CNFs vs. VNFs

Dan Kohn
Executive Director, CNCF

Cloud Native Computing Foundation

- Non-profit, part of the Linux Foundation; founded Dec 2015

Graduated


kubernetes
Orchestration


Prometheus
Monitoring

Incubating


OPENTRACING
Distributed Tracing API


fluentd
Logging


GRPC
Remote Procedure Call


container d
Container Runtime


rkt
Container Runtime


CNI
Networking API


envoy
Service Mesh


JAEGER
Distributed Tracing


TUF
Software Update Spec


notary
Security


Vitess
Storage


CoreDNS
Service Discovery


NATS
Messaging


LINKERD
Service Mesh


HELM
Package Management

Sandbox


ROOK
Storage


spiffe
Identity Spec


SPIRE
Identity


Open Policy Agent
Policy


cloudevents
Serverless


TELEPRESENCE
Tooling


HARBOR
Registry


OPENMETRICS
Metrics Spec


KV
Key/Value Store

- Platinum members:


Alibaba Cloud


aws


Azure


cisco


DELL Technologies


docker


FUJITSU


Google Cloud


HUAWEI


IBM Cloud


intel


JD.COM


MESOSPHERE


ORACLE

Pivotal

redhat

SAMSUNG
SAMSUNG SDS

SAP

vmware

TODAY THE LINUX FOUNDATION IS MUCH MORE THAN LINUX

Security

We are helping global privacy and security through a program to encrypt the entire internet.

Networking

We are creating ecosystems around networking to improve agility in the evolving software-defined datacenter.

Cloud

We are creating a portability layer for the cloud, driving de facto standards and developing the orchestration layer for all clouds.

Automotive

We are creating the platform for infotainment in the auto industry that can be expanded into instrument clusters and telematics systems.

Blockchain

We are creating a permanent, secure distributed ledger that makes it easier to create cost-efficient, decentralized business networks.

Web

We are providing the application development framework for next generation web, mobile, serverless, and IoT applications.

Let's Encrypt

LF NETWORKING

CLOUD NATIVE COMPUTING FOUNDATION

AUTOMOTIVE GRADE LINUX

HYPERLEDGER

node JS

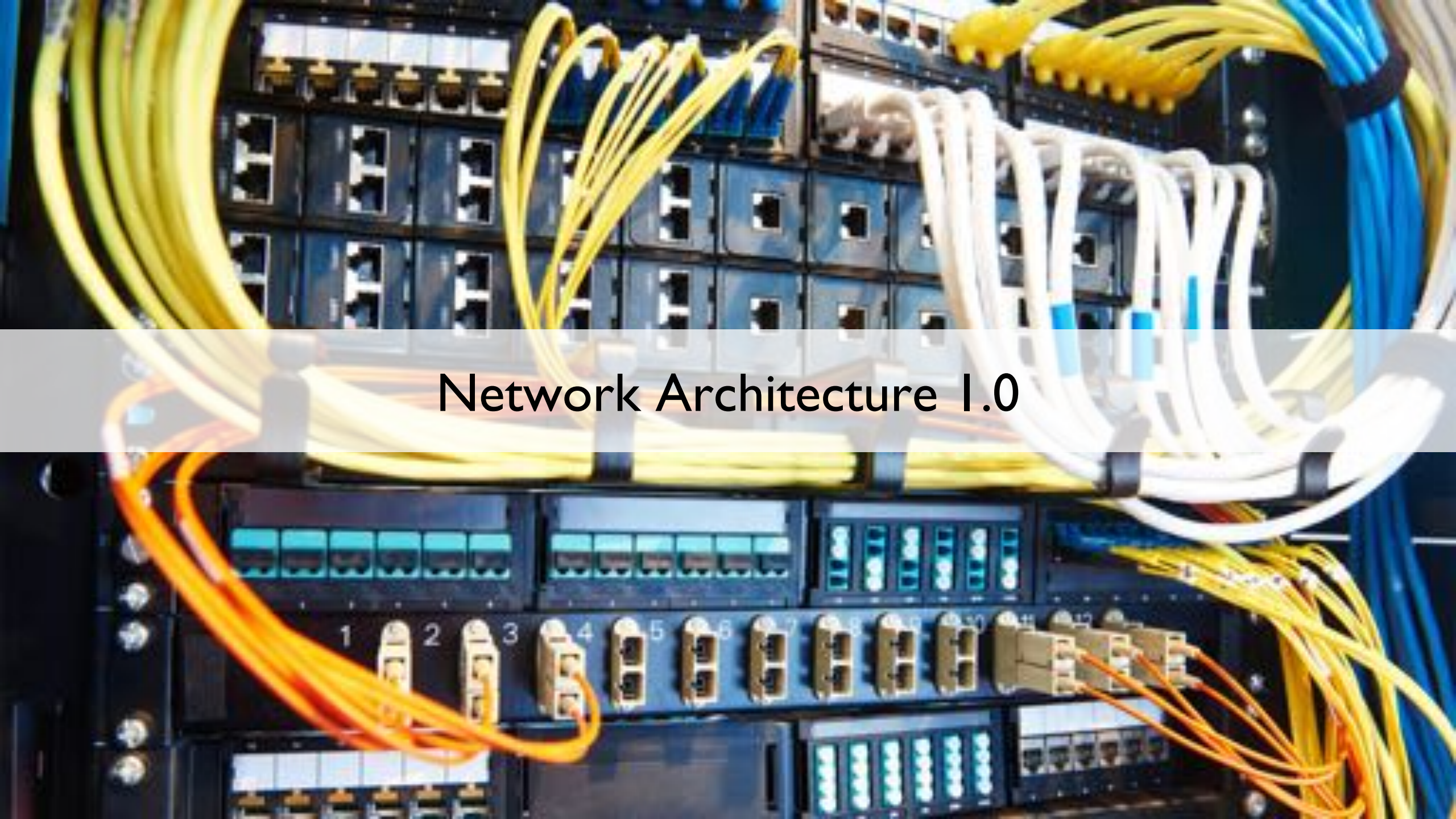
We are regularly adding projects; for the most up-to-date listing of all projects visit tlfprojects.org

KubeCon + CloudNativeCon

- China
  - Shanghai: November 14-15, 2018
  - Sponsorships open
- North America
  - Seattle: December 11 - 13, 2018
  - Sponsorships open
- Europe
  - Barcelona: May 21 - 23, 2019

Network Architecture Evolution

- 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack
- 3.0: Cloud-native Network Functions (CNFs) running on Kubernetes on public, private, or hybrid clouds



Network Architecture 1.0

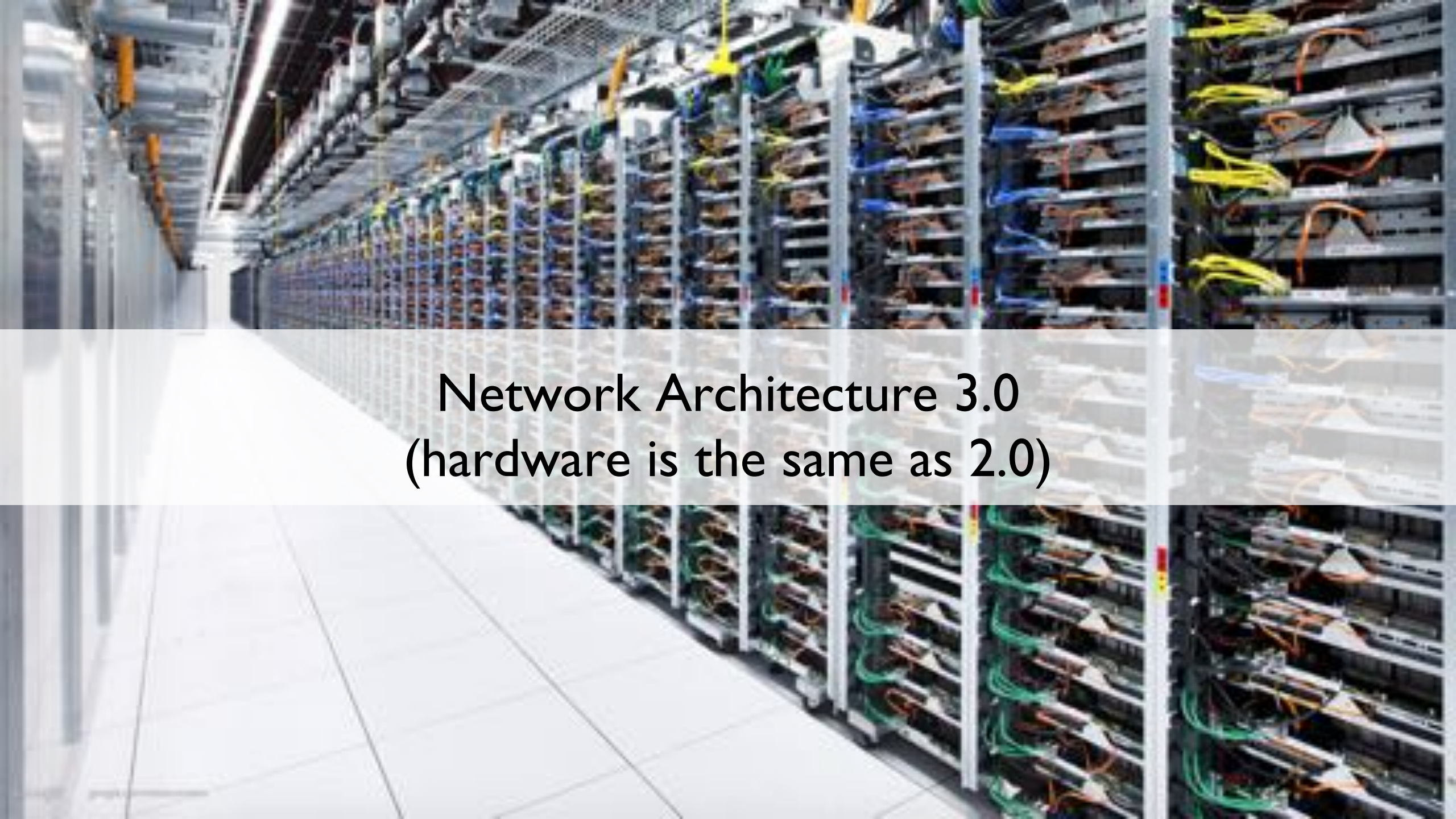
Network Architecture Evolution

- 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack

Network Architecture 2.0

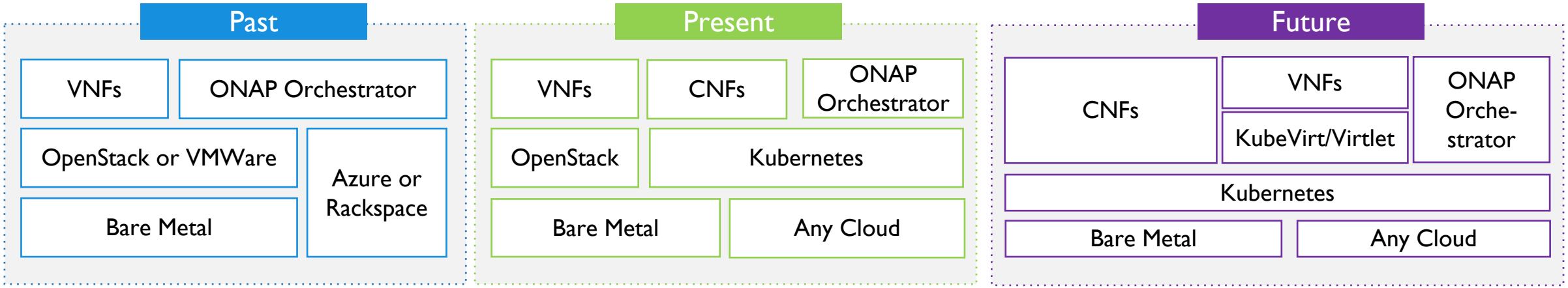
Network Architecture Evolution

- 1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)
- 2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack
- 3.0: Cloud-native Network Functions (CNFs) running on Kubernetes on public, private, or hybrid clouds



Network Architecture 3.0
(hardware is the same as 2.0)

Evolving from VNFs to CNFs



- ONAP Amsterdam (Past) runs on OpenStack, VMware, Azure or Rackspace
- ONAP Casablanca (Present) runs on Kubernetes and so works on any public, private or hybrid cloud
- Virtual Network Functions (VNFs) are virtual machines that run on OpenStack or VMWare, or can be run on K8s via KubeVirt or Virtlet

Major Benefits

1. Cost savings (with public, private, and hybrid clouds)
2. Development velocity
3. Resiliency (to failures of individual CNFs, machines, and even data centers)

The challenge of transitioning VNFs to CNFs

- Moving from network functionality from *physical* hardware to encapsulating the software in a *virtual* machine (P2V) is generally easier than *containerizing* the software (P2C or V2C)
- Many network function virtualization VMs rely on kernel hacks or otherwise do not restrict themselves to just the stable Linux kernel userspace ABI
  - They also often need to use DPDK or SR-IOV to achieve sufficient performance
- Containers provide nearly direct access to the hardware with little or no virtualization overhead
  - But they expect containerized applications to use the stable userspace Linux kernel ABI, not to bypass it

Areas for More Discussion

- The strength of no longer being locked into specific OSs
  - Any version of Linux >3.10 is acceptable
- Multi-interface pods vs. Network Service Mesh
- Complete parity for IPv6 functionality and dual-stack support in K8s
- Security, and specifically recommendations from Google and Jess that come into play when hosting untrusted, user-provided code
  - Possible use of isolation layers such as gVisor or Kata
- Scheduling container workloads with network-related hardware constraints (similar to what's been done for GPUs)
  - Network-specific functionality like QOS

Demo Plans Underway

- VNFs vs. CNFs
  - Working on a demo of boot-time and throughput of VNFs on OpenStack vs. CNFs on Kubernetes, where the networking code and underlying hardware is identical
  - Will deliver opens source installers and Helm charts
- Cloud-native Customer Premises Equipment (CCPE) Project
  - Modify the ONAP vCPE use case and VNF deployment to show VNF vs. CNF deployments of chained network functions

Roll-Out Plans

- Open Source Summit NA, Vancouver, August 28: Joint workshop by CNCF executive director Dan Kohn and LF Networking head Arpit Joshipura on Cloud-native Network Functions
- Open Network Summit Europe, Amsterdam, September 25: Marketing launch
- KubeCon + CloudNativeCon NA, Seattle, December 11: Planned demo
- Mobile World Congress, Barcelona, February 25: Major roll-out
- Ongoing close collaboration with LF Networking and specific carriers providing feedback (AT&T, Bell Canada, Vodafone, etc.)

The Networking aspects of Cloud Native

Arpit Joshipura
GM Networking
The Linux Foundation

Industry Direction: Any Cloud + Portable Apps in Containers

- Utilize the best of Cloud with the best of Telecom Networks
  - Promise of Containers – allow for any cloud portability
  - Promise of Network – full network automation & zero touch services
- Telecom Network Transformation require a hybrid strategy
  - Migration of VM to containers step by step approach to VNF/Workloads
  - VM and Container Interworking in a Multi-VIM environment

Two leading de-facto platforms –Networking & Cloud

LF NETWORKING

ONAP

&

CLOUD NATIVE COMPUTING FOUNDATION

kubernetes

- **Network Automation & Orchestration Platform**
- **ONAP has a multi-VIM strategy (Openstack, Vmware, Azure,..)**
- **Project within ONAP – OOM looking at Containers**

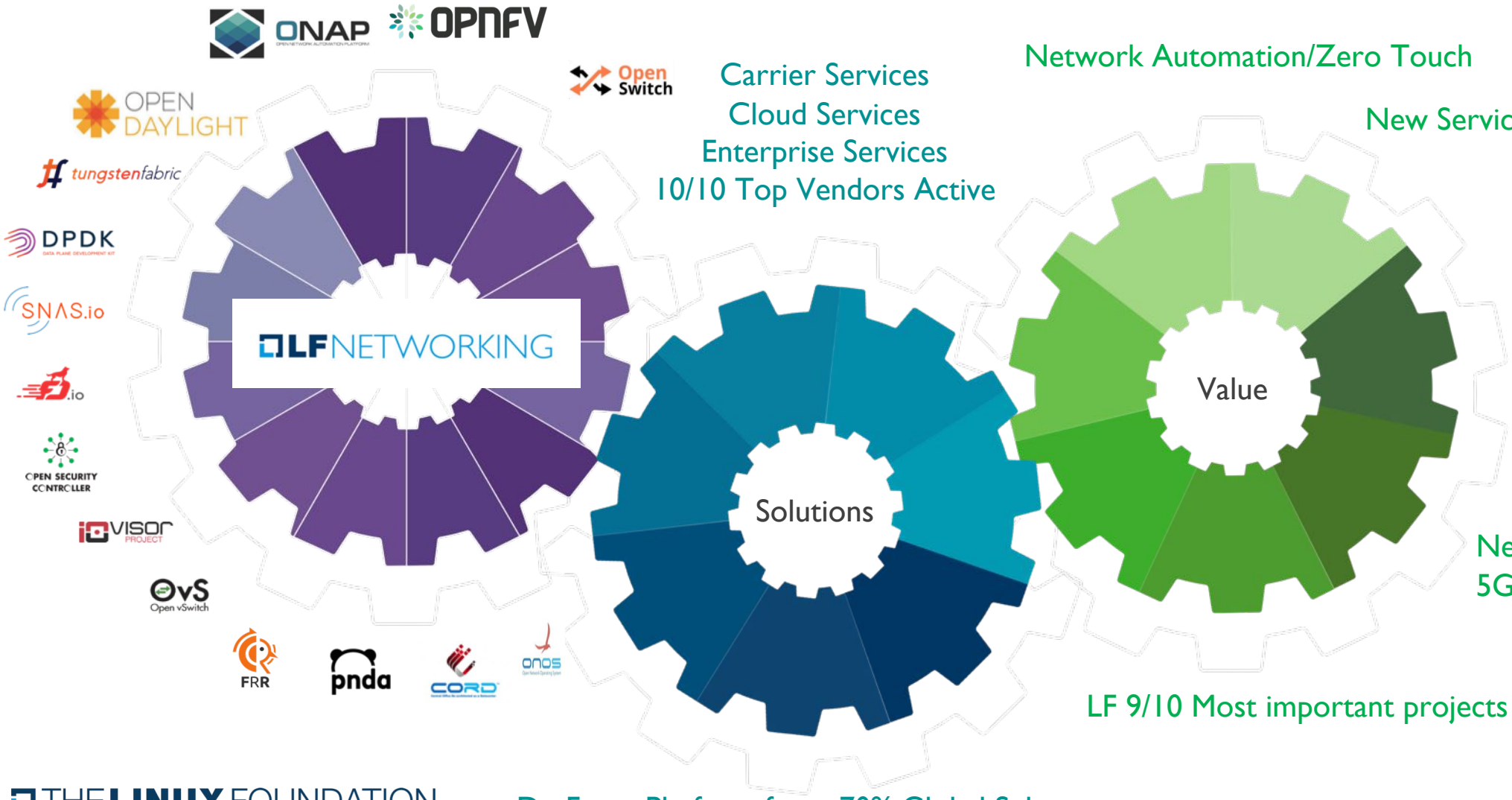
- **Cloud Automation & Orchestration Platform**
- **Project within CNCF looking at ONAP – Cross Cloud CI**

Open Source projects at LF can bring the best of both worlds to the Telecom Industry

THE LINUX FOUNDATION

Sustainable Innovation: Open Source Networking

Creating *De-Facto* Platforms to Enable Next Generation Solutions in Telecom, Enterprise & Cloud



Carrier Services
Cloud Services
Enterprise Services
10/10 Top Vendors Active

Network Automation/Zero Touch

New Services in Minutes

$576M Shared Innovation

SDO+OSS Harmonization

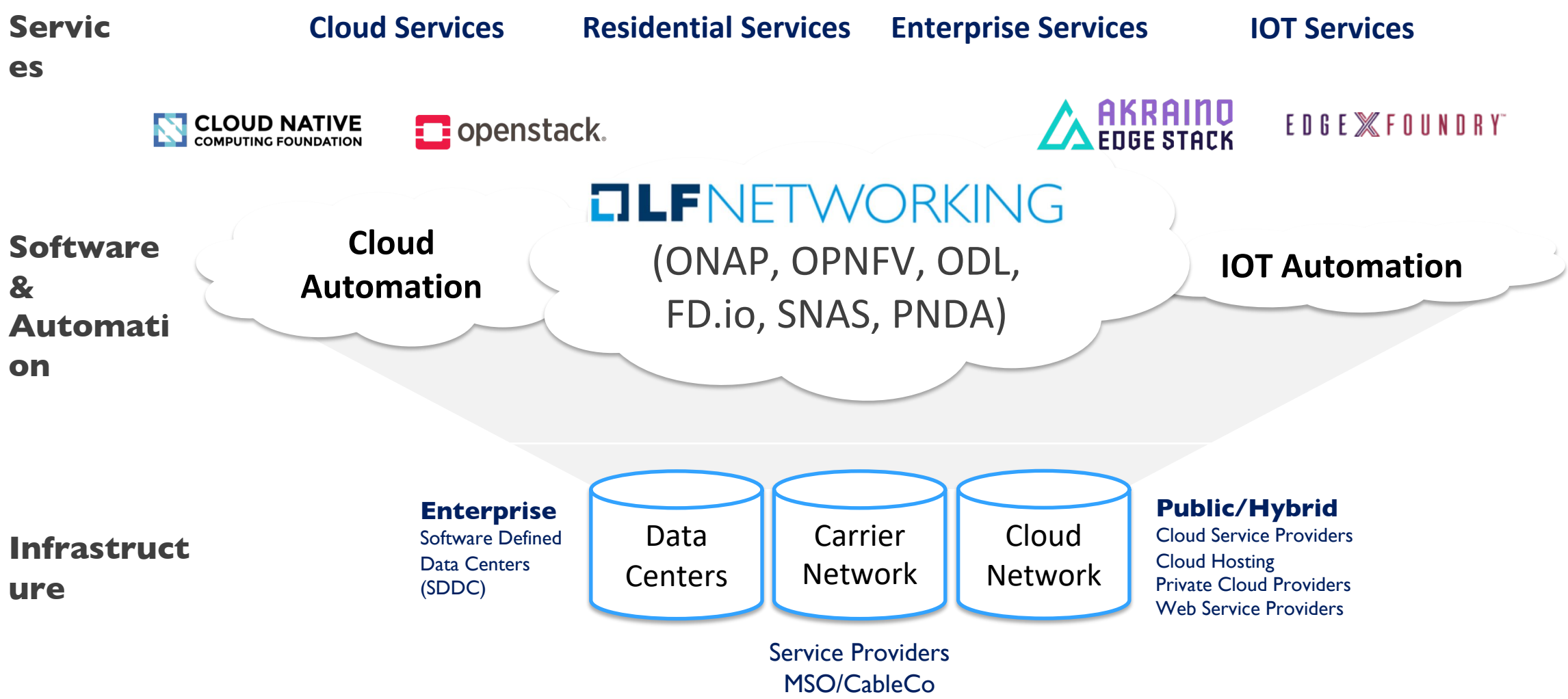
New Services 5G/IOT/Edge/AI

LF 9/10 Most important projects

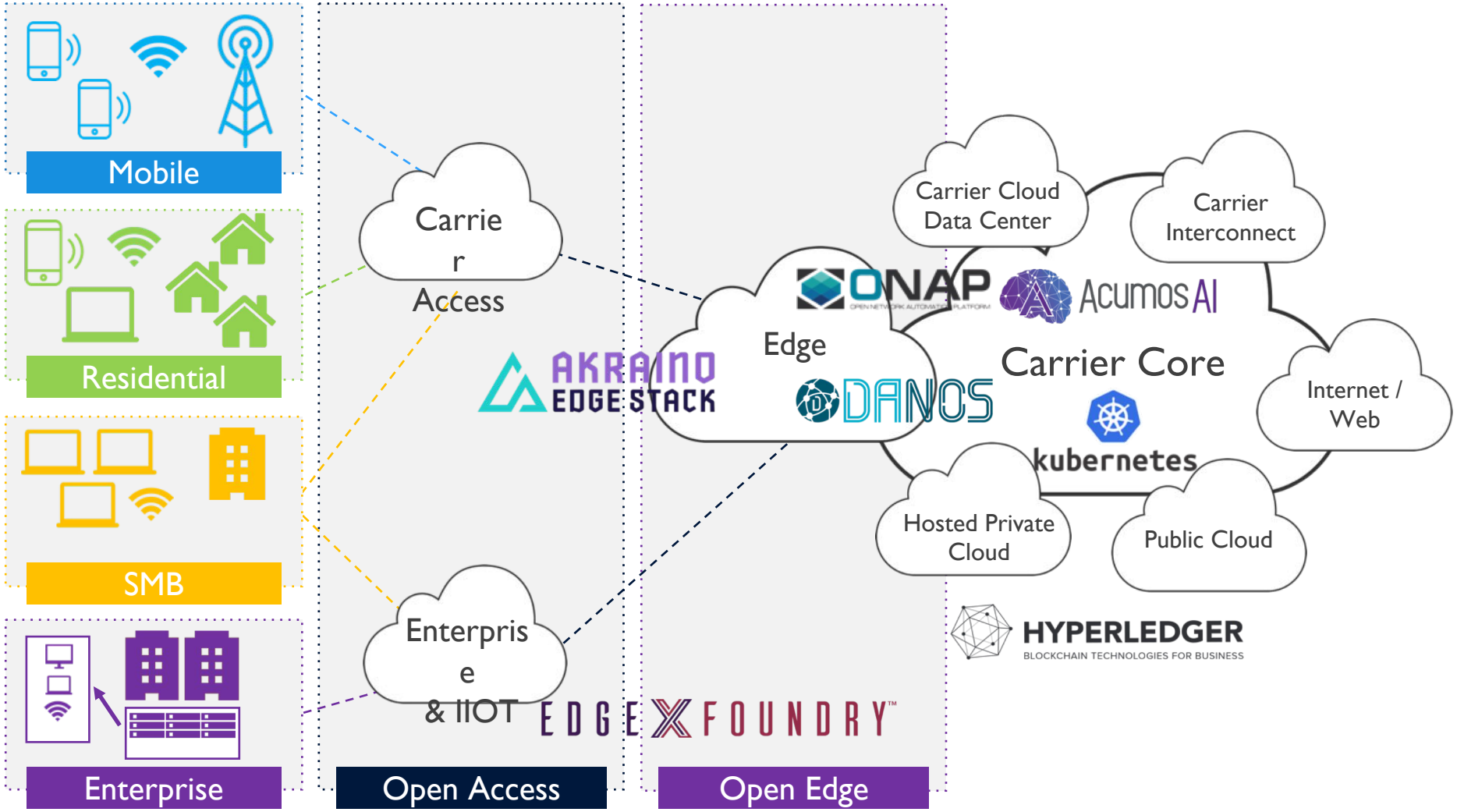
THE LINUX FOUNDATION

De-Facto Platform for ~70% Global Sub

LF Networking Vision: Automating Cloud, Network, & IOT Services



Bringing It All Together Core to Edge – LF Open Source
Network + Disaggregation + Edge + IOT + AI + Cloud + Blockchain



Standards for Edge

ETSI ISG MEC
World Class Standards

Ref Implementation

CORD

openEDGEcomputing

IoT, Gateway & Cloud Ref Arch

OpenFog

Other Edge Activities

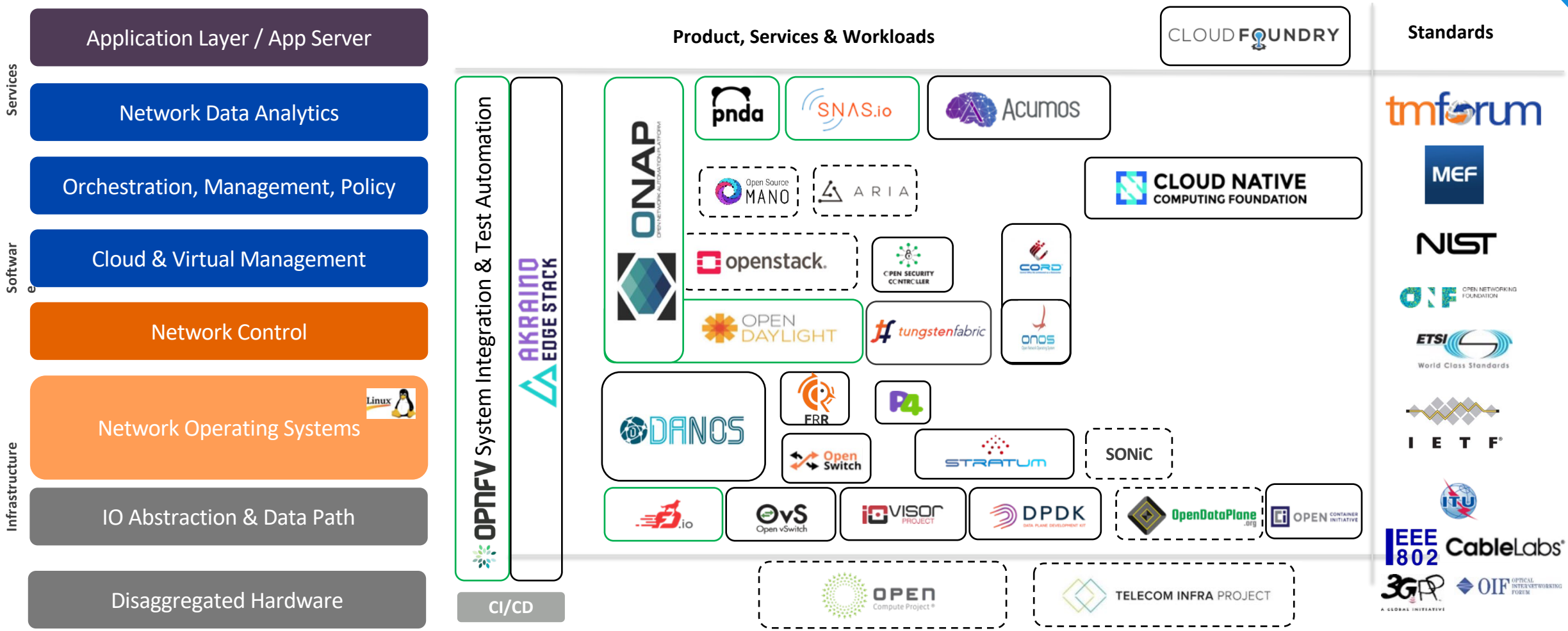
Open Source Networking Landscape
Linux Foundation hosts 9/10 Top projects

Updated

Linux Foundation Hosted

LF Networking Harmonized

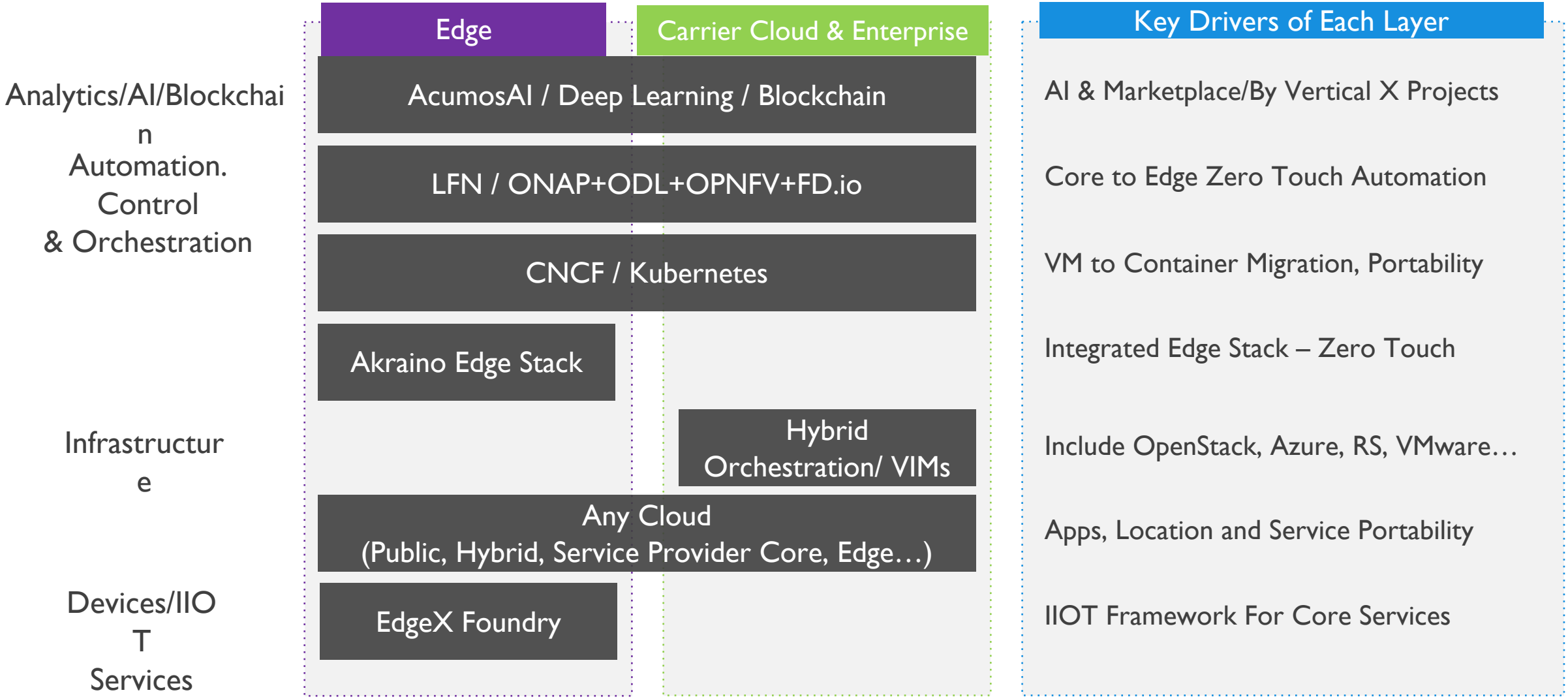
Outside Linux Foundation



THE LINUX FOUNDATION

Automation of Network + Infrastructure + Cloud + Apps + IOT

Linux Foundation Path to Open Source Harmonization 2.0

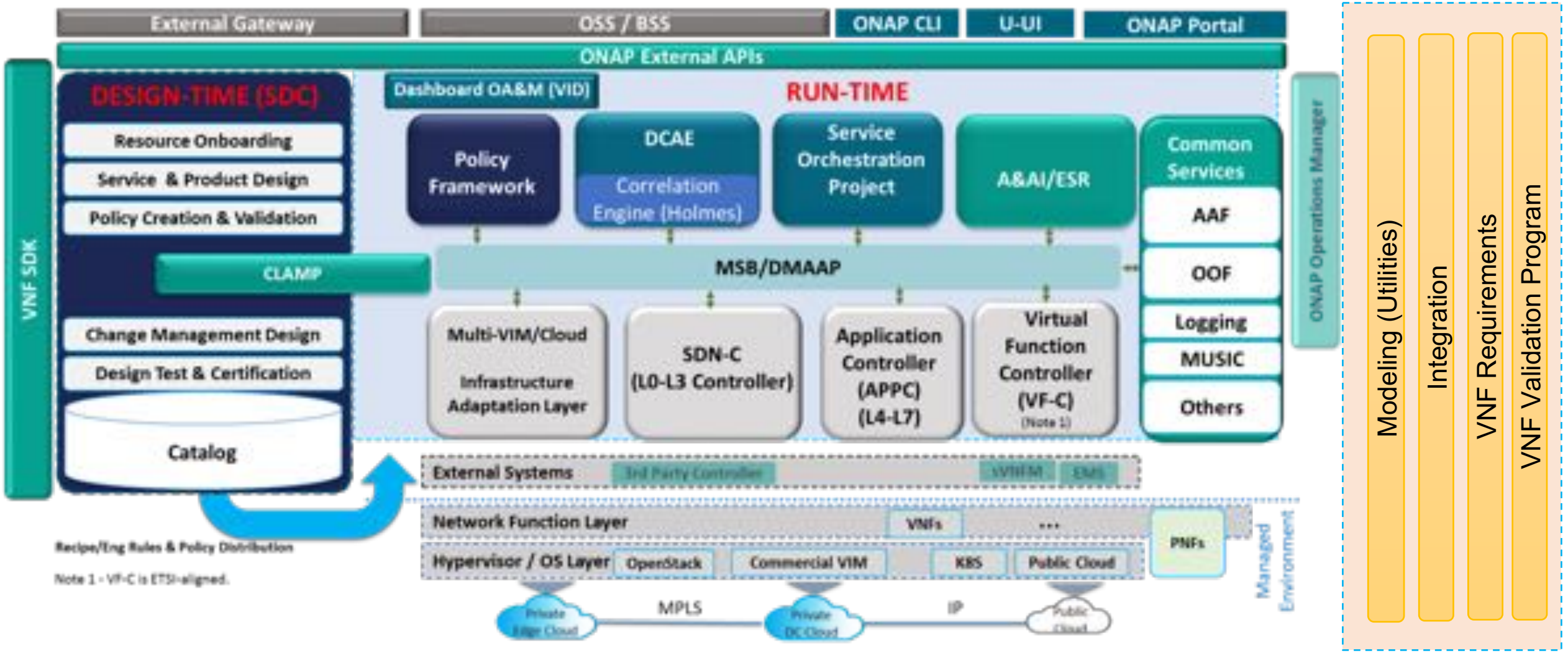


THE LINUX FOUNDATION

The deep dive – VNFs on ONAP & Cloud Native journey

THE LINUX FOUNDATION

ONAP Beijing Architecture

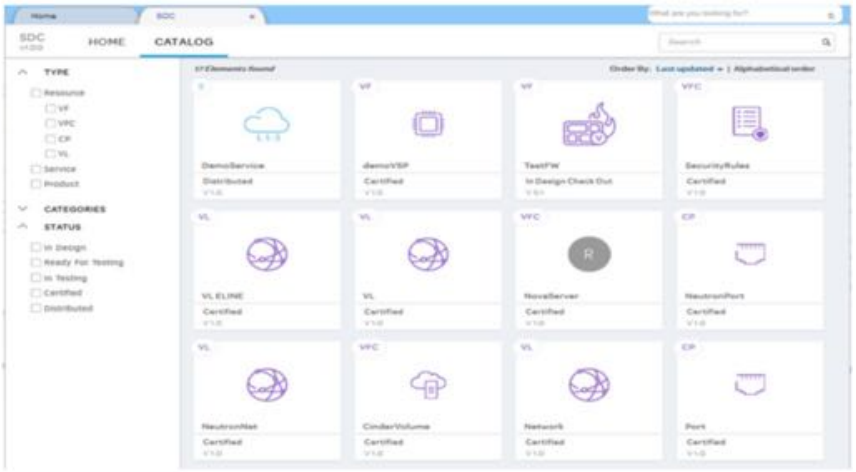


A Day in the Life of an ONAP Service

VNF

① Vendor provided VNF (cloud-hosted, optimized or native)

② Vendor packages VNF as per ONAP requirements; can use VNF SDK



Design

③ Design/test teams onboard VNFs

④ Designers create products, services, recipes

Run-Time

⑤ OSS/BSS system triggers service deployment

⑥ Service lifecycle management

Closed Loop

⑦ Constant data collection, analytics, event monitoring; S3P

Credit: Aarna Networks, ONAP Training course

Kubernetes Gap Analysis & Transition plans

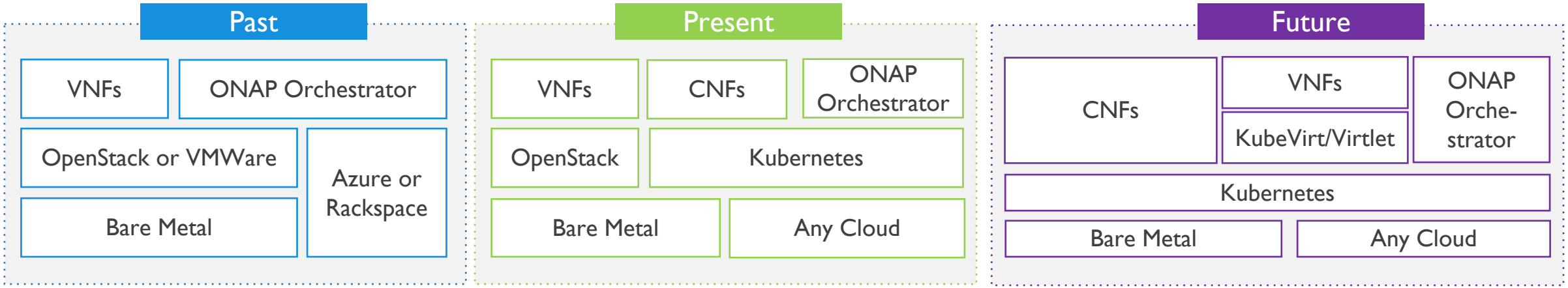
Top 3 Areas of Investigation

1. Support for VNFs/Apps from different vendors (Ref guest OS and VNF ecosystem alignment)
2. Security – Access rights and privileges, known rules for admins etc
3. Network Specific Requirements – focus on performance, scale and capabilities
   - Enabling scheduling container workloads with network-related hardware constraints (similar to what's been done for GPUs)
   - Multi-homed containers (Multis, a CNI plugin is working on this)
   - Functional parity when deployed with IPv6
   - Network-specific functionality like QOS
   - multiple vNIC for a given container, which is currently not supported.

Transition plan

- Demo of Any app, Any cloud with ONAP running on Kubernetes @ ONS opening keynote
- Kubernetes as the Virtual Infrastructure Manager (VIM) for running the ONAP Orchestrator
- Consider Kubernetes as the ONAP Application Controller (AppC)
- As VNFs can be containerized, do so and manage them with Kubernetes – prioritize use cases (eg CDN, DNS)

Evolving from VNFs to CNFs



- ONAP Amsterdam (Past) runs on OpenStack, VMware, Azure or Rackspace
- ONAP Casablanca (Present) runs on Kubernetes and so works on any public, private or hybrid cloud
- Virtual Network Functions (VNFs) are virtual machines that run on OpenStack or VMWare, or can be run on K8s via KubeVirt or Virtlet

Today's Agenda

BACKGROUND AND VISION

1:30 Introduction to VNFs and CNFs & Cross-cloud Dan Kohn

2:00 Networking & Telecom Automation: VNF to CNF journey Arpit Joshipura

REQUIREMENTS

2:15 Cloud Native lessons and requirement: A view from end user - Telus, Sanah Tariq

2:30 (Dan/Arpit facilitate) Why Telecom and Cloud Native technologies are coming together – discuss challenges and requirements

BREAK

PROJECTS AND ROADMAP

3:30 Overview of projects solving the migration Roadmap to Cloud Native

3:50 Network Service Mesh (VPP/Ligato) Ed Warnicke

4:10 Cross-Cloud CI working group Taylor

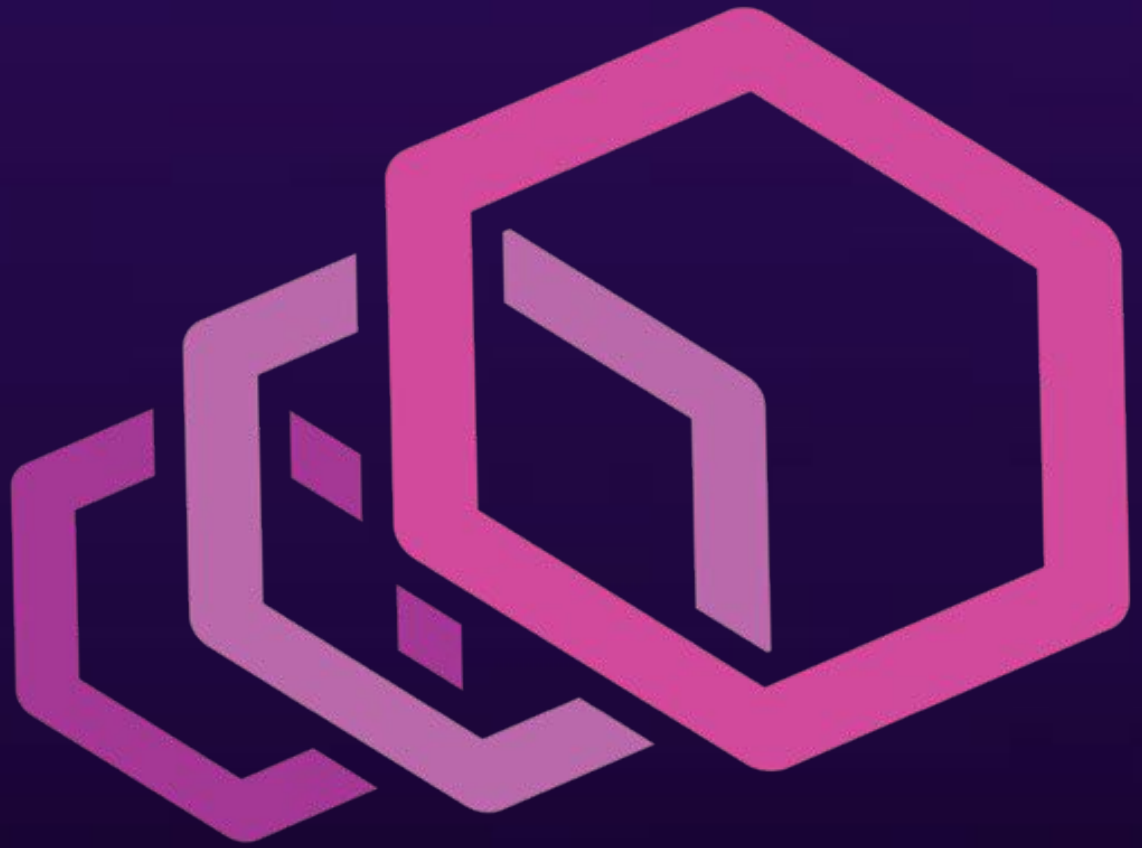
4:30 Wrap up and How to get involved


THE LINUX FOUNDATION

Envoy&Istio overview

Ihor Dvoretskyi, @idvoretskyi,
Developer Advocate, CNCF

CLOUD NATIVE
COMPUTING FOUNDATION



History / Community

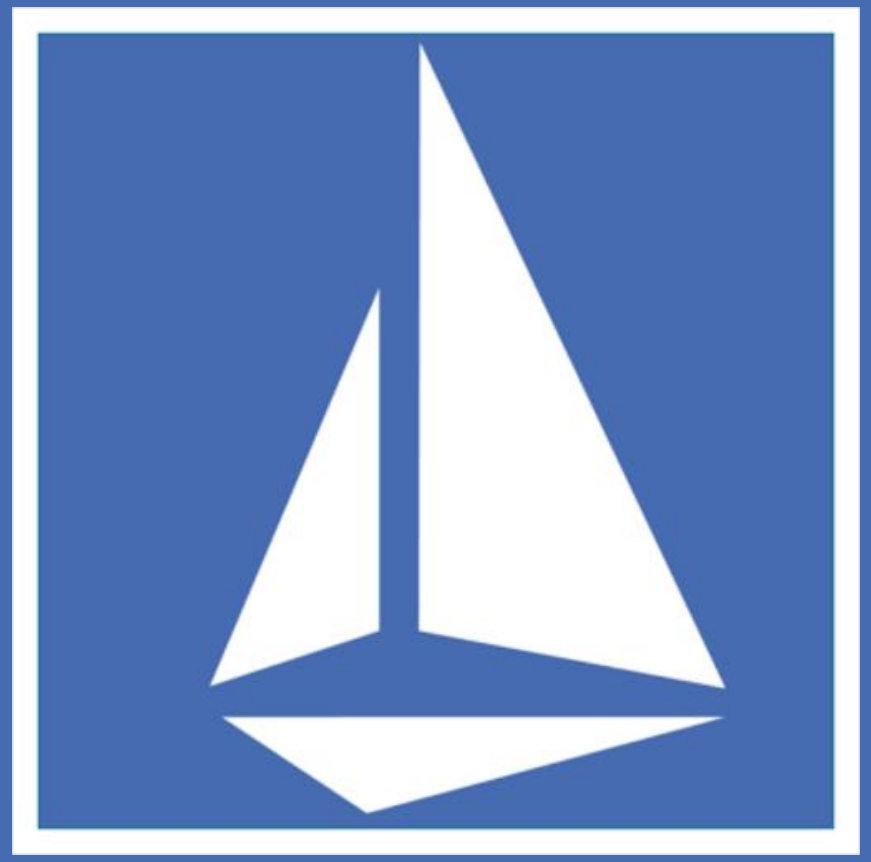
- **Inception** May ‘15
- **Open sourced** September ‘16: https://github.com/envoyproxy/envoy
- **Joined CNCF** September’17: https://www.envoyproxy.io/
- **Users/contributors** (partial list): Lyft, Google, IBM, Verizon, Apple, Microsoft, Pivotal, Red Hat, EBay, Stripe, VSCO, Tencent QQ, Twilio, Yelp … and more all the time.
- **Integrations** (partial list): Istio (Google/IBM), Nomad (Verizon), Tigera, Covalent, Turbine Labs, Datawire … more on the way.
- **Lyft deployment**: 10s of thousands of hosts, 100s of services, 3M + mesh RPS.

State of Service-Oriented Architecture networking

- **Protocols** (HTTP/1, HTTP/2, gRPC, databases, caching, etc.).
- **Infrastructures** (IaaS, CaaS, on premise, etc.).
- Intermediate **load balancers** (AWS ELB, F5, etc.).
- **Observability** output (stats, tracing, and logging).
- Implementations (often partial) of **retry**, **circuit breaking**, **rate limiting**, **timeouts**, and other distributed systems best practices.
- **Authentication** and **Authorization**.
- Per language **libraries** for service calls.

Envoy overview

- **Out of process architecture**: Let's do a lot of really hard stuff in one place and allow application developers to focus on business logic.
- **Modern C++ code base**: Fast and productive.
- **L3/L4 filter architecture**: A byte proxy at its core. Can be used for things other than HTTP (e.g., MongoDB, redis, stunnel replacement, TCP rate limiter, etc.).
- **HTTP L7 filter architecture**: Make it easy to plug in different functionality.
- **HTTP/2** first! (Including **gRPC** and a nifty gRPC HTTP/1.1 bridge).
- **Service discovery** and **active/passive health checking.**
- **Advanced load balancing**: Retry, timeouts, circuit breaking, rate limiting, shadowing, outlier detection, etc.
- Best in class **observability**: stats, logging, and tracing.
- **Edge proxy**: routing and TLS.



Istio overview

- Connect, manage and secure microservices
- Separation of concerns: developer and operations
- For both containerized and non-containerized workloads
- Leverage great functionality in Envoy, adding pluggable management and control planes
- Intelligent routing, load balancing, metrics collection, policy enforcement, end-to-end authentication
- Istio 1.0 announced in July’2018

Istio components

- **Mixer** - Central component that is leveraged by the proxies and microservices to enforce policies such as authorization, rate limits, quotas, authentication, request tracing and telemetry collection.
- **Pilot** - A component responsible for configuring the proxies at runtime.
- **Citadel** - A centralized component responsible for certificate issuance and rotation.
- **Node Agent** - A per-node component responsible for certificate issuance and rotation.

Istio and Envoy

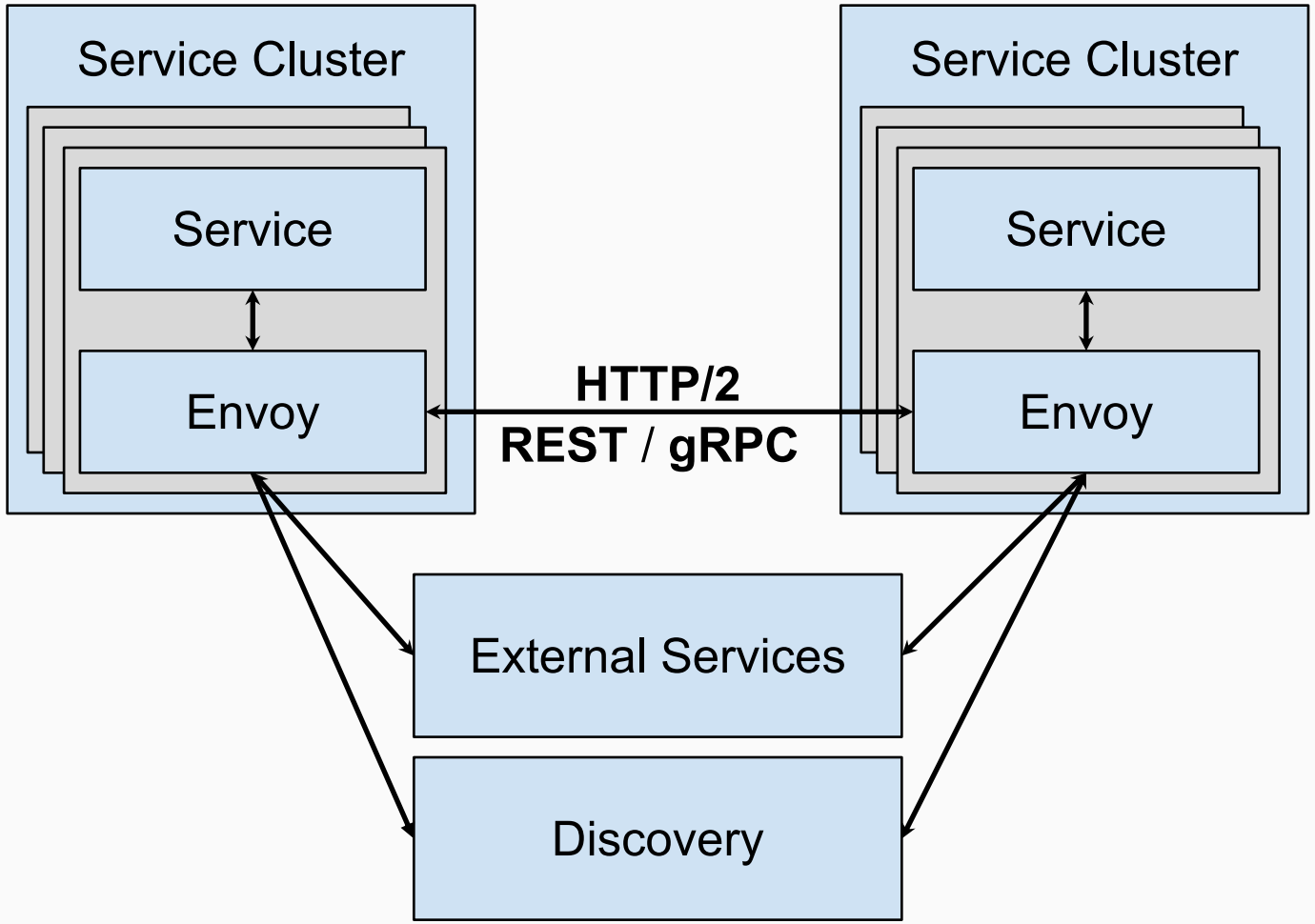
- **Envoy** - Sidecar proxies per microservice to handle ingress/egress traffic between services in the cluster and from a service to external services
- Providing a rich set of functions like:
    - discovery
    - l7 routing
    - circuit breakers
    - policy enforcement
    - telemetry recording/reporting functions.

More details

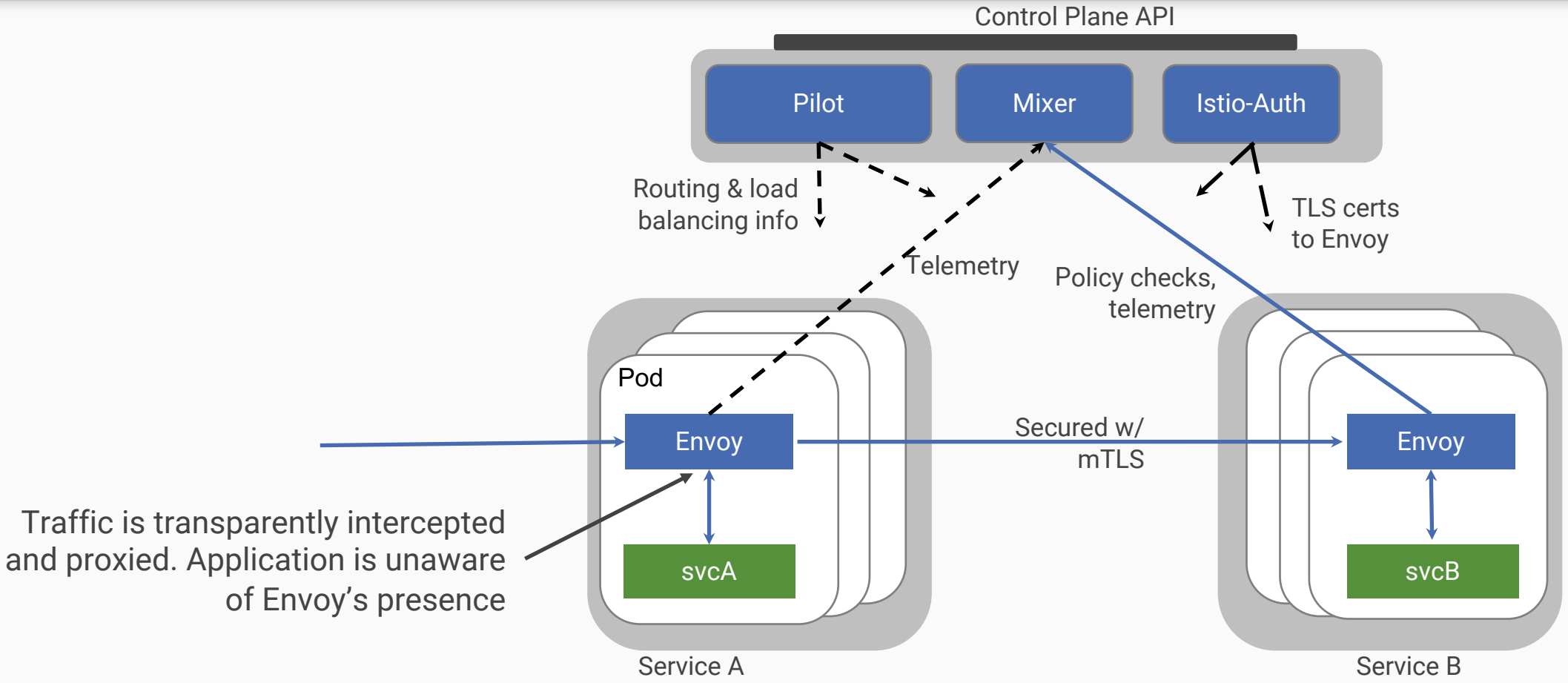
- **Envoy:**
  - https://www.envoyproxy.io/
  - https://github.com/envoyproxy/envoy
- **Istio:**
  - https://istio.io/
  - https://github.com/istio/istio

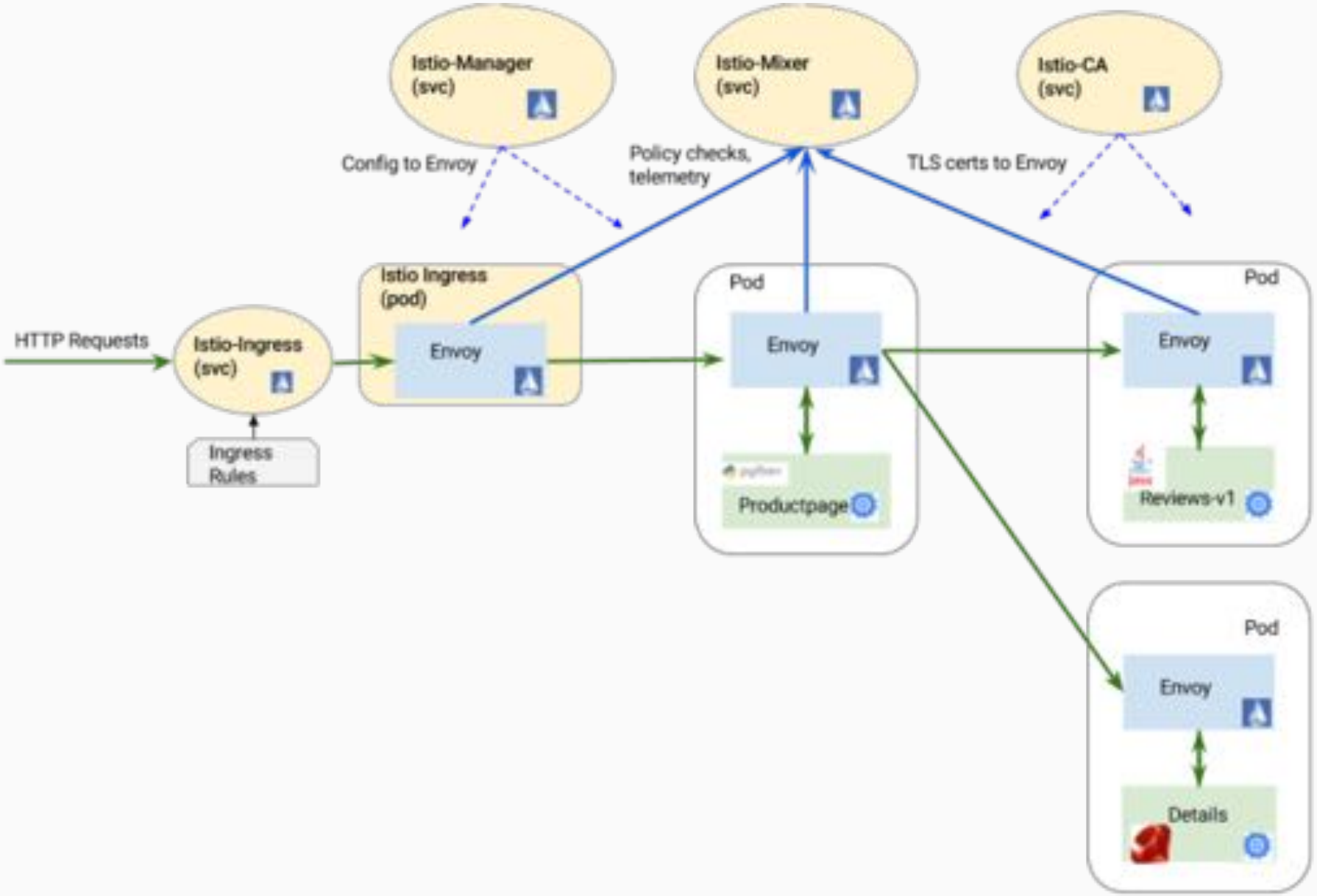
Extra

Envoy overview



Architectural overview





Bookinfo application sample

Cross-Cloud CI Overview

Cloud-native Network Functions Seminar

August 28, 2018

CLOUD NATIVE
COMPUTING FOUNDATION

Presented by:

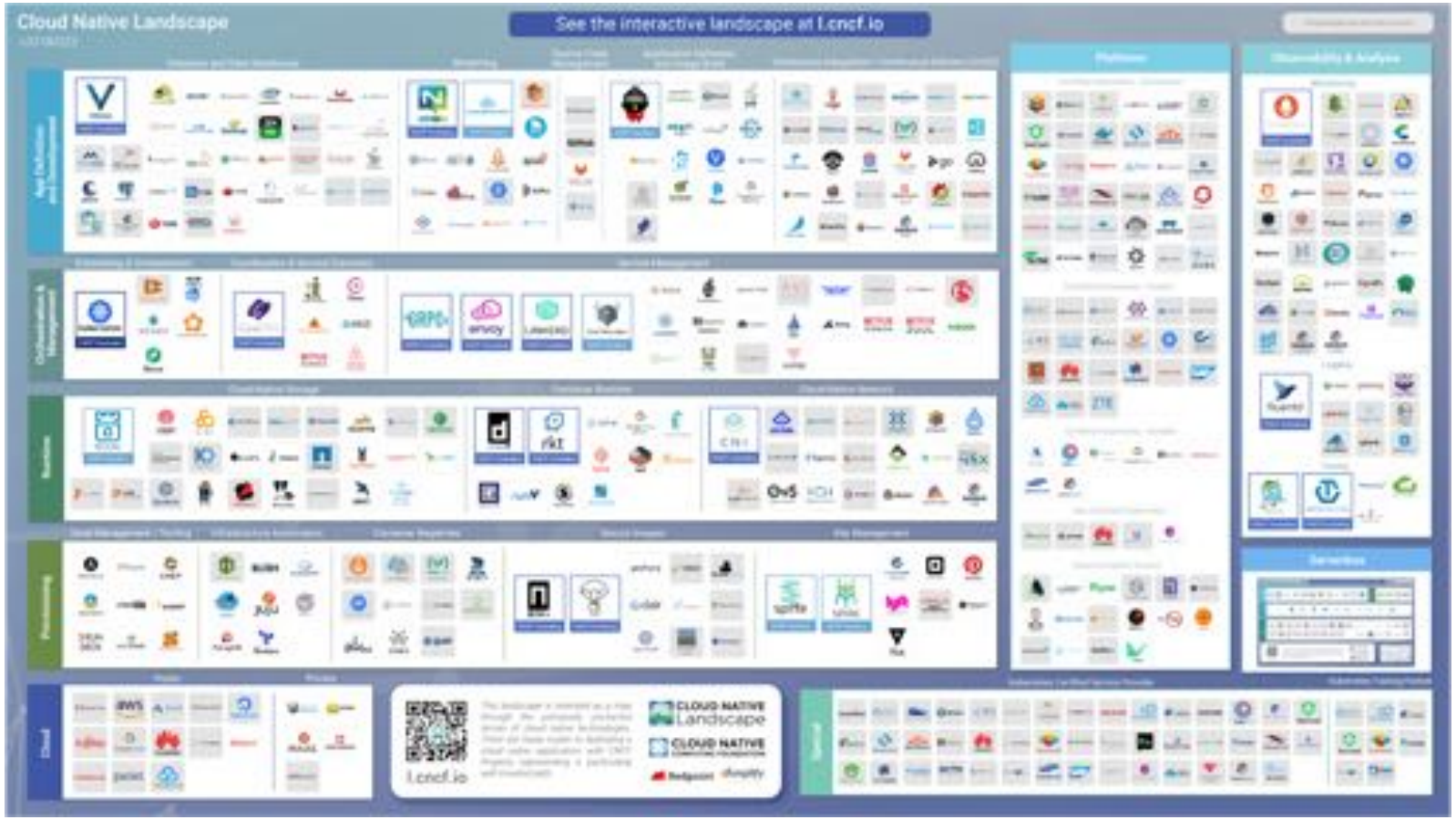
Taylor Carpenter, Vulk Coop
taylor@vulk.coop

Cross-Cloud CI Project Overview

Why?

Cross-Cloud CI Project Overview

Why? CNCF ecosystem is growing rapidly with new projects and cloud providers!



Cross-Cloud CI Project Overview

Why?

- The CNCF ecosystem is large, diverse and continues to grow. CNCF would like to ensure cross-project interoperability and cross-cloud deployments of all cloud native technologies and show the daily status of builds and deployments on a status dashboard.

Cross-Cloud CI Project Overview

Why?

- The CNCF ecosystem is large, diverse and continues to grow. CNCF would like to ensure cross-project interoperability and cross-cloud deployments of all cloud native technologies and show the daily status of builds and deployments on a status dashboard.

What?

- Cross-cloud testing system
- Status repository server
- Status dashboard

Build and provision CNCF projects

Graduated

Kubernetes
Orchestration

Prometheus
Monitoring

Incubating

OpenTracing
Distributed Tracing API

Fluentd
Logging

gRPC
Remote Procedure Call

containerd
Container Runtime

rkt
Container Runtime

CNI
Networking API

Envoy
Service Mesh

Jaeger
Distributed Tracing

Notary
Security

TUF
Software Update Spec

Vitess
Storage

CoreDNS
Service Discovery

NATS
Messaging

Linkerd
Service Mesh

Helm
Package Management

Sandbox

Rook
Storage

SPIFFE
Identity Spec

SPIRE
Identity

Open Policy Agent
Policy

CloudEvents
Serverless

Telepresence
Tooling

Harbor
Registry

OpenMetrics
Metrics Spec

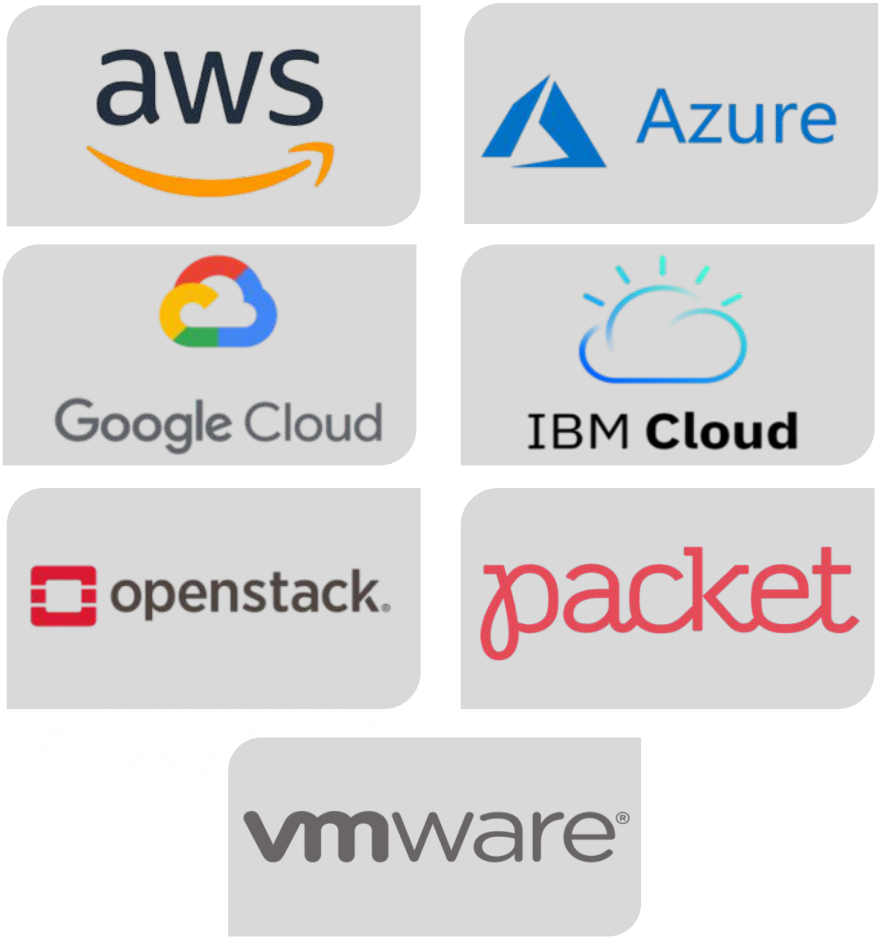
TiKV
Key/Value Store

Project CI artifacts and non-CNCF projects

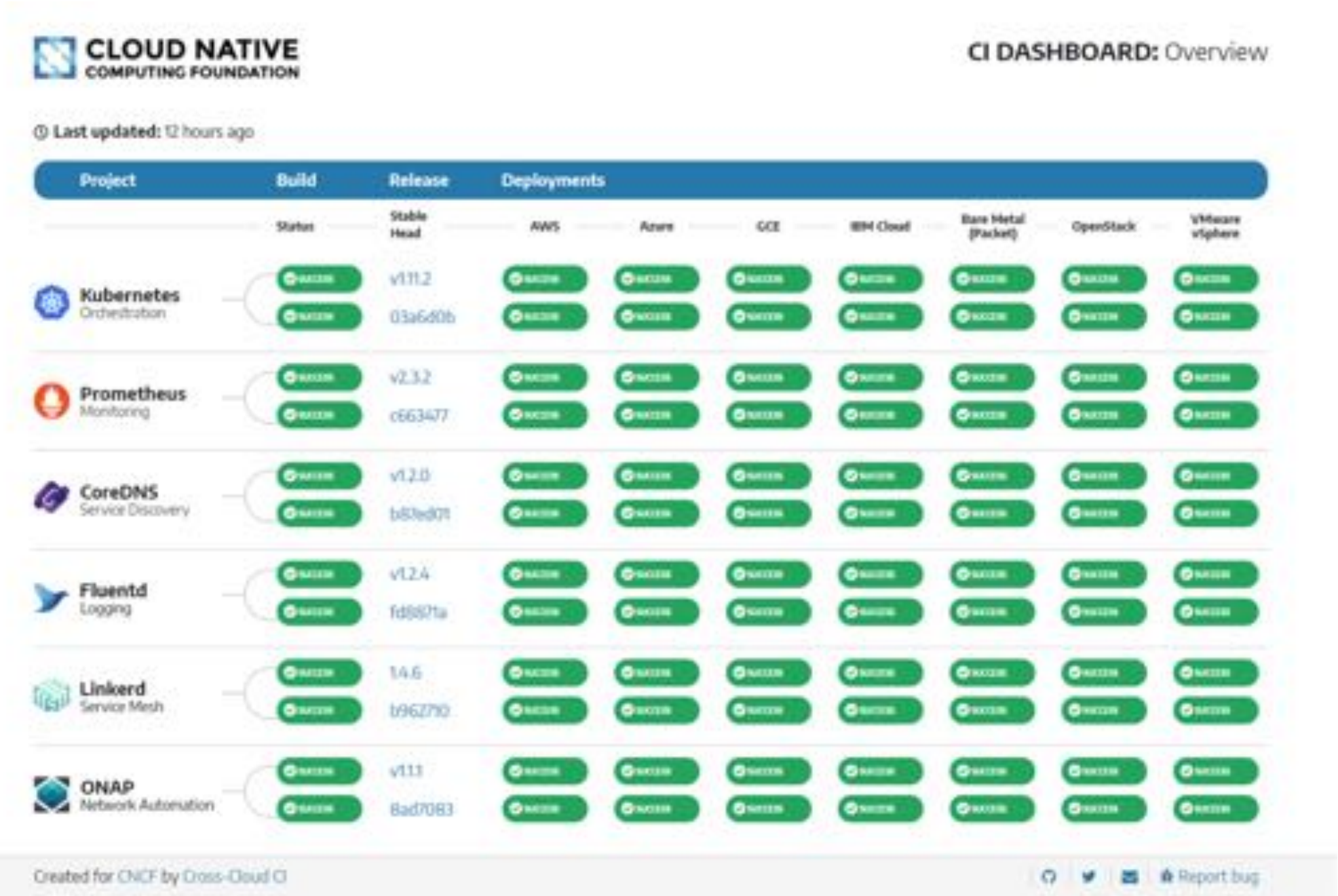
Implemented

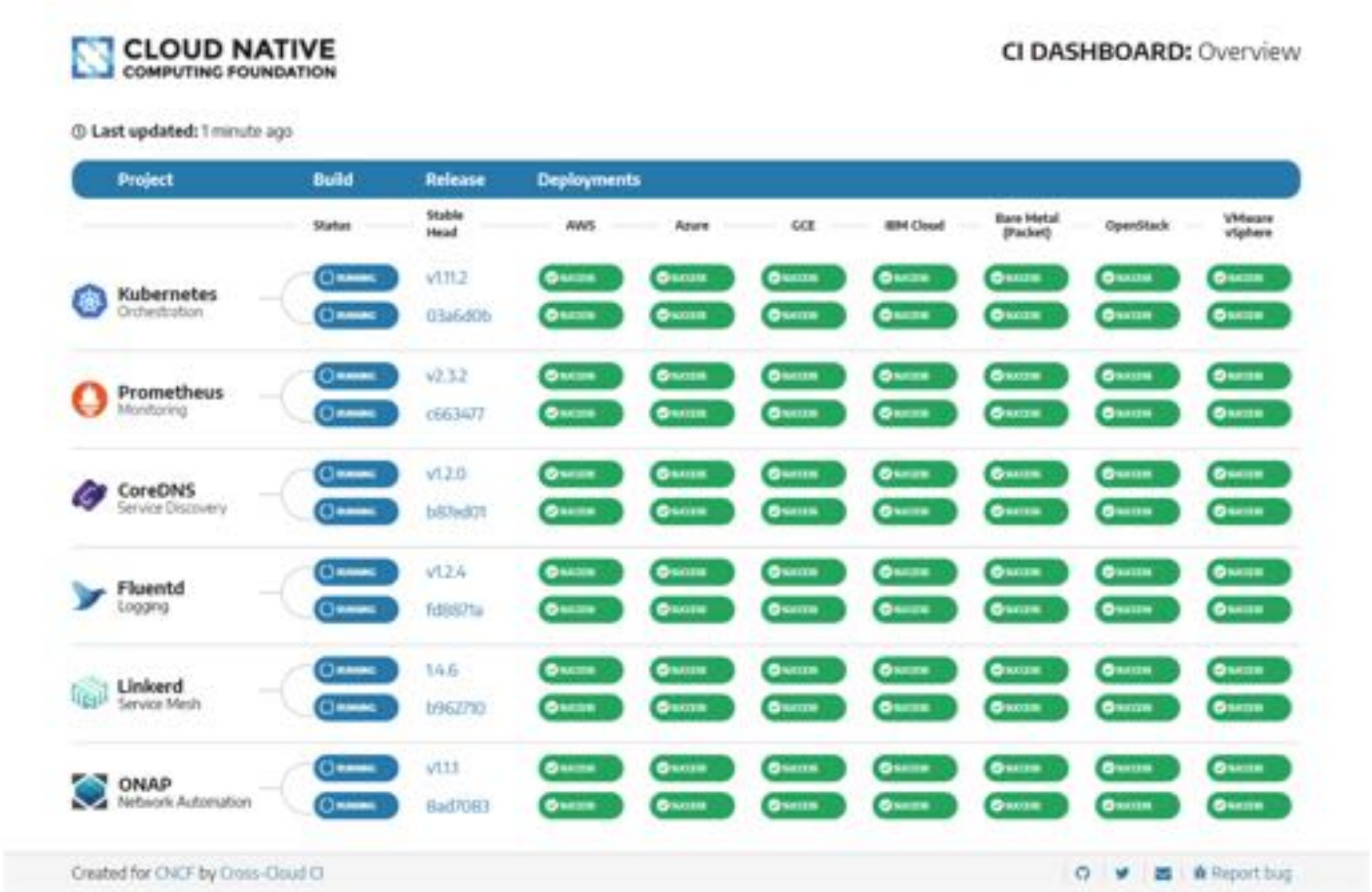
Deploy to public/bare metal/private clouds

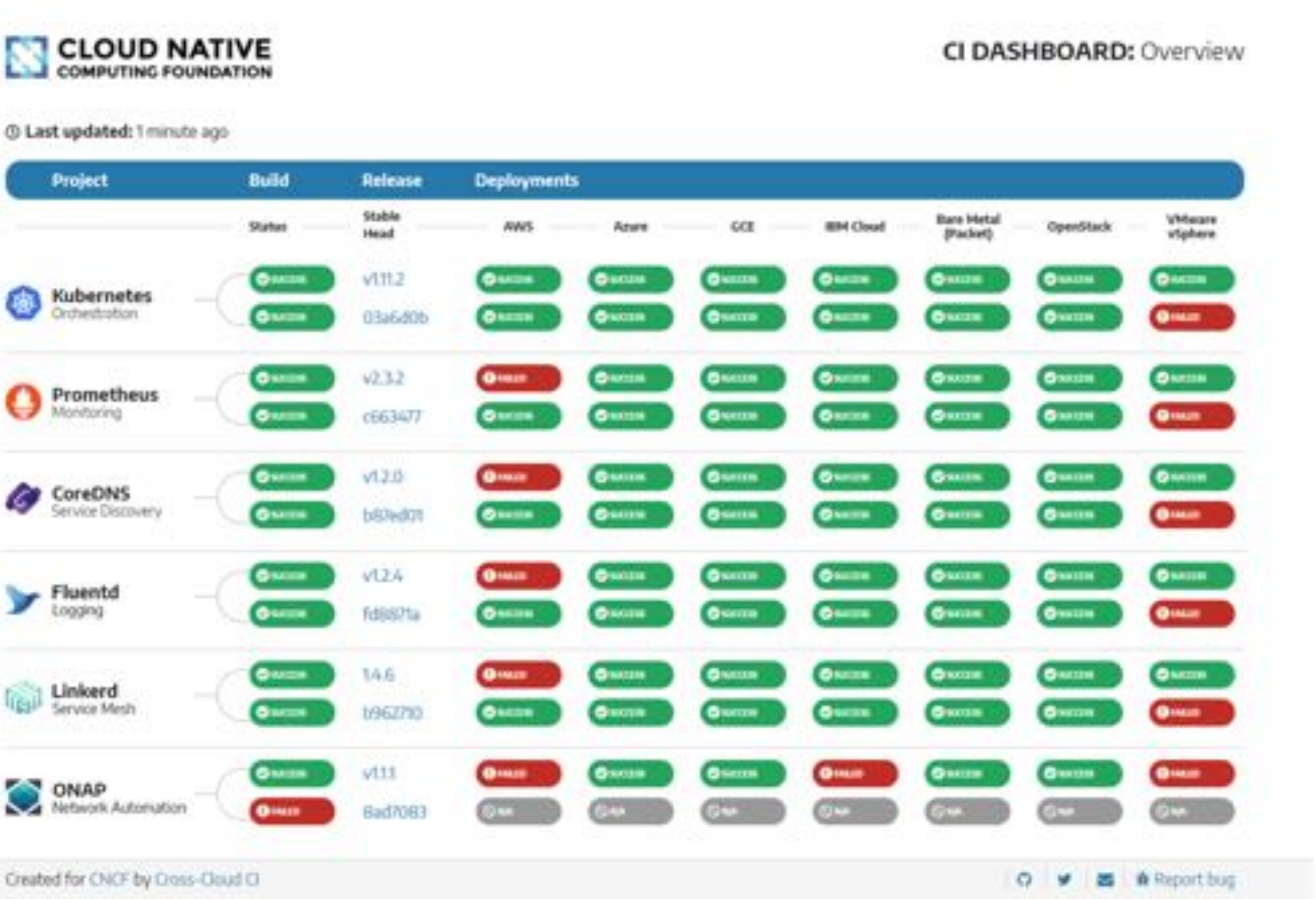
Implemented

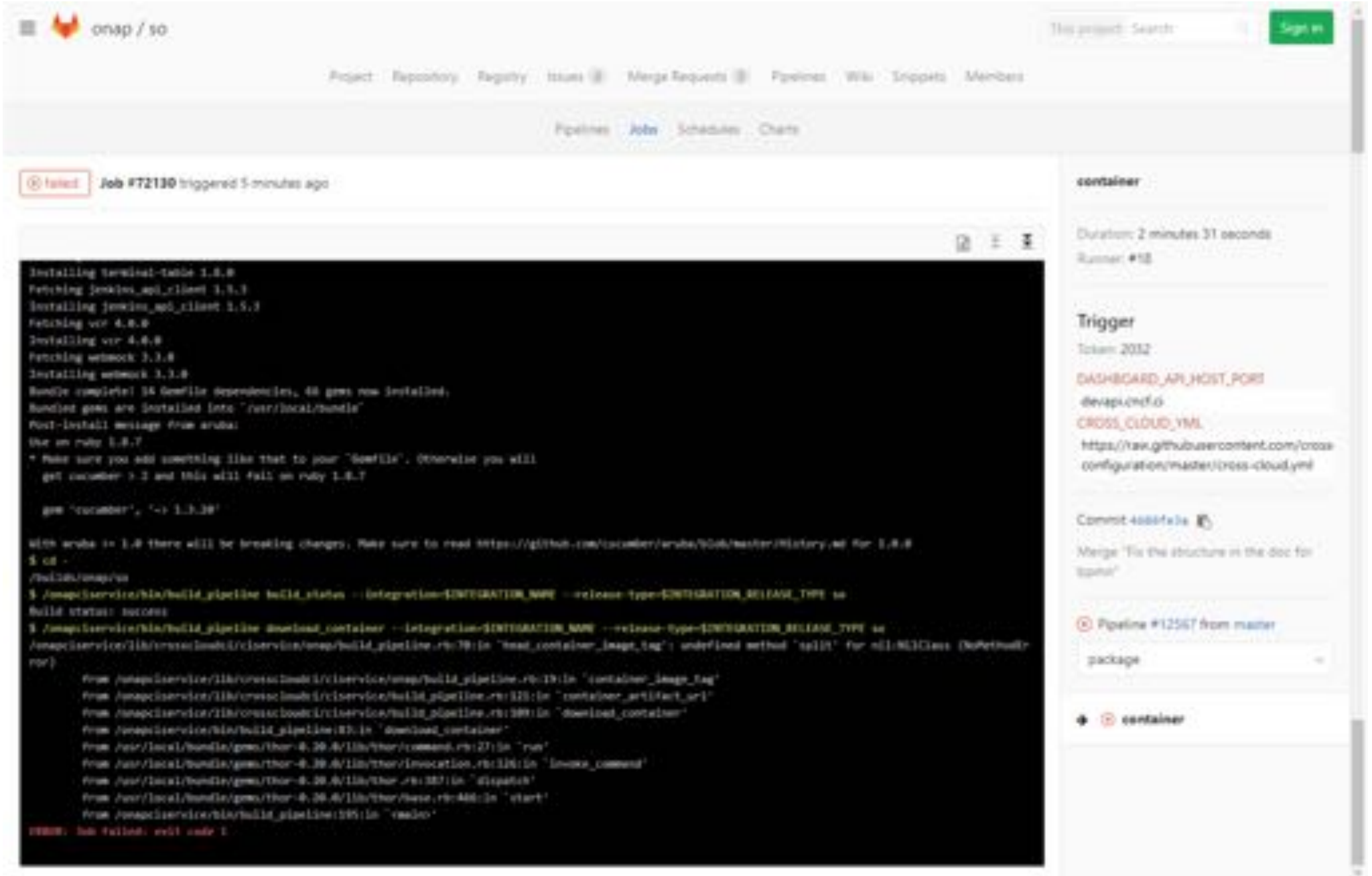


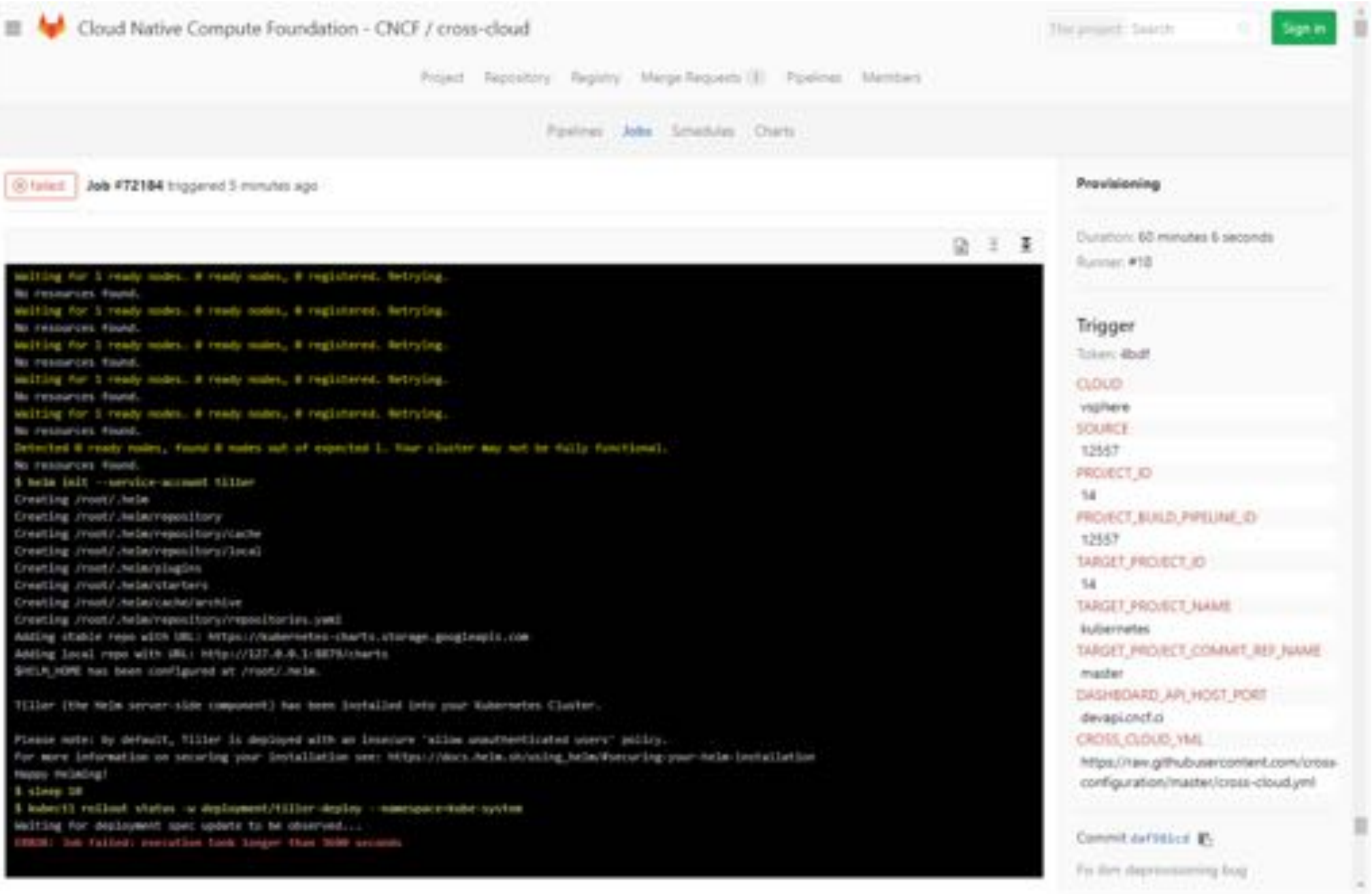
In Progress

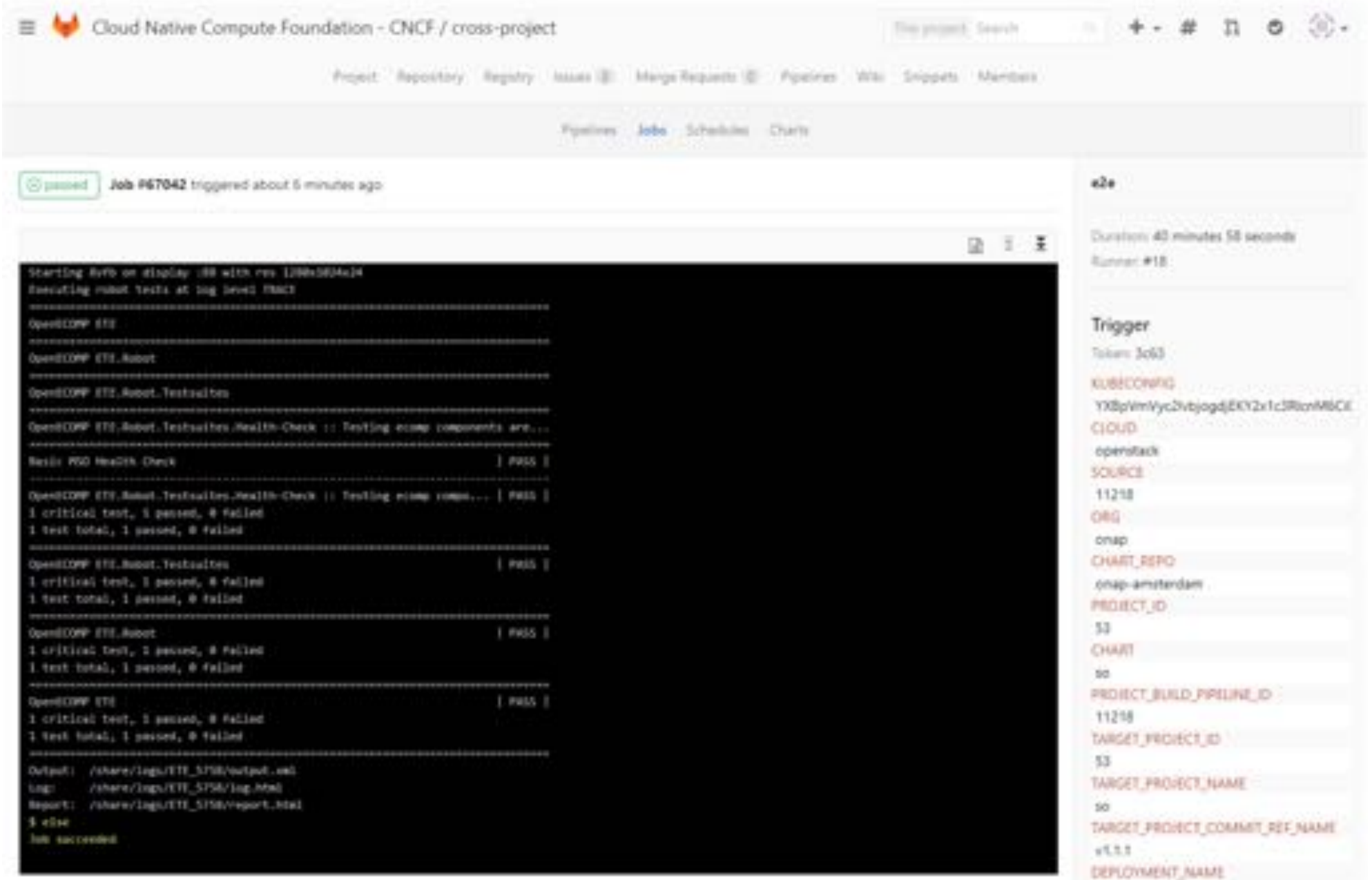












Technology Overview

CLOUD NATIVE
COMPUTING FOUNDATION

CI System Technology Overview

Unified CI/CD platform:

- GitLab

Cross-cloud provisioning:

- Terraform, Cloud-init, and per cloud K8s configuration

App deployments:

- K8s manifest management with Helm

E2e tests:

- Custom containers + Helm

Automated builds and deployments:

- Git + per project yaml configuration

Dashboard Technology Overview

Frontend:

- Vue.js

Status repository:

- Elixir and Erlang

Automated builds and deployments:

- Git + per project yaml configuration

Q&A

CLOUD NATIVE
COMPUTING FOUNDATION

How to Collaborate

Attend CI WG meetings:

- https://github.com/cncf/wg-ci
- 4th Tuesday of the month at 11:00am Pacific Time
- Next Meeting is on Tuesday, Sept 28th

Subscribe to the CNCF CI public mailing list:

- https://lists.cncf.io/g/cncf-ci-public

Create issues on GitHub:

- https://github.com/crosscloudci/cross-cloud/issues

Join the #cncf-ci channel on slack:

- Request invite at https://slack.cncf.io/
- Cloud-native.slack.com

Connect with Cross-Cloud CI

@crosscloudci

@crosscloudci

crosscloudci@vulk.coop

For more details and an in-depth demo, please contact Dan Kohn & Cross-Cloud CI team at CNCF booth at #OSSNA18

Also presenting at:

- KubeCon + CloudNativeCon China
  - November 13-14, Shanghai

CLOUD NATIVE COMPUTING FOUNDATION