OpenDaylight Architecture

ONF Member Work Day
February 12th, 2015

Colin Dixon, @colin_dixon
TSC Chair, OpenDaylight
Principal Engineer, Brocade
What is OpenDaylight

OpenDaylight is an **Open Source Software** project under the **Linux Foundation** with the goal of furthering the adoption and innovation of **Software Defined Networking (SDN)** through the creation of a common industry supported platform.

<table>
<thead>
<tr>
<th>Code</th>
<th>Acceptance</th>
<th>Community</th>
</tr>
</thead>
</table>
| To create a robust, extensible, open source code base that covers the major common components required to build an SDN solution | To get broad industry acceptance amongst vendors and users  
  - Using OpenDaylight code directly or through vendor products  
  - Vendors using OpenDaylight code as part of commercial products | To have a thriving and growing technical community contributing to the code base, using the code in commercial products, and adding value above, below and around. |
Base Network Service Functions

- Topology Manager
- Stats Manager
- Switch Manager
- Fwding Rules Mgr
- Host Tracker

API-Driven Service Abstraction Layer (AD-SAL)

- Clustering
- Shared Data Models
- RPCs and Notifications

Model-Driven Service Abstraction Layer (MD-SAL)

- DOCSIS Service
- LISP Service
- SDNI Aggregator
- Service Flow Chaining
- GBP Service
- L2 Switch

Controller Platform and Services

- OpenStack Neutron Service
- OVSDB
- VTN
- Plugin2OC

Abstraction Layers

- SDNI Aggregator
- Service Flow Chaining
- GBP Service
- L2 Switch

Northbound/REST APIs

- Northbound/REST APIs

Authentication

- Authentication
- Authorization

Applications and Orchestration Services

- Applications and Orchestration Services

Legend

- AAA: Authentication, Authorization & Accounting
- AuthN: Authentication
- BGP: Border Gateway Protocol
- COPS: Common Open Policy Service
- DLUX: OpenDaylight User Experience
- DOCSIS: Data Over Cable Service Interface Specification
- 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)
- SNMP: Simple Network Management Protocol
- TTP: Table Type Patterns
- VTN: Virtual Tenant Network

Core service wiring and dependencies

App/service-specific wiring and dependencies

Released October, 2014
1.87M+ lines of code, 28 Projects, 256 Contributors
Core Architecture

- MD-SAL
  - Model-Driven Service Abstraction Layer

- Components interact via YANG models
  - RPCs
  - Notifications
  - Data Store
Core Architecture

- Models are layered to increase abstraction
  - Path programming model
    - (over)
  - Flow programming model
    - (over)
  - OpenFlow model
- Common models provide unifying information to write many apps
  - Topology, Inventory, Neutron, etc.
Example YANG Model

- Network Topology
  - List of Nodes
  - List of Links
  - Links and Nodes can be “extended” later
  - Can specify constraints

```yango
container network-topology {
  description "...";
  key "topology-id";
  leaf topology-id {
    type topology-id;
    description "...";
  }

  list node {
    description "...";
    key "node-id";
    uses node-attributes;
  }

  list link {
    description "...";
    key "link-id";
    uses link-attributes;
  }
}
```
YANG Models as an Architecture

- **Good**
  - Easy to augment common models
    - Some apps just work
  - Modeling is great
    - Easily defined APIs between projects
    - You can do real work
  - Native NETCONF
  - Good Java tooling

- **Harder**
  - Interaction via shared data models is hard
    - No status codes
  - RPCs fix that, but can be less flexible
  - YANG isn’t perfect
    - No recursive self-inclusion
    - Less tools than alternatives
Architecture in Evolution
(OVSDDB/Neutron)

This project is a monolithic combination of:
(1) a network virtualization layer that is “hard-wired” to Neutron above and OVS below as well as (2) an OVSDDB protocol library.

$(NEW NAME FOR OVSDDB NETVIRT)
Network virtualization layer that is still “hard-wired” to Neutron above, but now uses more general APIs below.

Tunnel Mgmt
Traffic Direction

OVSDDB Plugin
NETCONF
OpenFlow +Nicira Extns?

Many h/w- and v-switches
Architecture in Evolution (OVSDDB/Neutron)

OpenStack
- Neutron (REST)
- Neutron (YANG)

REST/YANG Adapter
- Neutron ODL Policy Adapter

Higher-Level Network Virtualization API

$(NEW NAME FOR OVSDDB NETVIRT)
Network virtualization layer that now uses the more general APIs above and below.

Tunnel Mgmt - Traffic Direction

OVSD Plugin NETCONF OpenFlow
+Nicira Extns?

Relevant Southbound Protocol

Many h/w- and v-switches

www.opendaylight.org
Get Involved

A community-led and industry-supported open source platform to advance Software-Defined Networking (SDN) and Network Functions Virtualization (NFV).

- Pull the code and review documentation at [wiki.opendaylight.org](http://wiki.opendaylight.org)
- Connect with active developers in the community on the #opendaylight IRC channel at freenode.net [webchat.freenode.net](http://webchat.freenode.net)
- Join the conversation through [lists.opendaylight.org](http://lists.opendaylight.org) and [ask.opendaylight.org](http://ask.opendaylight.org)
- Propose a new project at [wiki.opendaylight.org/view/Project_Proposals:Main](http://wiki.opendaylight.org/view/Project_Proposals:Main)