OPEN SOURCE EPC
OPERATORS’ JOURNEY TOWARDS THE CLOUD NATIVE ONF-BASED TELCO CORE

Michal Sewera
T-Mobile PL
KEY MESSAGE: IT’S ALL ABOUT DISRUPTION

- New technology as a main source of disruption
  - Radical change instead of “small adjustments”
  - Significant increase of efficiency

- Examples
  - Transistor
  - Integrated circuits
  - Linux OS
WHY OPEN SOURCE?

- New business model - it’s about transparency of the software deliverables
- Telco service based architecture with higher flexibility and programmability (REST APIs) in the area of integration with 3rd party systems (i.e. network exposure functions)
- Cloud native ecosystem with micro service based architecture is already providing more transparent solutions exposing open source based components (i.e. istio service mesh) – this is true even for commercially available EPC solutions
- Operators perspective – gradual transition from RFQ/SLA mode of action towards full DevOps with build, run, release responsibilities
JOURNEY TOWARDS THE CLOUD NATIVE
TOWARD TELCO GRADE WEBSCALE FRAMEWORKS

12 Factors – properties of the cloud native apps*

- **Codebase** – use one codebase, even when building cross-platform apps
- **Dependencies** – explicitly declare and isolate all dependencies
- **Configuration** – don’t store config as constants in code
- **Backing services** – loosely-coupled resources attached to the app
- **Build, release, run** – strictly separate build and run stages
- **Processes** – execute the app as one or more stateless processes
- **Port binding** – use port binding to export services
- **Concurrency** – scale out apps horizontally, not vertically
- **Disposability** – use fast start-ups and graceful shutdowns
- **Dev/Prod parity** – facilitate continuous deployment
- **Logs** – treat logs as event streams
- **Admin processes** – run admin tasks as one-off processes from a machine in the production environment

*https://12factor.net/

Source: 5G-PPP Software Network Working Group (From Webscale to Telco, the cloud native journey)
OPEN SOURCE EPC – CASE STUDY
HIGH LEVEL STRATEGY & MAIN ASSUMPTIONS

- Fundamental shift towards Open Source based systems
- Initial Focus on FMS/FMC use case (high throughput but minimal set of EPC features)
- Bare-metal based distributed architecture (with DPDK) to simplify design and minimize cost per Gbps
- ONF based framework (based on OMEC and COMAC reference design)
TMPL FMS EPC USE CASE

• **Key business assumptions:**
  - „Start-small“ approach - focus on Fixed Mobile Substitution service only
  - Limited to 3GPP Gateway with distributed architecture
  - Minimum Viable Product approach (only mandatory features required to go for production)

• **Technical considerations:**
  - Initially no virtualization required (bare metal approach with max efficiency for user plane handling)
  - DPDK native application, but with support of standard Linux OS networking mechanisms (routing/switching/monitoring)
  - Critical business features:
    • Bandwidth cut to 20/60 Mbps per user (derived from subscription)
    • Lawful interception (required by law)
    • Simple billing (for data retention, required by law)
    • 2G/3G/4G support (initially only 4G for MVP POC)
FOCUS ON EPC GATEWAY (SGW/PGW)

Solution architecture:

- FMS traffic can be routed to relevant EPC Gateway (based on APN and subscriber charging characteristics profile – using APN resolution extension mechanism)
- Initially – only EPC Gateway will be used based on COMAC reference design

OMEC – (COMAC RD Phase-1)  
https://www.opennetworking.org/omec/
KEY TECHNICAL REQUIREMENTS

Interfaces to be implemented:
- S1-U
- S11
- S5/S8
- Sgi
- X1/X2/X3 (Lawful interface)
- GTPP (Offline billing)
- Gn (in case of implementing 2G/3G)

Generic functions
- Multiple MME support – ability to define MME IP range (CP config)
- CG-NAT (Optional support)
- GTP-C Echo sequence number
- DNS in PCO
- DHCP function (IP pool for APN configuration)
- Offline billing & LI

Packet routing and forwarding functions:
- MTU management (limitation)
- Support for IP packets fragmentation
- Static IP routing
- Dynamic IP routing
- User plane Ipv6 support (not in scope of MVP POC)

EPC procedures
- Attach / Detach
- Tracking Area Update / Routing Area Update
- Service Request
- S1 release
- Subscribed QoS Modification

Mobility management functions
- S/PGW overload control (signalling storm, restoration procedures)
- UL and DL rate enforcement based on APN-AMBR
- ARP/APN-AMBR/QCI for default bearer

CLI - Zero outage configuration management:
- configuration should be changed w/o stopping the S-PGW service e.g. MME IP range, APN Pool, ...
CONTINUOUS INTEGRATION / CONTINUOUS DELIVERY
FOCUS ON QUALITY IN BASICS

- Initially it is not about quantity, but **quality**!
- Basic features related with low level EPC core procedures must work perfectly – only if this is fulfilled You can start thinking about new features
- Operator perspective: platform lifecycle management aspects (integration with 3rd party systems, configuration, change management and troubleshooting)
- Recent work in the area of quality was addressing:
  - UE IP allocation
  - IP fragmentation and reassembly
  - CP & DP CLI
  - Billing & LI
ONF based multi-cloud environment with SPGW-C in Warsaw and user plane in Barcelona demonstrated @ MWC

Kubernetes based cloud native environment
OPEN SOURCE EPC - NEXT STEPS

- Thing Big, start small, scale fast...
  - Quality is King!
  - Initial focus: to prove, that basic features works stable with high quality

MVP for FMS use case – key features, billing & LI

Field trial with quality and reliability verification

Adding new features (potentially 2G/3G support to address IRATs)
THANK YOU