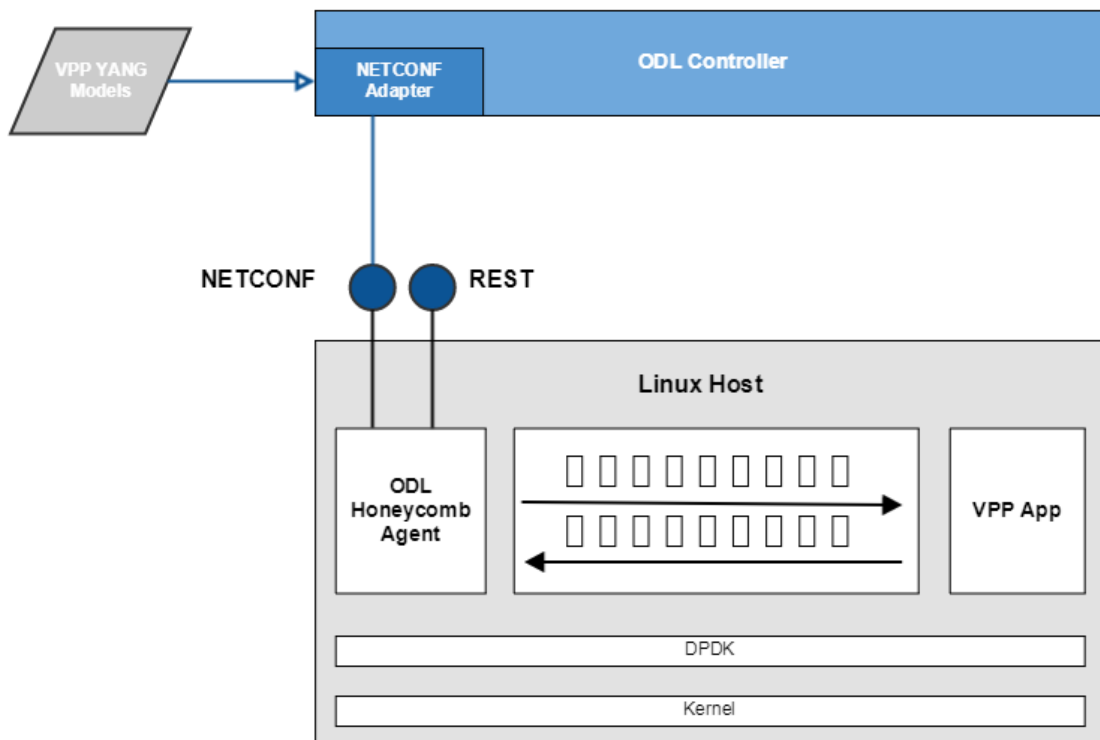# OpenDaylight NETCONF/YANG Adapter for VPP

## Abstract:

All NETCONF/YANG-enabled network functions - whether physical or virtualized - can be controlled by an OpenDaylight (ODL) SDN controller so long as an appropriate device adapter has been installed in the ODL controller. That ODL device adapter (or southbound plugin) can be generated from the network function's YANG model, and can be completed with Java programming.

Developers are developing such adapters for a range of NETCONF/YANG enabled devices. In this project, developers are able to develop the ODL device adapter for Vector Packet Processor (VPP), an open source, extensible framework that provides out-of-the-box production quality switch/router functionality. In order for VPP to be NETCONF managed by an ODL controller - it must be equipped with a generic NETCONF/RESTCONF java-based management agent (Honeycomb). The developer can then implement the VPP NETCONF adapter in ODL and manage VPP via NETCONF.

## Goals:

- Learn about:
  - Software Defined Network (SDN) - https://www.opennetworking.org
  - OpenDaylight SDN Controller – https://www.opendaylight.org/
  - NETCONF/YANG – https://www.ietf.org/slides/slides-edu-netconf-yang-00.pdf
  - VPP and Honeycomb – https://wiki.fd.io/view/VPP/What_is_VPP%3F
  - MEF LSO - https://wiki.mef.net/pages/viewpage.action?pageId=56165271
- Study relevant OpenDaylight tutorials including:
  - https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Startup_Project_Archetype
  - https://wiki.opendaylight.org/view/OpenDaylight_Controller:MD-SAL:Toaster_Step-By-Step
  - Optionally: https://devnetevents.cisco.com/events and https://learninglabs.cisco.com/modules/intro-device-level-interfaces
- Install VPP in a Linux environment
- Add the Honeycomb management agent to VPP
- Use the VPP YANG model(s) to develop the ODL device/function adapter with Java code
- Contribute resulting device/function adapter to the ODL Unimgr project
- Stretch goals

  - Test provisioning of layer 2 or layer 3 connectivity service using the device/function and ODL
  - Test orchestration of the device/function using LSO

## Requirements:

Java, Internet Networking Course

## Guided by:

Charles Eckel from

## About MEF:

The MEF is a collaborative code and specs development organization that starting in 2001 defined Carrier Ethernet and helped build the global Carrier Ethernet market now valued at $80Bn. MEF has 210 member companies, of which 130+ are the world's leading service providers, including AT&T, Verizon, Deutsche Telekom, BT, China Telecom, Cisco, Huawei, Ciena and many Israeli companies such as ECI, Amdocs, MRV, Gigaspaces, Contextream (HPE), RAD, Telco Systems and Ceragon. Carrier Ethernet services defined by the MEF and widely adopted by the telecoms industry include E-Line, E-LAN, E-Tree, E-Access and E-Transit.

The MEF's membership is now developing code and specs for **Third Networks** using NaaS (Network as a Service) principles in order to combine the agility and ubiquity of the Internet with the security and performance of Carrier Ethernet. Third Networks use a combination of LSO (Lifecycle Service Orchestration), SDN (Software Defined Networking) and NFV (Network Function Virtualization) to scale telecoms services to support the future tens of billions of IoT (Internet of Things) devices, billions of 5G users, and hundreds of millions of enterprises for whom cloud services and applications are essential.

### Important!

Developers working on this project will benefit from learning about industry leading technologies firsthand, and will be exposed to potential employment opportunities based on their experience in this project.