IMPLEMENTING A PROGRAMMABLE SERVICE EDGE

ONF CONNECT 2018

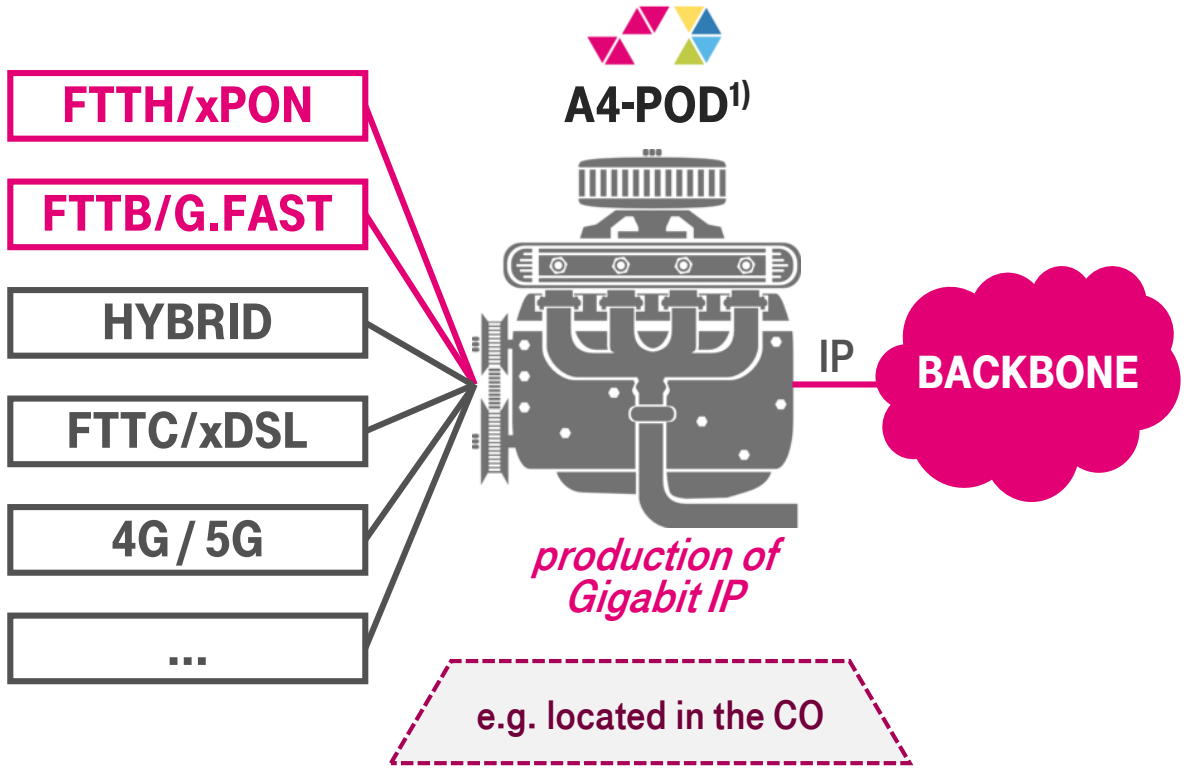
H.J. Kolbe and L. Nobach, Deutsche Telekom

LIFE IS FOR SHARING.

ACCESS 4.0 MISSION STATEMENT

RECAP FROM 2017: ACCESS 4.0 IS THE DESIGN OF A NEW ACCESS PLATFORM WITH TIGHT COUPLING TO A COST MODEL

GIGABIT READY TECHNOLOGY



COST MODEL

- Cost model developed **from day one**
- **Strong interworking** between architecture / design and Cost modelling
- Comprehensive application of **Design-to-cost** methods

1) Access 4.0 Point of delivery: production of broadband IP

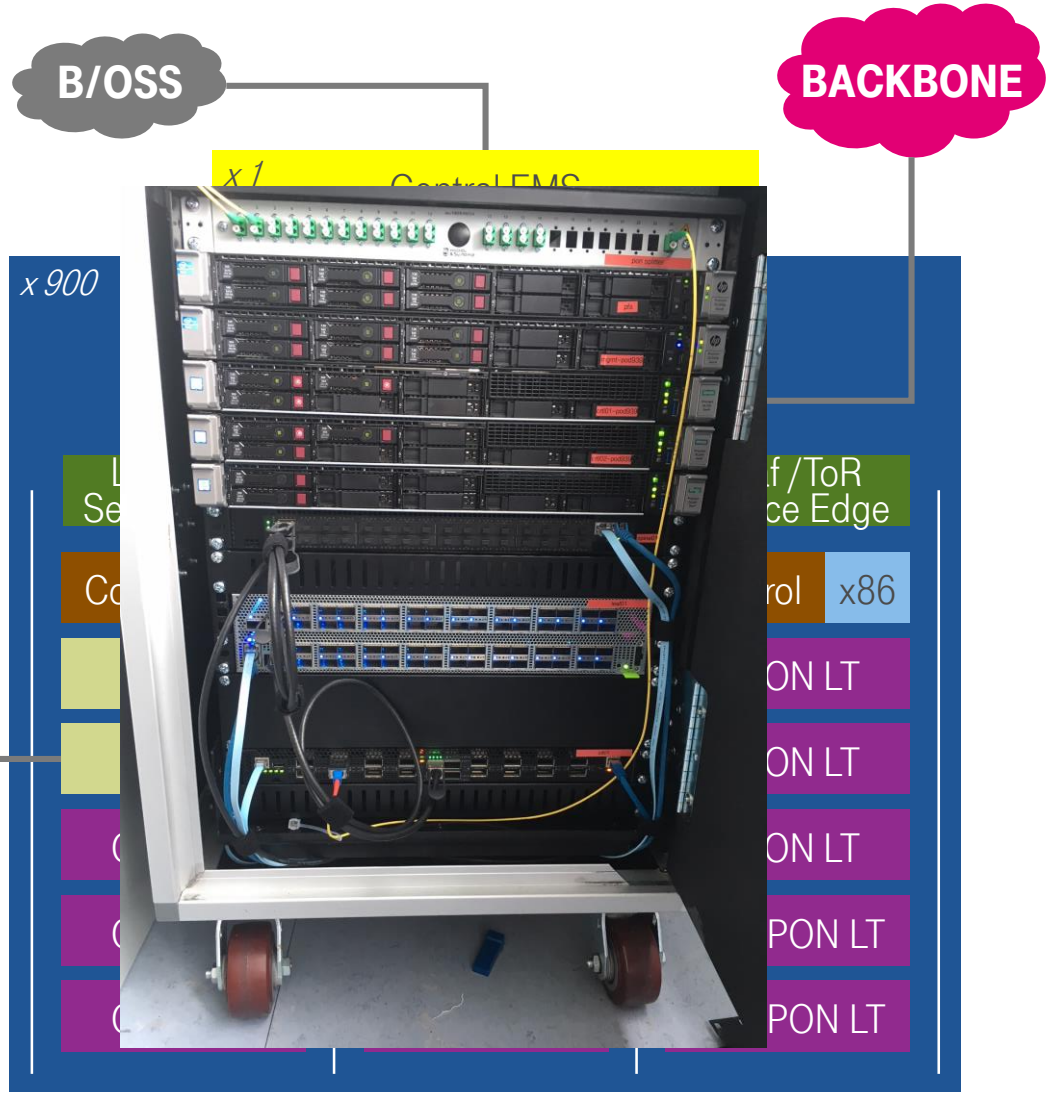
INTRO
WHAT IS ACCESS 4.0 / A4?

SCOPE OF ACCESS 4.0

- Extensive **D2C** project for **FTTH/B**
- Everything monitored in a **comprehensive Cost Model**
- Design and engineering using **bare metal / OCP hardware**, lots of **open-source software** as well as **merchant silicon**
- Application of **data center** principles, leaf/spine fabric, CI/CD, ...
- **Clean IT architecture** (Las Vegas principle)

OBJECTIVES OF ACCESS 4.0

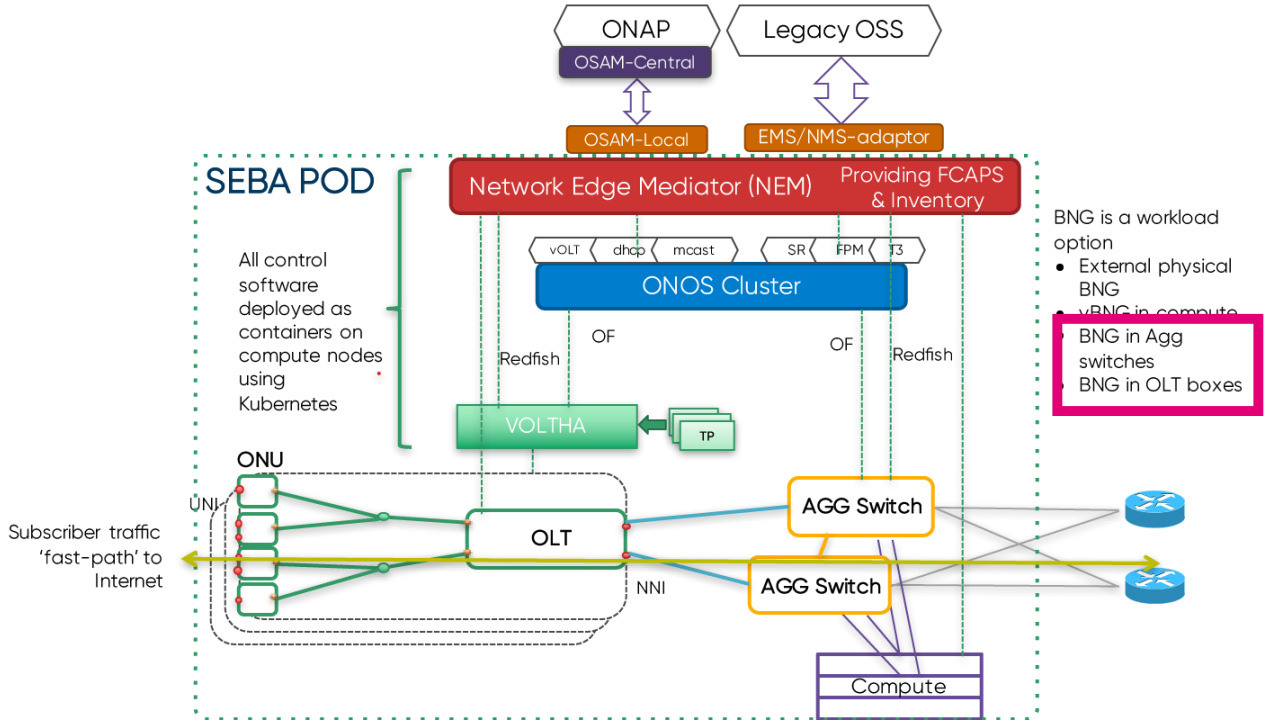
- **Save** CapEx and OpEx
- **Reduce vendor lock**; bring in new players
- Drive **automation**
- **Time-to-Market for services** (keep business logic SW in house)
- Increase **flexibility** for capacity mgmt, change-over, migration, ...



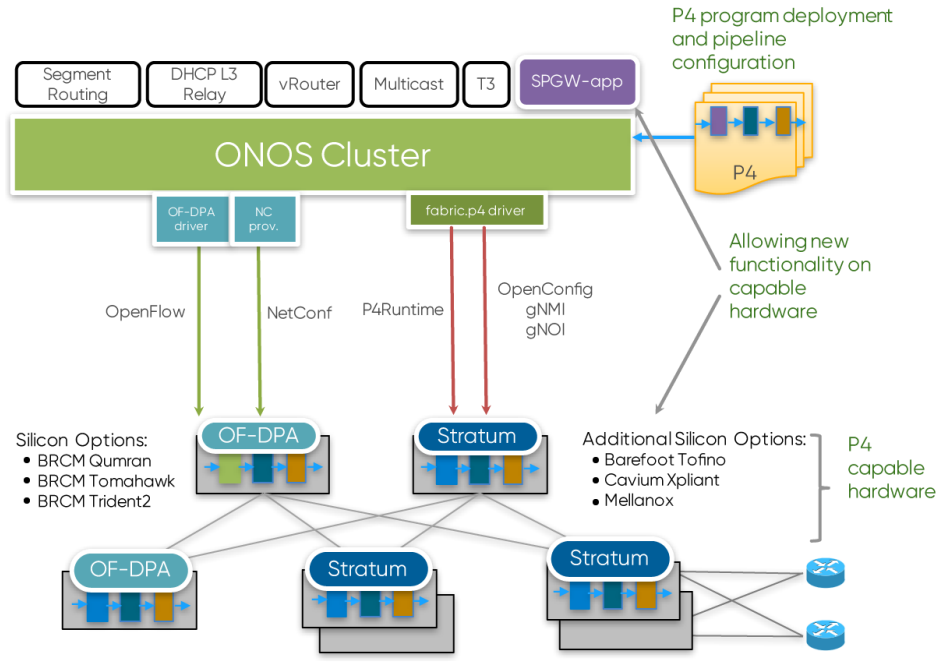
A4-RELATED WORK AT THE ONF

RELEVANT STREAMS

SEBA: SDN ENABLED BROADBAND ACCESS

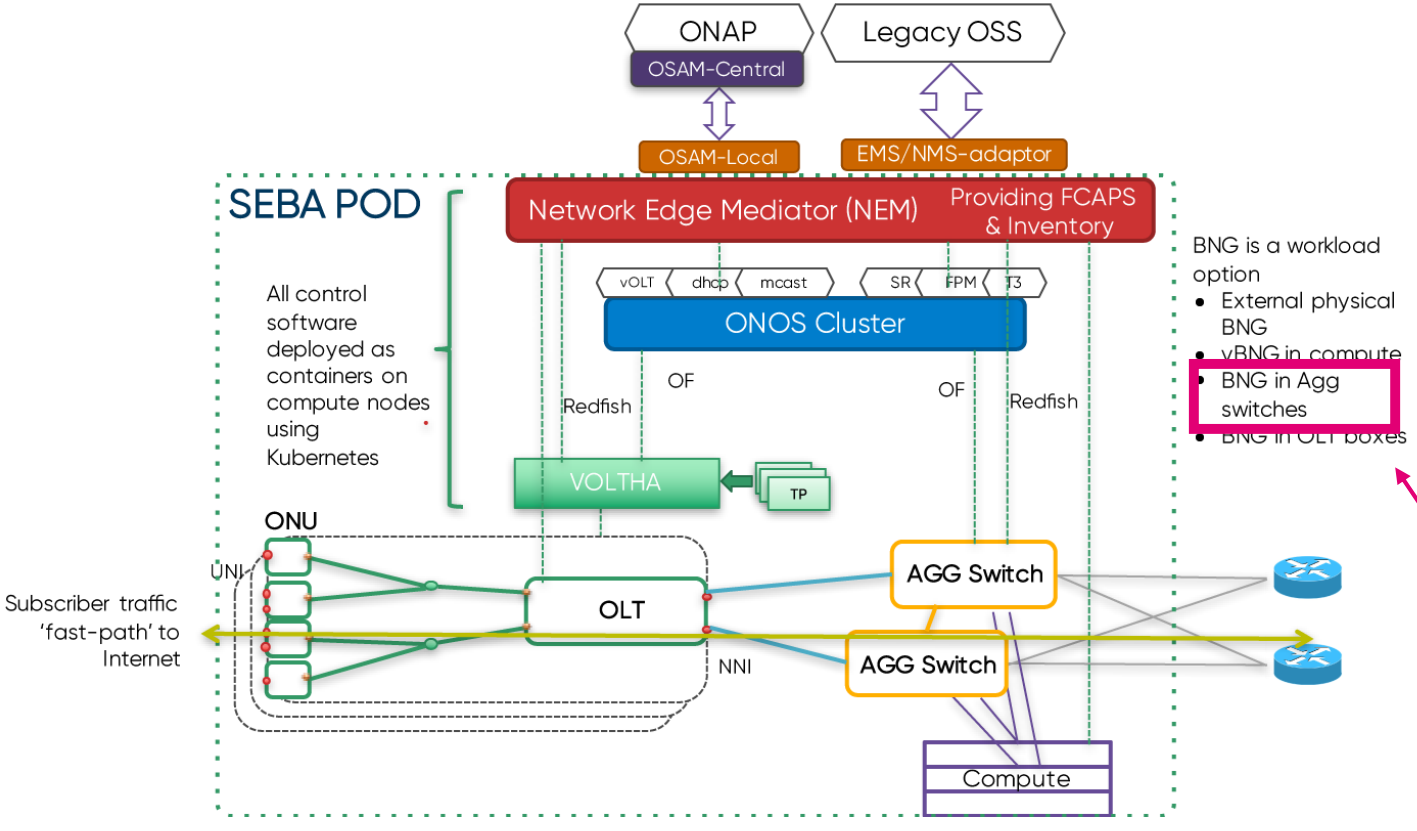


UPAN: UNIFIED, PROGRAMMABLE & AUTOMATED NETWORK



REFERENCE DESIGNS & EXEMPLAR IMPLEMENTATIONS
OPERATOR-DRIVEN, CONSENSUS AMONGST OPERATORS

CORE TECHNOLOGIES IN SEBA AND ACCESS 4.0

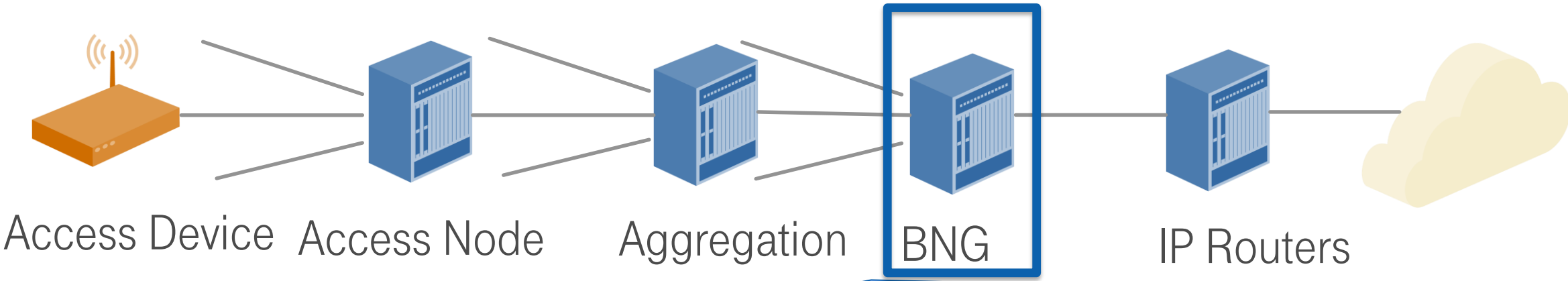


- SDN Control & Apps (incl. DT PPA)
- Open Hardware
  - OLT, Switch, DPU, ...
- Automation, CI/CD
- Network Management & IT Abstraction
  - "Las Vegas Principle" at DT
- Service Edge & Router (BNG)

KEY COMPONENT: BNG

BROADBAND NETWORK GATEWAY

BROADBAND NETWORK GATEWAY



Terminates **subscriber tunnels** and sessions (PPPoE / IPoE o L2)
Applies per subscriber **services**
Serves as full PE router

EVOLUTION OF BNG

BRAS

Subscriber Termination
Per Subscriber Services

BNG

+ PE Router
+ Single Service Creation Point

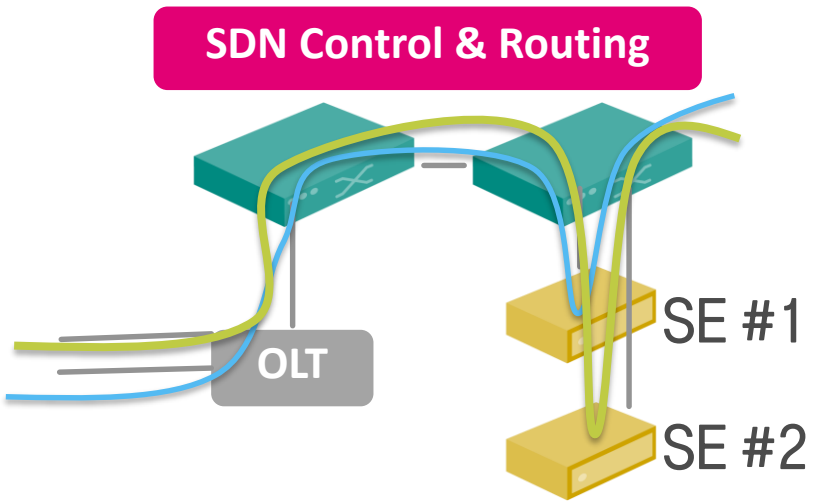
vBNG

+ virtualized
+ runs in DCs
+ CP/UP split

Growing Complexity. Growing Throughput

GROWING COMPLEXITY?? GROWING THROUGHPUT ??

- Complexity Issue
  - *"Single service creation point"*



- SDN-Traffic steering to service edge.
- IP Routing in fabric

- Performance Issue
  - *"x86 user plane performance through NFVI… well…"*

SE Control Plane → x86

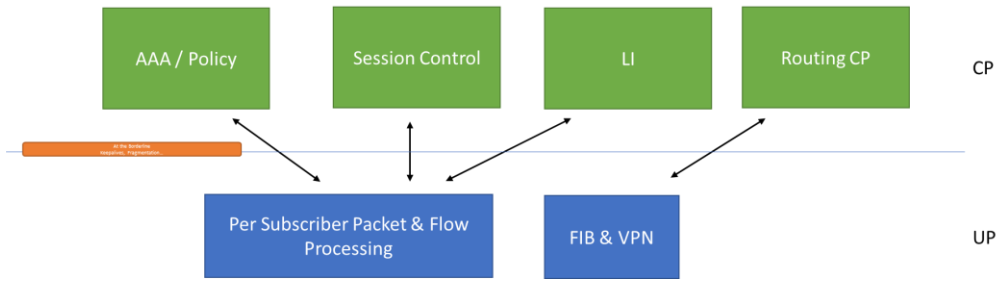
SE User Plane → ASIC

- Offload user plane to PNF
  - Cost and power per bit/s …..
  - Quite some #ASICs on the way to reach a VNF…

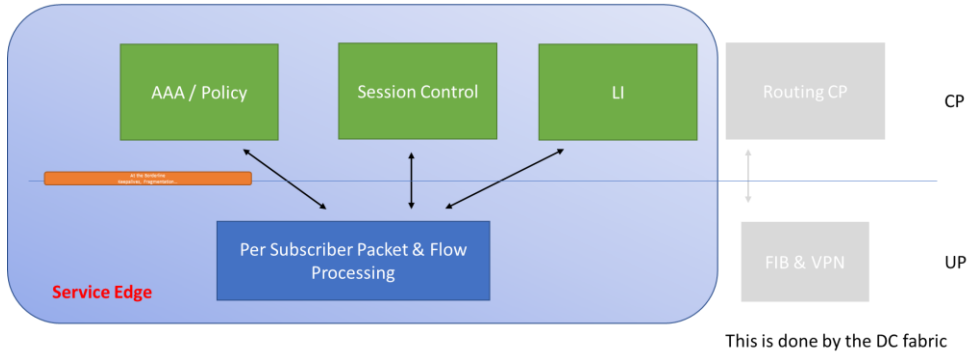
OFFLOADING THE BNG USER PLANE WORK LOAD

A BNG / SE ON MERCHANT SILICON: AT DT, SOON AT ONF

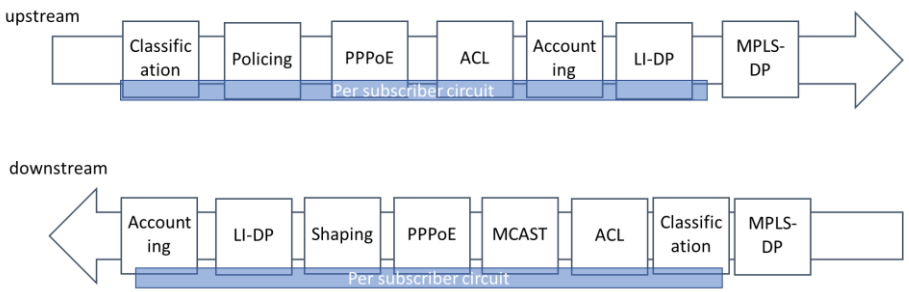
Split routing, CP, UP, near UP, cups



Take routing out, a new baby is born: the SE



The user plane: what is in there??



Project: Develop core protoype on merchant silicon: done

Deutsche Telekom, TU Darmstadt (pioneering work)
rtbrick (productization)

BNG/SE DATA PLANE ON MERCHANT SILICON? WORKS!

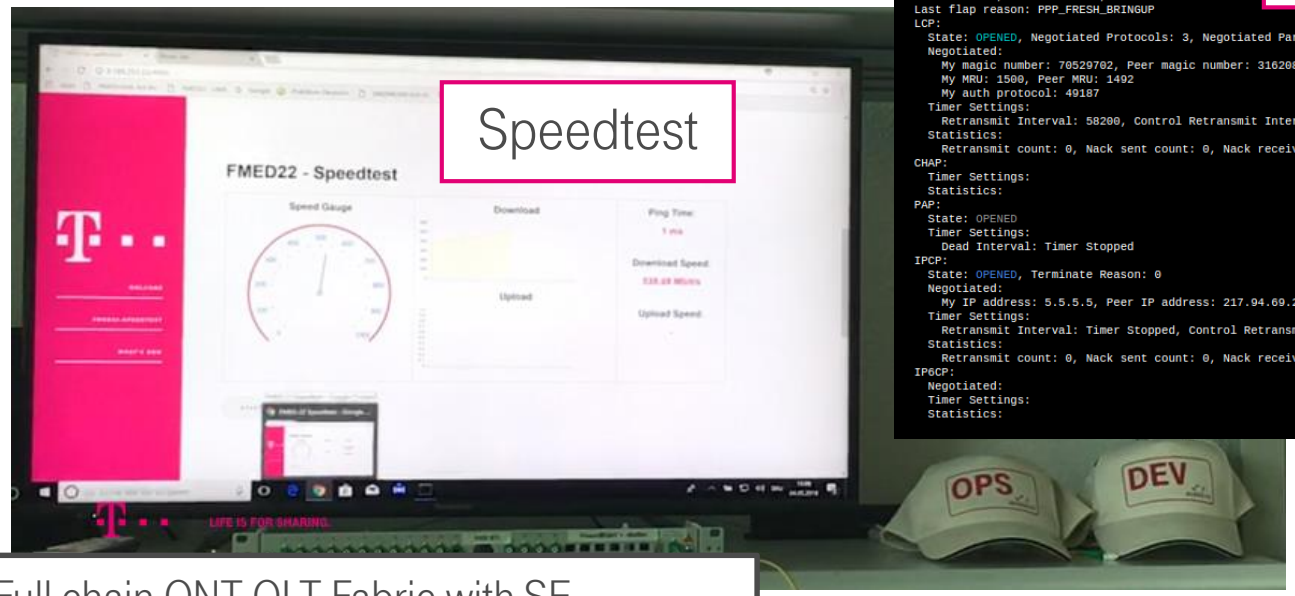
```
/*
 * D-Nets 6 P4 Service Edge
 * Authors: Jeremias Blendin, Leonhard Nobach
 * Copyright 2017-present Deutsche Telekom AG
 */


#define PPPOE_PROTO_DISCOVERY 0
#define PPPOE_PROTO_SESSION 1


header_type pppoe_md_t {
    fields {
        ppp_proto : 1;
        pad_1: 7;
        protocol : 16; // PPP protocol field
        totalLength : 16; // PPP lengths field
        mru : 16; // PPP maximum receive unit (RFC 4638)
        mru_check : 16 (saturating);
    }
}

header_type pppoe_t {
    fields {
        version : 4;
        typeID : 4;
        code : 8;
        sessionID : 16;
        totalLength : 16;
    }
}

header_type pppoes_protocol_t {
    fields {
        protocol : 16;
        /*
         * See http://www.iana.org/assignments/ppp-numbers/ppp-numbers.xhtml
         * Dataplane: IP:  0021, IPv6: 0057,
         * ControlPlane: LCP: c021, IPv6CP: 8057, IPCP:
         */
    }
}

header pppoe_t pppoe;
header pppoes_protocol_t pppoes_protocol;
```

P4 prototype: supporting PPP, L2TP work in progress

https://github.com/opencord/p4se/

Work is not at all limited to P4-programmable devices

- Works on two types of chipsets
- Productizing SE now



CLI

```
ubuntu@leaf1:~$ rtb accessd show ppp session detail

User: cpe1@a4.de, Session ID: 1, Client MAC address:
  Server MAC address: 00:1e:5e:00:00:1c, Service Name
  Ordinal: 104, Inner VLAN: 7, Outer VLAN: 128
  Last flap reason: PPP_FRESH_BRINGUP
  LCP:
    State: OPENED, Negotiated Protocols: 3, Negotiated Parameters: 42, Terminate Reason: 0
    Negotiated:
      My magic number: 70529702, Peer magic number: 31620830
      My MRU: 1500, Peer MRU: 1492
      My auth protocol: 49187
    Timer Settings:
      Retransmit Interval: 58200, Control Retransmit Interval: 5000, Echo Interval: 60000
    Statistics:
      Retransmit count: 0, Nack sent count: 0, Nack received count: 0
  CHAP:
    Timer Settings:
    Statistics:
  PAP:
    State: OPENED
    Timer Settings:
      Dead Interval: Timer Stopped
  IPCP:
    State: OPENED, Terminate Reason: 0
    Negotiated:
      My IP address: 5.5.5.5, Peer IP address: 217.94.69.251
    Timer Settings:
      Retransmit Interval: Timer Stopped, Control Retransmit Interval: 5000
    Statistics:
      Retransmit count: 0, Nack sent count: 0, Nack received count: 0
  IP6CP:
    Negotiated:
    Timer Settings:
    Statistics:
```

Prototype: Full chain ONT-OLT-Fabric with SE
Bare metal hardware + VOLTHA + our code + BNG CP

CHALLENGES
COMMON BASELINE AT ONF (?)

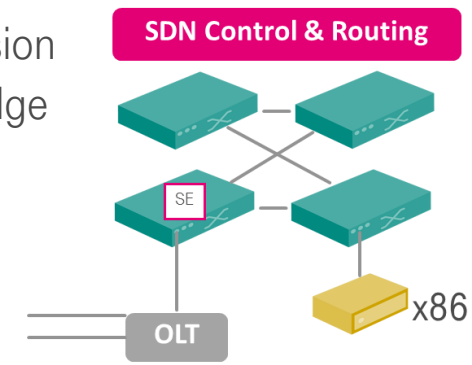
Data Path

- Tunnel termination (VLAN/SR)
- Session termination (PPP, IPoE) incl. OAM
- Tunnel switching (L2TP LAC)
- ACL per subscriber
- Accounting based on ACL
- Legal Intercept
- H-QoS per sub
- Multicast
- Uplink encap/decap to MPLS-based fabric

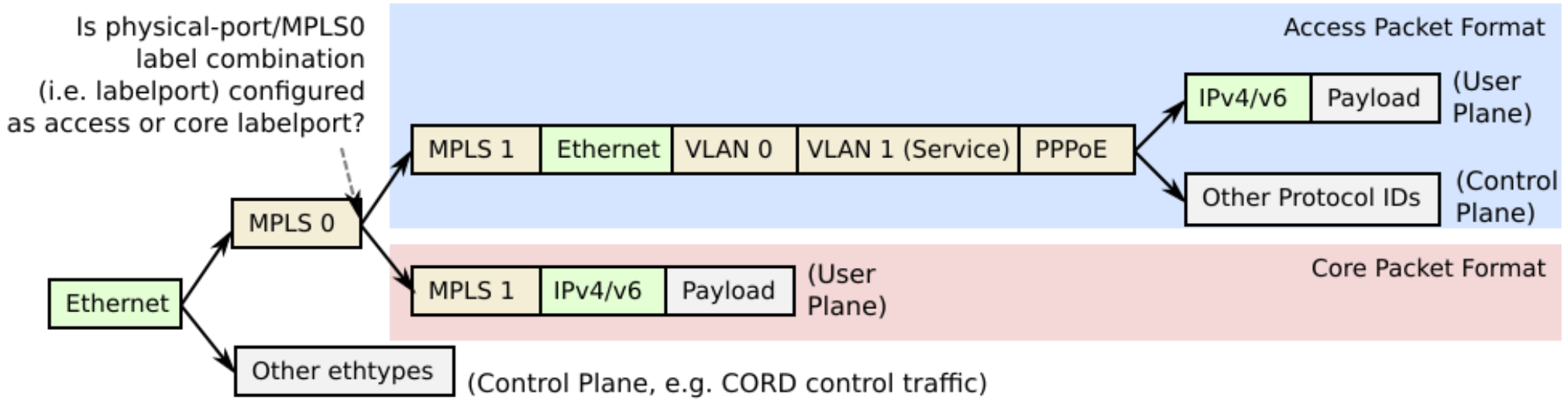
- + operator-specific functions

Deployment Options and Dependencies

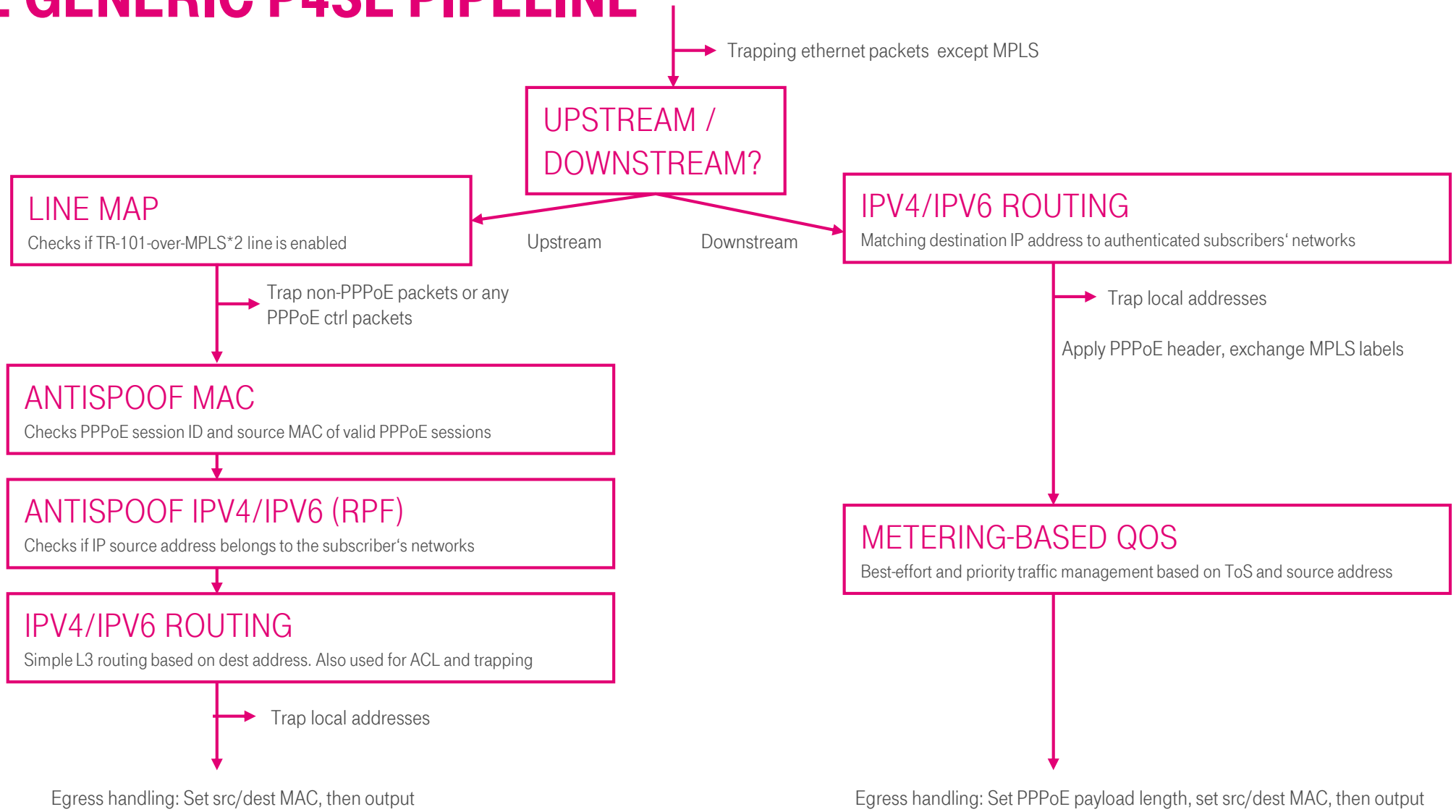
- The order of processing does matter
- Shapers and accounting closely related
- Dual-chip vs. single chip
- Distributed approach
  - E.g. separate QoS from session termination to allow multi-edge
  - Involve OLT (?)



P4SE PIPELINE PACKET FORMAT



THE GENERIC P4SE PIPELINE

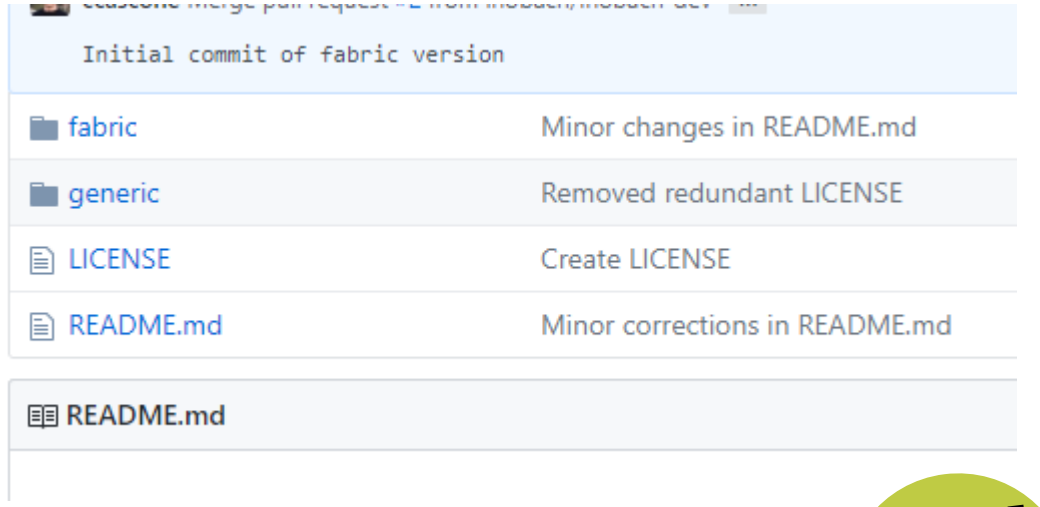


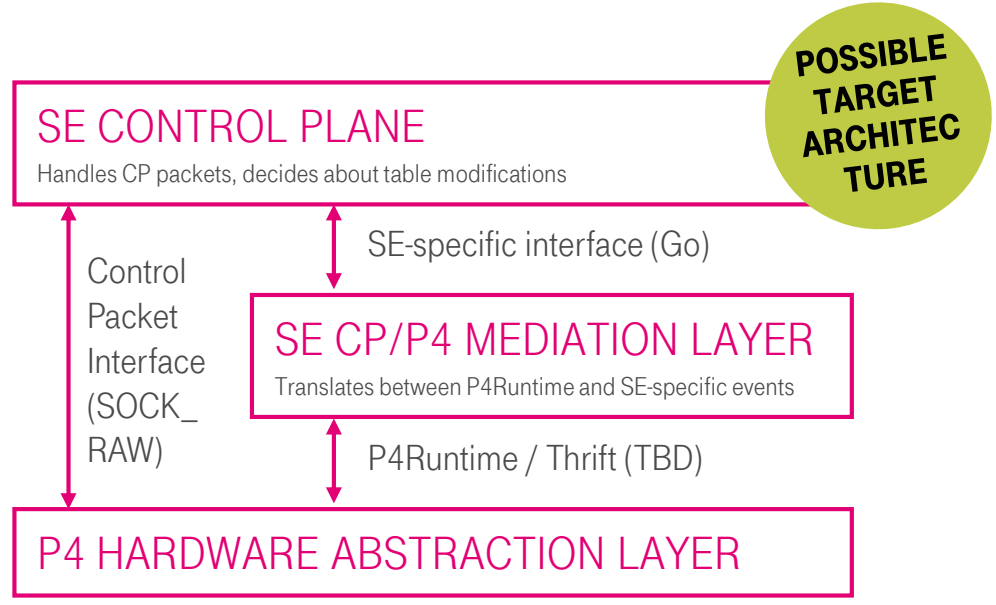
P4SE USER PLANE PIPELINE ON GITHUB https://github.com/opencord/p4se

- Two variants, **generic** and **fabric**
  - **generic**: Stand-alone SE, e.g. attached to separate fabric leaf switch
  - **fabric**: a CORD-like leaf switch with integrated SE functionality

- **Control plane** (CP) currently **missing** (except UP test stubs)
  - Idea to implement the CP in golang
  - Two-tiered architecture: use-case-specific mediation layer
  - CP uses P4Runtime or Thrift for UP communication

We are actively seeking for feedback and interest in the community!





THE BIG PICTURE

TOWARDS A COMMON SUBSCRIBER EDGE

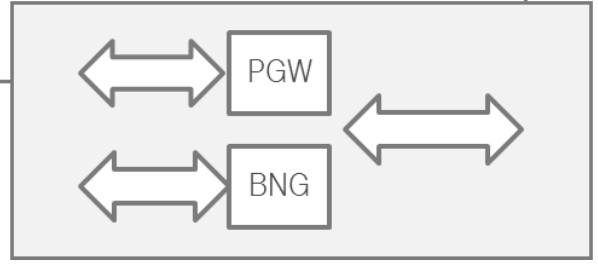
STRUCTURAL CONVERGENCE

FUNCTIONAL CONVERGENCE

Location consolidation

Traffic grooming, local coupling

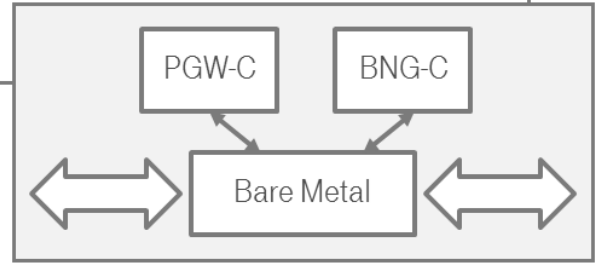
- H-CORD (hybrid)
- Edge Cloud
- Low Latency



User Plane consolidation

Same data path for fixed and mobile user plane (UP)

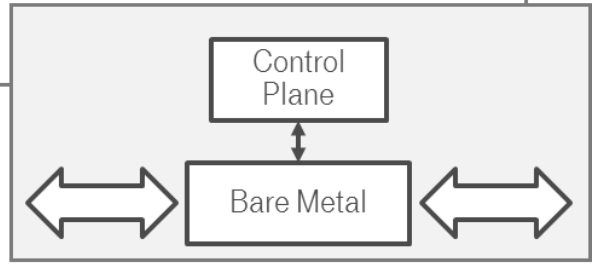
- Bare Metal-based UP,
- virtualized CP



Control Plane consolidation

converged control plane

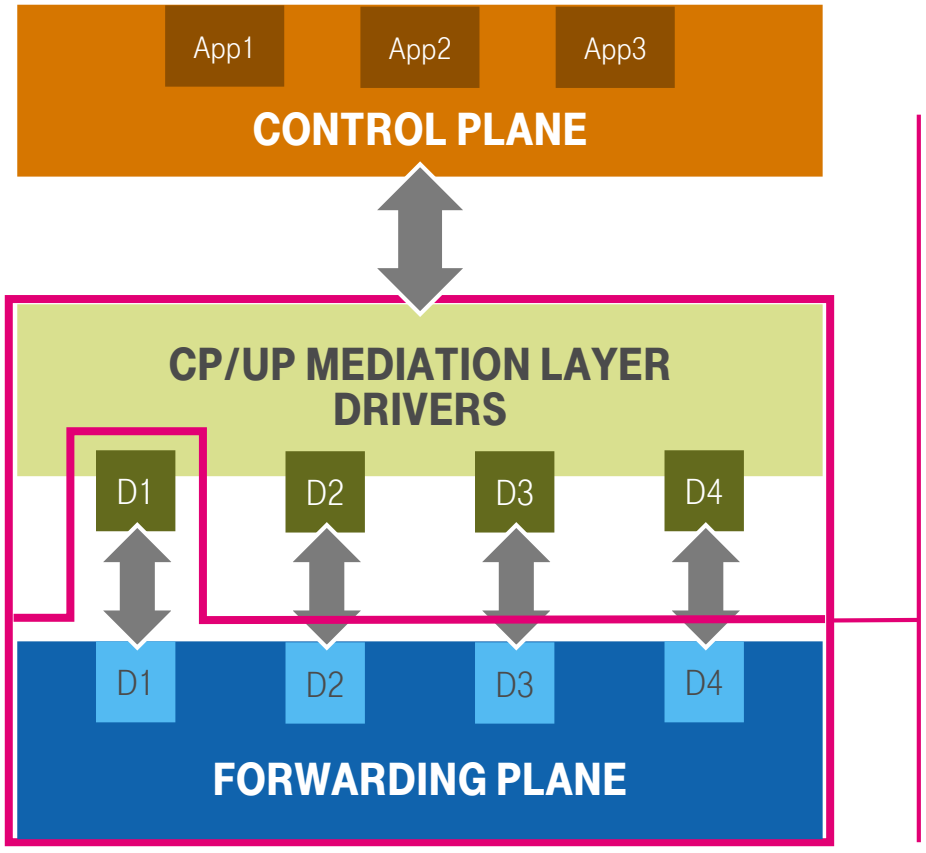
- includes slicing
- Following 3GPP+BBF work



DESIGN PARADIGM: CONTROL PLANE / USER PLANE SPLIT

NO FRAGMENTATION, PLEASE...

COMMUNITY WORK



Fixed / mobile or converged control applications

(usually on x86)

Platform Compatibility Framework with standard set of APIs

- avoids hardware lock-in
- provides compatibility to apps/features through common protocol and data model for forwarding
- provides compatibility of management tools and practices

Anything south of the line to be provided by hardware vendor

Programmable hardware on bare metal

(Differentiate through performance & exposed feature sets)

IMPLEMENTING ~~THE~~ A PROGRAMMABLE SERVICE EDGE

At Deutsche Telekom

ACCESS 4.0

- SE running in first field trial
- Developing a product grade solution (accounting, L2TP, QoS,...)
- Move in stages into the field
- Defining APIs – stay tuned.
- Fully decouple hardware from software
- Keep space for competition on silicon level

In the Community

- Published P4 code at DT site as well as ONF
- Provided Deployment Options to ONF
- Integration to ONF (SEBA / UPAN) as Reference Implementation (?)
- Define minimum requirements at ONF SEBA
- Go beyond “just” subscriber termination
- Work on APIs between CP and UP for real disaggregation (once the “homework” is done)

Proposal: Reference Implementation at ONF since there is no „one sitze fits all“ + agree on APIs

FIN