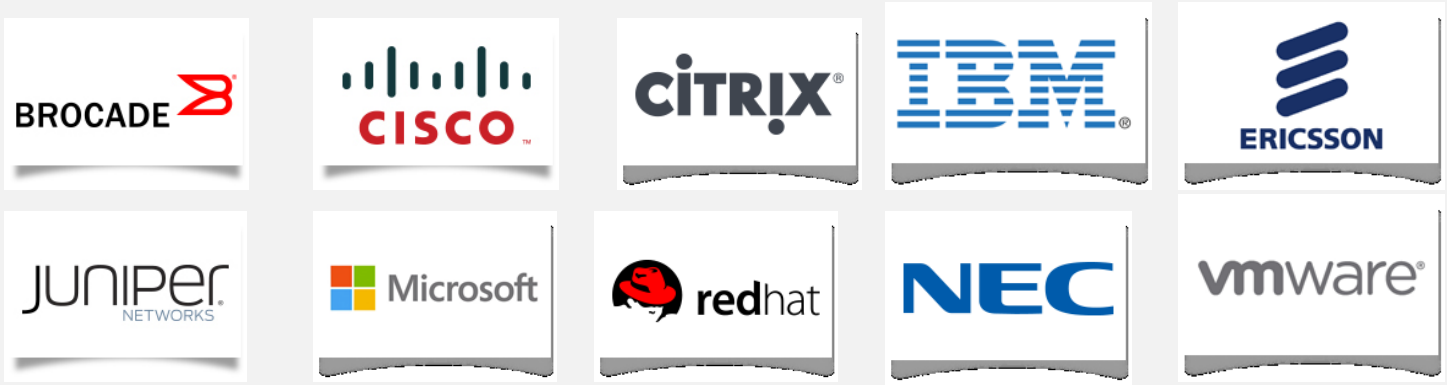
App Development Tutorial

Anirudh Ramachandran, Deutsche Telekom Labs & SDN Hub

Thanks to:
SDN Hub team
Srikanth Sundaresan, GA Tech
GPO Team

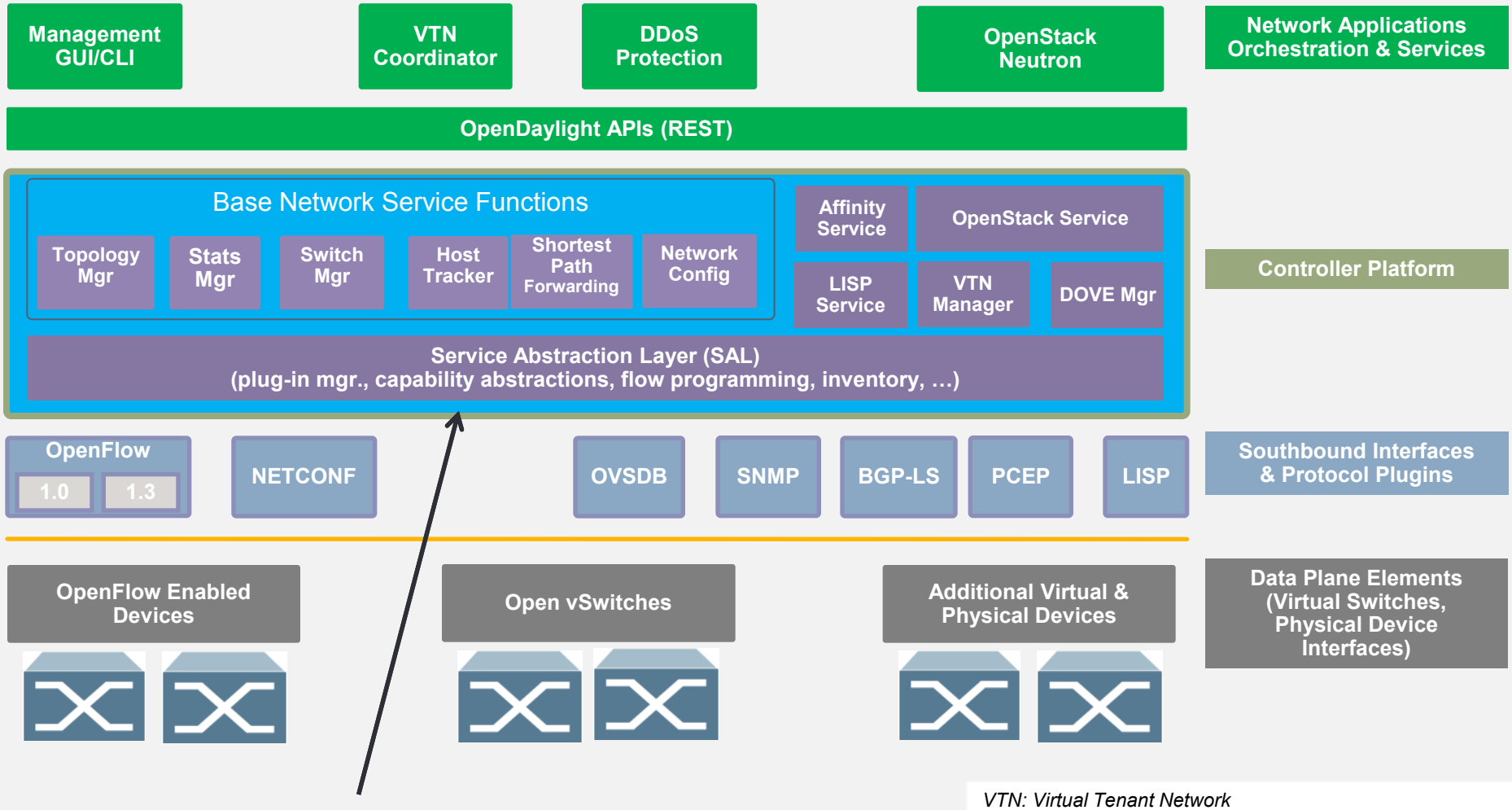
OpenDaylight 2-minute Intro

- Heavy industry involvement and backing



- Focused on having an open framework for building upon SDN/NFV innovations
  - Not limited to Openflow innovations, but in fact decoupled from it allowing the two to evolve independently

Opendaylight Hydrogen Release

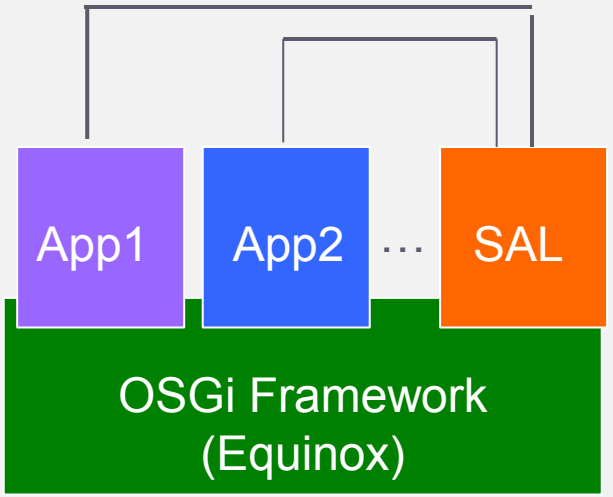


Main difference from other OpenFlow-centric controller platforms

VTN: Virtual Tenant Network
DOVE: Distributed Overlay Virtual Ethernet
DDoS: Distributed Denial Of Service
LISP: Locator/Identifier Separation Protocol
OVSDB: Open vSwitch DataBase Protocol
BGP: Border Gateway Protocol
PCEP: Path Computation Element Communication Protocol
SNMP: Simple Network Management Protocol

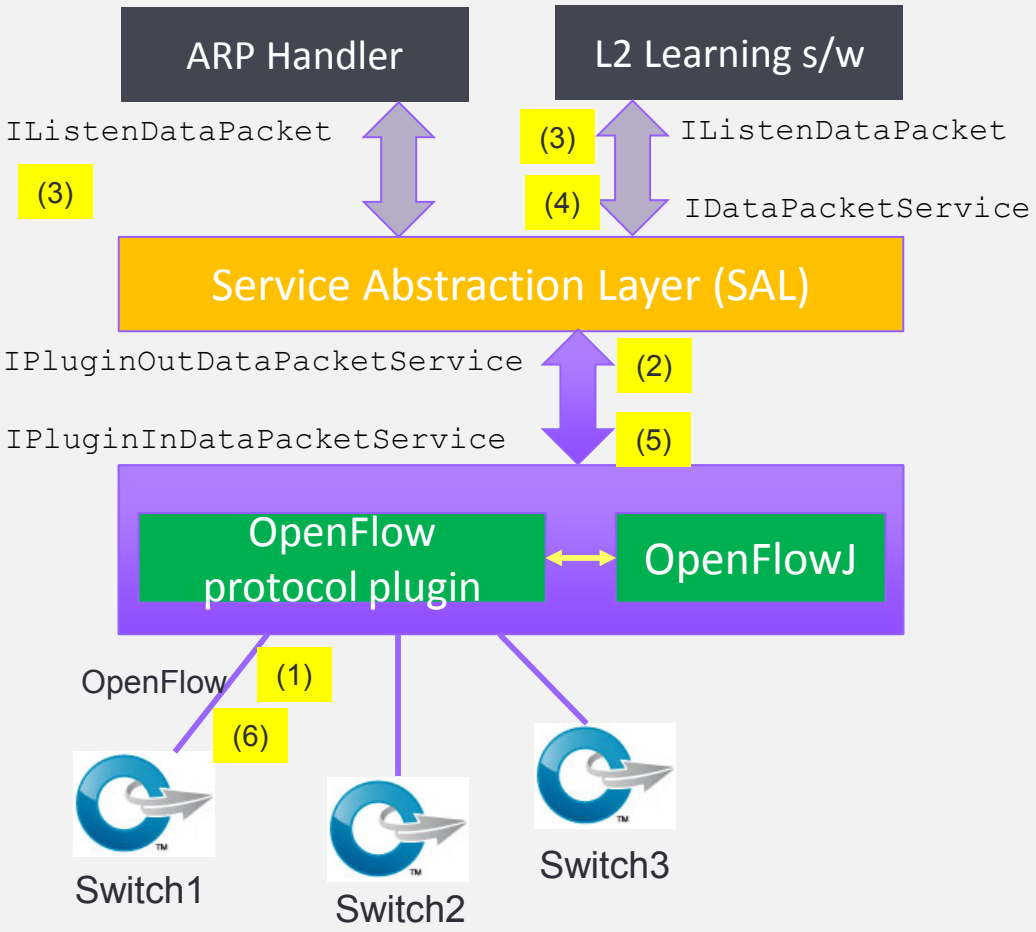
Java, Maven, OSGi, Interface

- 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

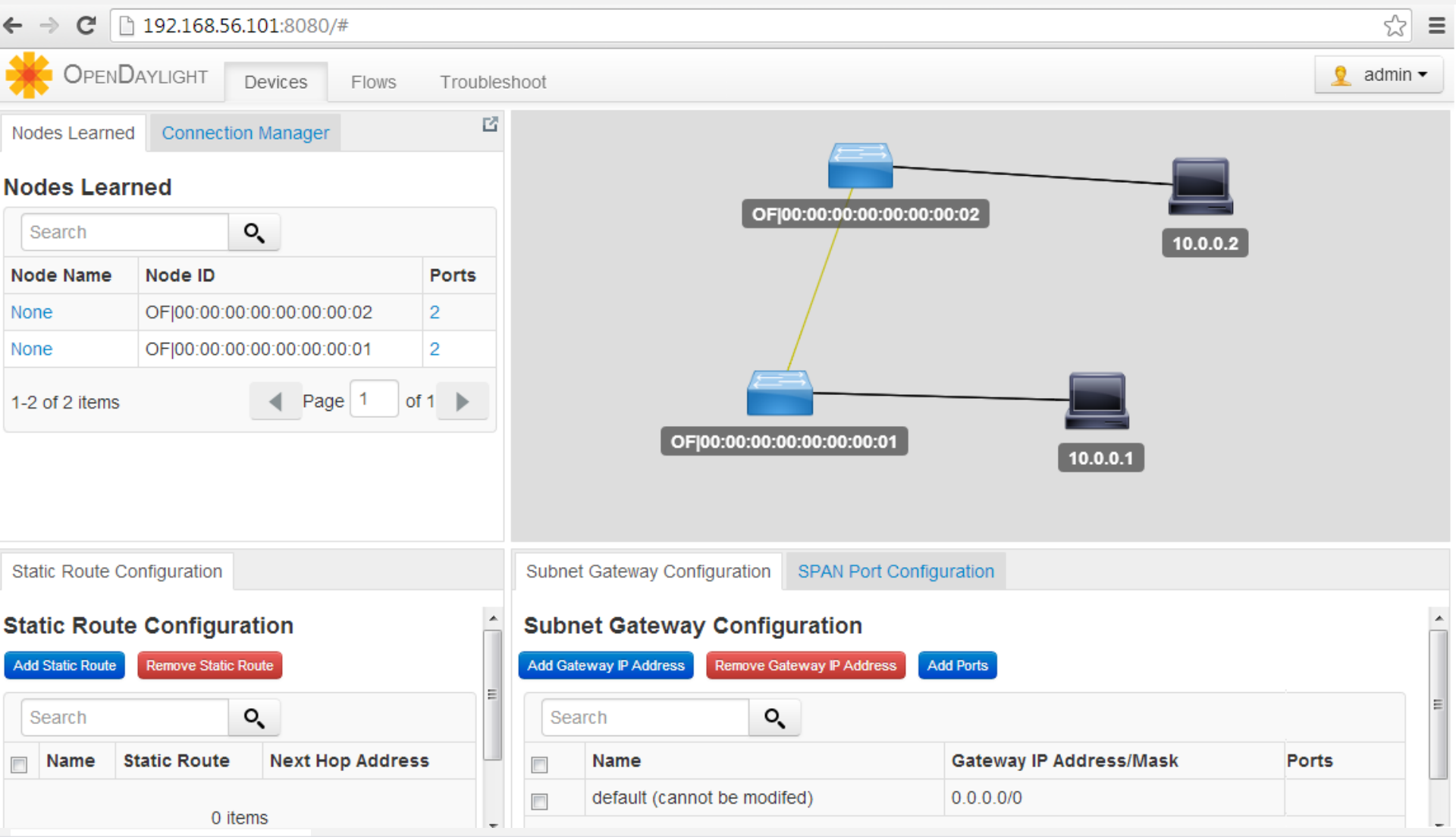


Life of a Packet

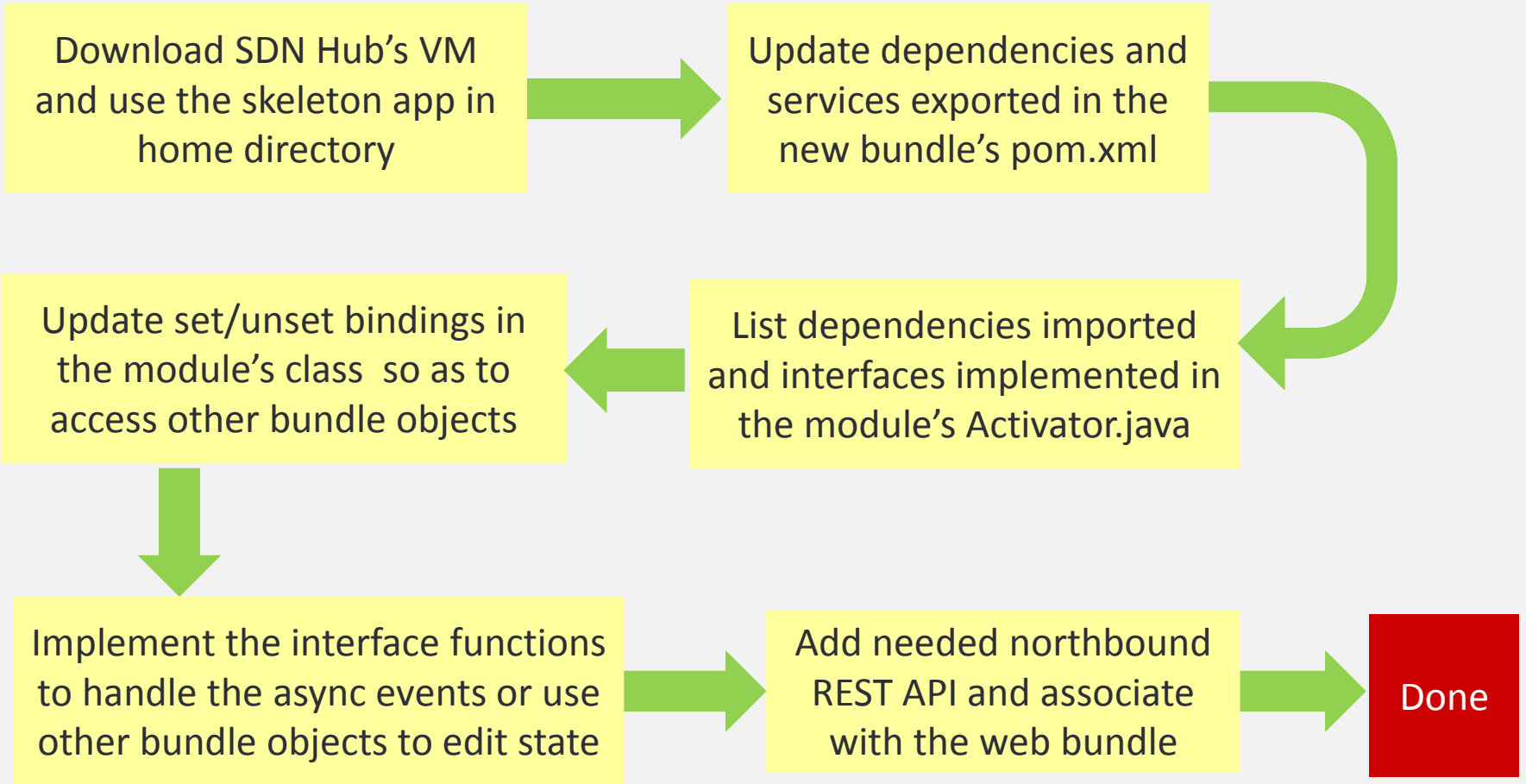
1. A packet arriving at Switch1 will be sent to the appropriate plugin managing the switch
2. The plugin will parse the packet, generate an event for SAL
3. SAL will dispatch the packet to the modules listening for DataPacket
4. Module handles packet and sends packet_out through IDataPacketService
5.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



OpenDayLight web interface



Steps for Writing a new application



Essential code constructs

	Beacon	OpenDaylight

Packet_in handling	public class XX implements IOFMessageListener {
	  public Command receive(IOFSwitch sw, OFMessage msg) throws IOException { … }
	}
		public class XX implements IListenDataPacket {
		  public PacketResult receiveDataPacket(RawPacket inPkt) { … }
		}

Packet parsing	Ethernet ethHdr = new Ethernet(pi.getPacketData()); IPv4 ipv4Hdr = (IPv4) ethHdr.getPayload();	Ethernet ethHdr = (Ethernet) this.dataPacketService.decodeDataPacket(inPkt); IPv4 ipv4Hdr = (IPv4) ethHdr.getPayload();

Send msg to switch	OFPacketOut pktOut = new OFPacketOut(packetData, actions,
	  OFPacketOut.BUFFER_ID_NONE);
	actions.add(new OFActionOutput(port));
	router.getOutputStream().write(pktOut);
		RawPacket destPkt = new RawPacket(inPkt);
		destPkt.setOutgoingNodeConnector(p);
		this.dataPacketService.transmitDataPacket(destPkt);

- 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 App Development Constructs
(See tutorial_L2_forwarding app)

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	IfIptoHost	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.