Accelerating Innovation with LEAN NFV

Linux Foundation Open Networking Summit
April 3, 2019 | San Jose, California

SCOTT SHENKER
SYLVIA RATNASAMY
CONSTANTINE POLYCHRONOPOULOS
Six Years After the NFV White Paper...

- **PROGRESS:**
  - A detailed architecture and large open-source codebases

- **PROBLEMS:**
  - Adoption and innovation much slower than expected

- **PARADOX:**
  - Why has this been so hard?
NFV Involves Three Basic Components

- Infrastructure Manager (VIM++):
  - Oversees the computational infrastructure

- Virtual Network Functions (VNFs):
  - Largely based on existing implementations

- NFV Manager (VNFM+NFV-O):
  - Oversees lifecycle management
Integration is the Problem

- Connected these components in complicated ways
  - Embedded NFV management in compute infrastructure
  - New features often require modifying pairwise APIs

- Makes deployment hard
  - Must change existing computational infrastructure

- Makes onboarding even harder
  - No useful guidance for how to easily integrate VNFs

- Makes innovation almost impossible
  - Because these pieces are so tightly interwoven
How Do We Fix This?

- Focus exclusively on integration
  - Provide universal integration mechanism (key-value store)
  - Do not require NFV-specific features in the VIM
  - Recognize that this is all we need to specify!

- Leave the rest of the design open for innovation
  - Components can evolve independently
  - Deployment barriers are greatly reduced

- This creates a lean, extensible, multi-vendor NFV ecosystem
Accelerating Innovation with Lean NFV

The Lean NFV Approach

Impact on the Ecosystem

Next Steps for Lean NFV
NFV Involves FOUR Basic Components

- Infrastructure Manager (VIM++):
  - Overseeing the computational infrastructure

- Virtual Network Functions (VNFs):
  - Existing code moved into a VM

- NFV Manager (VNFM+NFV-O):
  - Overseeing lifecycle management

- Key-Value (KV) Store
  - Universal point of integration
Key-Value (KV) Store

- Simple and general abstraction

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K1</td>
<td>AAA, BBB, CCC</td>
</tr>
<tr>
<td>K2</td>
<td>AAA, BBB</td>
</tr>
<tr>
<td>K3</td>
<td>AAA, DDD</td>
</tr>
<tr>
<td>K4</td>
<td>AAA, 2, 01/01/2015</td>
</tr>
<tr>
<td>K5</td>
<td>3, ZZZ, 5623</td>
</tr>
</tbody>
</table>

put(key, value)
value = get(key)
notification = watch(keys)
Key-Value (KV) Store

- Simple and general abstraction
- Many open source implementations & extensive deployment experience
Why a KV Store?

Crux of integration is allowing components to discover/exchange state

- VNF status
- configuration
- chain definition
- service load
- resource map
- events

KV store enables this in a manner that is lean and extensible
Integration with the NFV Manager

KV-based integration enables NFV MANO that is lean, extensible, and multi-vendor
Integration with the VNFs

PLACEMENT  LAUNCHING  CONFIG  CHAINING  MONITOR  SCALING  HEALING  ...

VNF, config  EMS_X, VNF_addr

VNF  EMS

KV
Integration with the VNFs

KV-based integration allows VNFs to evolve incrementally and independently
Integration with Infrastructure Managers

Specialization of Infrastructure Managers leads to a new form of lock-in!

NFV should rely on COMMON infrastructure management features
Integration with Infrastructure Managers
Integration with Infrastructure Managers

Avoid lock-in by treating NFV as just another workload
Accelerating Innovation with Lean NFV

The Lean NFV Approach

Impact on the Ecosystem

Next Steps for Lean NFV

SYLVIA RATNASAMY
CONSTANTINE POLYCHRONOPOULOS
SCOTT SHENKER
Lean NFV is Synergistic to ETSI MANO & ONAP

- Existing NFVI and VNF deployments can co-exist with Lean NFV based NFs and higher-level functions (O/M)
- Incremental integration into a Lean NFV framework
- Enabling value-add service creation around KV Store that can be instantaneously available to all VNFs and NFV infrastructure
Lean NFV: Enhanced flexibility for 5G
Core NFs and Reference Points TS 23.501

- Aligned with SBA req/repl & subscribe/notify architecture
- Support for microservices
- Highly distributed, vendor-independent framework
- Asynchronous architecture supports faster integration, less constrained NF development and new innovations (callback framework in KV Store?)
- Aligned with DevOps and cloud native strategy
Network Slicing & MEC Support in Lean NFV

- Decisive move toward distributed networks/infrastructure – key requirement for 5G
- NFV facilitates access aggregation, which in turn accelerates adoption of MEC
- By design, Lean NFV is suitable for highly distributed networks and infrastructure (MEC and Network Slicing)
  - Distributed KV Store for autonomous edge NFVI operations
  - Hierarchical organization of Keys and Values for distributed environments
  - KV store reads/updates on a need-to-know basis
Enhanced Segmentation in Network Slicing
Isolation of KV Store per slice greatly improves security

- SP KV Store: Part of common infrastructure and provisioning framework
- Tenant (Slice) KV Store: Private/shared options depending on security requirements
Accelerating Innovation with Lean NFV

SYLVIA RATNASAMY
The Lean NFV Approach

CONSTANTINE POLYCHRONOPOULOS
Impact on the Ecosystem

SCOTT SHENKER
Next Steps for Lean NFV
Summarizing Lean NFV

- Main technical points
  - Use key-value store as universal point of integration
  - Remove need for specialized VIMs

- Lean NFV is an open architecture
  - Lean, extensible, multi-vendor

- We expect many realizations of this architecture
  - Commercial, open-source, and combinations of both
Advantages of this Approach

- The minimal design that enables easy integration
  - Leaving the rest open for innovation

- Allows us to solve the problems of today...
  - Complementary to today's codebases and VNFs

- ...and the problems of tomorrow
  - Compatible with cloud-native, 5G, and beyond

- This is the path to increasing adoption and innovation
Next Steps...

- Learn the basics:
  - Read white paper at LeanNFV.org (endorsed by ten experts)
  - Visit demo in booth #605

- Explore more deeply:
  - Read more detailed technical documents (forthcoming)
  - Inspect demonstration code (forthcoming)
  - Attend future workshops (sign up at web site)