Exploring OpenDaylight

Matt Younkins

younkinsm@ou.edu
SDN, NFV and OpenDaylight

New Revenue
Open, Programmable APIs
Service Agility
Orchestration, Automation and MANO
SDN
NFV
Virtualization and Abstraction Layer
Lower Cost
Why SDN?

- New architecture with separate Control and Data planes
- Open Programmable Networks and APIs
- New business models and revenue opportunities
- Efficiency in both capital and operational expenses

Focus Area for OpenDaylight
SDN Architecture Characteristics

- Directly programmable
- Agile
- Centrally managed
- Programmatically configure
- Open standards-based and vendor-neutral
SDN Overview

SDN application

SDN application

Application layer
Application plane

SDN northbound interfaces (NBIs)

A-CPI: Application-controller plane interface

SDN controller

Control layer
Controller plane

D-CPI: Data-controller plane interface

SDN southbound interface

Network element

Network element

Network element

Infrastructure layer
Data plane
What is OpenDaylight?

• Open source project
• Modular, pluggable, and flexible controller at its core
• Implemented strictly in software
  • Contained within its own Java Virtual Machine (JVM)
• Deployable on any hardware and OS that supports Java
Who is in OpenDaylight Project?

Continuous Growth to 41 Members
Who makes products based on Open Daylight?
Is Opendaylight the only Open Source SDN Controller Available?

<table>
<thead>
<tr>
<th>Use-Cases</th>
<th>Trema</th>
<th>Nox/Pox</th>
<th>RYU</th>
<th>Floodlight</th>
<th>ODL</th>
<th>ONOS***</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Virtualization by Virtual Overlays</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>PARTIAL</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Hop-by-hop Network Virtualization</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>OpenStack Neutron Support</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Legacy Network Interoperability</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Service Insertion and Chaining</td>
<td>NO</td>
<td>NO</td>
<td>PARTIAL</td>
<td>NO</td>
<td>YES</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Network Monitoring</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Policy Enforcement</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>PARTIAL</td>
<td>YES</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Load Balancing</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Traffic Engineering</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>YES</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Dynamic Network Taps</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
</tr>
<tr>
<td>Multi-Layer Network Optimization</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Transport Networks - NV, Traffic-Rerouting, Interconnecting DCs, etc.</td>
<td>NO</td>
<td>NO</td>
<td>PARTIAL</td>
<td>NO</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
</tr>
<tr>
<td>Campus Networks</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>PARTIAL</td>
<td>NO</td>
</tr>
<tr>
<td>Routing</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
OpenDaylight Tools and Paradigms

• Java interfaces
  • for event listening, specifications and forming patterns

• Maven -
  • Build and dependency management

• OSGi -
  • Backend container framework that allows dynamically loading bundles

• Karaf -
  • OSGi based runtime
What is an Application Programming Interface (API)?

• set of rules ('code') and specifications that software programs can follow to communicate with each other.
What is a Representational State Transfer (REST) API?

• A REST API is an API in a specific architectural style
  • originally communicated by Roy Fielding in his doctoral dissertation
    • http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
What defines a ‘RESTful’ API?

- Six constraints
  - Start with Null Style
  - Client-Server
  - Stateless (Server)
  - Cache
  - Uniform Interface
  - Layered System
What distinguishes REST?

• Emphasis on uniform interface between components
  • Four constraints
    • identification of resources
    • manipulation of resources through representations
    • self-descriptive messages
    • hypermedia as the engine of application state.
Why a ‘RESTful’ API

• Principle of generality
  • Simplifies overall system architecture
  • Visibility of interactions improved
Questions? Thoughts?

Matt Younkins
younkinsm@ou.edu
Extra Slides
Legend

AAA: Authentication, Authorization & Accounting
AuthN: Authentication
BGP: Border Gateway Protocol
COPS: Common Open Policy Service
DLUX: OpenDaylight User Experience
DDoS: Distributed Denial Of Service
DOCSIS: Data Over Cable Service Interface Specification
FRM: Forwarding Rules Manager
GBP: Group Based Policy
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
PCEP: Path Computation Element Communication Protocol
PCMM: Packet Cable MultiMedia
Plugin2OC: Plugin To OpenContrail
SDNI: SDN Interface (Cross-Controller Federation)
SFC: Service Function Chaining
SNBI: Secure Network Bootstrapping Infrastructure
SNMP: Simple Network Management Protocol
TTP: Table Type Patterns
VTN: Virtual Tenant Network

OpenDaylight APIs (REST)

Base Network Service Functions
- Topology Manager
- Stats Manager
- Switch Manager
- FRM
- Host Tracker
- VTN Manager
- OVSDB Neutron
- Plugin2OC
- LISP Service
- L2 Switch
- SDNI Aggregator

Service Abstraction Layer (SAL)
(Plugin Manager, Capability Abstractions, Flow Programming, Inventory, etc.)

Controller Platform

GBP Renderers
- OpenFlow
- OVSDB
- NETCO
- PCMM/COPS
- SNBI
- LISP
- BGP
- PCEP
- SNMP
- Plugin2OC

Southbound Interfaces & Protocol Plugins

Data Plane Elements
Additional Virtual & Physical Devices

OpenFlow Enabled Devices
Open vSwitches
What is Karaf?

- Small OSGi based runtime
- Lightweight container
  - various components and applications can be deployed
What is OSGi (Open Service Gateway Initiative)?

• Java framework for developing and deploying modular software programs and libraries

• Two components
  • Specification for modular components called bundles
  • Java Virtual Machine (JVM)-level service registry
What are the drawbacks of REST?

• Uniform interface degrades efficiency
  • information transferred in a standardized form rather than form specific to an application's needs