Hybrid Orchestration Model for NFVi

Adam Dunstan
Centurylink’s NFVi Evolution

PSB – VMWare
Based on VMware
Single Use Case - FW
No Orchestration

PSB – OSP
Based on Openstack
Full Stack platform
All VM use cases

PSB – OSN
Based on Openstack
Full Stack platform
NFVi only use cases
PSB - OSN

NFVi Domain

- Domain Orchestration
- Domain Management
- Platform Manager

Switches

Openstack Platform

NFVi PODS
Master System Orchestration

- Autonomous functional domains
- Each domain provides complete functionality
- Simplifies Master Orchestrator & other shared functions
- Common shared components & domain specific tooling used
NFVi POD’s

• Openstack
  • Packaged from source
  • CentOS/KVM
  • Minimal components
  • Optimized for packet forwarding

• Centurylink SOS
  • White box switches – BRCM OCP
  • Internally developed NOS
  • Unique management tooling

Note: during installation
Workflow – New NFVi POD – 2 days

Physical - 0.5 day
- Rack, stack & wire
- "Ikea Install"

Install - 0.5 day
- Switches (ONIE)
- Remediate & OS (RHD)
- Openstack (Ansible)

Test/Burn - 24hours
- Continuous test
- Activate, send packets, destroy
- Expect 100% success

Hand to Operational Management
- Begin deploying VNF’s
Hybrid Orchestration

- Management functionality distributed in two platforms
- Victor functionality focused on CRUD
- Elvira has complete visibility of NFVi platform
- Use cases requiring synchronization driven by configured vs operational state comparison
- Other CRUD platforms do co-exist

*Victor knows what it created, Elvira knows what actually exists*
VICTOR

- Micro-service
- Combine multiple configuration mechanisms
- Graphical Service Designer
- UI & WS-API auto-generated from Service Model
- Support for chaining
Auto-Generated UI

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Customer Name</th>
<th>NPI</th>
<th>CID</th>
<th>Status</th>
<th>Completed At</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Service 1]</td>
<td>[Customer 1]</td>
<td>[NPI 1]</td>
<td>[CID 1]</td>
<td>[Status 1]</td>
<td>[Completed 1]</td>
</tr>
<tr>
<td>[Service 2]</td>
<td>[Customer 2]</td>
<td>[NPI 2]</td>
<td>[CID 2]</td>
<td>[Status 2]</td>
<td>[Completed 2]</td>
</tr>
<tr>
<td>[Service 3]</td>
<td>[Customer 3]</td>
<td>[NPI 3]</td>
<td>[CID 3]</td>
<td>[Status 3]</td>
<td>[Completed 3]</td>
</tr>
</tbody>
</table>

Swagger UI
Service Designer
Auto Generated WS-API
Vertically Integrated (VNF Onboarding)
Summary

• Decoupling simplified Control Infrastructure development
• Orchestration horizontally focused on Infrastructure provisioning
• Abstract network functionality to WS-API for IT acceleration
• Uses a couple of components from ONAP
• Micro-service application designed for containerization
• Releasing VICTOR to Opensource