



**Dr. Nick Feamster**  
Associate Professor

# Software Defined Networking



*In this course, you will learn about software defined networking and how it is changing the way communications networks are managed, maintained, and secured.*

# This Lesson: OpenDaylight

- Overview of OpenDaylight
  - Consortium
  - Architecture
- Demonstration
  - Life of a packet, Web interface
  - Essential ODL functions
- More information: <http://sdnhub.org/>

Slides adapted from: <http://goo.gl/k5Ba5J>  
Used with permission from SDNHub.

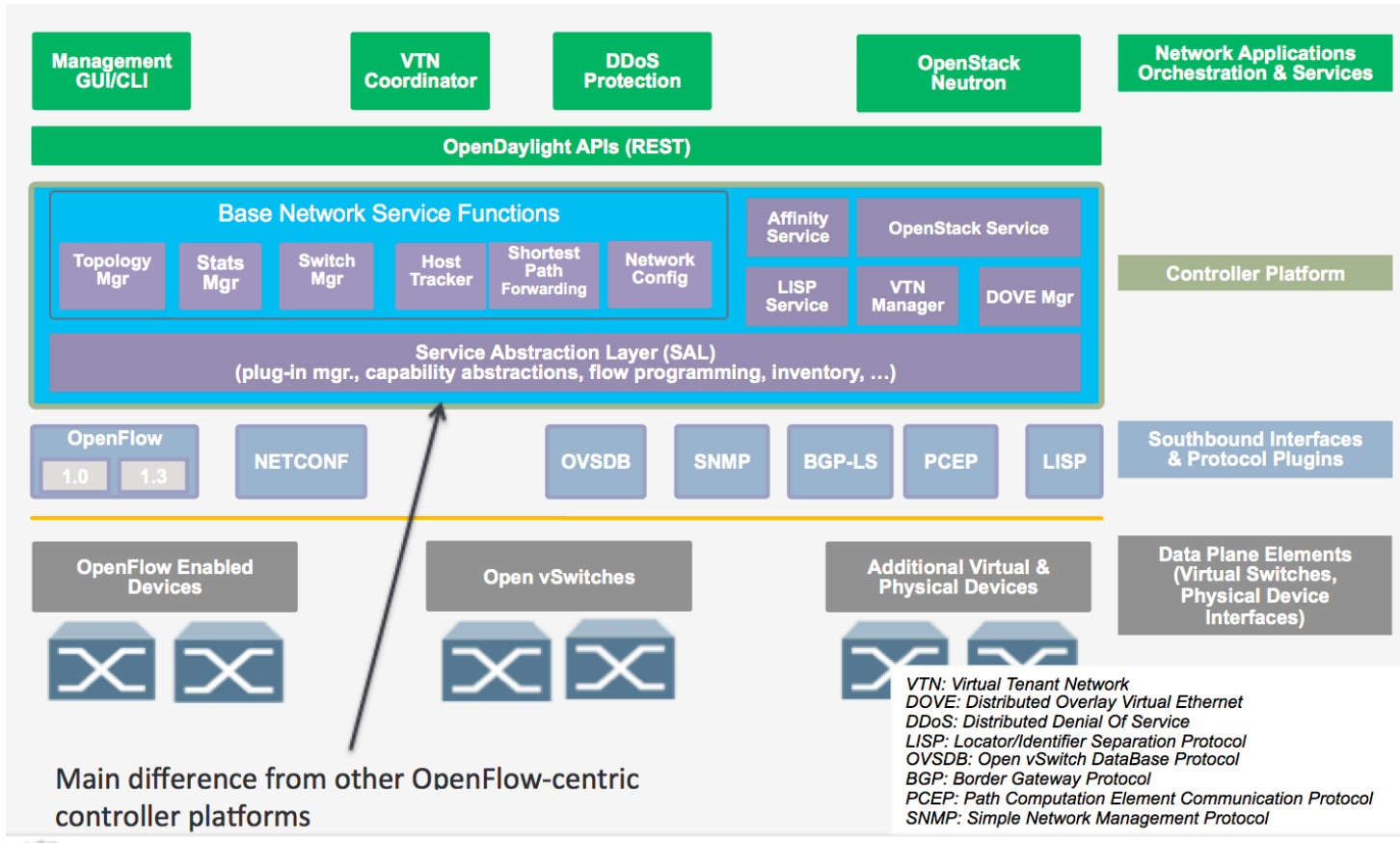
# OpenDaylight Consortium

- Heavy industry involvement and backing



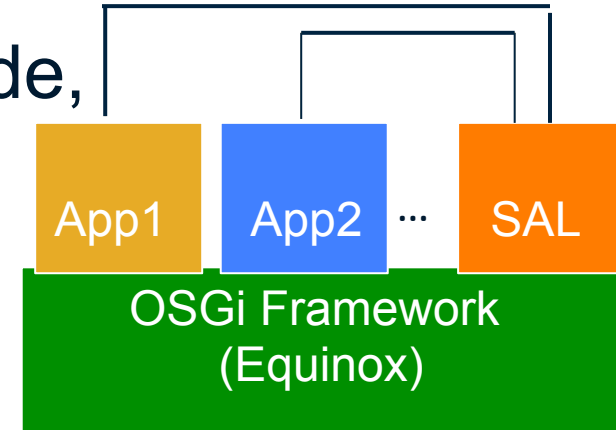
- Focused on having an open framework for building upon SDN/NFV innovations
  - Not limited to OpenFlow innovations

# Hydrogen Release



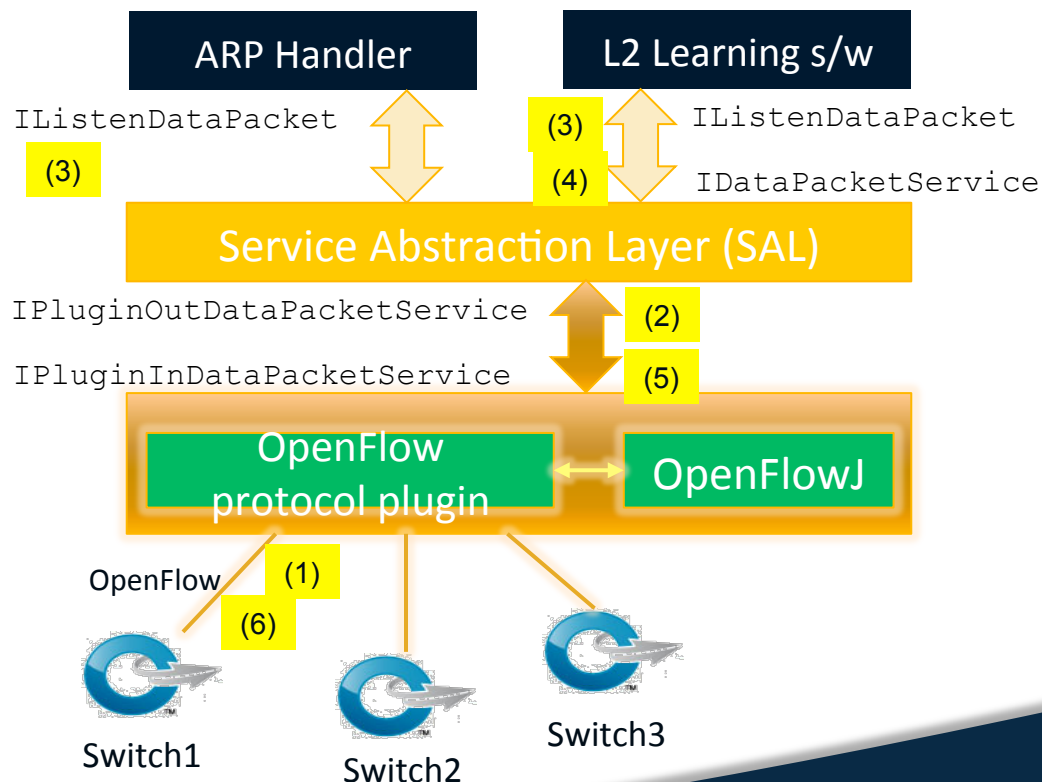
# Java, Maven, OSGi, Interface

- ⦿ **Java** chosen as an enterprise-grade, cross-platform compatible language
- ⦿ **Maven** – build system for Java
- ⦿ **OSGi:**
  - Allows dynamically loading bundles
  - Allows registering dependencies and services exported
  - For exchanging information across bundles
- ⦿ Java Interfaces are used for event listening, specifications, and forming patterns



# Life of a Packet

1. A packet arriving at Switch1 will be sent to the appropriate plugin managing the switch
2. The plugin will parse the packet, generate an event for SAL
3. SAL will dispatch the packet to the modules listening for DataPacket
4. Module handles packet and sends packet\_out through IDataPacketService
5. SAL dispatches the packet to the modules listening for DataPacket
6. OpenFlow message sent to appropriate switch



# OpenDaylight Web Interface

192.168.56.101:8080/#

OPENDAYLIGHT | Devices | Flows | Troubleshoot | admin

Nodes Learned | [Connection Manager](#)

### Nodes Learned

Search

Node Name	Node ID	Ports
None	OF 00:00:00:00:00:00:02	2
None	OF 00:00:00:00:00:00:01	2

1-2 of 2 items | Page 1 of 1

Static Route Configuration

### Static Route Configuration

[Add Static Route](#) [Remove Static Route](#)

Search

Name	Static Route	Next Hop Address
0 items		

Subnet Gateway Configuration | [SPAN Port Configuration](#)

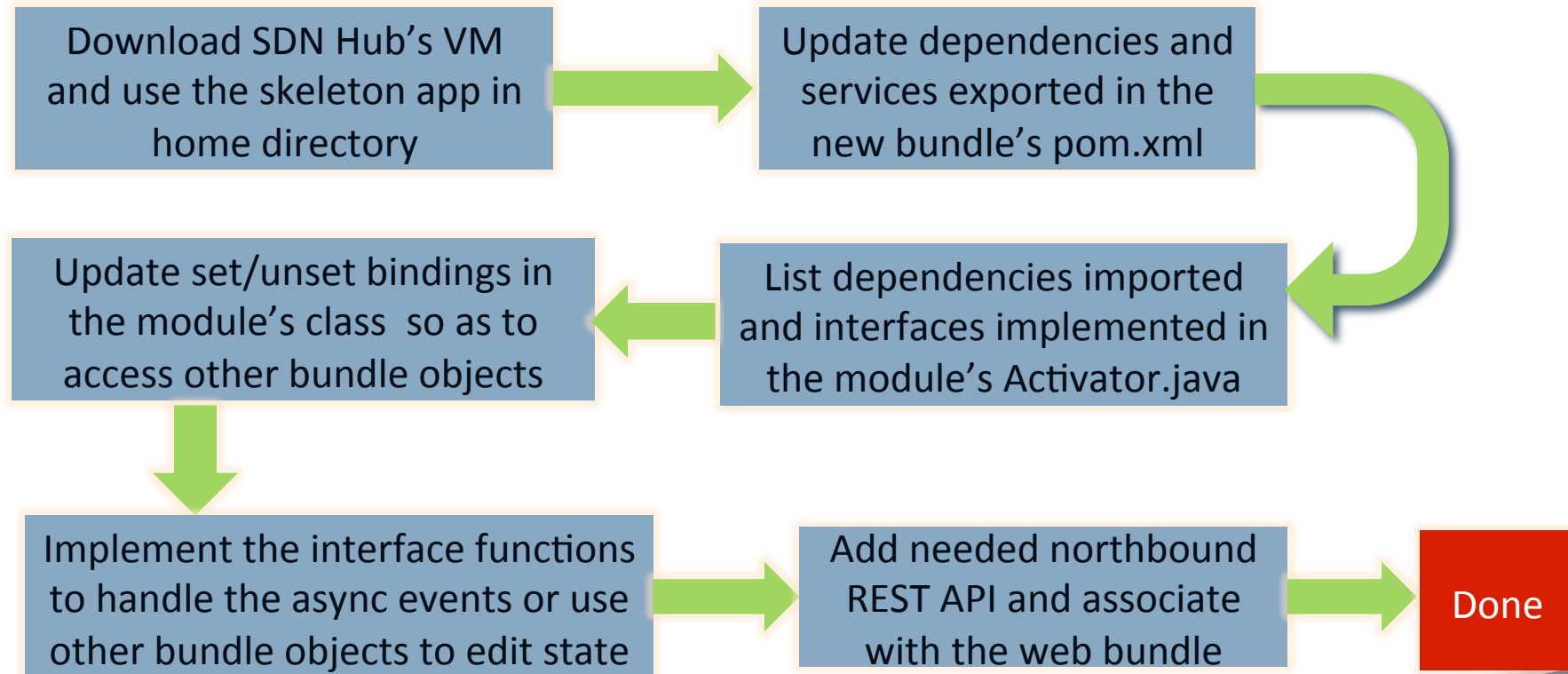
### Subnet Gateway Configuration

[Add Gateway IP Address](#) [Remove Gateway IP Address](#) [Add Ports](#)

Search

Name	Gateway IP Address/Mask	Ports
default (cannot be modified)	0.0.0.0/0	

# Steps for Writing a new application





# Essential Code Constructs

	Beacon	OpenDaylight
Packet_in handling	<pre>public class XX implements IOFMessageListener {     public Command receive(IOFSwitch sw, OFMessage msg) throws IOException { ... } }</pre>	<pre>public class XX implements IListenDataPacket {     public PacketResult receiveDataPacket(RawPacket inPkt) { ... } }</pre>
Packet parsing	<pre>Ethernet ethHdr = new Ethernet(pi.getPacketData()); IPv4 ipv4Hdr = (IPv4) ethHdr.getPayload();</pre>	<pre>Ethernet ethHdr = (Ethernet) this.dataPacketService.decodeDataPacket(inPkt); IPv4 ipv4Hdr = (IPv4) ethHdr.getPayload();</pre>
Send msg to switch	<pre>OFPacketOut pktOut = new OFPacketOut(packetData,     actions,         OFPacketOut.BUFFER_ID_NONE); actions.add(new OFActionOutput(port)); router.getOutputStream().write(pktOut);</pre>	<pre>RawPacket destPkt = new RawPacket(inPkt); destPkt.setOutgoingNodeConnector(p); this.dataPacketService.transmitDataPacket(destPkt);</pre>

## ○ Several similarities between Beacon and OpenDaylight

- This goes beyond just these two controller platforms
- The above three functions are basic to all controller platforms

# Main Constructs

## A. Packet in event handling:

- public class TutorialL2Forwarding implements IListenDataPacket
  - Indicates that the class will handle any packet\_in events
- public PacketResult receiveDataPacket(RawPacket inPkt) { ... }
  - Call-back function to implement in the class for receiving packets

## B. Packet parsing

- Packet formattedPak = this.dataPacketService.decodeDataPacket(inPkt);
- byte[] srcMAC = ((Ethernet)formattedPak).getSourceMACAddress();
- long srcMAC\_val = BitBufferHelper.toNumber(srcMAC);

## C. Send message (packet\_out or flow\_mod) to switch

- RawPacket destPkt = new RawPacket(inPkt);
- destPkt.setOutgoingNodeConnector(p);
- this.dataPacketService.transmitDataPacket(destPkt);

# Useful Interfaces and Bundles

Bundle	Exported interface	Description
arphandler	IHostFinder	Component responsible for learning about host location by handling ARP.
hosttracker	IFlptoHost	Track the location of the host relatively to the SDN network.
switchmanager	ISwitchManager	Component holding the inventory information for all the known nodes (i.e., switches) in the controller.
topologymanager	ITopologyManager	Component holding the whole network graph.
usermanager	IUserManager	Component taking care of user management.
statisticsmanager	IStatisticsManager	Component in charge of using the SAL ReadService to collect several statistics from the SDN network.

# Useful Interfaces and Bundles

Bundle	Exported interface	Description
sal	IReadService	Interface for retrieving the network node's flow/port/queue hardware view
sal	ITopologyService	Topology methods provided by SAL toward the applications
sal	IFlowProgrammerService	Interface for installing/modifying/removing flows on a network node
sal	IDataPacketService	Data Packet Services SAL provides to the applications
web	IDaylightWeb	Component tracking the several pieces of the UI depending on bundles installed on the system.

# Summary

- OpenDaylight is an industry-backed effort to develop broader set of SDN solutions
- SDN is no longer just OpenFlow!
  - Possible to integrate a broad set of cloud-based applications
  - Set of functions is similar to other controllers
- Learning curve is significant.  
SDN Hub has good starter kit!