Development Process and Testing

Session 4: 3:30pm - 5pm

In this session

- What does a typical SEBA development loop look like?
- How do we test functionality end-to-end? How do we test system scale?
- What are the next steps for SEBA? (last 30 minutes)

Development Loop

- Make changes to component source
- Pass unit tests for that component
- Build new container image for the component
- Deploy new container in SiaB
- Pass integration/E2E tests
- Lab #4 will walk through this process

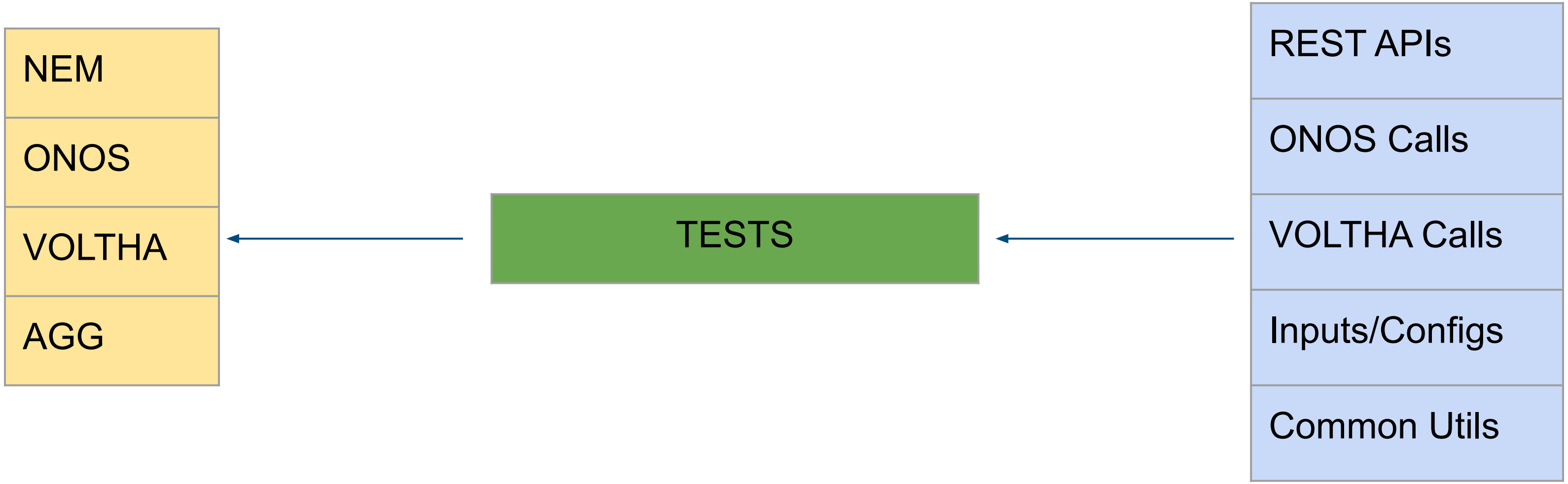
Cord-Tester

- The cord-tester is a test-automation framework written in Robot Framework and Python
- Test suites to validate functionality, regression, and stability of various components of CORD (xos, onos, voltha, siab, ponsim, etc.)
- Shared libraries between multiple test suites (physical pod, siab, xos-api-tests, integration-tests, etc)
- Current E2E test suites based on the ATT-Workflow

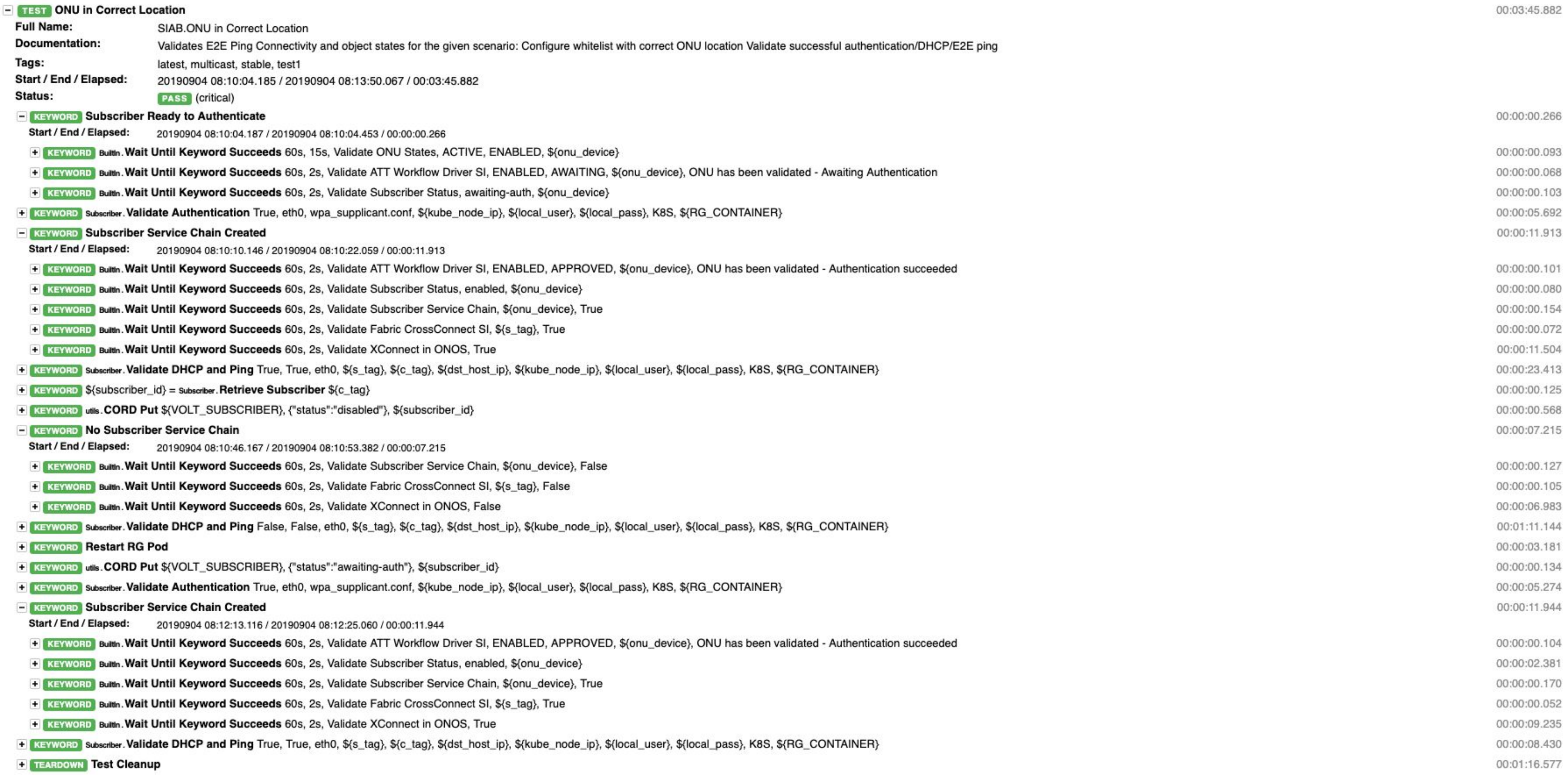
Example SEBA Test w/Framework Reference

- Push required configurations (tosca or json)
  - Tosca files (SEBA for various components - fabric,olt,subscribers etc)
  - Json (inputs for POD configurations)
- Validate states in voltha, onos and NEM for devices and subscribers
  - Checks in voltha and onos cli by sending various commands
- Authenticate
  - Login to the subscriber/specified host and send auth request via wpa_supplicant
  - Validate command output status
  - Validate NEM states
- DHCP
  - Send dhclient commands from the specified host
  - Validating command success states, verifying assigned address on the interface
  - Validate NEM states
- Pings
  - Send and validate ping requests

Current SEBA Tests

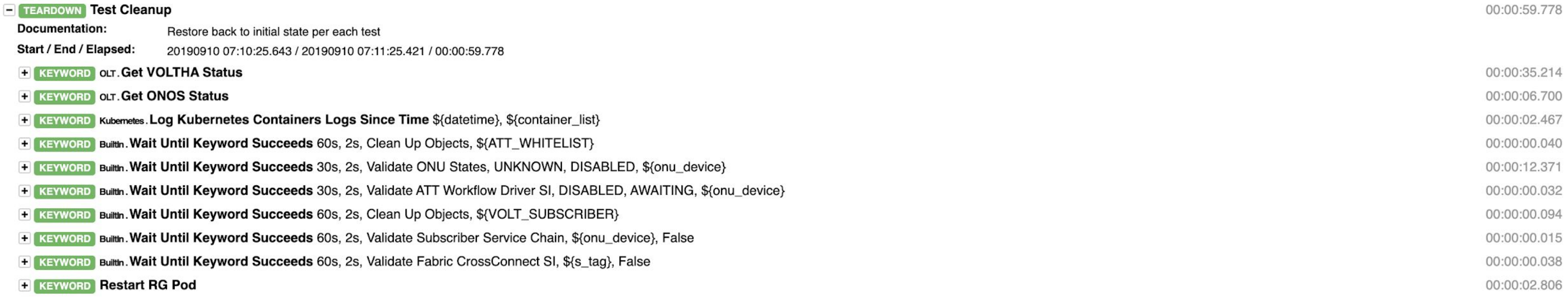


Test Case #1



```
− TEST ONU in Correct Location                                                                                                   00:03:45.882
Full Name:              SIAB.ONU in Correct Location
Documentation:          Validates E2E Ping Connectivity and object states for the given scenario: Configure whitelist with correct ONU location Validate successful authentication/DHCP/E2E ping
Tags:                   latest, multicast, stable, test1
Start / End / Elapsed:  20190904 08:10:04.185 / 20190904 08:13:50.067 / 00:03:45.882
Status:                 PASS (critical)
  − KEYWORD Subscriber Ready to Authenticate                                                                                     00:00:00.266
    Start / End / Elapsed:  20190904 08:10:04.187 / 20190904 08:10:04.453 / 00:00:00.266
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 15s, Validate ONU States, ACTIVE, ENABLED, ${onu_device}                 00:00:00.093
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate ATT Workflow Driver SI, ENABLED, AWAITING, ${onu_device}, ONU has been validated - Awaiting Authentication   00:00:00.068
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Status, awaiting-auth, ${onu_device}             00:00:00.103
  + KEYWORD Subscriber. Validate Authentication True, eth0, wpa_supplicant.conf, ${kube_node_ip}, ${local_user}, ${local_pass}, K8S, ${RG_CONTAINER}   00:00:05.692
  − KEYWORD Subscriber Service Chain Created                                                                                     00:00:11.913
    Start / End / Elapsed:  20190904 08:10:10.146 / 20190904 08:10:22.059 / 00:00:11.913
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate ATT Workflow Driver SI, ENABLED, APPROVED, ${onu_device}, ONU has been validated - Authentication succeeded   00:00:00.101
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Status, enabled, ${onu_device}                   00:00:00.080
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Service Chain, ${onu_device}, True               00:00:00.154
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Fabric CrossConnect SI, ${s_tag}, True                      00:00:00.072
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate XConnect in ONOS, True                                      00:00:11.504
  + KEYWORD Subscriber. Validate DHCP and Ping True, True, eth0, ${s_tag}, ${c_tag}, ${dst_host_ip}, ${kube_node_ip}, ${local_user}, ${local_pass}, K8S, ${RG_CONTAINER}   00:00:23.413
  + KEYWORD ${subscriber_id} = Subscriber. Retrieve Subscriber ${c_tag}                                                          00:00:00.125
  + KEYWORD utils. CORD Put ${VOLT_SUBSCRIBER}, {"status":"disabled"}, ${subscriber_id}                                          00:00:00.568
  − KEYWORD No Subscriber Service Chain                                                                                          00:00:07.215
    Start / End / Elapsed:  20190904 08:10:46.167 / 20190904 08:10:53.382 / 00:00:07.215
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Service Chain, ${onu_device}, False              00:00:00.127
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Fabric CrossConnect SI, ${s_tag}, False                     00:00:00.105
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate XConnect in ONOS, False                                     00:00:06.983
  + KEYWORD Subscriber. Validate DHCP and Ping False, False, eth0, ${s_tag}, ${c_tag}, ${dst_host_ip}, ${kube_node_ip}, ${local_user}, ${local_pass}, K8S, ${RG_CONTAINER}   00:01:11.144
  + KEYWORD Restart RG Pod                                                                                                       00:00:03.181
  + KEYWORD utils. CORD Put ${VOLT_SUBSCRIBER}, {"status":"awaiting-auth"}, ${subscriber_id}                                     00:00:00.134
  + KEYWORD Subscriber. Validate Authentication True, eth0, wpa_supplicant.conf, ${kube_node_ip}, ${local_user}, ${local_pass}, K8S, ${RG_CONTAINER}   00:00:05.274
  − KEYWORD Subscriber Service Chain Created                                                                                     00:00:11.944
    Start / End / Elapsed:  20190904 08:12:13.116 / 20190904 08:12:25.060 / 00:00:11.944
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate ATT Workflow Driver SI, ENABLED, APPROVED, ${onu_device}, ONU has been validated - Authentication succeeded   00:00:00.104
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Status, enabled, ${onu_device}                   00:00:02.381
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Service Chain, ${onu_device}, True               00:00:00.170
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate Fabric CrossConnect SI, ${s_tag}, True                      00:00:00.052
    + KEYWORD Builtin. Wait Until Keyword Succeeds 60s, 2s, Validate XConnect in ONOS, True                                      00:00:09.235
  + KEYWORD Subscriber. Validate DHCP and Ping True, True, eth0, ${s_tag}, ${c_tag}, ${dst_host_ip}, ${kube_node_ip}, ${local_user}, ${local_pass}, K8S, ${RG_CONTAINER}   00:00:08.430
  + TEARDOWN Test Cleanup                                                                                                        00:01:16.577
```

Test Clean-up

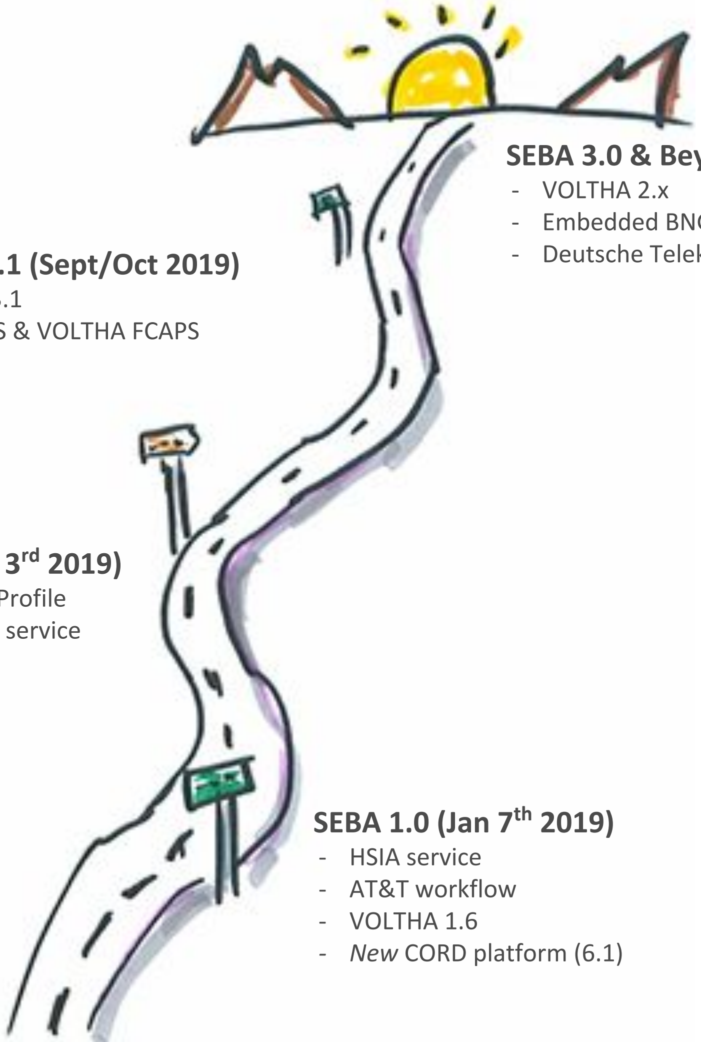


```
− TEARDOWN Test Cleanup                                                                                                      00:00:59.778
  Documentation:         Restore back to initial state per each test
  Start / End / Elapsed: 20190910 07:10:25.643 / 20190910 07:11:25.421 / 00:00:59.778
  + KEYWORD OLT.Get VOLTHA Status                                                                                            00:00:35.214
  + KEYWORD OLT.Get ONOS Status                                                                                              00:00:06.700
  + KEYWORD Kubernetes.Log Kubernetes Containers Logs Since Time ${datetime}, ${container_list}                              00:00:02.467
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 60s, 2s, Clean Up Objects, ${ATT_WHITELIST}                                  00:00:00.040
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 30s, 2s, Validate ONU States, UNKNOWN, DISABLED, ${onu_device}               00:00:12.371
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 30s, 2s, Validate ATT Workflow Driver SI, DISABLED, AWAITING, ${onu_device}  00:00:00.032
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 60s, 2s, Clean Up Objects, ${VOLT_SUBSCRIBER}                                00:00:00.094
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 60s, 2s, Validate Subscriber Service Chain, ${onu_device}, False             00:00:00.015
  + KEYWORD BuiltIn.Wait Until Keyword Succeeds 60s, 2s, Validate Fabric CrossConnect SI, ${s_tag}, False                    00:00:00.038
  + KEYWORD Restart RG Pod                                                                                                   00:00:02.806
```

Open Networking Foundation

SEBA Roadmap

Roadmap



SEBA 3.0 & Beyond

- VOLTHA 2.x
- Embedded BNG
- Deutsche Telekom/Turk Telecom Workflows

SEBA 2.1 (Sept/Oct 2019)

- BAL 3.1
- ONOS & VOLTHA FCAPS

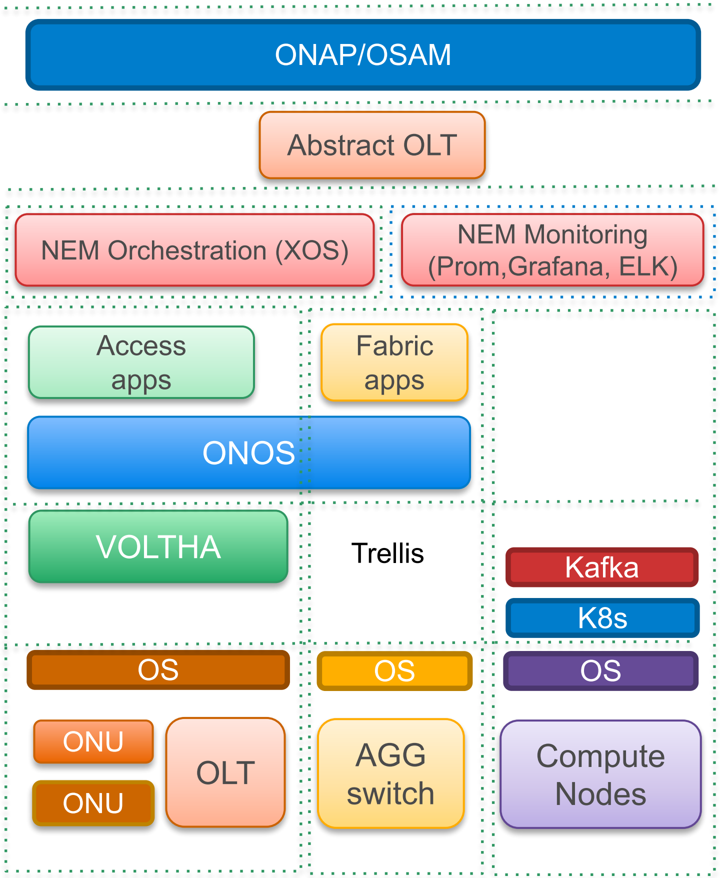
SEBA 2.0 alpha (July 3rd 2019)

- Technology & Speed Profile
- AT&T workflow, HSIA service
- VOLTHA 1.7, BAL 2.6

SEBA 1.0 (Jan 7th 2019)

- HSIA service
- AT&T workflow
- VOLTHA 1.6
- *New* CORD platform (6.1)

SEBA Modularity



Accelerating SEBA: MVP Requirements & Gaps

https://docs.google.com/document/d/1eM-Sdy4dAQVm9oP1i7V7H_LXKbyD2S7juIZOdFiPmXk/edit?ts=5d5dad90#

	TT	DT	ATT	Gap (Y=Yes)
Timeline				
Features desired by	YE2019	YE2019	YE2019	
Production with commercial subscribers	End 2019	End Q1 2020	End Q2 2020	
Features (docs, tests & API) (must include openolt)				
HSIA	✓	✓	✓	N
EAPOL			✓	N
Subscriber DHCP	✓		✓	N
multi-gem	✓	✓	✓	Coming to voltha 2.x
VoIP, VoD	✓	✓	✓	N
GPON (BAL3.1 dep)	✓	✓		in the works (M)

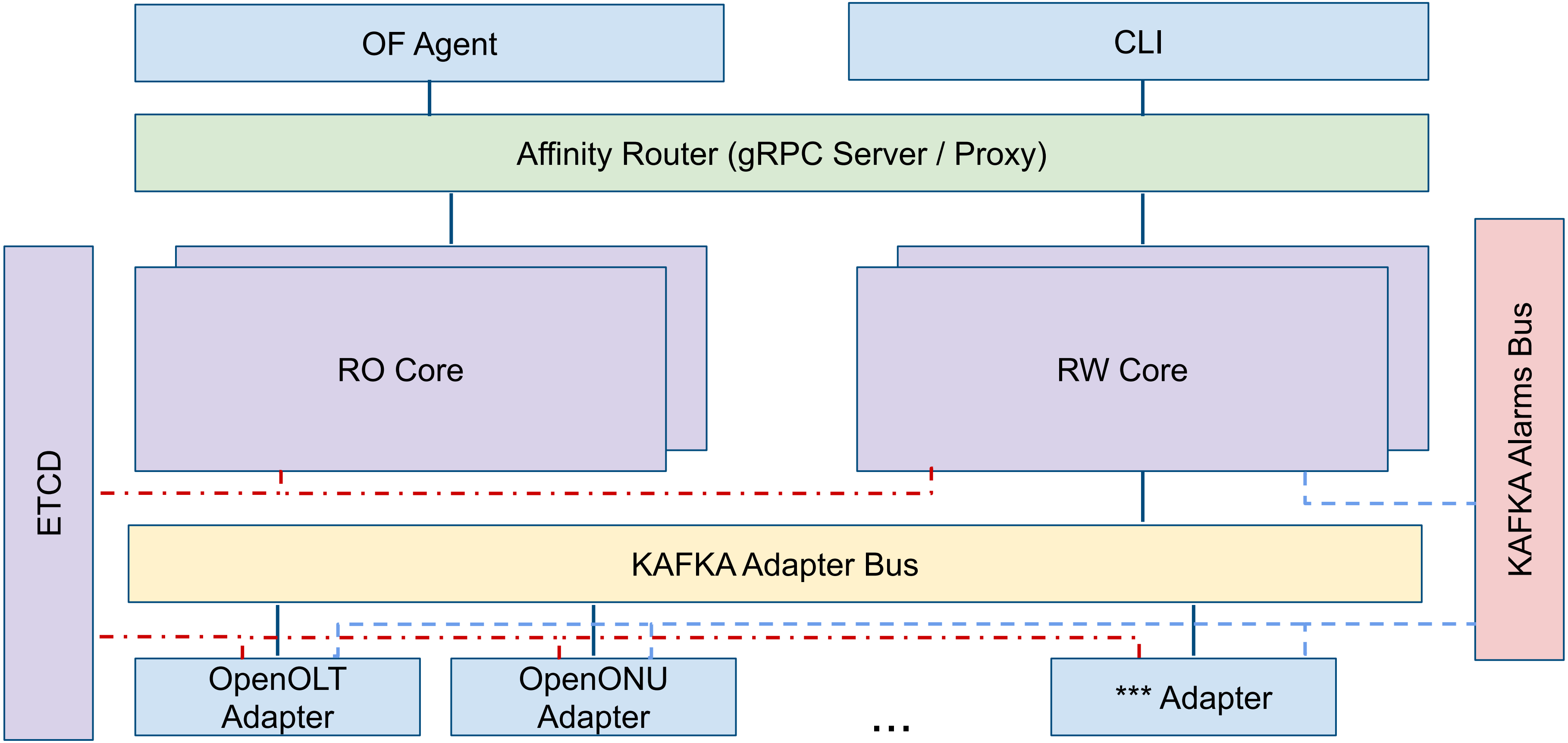
VOLTHA 2.x Integration

- VOLTHA 2.x was not at feature parity when SEBA 2.0 alpha was released.
- Community working extensively on stabilization of VOLTHA codebase, brigade focused on this.
- Current plan is to integrate when feature parity and stability goals are reached.

Key VOLTHA 2.x Changes

- Significant portions rewritten in go
- New High Availibility model
  - disaggregated core: api-server, rwcores, rocores
  - active-active to ensure fast failover
- New event and performance metric format
  - Protobuf-encoded Kafka messages

VOLTHA 2.x: architecture overview



VOLTHA 2.x: RW Core

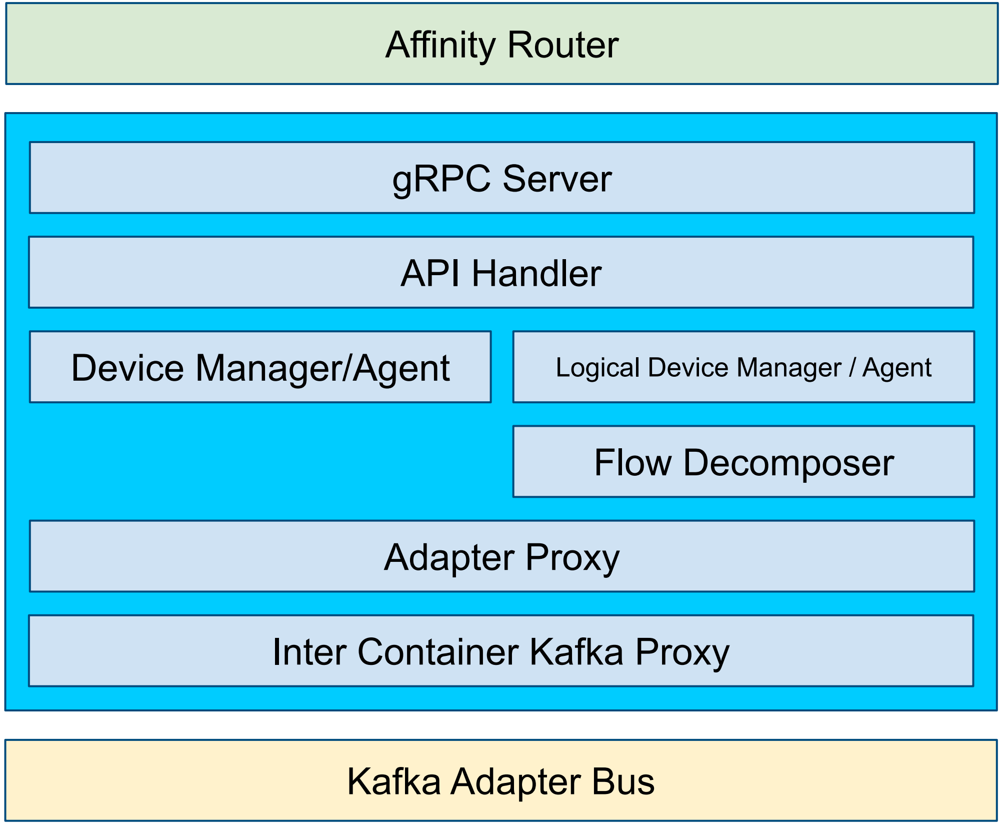
For each request* **Affinity Router** sends a message to 2 cores.

API Handler receives the message and decide whether to execute it or keep it in standby.

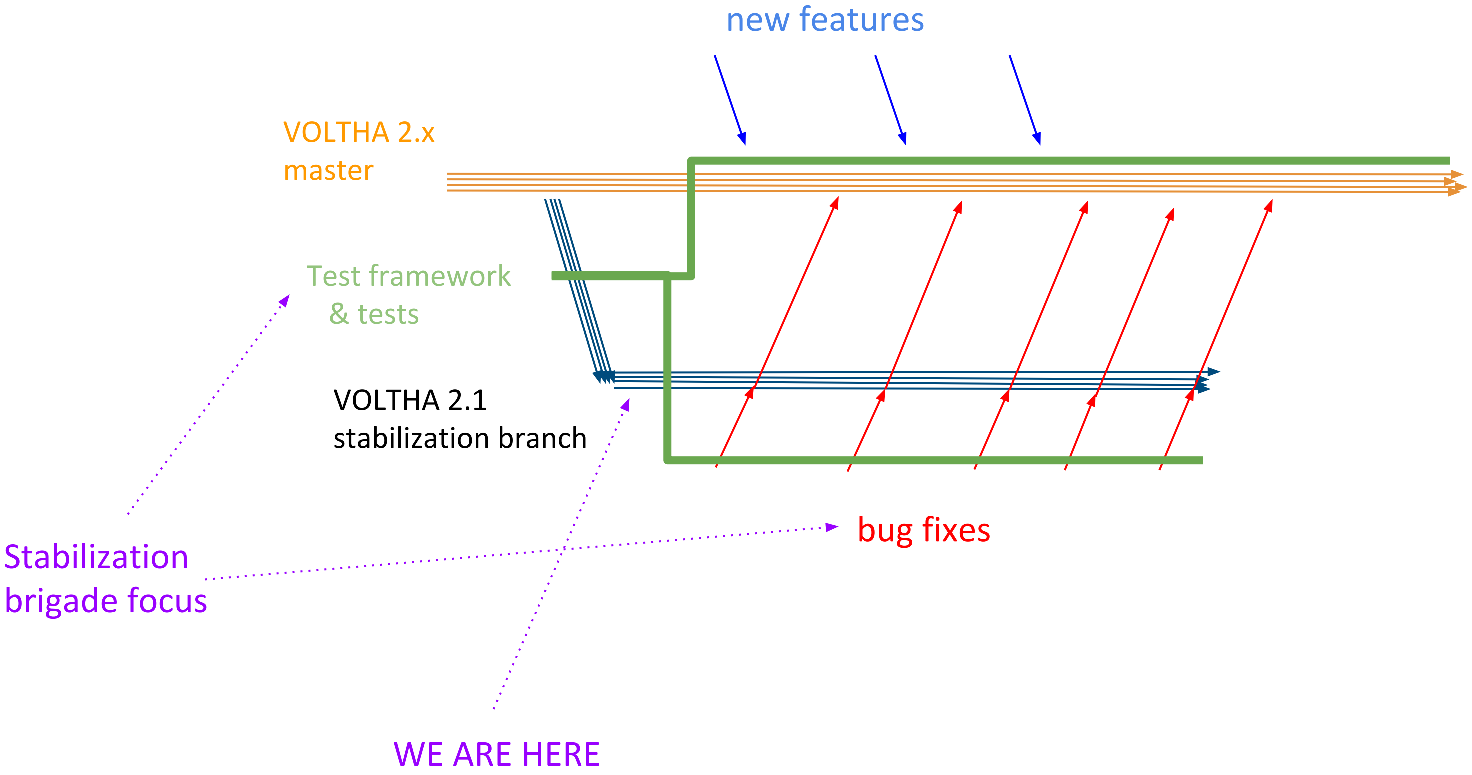
Depending on the request it can be executed by:

- **Device Agent** Manages the physical device lifecycle
- **Logical Device Agent (Flow Decomposer)** Manages the logical switch lifecycle and map the flows to the underlying topology

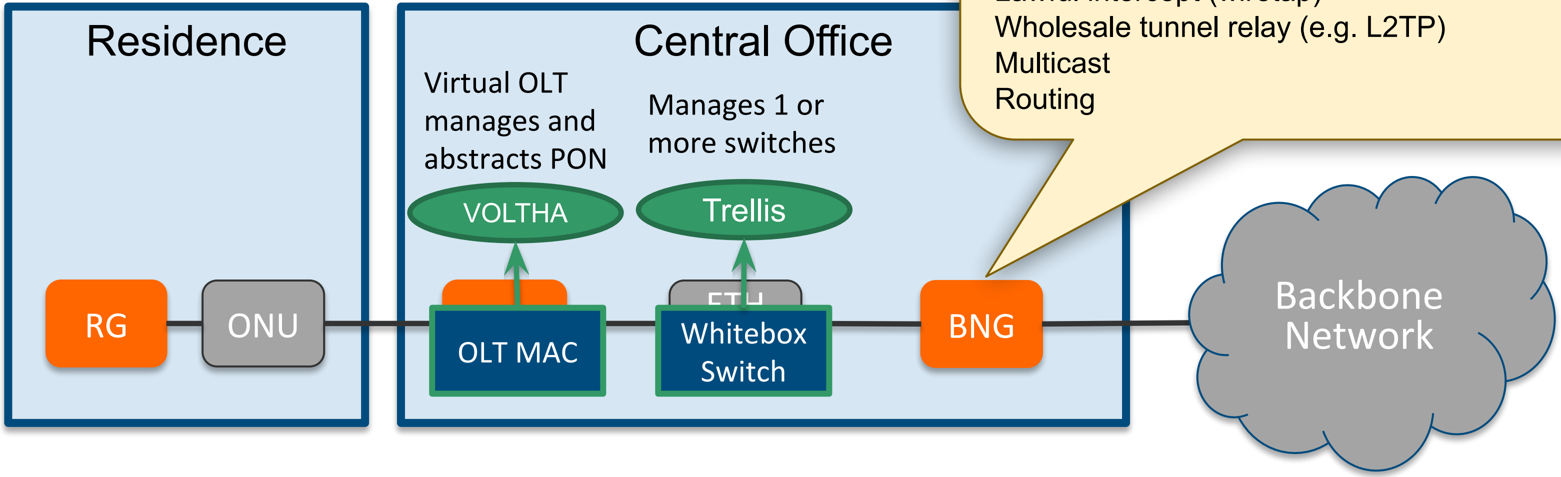
The **Adapter Proxy** acts as a shim layer to abstract the APIs while the **Inter Container Kafka Proxy** manages the request in an RPC fashion.



VOLTHA 2.x Stabilization Brigade

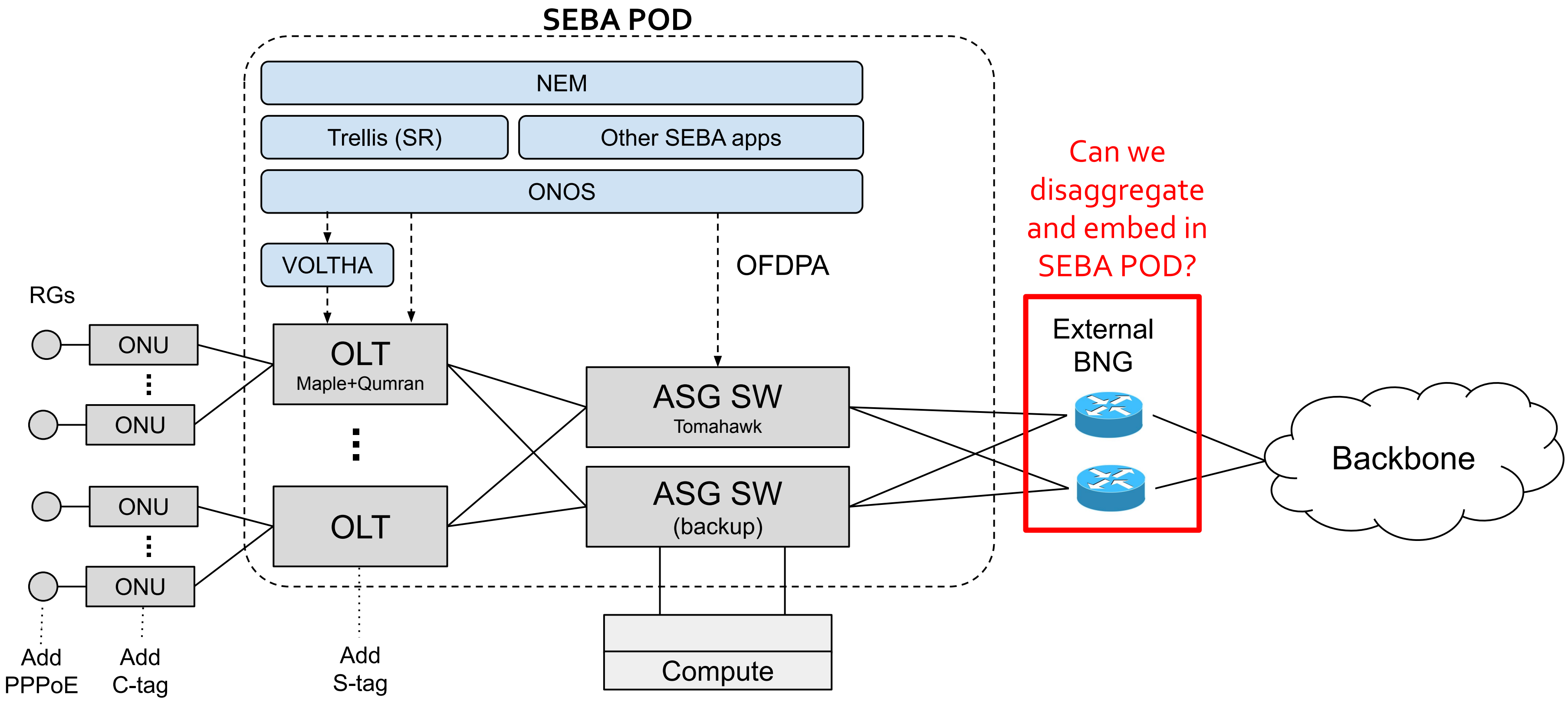


SEBA: SDN Enabled Broadband Access



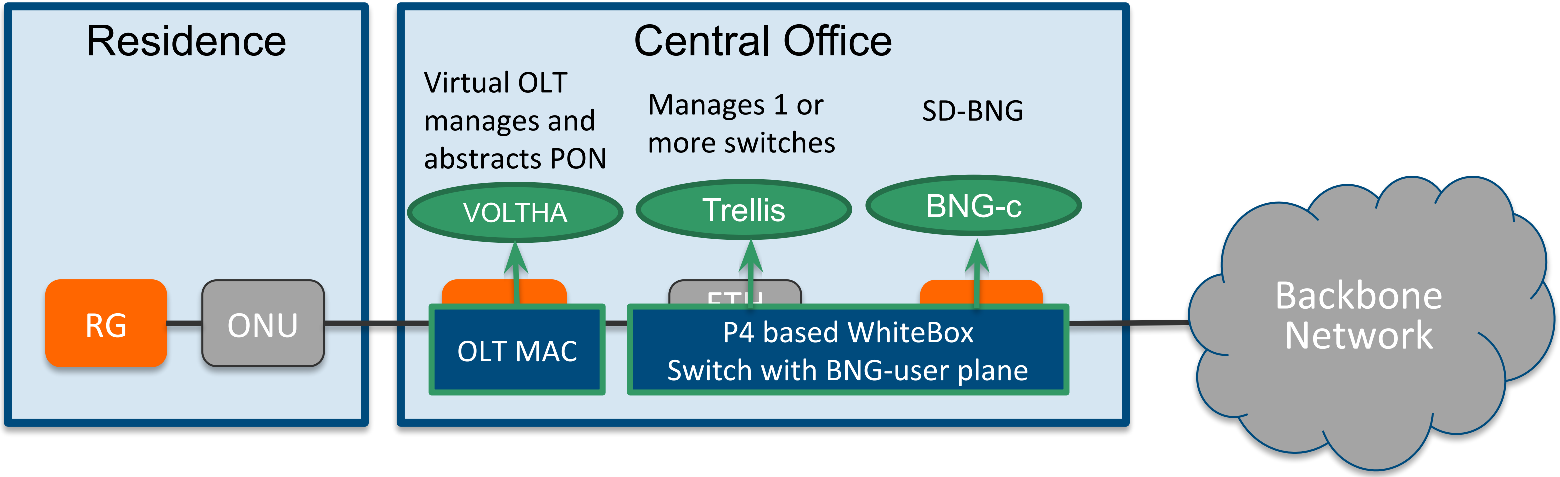
Many functions (Closed Integrated Device):
Subscriber tunnel termination (QinQ, PPPoe)
Accounting
Hierarchical QoS
Lawful intercept (wiretap)
Wholesale tunnel relay (e.g. L2TP)
Multicast
Routing

SEBA today



Can we disaggregate and embed in SEBA POD?

SEBA with SD-BNG


Residence
RG
ONU
Central Office
Virtual OLT manages and abstracts PON
Manages 1 or more switches
SD-BNG
VOLTHA
Trellis
BNG-c
ETH
OLT MAC
P4 based WhiteBox Switch with BNG-user plane
Backbone Network

Plan for BNG disaggregation in SEBA

- **BNG user plane (BNG-u)**

  Implement “in-fabric” using P4 and merchant silicon

- **BNG control plane (BNG-c)**

  Implement as an app running on top of ONOS

Initial focus on PPPoE based BNG

Acknowledgments:

- Deutsche Telekom: Initial P4 implementation of PPPoE-based BNG user plane

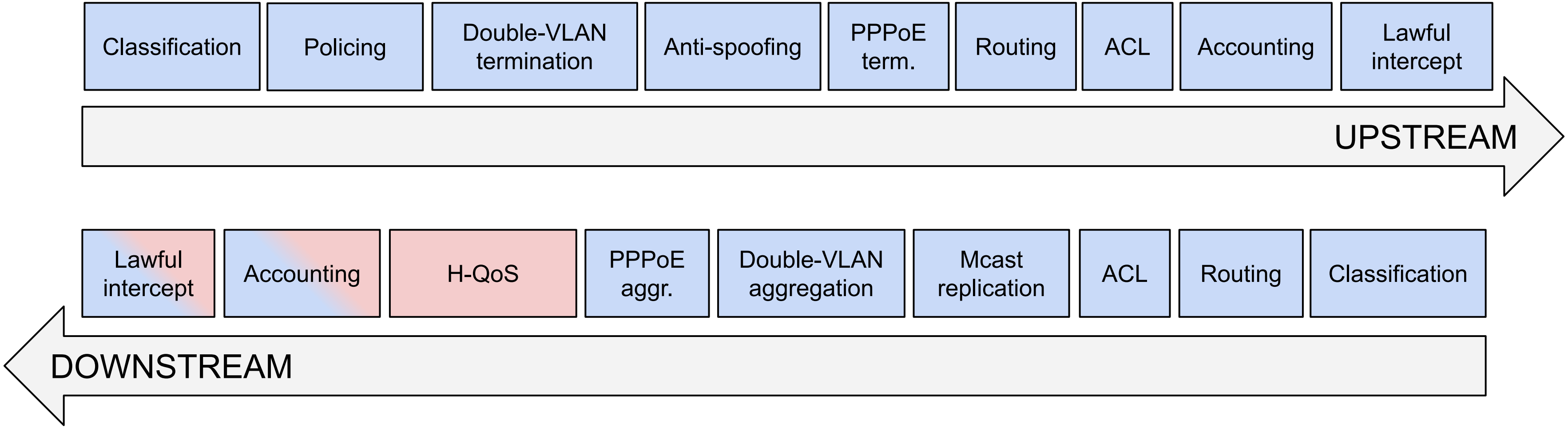
Proposed BNG-U functional distribution

🟦 **ASG SW (Barefoot Tofino)**

- DC chipset
- Implemented in P4
- Use P4Runtime to manage flow tables at runtime

🟥 **OLT (Broadcom Qumran)**

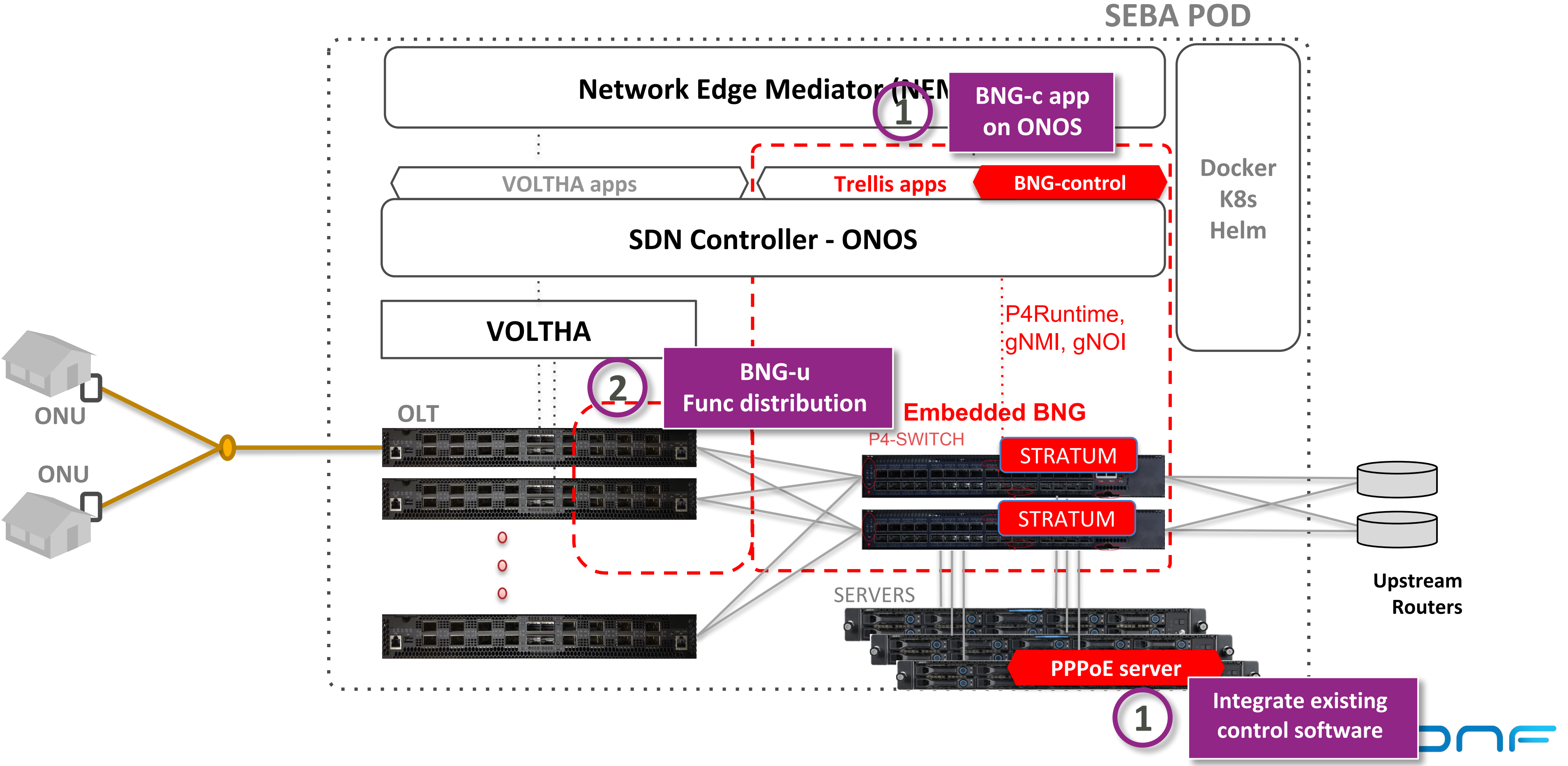
- Deep buffer chipset
- Which API to manage queues and hierarchical scheduling policies at runtime? BAL?



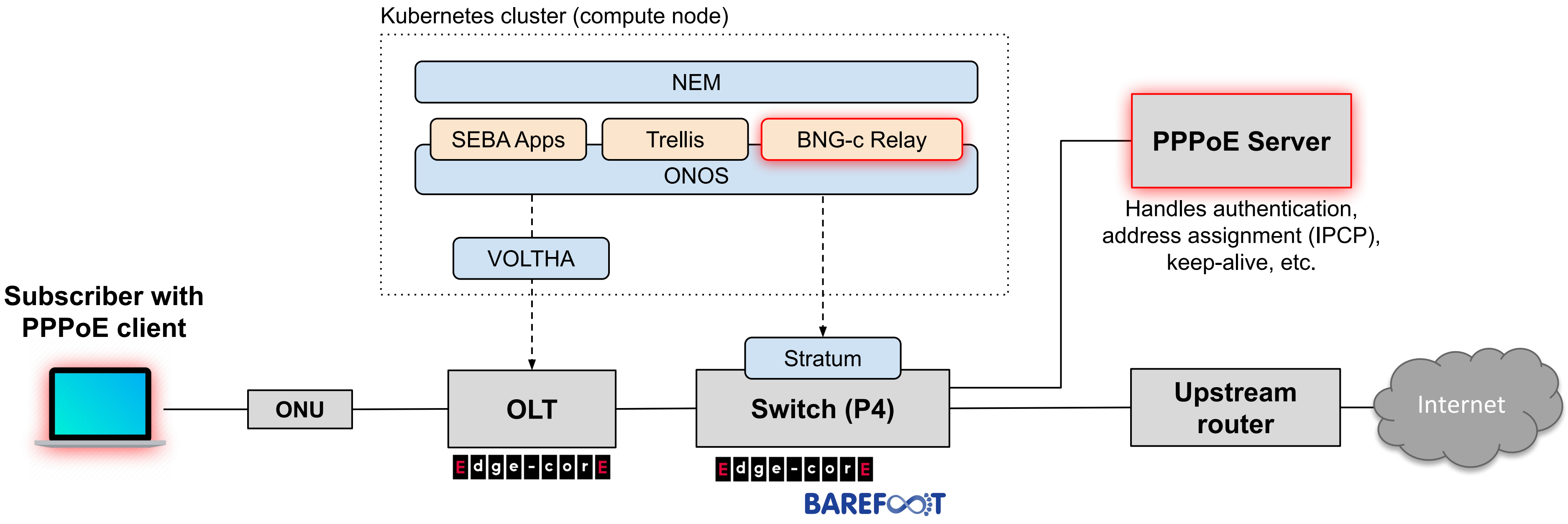
Classification
Policing
Double-VLAN termination
Anti-spoofing
PPPoE term.
Routing
ACL
Accounting
Lawful intercept
UPSTREAM

Lawful intercept
Accounting
H-QoS
PPPoE aggr.
Double-VLAN aggregation
Mcast replication
ACL
Routing
Classification
DOWNSTREAM

SEBA with SD-BNG



SD-BNG Integration with SEBA Demo at ONF Connect



Handles authentication, address assignment (IPCP), keep-alive, etc.

DEMO: Advancing SEBA with P4, Stratum & NG-SDN

Want to learn more?

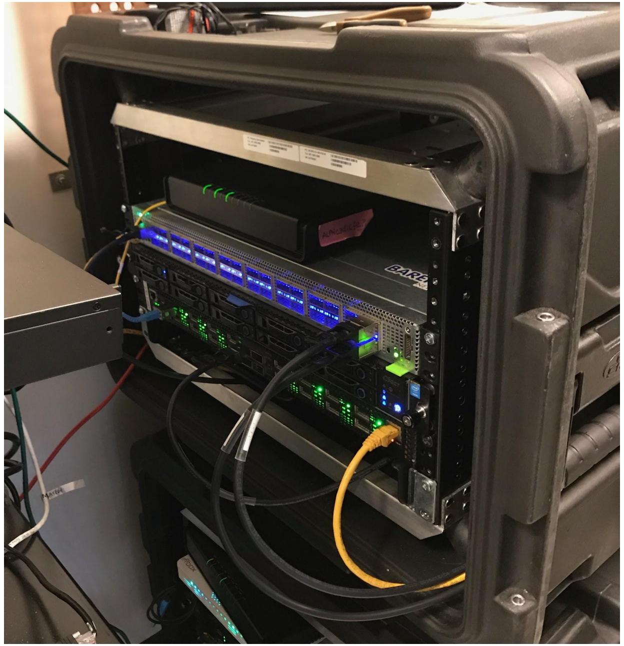
Visit the booth to see our demo

SEBA

Deutsche Telekom

BAREFOOT NETWORKS

STRATUM



What's missing / Next steps

- **Integration with NEM**
  - Push events related to subscriber management
  - FCAPS support
  - PPPoE based workflows (for TT and DT)
- **BNG-u improvements:**
  - Add Hierarchical QoS for downstream traffic
    - Plan is to use BCM Qumran chip inside OLT with BAL 3.1 API
  - Support service delivery protocols other than PPPoE
    - E.g. IPoE with DHCP-based subscriber address assignment
  - Missing P4 pipeline features -- help needed!

Thanks!

Questions?