INSTITUTE
OF COMMUNICATION,
INFORMATION
AND PERCEPTION
TECHNOLOGIES

Scuola Superiore
Sant'Anna

WiSHFUL

How much is Fronthaul Latency Budget Impacted by RAN Virtualisation ?

H. Gupta[2], D. Manicone[1],
F. Giannone[1], K. Kondepu[1], A. Franklin[2],
P. Castoldi[1], L. Valcarenghi[1]

[1]Scuola Superiore Sant'Anna, Pisa, Italy
[2]Indian Institute of Technology Hyderabad, India

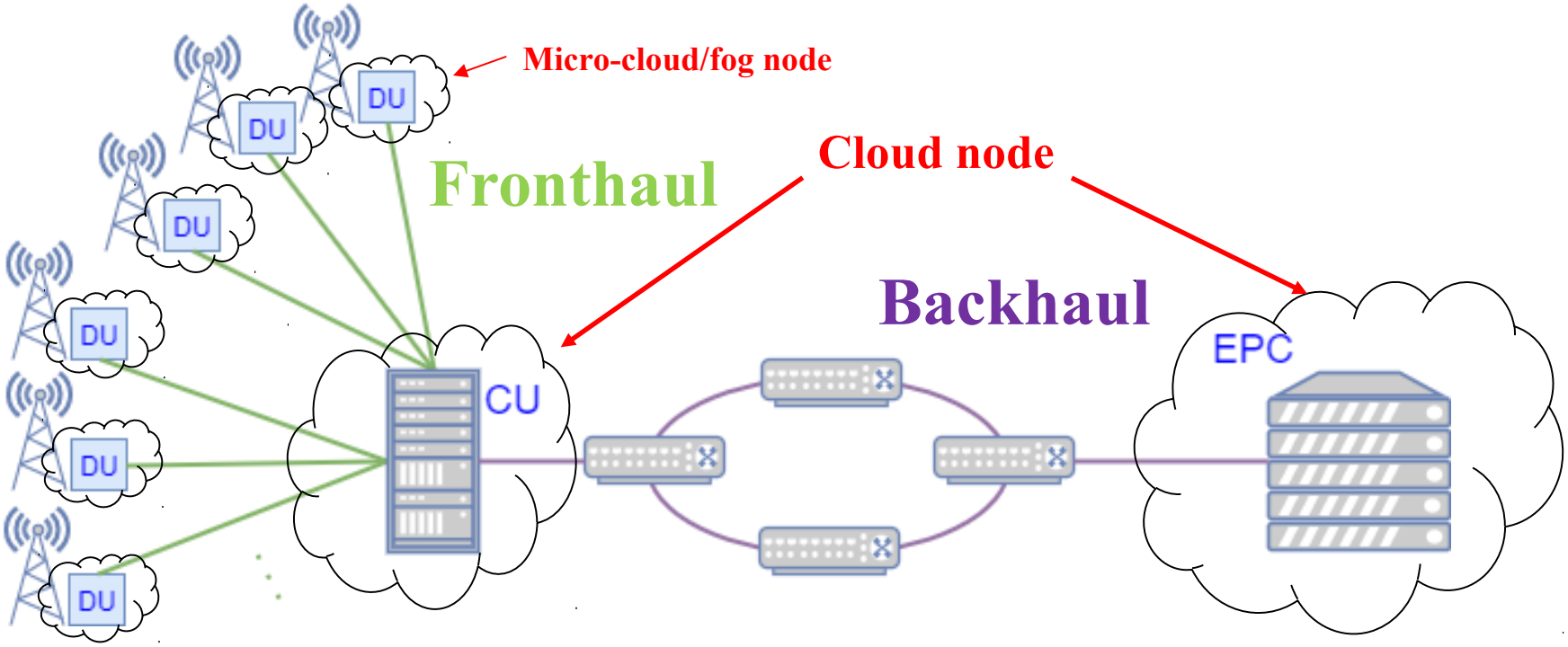
IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)
6-8 November 2017 – Berlin, Germany

Summary

- A Virtualized New Radio Acces Network
- Problem Description
- Investigated questions
- The Federated ARNO-5G Testbed
- Mobile Network Software - OpenAirInterface
- Performance Evalution Parameters
- Emulations of Latency and Jitter on the fronthaul
- Evaluation Scenario
- Considered Virtualisation Methods
- Experimental Results
- Conclusions

A New Virtualized Radio Access Network

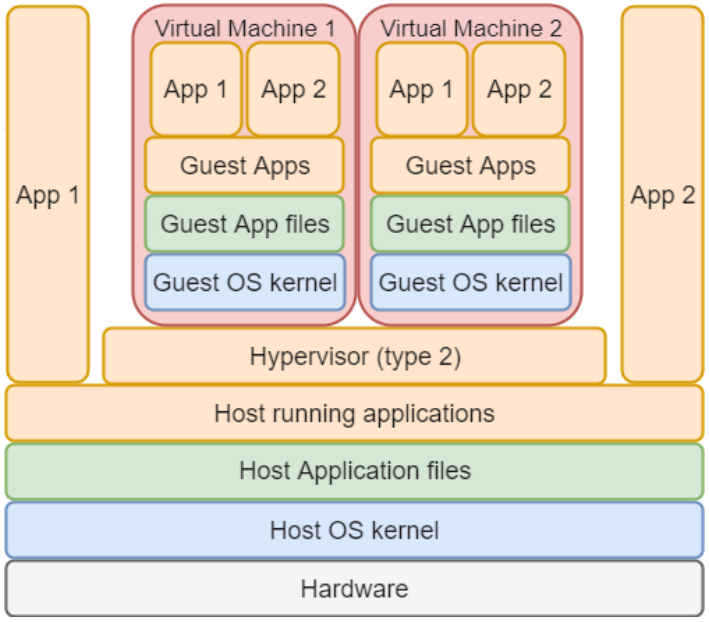
- A New Radio Access Network (New RAN) has been proposed to increase the performance with limited deployment costs



- eNodeB functional split
  - Distributed Unit (DU)
  - Central Unit (CU)
- RAN split
  - Fronthaul
  - Backhaul

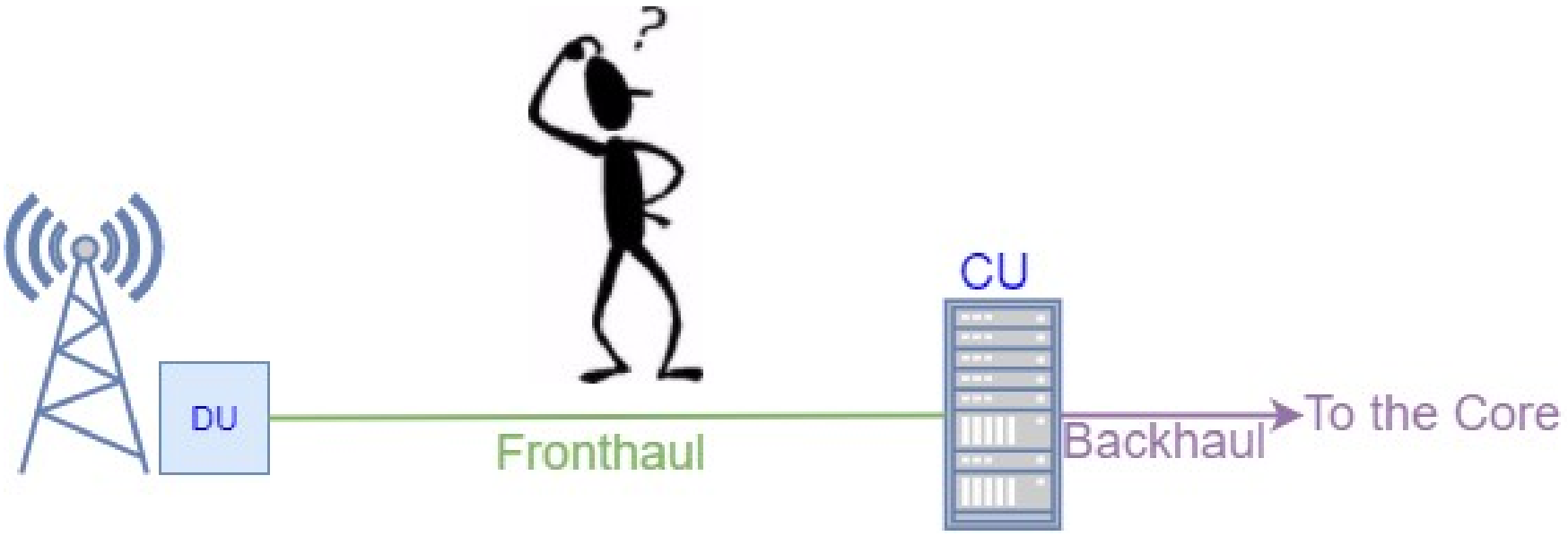
Problem description

- Different functional splits → different latency constraints (TR38.801):
  - ✓ Option 7a functional split max. allowed one-way latency → 250 [μs]
- What virtualization implies:
  - ✓ applications running in the guest host have “to cross” several layers of abstraction.
  - ✓ Extra levels of abstraction reduce workload performance
- Different virtualization types:
  - ✓ Hypervisor-based virtualizations:
    - ✓ allow to fully emulate a CPU architecture and OS;
  - ✓ Container-based virtualizations:
    - ✓ utilizes kernel features to create an isolated environment of the process using the host hardware.



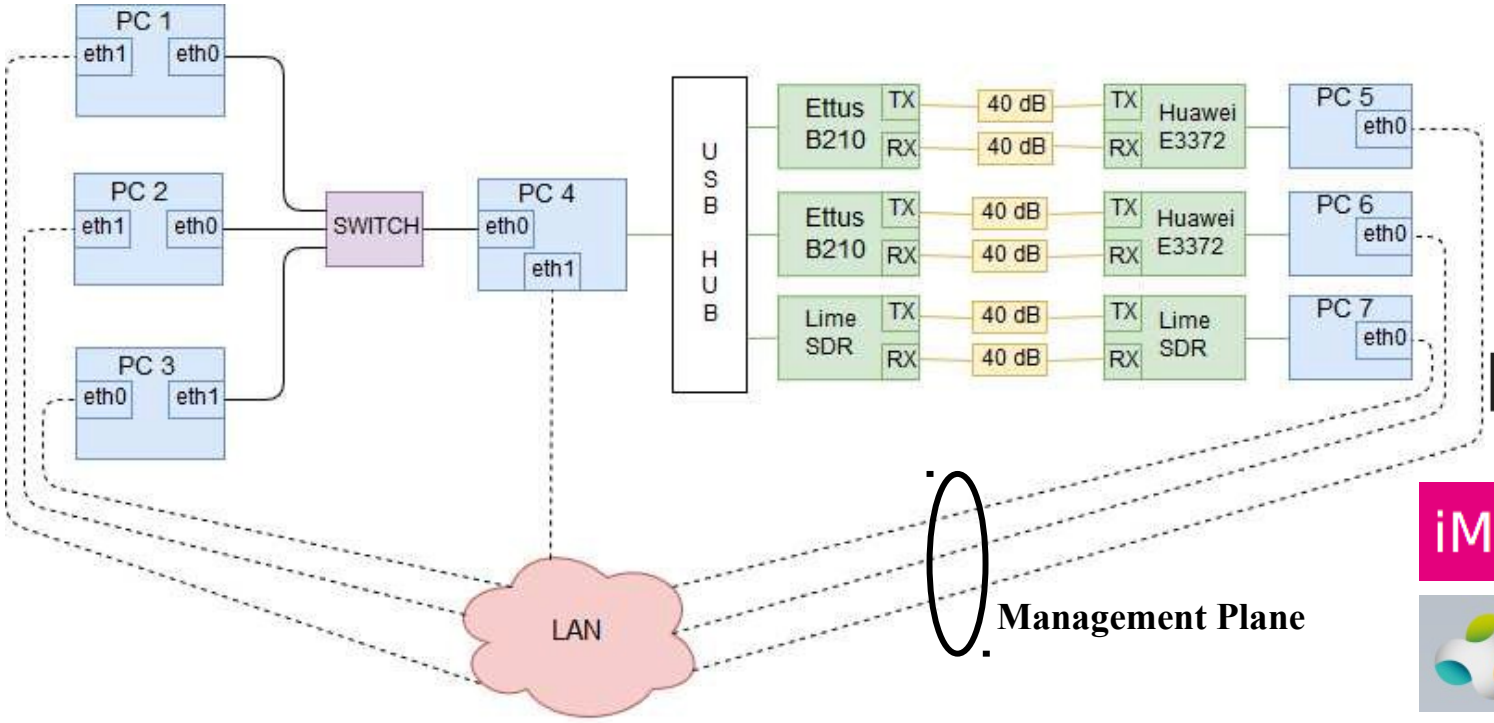
Investigated questions in this paper

1. Are the extra levels of abstraction impacting the fronthaul latency constraints?
2. Does the jitter impact the fronthaul link performance?



The Federated ARNO-5G Testbed

- In the ARNO-5G Testbed different virtualisation methods are considered in order to virtualize the EPC and the CU.



FED4FIRE

iMinds Authority

jFed Login

Devices Name	Devices Type	Processor Type	OS
PC 1	mini-pc (Up-board First Generation)	Intel Atom x5-Z8350 Quad Core Processor	Ubuntu 14.04 (4.7 kernel)
PC 2	Dell T410 PowerEdge desktop servers	Intel Xeon E5620	Ubuntu 14.04 (3.19 low-latency kernel)
PC 3	Dell T410 PowerEdge desktop servers	Intel Xeon E5620	Ubuntu 14.04 (3.19 low-latency kernel)
PC 4	Mini-ITX	Intel I7 7700 Quad Core (@ 4.0GHz)	Ubuntu 14.04 (3.19 low-latency kernel)
PC 5	mini-pc (Up-board First Generation)	Intel Atom x5-Z8350 Quad Core Processor	Ubuntu 14.04 (4.7 kernel)
PC 6	mini-pc (Up-board First Generation)	Intel Atom x5-Z8350 Quad Core Processor	Ubuntu 14.04 (4.7 kernel)
PC 7	Desktop Computer	Intel I7 7700 Quad Core (@ 4.0GHz)	Ubuntu 14.04 (3.19 low-latency kernel)

Mobile Network Software – OpenAirInterface

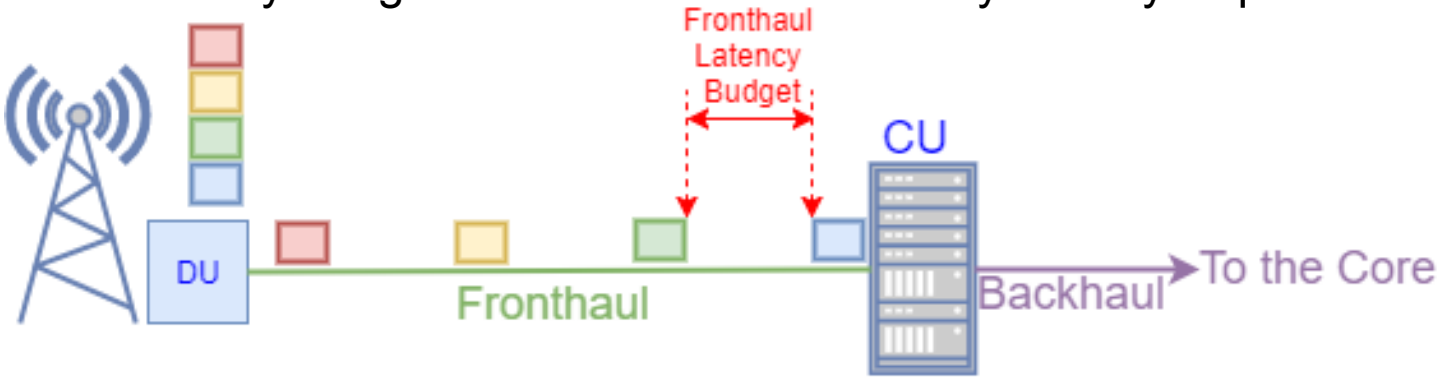
- For the Core and RAN implementation the OpenAir interface softwae are used.

Core Implementation	RAN Implementation
• openair-cn • Implements the EPC 3GPP specs • Contains the implemention of: ✓ Home Subscriver Server (HSS) ✓ Mobile Management Entity (MME) ✓ Serving Gateway (S-GW) ✓ PDN Gateway (PDN-GW)	• openairinterface5g • Implementation of Rel 10 LTE of: ✓ Evolved NodeB (eNB); ✓ User Equipment (UE). • Implemented functional splits options: ✓ IF4p5 → Option 7-1 (intra-PHY split) ✓ IF5 → Option 8 (PHY-RF split)

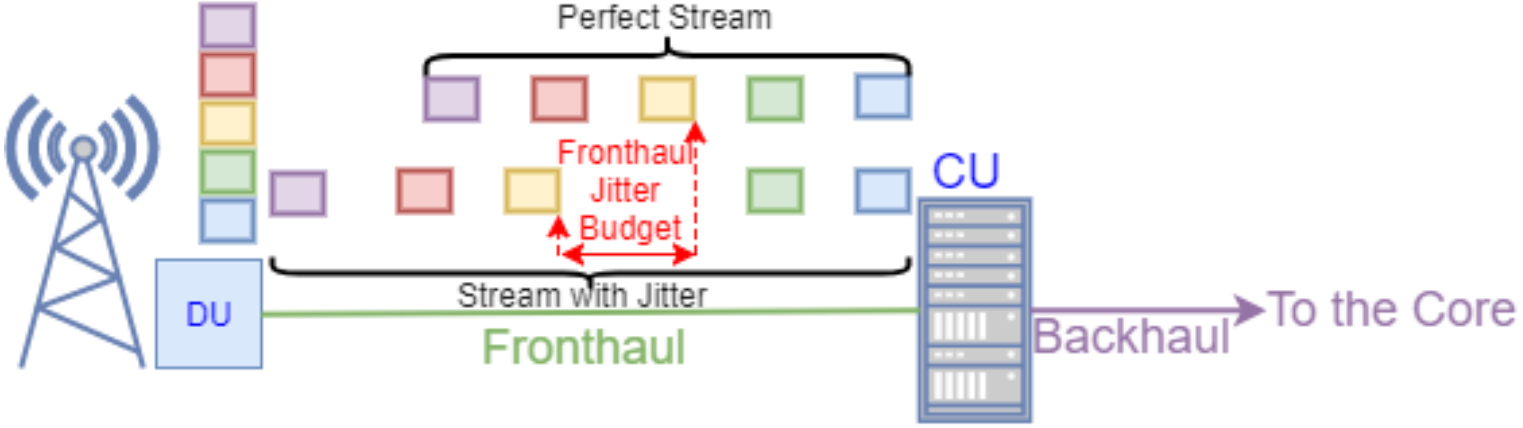
	Option 7-1	
	Uplink Direction	Downlink Direction
DU	FFT, CP removal and PRACH filtering	IFFT, CP addition and PRACH filtering
CU	Rest of PHY functions and the higher layers	Rest of PHY functions and the higher layers

Performance Evaluation Parameters

- When virtualised EPC and CU are considered a experimental evalution in Option 7-1 functional split scenario of the following parameters are performed:
  - ✓ Allowable Latency budget supported by the fronthaul;
  - ✓ Allowable Jitter budget supported by the fronthaul;
- The fronthaul latency budget is defined as the one-way latency requirement:



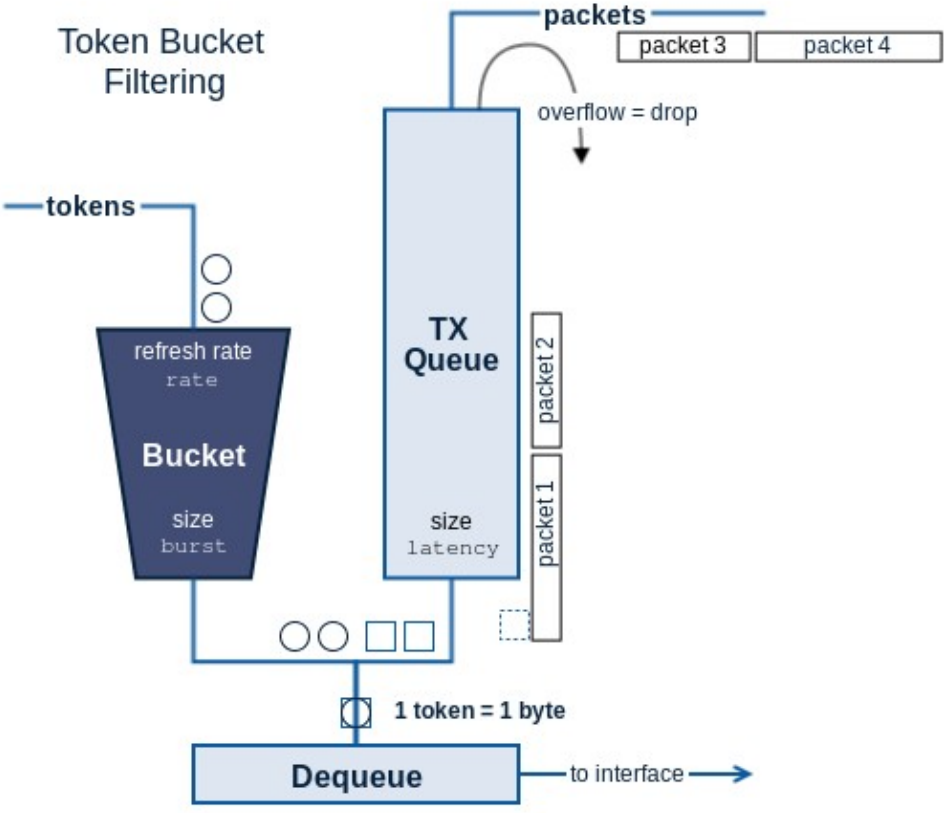
- The fronthaul jitter budget is defined as the maximum supported latency variation:



Emulations of Latency and Jitter on the fronthaul

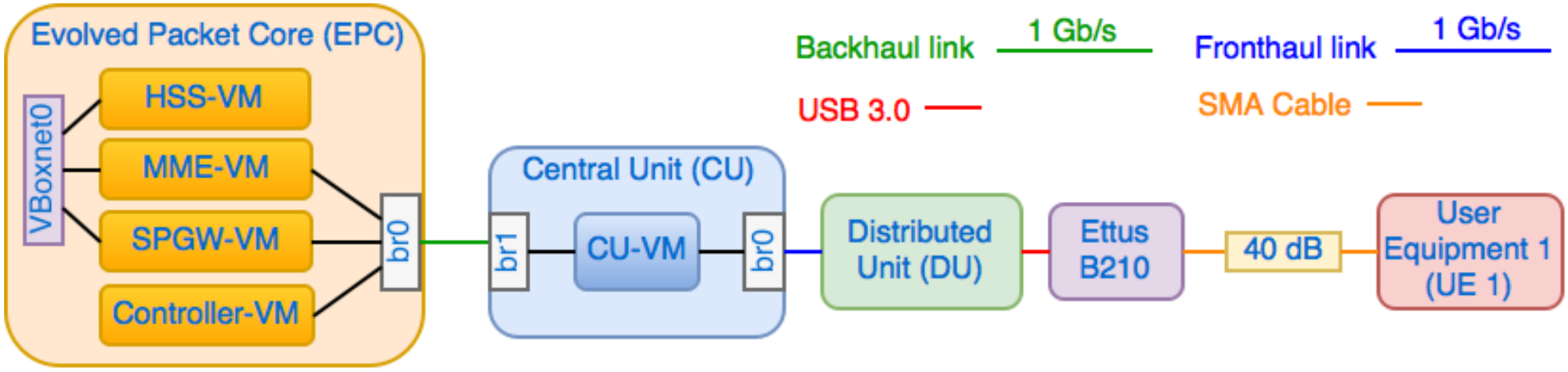
- The linux utility traffic control «tc-netem» is used

- A delay d0 is applied to the DU Ethernet interface towards the CU.
- A delay d1 is applied to the CU Ethernet interface towards the DU.
- Evaluation of the frontahul latency budget:
  - ✓ d0 and d1 are encreased with steps of 10 μs until DU, CU and UE disconect.
- Evalution of the fronthaul jitter budget:
  - ✓ A jitter following a normal distribution is added to the latency values d0 and d1 with steps of 10 μs.



Source: https://www.excentis.com/blog/use-linux-traffic-control-impairment-node-test-environment-part-2

Evaluation Scenario

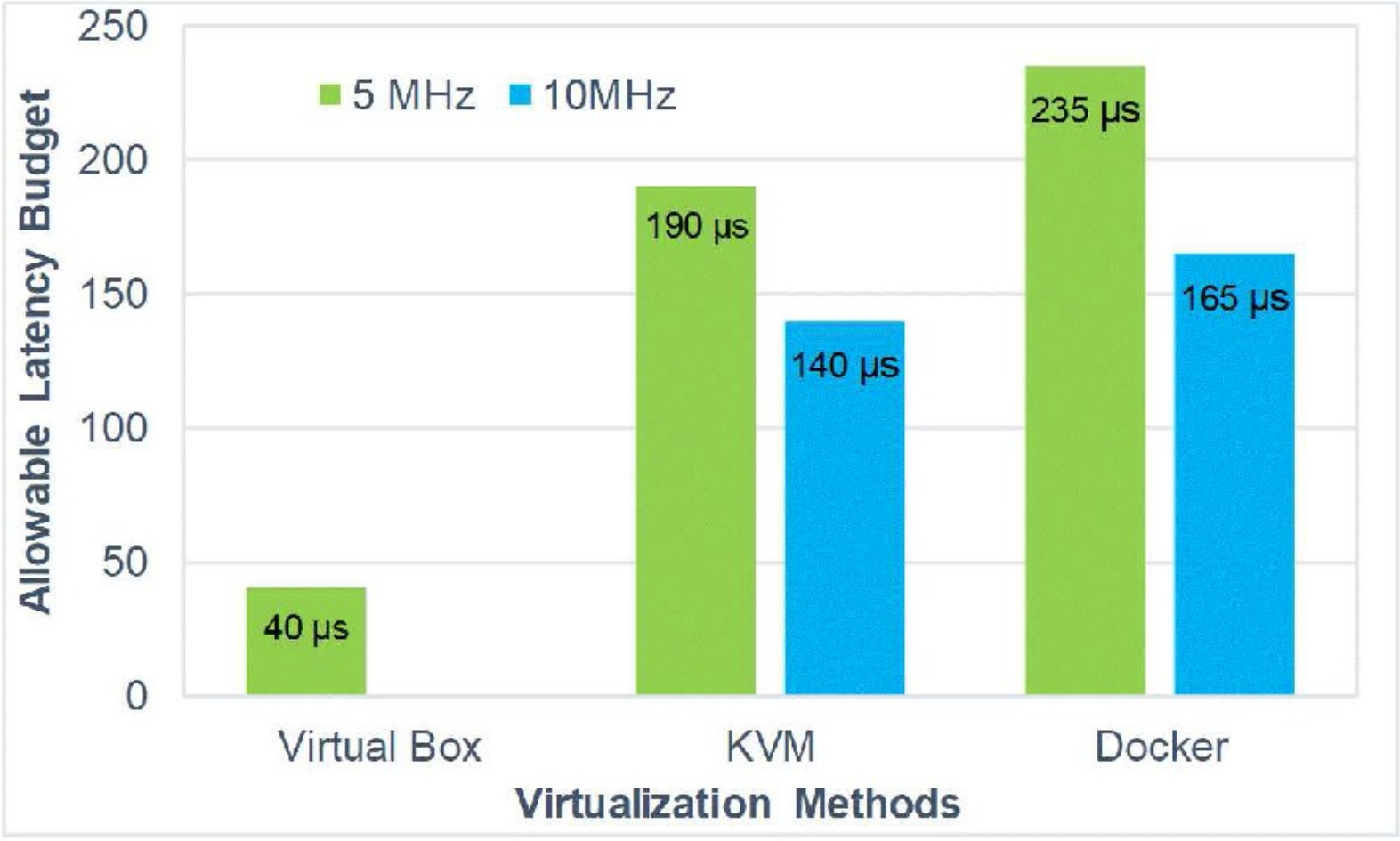


- The virtualized EPC:
  - ✓ The Mobile Management Entity (MME) is deployed in a VM (MME-VM);
  - ✓ The Home Subscriber Server (HSS) is deployed in a VM (HSS-VM);
  - ✓ The Serving Gateway (S-GW) and the PDN-Gateway (P-GW) are deployed in a VM (SPGW-VM).
- The virtualized Central Unit (CU) is deployed in a VM (CU-VM).
- The Distributed Unit (DU) runs directly in the physical machine.
- The User Equipment (UE) is deployed by means of a Huawei E3372 dongle attached to a PC. The UE is connected to the RAN through SMA cables with 40 dB of attenuation

Considered Virtualisation Methods

- Hypervisor-based virtualization and Container-based virtualization are analyzed.
- Considered Hypervisor-based virtualization methods:
  - ✓ VirtualBox;
  - ✓ Kernel-Based Virtual Machine (KVM);
- Considered Container-based virtualization method:
  - ✓ Docker Container.
- Using VirtualBox and KVM virtualisation methods:
  - ✓ The HSS-VM, MME-VM and SPGW-VM are created with the following characteristics:
    - ❖ Ubuntu 16.04 (4.8 generic kernel);
    - ❖ 1 core virtual CPU and 1 GB of RAM.
  - ✓ The CU-VM is created with the following characteristics:
    - ❖ Ubuntu 14.04 (3.19 low-latency kernel);
    - ❖ 8 core virtual CPU and 16 GB of RAM.
- Using the Docker Container virtualisation methods:
  - ✓ A Container is created in a physical machine for the deployment of the EPC and the elements belonging to it (i.e. MME, HSS, SPGW);
  - ✓ A Container is created in a second physical machine for the deployment of the CU.

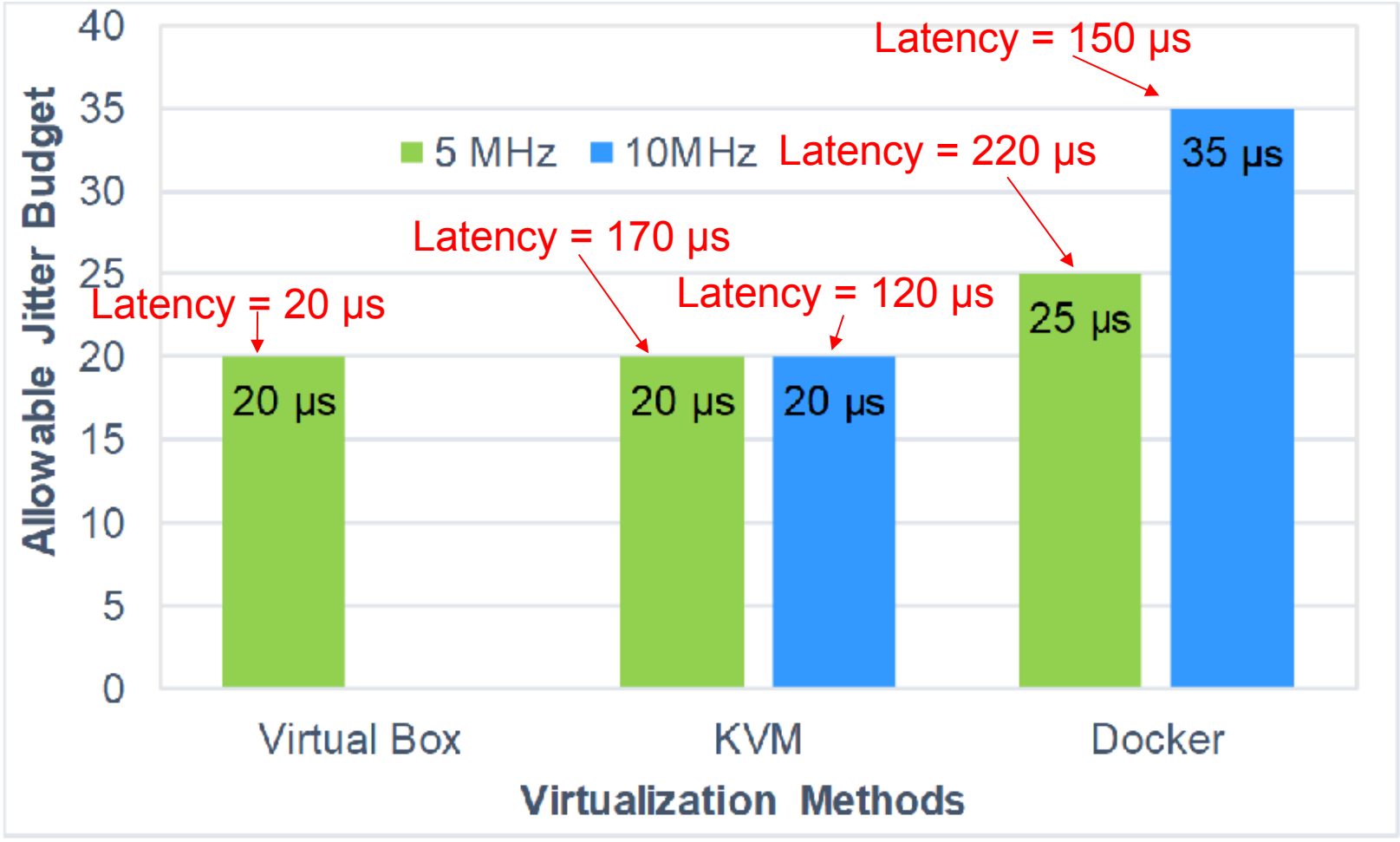
Experimental Results – Allowable Latency Budget



- Using Virtual Box the fronthaul allowable latency budget is very low.
- Using KVM and Docker Container the fronthaul allowable latency is close to the 3GPP constraints.

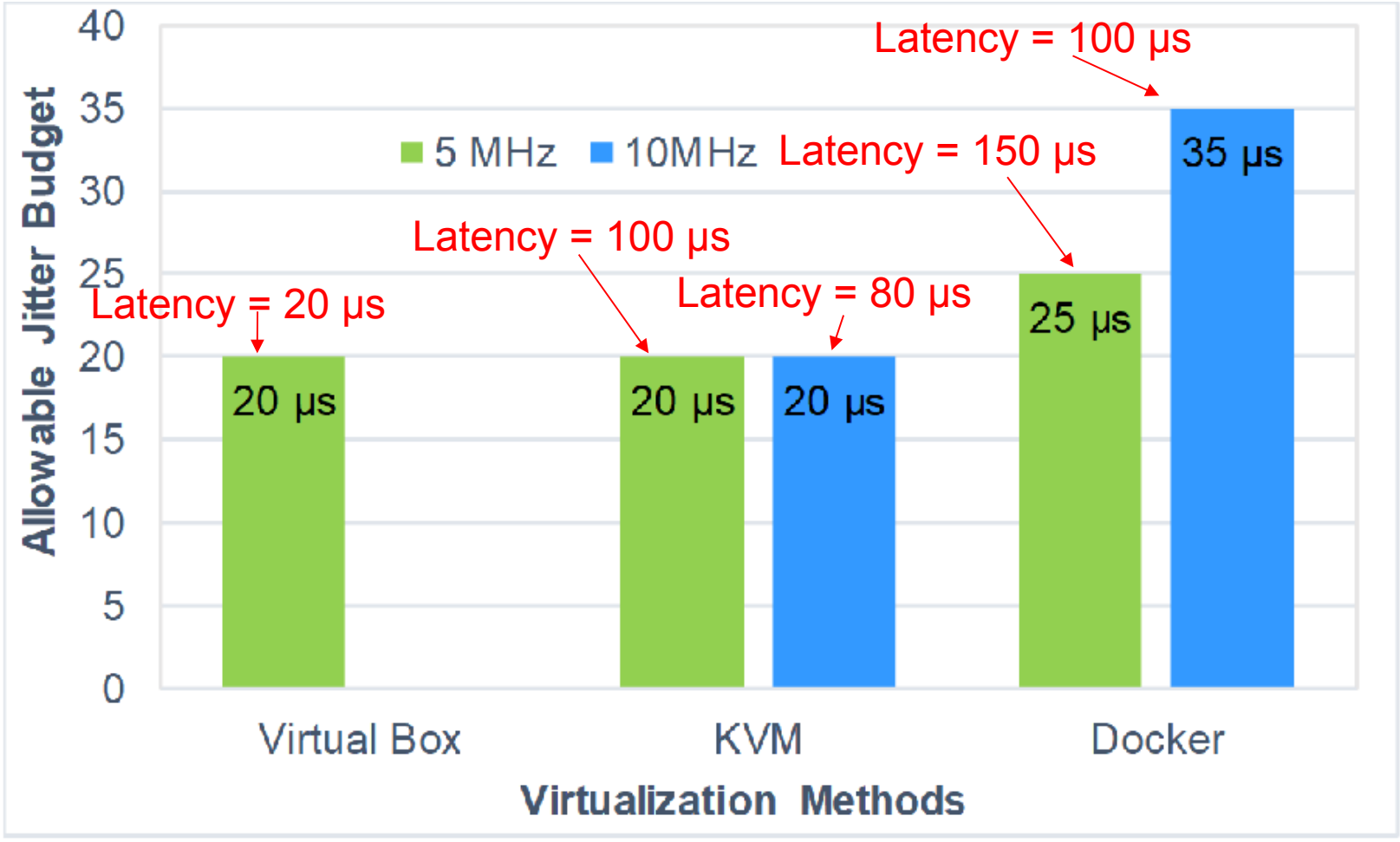
Experimental Results – Allowable Jitter Budget (1)

- The Jitter is applied to a latency value close to the fronthaul allowable latency budget.
- The fixed latency value is choosen according to the Virtualization Methods and the signal bandwidth



Experimental Results – Allowable Jitter Budget (1)

- The Jitter is applied to a latency value far to the fronthaul allowable latency budget.
- The fixed latency value is choosen according to the Virtualization Methods and the signal bandwidth



Conclusions

- Experimental evaluation of the impact of virtualizing eNB functions on the fronthaul latency and jitter budget are performed.
- Functional split Option 7-1 (i.e. intra-PHY) are applied.
- Different Virtualisation methods are considered:
  - ✓ VirtualBox;
  - ✓ KVM;
  - ✓ Docker Container.

✓ The lighter virtualisation methods (e.g. Docker Container) impact the fronthaul latency budget less than heavier virtualisation methods (i.e. VirtualBox).

✓ The fronthaul latency bandwidth reduction depends on the considered signal bandwidth (i.e. 5 MHz, 10 MHz).

✓ The performed experimental evaluation showed that a jitter of at most 40 us can be tolerated.

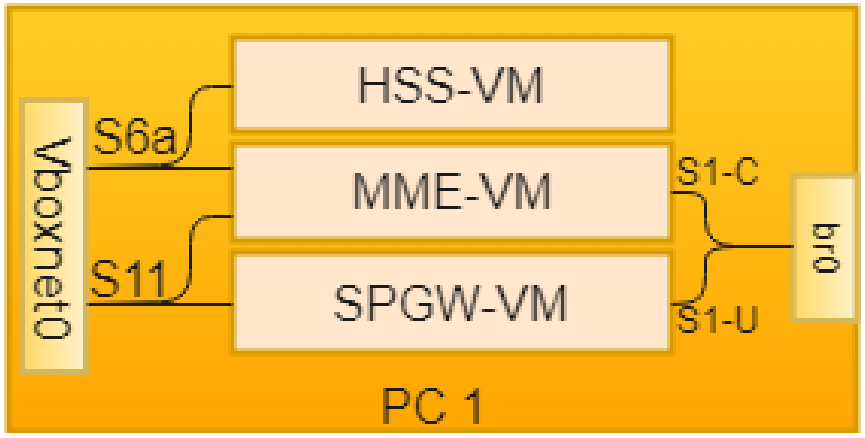
thank you!

email:
francesco.giannone@santannapisa.it

This work has been partially funded by the H2020-ICT-2014-1 Wishful project (grant no. 645274).

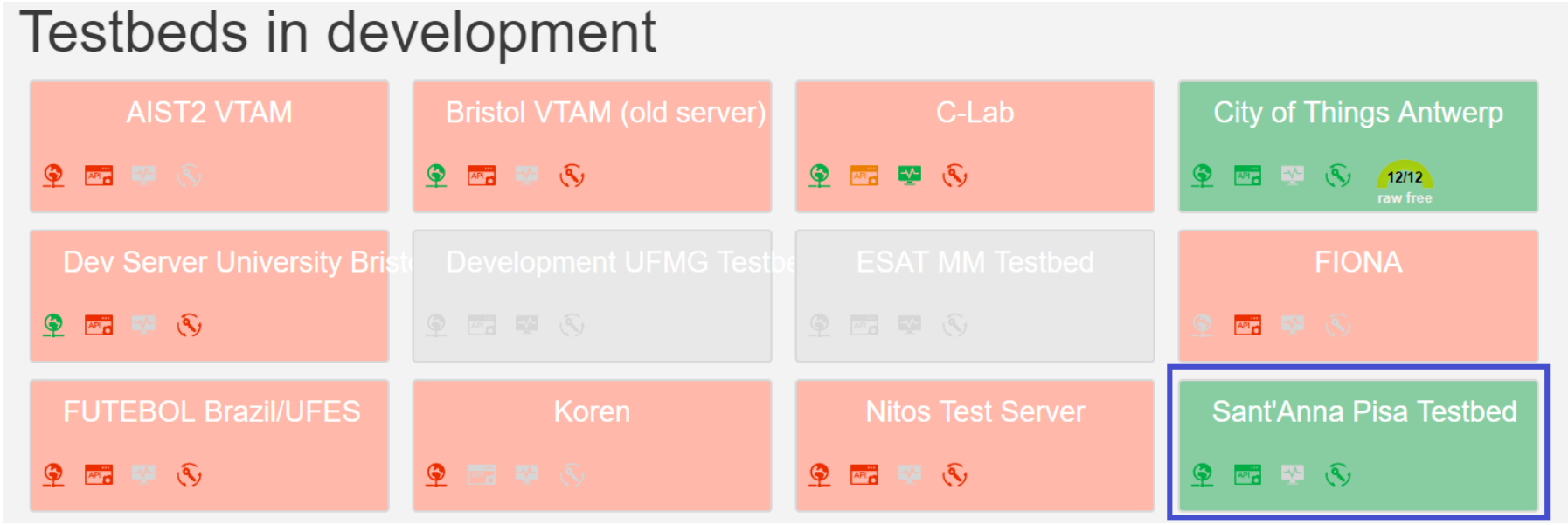
Backup Slides

Virtualized EPC and CU Network configuration



- HSS-VM:
  - ✓ 1 Virtual Interfaces in host-only networking (s6a interface);
- MME-VM:
  - ✓ 1 Virtual Interfaces in host-only networking mode (S6a interface);
  - ✓ 1 Virtual Interfaces in bridge networking mode (S1-C interface);
- SPGW-VM:
  - ✓ 1 Virtual Interfaces in host-only networking mode (S11 interface);
  - ✓ 1 Virtual Interfaces in bridge networking mode (S1-U interface)
- CU-VM:
  - ✓ 1 Virtual Interfaces in bridge networking mode (S1-U and S1-C interfaces);
  - ✓ 1 Virtual Interfaces in bridge networking mode (fronthaul link).

Federation of the ARNO-5G Testbed



- ARNO-5G Testbed:
  - ✓ Is federated in Fed4FIRE federation;
  - ✓ Accepts only trusted users from iMinds.
- Therefore, experimenters can:
  - ✓ Acces and reserve resources from multiple testbeds via jFed;
  - ✓ Configure experimentes interconnecting such resources.

iMinds Authority

jFed Login

INSTITUTE OF COMMUNICATION, INFORMATION AND PERCEPTION TECHNOLOGIES

Scuola Superiore Sant'Anna

FED4FIRE

Reserve Resources in ARNO-5G Testbed

- Through jFed tool an experimenter can:
  - ✓ Select the ARNO-5G Testbed, namely «Sant'Anna Pisa Testbed»;
  - ✓ Provide his slice name
- In this way a Docker Container in ARNO-5G Testbed is created;
- The ARNO-5G devices are now accessible;
- Each OAI component of ARNO-5G Testbed are reachable.
  - ✓ Through SSH based on the specific container.
- More details on how to reserve the components of ARNO-5G Testbed can be found in the ARNO-5G Testbed web page.