OPNFV Overview

July 2015

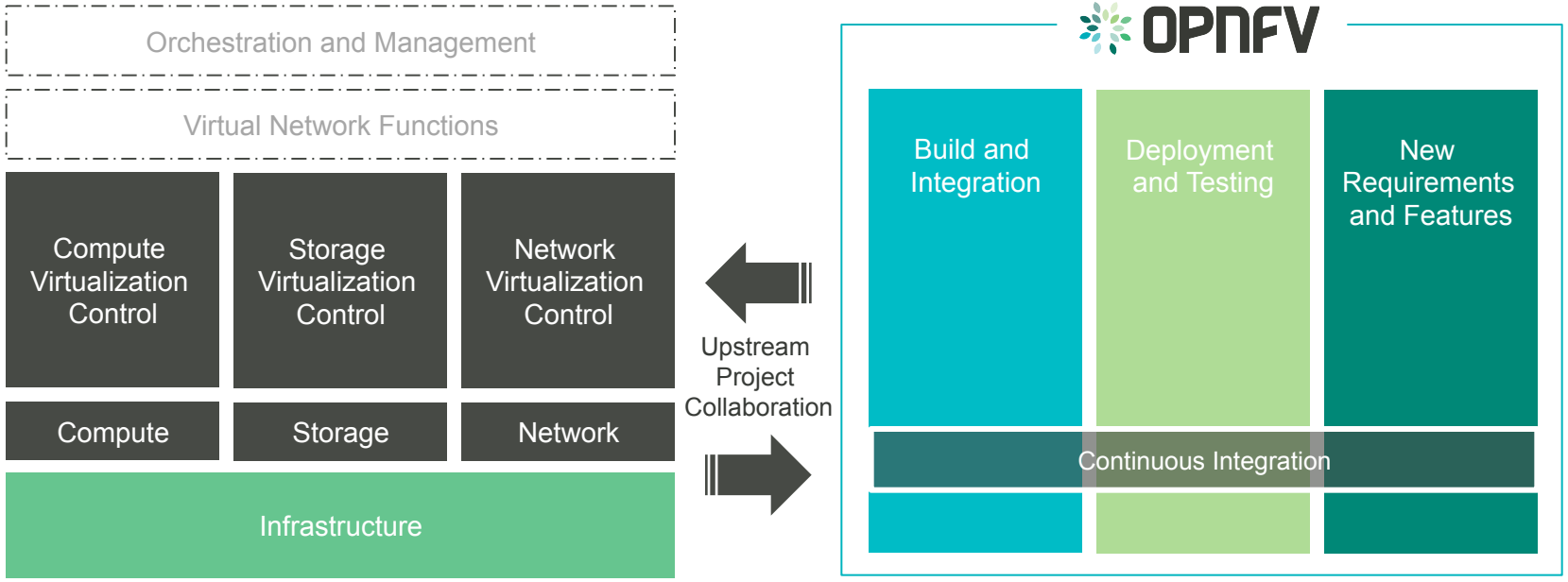
Heather Kirksey

LINUX FOUNDATION
COLLABORATIVE PROJECTS

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

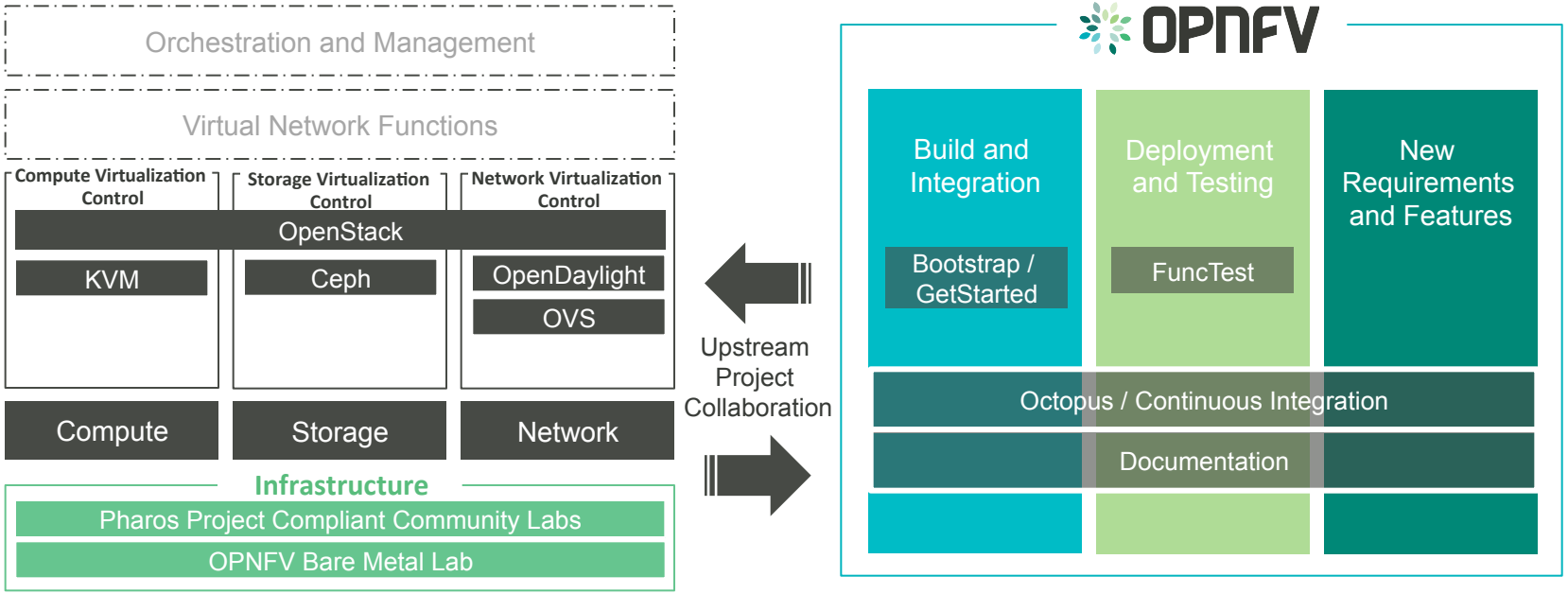
OPNFV

OPNFV Platform Overview



OPNFV

OPNFV Arno Overview



OPNFV

Creating a starting point for OPNFV – Build and Integration

Assemble a minimal set of base infrastructure to enable VNF deployments

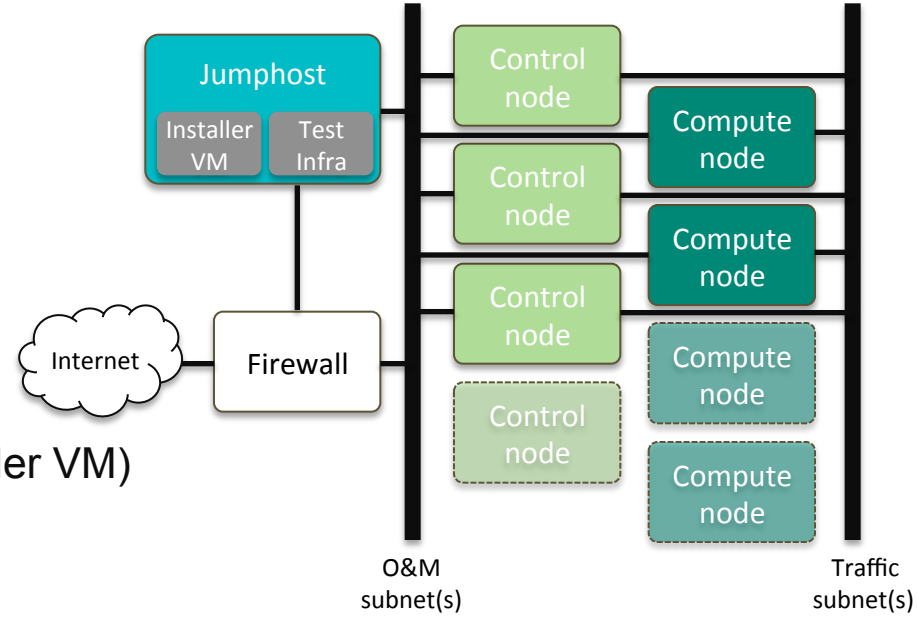
- Predictable performance – Deploy to bare metal
- Validated – Functional testing, multiple test environments
- Repeatable – Automatic deployment
- Carrier-class – High availability

See also: https://wiki.opnfv.org/get_started/get_started_system_state

OPNFV

Choose your Hardware: OPNFV Reference Infra

- Servers
  - >= 3 Control nodes
  - >= 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)



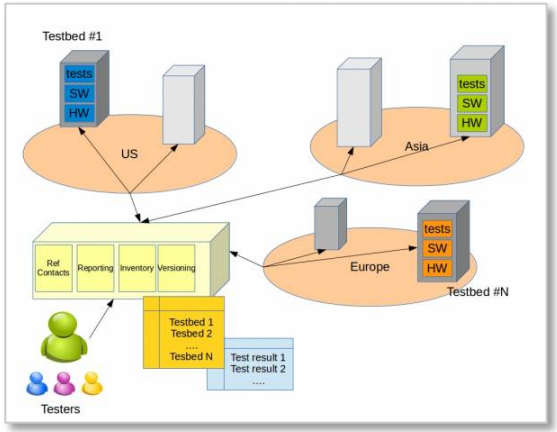
Example HW (Linux Foundation lab):
Blade servers with 80G connectivity each (Cisco UCS-B)
Per server:
Intel Xeon E5-2637V3 / 3.5 GHz processor
2 x 1.2 TB 6G SAS 10K RPM SFF disks, 32G Memory

See also: https://wiki.opnfv.org/pharos/pharos_specification

OPNFV

Build your own lab or choose an OPNFV community lab

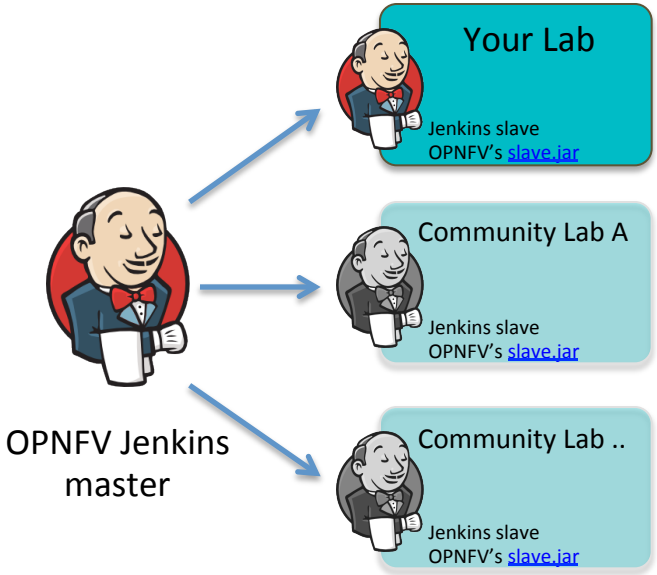
- OPNFV offers a set of community test labs
- Compliant to OPNFV hardware reference spec
- Individual hardware components for custom/advanced testing

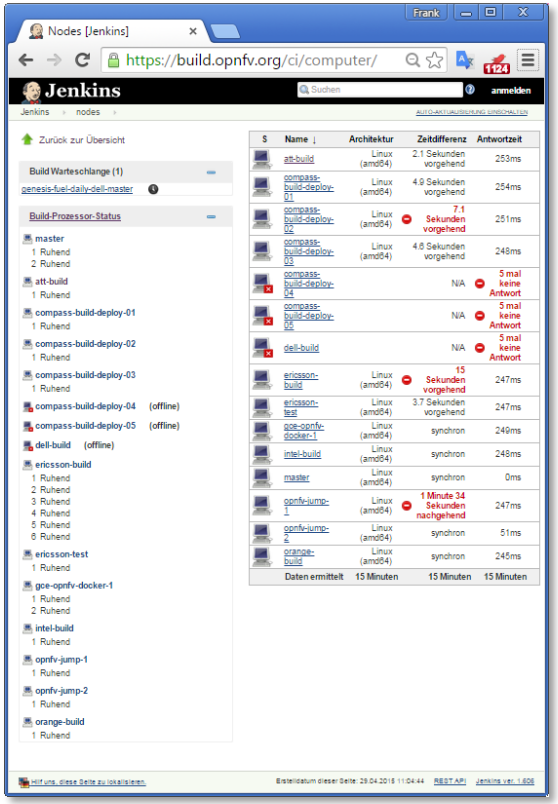


See also: https://wiki.opnfv.org/pharos
https://www.opnfv.org/sites/opnfv/files/release/pharos-spec.arno.2015.1.0.pdf

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



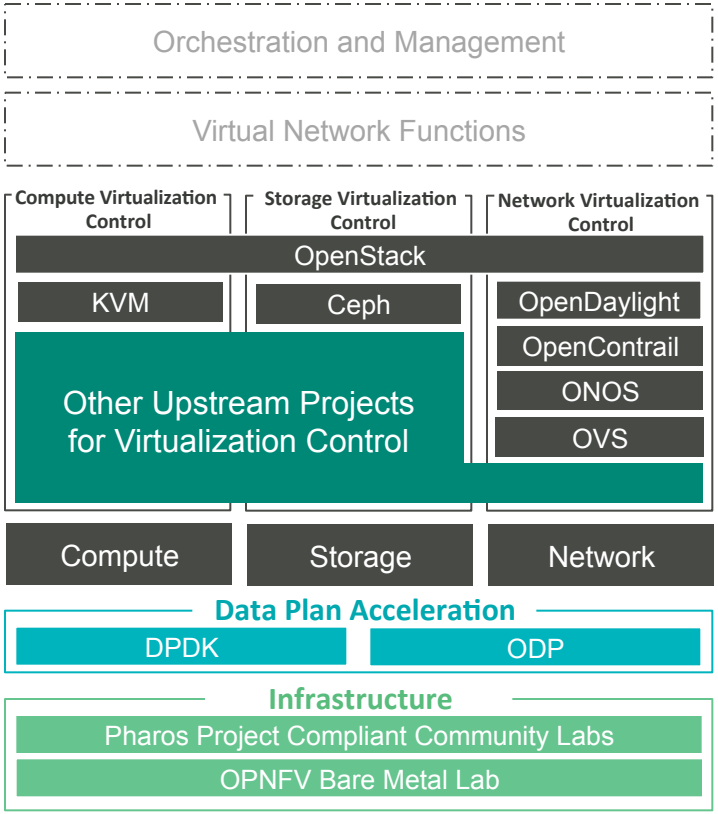


Further details: https://www.opnfv.org/sites/opnfv/files/release/opnfv-jenkins-slave-connection.arno.2015.1.0.pdf
Current Jenkins slaves: https://build.opnfv.org/ci/computer/

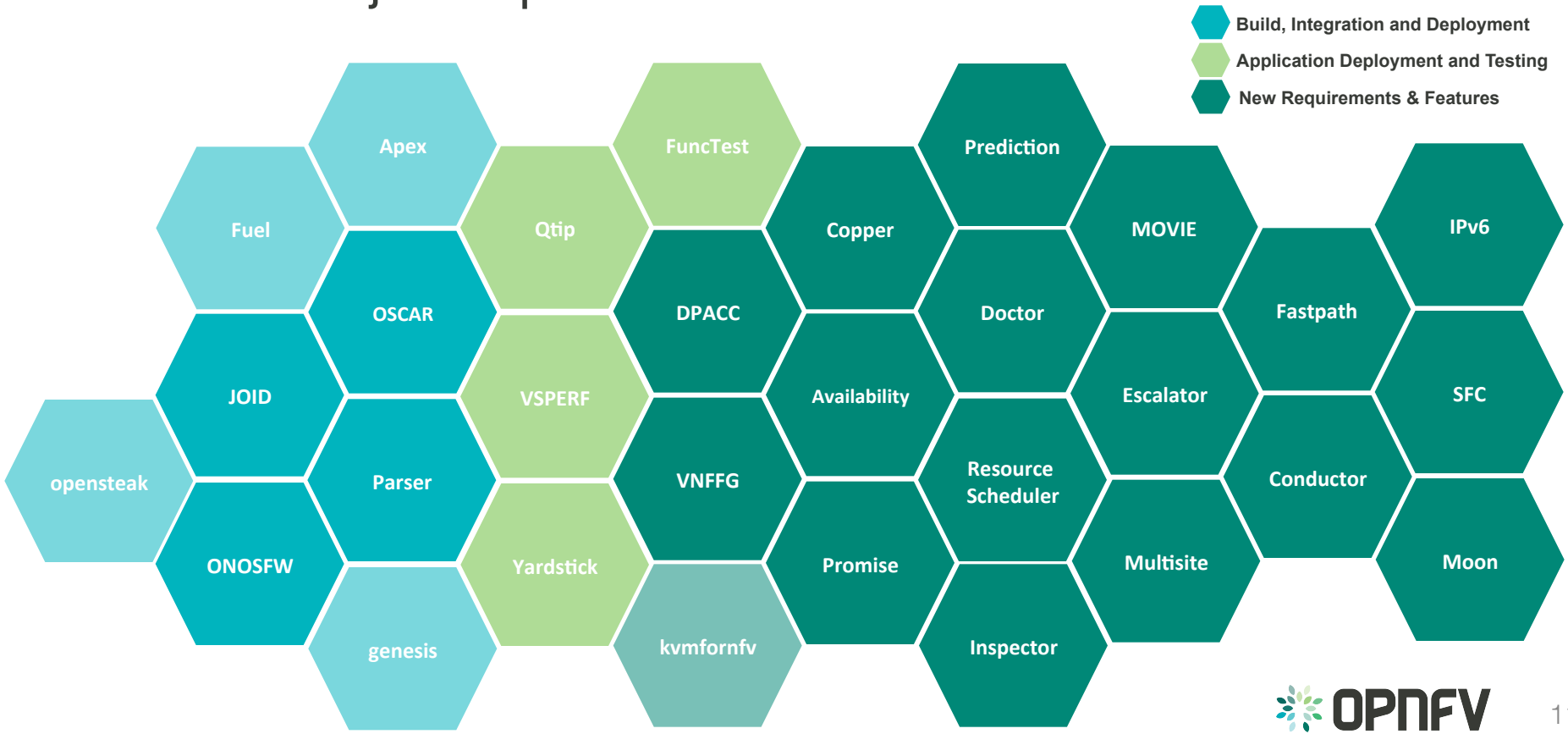
Working with Upstream Communities

- 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

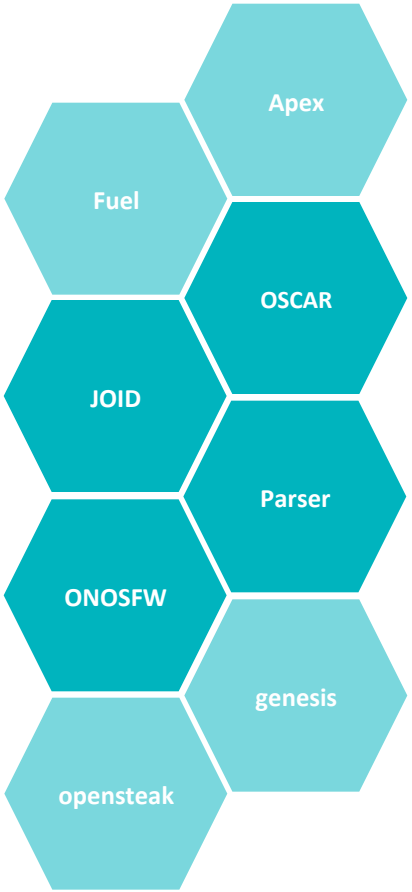


OPNFV Project Pipeline



Build Integration & Deployment

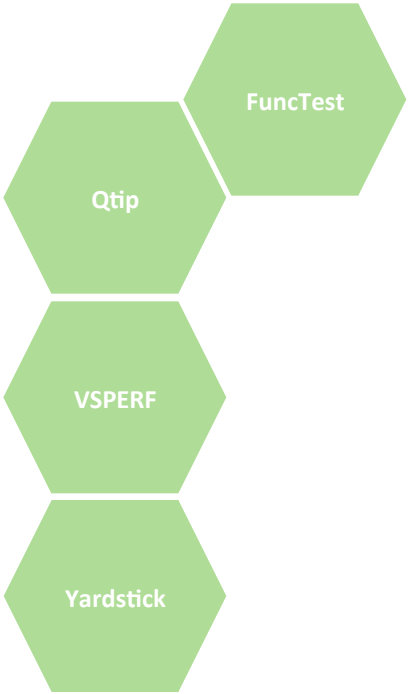
- Projects in this category include:
  - Deployment tools; Juju, Fuel, Foreman
  - Upstream component integration projects



OPNFV

Application Deployment & Testing

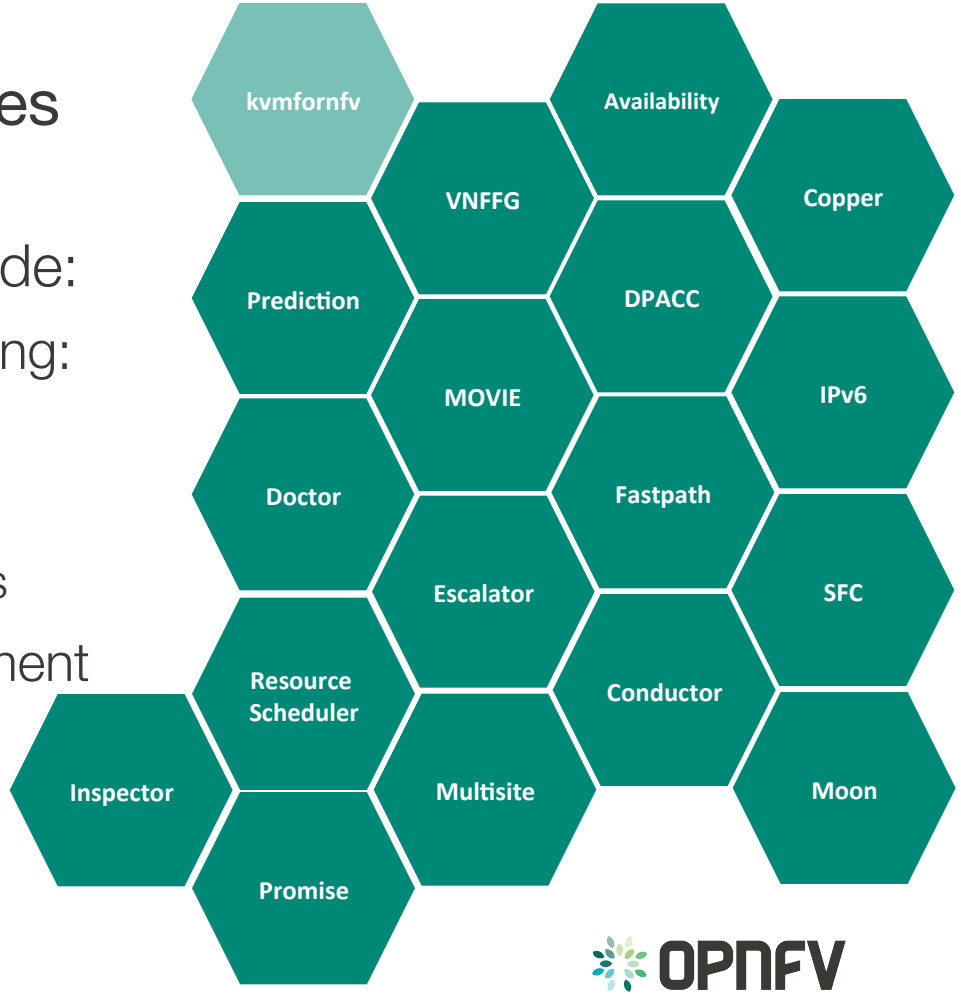
- Projects in this category include:
  - Platform test and characterization
  - Component test and characterization
  - Application deployment test and characterization
  - Test Frameworks



OPNFV

New Requirements & Features

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



OPNFV

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?

OPNFV