OpenDaylight Update

November 2013
Jan Medved, Reinaldo Penno
Agenda

- Intro to Opendaylight
- Hydrogen Release
- Major Feature: Model Driven Service Abstraction Layer (SAL)
- Asks from Netconf and Yang
What is OpenDaylight

- An OpenSource Project to build an ecosystem of OpenSource SDN software
  - Multi-project
  - Multi-vendor
OpenDaylight Project Goals

- **Code**: To create a robust, extensible, open source code base that covers the major common components required to build an SDN solution

- **Acceptance**: To get broad industry acceptance amongst vendors and users

- **Community**: 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.
Broad Industry Support for OpenDaylight
First Code Release “Hydrogen”
## Projects in the “Hydrogen” Release

<table>
<thead>
<tr>
<th>Project</th>
<th>Description</th>
<th>Originator (others)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>Modular, extensible, scalable, and multi-protocol SDN controller based on OSGi</td>
<td>Cisco (IBM, RedHat, NEC, etc.)</td>
</tr>
<tr>
<td>Virtual Tenant Network</td>
<td>Multi-tenant network virtualization application using OpenFlow</td>
<td>NEC</td>
</tr>
<tr>
<td>YANG Tools</td>
<td>Java-based NETCONF and YANG tooling for OpenDaylight projects</td>
<td>Cisco</td>
</tr>
<tr>
<td>OpenFlow Protocol Library</td>
<td>OF 1.3 protocol library implementation</td>
<td>Pantheon (IBM, Cisco, Ericsson)</td>
</tr>
<tr>
<td>OpenFlow Plugin</td>
<td>Integration of OpenFlow protocol library in controller SAL</td>
<td>Ericsson, IBM, Cisco</td>
</tr>
<tr>
<td>Affinity Metadata Service</td>
<td>APIs to express workload relationships and service levels</td>
<td>Plexxsi</td>
</tr>
<tr>
<td>Defense4All</td>
<td>DDoS detection and mitigation framework</td>
<td>Radware</td>
</tr>
<tr>
<td>BGP-LS/PCEP</td>
<td>Support for traffic engr with BGP-LS (BGP protocol library and topology model) and PCEP (path programming model)</td>
<td>Cisco</td>
</tr>
<tr>
<td>OVSDB</td>
<td>OVSDB configuration and management protocol support (e.g., for Open vSwitch and other OVSDB servers)</td>
<td>Univ. of Kentucky</td>
</tr>
<tr>
<td>LISP Flow Mapping</td>
<td>LISP (locator/identifier separation protocol) plugin, LISP mapping service (can be used to implement virtual networks)</td>
<td>ConteXtream</td>
</tr>
<tr>
<td>SNMP4SDN</td>
<td>SNMP protocol support; APIs to manage commodity Ethernet switches</td>
<td>Industrial Technology Research Inst.</td>
</tr>
<tr>
<td>Open DOVE</td>
<td>Multi-tenant network virtualization based on overlays, including ctrl plane and OVS-based data plane</td>
<td>IBM</td>
</tr>
</tbody>
</table>
Moving to Model-Driven SAL

SAL: Service Abstraction Layer
- AD-SAL: “API-Driven” SAL
- MD-SAL: “Model-Driven” SAL
Moving to Model-Driven SAL

Applications

JAVA SAL APIs (Generated)

MD-SAL

JAVA SAL APIs (Generated)

Network Elements

www.opendaylight.org
Request Routing (App->NE)

Path: "/restconf/datastore/nodes/node[id="NE1"]/mounted-data/f1"

Module code:
```
module node-feature-inventory {
  prefix nf;
  import opendaylight-inventory {prefix inv};
  import yang-ext {prefix ext};
  import mount {prefix mount};
  augment "/inv:nodes/inv:node" {
    ext:context-instance "node";
    ext:augment-identifier "netconf-node";
    mount:mountpoint "mounted-data" {
      mount:subtree "/";
    }
  }
}
```

Models

Routing Table:
- NE1
- NE2
- NEn

Network

Controller (Container Instance)

Application

Models

RESTCONF

Inventory
Yang Models used in ODL

- Yang Extensions
- Common Base Types:
  - IETF (ietf-types, inet-types), Common base flow types, L2 types, IEEE754 floating point types
- Services:
  - Topology, Inventory, Flow Programming Services, Affinity Service
- Protocols:
  - OpenFlow, BGP/BGP-LS, PCEP

Standardization Asks

- Controller’s NB API:
  - Address Space: the whole network as opposed to an NE
  - Policies (access, address space remaps, views, request routing)

- Application requirements:
  - I2RS (defining requirements)

- NETCONF:
  - RESTConf
  - Efficient binary encoding (e.g. draft-varga-netconf-exi-capability)
  - JSON encoding
  - Query language
Standardization Asks (Yang)

- Yang ODL extensions (e.g. request routing, Java API generation)
- Yang programming language bindings (Java, Python, ...)
- Standard Service Models:
  - Example: VPNs, DDoS, QoS, Topology, ...
- Standard Device Models:
  - Example: IP, ACL, RIB,
  - WADL/RSDL for RESTCONF clients
- Yang as IDL
Thank you
Resources

- More information and to join:
  - [wiki.opendaylight.org](http://wiki.opendaylight.org)
- Keep informed and join the conversation
  - IRC: #opendaylight on Freenone
  - Open mailing lists: [lists.opendaylight.org](http://lists.opendaylight.org)
  - [@openDaylightSDN](https://twitter.com/openDaylightSDN)
  - [#OpenDaylight](https://twitter.com/hashtag/OpenDaylight)