Towards a Carrier Grade ONAP Platform
SDN Architectural Evolution

Key Contributors:
Ramki Krishnan, Bin Hu, Dan Timoney, Brian Freeman
Agenda

➢ Multi-vendor SDN Architectural Approach

➢ Multi-vendor & Cloud SDN ONAP Integration

➢ MC Networking Challenges & Solution Direction
Multi-vendor SDN Architectural Approach

SDN Domain Mapping
- 1 physical DC could have multiple SDN Domains
- Multiple latency bound physical DCs could map to a SDN Domain
- One Domain Controller per SDN domain

Multi-vendor Interconnect DB
- Store interconnect information between Multi-vendor Overlay/Underlay/Gateway Controllers, for e.g. Vlan
- Function similar to SDN-C for connecting MC instance Gateway and WAN Controller

OpenStack Adaptor
- Enable smooth transformation to new Modelling Language (TOSCA, YANG etc.)

Model Translation Plugin(s)
Flexible Southbound
SDN-DC
HW Switch/PNF Inventory
Multi-vendor Interconnect DB

OpenStack Adaptor
Overlay Controller
SDN-OC
SDN-UC
Domain Controller

OpenStack Adaptor

REST etc.
OVSDB, REST, Netconf etc.
OVSDB, REST, Netconf etc.

Overlay Controller

vSwitch
vSwitch
vSwitch
vSwitch
vSwitch

vSwitch Overlay

SDN Domain Mapping

SD-DC

Flexible Southbound

SDN-OC

Overlay Controller

SDN-UC

Underlay Controller

SDN-GC

Gateway Controller

HW Gateway

VM - SW Gateway
Multi-vendor & Cloud SDN ONAP Integration

SDN-DC external to ONAP
- Multi-vendor Interconnect DB in SDN-DC
- HW Switch/PNF inventory exposed to ONAP SDN-C for common management and fine grained control

SDN-DC function implemented in ONAP SDN-C
- Multi-vendor Interconnect DB in SDN-C
- SDN-C fully aware of Overlay/Underlay/Gateway including HW Switch/PNF inventory

Overlay/Underlay/Gateway fully abstracted
- HW Switch/PNF inventory *not* exposed to ONAP

Full SDN Domain Abstraction (Azure, GCP etc.)

Model Translation Plugin(s)
## MC Networking Challenges & Solution Direction

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Workflow</th>
<th>Need</th>
<th>Solution Direction</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNFs (HW Gateway etc.) and Underlay (HW Leaf/Spine switches/routers) not managed by MC instance</td>
<td>Day 1 &amp; Beyond (Init and Deploy)</td>
<td>Foundation</td>
<td>PNFs and Underlay to be accounted in A&amp;AI. SDN Domain controller function per MC instance (implemented in SDN-C or externally).</td>
</tr>
<tr>
<td>Lack of standardized APIs for MC instance Underlay Networking</td>
<td>All</td>
<td>Foundation</td>
<td>Standardized Data Models (TOSCA, YANG etc.), Semantics and Model-driven APIs for MC instance Underlay. Model Translation plugins translate to appropriate data model for specific Underlay Controller. Underlay configuration can be Layer 2 (MLAG etc.) or Layer 3 (eBGP etc.).</td>
</tr>
<tr>
<td>Underlay Network Configuration per MC instance not automated as part of common workflow</td>
<td>Day 1 (Init)</td>
<td>Foundation</td>
<td>Follow steps similar to WAN underlay Init; need to do this per MC instance.</td>
</tr>
</tbody>
</table>