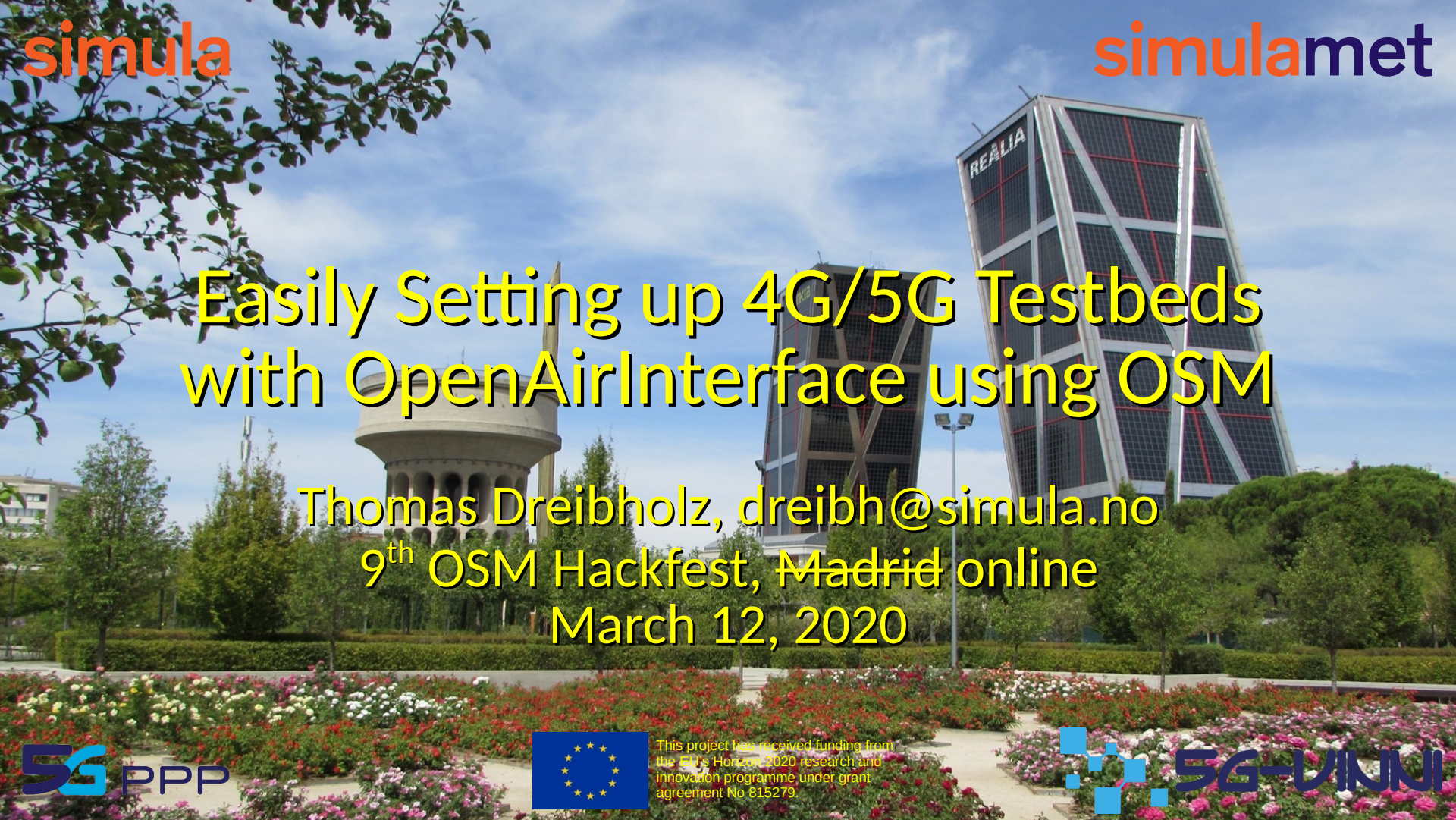


simula

simulamet

Easily Setting up 4G/5G Testbeds with OpenAirInterface using OSM

Thomas Dreibholz, dreibh@simula.no

9th OSM Hackfest, ~~Madrid~~ online

March 12, 2020

This project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 815279.

5G PPP

5G-VINNI

Table of Contents

OpenAirInterface (OAI)

- OpenAirInterface (OAI):
  - Open Source software for EPC and eNodeB (i.e. packet core and base stations)
  - Details: https://www.openairinterface.org
  - 4G LTE available, 5G under development
  - Ongoing work, with many different Git branches
- Idea:
  - Manage OAI setups in OSM (at least, the EPC part)
    - Automatic setup and deployment
  - Easy to add additional features (e.g. Mobile Edge Computing components)
  - Open Source, of course! → https://github.com/simula/5gvinni-oai-ns

Setting Up a 4G/5G Testbed

- Hardware:
  - User Equipment (modems, smartphones, etc.)
  - Programmable sim cards
  - Software-Defined Radio boards

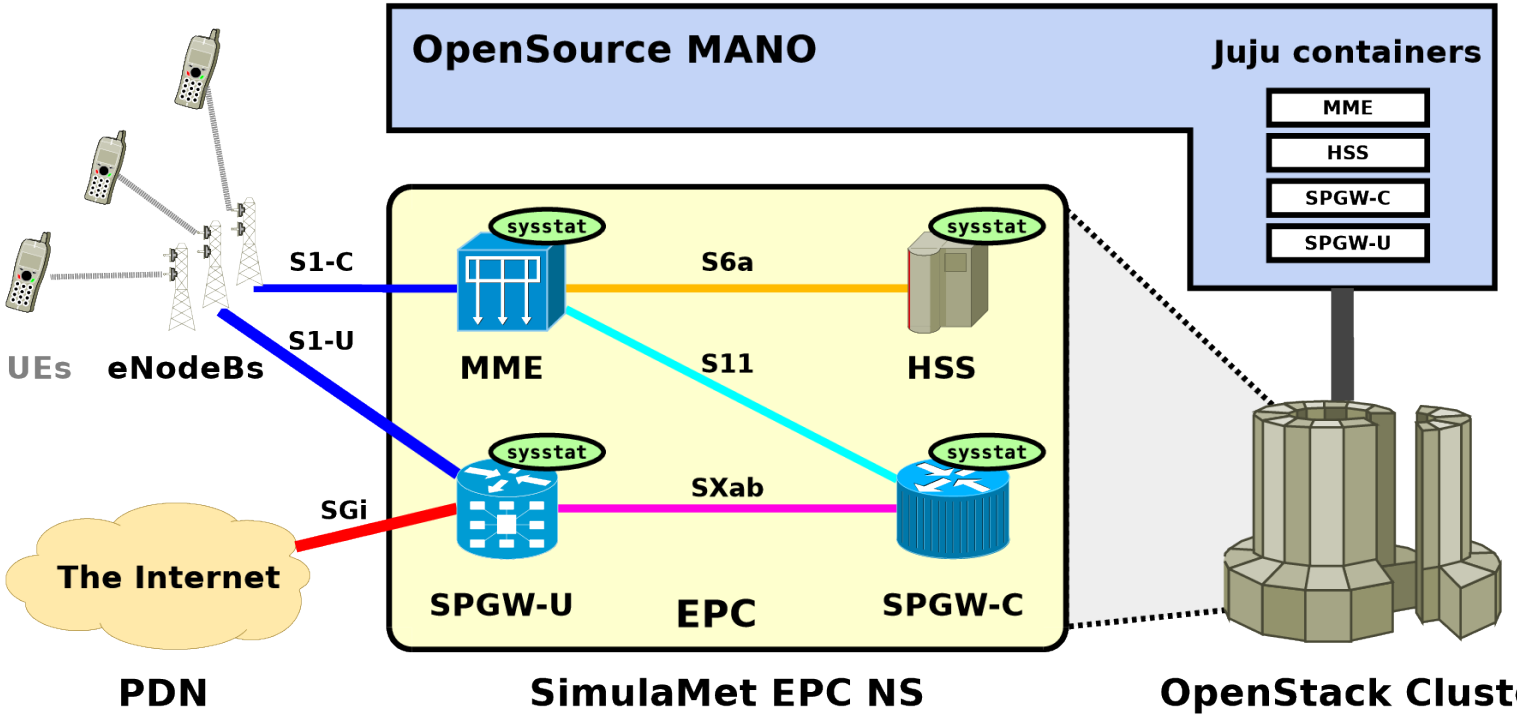
- For the rest (eNodeBs, EPC):
  - OpenAirInterface Open Source software
  - Running on regular Linux PCs
  - **But: difficult to install and maintain!**

5G-VINNI

Our Goal: An OpenAirInterface VNF

- Main purpose: testbed setups for research and development
- OAI EPC as VNF
  - Easy to use, EPC should (hopefully) work “out of the box”
  - Build of OAI software inside VMs, according to specified Git repositories and commits ⇨ get exactly the desired installation
- NSs using the VNF and possibly other VNFs
  - Example 1: add Mobile Edge Computing services to EPC
  - Example 2: get basic EPC to test extended eNodeB software
  - …

Basic Testbed Setup



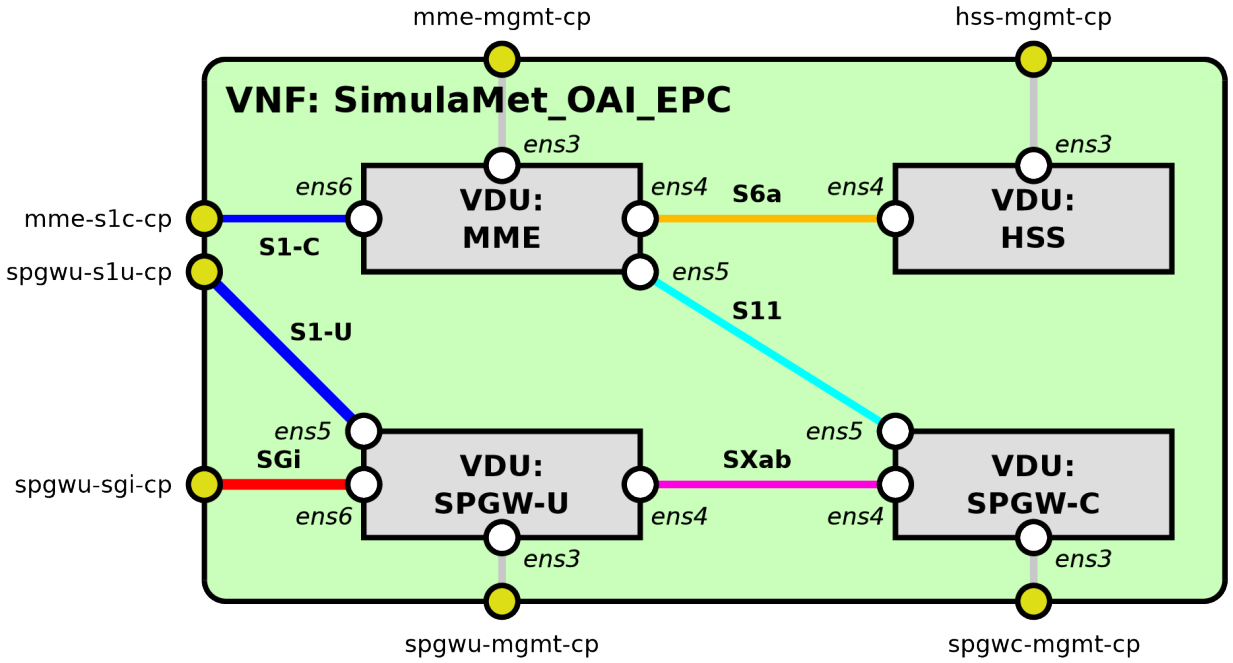
What is needed for the VNF?

- Base VDU image
- The VNF itself
- Juju Charms to configure the components
- Management of the build process

Base VDU Image

- VDU image goals:
  - Full-featured base VDU image, including development and debug tools
  - Different versions of Ubuntu LTS (Xenial, Bionic, Focal)
  - Up-to-date (i.e. all updates installed)
- Preseeding script:
  - Fully automatic Ubuntu installation from scratch (using virt-install)
  - Preseed configuration to include all necessary base packages
  - “late_command”: mainly work-arounds for bugs in the Ubuntu installer
    - add PPAs, update keyboard layout, ensure updates are installed

The SimulaMet EPC VNF



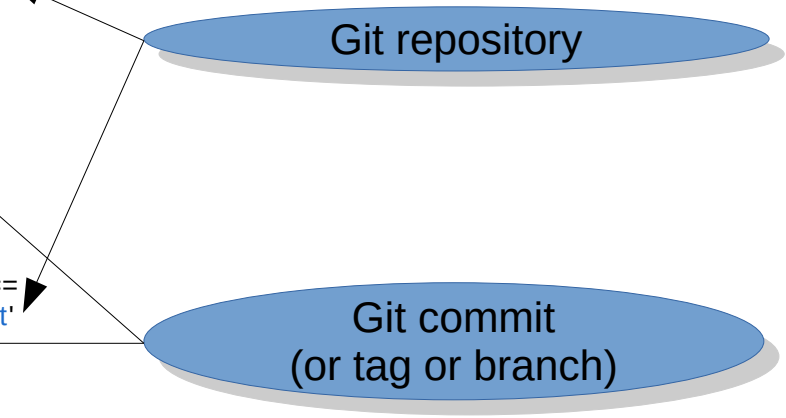
HSS: Home Subscriber Server
MME: Mobile Management Entity
SPGW-C:
Control Plane of the
Packet Data Network Gateway
SPGW-U:
User Plane of the
Packet Data Network Gateway

VNF Parameters Example

- # ====== HSS ==========
  hss_git_repository: 'https://github.com/OPENAIRINTERFACE/openair-cn.git'
  hss_git_commit: 'dreibh/cassandra-build-fix-17feb2020'
  hss_S6a_address: '172.16.6.129'
  network_realm: 'simula.nornet'
  network_k: '449C4B91AEACD0ACE182CF3A5A72BFA1'
  network_op: '1006020F0A478BF6B699F15C062E42B3'
  network_imsi_first: '242881234500000'
  network_msisdn_first: '24288880000000'
  network_users: '1024'

- # ====== MME ==========
  mme_git_repository: 'https://github.com/OPENAIRINTERFACE/openair-cn.git'
  mme_git_commit: '2019.w45'
  mme_S1C_ipv4_interface: '192.168.247.102/24'
  mme_S1C_ipv4_gateway: '0.0.0.0'
  mme_S1C_ipv6_interface: ''
  mme_S1C_ipv6_gateway: ''
  mme_S11_ipv4_interface: '172.16.1.102/24'
  mme_S6a_address: '172.16.6.2'
  network_mcc: '242'
  network_mnc: '88'

- ...



Configuration with Juju

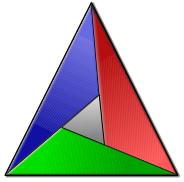
- Day-0/1: For each VDU (EPC component, i.e. HSS, MME, SPGW-C, SPGW-U):
  - Install necessary additional packages (depends on component)
  - Set up network configuration
  - Clone component sources (Git repository and commit)
  - Build the sources
  - Create/update component's configuration files
  - Write systemd unit file (for "sudo service <component> start|stop|restart")
  - Start the component
- Day-2: actions to start/stop/reconfigure components

Juju Proxy Charm Challenges

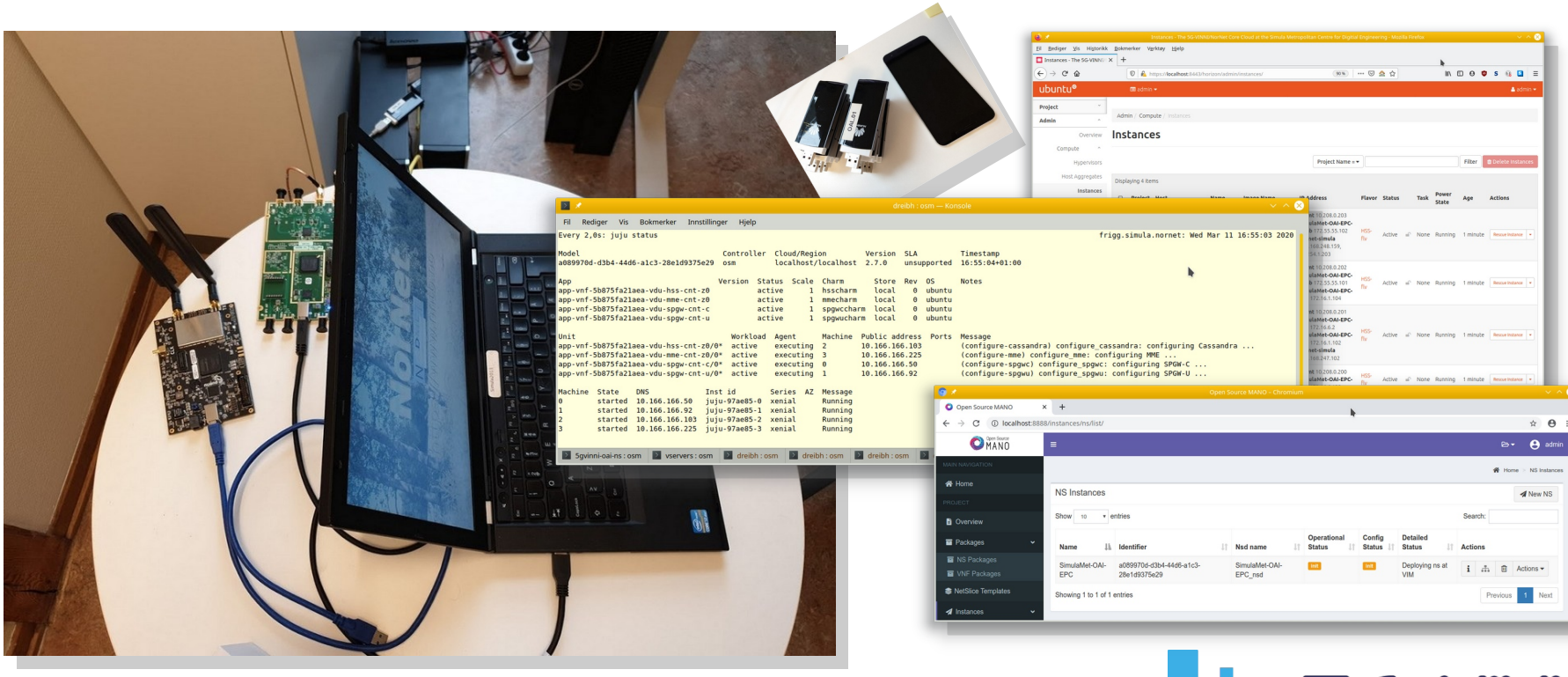
- charms.sshproxy._run(*COMMAND STRING*)
- String is processed in Python, then it is processed by ssh/bash shell of VDU
  - Escaping/double escaping required:
    - $ ⇨ \$; \ ⇨ \\ ; " ⇨ \\\" ; \" ⇨ \\\\\\"
    - Result: writing charm commands gets ugly and error prone
  - Wishlist: automatic escaping!
- Juju errors passed to OSM are usually not very helpful
  - Something went wrong, but not saying what went wrong
  - Wishlist: improved error reporting!

Managing VNFD/NSD Builds

- Multiple manual steps to generate and deploy VNFs and NSs
  - Strictly verify all YAML files with yamllint (useful, to avoid problems!)
  - Copy Charm files to VNFDs and build Charms (charm build ...)
  - Verify descriptor(s) and generate VNFD package(s) (validate_descriptor.py, generate_descriptor_pkg.sh)
  - Verify descriptor(s) and generate NSD package(s)
- Initial approach: write a Makefile
- Better approach:
  - Use Git for source management ⇨ information about all relevant source files
  - Use CMake to write Makefiles and take care of dependencies!



Demo



5G-VINNI

Sources

- Get the sources here: https://github.com/simula/5gvinni-oai-ns
  - Open Source, GPL-licensed
  - README: how to set up a testbed
  - images/: VDU preseeded image build script
  - juju/: The Juju Charms used by the VNF
  - SimulaMet-OAI-EPC_vnfd/: VNF descriptor
  - SimulaMet-OAI-EPC_nsd/: NS descriptor for simple example

simulamet

simula

5G-VINNI

Any Questions?

Thomas Dreibholz

dreibh@simula.no

https://www.simula.no/people/dreibh

This project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 815279.