SEBA reality check!
How to take the design to the next level?

Fabian Schneider, Deutsche Telekom
(Head of SDN Control & E2E System Development, Access 4.0 Program)
fabian-schneider@telekom.de

Tom Anschütz, AT&T
(Distinguished Member of Technical Staff)
tom.anschutz@att.com
Goals for SEBA

• Provide a basis for community development and support of broadband access systems.
• Enable/encourage 3rd parties to build offers based on SEBA.
• Integrate easily with existing OSS and NG ONAP management.
  • But also minimize system concerns in upstream systems
• Provide like capabilities across disparate access systems
• Avoid lock-in
SEBA Architecture
AT&T vAccess SEBA/VOLTHA Deployment Model

Cross-Domain Correlation & Automation

Access Domain Manager

Domain Specific Controller

OSS/BSS

ECOMP

VES

Helm charts & SW repo

Helm & K8s

Kafka & VES

K8s Metrics

Config & Status

Common Data Model

NEM Function

NC/Y Server

SEBA Site(s)

Kafka & VES (FM/PM/Logs)

ONOS & Control Apps

ASG

BMC

Status

VLAN Config

K8s

SEBA Services

Compute

AGG

Tiller

DPU(s)

ONT(s)

Gfast

DSC

vOLT(s)

Common Backbone

Akraino Regional Controller

Mgmt Network

• SEBA SW (in blue) deployed by Helm API to K8s containers and SW images
  ○ NEM Function - SEBA NC/Y Server – Mediator to DSC
  ○ NEM Function - YANG Adapters – to SEBA services
  ○ Kafka & VES – Agents for Collection & Reporting
  ○ VOLTHA, vOLT adapters, ONOS, Control Apps, BMC, ASG – SEBA services
Access 4.0 in a NUTSHELL

- Several other talks on what Access 4.0 does ⇒ see next slide
- We build many (~1000) A4-PODs, each with
  - Few servers for management & control plane containers only
  - White-box switches that provide both SE, fabric, and LER functions
  - White-box OLTs that enable Gigabit via FTTH and FTTB
  - Dedicated servers for IPsec, dedicated switch(es) for L2-BSA
- Few central management clusters, that
  - Implement interfaces to the OSS/BSS platform
  - Provide & distribute data to the A4-PODs
  - Collect and analyze logs, metrics, and events
SEBA reality check: Insights from ACCESS 4.0
Key Drivers for the Control Plane Design

- Extensible For Future Access Technologies
- SDN Principles
- Feature Parity with BNG
- Disaggregation
- Vendor Neutral — 2nd, 3rd Vendor
- Las Vegas: „... Stays in the A4-POD“
BIG PICTURE – addressing the Key Drivers

1. Multi Access Tech
2. Disaggregation
3. SDN + Las Vegas
4. Vendor Neutral

Technology Specific Control Stacks

- POD Access Orchestrator (PAO)
- Interfaces/APIs
  - 1/POD
    - OLT Controller
    - DPU Controller
    - DSL Controller
    - SE Controller
    - Fabric Controller
    - LER Controller
  - 1/Element (logical level)
    - OLT/ONT Agent
    - OLT/DPU Agent
    - DSL Agent
    - SE Agent
    - Fabric Agent
    - LER Agent

Hardware (ASICs, FPGAs, SmartNICs, CPUs, etc.)
BIG PICTURE – Imperfections & Orchestration Tasks

Subscriber Management: Port-up/down events, determine subscriber type/product/access technology, select SE, provision profiles and services, user authentication.

Overall topology is here, needs support (e.g. API for LLDP generation/reception) from lower level controllers. Required for HQoS config.

Path & QoS calculation, monitoring and control.

Link utilization, service surveillance, traffic accounting, OAM, diagnostics, debugging.

Single Radius proxy for the whole POD.

POD Access Orchestrator (PAO)

Policy & AAA (Radius)

Interfaces/APIs

1/POD

VOLTHA & ONOS

DSL Controller

SE Controller

Fabric Controller

LER Controller

1/Element (logical level)

OLT/ONT Agent

OLT/DPU Agent

DSL Agent

Hardware (ASICs, FPGAs, SmartNICs, CPUs, etc.)
Access 4.0 & SEBA Exemplar Implementation

Perfect match for Dataplane

A4 central: majority of mgmt tasks

A4 POD: CP & MP separated
What does this mean with respect to SEBA?

• Building 1000 A4-PODs requires lean and easy to operate building blocks
  ➢ Micro-service based components need to work in generic container management systems
  ➢ Components should not require complex helper services (such as Kafka)

• Allow different sources (in-house dev, open-source, proprietary) for different parts of the control plane
  ➢ SEBA must allow for more plug and play LEGO-like system building
  ➢ Leverage common internal control plane interfaces to bring in 2nd & 3rd vendors

• Harmonize those different sources through interfaces (APIs) which abstract functionality
  ➢ DT plans to bring these interfaces into the public, once we have a running example.
  ➢ Happy to improve these interfaces with and move their evolution into the ONF community

• The POD access orchestration component will be very operator/deployment/scenario dependent
  ➢ DT decided to co-develop the POD access orchestrator in-house
  ➢ Current focus on production readiness of technology specific controllers such as VOLTHA
Actively Bringing our Experience to SEBA Acceleration

Access4.0 mapped to 2019/2020
SEBA common core

SEBA RD1.0

Carrier Automation Platform (CAP)

Network Edge Mediator (NEM) / Edge Cloud Orch.

SEBA Site(s)

ODS/ODS

SEBA Scope

SEBA RD1.0

Network Edge Mediator (NEM) / Edge Cloud Orch.

SDM Controller

vOLT Adapter(s)

OLT(s)

ONT(s)

Gfast

DPU(s)

Fabric

Compute

RedFish Adapter

Config & Status

Common Data Model

POD API

POD Actor

POD DB

Kafka & Agents (FM/PM/Logs)

VOLTHA

ONOS

Fabric-C

SE-C

SEBA Access Control/Orchestrator

Link

LIF3/3rd pty

LER/LSR

CP

MP

LIFE IS FOR SHARING.

ACCESS 4.0

mapped to 2019/2020
SEBA common core
SEBA reality check: Insights from
Pain Points with the Current State

• There’s a strong desire to allow SEBA to run on open access hardware
  • So there’s also a desire to reduce complexity, footprint.
• There’s a need to adapt SEBA to existing large scale systems
• Economics need to be on-par with legacy solutions
• There are more opportunities for smaller scale than larger scale
• ONOS Apps are limited by OF protocol capabilities
SEBA possibilities

VOLTHA is by far the most valuable component in SEBA. It is also built on a robust and reliable platform. So focus on VOLTHA – make it indispensable to a variety of uses.

- Add capabilities for more silicon and devices
- Enable small systems with local APIs
- Use performant software languages
- Redefine what makes a MVP!

Let’s look at a similar example in an adjacent industry...
Adjacent Market Example (Original)
Adjacent Market Example (SAI)

Management Interfaces (Netconf, CLI, etc.)

NOS
Other Ctl
Ctl Plane Stacks

Switch Abstraction Interface

SDK
Driver
SDK
Driver
SDK
Driver

Switch ASIC 1
Switch ASIC 2
Switch ASIC 3

After SAI
Adjacent Market Example (SONiC)

SONiC

Management Interfaces
(Netconf, CLI, etc.)

NOS

Control Modules
State Service

Switch Abstraction Interface

SDK
Driver

Switch ASIC 1
Switch ASIC 2
Switch ASIC 3

After SONiC
So Let’s Envision

VOLTHA

Management Interfaces
(Netconf, CLI, etc.)

Control Modules

State Service

NOS

Access Abstraction Interface

SDK

Driver

Access ASIC 1

Access ASIC 2

Access FPGA 3
So Let’s Envision

This is what’s new...
Allow “ONOS Apps” to run directly against VOLTHA APIs
-or-
Allow for Remote/Central Control
In Summary

Focus on VOLTHA

• Adjust VOLTHA architecture to be more self-reliant
• Allow for extensibility, but don’t force it
• Enable both local and central functions
Thank You
Deutsche Telekom @ ONF Connect 2019
depth dives on Access4.0 and much more

Jochen Appel, 10:00Am
Keynote: Access Network Transformation

Dr. Hans-Joerg Kolbe, 11:00Am
Inside View Into Operator Business Cases

Manuel Paul, 4:30pm
COMAC and OMEC at DT

Bjoern Nagel, 2:15pm
VOLTHA Roadmap

Dr. Hans-Joerg Kolbe, 2:30pm
Implementing the Programmable Service Edge

Dr. Fabian Schneider, 5:30pm
SEBA Reality Check! How to Take the Design to the Next Level?

Robert Soukup, 4:30pm
Access4.0 Program update

Manuel Paul, 11:55Am
Panel: Technical Leadership Team (TLT)