SSY145 – Guest lecture on Network Slicing

Federico Tonini

CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni

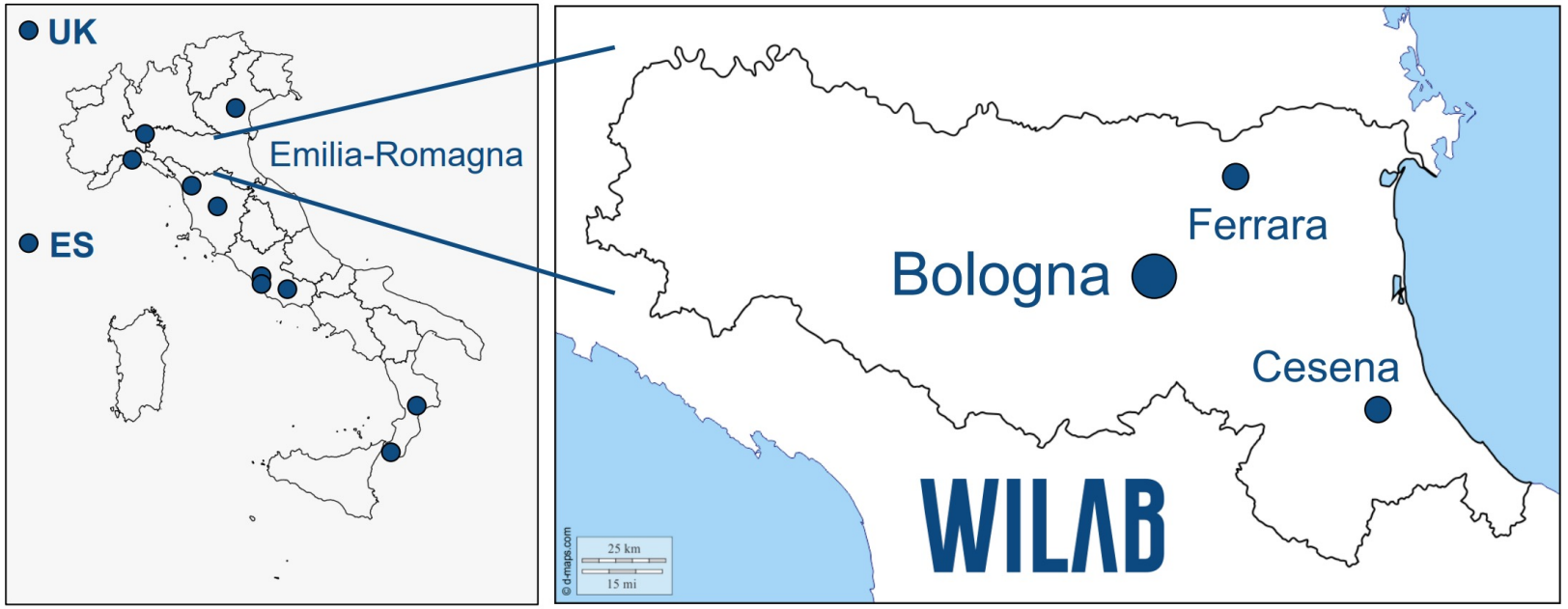
federico.tonini@wilab.cnit.it

CNIT

- CNIT (National Inter-University Consortium for Telecommunications) is a non-profit entity funded in 1995 and recognized by the MIUR that performs research, innovation and educational activities in the ICT sector
- CNIT has 7 national laboratories
  1) National Laboratory of Multimedia Communications, Naples
  2) National Laboratory of Advanced Optical Fibers for Photonics, L'Aquila
  3) National Laboratory of Radar and Surveillance Systems, Pisa
  4) Photonic Networks & Technologies National Laboratory, Pisa
  5) National Laboratory of Smart and Secure Networks, Genoa
  6) National Laboratory of Network Assessment, Assurance and Monitoring, Rome
  7) **National Laboratory of Wireless Communications, Federated (Italy)**

As of today – Associates

- Universities of Bologna, Ferrara, Genova, Padova, Siena, Pisa, Roma Tor Vergata, Roma La Sapienza, Calabria, Mediterranea, Cassino, Pavia
- Imperial College London (UK), University of Malaga (ES)



Headquarters: Via Paolo Nanni Costa 20, Bologna

As of today – WiLab Members

1 Emeritus affiliate

5 Affiliates

17 Employees

61 Associates

5 Collaborators

89 WiLab members *(Professors, University Researchers, Ph.D./M.Sc/B.Sc Students)*

WiLab activities

WiLab pursues both frontier research and activities geared towards proof-of-concepts.

Topics:

- Industrial and Indoor IoT
- Urban and Rural IoT
- Wireless Communications for Connected Vehicles
- Mobile Radio Network Architectures & Optimisation
- Joint Localisation and Communication
- Applied AI for Wireless Communications

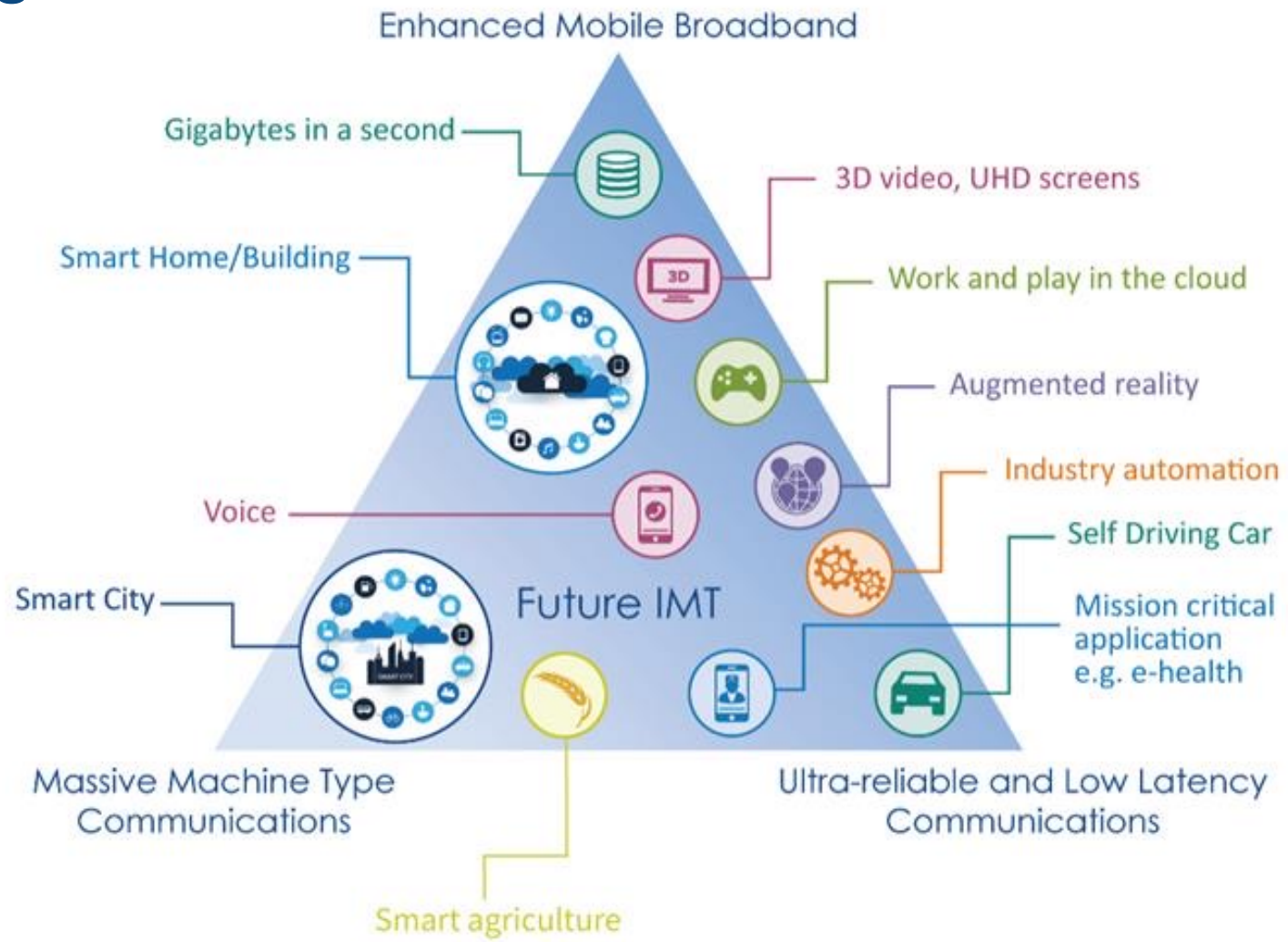
Outline of the lecture

- 5G Network slicing
  - Overview and definition
  - Involved resources and standardization
  - Slice creation process
- Examples
  - Slice deployment in the cloud
  - Reliable resource provisioning
- A quick look at 6G

Outline of the lecture

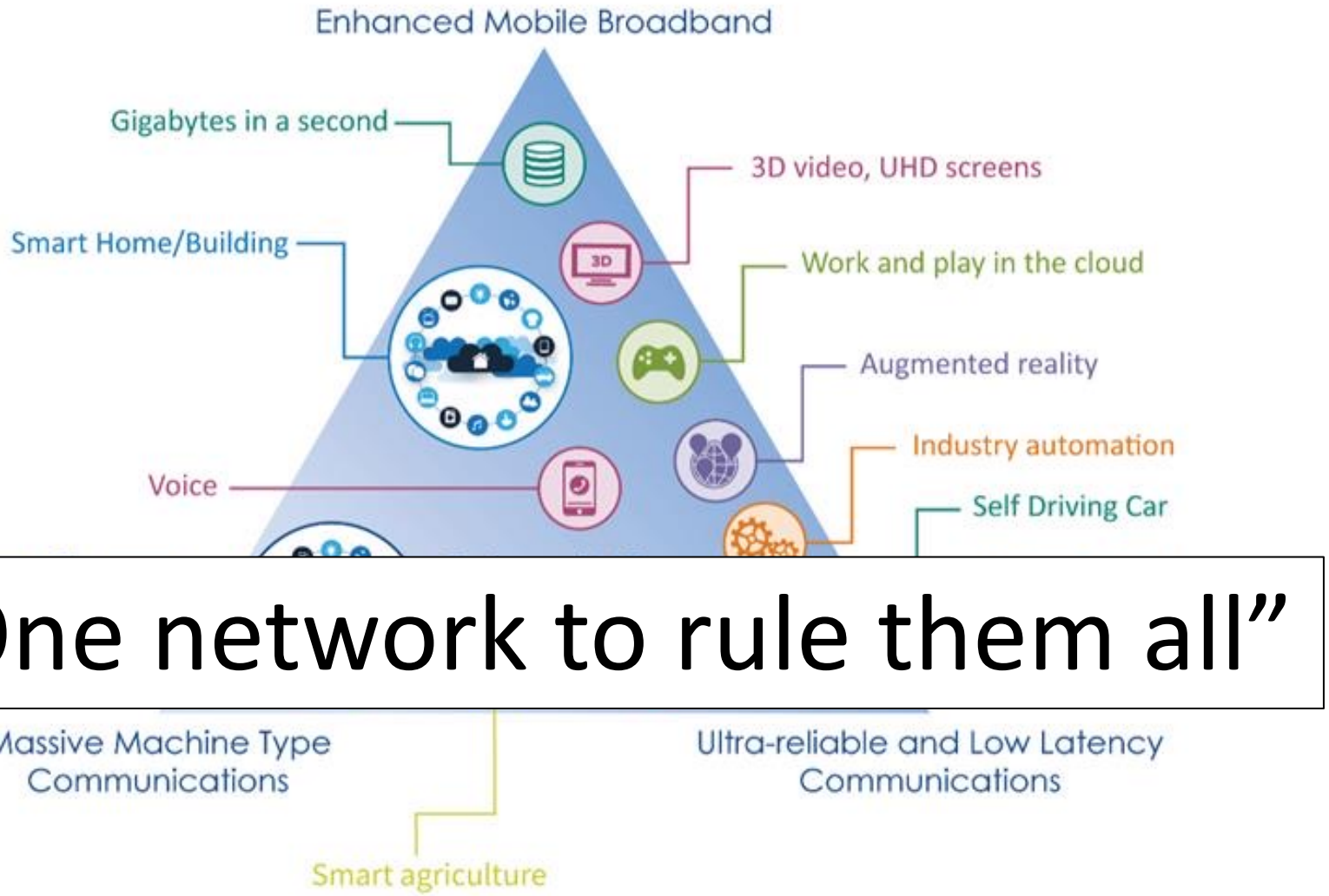
- 5G Network slicing
  - Overview and definition
  - Involved resources and standardization
  - Slice creation process
- Examples
  - Slice deployment in the cloud
  - Reliable resource provisioning
- A quick look at 6G

5G services



The 5G triangle; image credit: ETSI

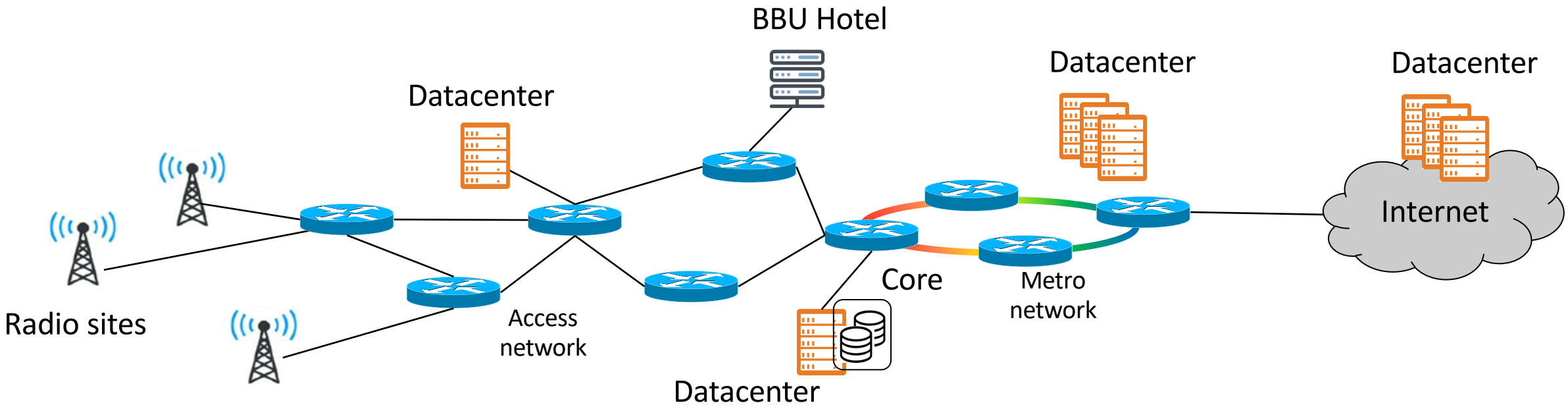
5G services



"One network to rule them all"

The 5G triangle; image credit: ETSI

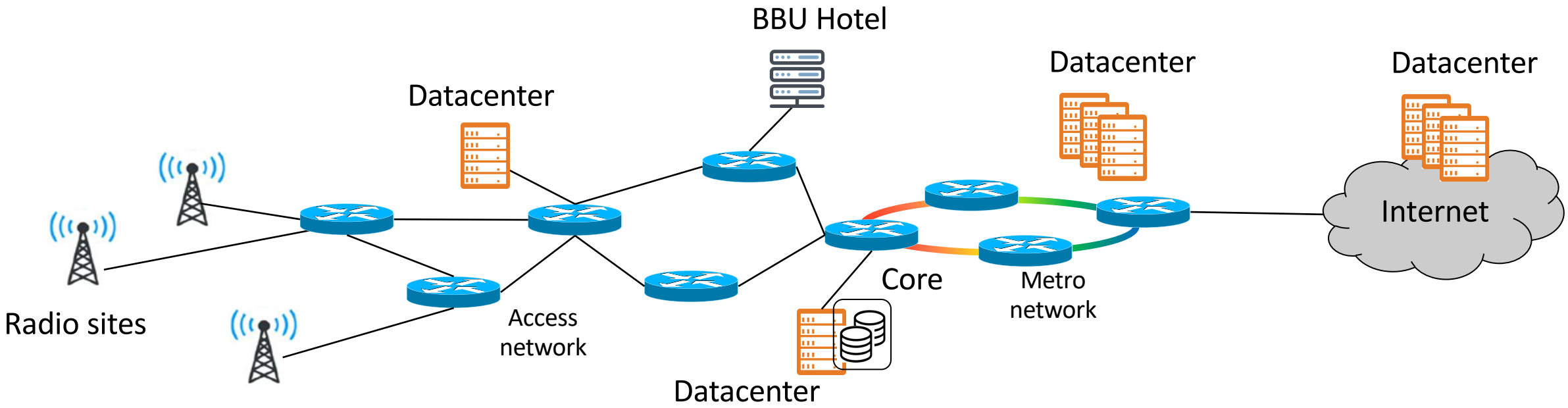
Mobile network domains



Mobile network domains

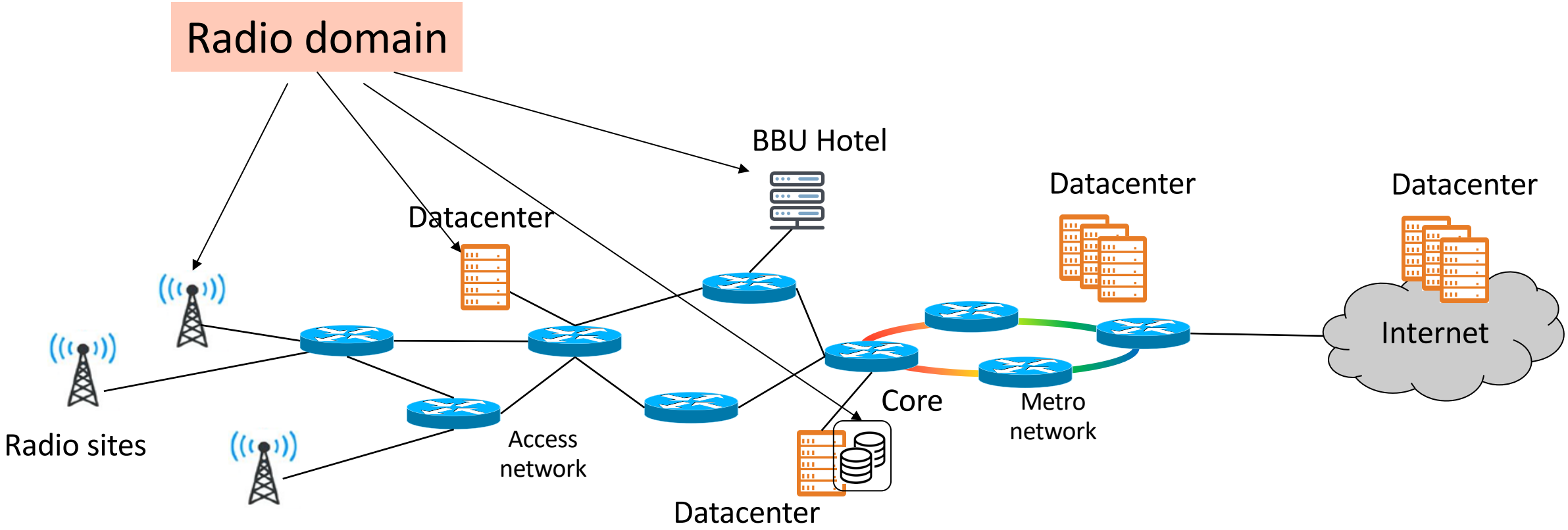
Radio domain

Cloud domain

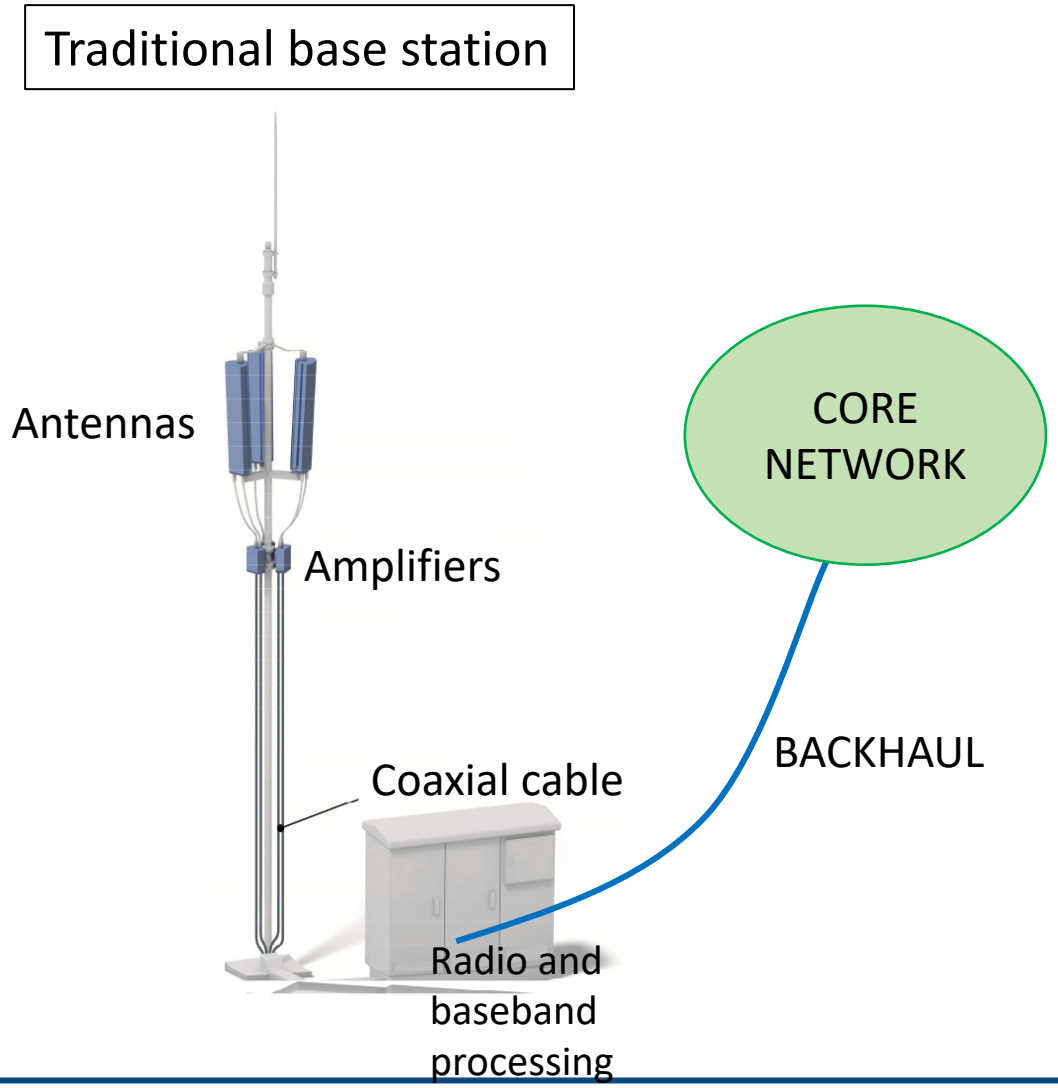


Transport domain

Mobile network domains



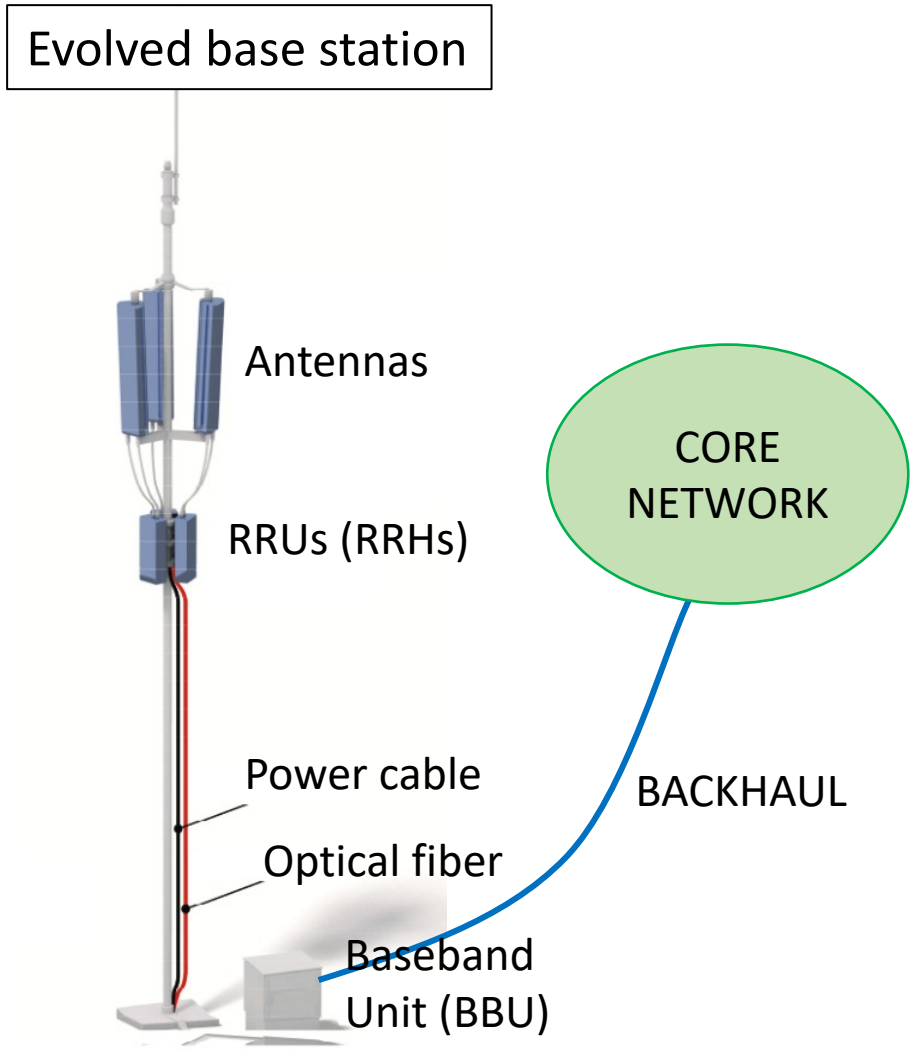
Traditional base station



Source: Phoenix Contact

- In a traditional BS, antennas are connected to radio and baseband processing functions through a coax. Cable
- The base station (BS) is connected to the core network through a fiber link (backhaul)

Evolved base station

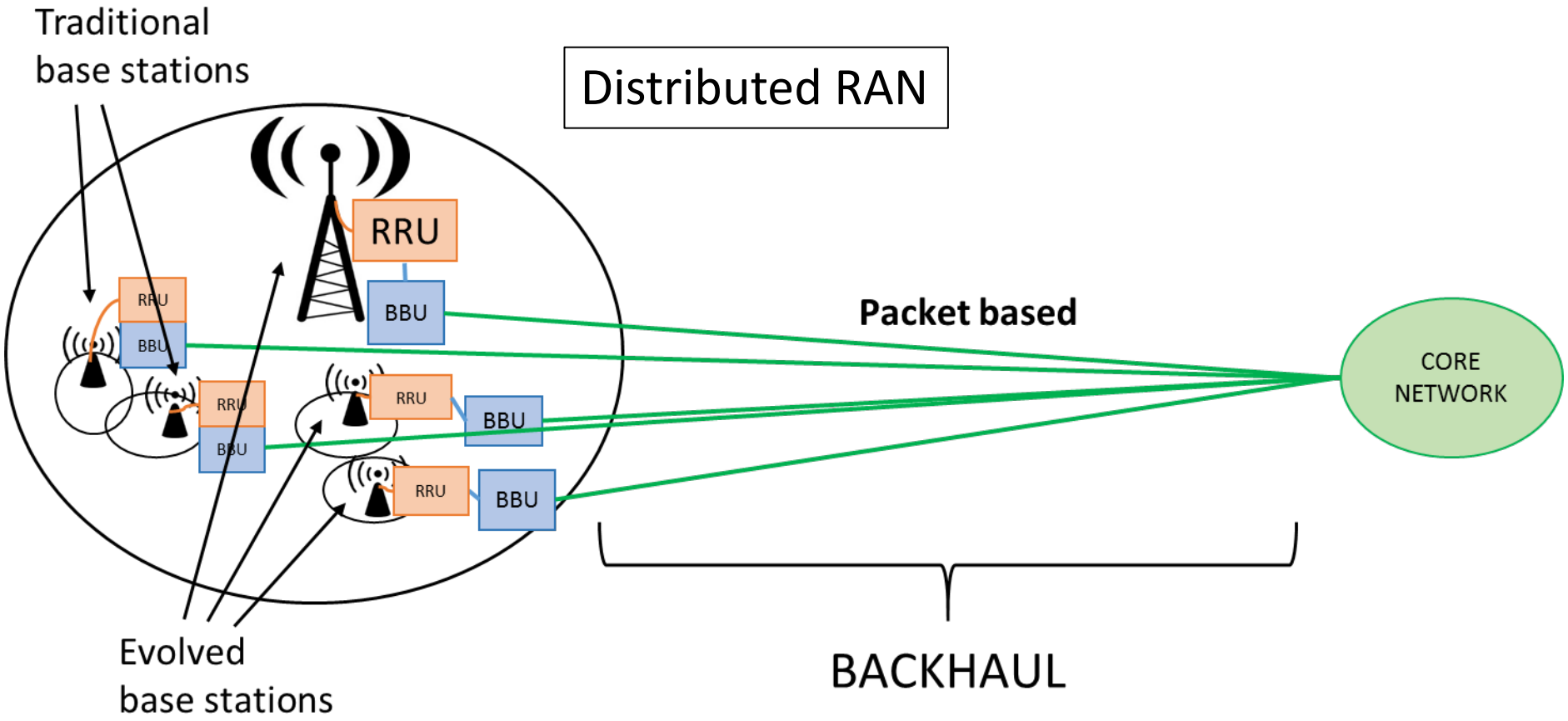


Source: Phoenix Contact

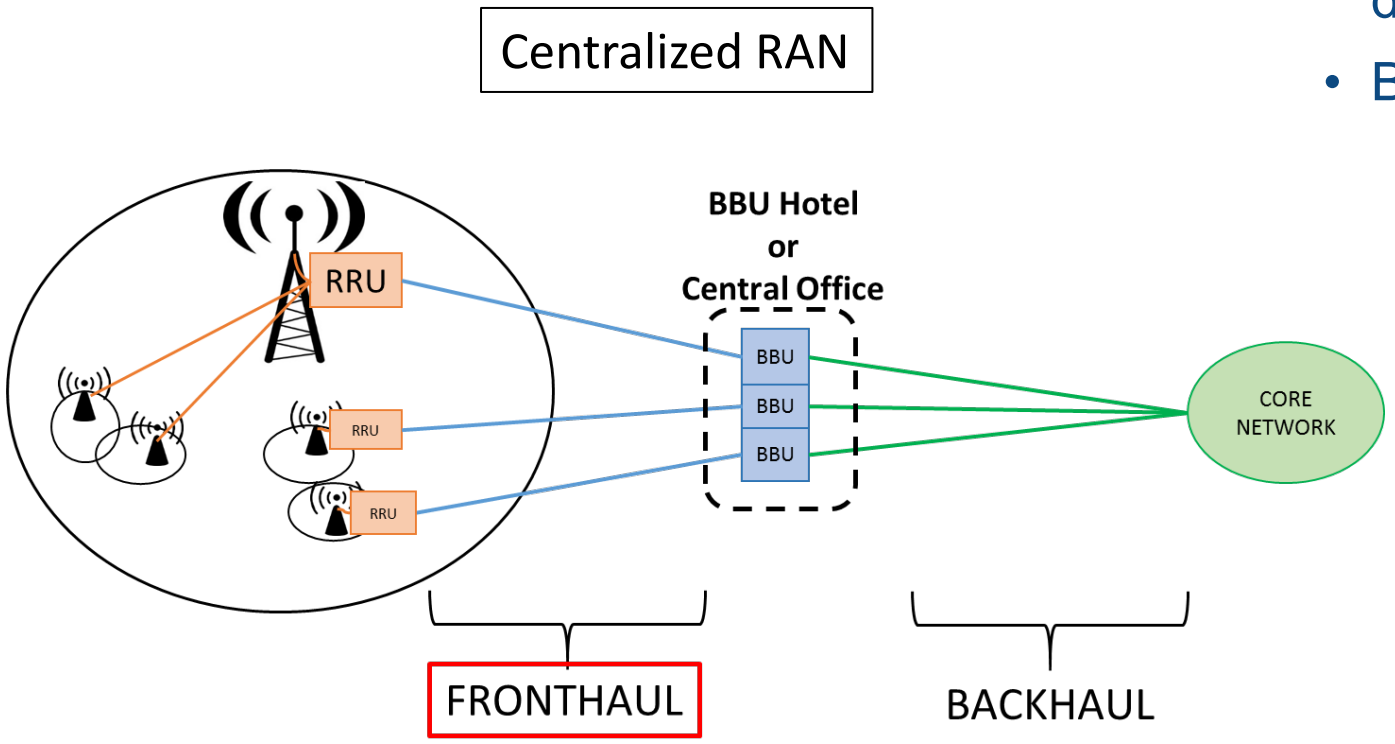
- Remote radio units (RRUs) are decoupled from baseband units (BBUs)
- RRUs performs power amplification, filtering, and DAC/ADC conversion
- BBUs performs digital baseband signal processing (e.g., modulation and coding) and upper layer functions
- In a RRU/BBU paradigm RRUs are placed close to antennas and connected to BBUs through fibers
- BBUs are connected to the core network through fiber cables (backhaul)

Distributed Radio Access Network (DRAN)

- A RAN can be distributed or centralized, depending on the BBUs locations

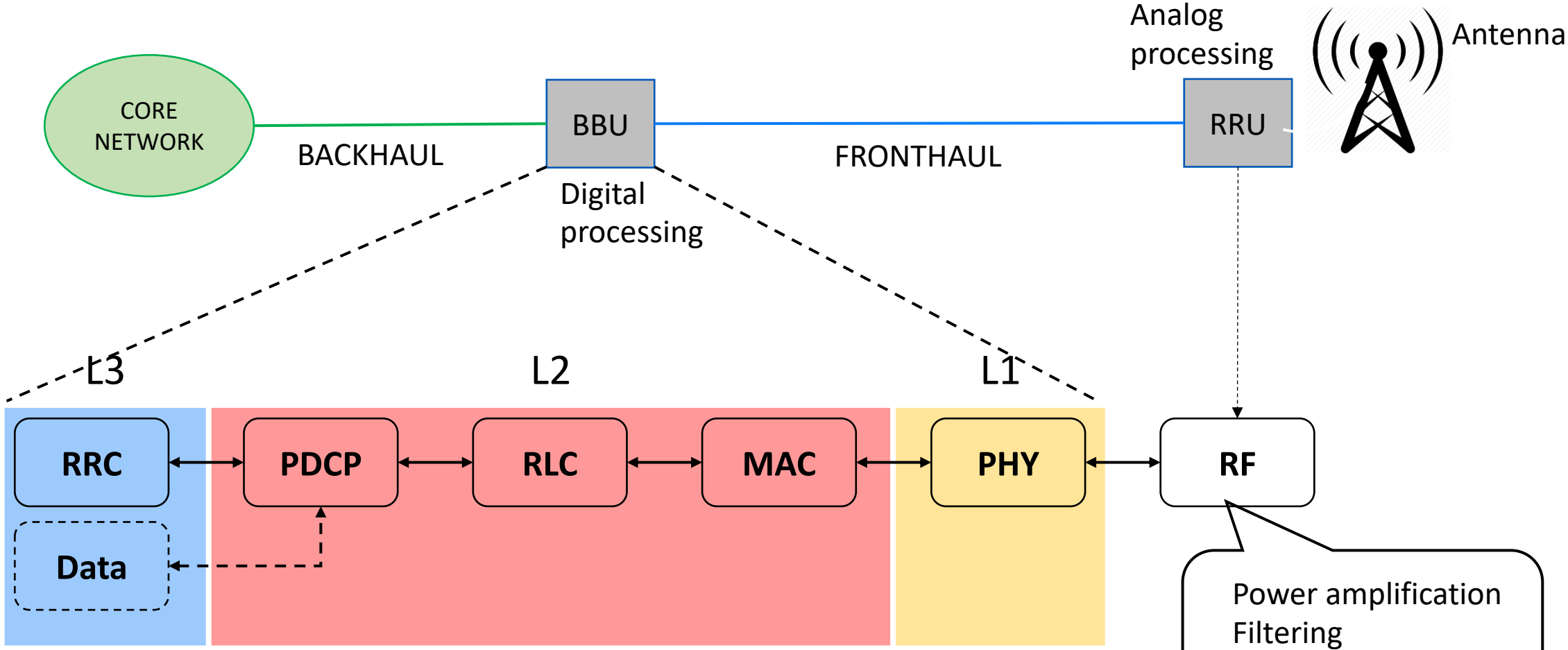


Centralized Radio Access Network (CRAN)



- A RAN can be distributed or centralized, depending on the BBUs locations
- Baseband centralization:
  - \+ Pooling gains
  - \+ Efficient coordination for interference management
  - \+ Simplified network deployment, maintanance and updates
  - \+ Software virtualization and management
  - \- Strict BW, delay and jitter requirements on fronthaul

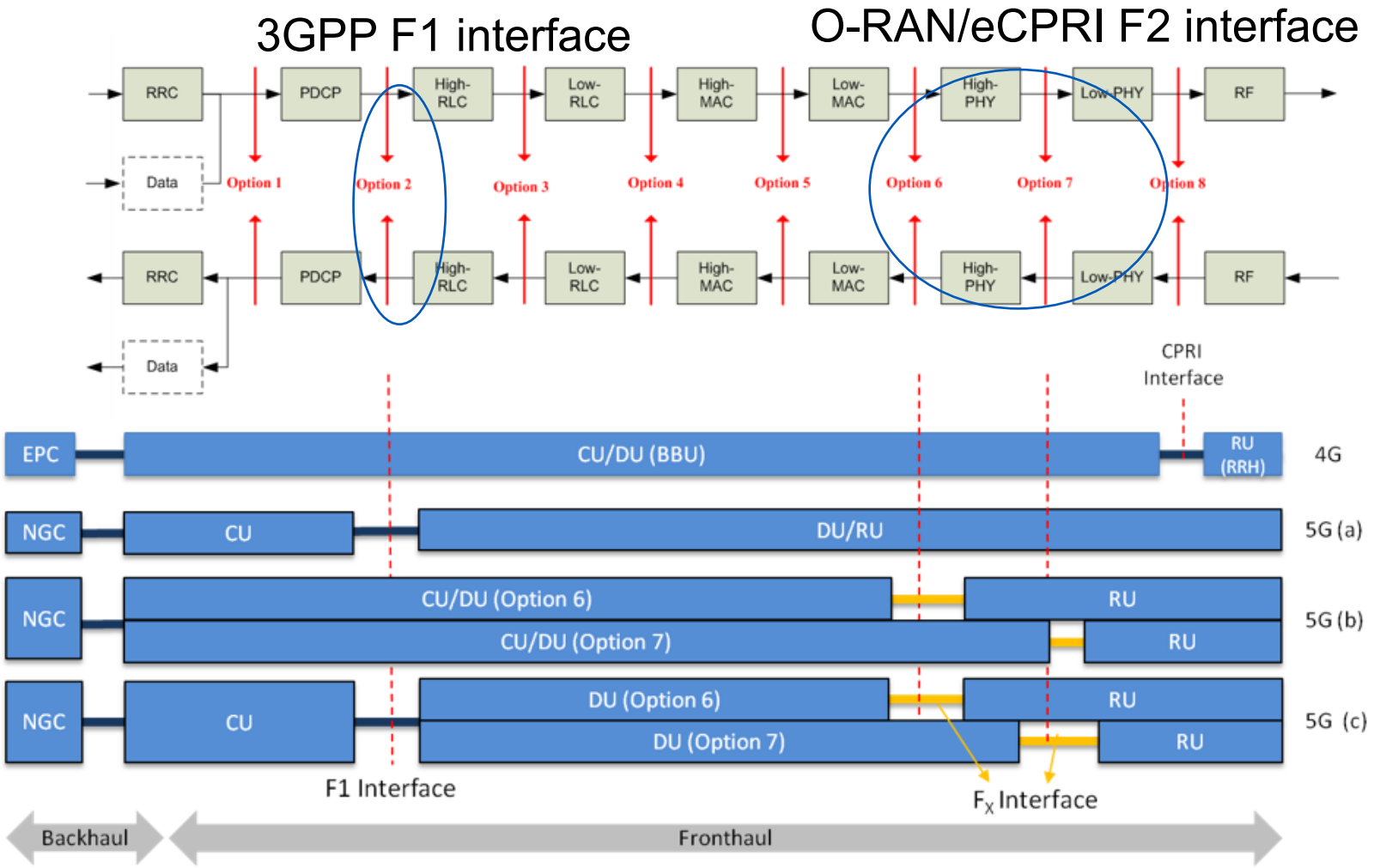
Fronthaul evolution



RRC: radio resource control
PDCP: packet data convergence protocol
RLC: radio link control

MAC: medium access control
PHY: physical layer
RF: radio frequency

5G baseband functional splits



Source: ITU-T, Transport network support of IMT-2020/5G, 2018

Functional splits requirements (example)

		Option 1	Option 2	Option 3	Option 4	Option 5	Option 6	Option 7a	Option 7b	Option 7c	Option 8
Required bandwidth	DL	4 Gbps	4016 Mbps	<Option 2	4 Gbps	4 Gbps	4133 Mbps	10.1÷22.2 Gbps	37.8÷86.1 Gbps	10.1÷22.2 Gbps	157.3 Gbps
	UL	3 Gbps	3024 Mbps	<Option 2	3 Gbps	3 Gbps	5640 Mbps	13.6÷21.6 Gbps	53.8÷86.1 Gbps	53.8÷86.1 Gbps	157.3 Gbps
Max. Allowed one way latency		10 ms	1.5÷10ms	1.5÷10ms	~100us	hundreds of us	250 us	250 us	250 us	250 us	250 us

Source: 3GPP TR 38.801 V14.0.0 (2017-05)

O-RAN standardization efforts

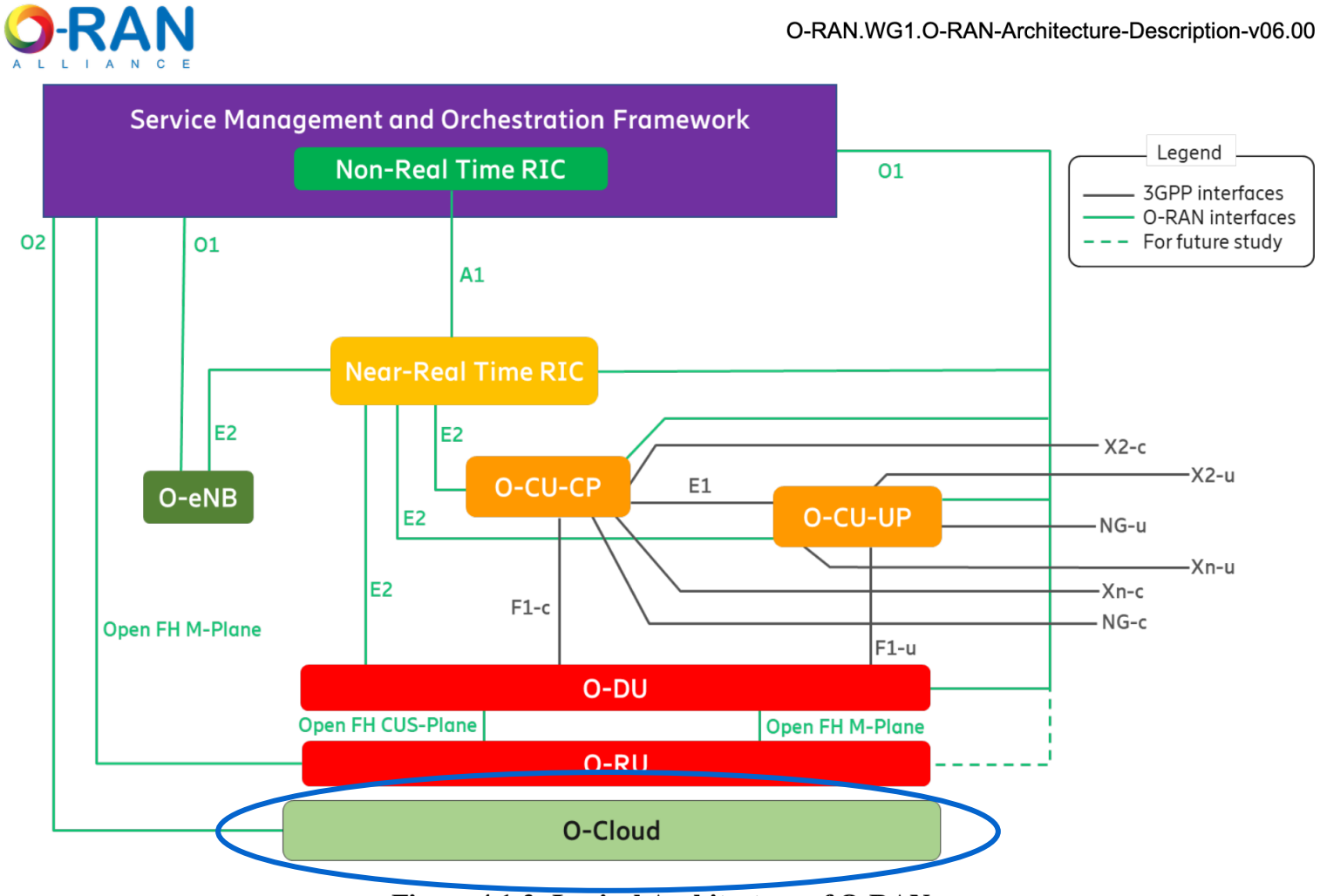
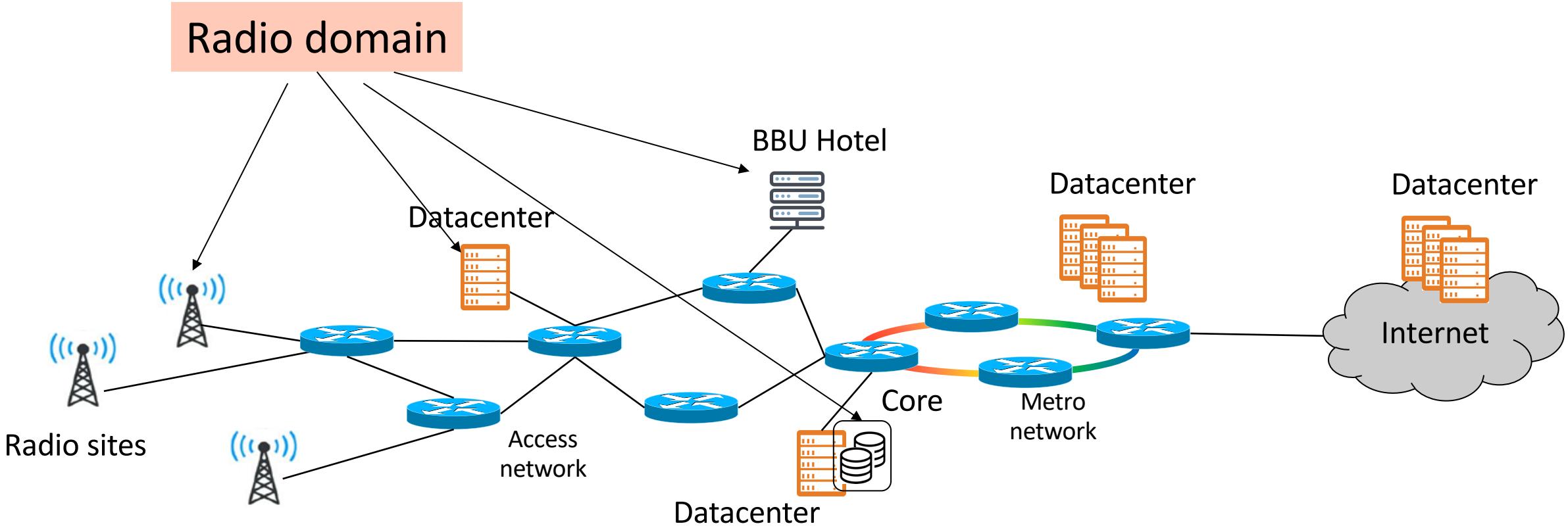
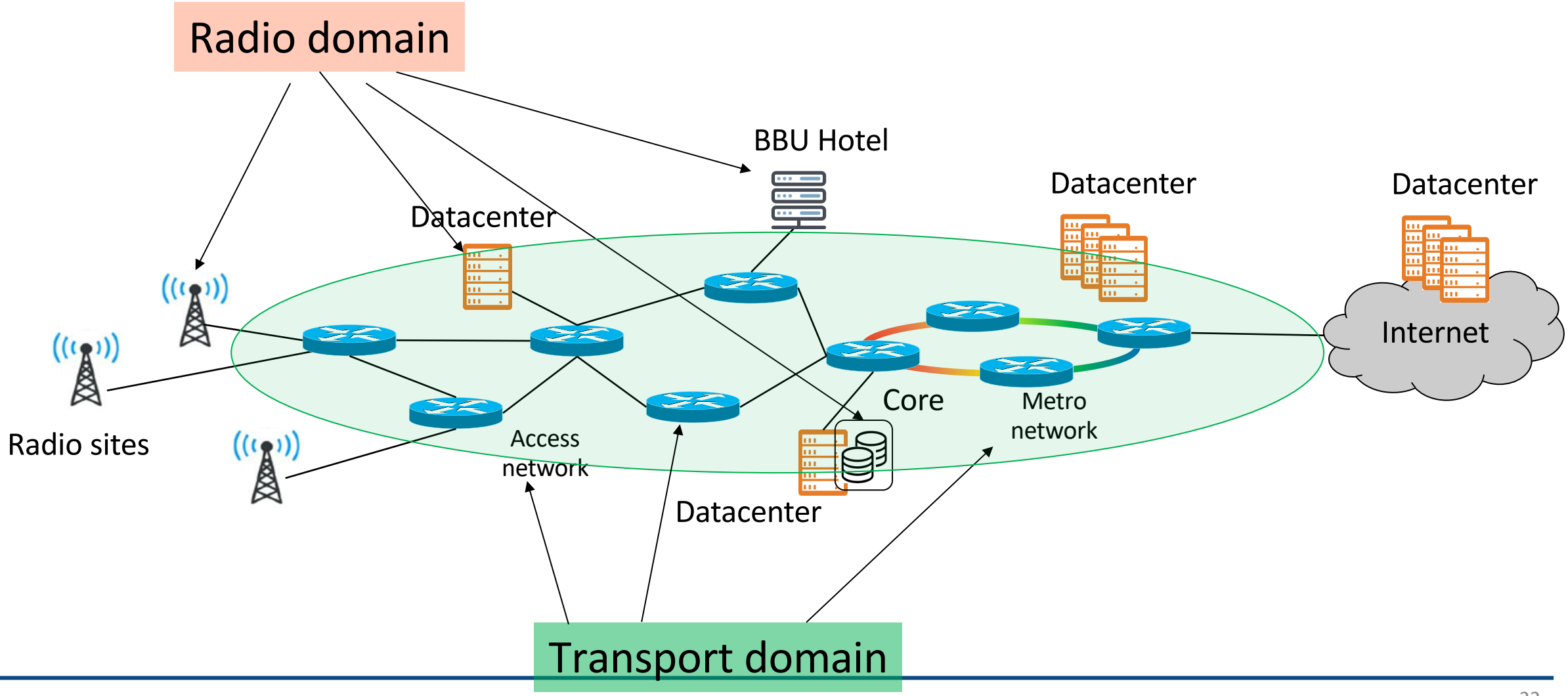


Figure 4.1-2: Logical Architecture of O-RAN

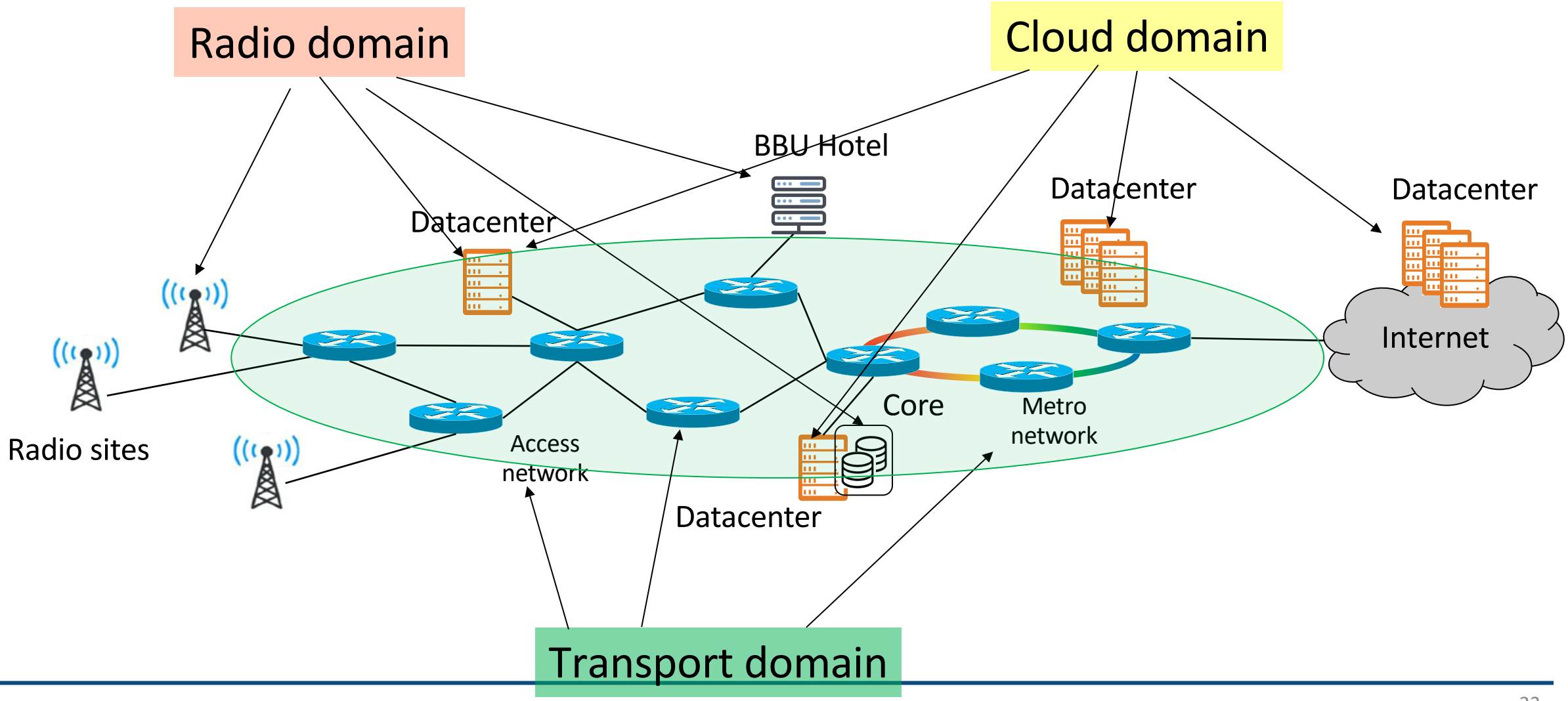
Mobile network domains



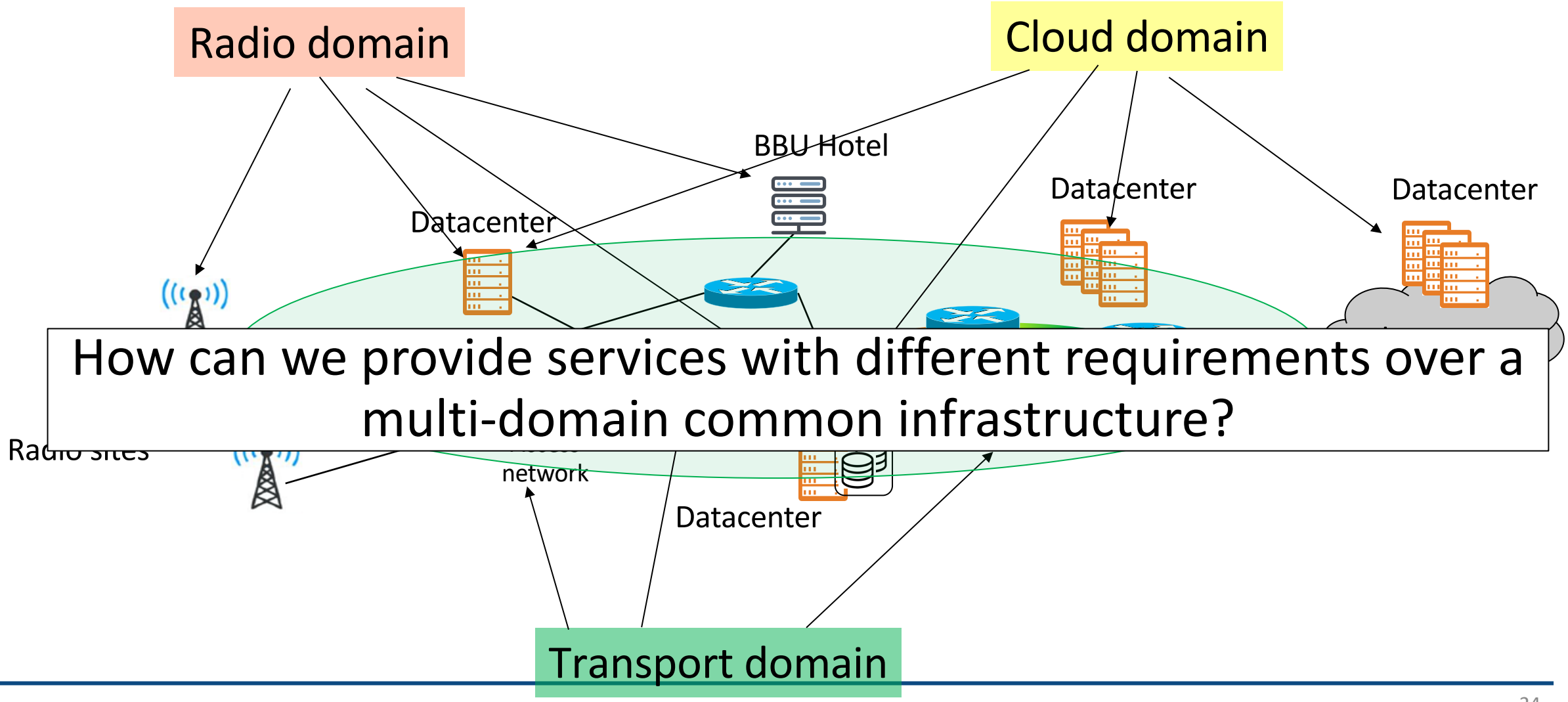
Mobile network domains



Mobile network domains

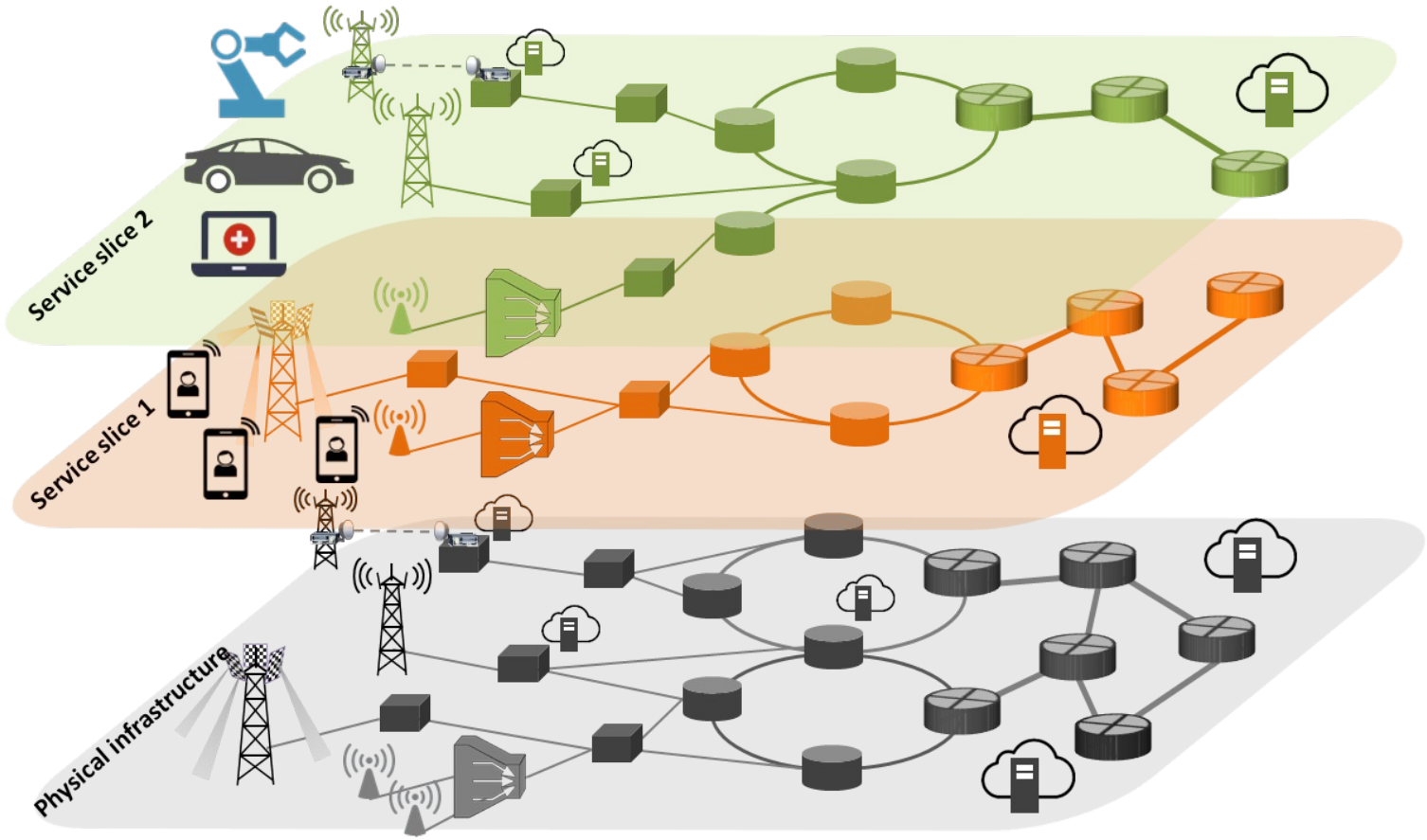


Mobile network domains



Network slicing

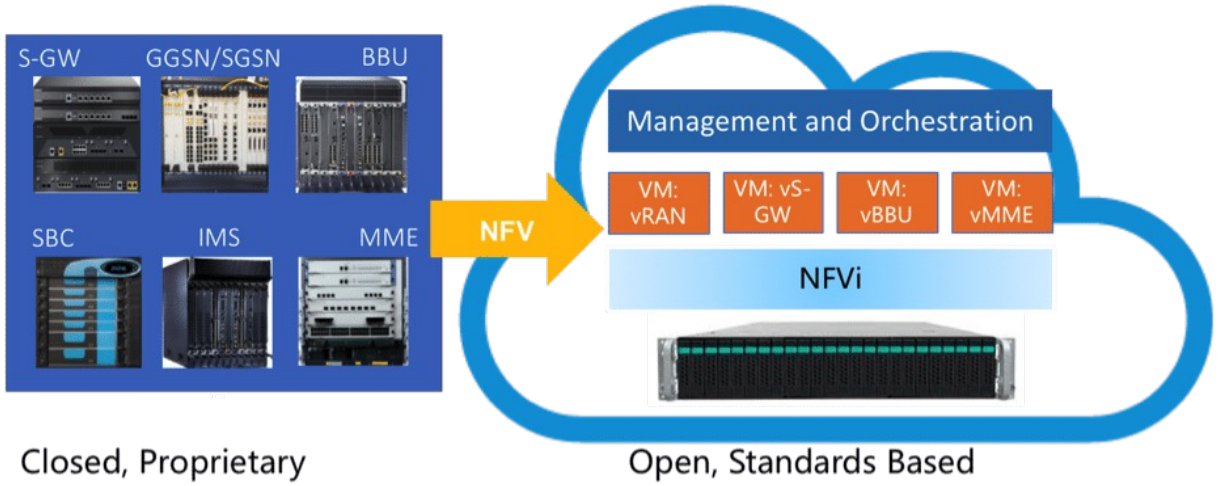
- A **network slice** is an end-to-end logical network that runs on a shared physical infrastructure, capable of providing a negotiated service quality
- A slice should be:
  - programmable
  - adaptive
  - reliable
  - secure
- We need “tools” to bring flexibility in the network



Definition: GSMA, An Introduction to Network Slicing, 2017

Enabling technologies: Network Function Virtualization (NFV)

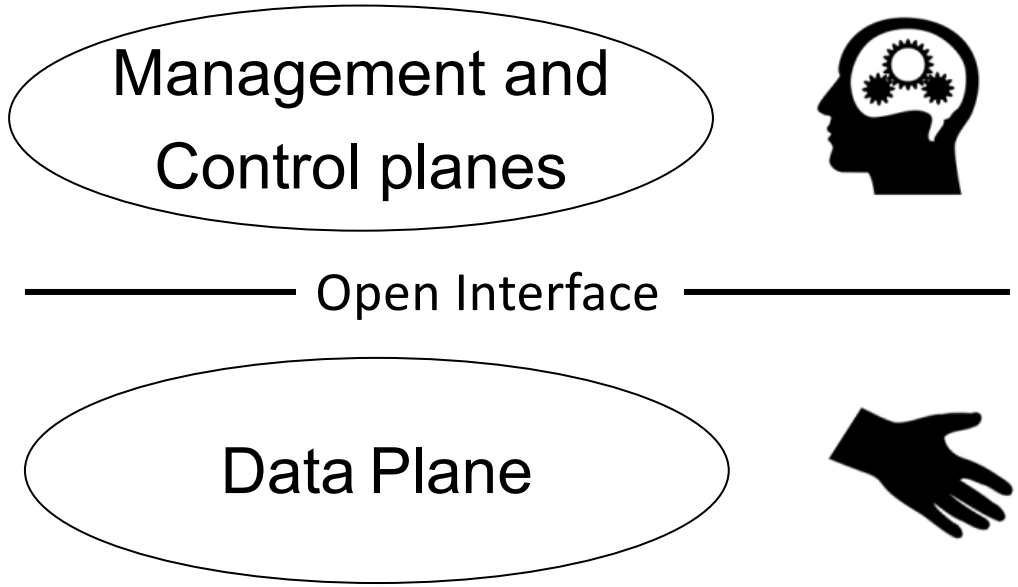
- Decouples physical network resources from the functions that run on them
- Decomposes each service into a set of Virtual Network Functions (VNFs)
  - implemented via software running on general purpose hardware
- The VNFs can be instantiated, moved, terminated at runtime in different locations
- However, VNFs must be linked to form a chain…



Source: https://www.rcrwireless.com/20180910/network-function-virtualization-nfv/nfv-virtualized-acceleration-offloads

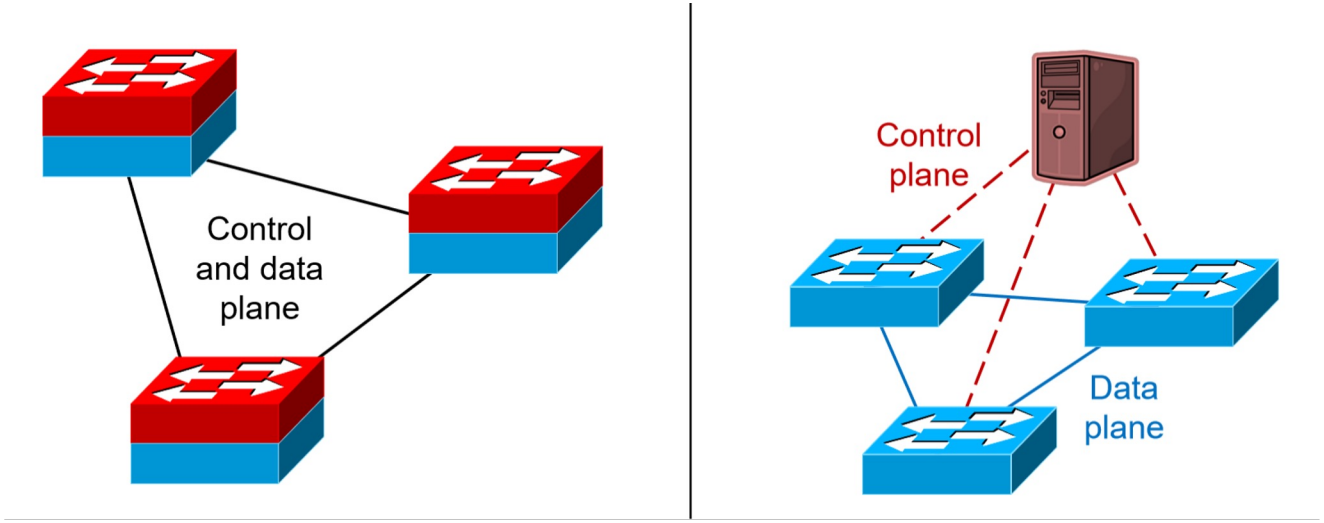
Enabling technologies: Software Defined Networking (SDN)

- Decouples control and data plane
- Network intelligence is logically centralized
- Makes the transport network programmable

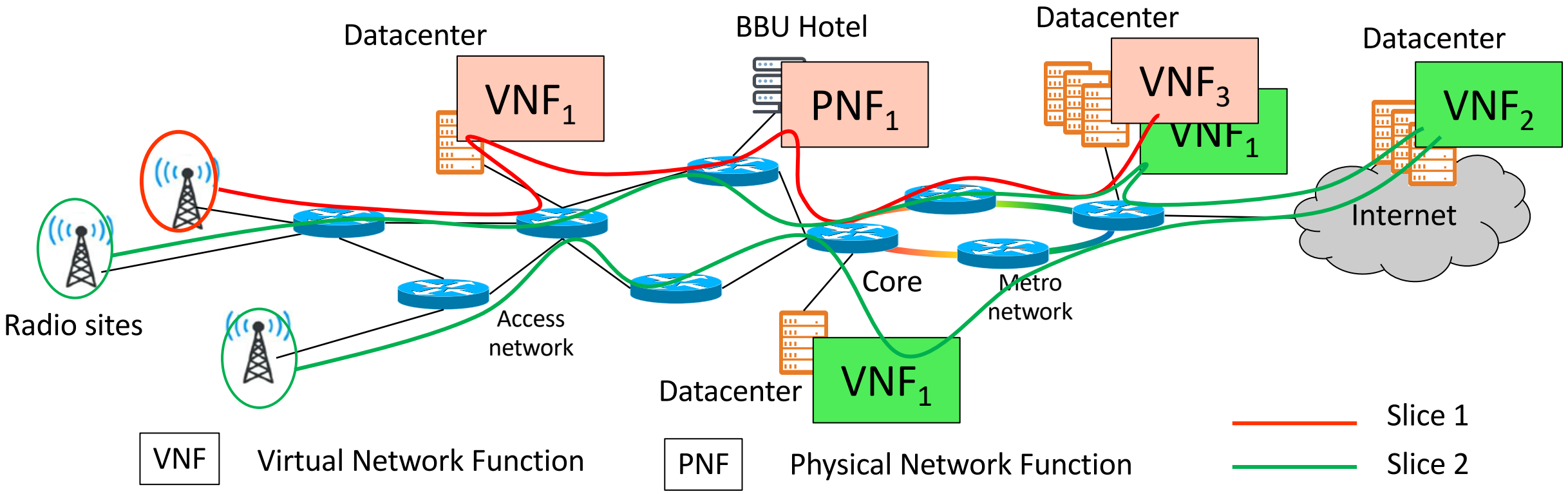


Enabling technologies: Software Defined Networking (SDN)

- Decouples control and data plane
- Network intelligence is logically centralized
- Makes the transport network programmable

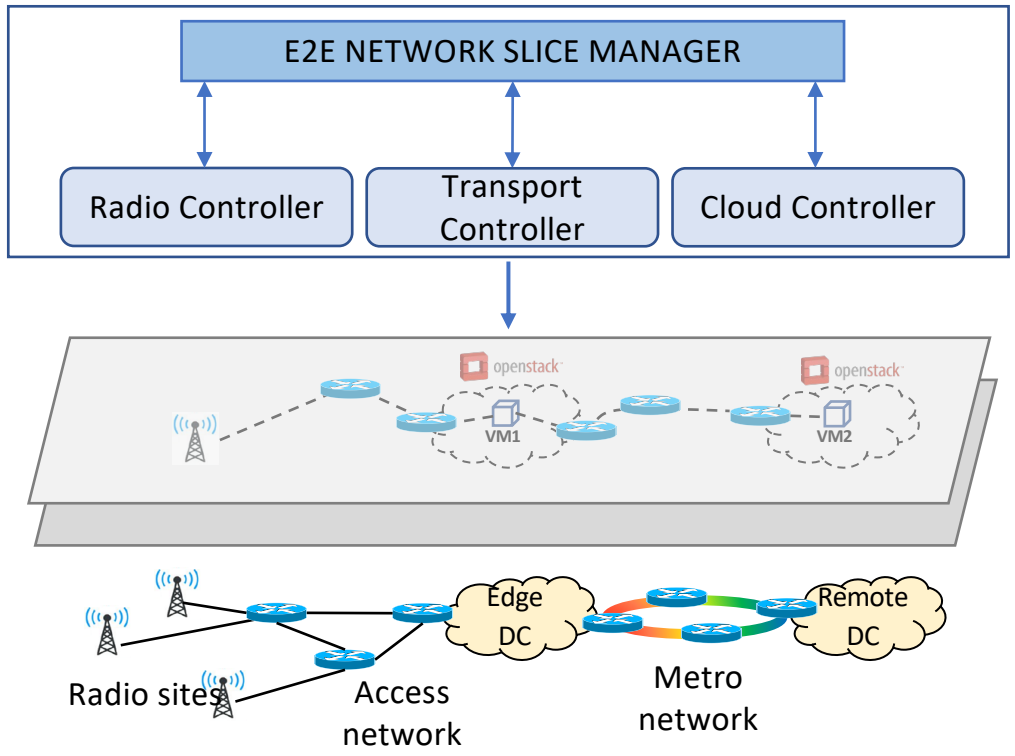


Chain of virtual/physical functions



Slice orchestration

- Controllers are in charge of handling resources within their own domain
- Slices span over multiple administrative and technological domains
  - E2E management and orchestration is required

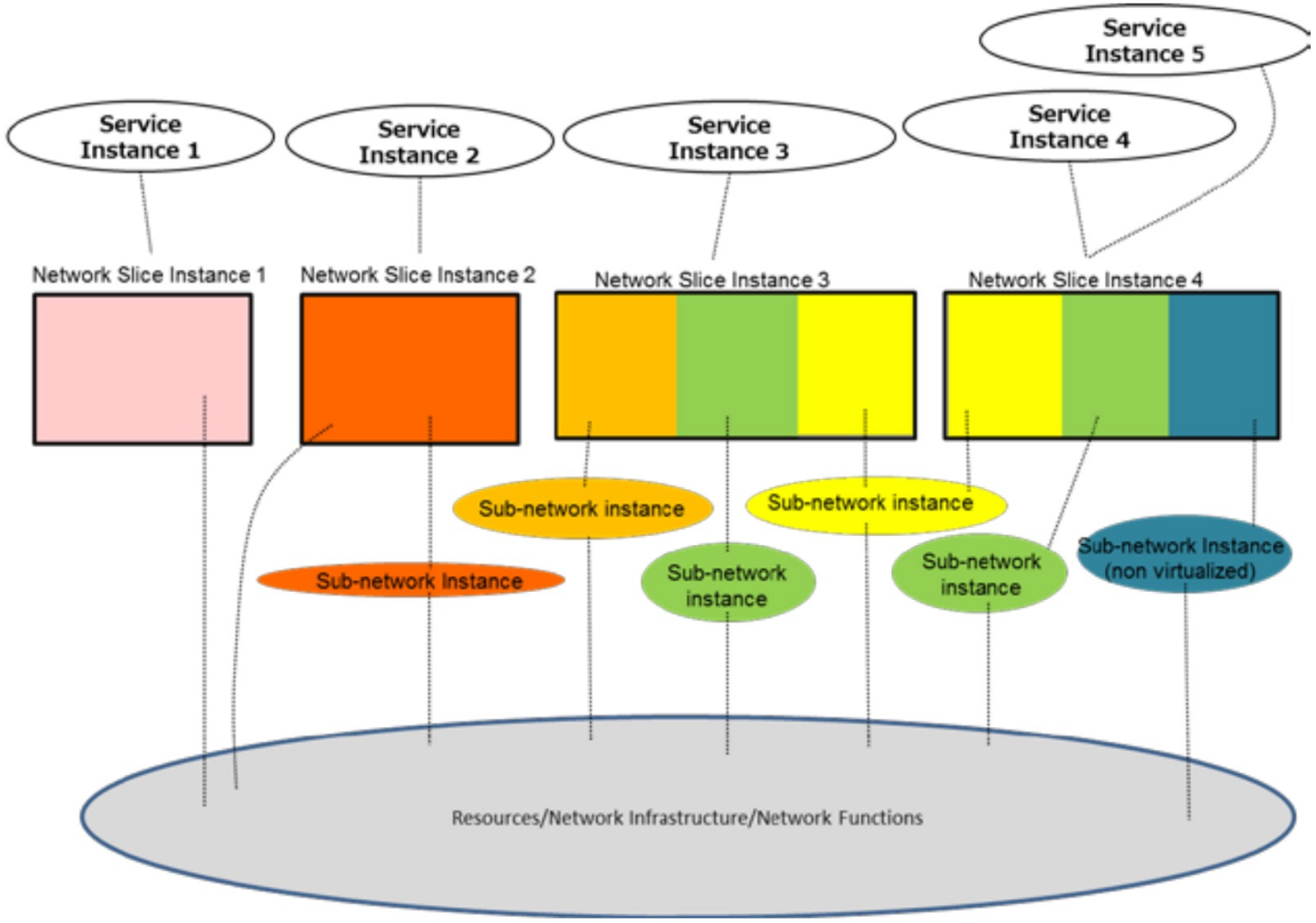


Main standardization bodies and industry fora involved

- E2E network slice implementations have to be aligned with the standards
  - Unfortunately, there is no E2E network slicing model available
- Several domains involved -> lots of bodies that need to provide a common way to interoperate (and the proper “means” to support this (e.g., interfaces, abstraction))

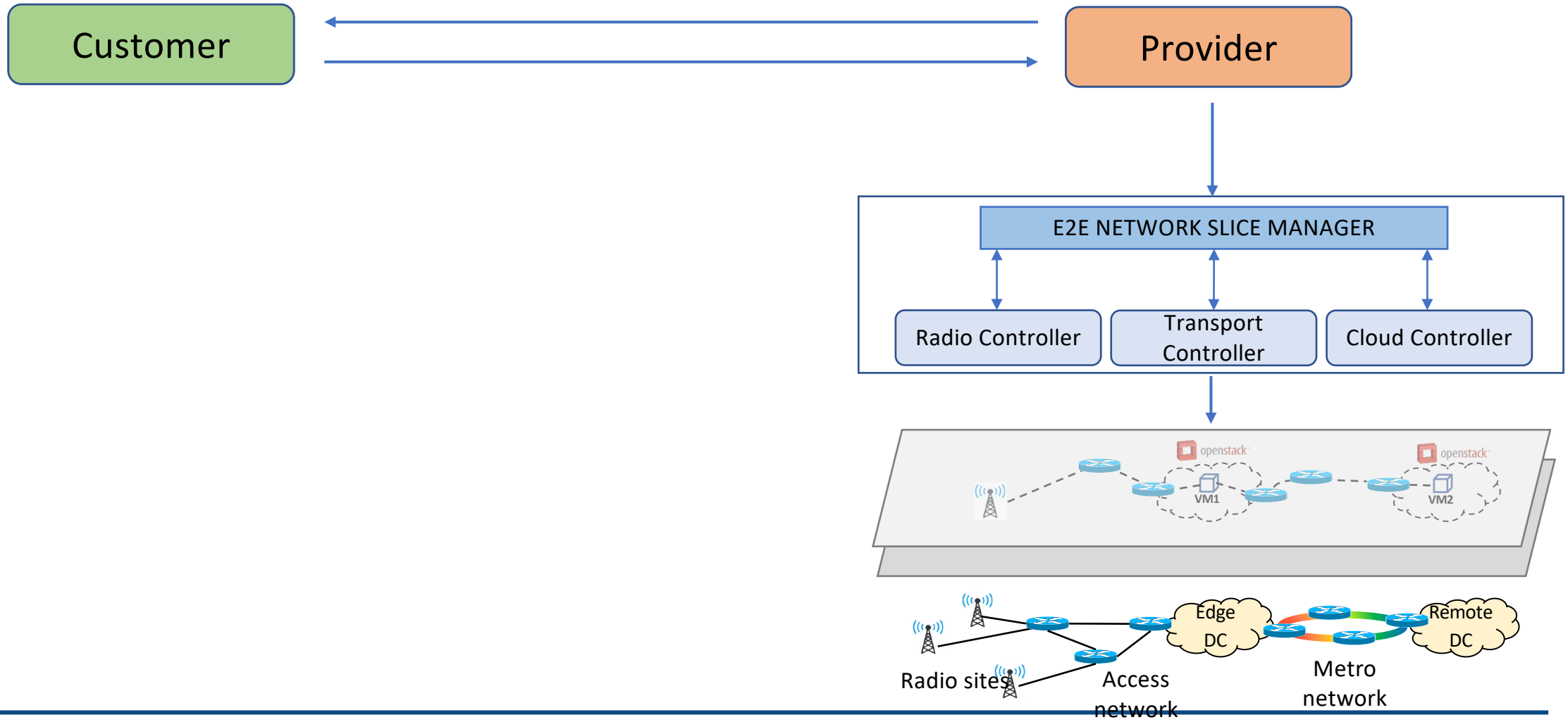
M. Chahbar, G. Diaz, A. Dandoush, C. Cérin and K. Ghoumid, "A Comprehensive Survey on the E2E 5G Network Slicing Model," in *IEEE TNSM*, 2021.

Very abstracted view of network slicing

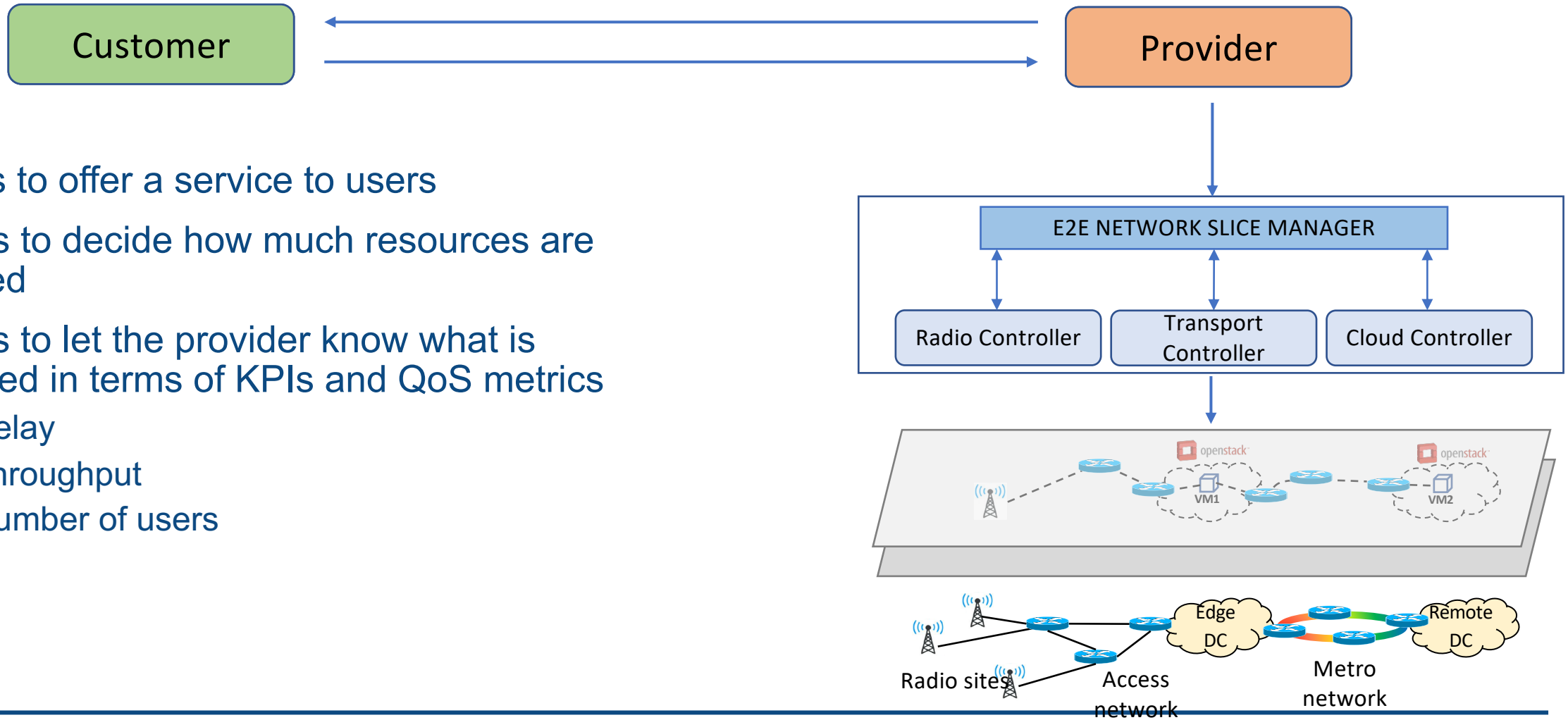


Source: NGMN

Slice as a service



Slice as a service



- Wants to offer a service to users
- Needs to decide how much resources are needed
- Needs to let the provider know what is required in terms of KPIs and QoS metrics
  - Delay
  - Throughput
  - Number of users
  - …

Slice as a service

- Wants to offer a service to users
- Needs to decide how much resources are needed
- Needs to let the provider know what is required in terms of KPIs and QoS metrics
  - Delay
  - Throughput
  - Number of users
  - …
- Needs to decide which resources are most suitable based on
  - Requirements (SLA)
  - Other optimization criteria (e.g., cost)
- Needs to reserve resources – the actual slicing

Slice as a service

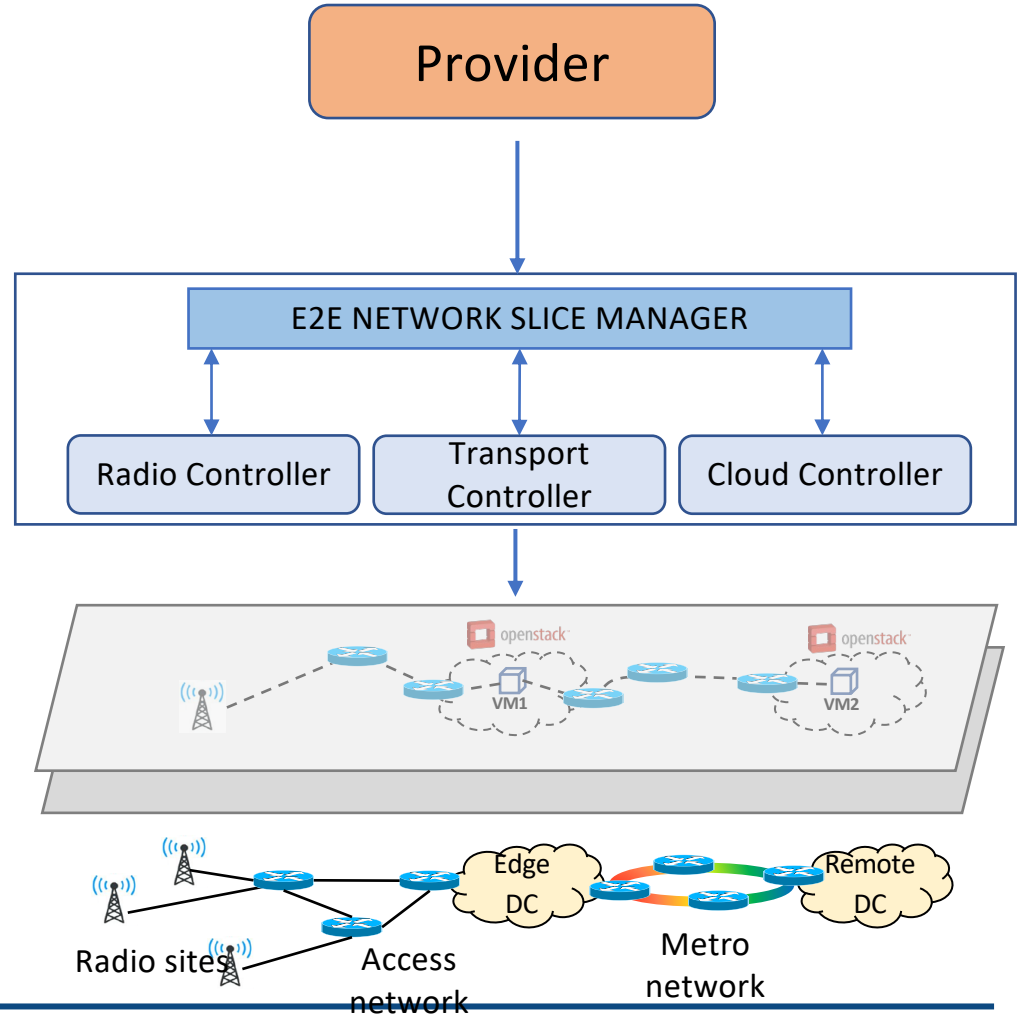
- Wants to offer a service to users
- Needs to decide how much resources are needed
- Needs to let the provider know what is required in terms of KPIs and QoS metrics
  - Delay
  - Throughput
  - Number of users
  - …
- Needs to decide which resources are most suitable based on
  - Requirements (SLA)
  - Other optimization criteria (e.g., cost)
- Needs to reserve resources – the actual slicing

This process must be standardized!

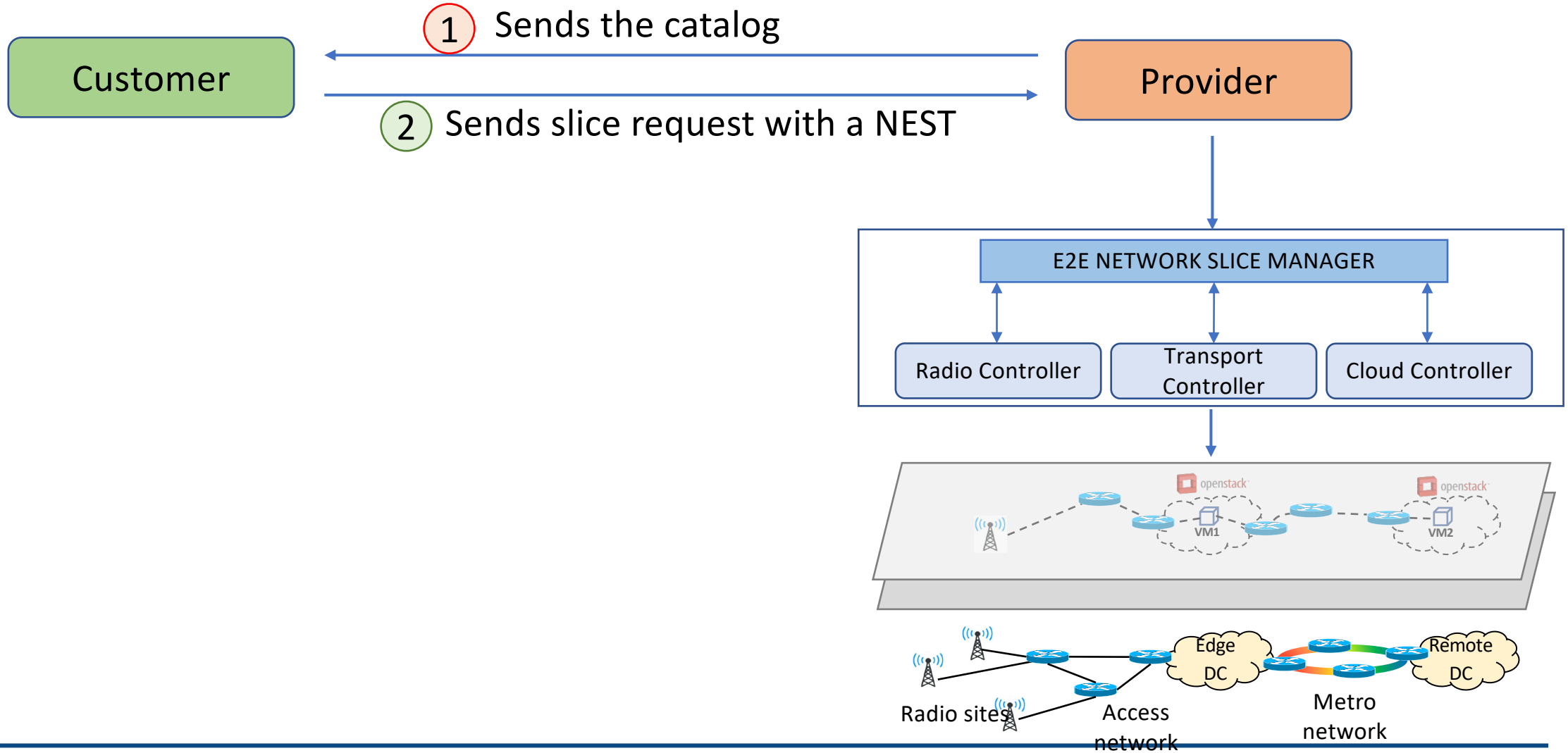
How do we describe a slice?

Customer

- “high level”, depending on service requirements or target performance (e.g., user data rate, latency, …), for the interaction with customers
- “low level”, for the software implementation of the slice
- Let’s take a look at the GSMA approach

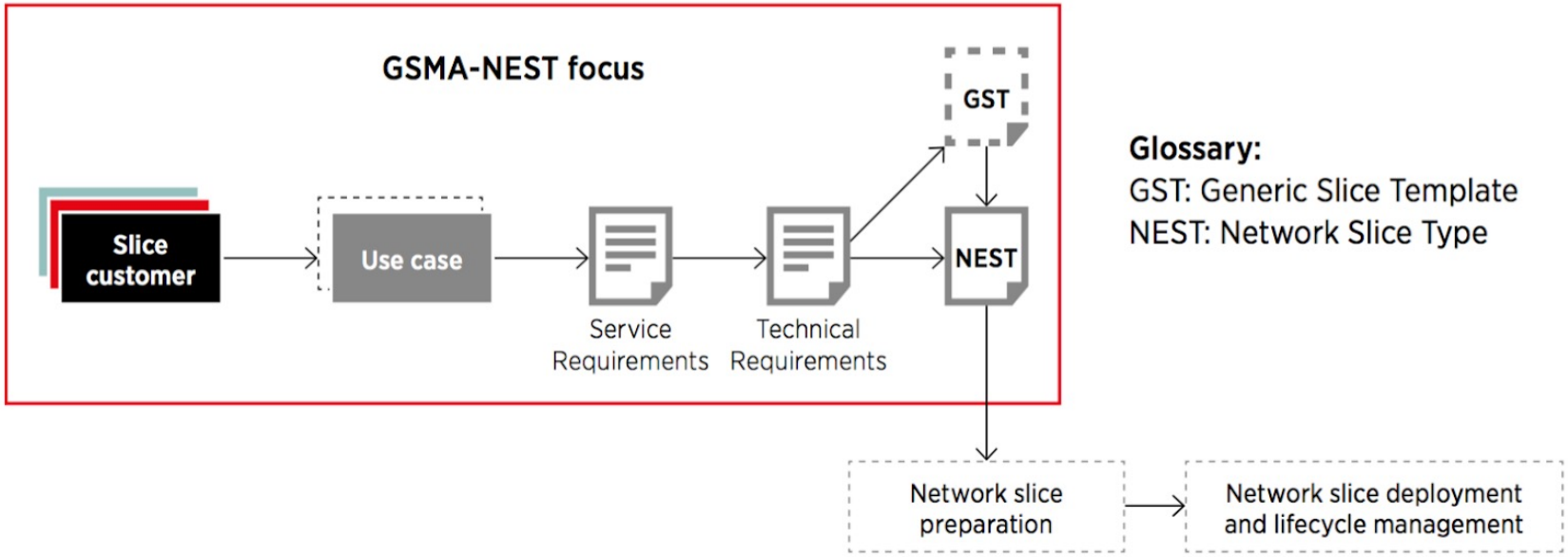


Slice creation process



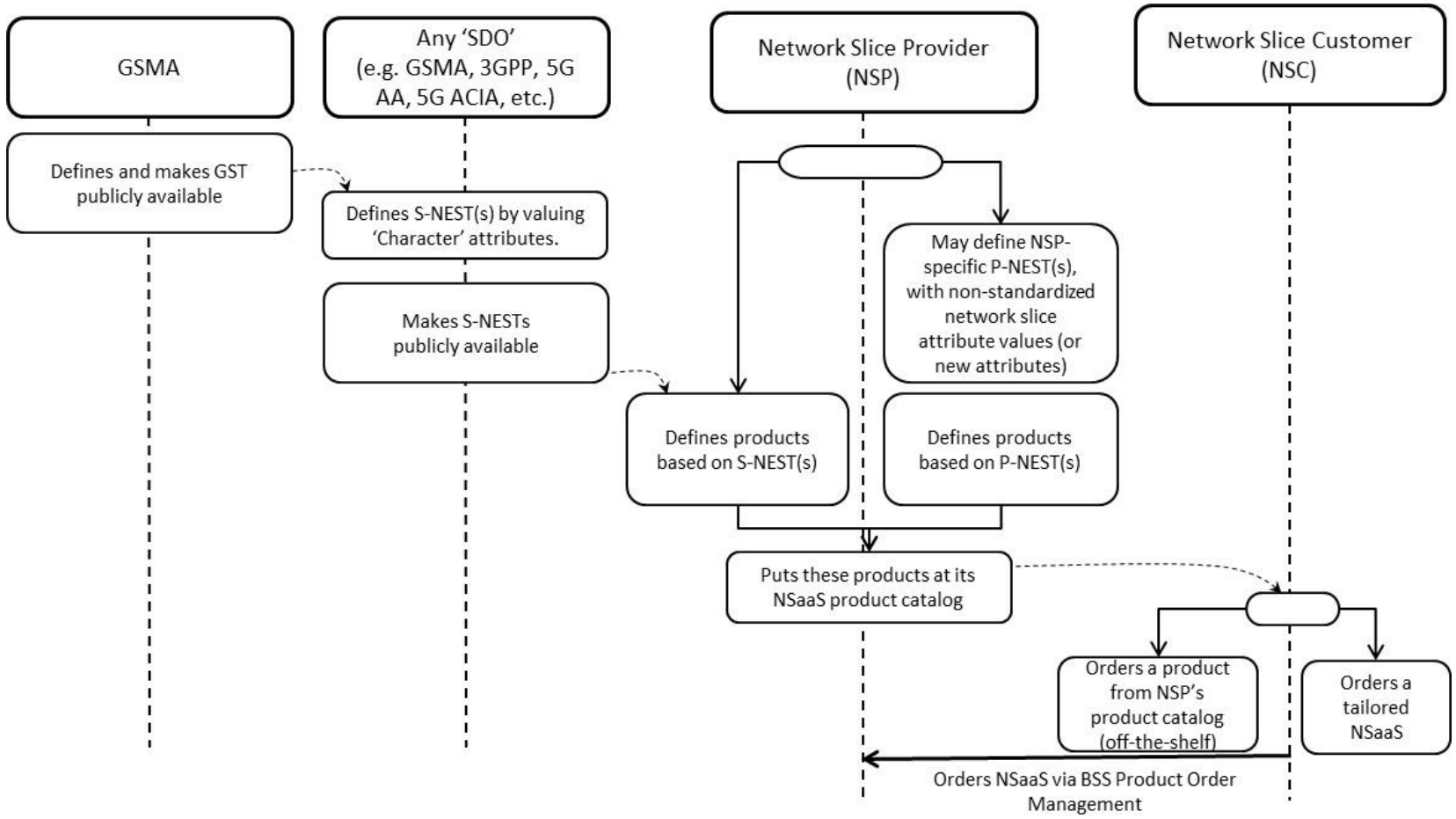
From GST to NEST

- **GST**: set of attributes (e.g. supported throughput, supported functionality, provided application programming interfaces (APIs), etc.) that characterize any slice.
- **NEST**: GST filled with values and/or ranges based on specific use case.
- S-NEST (standard) and P-NEST (private)



Source: GSMA, From Vertical Industry Requirements to Network Slice Characteristics, August 2018

From GST to S-(/P-)NEST based product ordering



Source: 3GPP TS 28.531 V16.3.0 (2019-09)

Example of a NEST (1/2)

Attribute	Attribute presence	Value
Character Attributes		
Coverage	Mandatory	2 (or 1 if roaming agreements are in place)
Delay tolerance	Mandatory	0
Deterministic communication	Mandatory	0
Isolation level	Mandatory	0
Group communication support	Mandatory	0
Location based message delivery	Mandatory	0
Maximum supported packet size	Optional	1500 (or 9000 if jumbo packets are supported)
Mission-critical support	Mandatory	0
MMTel support	Mandatory	1
Session and Service Continuity support	Mandatory	1
Slice quality of service parameters	Mandatory	1,2,5,6,7,8, 9
Support for non-IP traffic	Mandatory	0
Supported access technologies	Optional	4*

Example of a NEST (2/2)

Attribute	Attribute presence	Value
User management openness	Mandatory	0
User data access	Mandatory	0
Scalability Attributes		
Downlink throughput per network slice	Conditional	
Number of connections	Conditional	
Number of terminals	Conditional	
Radio spectrum	Optional	
Supported access technologies	Optional	
Uplink throughput per network slice	Conditional	

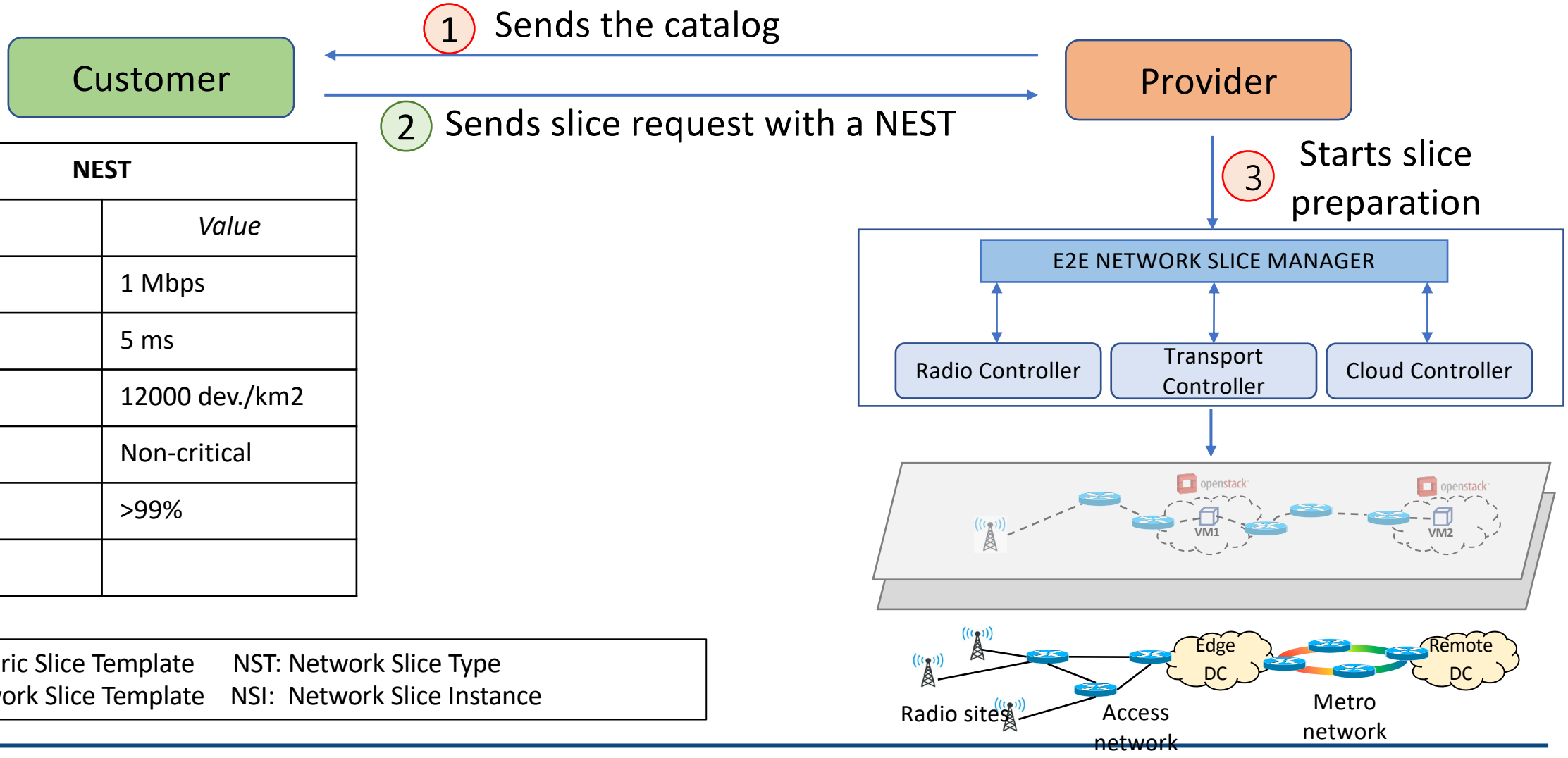
- Character Attributes - the attribute's value is in the NEST
- Scalability attributes - the attribute's value for scalability attributes depends on actual business case and each NSP is responsible for filling in the values.

Source: GSMA, NG.116 - Generic Network Slice Template, v.1, May 2019

NEST

- The Network Slice Type serves many purposes:
  - Vendors can use a NEST to define the features of their products
  - Vertical Industry customers (slice customers) can use a NEST as a reference to understand the contractual agreements with the network operator
  - Network operators (slice provider) can use a NEST with their “roaming” partners facilitating the definition of network slices in roaming agreements. An international roaming user can get a network slice with equivalent functionality of the slice used in the home network

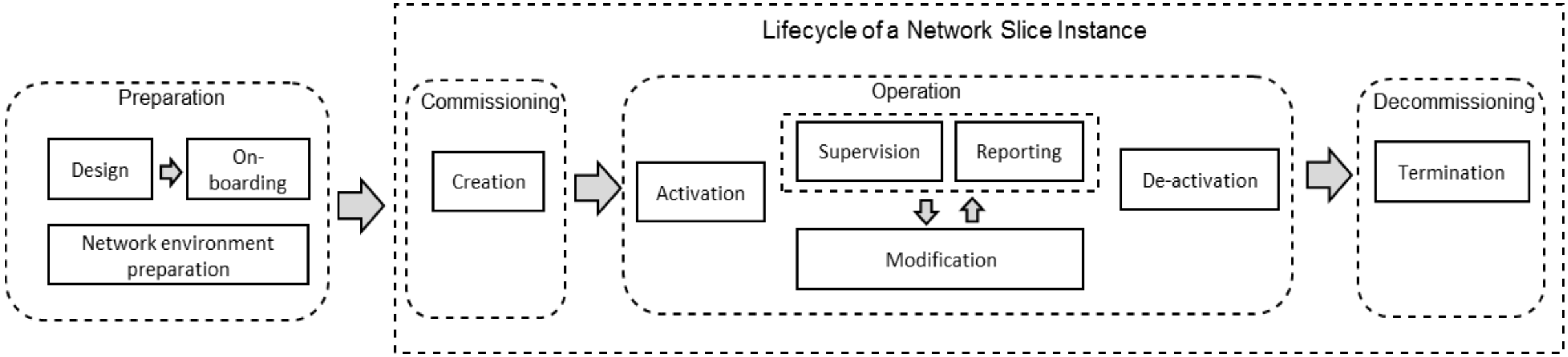
Slice creation process



NEST	
GST	*Value*
Data rate	1 Mbps
Latency	5 ms
Density	12000 dev./km2
Mobility	Non-critical
Reliability	>99%
…	

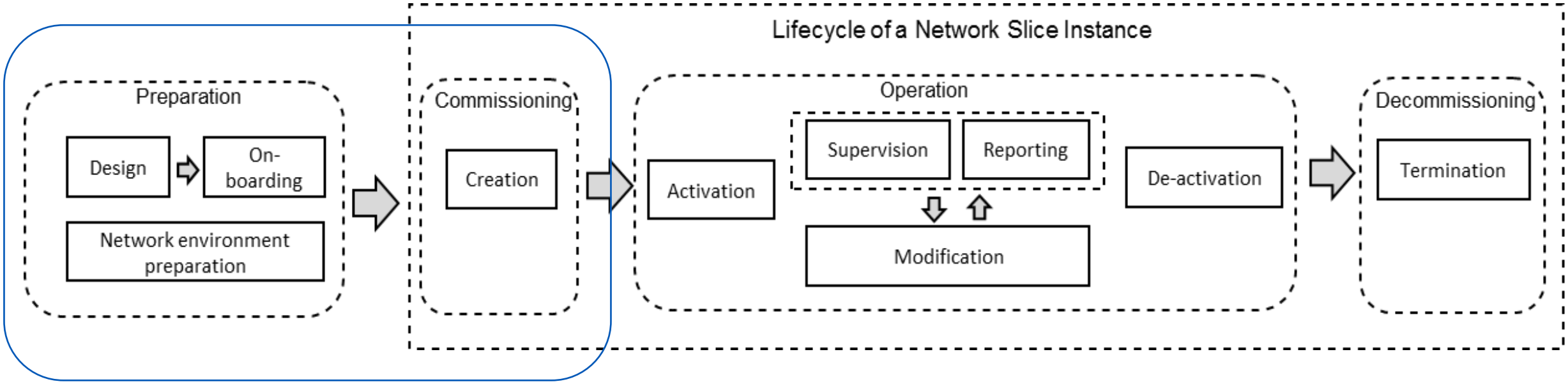
GST: Generic Slice Template NST: Network Slice Type
NEST: NEtwork Slice Template NSI: Network Slice Instance

Management aspects of a network slice instance



TS 28.530, Management and orchestration; Concepts, use cases and requirements, 3GPP Release 15 v15.1.0, 12/2018

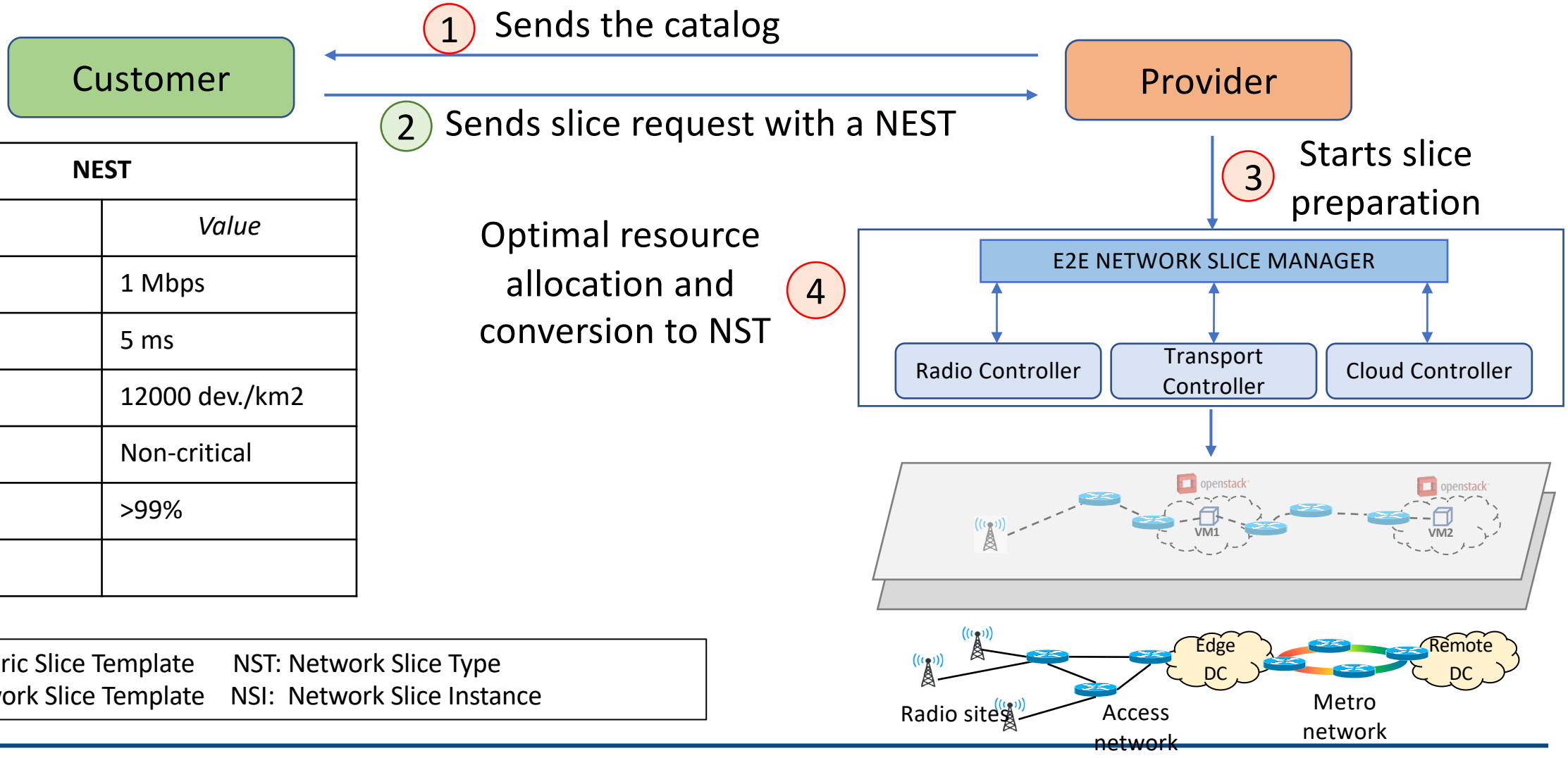
Management aspects of a network slice instance



Management aspects of a network slice instance

- Preparation: in this phase the NSI does not exist. The preparation phase includes network slice template design, network slice capacity planning, on-boarding and evaluation of the network slice requirements, preparing the network environment and other necessary preparations required to be done before the creation of an NSI.
- Commissioning: includes creation of the NSI. During NSI creation all needed resources are allocated and configured to satisfy the network slice requirements. The creation of an NSI can include creation and/or modification of the NSI constituents.
- Operation: this phase includes the activation, supervision, performance reporting (e.g. for KPI monitoring), resource capacity planning, modification, and de-activation of an NSI.
- Decommissioning: includes decommissioning of non-shared constituents if required and removing the NSI specific configuration from the shared constituents. After the decommissioning phase, the NSI is terminated and does not exist anymore.

Slice creation process



NEST	
GST	*Value*
Data rate	1 Mbps
Latency	5 ms
Density	12000 dev./km2
Mobility	Non-critical
Reliability	>99%
…	

GST: Generic Slice Template NST: Network Slice Type
NEST: NEtwork Slice Template NSI: Network Slice Instance

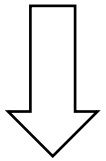
What and how to slice

- Network resources can be sliced
  - Radio domain (e.g., antenna/sectors, spectrum, core)
  - Transport domain (e.g., bandwidth, wavelengths)
  - Cloud domain (e.g., virtual machines/functions)

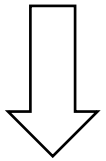
- A network slicer is required
  - orchestration of different domains
  - management and operation of services and virtual functions
  - resource optimization

Optimal resource allocation

- Given a set of resources, which ones do we assign to the slice?
- Service requirements/constraints to be satisfied (NEST)



Optimization problem (to be solved)

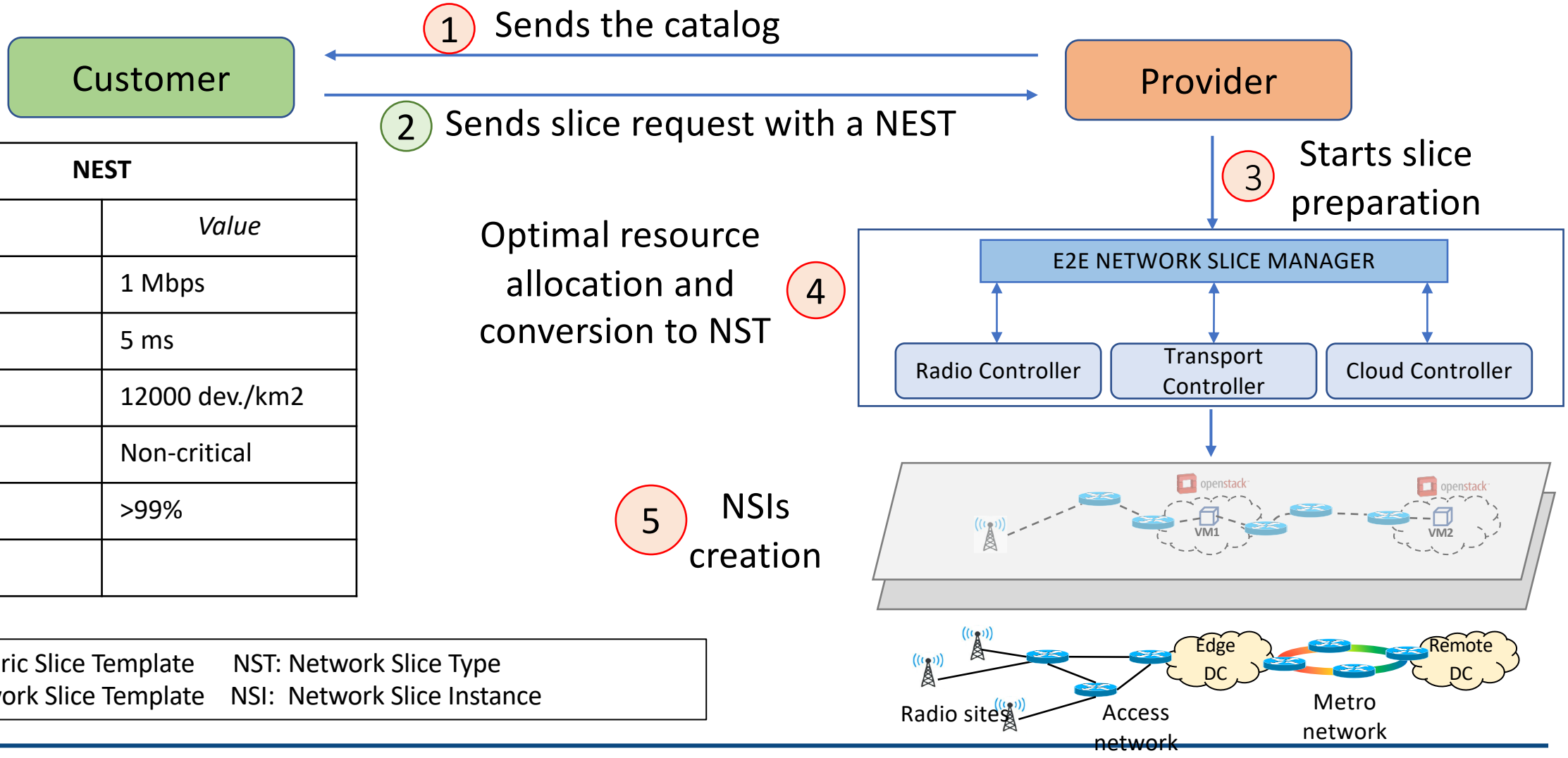


(Optimal) resource allocation

Conversion to Network Slice Type (NST)

- From optimal allocation to “low” level instructions
  - Selection from catalog (collection of available virtual network function and network service descriptors)
  - Set-up the whole network resources by means of standard languages and templates (e.g. YAML, TOSCA, …)
- NST distinguishes the kind of network resources needed to fulfil service requirements
- Different use cases can be mapped to the same NST

Slice creation process



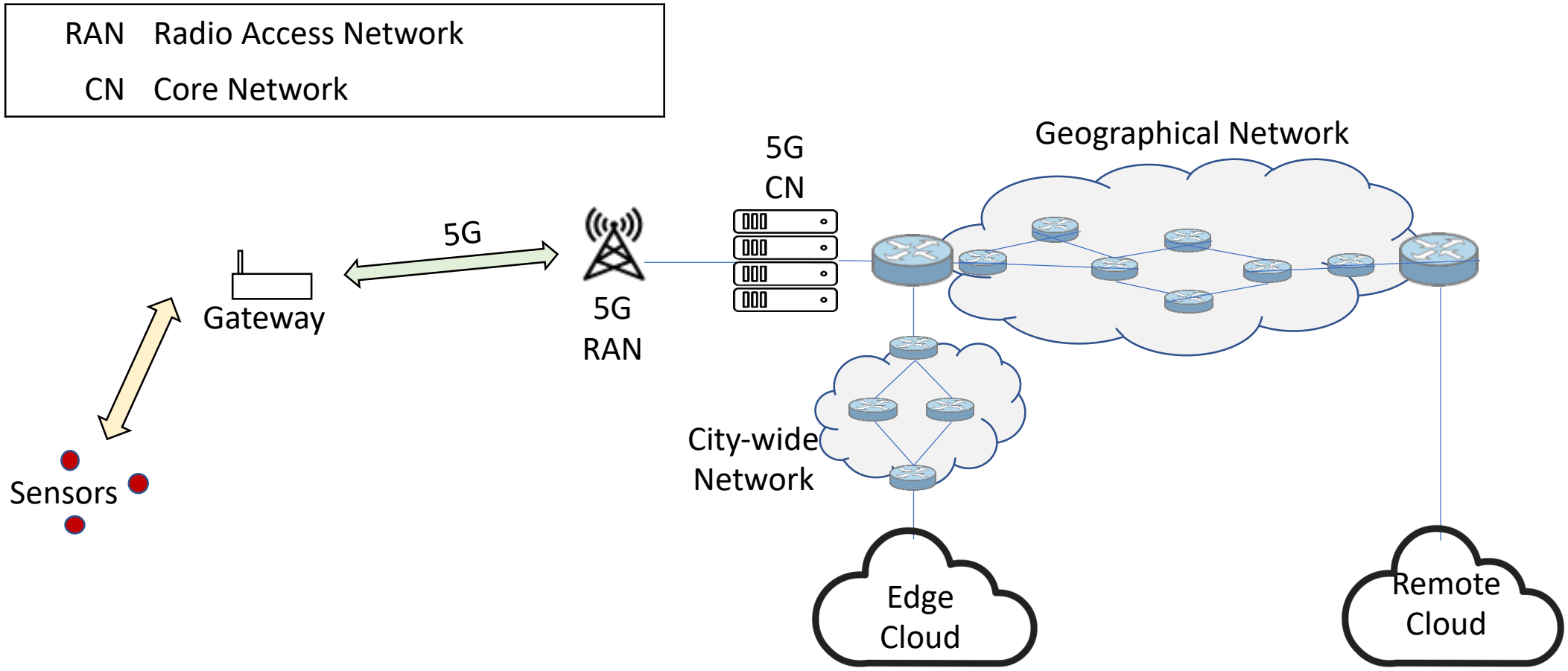
NEST	
GST	*Value*
Data rate	1 Mbps
Latency	5 ms
Density	12000 dev./km2
Mobility	Non-critical
Reliability	>99%
…	

GST: Generic Slice Template NST: Network Slice Type
NEST: NEtwork Slice Template NSI: Network Slice Instance

Outline of the lecture

- 5G Network slicing
  - Overview and definition
  - Involved resources and standardization
  - Slice creation process
- Examples
  - Slice deployment in the cloud
  - Reliable resource provisioning
- A quick look at 6G

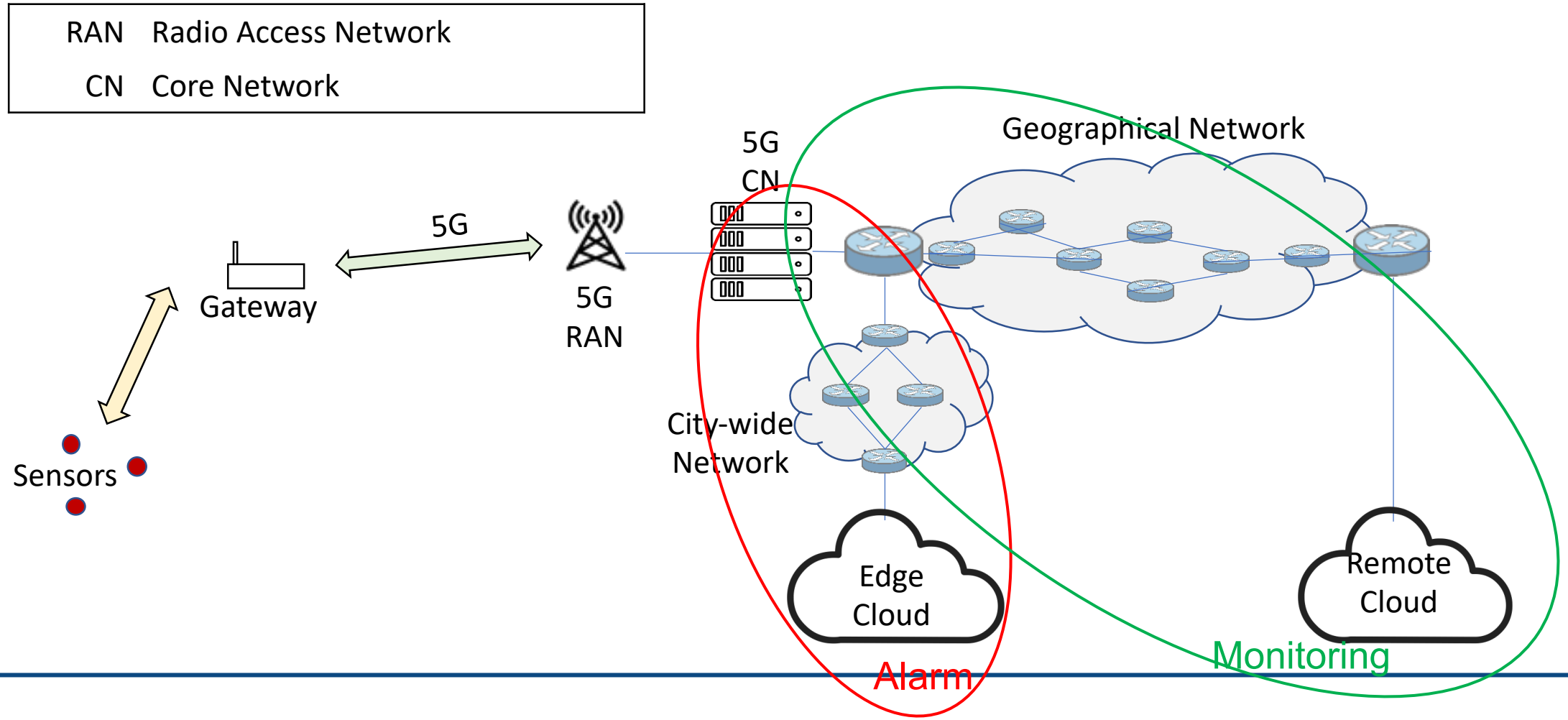
Reference scenario



Scenario description

- Some sensors are deployed collecting data (e.g., temperature, …)
- Data are sent through a gateway to a 5G network
- Data reaches different destinations (e.g. edge cloud, remote cloud) depending on the use case
- Applications with different requirements can be identified
  - Alarm
    - latency is critical (e.g., <5ms)
    - bandwidth is non-critical
  - Monitoring
    - latency is not critical (e.g., <50ms)
    - bandwidth is non-critical

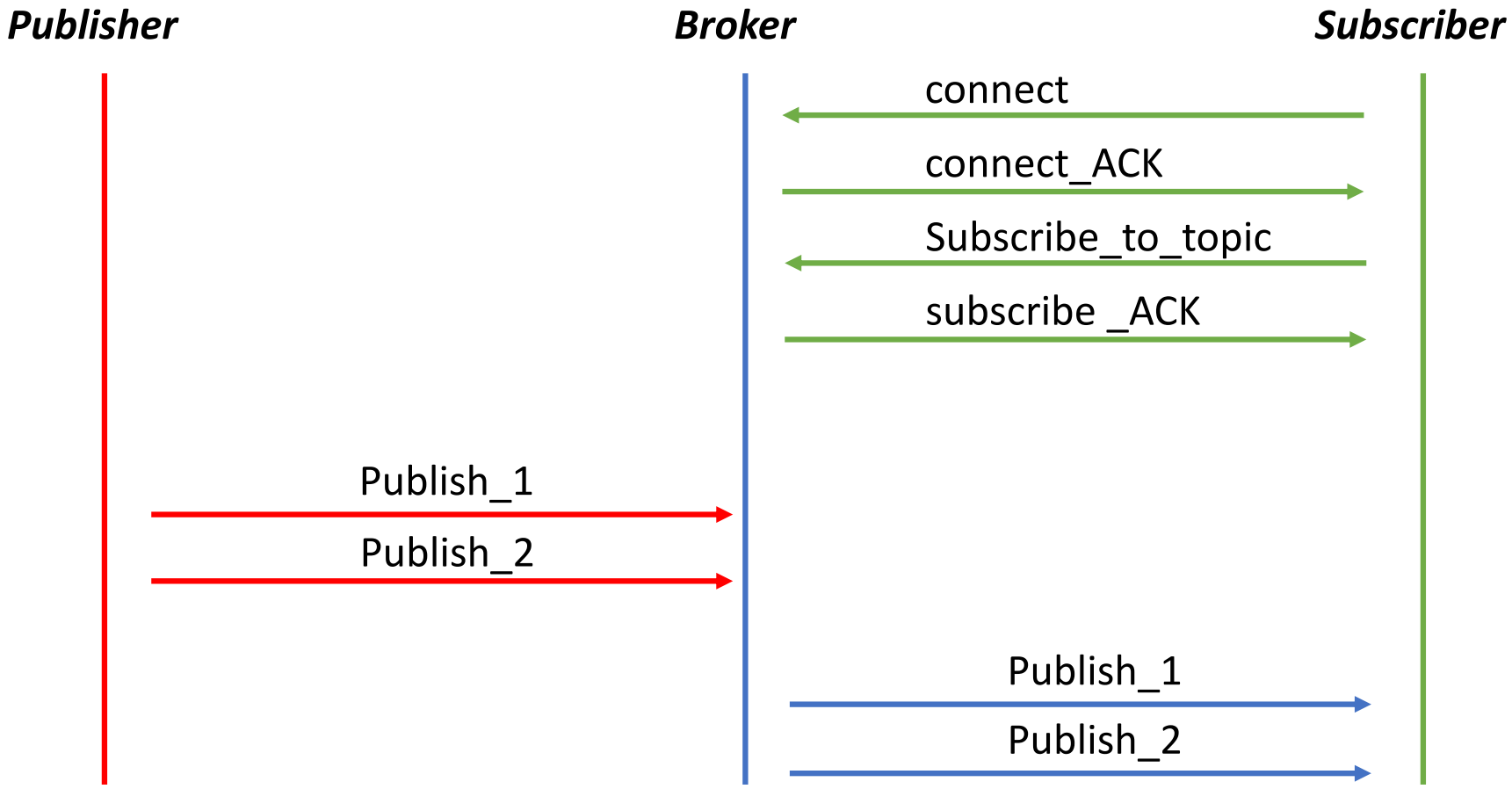
Two slices


RAN Radio Access Network
CN Core Network
5G
CN
Geographical Network
5G
Gateway
5G
RAN
City-wide
Network
Sensors
Edge
Cloud
Remote
Cloud
Alarm
Monitoring

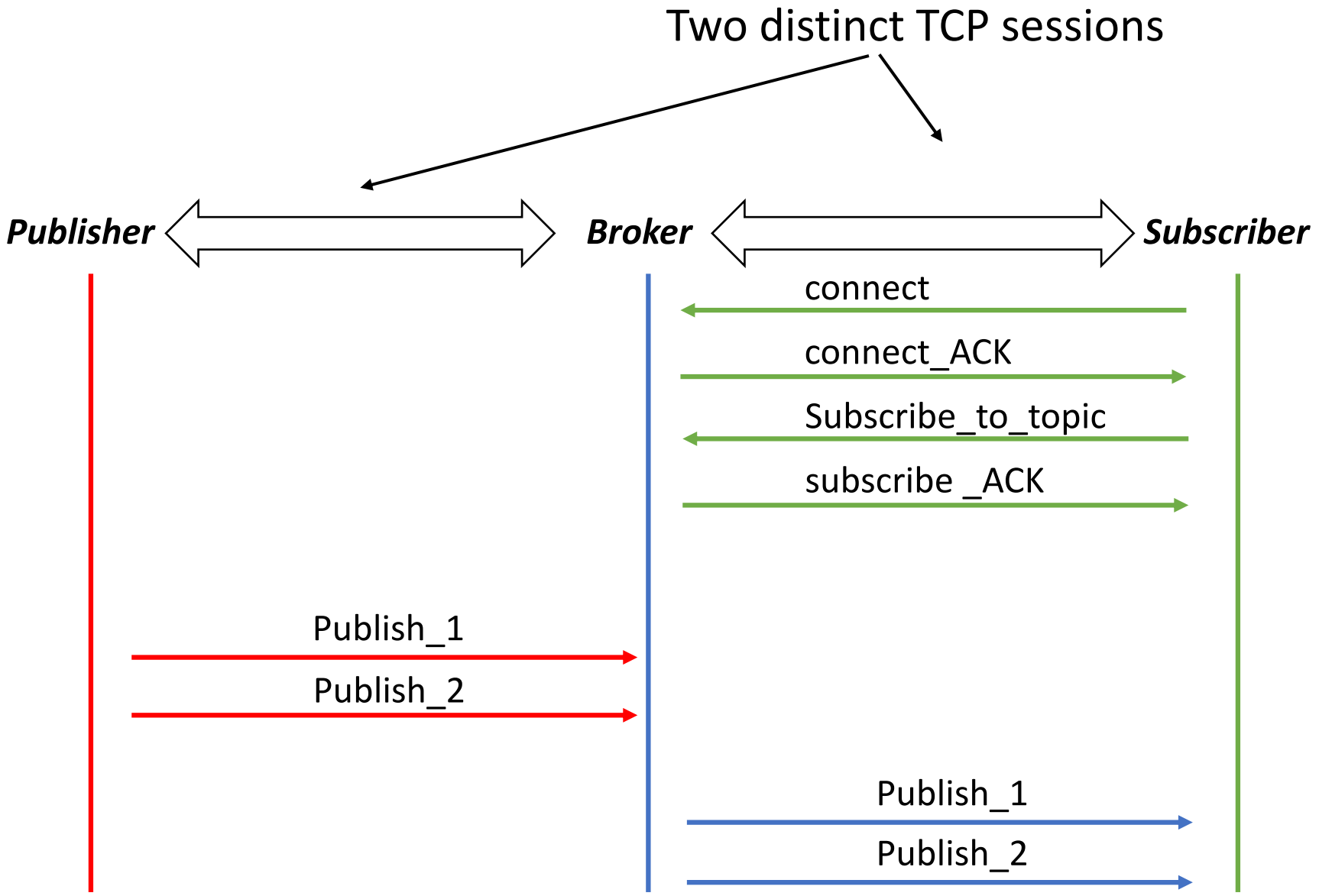
Message Queue Telemetry Transport (MQTT) protocol

- MQTT is a Client Server publish/subscribe messaging protocol
- The protocol runs over TCP/IP, or over other network protocols that provide ordered, lossless, bi-directional connections
- 3 entities:
  - ***Publisher***, sends the messages
  - ***Broker***, filters and forwards messages to interested subscribers
  - ***Subscriber***, subscribes to topics on a broker to receive messages published with that topic
- Topic refers to a string that the broker uses to filter messages for each connected client. Each message is published with a specific topic.

MQTT


Publisher
Broker
Subscriber
connect
connect_ACK
Subscribe_to_topic
subscribe _ACK
Publish_1
Publish_2
Publish_1
Publish_2

MQTT



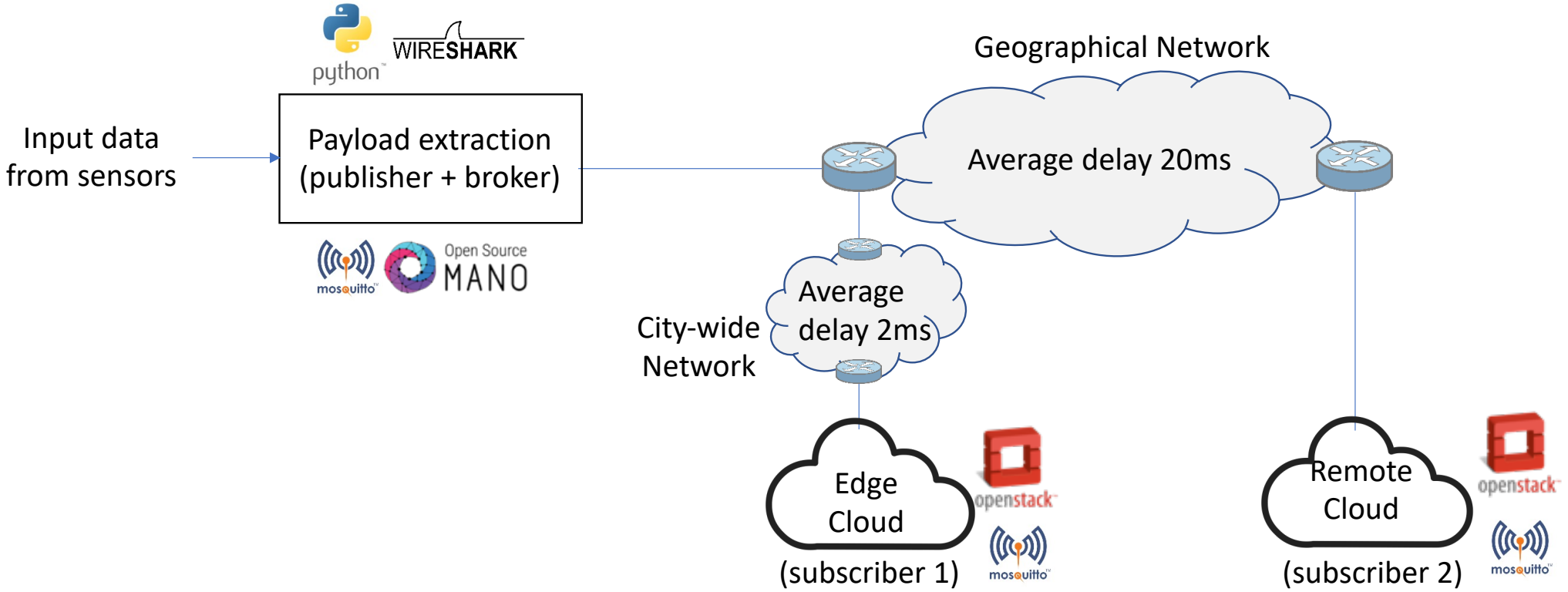
- Publish_2 is sent only after the reception of TCP ACK packet! -> the subscriber location impacts the delay

Software emulation of the testbed

- mosquitto implements MQTT message exchange with extracted payloads (100 messages/sec.)
- Python script extracts payload from real input traces and sends packets
- OSM instantiates cloud resources for each slice
- Cloud resources are located in two isolated Openstack Clusters (one for each slice)
- Wireshark for packet captures
- Delays are introduced with tc command

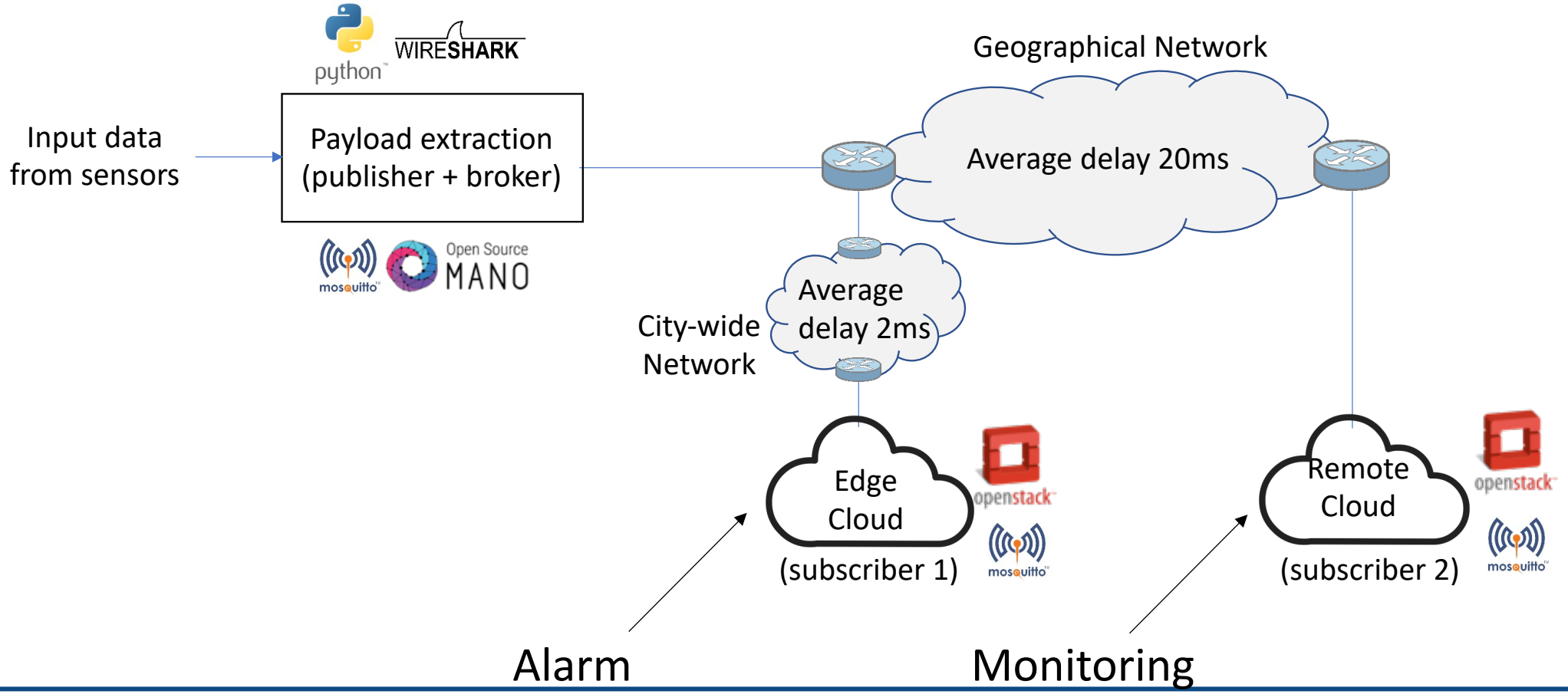
Testbed description

- Slicing only on cloud resources
- 5G network is neglected



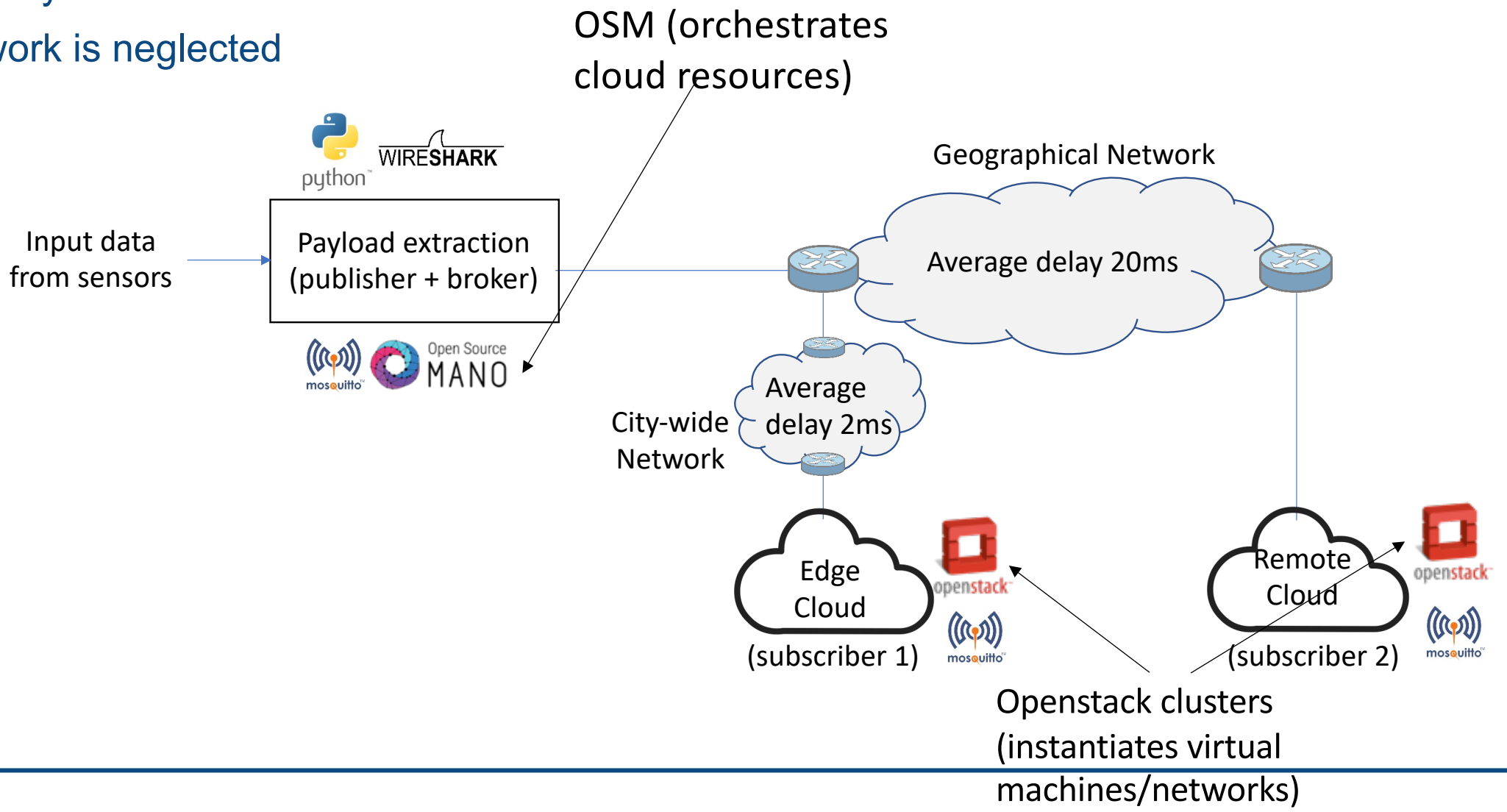
Testbed description

- Slicing only on cloud resources
- 5G network is neglected



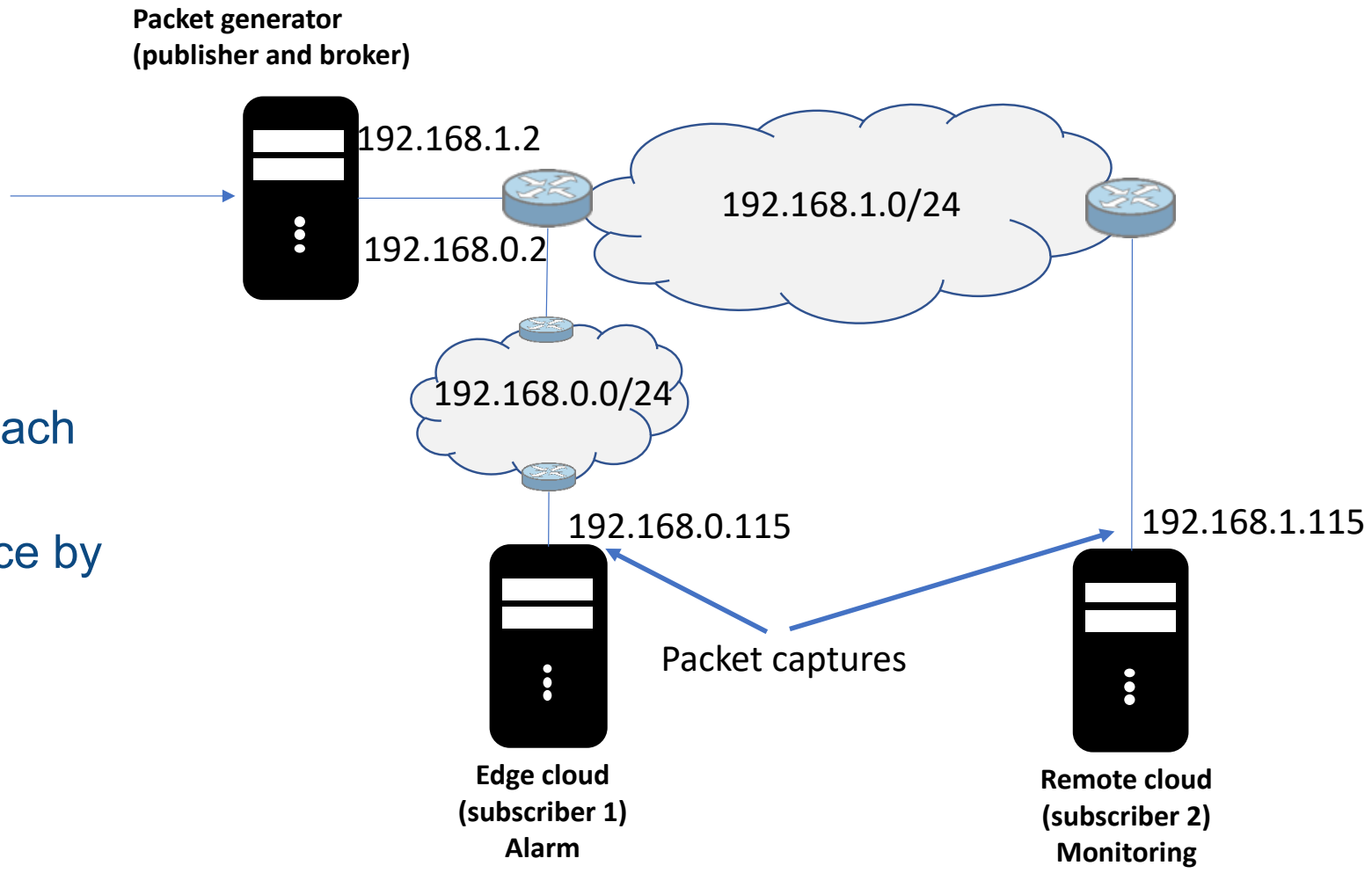
Testbed description

- Slicing only on cloud resources
- 5G network is neglected



Testbed configuration

- Two separate networks, one for each slice
- Delay is emulated at each interface by means of tc command



Slice descriptors (1/2)

- OSM relies on Network Service Descriptors (NSDs) to create service slices

- A NSD describes the connections between VNFs and virtual networks

```
nsd:nsd-catalog:
    nsd:
    -   constituent-vnfd:
        -   member-vnf-index: '1'
            vnfd-id-ref: allarme-vnf
        description: Allarme NS
        id: allarme-ns
        ip-profiles:
        -   description: rete di allarme
            ip-profile-params:
                dhcp-params:
                    enabled: true
                gateway-address: 192.168.10.1
                ip-version: ipv4
                subnet-address: 192.168.10.0/24
            name: rete_allarme
        name: allarme-ns
        short-name: allarme-ns
        version: '1.0'
        vld:
        -   id: mgmtnet
            name: mgmtnet
            short-name: mgmtnet
            type: ELAN
            vim-network-name: external
            vnfd-connection-point-ref:
            -   ip-address: 192.168.0.151
                member-vnf-index-ref: '1'
                vnfd-connection-point-ref: vnf-cp0
                vnfd-id-ref: allarme-vnf
```

Slice descriptors (2/2)

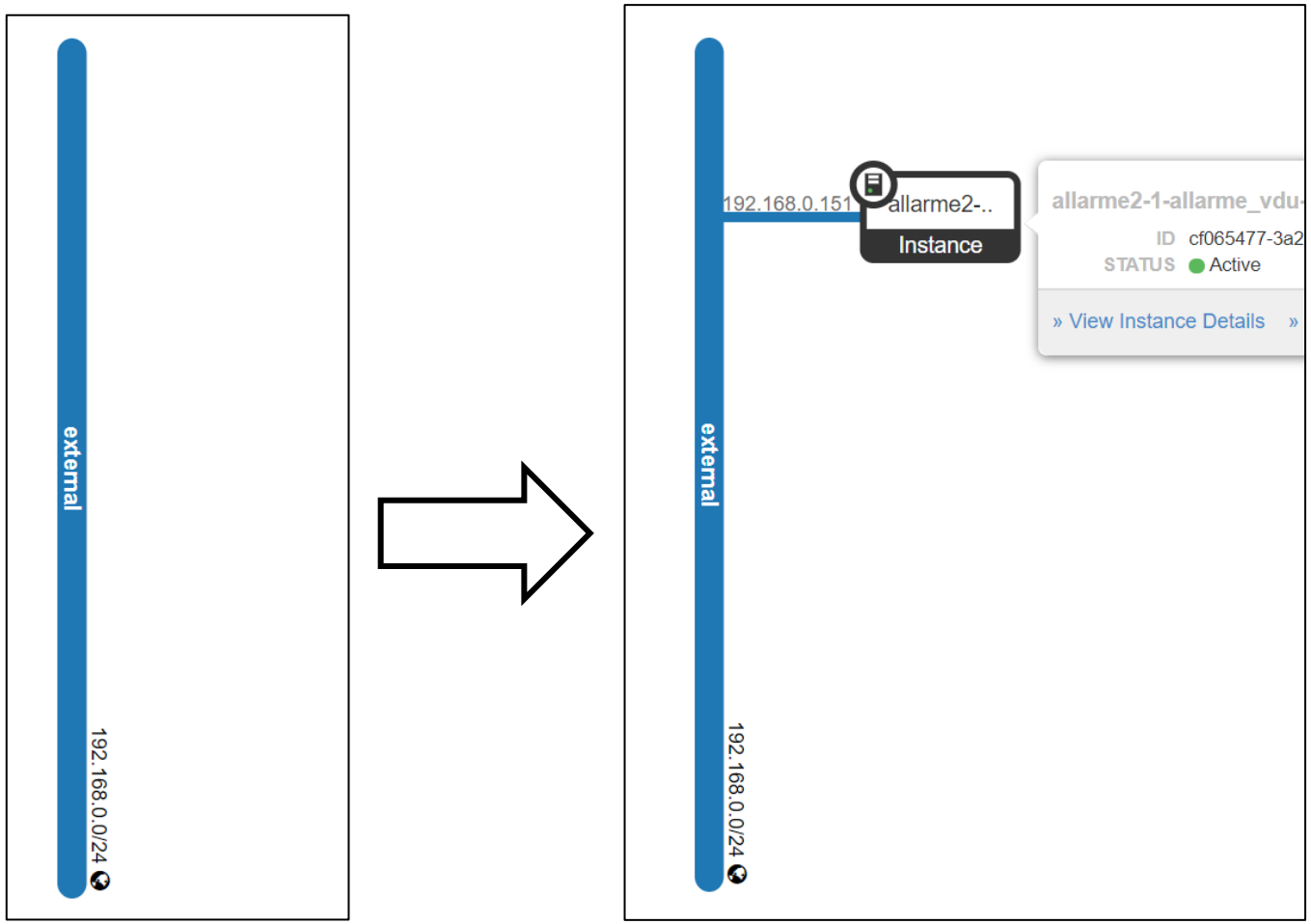
- Each VNF is described by a VNF descriptor (VNFD)

- A VNFD contains information on virtual machines and their characteristics (RAM, CPUs, ...)

```
vnfd:vnfd-catalog:
    vnfd:
    -   connection-point:
        -   name: vnf-cp0
            type: VPORT
        description: Semplice versionte di VNF formato da una VDU
        id: allarme-vnf
        mgmt-interface:
            cp: vnf-cp0
        name: allarme-vnf
        short-name: allarme-vnf
        vdu:
        -   cloud-init-file: cloud.txt
            count: '1'
            id: allarme_vdu-VM
            image: MqttServer
            interface:
            -   external-connection-point-ref: vnf-cp0
                name: vdu-eth0
                position: '1'
                type: EXTERNAL
                virtual-interface:
                    type: VIRTIO
            name: allarme_vdu-VM
            supplemental-boot-data:
                boot-data-drive: true
            vm-flavor:
                memory-mb: '4096'
                storage-gb: '20'
                vcpu-count: '4'
        version: '1.0'
```

Slice instantiation

- Openstack is a cloud OS that provides clusters of virtual resources interconnected by virtual networks
- OSM communicates with Openstack to instantiate the required resources (specified in the NSD)



Before

After

Example of MQTT subscriber connection establishment (subscriber 1)

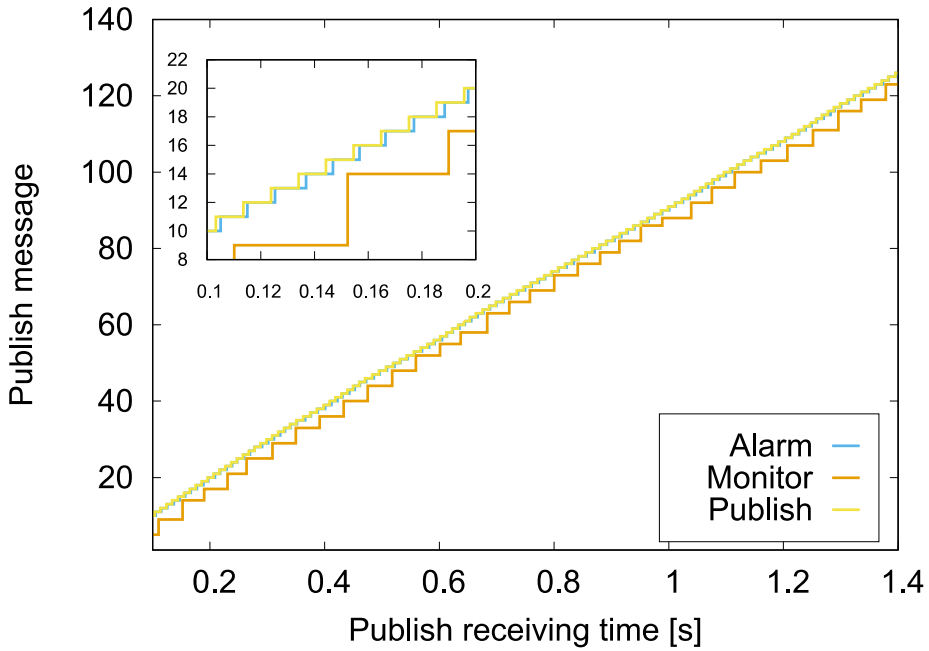
- To test the system we captured packets at each subscriber network interface

No.	Time	Source	Destination	Protocol	Length	Info
87	18.295313	192.168.0.115	192.168.0.2	TCP	74	36074 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=36534119 TSecr=0 WS=128
88	18.300376	192.168.0.2	192.168.0.115	TCP	74	1883 → 36074 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1832076126 TSecr=36534119 WS=1…
89	18.305611	192.168.0.115	192.168.0.2	TCP	66	36074 → 1883 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=36534122 TSecr=1832076126
90	18.305636	192.168.0.115	192.168.0.2	MQTT	105	Connect Command
91	18.310647	192.168.0.2	192.168.0.115	TCP	66	1883 → 36074 [ACK] Seq=1 Ack=40 Win=29056 Len=0 TSval=1832076128 TSecr=36534122
92	18.310790	192.168.0.2	192.168.0.115	MQTT	70	Connect Ack
93	18.315945	192.168.0.115	192.168.0.2	TCP	66	36074 → 1883 [ACK] Seq=40 Ack=5 Win=29312 Len=0 TSval=36534125 TSecr=1832076128
94	18.315954	192.168.0.115	192.168.0.2	MQTT	74	Subscribe Request (id=1) [#]
95	18.321013	192.168.0.2	192.168.0.115	MQTT	71	Subscribe Ack (id=1)
96	18.363894	192.168.0.115	192.168.0.2	TCP	66	36074 → 1883 [ACK] Seq=48 Ack=10 Win=29312 Len=0 TSval=36534137 TSecr=1832076131
97	24.048969	192.168.0.2	192.168.0.115	MQTT	91	Publish Message [xyz_data]
98	24.054215	192.168.0.115	192.168.0.2	TCP	66	36074 → 1883 [ACK] Seq=48 Ack=35 Win=29312 Len=0 TSval=36535559 TSecr=1832077563
99	24.087707	192.168.0.2	192.168.0.115	MQTT	91	Publish Message [xyz_data]

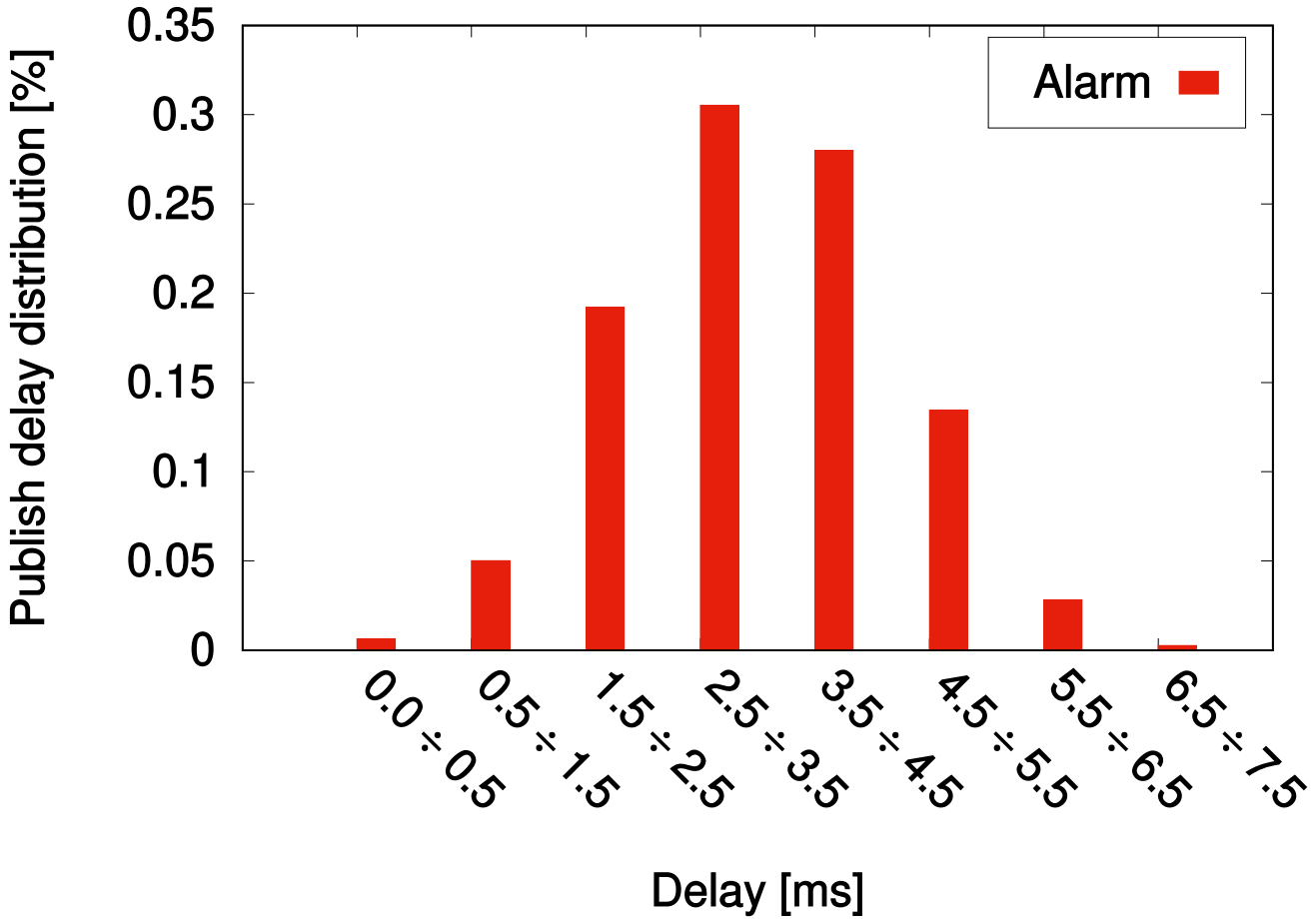
Packet aggregation at TCP level (subscriber 2)

No.	Time	Source	Destination	Protocol	Length	Info
35	71.919409	192.168.1.2	192.168.1.115	MQTT	291	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
36	71.971205	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=2210 Win=38912 Len=0 TSval=3692892123 TSecr=1832077790
37	72.021246	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
38	72.071479	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=2460 Win=39936 Len=0 TSval=3692892225 TSecr=1832077815
39	72.121518	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
40	72.171697	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=2710 Win=41088 Len=0 TSval=3692892326 TSecr=1832077840
41	72.221737	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
42	72.271915	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=2960 Win=42112 Len=0 TSval=3692892426 TSecr=1832077865
43	72.321977	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
44	72.372282	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=3210 Win=43136 Len=0 TSval=3692892526 TSecr=1832077890
45	72.422515	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…
46	72.475076	192.168.1.115	192.168.1.2	TCP	66	38598 → 1883 [ACK] Seq=48 Ack=3460 Win=44288 Len=0 TSval=3692892627 TSecr=1832077916
47	72.525124	192.168.1.2	192.168.1.115	MQTT	316	Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_data], Publish Message [xyz_dat…

Latency and TCP aggregation effects (pub/sub MQTT protocol)

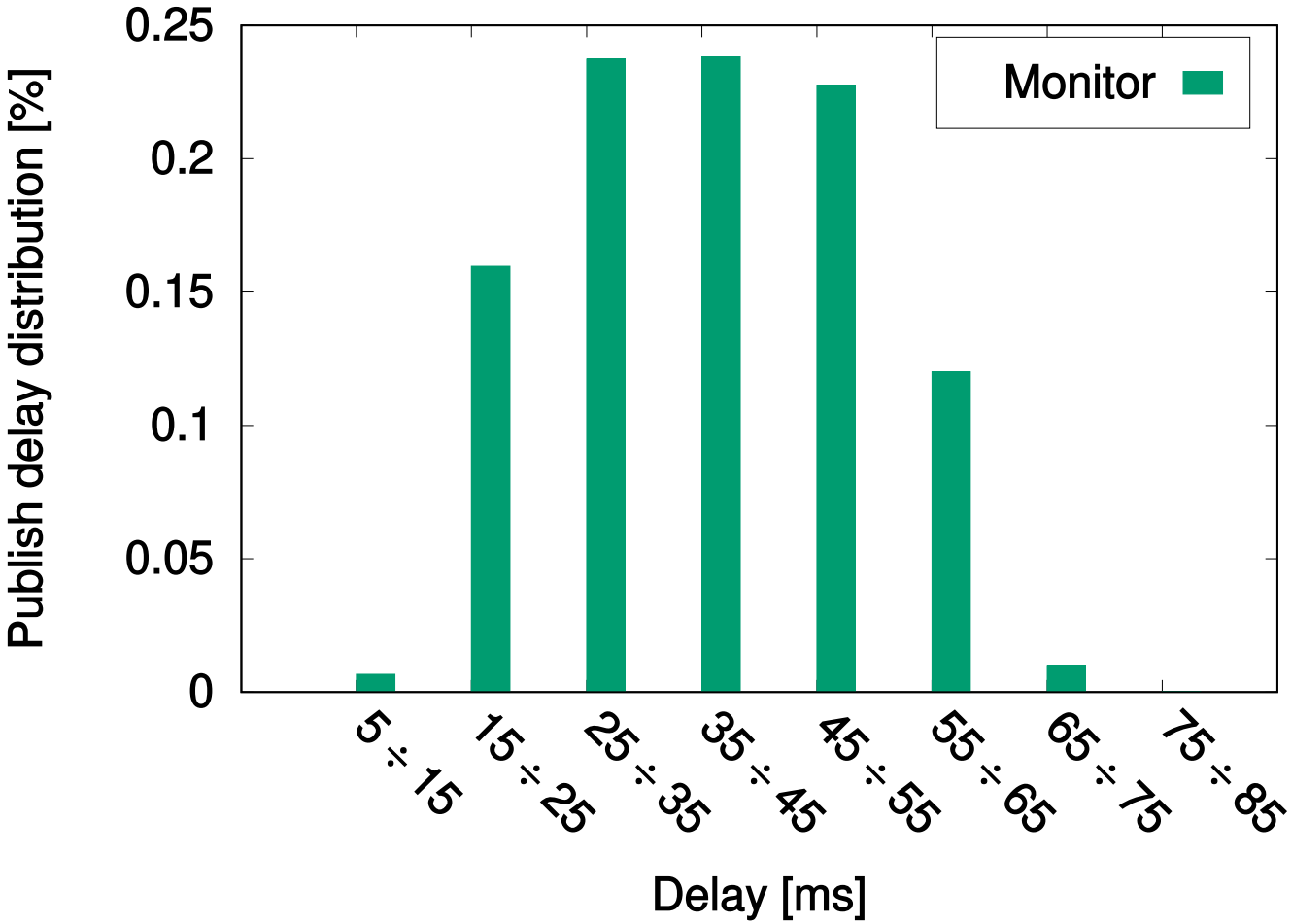


Publish average delay towards edge cloud (alarm)



- Avg. emulated delay 2ms (one-way)
- 1 Publish message every 10ms
- Additional delay due to physical (real) network

Publish average delay towards remote cloud (monitoring)



- Avg. emulated delay 20ms (one-way)
- 1 Publish message every 10ms
- Publish message aggregation creates additional delay

Outline of the lecture

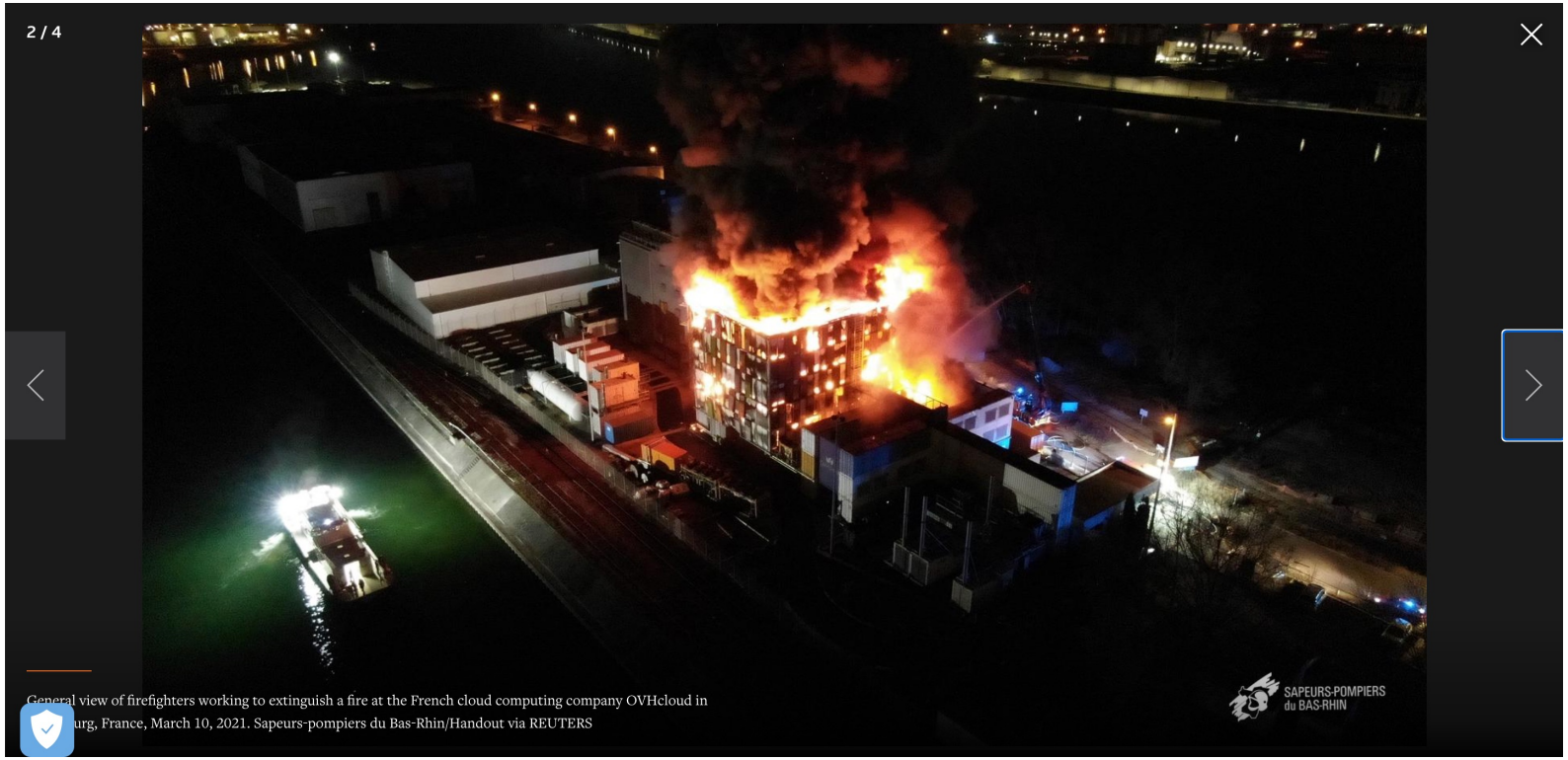
- 5G Network slicing
  - Overview and definition
  - Involved resources and standardization
  - Slice creation process
- Examples
  - Slice deployment in the cloud
  - Reliable resource provisioning
- A quick look at 6G

Reliability in network slicing

- Several services are relying on a single infrastructure
  - The infrastructure is subject to failures that potentially affect several slices
  - Some services are more critical than others (e.g., vehicular networks, real time mobile trading)
- Example of URLLC service requirements
  - Latency: ~1ms
  - Bandwidth: non-critical
  - Reliability: high
- Each slice spans across different domains (e.g., radio, transport, cloud)
  - Each domain must be properly protected

Why is reliability important?

- OVHCloud is the Europe's largest cloud services provider
- Millions of websites affected
- Several companies suffered data losses



General view of firefighters working to extinguish a fire at the French cloud computing company OVHcloud in urg, France, March 10, 2021. Sapeurs-pompiers du Bas-Rhin/Handout via REUTERS

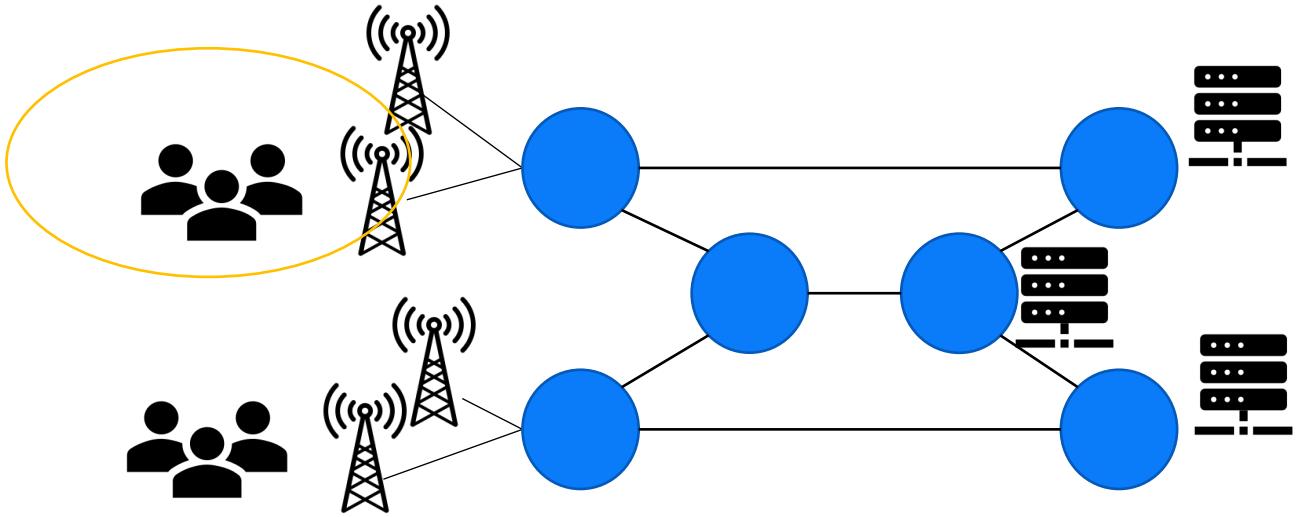
Failures

- Example of failures
  - Fiber cuts
  - Optical equipment failures (e.g., transponders, amplifiers, …)
  - IT equipment failures (e.g., servers, VMs)
  - Power outages
  - …
- To avoid severe service interruptions optimized provisioning of extra resources is required
  - We want to keep the number of backup resources low (cost!)
- The focus of this work is on single failure

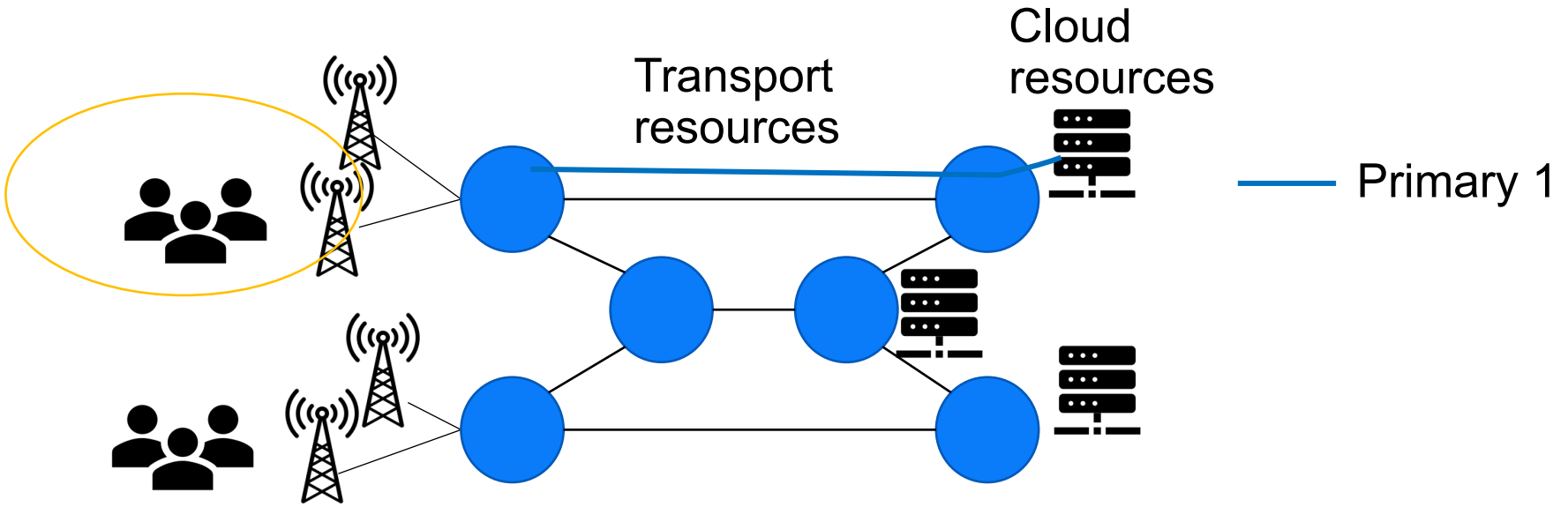
Network slicing and reliable slice-as-a-service provisioning

- Network slicing allows provisioning of several services (slices) over the same 5G infrastructure
- Providers can slice the 5G network, offering slices-as-a-service tailored to the different needs
- Providers must satisfy slice requirements
  - Slices are admitted on an infrastructure if there are enough free resources, according to the specific requirements. If not, slice requests are blocked (rejected).
- This work focuses on slices with strict reliability and latency requirements in 5G metro networks
  - Resources are usually scarce
  - Efficient techniques are needed to minimize provisioning of extra resources for backup

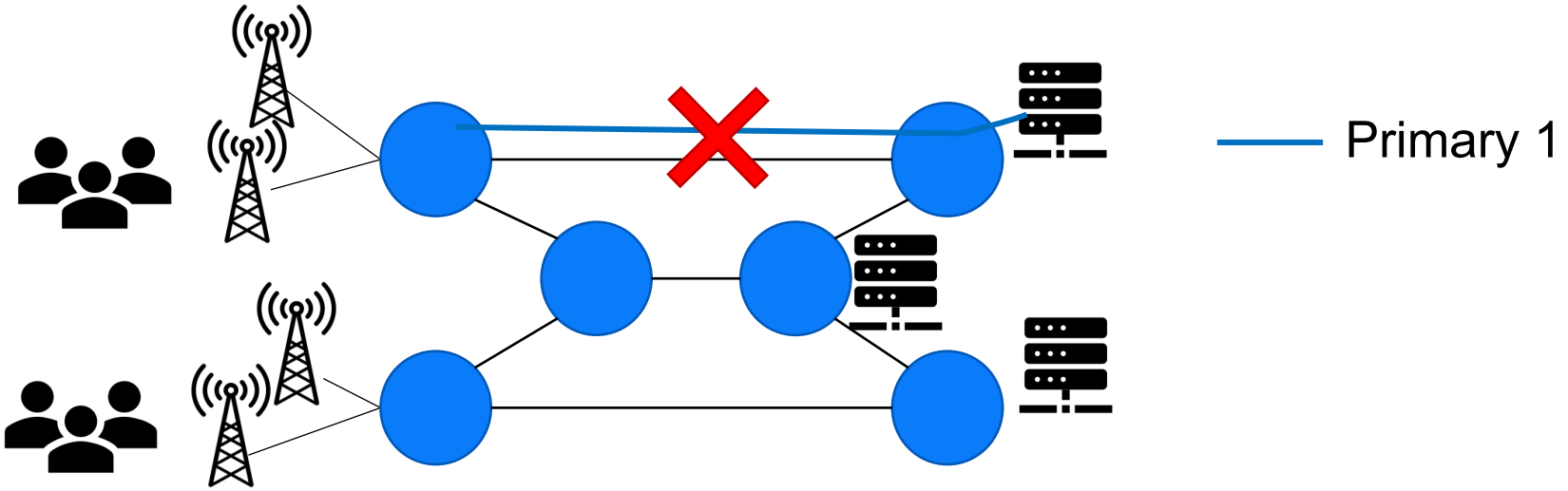
Example



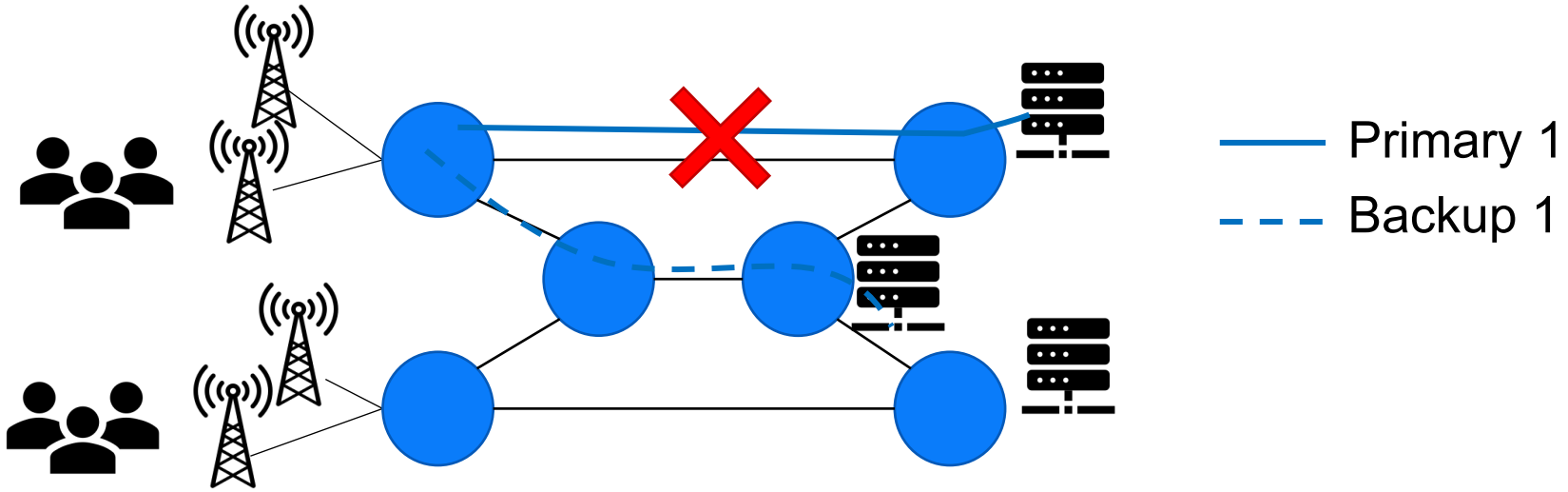
Example



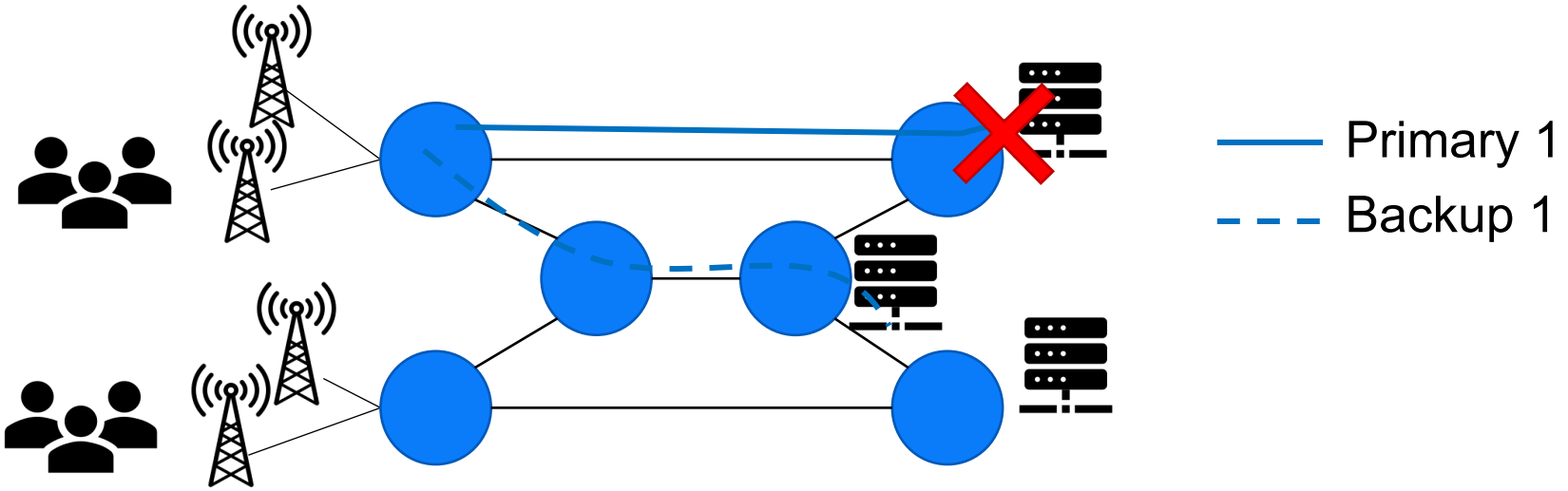
Example



Example



Example

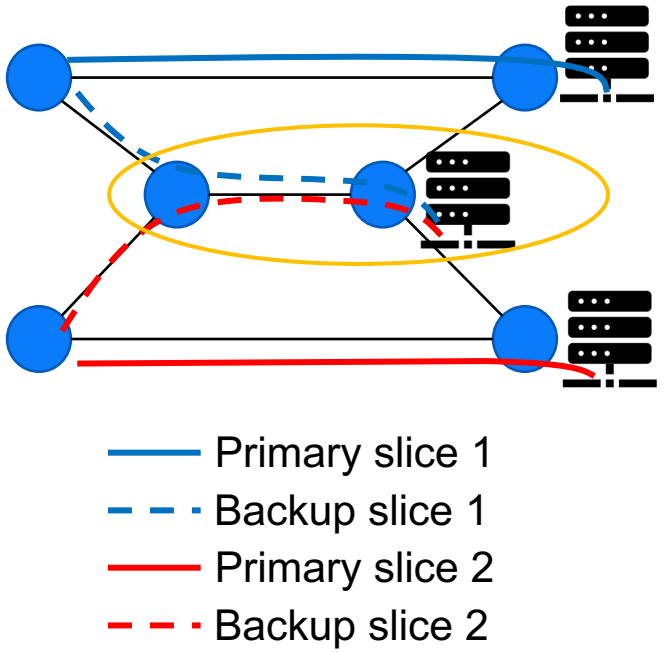


Problem formulation for the reliable slice provisioning

- Given:
  - a network topology with available network resources (bandwidth and processing units (PU))
  - the requirements of a slice to be provisioned
- To find:
  - a suitable slice provisioning, such that the allocated network resources are minimized
- To ensure:
  - reliability against single link or node failure
  - the bandwidth and PU resources allocated at each link and node do not exceed the available resources
  - the maximum distance between a source node and a target node is enforced (latency).

Dedicated vs. Shared Protection

- A conventional way is to provide Dedicated Protection (DP)
  - backup resources are dedicated for each slice
- Shared Protection (SP) allows to share backup resources among slices
  - Backup resources can be shared between two (or more) slices if primary resources (connectivity and compute) are disjoint
- SP leads to:
  - Lower resource required per slice
  - Increased slice admission (or lower blocking)
  - Slightly higher recovery time due to the need to switch from primary to backup



The algorithm

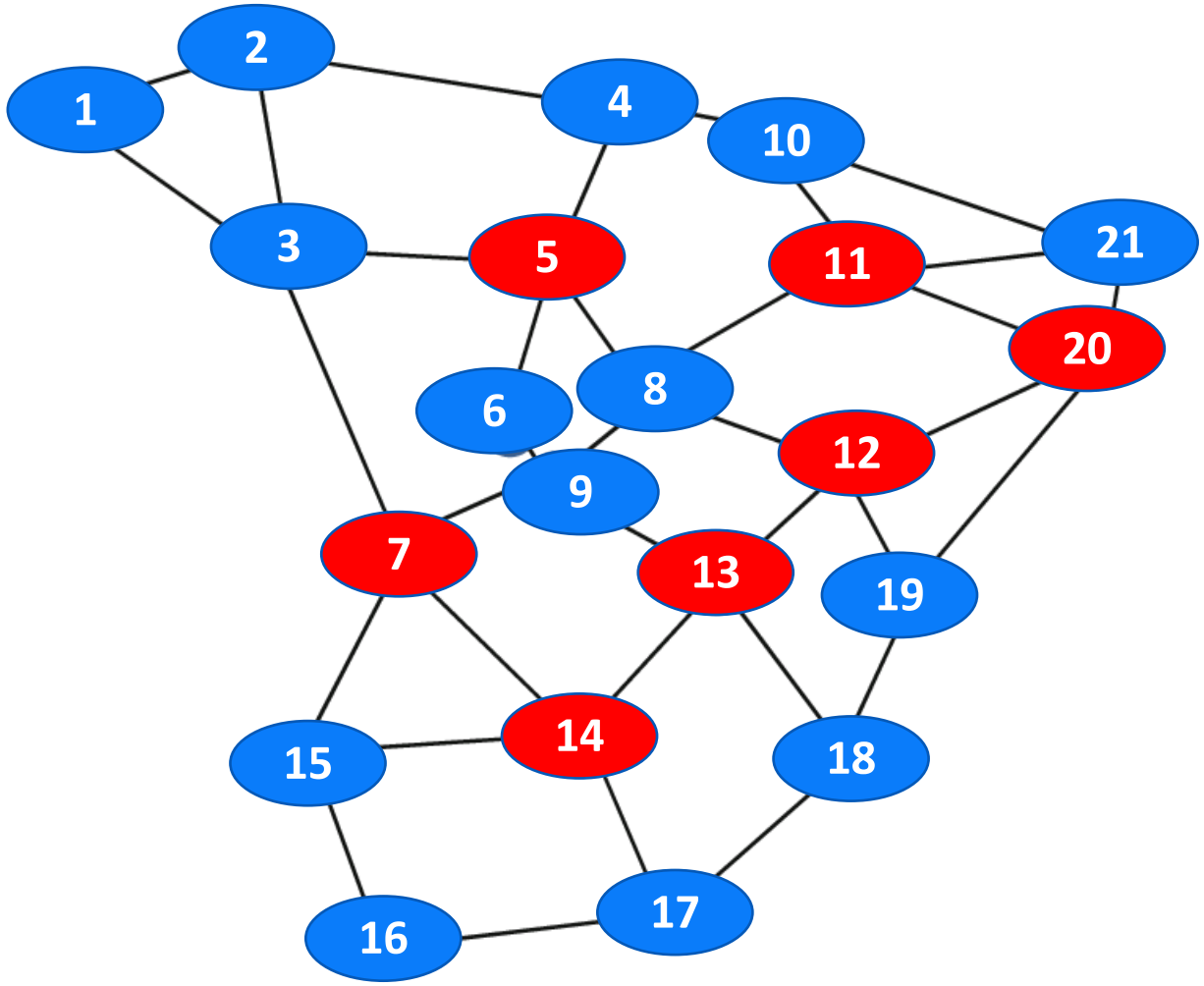
- Upon slice request, an exhaustive search of all possible combinations is performed to find the primary and backup resources
  - Connectivity and compute must not exceed available resources
  - Latency must not be exceeded
- Among all the combinations, the algorithm picks the one with the lowest cost

$$Cost = \alpha * \sum_{i \in Links} Band_i + \beta * \sum_{j \in Nodes} Compute_j$$

- Slice resources are released when slice expires
- We developed a simulator to compare DP and SP

Simulation settings

- 14 source nodes (blue), 7 target nodes (red)
- Each slice requires
  - 24 Gbps bandwidth per link
  - 12 PU (baseband + service)
  - Strict reliability and latency (max. 4 hops)
- Each link has 1Tbps capacity, the same length (1-hop)
- Two scenarios
  - 2000 PU available per target node
  - 500 PU available per target node
- $\alpha = \beta = 1$
- 1 slice request per time unit
- Service time is exponential and varies



Blocking probability

- π_b= blocking probability: prob. that a slice request is not accepted
- SP considerably reduces π_b thanks to sharing in both scenarios

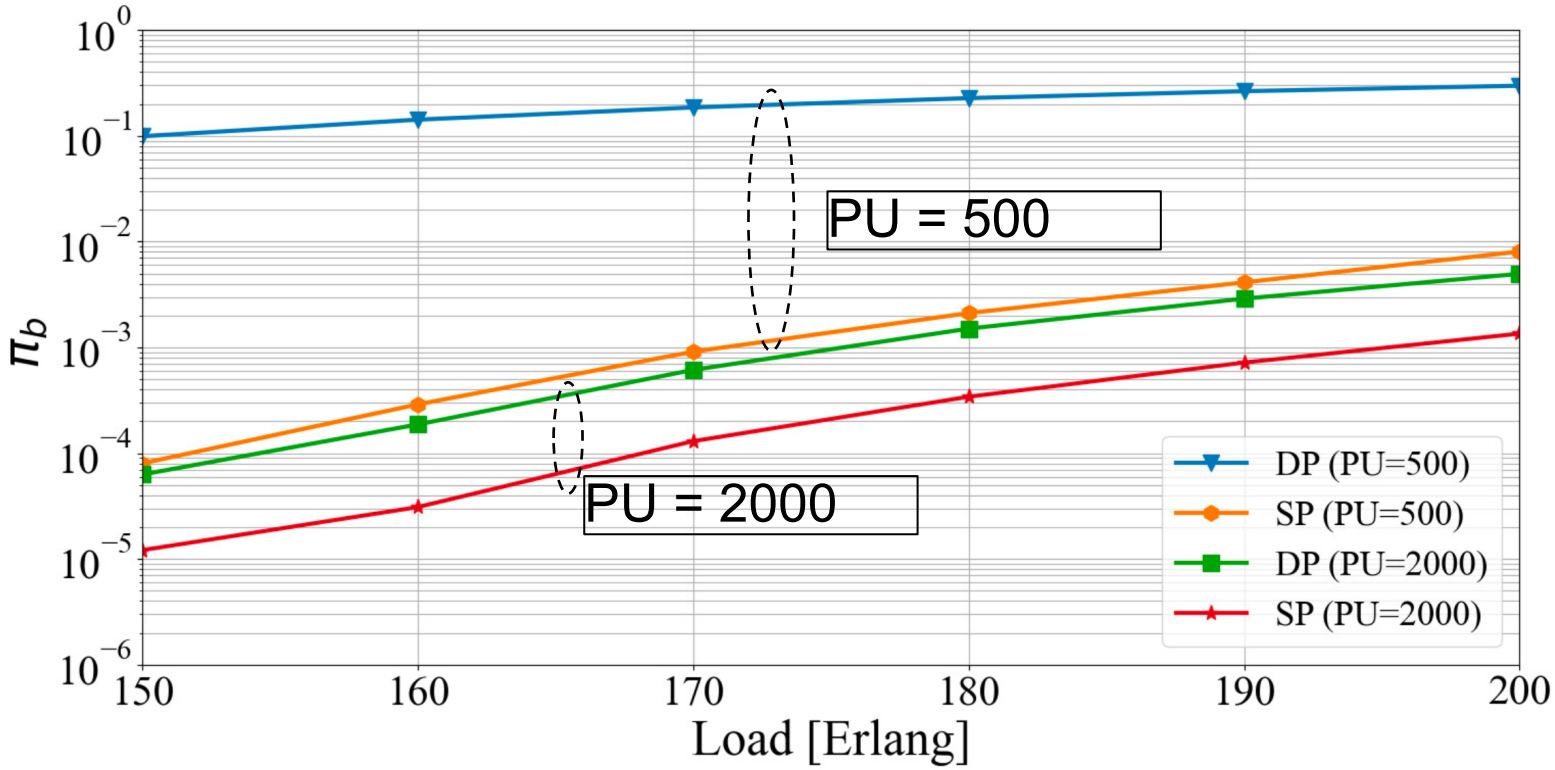


Fig. 2. Blocking probability as a function of the load.

Average bandwidth per slice

- Bs = average bandwidth per slice
- When PU=500
  - DP +10% from low to high load
  - DP requires up to 28% more band.
- When PU=2000, band is almost constant (PUs are not an issue in this case)

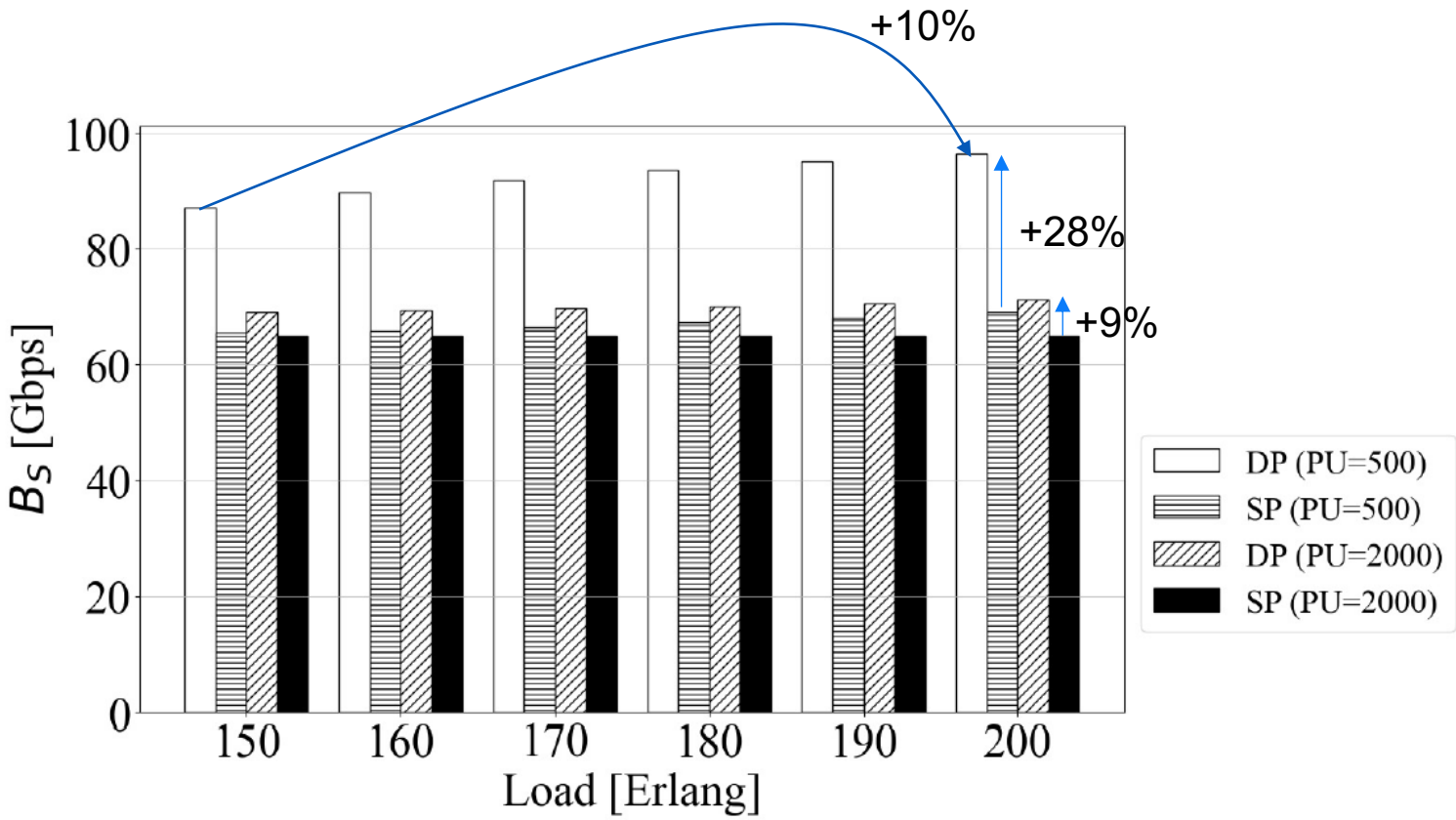


Fig. 3. Average bandwidth per slice (B_S) as a function of the load.

Average compute (PU) per node when load=200

- PU_U^j= avg. PU per target node j
- When PU=500, DP saturates the nodes, hence high blocking
- When PU=2000, PU utilization is less than 50% (not an issue)

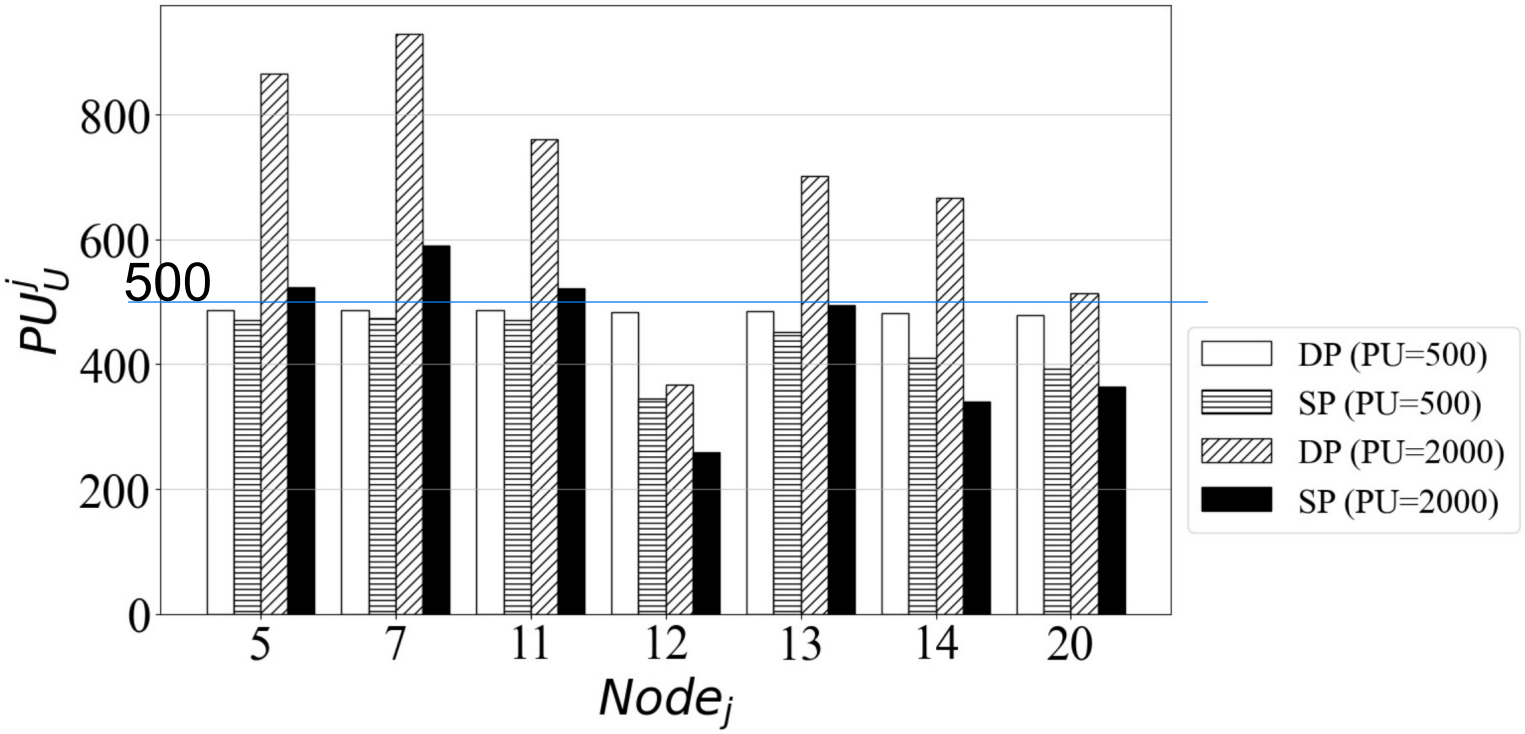


Fig. 4. Average PU per target node j (PU_U^j) with load = 200.

Average bandwidth per link when load=200

- $B_U^{i,j}$ = avg. band per link i-j
- When PU=2000, some links are close to saturation, hence the higher blocking for DP

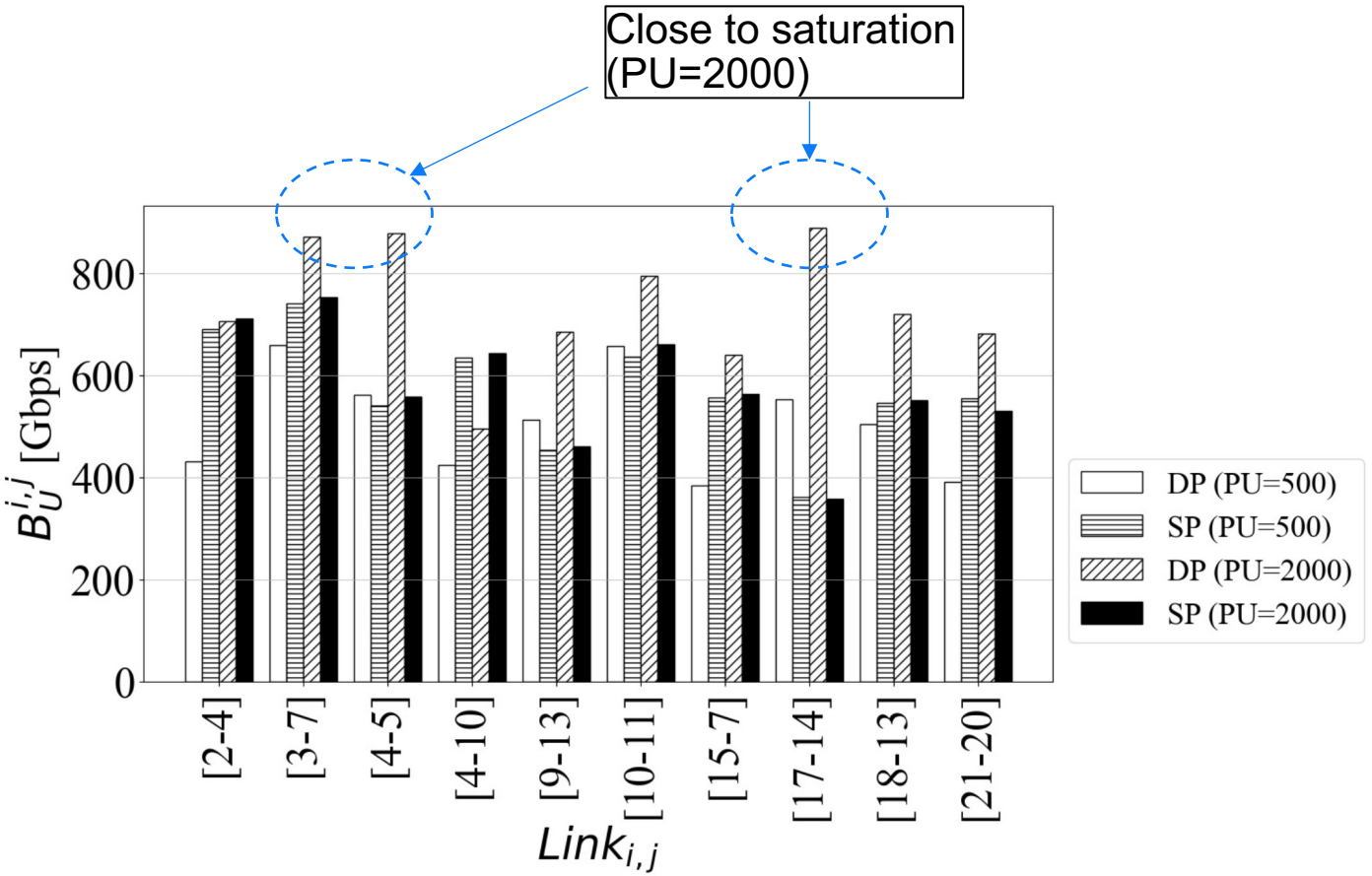


Fig. 5. Average bandwidth per link $i-j$ ($B_U^{i,j}$) with load = 200. Only links with utilization $> 60\%$ are reported.

Compute savings: SP vs. DP

TABLE I
AVERAGE NUMBER OF PU REQUIRED PER SLICE (PU_S) AND SP SAVINGS.
DP DOES NOT ALLOW ANY SHARING.

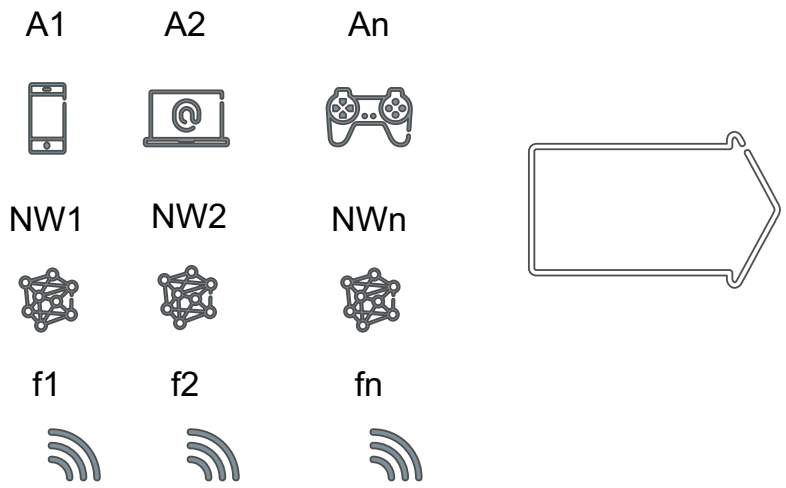
Load	DP	SP PU=500	Savings	SP PU=2000	Savings
150	24 PU	15.45 PU	35.62%	15.50 PU	35.41%
160	24 PU	15.39 PU	35.88%	15.48 PU	35.50%
170	24 PU	15.32 PU	36.16%	15.45 PU	36.61%
180	24 PU	15.25 PU	36.47%	15.43 PU	36.71%
190	24 PU	15.18 PU	36.75%	15.41 PU	36.79%
200	24 PU	15.11 PU	37.03%	15.40 PU	37.85%

Outline of the lecture

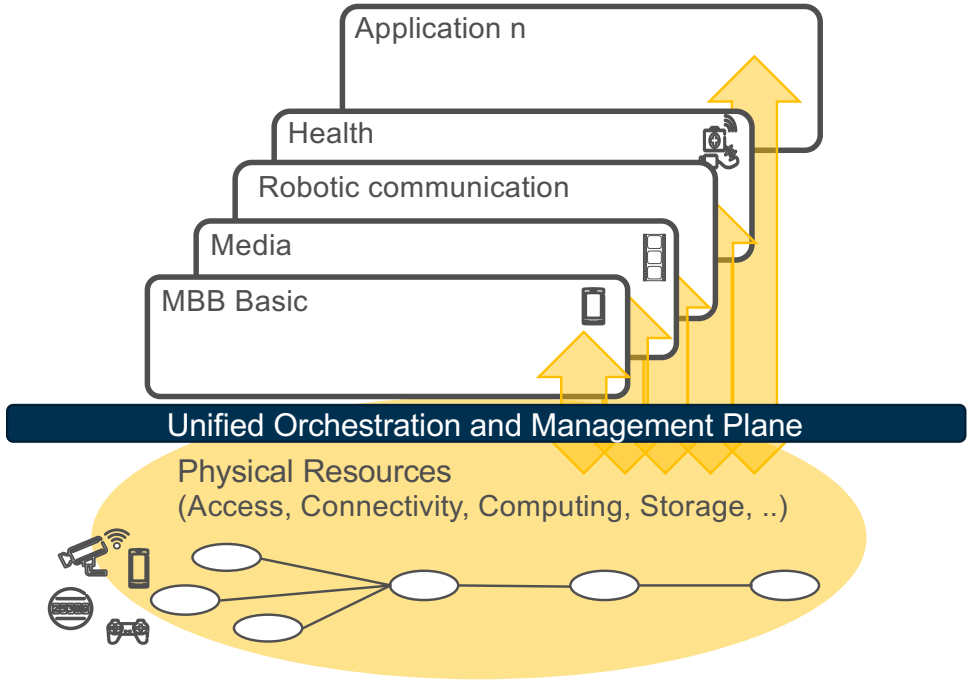
- 5G Network slicing
  - Overview and definition
  - Involved resources and standardization
  - Slice creation process
- Examples
  - Slice deployment in the cloud
  - Reliable resource provisioning
- A quick look at 6G

5G vision: one network – multiple industries

- From dedicated physical networks with dedicated control and dedicated services/resources for different applications…

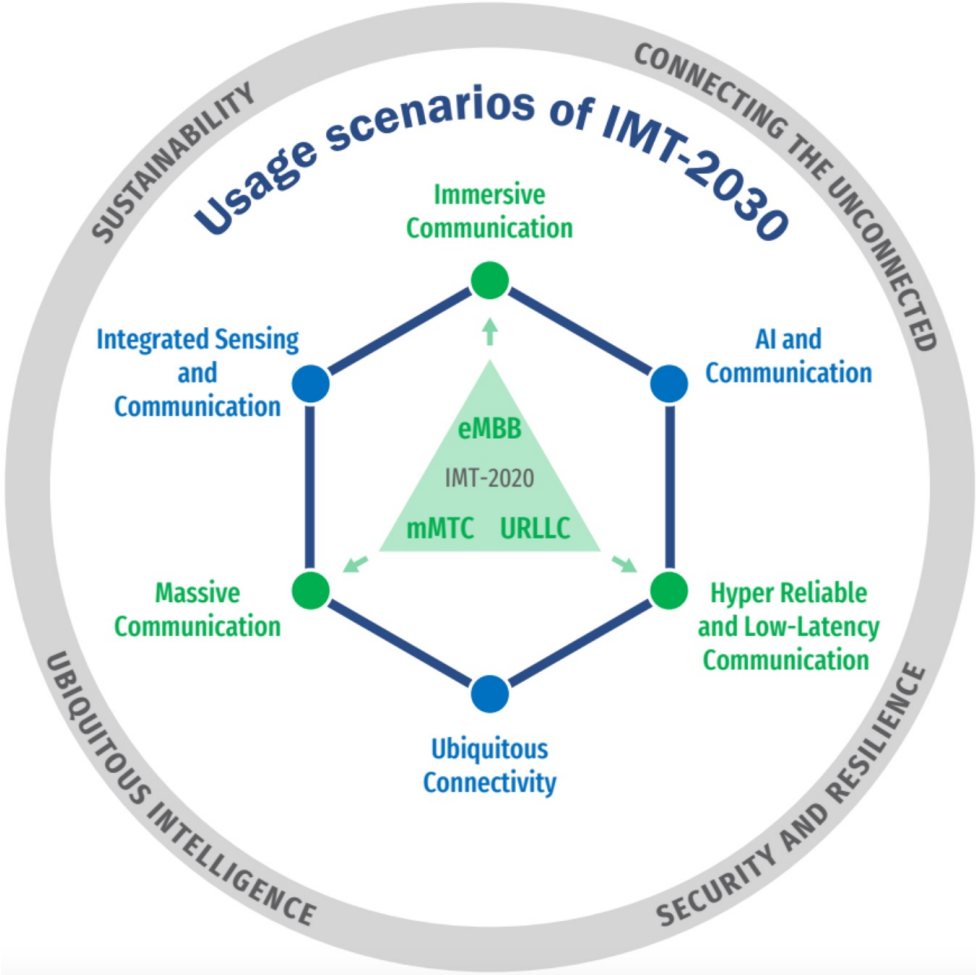


…to a "network factory" where resources and network functions are traded and provisioned: new infrastructures and services are "*manufactured by SW*"



Usage scenarios of IMT-2030

- Ubiquitous Connectivity: is intended to enhance connectivity with the aim to bridge the digital divide
- AI and Communication: support to distributed computing and AI applications (e.g., assisted automated driving, offloading devices, creation of and prediction with digital twins)
- Integrated Sensing and Communication: to offer multi-dimensional sensing that provides spatial information about unconnected objects as well as connected devices and their movements and surroundings



Recommendation ITU-R M.2160-0 (11/2023) - Framework and overall objectives of the future development of IMT for 2030 and beyond

Thank you!

SSY145 – Guest lecture on Network Slicing

Federico Tonini

CNIT - Consorzio Nazionale Interuniversitario per le Telecomunicazioni

federico.tonini@wilab.cnit.it