

What is Cloud Native and Why You Should Care

Dan Kohn
Executive Director, CNCF



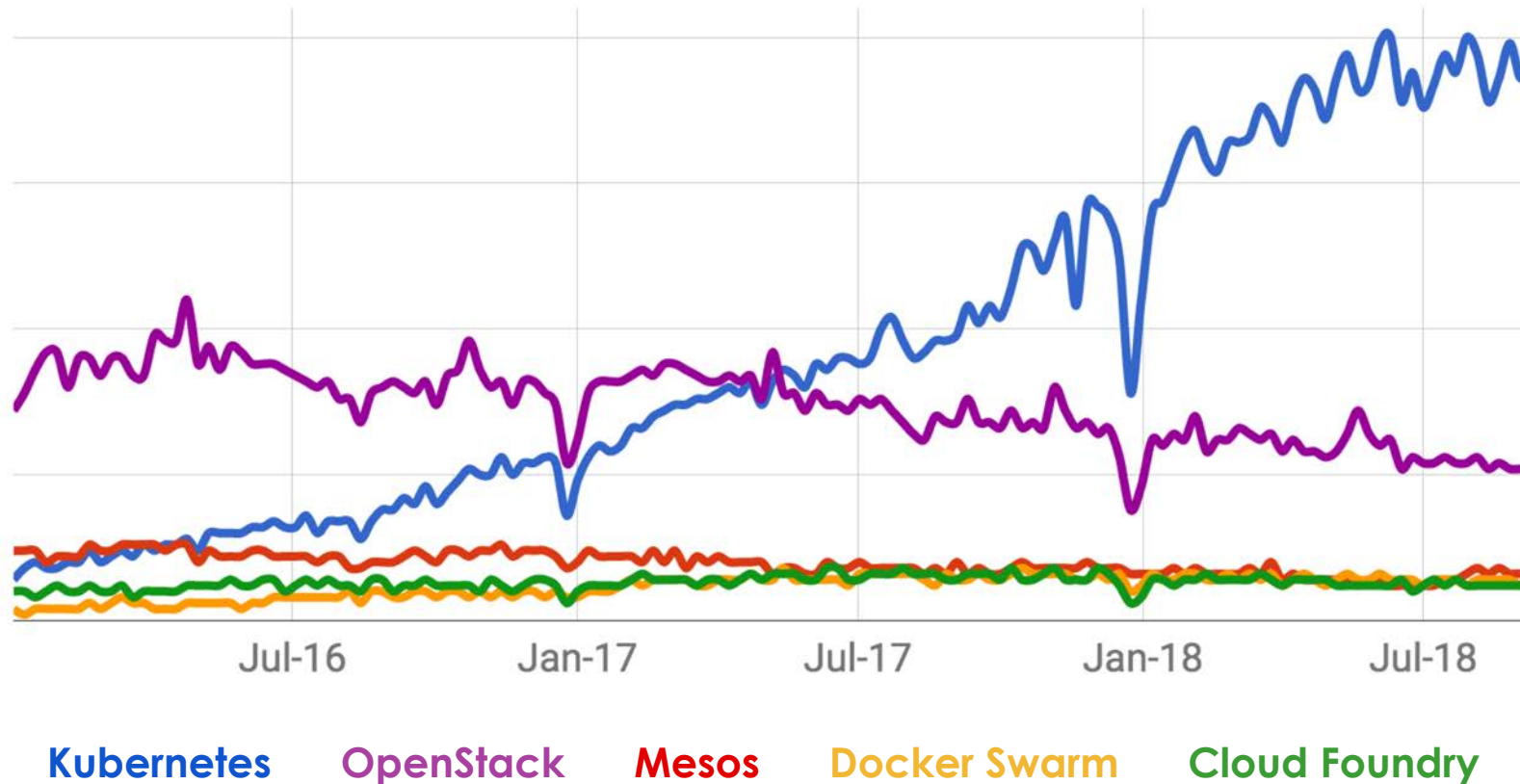
kubernetes

- 4,550 unique authors
- 23,347 unique issue commenters
- 2nd fastest development velocity (behind Linux)
- See devstats.cncf.io

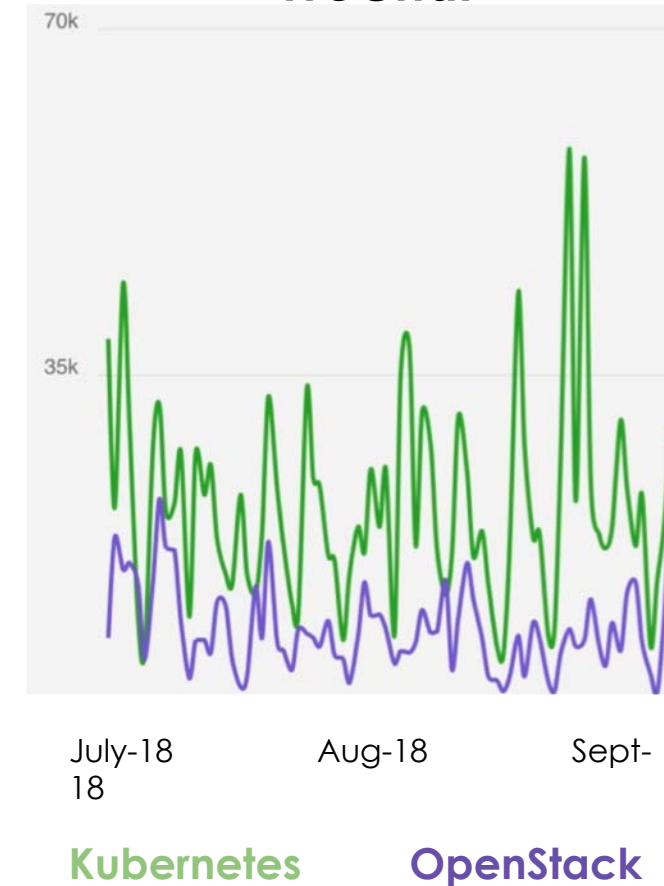


Kubernetes in Search Trends

Google Trends



WeChat



61 Companies in CNCF's End User Community



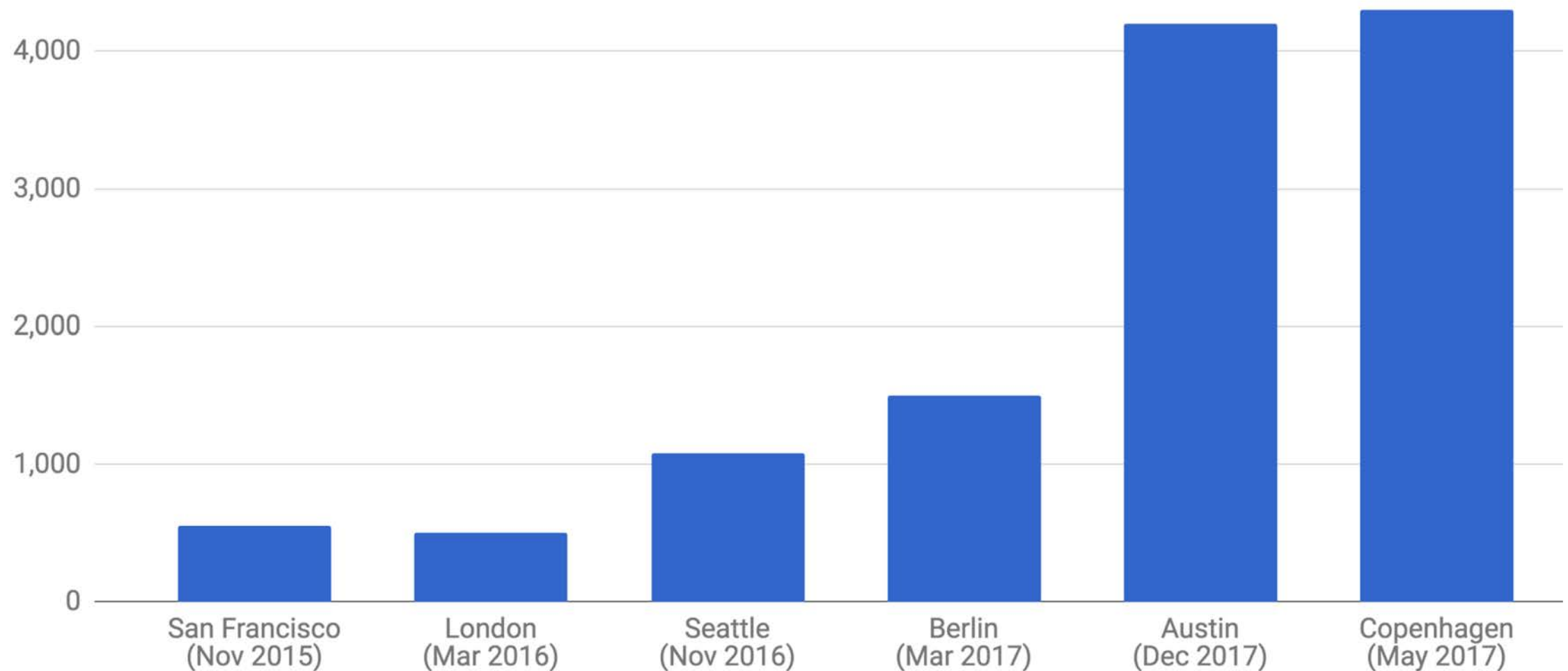
61 Kubernetes Certified Service Providers



71 Certified Kubernetes Partners



KubeCon + CloudNativeCon Attendees





SaaS Architecture: Move Fast and Bake Things

Bootstrapped with single-donut mixer, fryer, and topper

Built with OpenTracing-aware Mutex wrapper

```
graph TD; WC[Web Client] --> WS[Web Server]; WS --> M[Mixer]; WS --> F[Fryer]; WS --> T[Topper];
```

Cloud Native Con Europe 2019 KubeCon

SUPPORTING THE CLOUD NATIVE ECOSYSTEM

www.cncf.io | @CloudNativeFdn

**ORCHESTRATION.
CONTAINERIZATION.
MICROSERVICES.**





CLOUD
NATIVE
CON
Europe 2017



KubeCon
A CNCF EVENT

ORCHESTRATION.
CONTAINERIZATION.
MICROSERVICES.



Cloud Native Definition (1 of 3)

Cloud native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as **public, private, and hybrid clouds**. Containers, **service meshes**, microservices, immutable infrastructure, and declarative APIs exemplify this approach.

cncf.io/d



Cloud Native Definition (2 of 3)

These techniques enable ***loosely coupled systems*** that are resilient, manageable, and observable. Combined with ***robust automation***, they allow engineers to make ***high-impact changes*** frequently and predictably with minimal toil.

cncf.io/d

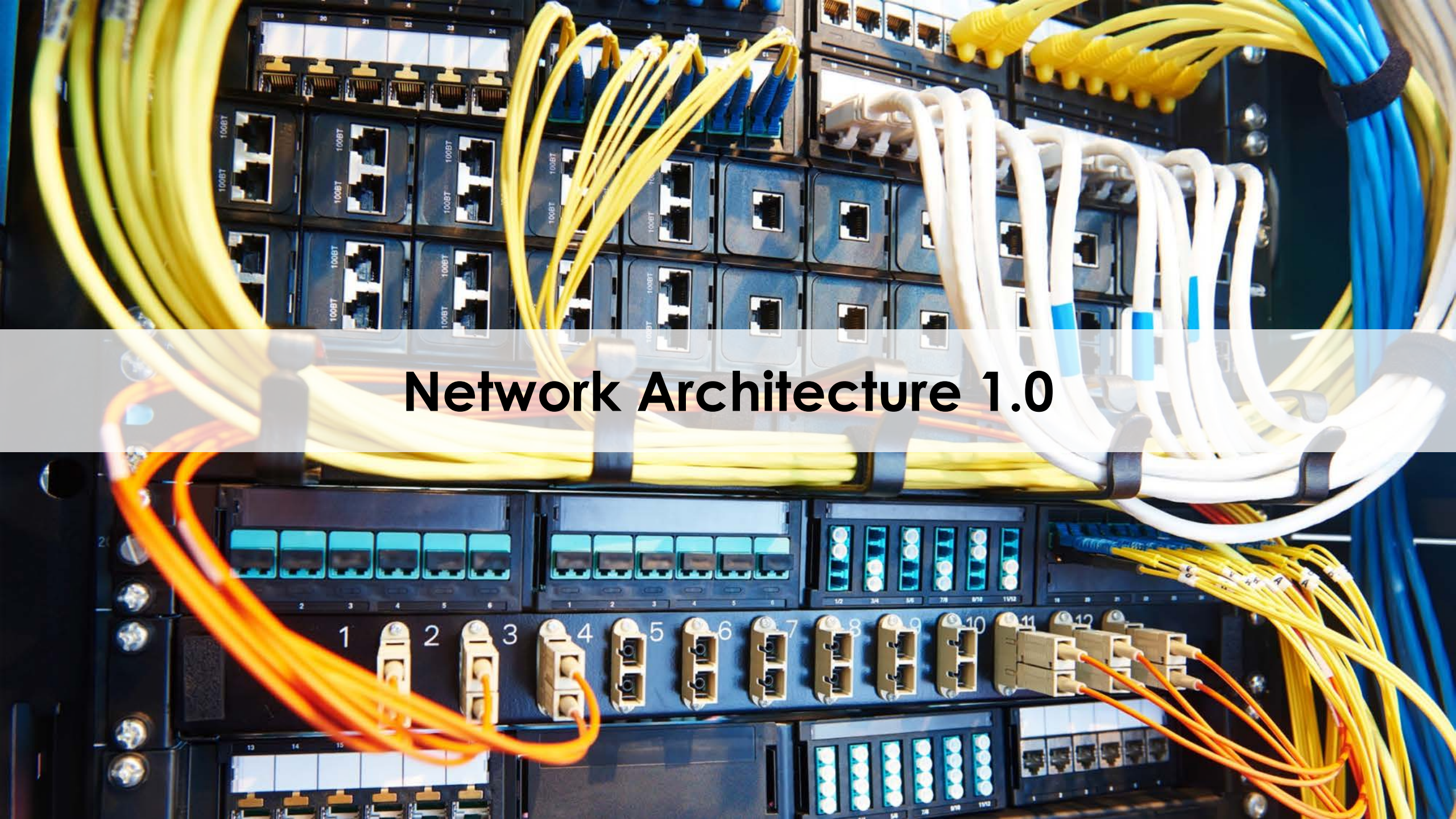


Cloud Native Definition (3 of 3)

The Cloud Native Computing Foundation seeks to **drive adoption** of this paradigm by fostering and sustaining an **ecosystem of open source, vendor-neutral projects**. We democratize **state-of-the-art patterns** to make these innovations accessible for everyone.

cncf.io/d





Network Architecture 1.0

Network Architecture Evolution 1.0

1.0: Separate physical boxes for each component (e.g., routers, switches, firewalls)



A wide-angle photograph of a modern data center aisle. On the right, multiple rows of server racks are visible, densely packed with network equipment. A complex web of colorful Ethernet cables (yellow, orange, blue, green) is plugged into the front of the racks, some bundled together. The floor is a light-colored, polished tile. The perspective is from the end of the aisle, looking down its length.

Network Architecture 2.0

Network Architecture Evolution 2.0

2.0: Physical boxes converted to virtual machines called Virtual Network Functions (VNFs) running on VMware or OpenStack



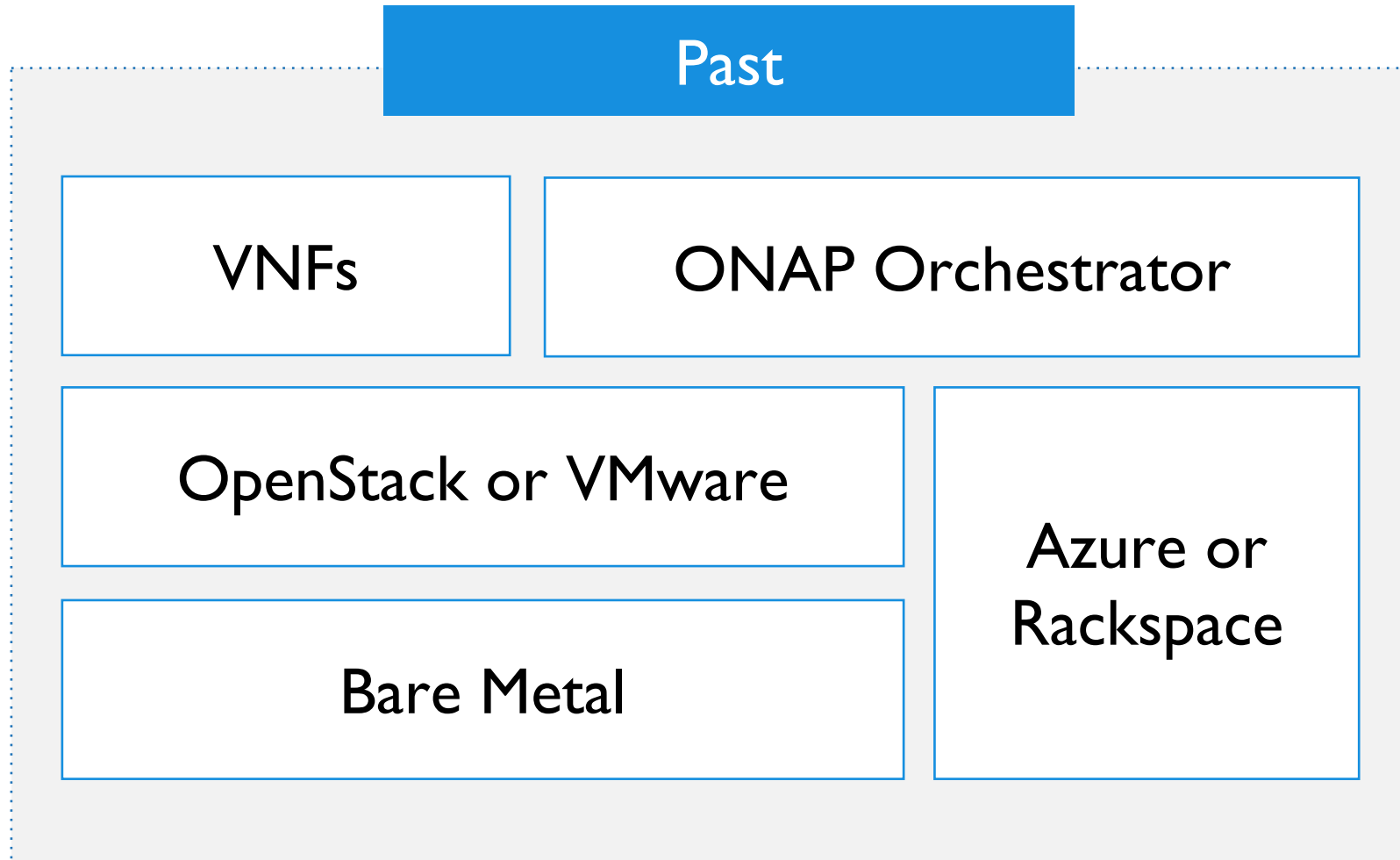
Network Architecture 3.0
(hardware is the same as 2.0)

Network Architecture Evolution 3.0

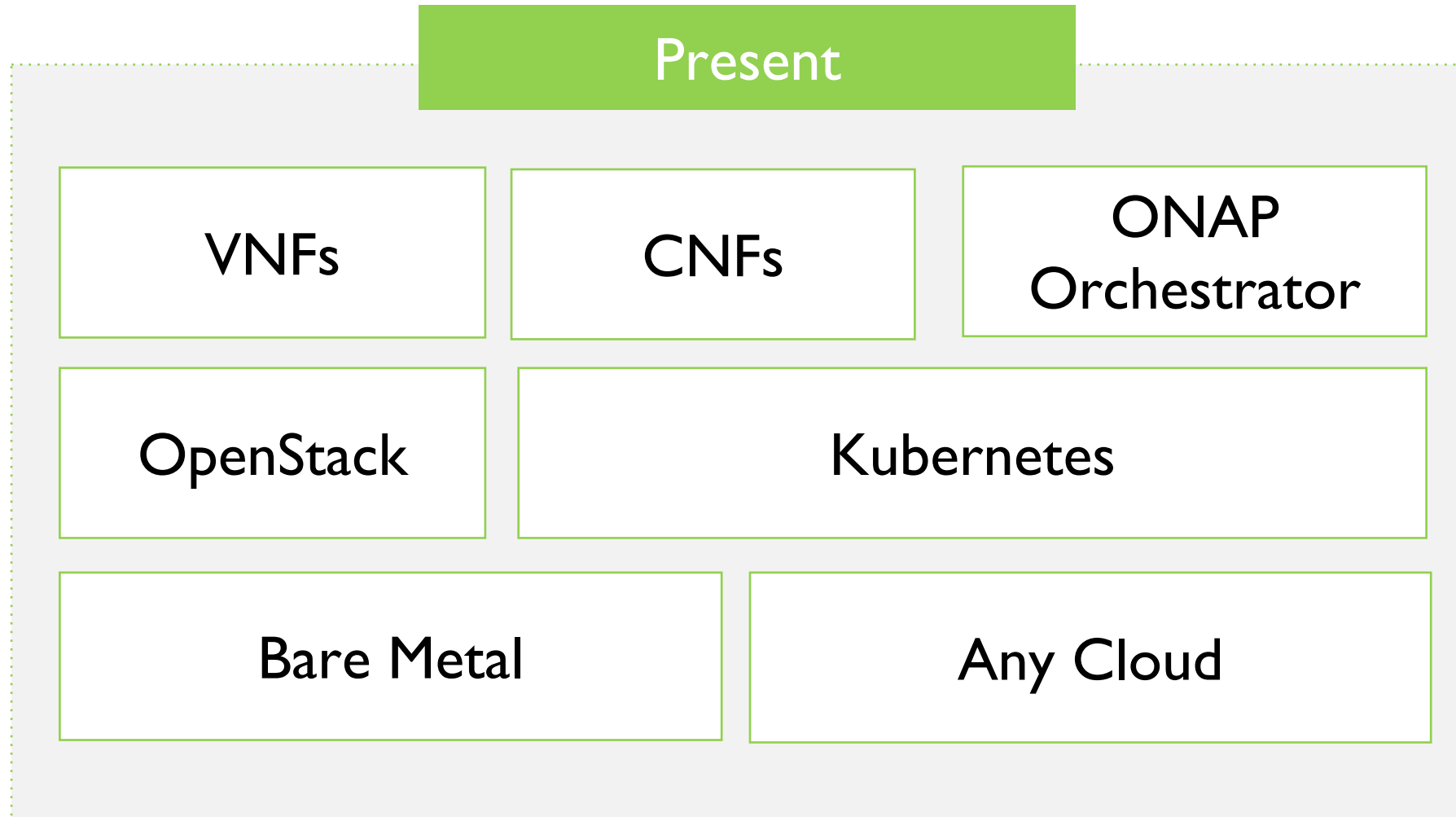
3.0: Cloud-native Network Functions (CNFs) run on Kubernetes on public, private, or hybrid clouds



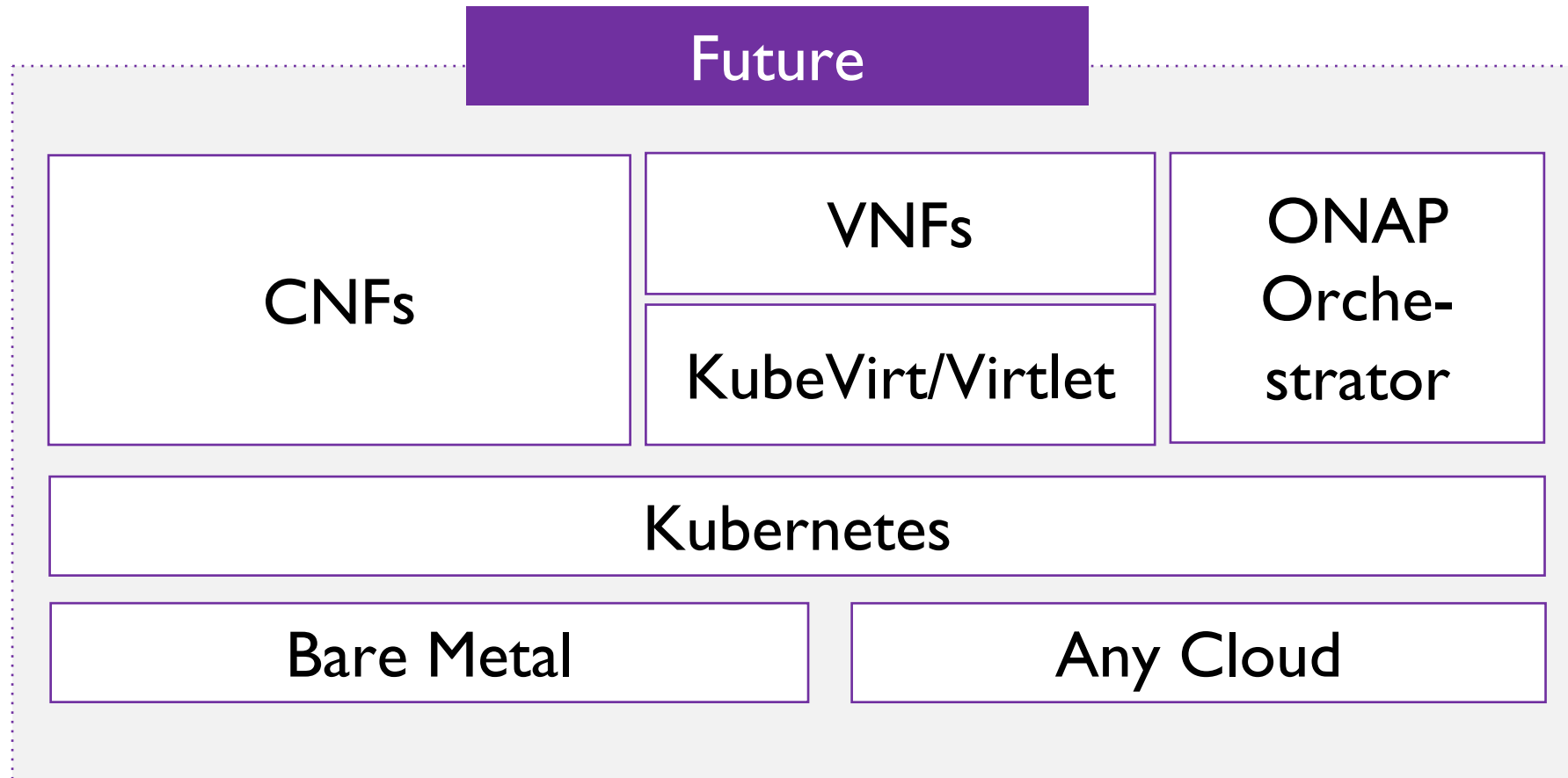
Evolving from VNFs to CNFs (Past)



Evolving from VNFs to CNFs (Present)



Evolving from VNFs to CNFs (Future)



Three Major Benefits

1. Cost savings



Three Major Benefits

2. Improved resiliency (to failures of individual CNFs, machines, and even data centers)



Three Major Benefits

3. Higher development velocity



The challenge of transitioning VNFs to CNFs

- Moving 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 VNFs rely on kernel hacks or otherwise do not restrict themselves to just the **Linux userspace ABI**



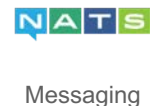
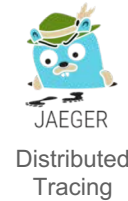
Cloud Native Computing Foundation

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

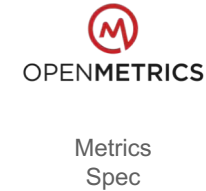
Graduated



Incubating



Sandbox



- Platinum members:



Cloud Native Trail Map

Trail Map: l.cncf.io



CLOUD NATIVE TRAIL MAP

The Cloud Native Landscape l.cncf.io has a large number of options. This Cloud Native Trail Map is a recommended process for leveraging open source, cloud native technologies. At each step, you can choose a vendor-supported offering or do it yourself, and everything after step #3 is optional based on your circumstances.

HELP ALONG THE WAY

A. Training and Certification

Consider training offerings from CNCF and then take the exam to become a Certified Kubernetes Administrator or a Certified Kubernetes Application Developer

cncf.io/training

B. Consulting Help

If you want assistance with Kubernetes and the surrounding ecosystem, consider leveraging a Kubernetes Certified Service Provider

cncf.io/kspp

C. Join CNCF's End User Community

For companies that don't offer cloud native services externally.

cncf.io/enduser

WHAT IS CLOUD NATIVE?

Cloud-native technologies, such as containers and microservices, empower organizations to develop and deploy scalable, agile applications and services in dynamic, distributed environments. By taking into account these characteristics, such systems are designed to be resilient, elastic, and loosely coupled, via manageable abstractions and declarative APIs, thereby enabling effective, reliable automation. This allows engineers to observe the applications and to safely make impactful changes, and results in processes and workflows that fully take advantage of these environments and minimize toil.

The Cloud Native Computing Foundation seeks to drive adoption of these techniques by fostering an ecosystem of open-source, vendor-neutral projects that align with these objectives, and which are portable to public, private, and hybrid clouds. We democratize the state-of-the-art patterns and practices to ensure innovations remain open and accessible for everyone.

l.cncf.io

v20180604



1. CONTAINERIZATION

- Commonly done with Docker containers
- Any size application and dependencies (even PDP-11 code running on an emulator) can be containerized
- Over time, you should aspire towards splitting suitable applications and writing future functionality as microservices

3. ORCHESTRATION & APPLICATION DEFINITION

- Kubernetes is the market-leading orchestration solution
- You should select a Certified Kubernetes Distribution, Hosted Platform, or Installer: cncf.io/ckd
- Helm Charts help you define, install, and upgrade even the most complex Kubernetes application



5. SERVICE MESH AND DISCOVERY

- CoreDNS is a fast and flexible tool that is useful for service discovery
- Envoy and Linkerd each enable service mesh architectures
- They offer health checking, routing, and load balancing



7. DISTRIBUTED DATABASE

When you need more resiliency and scalability than you can get from a single database, Vitess is a good option for running MySQL at scale through sharding.



9. CONTAINER RUNTIME

You can use alternative container runtimes. The most common, all of which are OCI-compliant, are containerd, rkt and CRI-O.



2. CI/CD

- Setup Continuous Integration/Continuous Delivery (CI/CD) so that changes to your source code automatically result in a new container being built, tested, and deployed to staging and eventually, perhaps, to production
- Setup automated rollouts, roll backs and testing

4. OBSERVABILITY & ANALYSIS

- Pick solutions for monitoring, logging and tracing
- Consider CNCF projects Prometheus for monitoring, Fluentd for logging and Jaeger for Tracing
- For tracing, look for an OpenTracing-compatible implementation like Jaeger



6. NETWORKING

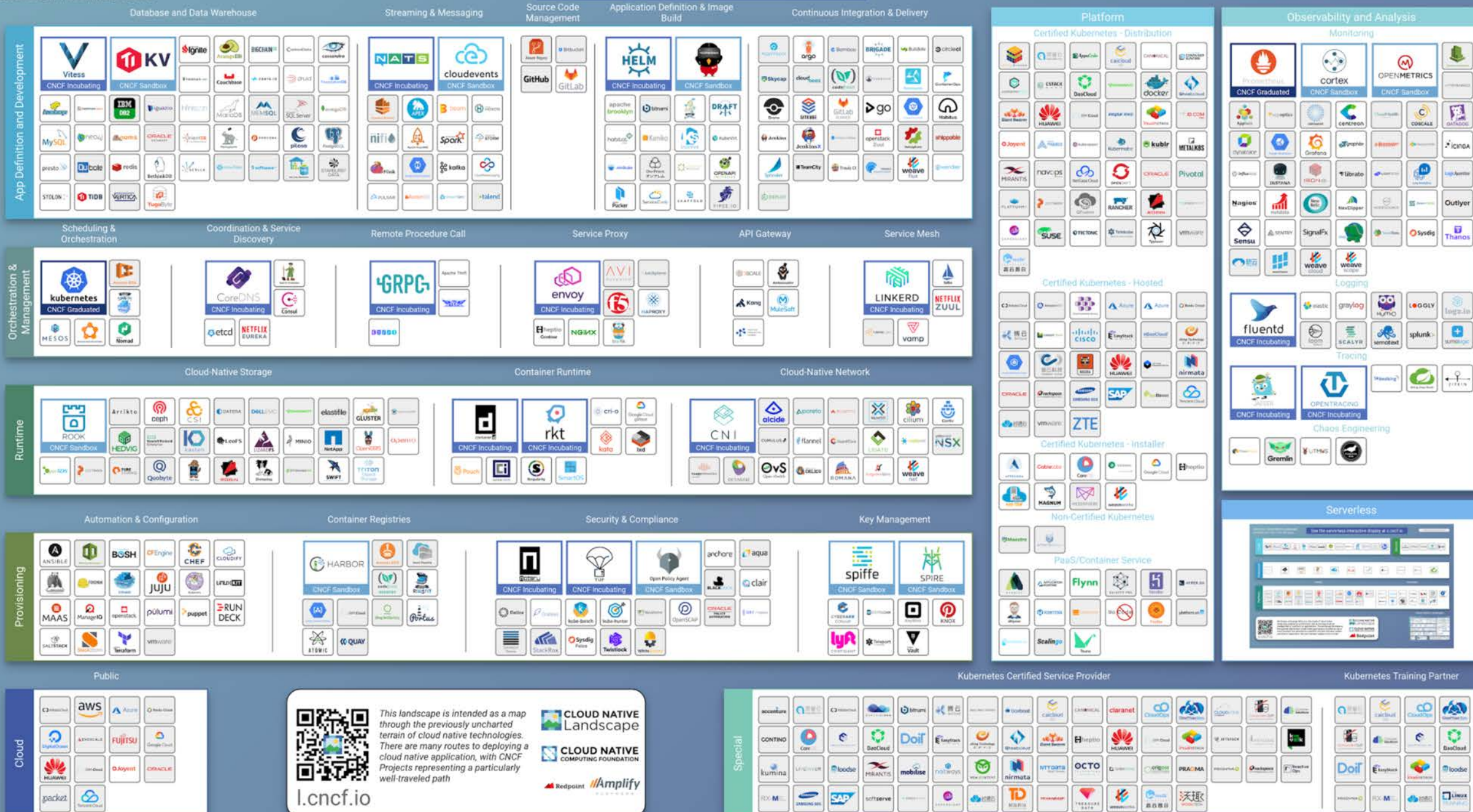
To enable more flexible networking, use a CNI-compliant network project like Calico, Flannel, or Weave Net.



8. MESSAGING

When you need higher performance than JSON-REST, consider using gRPC. NATS is publish/subscribe message-oriented middleware.







PLEASE TRY THE
INTERACTIVE
LANDSCAPE NOW:

l.cncf.io

KubeCon + CloudNativeCon

- China
 - [Shanghai](#): November 13-15, 2018
 - Sponsorships [open](#)
- North America
 - [Seattle](#): December 10-13, 2018
 - Sponsorships [open](#)
- 2019
 - [Barcelona](#): May 20-23, 2019
 - [Shanghai](#): June 26-28, 2019
 - [San Diego](#): November 18-21, 2019

