



## OPNFV Overview

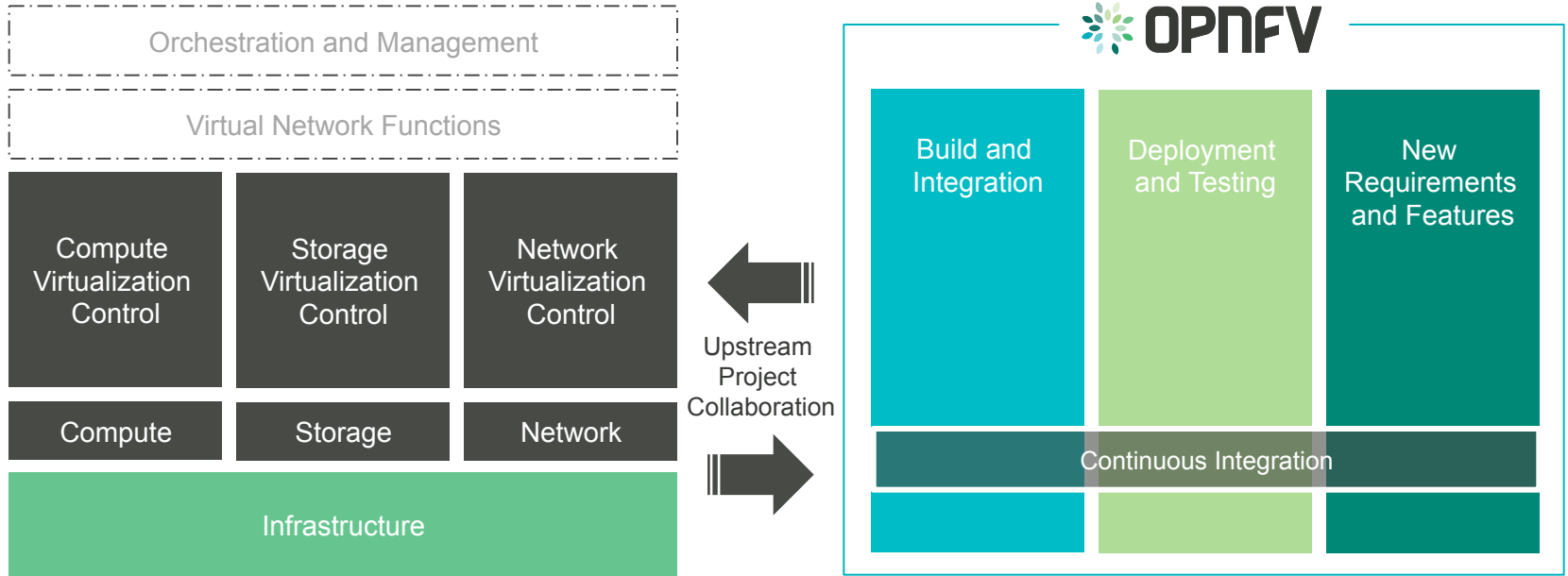
July 2015

Heather Kirksey

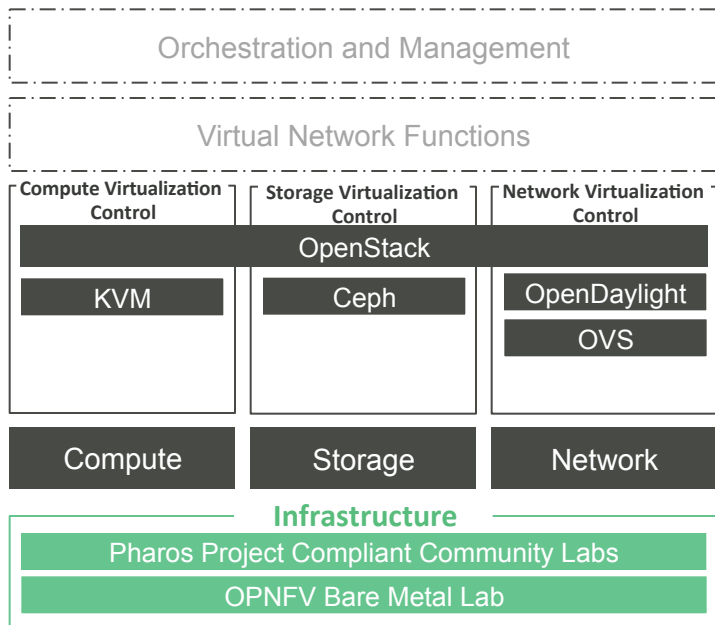


OPNFV is a carrier-grade, integrated, open source platform to accelerate the introduction of new NFV products and services.

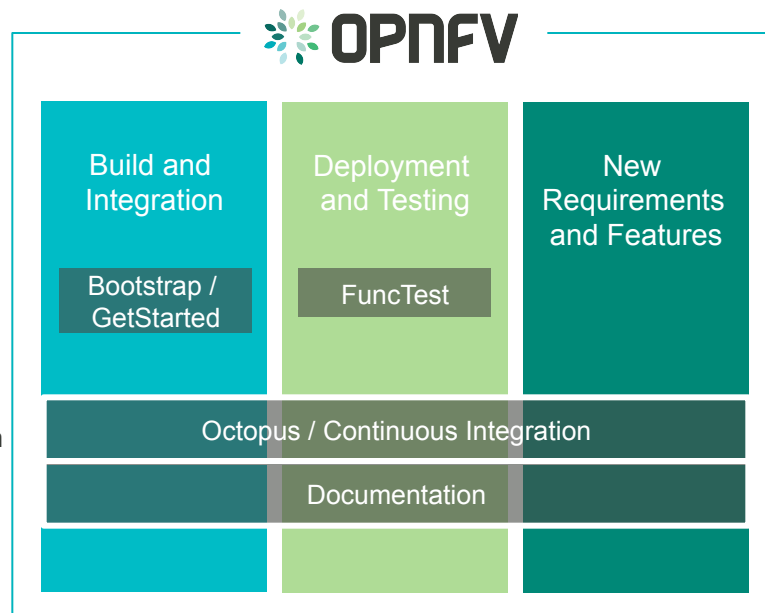
# OPNFV Platform Overview



# OPNFV Arno Overview



Upstream Project Collaboration



# Creating a starting point for OPNFV – Build and Integration

Assemble a minimal set of base infrastructure to enable VNF deployments

- Predictable performance – Deploy to bare metal
- Validated – Functional testing, multiple test environments
- Repeatable – Automatic deployment
- Carrier-class – High availability

See also: [https://wiki.opnfv.org/get\\_started/get\\_started\\_system\\_state](https://wiki.opnfv.org/get_started/get_started_system_state)

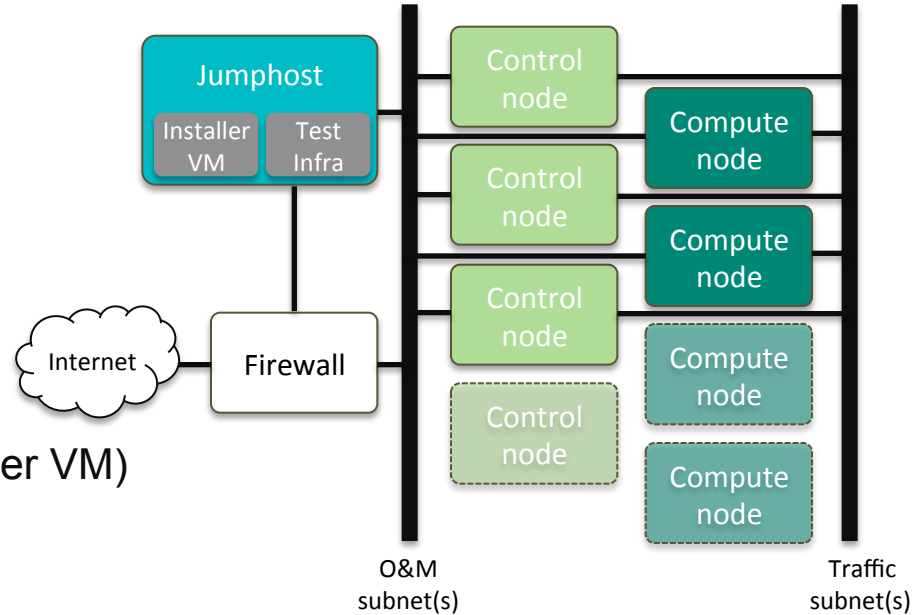
# Choose your Hardware: OPNFV Reference Infra

- Servers

- $\geq 3$  Control nodes
- $\geq 2$  Compute nodes
- 1 Provisioning node (“jumphost”)  
(Centos 7, runs test infra and installer VM)

- Services

- No additional services (e.g. DHCP) available on subnets. Installers run services that are required themselves
- Remote access for community members
- Lights-out-management (IPMI, PXE boot)

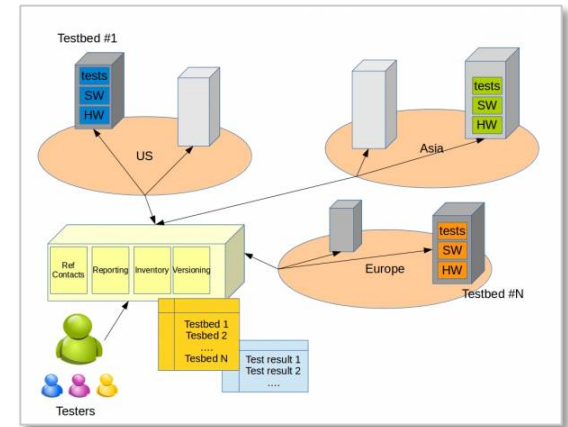


Example HW (Linux Foundation lab):  
Blade servers with 80G connectivity each (Cisco UCS-B)  
Per server:  
Intel Xeon E5-2637V3 / 3.5 GHz processor  
2 x 1.2 TB 6G SAS 10K RPM SFF disks, 32G Memory

# Build your own lab or choose an OPNFV community lab



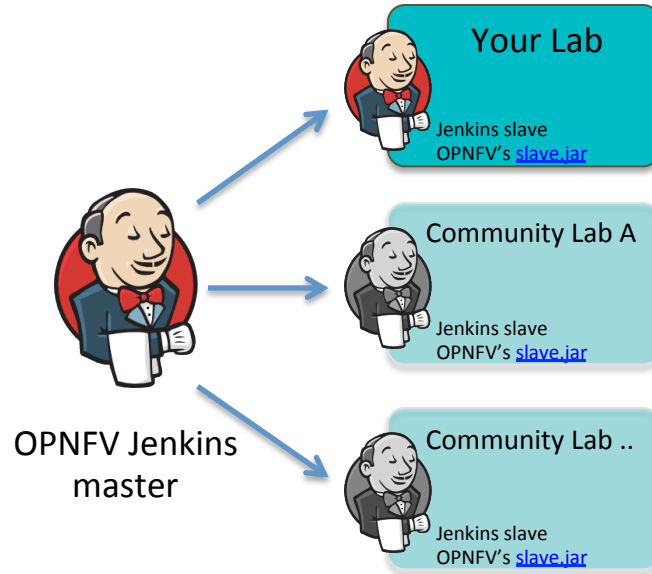
- OPNFV offers a set of community test labs
- Compliant to OPNFV hardware reference spec
- Individual hardware components for custom/advanced testing



See also: <https://wiki.opnfv.org/pharos>  
<https://www.opnfv.org/sites/opnfv/files/release/pharos-spec.arno.2015.1.0.pdf>

# Synchronize your Tool Chain with OPNFV: Continuously deploy OPNFV to your lab

- Create an OPNFV compliant lab
- Hook-up your local Jenkins to OPNFV
- Continuously deploy to your lab



Further details: <https://www.opnfv.org/sites/opnfv/files/release/opnfv-jenkins-slave-connection.arno.2015.1.0.pdf>  
Current Jenkins slaves: <https://build.opnfv.org/ci/computer/>

The screenshot shows the Jenkins web interface for the 'Nodes [Jenkins]' environment. The browser address bar shows 'https://build.opnfv.org/ci/computer/'. The interface displays a list of build jobs with columns for Name, Architektur, Zeitdifferenz, and Antwortzeit. The jobs are categorized into 'Build Warteschlange (1)', 'Build-Prozessor-Status', and 'offline'.

S	Name	Architektur	Zeitdifferenz	Antwortzeit
	an-build	Linux (amd64)	2.1 Sekunden vorgehend	253ms
	compass-build-deploy-01	Linux (amd64)	4.0 Sekunden vorgehend	254ms
	compass-build-deploy-02	Linux (amd64)	7.1 Sekunden vorgehend	251ms
	compass-build-deploy-03	Linux (amd64)	4.0 Sekunden vorgehend	248ms
	compass-build-deploy-04		N/A	5 mal keine Antwort
	compass-build-deploy-05		N/A	5 mal keine Antwort
	dell-build		N/A	5 mal keine Antwort
	ericsson-build	Linux (amd64)	15 Sekunden vorgehend	247ms
	ericsson-bsp	Linux (amd64)	3.7 Sekunden vorgehend	247ms
	goe-opnfv-docker-1	Linux (amd64)	synchro	249ms
	intel-build	Linux (amd64)	synchro	248ms
	master	Linux (amd64)	synchro	0ms
	opnfv-jump-1	Linux (amd64)	1 Minute 34 Sekunden nachgehend	247ms
	opnfv-jump-2	Linux (amd64)	synchro	51ms
	orange-build	Linux (amd64)	synchro	245ms
Daten ermittelt: 15 Minuten				15 Minuten 15 Minuten

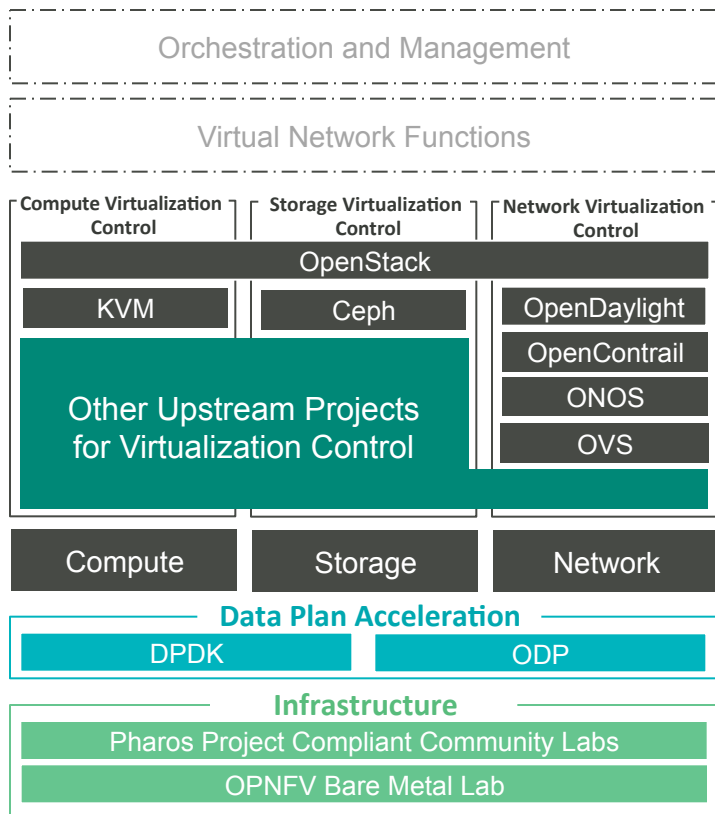


# Working with Upstream Communities






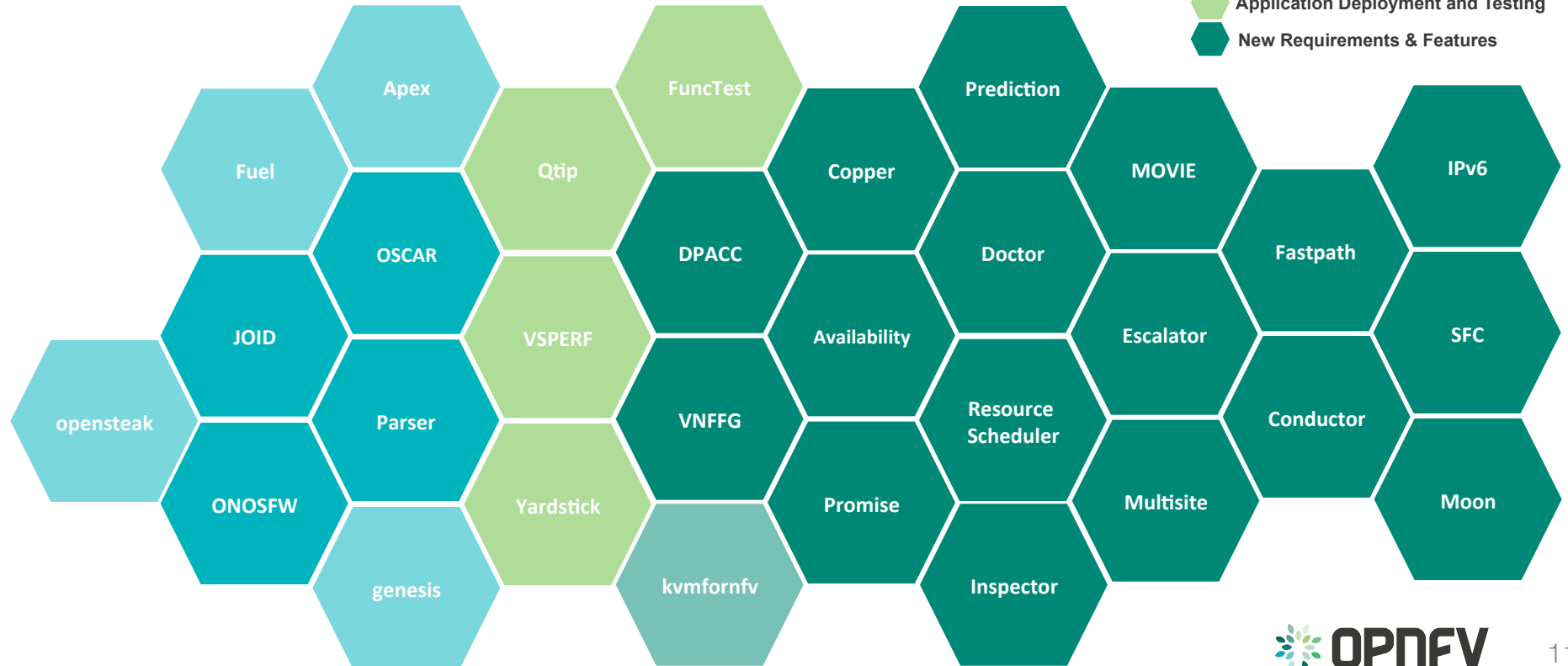
- Upstream First
  - Work within upstream, don't fork
- Learn the norms, culture, process, expectations, timelines of upstream communities and work within them
- Requirements projects identify gaps/features/blueprints based on telco use cases
  - E.g., Fault Management, Resource Management, Policy Management
- Deployment/Integration Issues
  - E.g., OpenStack and OpenDaylight integration issues
- Gaps/issues/bugs discovered via testing
  - E.g., testing in specific HA scenarios, performance testing

# Post-Arno Stack Evolution



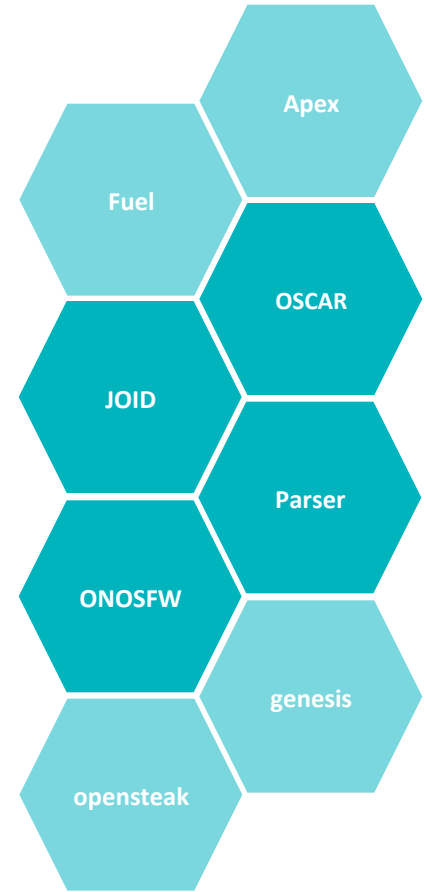
# OPNFV Project Pipeline

-  Build, Integration and Deployment
-  Application Deployment and Testing
-  New Requirements & Features



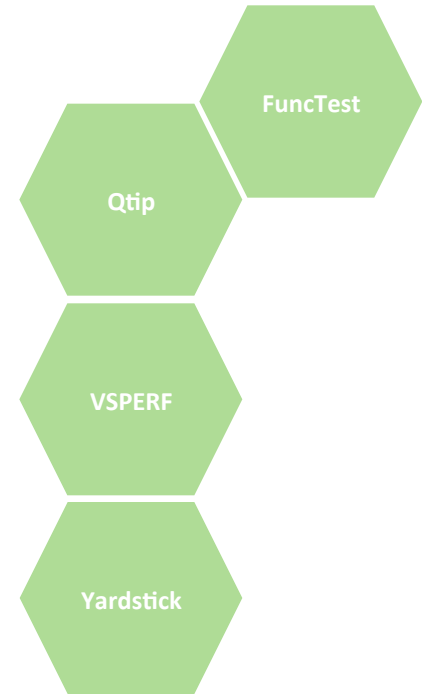
# Build Integration & Deployment

- Projects in this category include:
  - Deployment tools; Juju, Fuel, Foreman
  - Upstream component integration projects



# Application Deployment & Testing

- Projects in this category include:
  - Platform test and characterization
  - Component test and characterization
  - Application deployment test and characterization
  - Test Frameworks



# New Requirements & Features

- Projects in this category include:
  - Requirements projects defining:
    - Platform frameworks
    - Required features
    - Domain area focus activities
  - Focused upstream development projects



# Get Involved

- Website: <http://www.opnfv.org/>
- Wiki: <https://wiki.opnfv.org/>
- Arno: <https://www.opnfv.org/arno>
- Join the conversation on mailing lists and social media
- Join as a member
- Developers: Download Arno, join approved projects, propose a project, write documentation, contribute use cases, define tests, analyze requirements, build upstream relationships, contribute code, contribute upstream code, define processes, resource a community lab, answer questions, give training, evangelize





Questions?