



# 5G NETWORK SLICING AND OPENSTACK

**Curtis Collicutt**

OpenStack Architect  
Interdynamix  
curtis@interdynamix.com

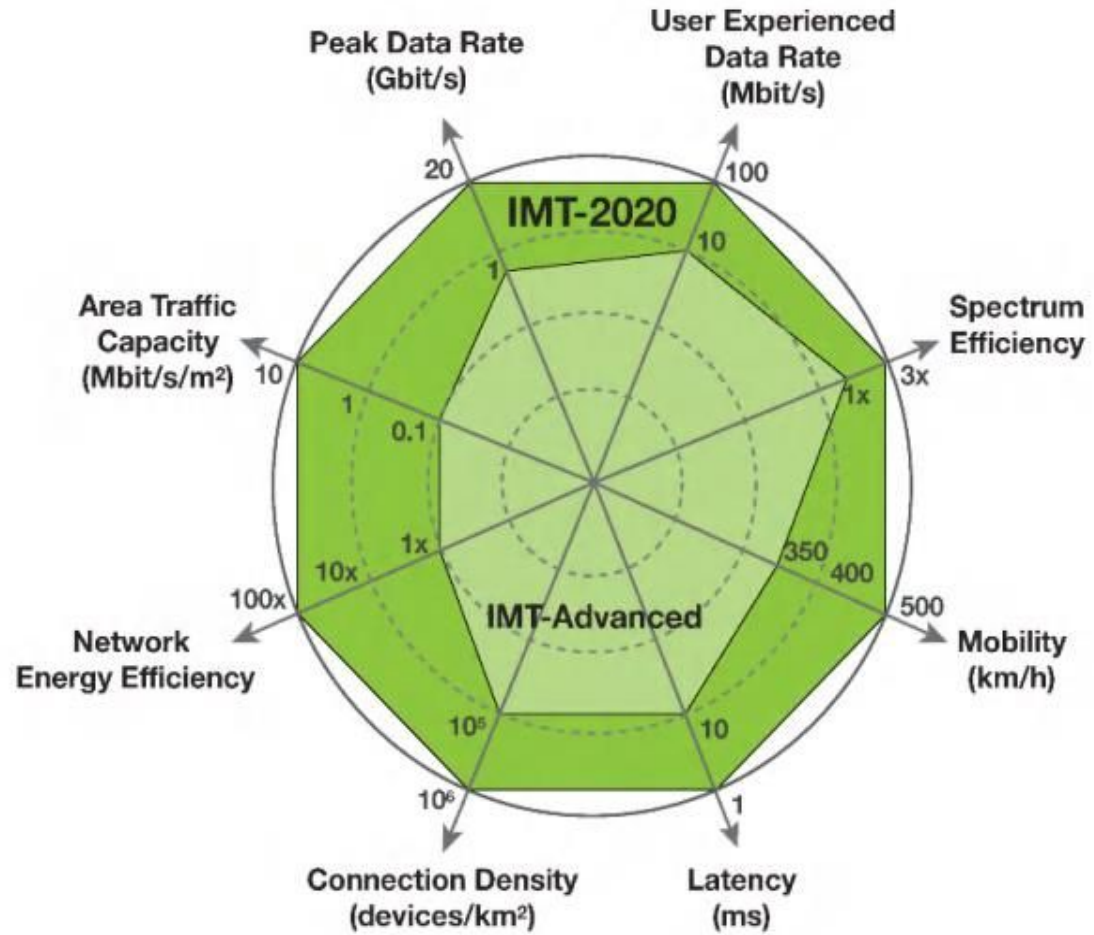
**Corey Erickson**

OpenStack Architect  
Interdynamix  
corey@interdynamix.com

1. What is network slicing and how does (or will) it work?
2. From a high level, we know it needs to be “end-to-end” which suggests that something is automating each “domain”...what is that thing?
  - a. OpenStack is not end to end (not supposed to be)
3. What role will OpenStack play in network slicing?
4. What features does OpenStack have now that meet the requirements of network slicing...and what else might be needed?

# 5G INTRODUCTION

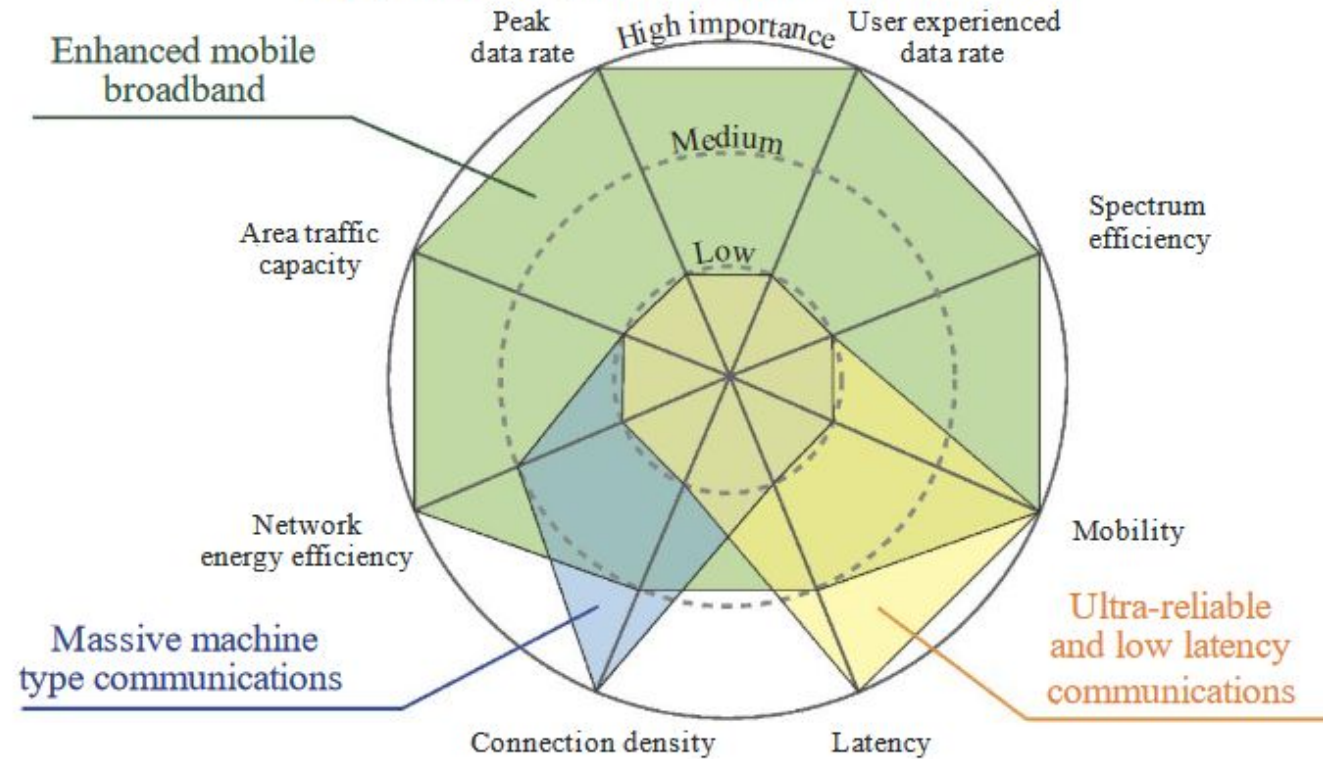
- Core Tenets
  - Transformation of the Network
  - Service Based Architecture
  - Virtualization and Slicing
  - Softwarization/Cloudification
  - APIs
  - Harmonized Protocols
  - Exposure to 3rd Parties
- Oft-stated performance requirements:
  - 1000X traffic demand
  - 10-100X connected devices
  - 3X spectrum efficiency
  - $\leq 4$ ms latency
  - 10 Gbits peak
  - 100 Mbits avg
  - 500 km/hr mobility
  - 10x connection density/km<sup>2</sup>
  - 10x area traffic capacity (mbits per m<sup>2</sup>)
  - 1000x lower energy/bit
- Also see ITU-R M.2410-0



Source: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52016SC0306>

FIGURE 4

The importance of key capabilities in different usage scenarios



M.2083-04

Source: ITU-R  
M.2083-0

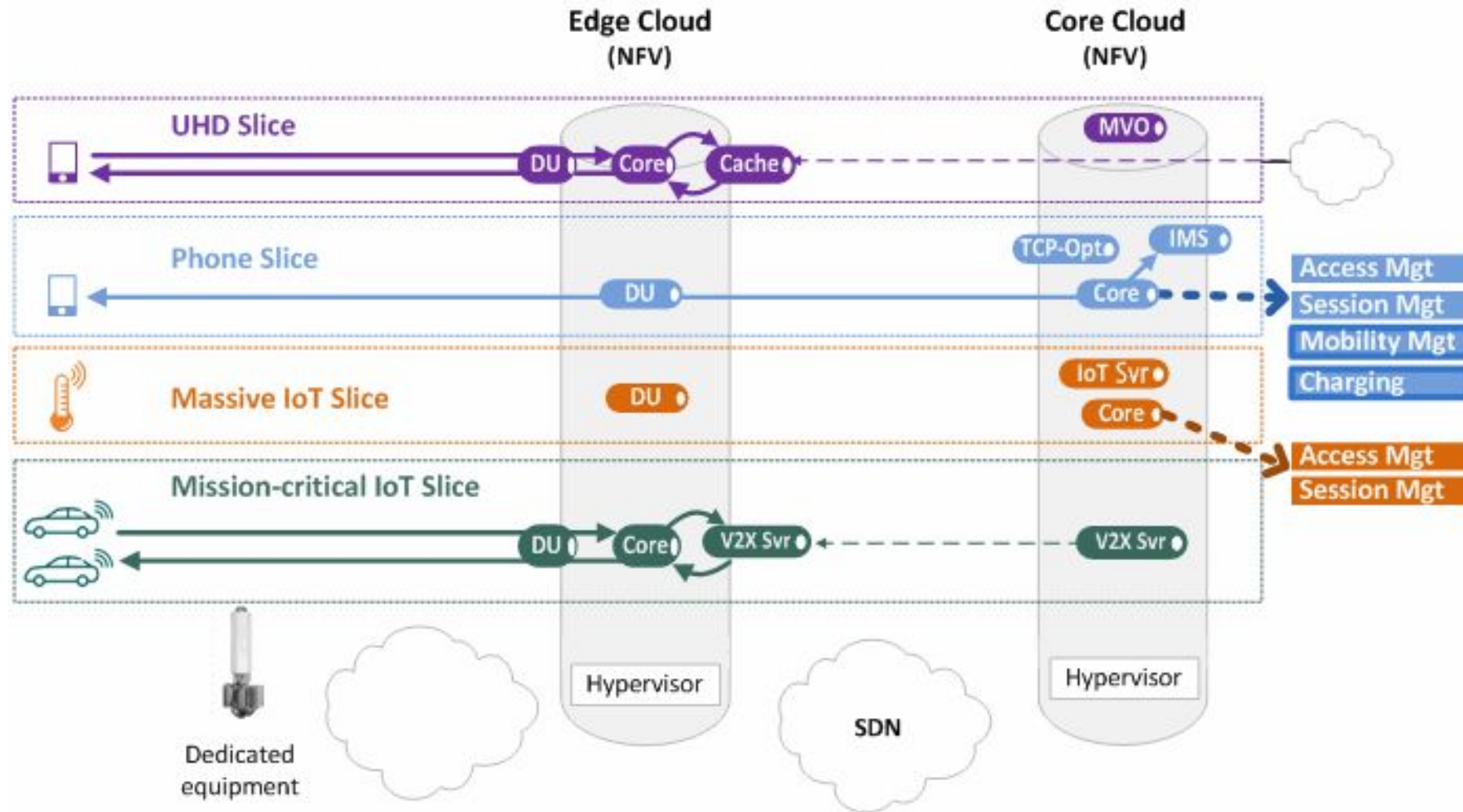
- eMBB - enhanced Mobile Broadband
- URLLC - Ultra Reliable Low Latency Communications
- mMTC - massive Machine Type Communications

	Latency	Mobility	Spectrum Efficiency	User experienced data rate	Peak data rate	Area traffic capacity	Network energy efficiency	Connection density
<b>eMBB</b>	Med	High	High	High	High	High	High	Med
<b>URLLC</b>	High	High	Low	Low	Low	Low	Low	Low
<b>mMTC</b>	Low	Low	Low	Low	Low	Low	Med	High

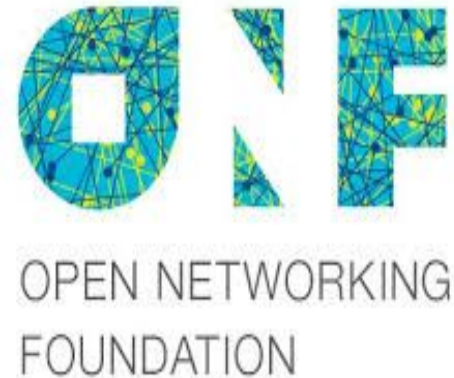


# 5G NETWORK SLICING

How do we support diverse use cases?



Source: E2E Network Slicing - Key 5G technology



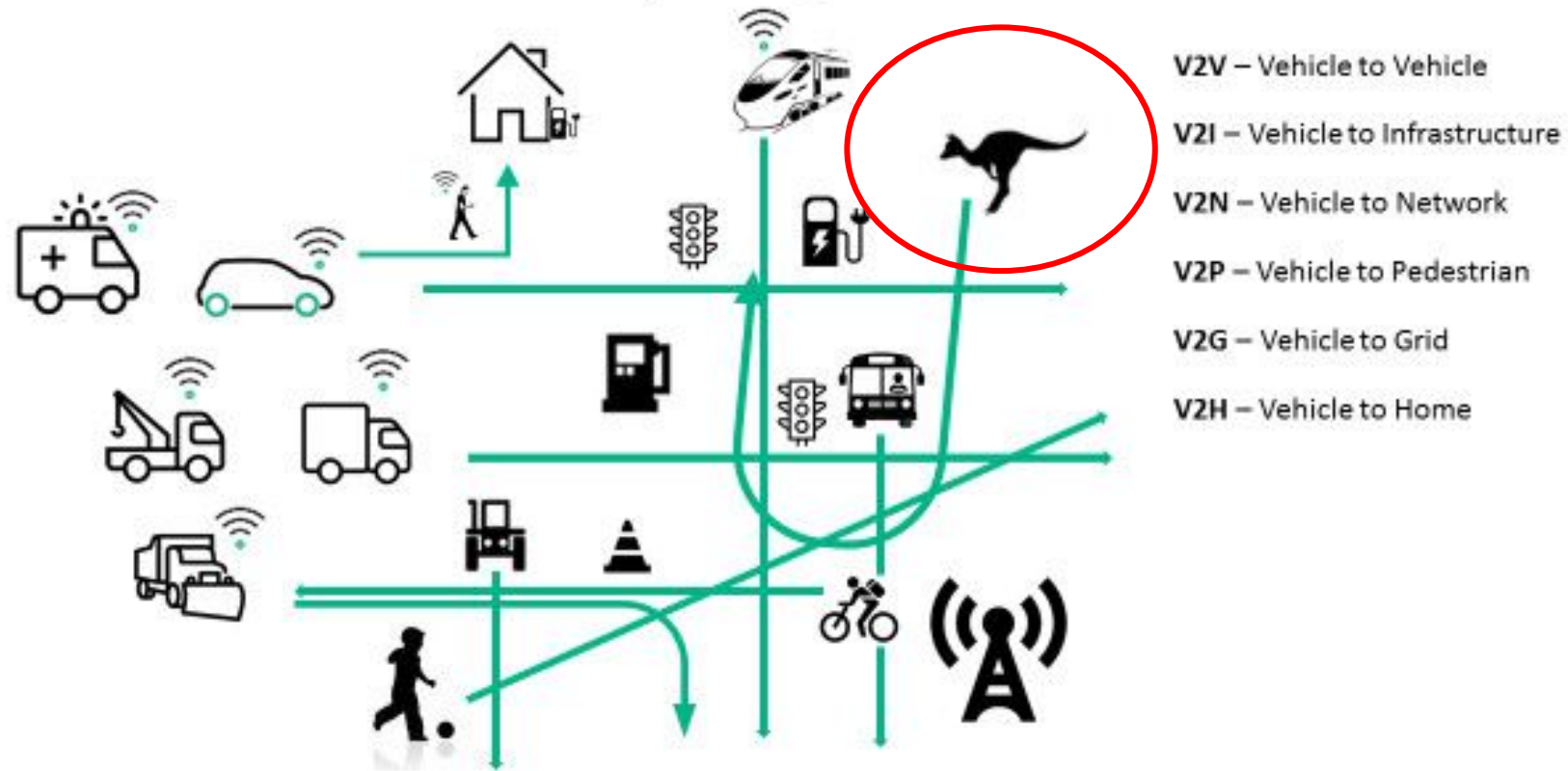
“We define network slices as **end-to-end(E2E) logical networks running on a common underlying (physical or virtual) network**, mutually isolated, with independent control and management, and which can be created on demand.” - *Network Slicing for 5G with SDN/NFV*

“The industry consensus is that 5G should be known not only for its cutting-edge radio access technologies, but also for the way it integrates **cross-domain networks** so operators can provide networks on a need-for-service basis.” - *5G Americas Network Slicing*

- **End to End** - RAN, CN, Transport, etc, etc
- **Orchestration** - Manage heterogeneous network resources
- **On Demand** - CRUD actions at any time
- **Elastic** - Grow/shrink network services resources based on need
- **Extensible** - Expand network slice with additional functionality and characteristics
- **Safety** - Failures in one slice not causing failures in others
- **Protection** - Events in one slice do not have a negative impact on other slices

- **Recursion** - Ability to build a new network slice out of existing slices
- **Isolation** - Guaranteed isolation and non-interference between network slices in data and control plane
- **Flexible** - Simultaneously accommodate diverse use cases
- **Exposure** - Provide secure access to 3rd parties
- **Cross Domain** - Multiple operators
- **Network Functions** - Not just switches and routers
- All that and a Kangaroo...

## V2X – Vehicle to Everything



Source: <https://sdn.ieee.org/newsletter/december-2017/v2x-and-network-slicing>

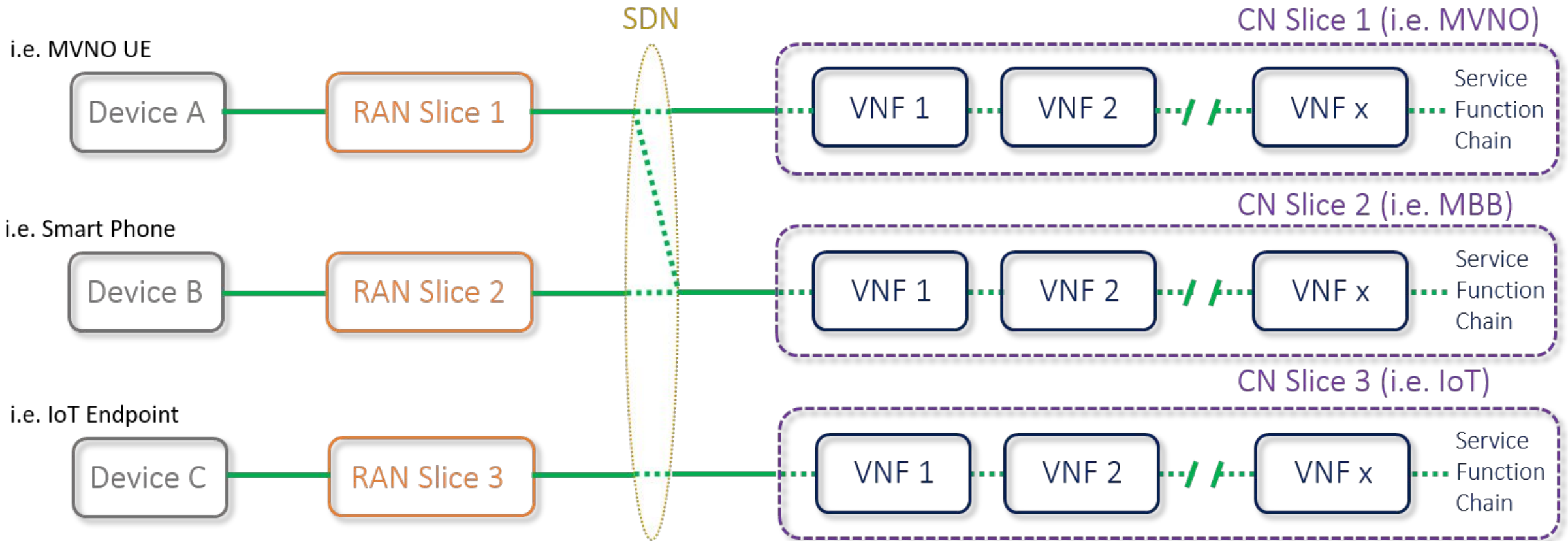
# NETWORK SLICING AND NETWORK FUNCTION VIRTUALIZATION

How will NS and NFV work together?



“Through technologies like **Software-Defined Networking (SDN)** and **Network Function Virtualization (NFV)**, network softwarization can provide the programmability, flexibility, and modularity that is required to create multiple logical (virtual) networks, each tailored for a given use case, on top of a common network.” - *Network Slicing for 5G with SDN/NFV: Concepts, Architectures and Challenges*

- NS is not an NFV technology...not part of the standard NFV definition
- But NS would likely need NFV to be viable
- Also when using the term NFV, are thinking about “ETSI NFV” and the surrounding ecosystem



Source: 5G Network Slicing – Separating the Internet of Things from the Internet of Talk

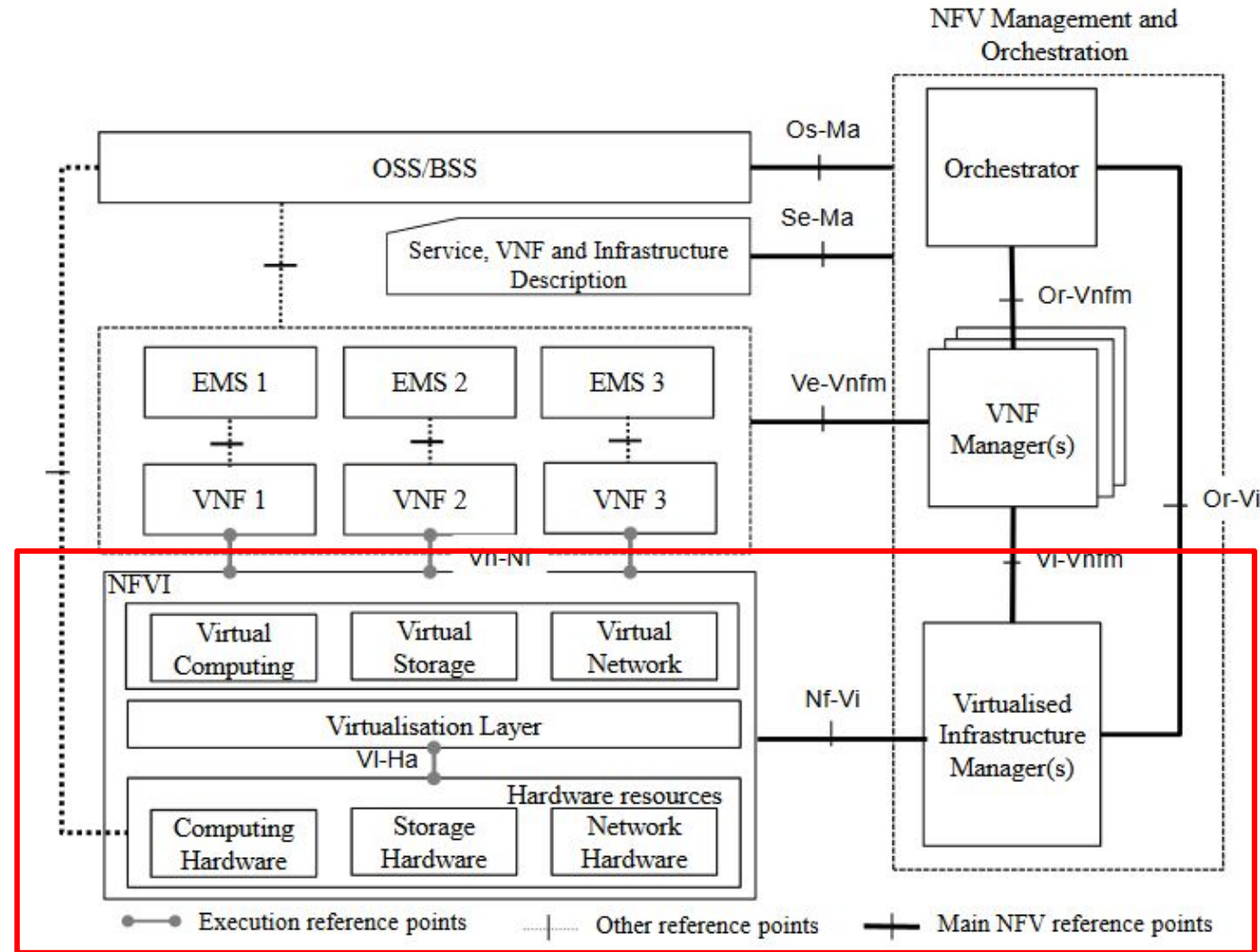


Figure 4: NFV reference architectural framework

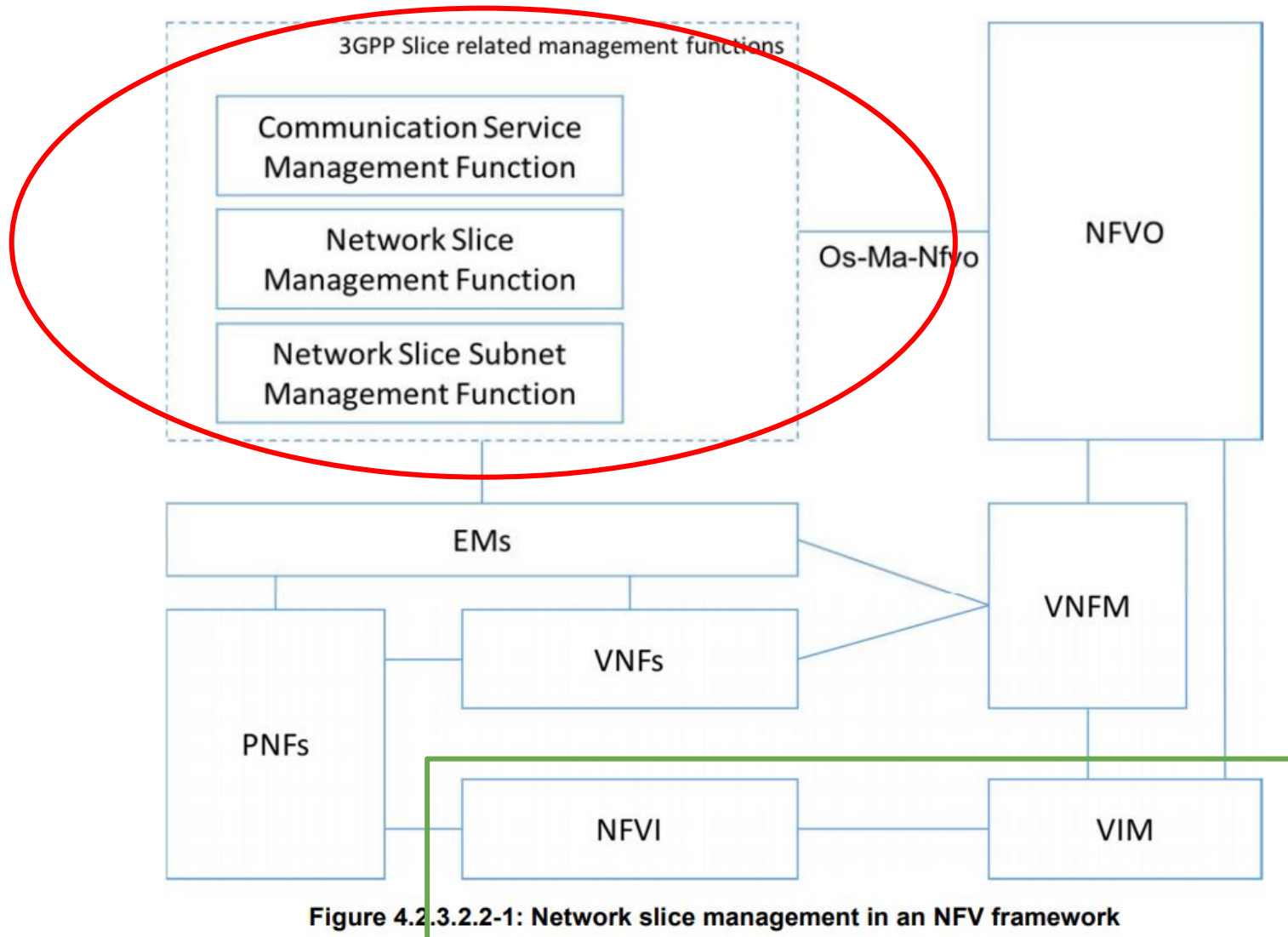


Figure 4.2.3.2.2-1: Network slice management in an NFV framework

- Attempts to map NGMN, 3GPP, and ONF definitions of NS to NFV
- “[...]document describes how these use cases **could be** mapped to the current NFV concepts supported by the ETSI NFV architectural framework and by NFV-MANO”
- “**...there is no common definition**” of Network Slicing
- Slice vs Service: “An NFV Network Service (NS) can thus be regarded as a **resource-centric view of a network slice**, for the cases where a Network Slice Instance (NSI) would **contain at least one virtualized network function.**”
- Network functions, both physical and virtual, will be part of network slices
- Note: Typically ETSI MANO does not deal with PNFs

Tenant SDN Controller (VNF)

Infrastructure SDN Controller

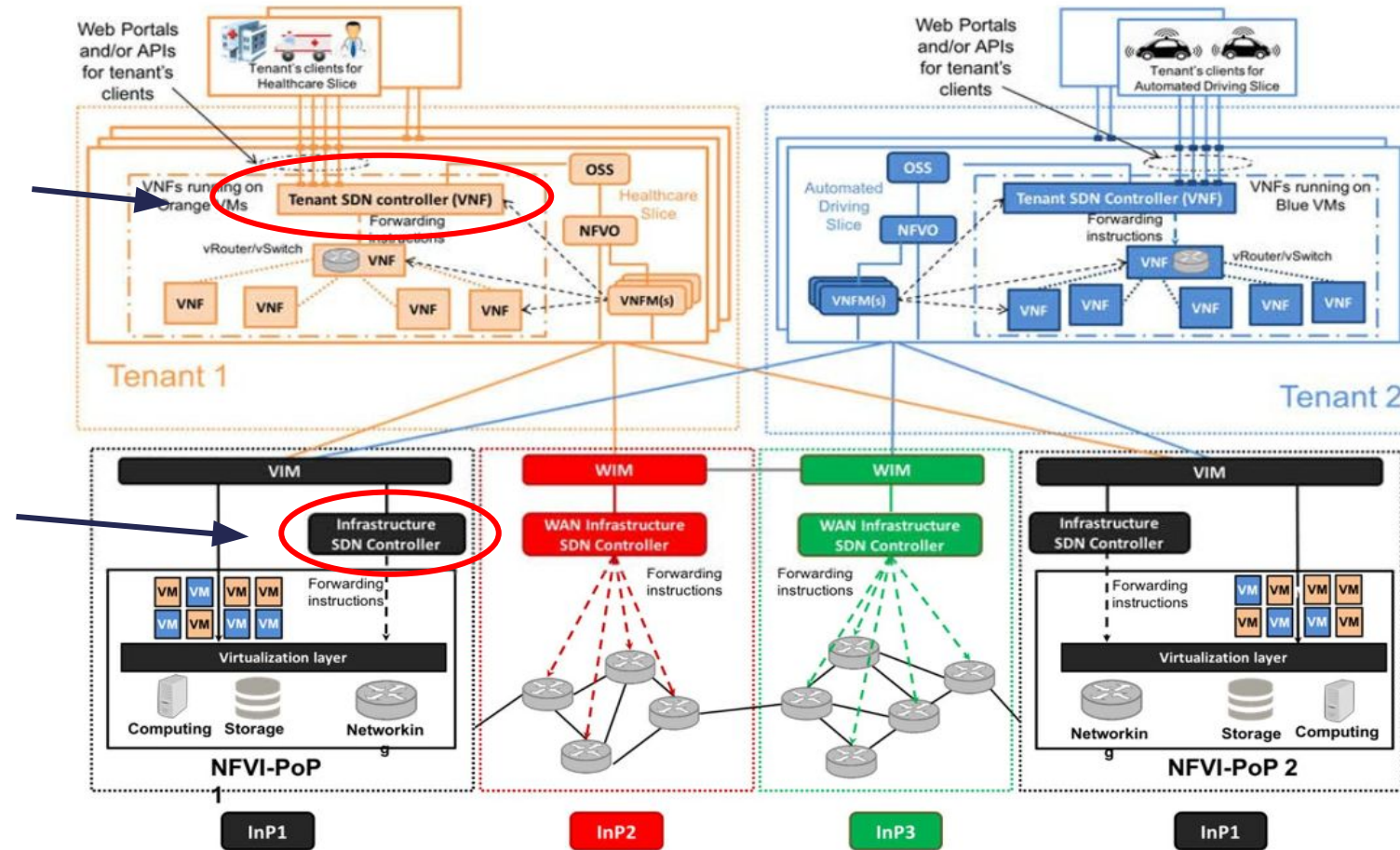


Figure 4.3-1: Network slicing deployment applying NFV concepts to achieve isolation

Source: ETSI GR NFV - EVE 012 V3.1.1 (2017 12)

# OPENSTACK AND NETWORK SLICING

- Isolation and multi-tenancy
- Management APIs
- Cloud native, elasticity, flexibility
- Defacto standard API for VIM
- As NFVi, manages compute, storage, network
- Multiple SDN integrations
- Continual NFV related improvements
- Emerging trends:
  - Edge computing
  - Multi-cloud management





- Security group rules
- neutron-fwaas
- **networking-sfc**
- **networking-bgpvpn**
- TAP as a Service (TAPaaS)
- **Neutron QoS features**
- Congress
- **Group Based Policy**
- Tacker
- Spec: [Neutron-Neutron Interconnect](#)
- **Neutron Common Classification Framework**
- Octavia
- **Neutron RBAC**
- **Nova Placement API**
- **Network aware scheduling**
- Enhanced Platform Awareness
- Multisite/region
- Tricircle
- Others?

- Scheduling is extremely important for compute, but just as important in networking for NS
- A lot of work is being done with the Nova Placement API
- “This is a separate REST API stack and data model used to **track resource provider inventories and usages, along with different classes of resources.**”
- “...NFV means ‘never-ending feature velocity’” - Jay Pipes, OpenStack Dev List



“Group Based Policy (GBP) is an optional service plugin for Neutron that provides **declarative abstractions** for achieving scalable intent-based infrastructure automation. GBP complements the OpenStack networking model with the notion of policies that can be applied between groups of network endpoints.” - [RDO](#) Docs on GDP

“Network service chaining is a key capability of Group-Based Policy (GBP) that describes the requirements for ordered chains of services by separating out network-specific policies from service-specific details.” - [GBP](#)

- Can share “objects”, eg. a network, between tenants
- QoS Policy is a shareable object as well
- Would help with hierarchical/recursion requirements

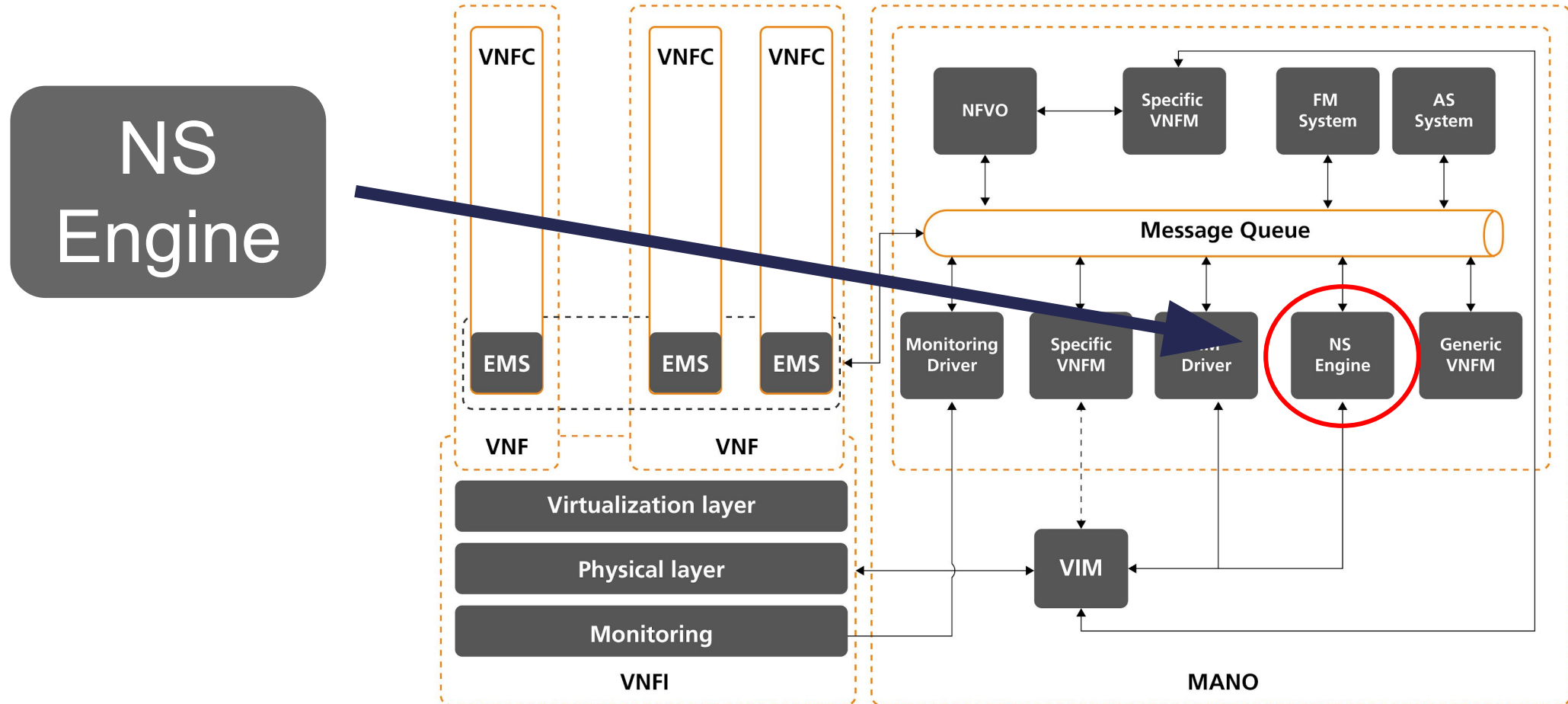
```
neutron rbac-create --target-tenant \  
<tenant ID> \  
--action access_as_shared --type qos-policy \  
<policy ID>
```

- Neutron Common Classification Framework
  - Centralize common classifications of traffic, for example networking-sfc, fwaas, others classify traffic
  - Spec - <https://review.openstack.org/#/c/190463/>
- Network Aware Scheduling
  - Mostly focussed around rack aware scheduling, but presumably other scheduling could be supported in future...”Supporting Neutron routed networks is a dependency for using Neutron with multiple cells.”

- Can build QoS policies with rules
- Applied to port or network
- Currently
  - Egress bandwidth limit - Max BW limit, optionally max burst
  - DSCP markings - RFC 2474
- Open Baton NSE can use these policies
- Future: Some mention of minimum bandwidth policies as well
- Overall, QoS is a difficult problem

# OPEN BATON NETWORK SLICING ENGINE

A Reference Implementation of ETSI MANO with a NSM



Source: <https://sdn.ieee.org/newsletter/july-2016/open-baton>



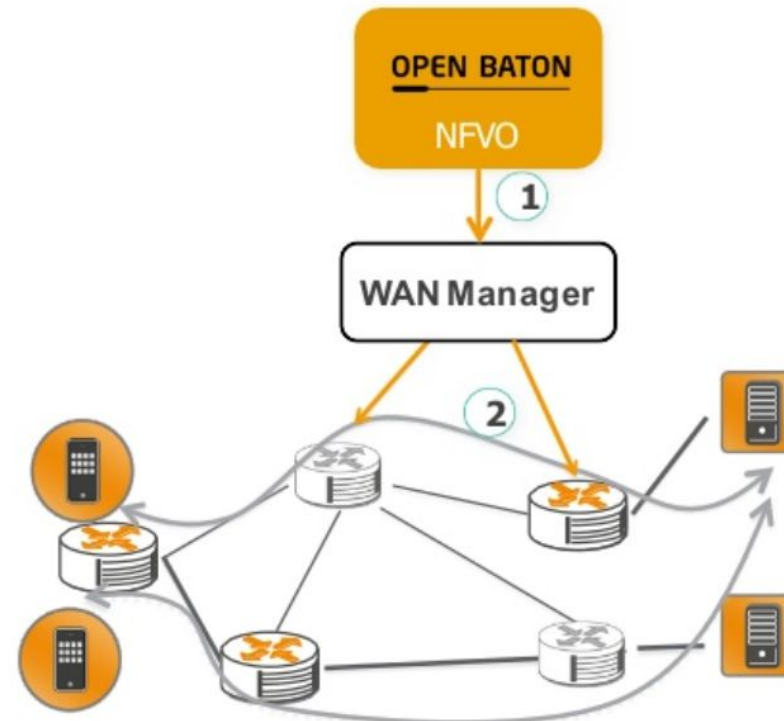
## Integration with the SDN WAN for Network Slicing use cases

For an end-to-end network requirements control the NFVO can be integrated with the WAN Manager

Instantiation of multiple slices on top of the real network for connecting VNFs

- Realizing traffic steering/QoS
- Dynamic adaptation to network conditions

Service chaining: realizing data paths for the end subscriber sessions



- QoS policies defined/hardcoded in OpenBaton NSE code

```
23  */
24  public enum Quality {
25      // 1126 mb/s burst - 1024 mb/s
26      PLATINUM("1181116006", "1073741824"),
27      // 110 mb/s burst - 100 mb/s
28      GOLD("115343360", "104857600"),
29      // 11 mb/s burst - 10 mb/s
30      SILVER("11534336", "10485760"),
31      // 1.1 mb/s burst - 1 mb/s
32      BRONZE("1153433", "1048576"),
33      // 0.2 mb/s burst - 0.1 mb/s
34      COAL("209715", "104857");
35  }
```

1. Using OpenBaton, we will instantiate a single VNF into an OpenStack PoP
  - a. The pre-defined VNF descriptor includes QoS policy requirements
2. When the VNF registers the NSE will receive notification
  - a. The notification will include QoS policy requirements
3. The NSE will ensure the QoS policy exists in Neutron or create it if not
4. The policy will be applied to the VNF's port

The image shows a terminal window with a dark background and light-colored text. The text is a log of OpenStack operations. A large, semi-transparent white box is overlaid on the center of the terminal, containing the text "OpenStack QoS policy created and applied to OVS port". The log text includes various timestamps and thread identifiers, such as "[openstack-0] 2018-05-01 14:28:29.818". The log shows the creation of a QoS policy and its application to a specific port.

Demo: Deploying a VNF with a QoS Policy via OpenBaton NSE

- Issues
  - Had issues with openstack-4j-plugin, due to openstack-4j upstream issues with images
    - <https://github.com/openbaton/NFVO/issues/291>
    - Had to recompile openstack-4j-plugin from master
  - A few documentation issues, such as minimum\_bandwidth
  - Usually important to compile from source
- Positives
  - NSE has a small code base, only 19 Java files and < **3000** lines of code
  - Much smaller than ONAP; useful for PoC/small lab
  - Great that they have helper scripts for start/stop/compile in dev env

# OVERALL CONCLUSIONS

- Questions around how will we meet Hierarchies/Recursion requirement
- ETSI proposal has IC SDN and TC SDN, ie. multiple SDN controllers
- QoS
  - Neutron currently two QoS mechanisms - bandwidth (max/min) and DSCP
  - Tough to schedule minimum networking requirements
- Monitoring - How much jitter? How much latency? Where?
- Group Based Policy was probably ahead of its time--where is it now?
- Multi-cloud management - Regions yes, but that is not all that is needed
- If network slicing also includes SFC...anything can be done, almost too flexible
- A 3rd party needs access to a network slice, would it also need access to OpenStack APIs?

- If SFC is used, then how can OpenStack monitor multiple slices? ie. would have to include probes somehow in chain; eg. path tracer
- Orchestration - “the **continuing** process of selecting resources to fulfill client service demands in an optimal manner”
- When an SDN is in place, what exactly does OpenStack do?
- Many high level papers; few discuss application of NS
- Inter-cloud networking is challenging
- Already several requirements met...eg. private networks, APIs, etc
- Overall, lots of great places for work to improve NS support



## POTENTIAL SCENARIOS

Where **could** we go from here?

- Things are fine and while perhaps moving slowly, seem to be working; OpenStack is growing in the NFV space
- OpenStack has no real concept of NS, no need to
- OpenStack builds basic NFs and external SDN manages networking
- Something monitors KPIs, SDN provides network metrics
- Most components are outside of OpenStack
  - Eg. NFVO systems are the orchestrators
- Meeting recursion/hierarchy and QoS requirements may be too difficult, costly, or lack of development resources
- Perhaps other things for OpenStack to be focussed on (upgrades, stable versions etc)

\* Please note this is a potential scenario, not a statement of fact

- OpenStack improves/adds QoS features
- Potentially network scheduling with minimums
- Some concept of NS, perhaps involving networking-sfc or GBP; some API development in this area
- Ability to deploy network probes (ala Octavia?)
- Edge model gets moving
- Begin to accept that OpenStack's best fit may actually be in Telecoms (?)

\* Please note this is a potential scenario, not a statement of fact

- Scenario #2 plus...
- OpenStack gets a NS API of some kind; maybe own Network Slice Manager?
- QoS dialed up to 11
- Multi-cloud networking becomes first class citizen
- Online scheduling available in some form to re-deploy NFs based on usage
- Ability to somehow build sub-APIs or otherwise meet hierarchy/recursion/sharing requirements

\* Please note this is a potential scenario, not a statement of fact

- Thank you
- Bibliography available at:
  - <https://github.com/idx-labs/openstack-network-slicing/blob/master/BIBLIOGRAPHY.md>
- Email [curtis@interdynamix.com](mailto:curtis@interdynamix.com) or [corey@interdynamix.com](mailto:corey@interdynamix.com) with any questions/comments
- Special thanks to the OpenBaton team for their work
- Questions?

Interdynamix Edmonton  
620 Manulife Place  
10180 – 101 Street NW  
Edmonton, AB T5J 3S4  
780.423.7005

Interdynamix Toronto  
140 Yonge St.  
Suite 200  
Toronto, ON M5C 1X6