MESOS, DC/OS AND THE ARCHITECTURE OF THE NEW DATACENTER

MESOSPHERE

WHOAMI

benjamin.bannier@mesosphere.io

MESOSPHERE

ParStream

STONY BROOK
STATE UNIVERSITY OF NEW YORK

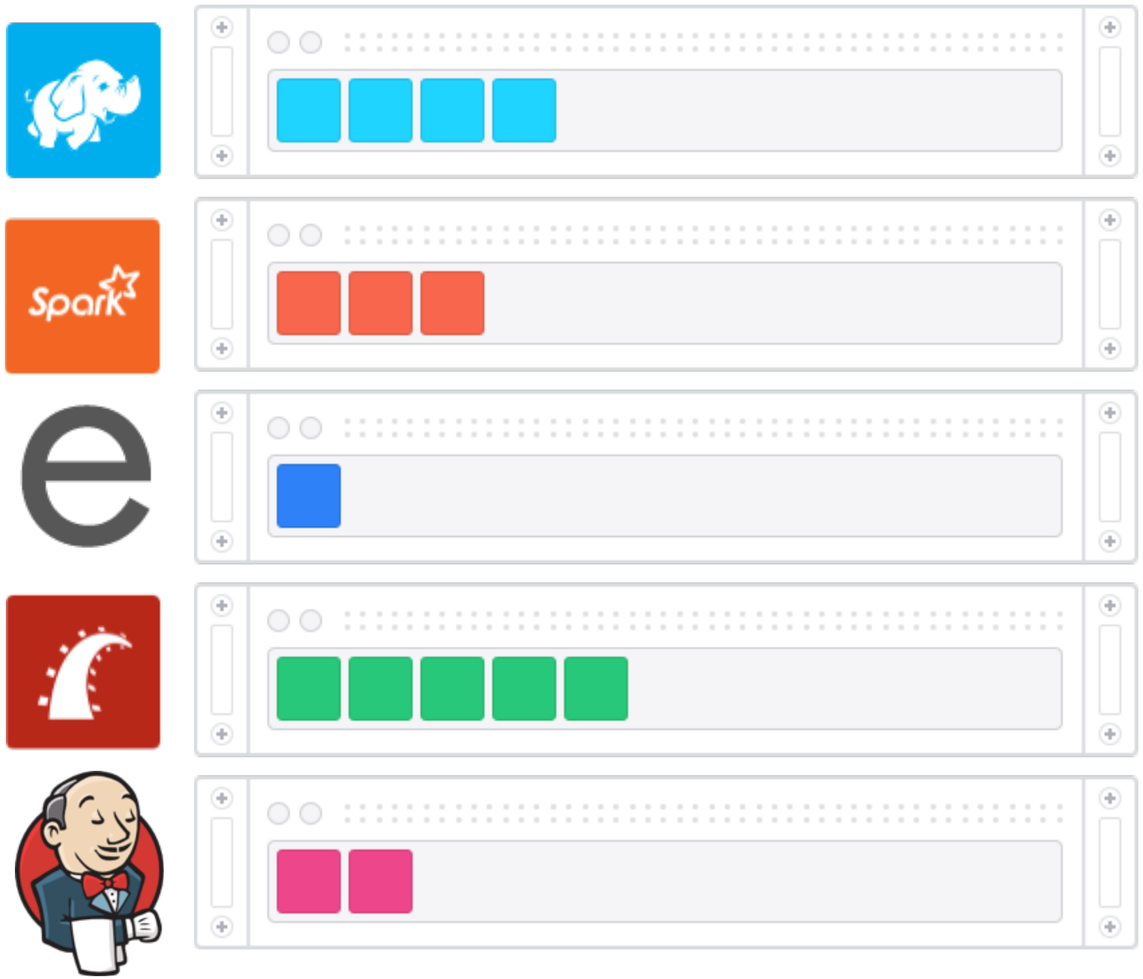
2015 -	Mesosphere
2014 - 2015	Distributed databases
2007 - 2014	PhD, High energy nuclear physics

EVOLUTION OF APPLICATIONS

NEW FORM FACTOR FOR DEVELOPING AND RUNNING APPS

UNIT OF INTERACTION	PARTITION (LPAR)	SERVER	VIRTUAL MACHINE	**DATACENTER**
				
	MAINFRAME	**PHYSICAL (x86)**	**VIRTUAL**	**HYPERSCALE**
DEFINITIVE APPS AND OS	• DATA / TRANSACTION PROCESSING • UNIX, IBM OS/360	• ERP, CRM, PRODUCTIVITY, MAIL & WEB SERVER • LINUX, WINDOWS	• ERP, CRM, PRODUCTIVITY, MAIL & WEB SERVER • HYPERVISOR + GUEST OS	• BIG DATA, INTERNET OF THINGS, MOBILE APPS • **DATACENTER OPERATING SYSTEM (DC/OS)**

STATIC PARTITIONING

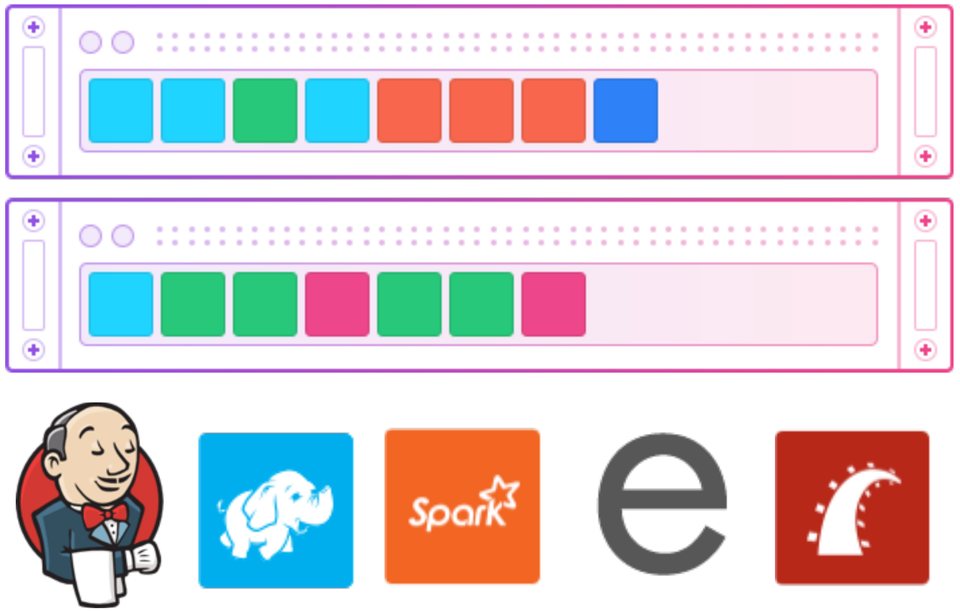


Deals well with heterogeneous workloads.

Headroom left for peak workloads will lead to poor overall resource utilization.

When resource requirements change potentially costly reconfigurations.

SHARED RESOURCES



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

Run everything in containers!


tex
tex


tex

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.

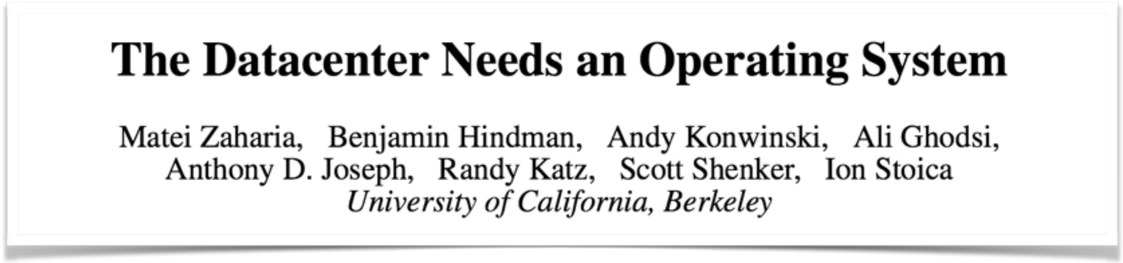
April 2016

DC/OS

VISION


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


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

Sharing resources between batch processing frameworks

- Hadoop
- MPI
- Spark

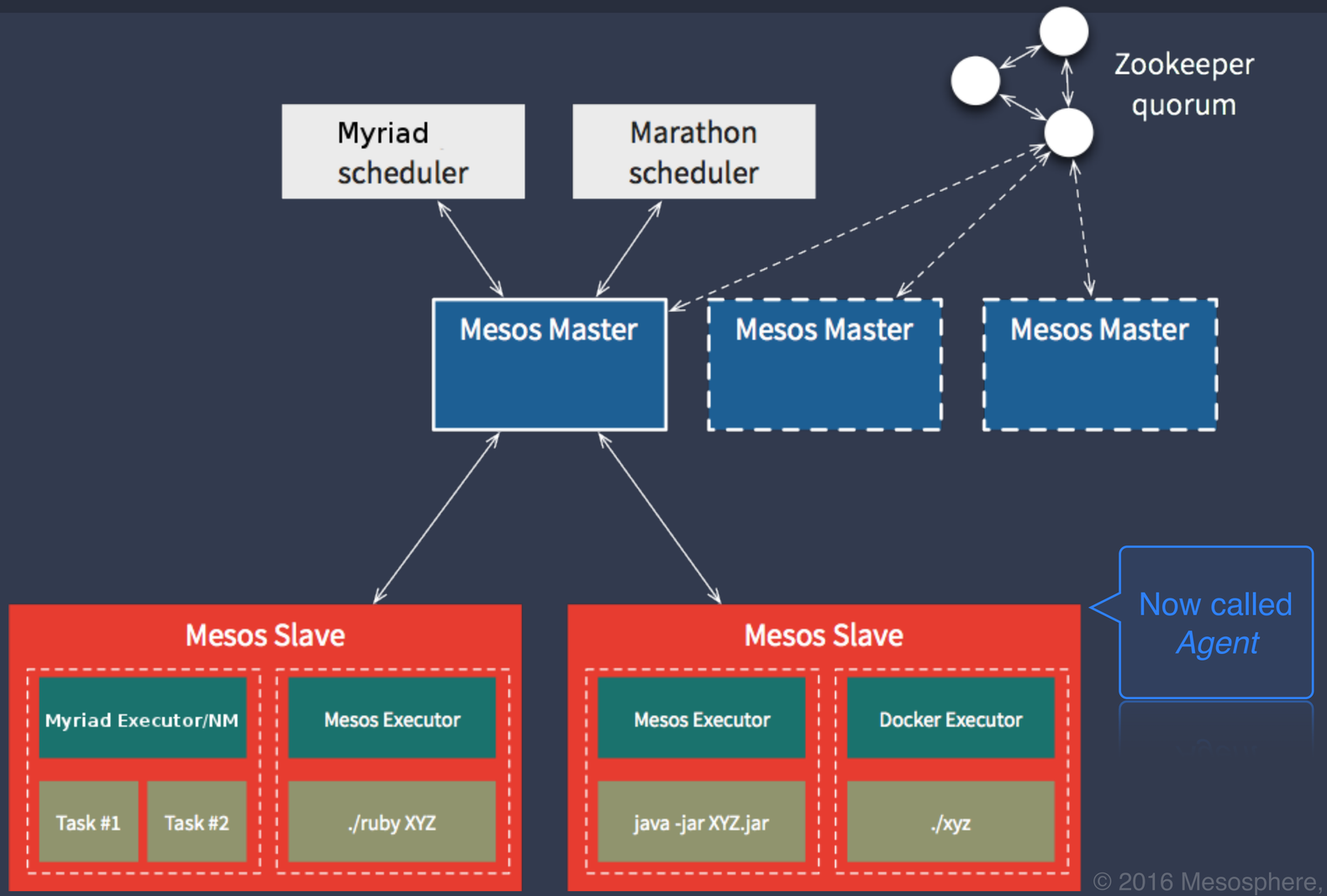
What does an operating system provide?

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

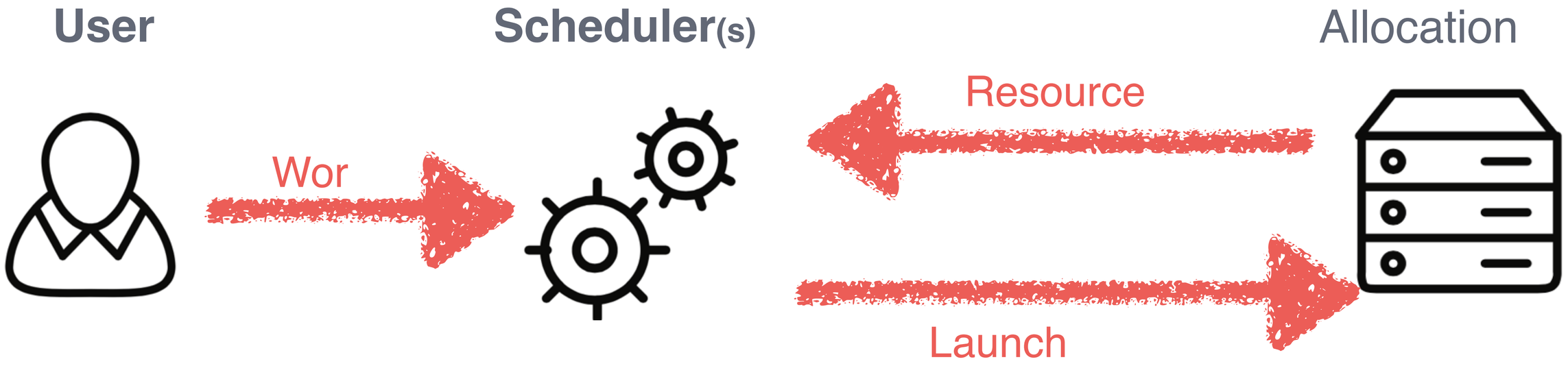
Apache Mesos

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

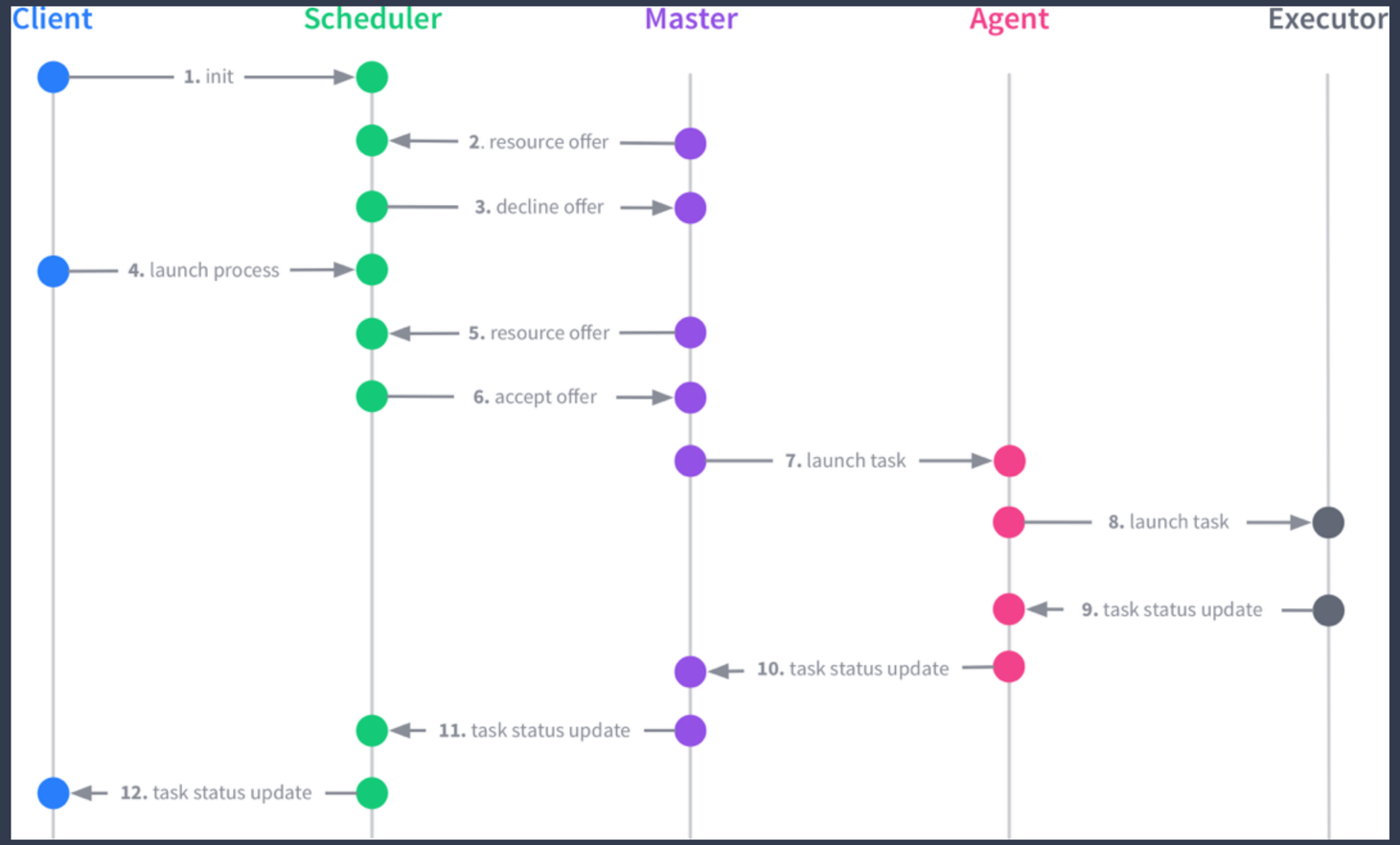
ARCHITECTURE

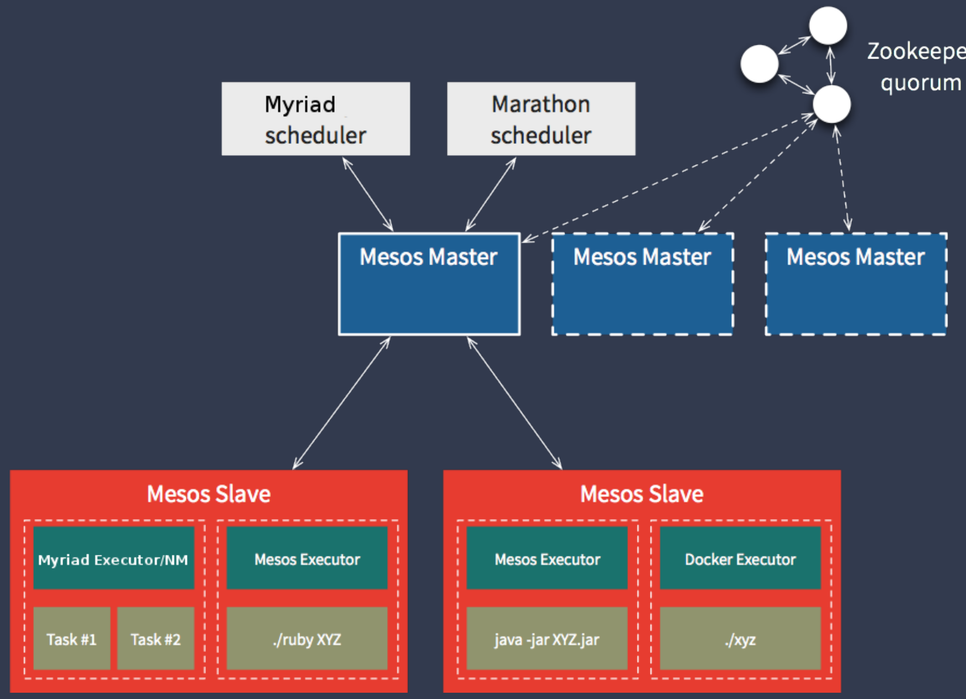


TWO-LEVEL SCHEDULING

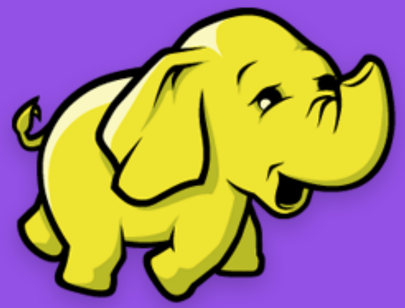


TWO-LEVEL SCHEDULING





apache™ myriad



YARN

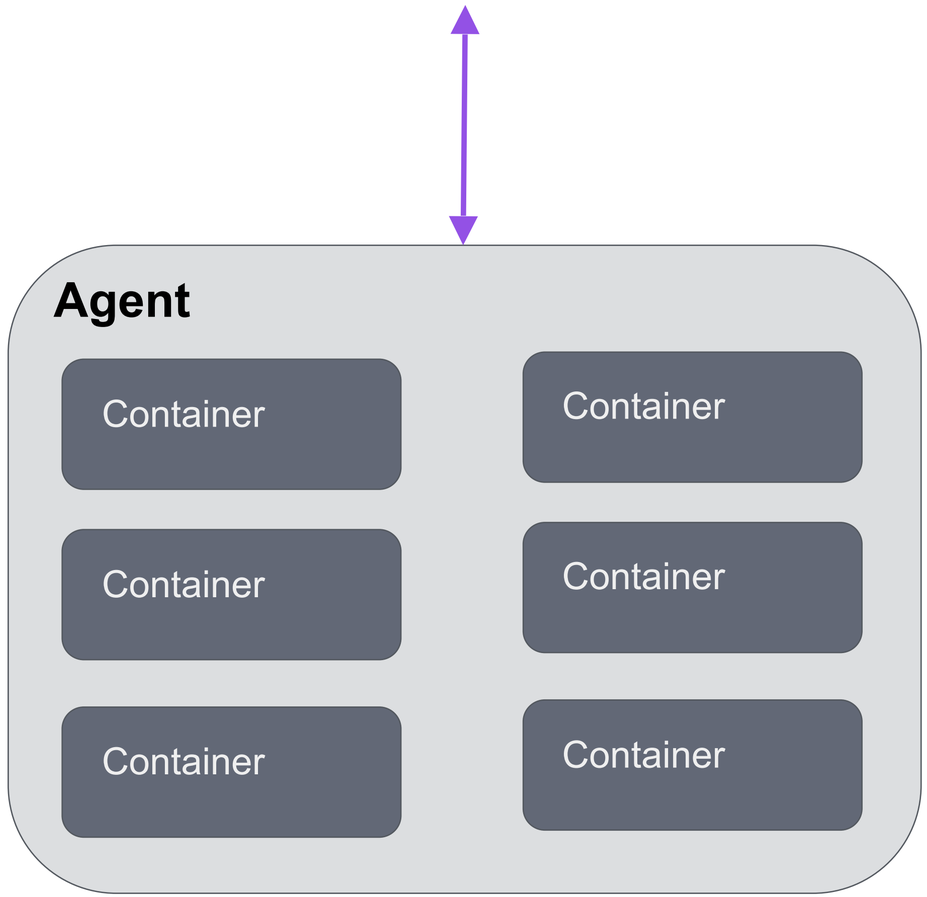
CONTAINER NETWORKING

CONTAINER NETWORKING

Containers isolate tasks on the agent, but what about their communication?

The status quo in a Mesos cluster: one IP per agent.

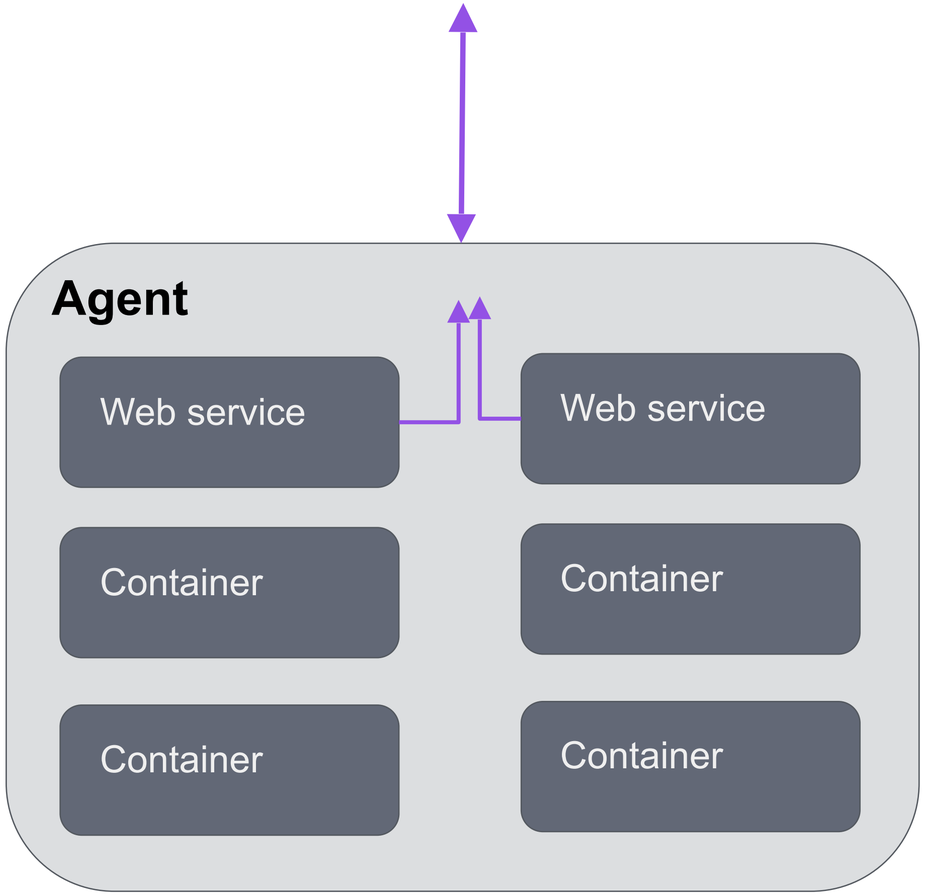
Many containers per agent: they must share a single IP.



CONTAINER NETWORKING

This causes headaches:

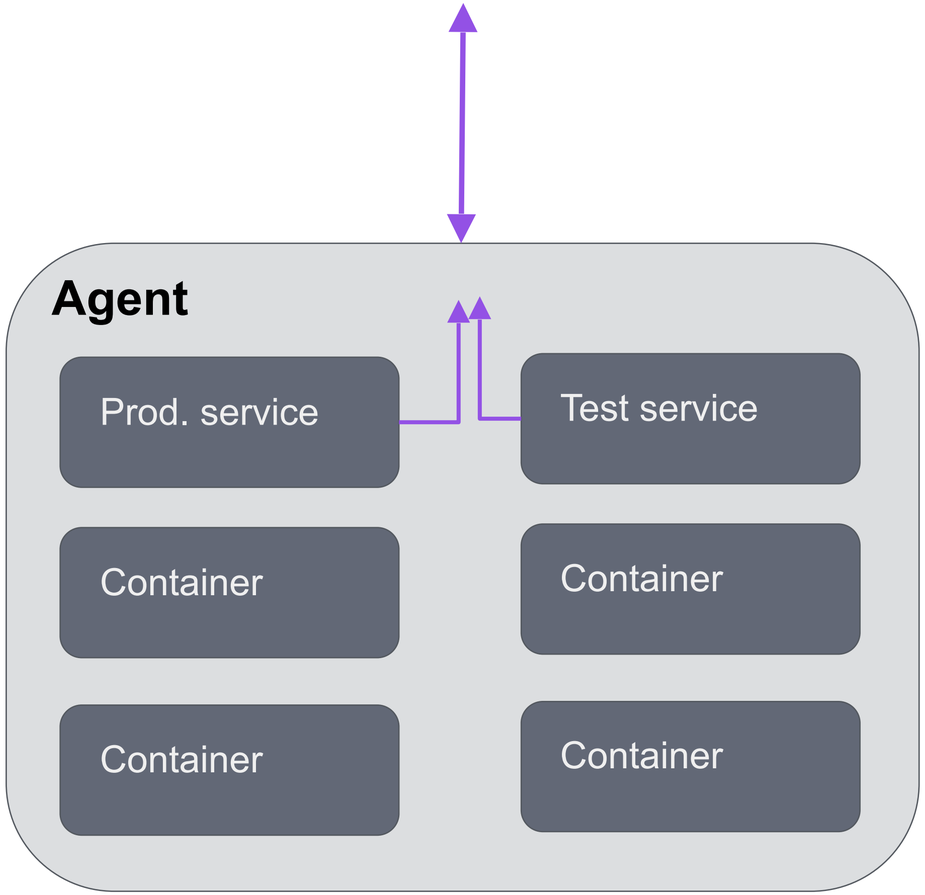
- Port conflicts
- Security compromises
- Performance
- Service discovery



CONTAINER NETWORKING

This causes headaches:

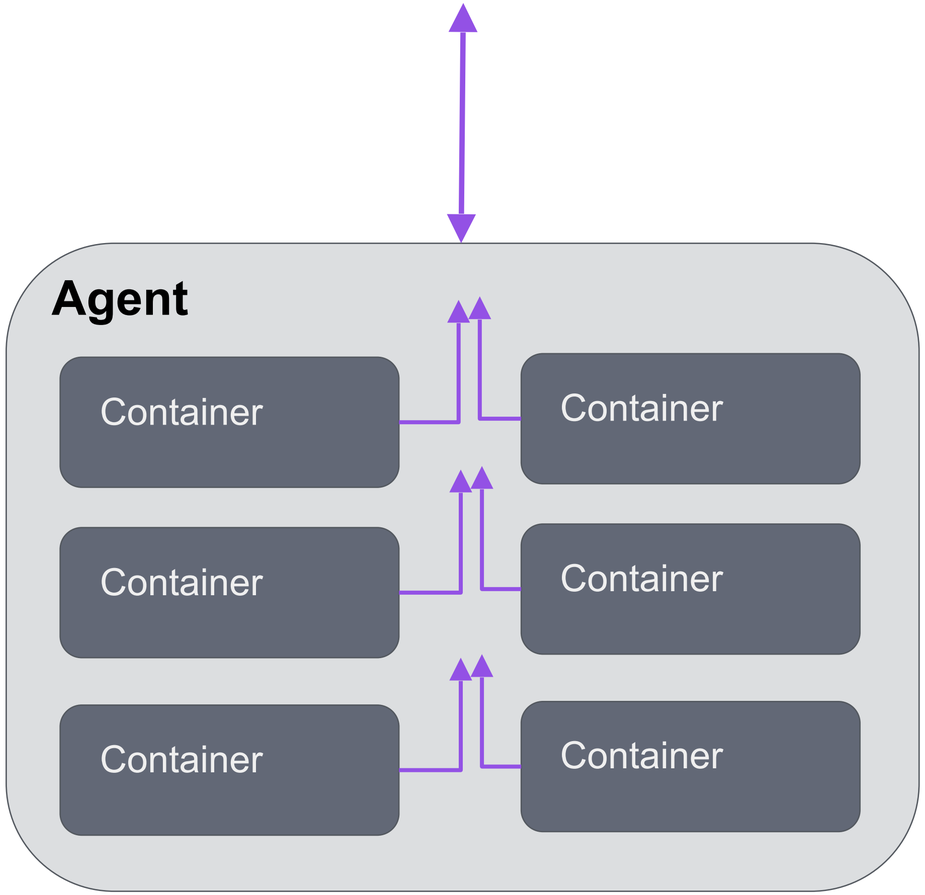
- Port conflicts
- Security compromises
- Performance
- Service discovery



CONTAINER NETWORKING

This causes headaches:

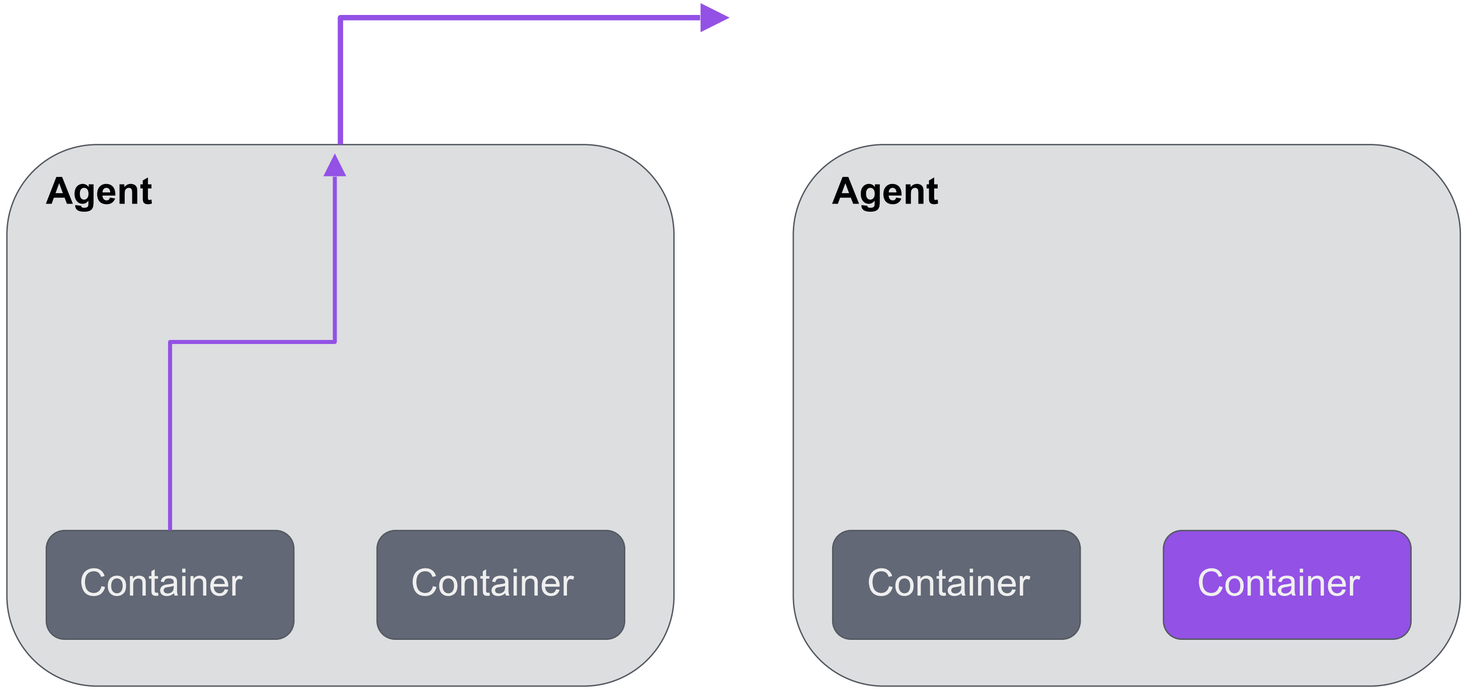
- Port conflicts
- Security compromises
- Performance
- Service discovery



CONTAINER NETWORKING

This causes headaches:

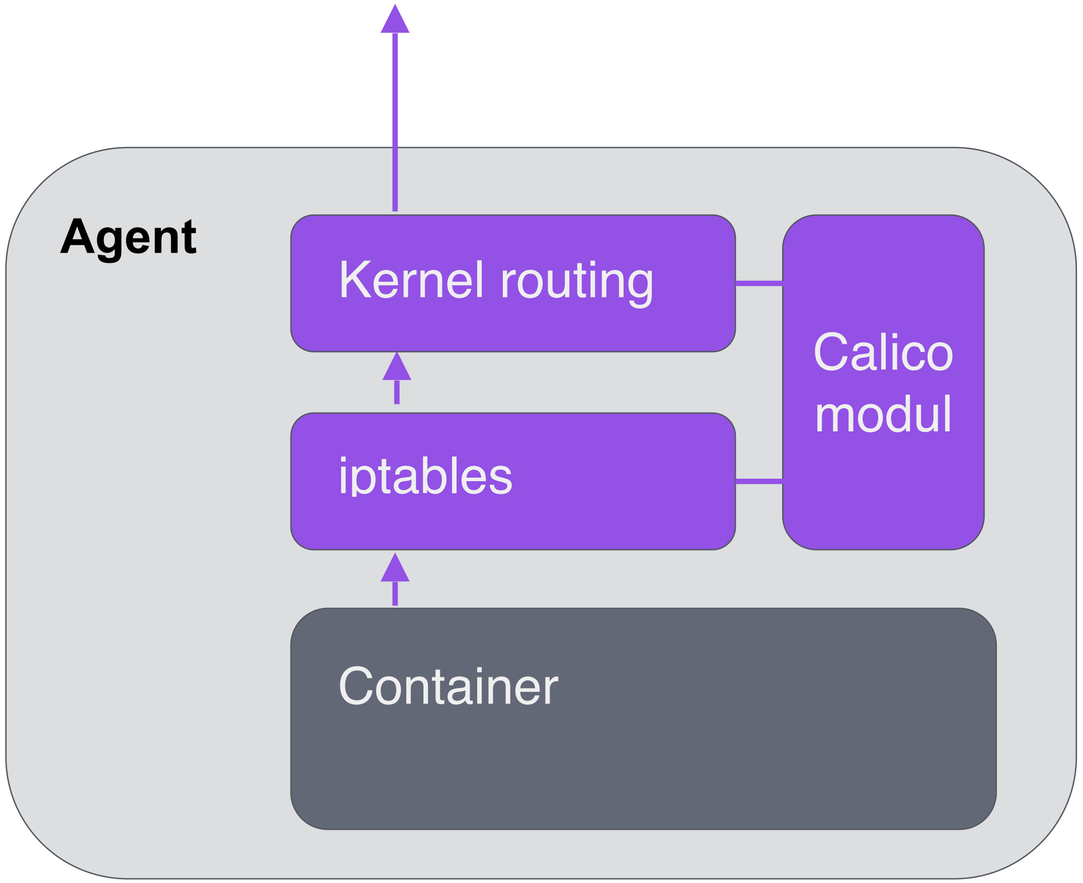
- Port conflicts
- Security compromises
- Performance
- Service discovery



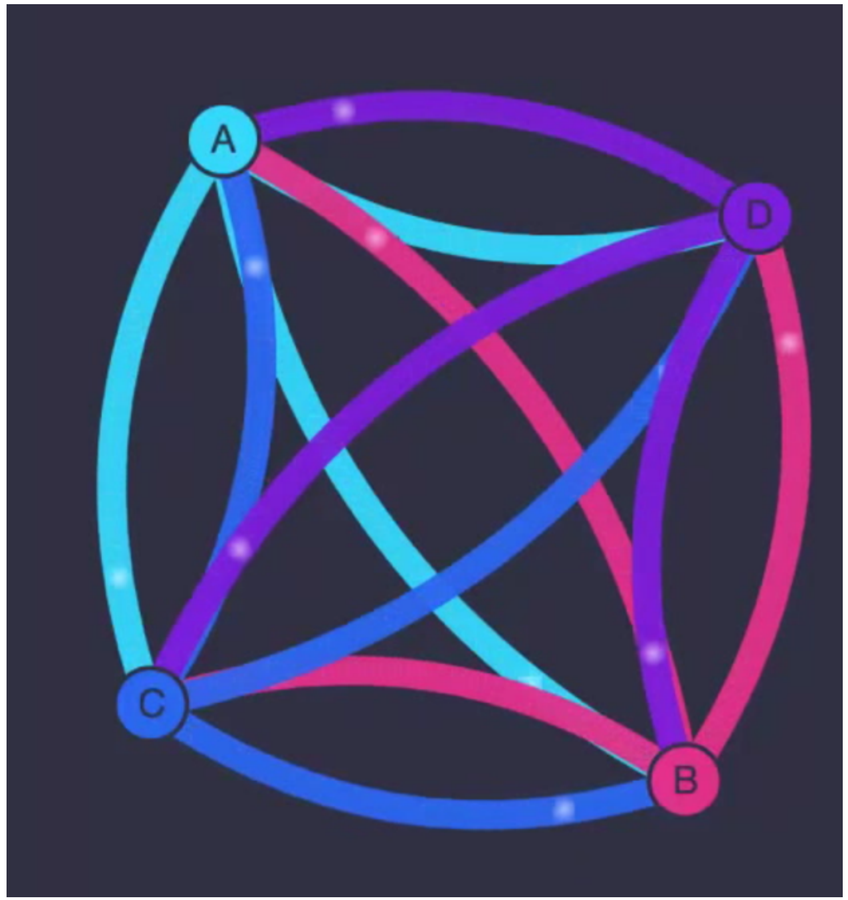
IP PER CONTAINER

Segregating containers' network traffic can solve these problems in an elegant, maintainable way.

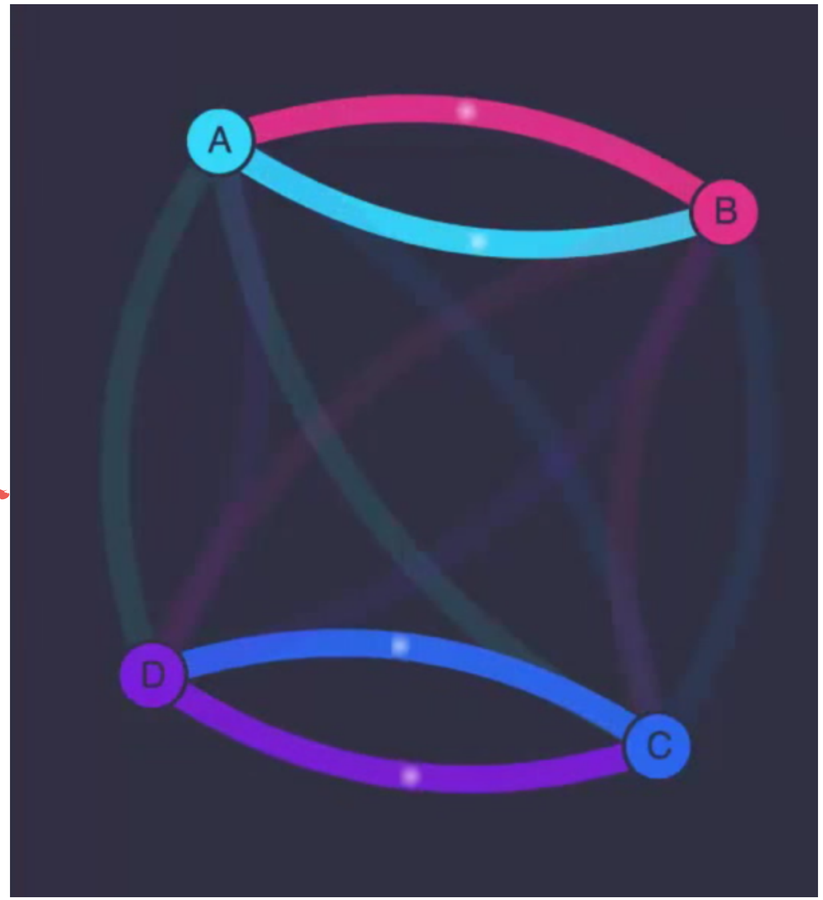
CALICO NETWORK ISOLATION



CALICO NETWORK ISOLATION



PROJECT CALICO



“Developers will ask for an API not for a VM to run their software”

DISTRIBUTED SYSTEMS SDK

Distributed System SDK

Focus on application logic, not on data-center structure

- Avoid networking-related code
- Reuse of built-in fault-tolerance and high availability
- Reuse distributed (infrastructure) frameworks (e.g., storage)

FIRST FRAMEWORK/SERVICE

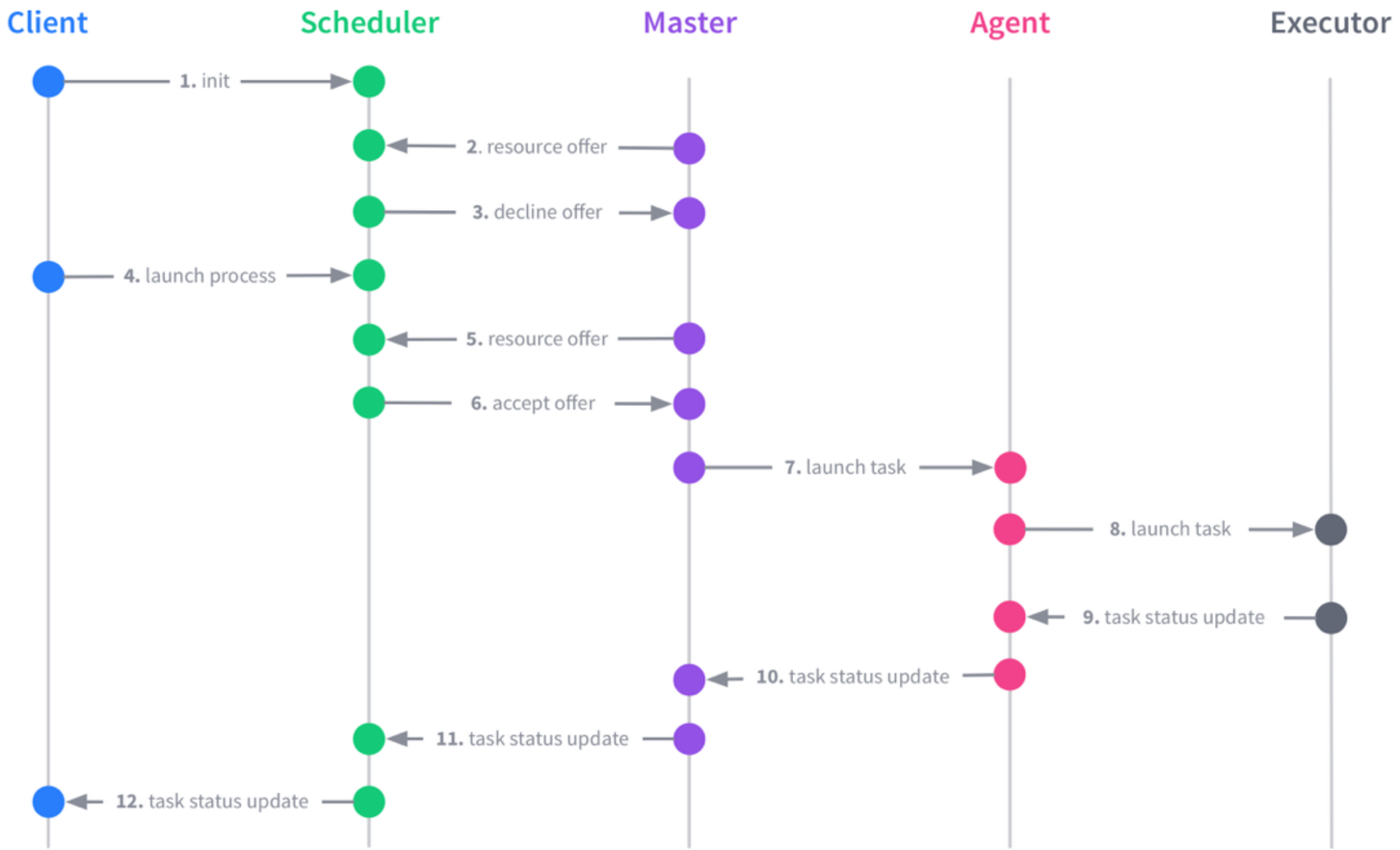
Scheduler Interface

Utilize resources

`ResourceOffer`

React on Tasks Events

`TaskStatus` Update



RENDLER

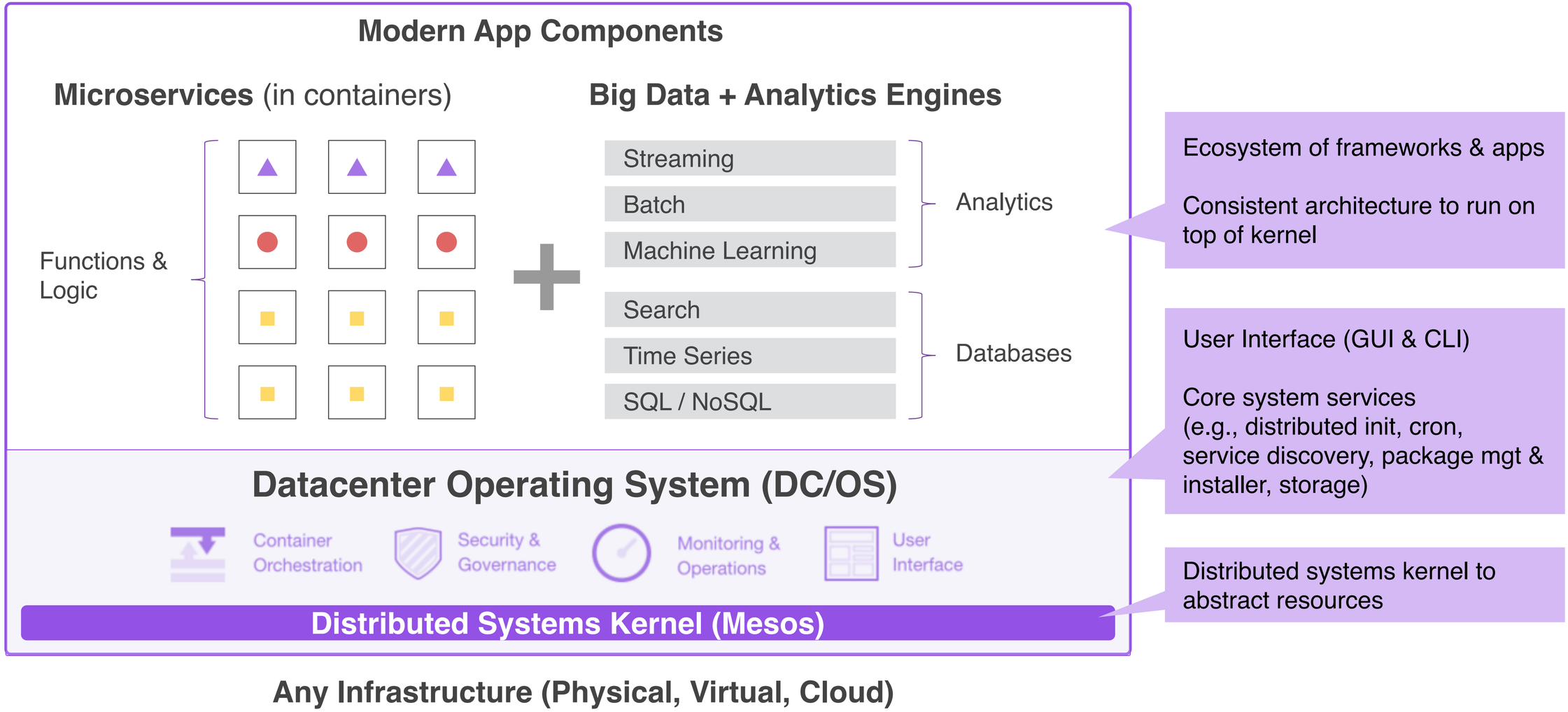
Scheduler Interface

Examples implementations for different languages:

C++, Go, Haskell, Java, Python, Scala

https://github.com/mesosphere/RENDLER

DC/OS ENABLES MODERN DISTRIBUTED APPS



DC/OS is …

- 100% open source (ASL2.0)
  - A big, diverse community
- An umbrella for ~30 OSS projects
  - Roadmap and designs
  - The build tool chain
  - Docs and tutorials
- Not limited in any way
- Familiar, with a few new features

DC/OS

Why DC/OS Install Docs Commu

The easiest way to run containers in production.

Get Started Play Video

Containers & Big Data

Easily deploy and run stateful or stateless distributed workloads including Docker containers, big data, and traditional apps.

100% Open Source

Run DC/OS in production without any scale or performance limitations, on any on-premises or cloud platform.

Production Pro

Built on Apache Me the experience of Mesosphere, Yelp, T Airbnb, and many o most innovative com

DC/OS

Why DC/OS Install Docs Community GitHub Get Started

Run: Jenkins On: Azure

Step 1 **Select a service** Step 2 Select a platform Step 3 Run it

Marathon

Use to launch long-running apps, for example a webserver.

Spark

Use for elastic data processing, batch or streaming.

ArangoDB

Use to store and query tree-like data as well as graphs.

Cassandra

Use to store and query semi-structured data in wide tables.

CHR

QUESTIONS?

http://apache.mesos.org

http://dcos.io

http://mesosphere.com