A New Way to Achieve Rough Consensus and (a lot of) Running Code

Colin Dixon, PhD, Principal Engineer, Brocade
colin@colindixon.com (http://colindixon.com)
@colin_dixon

Some content borrowed from David Meyer, Kyle Mestery, Anees Shaikh, Luis Gomez, and likely others
Why am I here giving this talk?

“standards bodies are too slow”

“While the pace at which standards are written hasn’t changed in many years, the pace at which the real world adopts software has become orders of magnitude faster.”

Who cares and why?

- Networking people
- Because Networking is all about standards
  - (until recently) not about software
- (Open Source) software has started to change that
  - Open vSwitch
  - OpenStack
  - OpenDaylight
  - CloudStack
Open source is not new…

…but it *is* new to networking
Where I’m coming from

- OpenDaylight Contributor
  - Dixon-Erickson Merged Controller Plan
  - Controller, Table Type Patterns, etc.

- Open Networking Foundation (ONF)
  - Forwarding Abstractions Working Group (FAWG)
  - Helped created Table Type Patterns…

  …which I’m now implementing in OpenDaylight
Are Standards Dead?

No.

- Interoperability is more than “here, just run this code”
  - Really!

- All? Successful Open Source projects rely on standards to get their jobs done

- Real Question:
  - What can open source and standards do together?
What it is really about

- "We need good, honest, thorough collaboration that results in both documentation and running code.”
  —Benson Schliesser

- This will involve both open source and standards

- The interplay is going to be interesting and important
  - Somewhat uncharted territory
Agenda

- What is OpenDaylight
- Some things I’ve learned
- Open Source in support of Standards (and vice versa)
- Where OpenDaylight is Going
Networks are Evolving

Applications and services that ride on, optimize, exploit, and manage the network

Software Defined Networking is a software layer that makes the network more programmable, agile, and efficient

A network that provides the physical connectivity

Focus Area for OpenDaylight
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 SDN solutions | To get broad industry acceptance amongst vendors and users  
• Users consuming OpenDaylight code directly or through vendor products  
• Vendors using OpenDaylight code as part of commercial products | To have a thriving technical community contributing to the code base, using the code in commercial products, and adding value above, below and around. |
What is OpenDaylight building?

- An evolvable SDN platform capable of handling diverse use cases and implementation approaches
  - Common abstractions for people to program
    - “Northbound” Interfaces
  - Southbound “drivers”, e.g., OpenFlow, OVSDB, BGP-LS
  - Intermediation between north and south
  - Programmable Network services
  - Network Applications
  - Whatever else we need to make it work
Project Framework

- Network applications, orchestration, and services
- Controller platform
- Southbound interfaces & protocols
- Data plane elements (virtual switches, physical device interfaces)

Service Abstraction Layer (SAL)

OpenDaylight APIs (REST)

network service functions
platform services
extensions

protocols (ONF, IETF, ...)
standard specific interfaces

OpenFlow
RESTCONF APIs

Base Network

Service Functions

GUI

REST APIs

AD-SAL

VTN Coordinator

VTN: Virtual Tenant Network

doDMC: Open Dove Management console
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch Data Base protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Protocol
SNMP: Simple Network Management Protocol

OpenFlow Enabled Devices

Open vSwitches

Additional Virtual & Physical Devices

OF: OpenFlow
D4A: Defense for All
VTN: Virtual Tenant Network
oDMC: Open Dove Management console
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Protocol
SNMP: Simple Network Management Protocol
Major Architectural Feature: Service Abstraction Layer (SAL)

Hard SAL (AD-SAL)

[Diagram of HARD SAL]


MD SAL
Who is OpenDaylight? (Members)

PLATINUM MEMBERS

GOLD MEMBERS

SILVER MEMBERS
Who is OpenDaylight? (Really)

- Like any Open Source Project, OpenDaylight primarily consists of those who show up to do the work.
  - Currently commits from over 180 contributors from many different organizations (and unaffiliated individuals)

- Running around 100 commits per week
  - **30 Days:** 369 commits, 50 contributors
  - **12 Months:** 6415 commits, 181 contributors

- Strong integration and testing community
  - This stuff really matters

http://www.ohloh.net/p/opendaylight
The Hydrogen Simultaneous Release

- First release of OpenDaylight on February 3rd, 2014
  - Codename: Hydrogen
  - 15 different projects
  - Lots of integration and testing

- Several “editions” to group related functionality
  - base, virtualization, service provider
  - virtualization edition provides OpenStack integration

- We all learned A LOT
OpenDaylight uses all the standards

- OpenFlow
- LISP
- NETCONF
- RESTCONF
- OVSDB
- TTPs
- SNMP
- BGP
- PCEP
- YANG
- ...
- ALL
- OF THEM!!!
OpenStack Integration

- OpenDaylight exposes a single common OpenStack Service
  Northbound
  - API matches Neutron precisely
  - multiple implementations to choose from

- Shipped in OpenStack Icehouse/OpenDaylight Hydrogen
  - Enhancements coming
Agenda

- What is OpenDaylight
- Some things I’ve learned
- Open Source in support of Standards (and vice versa)
- Where OpenDaylight is Going
Key Learnings

- **Community building** is a core objective
  - In fact, innovation through collaboration is one of the most powerful features of open source development

- **Code** is the coin of the realm

- **Engineering systems** are as important as artifacts

**Putting this all Together ➔**

http://www.sdncentral.com/education/david-meyer-reflections-opendaylight-open-source-project-brocade/2014/03/
Trend: Engineering artifacts are no longer the source of sustainable advantage and/or innovation.

Perhaps surprisingly, the “hyper-scale” and open source communities have taught me that actual artifacts (in our case network applications as well as HW/SW) are ephemeral entities and that the only source of sustainable advantage/innovation consists of

- Engineering Systems
- Culture
- People/Process

http://en.wikipedia.org/wiki/Aeroelasticity - Flutter
Factories vs. Babies

- “Most vendors develop product like an overly anxious parents making a baby. There is a lot preparation and planning and once the baby is “born” the product requires ongoing attention to reach maximum potential.”
- “By comparison, … has organized itself as a product factory. Each product is the result of a unified production line and the next product or feature is just a year or two away. Each product builds on the previous product.”
  - Even faster in open source software.
  - Networking hasn’t seen this yet.

http://etherealmind.com/difference-arista-competitors-factories-babies/
Transparency

- Transparency matters

- When there are disagreements in the community
  - Transparency makes everyone feel heard
  - Transparency makes sure the community does not fracture

- OpenDaylight is transparent to the extreme
  - All calls, mailing lists, wikis, etc. are open to the public
  - Even the technical steering committee calls
Agenda

- What is OpenDaylight
- Some things I’ve learned
- Open Source in support of Standards (and vice versa)
- Where OpenDaylight is Going
“We reject: kings, presidents and voting. We believe in: rough consensus and running code.”

—David Clark, 1992
Rough Consensus and Running Code

- What about Open Source?

- Open Source is all about running code.

- When done right, it’s also about rough consensus
  - Community
  - Transparency
  - Governance
Standards Supporting Open Source Global YANG Repository

IETF
- IETF Working Groups
- RESTCONF / NETCONF
- RFC Process Standardized YANG Models

Open Source Community
- YANG GitHub Channel
- YANG Models
- Vendor-Proprietary Yang Models

OpenDaylight
- AUTO-Generated API YANG Tools
- Model-Driven MD-SAL
- Manually Generated API
- API-Driven AD-SAL

https://github.com/YangModels/yang

www.opendaylight.org
typedef ip-version {
  type enumeration {
    enum unknown {
      value "0";
      description
      "An unknown or unspecified version of the Internet protocol.";
    }
    enum ipv4 {
      value "1";
      description
      "The IPv4 protocol as defined in RFC 791.";
    }
    enum ipv6 {
      value "2";
      description
      "The IPv6 protocol as defined in RFC 2460.";
    }
    description
    "This value represents the version of the IP protocol.
    In the value set and its semantics, this type is equivalent to the InetVersion textual convention of the SMIv2.";
    reference
    "RFC 791: Internet Protocol
    RFC 4001: Textual Conventions for Internet Network Addresses";
  }
}
Example YANG Model

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;
}

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

- ONF’s SampleTap app for OpenDaylight
- ONF also publishes openflow.h
  - C structs to complement the spec
  - Imported directly into open source, e.g., OVS
- I’m sure many other examples I’m missing
The Interplay is Where It’s Interesting

- Open source proves out standards
  - Fast way to get feedback
  - What people use and what they don’t

- Open source can make standards more useful
  - Libraries on top of standards, e.g., OpenFlowJ
  - Showing behavior as well as specs
  - Providing useful combinations of standards

- Standards make open source more interoperable
  - e.g., YANG for data modeling
Interplay in OpenDaylight

- Combination of Standards
  - OVSDB to establish tunnels
  - OpenFlow to route traffic in to/out of tunnels

- Feedback loop with standards
  - Northbound API group in ONF
  - Table Type Patterns in ONF

- YANG extensions
  - Model augmentation
  - New models
Agenda

- What is OpenDaylight
- Some things I’ve learned
- Open Source in support of Standards (and vice versa)
- Where OpenDaylight is Going
Current Projects

14 more project proposals in some state of preparation

- Controller
- Virtual Tenant Network (VTN)
- Open DOVE
- OpenFlow Plugin
- Affinity Metadata Service
- YANG Tools
- LISP Flow Mapping
- OVSDB
- OpenFlow Protocol Library
- BGP-LS/PCEP
- Defense4All
- SNMP4SDN
- Integration Group
- Dlux
- Group-based Policy
- OpenDaylight Toolkit
- PacketCable PCMM
- OpFlex Implementation
- Documentation
- Dynamic Resource Reservation
- Table Type Patterns (TTPs)
- SDNi
- OpenContrail

(Red are new since Hydrogen)

https://wiki.opendaylight.org/view/Project_Proposals:Main
Other Future Technical Work

- Core Infrastructure
  - Factoring apart the controller, e.g., MD-SAL, etc.
  - Data Persistence, DOM manipulation, etc.
  - Distributed Systems (Infinispan, Akka, …)
  - Performance, Scalability, Stability
  - Code Quality, Test Coverage,

- We need more code that writes code
  - MD-SAL is an example
  - Helps automate interactions with standards
  - More automation is more better
Non Technical Work

- Continue to build/refine our community
  - Increasing committer diversity across projects
    - Weekly status meetings
    - More transparency is more better
  - “Staffing”
    - Release engineering
    - Documentation

- Continue to refine our engineering systems
  - Thanks Andrew!
  - Fewer humans in the loop
  - SDN Simulation Platform
Getting Started and Involved

- Developer documentation is at wiki.opendaylight.org
  - List of current projects and documentation
  - List of recurring meetings
  - Information on proposing new projects for OpenDaylight

- Open mailing lists: lists.opendaylight.org
  - Discussion groups on specific projects
  - Cross-project discussions
  - Announcements

- #opendaylight IRC channel at freenode.net
Conclusions

- It’s Standards and Open Source
  - Not Standards vs. Open Source

- The interplay between the two will define the future

- You can (and should) get involved
  - [http://www.opendaylight.org/](http://www.opendaylight.org/)
  - [https://github.com/YangModels/yang](https://github.com/YangModels/yang)
  - [http://www.openstack.org/](http://www.openstack.org/)
  - [http://openvswitch.org/](http://openvswitch.org/)