



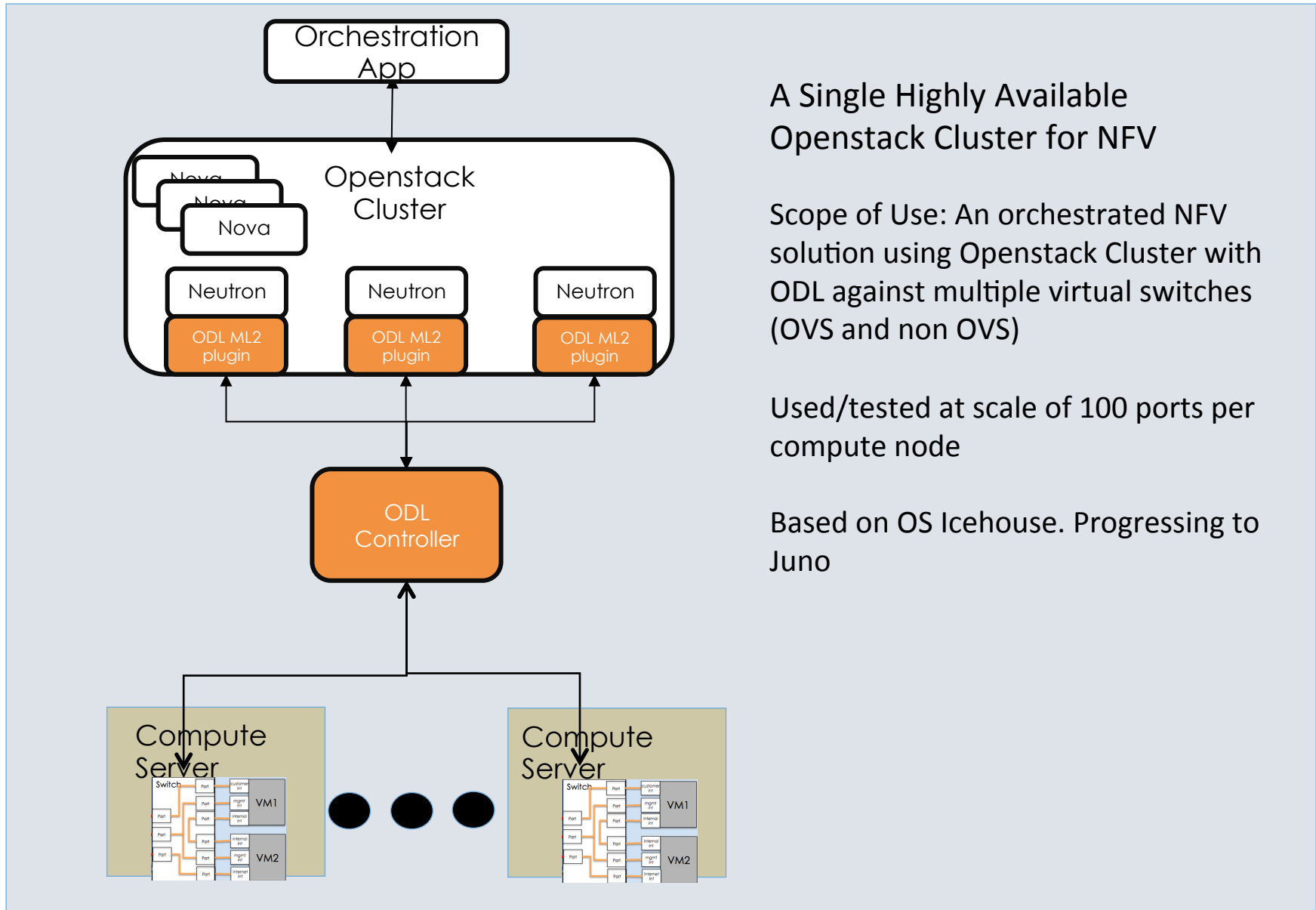
# Experiences in using Neutron ODL

Wojciech Dec – [wdec@cisco.com](mailto:wdec@cisco.com)

July 2015



# Setup Overview



A Single Highly Available Openstack Cluster for NFV

Scope of Use: An orchestrated NFV solution using Openstack Cluster with ODL against multiple virtual switches (OVS and non OVS)

Used/tested at scale of 100 ports per compute node

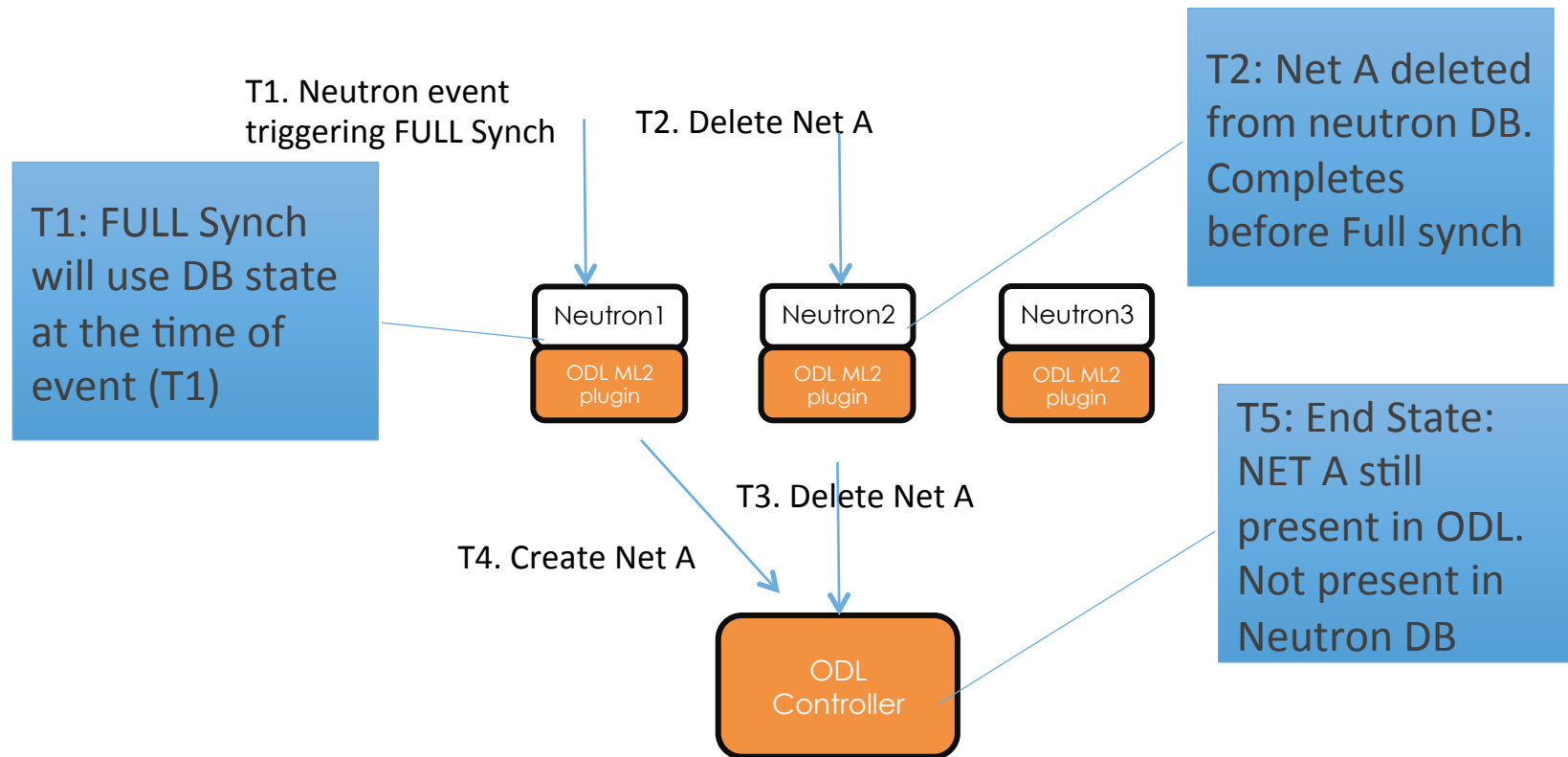
Based on OS Icehouse. Progressing to Juno

# Observation #1

- Current Openstack ML2 ODL driver is single threaded and blocks Neutron Server (post-commit) until ODL responds
  - Throughput of Neutron Requests gated by ODL ML2
  - Perhaps not a major issue for Network or subnets
- Can be addressed by running multiple Neutron servers but...
  - Concurrency and race-conditions...

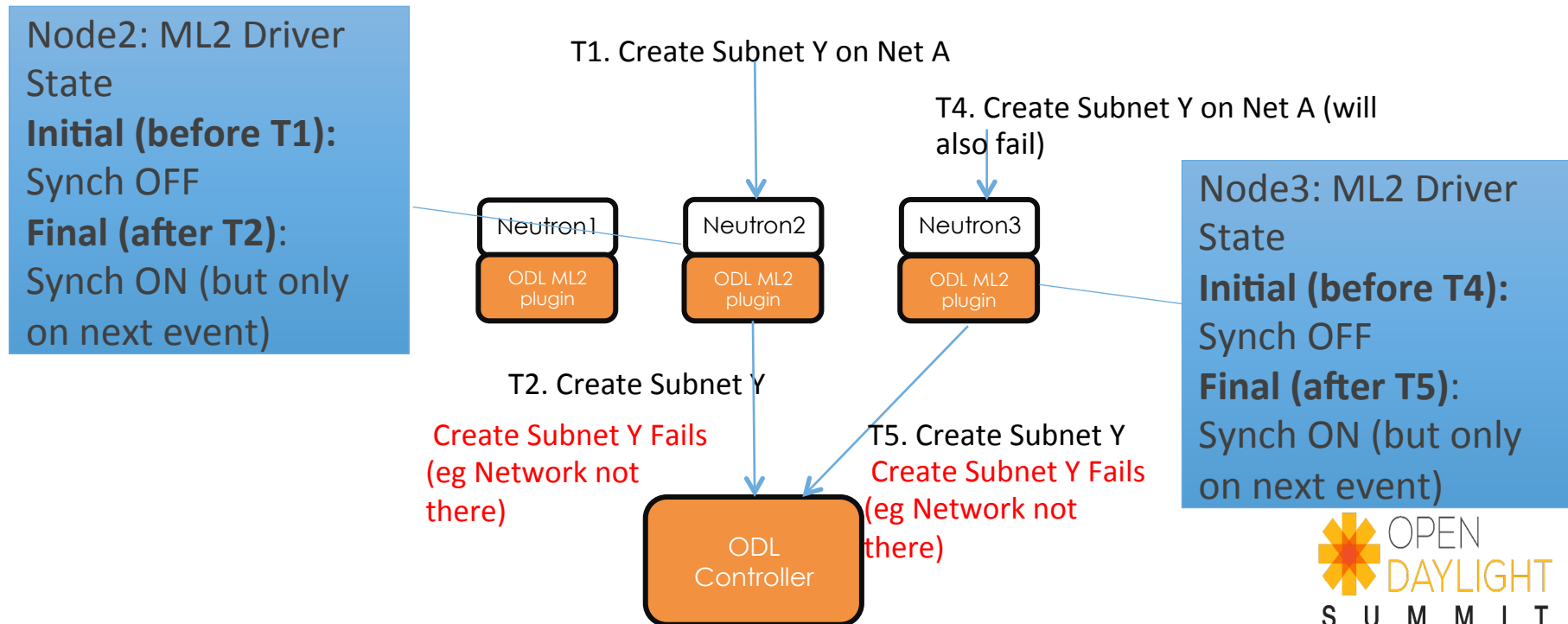
# Observation #2

- DB Synchronization concurrency issue



# Observation #3

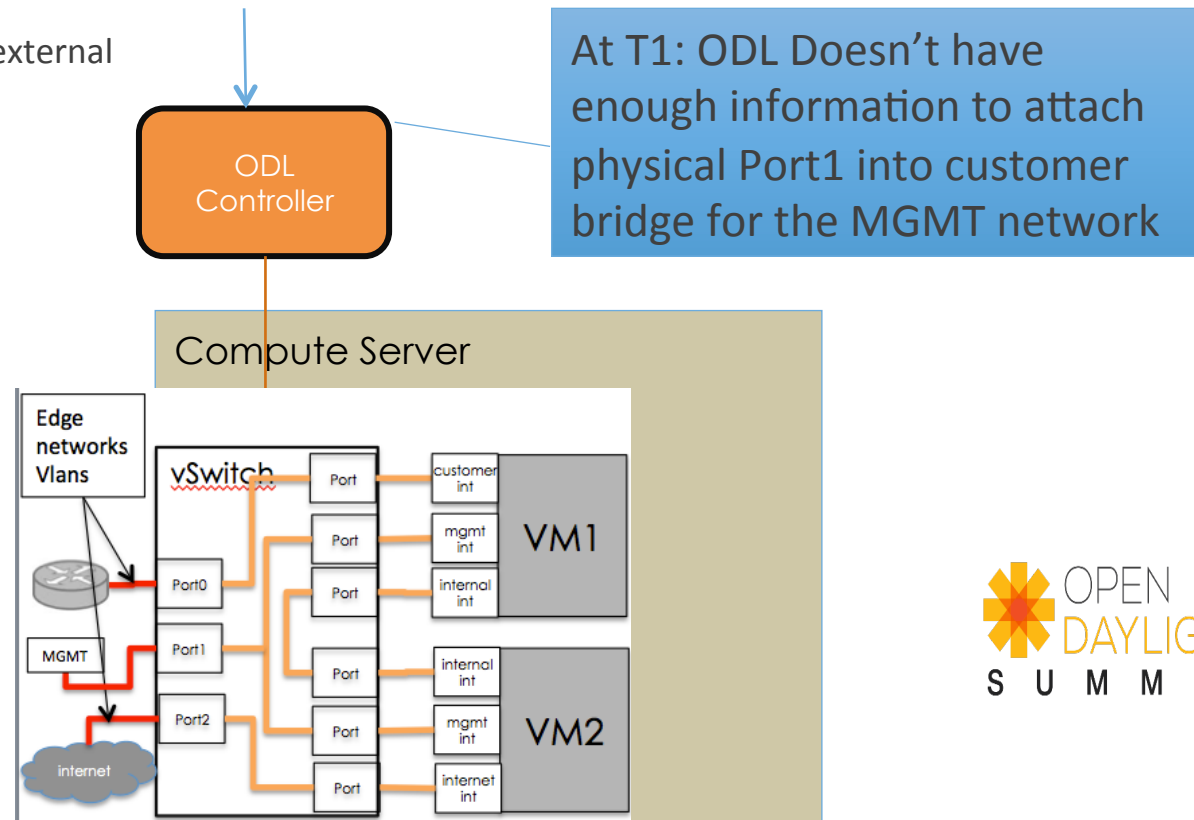
- Current ML2 ODL driver synch mechanism only run when triggered by a new Neutron event
  - leads to repeat failures



# Observation #4

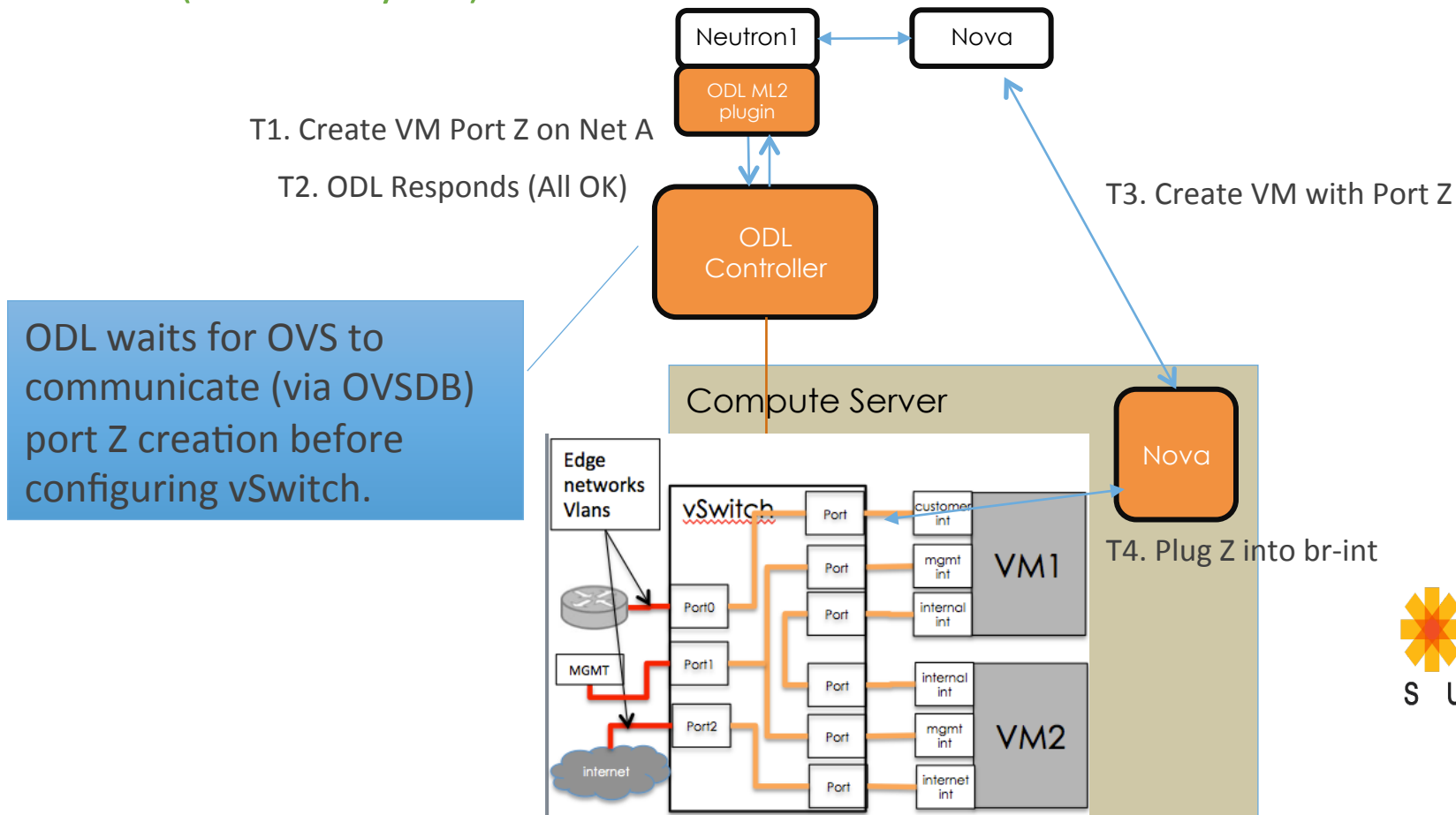
- No switch type independent configuration for mapping of Neutron physical networks to physical interfaces
  - Current solution: Network Configuration being switch specific and stored on compute nodes
  - Proposed Solution: An ODL based map configuration API, independent of switch type

T1. Create VM Port Z on Net A, with external Network MGMT



# Observation #5

- No feedback to Openstack regarding network rendering errors
  - Success of Neutron transaction <> Successfully rendered Network configuration. Openstack Neutron state not reflecting actual state
  - Particularly evident with OVS; network rendering only after OVSDb update (which may fail)



# Observation #6

- No ordered message delivery to ODL leading to “oddball” failures (i.e. race condition)

