mesosphere

Software for the hyperscale datacenter

Florian Leibert, CEO & Founder
flo@mesosphere.io

Imagine if…

All your servers in your datacenter and cloud

were pooled together

So they behave like one big computer

as easy as

play!
CHRONOS
Spark
JBoss
by Red Hat

…and building new datacenter apps is as easy as building an app for one machine

Fantasy?

mesosphere

Mesos Users Today

vimeo

OpenTable™

NETFLIX

Atigeo™

airbnb™

sharethrough

MediaCrossing™
BRIDGING THE MARKET™

salesforce

CommonwealthBank

UCSF

DueDil

HubSpot

xogito
...radical thinking...

PayPal™

shopify

CONVIVA®

Sigmoid Analytics

BEST BUY®

CLOUD PHYSICS

device scape™

CATEGORIZE

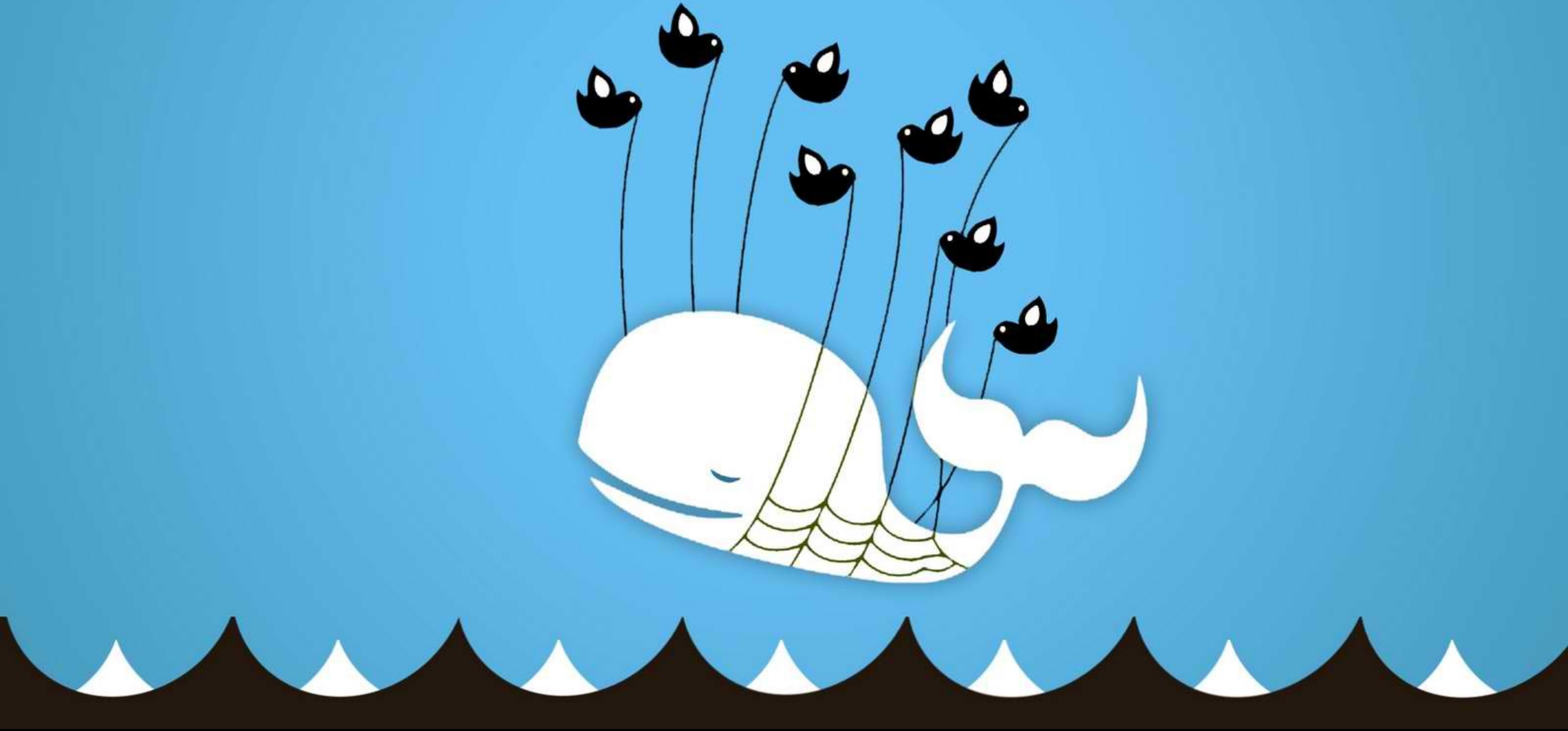
pb pinkbike

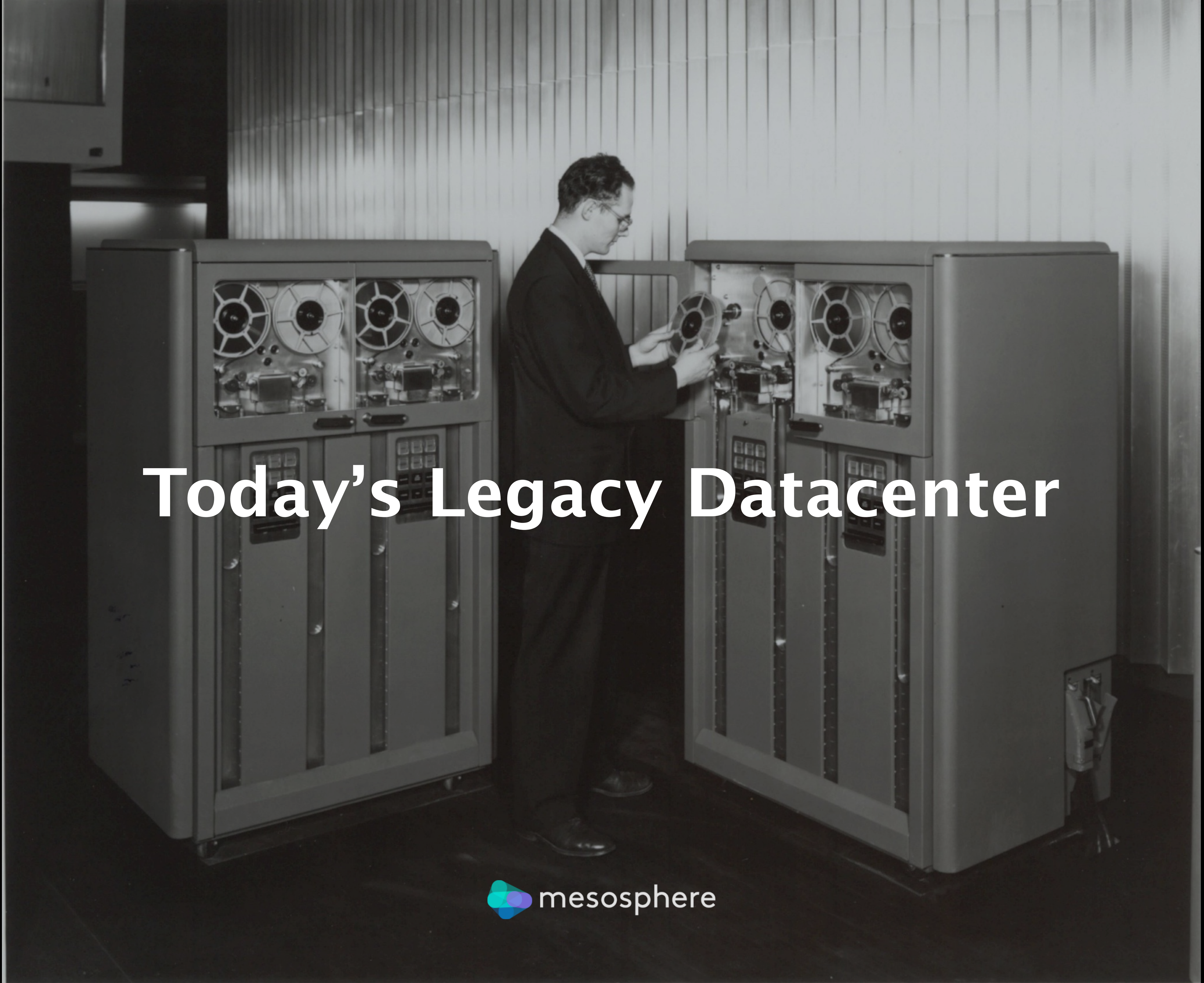
medidata

ignidata
igniting business with data

iQIYI 爱奇艺

mesosphere

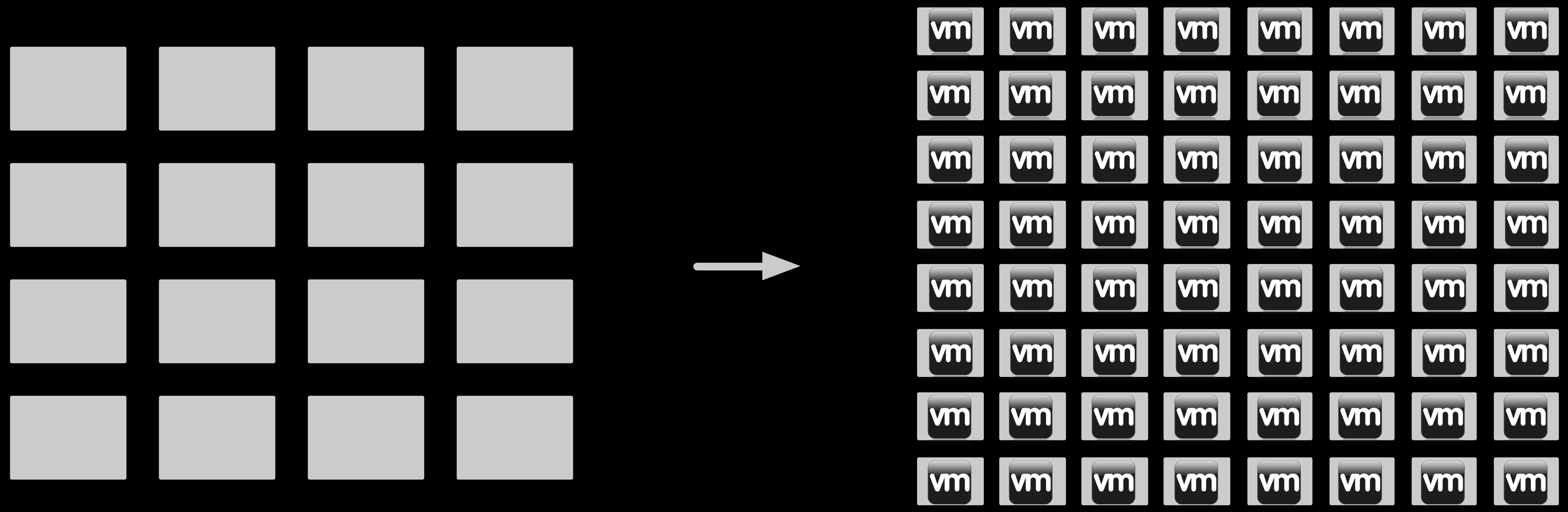




Today’s Legacy Datacenter

mesosphere

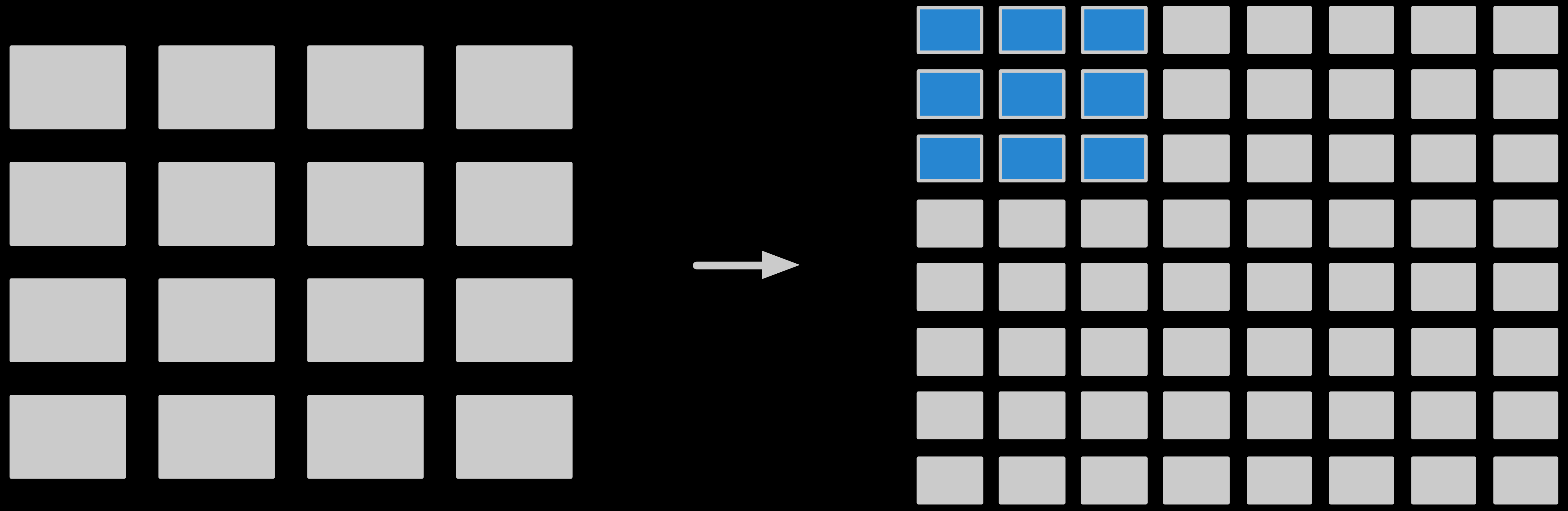
Today's Legacy Datacenter



Provision VMs in the cloud or on physical servers

mesosphere

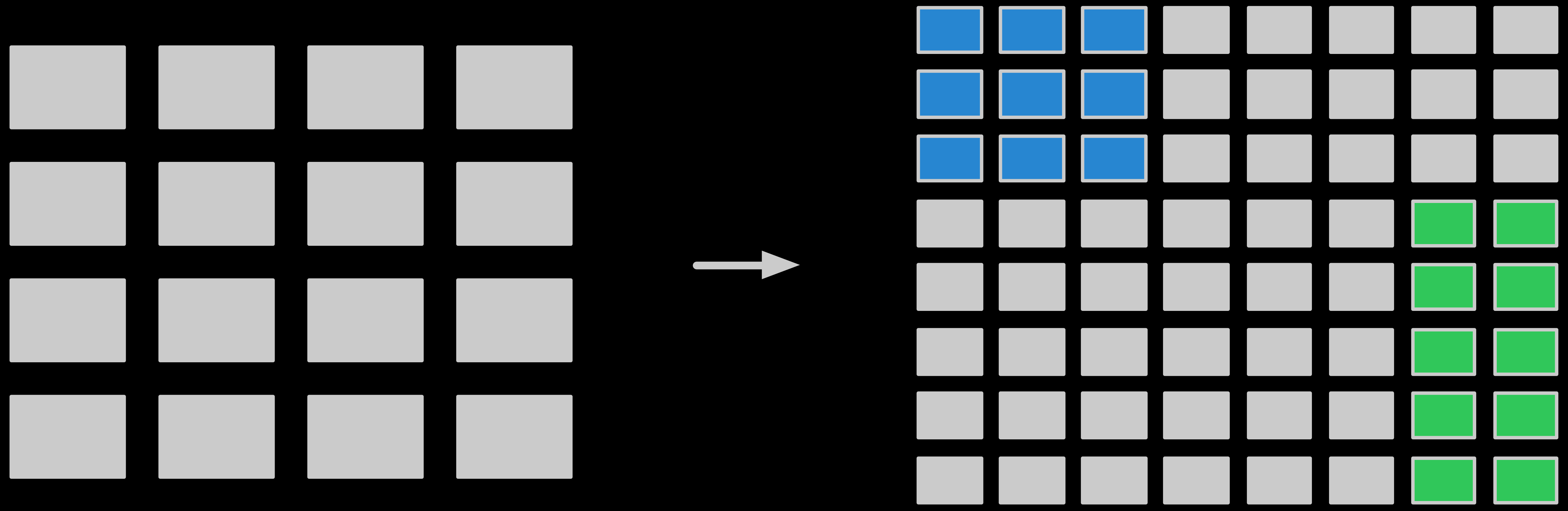
Installing an Application with Static Partitioning



Install Hadoop on a static set of machines

mesosphere

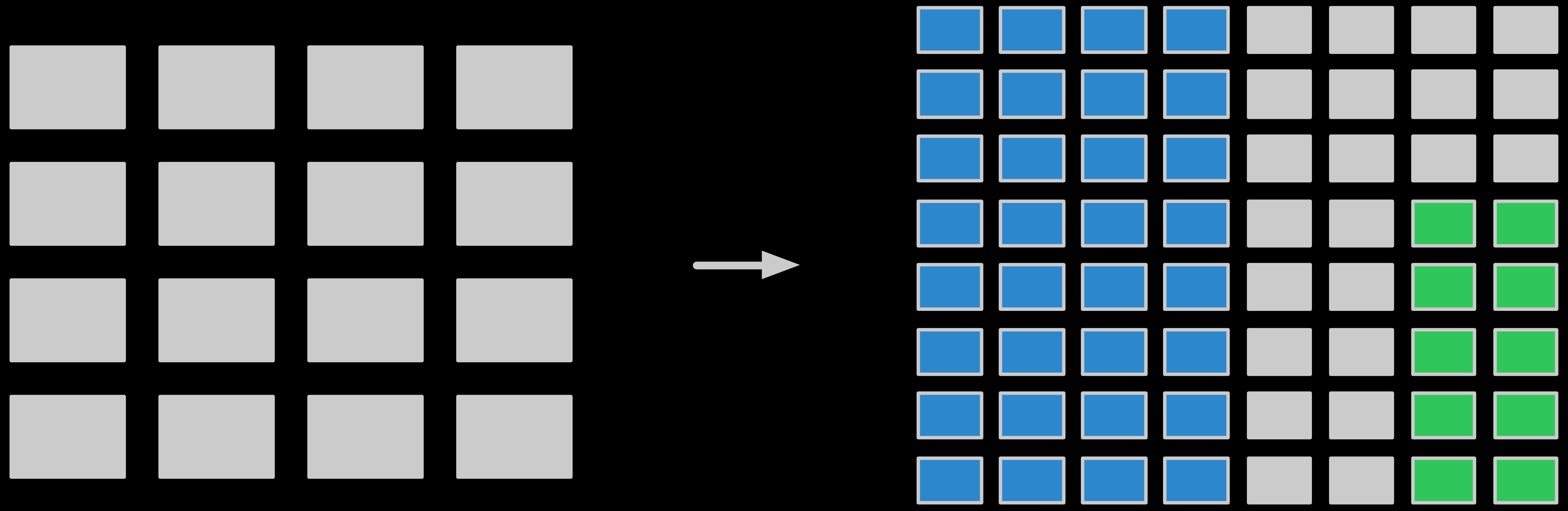
Installing an Application with Static Partitioning



Install Web Server on a static set of machines

mesosphere

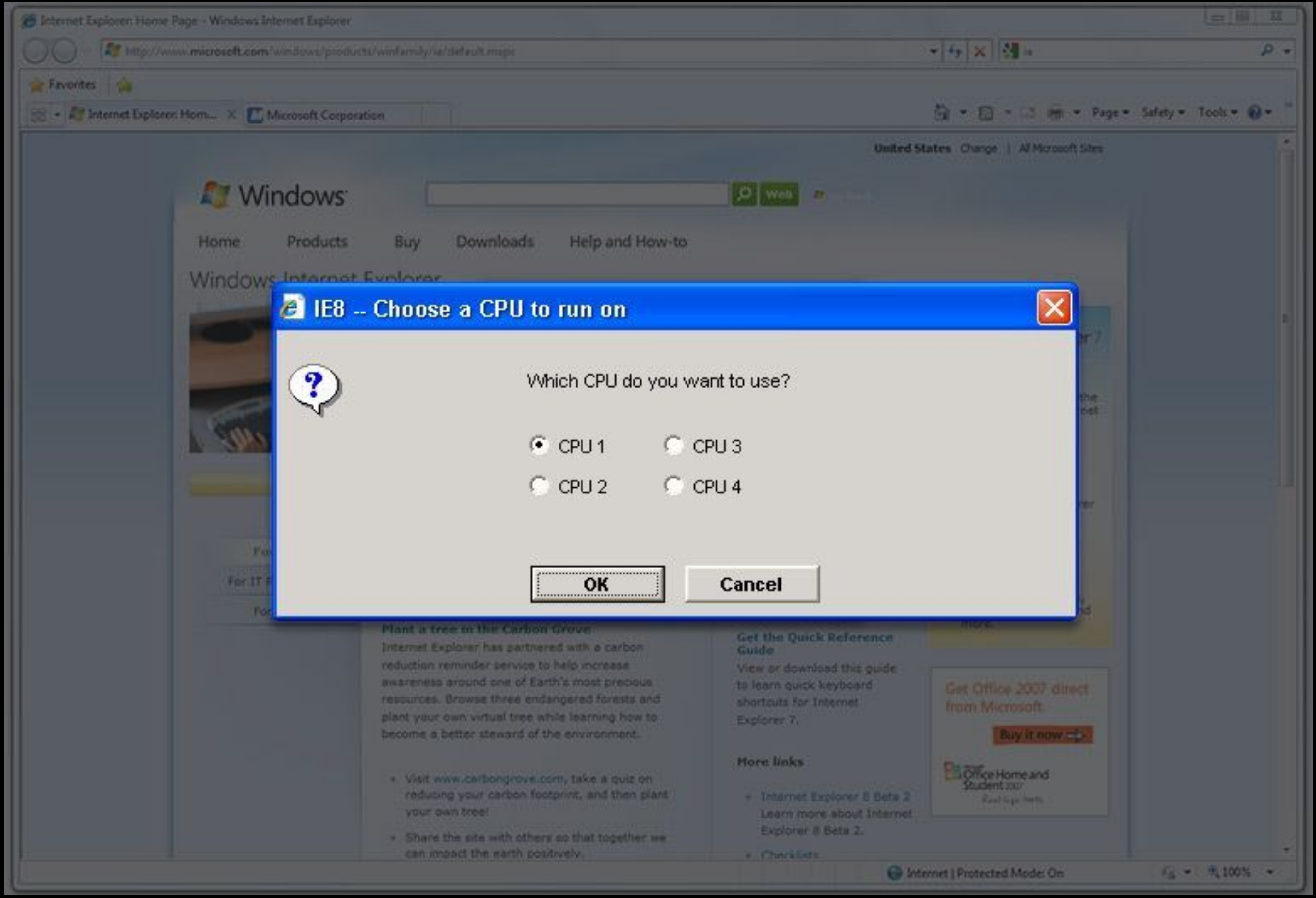
Resizing an Application with Static Partitioning



Scale up Hadoop manually

mesosphere

What if your Laptop was operated like your Data



Which CPU do you want to use?

[✓ CPU 1] CPU 3
CPU 2 CPU 4

OK Cancel

Issues with Statically Partitioned Data Centers

Complex
Machine sprawl, manual resize/scale

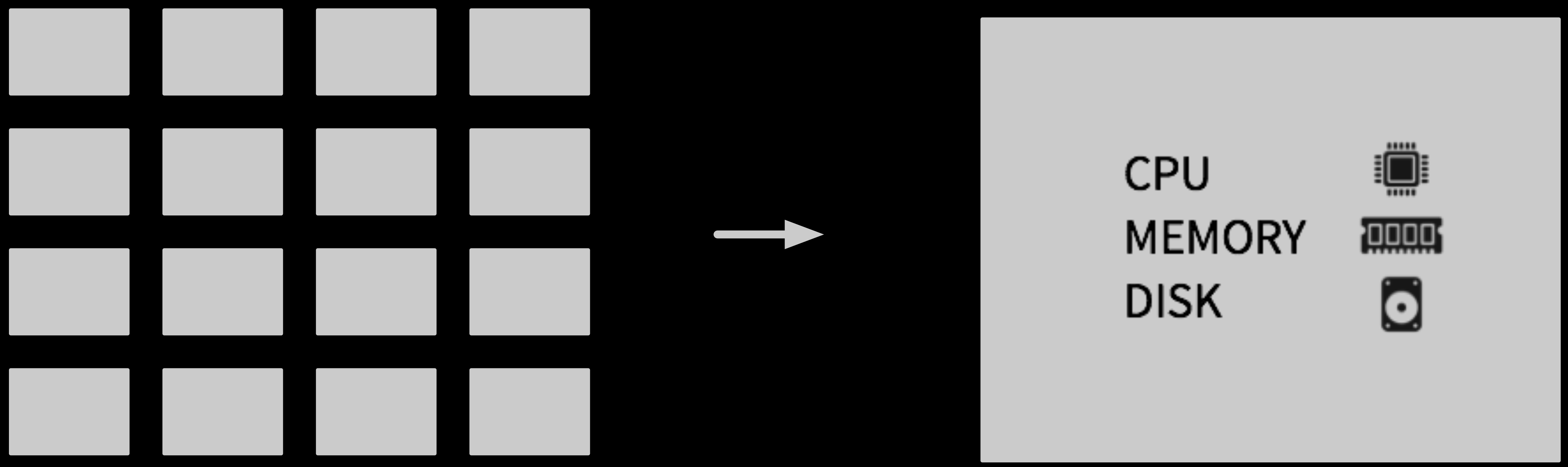
Limited
No software failure handling, “black box”

Inefficient
Static partitioning, overhead

Not Developer-Friendly
Long time to roll out software, development starts at the machine level

mesosphere

Aggregation

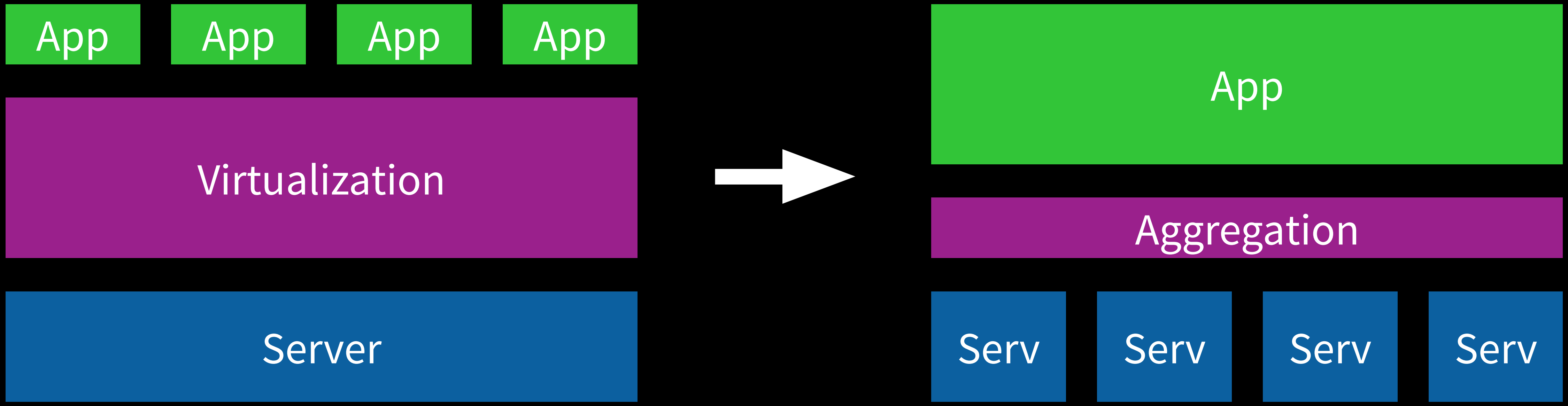


Mesosphere aggregates resources, makes a data center look like one big computer

Mesosphere runs on top of a VM or on bare metal

mesosphere

Applications in the Cloud Era



Client-Server Era:
Small apps, big servers

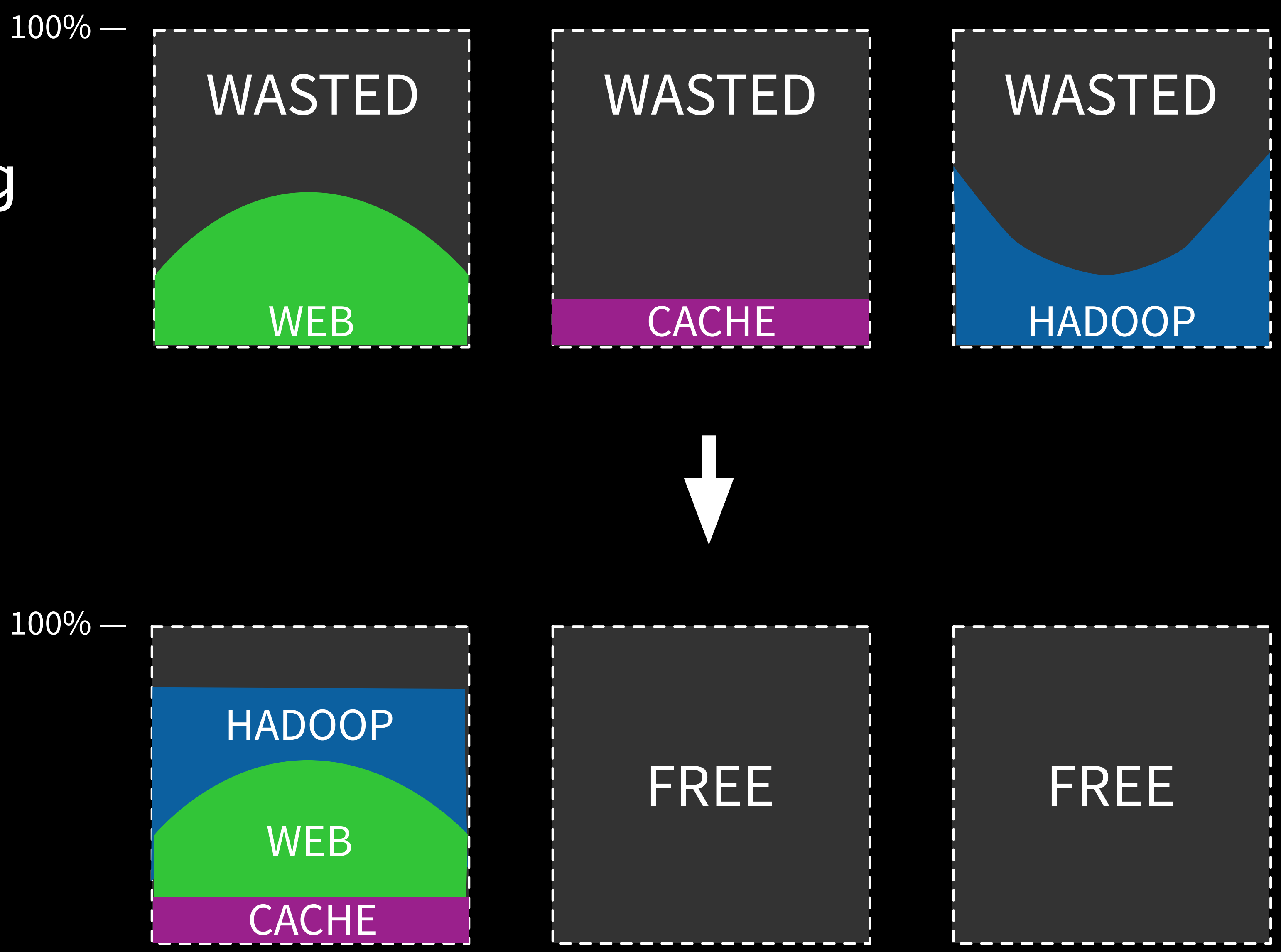
Cloud Era:
Big apps, small servers

mesosphere

From Static Partitioning to Elastic Sharing

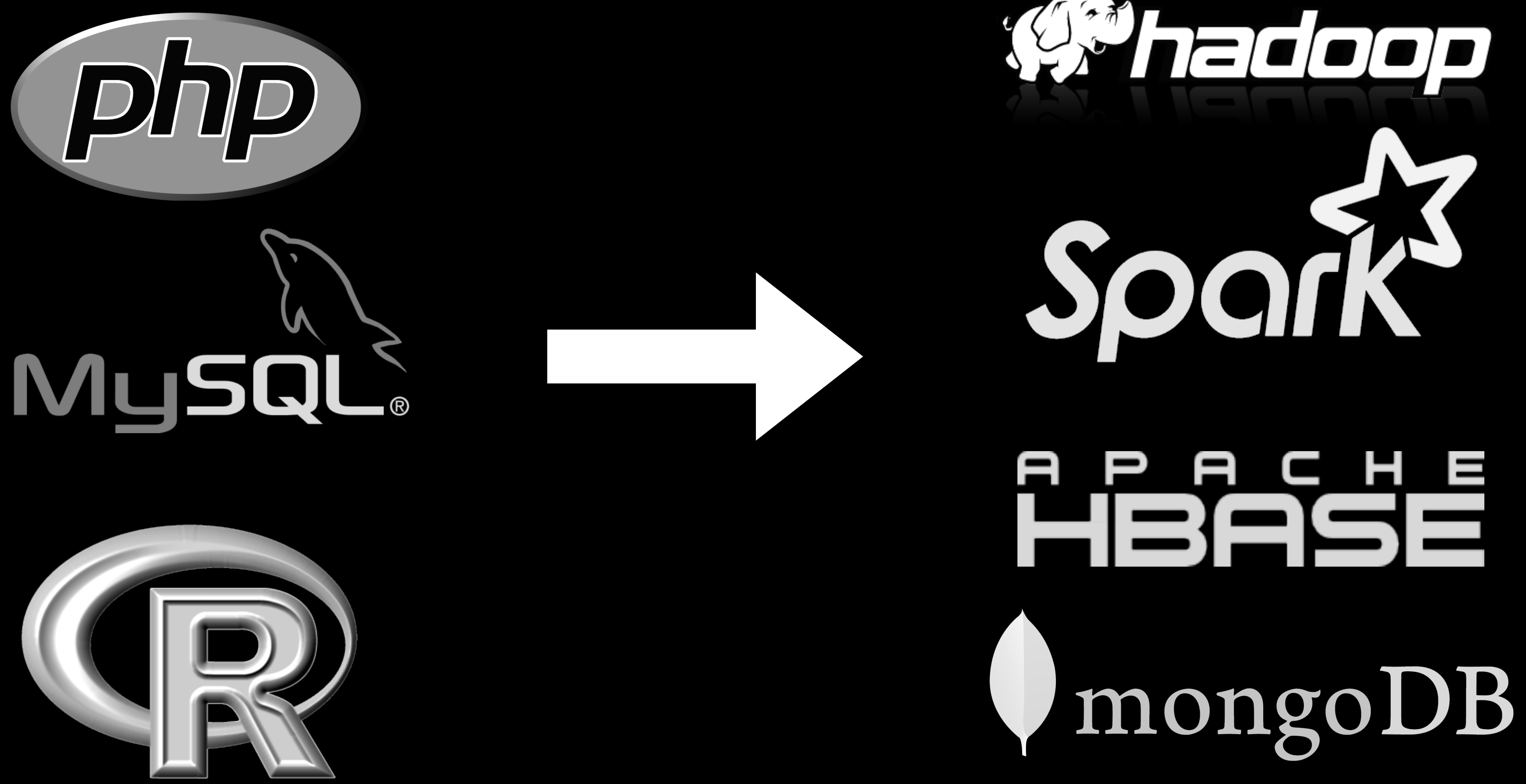
Static Partitioning

Elastic Sharing



mesosphere

Applications are Changing



Deployments

CLOUD PHYSICS

UCSF

NETFLIX

airbnb

OpenTable

BEST BUY

vimeo

CONVIVA

xogito [kä-gi-tō]

medidata

PayPal

CommonwealthBank

salesforce

HubSpot

GROUPON

mesosphere

Mesos Facts

Scales to 10,000s of nodes

Top-level Apache project

Twitter and Airbnb are major users and contributors

APIs for C++, Python, JVM, Go

Pluggable CPU, memory, IO isolation

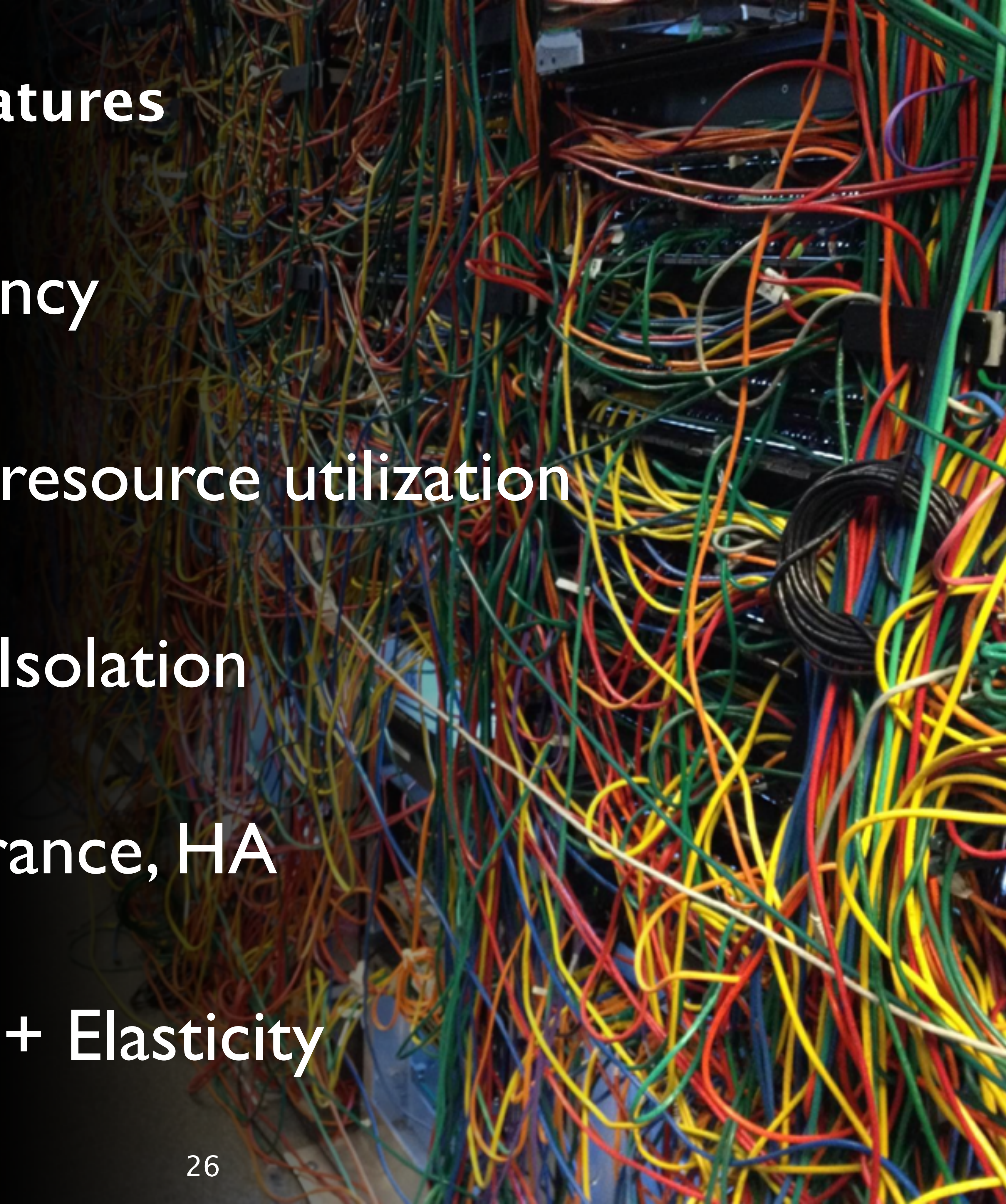
Packages and commercial support through Mesosphere

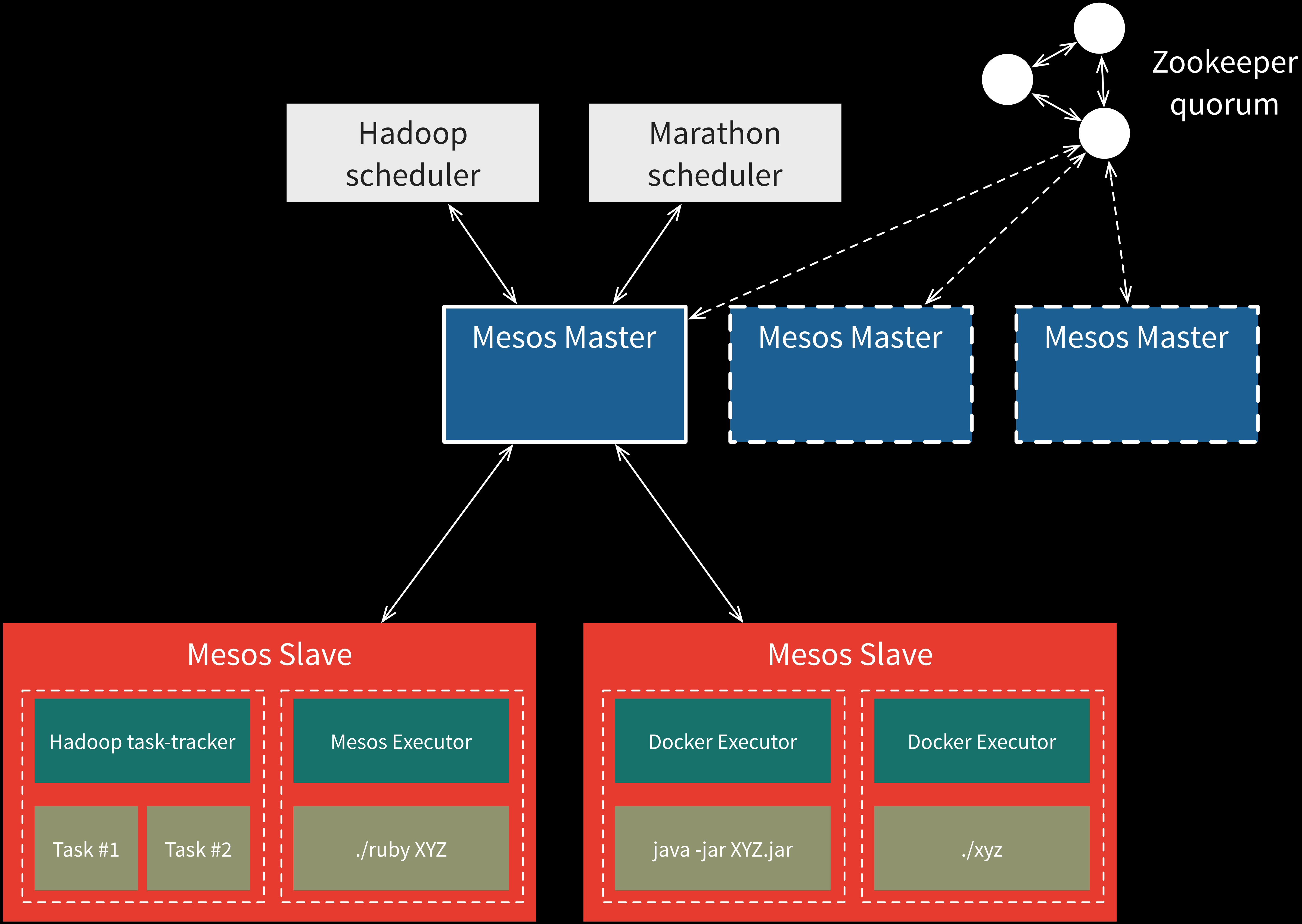
Highly available, scalable, elastic

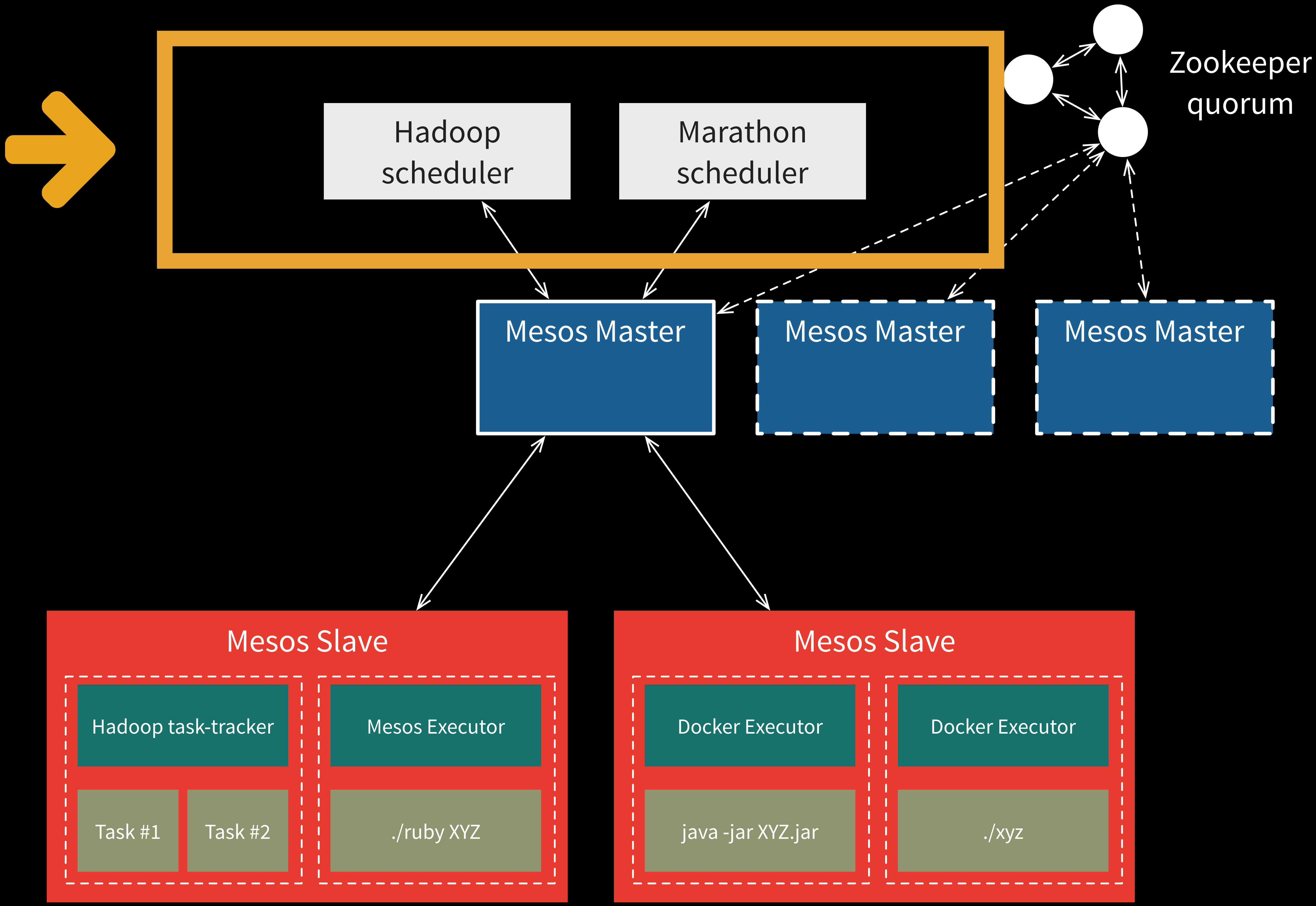
mesosphere

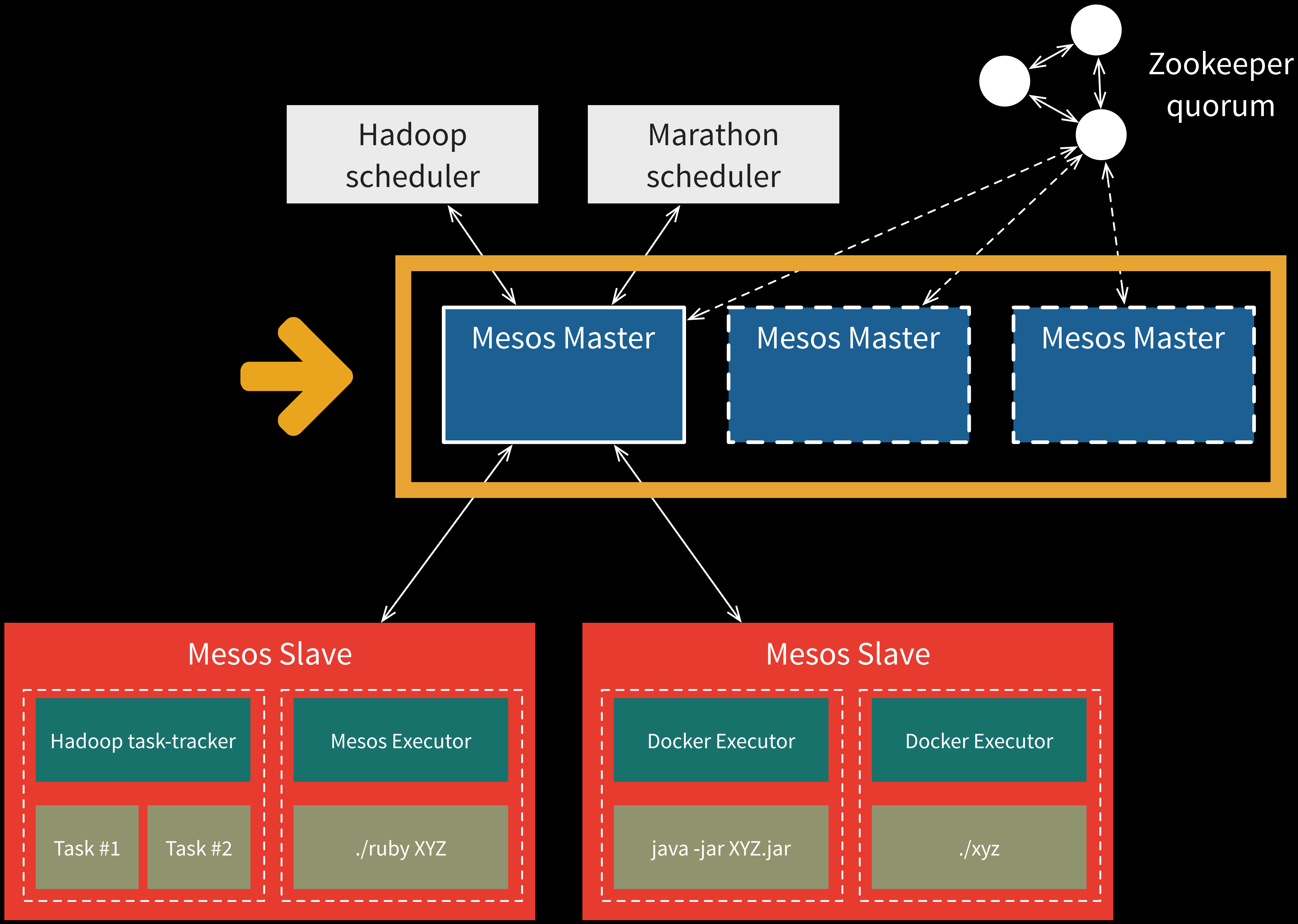
Apache Mesos Features

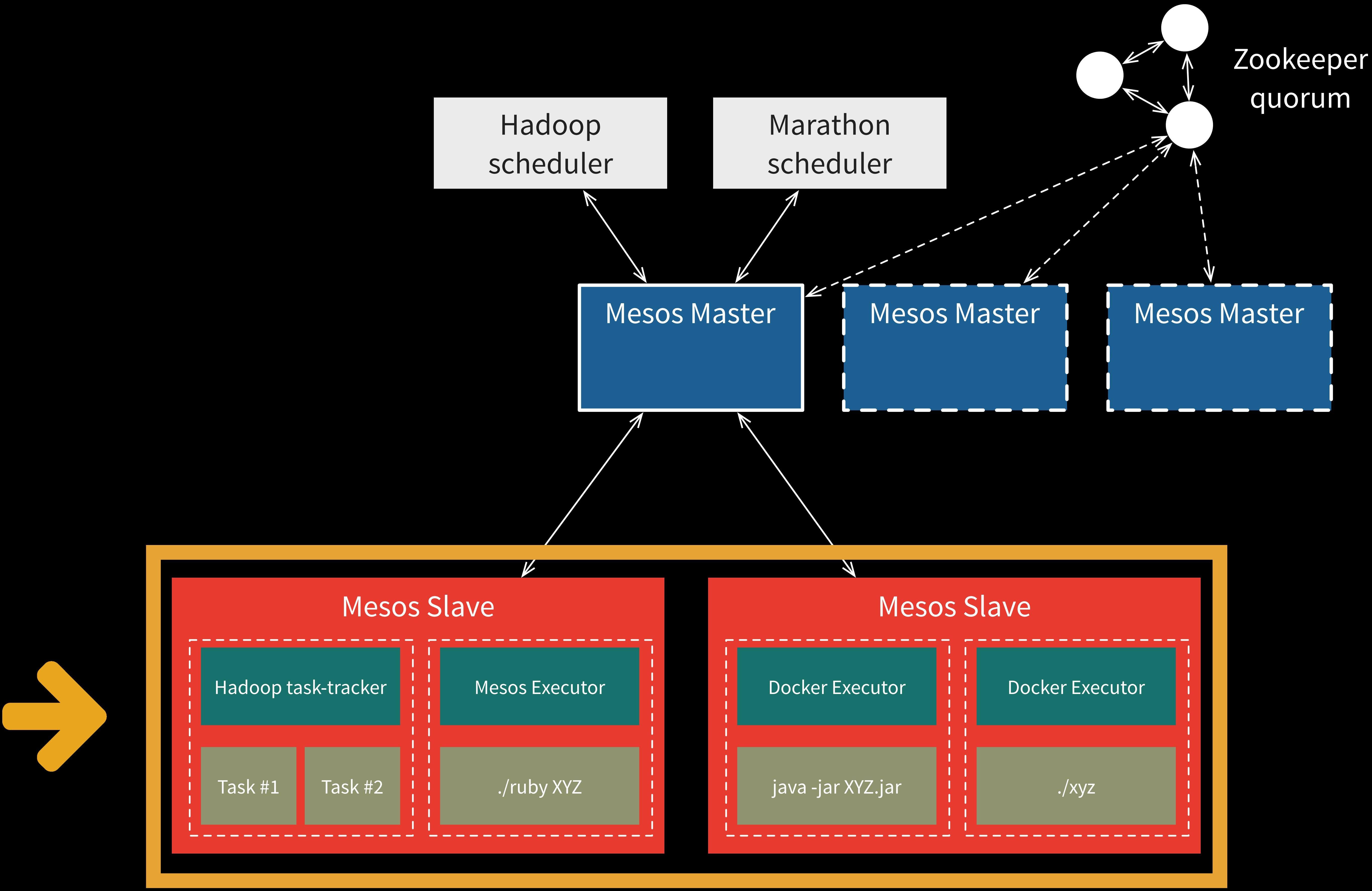
- Multi-tenancy
- Improved resource utilization
- Resource Isolation
- Fault-tolerance, HA
- Scalability + Elasticity



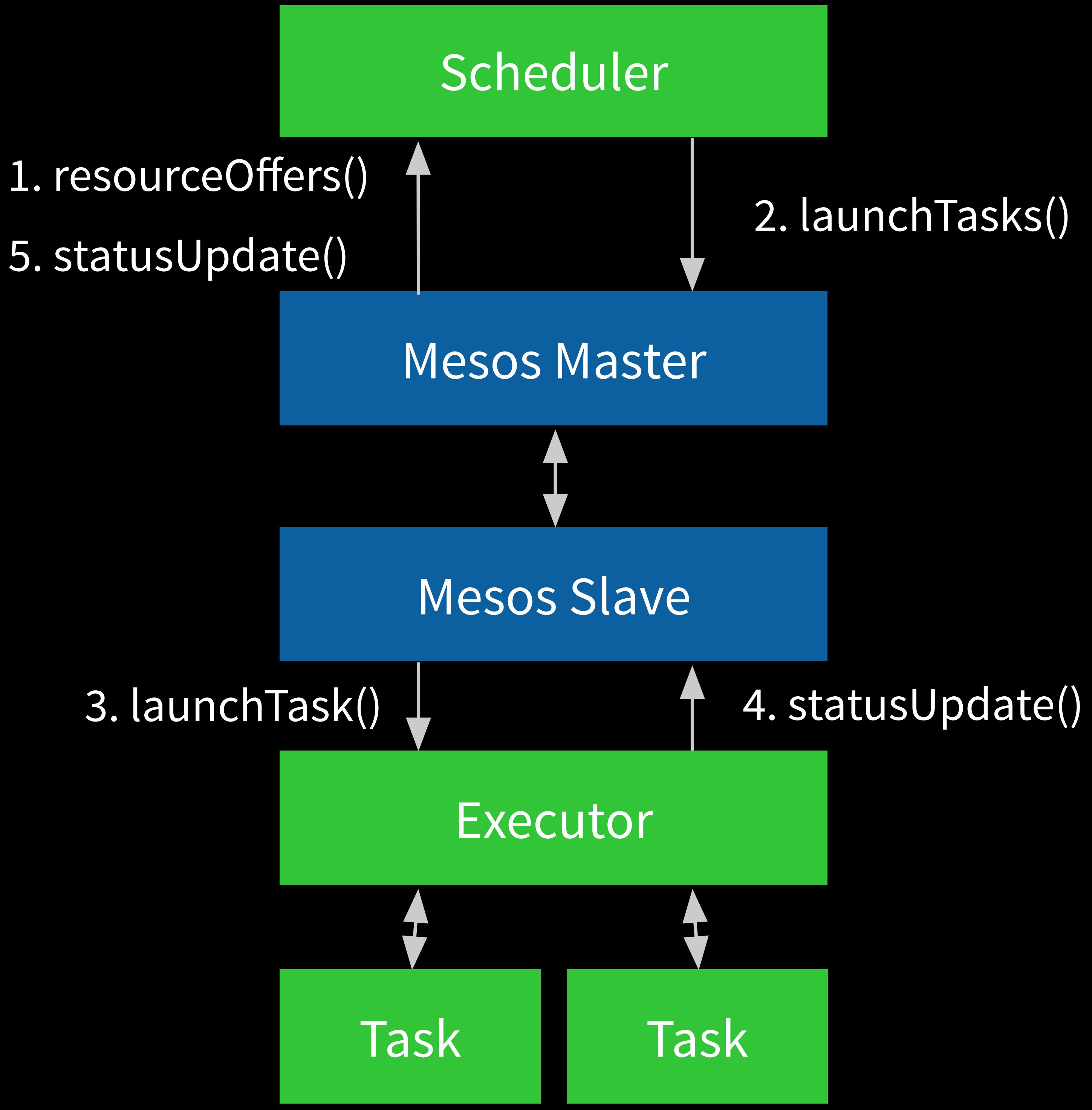

Zookeeper quorum
Hadoop scheduler
Marathon scheduler
Mesos Master
Mesos Master
Mesos Master
Mesos Slave
Hadoop task-tracker
Mesos Executor
Task #1
Task #2
./ruby XYZ
Mesos Slave
Docker Executor
Docker Executor
java -jar XYZ.jar
./xyz


Hadoop scheduler
Marathon scheduler
Zookeeper quorum
Mesos Master
Mesos Master
Mesos Master
Mesos Slave
Hadoop task-tracker
Mesos Executor
Task #1
Task #2
./ruby XYZ
Mesos Slave
Docker Executor
Docker Executor
java -jar XYZ.jar
./xyz


Zookeeper quorum
Hadoop scheduler
Marathon scheduler
Mesos Master
Mesos Master
Mesos Master
Mesos Slave
Hadoop task-tracker
Mesos Executor
Task #1
Task #2
./ruby XYZ
Mesos Slave
Docker Executor
Docker Executor
java -jar XYZ.jar
./xyz

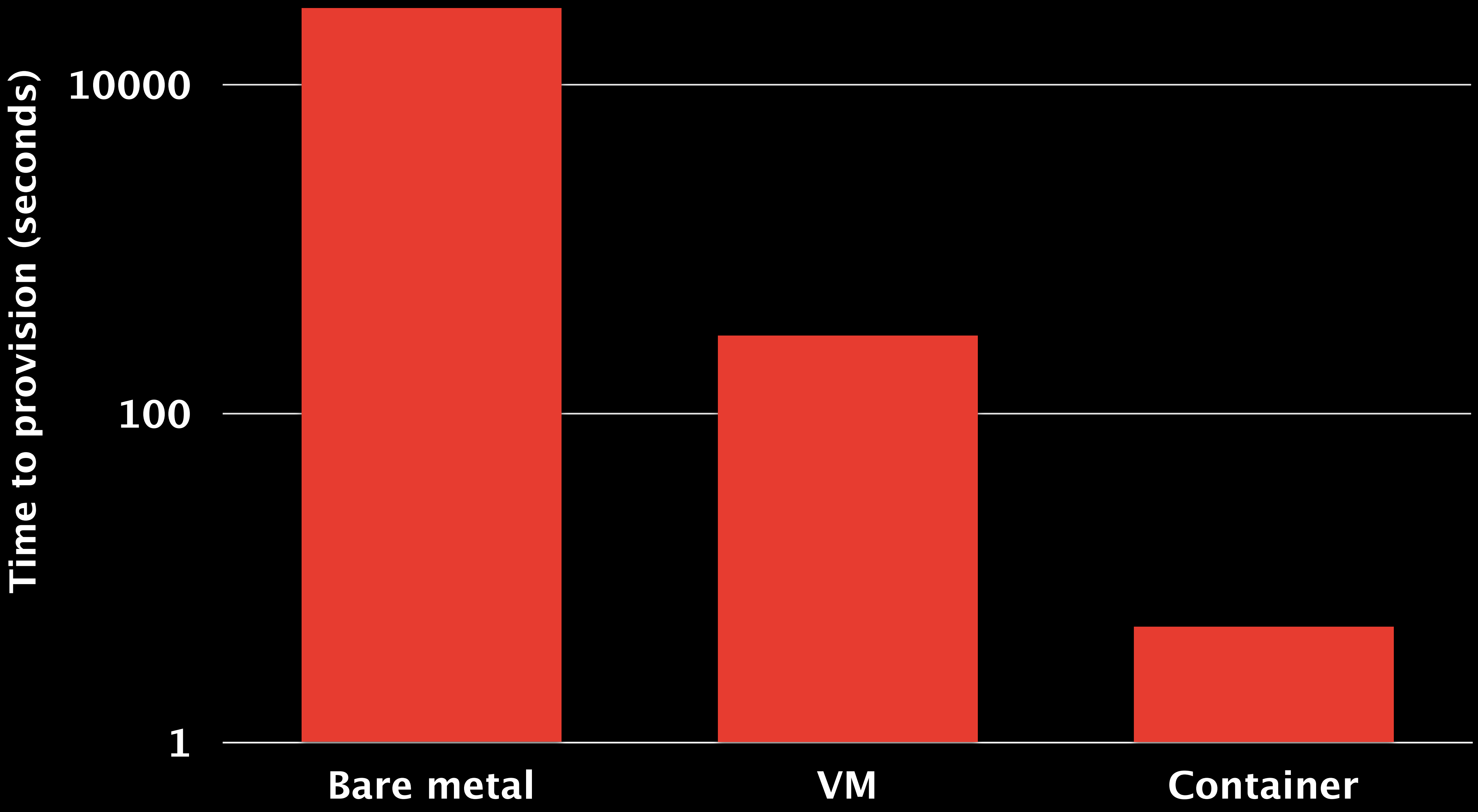

Zookeeper quorum
Hadoop scheduler
Marathon scheduler
Mesos Master
Mesos Master
Mesos Master
Mesos Slave
Hadoop task-tracker
Mesos Executor
Task #1
Task #2
./ruby XYZ
Mesos Slave
Docker Executor
Docker Executor
java -jar XYZ.jar
./xyz

Resource Offers and Launching a Task



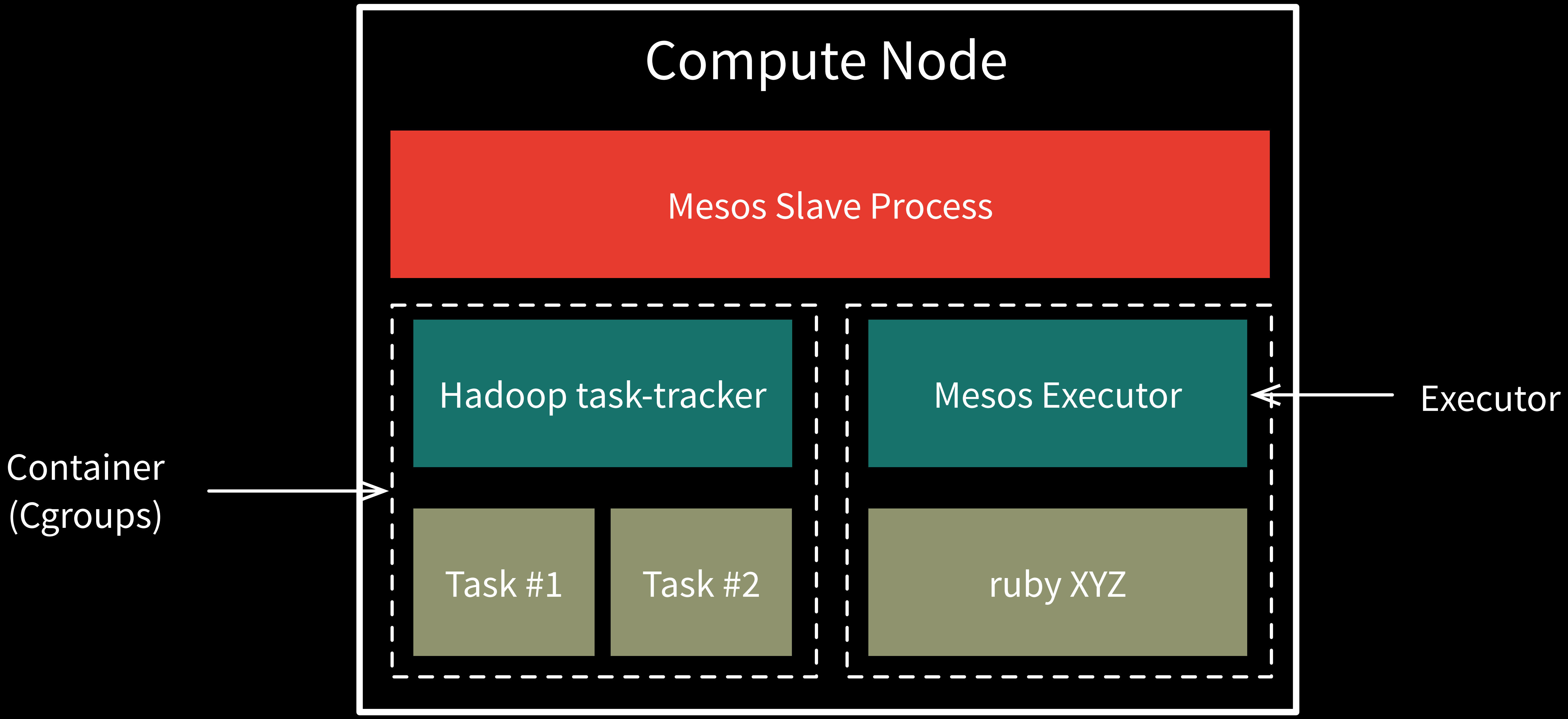
mesosphere

Response times and overhead are significantly

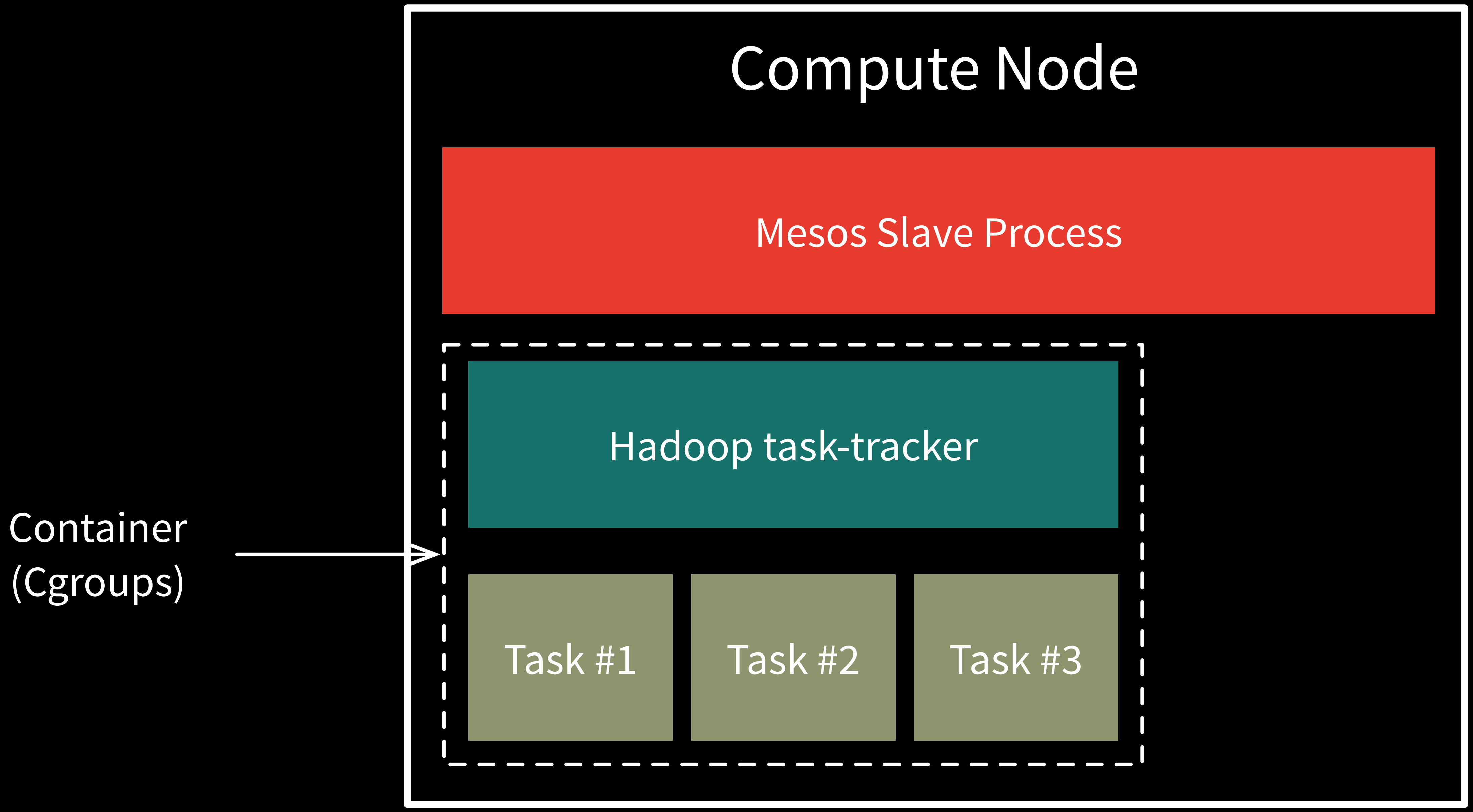


Inspired by Tomas Barton’s Mesos talk at InstallFest in Prague

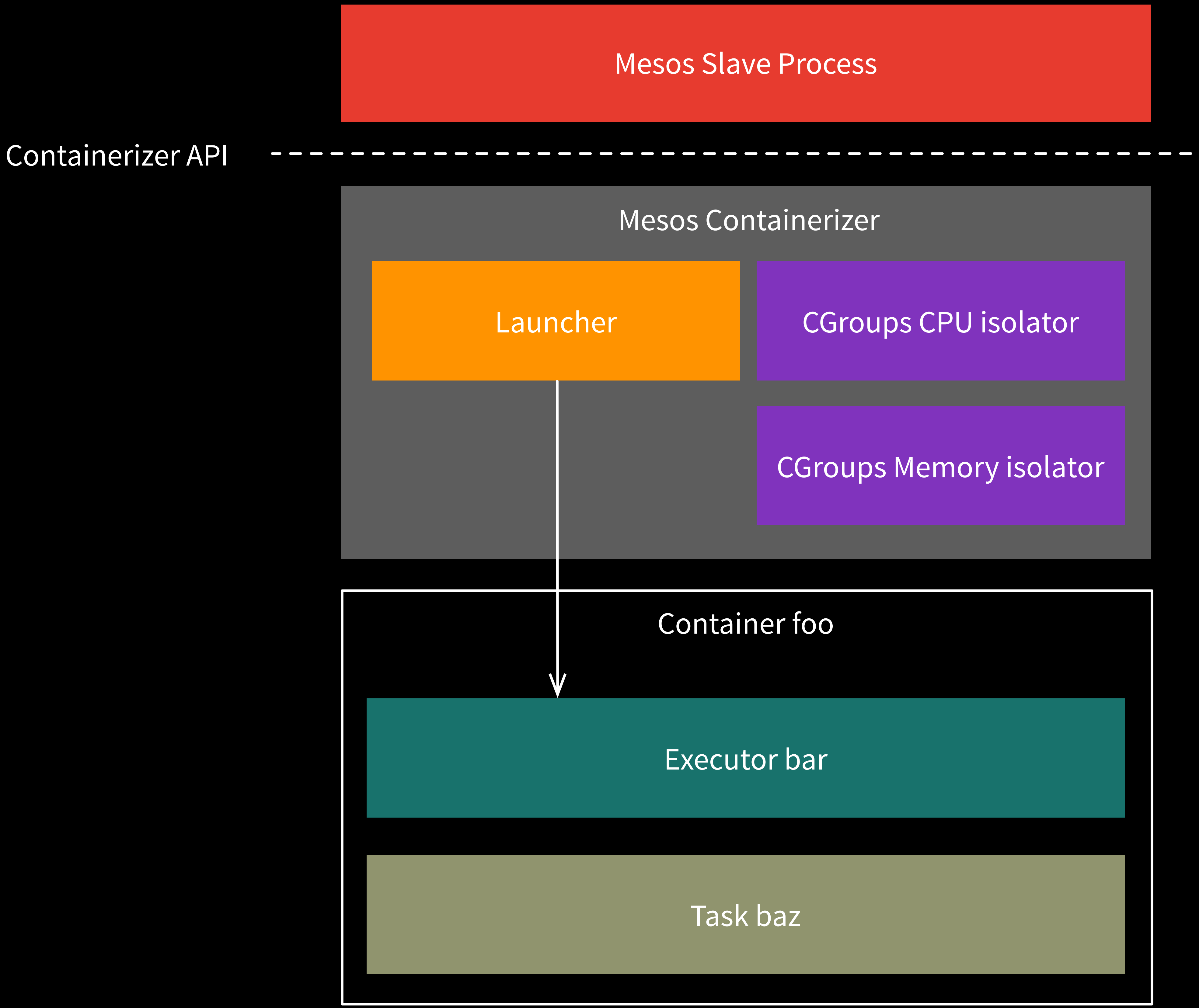
Mesos provides fine-grained resource isolation



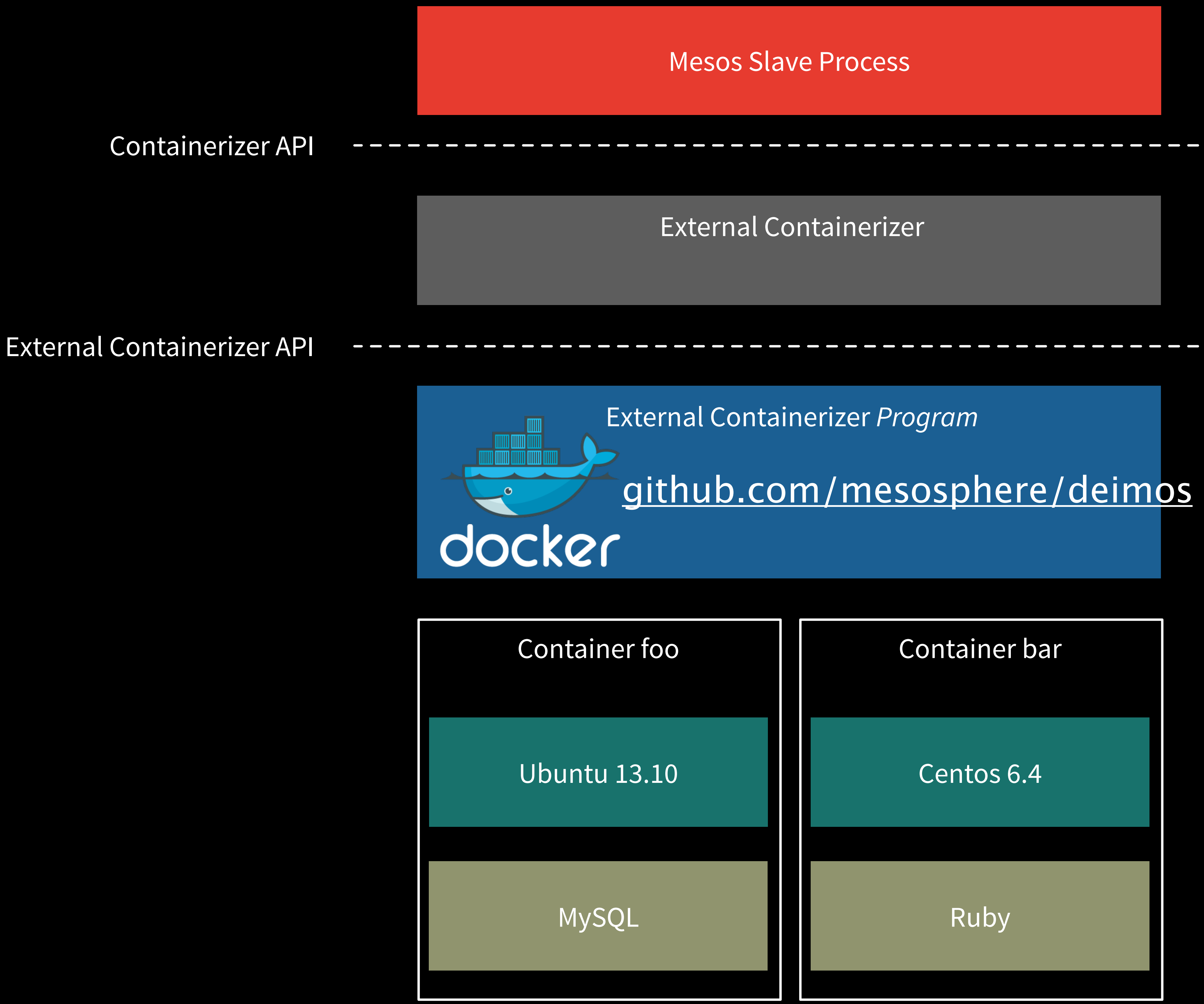
Mesos provides fine-grained resource isolation



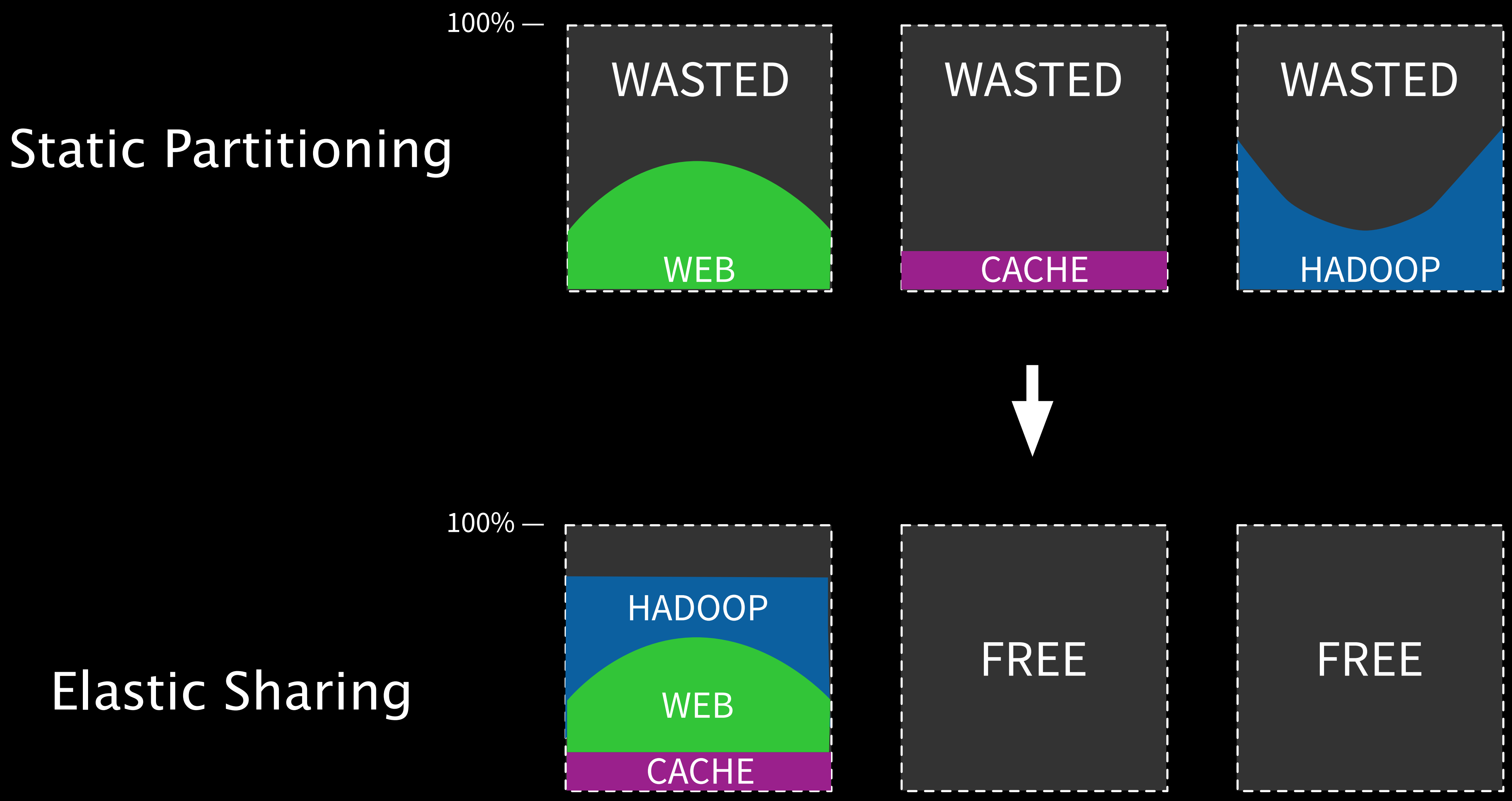
Mesos provides componentized resource



Mesos provides pluggable resource isolation

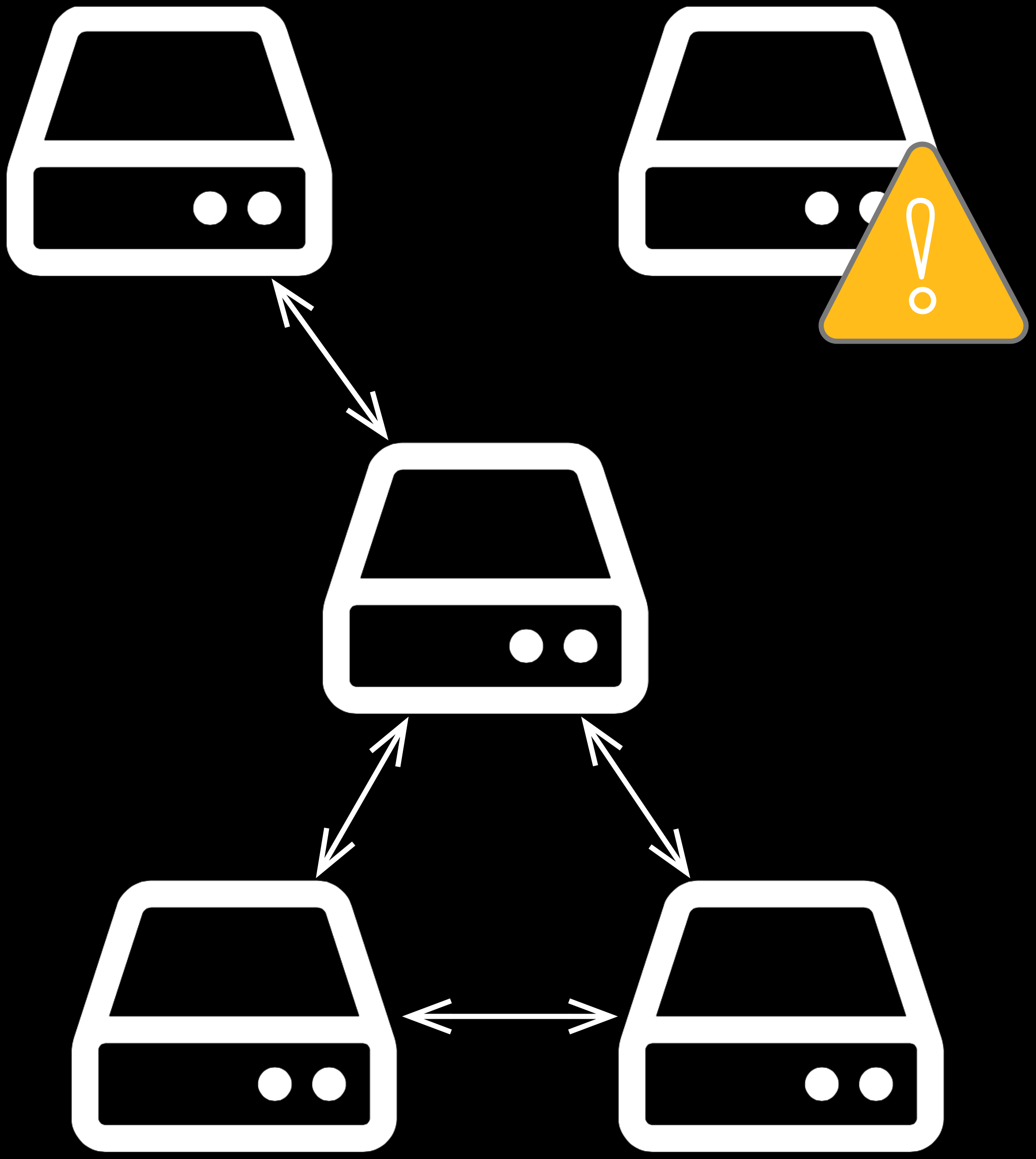


From Static Partitioning to Elastic Sharing



mesosphere

Mesos has no single point of failure

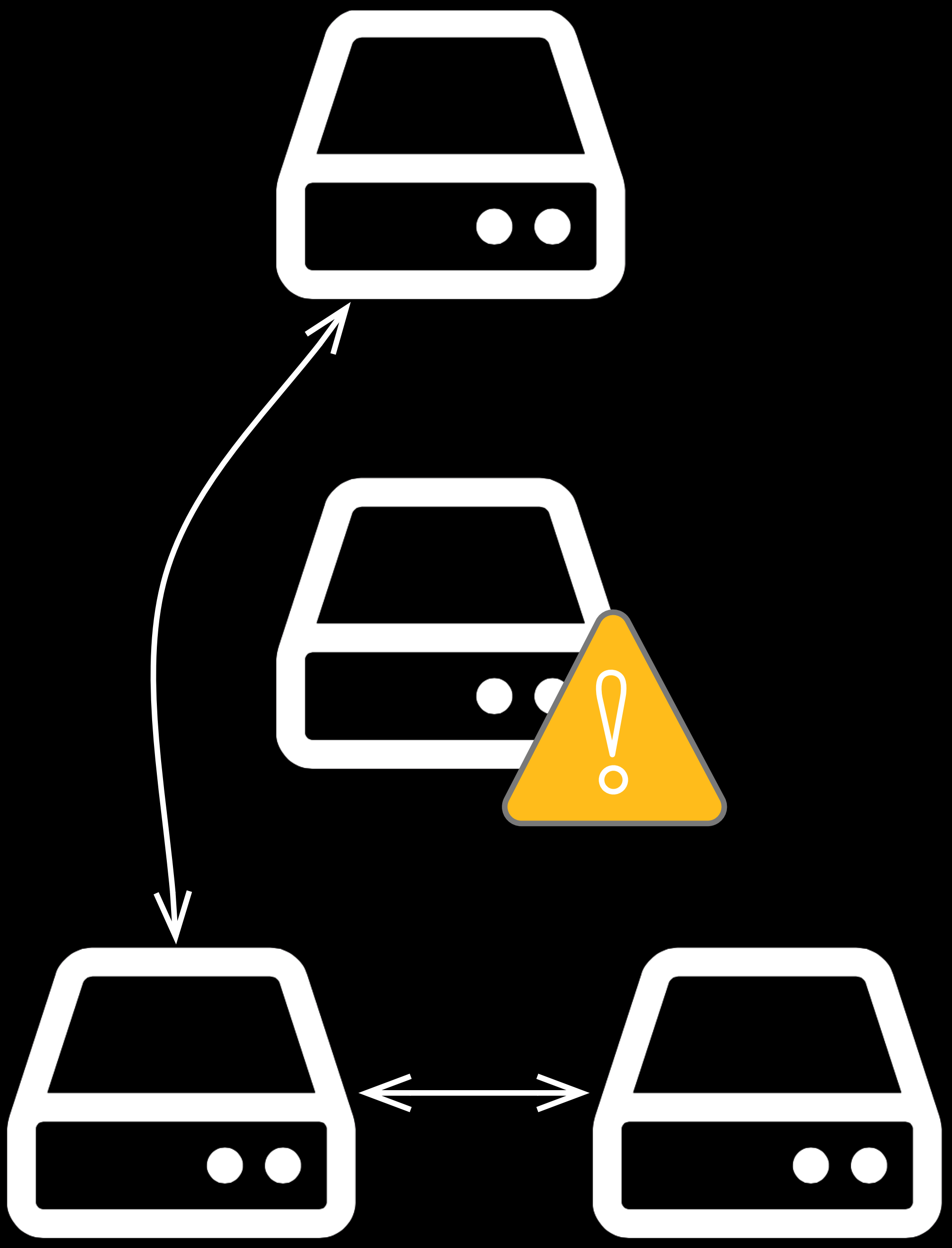


Framework

Masters

Tasks keep running!

Master node can fail-over

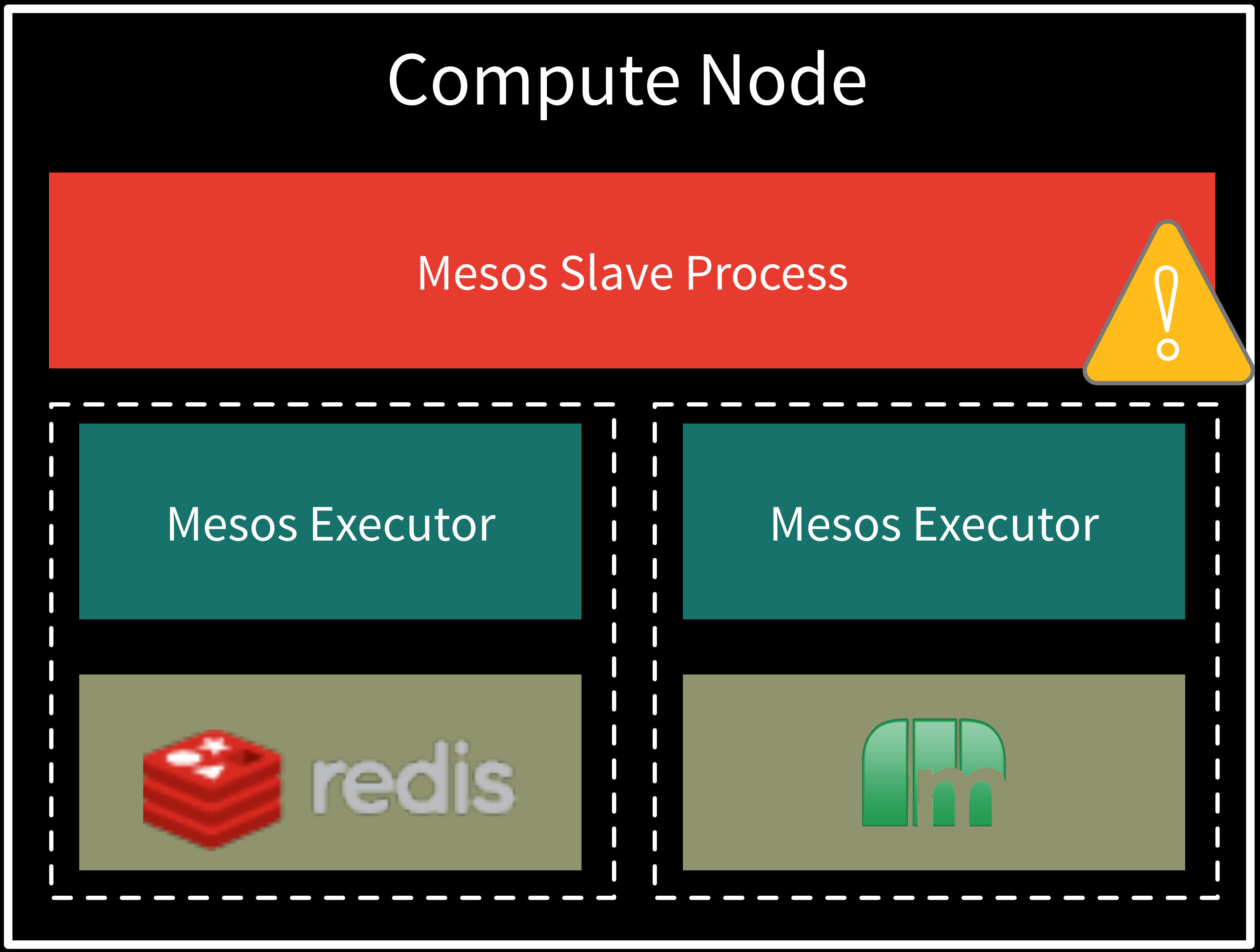


Framework

Masters

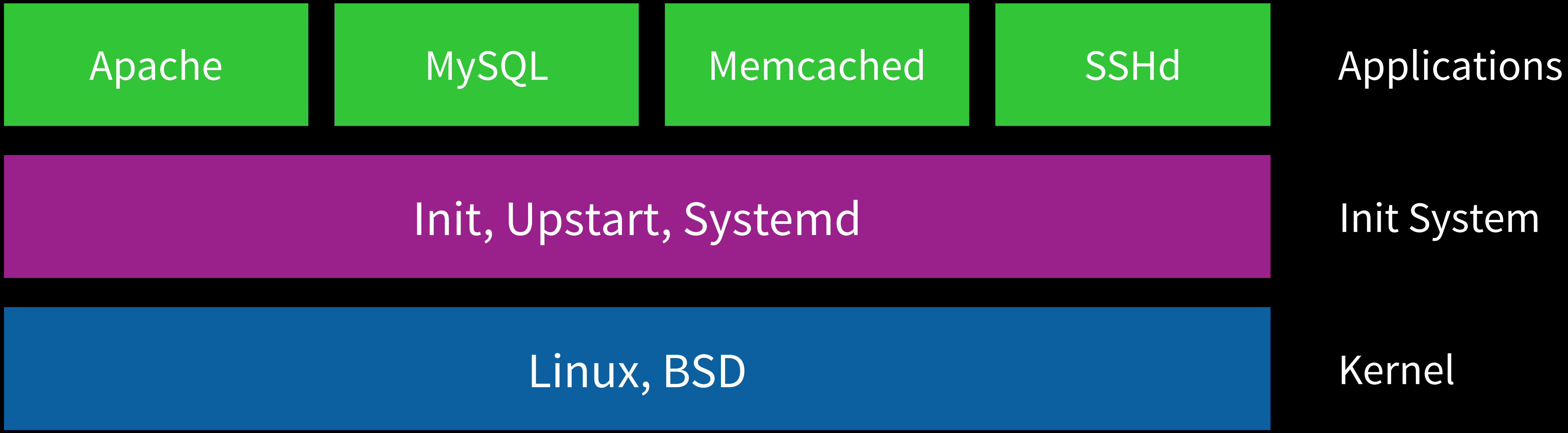
Tasks keep running!

Slave processes can fail-over



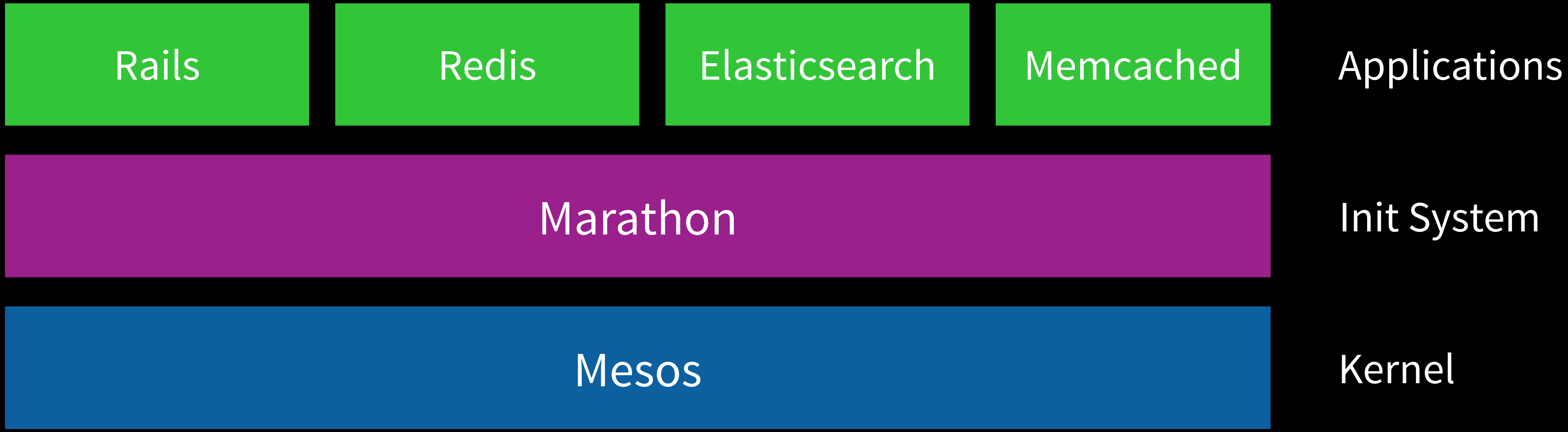
Tasks keep running!

The UNIX Operating System Stack



mesosphere

The Mesos Stack



mesosphere

Mesosphere Demo

mesosphere

Marathon

mesosphere

What is Marathon?

"Init Daemon" for the data center

- Runs any Linux binary without modification (e.g. Rails, Tomcat, …)
- Cluster-wide process supervisor

Private PaaS

- Service discovery
- Automated software and hardware failure handling
- Deployment and scaling

mesosphere

Marathon Design Goals

Simplify
Fewer things to manage = fewer sources of error
All machines have the same configuration

Automate
Nobody likes to get paged at night
Automatically respond to hardware & software failures

Improve efficiency
Humans are bad at estimating resource requirements
Use software-controlled elastic resource sharing

Self-serve API
Developers want to have control over their apps
Give them direct access to cluster resources

mesosphere

Marathon Key Features

REST/JSON API

Easy to use web interface

Authentication & SSL

Highly available – no single point of failure

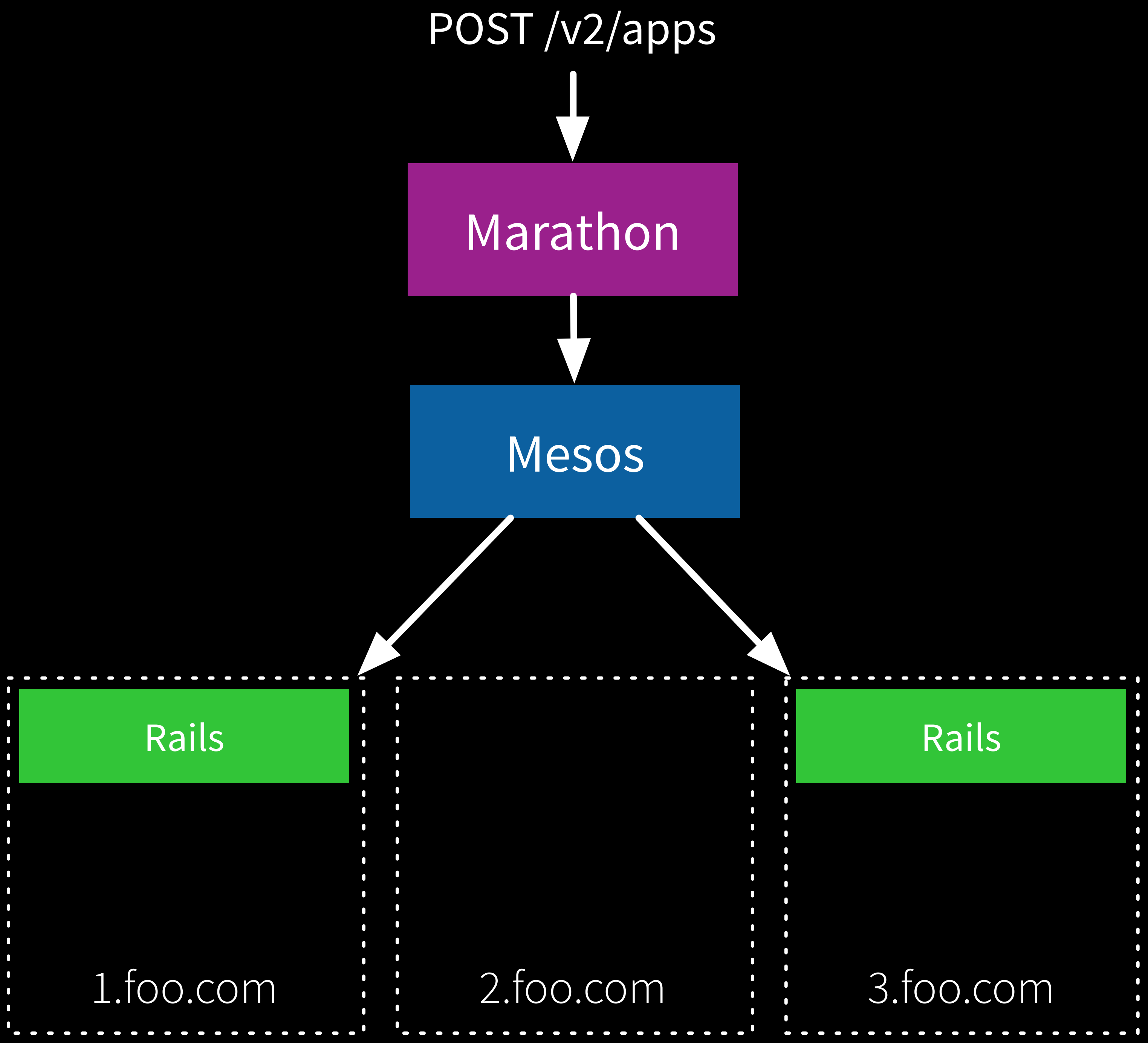
Placement constraints (nodes, racks, etc.)

Service discovery & load balancing via HAProxy

Event system for integration with 3rd party components, like load balancers

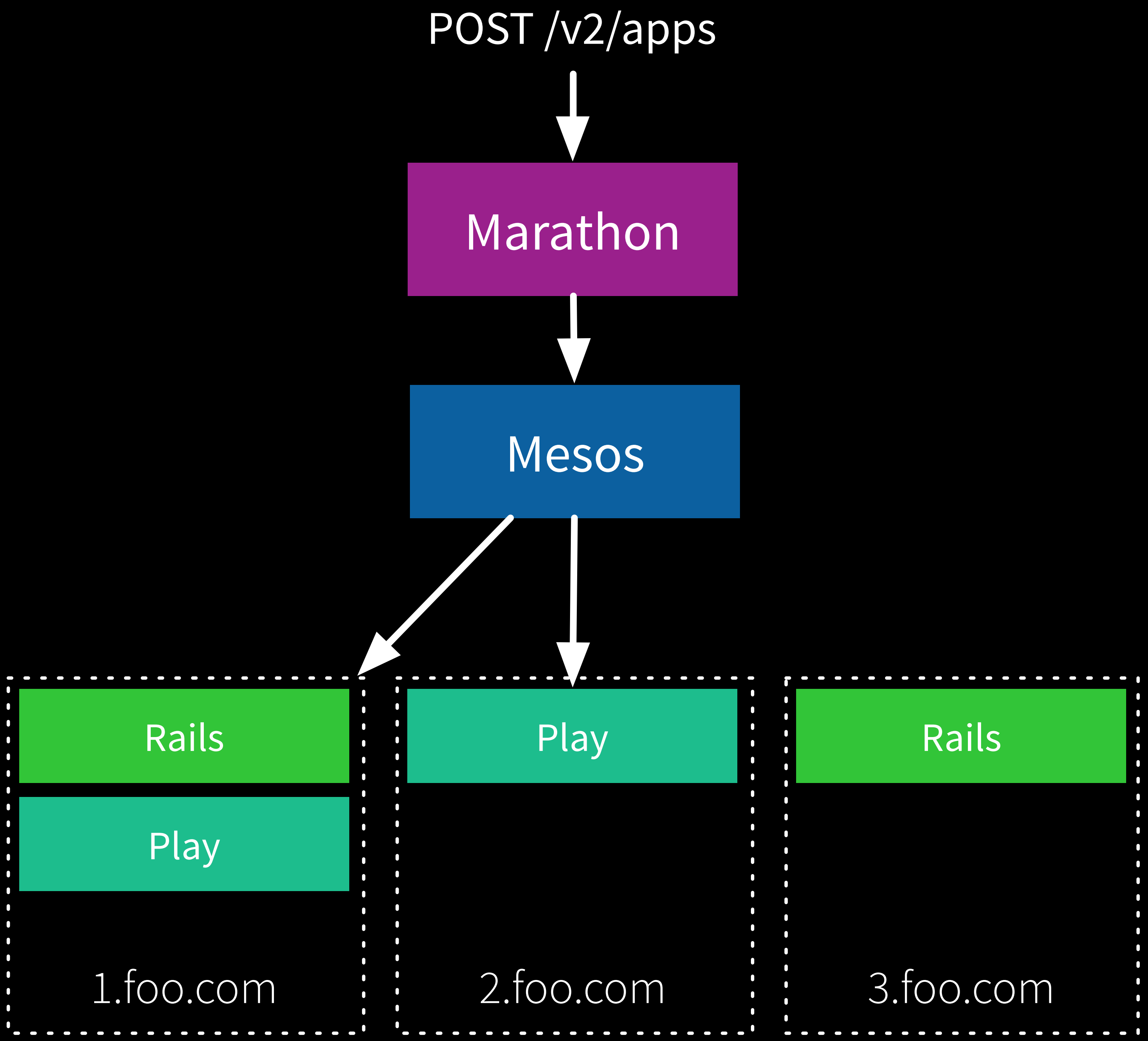
mesosphere

Marathon Workflow



mesosphere

Marathon Workflow



mesosphere

Marathon API - Launching Self-Contained Apps

- Command to start the app
- URL(s) to the app archive/configuration
- Environment variables

```
POST /v2/apps
{
    “id”: “Play”,
    “uris”: [“http://downloads.mesosphere.io/tutorials/
PlayHello.zip”]
    “cmd”: “./Hello-*/bin/hello -Dhttp.port=$PORT”,
    “env”: {“SECRET”: “password123”}
}
```

mesosphere

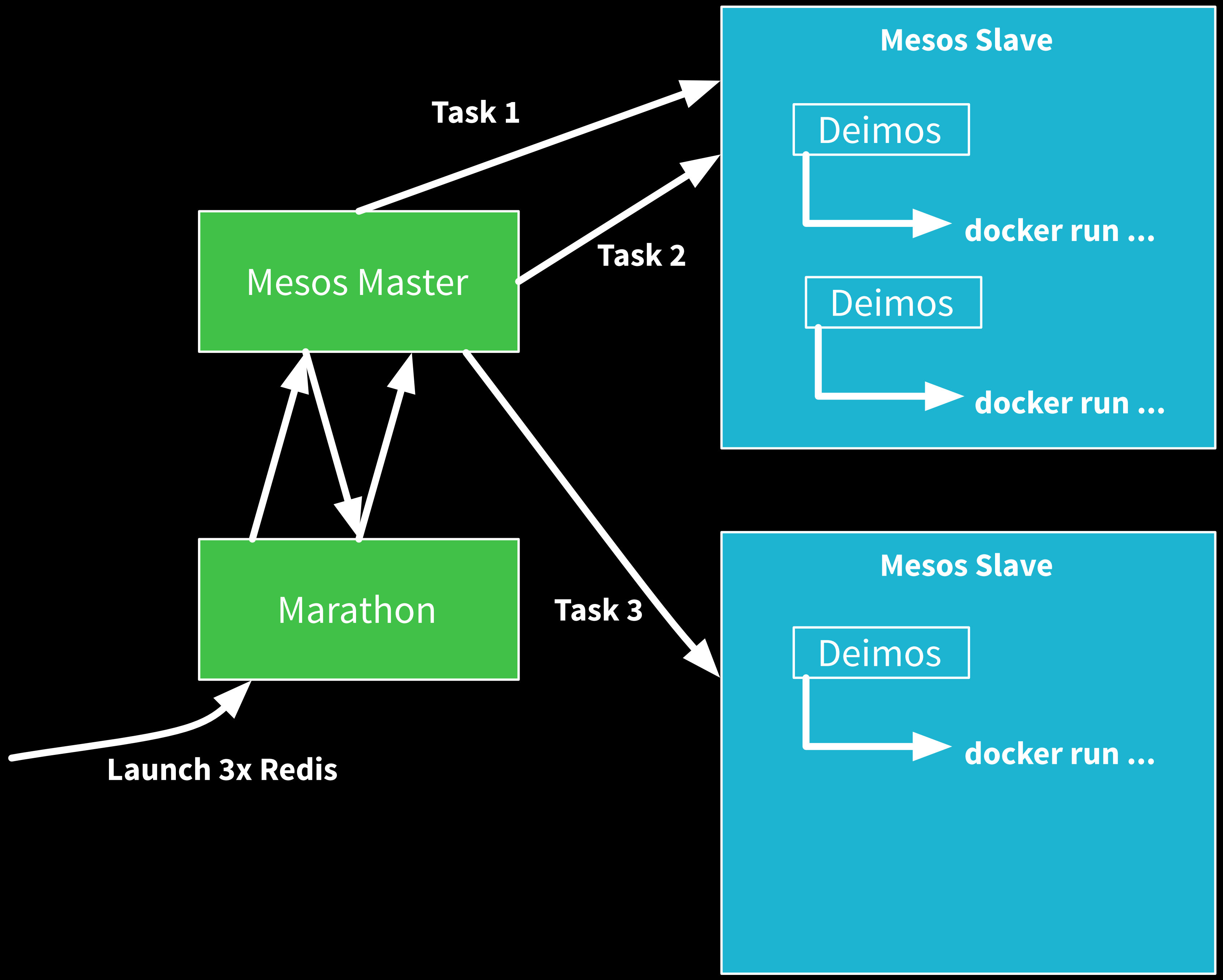
Marathon API - Launching Dockers

- Starting with Mesos 0.19 containers are 1st class citizens
- Deimos is the Docker containerizer

```
POST /v2/apps
{
    “id”: “Cassandra”,
    “container”: {
        “image”: “docker:///mesosphere/cassandra:2.0.6”,
        “options”: [“-v”, “/mnt:/mnt:rw”, “-e”,
“CLUSTER_NAME=prod”]
    }
}
```

mesosphere

Marathon API – Integration with Deimos

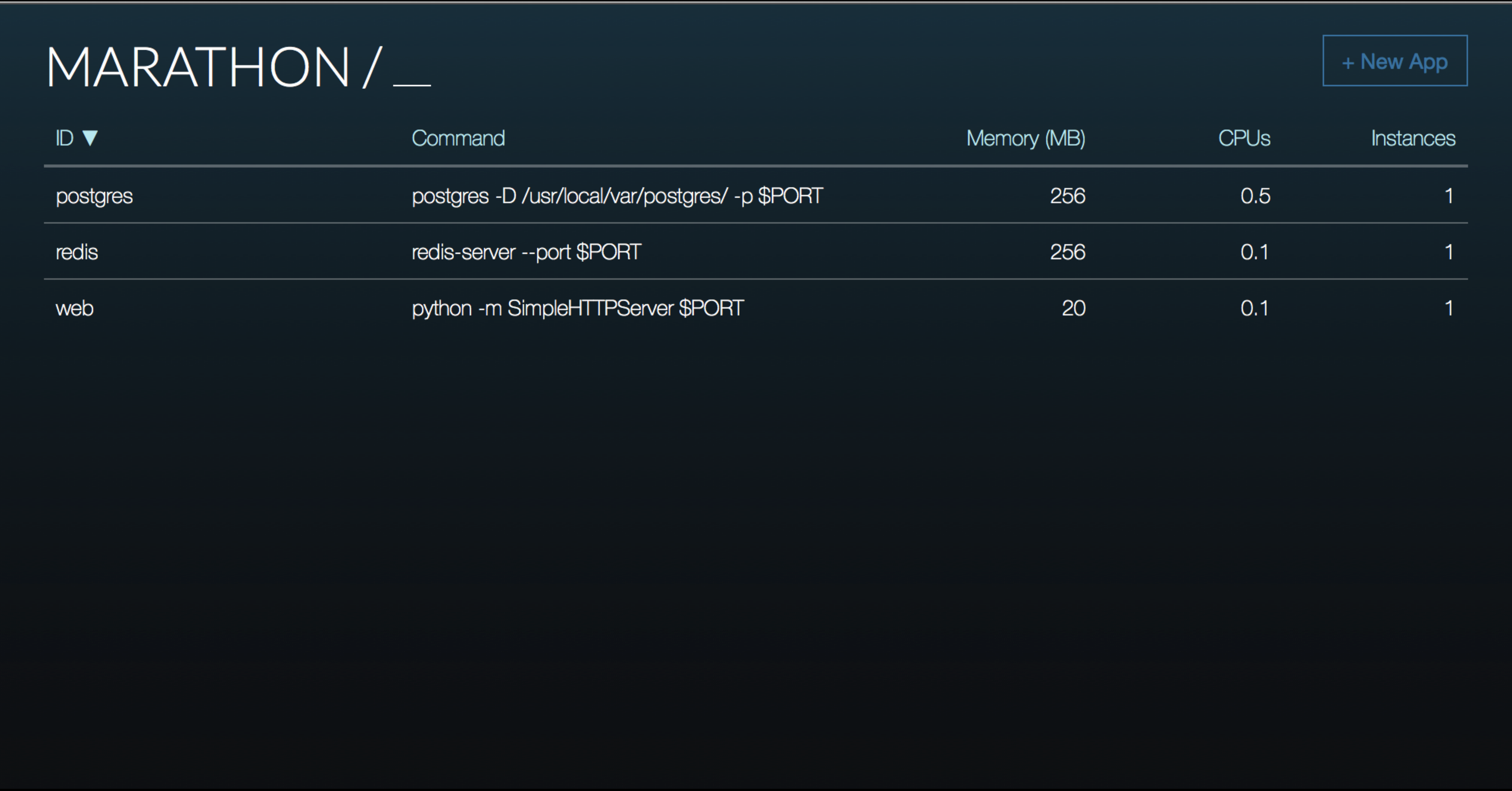


mesosphere

Marathon API - Scaling Apps

- Just tell Marathon how many you want!

```
PATCH /v2/apps/Play
{
    “instances”: 4
}
```

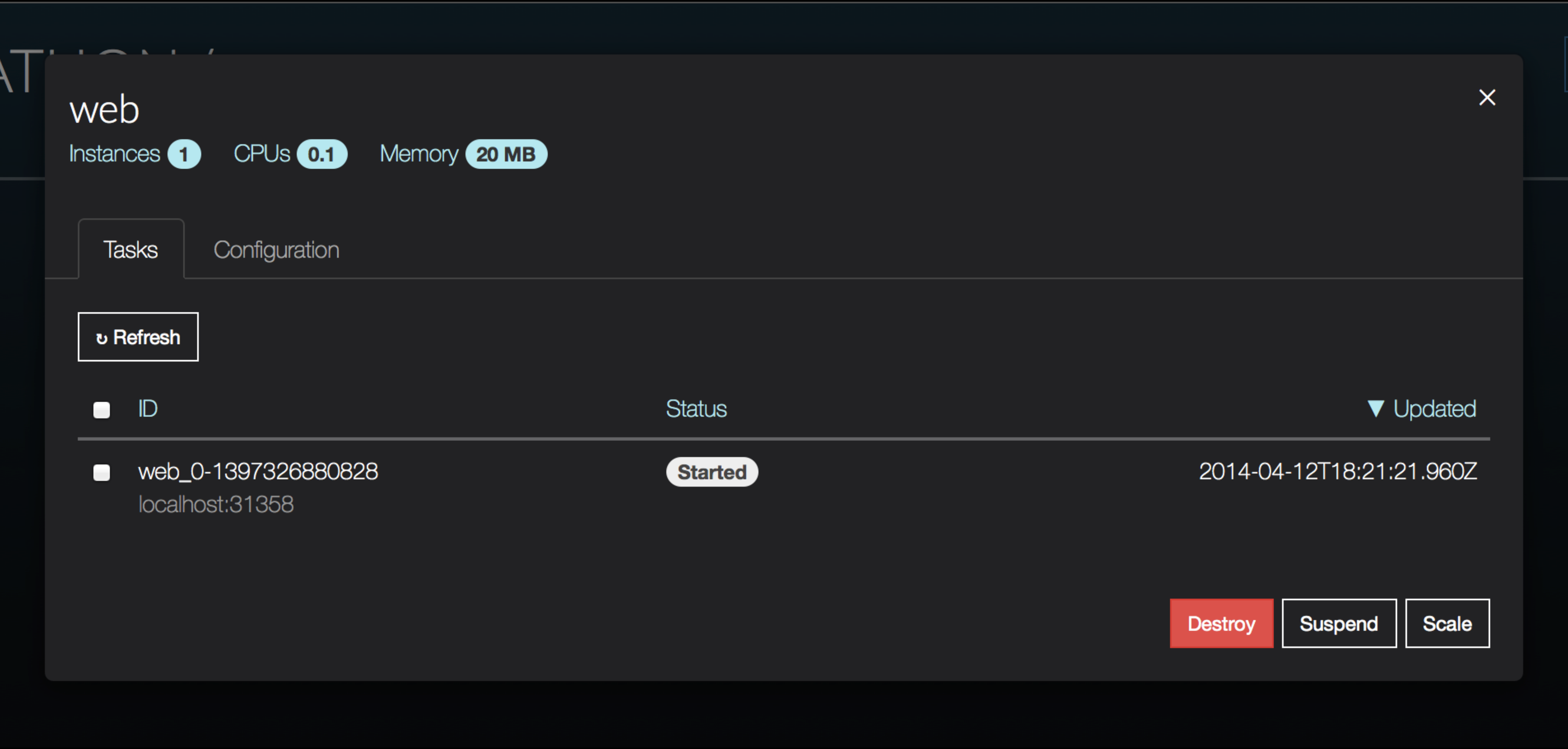


MARATHON / _

+ New App

ID ▼	Command	Memory (MB)	CPUs	Instances
postgres	postgres -D /usr/local/var/postgres/ -p $PORT	256	0.5	1
redis	redis-server --port $PORT	256	0.1	1
web	python -m SimpleHTTPServer $PORT	20	0.1	1

mesosphere

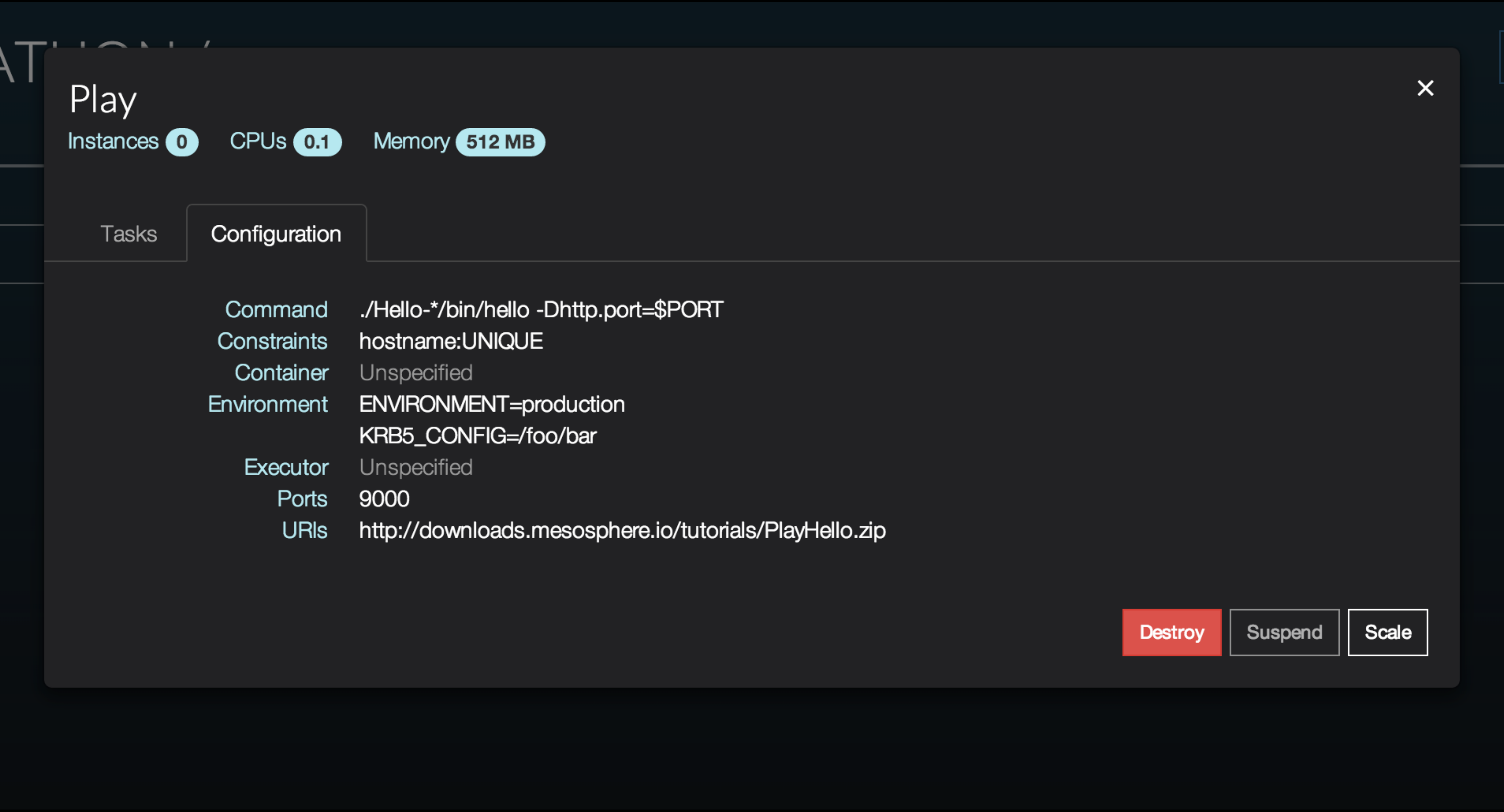


web

Instances 1 CPUs 0.1 Memory 20 MB

ID	Status	▼ Updated
web_0-1397326880828 localhost:31358	Started	2014-04-12T18:21:21.960Z

mesosphere



Play

Instances 0 CPUs 0.1 Memory 512 MB

Command	./Hello-*/bin/hello -Dhttp.port=$PORT
Constraints	hostname:UNIQUE
Container	Unspecified

Environment	ENVIRONMENT=production
	KRB5_CONFIG=/foo/bar

Executor	Unspecified
Ports	9000
URIs	http://downloads.mesosphere.io/tutorials/PlayHello.zip

mesosphere

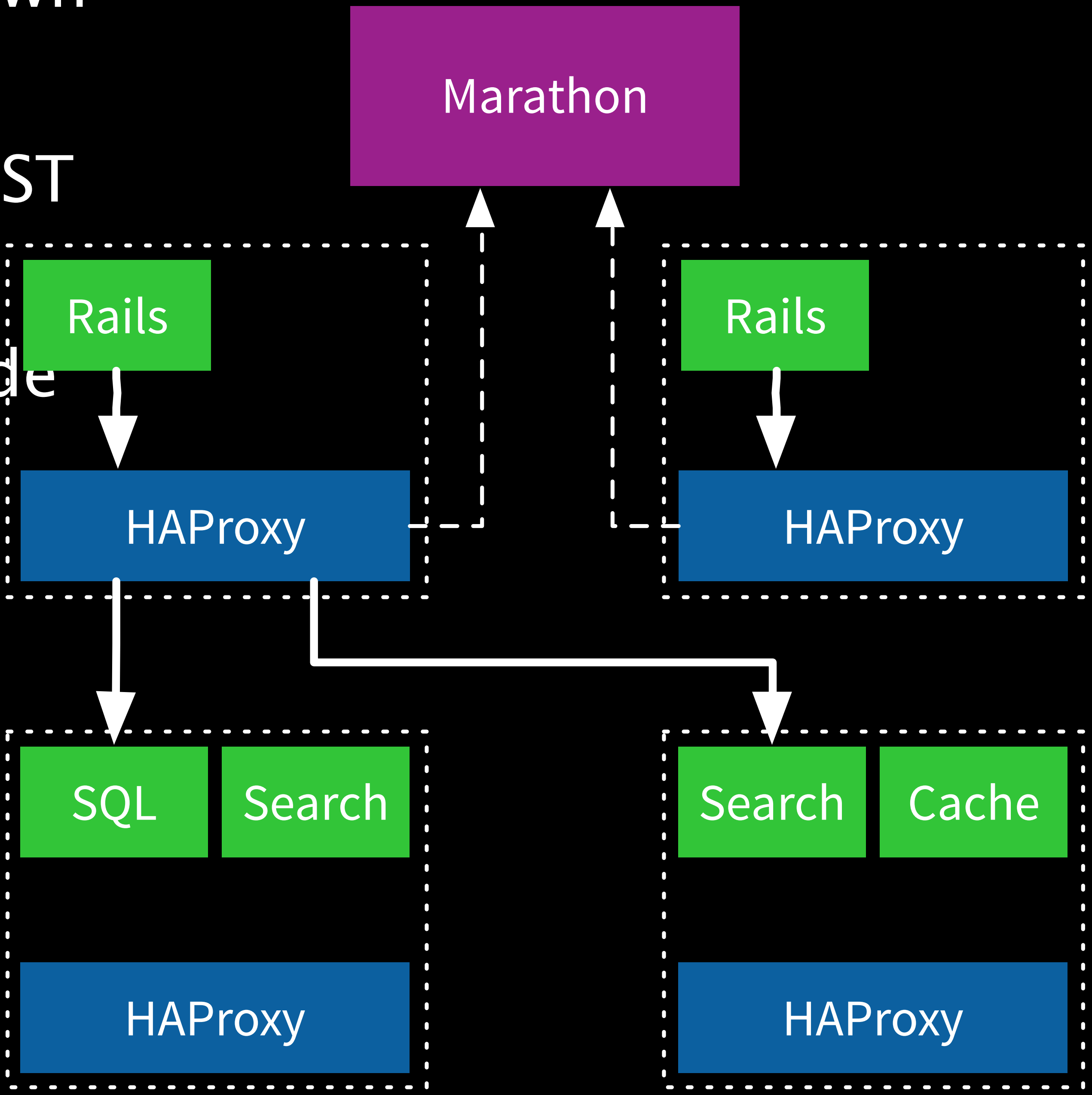
Marathon Service Discovery with HAProxy

Apps available on localhost & known port

HAProxy updates via Marathon REST API

HAProxy runs on every cluster node

Configurable policies



mesosphere

Mesosphere Products & Services

Infrastructure

- Try out Mesos on Amazon Web Services

  https://elastic.mesosphere.io

- On premise and cloud provisioning tools
- Linux Packages
- Mesosphere Plugins and Tools
- Professional Services
  - Training, Installation, Support Contracts

mesosphere

Thank you.

mesosphere

mesosphere