CLOUD NATIVE
COMPUTING
FOUNDATION

Cloud Native and Container Technology Landscape

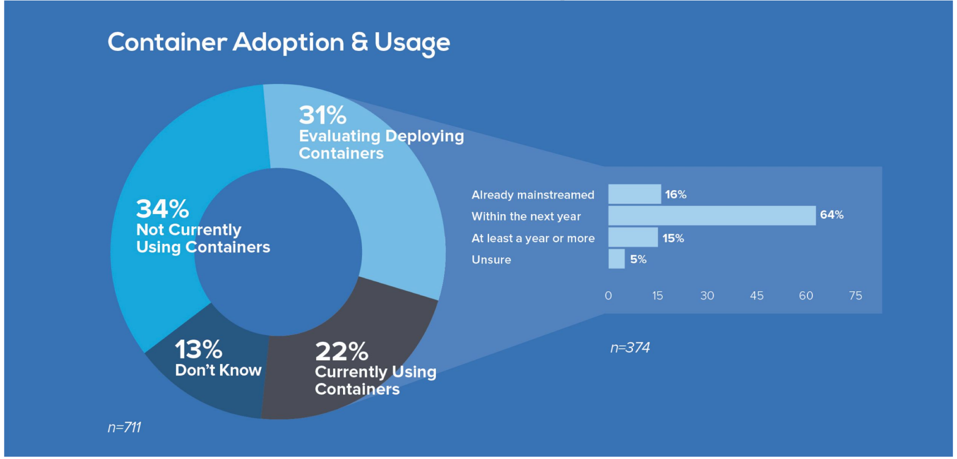
Chris Aniszczyk (@cra)

Rise of Containers and Cloud Native Computing!

- Google running 2B+ containers per week!
  - Internet scale companies are running containers too: Facebook, Twitter, Netflix, etc
- 75%+ companies are experimenting with containers!
  - https://www.blackducksoftware.com/2016-future-of-open-source
- PokemonGo on containers (via Kubernetes and GCE)!
  - https://cloudplatform.googleblog.com/2016/09/bringing-Pokemon-GO-to-life-on-Google-Cloud.html

Containers Adoption is Still Growing (But Fragmented)!

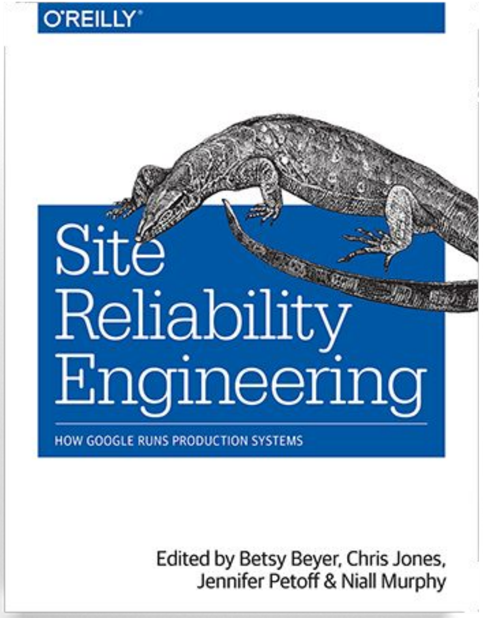
- Rapid growth in container adoption has led to the **need to standardize, integrate and collaborate on container technology…**
- **Fragmentation: Docker, rkt, Kurma, LXC/LXD, Hyperd, OpenVZ, …**
- Desire to not be bound to orchestration system, OS, arch, vendor, cloud etc…



https://www.cloudfoundry.org/wp-content/uploads/2016/06/Cloud-Foundry-2016-Container-Report.pdf

Lessons via Internet Scale Companies (i.e., Google)

- Sysadmins (Traditional Approach):
  - respond to events/issues as they occur (manual work)
  - grow team to absorb work as service grows
  - ops is fundamentally at odds with dev (resistance to changes)
- Site Reliability Engineers [SRE] (Cloud Native Approach)
  - software engineers do operations! automation vs manual labor
  - SREs get bored doing manual tasks, **automate them!**
  - culture of blameless postmortems
- Google: 1 SRE per 10000+ machines
- How did they get there?


O'REILLY
Site Reliability Engineering
HOW GOOGLE RUNS PRODUCTION SYSTEMS
Edited by Betsy Beyer, Chris Jones, Jennifer Petoff & Niall Murphy

https://landing.google.com/sre/book.html

Non-Virtualized Servers: Sun (2000)

- Launching a new application? Buy a new server; or a rack of them!
- Building block of your application is physical servers

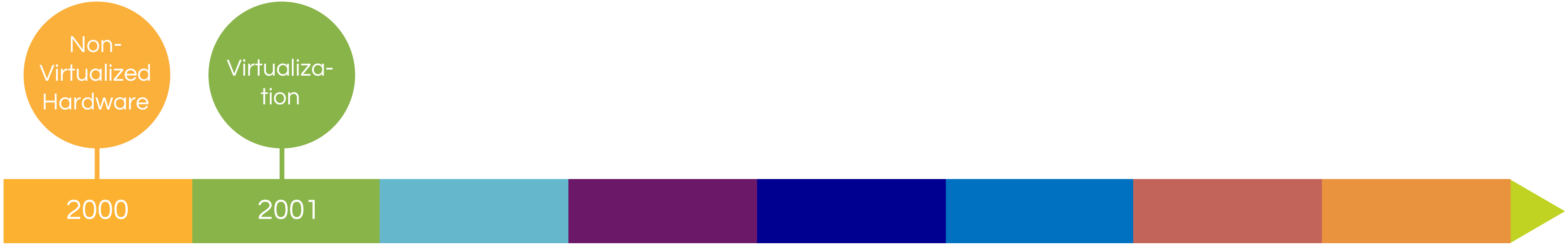
Non-Virtualized Hardware

2000

Virtualization: VMWare (2001)

vmware®

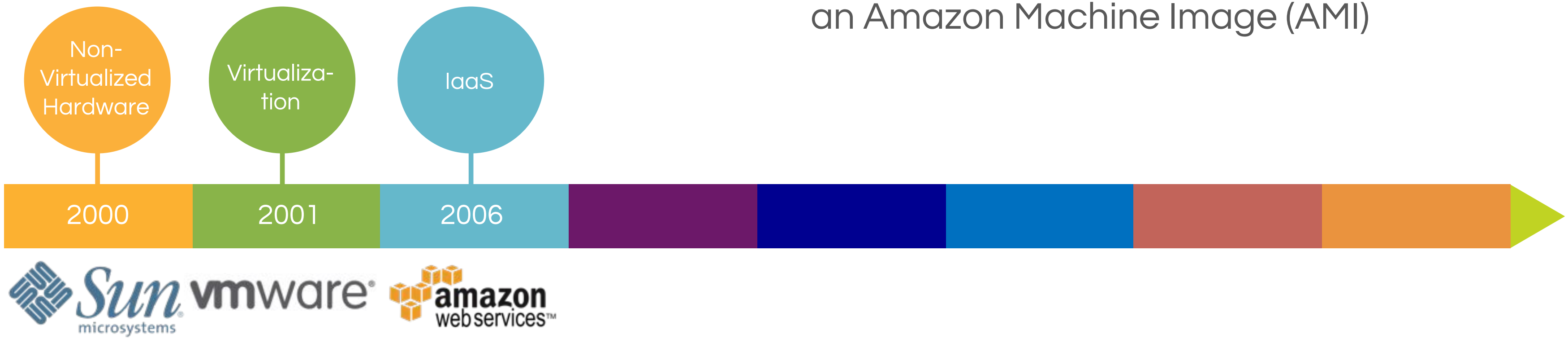
- Releases for server market in 2001
- Popularizes virtual machines (VMs)
- Run many VMs on one physical machine, meaning you can buy less servers!
- Architectural building block becomes a VM



Sun microsystems vmware®

IaaS: AWS (2006)

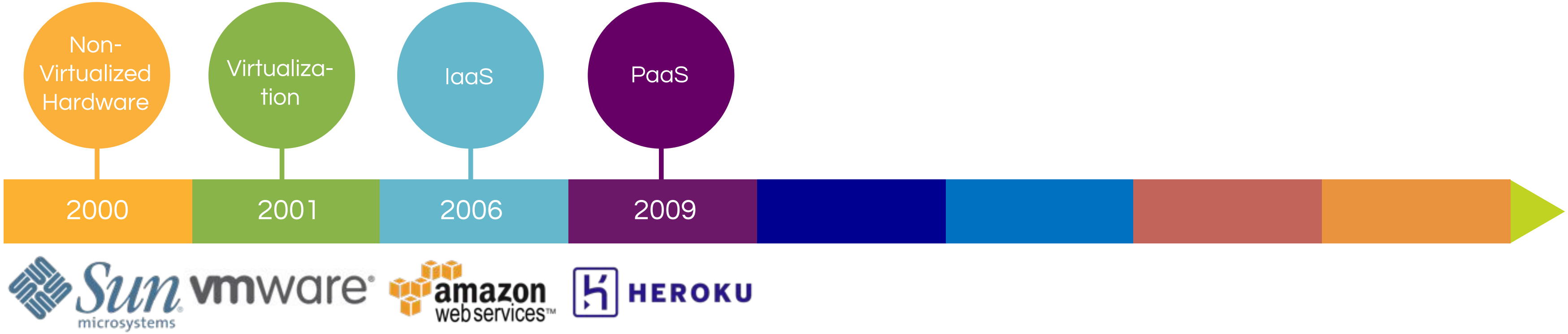
- Amazon Web Services (AWS) creates the Infrastructure-as-a-Service market by launching Elastic Compute Cloud (EC2) in 2006
- Rent servers by the hour
- Convert CapEx to OpEx
- Architectural building block is also a VM, called an Amazon Machine Image (AMI)



PaaS: Heroku (2009)

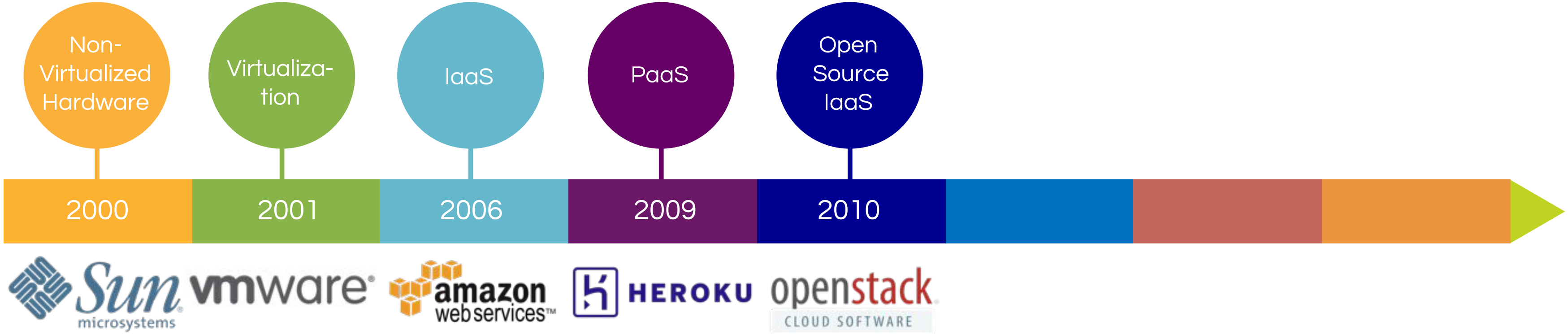
HEROKU

- Heroku popularizes Platform-as-a-Service (PaaS) with their launch in 2009
- Building block is a buildpack, which enables containerized 12-factor applications
  - The process for building the container is opaque, but:
  - Deploying new version of an app is just: `git push heroku`



Open Source IaaS: OpenStack (2010)

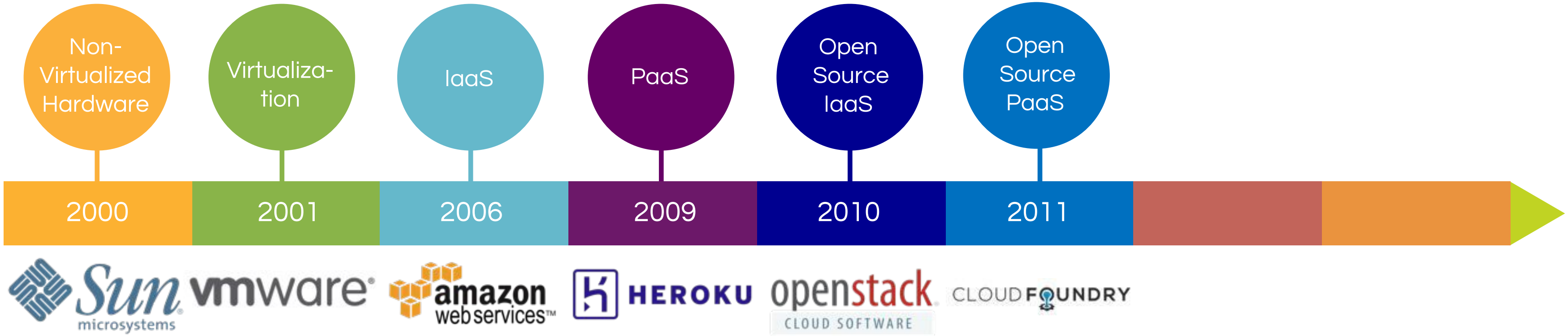
- OpenStack brings together an extraordinarily diverse group of vendors to create an open source Infrastructure-as-a-Service (IaaS)
- Competes with AWS and VMWare
- Building block remains a VM



Open Source PaaS: Cloud Foundry (2011)

CLOUD FOUNDRY

- Pivotal builds an open source alternative to Heroku's PaaS and launches the Cloud Foundry Foundation in late 2014
- Building block is Garden containers, which can hold Heroku buildpacks, Docker containers and even non-Linux OSes



Non-Virtualized Hardware — 2000 — Sun microsystems

Virtualization — 2001 — vmware

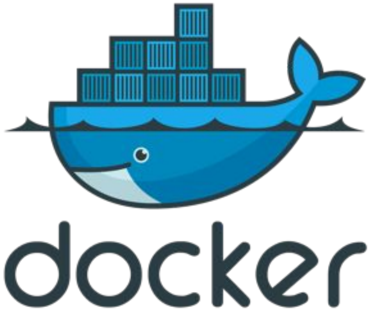
IaaS — 2006 — amazon webservices™

PaaS — 2009 — HEROKU

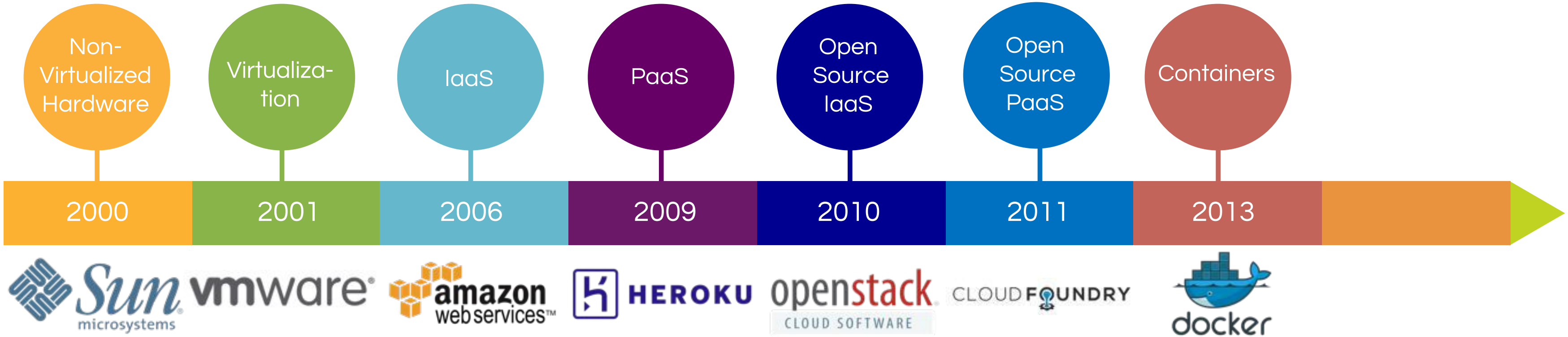
Open Source IaaS — 2010 — openstack CLOUD SOFTWARE

Open Source PaaS — 2011 — CLOUD FOUNDRY

Containers: Docker (2013)



- Docker combines LXC, Union File System and cgroups to create a containerization standard adopted by millions of developers around the world
- Fastest uptake of a developer technology ever
- Enables isolation, reuse and immutability


Non-Virtualized Hardware
2000

Virtualization
2001

IaaS
2006

PaaS
2009

Open Source IaaS
2010

Open Source PaaS
2011

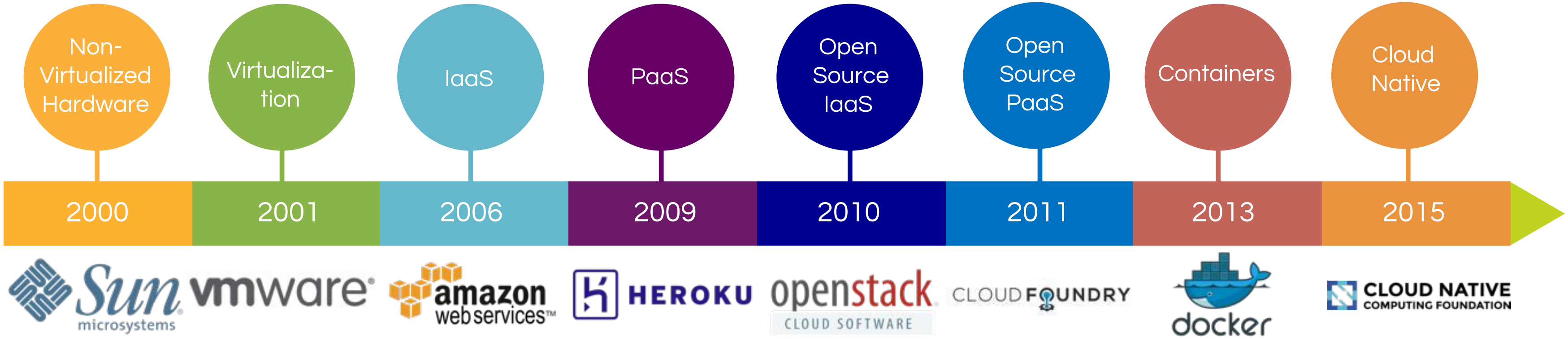
Containers
2013

CNCF and OCI (2015)

CLOUD NATIVE
COMPUTING FOUNDATION

OPEN CONTAINER INITIATIVE

- Cloud native computing uses an open source software stack to:
  - deploy applications as *microservices*,
  - packaging each part into its own *container*
  - and dynamically *orchestrating* those containers to optimize resource utilization
- Standardization: https://www.opencontainers.org/



2000 – Non-Virtualized Hardware
2001 – Virtualiza-tion
2006 – IaaS
2009 – PaaS
2010 – Open Source IaaS
2011 – Open Source PaaS
2013 – Containers
2015 – Cloud Native

So… What Have We Learned?

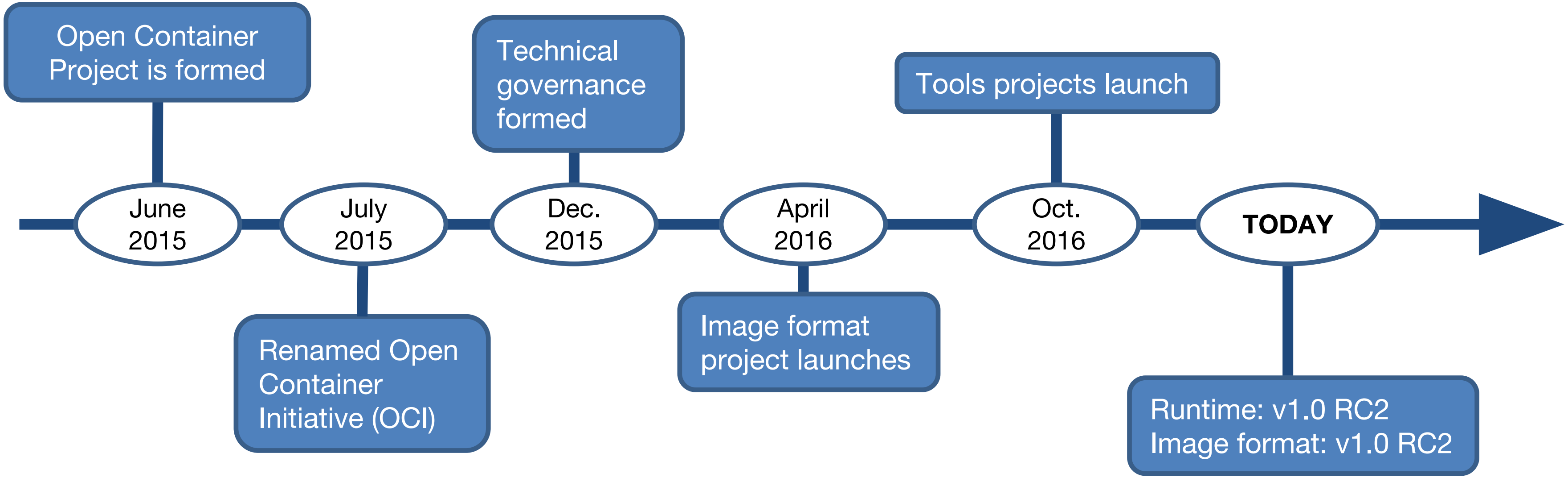
- Core Building Block:
  - Servers ➡ Virtual Machines ➡ Buildpacks ➡ Containers
- Isolation Units
  - From heavier to lighter weight, in spin-up time and size
- Immutability
  - From pets to cattle
- Provider
  - From closed source, single vendor to open source, cross-vendor

CLOUD NATIVE
COMPUTING
FOUNDATION

OCI + CNCF in Detail

Open Container Initiative (OCI)

- Founded in June 2015: https://www.opencontainers.org/
- Mission: Develop and promote a set of common, minimal, open standards and specifications around container technology (backed by a certification program)



June 2015 – Open Container Project is formed

July 2015 – Renamed Open Container Initiative (OCI)

Dec. 2015 – Technical governance formed

April 2016 – Image format project launches

Oct. 2016 – Tools projects launch

TODAY – Runtime: v1.0 RC2
Image format: v1.0 RC2

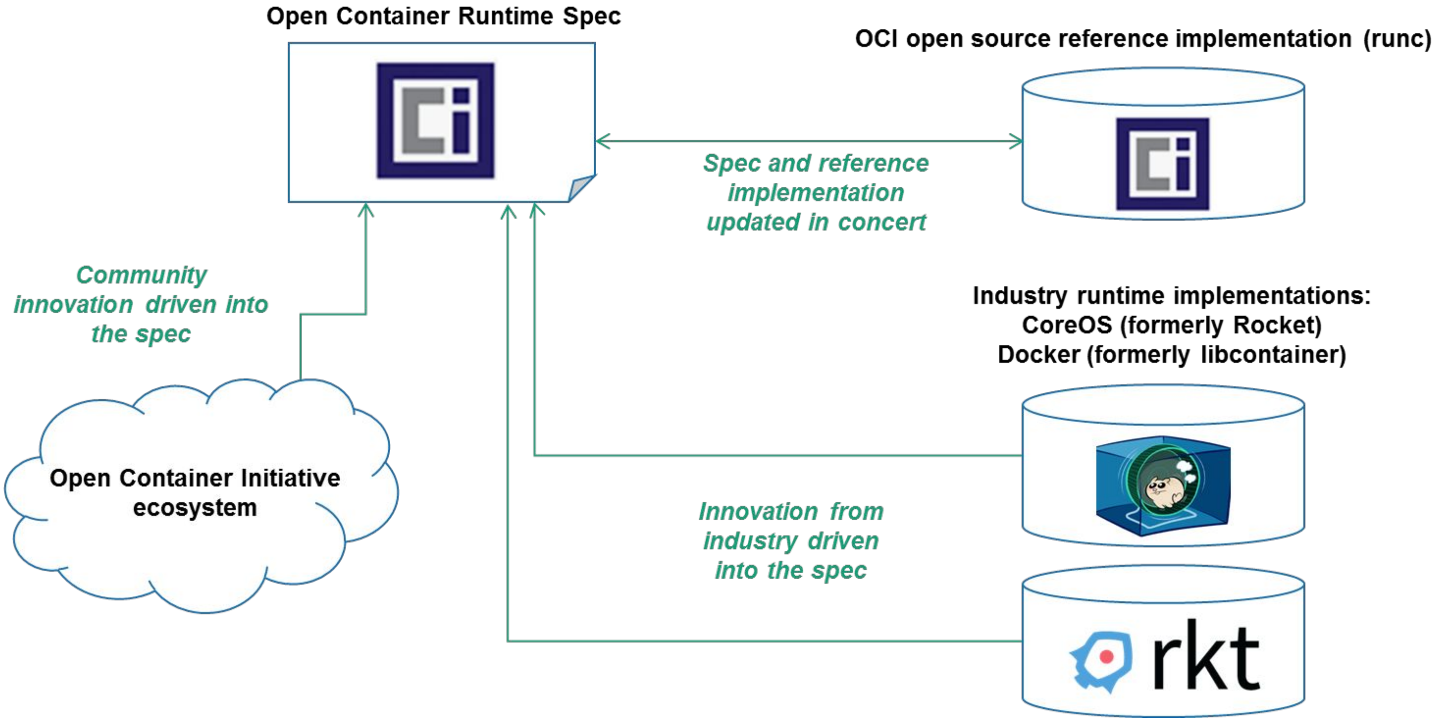
CLOUD NATIVE COMPUTING FOUNDATION

OCI Projects

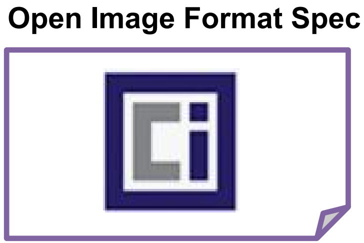
- Runtime spec: a spec for managing the container runtime
- Runtime tools: tools for testing container runtimes
- Runc: runs containers (implementation of runtime-spec)

- Image spec: a container image format spec
- Image tools: tools for testing of container images implementing the OCI image specification

OCI Projects

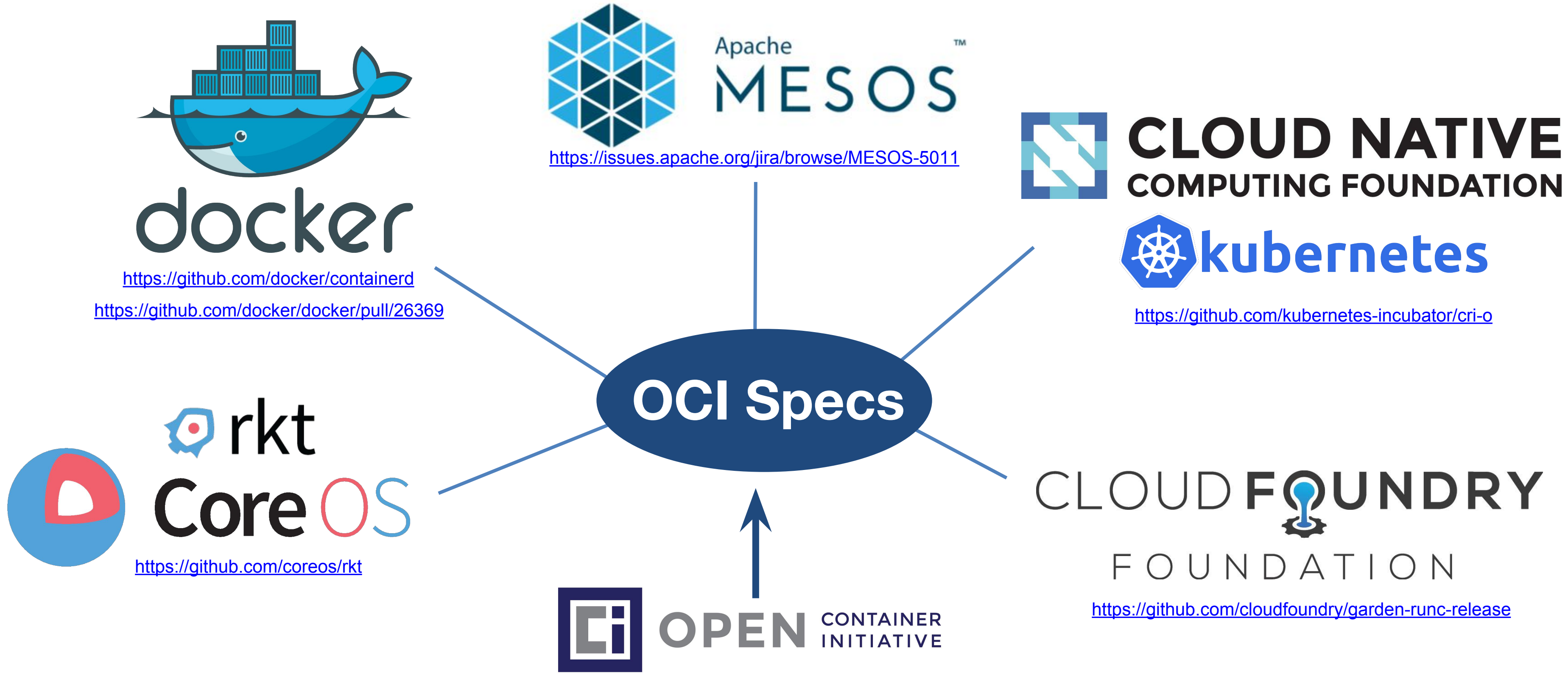


Open Container Runtime Spec
OCI open source reference implementation (runc)
Spec and reference implementation updated in concert
Community innovation driven into the spec
Open Container Initiative ecosystem
Industry runtime implementations: CoreOS (formerly Rocket) Docker (formerly libcontainer)
Innovation from industry driven into the spec
rkt



- Open Specification for *Container Image*
- Started with Docker v2.2
- Announced April 14, 2016

OCI Adopters



https://github.com/docker/containerd

https://github.com/docker/docker/pull/26369

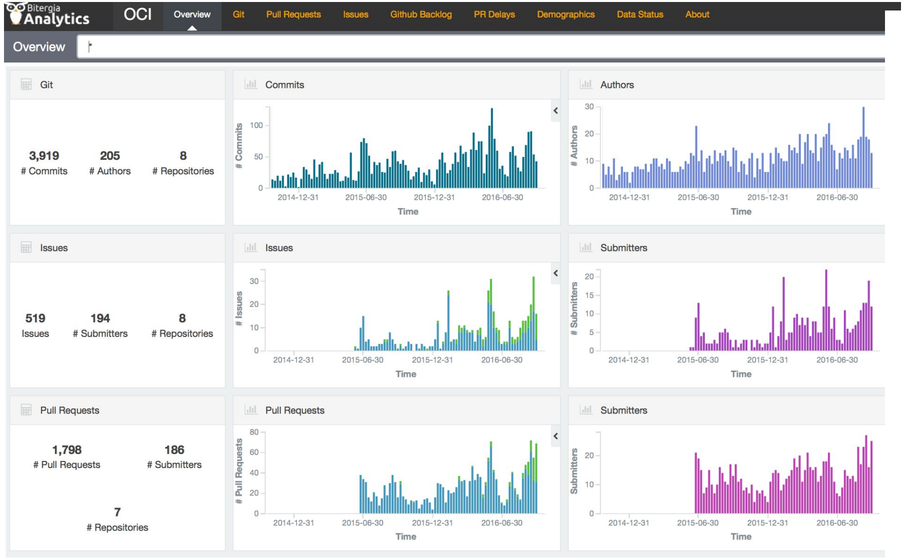
https://issues.apache.org/jira/browse/MESOS-5011

https://github.com/kubernetes-incubator/cri-o

https://github.com/coreos/rkt

https://github.com/cloudfoundry/garden-runc-release

OCI Contributors



Git

3,919 # Commits | 205 # Authors | 8 # Repositories

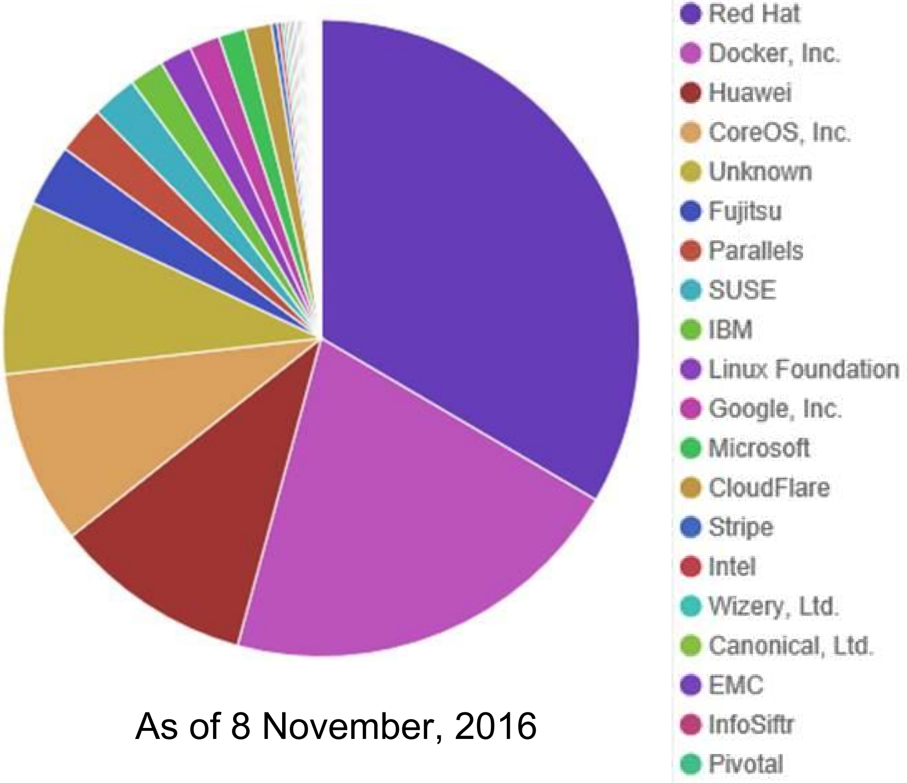
Issues

519 Issues | 194 # Submitters | 8 # Repositories

Pull Requests

1,798 # Pull Requests | 186 # Submitters | 7 # Repositories

- The top 15 groups contributing to the OCI represent a broad and diverse group of companies
- View the OCI dashboard: http://oci.biterg.io/



As of 8 November, 2016

Cloud Native Computing Foundation (CNCF)

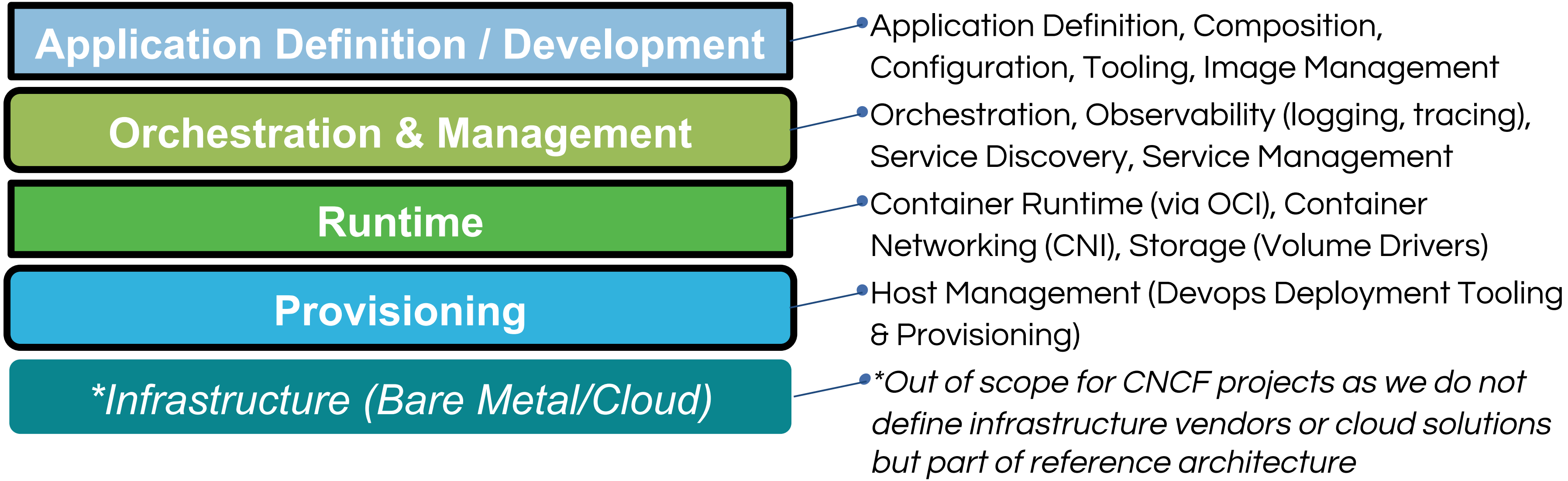
- Founded December 2015: https://www.cncf.io/
- Non-profit, part of the Linux Foundation
- Initial projects are Kubernetes, donated by Google, and Prometheus, originally from SoundCloud
- Platinum members:

CISCO CoreOS docker FUJITSU Google HUAWEI

IBM intel Joyent MESOSPHERE redhat SUPERNAP A Switch Company

- Plus 40 additional members

Cloud Native [End User] Reference Architecture



Application Definition / Development

- Application Definition, Composition, Configuration, Tooling, Image Management

Orchestration & Management

- Orchestration, Observability (logging, tracing), Service Discovery, Service Management

Runtime

- Container Runtime (via OCI), Container Networking (CNI), Storage (Volume Drivers)

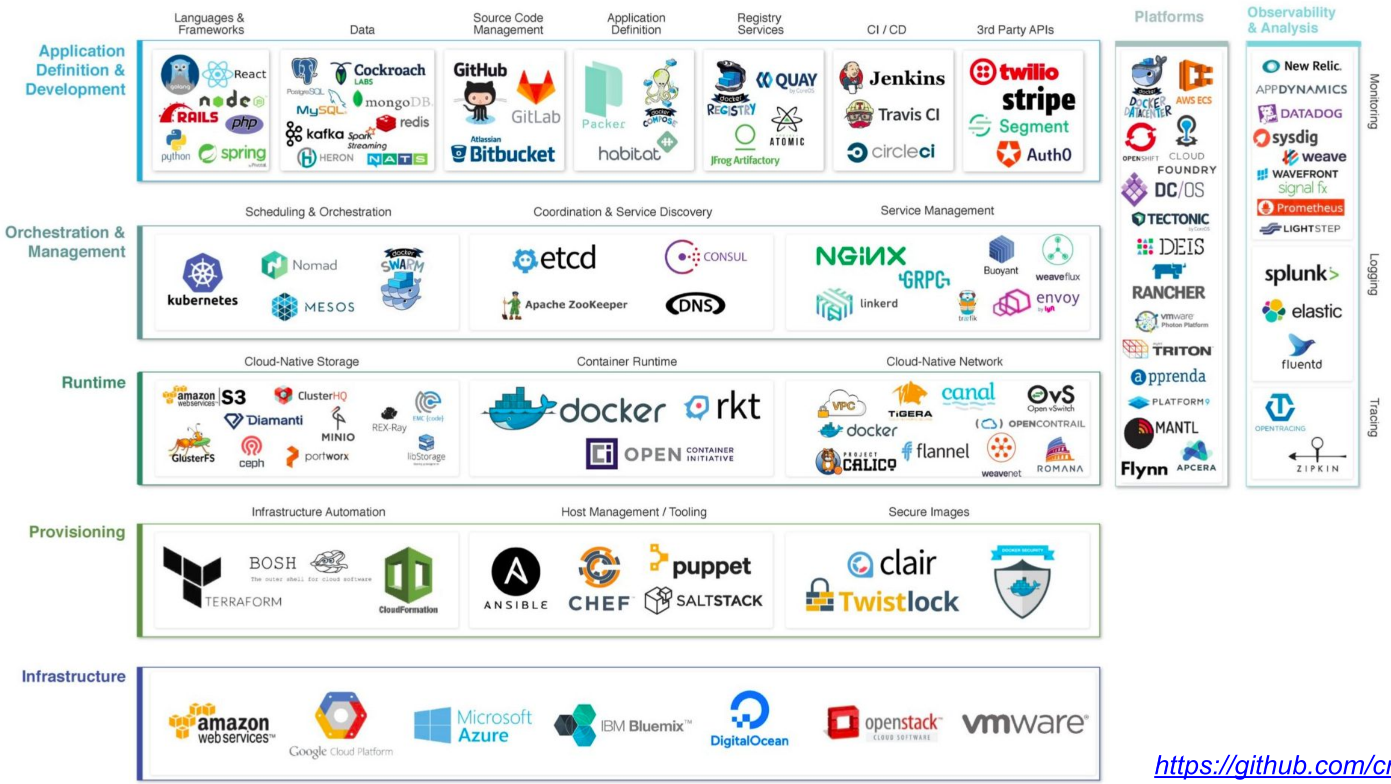
Provisioning

- Host Management (Devops Deployment Tooling & Provisioning)

**Infrastructure (Bare Metal/Cloud)*

- **Out of scope for CNCF projects as we do not define infrastructure vendors or cloud solutions but part of reference architecture*

Cloud Native Landscape (github.com/cncf/landscape)



Application Definition & Development

Languages & Frameworks · Data · Source Code Management · Application Definition · Registry Services · CI / CD · 3rd Party APIs

Orchestration & Management

Scheduling & Orchestration · Coordination & Service Discovery · Service Management

Runtime

Cloud-Native Storage · Container Runtime · Cloud-Native Network

Provisioning

Infrastructure Automation · Host Management / Tooling · Secure Images

Infrastructure

Platforms

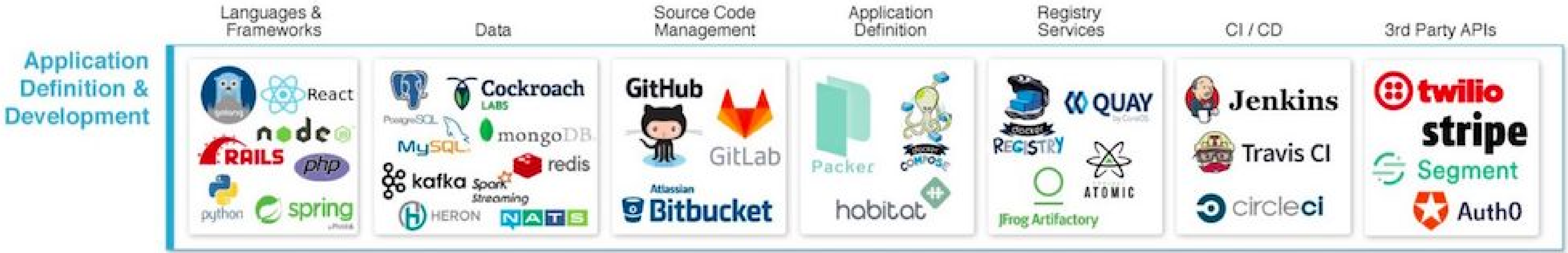
Observability & Analysis

Monitoring · Logging · Tracing

https://github.com/cncf/landscape

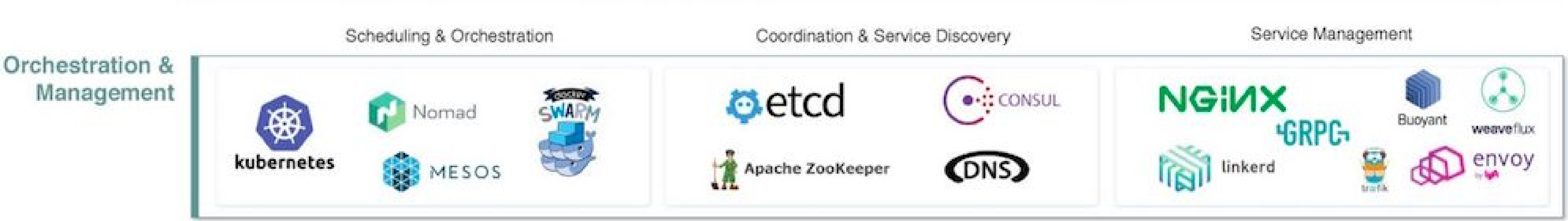
CLOUD NATIVE COMPUTING FOUNDATION

Cloud Native Landscape: App Definition + Development



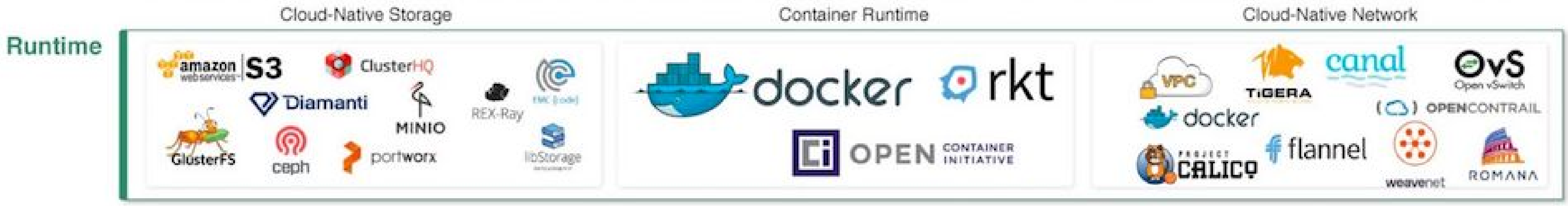
- Includes Languages, Frameworks, Data, SCM, App Definition, Registry Services, CI/CD

Cloud Native Landscape: Orchestration + Management



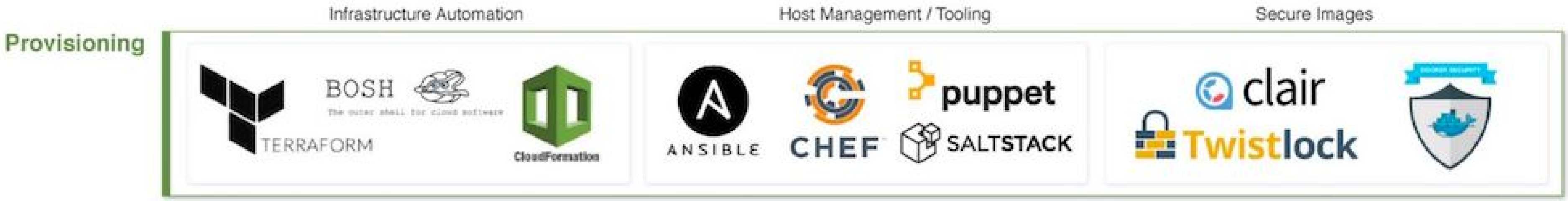
- Orchestration: Kubernetes, Mesos, Swarm, Nomad
- Service Discovery: etcd, Consult, ZK, CoreDNS
- Service Management: linkerd, gRPC, envoy

Cloud Native Landscape: Runtime



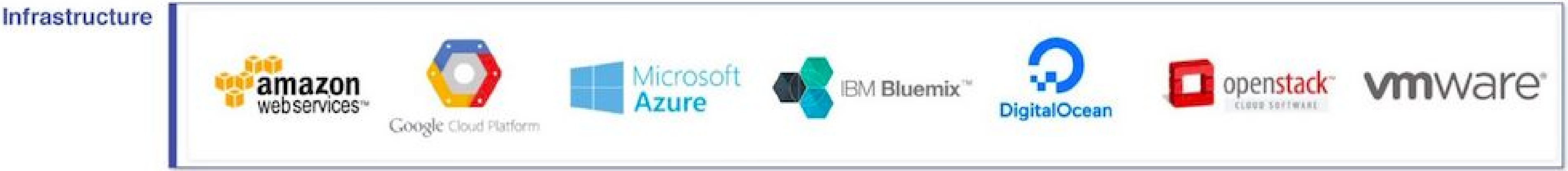
- Storage: Minio, ClusterHQ, ceph, GlusterFS
- Container Runtime: OCI, Docker, Rkt
- Networking: Canal, CNI, weavenet, libnetwork

Cloud Native Landscape: Provisioning



- Infra Automation: Terraform, CloudFormation
- Host Management: Ansible, Chef, Puppet, Salt
- Secure Image: Clair, Twistlock

Cloud Native Landscape: Infrastructure



- AWS, GCP, Azure, Bluemix, DigitalOcean, Openstack, etc

- *Note: **OUT OF SCOPE** for CNCF projects*

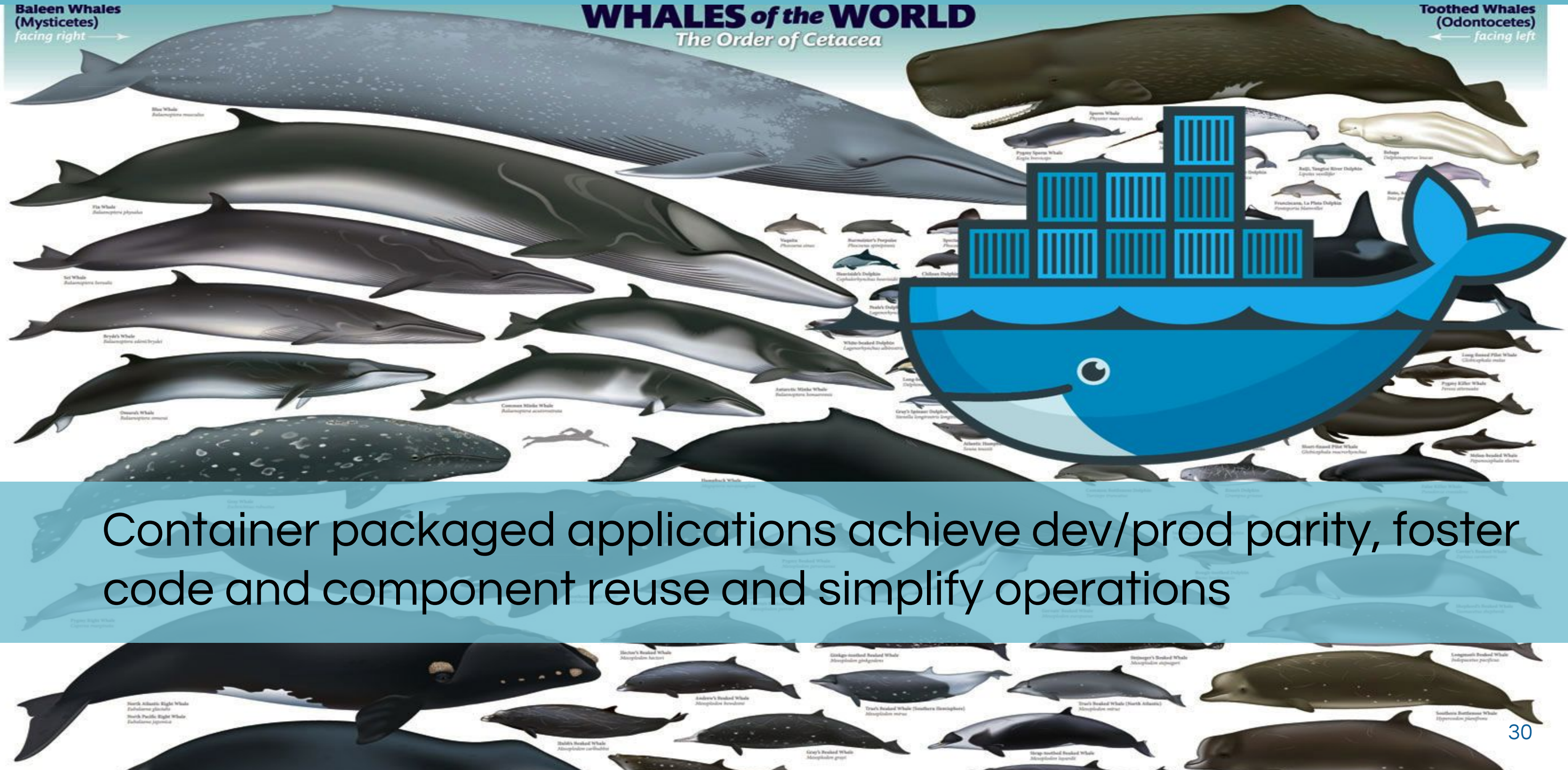
CNCF Potential Projects and Community

- Potential future project areas:
  - Logging (Fluentd): http://www.fluentd.org/
  - Networking (CNI/Flannel/Calico/Weave): https://github.com/containernetworking/cni
  - Messaging (NATS): http://nats.io/
  - Configuration (etcd): https://github.com/coreos/etcd
  - Storage (Minio): https://github.com/minio/
  - RPC (GRPC): http://www.grpc.io/
  - Tracing (OpenTracing, OpenZipkin): http://opentracing.io/
  - Streaming (Heron): http://heronstreaming.io
  - ...and more! https://github.com/cncf/toc#scheduled-community-presentations

CLOUD NATIVE
COMPUTING
FOUNDATION

Cloud Native Value Propositions

Isolation



Container packaged applications achieve dev/prod parity, foster code and component reuse and simplify operations

No Lock-in

Open source software stack enables deployment on any public or private cloud (or in combinations)

Unlimited Scalability

Optimized for modern distributed systems environments capable of scaling to tens of thousands of self healing multi-tenant nodes
(e.g., Google starts 2 billion containers per week)

Improved Efficiency and Resource Utilization

Via a central orchestrating process that dynamically manages and schedules microservices. This reduces the costs associated with maintenance and operations.

Resiliency

To failures of individual containers, machines, and even data centers and to varying levels of demand

CLOUD NATIVE
COMPUTING
FOUNDATION

Hosting with the CNCF?

Software Foundations in a Post-GitHub World

- No one is impressed today by a software repo, mailing list, or website
- Foundations need to offer a different set of services
- CNCF’s goal is to be the best place to host cloud native software projects

Why You Should Host Your Project at CNCF

- Neutral home increases contributions
- Endorsement by CNCF’s Technical Oversight Committee
- Priority access to $15 million, 1000 node Community Cluster
- Engagement with End User Board
- Full-time press relation and analyst relation teams
- $20 K per year to improve your project documentation
- Maintain your committers; just agree to unbiased process
- Full-time staff eager to assist
- World-class events team, track at CloudNativeCon/KubeCon around the world, and custom events for your project
- Worldwide meetup groups and Cloud Native Roadshows
- Inclusion in the CNCF marketing demo

CLOUD NATIVE
COMPUTING
FOUNDATION

Why You Should Join the CNCF and OCI?

Help Set the Direction of Cloud Native and Containers!

CLOUD NATIVE
COMPUTING FOUNDATION

OPEN CONTAINER INITIATIVE

- Participate in our hosted projects and attend our events and roadshows!
- Design your applications and services to work with a cloud native platform of orchestrated containers of microservices
- Become a member of the Cloud Native Computing Foundation (CNCF): https://cncf.io/join
- Become a member of the Open Container Initiative (OCI): https://opencontainers.org/join
- Contact: cra@linuxfoundation.org

CLOUD NATIVE
COMPUTING
FOUNDATION

Thank you! Q&A?

@cra

CLOUD NATIVE
COMPUTING
FOUNDATION

Extra Slides

CNCF Members

cisco

CoreOS

docker

FUJITSU

Google

HUAWEI

IBM

intel

Joyent

MESOSPHERE

redhat

SUPERNAP
A Switch Company

ebay

APCERA

AppFormix

apprenda

aqua

NetApp

AVI Networks

Centrify

超[illegible]云

cloudsoft

SEL
Software Engineering Lab
Zhejiang University

ClusterHQ
The Container Data People

Container Solutions

CRUNCHY
Enterprise PostgreSQL

DEIS

Goldman Sachs

Diamanti

eldarion

iguaz.io
VOLUME. VELOCITY. VARIETY

Infoblox
CONTROL YOUR NETWORK

WILLIAMS GARCIA

ishi systems

PROJECT CALICO
SUPPORTED BY METASWITCH

NEC

NGINX

portworx

SUPERGIANT

RANCHER

RX-M

weaveworks

STORAGEOS

sysdig

TREASURE DATA

Twistlock

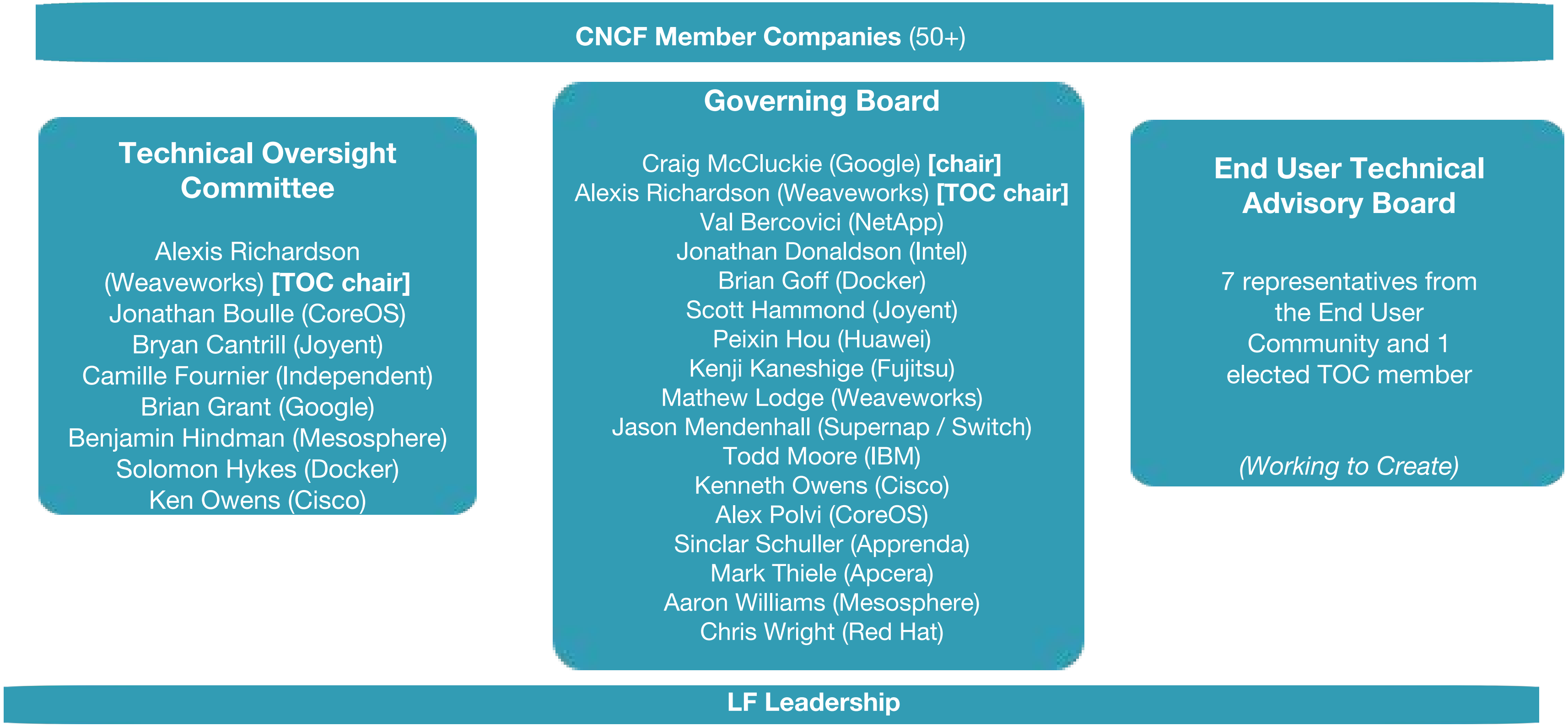
NCSOFT

UNIVA

Virtuozzo

vmware

CNCF Governance Structure



CNCF Member Companies (50+)

Technical Oversight Committee

Alexis Richardson (Weaveworks) **[TOC chair]**
Jonathan Boulle (CoreOS)
Bryan Cantrill (Joyent)
Camille Fournier (Independent)
Brian Grant (Google)
Benjamin Hindman (Mesosphere)
Solomon Hykes (Docker)
Ken Owens (Cisco)

Governing Board

Craig McCluckie (Google) **[chair]**
Alexis Richardson (Weaveworks) **[TOC chair]**
Val Bercovici (NetApp)
Jonathan Donaldson (Intel)
Brian Goff (Docker)
Scott Hammond (Joyent)
Peixin Hou (Huawei)
Kenji Kaneshige (Fujitsu)
Mathew Lodge (Weaveworks)
Jason Mendenhall (Supernap / Switch)
Todd Moore (IBM)
Kenneth Owens (Cisco)
Alex Polvi (CoreOS)
Sinclar Schuller (Apprenda)
Mark Thiele (Apcera)
Aaron Williams (Mesosphere)
Chris Wright (Red Hat)

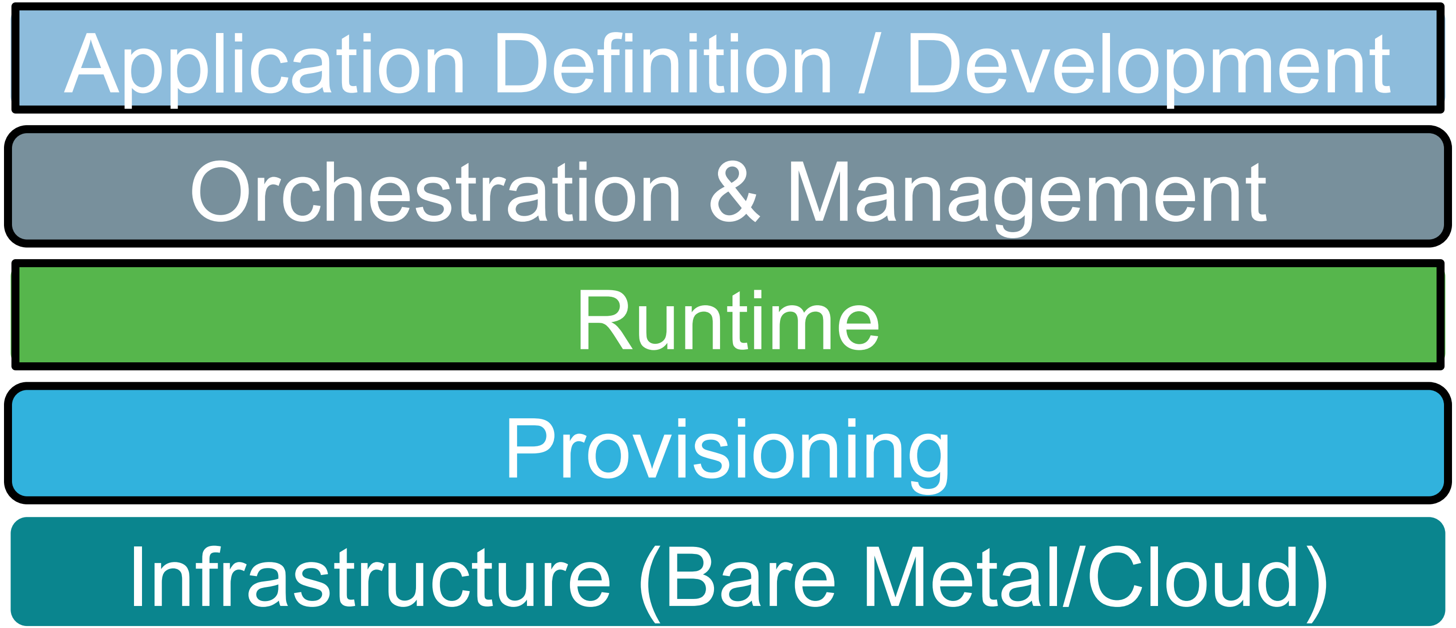
End User Technical Advisory Board

7 representatives from the End User Community and 1 elected TOC member

(Working to Create)

LF Leadership

Cloud Native Reference Architecture



Application Definition / Development

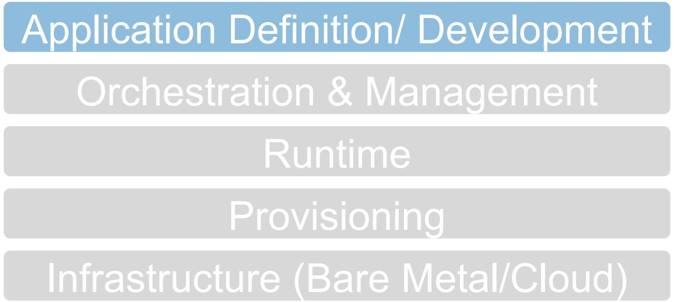
Orchestration & Management

Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

Application Definition/ Deployment Layer



Application Definition/ Development

Orchestration & Management

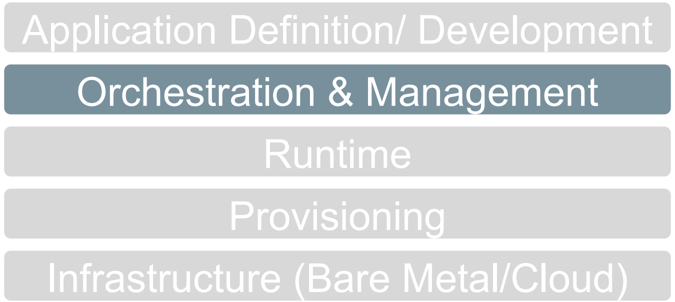
Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

- Application Definition, Composition, configuration, and reuse
- Development Frameworks
- Tooling
- CI/CD
- Image Management (Registry, governance, policy)

Orchestration & Management Layer



Application Definition/ Development

Orchestration & Management

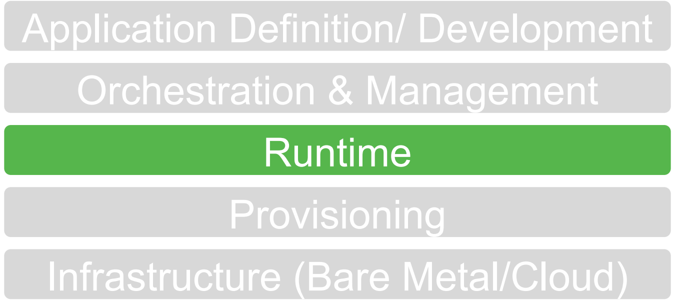
Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

- Observability
  - View / Filter / Replay
  - Monitoring / Trace / Stream / Log
  - Business Intelligence
- Orchestration and scheduling
- Name resolution and service discovery (e.g., DNS)
- Service Management
  - Routing / Proxy / Load Balancer
  - Policy / Placement / Traffic Management

Runtime Layer



Application Definition/ Development

Orchestration & Management

Runtime

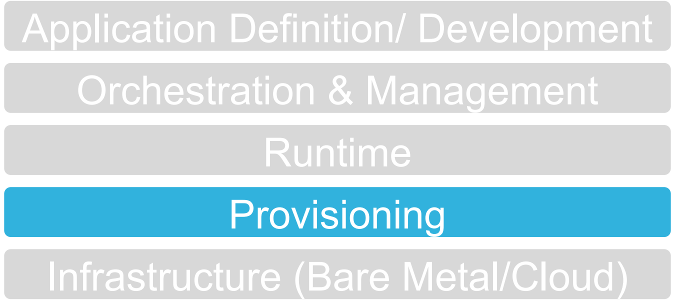
Provisioning

Infrastructure (Bare Metal/Cloud)

Note: Container runtime and format are adopted from OCI

- Resource Management
  - Image Management
  - Container Management
  - Compute Resources
- Cloud Native – Network
  - Network Segmentation and Policy
  - SDN & APIs (e.g., CNI, libnetwork)
- Cloud Native- Storage
  - Volume Drivers/Plugins
  - Local Storage Management
  - Remote Storage Access

Provisioning Layer



Application Definition/ Development

Orchestration & Management

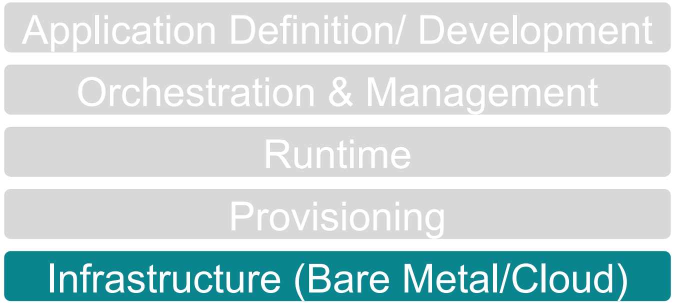
Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

- Host Management
- Secure OS Images
- Host level Devops Deployment Tooling & Provisioning
- Infrastructure Automation
  - Compute
  - Network
  - Storage

Infrastructure (Bare Metal/Cloud) Layer



Application Definition/ Development

Orchestration & Management

Runtime

Provisioning

Infrastructure (Bare Metal/Cloud)

- Out of scope for CNCF projects as we do not define infrastructure vendors or cloud solutions but part of reference architecture
- Potentially in the future we will provide “certification”