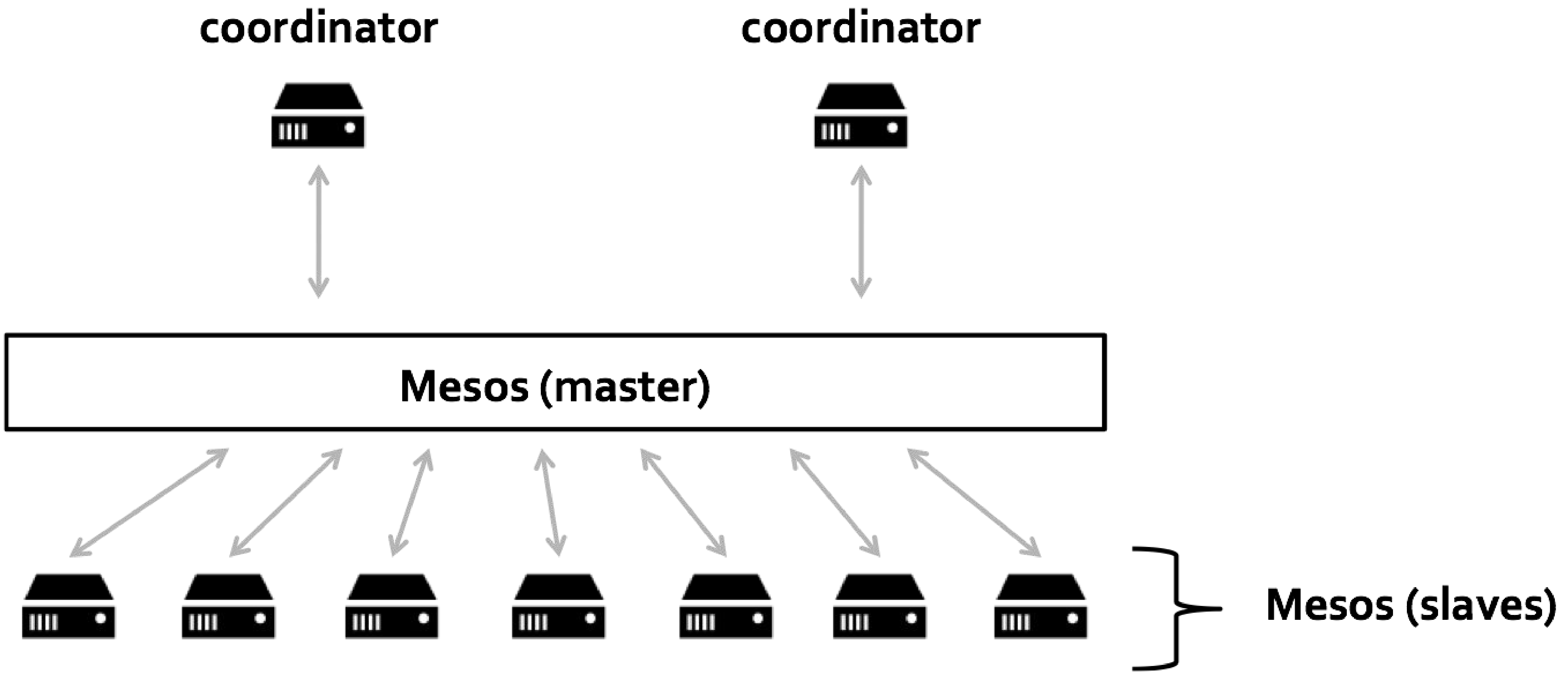
- Servers were handcrafted.
- Deploying new or updated software would require oversight and involvement from both Alice and Dan.

Now

- Dan provides Alice with her own instance of Marathon that makes it hard for her to take down someone else’s application.
- Running applications are isolated from each other by Mesos.
- Marathon offers a nice API that allows Alice to easily deploy new versions safely.

Apache Mesos

LAYER OF ABSTRACTION



coordinator

coordinator

Mesos (master)

Mesos (slaves)

Apache Mesos

INTRODUCTION

INTRODUCTION

Apache Mesos is a **cluster resource manager**.

It handles:

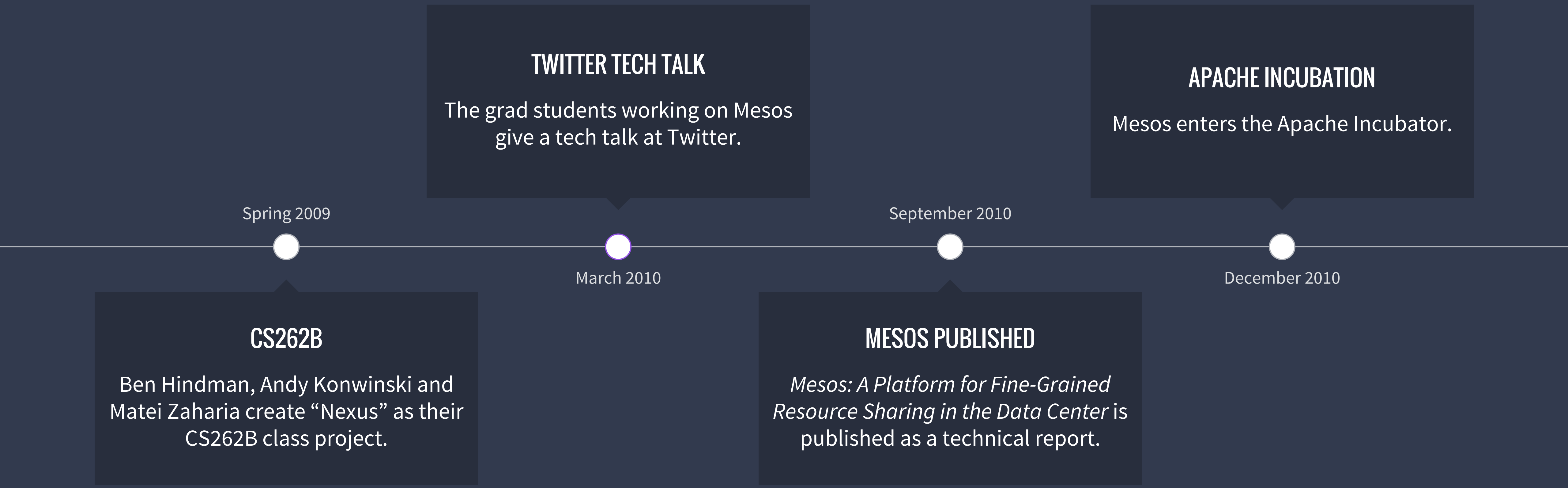
- **Aggregating resources** and **offering them to schedulers**
- **Launching tasks** (i.e. processes) on those resources
- **Communicating the state of those tasks** back to schedulers

PRODUCTION CUSTOMERS AND MESOS USERS

Bloomberg
verizon
NETFLIX
TWO SIGMA
PayPal
ebay
airbnb
HubSpot
Government Agencies
yelp

MESOS: ORIGINS

THE BIRTH OF MESOS



Spring 2009

CS262B

Ben Hindman, Andy Konwinski and Matei Zaharia create “Nexus” as their CS262B class project.

March 2010

TWITTER TECH TALK

The grad students working on Mesos give a tech talk at Twitter.

September 2010

MESOS PUBLISHED

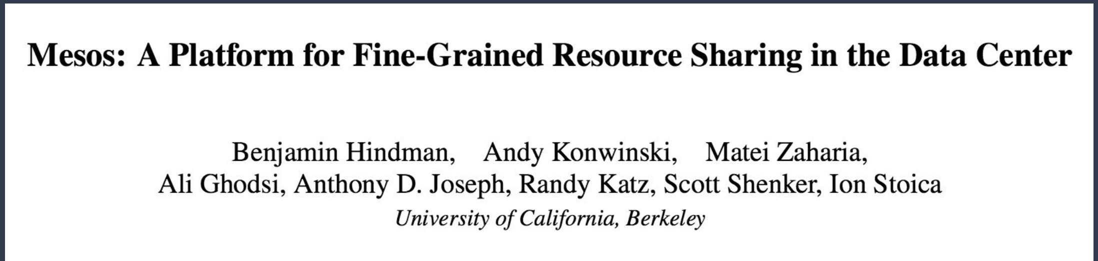
Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center is published as a technical report.

December 2010

APACHE INCUBATION

Mesos enters the Apache Incubator.

TECHNOLOGY



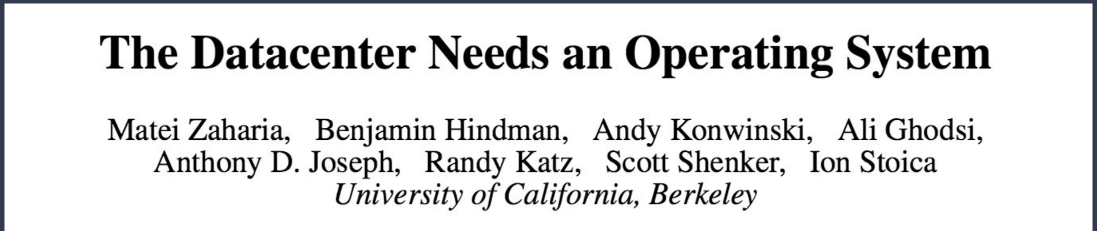
Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center

Benjamin Hindman, Andy Konwinski, Matei Zaharia,
Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
University of California, Berkeley

Sharing resources between batch processing frameworks

- Hadoop
- MPI
- Spark

VISION



The Datacenter Needs an Operating System

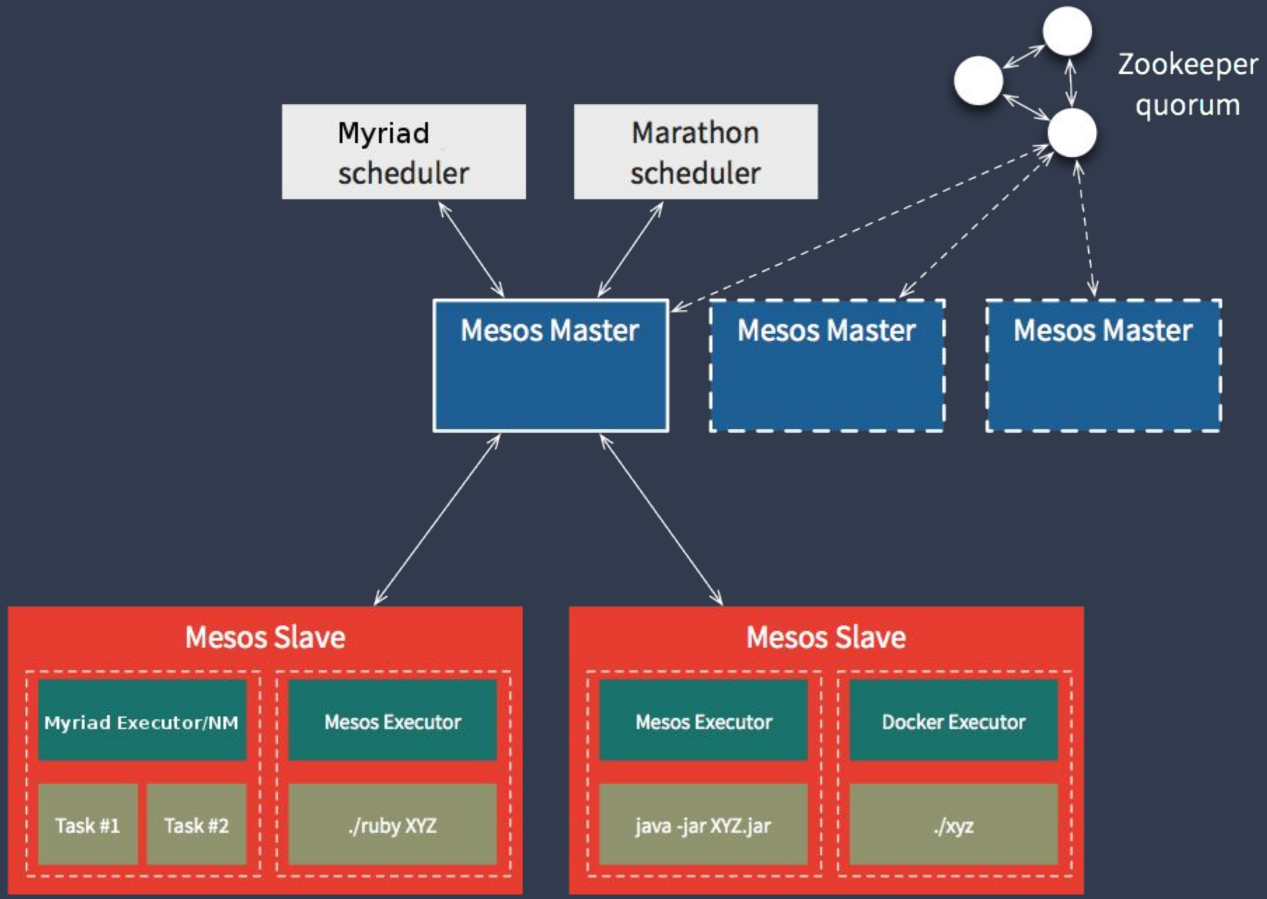
Matei Zaharia, Benjamin Hindman, Andy Konwinski, Ali Ghodsi,
Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
University of California, Berkeley

What does an operating system provide?

- Resource management
- Programming abstractions
- Security
- Monitoring, debugging, logging

MESOS FUNDAMENTALS

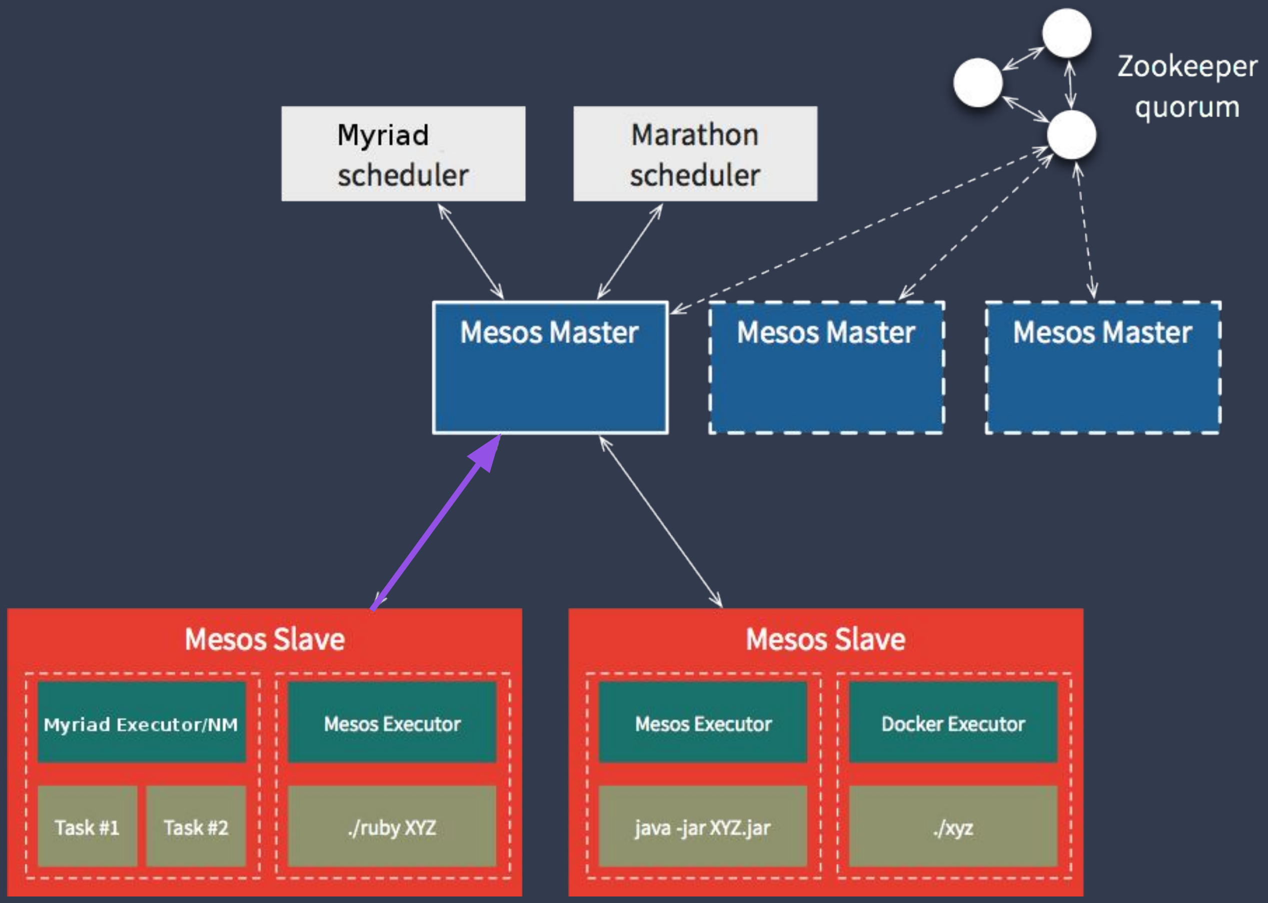
ARCHITECTURE



MESOS FUNDAMENTALS

ARCHITECTURE

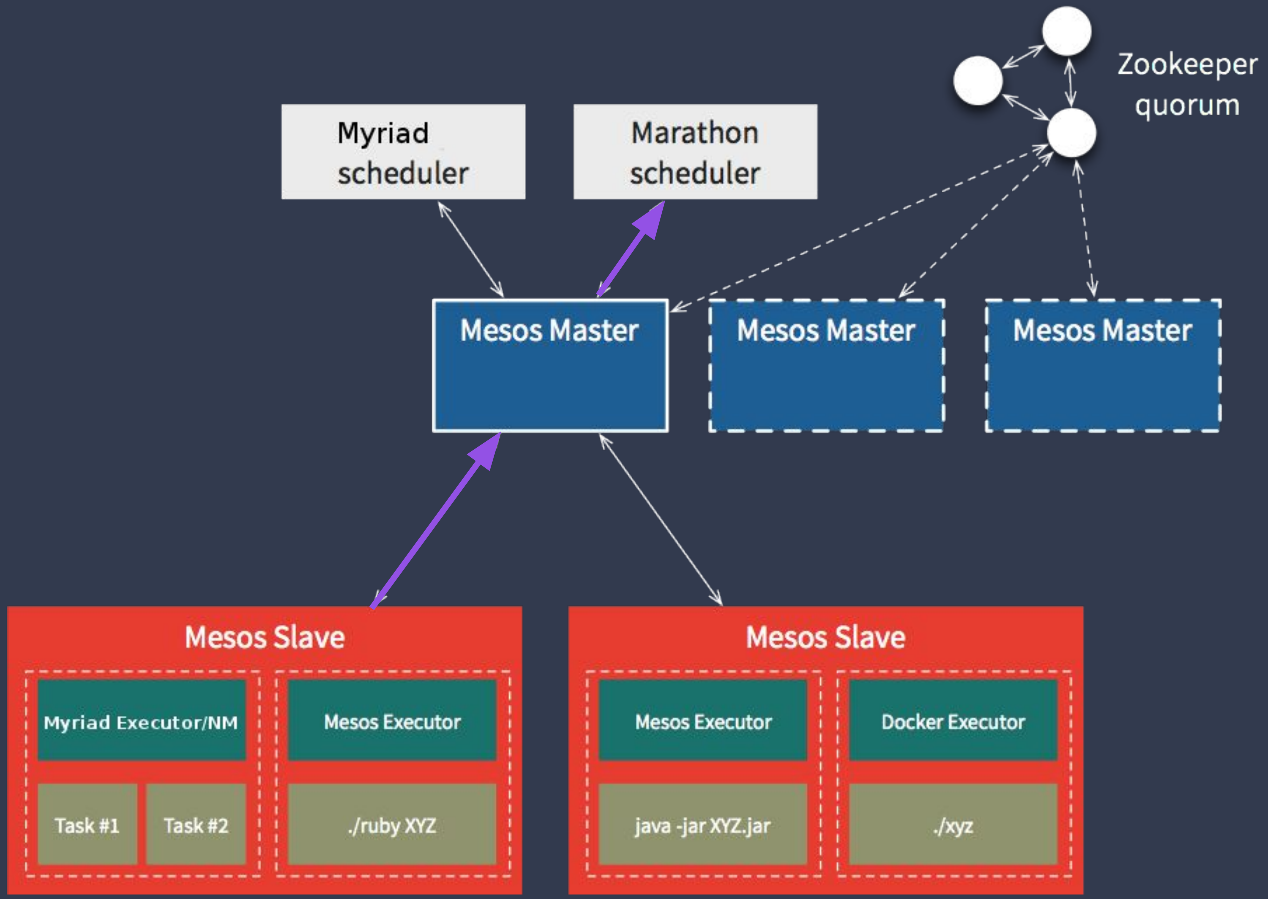
- Agents advertise resources to Master
- Master offers resources to Scheduler
- Scheduler rejects/uses resources
- Agents report task status to Master



MESOS FUNDAMENTALS

ARCHITECTURE

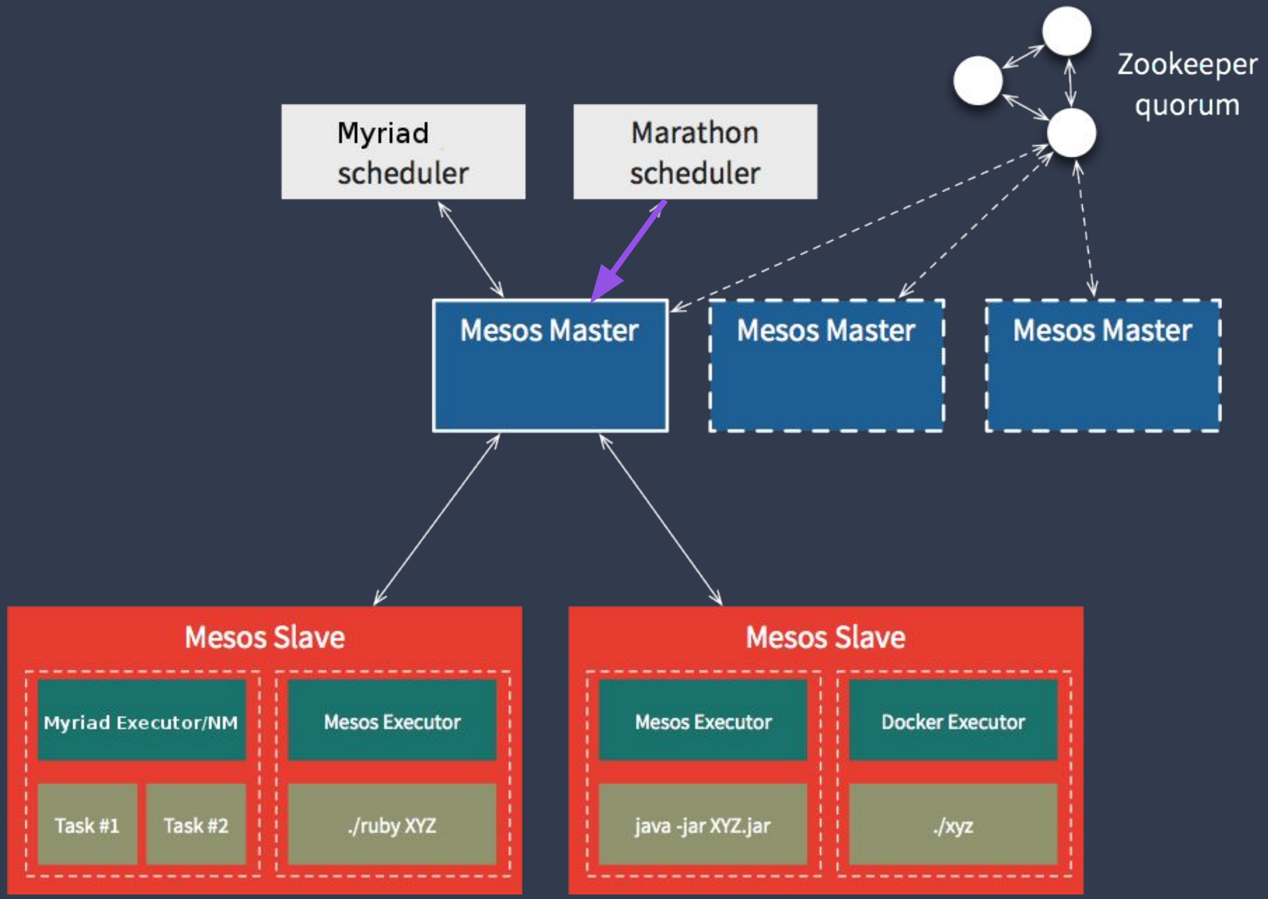
- Agents advertise resources to Master
- Master offers resources to Scheduler
- Scheduler rejects/uses resources
- Agents report task status to Master



MESOS FUNDAMENTALS

ARCHITECTURE

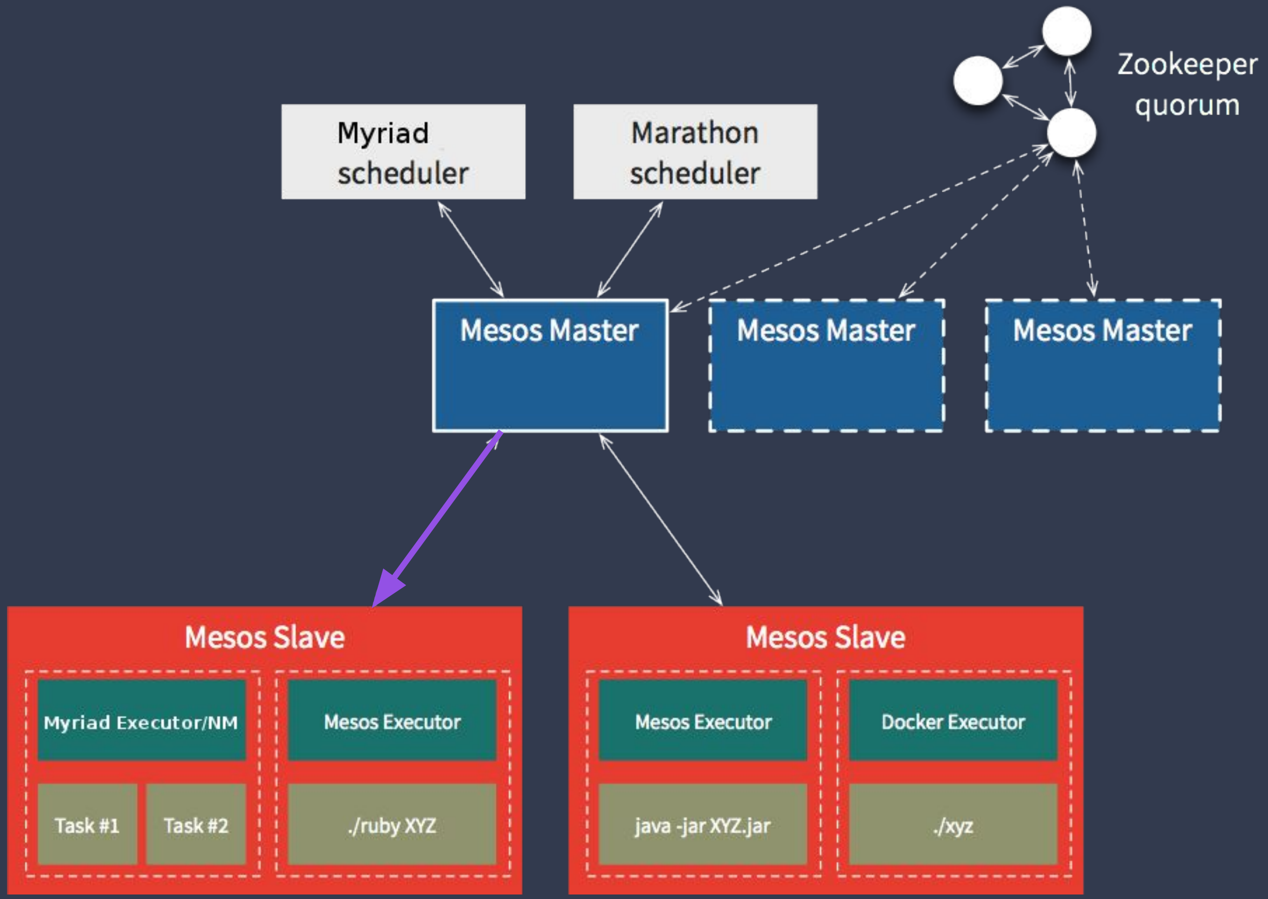
- Agents advertise resources to Master
- Master offers resources to Scheduler
- Scheduler rejects/uses resources
- Agents report task status to Master



MESOS FUNDAMENTALS

ARCHITECTURE

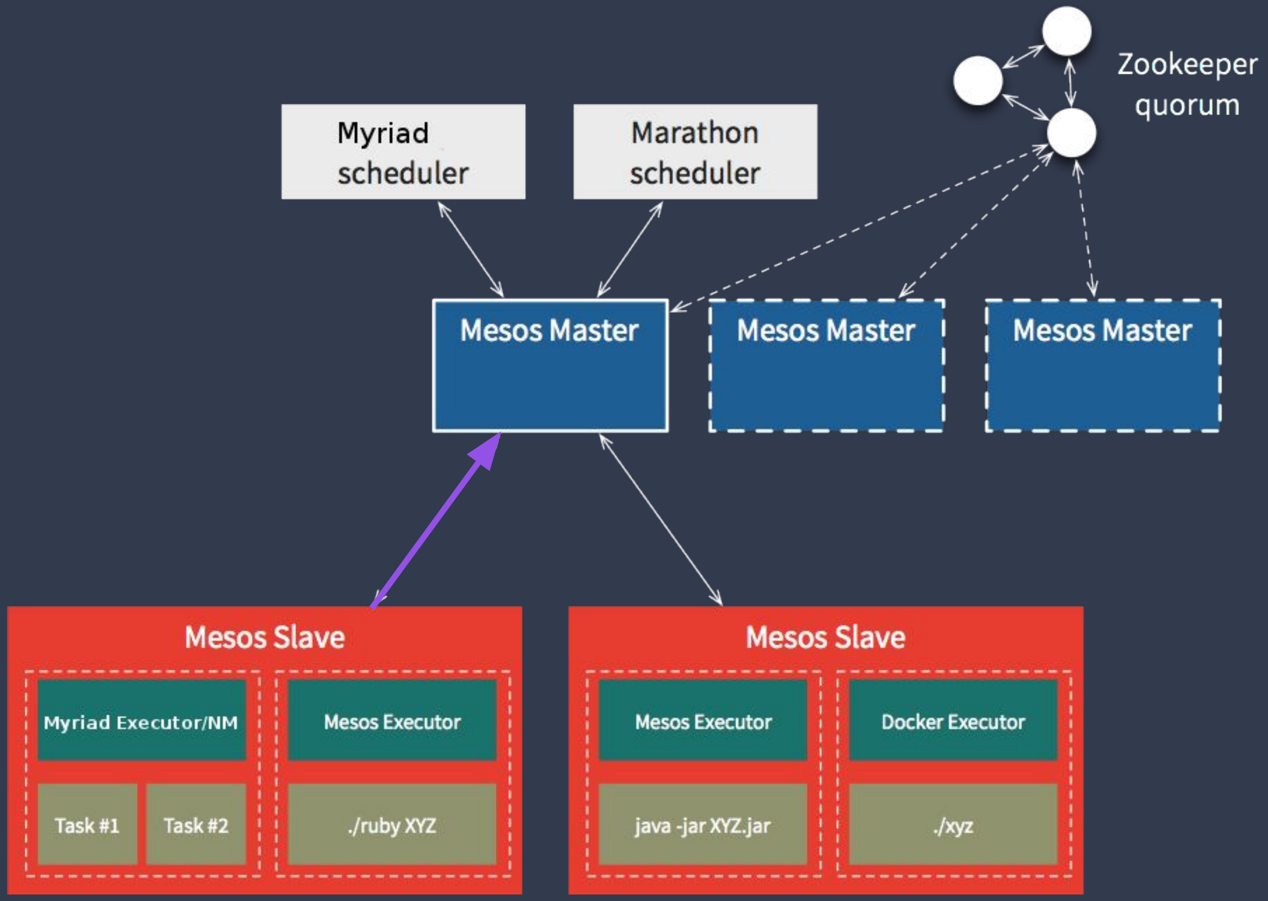
- Agents advertise resources to Master
- Master offers resources to Scheduler
- Scheduler rejects/uses resources
- Agents report task status to Master



MESOS FUNDAMENTALS

ARCHITECTURE

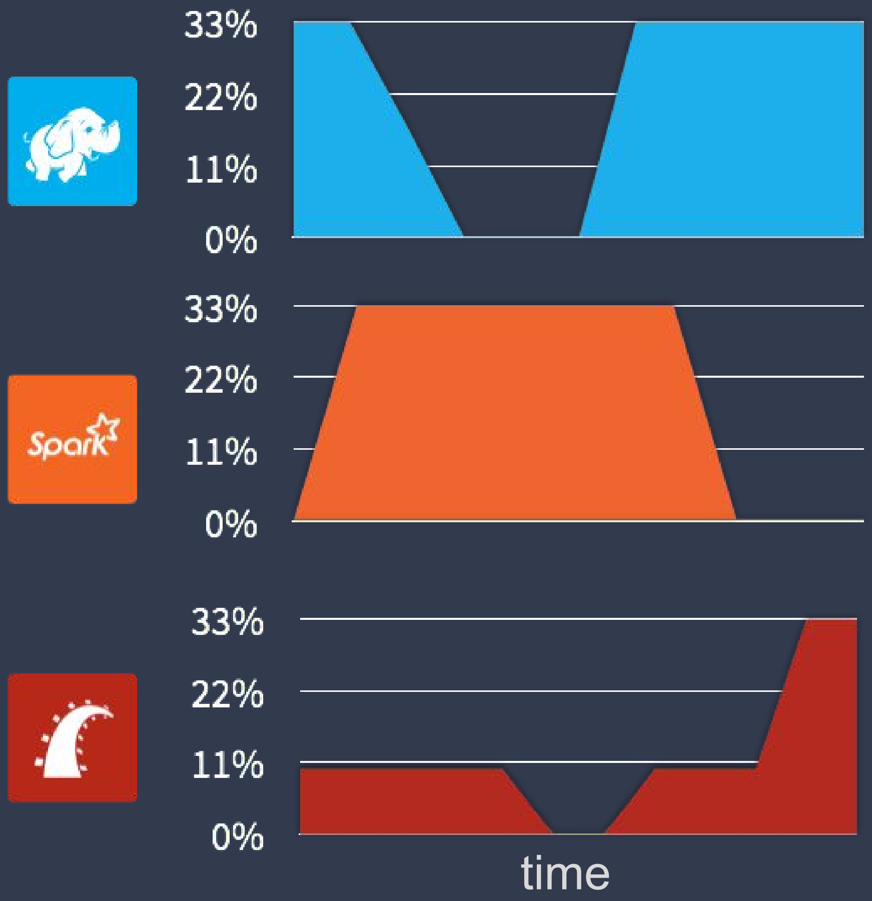
- Agents advertise resources to Master
- Master offers resources to Scheduler
- Scheduler rejects/uses resources
- Agents report task status to Master



KEEP IT STATIC

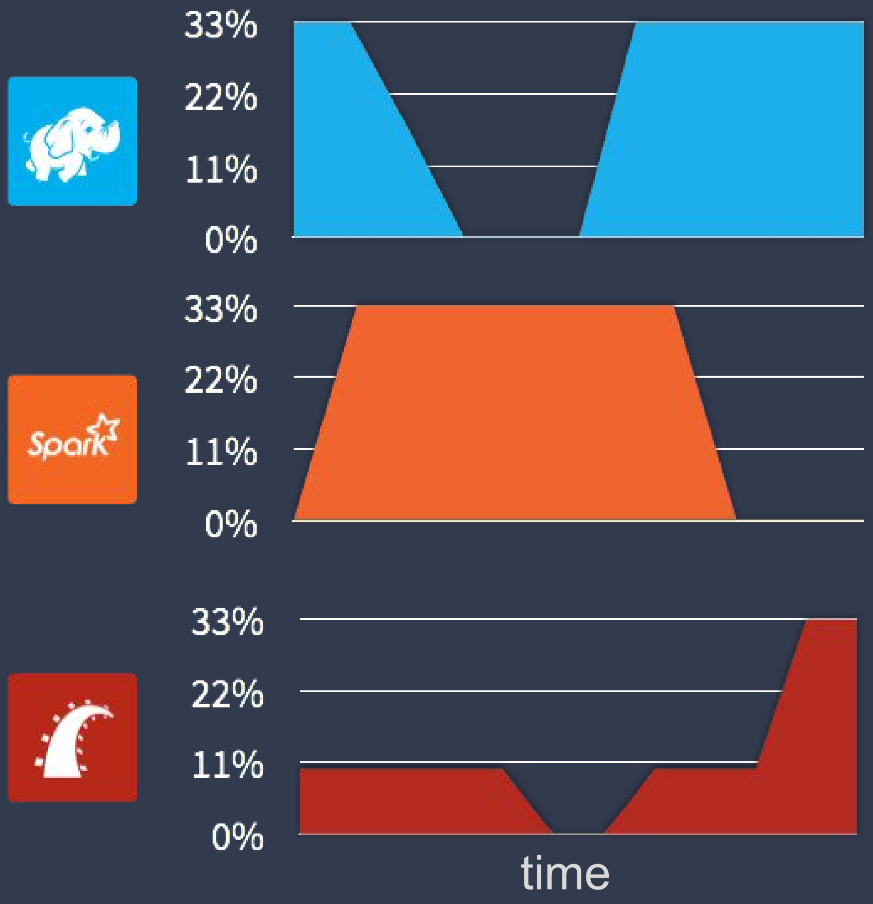
A naive approach to handling varied app requirements: **static partitioning**.

This can cope with heterogeneity, but is very expensive.



KEEP IT STATIC

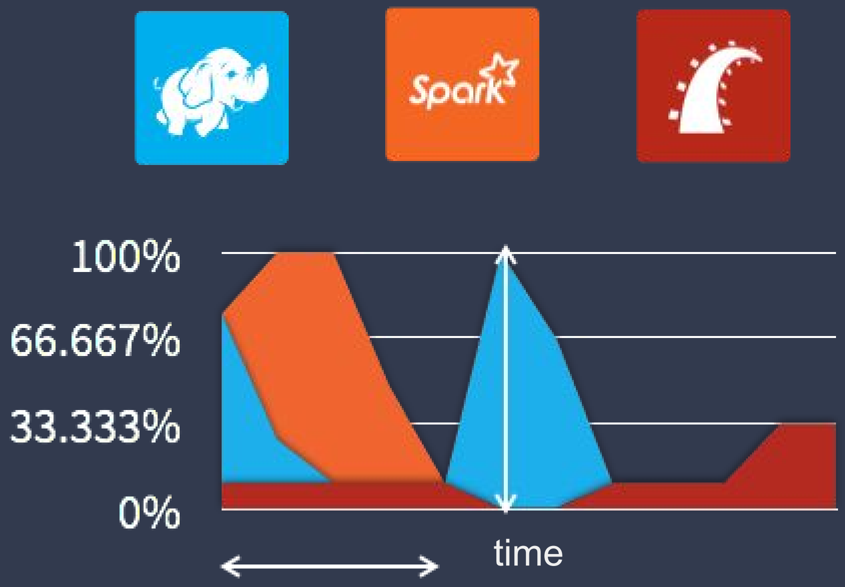
Maintaining sufficient headroom to handle peak workloads on all partitions leads to **poor utilisation** overall.



© 2015 Mesosphere, Inc. All Rights Reserved.

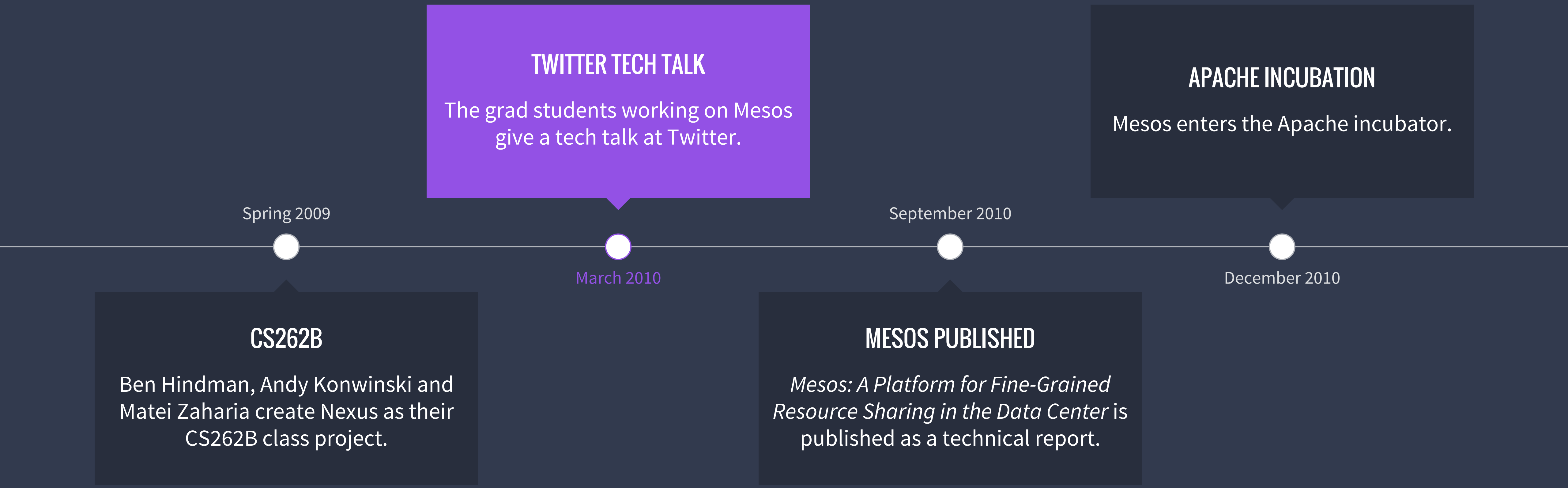
SHARED RESOURCES

Multiple frameworks can use the same cluster resources, with their share adjusting dynamically.



TWITTER & MESOS

THE BIRTH OF MESOS



Spring 2009

CS262B

Ben Hindman, Andy Konwinski and Matei Zaharia create Nexus as their CS262B class project.

March 2010

TWITTER TECH TALK

The grad students working on Mesos give a tech talk at Twitter.

September 2010

MESOS PUBLISHED

Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center is published as a technical report.

December 2010

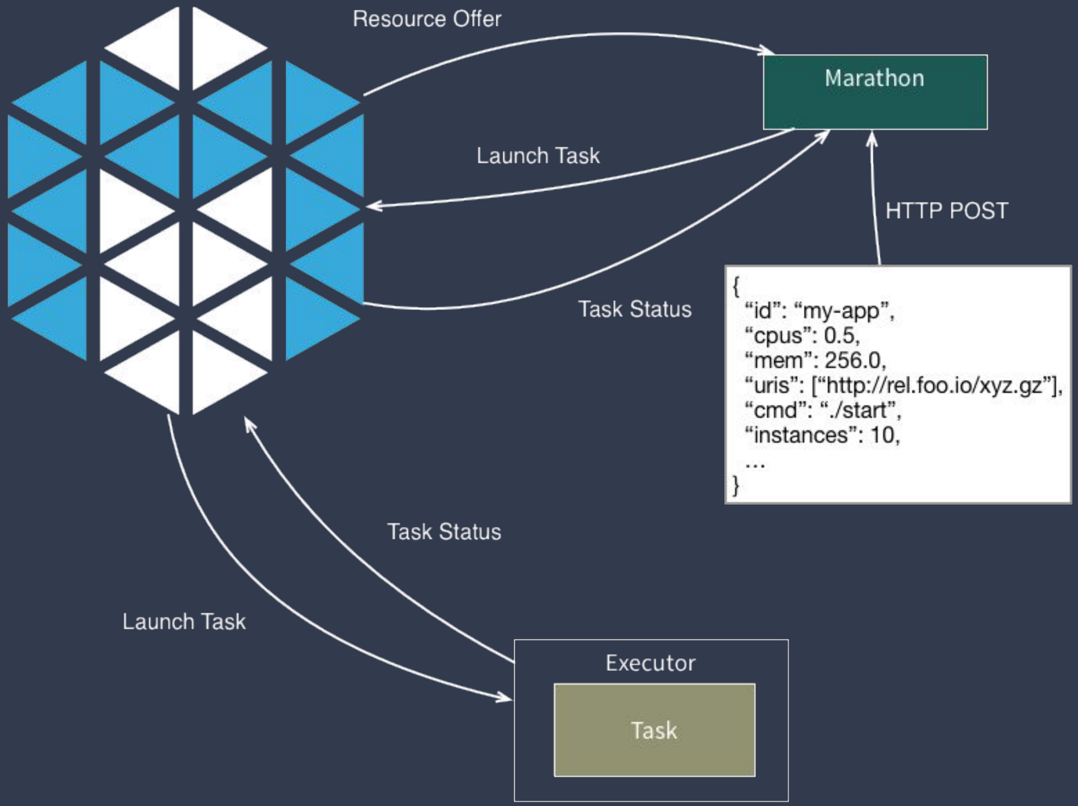
APACHE INCUBATION

Mesos enters the Apache incubator.

MESOS REALLY HELPS

- Former Google engineers at Twitter thought Mesos could provide the same functionality as Borg.

- Mesos actually works pretty well for long running services.



```
{
  “id”: “my-app”,
  “cpus”: 0.5,
  “mem”: 256.0,
  “uris”: [“http://rel.foo.io/xyz.gz”],
  “cmd”: “./start”,
  “instances”: 10,
  …
}
```

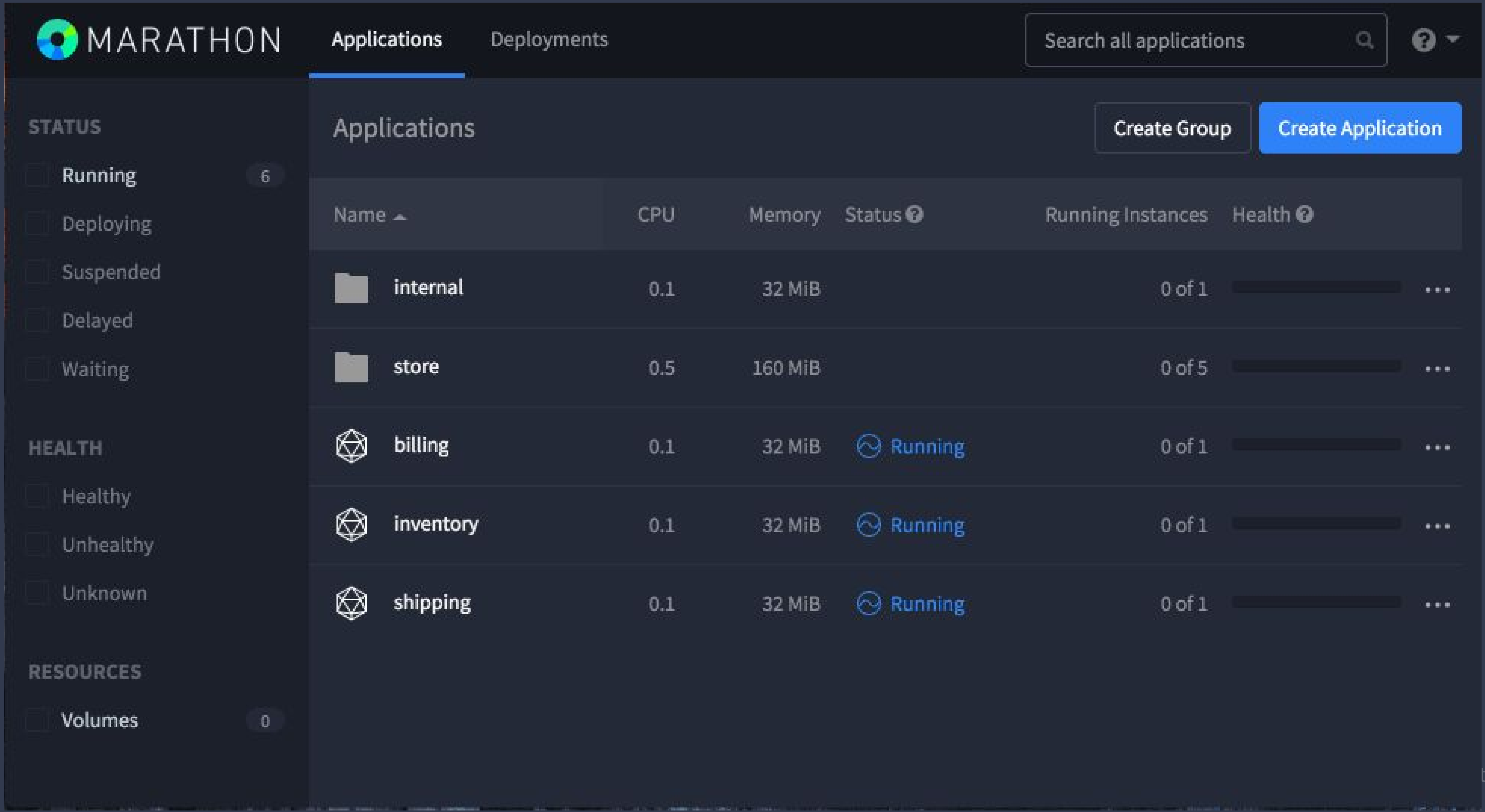
MESOS WITH MARATHON IN PRODUCTION

WHAT IS MARATHON?

- Service scheduler for Mesos
- `init.d` for long-running apps
- Your own private PaaS

Mesos with Marathon in Production

WHAT IS MARATHON?



Name	CPU	Memory	Status	Running Instances	Health
internal	0.1	32 MiB		0 of 1	
store	0.5	160 MiB		0 of 5	
billing	0.1	32 MiB	Running	0 of 1	
inventory	0.1	32 MiB	Running	0 of 1	
shipping	0.1	32 MiB	Running	0 of 1	

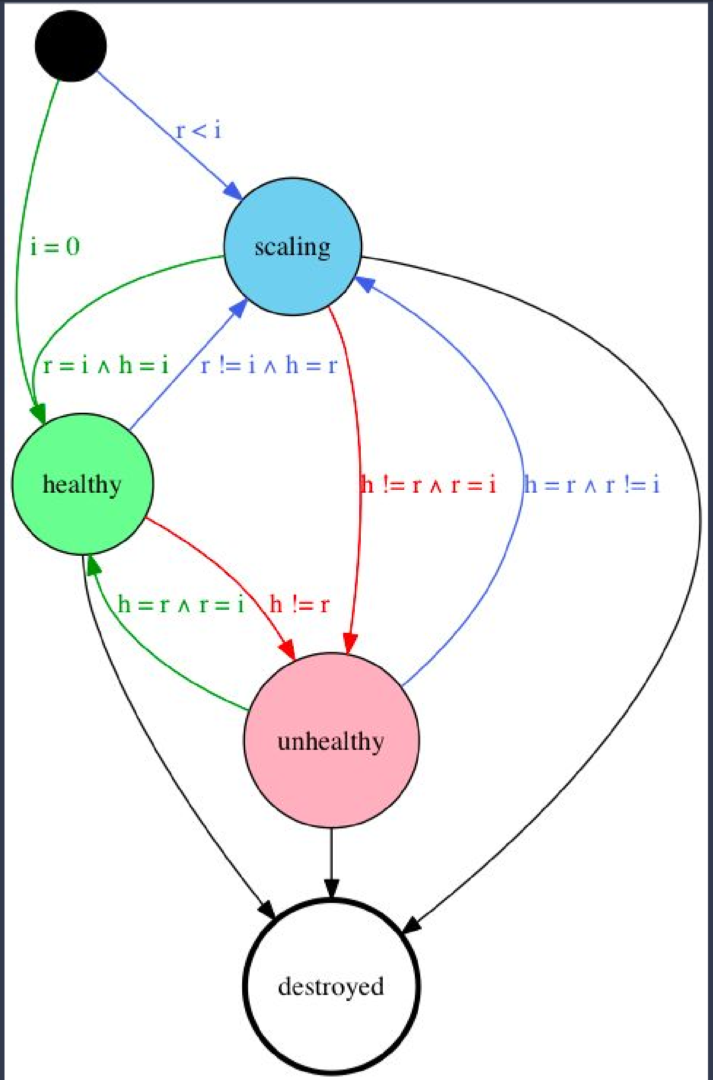
Mesos with Marathon in Production

USEFUL MARATHON FEATURES

- Start, stop, scale, update apps
- Highly available, no SPoF
- Native Docker support
- Powerful Web UI
- Fully featured REST API
- Pluggable event bus
- Artifact staging

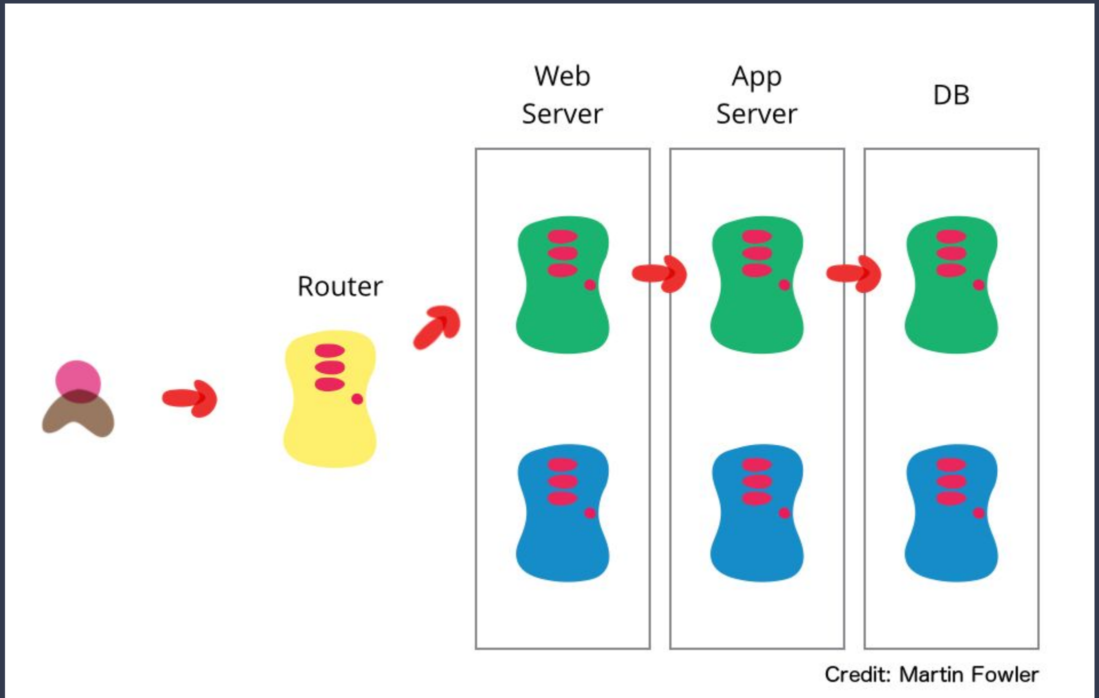
Mesos with Marathon in Production

USEFUL MARATHON FEATURES: DEPLOY LIKE FACEBOOK



- Application versioning
- Rolling deploy / restart
- Deployment strategies

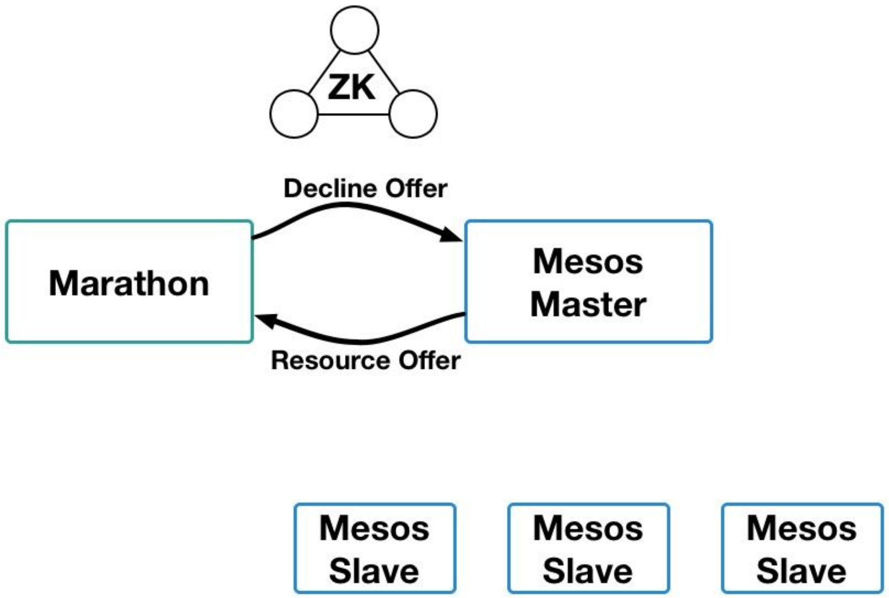
USEFUL MARATHON FEATURES: DEPLOY LIKE A TELCO

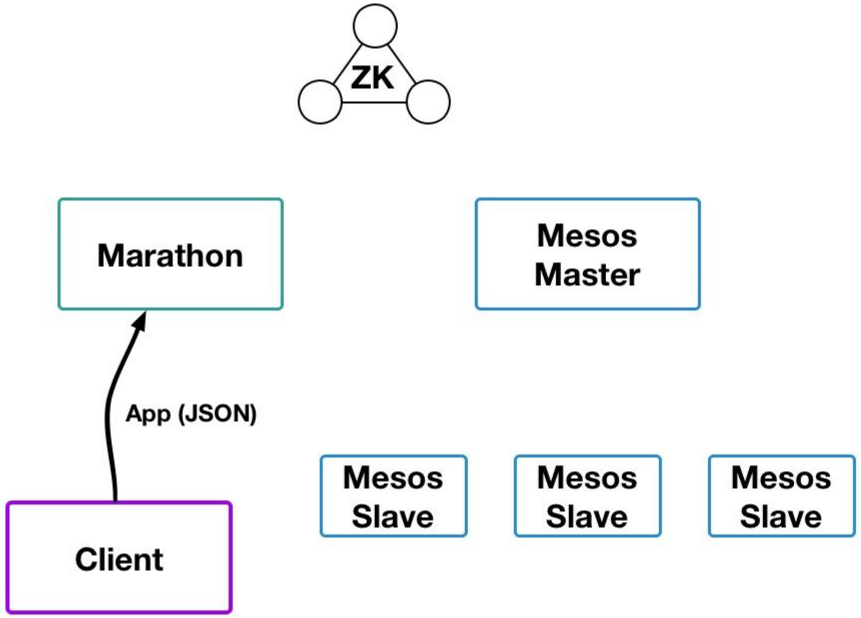


Credit: Martin Fowler

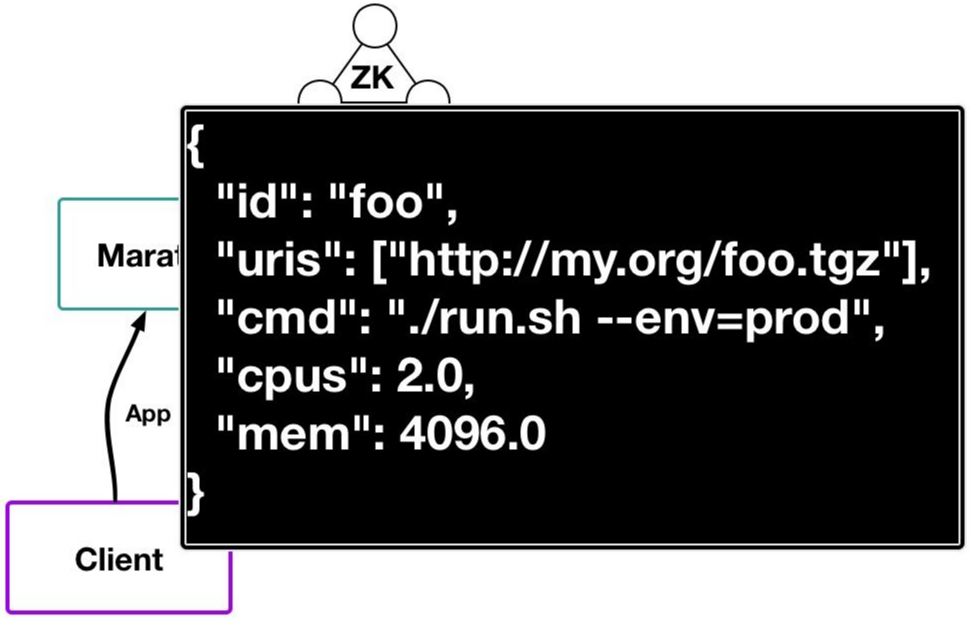
- Application versioning
- Hot/hot new/old clusters
- Authentic scale testing
- Manual *and* automated testing

MESOS WITH MARATHON IN ACTION

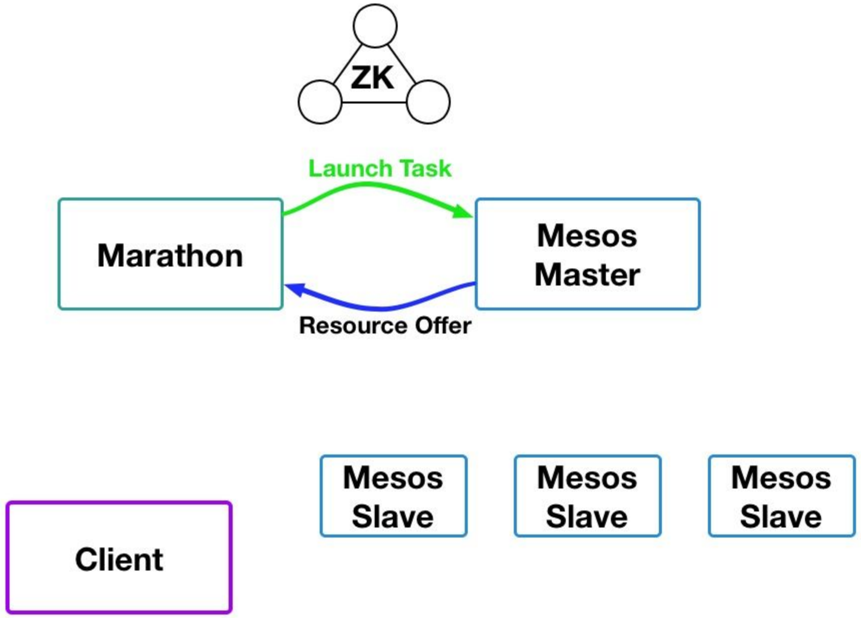

ZK
Decline Offer
Marathon
Mesos Master
Resource Offer
Mesos Slave
Mesos Slave
Mesos Slave

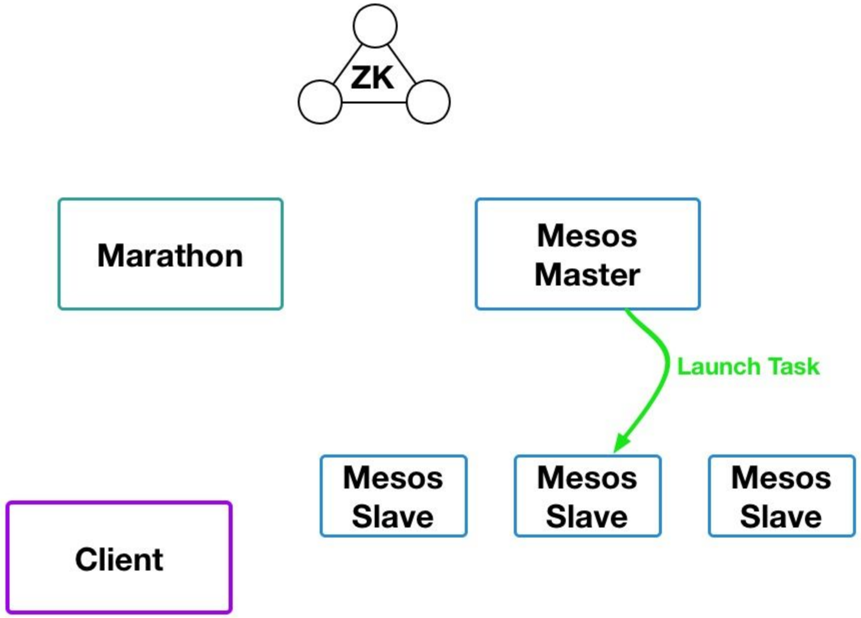


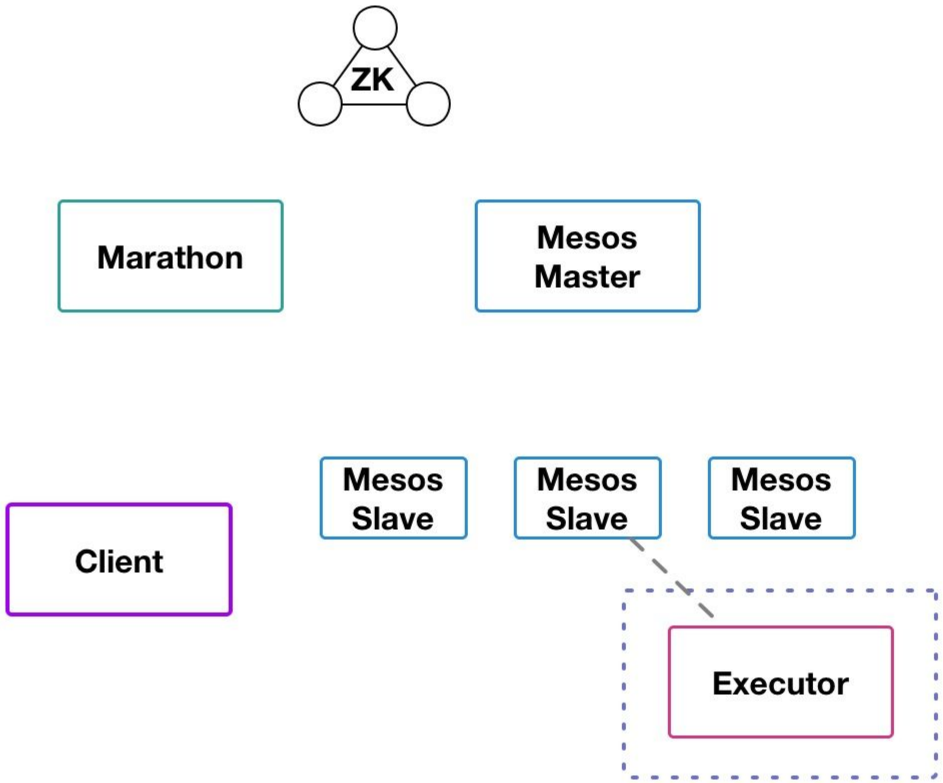
© 2015 Mesosphere, Inc. All Rights Reserved.

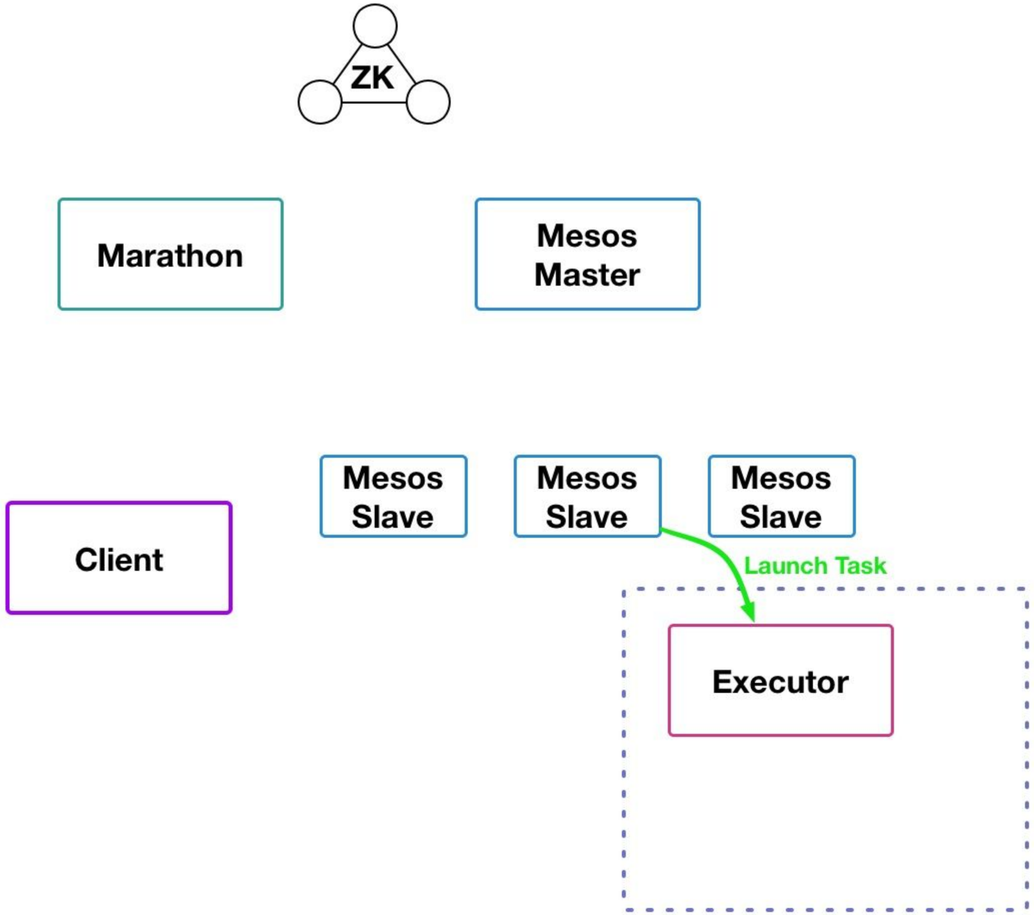


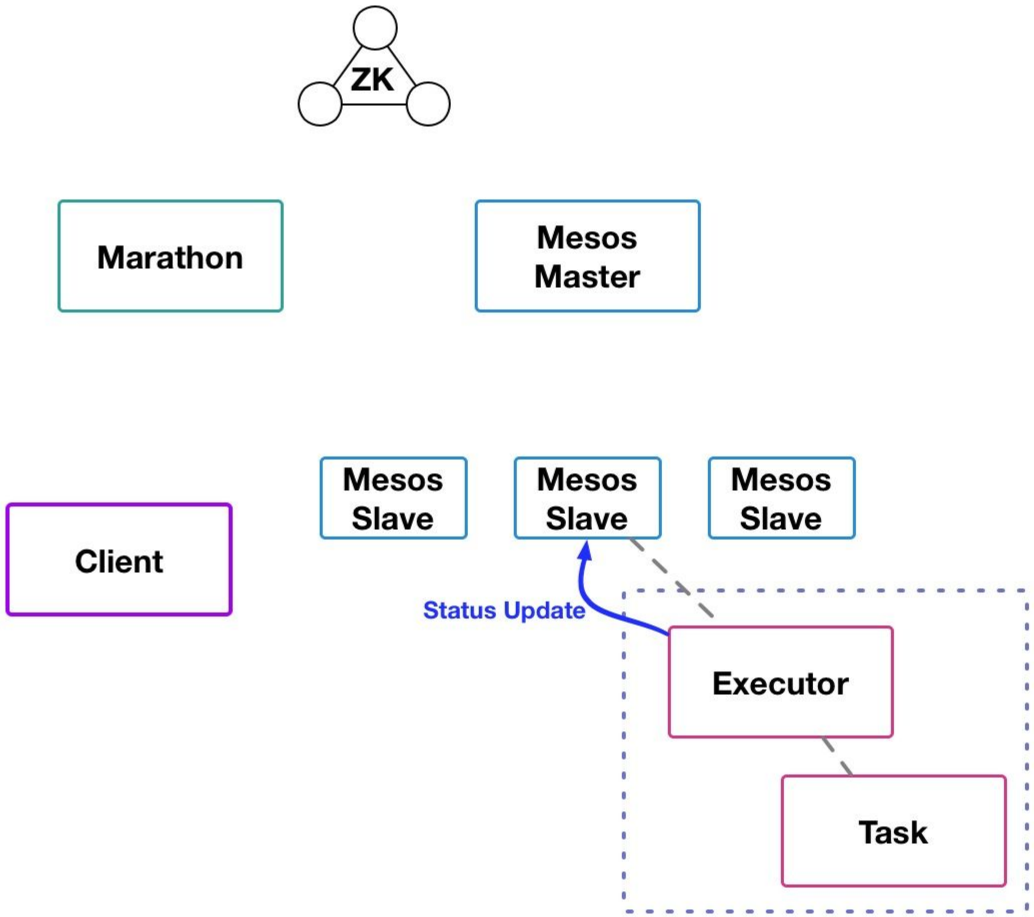
```
{
  "id": "foo",
  "uris": ["http://my.org/foo.tgz"],
  "cmd": "./run.sh --env=prod",
  "cpus": 2.0,
  "mem": 4096.0
}
```

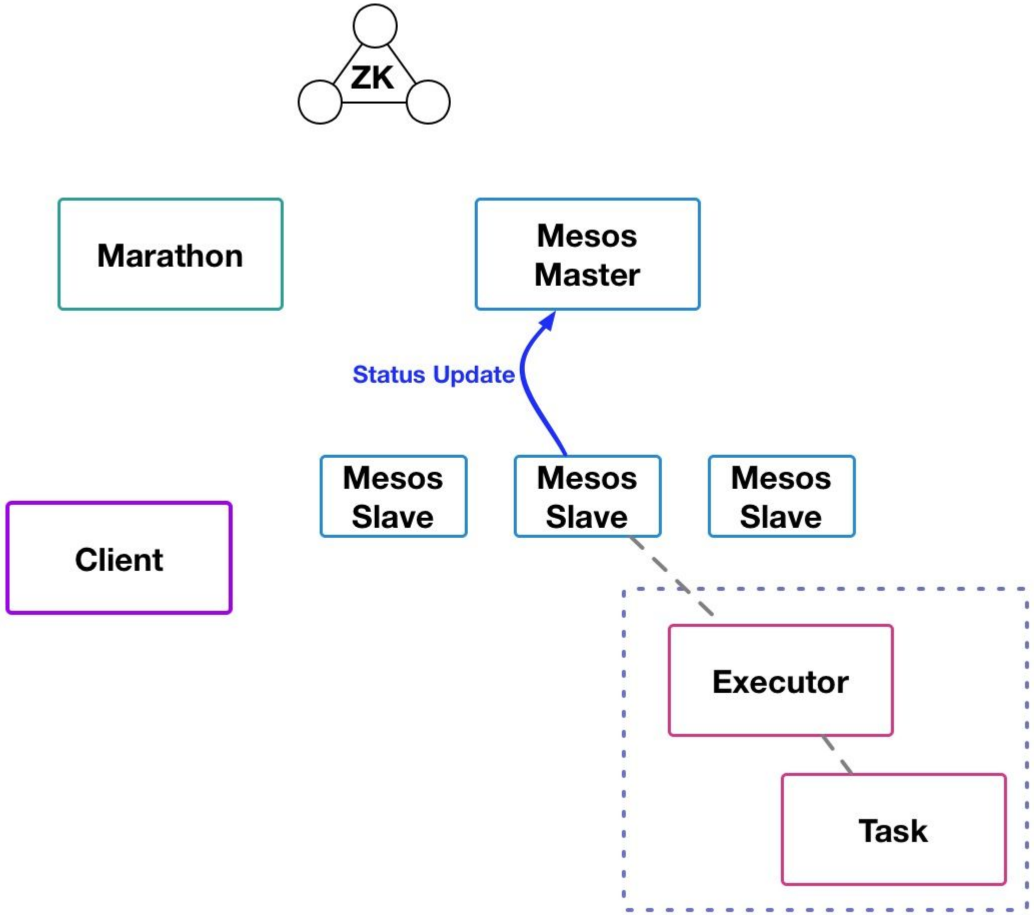

ZK
Launch Task
Marathon
Mesos Master
Resource Offer
Mesos Slave
Mesos Slave
Mesos Slave
Client

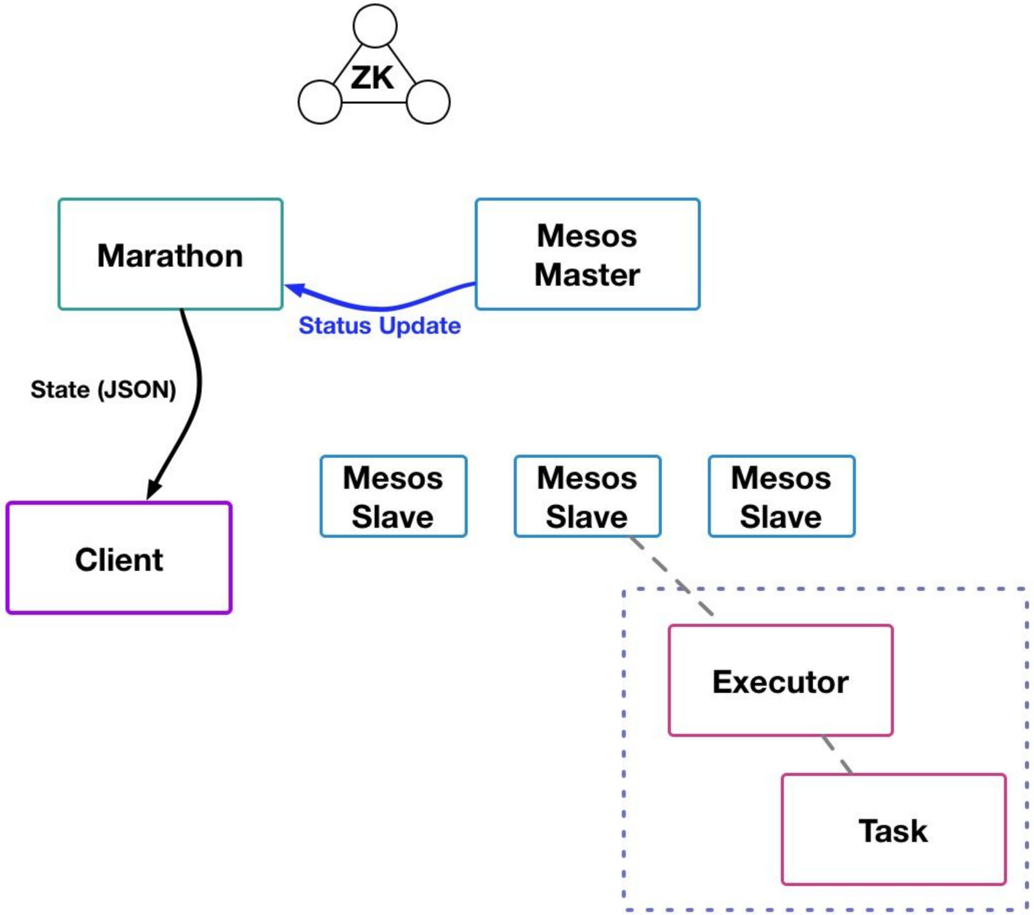

ZK
Marathon
Mesos Master
Launch Task
Mesos Slave
Mesos Slave
Mesos Slave
Client

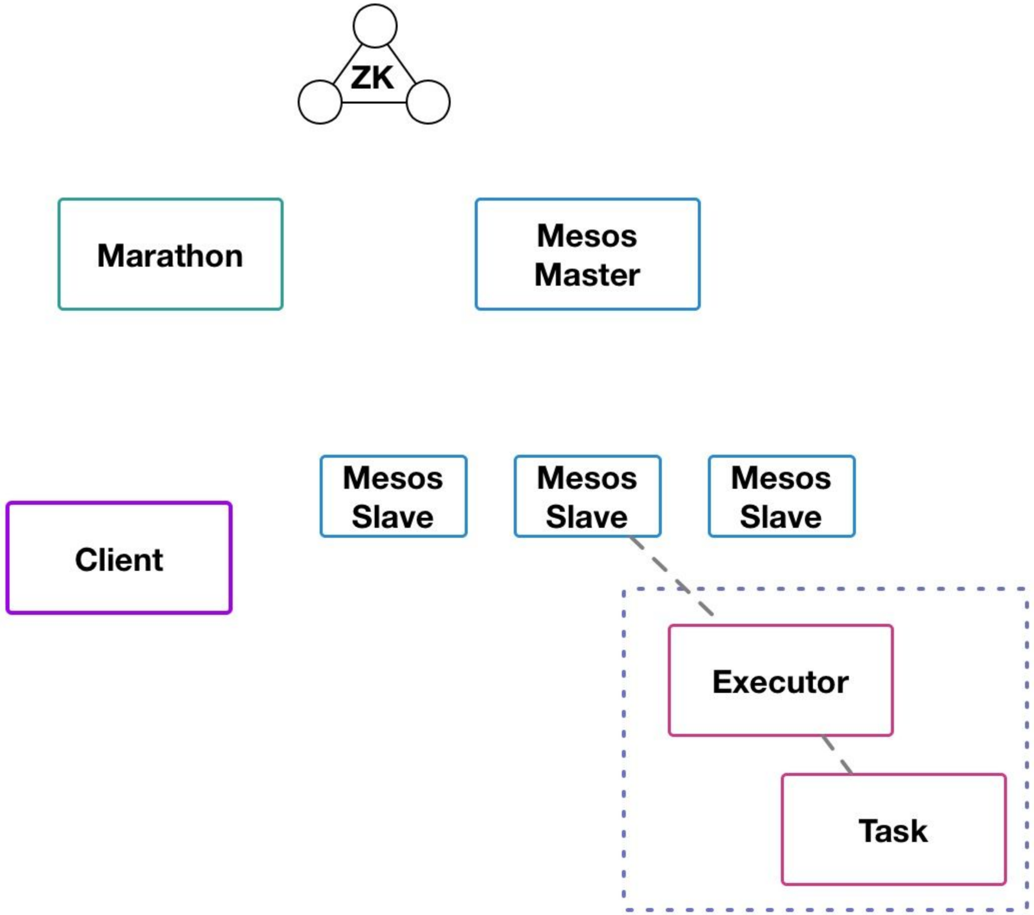

ZK
Marathon
Mesos Master
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor


ZK
Marathon
Mesos Master
Mesos Slave
Mesos Slave
Mesos Slave
Client
Launch Task
Executor


ZK
Marathon
Mesos Master
Mesos Slave
Mesos Slave
Mesos Slave
Client
Status Update
Executor
Task

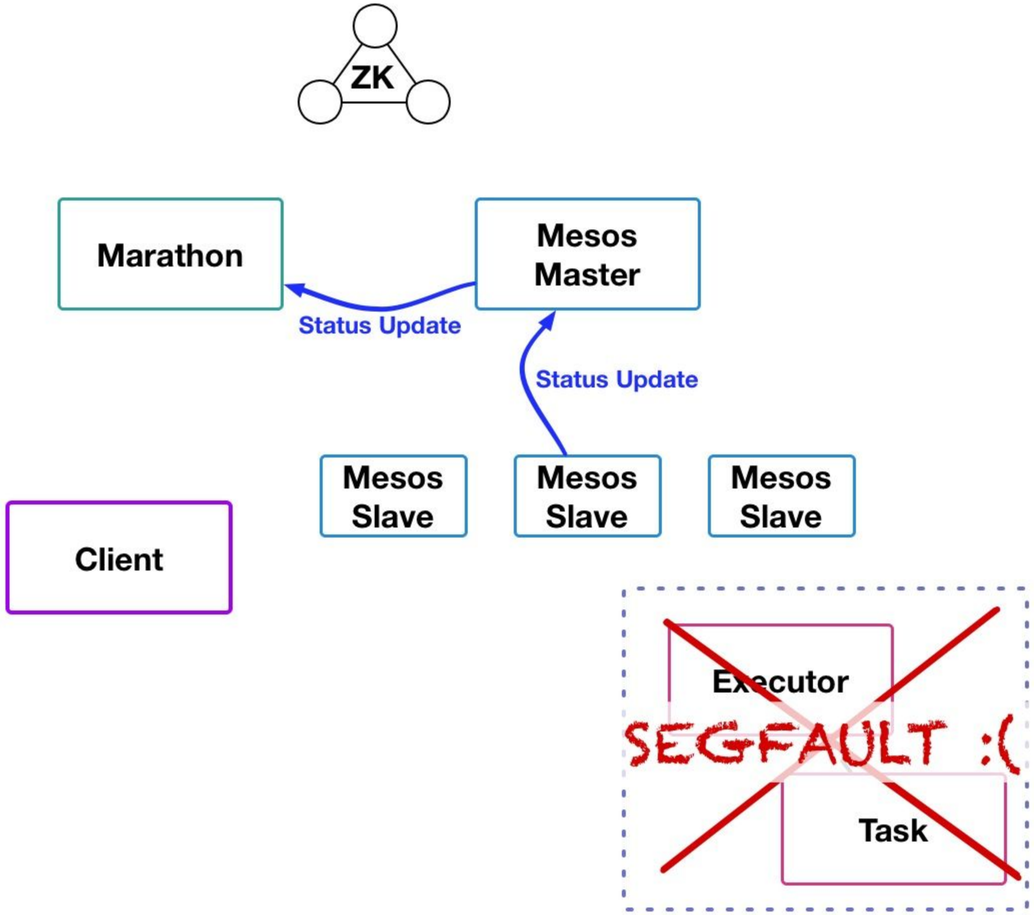

ZK
Marathon
Mesos Master
Status Update
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor
Task

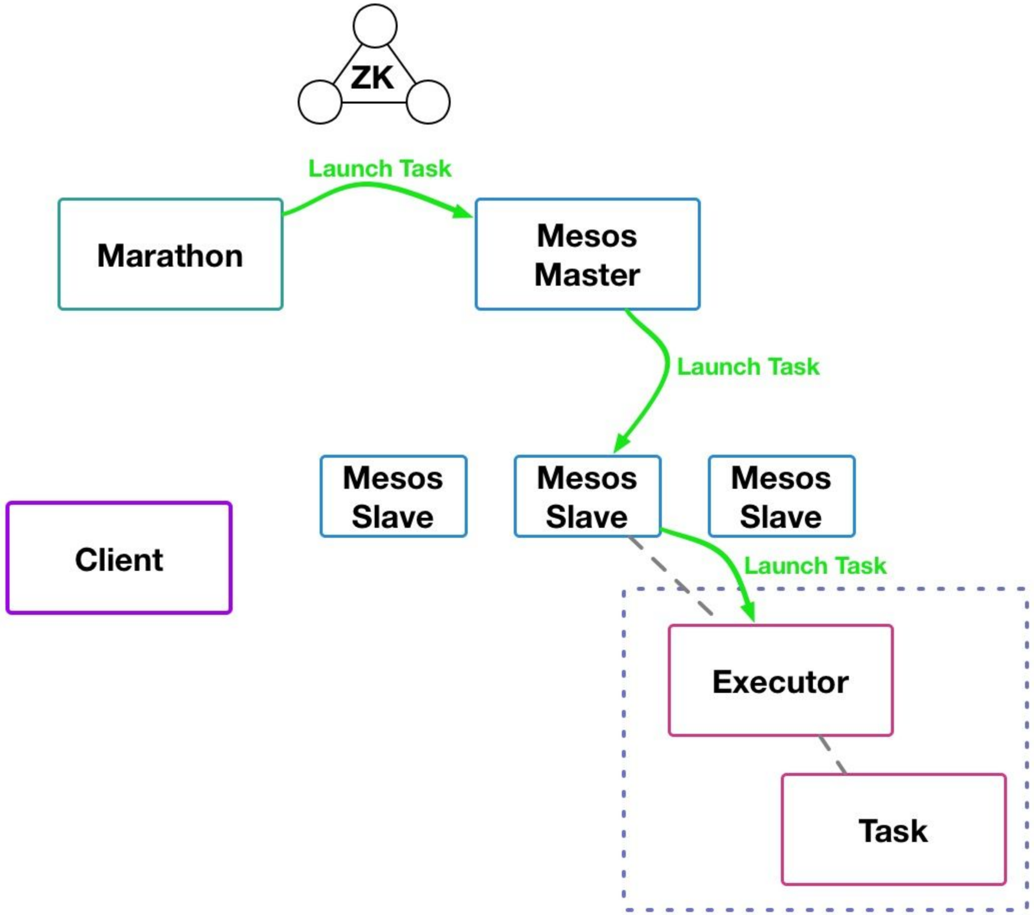

ZK
Marathon
Mesos Master
Status Update
State (JSON)
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor
Task

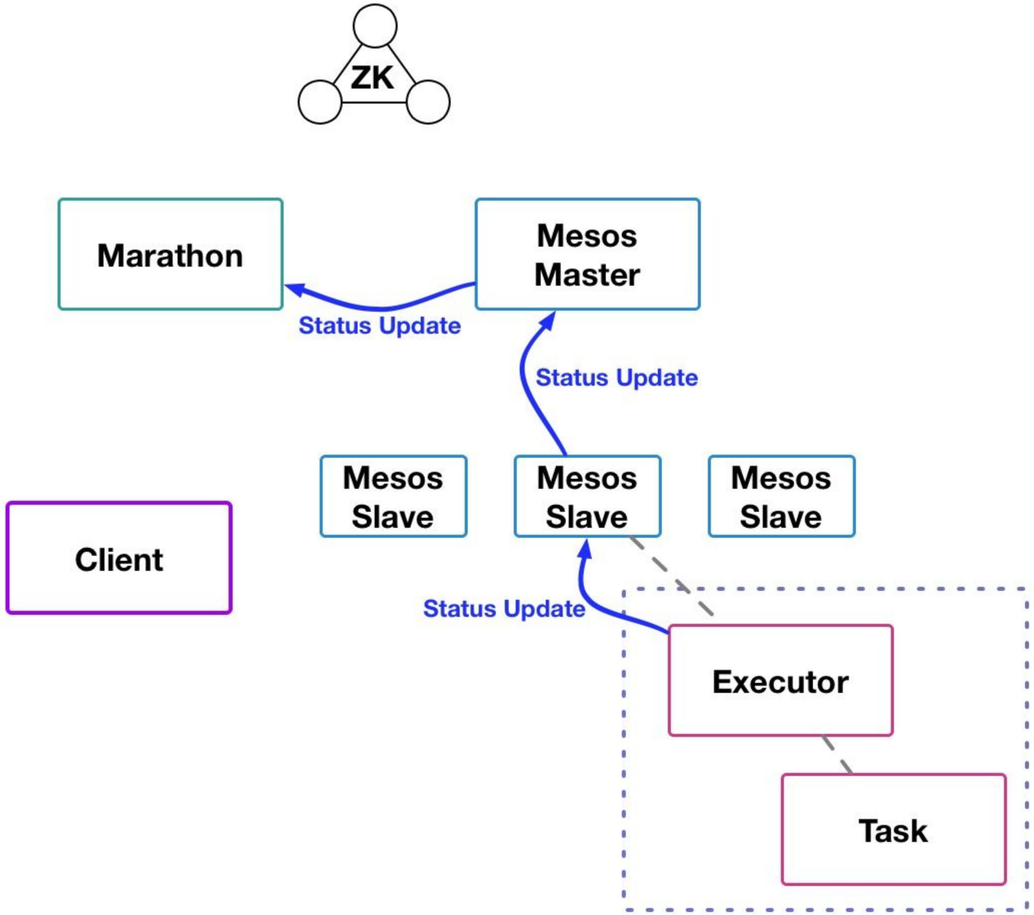

ZK
Marathon
Mesos Master
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor
Task

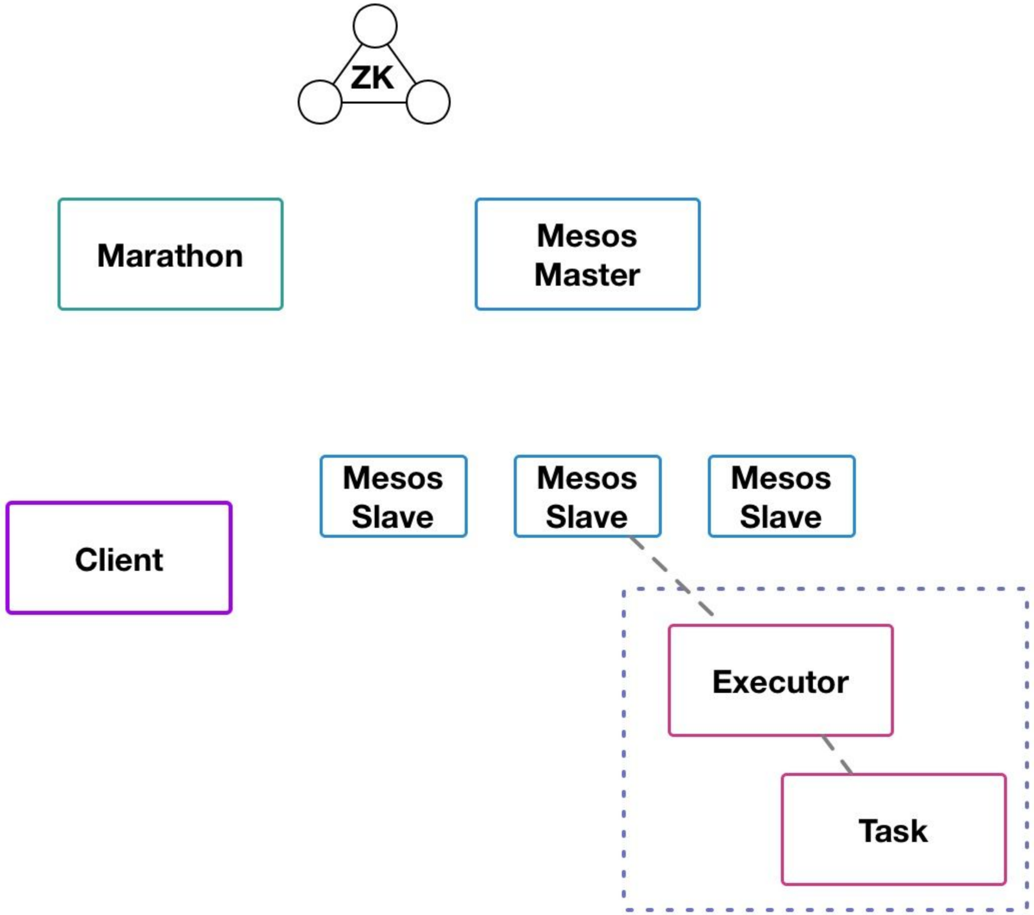
Mesos with Marathon in Action

TASK FAILURE


ZK
Marathon
Mesos Master
Status Update
Status Update
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor
SEGFAULT :(
Task


ZK
Launch Task
Marathon
Mesos Master
Launch Task
Mesos Slave
Mesos Slave
Mesos Slave
Client
Launch Task
Executor
Task


ZK
Marathon
Mesos Master
Status Update
Status Update
Mesos Slave
Mesos Slave
Mesos Slave
Client
Status Update
Executor
Task


ZK
Marathon
Mesos Master
Client
Mesos Slave
Mesos Slave
Mesos Slave
Executor
Task

Mesos with Marathon in Production

SERVICE DISCOVERY

How do my applications discover each other?

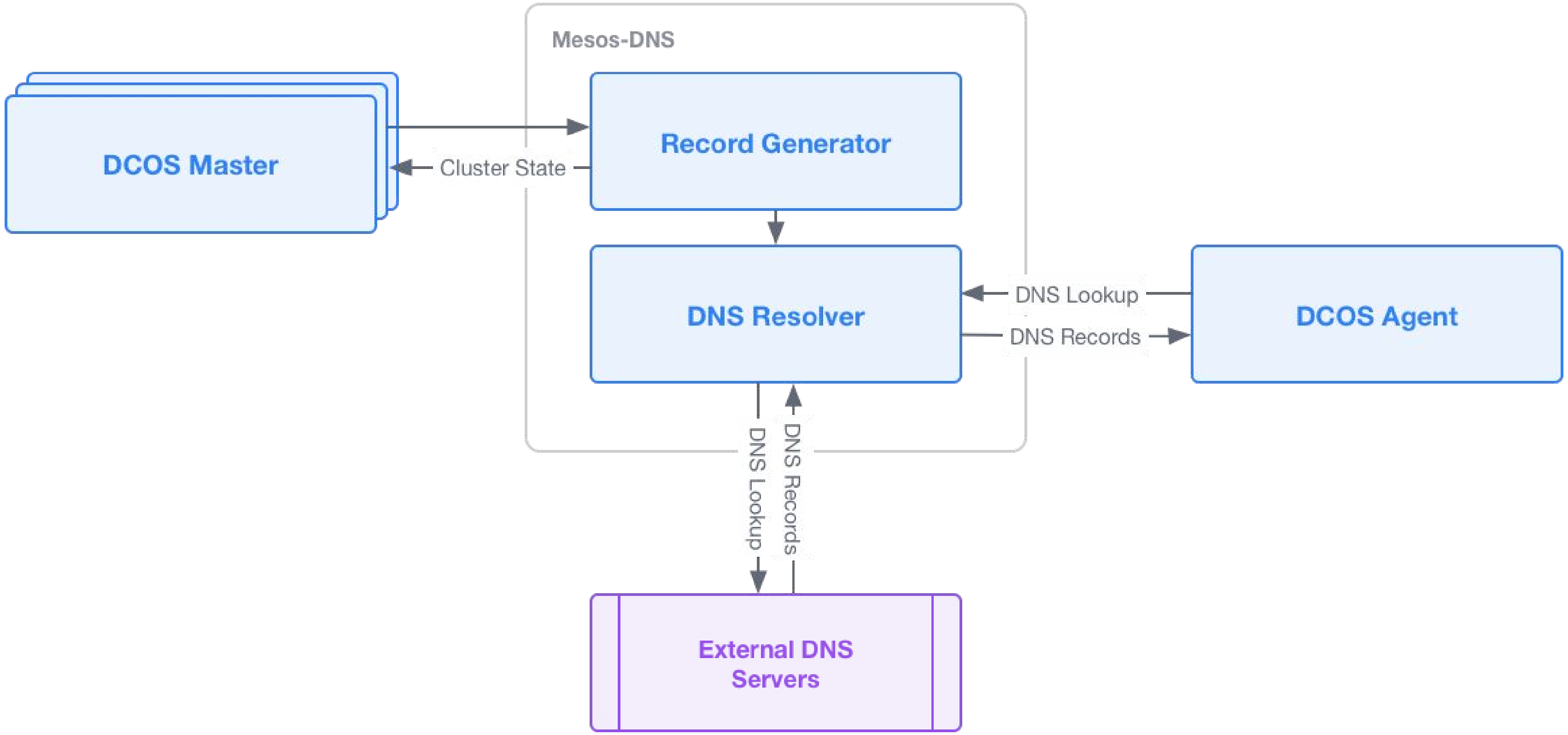
Two main service discovery mechanisms:

1. DNS based (Mesos-DNS)
2. HAProxy based (Marathon-lb)

Service Discovery

MESOS-DNS

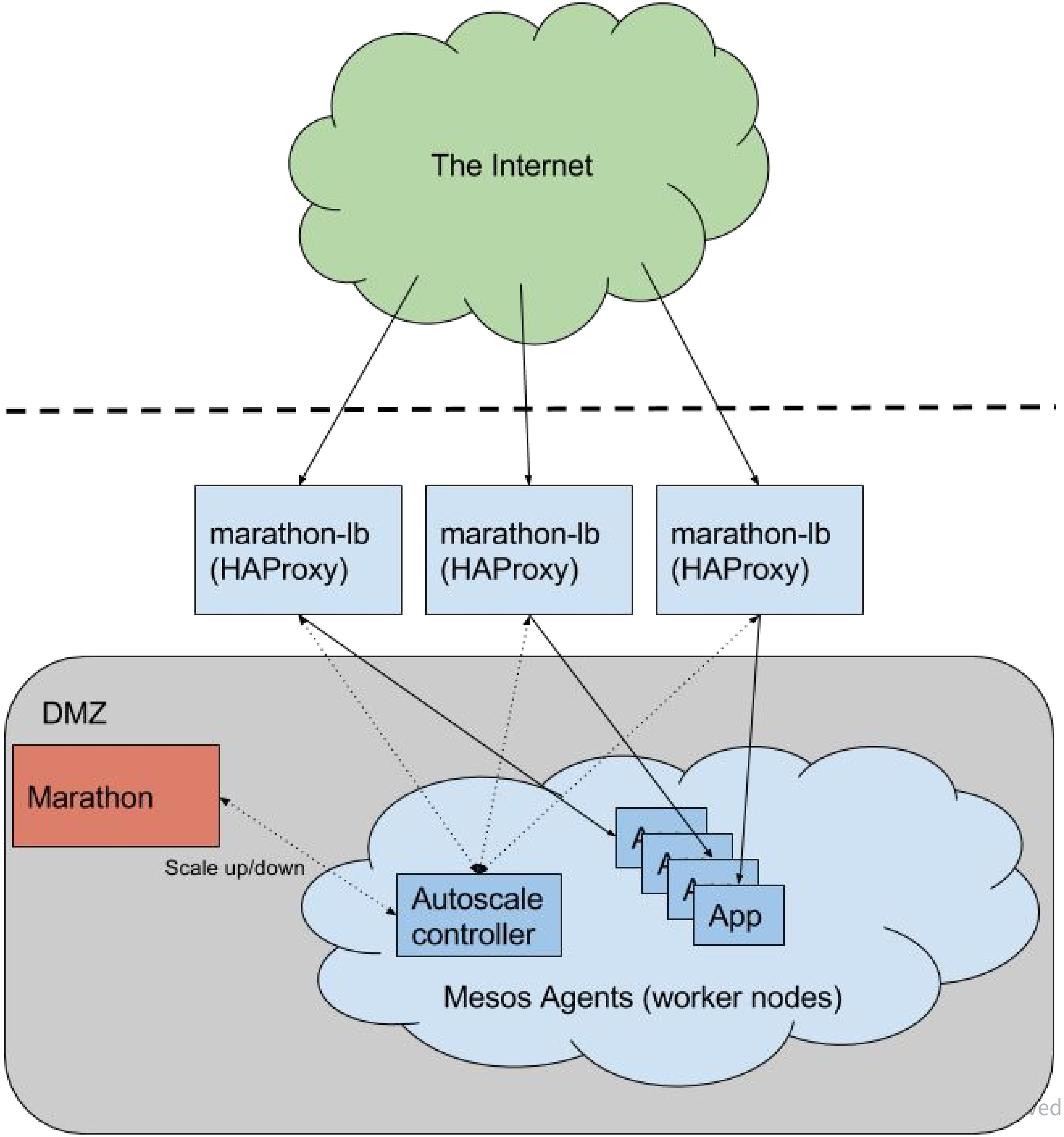
- Ingests cluster state periodically.
- Uses cluster state to generate DNS records for all running Mesos tasks.
- Services query DNS server to discover IP address and port of other services.
- Primarily used for internal service discovery.
- No extra configuration required!



Service Discovery

MARATHON-LB

- Ingests state of running Marathon applications.
- Regenerates HAProxy configuration.
- Supports virtual hosts!
- Can be used for both internal and external service discovery.
- Must add HAPROXY_GROUP and HAPROXY_0_VHOST variables to your marathon.json.

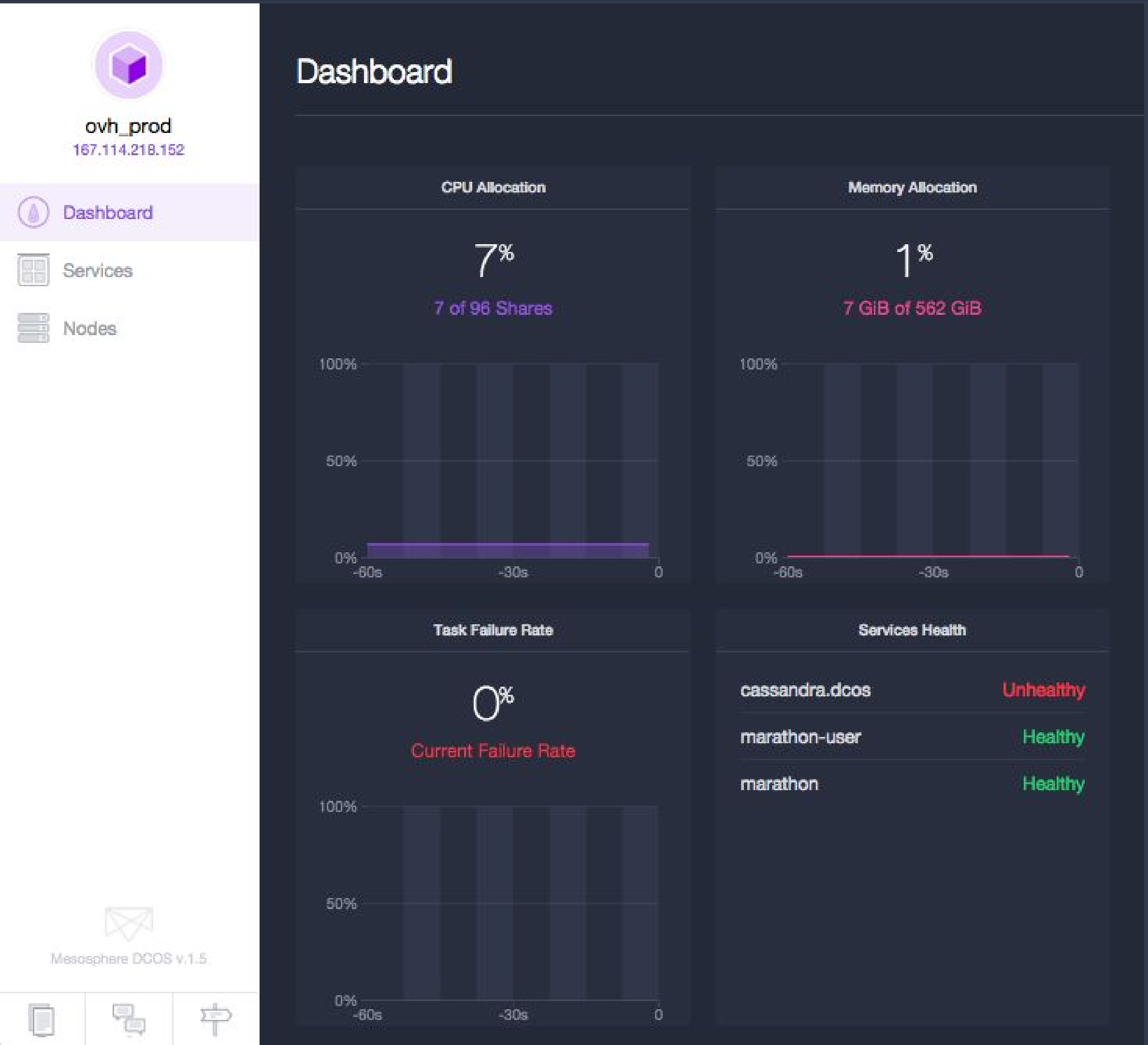


HOW TO DEPLOY A MESOS CLUSTER (THE HARD WAY)

- Using `chef/puppet/ansible` (or a reliable intern)
- Install ZooKeeper and Mesos
- Install your scheduler (Marathon)
- Deploy some long-running services.
- See https://open.mesosphere.com/getting-started/tools/ for more docs

Mesos as the Datacenter Kernel

HOW TO DEPLOY A MESOS CLUSTER (OUR WAY)



Dashboard

CPU Allocation

7%

7 of 96 Shares

Memory Allocation

1%

7 GiB of 562 GiB

Task Failure Rate

0%

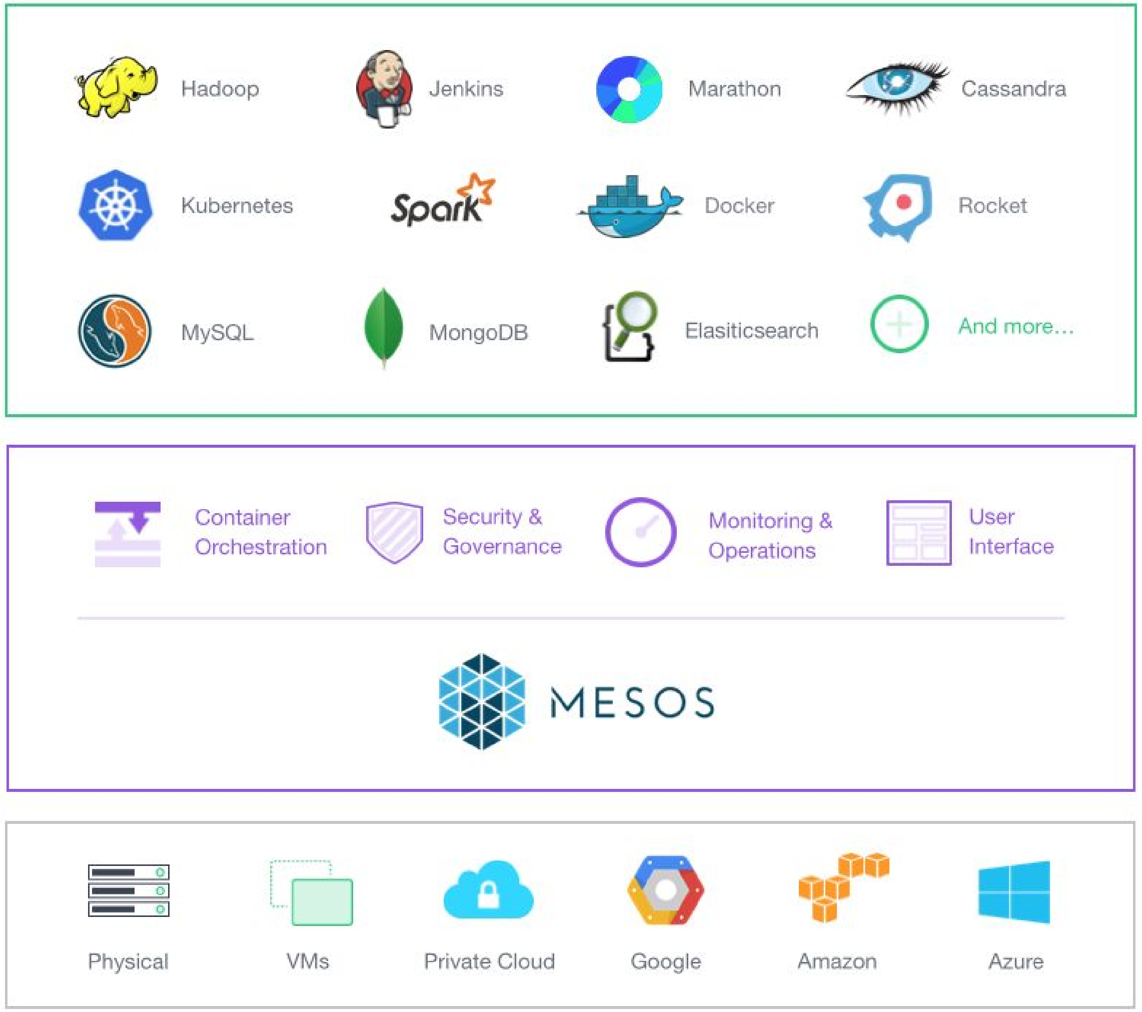
Current Failure Rate

Services Health

cassandra.dcos	Unhealthy
marathon-user	Healthy
marathon	Healthy

- Visit http://mesosphere.com
- Hit the ‘Get Started’ button

MESOS AS THE DATACENTER KERNEL



Hadoop

Jenkins

Marathon

Cassandra

Kubernetes

Spark

Docker

Rocket

MySQL

MongoDB

Elasiticsearch

And more…

Services & Containers

Container Orchestration

Security & Governance

Monitoring & Operations

User Interface

MESOS

Mesosphere DCOS

Physical

VMs

Private Cloud

Google

Amazon

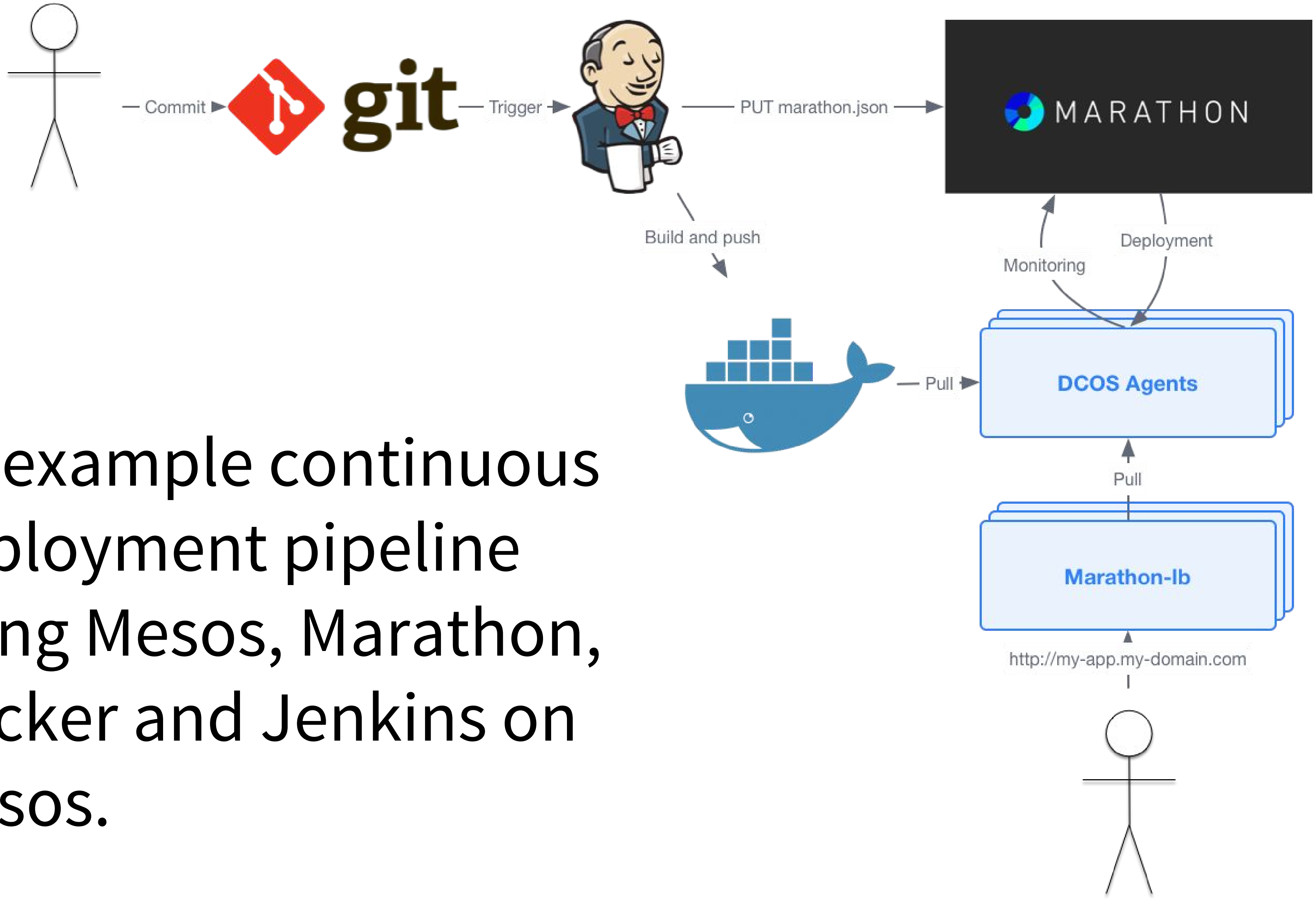
Azure

Existing Infrastructure

JENKINS: BUILD RESOURCE POOLING

Jenkins on Mesos allows you to share build resources between multiple Jenkins masters.

- PayPal does this with hundreds of Jenkins masters
- Between them, they use less than a hundred build slaves to service several thousand developers.
- Combining Jenkins with a PaaS like Marathon or Kubernetes allows you to practice easy continuous deployment.

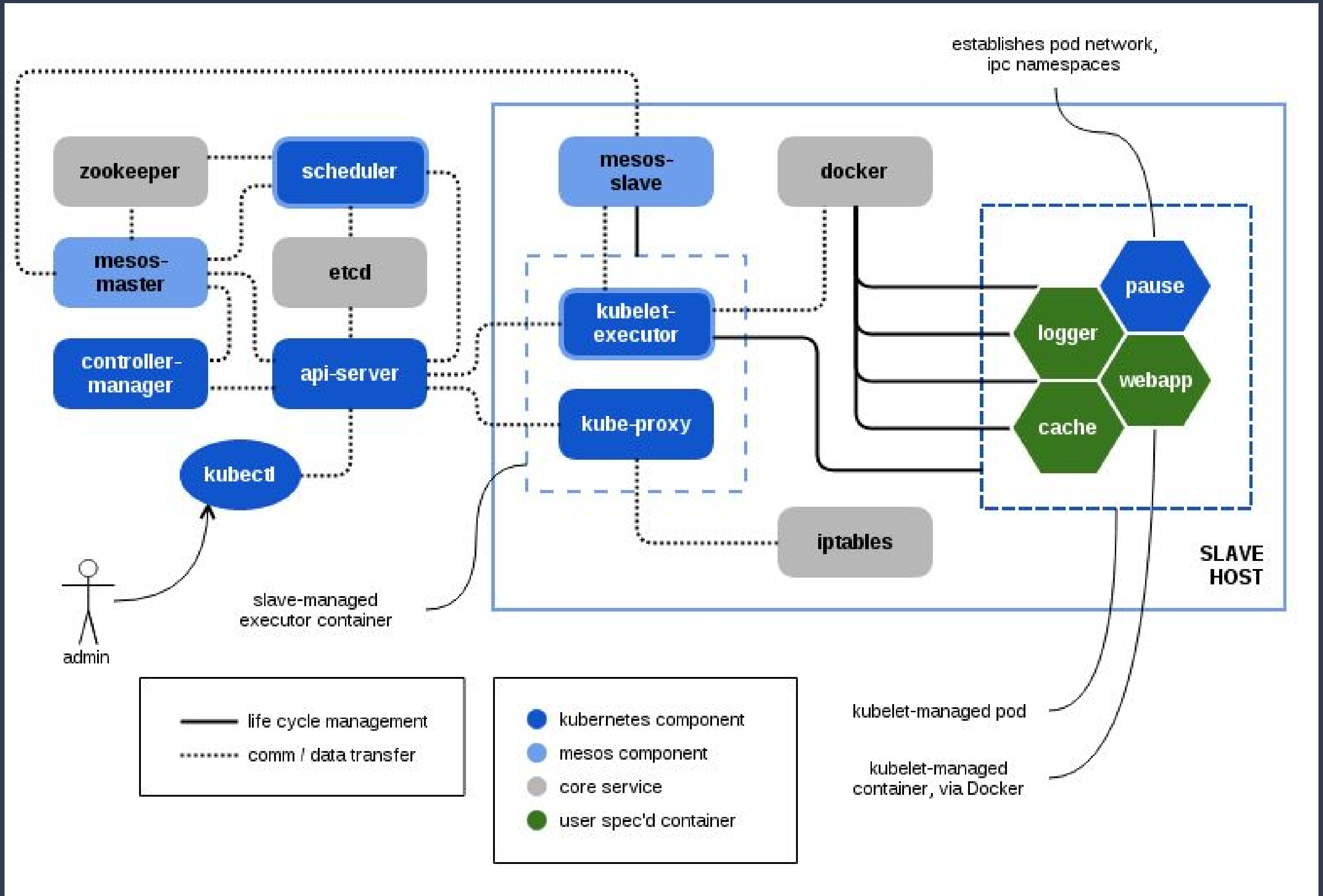


An example continuous deployment pipeline using Mesos, Marathon, Docker and Jenkins on Mesos.

CONTINUOUS DELIVERY DEMO

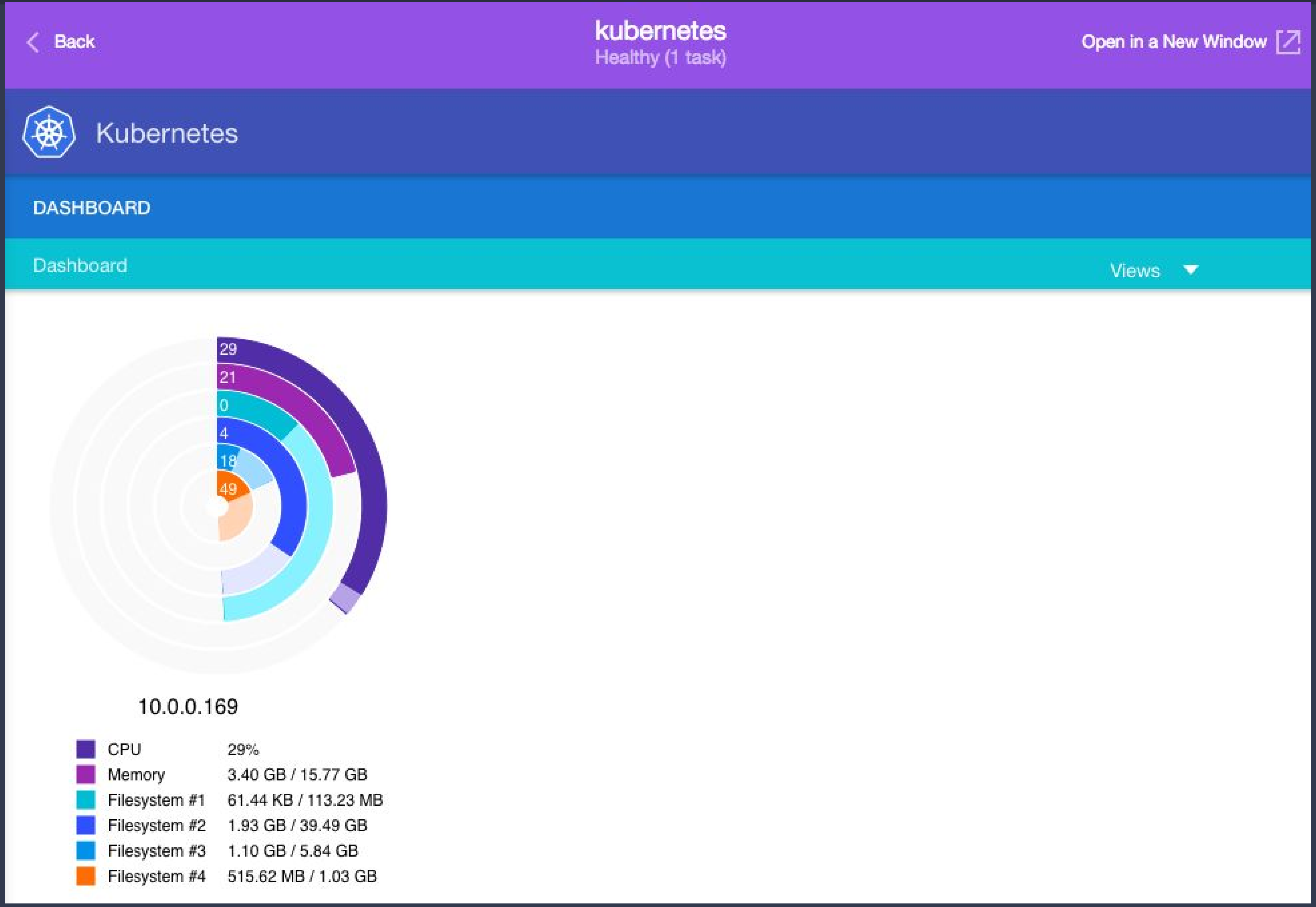
Mesos as the Datacenter Kernel

KUBERNETES ON MESOS



Mesos as the Datacenter Kernel

KUBERNETES ON MESOS



10.0.0.169

■	CPU	29%
■	Memory	3.40 GB / 15.77 GB
■	Filesystem #1	61.44 KB / 113.23 MB
■	Filesystem #2	1.93 GB / 39.49 GB
■	Filesystem #3	1.10 GB / 5.84 GB
■	Filesystem #4	515.62 MB / 1.03 GB

BIG DATA ON MESOS

Mesos was built for and is great for running big data workloads:

- Chronos (time scheduled jobs)
- Spark
- Cassandra
- Kafka
- Hadoop/YARN (via Myriad)

QUESTIONS?
THANK YOU!

Come and talk to us!

- Email us at philip@mesosphere.io, sunil@mesosphere.io
- Slides will be up at http://mesosphere.github.io/presentations