Developing OpenDaylight Apps with MD-SAL

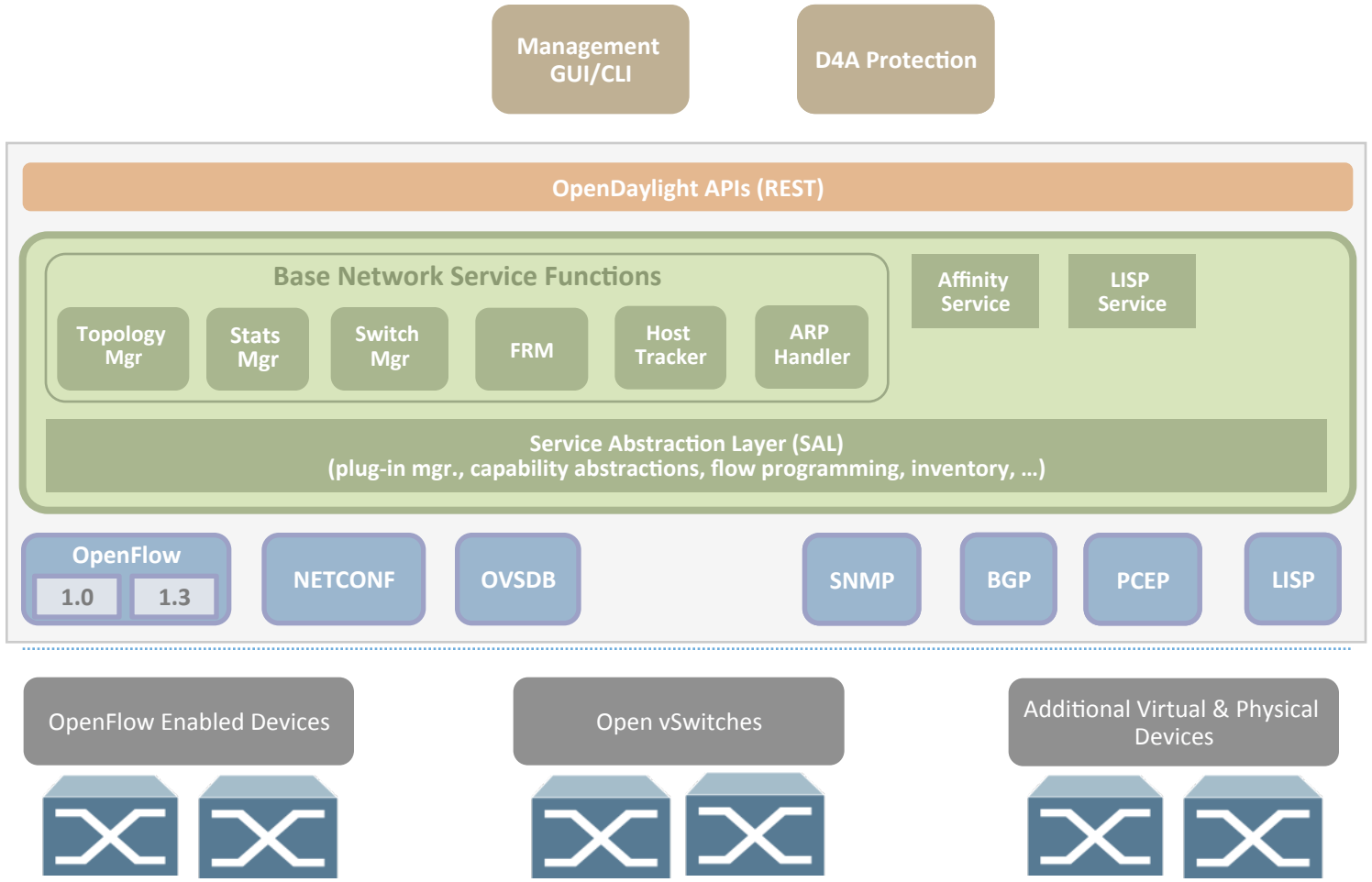
J. Medved, E. Warnicke, A. Tkacik. R. Varga
Cisco

Sample App: M. Rehak, Cisco

February 04, 2014

OPEN DAYLIGHT SUMMIT

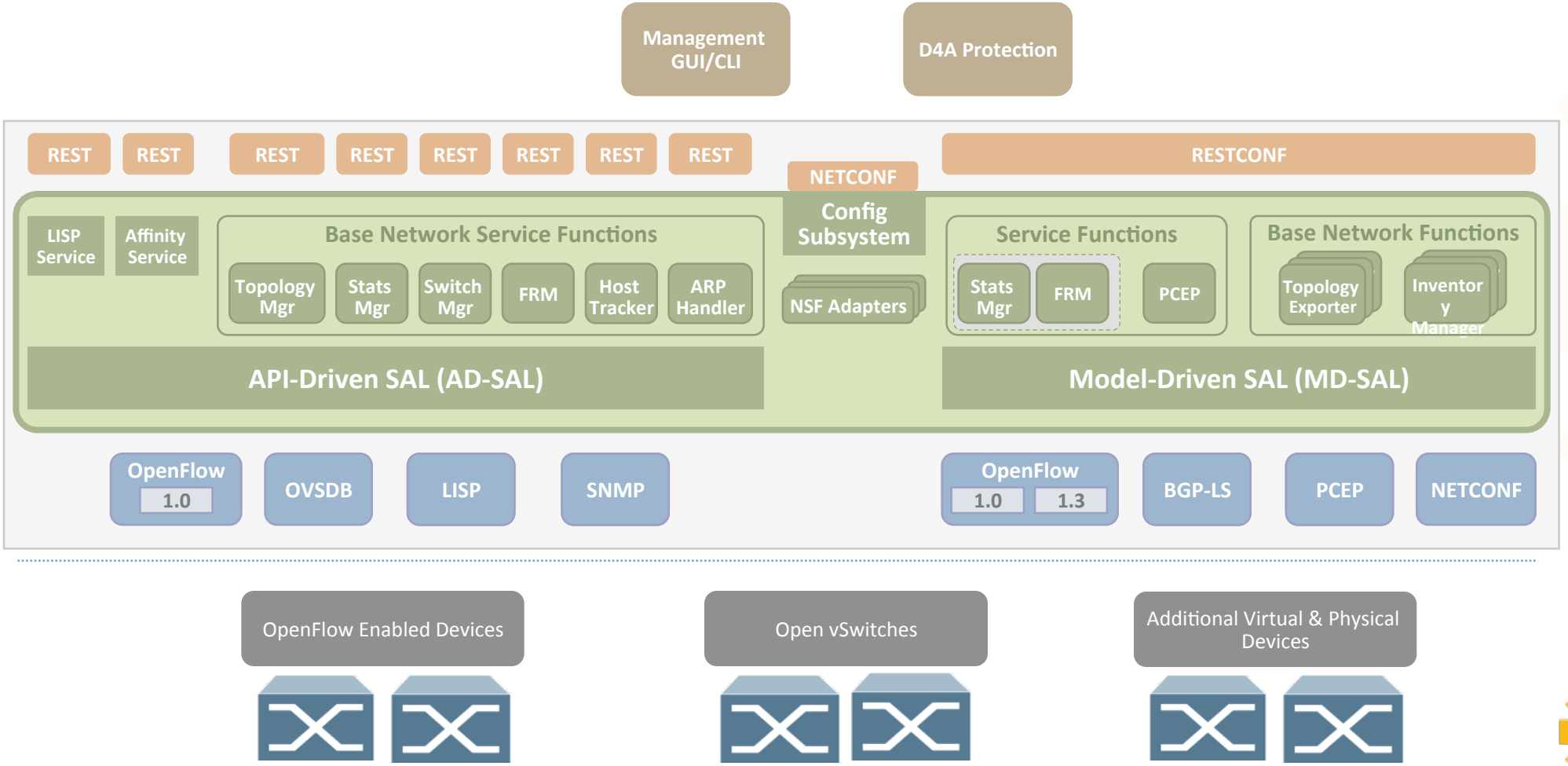
Controller Architecture



Example: Service Provider Edition

OPEN DAYLIGHT SUMMIT

Hydrogen Implementation

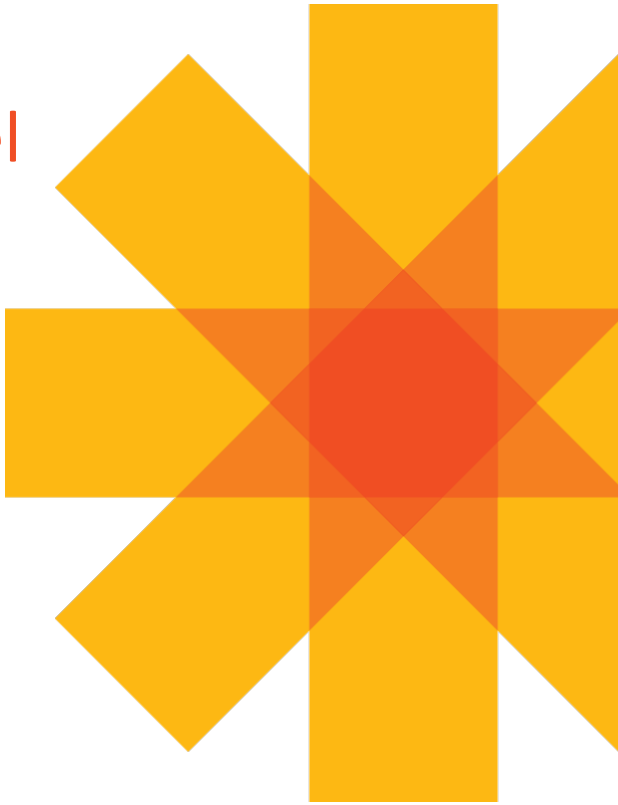


Example: Service Provider Edition

OPEN DAYLIGHT SUMMIT

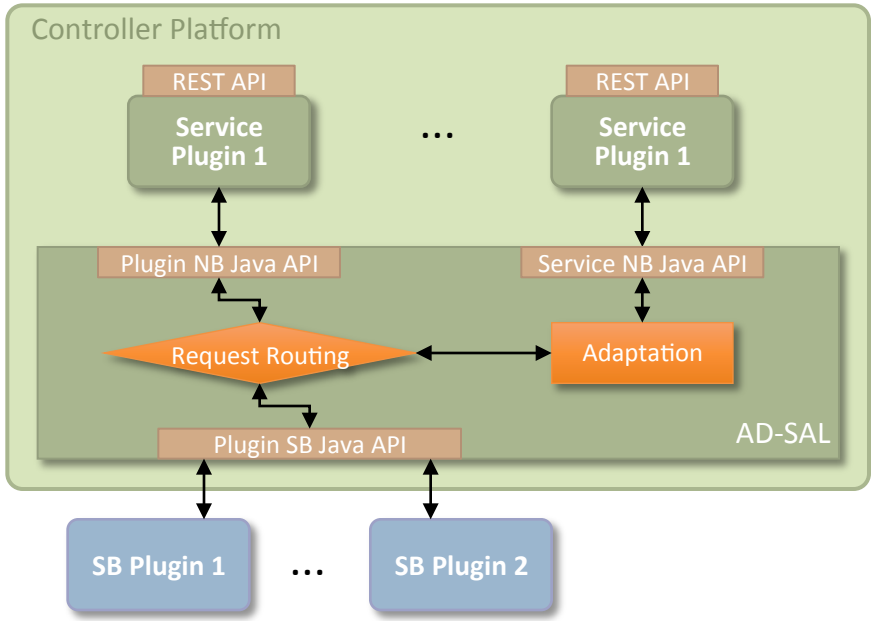
MD-SAL – Motivation & Requirements

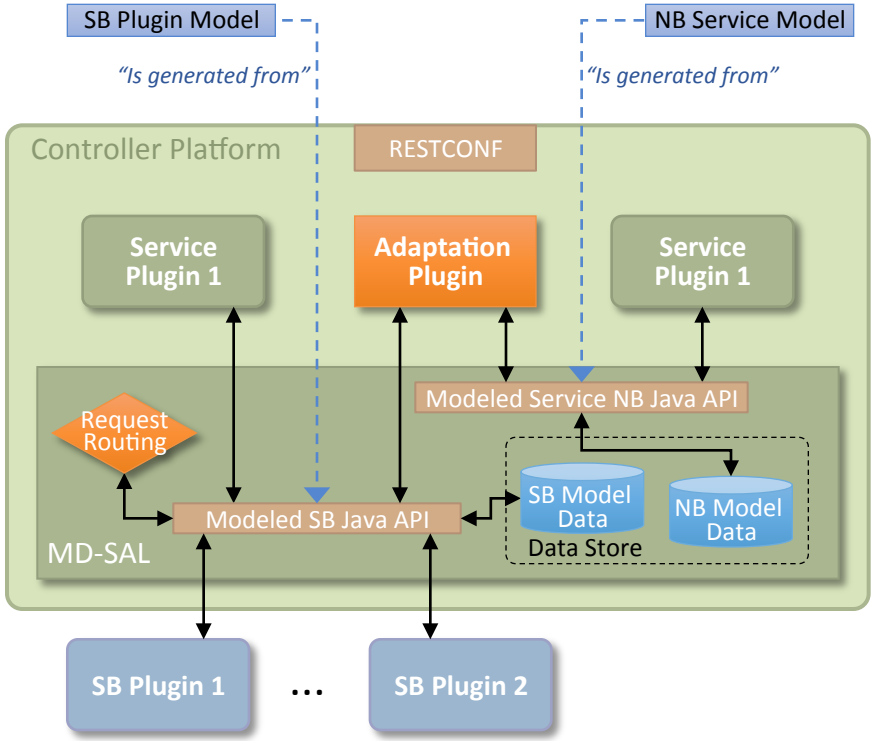
- Flexibility, but common framework and programming model
  - Support API governance
  - Functionally equivalent APIs for different language bindings
- Run-time Extensibility:
  - Augment existing functionality
  - Load new models (extends controller's functionality)
- Avoid module/sub-system hotspots
- Performance & scale



OPEN DAYLIGHT SUMMIT

From AD-SAL to MD-SAL





AD-SAL key services:

- Request routing
- Service Adaptation

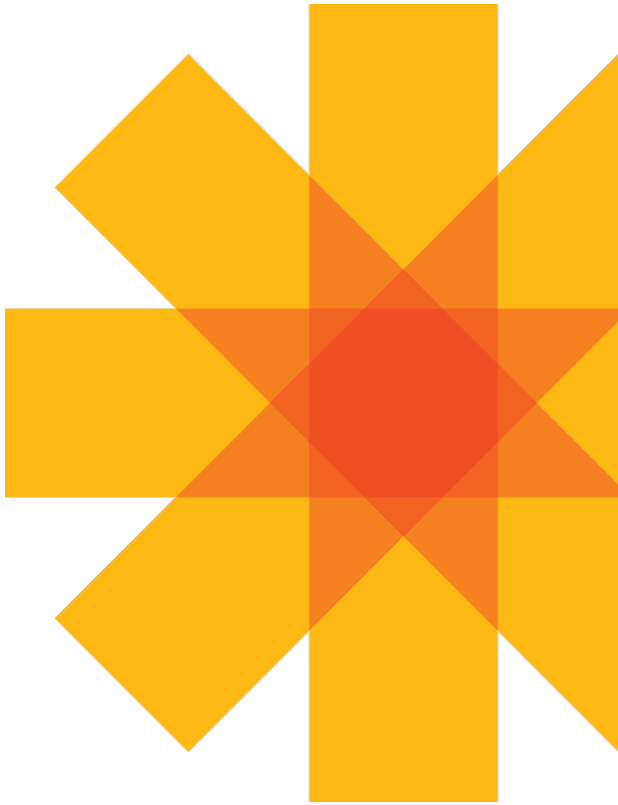
MD-SAL key services:

- Request (RPC) and notification routing
- Data Storage

OPEN DAYLIGHT SUMMIT

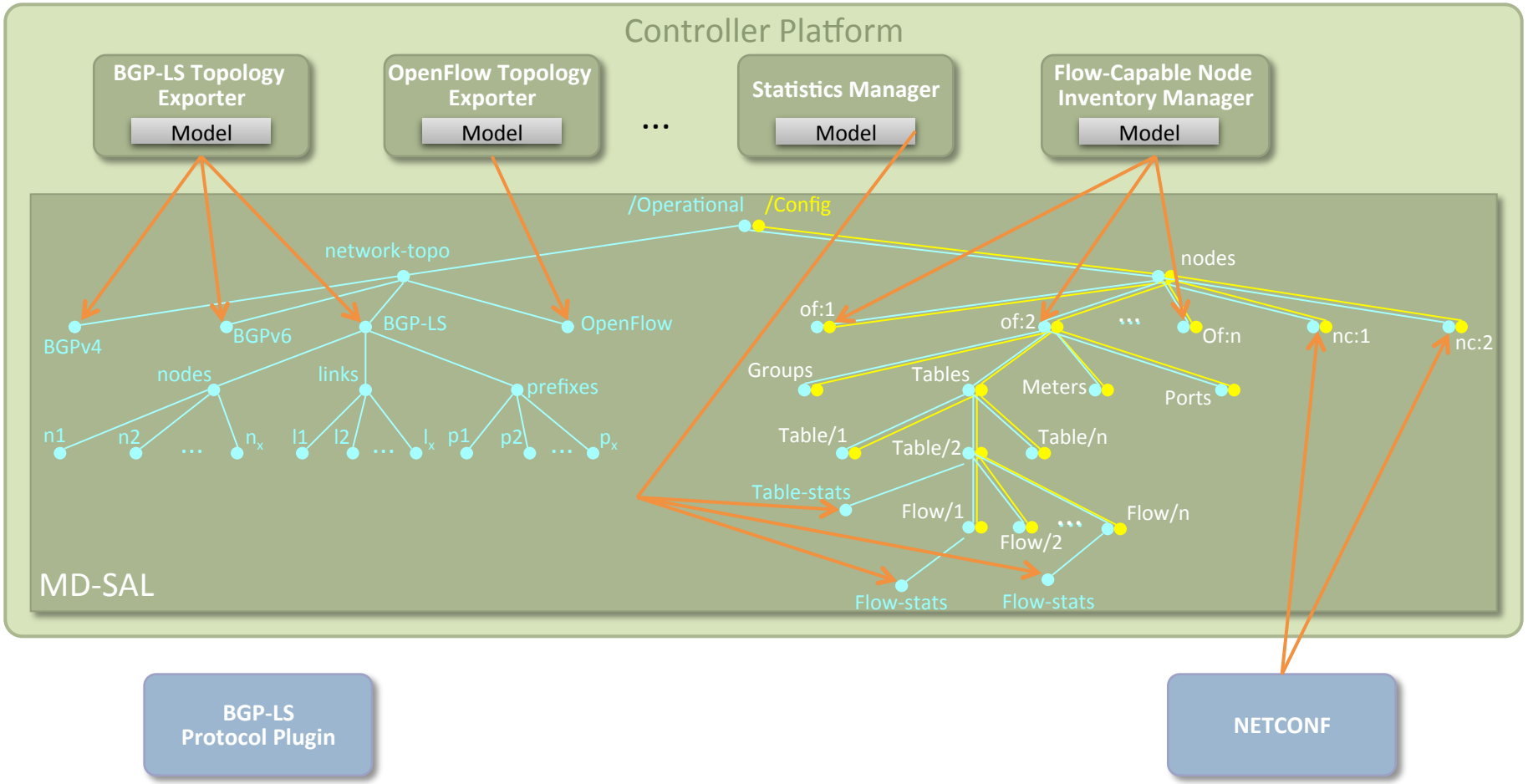
MD-SAL – Key Tenets Summary

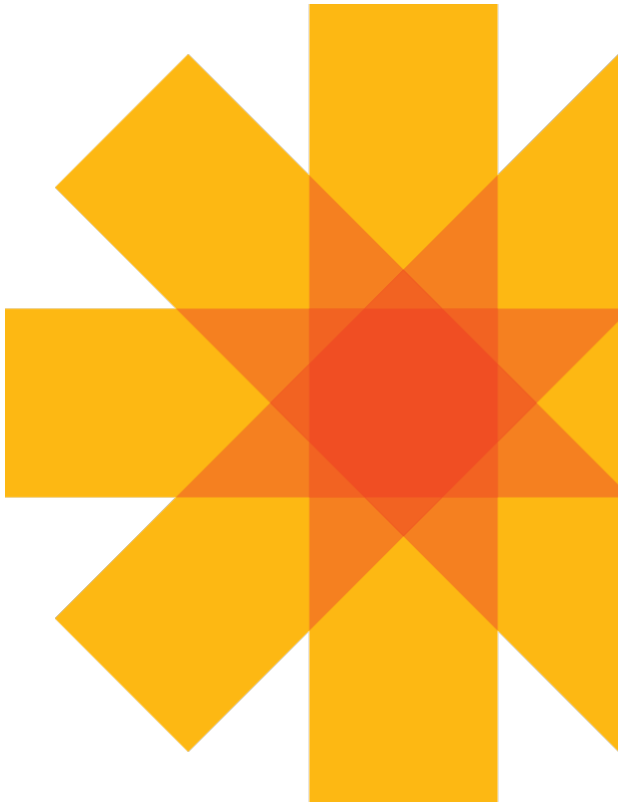
- Modeling language: YANG (rfc6020)
  - Used also as IDL
  - A couple of minor extensions
- Dynamic late binding
- Runtime & Compile time code generation



OPEN DAYLIGHT SUMMIT

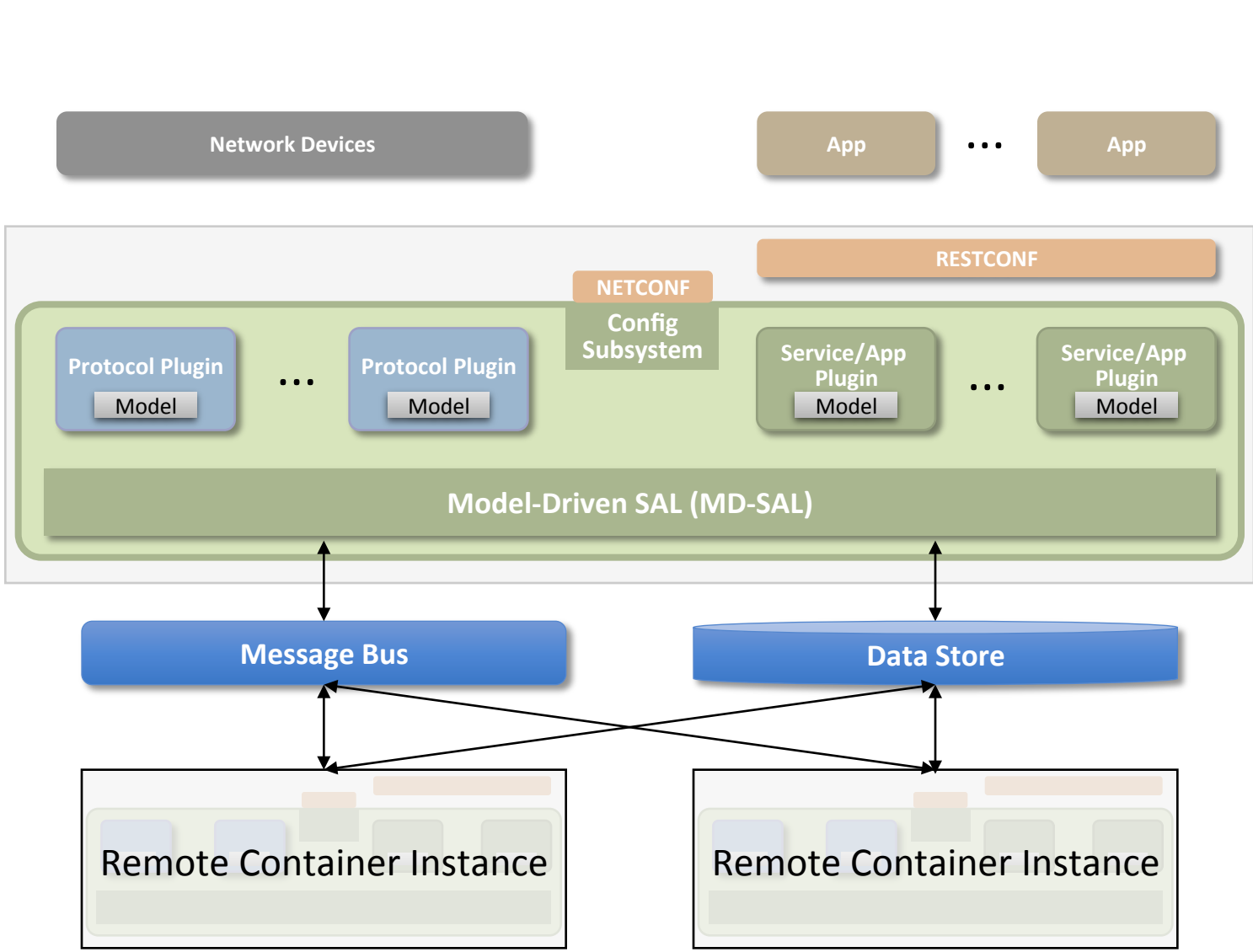
Creating the View of the Network





OPEN DAYLIGHT SUMMIT

MD-SAL: Software Architecture View



Network Applications Orchestration & Services

Data Plane Elements

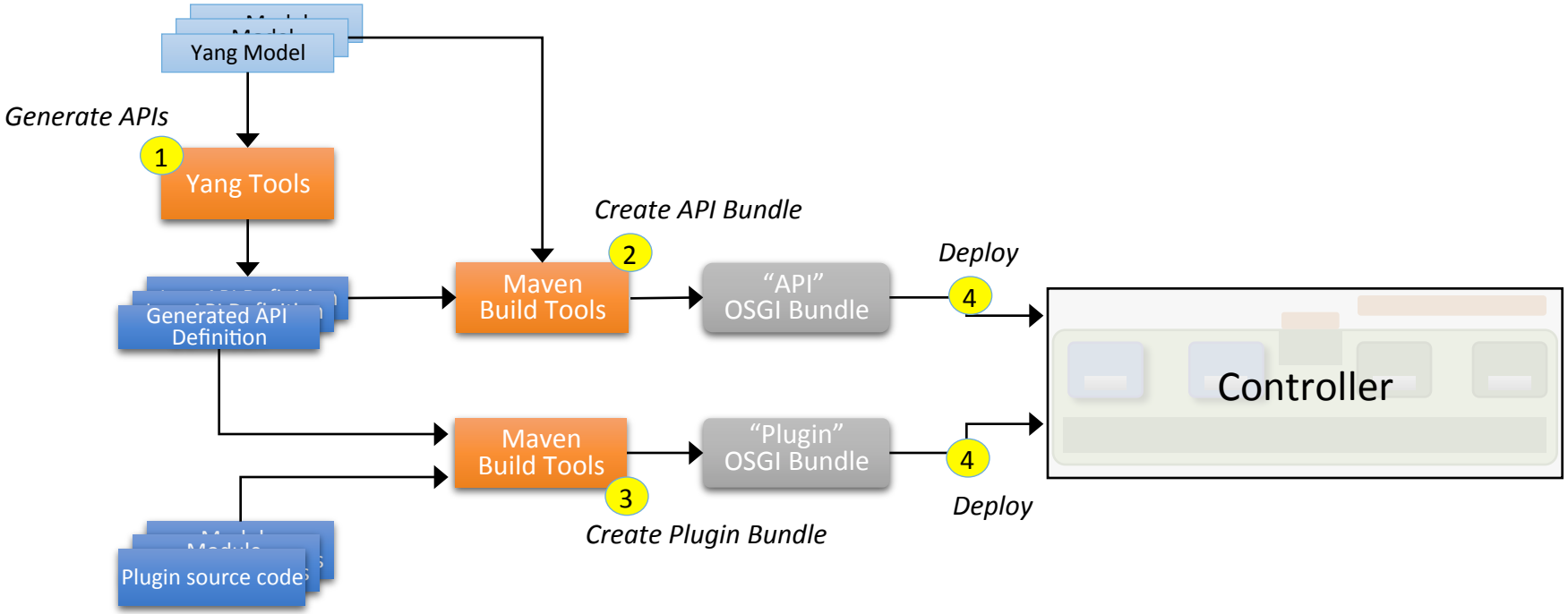
Controller Platform

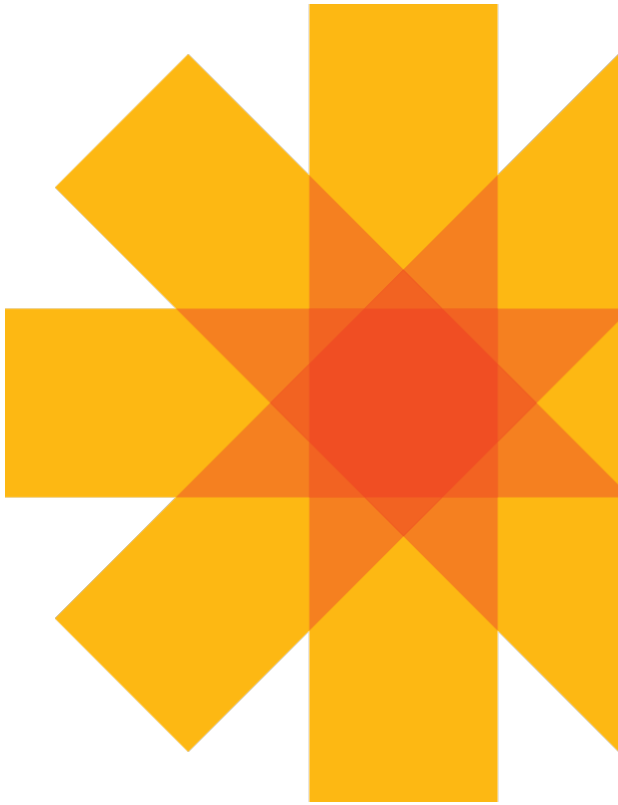
Southbound Interfaces & Protocol Plugins

Clustering

OPEN DAYLIGHT SUMMIT

Building a Plugin/Application

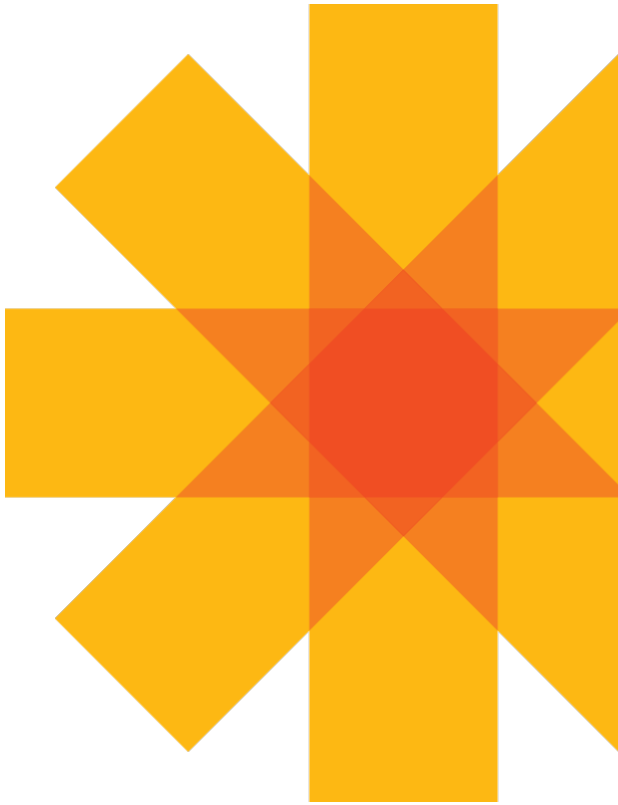




OPEN DAYLIGHT SUMMIT

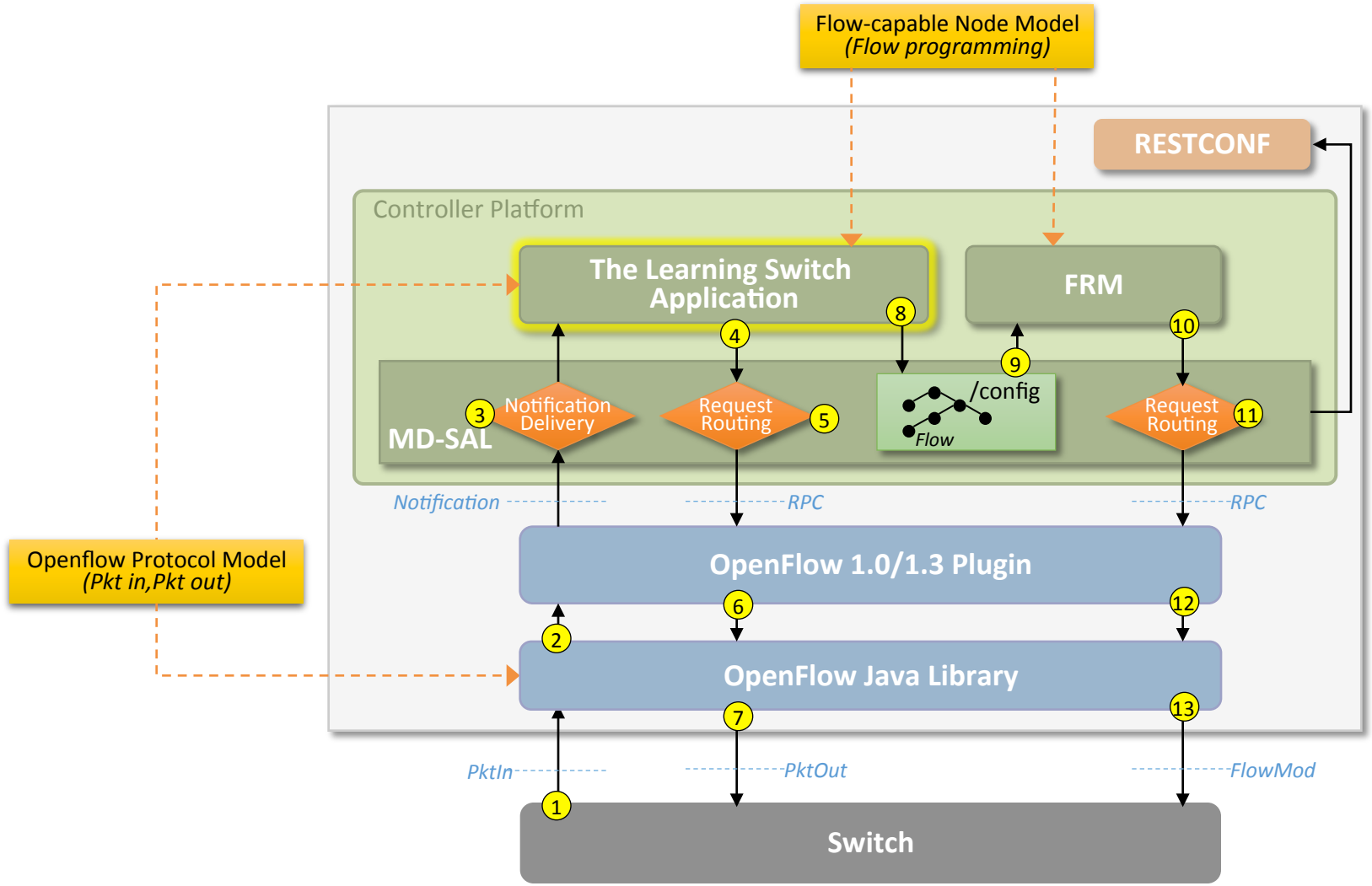
The Learning Switch Application

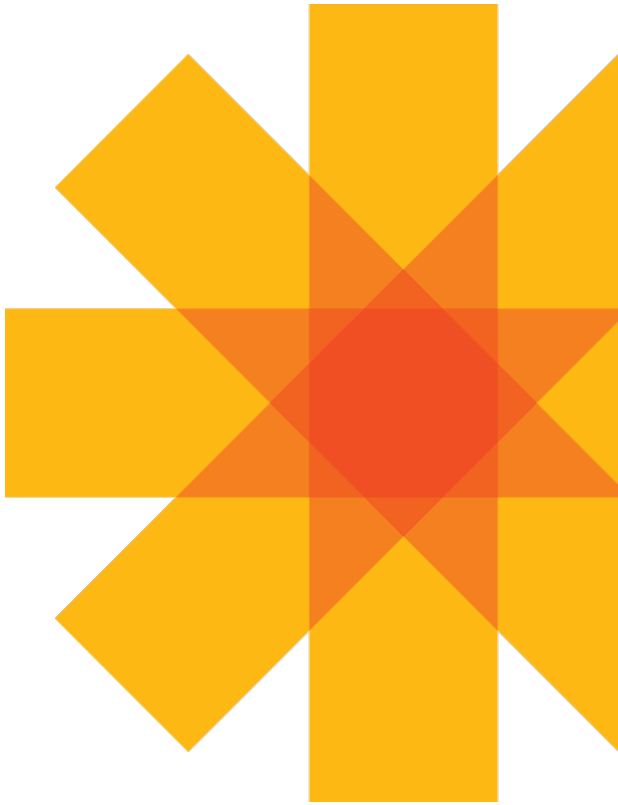
- Inspired by 'OpenDaylight Application Developers' Tutorial: http://sdnhub.org/tutorials/opendaylight/
- ODL Wiki: TBD.
- Functionality:
  - Create a HashMap called mac_to_port
  - On packet_in, Pprse packet to get src and dst MAC address
  - Store in a hashmap mapping between src_mac and in_port
  - Lookup dst_mac in mac_to_port map to find next hop
  - If next hop is found, create flow_mod and send
  - Else, flood like hub.



OPEN DAYLIGHT SUMMIT

Under the Hood: How it Works

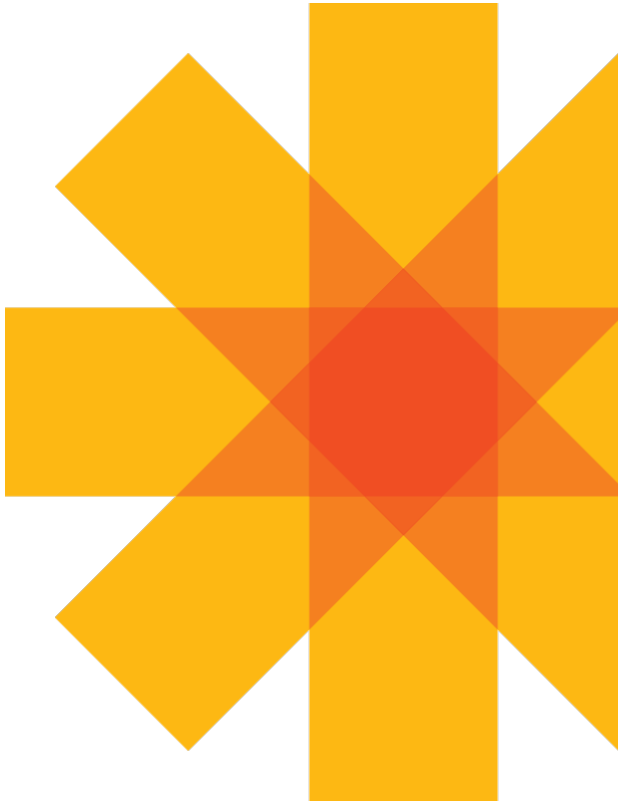




OPEN DAYLIGHT SUMMIT

Building the Sample Application

- Prerequisites:
  - Java 7, Maven 3.0.5 or later, Linux or Mac
  - Mininet 2.1 with OpenFlow 1.3 virtual switches or CPQD (pointers)
  - OpenDaylight Base Edition (download & installation instructions)
- Download the application code from ODL OpenFlow Plugin repo:
  ```
  > git clone https://git.opendaylight.org/gerrit/p/openflowplugin.git
  ```
- Build the application:
  ```
  > cd  openflowplugin/samples/learning-switch/
  > mvn clean install
  ```
- The build creates the 'learning-switch-0.0.3-SNAPSHOT.jar' bundle in 'openflowplugin/samples/learning-switch/target'
- Analyzing code in Eclipse:
  - Prerequisites: Install maven plugin for Eclipse
- More information:
  - https://wiki.opendaylight.org/view/OpenDaylight_Controller:Hydrogen_Developer_Guide:MD-SAL_App_Tutorial



OPEN DAYLIGHT SUMMIT

Starting the Environment

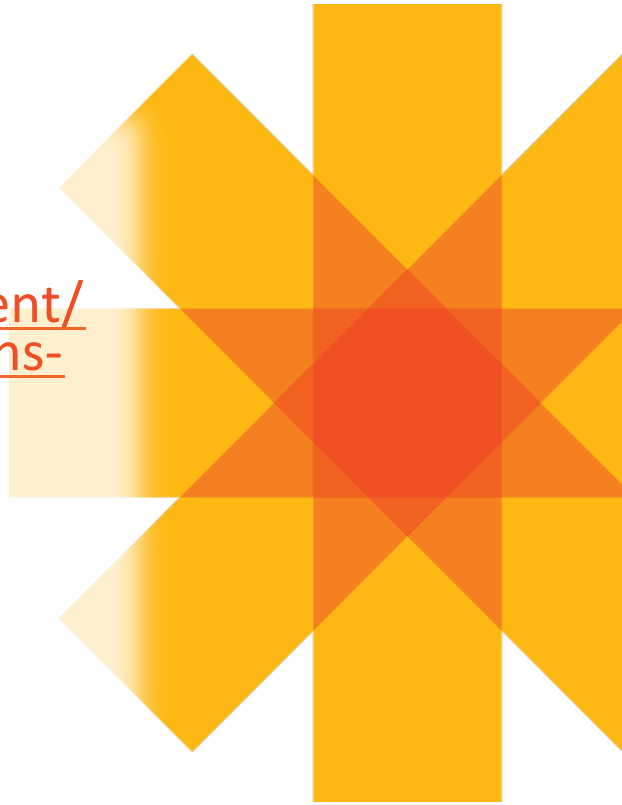
- Prequisites:
  - Java 7, Maven 3.0.5 or later, on Linux or Mac
  - Mininet 2.1 with OpenFlow 1.3 virtual switches: or CPQD
- Download OpenDaylight Base Edition: https://nexus.opendaylight.org/content/repositories/opendaylight.release/org/opendaylight/integration/distributions-base/0.1.1/
- Prepare the controller & application:
  - Unzip the downloaded controller package
  - Delete the 'Simple Forwarding Application' bundle from the distribution:

    ```
    > rm opendaylight/plugins/
      org.opendaylight.controller.samples.simpleforwarding-0.4.1.jar
    ```
  - Upload the created bundle:

    Put the 'learning-switch-0.0.3-SNAPSHOT.jar' bundle into the 'opendaylight/plugins' folder
  - Update logger configuration:

    Optionally, add the following line to configuration/logback.xml file:

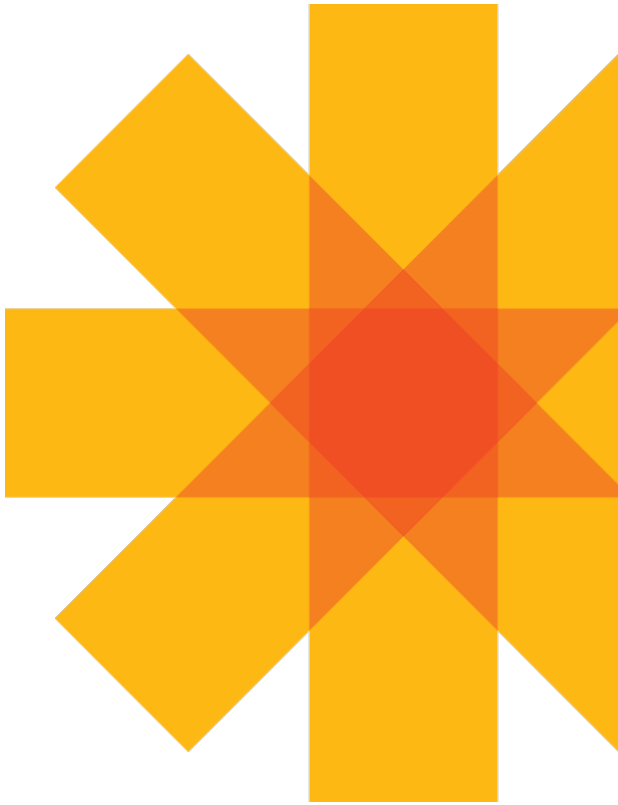
    ```
    <logger name="org.opendaylight.openflowplugin.learningswitch" level="TRACE"/>
    ```



OPEN DAYLIGHT SUMMIT

Starting the Environment (Continued)

- Run the controller:
  ```
  > ./run.sh -of13
  ```
- Check that the application bundle is active. On the controller console, type:
  ```
  osgi > lb learn
  ```
- You should see something like:
  ```
  START LEVEL 6
     ID|State       |Level|Name
    103|Active      |    4|learning-switch (0.0.3.SNAPSHOT)
  osgi>
  ```
- Optionally, check that the controller is listening on Ports 6633 and 6653.
  On a Linux console, type:
  ```
  > netstat -lnp | grep 'java'
  ```
  On a Mac OSX console, type:
  ```
  > lsof -i | grep LISTEN |grep java
  ```
- Start Mininet:
  ```
  > sudo mn --topo single,10  --controller 'remote,ip=<controller-ip-address>,port=6633' --switch ovsk,protocols=OpenFlow13
  ```



OPEN DAYLIGHT SUMMIT

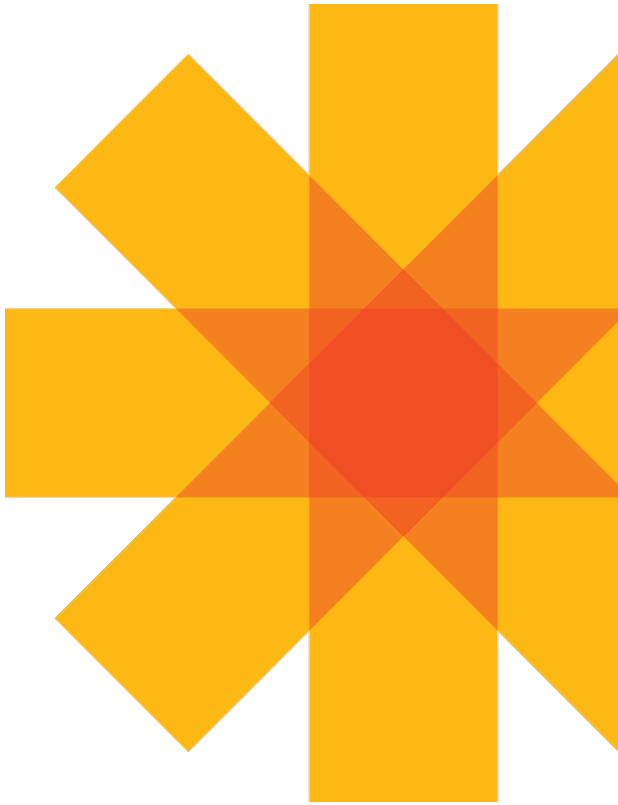
Core Concepts: Instance Identifier

- Unique identifier of an element (location) in the yang data tree;
  - Basically, the path to the node that uniquely identifies all the node's parent nodes
- Examples:
  - Java:

```
InstanceIdentifier<Node> identifier =
InstanceIdentifier.builder(Nodes.class).child(Node.class,new
NodeKey("foo")).build();
```

  - REST:

```
http://localhost:8080/restconf/config/inventory:nodes/node/foo
```



Running & Troubleshooting the App

- Use RESTCONF to validate that the app started and installed the initial flow into the switch:

  On a REST console, issue:

  ```
  GET http://<contr-ip-address>:8080/restconf/operational/
  opendaylight-inventory:nodes/node/openflow:1/table/0
  Accept Header: application/xml
  ```

  There should be one flow on the switch forwarding all packets to the controller

- On Mininet, issue pings between each pair of hosts:

  ```
  mininet> pingall
  ```

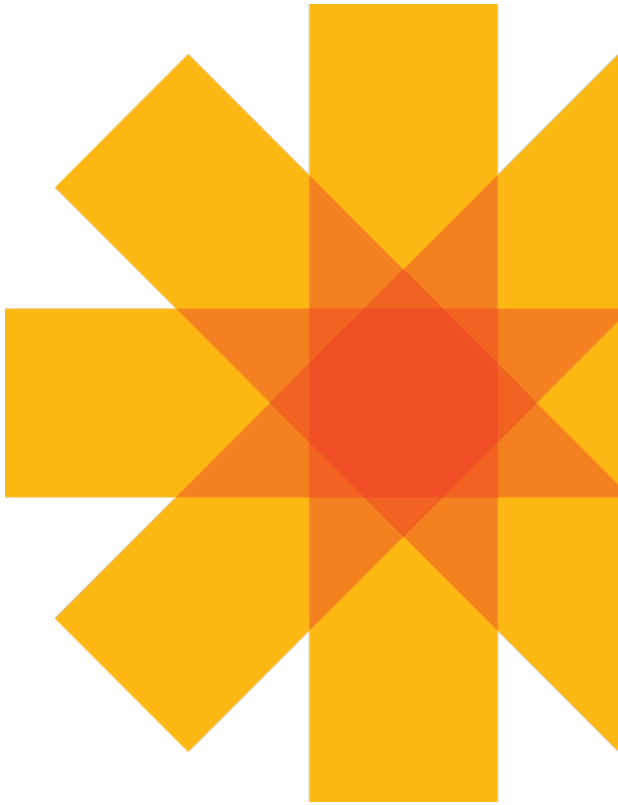
  6 more flows should be installed in the host.

OPEN DAYLIGHT SUMMIT

Other Things to Explore

- Try a network of switches; start Mininet with:

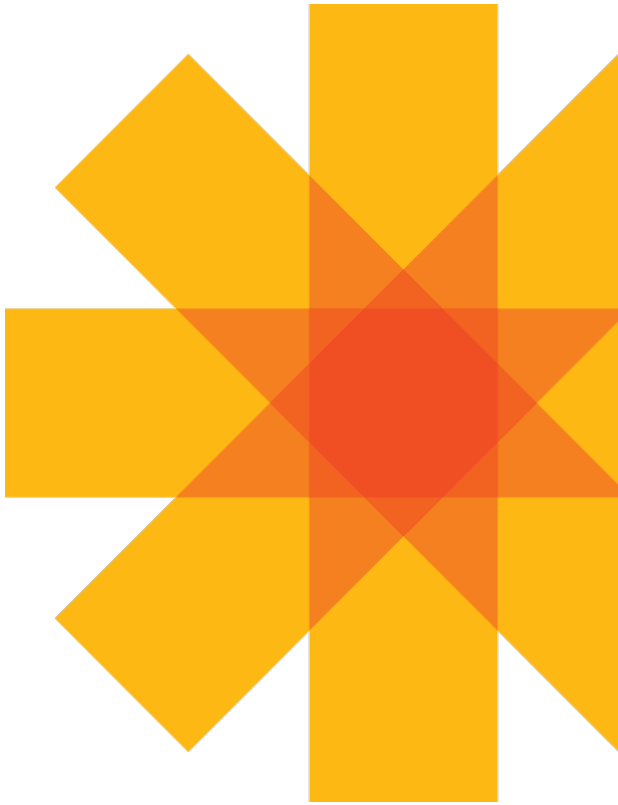
  sudo mn --topo tree,3 --controller 'remote,ip=192.168.4.1:6653' --switch ovsk,protocols=OpenFlow13



OPEN DAYLIGHT SUMMIT

Summary

- OpenDaylight:
  - Controller Platform
  - Model-Driven Application Development Environment

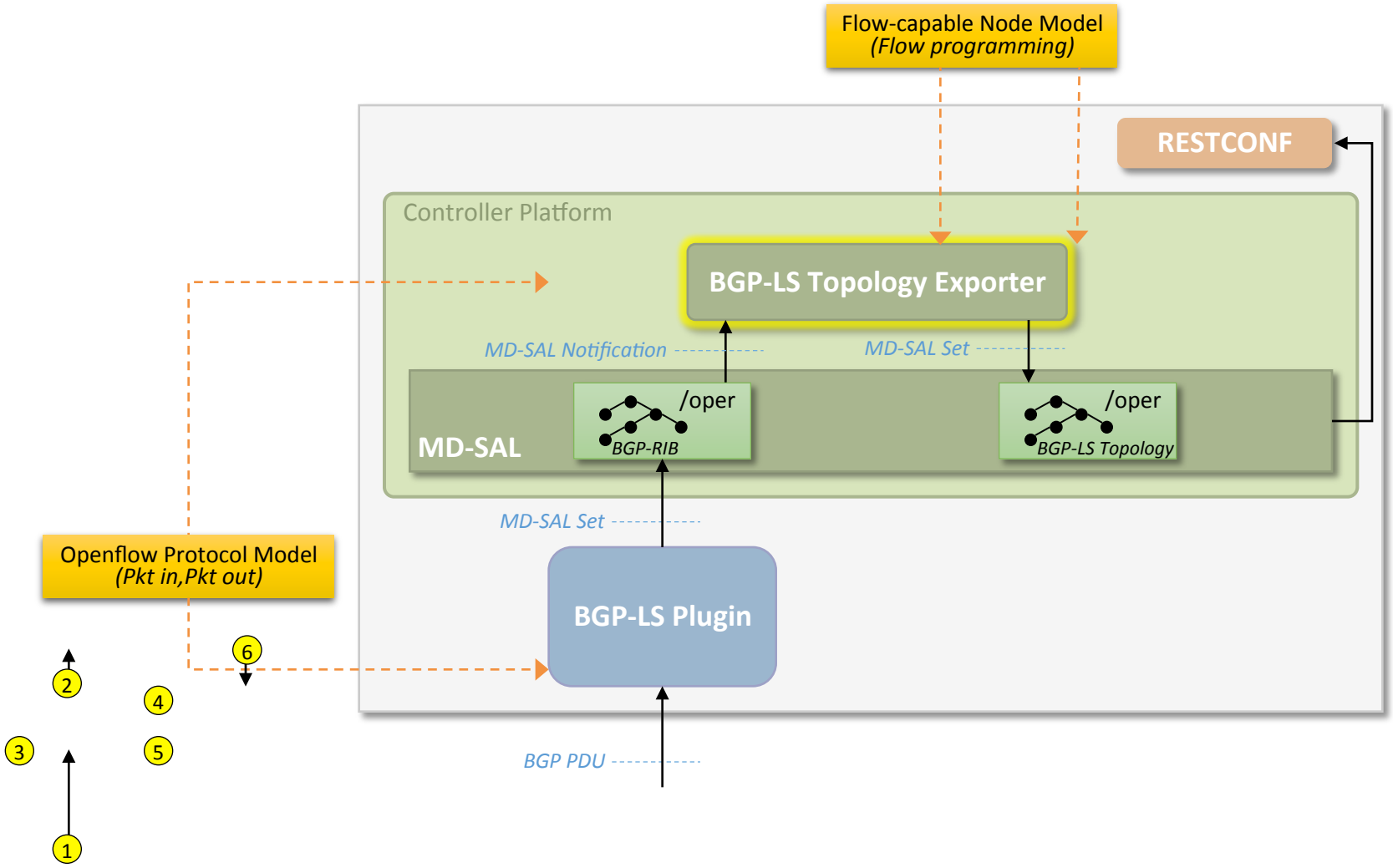


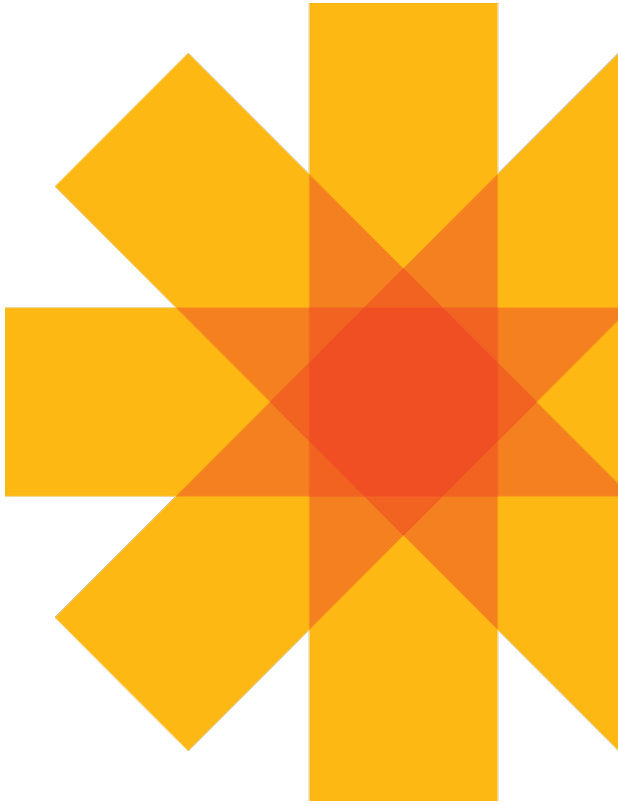
OPEN DAYLIGHT SUMMIT

Backup

OPEN DAYLIGHT SUMMIT

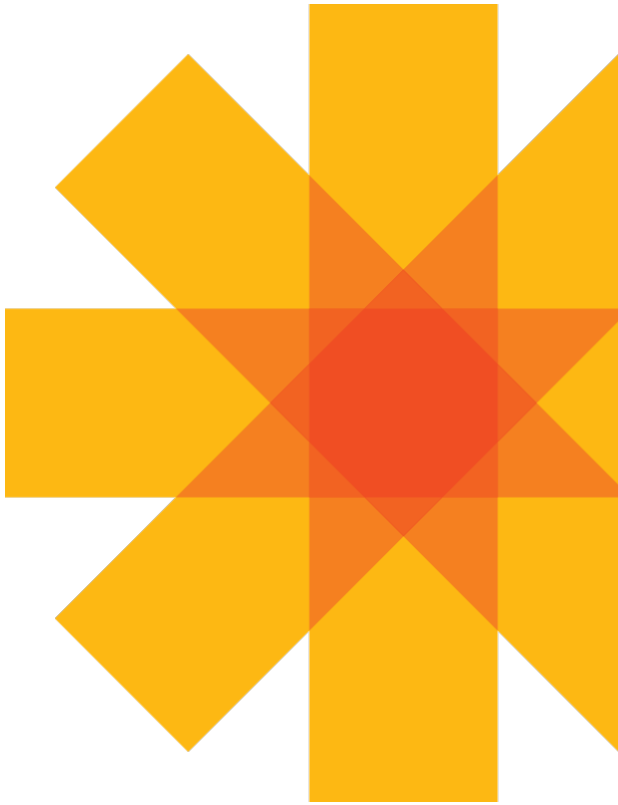
BGP-LS Flow





OPEN DAYLIGHT SUMMIT

BGP-LS System Flow



OPEN DAYLIGHT SUMMIT

[Sub-Section Title]

OPEN DAYLIGHT SUMMIT