
**REFERENCE DESIGN**

# SDN Enabled Broadband Access (SEBA)

ONF TS-103

Version 2.0 | March 2021

## About ONF Reference Designs

Reference Designs (RDs) represent a particular assembly of components that are required to build a deployable platform. They are "blueprints" developed by ONF's Operator members to address specific use cases for the emerging edge cloud.

RDs are the vehicles to describe how a collection of projects can be assembled into a platform to address specific needs of operators. By defining RDs, ONF's operator members are showing the industry the path forward to solutions they plan to procure and deploy.

Each RD is backed by specific Operator partner(s) who plan to deploy these designs into their production networks and will include participation from invited supply chain partners sharing the vision and demonstrating active investment in building open source solutions. The RD thus enables a set of committed partners to work on the specification and a related open source platform.

Assembling the set of selected components defined by the RDs into a platform enables a proof-of-concept to allow the test and trial of the design. These platforms are called Exemplar Platforms and each of them will be based on a Reference Design and will serve as reference implementations. These platforms are designed to make it easy to download, modify, trial and deploy an operational instantiation and thereby speed up adoption and deployment.

## About the Open Networking Foundation

The Open Networking Foundation (ONF) is an operator led consortium spearheading disruptive network transformation. Now the recognized leader for open source solutions for operators, the ONF first launched in 2011 as the standard bearer for Software Defined Networking (SDN). Led by its operator partners AT&T, China Unicom, Comcast, Deutsche Telekom, Google, NTT Group and Turk Telekom, the ONF is driving vast transformation across the operator space. For further information visit http://www.opennetworking.org

**Disclaimer**

THIS DOCUMENT HAS BEEN DESIGNATED BY OPEN NETWORKING FOUNDATION ("ONF") AS A **FINAL SPECIFICATION** AS SUCH TERM IS USED IN THE ONF INTELLECTUAL PROPERTY RIGHTS POLICY.

THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. WITHOUT LIMITATION, ONF DISCLAIMS ALL LIABILITY, INCLUDING LIABILITY FOR INFRINGEMENT OF ANY PROPRIETARY RIGHTS, RELATING TO USE OF INFORMATION IN THIS SPECIFICATION AND TO THE IMPLEMENTATION OF THIS SPECIFICATION, AND ONF DISCLAIMS ALL LIABILITY FOR COST OF PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES, WHETHER UNDER CONTRACT, TORT, WARRANTY OR OTHERWISE, ARISING IN ANY WAY OUT OF USE OR RELIANCE UPON THIS SPECIFICATION OR ANY INFORMATION HEREIN.

No license is granted herein, express or implied, by estoppel or otherwise, to any intellectual property rights of the Open Networking Foundation, any ONF member or any affiliate of any ONF member.

A license is hereby granted by ONF to copy and reproduce this specification for internal use only. Contact the ONF at http://www.opennetworking.org for information on specification licensing through membership agreements.

**WITHOUT LIMITING THE DISCLAIMER ABOVE, THIS SPECIFICATION OF ONF IS SUBJECT TO THE ROYALTY FREE, REASONABLE AND NONDISCRIMINATORY ("RANDZ") LICENSING COMMITMENTS OF THE MEMBERS OF ONF PURSUANT TO THE ONF INTELLECTUAL PROPERTY RIGHTS POLICY.** ONF DOES NOT WARRANT THAT ALL NECESSARY CLAIMS OF PATENT WHICH MAY BE IMPLICATED BY THE IMPLEMENTATION OF THIS SPECIFICATION ARE OWNED OR LICENSABLE BY ONF'S MEMBERS AND THEREFORE SUBJECT TO THE RANDZ COMMITMENT OF THE MEMBERS. A COPY OF THE ONF INTELLECTUAL PROPERTY RIGHTS POLICY CAN BE FOUND AT: https://www.opennetworking.org/organizational-documents/

 

# SDN Enabled Broadband Access (SEBA)

## Reference Design v2.0

Document Revision Date: March 17, 2021
Document Revision Number: v2.0
Document Release Status: Final Specification

*This reference design specification was authored by an operator-led Reference Design Team (RDT) composed of experts from:*

**Operator Group:**

**AT&T, Deutsche Telekom, Google Access, NTT, Türk Telecom / Netsia**

**Operator Group Lead Contacts:**

**Deutsche Telekom:**

Manuel Paul – manuel.paul@telekom.de
Hans-Joerg Kolbe – hans-joerg.kolbe@telekom.de
Bjoern Nagel – nagelb@telekom.de
Mario Kind – mario.kind@telekom.de
Olaf Bonness – olaf.bonness@telekom.de

**NTT:**

Kota Asaka – kota.asaka.mg@hco.ntt.co.jp
Keita Nishimoto – keita.nishimoto.wr@hco.ntt.co.jp

 

Tomoya Hatano – tomoya.hatano.yb@hco.ntt.co.jp
Jun-ichi Kani – junichi.kani.wb@hco.ntt.co.jp

**Türk Telekom / Netsia:**

Bora Eliacik – bora.eliacik@netsia.com
Burak Gorkemli – burak.gorkemli@netsia.com
Cemil Soylu – cemil.soylu@turktelekom.com.tr

**Contributors**

**Deutsche Telekom:** Hans-Joerg Kolbe, Manuel Paul, Olaf Bonness, Mario Kind

**NTT:** Kota Asaka

**Turk Telekom / Netsia:** Burak Gorkemli, Bora Eliacik, Cemil Soylu

**Ciena:** David Bainbridge

**Radisys:** Rajesh Chundury, Shaun Missett

**ONF Liaison:** Aseem Parikh

*Write to rdspec@opennetworking.org with comments or questions.*

**Document Revision History**

| Date | Revision | Description |
|---|---|---|
| 3/13/2019 | 1.0 | Final Specification for public release |
| 11/4/19 | 1.3 | Draft Specification by SEBA RD Team.<br><br>Multiple Sections - Generalize Reference Model for Access. Redefine AN as Access Network type, instead of an Access Node, and the AN type defines its device (node) types.<br><br>Section 2.3.2.1 - Add SEBA convergence with COMAC |

 

| | | |
|---|---|---|
| | | (Tom Anschutz, AT&T)<br><br>Section 2.3.2.11 Fixed Wireless Access (FWA) / mmWave |
| 11/18/19 | 1.4 | Sections 2 and 2.3.2.8 - Changed "AN" back to Access Node as in SEBA RD 1.0 and defined an AN type as the access technology.<br><br>Section 3.1.3 Security - updated based upon SEBA development community feedback<br><br>Section 3.1.8.1 Abstract OLT - removed this concept<br><br>Section 3.1.8.3 Removes reference to Abstract OLT<br><br>Section 3.1.14.1 Day 0 - added description of Akraino project that automates configuration & orchestration of SEBA |
| 10/14/20 | 1.5 | <1.5 depends upon feature updates for G.fast, BNG, and Common Data Modeling. Please add description of updates by section(s), per feature.><br><br>Section 3.1.13 Address Common Data Modeling by enhancements to SEBA NB API descriptions |
| 12/17/21 | 2.0 | Candidate of release to ONF Membership for review, ending on February 17, 2021.<br><br>Page 61: "/onu/uniport/reset" URL should be removed from "Reset ONT UNI port"<br><br>Page 62: "/technologyprofile/add" URL should be removed from "Create technology profile"<br><br>Page 62: "/technologyprofile/delete" URL should be removed from "Delete technology profile"<br><br>Page 62: "/servicedefinition/delete" URL should be removed from "Delete service definition"<br><br>Page 62: "/servicedefinition/get" URL should be removed from "Get service definition"<br><br>Page 62: "/servicedefinition/list" URL should be removed from "List All service definitions" |
| 3/17/21 | 2.0 | Final Specification for public release |

# TABLE OF CONTENTS



 

## TABLE OF FIGURES



 

## LIST OF ABBREVIATIONS

| | |
|---|---|
| 3GPP | The 3rd Generation Partnership Project |
| AAA | Authentication, Authorization and Accounting |
| AN | Access Node |
| ASG | Aggregation and Service Gateway |
| BBF | Broadband Forum |
| BMC | Board Management Controller |
| BNG | Broadband Network Gateway |
| CAP | Carrier Automation Platform |
| CI/CD | Continuous Integration / Continuous Delivery |
| CORD | Central Office Rearchitected as a Datacenter |
| CUPS | Control and User Plane Separation |
| DOCSIS | Data Over Cable Service Interface Specification |
| DPU | Distribution Point Unit |
| EPON | Ethernet Passive Optical Network |
| FCAPS | Fault-management, Configuration, Accounting, Performance, & Security |
| FWA | Fixed Wireless Access |
| NBI | NorthBound Interface |
| NEM | Network Edge Mediator |
| NFV | Network Function Virtualization |
| NG-PON2 | Next-Generation Passive Optical Network 2 |
| OLT | Optical Line Terminal |
| ONAP | Open Network Automation Platform |
| ONF | Open Networking Foundation |
| ONOS | Open Network Operating System |
| ONT | Optical Network Terminal |
| ONU | Optical Network Unit |
| OSS | Operations Subsystem |
| PNF | Physical Networking Function |
| POD | Point of Delivery |
| PON | Passive Optical Network |
| RD | Reference Design |
| SDN | Software-Defined Networking |
| SEBA | SDN-Enabled Broadband Access |
| VOLTHA | Virtual Optical Line Terminal Hardware Abstraction |
| xDSL | x Digital Subscriber Line |
| XGS-PON | 10-Gigabit-capable Symmetric Passive Optical Network |

# 1 INTRODUCTION

This Open Networking Foundation (ONF) Reference Design (RD) describes the SDN-Enabled Broadband Access (SEBA) exemplar platform. SEBA is intended to support network and feature needs of multiple operators with a common architecture. The SEBA RD provides a high-level template or architecture for supporting broadband access with a minimal prescription of technology choices.

The approach allows for multiple implementation streams to meet the SEBA requirements in whole or in parts as a set of modules and compositions that allow for a mix of SDN, NFV and also legacy PNF components to be used as compositional elements in a deployment. In addition to the SEBA RD, ONF will potentially develop exemplar implementations and implementation streams that derive from the exemplar platform.

## 1.1 PURPOSE & SCOPE

SEBA is created to provide an architecture pattern for developing solutions for carrier broadband access. The purpose is to define a common infrastructure component that would be considered non-differentiating both for operators as well as suppliers. The commonality helps create efficiency in the development of open source and white-boxes, and then commercial products and support for those entities. To drive toward this purpose, the operator group involved in developing this RD is expected to make use of the resulting implementation stream work product in some form or fashion beyond just lab or field trials.

The scope of the SEBA RD is intended to cover a broad set of wireline and fixed wireless access technologies and related Service Edge capabilities. These include, but are not limited to: PON, XGS-PON, NG-PON2, EPON, future PON technologies, Gfast, Ethernet, fixed wireless, DOCSIS and xDSL. The scope should allow easily adapting new technologies, new silicon supporting these technologies and new devices that incorporate these technologies and silicon into deployable elements. This should be possible without re-writing major sections of the subcomponents that make up SEBA and should not require new fundamental interactions northbound to carrier automation platforms. The RD supports the POD approach which means that all necessary components for service access delivery are covered. In addition to supporting plain access network nodes, the RD also may include other elements such as a leaf-spine architecture. A switching fabric enables

 

Aggregation and Service Edge functions. The RD supports requirements of operators who want to deliver broadband services without Service Edge capabilities as well as operators delivering IP services including service edge capabilities within the POD.

Direct support of wireless mobility access, like that defined by 3GPP, is not in scope of this RD, however many in the operator group have voiced the desire to have a common infrastructure layer and common components support both, so a keen eye is given to coordinating with a future wireless ONF project(s).

## 1.2 ASSUMPTIONS, DEPENDENCIES, PROCESS VARIANCES, OUT-OF-SCOPE SUMMARY

This document assumes the following relationships among Reference Design, target, and time to market solutions.

The Reference Design is described in terms of one or more solution elements. The particular implementation stream document will define how to deliver the elements and how to produce control, data and management planes. The most important of these steps is the target or end state. Target describes the most desired Reference Design in the context of the preferred future environment. The RD may also define a series of well-known intermediate elements that might be needed to go to market sooner or with near term constraints that prevent moving directly to the target. These are defined as Time-to-Market solutions.

Given these relationships, the target Reference Design describes both a set of assumptions and dependencies in addition to the body of the design itself. Because the design may change over time, it is also expected that the assumptions and dependencies may also change with the Time-to-Market solutions.

The assumptions for the SEBA Reference Design are:

1. Operators are interested in initial commercial deployments in 2020;
2. The existing carrier automation platforms include both legacy OSS as well as new orchestration systems, e.g. ONAP;
3. There are no strictly-greenfield deployments envisioned. This means that SEBA will need to be able to work within larger, already existent networks, services, and operational models;
4. There are requirements for multiple options to support Broadband Network Gateway (BNG) functions and these functional aggregation in

 

an exterior legacy device (PNF) and disaggregated placement of functions within the SEBA POD – either as Software of SDN VNFs;
5. The RD will reduce operational complexity by hiding it in layered abstractions, as is typical of IT systems. This also means that the design will incorporate self-sufficiency and automation for its assembly, failure recovery, and performance;
6. SEBA will be deployed in infrastructure aggregates, often called PODs. A point of delivery, or POD, is a "module of network, compute, storage, and application components that work together to deliver services. The POD is a repeatable design pattern, and its components maximize the modularity, scalability, and manageability of data centers" (reference: Cisco Nexus 2000 Series Fabric Extenders Data sheet);
7. SEBA will be constructed using containers run in Kubernetes as the cloud underlayer. It may use existing projects like Akraino or Airship to provide these functions and is expected to be loosely coupled to such layers.

It is recommended that implementations of the SEBA Reference Design are built using:

1. Working Kubernetes Environment
2. CI/CD Tools (e.g. Jenkins, etc.) for development as well as deployment instances.

SEBA is related to several mature projects at ONF, including ONOS, VOLTHA, XOS, and R-CORD. Because this work interacts with existing released work in active communities at ONF, it is likely that some of the processes defined for normal new RD work may need to be adjusted to ensure that the existing communities and their work do not become disenfranchised.

## 1.3 SEBA AND EXISTING MANAGEMENT AND ORCHESTRATION PLATFORM

Most carriers will need to develop adapters or agents that allow interworking between SEBA and existing management and orchestration platforms. To the extent that this work affects requirements and common aspects within SEBA, such work will be adopted in a common Network Edge Mediation module (NEM described in more detail below). However, aspects that are unique to a single carrier, product or deployment are considered out of scope for this RD.

 

## 1.4 AUDIENCE

The ONF partners are the current audience of this pre-Alpha version of the document, per the ONF Reference Design Process, and at this stage this document should not be shared outside of the ONF Partners defined at the top of the ONF membership page.

Upon reaching the criteria for an Alpha stage RD, the ONF TLT at its discretion will send drafts to the full ONF membership list.

Following the ONF RD process for the timeframe for members to review and comment, and following review of comments by the TLT, the Orchestration TLT will provide decisions about revision of the document and when to release the RD as a Final Specification.

## 1.5 DOCUMENT RELATIONSHIP

The SEBA RD is a standalone document in the SEBA process.

The ONF site provides a SEBA wiki that provides references to the designs, code, workflows, JIRA board, meeting times, meeting recordings, developer meeting list and Slack channel. Solution development represented in the artifacts at the SEBA wiki should be considered as a more detailed snapshot of implementation(s) for SEBA.

## 1.6 SOFTWARE RELEASES

The SEBA project should define software releases as a solution set for the software components, including but not necessarily limited to Network Edge Mediator/Edge Cloud Orchestrator, SDN controller, Access Node driver, and Aggregation and Service Gateway driver.

The SEBA software release documentation should provide the solution set information for these software releases.

The SEBA software release documentation should also provide lifecycle management of the compatible releases between these components, in order to define flexibility and dependencies for coordinated upgrades of the components.

The hardware from vendors may also include embedded software for controlling, monitoring and abstracting low level functions of the hardware, including BIOS, firmware, board support drivers and board management

 

controllers (BMCs). The vendors shall identify the required versions of these embedded software components, and how to upgrade these embedded software components using open software lifecycle management procedures.

### 1.7 HARDWARE RELEASES

The carriers define the hardware solution, including vendors, models, and releases. ONF suppliers do provide value to identify hardware for an ONF Reference Design, and to update the carriers with roadmaps and new product information for enhancements and improved cost.

# 2 REFERENCE DESIGN TARGET

SEBA is designed as a set of container elements that run in a Kubernetes environment. The system is modularized per typical microservice system architecture, and there is a hierarchy of modularity used to allow flexible compositions at different scales.

As is shown in Figure 1, SEBA is comprised of a few high-level software modules, including:

- Network Edge Mediator (NEM)
- SDN Control
- Control Applications
- Access Node (AN) Driver
- Aggregation and Service Gateway (ASG) Driver
- Device Manager

Figure 1 also shows some of the typical hardware equipment that comprises SEBA, including:

- Access Node types: PON, DPU, Fixed Wireless Access (FWA), other
- Access Node device types: Device types defined by the AN type (e.g. PON, DPU, FWA). For example, a PON AN type includes OLT and ONU device types.
- Aggregation and Service Gateway: one or several switches/routers in a setup that supports options for layer 2 aggregation, layer 3 service aggregation, Service Edge/BNG (and/or S/P-GW) functions and

 

supports composing an SDN-controlled leaf-spine fabric. The fabric design can be VxLAN based as well as MPLS based. It provides a mesh for the localized ANs to access the network, and optionally a management network between the compute functions, the ANs, AN drivers and ASG drivers.

- Compute: servers that host the AN driver(s), and ASG driver, as well as control and management plane modules

Finally, Figure 1 below also shows various interfaces among the physical and software entities and also the Technology Profiles, labeled TP, that provide abstraction hints for controlling technologies that are not similar to Ethernet.

The RD expects downstream implementation stream documents to develop and maintain specific implementation details, both from choices of components and also from instances or releases of those components.

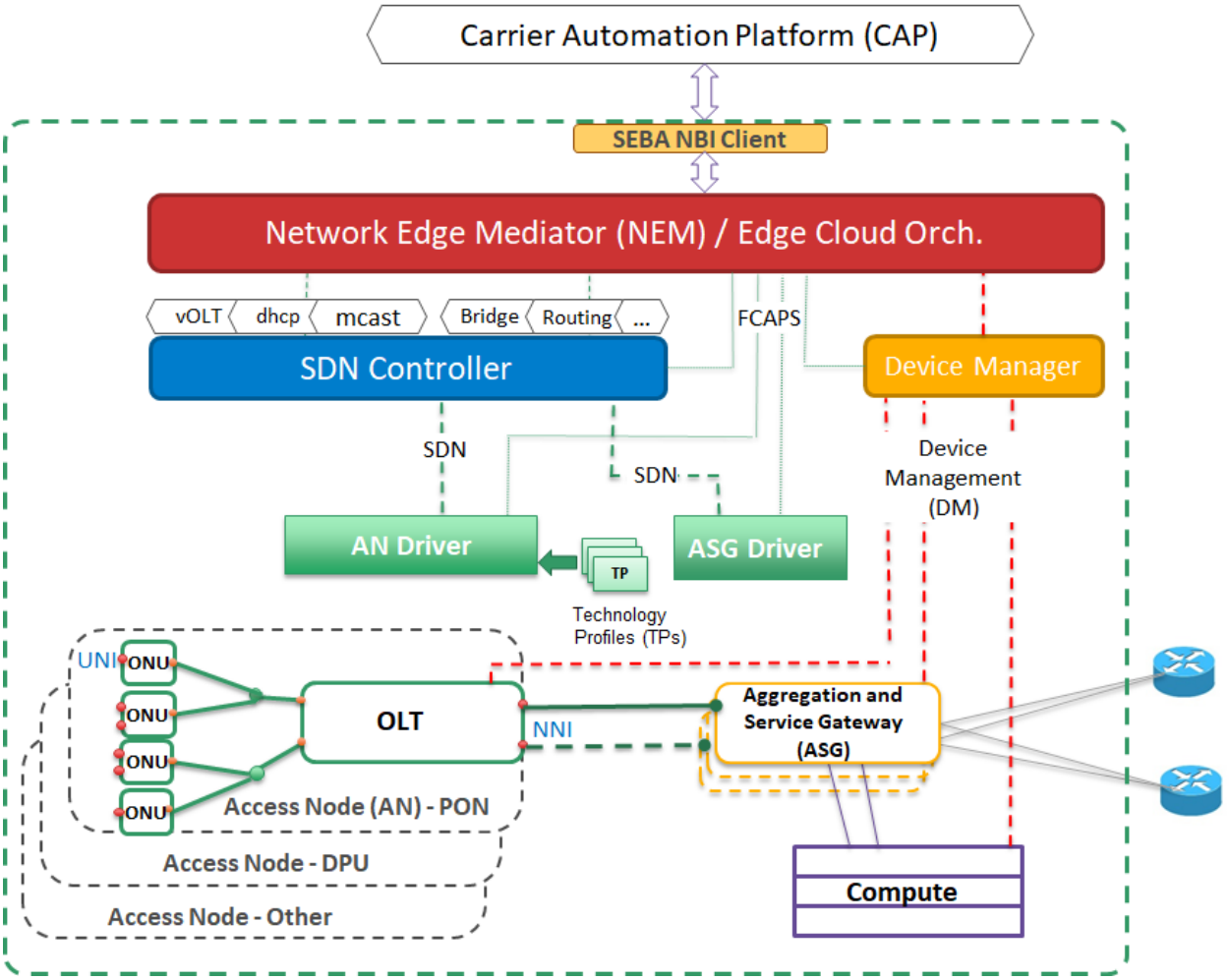


Figure 1: High Level Target Architecture

The target architecture diagram requires some principles and definitions to be noted:

- The Infrastructure Layer is not denoted but it includes the physical components - the Access Nodes (ANs), the Aggregation switches, and Compute.
- The Service Layer defines the binding of the components in the Infrastructure layer to deliver a service.
- The SDN controller maintains autonomy of the control structure to each component of the Infrastructure Layer involved in a service.
- ASG is only a functional block that supports aggregation, switching and routing of data plane, control plane and management plane traffic within a POD, and supports Service Edge capabilities. Use of multiple ASG devices, as depicted in figure 1, is a deployment option an operator may select, but the ASG setup may also be non-redundant.
- Device Management (DM) is a functional definition for the interface to equipment management functions.

## 2.1 SALIENT CHARACTERISTICS OF THE END STATE

Over the past several years, the SDN and NFV ecosystem has moved beyond skepticism and doubts into actionable strategies being adopted and deployed by operators around the world. For L4+ services (e.g., IMS, DNS, etc) and L1-3 (e.g., Optical and IP/MPLS, etc.) services, approaches, while not all optimal, are all fairly well understood and achieve many of the key benefits of decoupling, common off the shelf systems, and disaggregation of SW. While the industry is not at an ideal state, it is well on its way and headed in a common direction, thanks in a large part to organizations like ONF who were the early trailblazers. Moving forward, operators' access networks and service delivery platforms stand to benefit from such a focus. Access plant remains highly proprietary and capital intensive, representing a large component of capital outlay of most large operators. Access networks also come with unique constraints, such as environmental, regulatory, space, and power that tend to favor small edge compute platforms and highly distributed open-spec peripherals.

In a sense, the "easy work" has been done and the largest operational and capital outlays for operators remain fertile ground for transformation. The SEBA effort will drive access networks across the globe to deliver on the promise of open, software-driven systems in new key areas such as multi-gigabit fiber access networks and will ultimately look to reuse the concepts presented here in emergent wireless access.

Each of the key characteristics below represent key industry drivers for

 

highly performant, secure, flexible common solutions that aim to represent a lowest TCO infrastructure for operators across the globe.

1. Automated (Zero-Touch), secure, reliable
2. Affordable and transparent – Startup cost & operations
3. Highly Modular HW and SW, including peripherals and acceleration
4. Architecture of HW and SW that enables small start and scalability
5. Open Systems First, Disaggregation by Design, Modularity by Design, Loose Coupling
6. May be sourced using a variety of business and packaging models – Operational Abstractions is a Priority
7. Performant. Real-time, low latency
8. Integrally considers and intersects with automation and management systems and approaches
9. Multi-access by design
10. Multi-Cloud/Hybrid Cloud support
11. Is not defined by physical location (e.g., CO)

## 2.2 TARGET REALIZATION APPROACHES

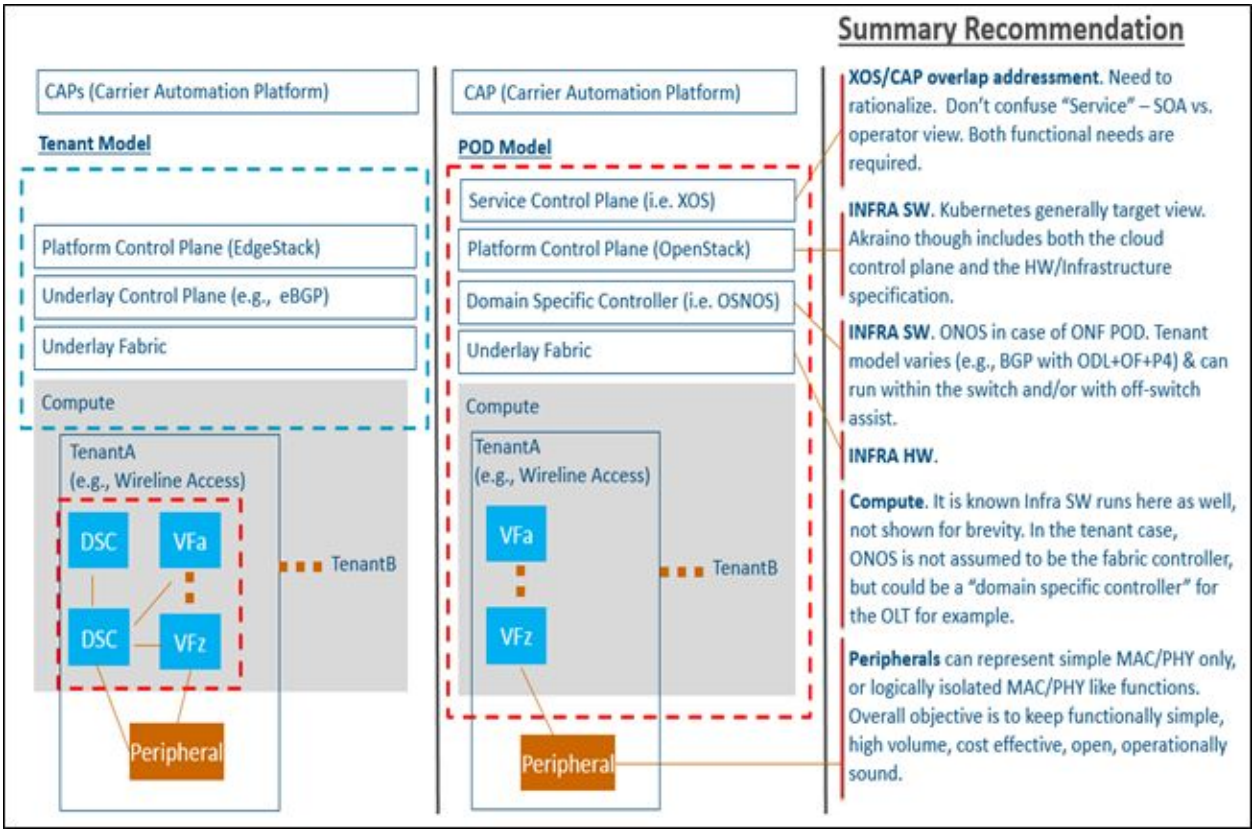


Figure 2: Target Realization Approaches

SEBA may be realized in a variety of different ways, depending on the operator and/or situation. Most important in this regard is the clear decoupling and separation of "service" from "infrastructure". This also provides for the realization of SEBA on a variety of different infrastructure platforms, including, but not limited to those that are CORD based. Modularity through good functional decomposition should provide for extensive reuse across the ONF solutions providing for high volume, consistent, SW assemblies possible from the ecosystem.

### 2.2.1 POD Approach(es)

Generally, this would be inclusive of infrastructure and service layer and be operated as SDN based access platforms. Controller can be operated inside a POD or as a "cloud in a box" (e.g. based on cloud control plane functions embodied within the POD).

### 2.2.2 Tenant Approach

This would be the case where an operator already has a cloud environment specified, and SEBA is either a tenant set of workloads and peripherals or is itself hosting a tenancy environment, with the peripherals managed and operated primarily at the service layer.

Both POD approach and Tenant approach do not limit how the broadband access services are deployed. SEBA supports the Access network infrastructure provider to deliver the broadband access services, or the infrastructure provider to deliver the network to third party broadband access service providers under a wholesale agreement.

## 2.3 MAJOR FUNCTIONAL COMPONENTS

### 2.3.1 SEBA Infrastructure

#### *2.3.1.1 Hardware*

Hardware includes physical components - the ANs (including OLTs, ONUs), switches, compute servers, physical interface plugins, fibers, cabling and powering.

#### *2.3.1.2 Software*

The SEBA project delivers a set of software components specified in the high-level architecture, including but not necessarily limited to NEM/Edge Cloud Orchestrator, SDN controller, Access Node (AN) driver and Aggregation & Service Edge (ASG) driver.

Note that Service Providers specify the SEBA NBI client software for the interface to their Carrier Automation Platform (CAP).

The hardware from vendors may also include embedded software for controlling, monitoring and abstracting low level functions of the hardware, including BIOS, firmware, board support drivers and board management controllers (BMCs).

 

The infrastructure services for instantiating a POD and remotely maintaining the lifecycle of the POD requires open and automated management approaches.

## 2.3.2 SEBA Service Layer

### *2.3.2.1 Disaggregated BNG with Service Edge CUPS*

SEBA supports multiple deployment options of BNG functionality:

1. Standalone, external BNG - SEBA acts as a smart aggregation network
2. BNG included into SEBA POD - either as PNF or VNF
3. BNG embedded into SEBA POD – functionally decomposed into Service Edge (BNG-SE) and Router and deployed as Physical Network Function (PNF) or Virtual Network Function (VNF) (with the PNF possibly being embedded on AN or ASG hardware)

The SEBA RD 1.0 described the general functionality of a Broadband Network Gateway (BNG). In addition, it has introduced the functional disaggregation of the BNG into a service edge (BNG-SE) and routing part, which can be deployed as a PNF or VNF with the PNF possibly being embedded in the AN or ASG platforms that are optimized for packet forwarding at high bandwidth and make use of programmable silicon.

While all deployment options of SEBA RD1.0 continue to be supported, this release of the RD focuses on enhancements for option 3 above.

This approach is further detailed and put into context with the Control and User Plane Separation (CUPS) following 3GPP and BBF design philosophies.

**BNG Functional Architecture**

The BNG (Broadband Network Gateway) is a key component in fixed broadband access networks. It resides at the demarcation point between the access network (usually based on L2 tunnels) and the routed IP/MPLS network. The BNG provides per-subscriber services and is the highest-tier network element that has the full per-subscriber context. Beyond this point, traffic can be correlated to a subscriber using the IP address, but the complete view is gone. The functional requirements on BNG types are described by the Broadband Forum (BBF, TR-178 and related documents).

 

The core functions of a BNG at the access side and towards the core, but not limited to, are:

- Aggregation of L2 access tunnels / VLANs / MPLS PWs
- Termination of network attachment (PPPoE or IPoE tunnels), authentication, (dynamic) policy enforcement, AAA client
- Tunnel switching and termination (e.g., L2TP LAC)
- Traffic filtering and shaping
- Lawful interception
- Anti-Spoofing
- Split horizon rules
- Per-subscriber OAM (e.g. using keepalives)
- Accounting

The BNG also acts as a router as it is part of the IP core network of the service provider. This covers, amongst others, the following functions:

- Routing protocols (IGPs, EGPs)
- MPLS control and user planes

The BNG directly interfaces with Policy control systems and AAA services.

### SEBA BNG Functional Decomposition

The BNG is decomposed into the following control plane and user plane building blocks as shown in figure below.

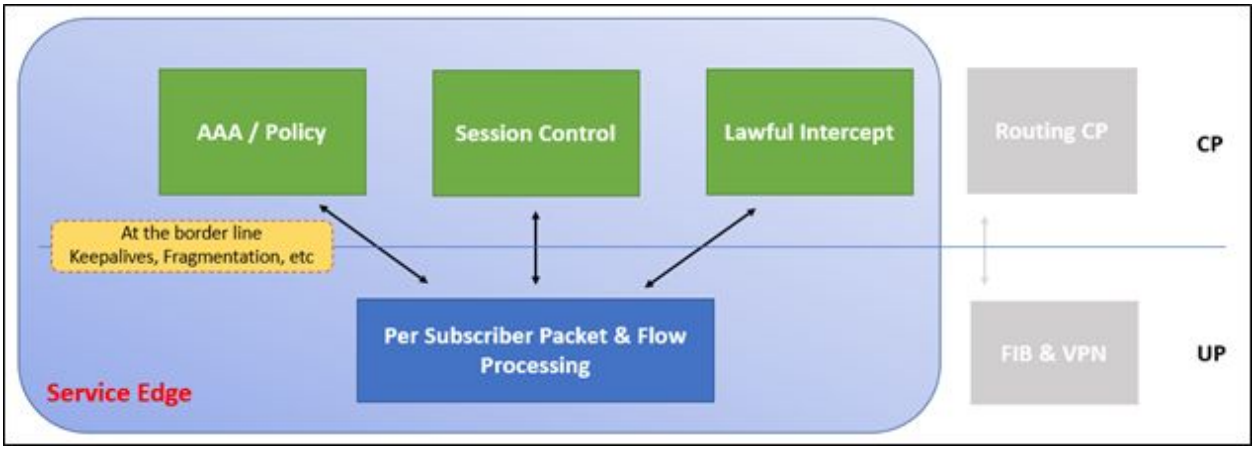


Figure 3: BNG Building Blocks

The Routing Control Plane (CP) as well as User Plane (UP) are not dependent on the per-subscriber functions and can thus become separated. Doing so leads to a look at a BNG as being split into a subscriber-facing function which we call Service Edge (BNG-SE), and a router function.

Further steps of disaggregation like CUPS are possible and discussed in the following sections.

### Decomposed BNG Deployment Options

When going for a decomposed BNG, a natural choice is to embed the routing function (CP and UP/FIB) into the switching fabric which is constituted of Aggregation and Service Gateway (ASG) devices. The CP may run on a centralized SDN controller such as Trellis, it may also run as a distributed routing process on the ASG.

The BNG-SE may run:

- Embedded in ASG devices by making use of programmable silicon
- Embedded in AN devices by making use of programmable silicon
- Embedded but spread across ASG and AN devices by making use of programmable silicon
- On a “black box” mounted into a rack in the POD as PNF on dedicated hardware or as VNF on generic servers

The decision where to place functions strongly depends on the actual requirements of the service providers. SEBA shall cater for all the above, leaving sufficient flexibility for placement.

An important aspect is the case where components of the POD, e.g. the ANs are located in geographically further distributed locations.

All cases, including the one with a non-decomposed BNG directly attached to a POD have the need for a traffic steering mechanism in common. The SDN control function needs to steer the L2 tunnels of the customers to the service edge. Once such a steering mechanism is in place, the service provider can even steer customers to different BNG-SEs. Slices can be built. One can e.g. imagine dedicated BNG-SEs for enterprise or IoT customers. Those can even be implemented using different technologies (e.g. virtualized on x86 for IoT, on programmable switching silicon for enterprise customers).

### Definition of CUPS for a Disaggregated BNG

A consequent next step is the mentioned decomposition of the functional part of the BNG with the help of the Control and User Plane Separation

 

(CUPS) concept. While essential design aspects from BBF TR-459 (BNG CUPS specification) are taken into account, this SEBA RD's complimentary focus is on architecture, requirements and SDN APIs for disaggregated BNG implementations leveraging open and programmable hardware.

The separation between control and user plane needs to be explained in order to avoid confusion with more common control and data plane separation. CUPS is introduced to support better scaling and redundancy of each plane. Many control plane functions do not need to reside on the device itself and could be outsourced to a central control plane block. The control plane could run on different more optimal hardware than for example rather fixed ASICs. In addition, protection mechanisms can be implemented easier with today's cloud like concepts, e.g. scale-out in container clusters, or the control plane can be moved to more centralized places.

But some control plane functions might have to reside close to the data plane in the local control plane. Especially OAM functions for detection of session loss or the collection of statistical data must be implemented in the user plane and cannot be offloaded to the central control plane. In the following, CUPS is used as a general design principle, while detailed explanation of the distribution of functions and their respective implementation is outside the scope of this document.

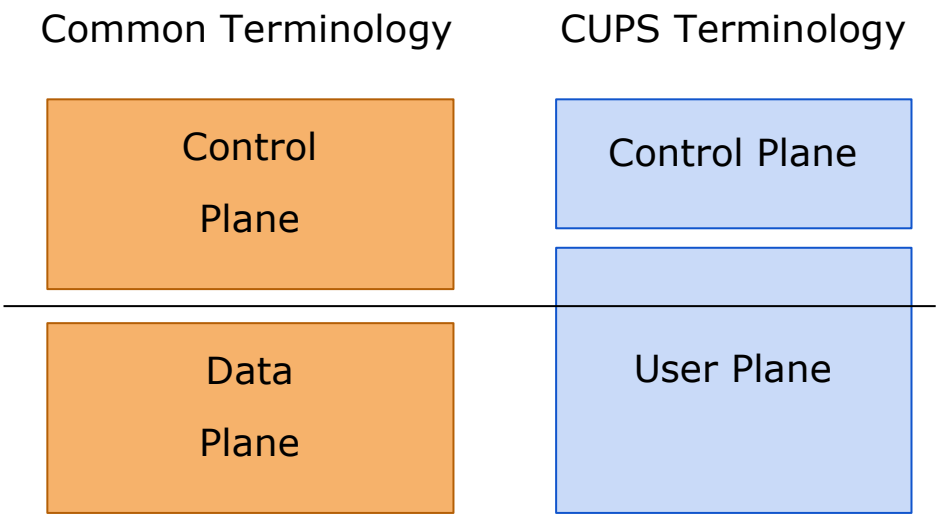


Figure 4: Common vs. CUPS Terminology

The routing part of the BNG can be fulfilled by a routing function on a "classical" SDN Controller. Still, the BNG-SE part has to be covered. In the following it is concentrated on the BNG-SE only. The control plane will be referenced as BNG-$SE_C$ and the user plane as BNG-$SE_U$.

### Requirements on a Disaggregated CUPS BNG

In order to realize separate scaling of CP and UP, it shall be possible to locate BNG-$SE_C$ and BNG-$SE_U$ on different physical devices as VNF or PNF. Still for smaller deployments or by service provider choice, a combination of both in a single box should be possible, especially when there is no use case for centralizing subscriber state of multiple user plane instances.

The BNG-$SE_C$ must be able to control various BNG-$SE_U$. Depending on the business needs like different service classes or wholesale, it is desirable to have multiple BNG-$SE_C$ controlling a single BNG-$SE_U$ function. In consequence, a N:M relationship between BNG-$SE_C$ and BNG-$SE_U$ should be possible. In case multiple BNG-$SE_C$ are accessing a BNG-$SE_U$, quality and security assurance with the help of policy control needs to be implemented.

For the support of service specific BNG-$SE_{C/U}$ combinations (e.g. wholesale, special business customers, IoT), a POD needs a common traffic steering mechanism for the SE allowing to direct traffic to different BNG-$SE_U$ types.

From a hardware perspective, the BNG-$SE_C$ shall be independent from the underlying BNG-$SE_U$ and work with different silicon classes like ASIC, FPGA or X86 and support device designs like single pizza box designs or multiple line card systems. In practice, this is a far-reaching goal and must be carefully evaluated in further implementation decisions.

The support for different services running on a single BNG-$SE_U$ for residential and business customers varying in e.g. QoS profiles is another important requirement.

From an architectural perspective, the disaggregation of the SE part needs three main interfaces between BNG-$SE_C$ and BNG-$SE_U$:

- Management interface for general aspects of the user plane like monitoring, telemetry or profiles for QoS handling
- State control interface for programming the forwarding in the data plane by the management or control plane, e.g. after accounting events
- Control packet redirection interface for sending control plane information like PPP LCP keepalives between data plane and control plane

Currently, there are two relevant realizations developed. First, is the telco favored PFCP based concept as e.g. detailed in BBF TR-459. The second one is the more IT data center-oriented model with P4/P4Runtime and gNMI/OpenConfig. Both tackle different levels of abstractions with the latter focusing on the chipset level. In any case, common, open sourced and industry accepted implementation and data models must be used.

 

### Deployment Options

There are different options to implement BNG CUPS in the SEBA reference platform and it should be left to the operator to decide which one to use. The three options are:

- Native BNG-$SE_C$ integrating the BNG-$SE_C$ functions on top of a SDN Controller
- Standalone BNG-$SE_C$ function which informs the SDN Controller via an API to program the correct forwarding state into the forwarding plane
- Combined BNG-$SE_C$ and BNG-$SE_U$ with a simplified SDN control layer

Still, there are different options to host the BNG-$SE_U$ plane (see SEBA RD Version 1.0 Section BNG) like ASG or AN. For simplification reasons, only the deployment in combination with a SDN Controller and an ASG as BNG-$SE_U$ is presented hereafter.

For simplification reasons, hereafter is assumed that no control plane function is required in the user plane. Other cases need to be discussed in upcoming versions of the SEBA Reference Design or separate BNG CUPS API definitions.

### Native BNG-SE$_C$

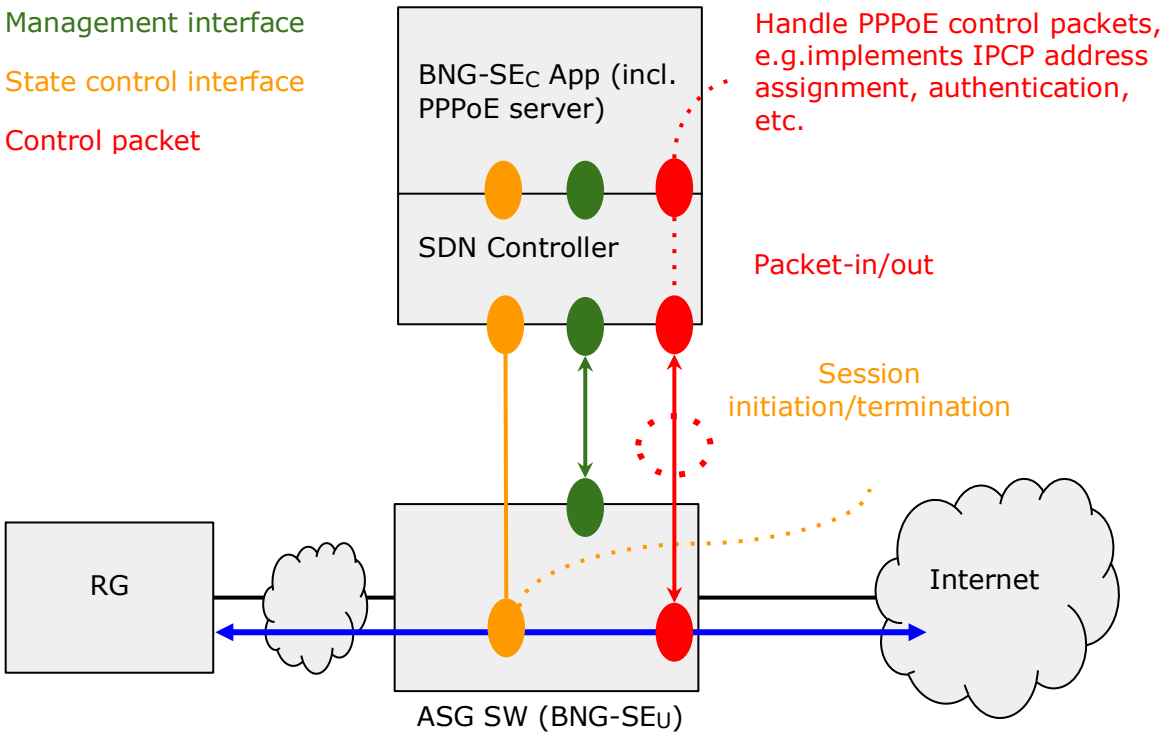


Figure 5: Native BNG-SE$_C$ Architecture

This is the case where the BNG-SE$_C$ is integrated directly in the SDN Controller. The SDN Controller is the only element controlling the BNG-SE$_U$. The BNG-SE$_C$ App signals via the Management Interface to the SDN Controller the desired configurations of the BNG-SE$_U$ (including the state control and control packet redirection interface) and the SDN Controller implements accordingly. PPP control packets are intercepted from the ASG SW and delivered to the app by means of packet-ins via the POD control and management network. The app implements logic such as IPCP address assignment and authentication, generating replies to such control packets and manages the BNG-SE$_U$ forwarding state via SDN Controller requests. Replies to control plackets are delivered to the RG by means of packet-out. The native option provides the best integration with an SDN controller as it allows access to the entire network state including multiple BNG-SE$_U$ distributed over the SDN fabric, however, it requires to implement from scratch all PPPoE server functions. In addition, it requires a detailed specification of the interface between SDN Controller and BNG-SE$_C$ App to allow interoperability and exchange of the BNG-SE$_C$ supplier.

### Standalone BNG-SE$_C$

The previous option faces two problems: implementation of an application on top of SDN Controller and the use of the SDN Controller infrastructure to transport PPPoE control packets. A potential solution is a standalone BNG-SE$_C$ where control packets via Control Packet Redirection Interface are forwarded entirely in the data plane network.

To manage the BNG-SE$_U$ management and forwarding state (Management and State Control Interface respectively), two extremes and cleanest solutions are possible. Either the BNG-SE$_C$ always communicates via the SDN Controller to the BNG-SE$_U$ (left part of the figure below) or the BNG-SE$_C$ communicates directly with the BNG-SE$_U$ and a master SDN Controller manages only the access to resources like tables in the BNG-SE$_U$ (depicted in the right part of the figure below). In between these two models exist a number of variants, where one interface uses the SDN controller and the other communicates directly from the BNG-SE$_C$ to the BNG-SE$_U$. Currently, this split of responsibilities does not look like a recommended way forward.

The first model (left side of the figure) assumes that exclusive control over the management and forwarding state of the BNG-SE$_U$ is enforced by the SDN Controller. All requests to modifications from the BNG-SE$_C$ are sent to the SDN Controller, who implements them in the forwarding tables of the BNG-SE$_U$. This for example could hide all details about the distribution of the BNG-SE$_U$ from the BNG-SE$_C$ and might hide physical specifics as well. Nonetheless, it would complicate the features and functions on the interfaces and might require the limiting of available physical details to the least common set. The second option on the right, assumes a policy function in the user plane. This function “slices” the resources like the forwarding tables and gives control to applications like the BNG-SE$_C$ to freely program the forwarding state in this slice. The SDN Controller will allow, monitor and remove the access to these resources.

In any case, both options of this solution end in a modification of existing servers to support these new API. The BNG-SE$_C$ application must be aware of the physical capabilities of the devices as well - or it must be limited to less optimal support of hardware features.

A potential option for the interfaces between BNG-SE$_C$ and SDN Controller or BNG-SE$_C$ and the BNG-SE$_U$ are northbound APIs of SDN Controllers (e.g. using REST or gRPC).

The major advantage of this solution is the flexibility to attach other control plane applications to the user plane without the need to integrate them into

the SDN controller and limited adaptation to essential interfaces for management and state control.

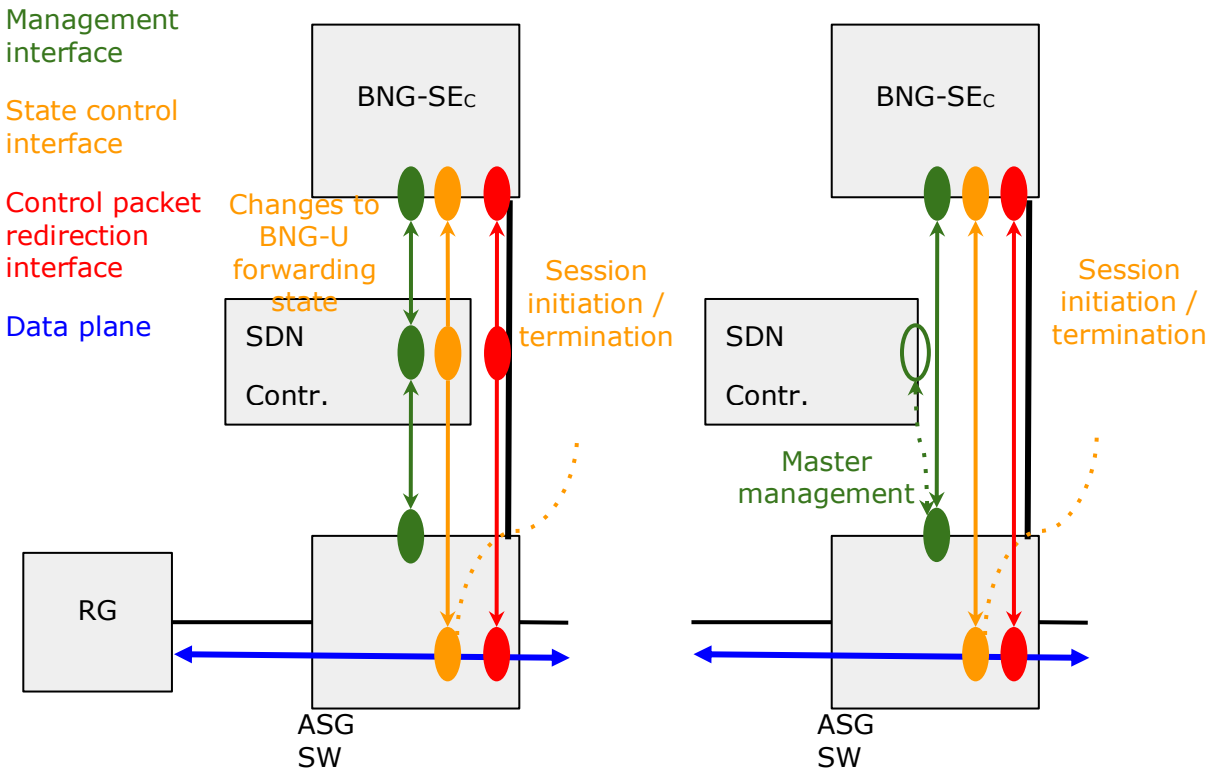


Figure 6: Standalone BNG-SE$_C$ Architecture

**Combined BNG-SE$_C$ and BNG-SE$_U$ with a Simplified SDN Control Layer**

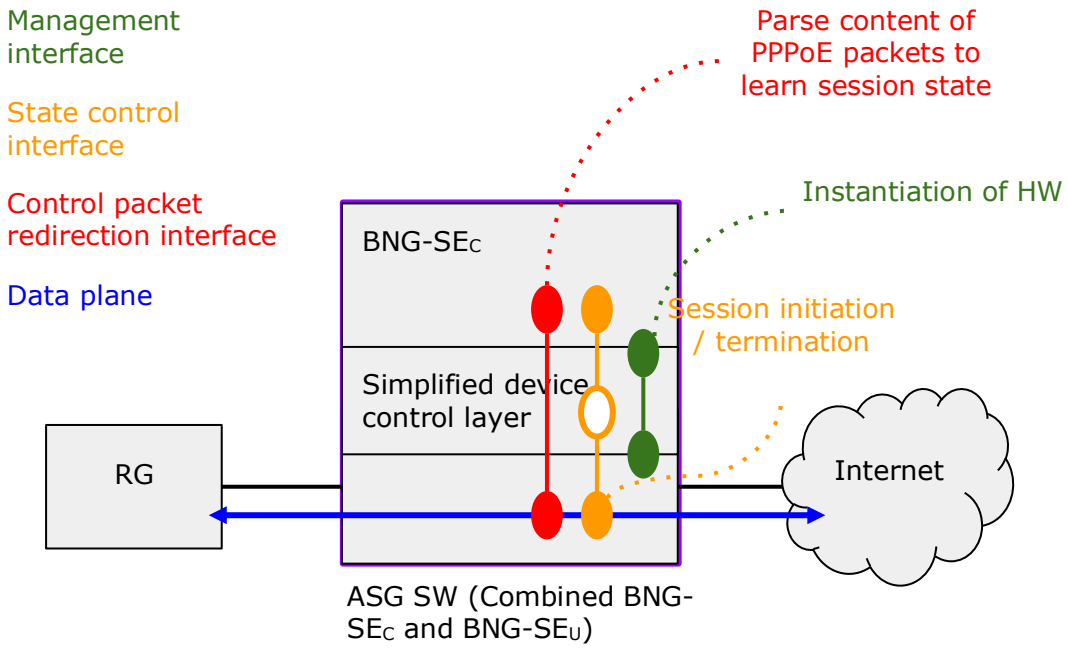


Figure 7: Combined BNG-SE$_C$ and BNG-SE$_C$ Architecture

This case is an option for simplified deployment scenarios, where no pod-fabric is needed or the BNG is run as a standalone function inside the pod, but it is desired to reuse the BNG control plane. The AGS switch hosts the BNG-SE$_C$ and a simplified device control layer. This device control layer is reduced to the bare minimum, e.g. protocol translation of interfaces between BNG-SE$_C$ and BNG-SE$_U$. All other functionality is similar to the Standalone BNG-SE$_C$ option with the only exception that no policy enforcement is done.

From an implementation perspective, this could be realized in processes or containers.

#### *2.3.2.2 NEM*

The Network Edge Mediator (NEM) serves as the mediation layer between the edge/access system and the service provider backend and global automation frameworks. NEM will provide the interfaces and components to support FCAPS functionalities required by the service provider for managing the access network components and broadband service subscribers the SEBA

 

POD is designed to offer and support. A variety of operator OSS/BSS and global orchestration frameworks can be integrated northbound for specific deployment needs.

#### *2.3.2.3 SEBA NBI Client*

The SEBA NorthBound Interface (NBI) Client provides an application layer for management interfaces between the Carrier Automation Platform and NEM.

The SEBA NBI client is tightly coupled with the Carrier Automation Platform and so is specific to the Service Provider.

#### *2.3.2.4 SDN Control*

SDN comes with three major capabilities:

- A means to take control functions out of a dedicated box and centralize them and create applications for such purposes
- A means to dynamically program data paths through the network
- A means to directly program packet processing on a chipset

The first two play a key role when steering subscriber traffic. The last one enables programming user plane packet processing for a service edge onto programmable silicon.

When looking at a customer / CPE attachment process, there are two major stages:

**Stage 1: Device Attachment and Recognition**

A device is powered on and attaches to the access network of the service provider domain (usually the AN). Layer 1 comes up and the AN needs to enable the L2/L3 connection set up. To do so, an AN can create an event like a 'Port Up' message that is processed by the SDN control framework. Based on implementation, a network path can be created to enable step 2, where the device attaches to the service edge / BNG. CORD does this using 802.1x port authentication.

**Stage 2: Subscriber Session Establishment**

Prerequisite of this step is the reachability of the BNG/SE. Depending on the service provider policy, this additional step is required here for CPE

 

authentication and access protocol establishment (e.g. PPPoE termination and PPPoE/L2TP).

The actual SDN and BNG implementation may benefit from external state databases. For the traffic steering mechanism in stage 1, it is obvious that these created flows need to be stored in a central database inside or attached to the SDN controller. This may also include implicit authentication states such as e.g. via line IDs.

For the subscriber session state (stage 2), the options depend on the BNG option chosen. For external BNGs, storing the subscriber session state is out of scope for the SEBA POD.

In case the BNG resides in the SEBA POD, independently on whether it is embedded in an ASG element or a standalone PNF/VNF, session state may be kept internally to this instance or, in order to e.g. allow for fast failovers, be stored in a centrally accessible state database inside the POD.

#### *2.3.2.5 Aggregation and Service Gateway (ASG)*

Aggregation and Service Gateway (ASG) devices (switches) support Layer 2 or Layer 3 network aggregation, switching, and routing of data plane, control plane and management network connectivity within the POD as well as to external data networks, and supports Service Edge/BNG capabilities.

There may be one or more ASG devices, and setups as switching fabric, depending upon the implementation.

#### *2.3.2.6 AN Driver*

The AN driver shall be a collection of loosely coupled services which provide an abstract interface from the SDN controller to target device hardware. Different AN drivers can support many technology types such as PON, XGS-PON, NG-PON2, Gfast, Fixed Wireless Access (FWA) or DOCSIS.

A PON AN Driver shall be developed to support XGS-PON and similar technologies such as GPON, EPON, or NG-PON2. OLT/ONT hardware can be delivered in many forms. Vendor OLTs/ONTs, whitebox OLTs/ONTs, and pluggable OLTs are all supported.

Adapters provide an interface from the core AN Driver to the specific hardware implementation and devices within an AN (e.g. OLT and ONU for PON). The PON AN Driver hides PON level details (T-CONT, GEM ports, OMCI

 

etc.) from the SDN controller, and abstracts each PON as a pseudo-Ethernet switch easily programmed by the SDN controller.

The AN Driver has the responsibility of establishing the data plane connections through the hardware by interpreting service requests from the SDN controller and transforming them into requests to be fulfilled by the appropriate adapter.

The PON AN Driver has the responsibility of forwarding control plane requests to the SDN controller. Control plane requests correspond to authentication protocols (802.1x, PPPoE, DHCP) and multicast service such as IGMP, and OMCI messaging.

The PON AN Driver provides to the SDN controller the ability to manage and control the ONU through OMCI or eOAM messaging.

A DPU AN Driver provides the SDN controller to manage and control the aggregation functions of the DPU and the Gfast access interface functions of the DPU.

#### *2.3.2.7 ASG Driver*

The ASG Driver provides the management and control functions for the ASG devices. Functions for user plane aggregation include create, delete, update and retrieve L2 or L3 connections between access ports and uplink ports, and to monitor these connections. Functions for management control plane connectivity include (as applicable) to create, delete, update and retrieve management and control paths between ASG devices and compute servers, between certain ANs and compute servers, and from ASG devices to external BNGs or routers.

Note that an ASG device does not necessarily attach to all types of ANs. ASG may attach for example to PON AN, and the PON AN may in turn attach to a DPU (G.fast) AN.

#### *2.3.2.8 Access Node (AN) Types*

The Access Node Types are a specific implementation of a broadband access technology, such as PON technology. An AN type includes device types within a technology, such as network aggregation devices, intermediate devices, or customer network terminations. Vendors can produce AN devices providing drivers to interface to the required adapter. White box AN devices based on industry standard chipsets are used. Drivers provide a bridge

 

between hardware supplied SDKs (software development kits) and the required adapter.

AN devices provide the physical layer termination of the network access ports and the aggregation of the traffic to the ASG switch. The number of ports provided can vary based on the hardware implementation, such as 8, 16, and 24 access ports and beyond.

#### *2.3.2.9 Profiles*

A Technology Profile (TP) helps to define a subscriber service. It contains AN specific parameters specific to a technology such as GPON, XGS-PON, NG-PON2, EPON, future PON technologies, DOCSIS, Fixed Wireless, Ethernet, xDSL etc. Thus, the profile is specific to the technology.

A device adapter interprets the technology profile. Multiple technology profiles can be defined for a specified technology type. These profiles define the service level characteristics. A residential service could use a weighted 4 queue model while a business service could require a strict priority 8 queue model.

A speed profile will define the service parameters related to the bandwidth achievable by a subscriber. Depending on the type of service being offered different parameters may be defined. A residential service could define a minimum and maximum speeds. A business service could define minimum, maximum, and guaranteed speeds.

Different technologies may implement speed profiles differently. Some may use simple meter bands (XGS-PON) while others may manage the physical line (Gfast sync rates).

#### *2.3.2.10 Access Technology - PON*

SEBA is expected to support various kinds of PON-related technology (e.g. GPON, XGS-PON, EPON, Gfast etc..) and physical devices. In this architecture, these PON-specific features and devices (i.e. OLT/ONUs) are abstracted by AN Driver into a pseudo-Ethernet switch whose ports correspond to ONU-UNIs and OLT-NNIs. This abstraction provides operators with various options to deploy PONs that have different equipment structures while managing them in a common SDN architecture. For instance, it is possible to use both types of OLT: Box-type OLT (refer to Figure 1, "High Level Target Architecture") and pluggable module-type OLT (refer to Figure 4). In the latter case, the control messages sent from the AN

 

Driver to OLT go through the ASG, while the messages are directly sent to the OLT in the former case.

It is also possible to run TC (Time Critical) functions (e.g. DBA (Dynamic Bandwidth Allocation) function) apart from hardware (refer to Figure 5 (right)) instead of running these functions inside the hardware (refer to Figure 5 (left)). In any case, the PON-specific features are managed under the AN Driver.

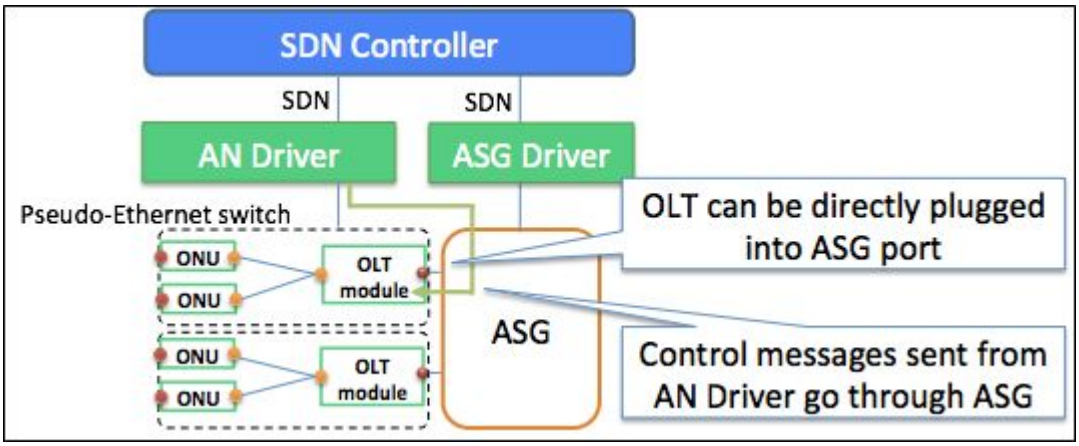


Figure 8: Architecture of pluggable module-type PON-OLT.

(Refer to Figure 9 for Box-type PON-OLT architecture)

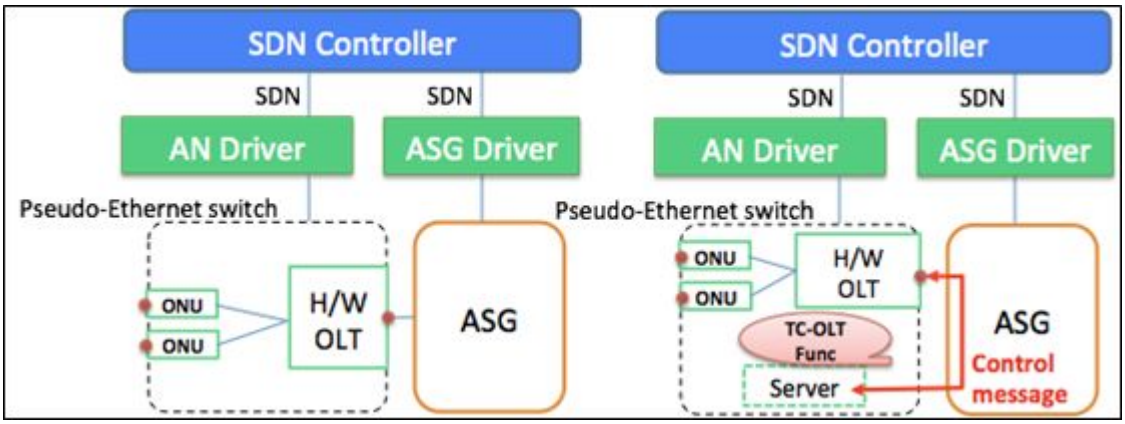


Figure 9: Options for PON-OLT architecture

(Left) Single H/W-type PON-OLT architecture.

(Right) Functional decomposition-type PON-OLT architecture.

#### *2.3.2.11 Access Technology – Fixed Wireless Access (FWA) / mmWAVE*

According to SEBA's basic modularized design approach, the SEBA reference architecture is also able to cover further access network technologies besides various kinds of PON. Today, many telcos and service providers consider access networks that attach their customers via a Fixed Wireless Network Access technology (e.g. mmWave – 28 or 60 GHz or WiFi) to their fiber-based backhaul networks.

Many other usage scenarios can be based on FWA technology, for instance small cell backhauling, WiFi backhauling, WTTB (Wireless to the building) etc.

The realization of FWA can broadly vary in terms of technologies (mmWave, WiFi), network topologies (tree, mesh, star, pt-to-pt, pt-to-mpt etc.) and transport mechanisms (L2 or L3 based), which all will be abstracted and managed by the FWA AN driver.

Furthermore SEBA is considered as the right architectural approach in order to cover all of these different deployment options and provide telcos and service providers with a common tool to produce different kinds of customer network access (FWA, PON, G.fast, …) in the same way with (mostly) the same hardware and processes.

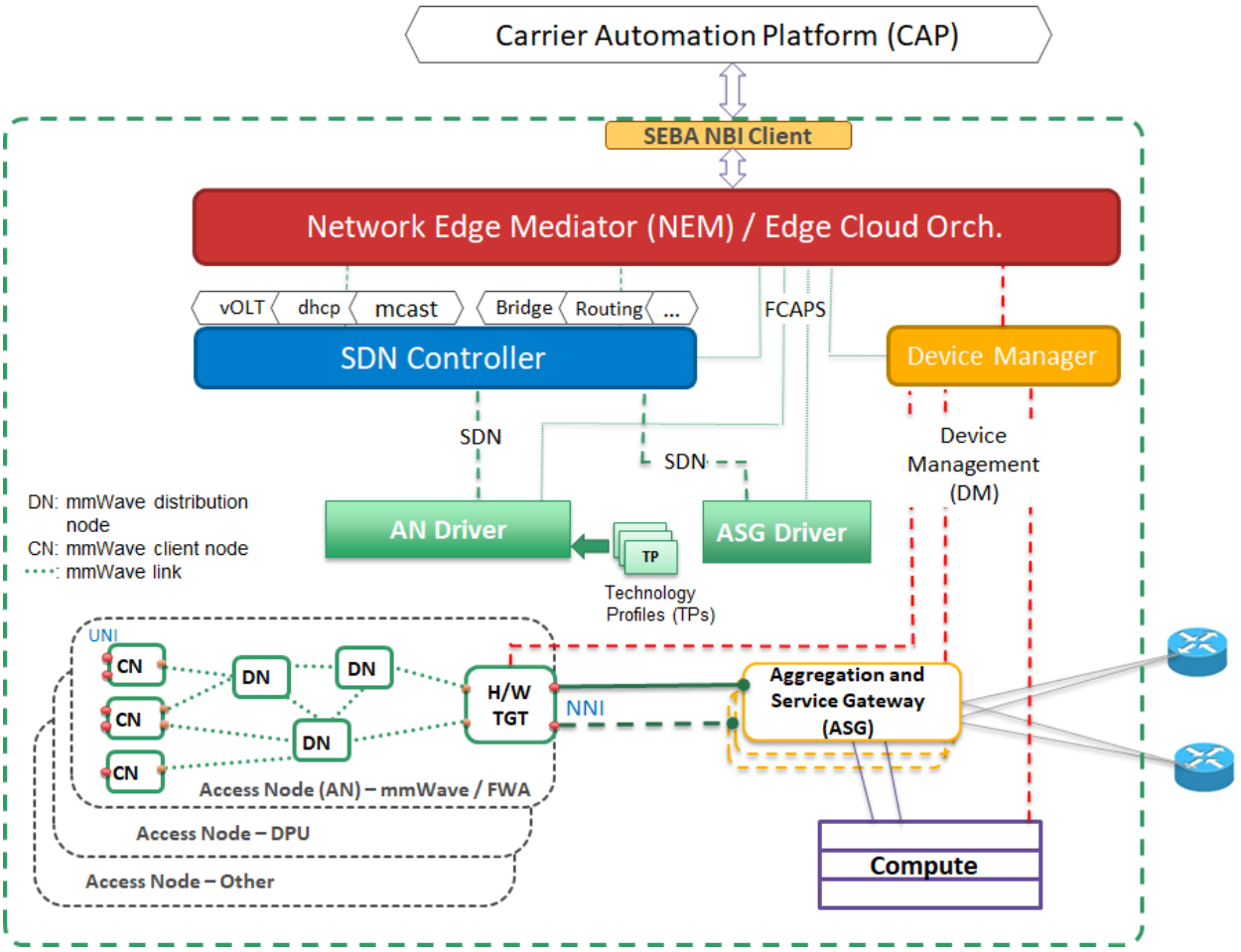


Figure 10: Exemplary mmWave-based ("Terragraph") FWA Integration into SEBA
(DN: mmWave Distribution Node, CN: mmWave Client Node, TGT: Terragraph Termination - HW / Functional building blocks for providing missing telco features to FWA domain)

In the recent RD the exemplary mmWave based FWA domain is assumed to be backhauled either by point-to-point fiber connections or using already deployed PON networks. These different deployment models are sketched with the following figures.

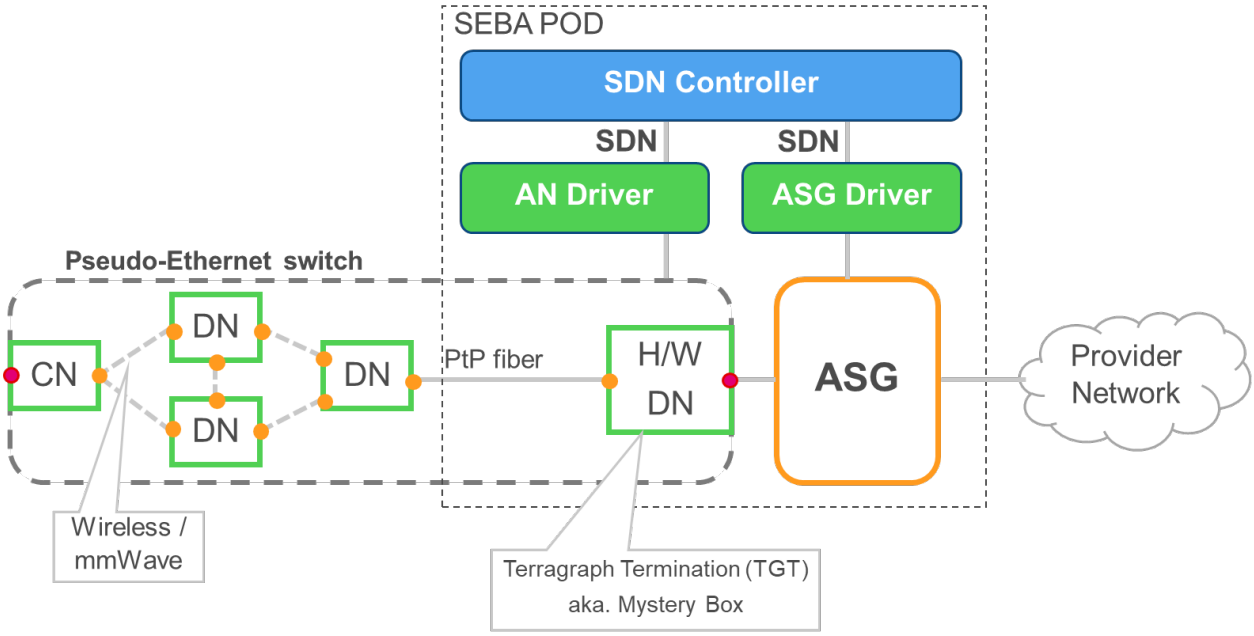


Figure 11: PtP Fiber – "Simple" FWA Model
(DN: Distribution Node, CN: Client Node)

Within the "Simple" FWA model, the FWA domain is backhauled using a PtP fiber connection. Additional functionality for providing eventually missing carrier grade / telco features to the FWA access domain are realized by the so-called TGT / Mystery box within the SEBA POD.

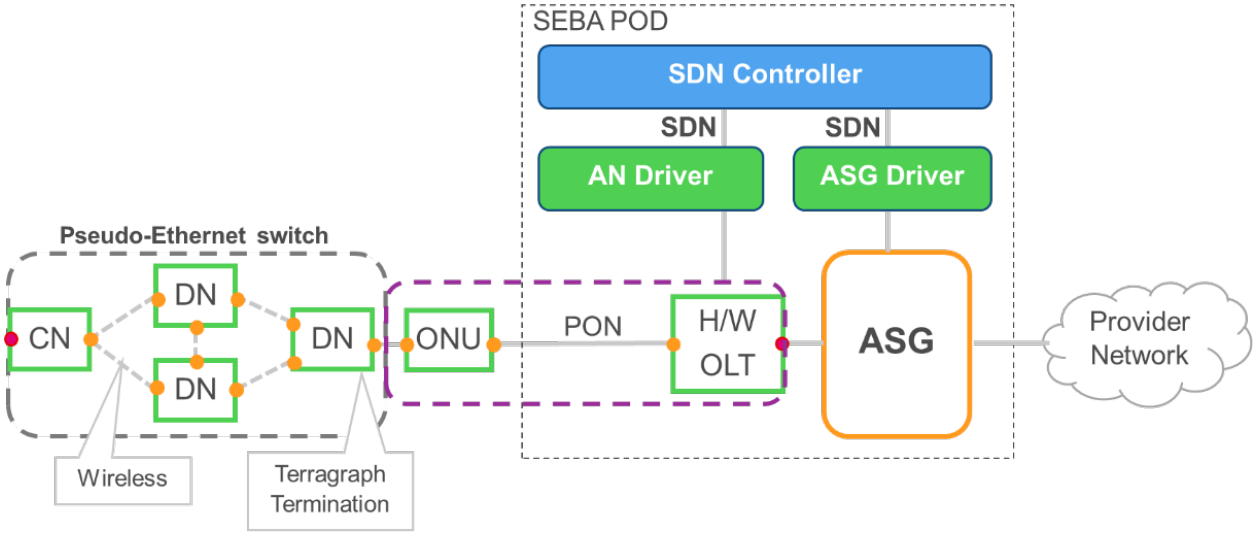


Figure 12: PON based backhauling of FWA domain – "Cascaded" FWA Model

The "Cascaded" FWA deployment model backhauls the FWA domain through existing PON network infrastructure. Assuming that the PON backhaul network as well as the FWA domain are both managed by the SEBA POD, this deployment scenario can be considered as cascading two SEBA managed instances

In this scenario, the TGT has to be provided by the rightmost POP DN of the FWA domain.

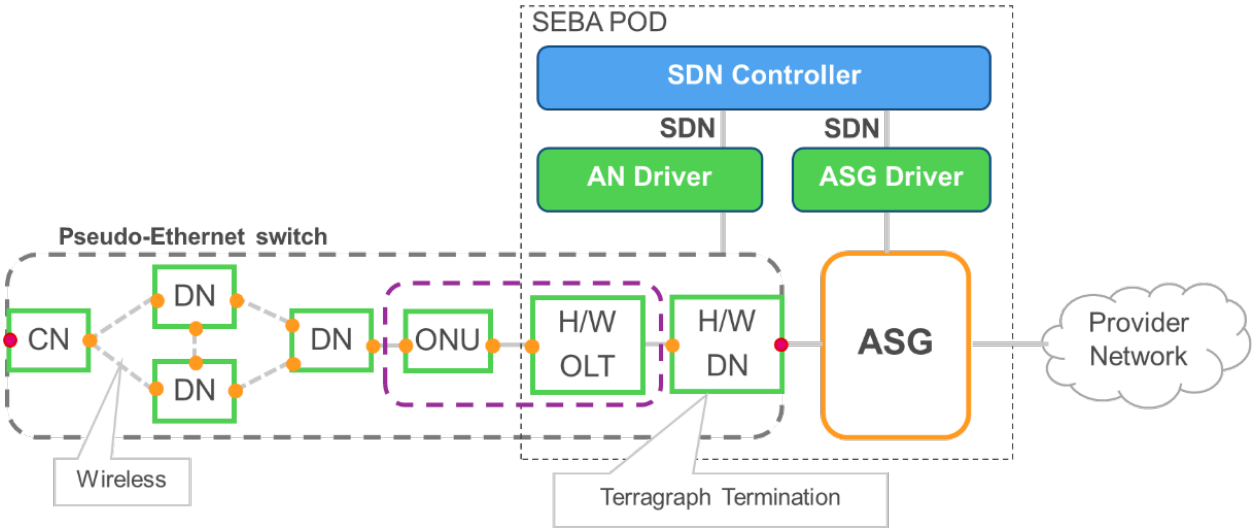


Figure 13: PON based backhauling of FWA domain – "Nested" FWA Model

The "Nested" FWA Model assumes that the wireless access network domain is attached via PON towards the SEBA POD and that the FWA network itself does not provide all the needed carrier / telco / service provider functionality. In this case, the missing functionality (TGT) has to be placed inside the SEBA POD and the existing PON infrastructure is used as the wiring in a distributed pseudo-Ethernet switch. When the SEBA POD is managing both - the FWA domain and the internally used / nested PON access network infrastructure - a "nested" deployment scenario comes up.

Focusing on the most likely deployment of the "Simple" FWA Model, the AN Driver (which is abstracting the concrete realization of the FWA domain) can be realized by choosing one of the following three implementation approaches:

1. A dedicated FWA AN Driver uses (like VOLTHA) the SEBA southbound interfaces of the SDN controller and NEM and provides all the needed abstractions related to the specific FWA deployment. Hence, the whole

FWA domain is presented towards the SDN Controller like a distributed Ethernet switch.
2. FWA AN attaches to the VOLTHA southbound interface from "below" and acts / behaves like an OLT.
3. FWA uses the existing VOLTHA implementations with a dedicated / special FWA technology profile.

#### *2.3.2.12 Infrastructure Requirements*

**Scaling**

The scalability requirement is deployment dependent. It depends heavily on the implementation network architecture and services offered from the POD. The SEBA community shall conduct the scalability study and provide the deployment guidelines which meet the service provider implementations.

SEBA POD must support scaling the POD elements horizontally to accommodate from the initial deployment to the larger footprint by adding the ANs and resources.

Vertical scaling of a POD means adding more resources to VNFs or into operating hardware. Horizontal scaling is preferred in clouds to simplify operations. The application of vertical scaling must be justified as an alternative to horizontal scaling.

An exemplar implementation for horizontal scaling is the per OLT VOLTHA Stack Model, as shown in the figure below, where a VOLTHA stack, consisting of Read/Write Core, OpenFlow Agent, OLT Adapter, and ONU Adapter modules, supports the deployment of a single OLT. When an additional OLT is required, a VOLTHA stack is initiated to support the new OLT. Likewise, when an OLT is removed then the stack supporting that OLT can be removed, freeing compute, storage and network resources. It is also possible to scale the ONU Adapter modules in a given stack, in case of addition/removal of a vendor ONU into an OLT or an increase/decrease in the number of ONUs attached to an OLT.

 

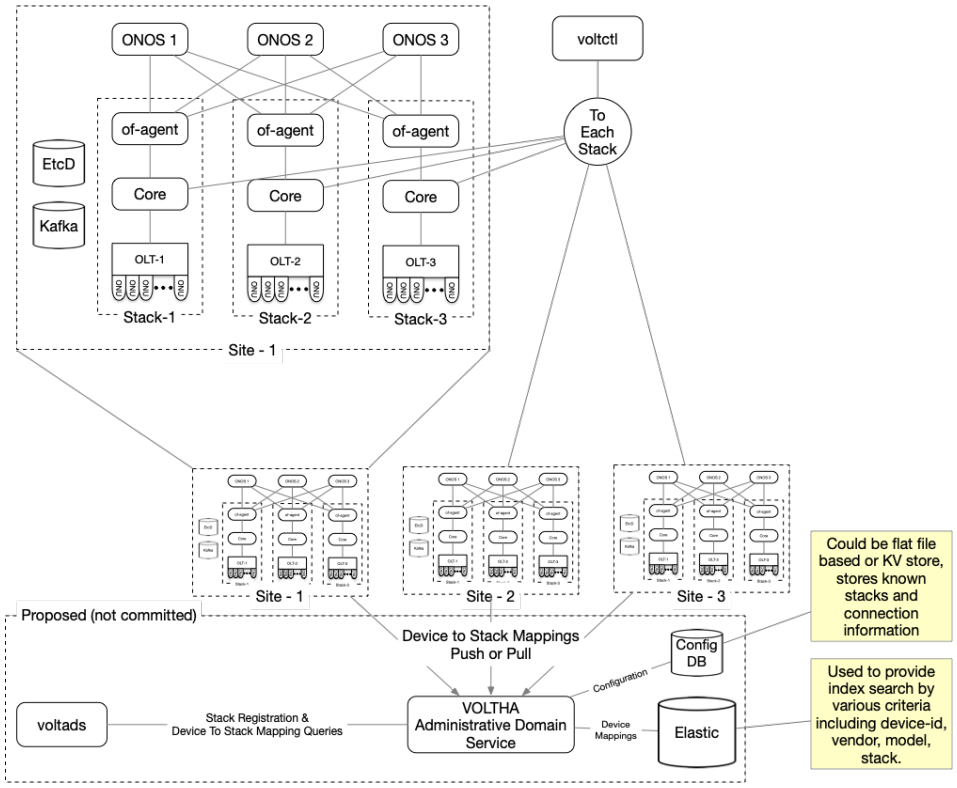


Figure 14: Per OLT VOLTHA Stack Model

### Resource Management

Resource management includes the definition of compute, memory, storage and network connectivity and bandwidth required for the initial POD installation, and the modeling and capacity management plan as additional utilization grows by ANs and subscribers.

### Data Collection

The status and the health of the SEBA POD, including all physical and local components and services, must be monitored and provide on-demand and periodic reporting mechanism to the CAP through the SEBA NBI Client.

Operation configurable Threshold Crossing Alert (TCA) must be utilized for raising the event to inform the operator about the state of the POD elements.

### Overlay Networking Requirements

SEBA POD must provide secure communication channels internally within the POD and externally to the provider's management network. The data path channel must not be compromised and provide unlawful access to the

 

management and control channels into the POD infrastructure and provider's management network.

**Speeds/Feeds, Performance (e.g., compute and I/O)**

The physical connectivity between the POD to the access network and backbone must maintain the SLA of the services offered of the POD. The connectivity requirement may vary based on the overall access network architecture design and the type of the services offered. The SEBA exemplar implementation of a provider will determine the requirement. The requirement shall define whether it is Layer 2 and/or Layer 3 connections.

Defined by the implementation, the oversubscription over the physical bandwidth may be required.

The SEBA project shall provide the compute and storage performance based on the provider's implementation and scalability requirements.

### 2.3.3 POD Assembly

The assembly of a POD includes the definition of the virtualization approach of the software to the hardware resources.

The exemplar platform begins with a set of container elements run in a Kubernetes environment to optimize the utilization of compute resources.

Large-scale orchestration and lifecycle management of PODs from an operator's cloud environment may lead to approaches to standardize on other variants of infrastructure environments.

A VIM or Multi-VIM approach may enable SEBA to run on other infrastructure environments.

A Virtual Infrastructure Manager (VIM) controls and manages the NFV infrastructure (NFVI) compute, storage and network resources, usually within one operator's infrastructure domain. VIMs can direct a multidomain environment or optimize to a specific NFVI environment. A VIM coordinates the physical resources to deliver network services.

### 2.3.4 Use Cases and Flow

The workflow branch under the SEBA project wiki page hosts the SEBA use cases and workflows from multiple Service Providers and will become the basis for the SEBA implementation streams.

 

## 2.4 COMPLIANCE WITH END STATE

This section defines guidelines that the RD should follow to achieve the desired end state. Include Key Performance Indicators (KPIs) to achieve the technical and business goals.

Guidelines for the end state of the RD include:

- Suppliers can use this RD to build the solution
- Providers can verify compliance by the suppliers to the RD
- Reliability above a defined number of "9s" for availability of the solution

# 3 TIME TO MARKET SOLUTIONS

There will be multiple options, some trivial and some substantial. The motivation is for an operator to realize a substitutional model for hardware and software selection to enhance features and reduce costs. There may also be variants amongst the operators in the operator group. It is highly desired that operator variants be minimized to the functional components rather than the interfaces between the components.

## 3.1 SOLUTION ELEMENTS

### 3.1.1 Major Functional Elements

The following sections define the functional building blocks.

#### *3.1.1.1 Carrier Automation Platform (CAP)*

The Carrier Automation Platform that is external and northbound of the SEBA is a robust design framework that allows specification of the service in all aspects – modeling the resources and relationships that make up the service, specifying the policy rules that guide the service behavior, and the applications, analytics and closed-loop events needed for the elastic management of the service. ONAP is an example of a CAP.

 

The orchestration and control framework (Service Orchestrator and Controllers) is recipe/policy-driven to provide automated instantiation of the service when needed and managing service demands in an elastic manner.

The analytic framework closely monitors the service behavior during the service lifecycle based on the specified design, analytics and policies to enable response as required from the control framework, to deal with situations ranging from those that require healing to those that require scaling of the resources to elastically adjust to demand variations.

#### *3.1.1.2 Infrastructure Layer*

The Infrastructure layer includes the hardware in the solution, including the devices, racks and shelves, powering equipment and connections, and external fibers or electrical cables, and other passive devices.

The devices in the POD include the ASG devices, compute servers, OLTs, or other devices defined by the technology (e.g., other types of devices for Wireless or DOCSIS solutions that become defined for SEBA).

The external fibers are part of the infrastructure as they connect to ports of the devices and carry user plane and/or management traffic, and the devices monitor performance of signals and protocols carried by the fiber media.

The passive devices include PON splitters, patch panels, and other equipment that do not monitor the signals, but which perform essential functions to connect paths, combine signals, split signals or filter signals.

The combination of components in the Infrastructure layer compose the physical inventory of the solution that suppliers plan for delivery, fulfillment solution providers aggregate in a supply chain, and that installers place and validate.

#### *3.1.1.3 Physical Topology*

The physical topology presents more detail about the detailed organization and connectivity of the infrastructure layer components into a complete solution.

 

#### *3.1.1.4 Service Layer*

The Service Layer defines the attributes of the service, and the configuration and binding of the components in the Infrastructure layer to deliver a service.

#### *3.1.1.5 Application Layer*

The application layer is the Open Systems Interconnection (OSI) layer closest to the user. This layer establishes communications between applications, and to the user. An example SEBA component in the Application Layer is the SEBA NBI client.

### 3.1.2 Interfaces and Interior APIs

The functional descriptions of the internal interfaces include:

- API between CAP (Carrier Automation Platform) and the SEBA NBI client
- API between SEBA NBI Client and NEM
- API between NEM and SDN Controller
- API between NEM and AN Driver (FCAPS)
- API between NEM and ASG Driver (FCAPS)
- API between NEM and Device Manager (DM and FCAPS)
- API between Device Manager and AN Driver (Device Management - DM)
- API between Device Manager and ASG Driver (DM)
- API between Device Manager and Compute (DM)
- API between Edge Cloud Orch and Compute (orchestration and life cycle management)
- API between SDN Controller and AN Driver (SDN)
- API between SDN Control and ASG Driver (SDN)
- API between AN Driver and AN
- API between ASG Driver and ASG
- Interface between DPU of a G.fast AN to an ONU of a PON AN
- NNI interface between AN and ASG
- NNI interface from ASG to external BNG/router

 

### 3.1.3 Security

This section has been generalized since RD 1.0, following discussions with the SEBA development community.

As a managed virtualized environment, SEBA should as possible derive and leverage security requirements, security architecture, security best practices and open source security solutions from other open communities.

An operator will need to adapt SEBA to its security practices. SEBA can implement some core functions for security that may help to implement security at perimeters such as APIs, and OS and Kubernetes.

Operators should specify security implementations to the SEBA development community, if those implementations benefit core security functions that are better maintained in SEBA open source. For example, SEBA transactions that change any configurations or perform any operator actions from any operator or automated interface (CLI, Northbound API, control loop action) could use a Transaction ID that passes through the SEBA processing of configuration actions, in order to perform effective security Audit Logging & other Logging.

### 3.1.4 Reliability and Resiliency

Reliability is the definition of the probability of a system or component to function under stated conditions for a period of time. The reliability is also defined in terms of availability of a system or component in terms of number of “9s”, such as “five 9s” indicating 99.999% availability.

A reliability analysis is recommended for the components of a POD, and of the entire POD to predict the availability of the POD as a system.

Resiliency is the ability of a server, network component, or POD to recover from a failure (such as a power failure, or equipment failure) and quickly resume operations.

Redundancy or clustering is employed to help improve both reliability of a POD, and resiliency of operations in a POD.

### 3.1.5 System Performance

System performance measurements in the internal processing functions of the POD are important to measure and understand with respect to system response for control and management transactions, and for scalability of the system to provide a determinate number of operations of a certain type, while the system is operating under a defined load profile (profile of a variety of defined operations at defined frequencies over a defined period of time).

The implementation of monitoring tools to monitor system performance must be considered and selected to minimize their own impact on system performance and resources.

### 3.1.6 Capacity Management

Capacity management provides analytics and reporting of the resources needed for a solution. It derives capacity measurements from the Performance Management element and from the resources that define the capacity of a solution element.

The implementation of capacity management may occur in the Carrier Automation Platform (CAP) that receives performance measurements from the POD(s), and so the Performance Management collection requires forwarding to the CAP for that function.

A provider may determine that Capacity Management functions are implemented in the local SEBA POD and provided through a local operator interface.

### 3.1.7 Fault Management

Fault Management applies at a SEBA POD level. A Carrier Automation Platform (CAP) that attaches to the SEBA POD through the SEBA NBI client typically provides two functions at an enterprise level for fault management: (1) Show Current Active Alarms, and (2) Show Alarm & Event History.

For purposes of this discussion, an Alarm can also be a “standing condition” that has a set/cleared state but is mapped to a “not alarmed” severity.

It is desired for NEM to provide a normalized collector implemented in Kafka for all faults for external northbound OSSs/Orchestrators to be able to receive streams of fault history in a deterministic manner. As a streaming platform, Kafka provides the capabilities to publish and subscribe to streams of records, to store streams of records in a fault-tolerant durable way, and to process streams of records as they occur.

While VOLTHA publishes faults to its Kafka bus that NEM can provide to its northbound clients, other faults may be derived from other APIs in a SEBA POD such as Redfish for device management. NEM can develop a "transformer" as necessary to export data from other APIs to a Kafka topic, and thus provide all faults through a normalized collector in Kafka.

Northbound OSSs/Orchestrators can implement and install an agent as a microservice in the SEBA NBI client that subscribes to the normalized Kafka collector and transforms the faults to a desired format (e.g., VES for ONAP, IPFIX for another OSS, etc.) for transport to the Northbound CAP. The SEBA NBI client for a CAP is not part of SEBA, as different CAP implementations may use different management protocols.

As a SEBA NBI client can discover or re-attach to a SEBA POD at any time, it may be able to depend upon cached alarms and events in the SEBA POD and to re-sync with this incremental history to correctly update its "Show Current Active Alarms" function, if and only if the SEBA NBI client can resync from the actual last alarm or event from a prior attachment to the SEBA POD.

The Kafka bus implementation in SEBA should allow the SEBA NBI client to determine if it is able to sync to the last known fault management state of the SEBA POD. If the SEBA NBI client is not able to sync to the last known state, then it will have missing alarms and events. In that event, then the SEBA POD should provide a "Retrieve all Active Alarms from SEBA POD" function.

### 3.1.8 Configuration

SEBA shall provide abstract configuration interfaces to provide subscriber level services. SEBA can be used in multiple deployment environments. Each implementation should provide simple easy to use APIs providing the minimum required parameters.

#### *3.1.8.1 Abstract OLT (Removed)*

The previous RD version 1.0 discussed a model to represent multiple AN devices as a single device to northbound systems. At this time this will no longer be pursued in favor of a per AN device model.

#### *3.1.8.2 Profile Configuration*

SEBA shall provide the ability to manage profiles including technology profiles and speed profiles and associate these profiles to service types defined by the operator.

#### *3.1.8.3 Service Configuration*

Service configuration will identify the physical location (i.e. device name, port, pon id for PON), service type, speed profiles, and VLAN tags. Service types are defined by the operator (RESIDENTIAL or BUSINESS) as their service models require. Likewise, speed profiles are defined by required service tiers (100M, 500M, 1G). VLAN tags define the model expected on the AG switch uplink ports interfacing to the external BNG.

#### *3.1.8.4 Backup and Recovery*

The NEM must provide the capability to periodically collect configuration information from each of the POD components and export them to another safe location. Then, in the event of a software or equipment failure it must be possible to restore the SEBA POD and the customer's service using the backed-up system configuration information.

SEBA shall provide an API to backup the configuration information to an external system.

#### *3.1.8.5 Restore From Backups*

A service-affecting event may occur in which a part of or all of the SEBA POD components have been corrupted and rendered inoperable. In this scenario

 

the recovery plan would be to restore the POD components using recent backup files.

1. SEBA shall provide an API to restore a recent set of backup configurations.
2. The restore process shall be able to be monitored for progress and success.

#### *3.1.8.6 Software Lifecycle Management*

SEBA shall provide APIs capable of managing software for the components of the POD. For a PON AN type, this includes the physical equipment resident in the POD as well as all the ONT devices connected to the PON ports.

The components of the POD shall be managed independently. Where new services require software upgrades to various components of the POD those services will not be configurable until all parts of the POD have been upgraded.

During an upgrade APIs will be provided to track the progress. The APIs shall be able to determine the success of the upgrade activities and identify any fallout activities which are required.

Rollback to previous software components must be available if failure criteria are met.

Activities impacting customer service shall be performed in maintenance periods, usually 2 to 4 hours. Expected interruption to subscriber service shall be less than 5 minutes.

### 3.1.9 Accounting and Status

To operate the SEBA POD, interfaces are required to determine the operational status of the POD. Real time status shall be provided in the following areas:

- 802.1x Authenticator
- 802.1x Diagnostics
- 802.1x Session
- RADIUS Accounting Server
- ONT Status
- ONT Alarm Thresholds

 

- ONT UNI Port
- Current Optical Data
- Historical Optical Data
- PON
- PON SFP
- PON Utilization Data

### 3.1.10 Performance Management

VOLTHA publishes performance monitoring data to its Kafka bus that NEM can transform into bulk collections of data (such as organized by a collection interval).

NEM can also poll other performance monitoring data for other functions in a SEBA POD, for example using a Redfish API for device performance monitoring. NEM can develop a "transformer" as necessary from another API to a Kafka topic, and thus provide all performance monitoring through a normalized collector in Kafka.

Northbound OSSs/Orchestrators can implement and install an agent as a microservice in the SEBA POD that subscribes to the normalized Kafka collector and transforms the performance monitoring collections to a desired format (e.g., VES for ONAP, IPFIX for another OSS, etc.) for transport to the Northbound OSS/orchestrator.

The SEBA NB API should provide an API to retrieve all PM data and metrics for the "current" interval(s) - e.g. 15-minute and daily.

If it is possible for a Carrier Automation Platform (CAP) to be out-of-sync with the SEBA POD historical PM interval collection upon discovery or re-attachment to a SEBA POD, then the mechanism to re-sync the CAP from the Kafka bus needs to be determined and implemented.

### 3.1.11 Inventory

Inventory is the definition of the system components and interfaces. In a life cycle view, the planned inventory includes the expected components and interfaces that need to be operating in the infrastructure layer to deliver the POD functions. The actual or discovered inventory requires discovery and

 

validation against the planned inventory to provide the required infrastructure for services and operations of the POD.

### 3.1.12 Telemetry, Monitoring and Logging, Analytics and Policy Functions

Telemetry involves automatically recording and transmitting data from remote or inaccessible sources to a management system for monitoring and analysis. For this solution, it is encouraged to direct the collected telemetry data to the performance monitoring management subsystem for common processing.

Note that SEBA provides an optional Monitoring and Logging framework. See the SEBA Monitoring & Logging Infrastructure in the SEBA Design Docs. This optional framework is included via helm statements for monitoring and logging functions in the SEBA startup.

Note that the SEBA project currently does not include or plan to include an Analytics engine within SEBA, such as described in the proposed Analytics for CORD project (A-CORD). In the absence of an Analytics engine from the A-CORD project, an operator may develop its own Analytics applications in the POD that could interface to the SEBA Monitoring and Logging infrastructure.

Note that instead of building analytics applications in the POD, an operator may build centralized Analytics and Policy functions in its northbound CAP. The CAP will receive Faults, Telemetry and Performance Monitoring from the NEM adapters (see also sections above for Fault Management and Performance Monitoring) which subscribe to the Kafka bus and the transformers in the SEBA Monitoring & Logging Infrastructure, in order to provide centralized processing for Fault Management, Performance Management, Telemetry and any Analytics and Policy functions.

 

### 3.1.13 Automation and Management (includes Exterior APIs)

In a typical deployment scenario, there will be up to thousands of SEBA PODs installed.

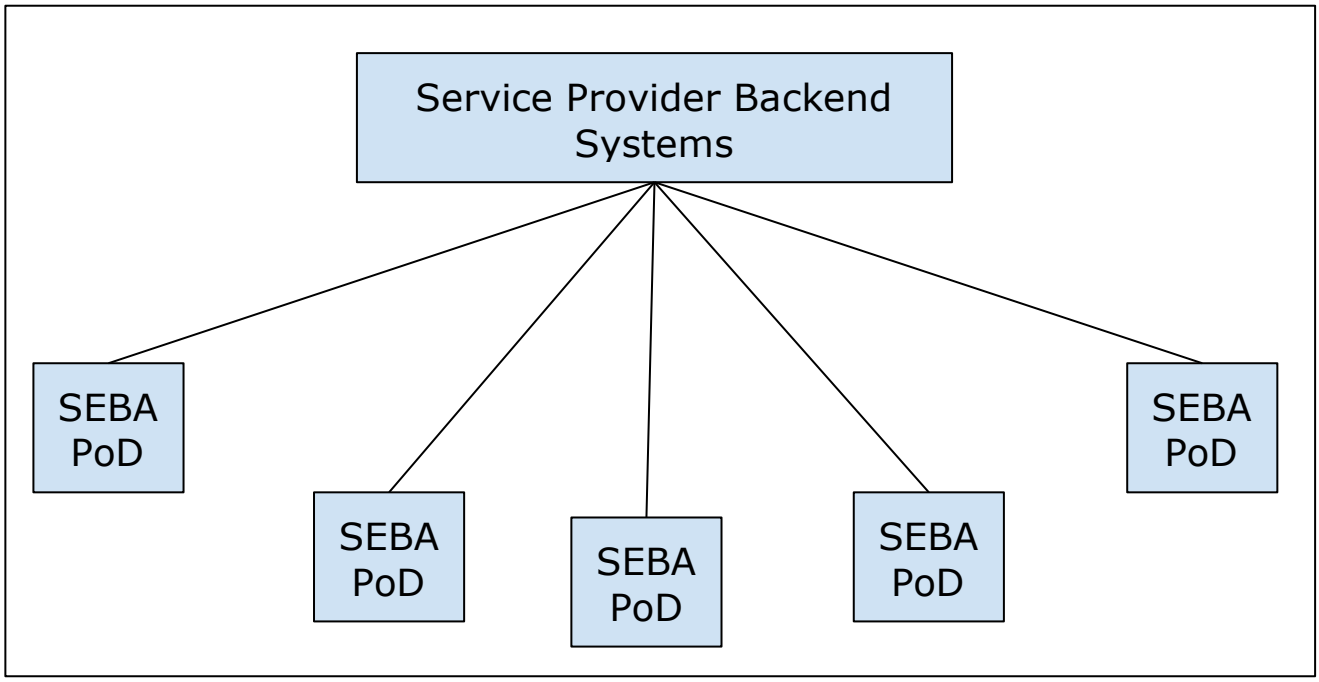


Figure 15: Service Provider Backend Systems to Many SEBA PODs

SEBA should be self-contained and be able to work at any service provider environment. To achieve these goals, we need to have an external adaptation box that sits on top of SEBA Northbound API and may be remotely positioned. An operator may instantiate the SEBA NBI client within a SEBA POD.

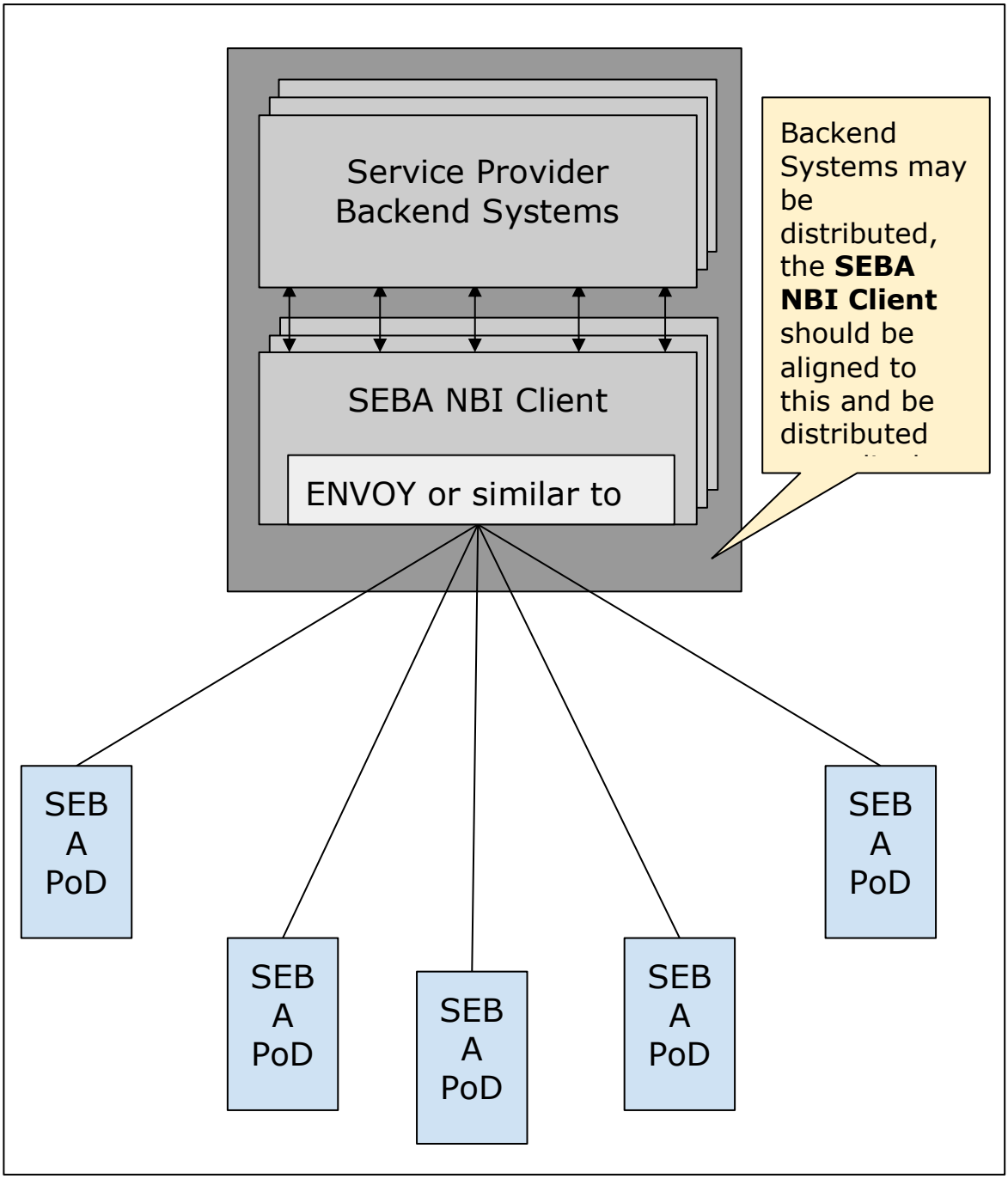


Figure 16: SEBA NBI Client Role

The SEBA NBI Client is tightly coupled with the Service Provider backend and will be Service Provider specific.

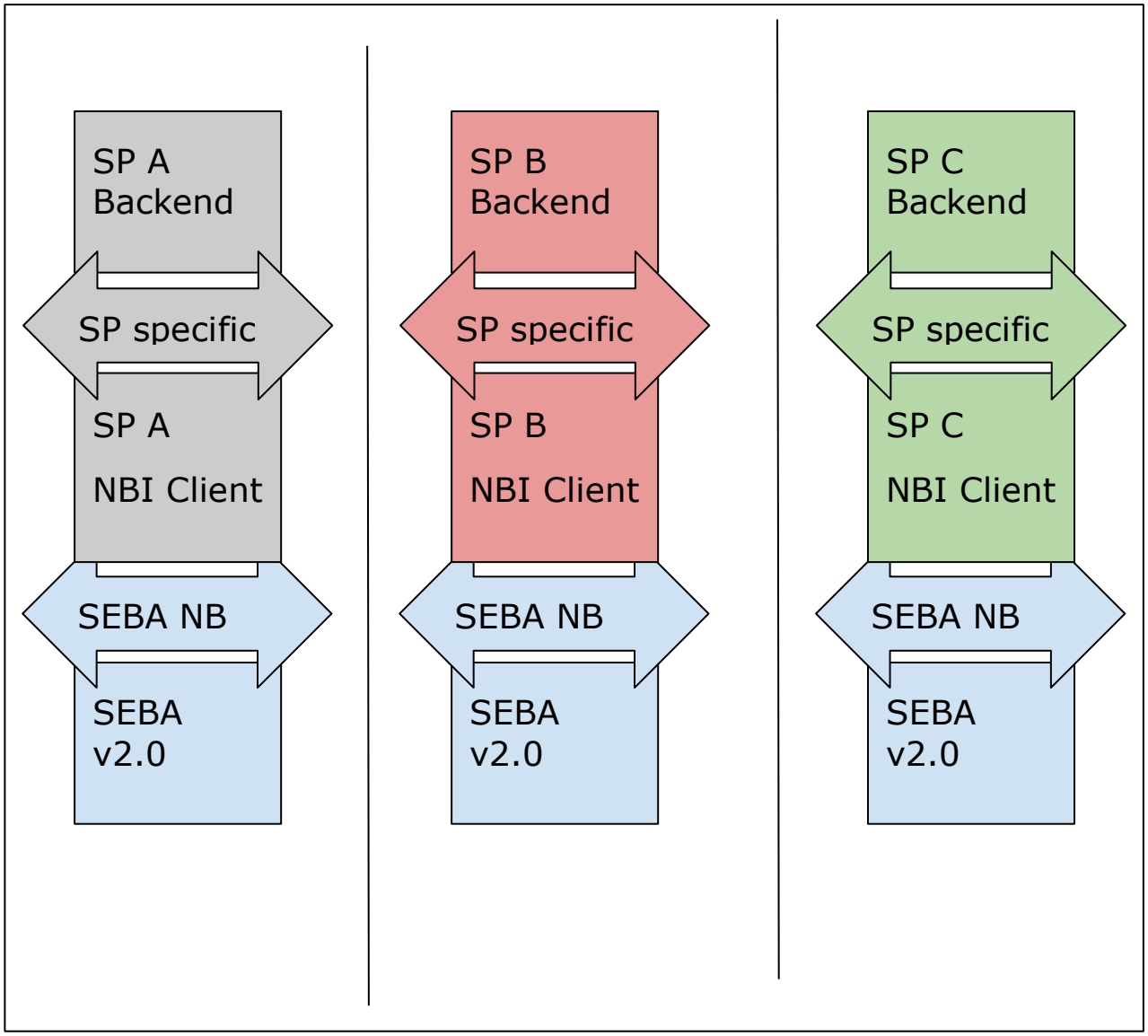


Figure 17: SP Backend, SEBA NBI Client (per SP), and SEBA NB API

SEBA should be self-contained and shall not be affected by any changes that may occur in the deployment environment.

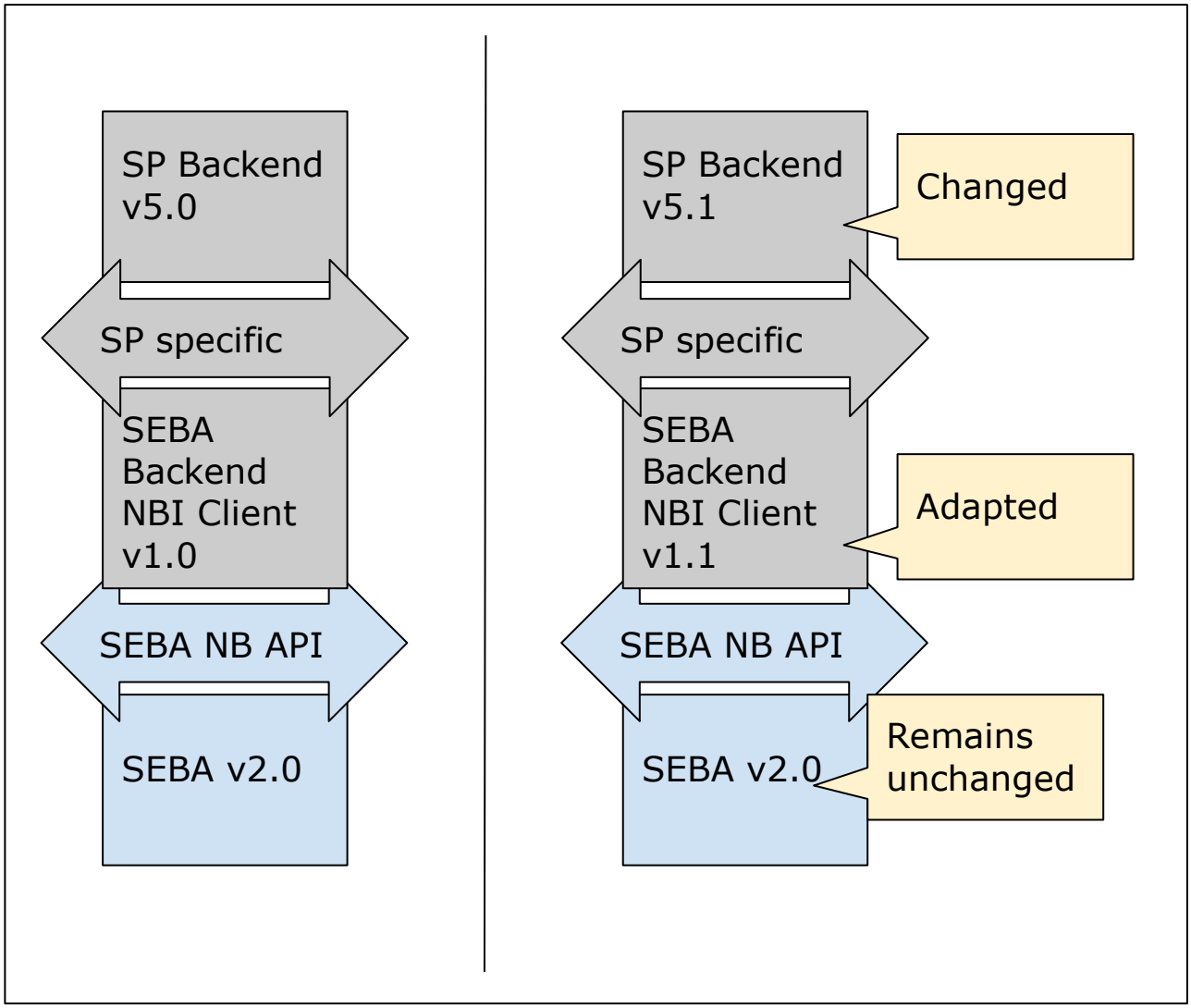


Figure 18: Example - No change to SEBA NB API

Similarly, version upgrades of SEBA shall not affect the backend systems.

As shown in the figure below, the SEBA NB API version may change. SEBA releases should define an NB API version with a documented API catalog and release notes.

The SEBA NB API, catalog and version should aggregate the APIs of all NBI services. For example, the SEBA NBI should include - VOLTHA NB APIs, SADIS (Subscriber / Access Device Information Service) NB APIs, and other NBI services in SEBA.

The SEBA NB API should remain backward compatible. An API call can be backward compatible by only adding parameters or parameter values. To propose to replace an API, there should be a new API defined to replace a current API. Only after confirming there are no users of an API to be replaced, is it possible to delete the former API. The release notes of a new API version shall document all API enhancements, new APIs, APIs proposed to be replaced, and deleted APIs.

The SEBA management interfaces can benefit from standardized modeling of Access devices and technologies using YANG models from standards development organizations (SDOs) - see github directory for SDOs. In this github directory, the Broadband Forum (BBF) maintains published YANG models for ITU-T PON (per BBF TR-385), FTTdp for G.fast and G.hn (TR-355), and Common YANG (BBF TR-383). The development of the SEBA NB API should consider conformance to these standardized YANG models for integration to a Carrier Automation Platform (CAP).

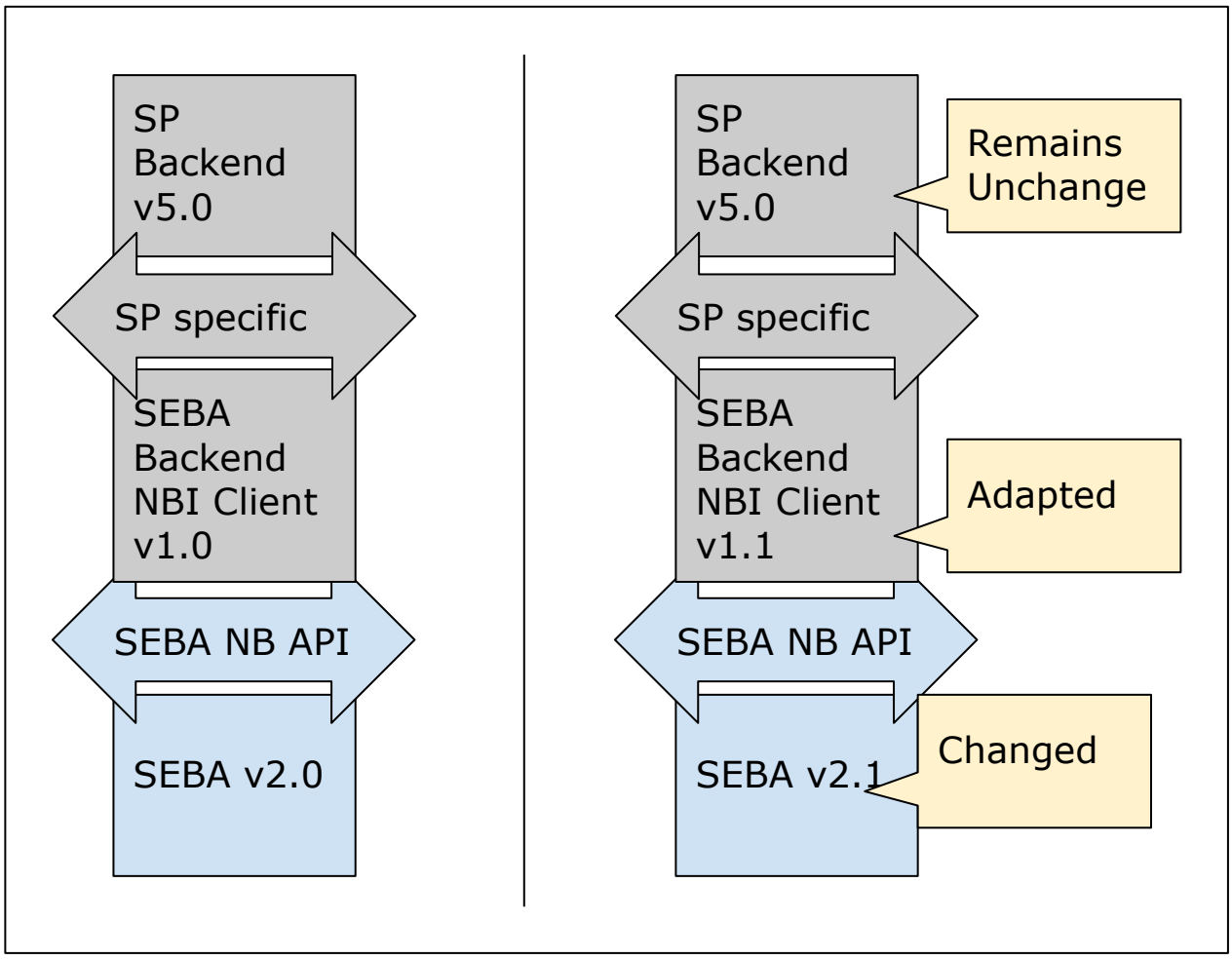


Figure 19: Example - No change to SP Backend

The adapter shall invoke relevant methods on the SEBA NB API and implements a set of callback API's for SEBA to call. The remote invocation shall be made through gRPC.

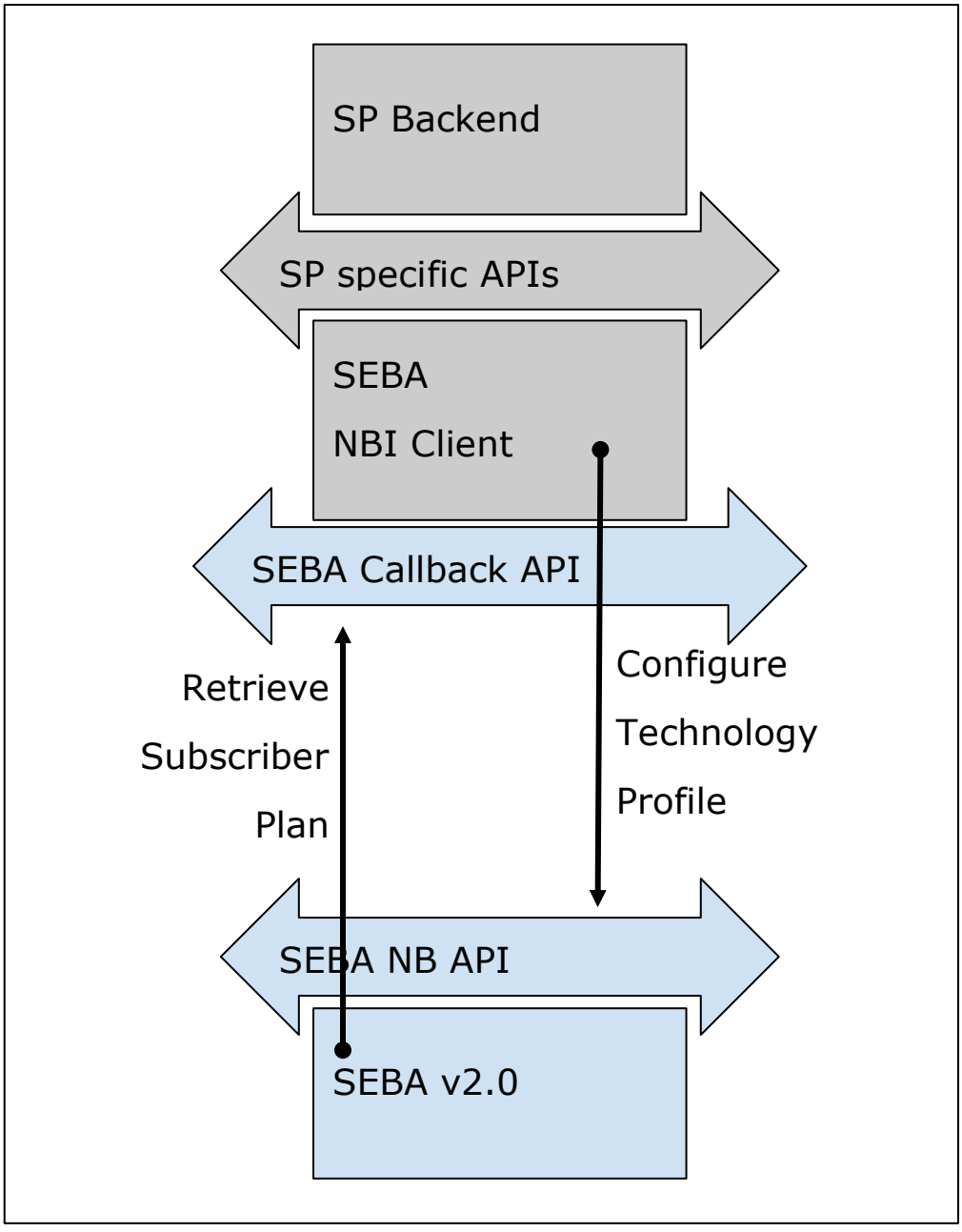


Figure 20: SEBA Callback API

The SEBA NB API will provide the following functionalities:

**POD Management**

- Provide inventory information for common hardware components
  - Download and manage software upgrades for ONOS, ONOS Apps, and VOLTHA components
- Monitor common hardware resources
  - Provide detail views on CPU utilization, component states, POD status, Container status
- Status Reporting
- Alarm Management
- Performance Monitoring

**OLT Management**

- Provision OLT hardware
  - Input: OLT device information
  - Return: OLT device information
- Assign CLLI associated to specific hardware inventory via serial number (or other unique identifier)
- Retrieve list of OLT devices
  - Input: None
  - Return: List of OLT device information
- Retrieve OLT hardware inventory information
  - Input: Device ID or OLT no
  - Return: OLT device information
- Retrieve list of OLT NNI ports
  - Input: Device ID or OLT no
  - Return: List of NNI ports
- Retrieve list of OLT PON ports
  - Input: Device ID or OLT no
  - Return: List of PON ports
- Retrieve OLT PON port information
  - Input: PON port ID
  - Return: PON port information
- Manage OLT software and upgrades
  - Input: Device ID, software
  - Return: Operation result
- Reset OLT hardware
  - Input: Device ID or OLT no
  - Return: Operation result
- Delete OLT hardware
  - Input: Device ID
  - Return: Operation result

 

- Run available OLT diagnostics and retrieve results
  - Input: Device ID
  - Return: Operation result
- Retrieve Operational Status - PON ports, optics, IGMP, multicast source table, authentication status
  - Input: Device ID
  - Return: Status
- Retrieve inventory information for SFP devices plugged into OLT ports
  - Input: Device ID
  - Return: Inventory information
- Disable OLT hardware
  - Input: Device ID
  - Return: Operation result
- Enable OLT hardware
  - Input: Device ID
  - Return: Operation result

**ONT Management**

- Provision ONT hardware
  - Input: OLT no, OLT PON port no, ONT serial no
  - Return: ONT no
- Update ONT hardware serial number
  - Input: ONT no, PON port no, ONU no, serial no
  - Return: Operation result
- Map upstream ONT identifications (OLT CLLI ONT port) to dynamic VOLTHA assignments
- Retrieve list of ONT devices
  - Input: OLT no, OLT PON port no
  - Return: List of ONT device information
- Retrieve ONT hardware inventory information·
  - Input: Device ID or serial no
  - Return: ONT device information
- Retrieve list of ONT UNI ports
  - Input: Device ID or (OLT no, OLT PON port no, ONT no)
  - Return: List of UNI ports
- Manage ONT software and upgrades
- Reset ONT hardware
  - Input: Serial no
  - Return: Operation result
- Manage associated ONT database configurations
- Delete ONT hardware
  - Input: (OLT no, PON port no, ONU no) or ONT serial no

    - Return: Operation result
- Run available ONT diagnostics and retrieve results
    - Input: Device ID
    - Return: Operation result
- Retrieve Operational Status - ONT, UNI ports, optics
    - Input: Device ID
    - Return: Status
- Retrieve inventory information for SFP device plugged into the ONT
    - Input: Device ID
    - Return: Inventory information
- Disable ONT hardware
    - Input: Serial no
    - Return: Operation result
- Enable ONT hardware
    - Input: Serial no
    - Return: Operation result
- Reset ONT UNI port
    - Input: ONT no, PON port no, ONU no, UNI port no
    - Return: Operation result
- Enable/Disable ONT UNI
    - Input: ONT no, PON port no, ONU no, UNI port no, status (ENABLE/DISABLE)
    - Return: Operation result

**Service Management**

- Provision service subscription
    - Input: OLT no, OLT PON port no, ONT no, UNI no, list of services
    - Return: Operation result
- Delete service subscription
    - Input: OLT no, OLT PON port no, ONT no, UNI no, service name
    - Return: Operation result
- Delete list of service subscriptions
    - Input: OLT no, OLT PON port no, ONT no, UNI no, list of service names
    - Return: Operation result
- Delete all service subscriptions
    - Input: OLT no, OLT PON port no, ONT no
    - Return: Operation result
- Enable/Disable service subscription
    - Input: OLT no, OLT PON port no, ONT no, UNI no, service name, service status (ENABLE/DISABLE)

 

    - Return: Operation result
- Create technology profile
    - Input: Profile ID, profile name, profile data
    - Return: Operation result
- Delete technology profile
    - Input: Profile ID or profile name
    - Return: Operation result
- Create service definition
    - Input: ID, name, ctag, stag, upstream/downstream ctag priority, upstream/downstream stag priority, vlan, technology profile id
    - Return: Operation result
- Delete service definition
    - Input: Service definition ID or name
    - Return: Operation result
- Get service definition
    - Input: Service definition ID or name
    - Return: Service definition
- List All service definitions
    - Input: Empty
    - Return: List of service definitions
- Create speed profile
    - Input: Profile ID, profile name, profile data
    - Return: Operation result
- Delete speed profile
    - Input: Profile ID or profile name
    - Return: Operation result
- Get speed profile /speedprofile/get
    - Input: Speed profile ID or name
    - Return: Speed profile
- List all speed profiles
    - Input: Empty
    - Return: List of speed profile data
- List ONTs having specific service
    - Input: OLT no, OLT PON port no, service name
    - Return: List of ONTs having subscriptions to the service
- List UNIs having specific service
    - Input: (OLT no, OLT PON port no, ONT no) or (ONT serial no), service name
    - Return: List of UNI ports having subscriptions to the service
- Get service subscription info
    - Input: (OLT no, OLT PON port no, ONT no) or (ONT serial no), UNI no, service name
    - Return: Service subscription information

 

**Device Management**

Device Management (DM) interface should support the data models and API specified by the SEBA/VOLTHA community. DM should include necessary mechanisms to perform the actions listed below.

- Inventory Management: It should be possible to automatically discover networked elements, as well as collect configuration and status data from them
- Status Monitoring: DM should support monitoring the status of devices, to proactively detect and remedy issues (e.g. device temperature, FAN failure)
- Log Management: It should be possible to manage logs and events originated from the devices, to help debug system issues.
- Device SW Maintenance: DM should provide support for upgrade/downgrade scenarios for all SW components, including firmware, ONIE, NOS, etc.
- Device Reboot: DM should enable the reboot of any device, to recover from errors that cannot be resolved by any other means.

Note: Since AN Driver and ASG Driver provide the major resource abstractions and handle key operations, Device Management only comprises the minimal set of (common) operations required to manage the SEBA devices, which is not already covered via AN / ASG drivers.

### 3.1.14 Design in Motion – Use Cases (SEBA POD for PON Technology)

#### *3.1.14.1 Day 0*

SEBA HW Installation

- SEBA Rack Installation
- SEBA vOLT Installation & Fiber wiring
- SEBA Compute Nodes Installation
- SEBA HW location information registration to SP backend

SEBA POD Configuration

- Done via human readable site.config (file)
- Applies POD id and basic configurations

 

- Configures the external management path
- Configures management networking within the POD, between the Compute, ASG, vOLT

SEBA Platform Installation and Orchestration

- Container based
- Uses precompiled container images
- Able to install with a single command line step
- Container and software upgrades

SEBA POD Activation

- The SEBA platform runs a sanity test suite and becomes active if it passes
- Calls a POD_Activated(POD_ID) on the callback endpoint.
- If the sanity test fails, use API to raise an error on the callback endpoint

SEBA platform installation automates installation and configuration of the host OS, network, Kubernetes cluster and SEBA container images.

SEBA workload orchestration includes container and software upgrades and lifecycle management (LCM).

The goals for the SEBA platform installation and orchestration include:

- Install host operating system (OS) on minimum three servers running in HA mode
- Validate/test host OS environment
    - OS release version
    - Connectivity to repositories
    - NIC interfaces
    - RAM
    - Storage
    - Permissions
- Install Kubernetes cluster on three servers
- Install SEBA software and supporting containers
- Validate running containers
- Expose northbound APIs for container and software upgrades

 

#### *3.1.14.2 Steady State*

At least the following basic use cases need to be supported for POD lifecycle management:

- SEBA POD Bringup
- Software Components (services) health check/monitoring, e.g. by Kubernetes pod supervision
- Network Elements "ready for operation" checks/monitoring
- Add Network Element (Server, OLT, ASG)
    - Including add container(s) as needed, download software, apply configuration
- Remove Network Element (Server, OLT, ASG)
    - Combination of remove/add can be used to handle hardware replacement.
- Upgrade Network Element (Server, OLT, ASG, ONU)
- Reboot Network Element (Server, OLT, ASG, ONU)

Notifications (start/finish/failure) to operations support systems should be supported for all lifecycle operations.

#### *3.1.14.3 Fault Detection and Recovery*

There can be local or global control loops for fault detection and recovery.

This section should provide uses cases for these control loops:

- Local control loops within SEBA
- Global control loops (outside of SEBA within the Carrier Automation Platform)

### 3.1.15 Tooling

Tooling of software is oriented to the effectiveness and efficiency of operations.

Operators have experience with the development of operator portals, logging and search mechanisms, correlation and analytics functions to improve the effectiveness of operations teams to troubleshoot and correct issues proactively or reactively for customers.

 

Operators should collaborate on Tooling features in SEBA that provide the most value in the exemplar platform.

## 3.2 SUPPORTING ACTIVITIES

### 3.2.1 Operational Plan

#### *3.2.1.1 Physical Environment*

The physical environment for a SEBA POD is likely to be mostly in a Central Office, but operators may pursue options to deploy a SEBA POD or its elements in a Data Center.

#### *3.2.1.2 Physical Requirements*

The physical requirements for a SEBA POD cover multiple areas such as space, rack placement, power, operating temperature ranges, cooling and heat dissipation.

Standards for a Central Office derive from telco industry standards and are referenced in Open Compute Project (OCP) definitions for Telco. Operators may also provide more specifics about their Central Office environments.

Standards for a Data Center derive from the data center industry, and support a wider range of open devices, as the standards may be less constraining for some requirements such as the operating temperature range.

### 3.2.2 Ecosystem Component Assessment

#### *3.2.2.1 Open Source Software*

Open source software should follow the guidelines of ONF as to the open software licenses that ONF projects can use to incorporate open source.

Operators and members contributing code to the ONF open source are responsible to conform to the guidelines of their companies or organizations to contribute code to ONF.

 

#### *3.2.2.2 Open Hardware*

The classification of open hardware will follow the definitions within the OCP, such as the "OCP Accepted" and "OCP Inspired" trademark definitions.

#### *3.2.2.3 Functional Decomposition Supplier Consistency*

Supplier consistency in the functional decomposition of the SEBA exemplar platform and into specific implementations is desired and encouraged in order to align with suppliers and developers, while not limiting innovation to improve cost, performance or reliability.

The operators, suppliers and developers should proactively collaborate to communicate about implementations and product roadmaps and evolving open software technologies, and to propose the controlled evolution of solutions.

### 3.2.3 Operator Specific Addenda (System Impacts, etc.)

Operators should provide specific addenda here that require special attention to the SEBA project, as derived out of their experience, requirements, or consequences of their SEBA implementations.

### 3.2.4 Key Outstanding Questions

This section will capture any current outstanding questions from the provider, supplier or member groups for ONF and SEBA.

There are currently no open questions.

 

*Write to rdspec@opennetworking.org with comments or questions.*


## End of

## SDN Enabled Broadband Access (SEBA)

## Reference Design