

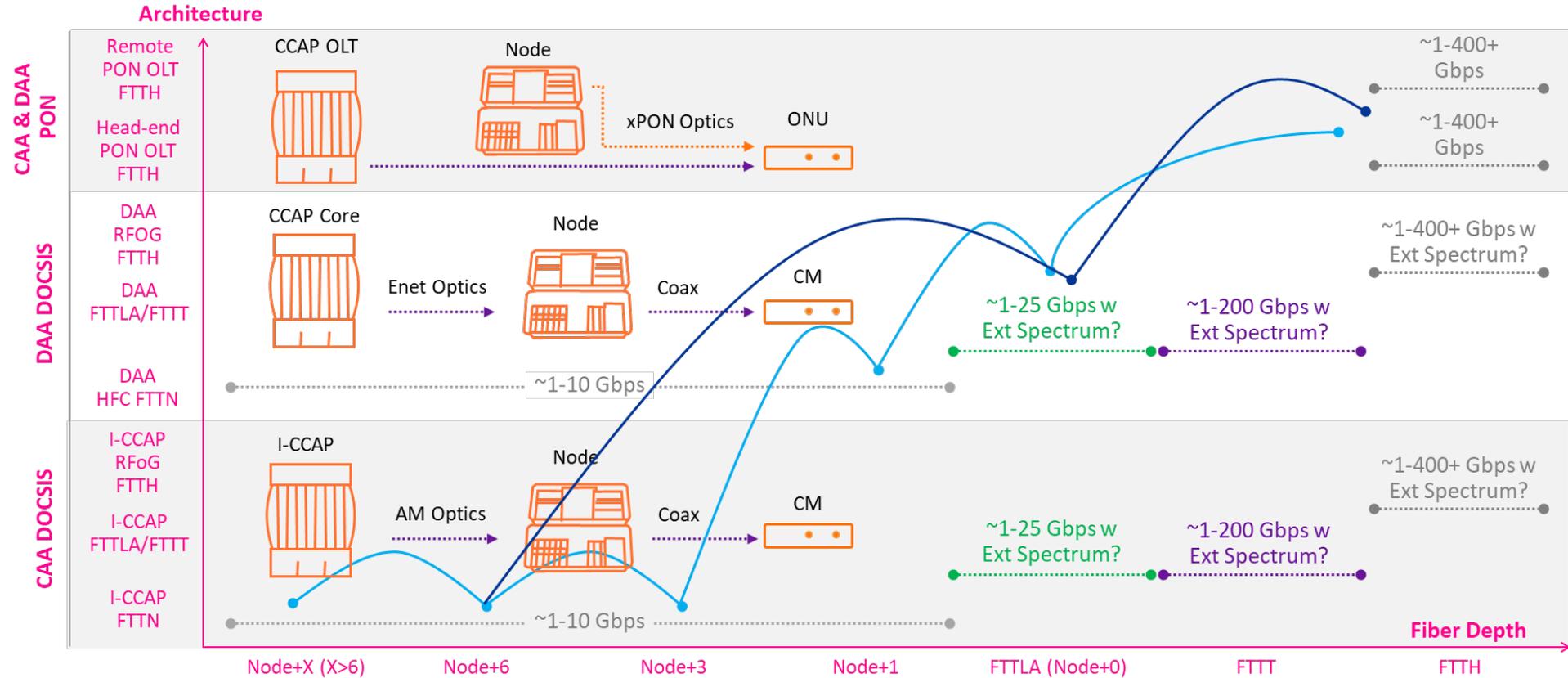


Bringing DOCSIS to SEBA

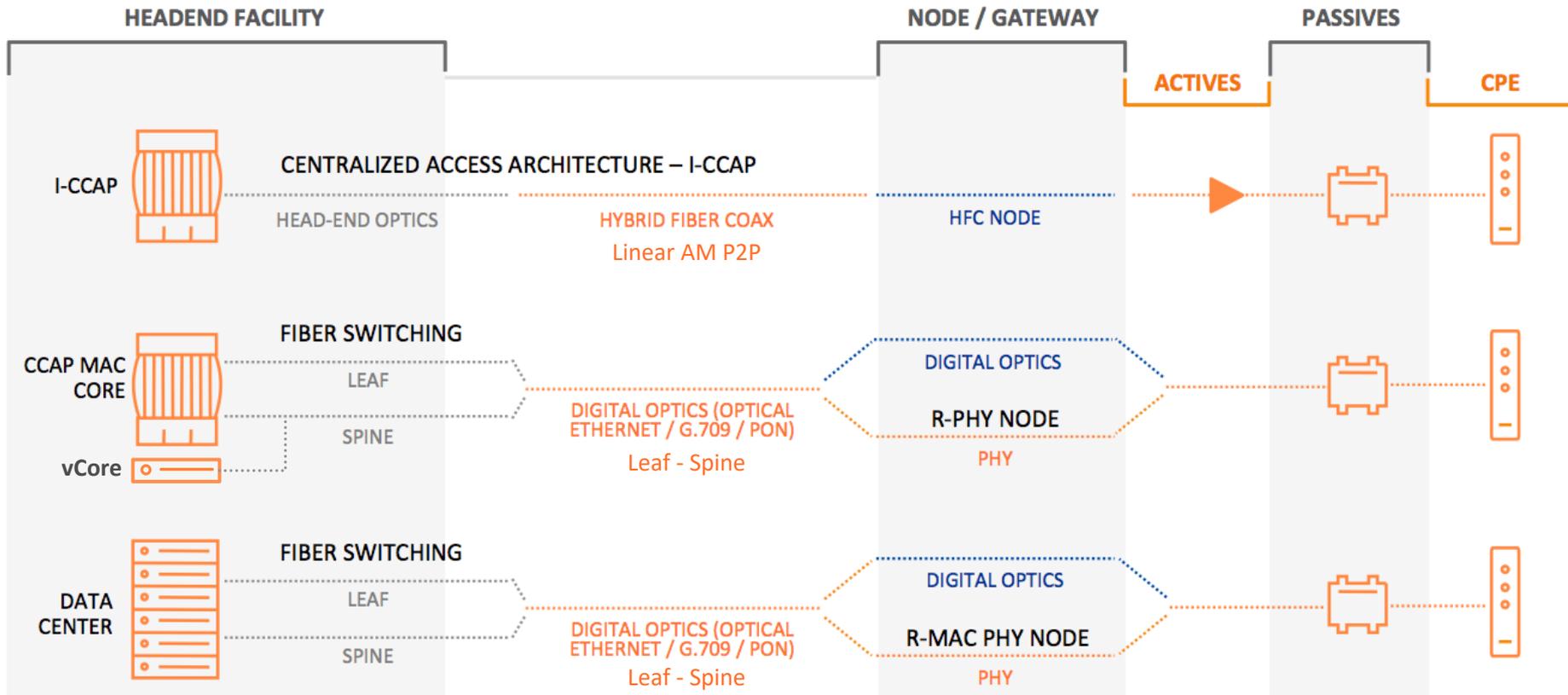
How Cable DAA & PON Come Together

Chris Busch Jaspreet Sachdev
CommScope

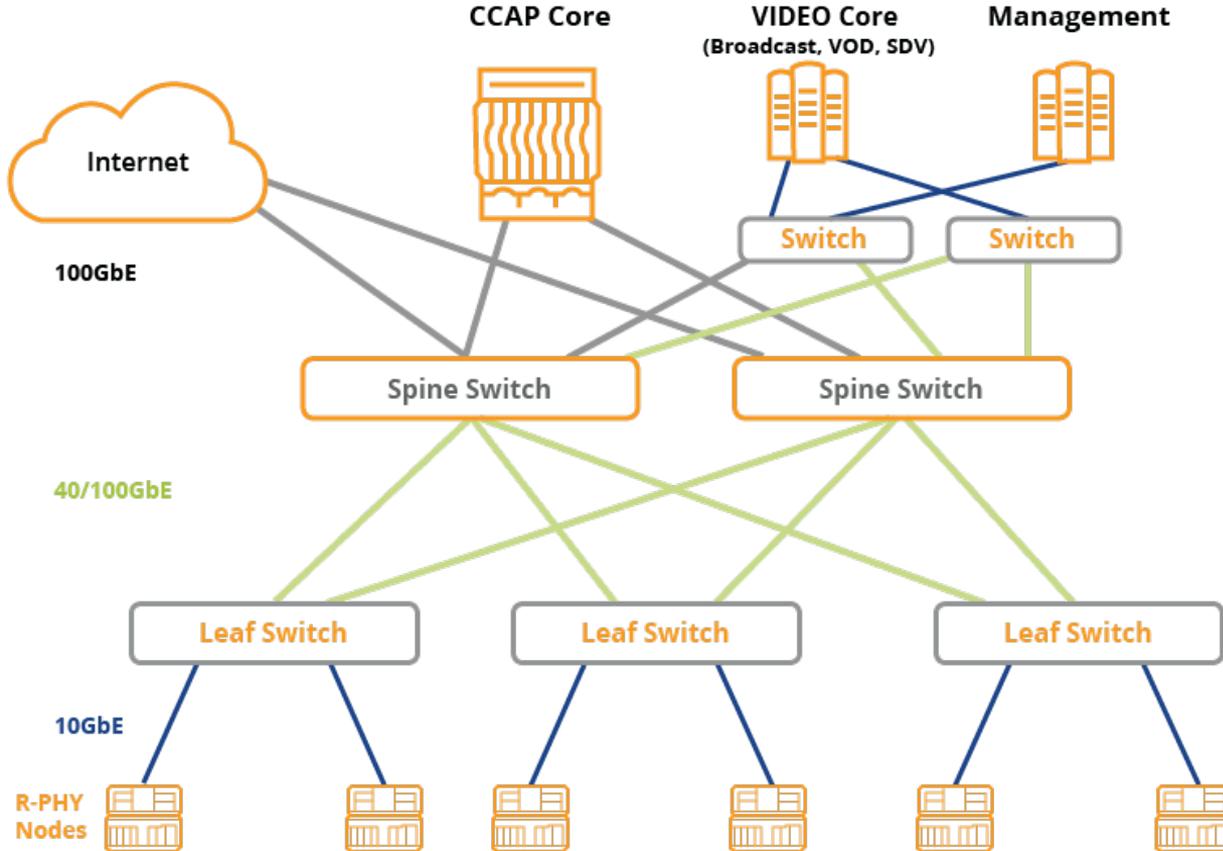
Cable Access Architecture Options



HFC to DAA to Virtualization



Top Level DAA Systems

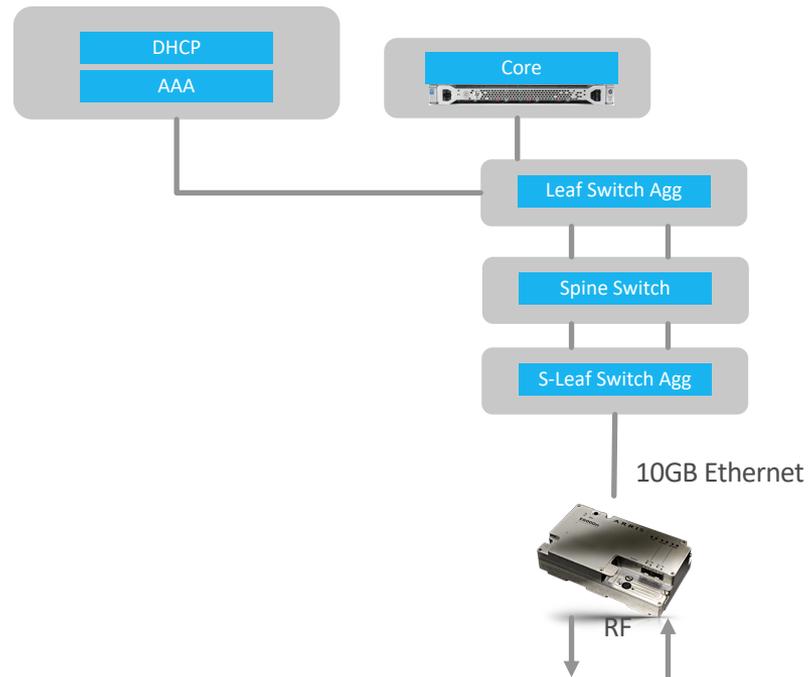


Tier 1 Operator Request

- Customer:
 - “We want to orchestrate services regardless of access technology”
 - “How can we deploy DOCSIS DAA and also PON in VOLTHA?”
- Options:
 - Disaggregate the Remote-PHY Device
 - Very hard to see this happen – CableLabs governs Remote-PHY standards
 - Re-Use the Remote-PHY Device Control Plane
 - Work with current control plane – make VOLTHA ‘DOCSIS Aware’

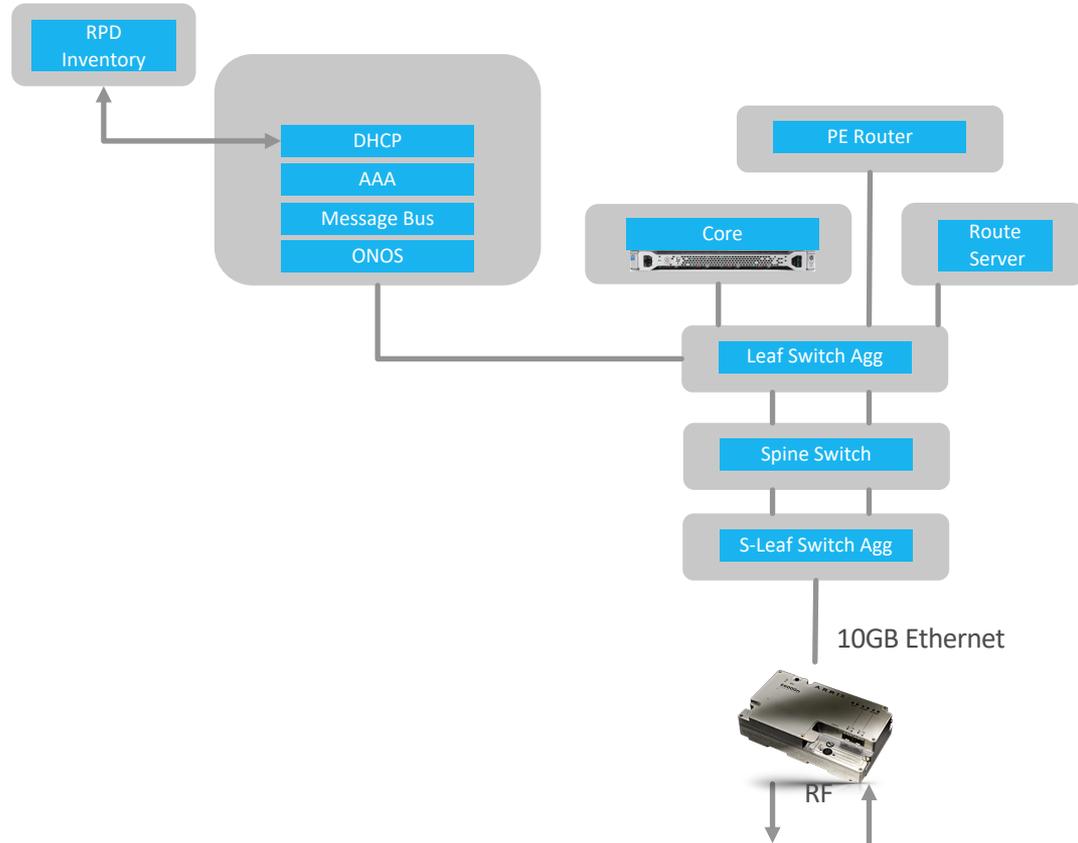
How Remote-PHY Devices Initialize

- 10GigE Optical Link(s) + MACSEC s-Leaf
- 802.1x Authenticated EAP CableLabs Root-CA per s-Leaf port
- DHCP direction RPD to Cores
 - RPD Connects with a Generic Configuration Protocol
 - GCP TLVs configure RF and any other operating parameters in RPD
- Provisioning an RPD needs GCP
 - This can be a Core or a Provisioning Service
- Onboarding an RPD end to end can be a SEBA solution



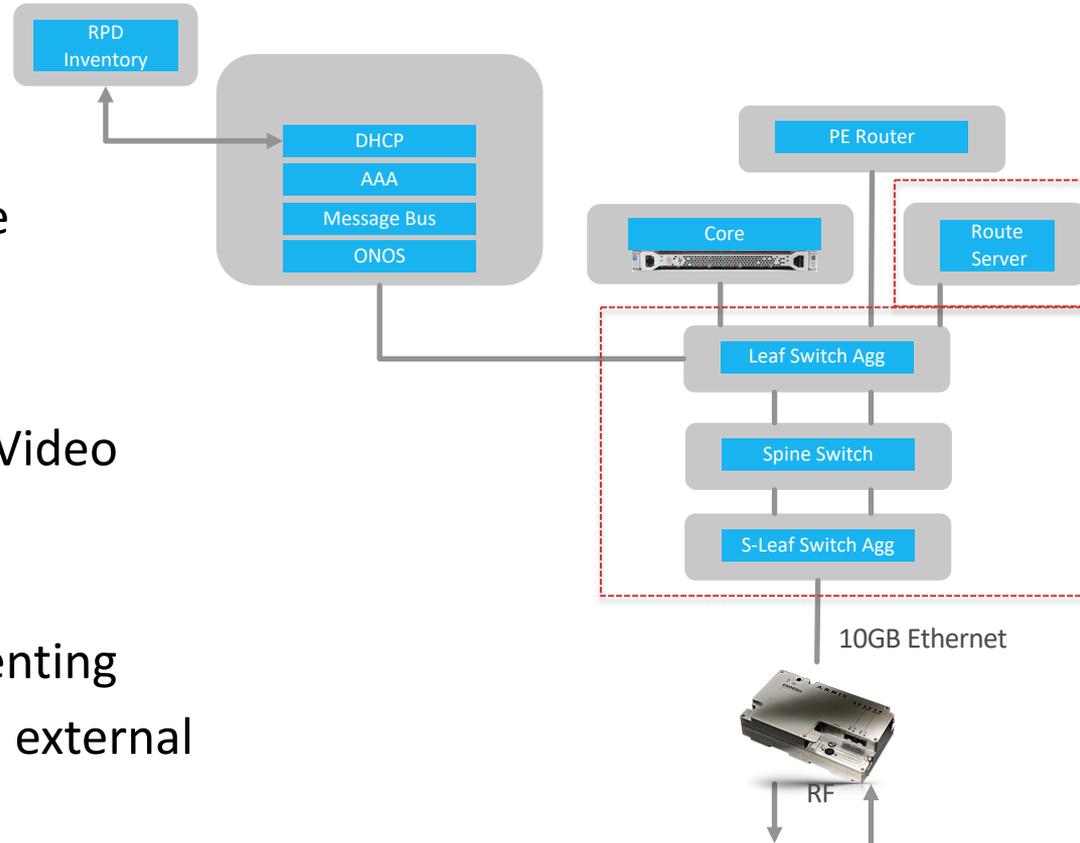
Trellis Remote-PHY Devices Initialize

- RPD attach to s-Leaf
- Packet_IN
- AAA
- DHCP
- Message Bus



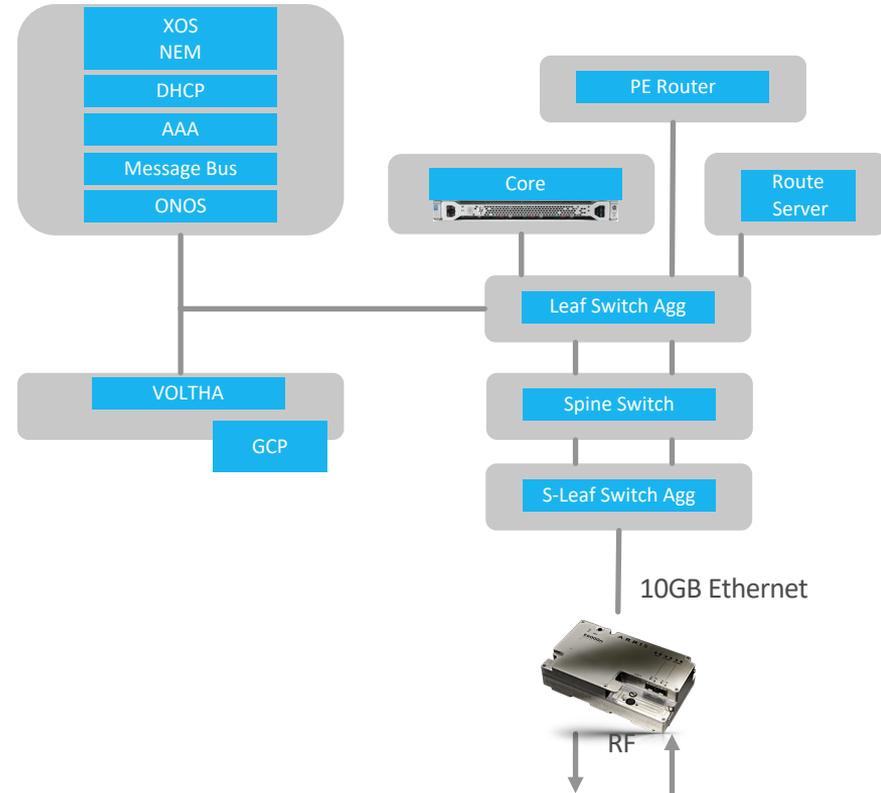
Trellis Remote-PHY Devices Initialize

- With Trellis
 - RPD Onboarding fairly simple
 - Segment Routing Topology
 - Route Server + Fabric Peer
 - Multicast Traffic support for Video Distribution
 - Traffic Visibility
 - De-coupled Message Bus Eventing
 - DHCP remains dependent on external actor for RPD Core Direction



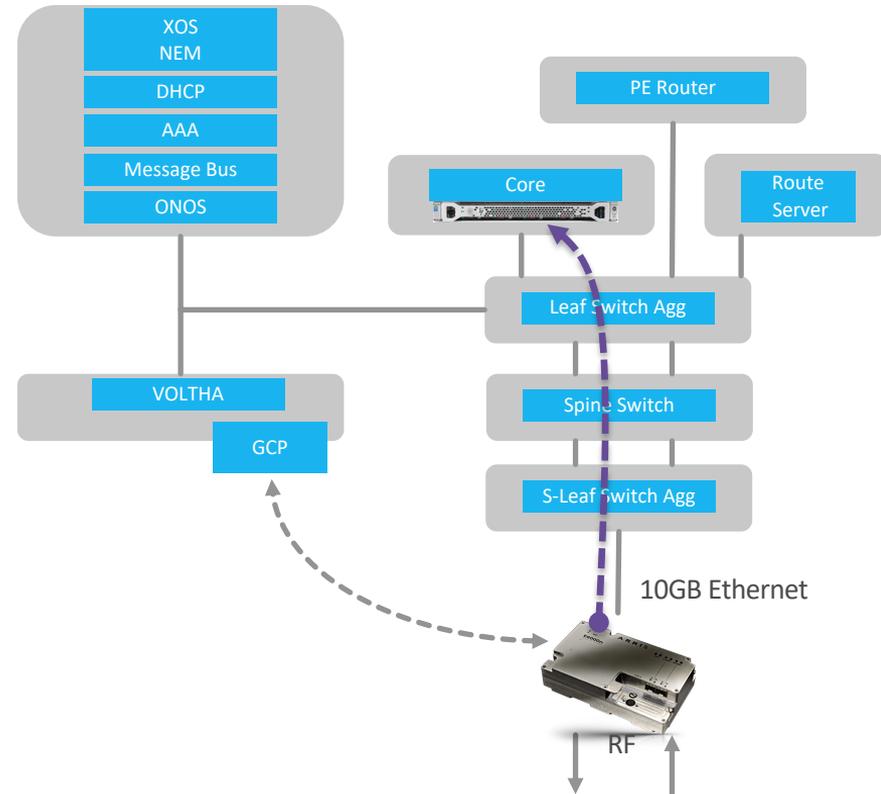
SEBA Remote-PHY Devices Initialize

- Introducing SEBA
 - XOS and NEM
 - Orchestration possible among actors
 - Fabric cross connect service
 - K8s VNF onboarding
 - Inventory
 - RPD as a NEM 'vOLT'
 - vCore as a NEM 'vBNG'
 - VOLTHA
 - Create GCP Adaptor
- External to SEBA
 - CommScope vManager DAA Inventory



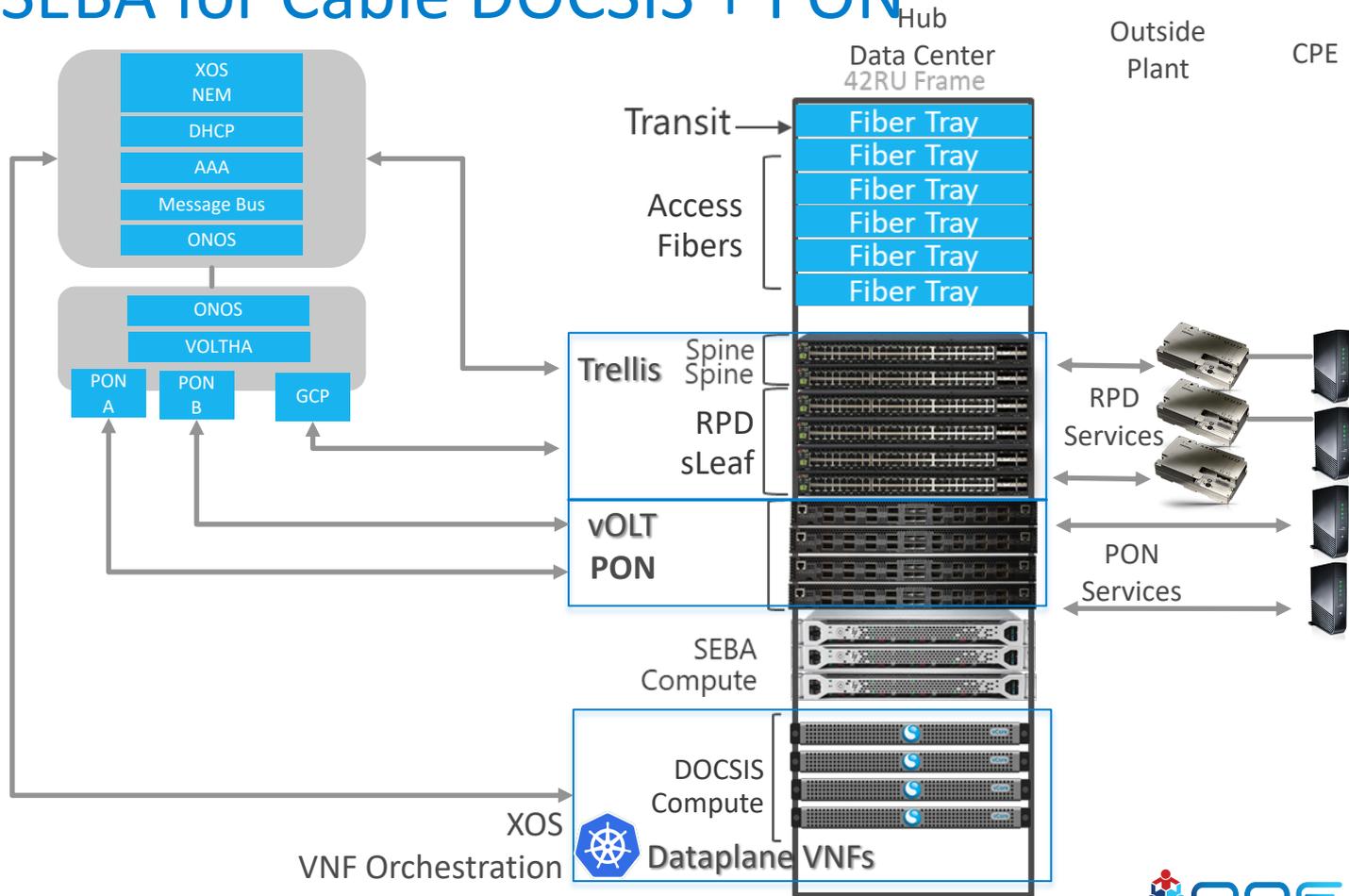
SEBA Remote-PHY Devices Initialize

- SEBA for DAA
 - DHCP answers ALL RPD device types with VOLTHA GCP IP Address
 - XOS RPD as a Service
 - RPD Inventory association with vCore VNF on K8s
 - Incoming GCP
 - VOLTHA speaks GCP_init pushes Packet_IN
 - GCP via VOLTHA Core or Not
 - Can be directed via NEM for vCore
 - Can make external Inventory lookup
 - Lookup for Core
 - Causes GCP Redirect TLV to RPD
 - Results in VOLTHA Message to Bus
 - Enables RPD Service over Trellis to be synchronized



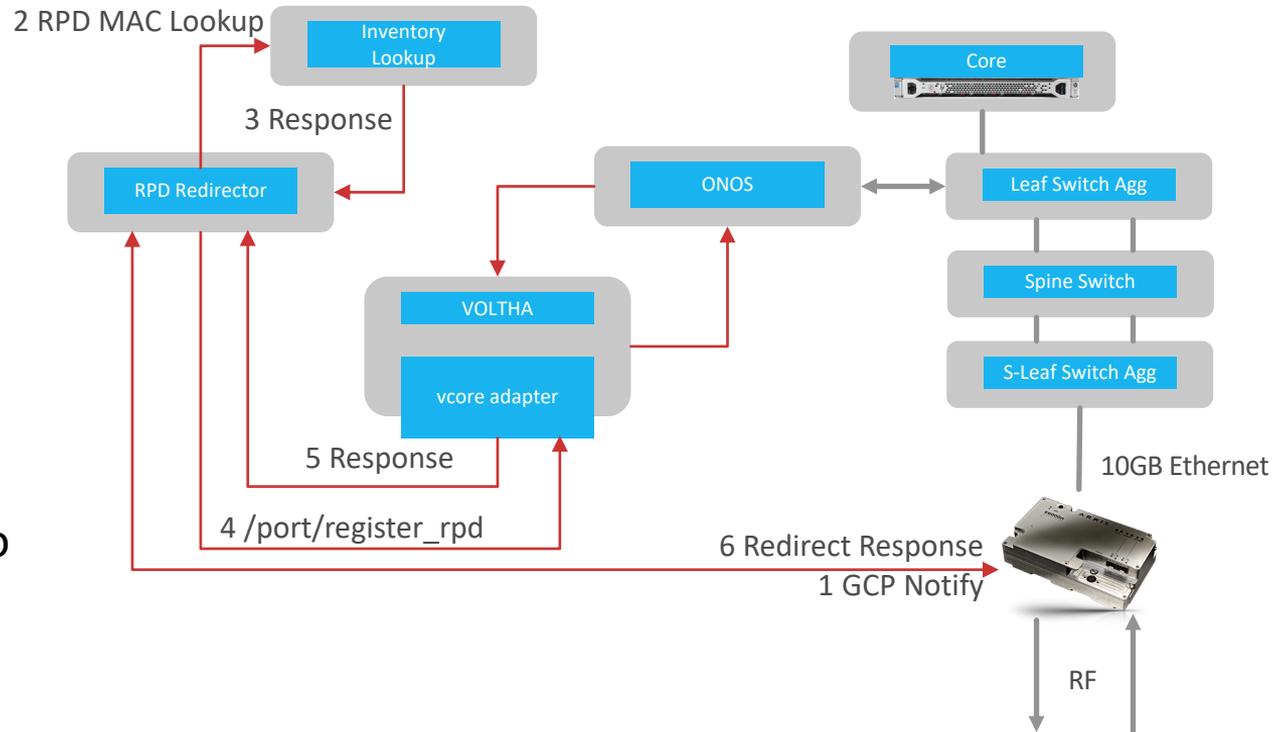
SEBA for Cable DOCSIS + PON

- Common Fabric
 - Trellis
- XOS
 - API to DOCSIS VNF
 - ATT Workflow for PON
- VOLTHA
 - Multi-vendor / white box PON
 - GCP Redirect DOCSIS
- Common Onboarding DOCSIS and PON



VOLTHA GCP Design

- vCore Adapter
 - New VOLTHA port 1881 for GCP communications
- RPD Redirector
 - Standalone service
- NOTIFY_URL sends request to VOLTHA port 1881
- REST_URL lookup to DAA Inventory
 - Future NEM



VOLTHA Perspective with RPD

```
harris@R2SEBA:~/cord/helm-charts$ kubectl exec -it vol1-67c996f87d-v79df -n voltha su - voltha
VOLTHA (GIT)
(to exit type quit or hit Ctrl-D)
(voltha) adapters
adapters:
-----
| id | vendor | version |
-----
| acme | Acme Inc. | 1.36 |
| adtran_olt | ADTRAN, Inc. | 1.36 |
| adtran_onu | ADTRAN, Inc. | 1.25 |
| asfvoltc_olt | Edgeworks | 0.98 |
| bcmc_openomci_onu | Voltha project | 0.50 |
| broadcom_onu | Voltha project | 0.46 |
| cig_openomci_onu | CIG Tech | 0.10 |
| dpoe_onu | Sumitomo Electric, Inc. | 0.1 |
| maple_olt | Voltha project | 0.4 |
| microsemi_olt | Microsemi / Celestica | 0.2 |
-----
| openolt | OLT white box vendor | 0.1 |
| pmcs_onu | PMCS | 0.1 |
-----
| ponsim_olt | Voltha project | 0.4 |
| ponsim_onu | Voltha project | 0.4 |
| rpd | ComScope | 0.1 |
| simulated_olt | Voltha project | 0.1 |
| simulator_onu | Voltha project | 0.1 |
| tellabs_olt | Tellabs Inc. | 0.1 |
| tellabs_openomci_onu | Tellabs Inc. | 0.1 |
| tibit_olt | Tibit Communications Inc. | 0.1 |
-----
| tibit_onu | Tibit Communications Inc. | 0.1 |
| tlgs_onu | TLGS | 0.1 |
-----
(voltha)
```

Load VOLTHA RPD Adapter

Preprovision RPD

```
(voltha) preprovision_olt -t rpd
success (device id = 0001fbcc9bbf0ddc)
(voltha) enable
enabling 0001fbcc9bbf0ddc
waiting for device to be enabled...
waiting for device to be enabled...
success (device id = 0001fbcc9bbf0ddc)
```

List RPD Devices

```
(voltha) devices
Devices:
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | type | root | parent_id | serial_number | mac_address | admin_state | oper_status | connect_status |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 0001fbcc9bbf0ddc | rpd | True | 0001000ce2314000 | 63e8dcf584eb442bb6b5512c060de365 | 00:0c:e2:31:40:00 | ENABLED | ACTIVE | REACHABLE |
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
(voltha)
```

Verify Device UNI/NNI

```
(voltha) device 0001fbcc9bbf0ddc
(device 0001fbcc9bbf0ddc) ports
Device ports:
-----+-----+-----+-----+-----+-----+
| port_no | label | type | admin_state | oper_status | device_id |
-----+-----+-----+-----+-----+-----+
| 1 | NNI facing Ethernet port | ETHERNET_NNI | ENABLED | ACTIVE | 0001fbcc9bbf0ddc |
| 2 | UNI port | ETHERNET_UNI | ENABLED | ACTIVE | 0001fbcc9bbf0ddc |
-----+-----+-----+-----+-----+-----+
(device 0001fbcc9bbf0ddc)
```

ONOS RPD View

- VOLTHA representing the RPD as an OpenFlow Switch

/v1/devices

```
{
  "id": "of:0000000ce2314000",
  "type": "SWITCH",
  "available": true,
  "role": "MASTER",
  "mfr": "VOLTHA Project",
  "hw": "RPD",
  "sw": "RPD",
  "serial": "3905326883914afe9e77c6f71b6b46e7",
  "driver": "voltha",
  "chassisId": "ce2314000",
  "lastUpdate": "1567038377833",
  "humanReadableLastUpdate": "connected 17m35s ago",
  "annotations": {
    "channelId": "192.168.0.69:58392",
    "managementAddress": "192.168.0.69",
    "protocol": "OF_13"
  }
}
```

Next Development Goals

- GCP
 - Move GCP into VOLTHA 2.x as an adapter
 - Extend to full AUX Core for ongoing provisioning
 - Adds ongoing counters & telemetry to VOLTHA for RPDs (pushed to ONOS & Bus)
- XOS
 - Extend to support DOCSIS VNF Lifecycle
- NEM
 - Extend to support RPD as Inventory Device
 - OLT Device Type refactored for Remote PHY Devices
 - Extend to support DOCSIS VNF
 - vBNG Device Type refactored for DOCSIS Virtualization



Thank You