Open Mobile Evolved Core (OMEC)

Christian Maciocco

Principal Engineer, Director of Telecom Systems Research – Intel Labs

CORD Workshop @ Big Communication Event – May 6th, 2019 – Denver, CO

Intel Labs

Legal Disclaimer

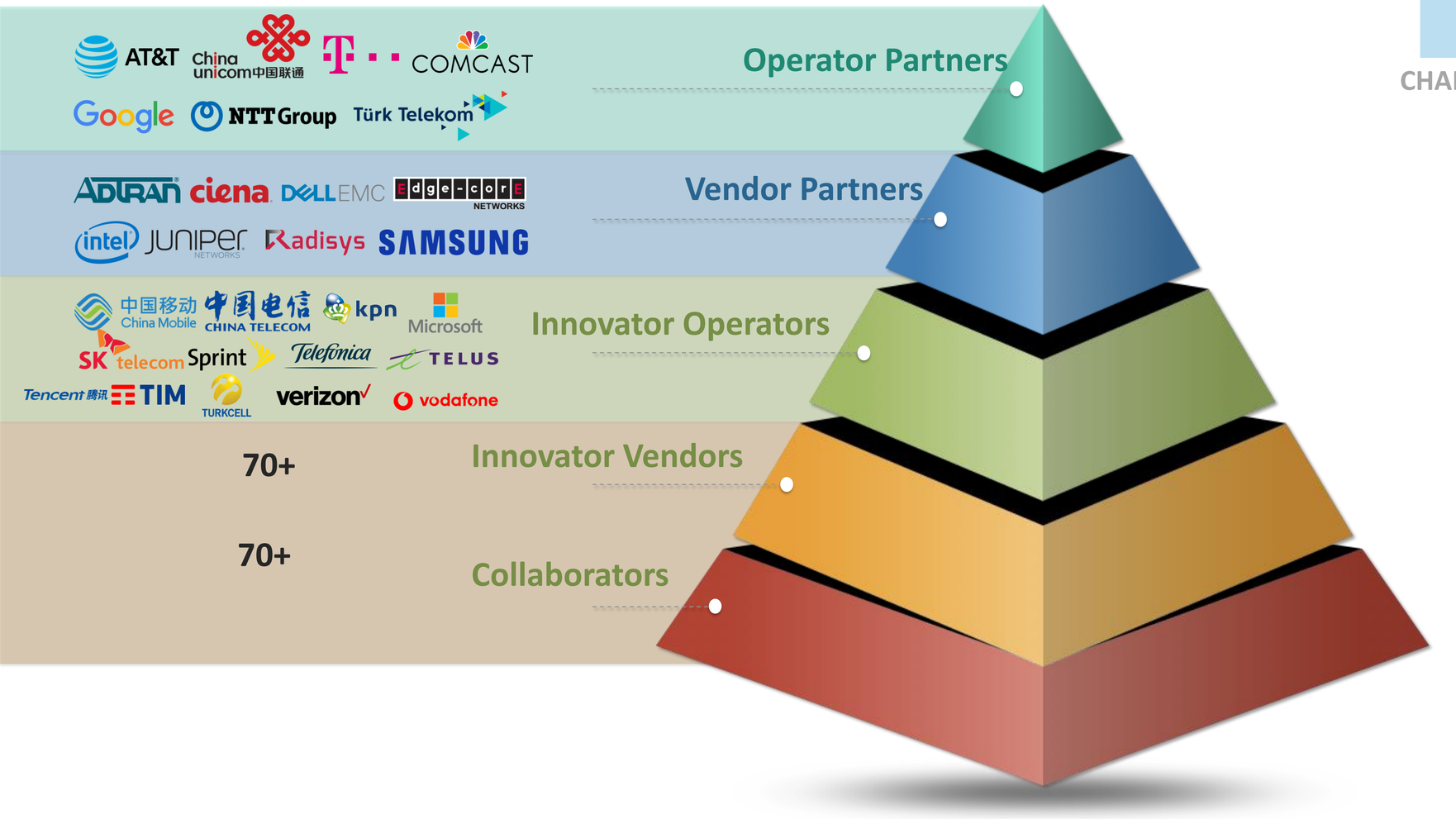
- This presentation contains the general insights and opinions of Intel Corporation (“Intel”). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Use at your own risk! Intel makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. Intel accepts no duty to update this presentation based on more current information. Intel is not liable for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of the information in this presentation.
- Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
- Intel, the Intel logo and Xeon are trademarks of Intel Corporation in the United States and other countries.
- *Other names and brands may be claimed as the property of others.
-

Agenda

- Open Networking Forum (ONF) Background
- Open Mobile Evolved Core (OMEC)
  - History, features, deployment options (VMs, containers, …)
- OMEC in ONF
- Summary / Next Steps

The ONF Ecosystem – 160+ Members Strong

Vibrant Operator Led Consortium Positioned for Success

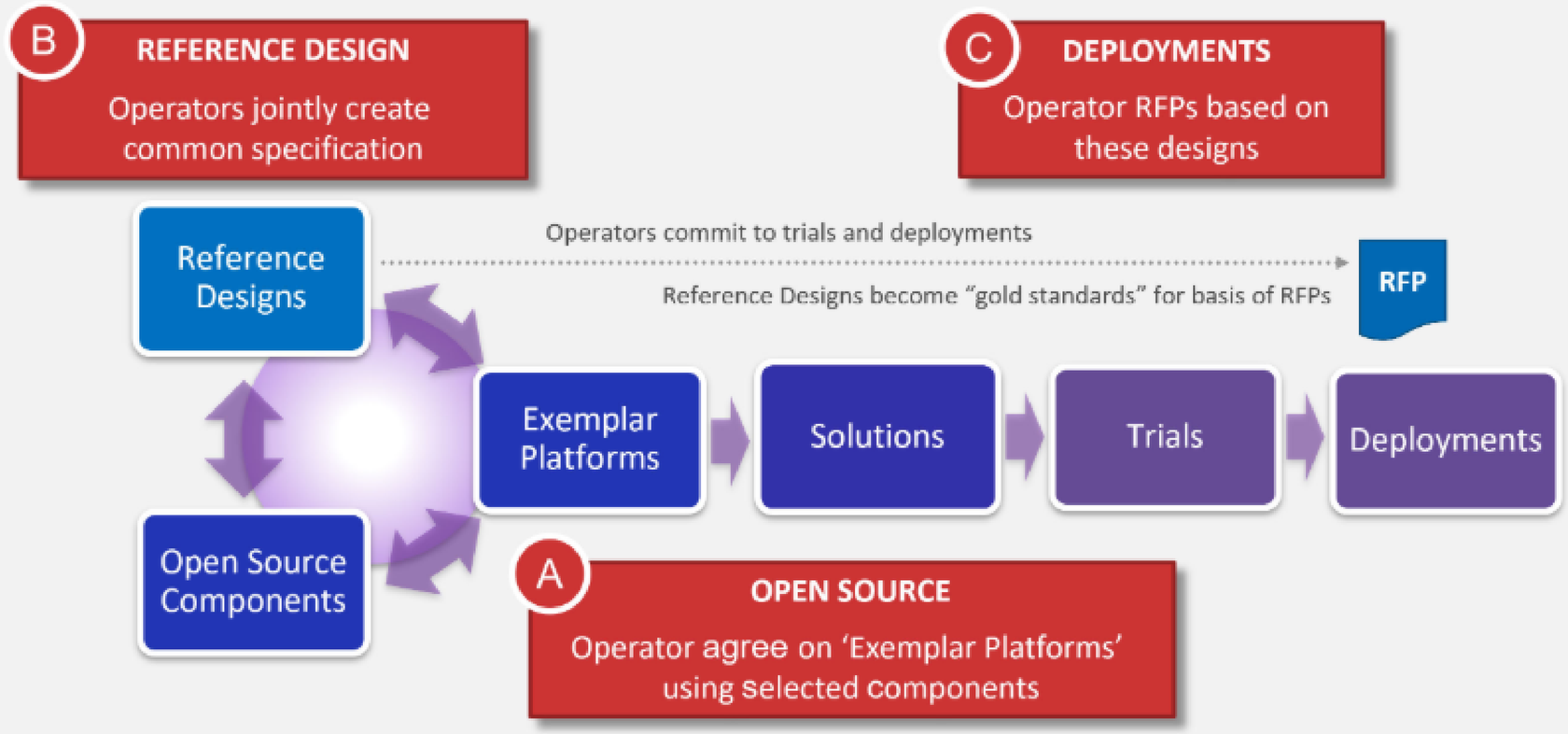


ONF BOARD	E-SAB* Board
CHAIR: AT&T	Intel
T-Mobile	Radisys
Google	Samsung
NTT Group	Dell EMC
Comcast	Ciena
China Unicom	Juniper Networks
Telefónica	ADTRAN
Türk Telekom	Edge-corE Networks
Stanford	Infosys
ONF, Stanford	Tech Mahindra

E-SAB*: Executive Supplier Advisory Board

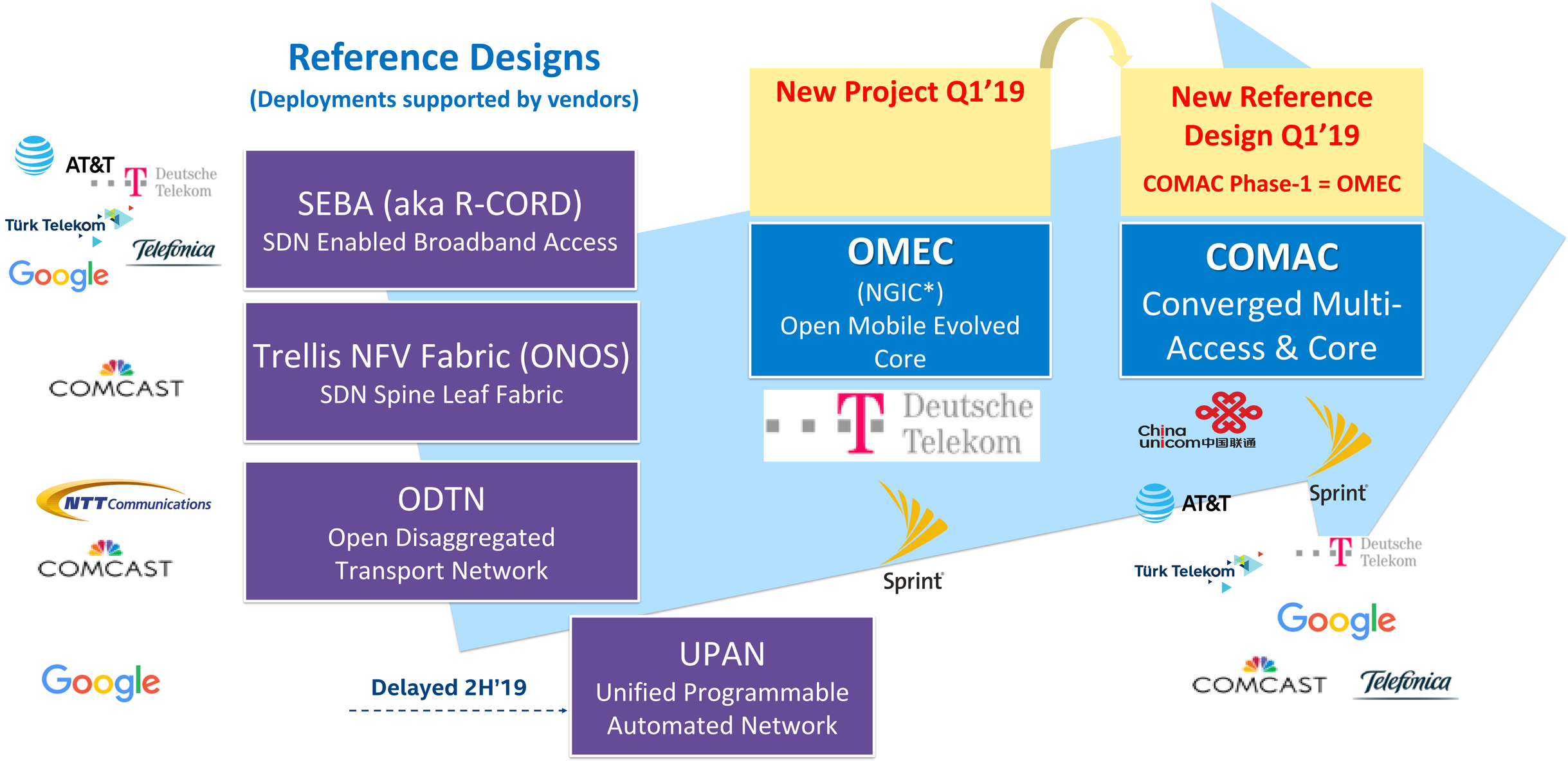
ONF Process & Go-To-Market (and Projects Maturity)

Reference Designs Provide Clear Path to Deployment and Monetization



Each Reference Design is paired with an open source Exemplar Platform (EP). These advance in parallel in order to circumvent the historical tendency to draft standards in isolation from working software.

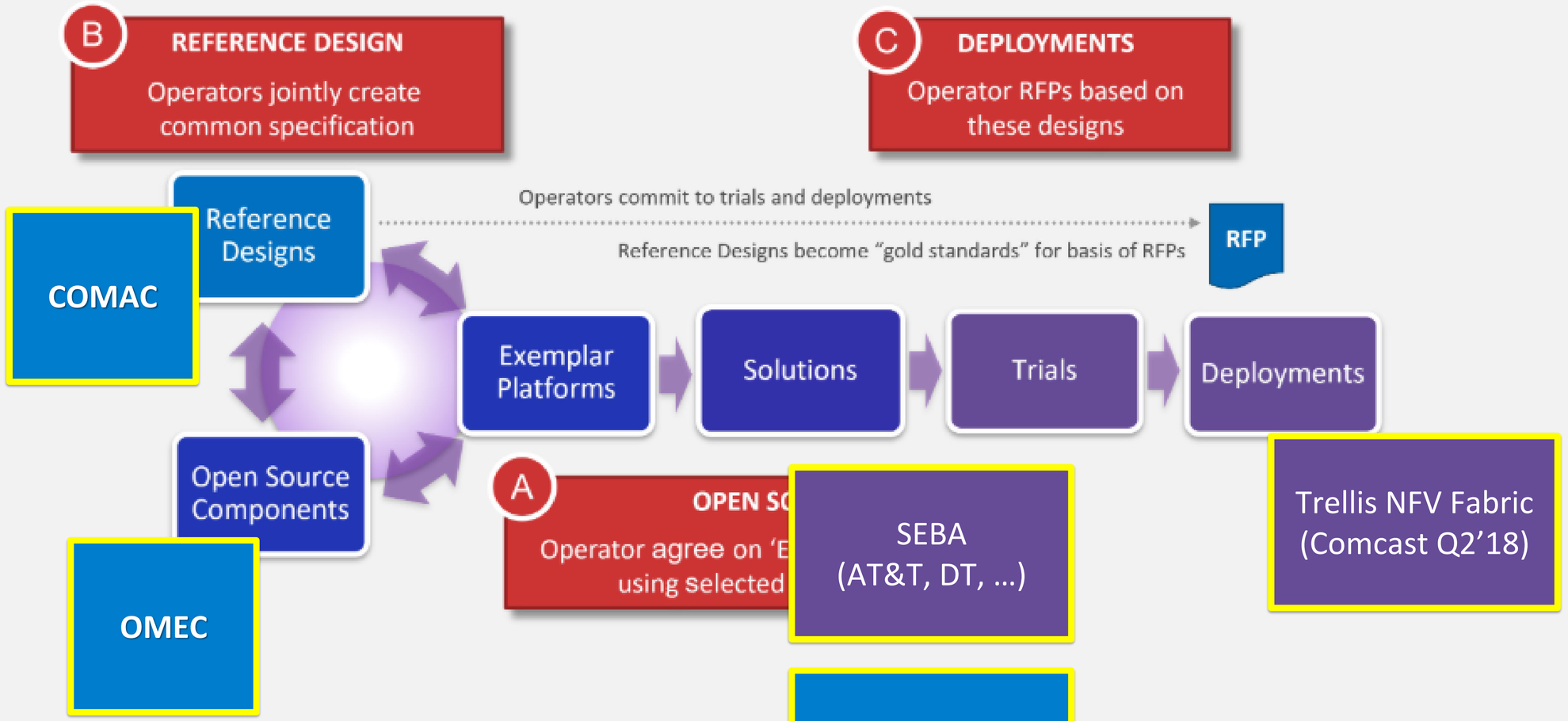
ONF Reference Designs ➔ Operators' Commitment to field trials/deployments



NGIC*: Next Gen Infrastructure Core

ONF Process & Go-To-Market (and Projects Maturity)

Reference Designs Provide Clear Path to Deployment and Monetization



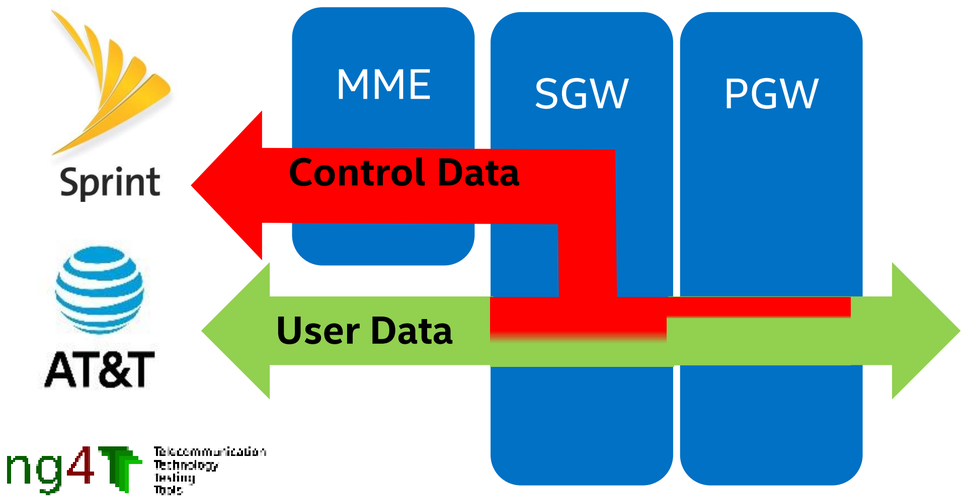
Each Reference Design is paired with an open source Exe ese advance in parallel in order to circumvent the historical tendency to draft s om working software.

Agenda

- Open Networking Forum (ONF) Background
- Open Mobile Evolved Core (OMEC)
  - History, features, deployment options (VMs, containers, …)
- OMEC in ONF
- Summary / Next Steps

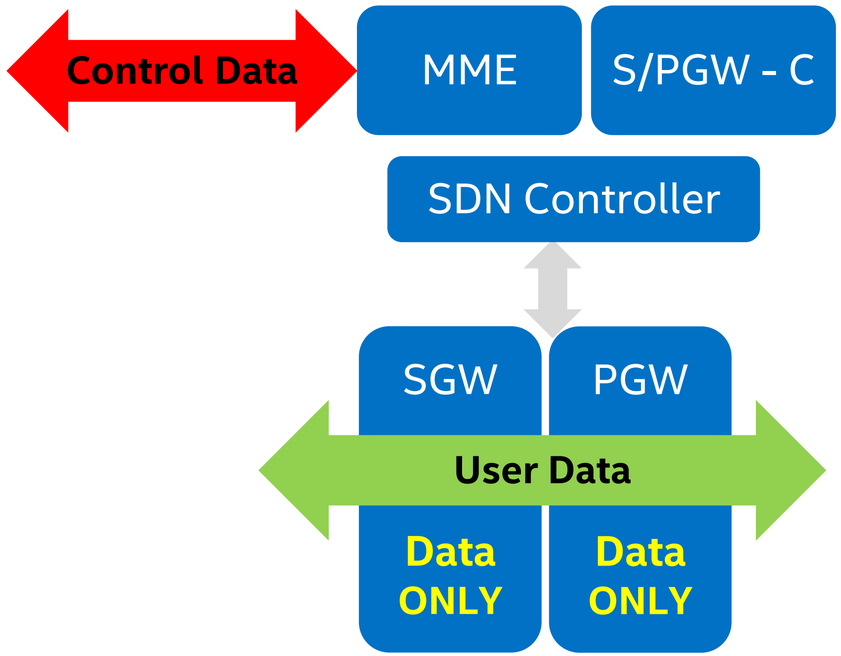
What led to OMEC ?

Traditional EPC Architecture



SW Defined Network (Separation of Control and Data)

Disaggregated Architecture



- Operators' real traffic (San Jose, Houston, Chicago, …)
- Identified system's bottleneck
  - "Understanding Bottlenecks in Virtualizing Cellular Core Network functions", IEEE LANMAN '15
- No independent control or data scaling

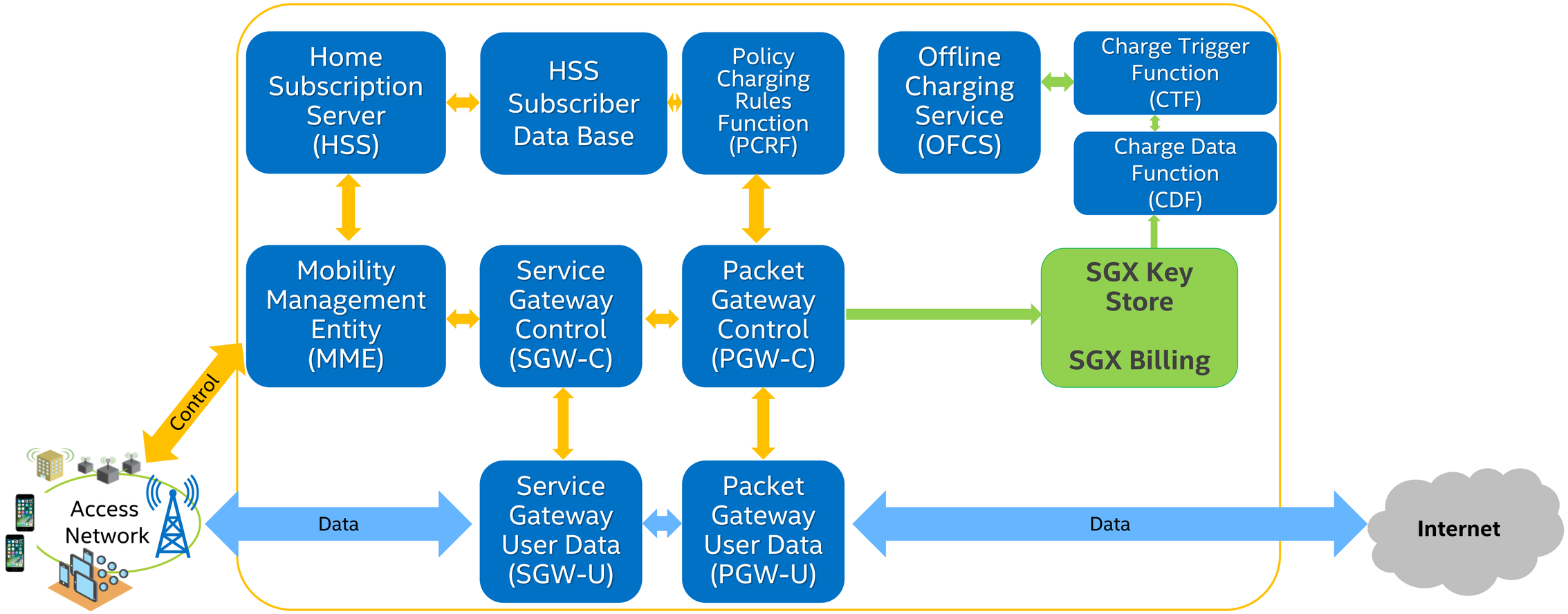
- SDN based architecture
- High Perf Match/Action semantic data plane
- Independent & scalable control & data
- Functional EPC per operator's requirements

OMEC – (COMAC RD Phase-1)

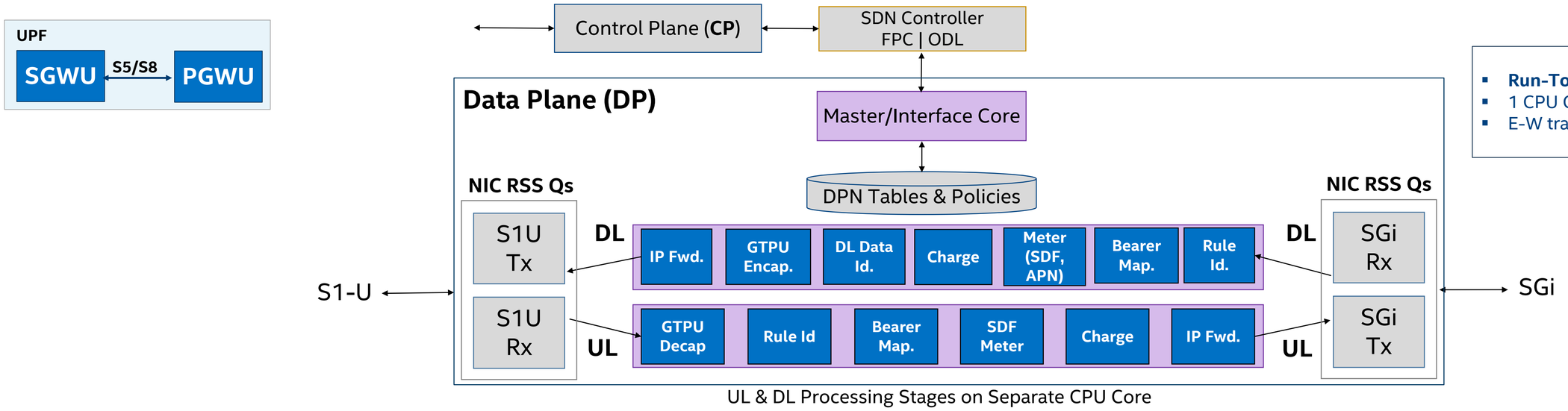
https://www.opennetworking.org/omec/

MME: Mobility Management Engine
HSS: Home Subscriber Services
PCRF: Policy and Charging Rules Function
SGW-C: Service Gateway Control
SGW-U: Serving Gateway User
PGW-C: Packet Gateway Control
PGW-U: Packet Gateway User
OFCS: Offline Charging Service
CTF: Charge Trigger Function
CDF: Charge Data Function

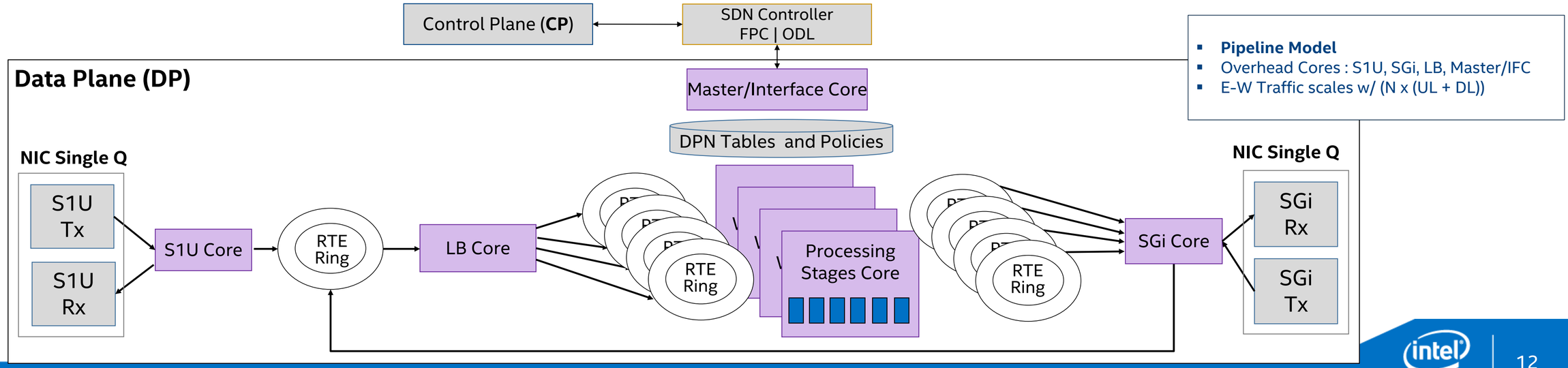
- Complete connectivity, billing and charging
  - Default bearers
  - Offline billing
  - Child protections (domain or 5-tuple)
  - Basic MME (initial attach/detach, etc)
- 3GPP Rel 13 compatibility
- DPDK based data plane, large number of subs
- Optimized for lightweight cost effective deployment
- ONF CI/CD test and verification infrastructure
  - Performance (w/ Polaris emulator)
  - 3GPP compliance (w/ Polaris)
- Future
  - TBD: Based on users' requests and contributions



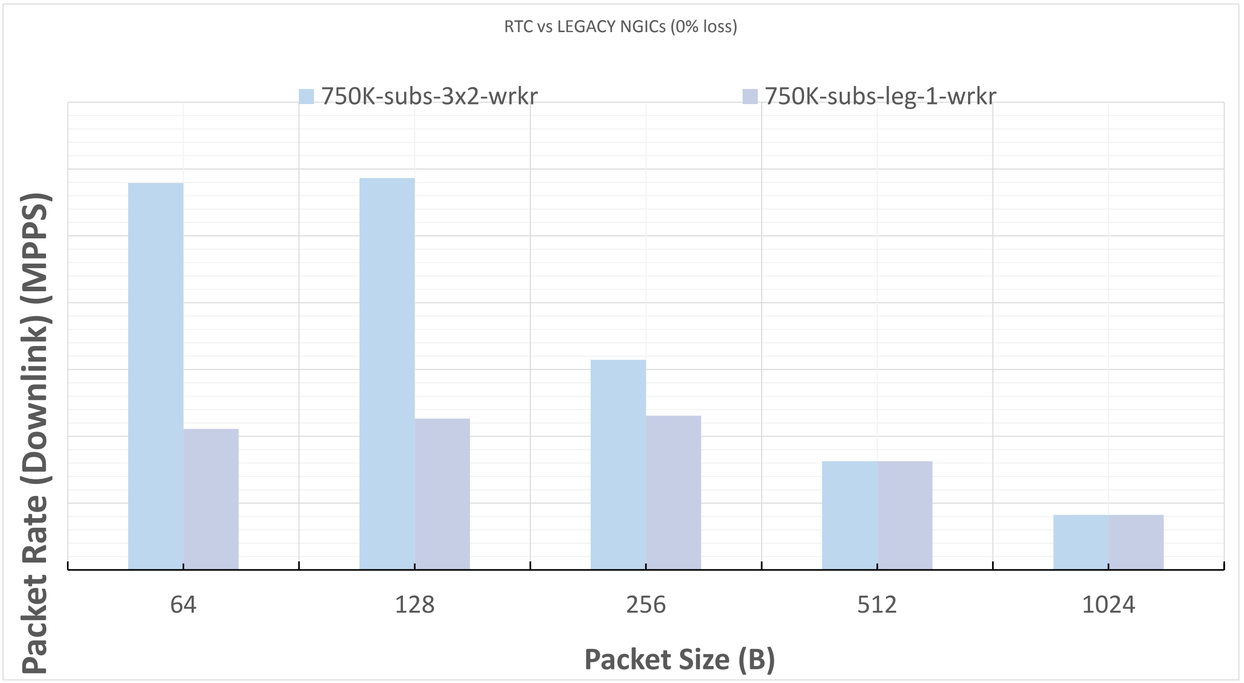
OMEC Data Plane Processing :– Pipeline or Run-to-Completion Model

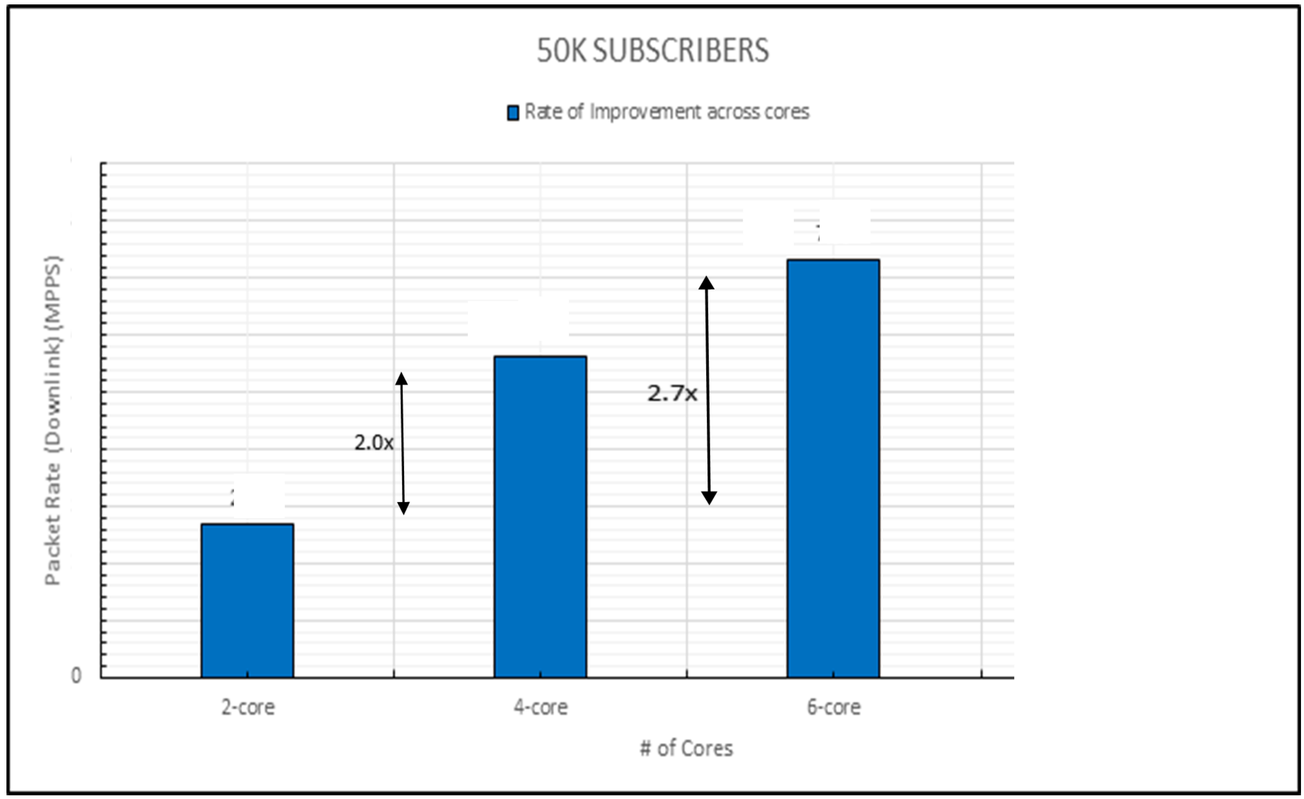


UL & DL Processing Stages on Separate CPU Core



Sample Performance Data





Intel® Software Guard Extensions (Intel® SGX) – The philosophy

❑ Enclaves

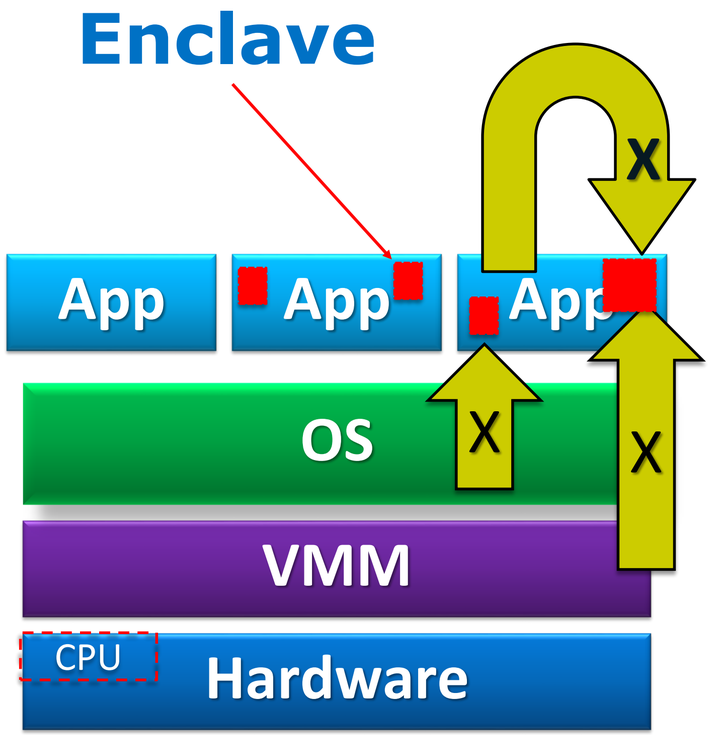
- Confidentiality and Integrity-protected data & code
- Controlled access to secrets with HW support for local and remote attestation
- Smaller attack surface

❑ Familiar development/debug

- Standard OS environment and programming model
- Single application environment
- Builds on existing ecosystem expertise

❑ Familiar deployment model

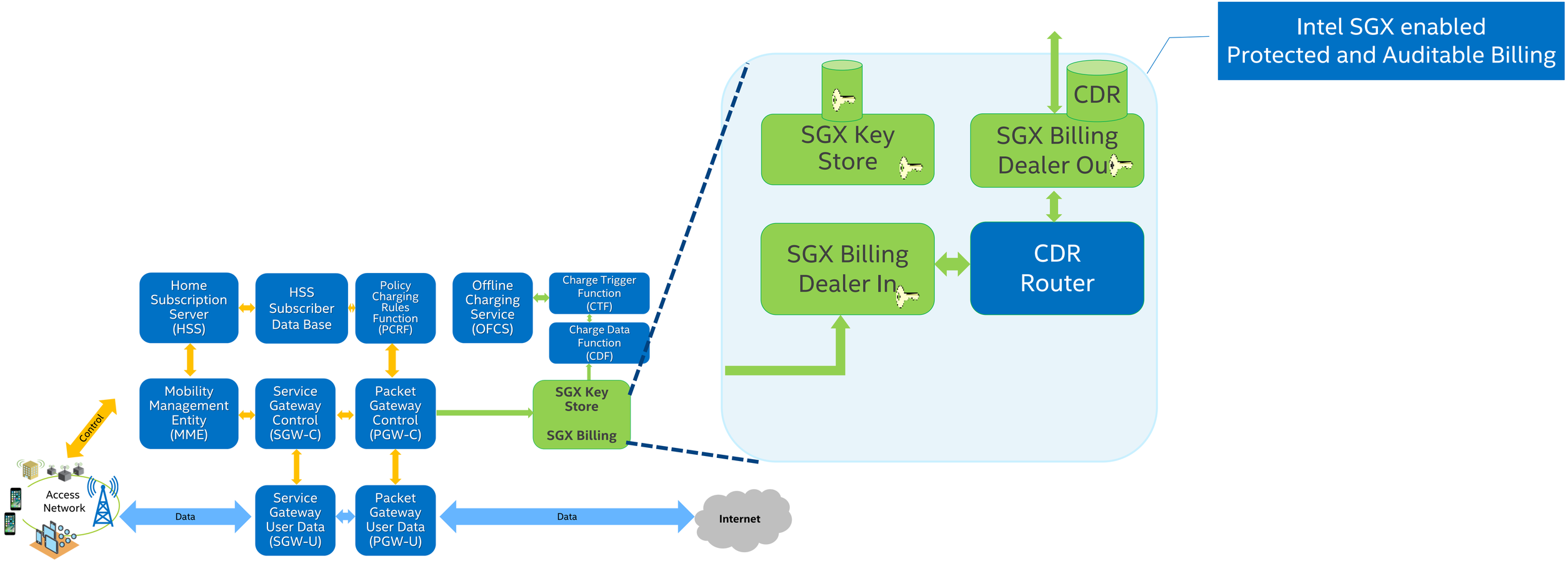
- Platform integration not a bottleneck to deployment of trusted apps



Attack Surface with Intel® SGX

Scalable security within mainstream environment

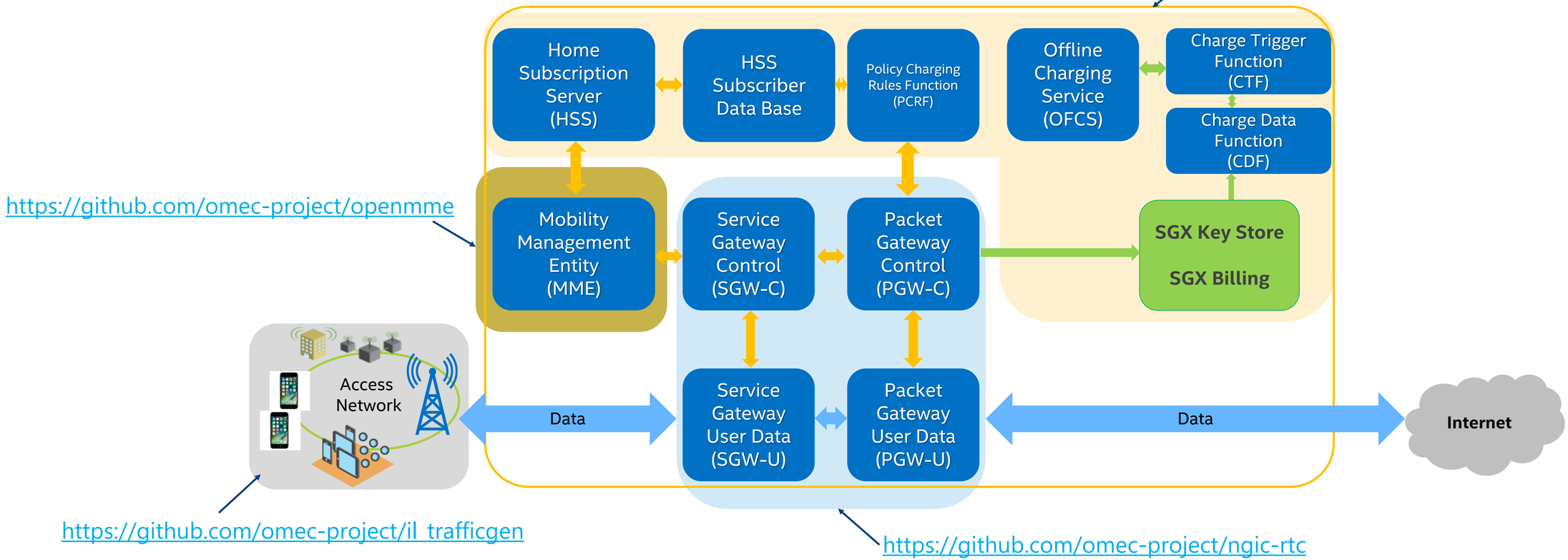
OMEC with Intel® Software Guard Extensions (Intel® SGX)



OMEC github Repositories

https://github.com/omec-project

https://github.com/omec-project/c3po



https://github.com/omec-project/openmme

https://github.com/omec-project/il_trafficgen

https://github.com/omec-project/ngic-rtc

Additional repos:

- CI/CD: https://github.com/omec-project/omec-project-ci

- Deployment: https://github.com/omec-project/deployment

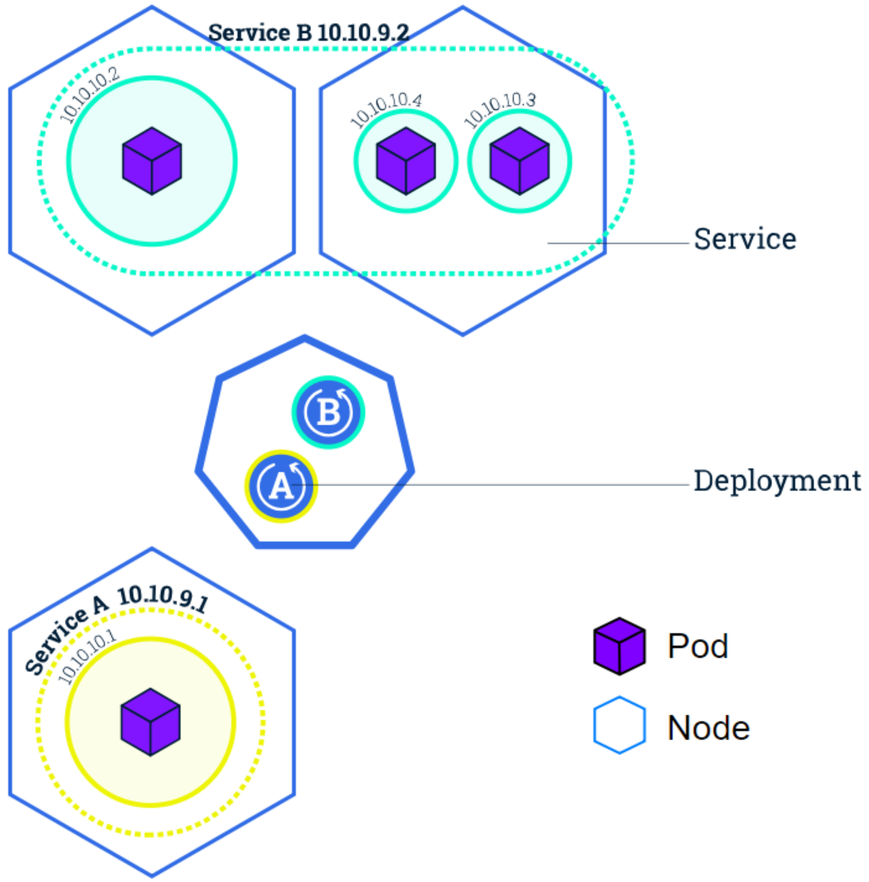
- Free Diameter: https://github.com/omec-project/freediameter

- CLI, etc: https://github.com/omec-project/oss-util

Agenda

- Open Networking Forum (ONF) Background
- Open Mobile Evolved Core (OMEC)
  - History, features, deployment options (VMs, containers, …)
- OMEC in ONF
- Summary / Next Steps

Overview: Kubernetes Taxonomy



- Pods
  - Smallest deployable object in the Kubernetes object model
- Pods created directly/indirectly are ephemeral
  - IPs cannot be relied upon, need way to reliably identify and target logical set of pods
- Pods targeted by a Service is (usually) determined by a Label Selector
  - Service is set to track “back-end” pods with a set of labels
  - Dynamically update Endpoints for the Service object, as back-ends change
- Controllers
  - Deployment - provides declarative updates for Pods and ReplicaSets, rolling updates, rollbacks

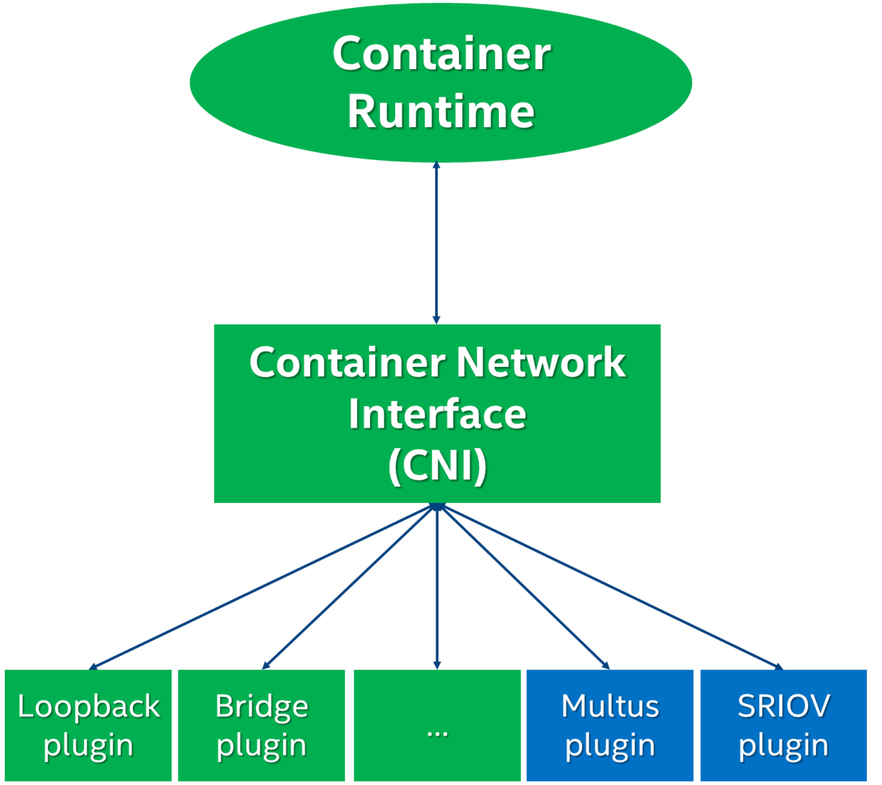
https://kubernetes.io/docs/tutorials/kubernetes-basics/expose/expose-intro/

OMEC based Containers orchestrated by Kubernetes

- ✓ Multiple networks and high-throughput I/O
- ✓ Ability to do service discovery on other networks
- ✓ Performance optimizations
  - ✓ CPU core pinning and isolation
  - ✓ Huge Pages

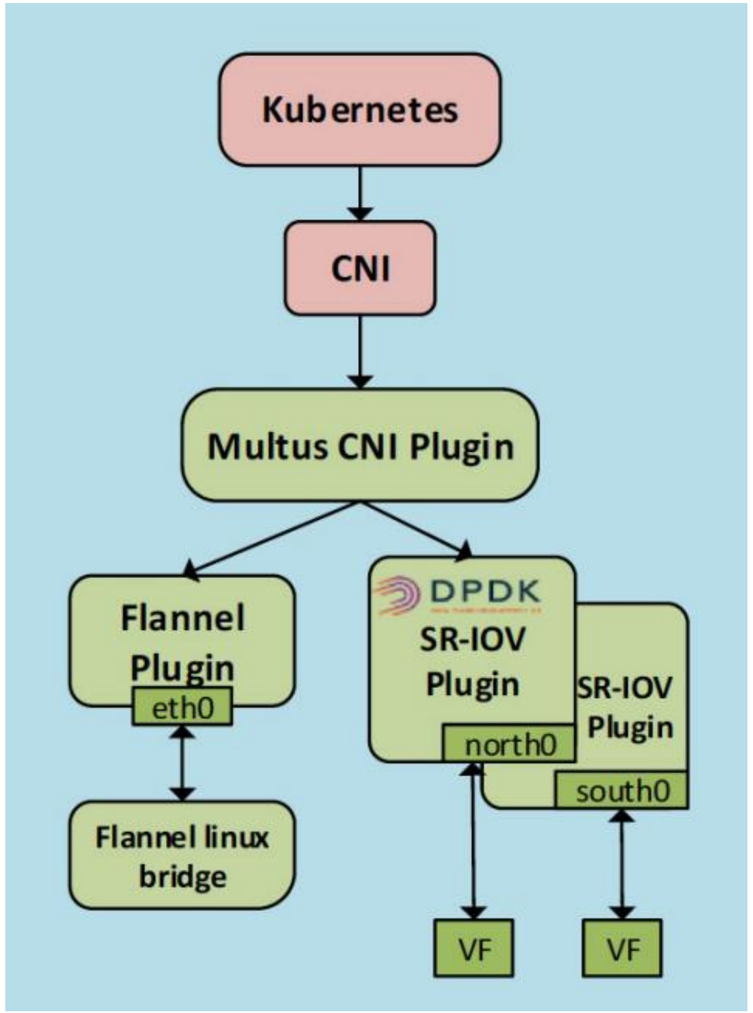
Overview: Kubernetes Networking – Container Network Interface

- K8s networking model limits one IP/interface per pod
- On Pod bring up, kubelet, the agent on the node, calls out to the CNI registered on the node to setup networking
- For multi-interfaces we use Multus CNI which acts as a proxy, to set up extra networks

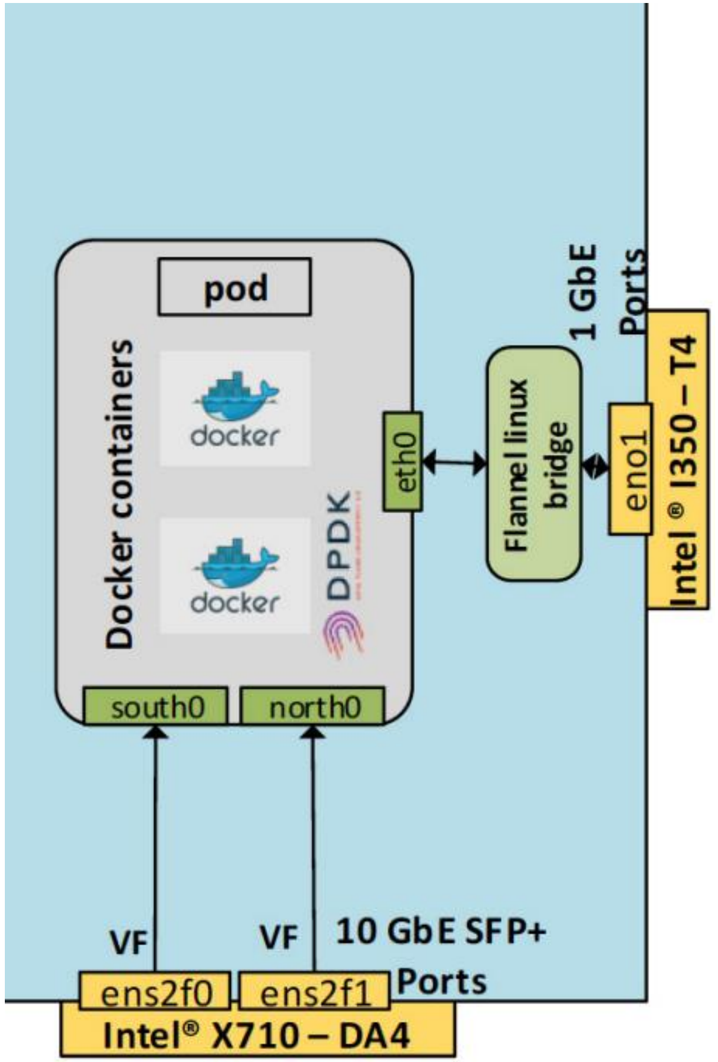


Overview: Multi-Network in Kubernetes (1/2)

Logical



Physical Manifestation

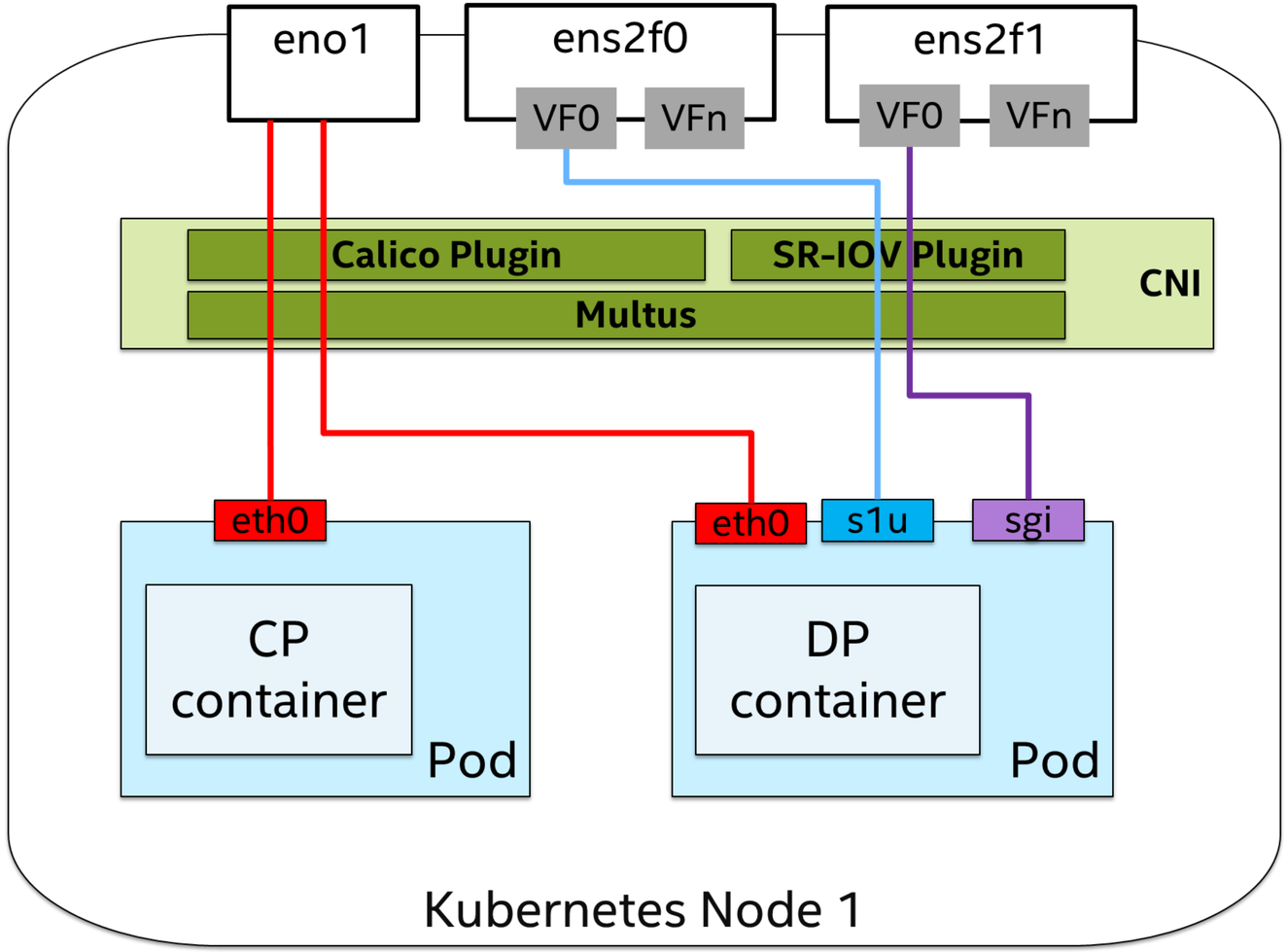


https://builders.intel.com/docs/networkbuilders/enabling_new_features_in_kubernetes_for_NFV.pdf

Overview: Multi-Network in Kubernetes (2/2)

✓ Multiple networks and high-throughput I/O for DP

- Multus CNI plugin and SR-IOV CNI plugin (enables VFs + DPDK user space drivers)

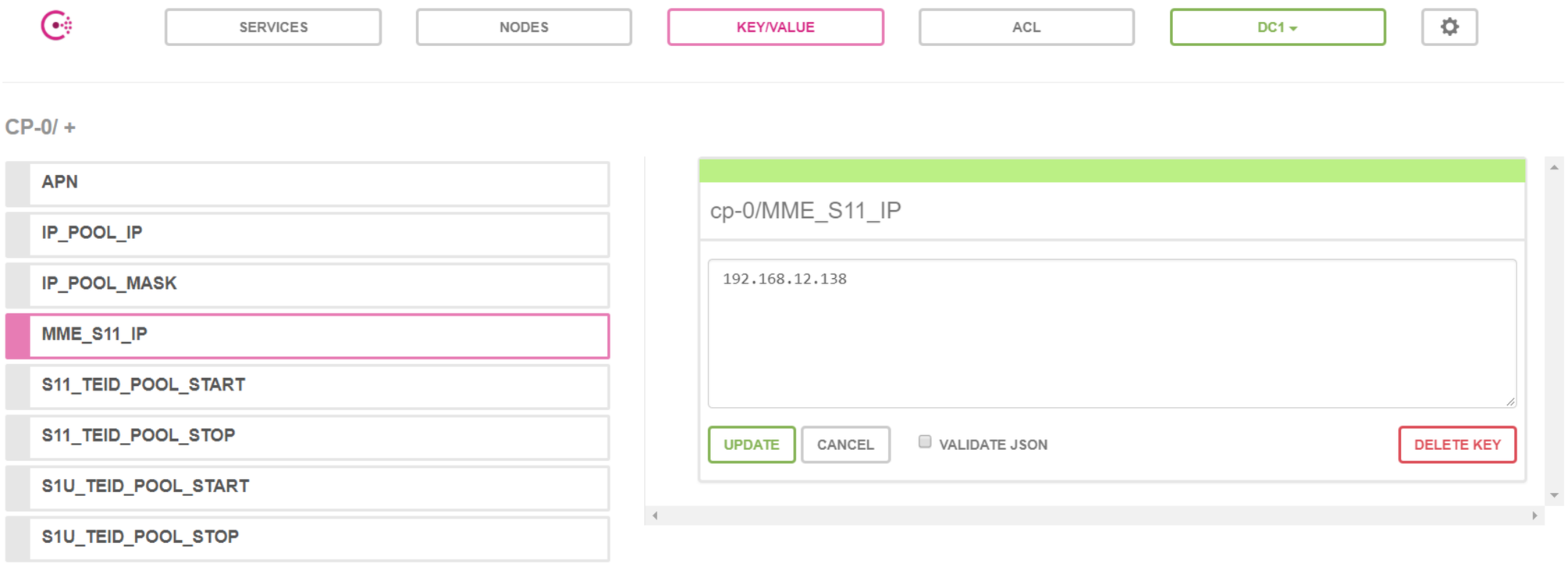


Overview: Service Discovery

✓ Multiple networks and high-throughput I/O for DP

✓ Ability to do service discovery on other networks

- Used Consul* to store and distribute discovery/configuration data



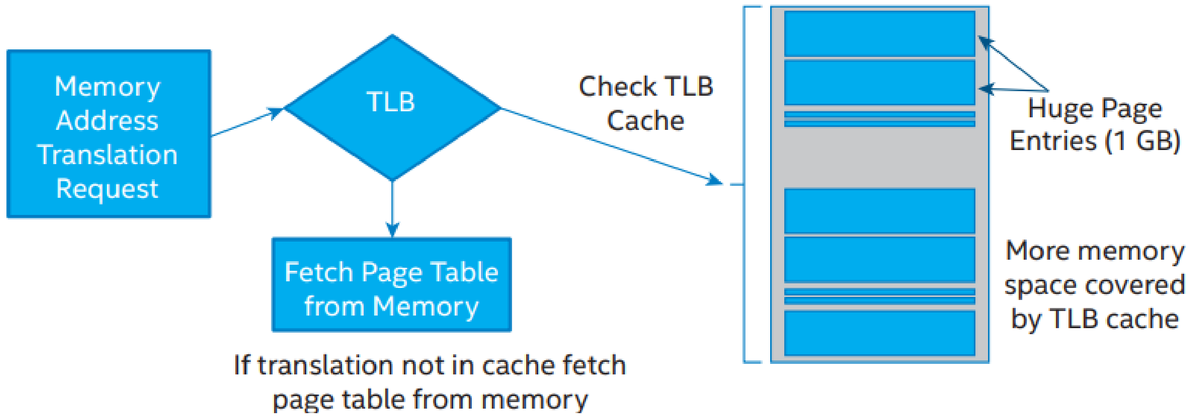
*Other names and brands may be claimed as the property of others.

Overview: Performance w/ Kubernetes (1/3)

- ✓ Multiple networks and high-throughput I/O for DP
- ✓ Ability to do service discovery on other networks
- ✓ Core pinning and isolation
  - CPU manager for k8s (beta): automated core mask gen for DPDK apps

Overview: Performance w/ Kubernetes (2/3)

- ✓ Multiple networks and high-throughput I/O for DP
- ✓ Ability to do service discovery on other networks
- ✓ Core pinning and isolation
- ✓ Huge Pages
  - Native resource in k8s (beta)



https://networkbuilders.intel.com/docs/kilo-a-path-to-line-rate-capable-nfv-deployments-with-intel-architecture-and-the-openstack-kilo-release.pdf

Overview: Performance w/ Kubernetes (3/3)

- Native – Running the binaries manually. No containers, no orchestration
- Kubernetes – Container version orchestrated with perf knobs toggled

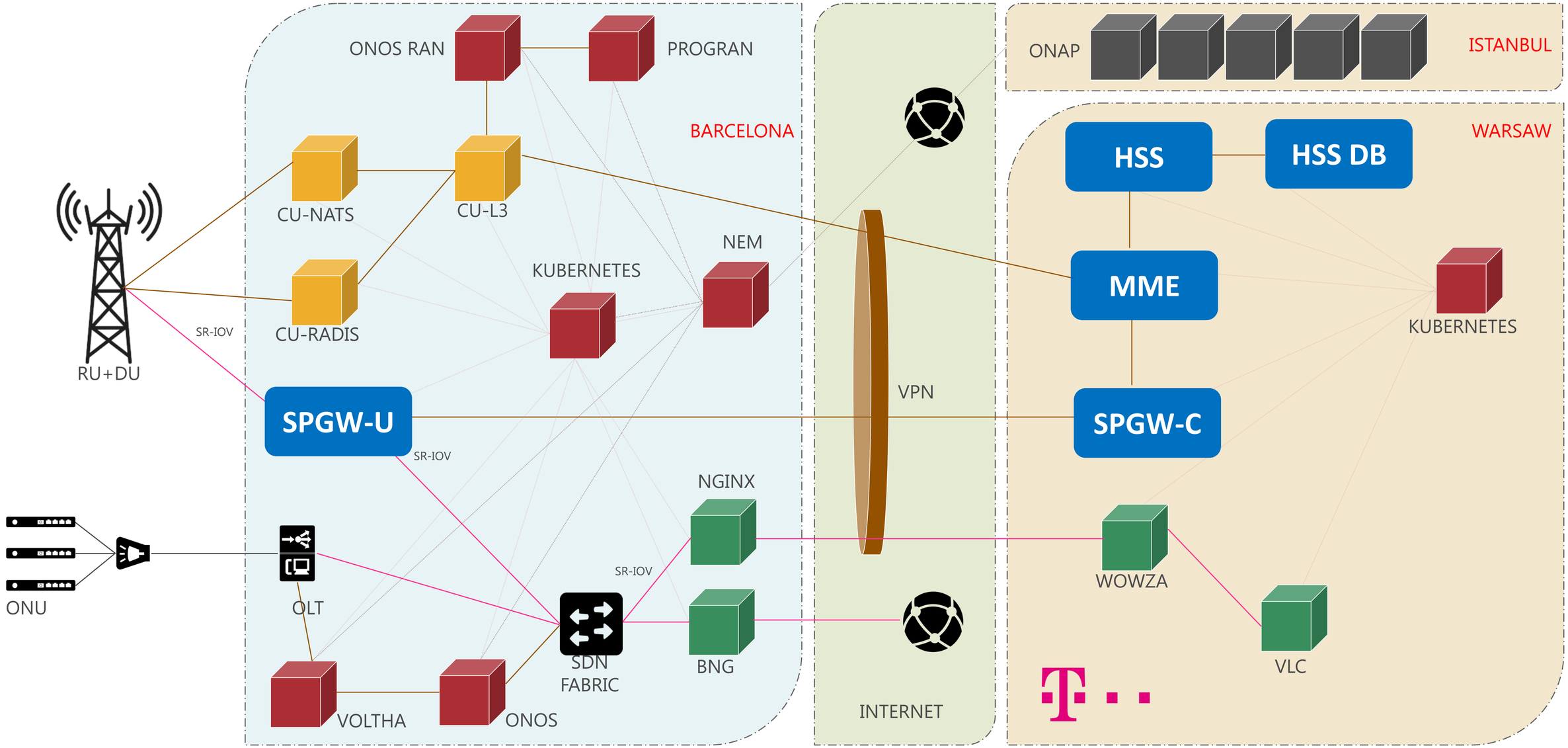
Test	Usr Sp Drv	Pinning	Huge	Pkts/sec*	*(w/noise)*
Native	yes	yes	yes	**1,550K**	*(1,100K)*
Kubernetes	yes	yes	yes	**1,450K**	*(1,150K)*
Kubernetes	**no**	yes	yes	**750K**	*(650K)*
Kubernetes	yes	**no**	yes	**1,450K**	***400K***
Kubernetes	yes	yes	**no**	**1,200K**	*(1,100K)*

* 50K Granularity (1 Worker Core)

Agenda

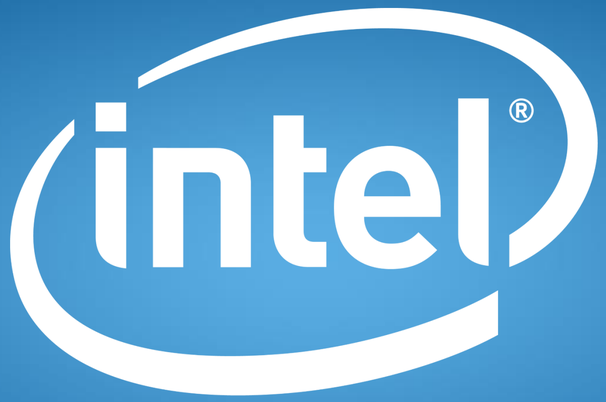
- Open Networking Forum (ONF) Background
- Open Mobile Evolved Core (OMEC)
  - History, features, deployment options (VMs, containers, …)
- OMEC in ONF
- Summary / Next Steps

OMEC @ MWC '19 : MULTI-CLOUD DEPLOYMENT



Summary

- OMEC is available
  - Sprint and DTAG/T-Mobile Poland announced field trials in ‘19
  - System Integrators involved, e.g. GS.Lab, HCL, Infosys
- OMEC is ONF Converged Multi-Access & Core Phase-1
- OMEC needs your contributions
  - Join OMEC github: https://github.com/omec-project
  - Contribute to any of the repos



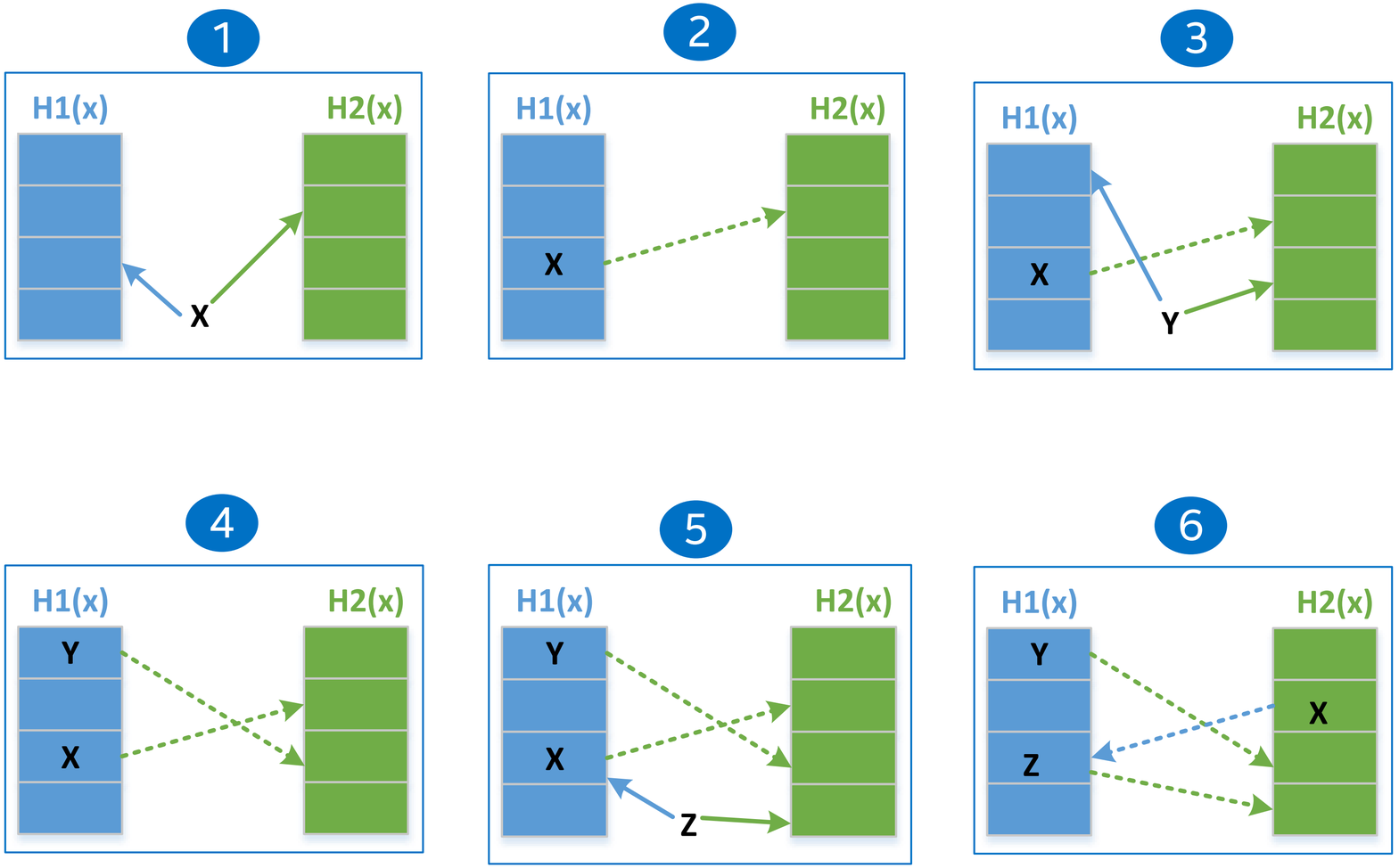
experience
what's inside™

Backup

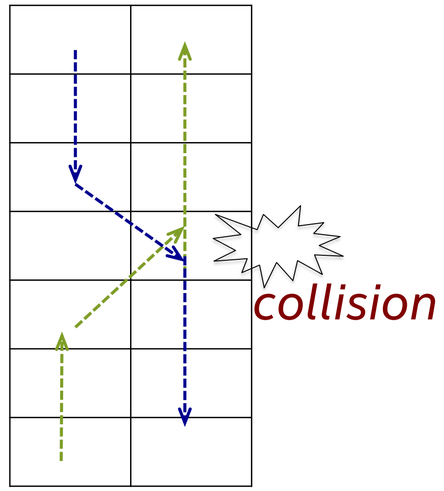
Efficient MATCH/Action Semantic Data Plane (1/2)

Match/Action : Optimized Table Lookup with Cuckoo Hashing[Pagh 01]

- "Scalable, High Performance Ethernet Forwarding with CuckooSwitch", Dong Zhu, Bin Fan, Dave Anderson (CMU), M. Kaminsky (Intel)



One `Insert` may move a lot of items especially at high table occupancy. Optimal multi-writer insertion using Intel® TSX



Efficient MATCH/Action Semantic Data Plane (2/2)