5GCity

WEBINAR

Webinar info

Contents

- **Introduction to the 5GCity project**
- **Technology enablers: NFV and SDN**
- **The 5GCity platform**: architecture, components and dashboard
- **Service design with the 5GCity SDK**
- **Unikraft**: toolkit for crafting unikernels

Speakers

Shuaib Siddiqui

from **i2Cat**
Deputy 5GCity
Project Manager

Gino Carrozzo

from **Nextworks**
5GCity
Technical Manager

Apostolos Papageorgiou

from **i2Cat**
Resp. 5GCity
orchestration platform

Leonardo Agueci

from **Nextworks**
5GCity SDK software team

Santiago Pagani

from **NEC Europe**
Unikernel/Unikraft team

Introduction to the 5GCity project

Shuaib Siddiqui

from **i2Cat**

Deputy 5GCity Project Manager

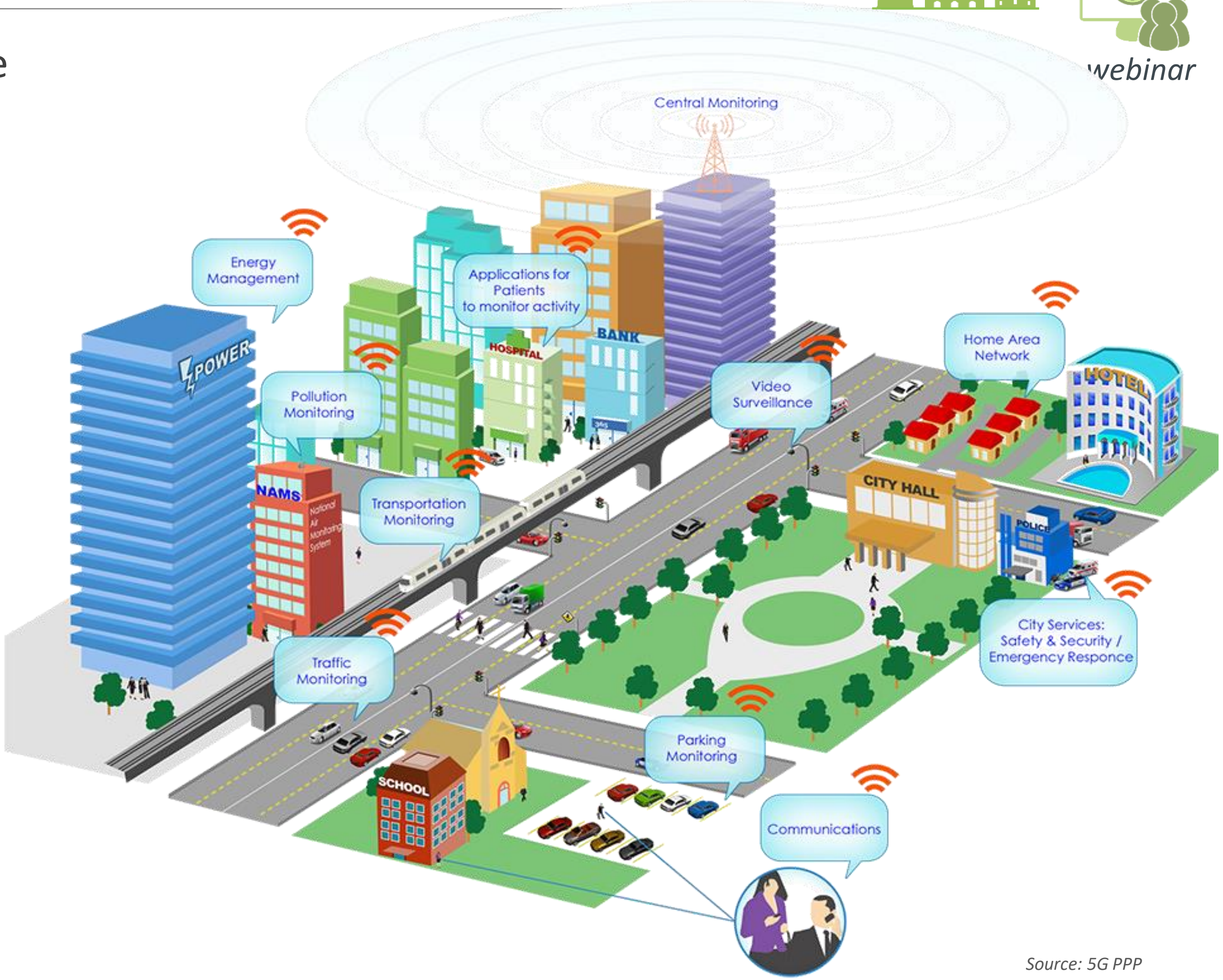
Cities, a flourishing business ecosystem for 5G

Futuristic scenarios (beyond connectivity) are becoming today's service offering

- Media, Entertainment, Tourism services
  - AR/VR for immersive visits
  - UHD distribution
  - Participatory media in public events/places
- Smart City services
  - Public safety & video-surveillance for security
  - Transportation and environment monitoring
  - Energy management
- Connected cars
  - Traffic monitoring and optimal routing

New business opportunities for traditional & new stakeholders

- Telcos/NSPs, Vendors
- Virtual Service Providers, Service brokers
- **SMEs & start-ups**
- **Municipalities**



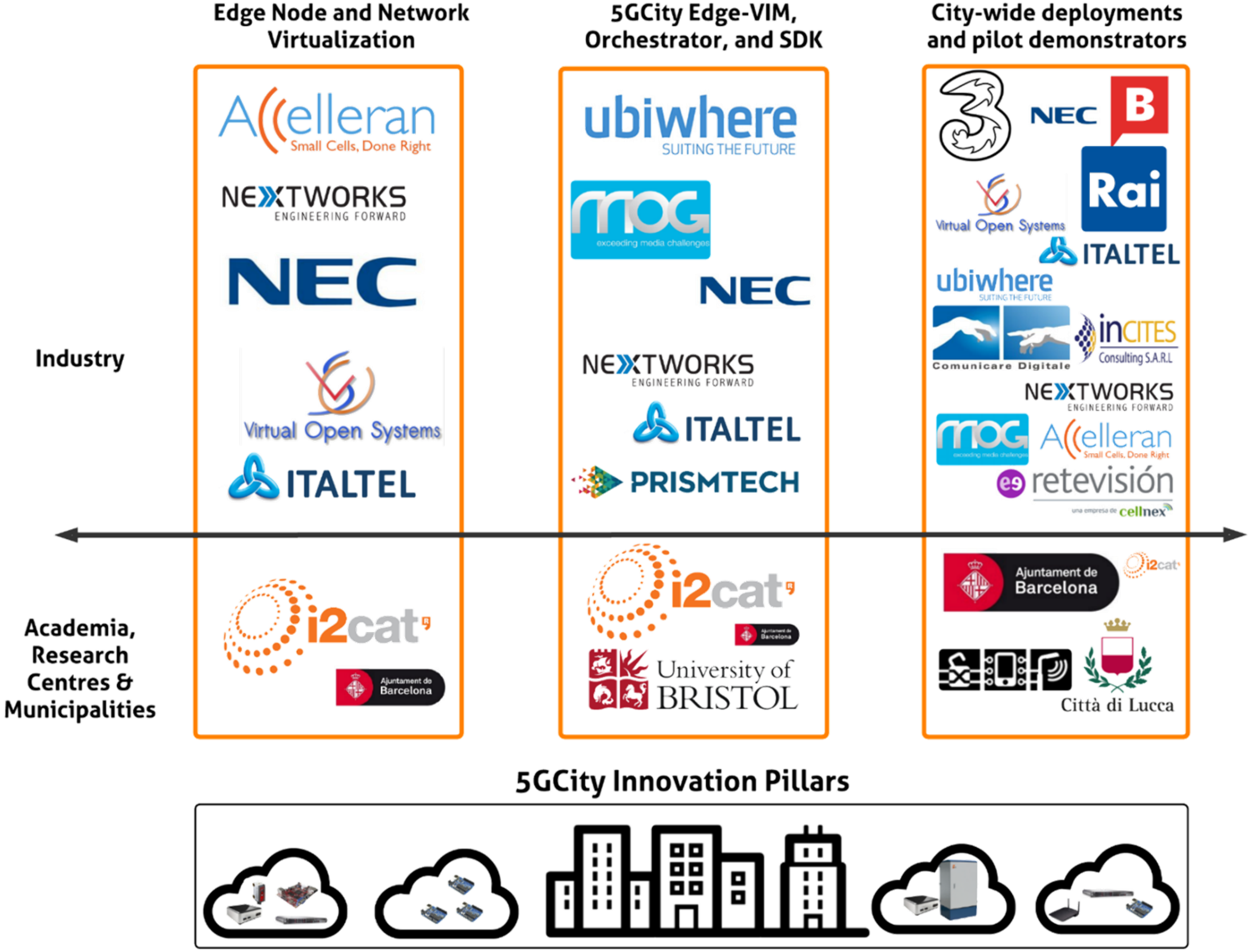
Source: 5G PPP

5GCity project at a glance

Web:
www.5gcity.eu

- 5G PPP Phase 2 Innovation Action
- **i2CAT** (PC) & **NXW** (TM)
- 34 months Duration (2017 – 2020)
- 18 partners form 7 EU countries
- Twitter: @5GCity

5GCity is a large collaboration of industries



Edge Node and Network Virtualization

5GCity Edge-VIM, Orchestrator, and SDK

City-wide deployments and pilot demonstrators

Industry

Academia, Research Centres & Municipalities

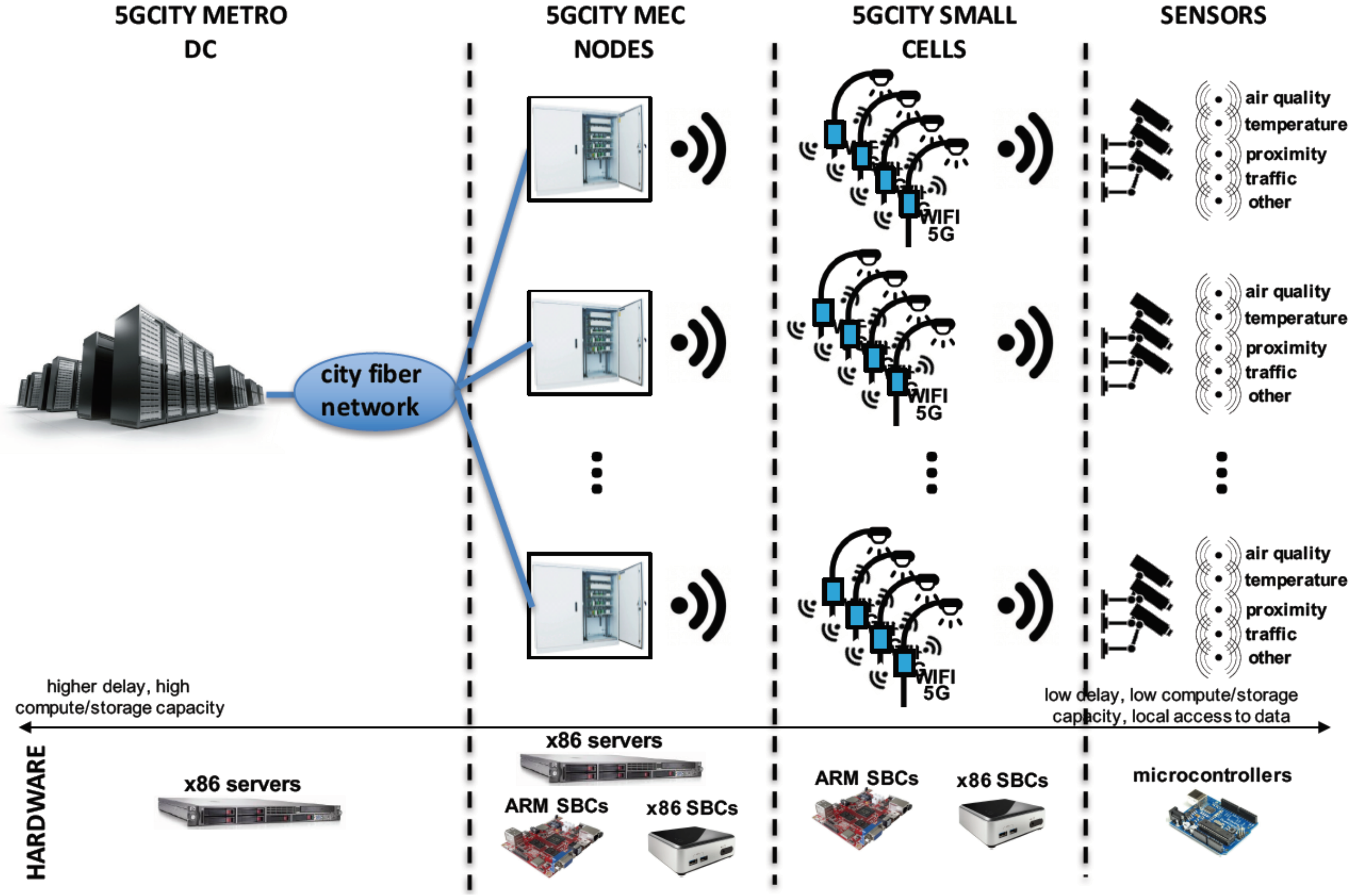
5GCity Innovation Pillars

Accelleran Small Cells, Done Right
NEXTWORKS ENGINEERING FORWARD
NEC
Virtual Open Systems
ITALTEL
i2cat
Ajuntament de Barcelona
ubiwhere SUITING THE FUTURE
MOG exceeding media challenges
NEC
NEXTWORKS ENGINEERING FORWARD
ITALTEL
PRISMTECH
i2cat
Ajuntament de Barcelona
University of BRISTOL
3
NEC
B
Rai
Virtual Open Systems
ITALTEL
ubiwhere SUITING THE FUTURE
Comunicare Digitale
inCITES Consulting S.A.R.L.
NEXTWORKS ENGINEERING FORWARD
MOG
Accelleran Small Cells, Done Right
retevisión una empresa de cellnex
Ajuntament de Barcelona
i2cat
Città di Lucca

What 5GCity is doing

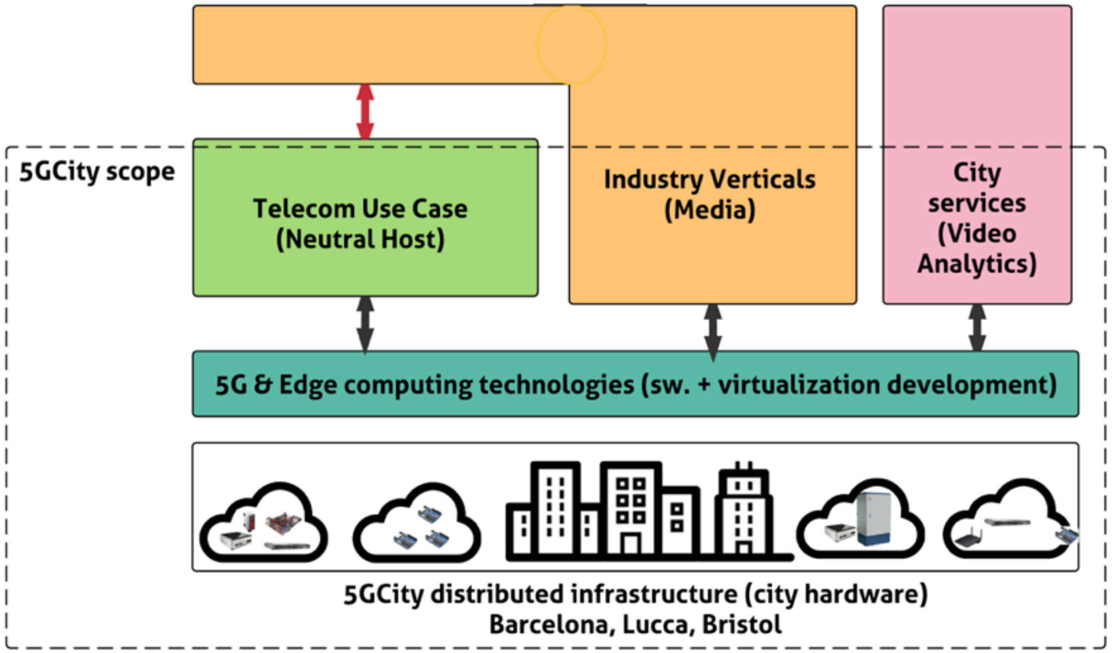
Our team is designing, developing, deploying and demonstrating a distributed cloud and radio platform for municipalities acting as 5G neutral hosts

Live pilot trials in 3 cities: **Barcelona (ES) – Bristol (UK) – Lucca (IT)**



6 key use cases to evaluate benefits in 5GCities

3 Groups: Telco, Media, Security

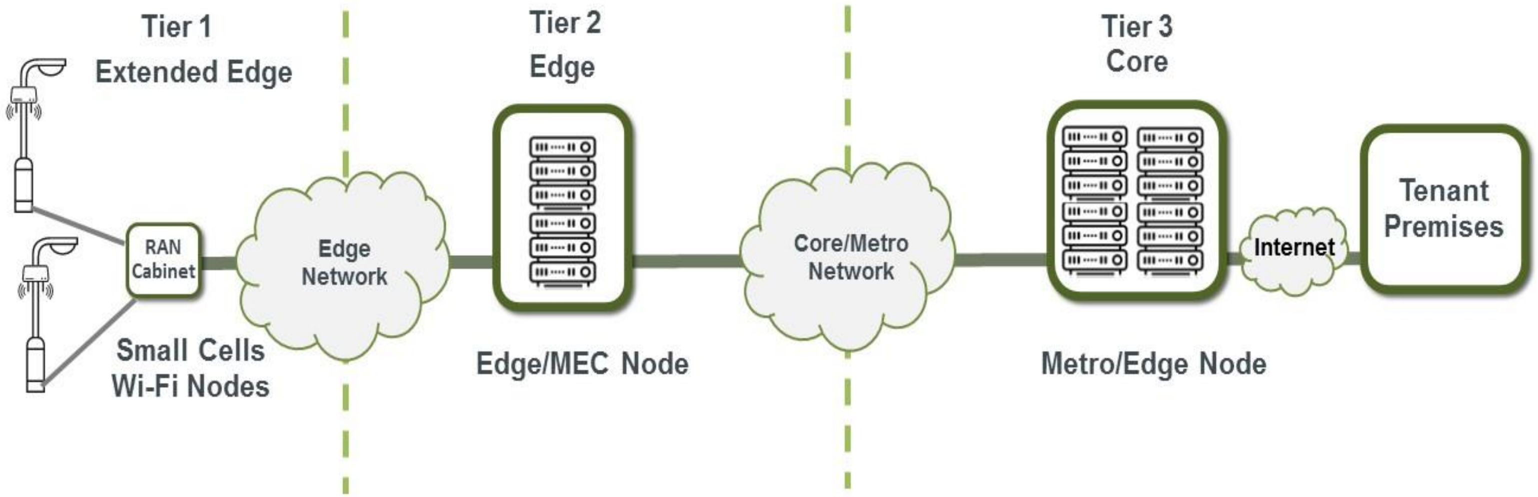

5GCity scope
Telecom Use Case (Neutral Host)
Industry Verticals (Media)
City services (Video Analytics)
5G & Edge computing technologies (sw. + virtualization development)
5GCity distributed infrastructure (city hardware)
Barcelona, Lucca, Bristol

UC ID	Use Case	City		
		BCN	BRS	Lucca
UC1	Unauthorized Waste Dumping Prevention			✓
UC2	Neutral Host	✓	✓	✓
UC3	Video Acquisition and Production + Community media engagement in live events	✓	✓	
UC4	UHD Video Distribution Immersive Services		✓	✓
UC5	Mobile Backpack Unit for Real-time Transmission	✓		
UC6	Cooperative, Connected and Automated Mobility (CCAM)	✓		

Great focus on Media

Core Infrastructure and Platform concepts in 5GCity

- **Distributed, 3-tiers architecture**



- **MEC Node Virtualization Platform and Guest Optimizations**
  - **Unikernels and containers for reduced boot times** and **lightweight images** (i.e., in the few MBs)

- **Innovative Network virtualization**
  - **Wireless virtualization for 802.11 radio and for LTE trusted ARM computing zones**

- **Scalable edge management & orchestration and SDK**

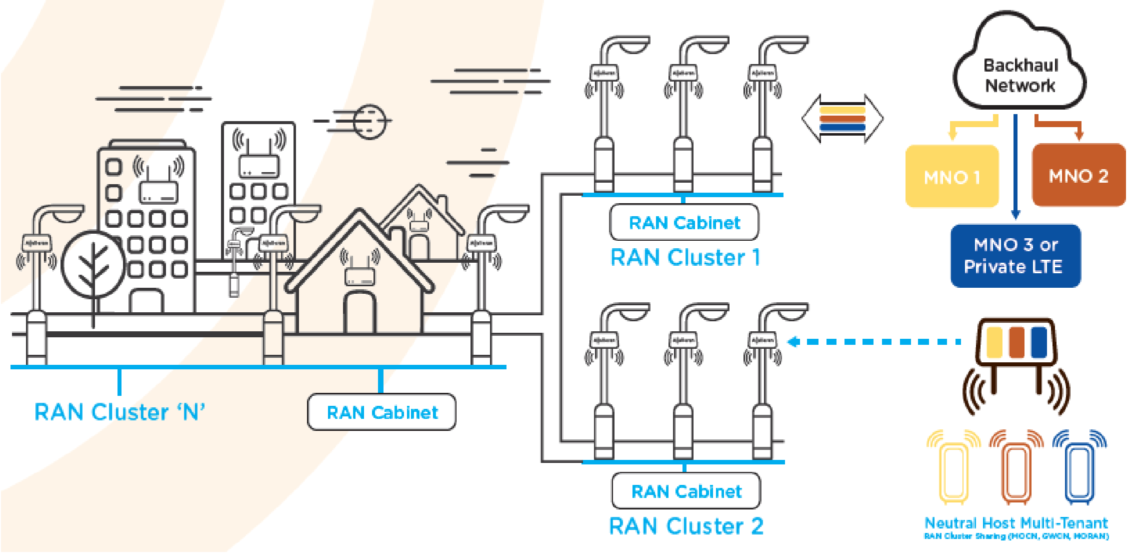
5G Neutral Host model

Network slicing mechanisms developed in 5GCity

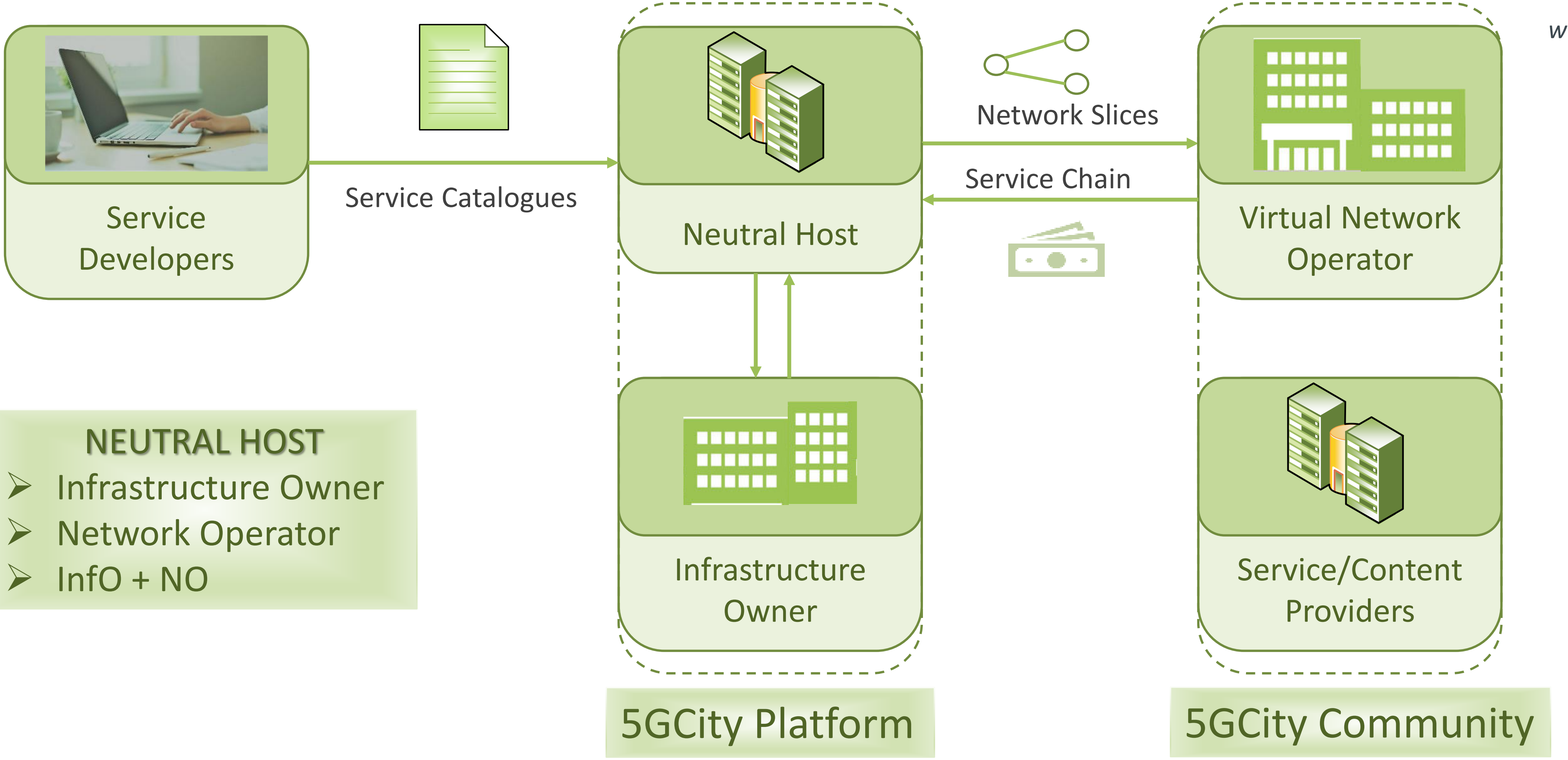
- The network infrastructure hosts any entity that, in turn, uses this slice to provide services to end-users
- The "Neutral Host" (infrastructure owner) is able to operate a partition of its resources and to arrange them in a set of homogeneous tenants (or slices)

Architectural approach for neutral 5G operator in 5GCity

- *MOCN – Multiple Operator Core Network*
  - The cabinet hosts centralised Small Cell components dedicated to the control of a cluster of physical Small Cells deployed in lampposts and other urban furniture
  - Spectrum is shared among different Mobile Operators



Our 5G Neutral Hosting model in practice



NEUTRAL HOST

- Infrastructure Owner
- Network Operator
- InfO + NO

Technology enablers: NFV and SDN

Gino Carrozzo

from **Nextworks**

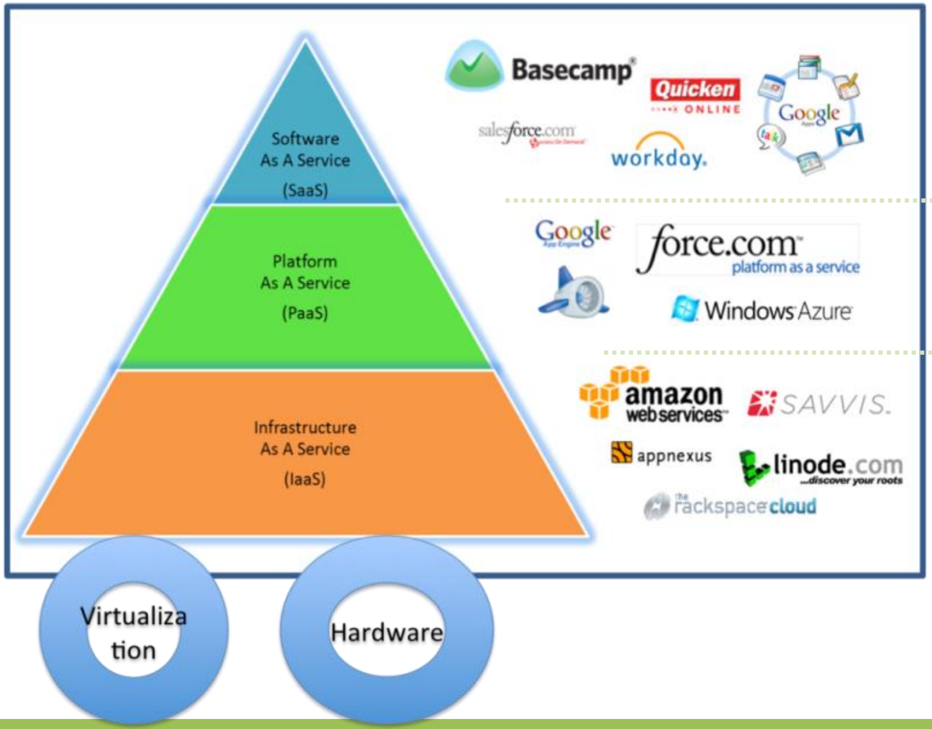
5GCity

Technical Manager

Drivers for innovation in ICT and networks

Telco market is rapidly moving towards an Everything-as-a-Service model, solidly sustained by

- **Virtualization** & **softwarization** of traditionally in-the-box network functions (Network Functions Virtualization – NFV)
- **Deep network programmability** tools/technologies
- Advanced **Infrastructure as a Service** platforms and orchestration tools



SaaS:	Email, calendars, DMS, CRM, etc.
PaaS:	OS, DBMS, LAMP/WAMP [virtual] servers, etc.
IaaS:	Enterprise infrastructure, Cloud hosting, Virtual Data Centers, etc.

5G is driving Telcos’ transformation

What 5G will bring

Speed
1000 TIMES

Users
20 BILLION
HUMAN-ORIENTED TERMINAL

Devices
1 TRILLION

Power consumption
90%

Latency
<5MS LATENCY

Availability
99.999%

Source: 5G PPP

A radical transformation of vertical sectors

Digitalization of vertical markets

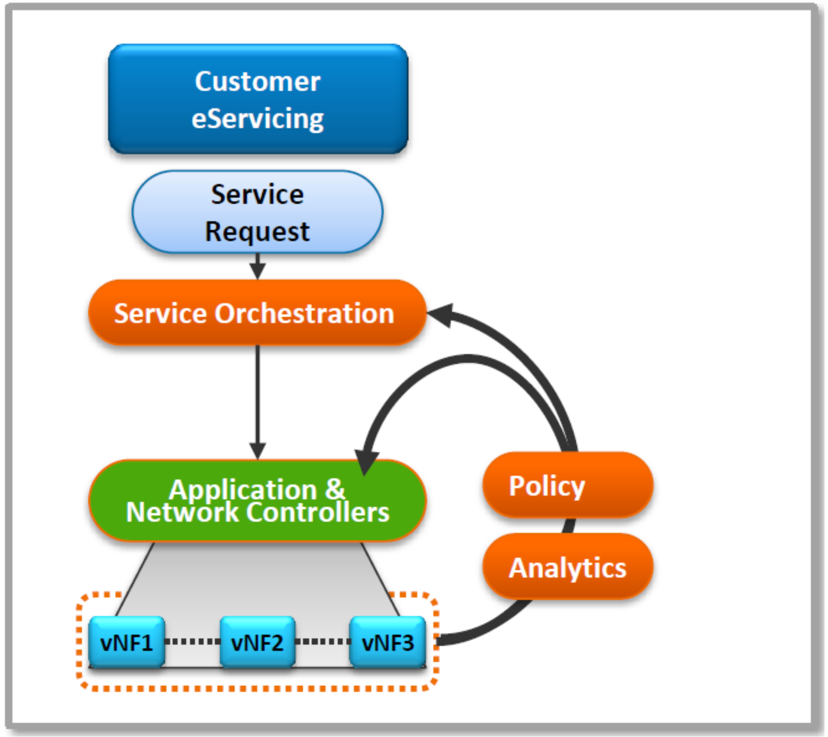
- Automotive, transportation, manufacturing, banking, city management, energy, utilities, real estate, retail….

Sustainable and scalable technology

- Growth in number of terminal devices and traffic
- Heterogeneous network layouts

But more importantly, enriched and automated service model

- Auto-provisioned service through central end-to-end orchestration
- Network analytics also exposed to Verticals



Customer eServicing

Service Request

Service Orchestration

Application & Network Controllers

Policy

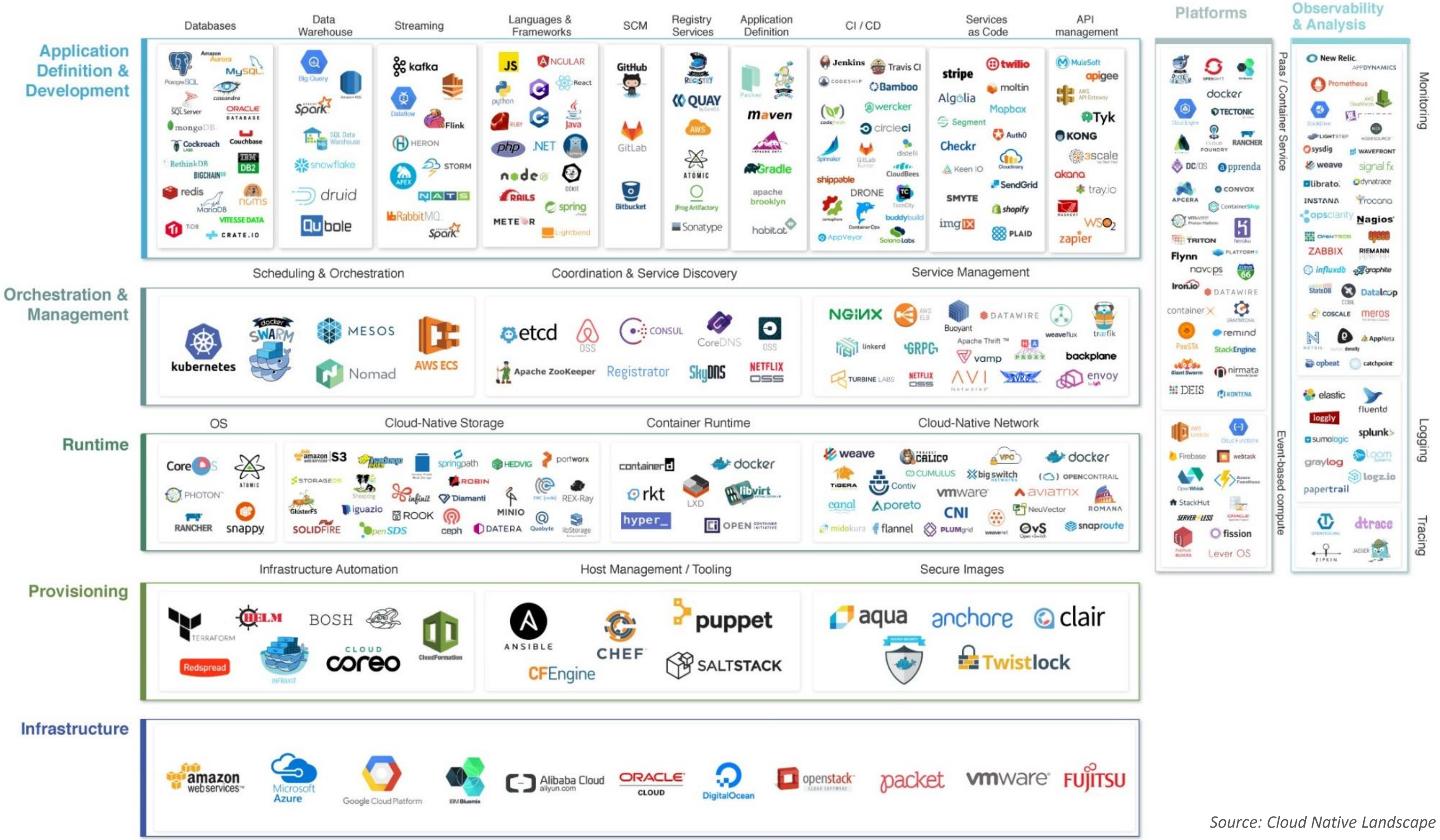
Analytics

vNF1

vNF2

vNF3

There is a complex landscape of technologies and tools for this

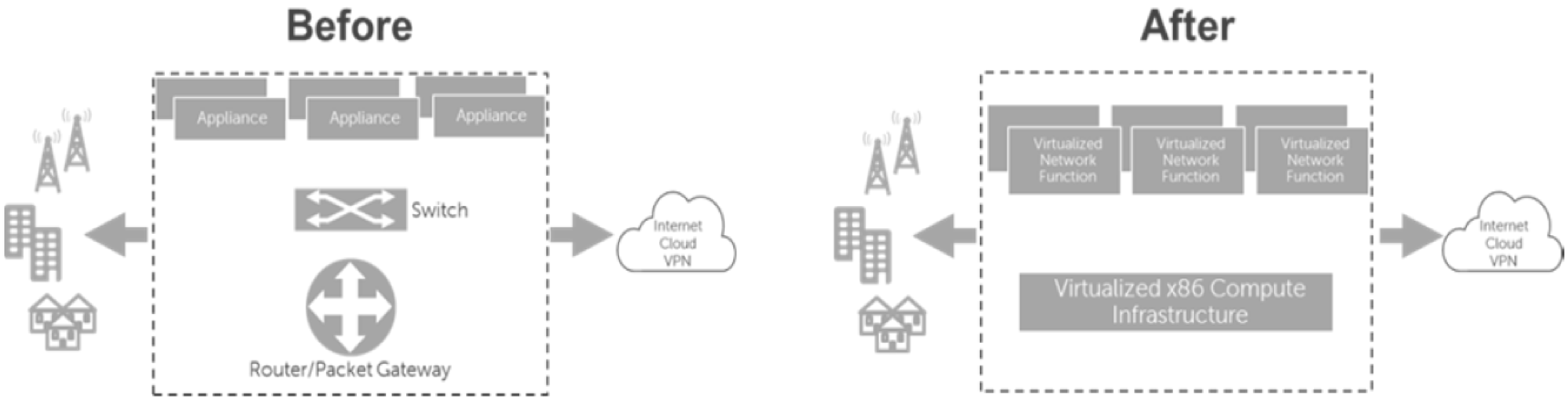


Source: Cloud Native Landscape

NFV principle

Network functions virtualization (NFV) decouples network services from the hardware that delivers them

- functions, such as network address translation (NAT), firewalling, intrusion detection, domain name service (DNS) and caching, can be delivered in software and deployed on general purpose appliances
- Result is a lot more flexibility in design, deployment and management of network services



NFV rationale

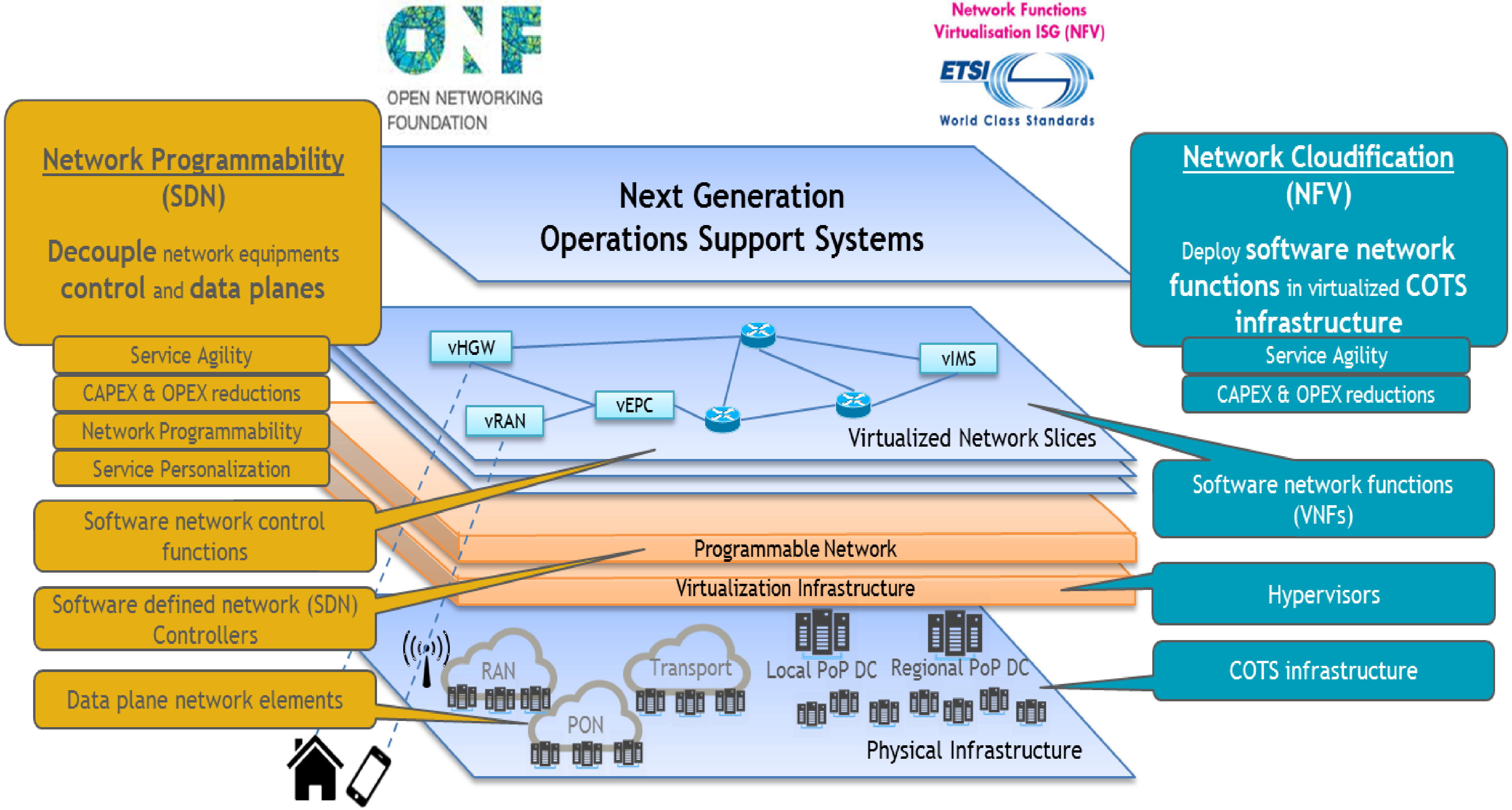
Cloud computing and virtualization have had profound impact on

- Data center & network design
- Service models (X-as-a-Service)
- Performance & efficiency

Supporting a common sense/desire for

- Agile, on-demand provisioning of new services and service architectures
- Open and standards-based hardware and software components
- Common shared infrastructure for workloads with varied SLA
- Common Management & Operations and service lifecycle across different service silos
- Implicit user and mobility-aware, end-to-end security and policy model
- Autonomic elastic workload scaling
- Modern, open APIs for every logical component
- Powerful developer tools and platform services to accelerate innovation and new service delivery

Paradigm Shift #1: towards the agile Software Network



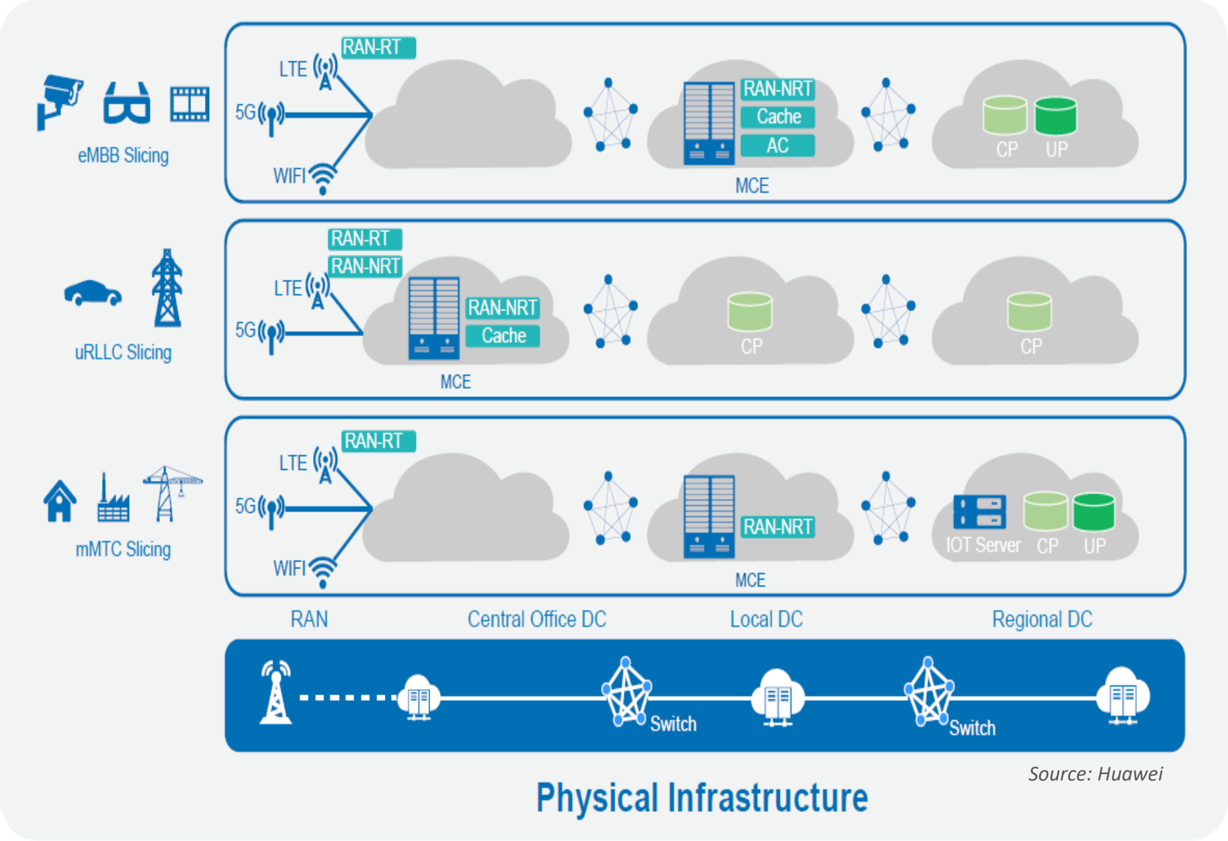
OPEN NETWORKING FOUNDATION
Network Functions Virtualisation ISG (NFV)
ETSI World Class Standards
Network Programmability (SDN)
Decouple network equipments control and data planes
Service Agility
CAPEX & OPEX reductions
Network Programmability
Service Personalization
Software network control functions
Software defined network (SDN) Controllers
Data plane network elements
Next Generation Operations Support Systems
vHGW
vIMS
vRAN
vEPC
Virtualized Network Slices
Programmable Network
Virtualization Infrastructure
RAN
Transport
Local PoP DC
Regional PoP DC
PON
Physical Infrastructure
Network Cloudification (NFV)
Deploy software network functions in virtualized COTS infrastructure
Service Agility
CAPEX & OPEX reductions
Software network functions (VNFs)
Hypervisors
COTS infrastructure

Paradigm Shift #2: Network Slicing through SDN/NFV

- Slicing is a way to provide fully or partly, logically and/or physically isolated sub-networks on top of a shared infrastructure [see NGNM]
  - Set of network (and non-network) functions and resources (both phy and virt)
  - A complete instantiated logical network fulfilling the characteristics of the Service Instance(s)
  - Autonomous control loop (orchestrators and controller chain) in the slice

Current 3GPP R15 networks do not support a notion of slicing, but R16 will

Virtualization, SDN/NFV and MEC are key enablers of Network Slicing

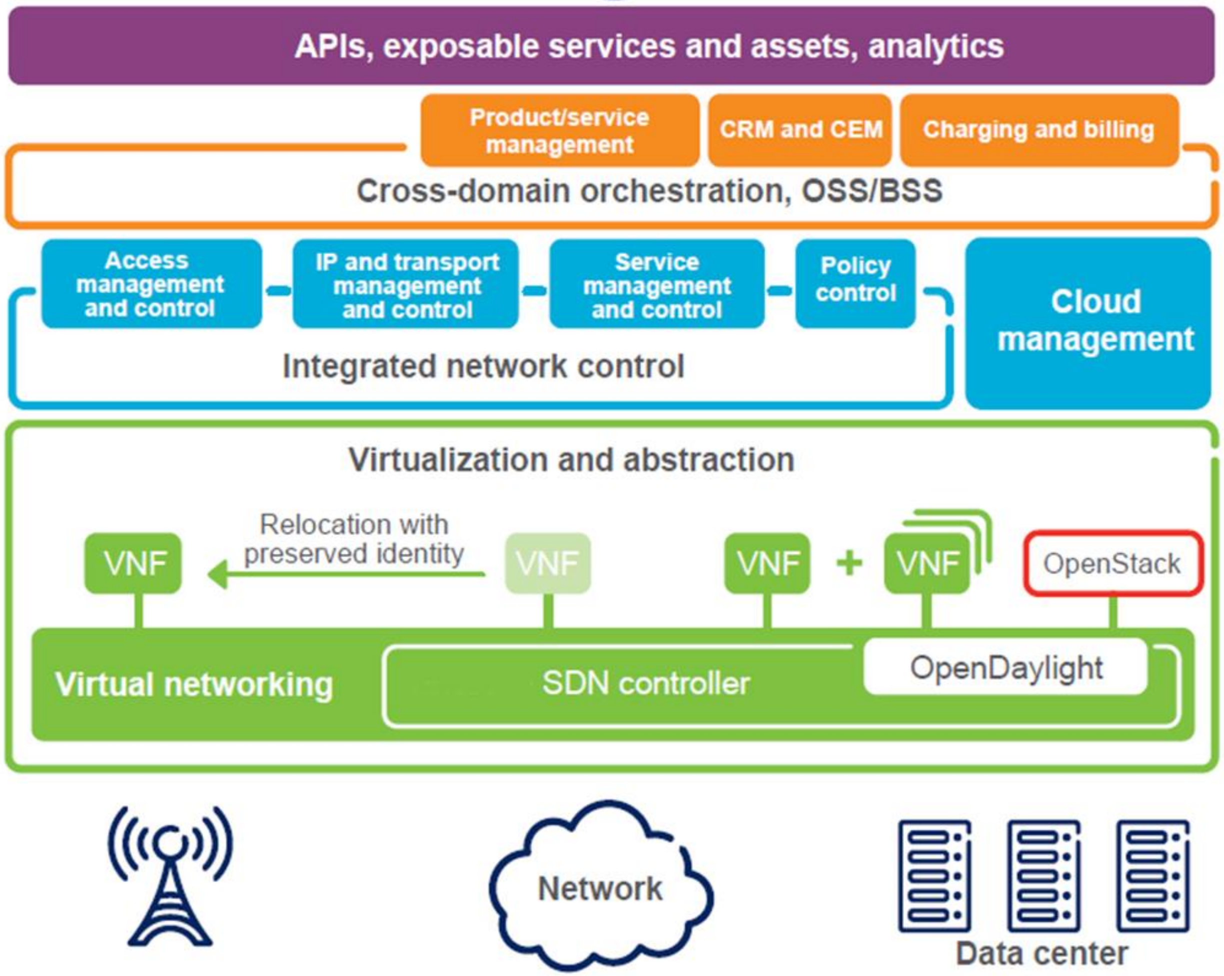


Source: Huawei

eMBB – Enhanced Mobile Broadband
uRLLC – Ultra-reliable and Low-latency Communications
mMTC - Massive Machine Type Communications

NFV, SDN and cloud technology

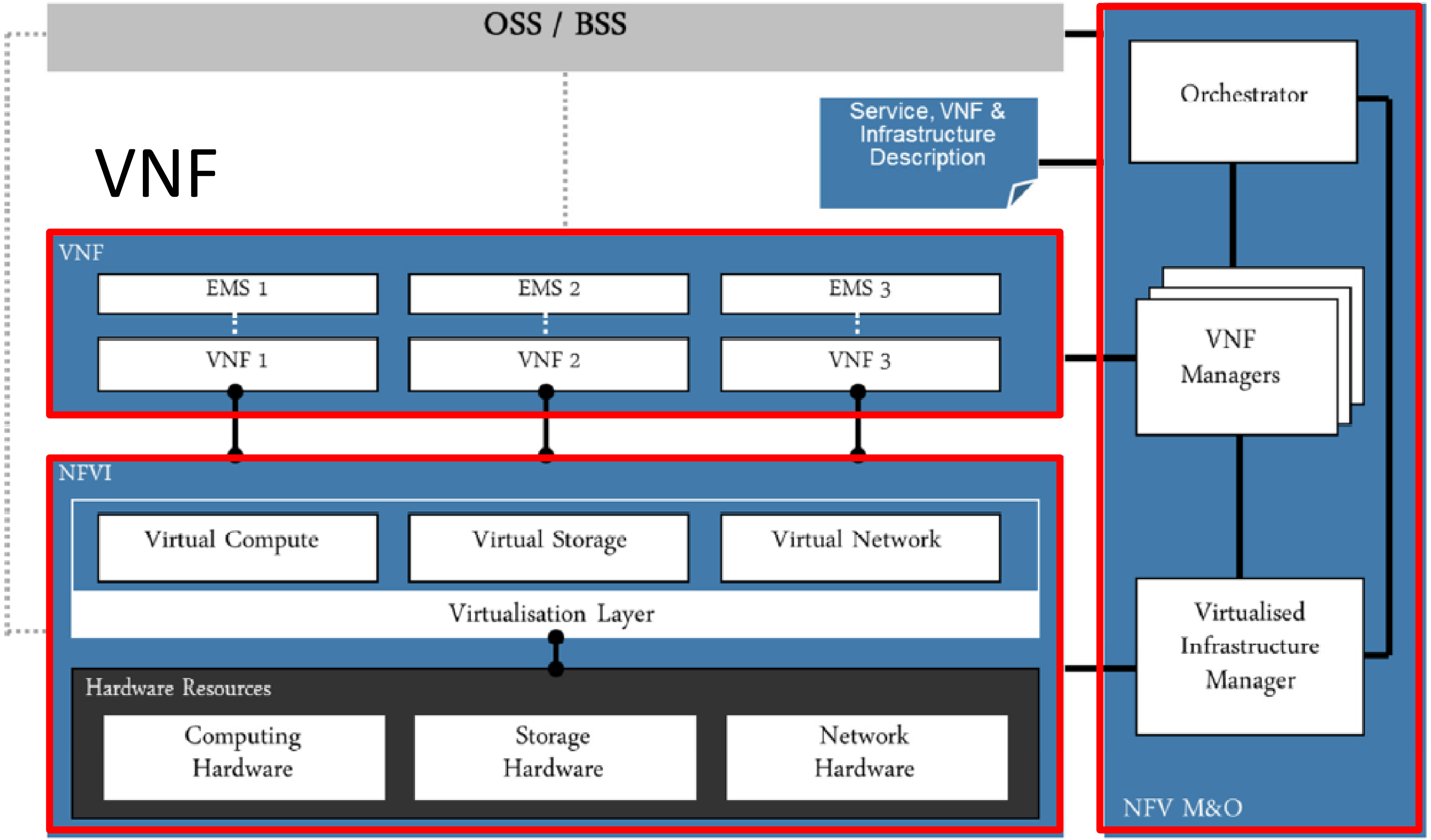
Towards combined Network-enabled Cloud, NFV & SDN &MEC


APIs, exposable services and assets, analytics
Product/service management
CRM and CEM
Charging and billing
Cross-domain orchestration, OSS/BSS
Access management and control
IP and transport management and control
Service management and control
Policy control
Cloud management
Integrated network control
Virtualization and abstraction
VNF
Relocation with preserved identity
VNF
VNF
+
VNF
OpenStack
Virtual networking
SDN controller
OpenDaylight
Network
Data center

Source: Ericsson

ETSI NFV Architectural framework

5GCity webinar



VNF

NFVI

NFV MANO

Network Functions Virtualisation ISG (NFV)

ETSI World Class Standards

ETSI NFV Architectural framework [2]

webinar

Network Functions Virtualisation Infrastructure (NFVI)

- Totality of all hardware and software components which build up the environment in which VNFs are deployed, managed and executed
- COTS hardware, accelerator components where necessary, and software layer to virtualise and abstracts the underlying hardware

Virtualised Network Function (VNF)

- Software implementation of a network function which is capable of running over the NFVI
- Can be accompanied by an Element Management System (EMS)
- Corresponds to today’s network nodes, expected to be delivered as pure software free from hardware dependency

NFV Management and Orchestration (MANO)

- Covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualisation, and the lifecycle management of VNFs
- Interacts with the (NFV external) OSS/BSS

Network Functions Virtualisation ISG (NFV)

ETSI World Class Standards

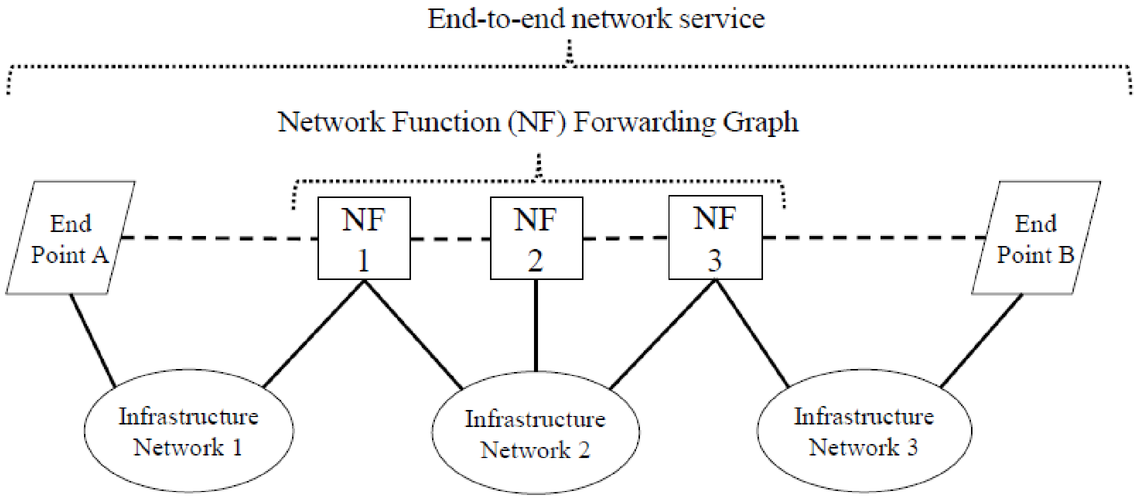
Key terminology

VNF Forwarding Graph (VNF-FG)

- To describe network connectivity between VNFs, e.g. a chain of VNFs on the path to a web server tier (e.g. firewall, NAT, load balancer)

Network service (NS)

- A forwarding graph of Network Functions (NFs) interconnected by supporting network infrastructure
- NF can be implemented in a single operator network or interwork between different operator networks
- End points and NFs of a NS are represented as nodes and correspond to devices, applications, and/or physical server applications



source: ETSI

Network Functions Virtualisation ISG (NFV)

Packaging a VNF

webinar

A VNF is typically built of many software components and auxiliary files

VNF vendors provide the required VNF software components and auxiliary files in a “VNF Package”

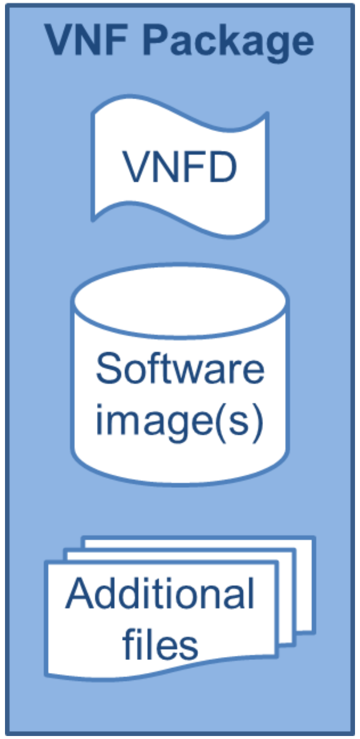
The VNF Package contains:

- VNF descriptor (VNFD) that defines metadata for package onboarding and VNF management
- the software images needed to run the VNF
- (optional) additional files to manage the VNF (e.g. scripts, vendor-specific files, etc.)

The VNF Package is digitally signed and delivered by the VNF provider as a whole

The VNF Package is immutable (protected from modification)

The VNF Package can be accessed by OSS/BSS, NFVO and VNFM

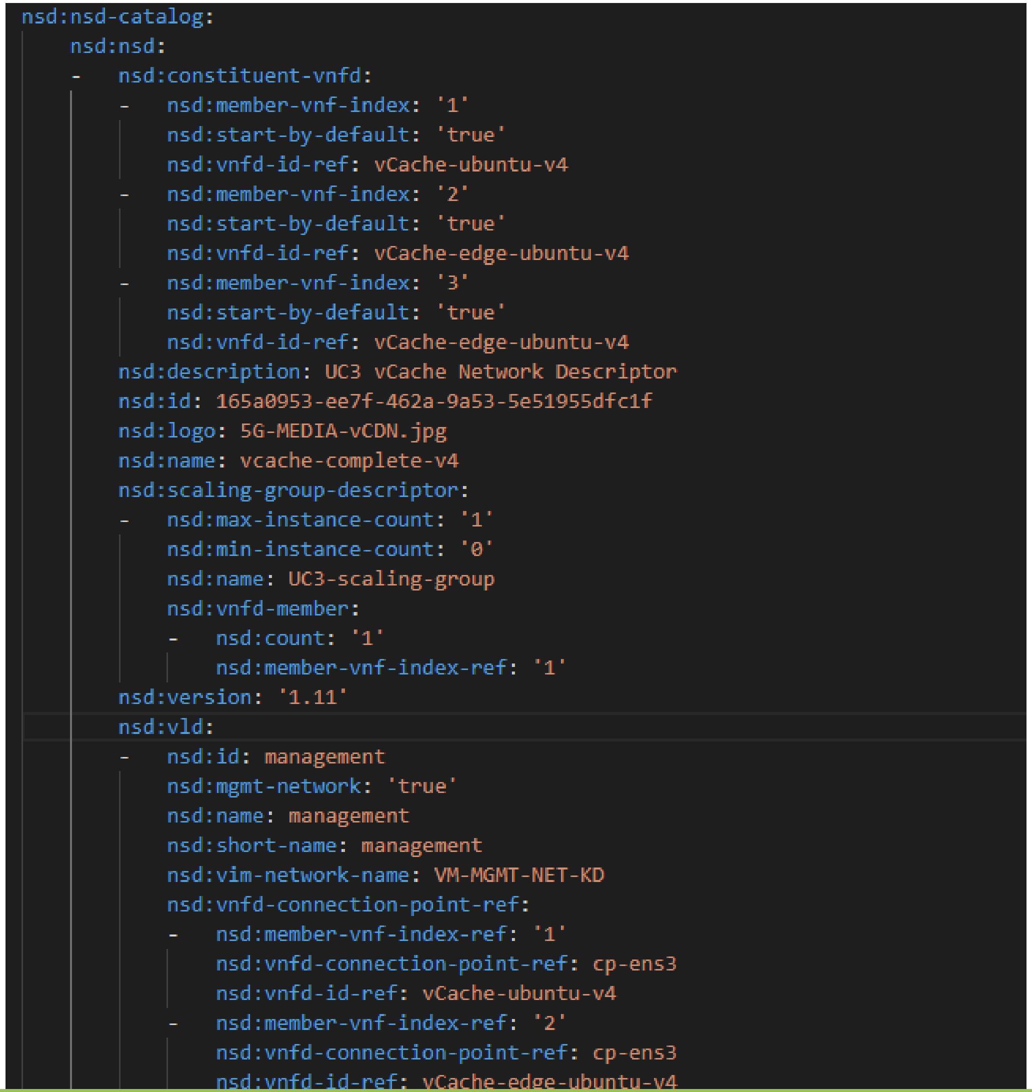


source: ETSI

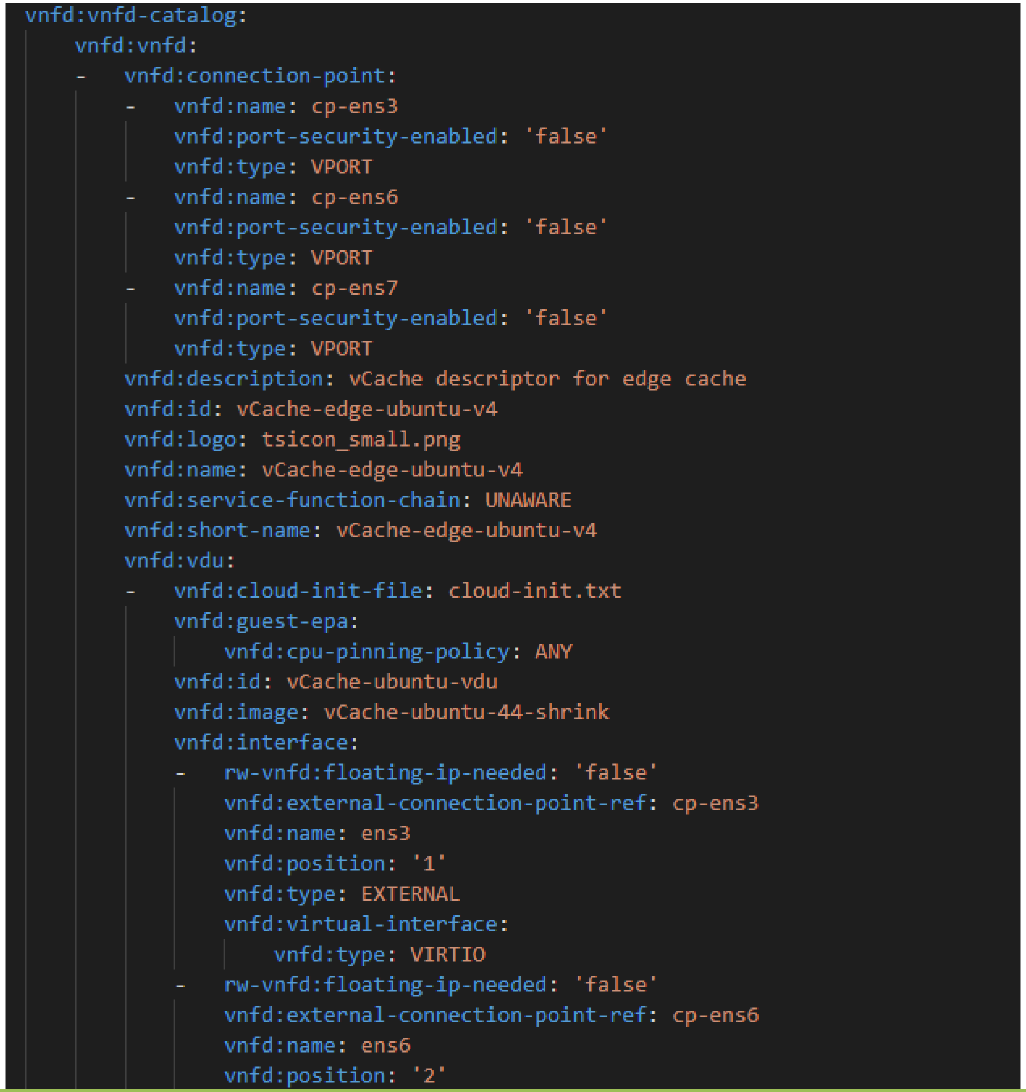
Network Functions Virtualisation ISG (NFV)

ETSI World Class Standards

Examples of NS and VFN descriptors

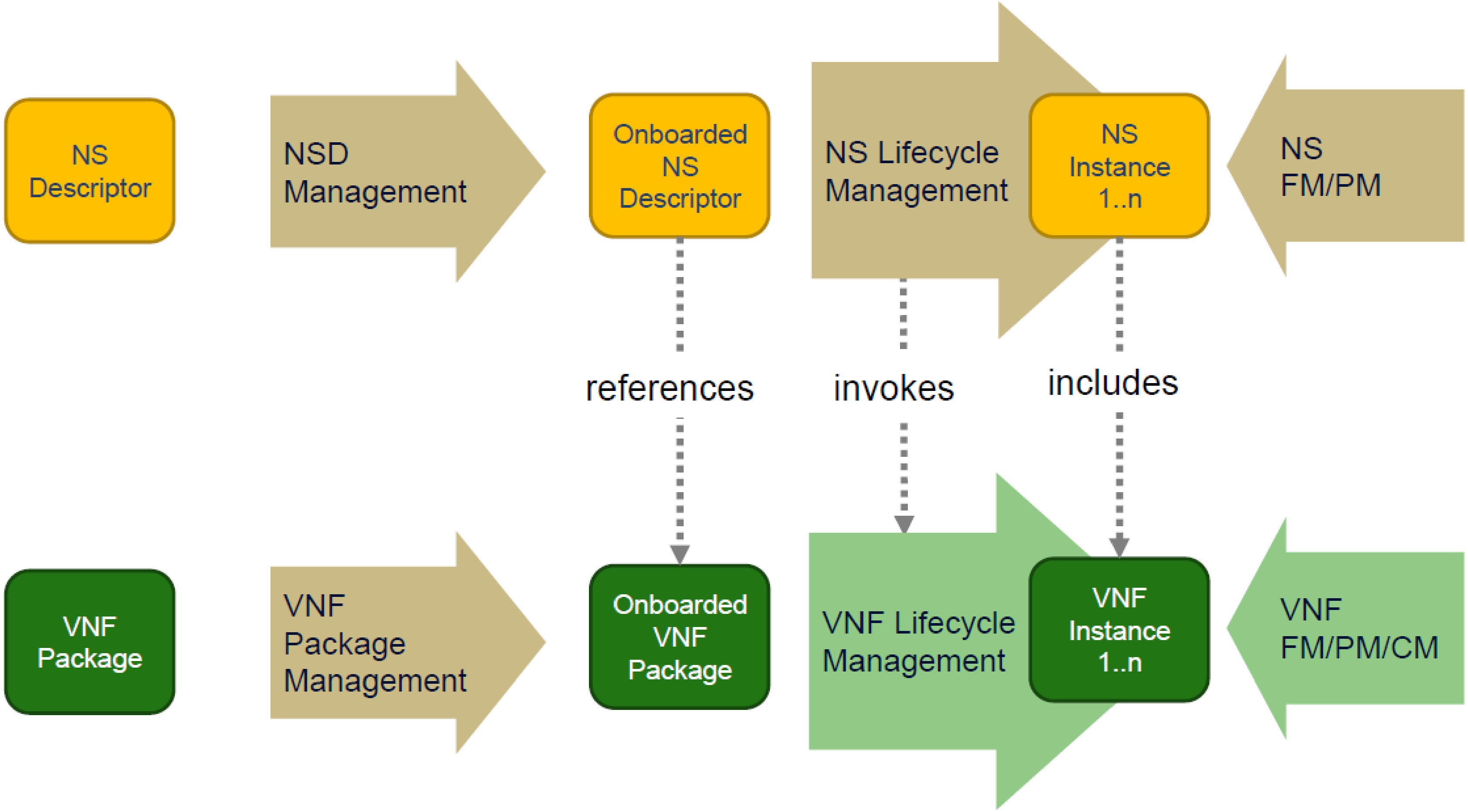


```
nsd:nsd-catalog:
  nsd:nsd:
  - nsd:constituent-vnfd:
    - nsd:member-vnf-index: '1'
      nsd:start-by-default: 'true'
      nsd:vnfd-id-ref: vCache-ubuntu-v4
    - nsd:member-vnf-index: '2'
      nsd:start-by-default: 'true'
      nsd:vnfd-id-ref: vCache-edge-ubuntu-v4
    - nsd:member-vnf-index: '3'
      nsd:start-by-default: 'true'
      nsd:vnfd-id-ref: vCache-edge-ubuntu-v4
    nsd:description: UC3 vCache Network Descriptor
    nsd:id: 165a0953-ee7f-462a-9a53-5e51955dfc1f
    nsd:logo: 5G-MEDIA-vCDN.jpg
    nsd:name: vcache-complete-v4
    nsd:scaling-group-descriptor:
    - nsd:max-instance-count: '1'
      nsd:min-instance-count: '0'
      nsd:name: UC3-scaling-group
      nsd:vnfd-member:
      - nsd:count: '1'
        nsd:member-vnf-index-ref: '1'
    nsd:version: '1.11'
    nsd:vld:
    - nsd:id: management
      nsd:mgmt-network: 'true'
      nsd:name: management
      nsd:short-name: management
      nsd:vim-network-name: VM-MGMT-NET-KD
      nsd:vnfd-connection-point-ref:
      - nsd:member-vnf-index-ref: '1'
        nsd:vnfd-connection-point-ref: cp-ens3
        nsd:vnfd-id-ref: vCache-ubuntu-v4
      - nsd:member-vnf-index-ref: '2'
        nsd:vnfd-connection-point-ref: cp-ens3
        nsd:vnfd-id-ref: vCache-edge-ubuntu-v4
```



```
vnfd:vnfd-catalog:
  vnfd:vnfd:
  - vnfd:connection-point:
    - vnfd:name: cp-ens3
      vnfd:port-security-enabled: 'false'
      vnfd:type: VPORT
    - vnfd:name: cp-ens6
      vnfd:port-security-enabled: 'false'
      vnfd:type: VPORT
    - vnfd:name: cp-ens7
      vnfd:port-security-enabled: 'false'
      vnfd:type: VPORT
    vnfd:description: vCache descriptor for edge cache
    vnfd:id: vCache-edge-ubuntu-v4
    vnfd:logo: tsicon_small.png
    vnfd:name: vCache-edge-ubuntu-v4
    vnfd:service-function-chain: UNAWARE
    vnfd:short-name: vCache-edge-ubuntu-v4
    vnfd:vdu:
    - vnfd:cloud-init-file: cloud-init.txt
      vnfd:guest-epa:
        vnfd:cpu-pinning-policy: ANY
      vnfd:id: vCache-ubuntu-vdu
      vnfd:image: vCache-ubuntu-44-shrink
      vnfd:interface:
      - rw-vnfd:floating-ip-needed: 'false'
        vnfd:external-connection-point-ref: cp-ens3
        vnfd:name: ens3
        vnfd:position: '1'
        vnfd:type: EXTERNAL
        vnfd:virtual-interface:
          vnfd:type: VIRTIO
      - rw-vnfd:floating-ip-needed: 'false'
        vnfd:external-connection-point-ref: cp-ens6
        vnfd:name: ens6
        vnfd:position: '2'
```

Management and Orchestration of VNF and Network Services



source: ETSI

Network Functions Virtualisation ISG (NFV)

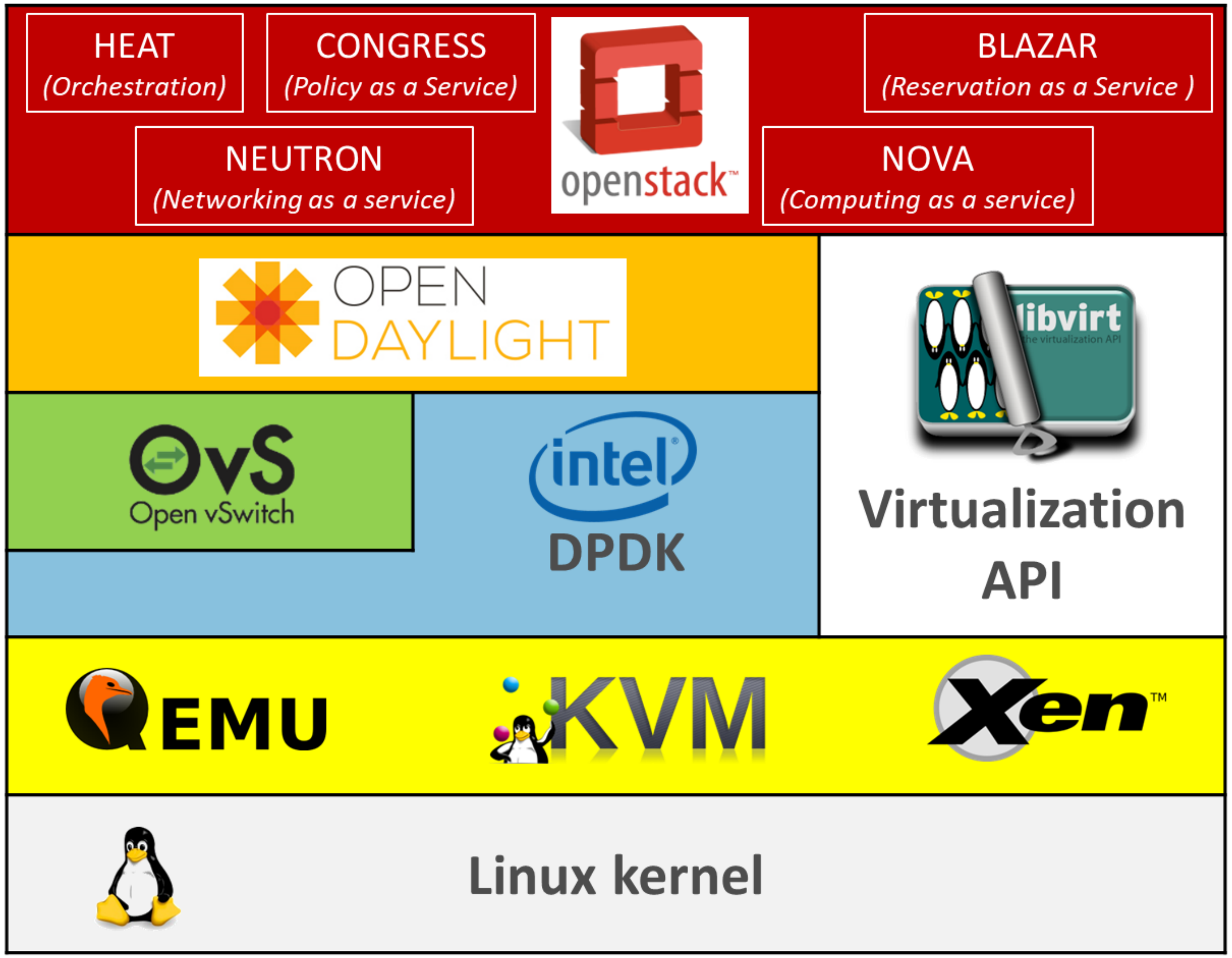
ETSI World Class Standards

Basic bricks for SDN/NFV & MEC

5GCity webinar

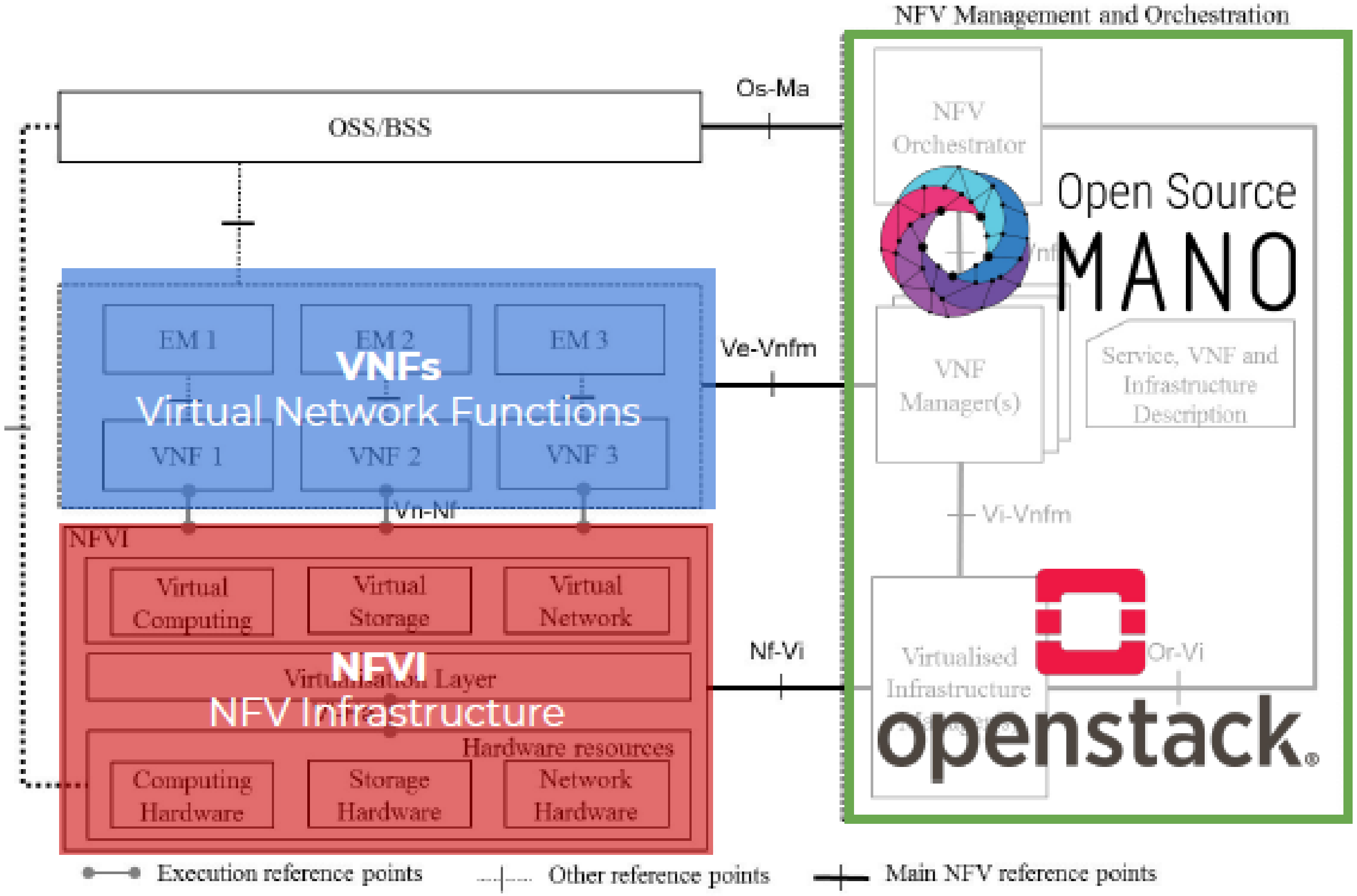
Network Functions Virtualisation ISG (NFV)
ETSI World Class Standards

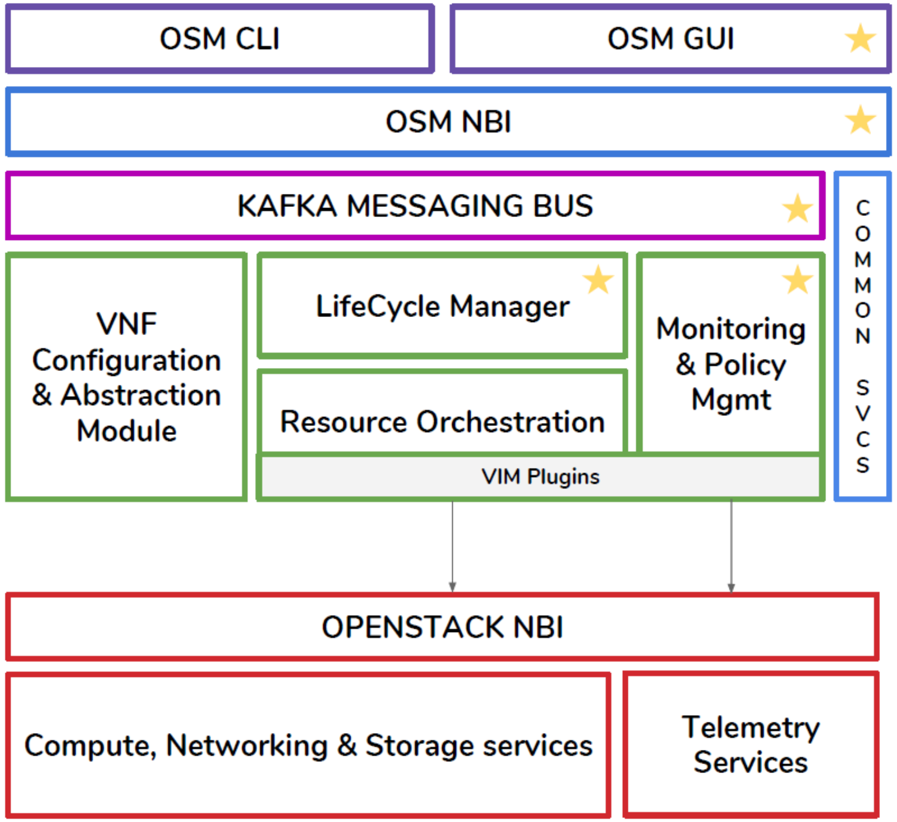
OPNFV



NFV Mangement & Orchestration tools

5GCity webinar





OSM CLI

OSM GUI ★

OSM NBI ★

KAFKA MESSAGING BUS ★

COMMON SVCS

VNF Configuration & Abstraction Module

LifeCycle Manager ★

Resource Orchestration

Monitoring & Policy Mgmt ★

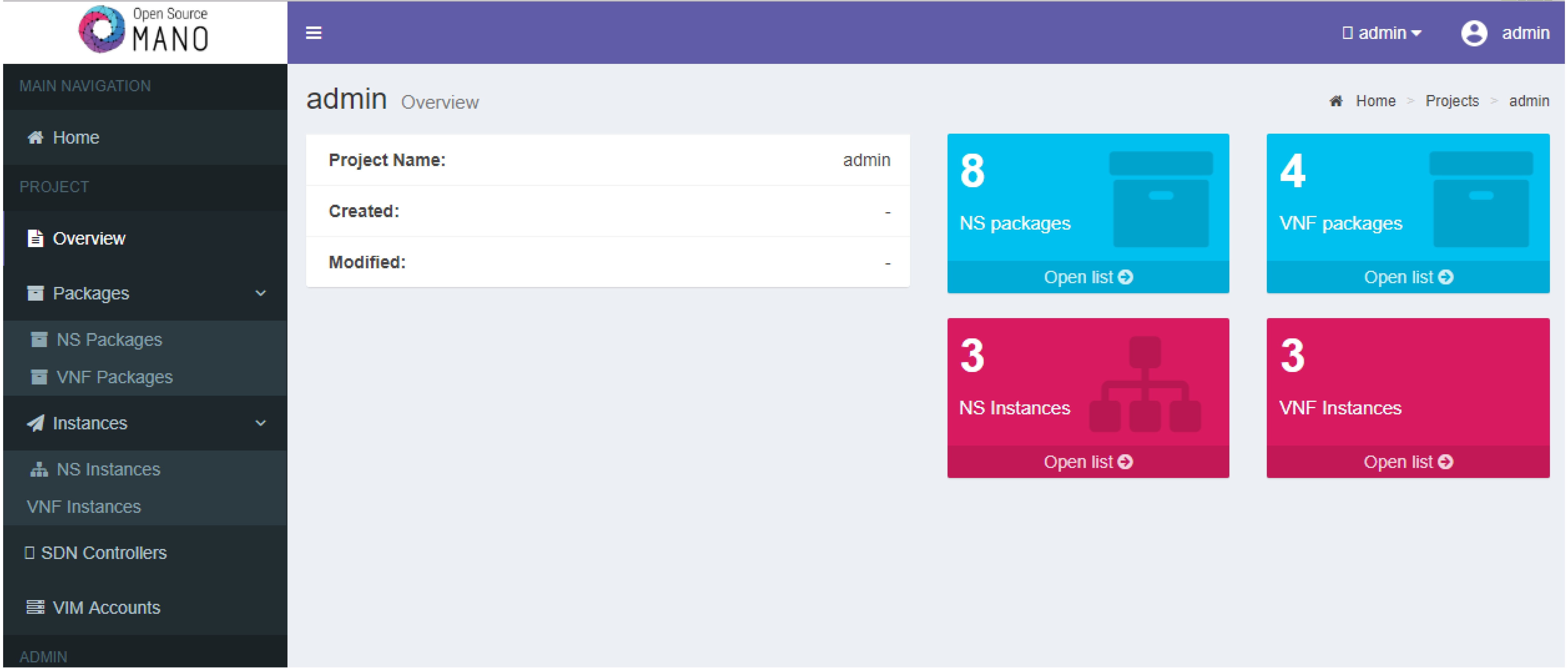
VIM Plugins

OPENSTACK NBI

Compute, Networking & Storage services

Telemetry Services

OSM dashboard



admin Overview

Project Name:	admin
Created:	-
Modified:	-

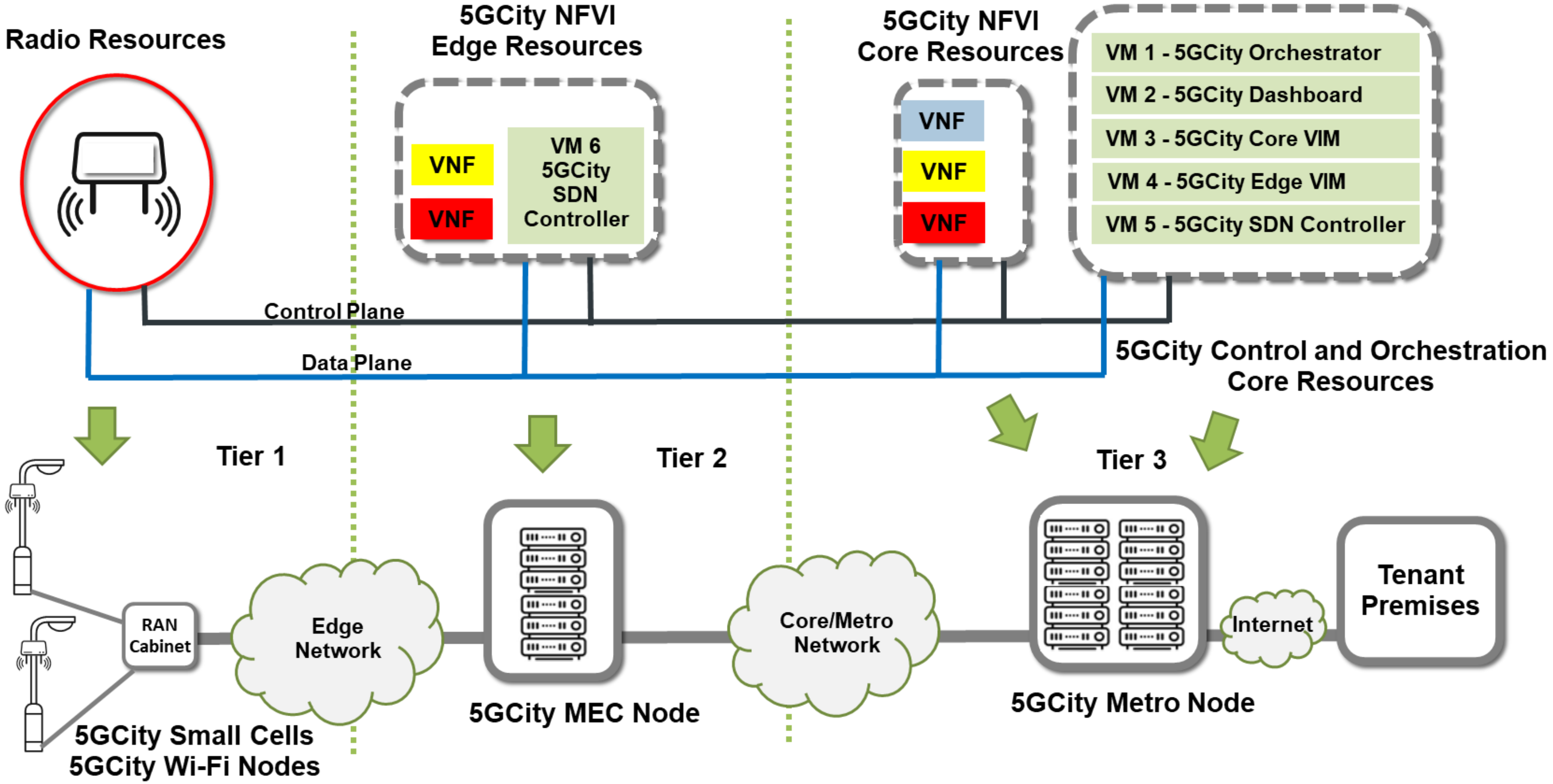
8 NS packages Open list

4 VNF packages Open list

3 NS Instances Open list

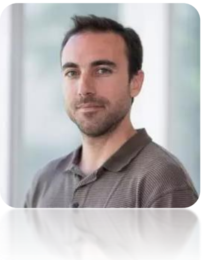
3 VNF Instances Open list

5GCity and NFV



5GCity platform

ARCHITECTURE, COMPONENTS AND DASHBOARD



Apostolos Papageorgiou

from **i2Cat**

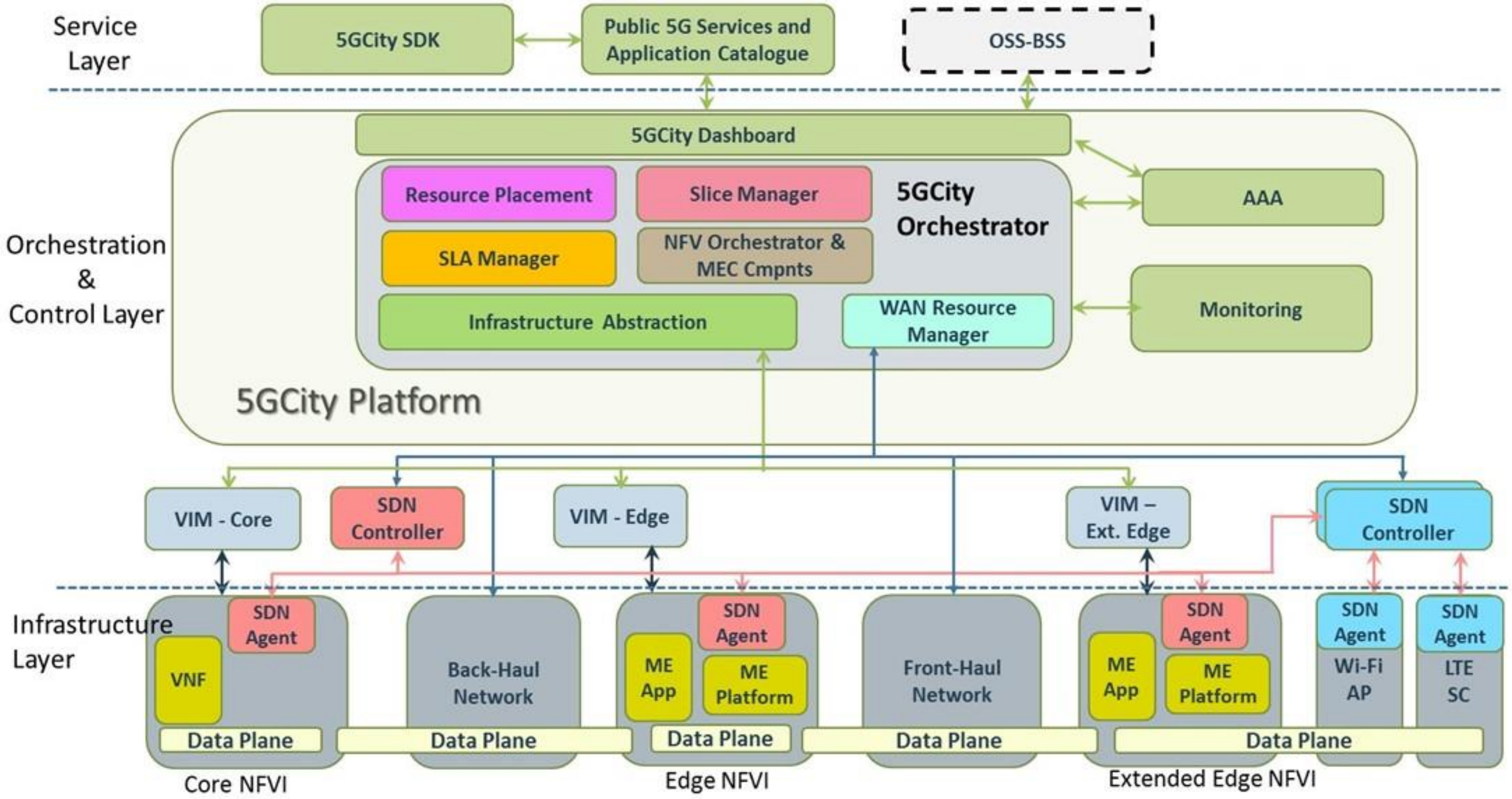
Resp. 5GCity
orchestration platform

5GCity architecture principles

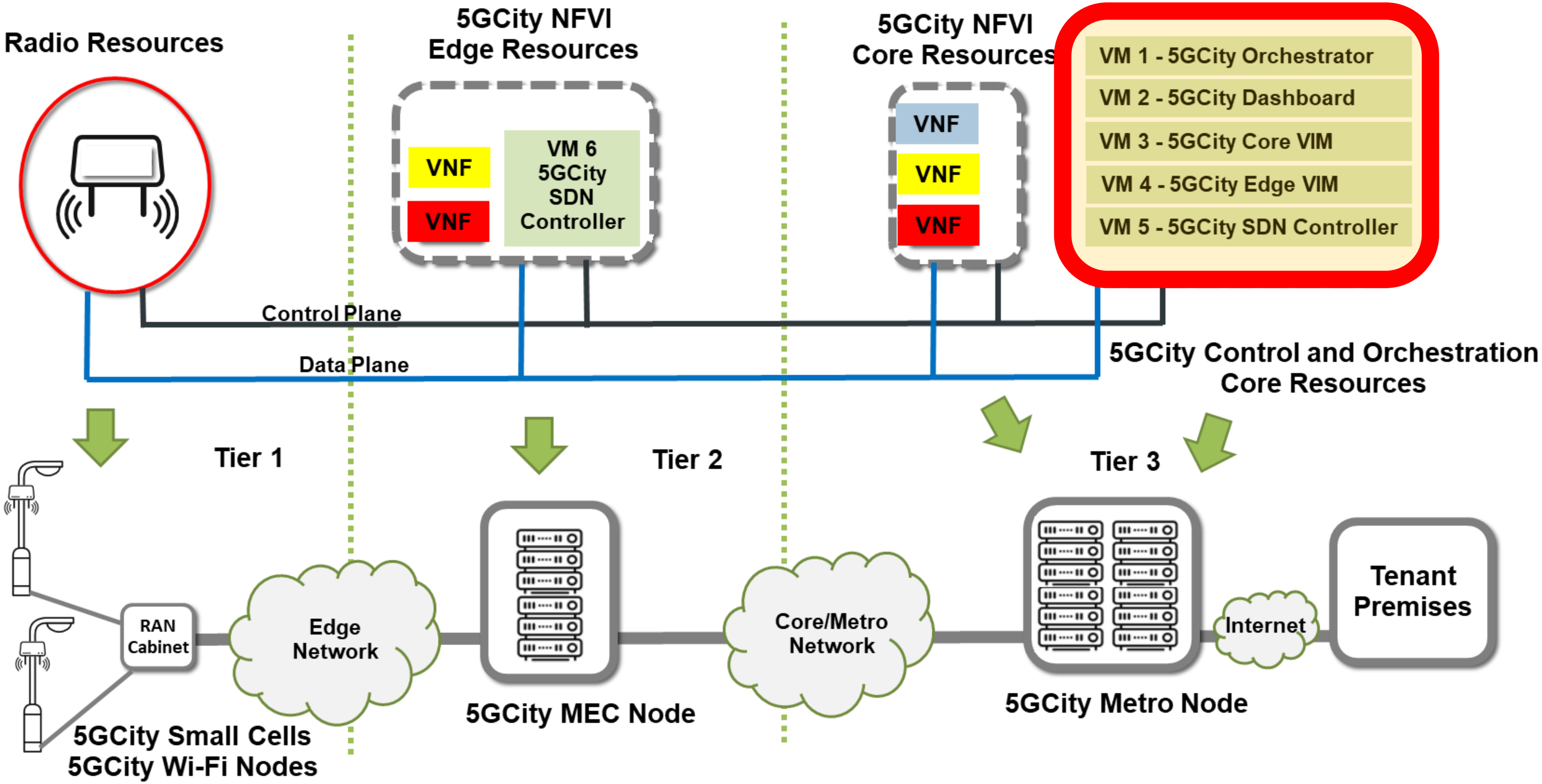
Split vertically across three layers

- **Service/Application Layer**
  - specific set of functions/tools of the proposed 5GCity architecture available for the operators of the infrastructure, their customers, subcontractors and any third party actor
- **Orchestration & Control layer**
  - entry point of network services (Dashboard), core orchestration components (5GCity orchestrator), as well as control between the central orchestration platform and the infrastructure (WAN managers, VIMs, and SDN controllers)
- **Infrastructure layer**
  - The actual radio and computing virtualization infrastructure spanning from far-edge to datacenter

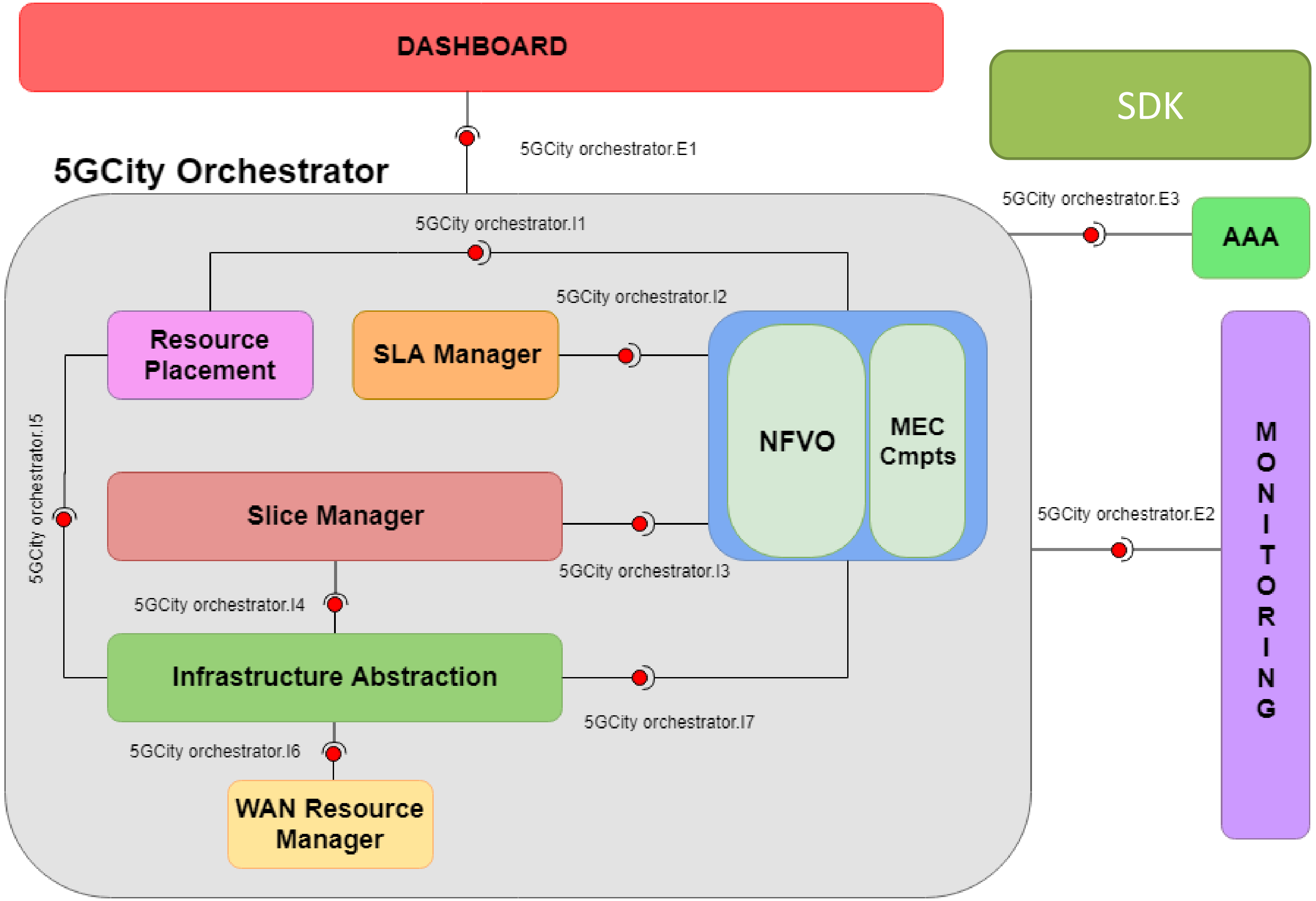
5GCity architecture



5GCity Platform deployment



5GCity Platform high level architecture



Mapping into software modules

Implemented as:
- **5GCity-Dashboard**

Implemented as:
- **5GCity-resource-placement**

Implemented as:
- **5GCity-slice-manager**

Implemented as:
- **5GCity-infrastructure-abstraction**

Implemented as proprietary code

Using:
- **ETSI OpenSourceMano (OSM)**

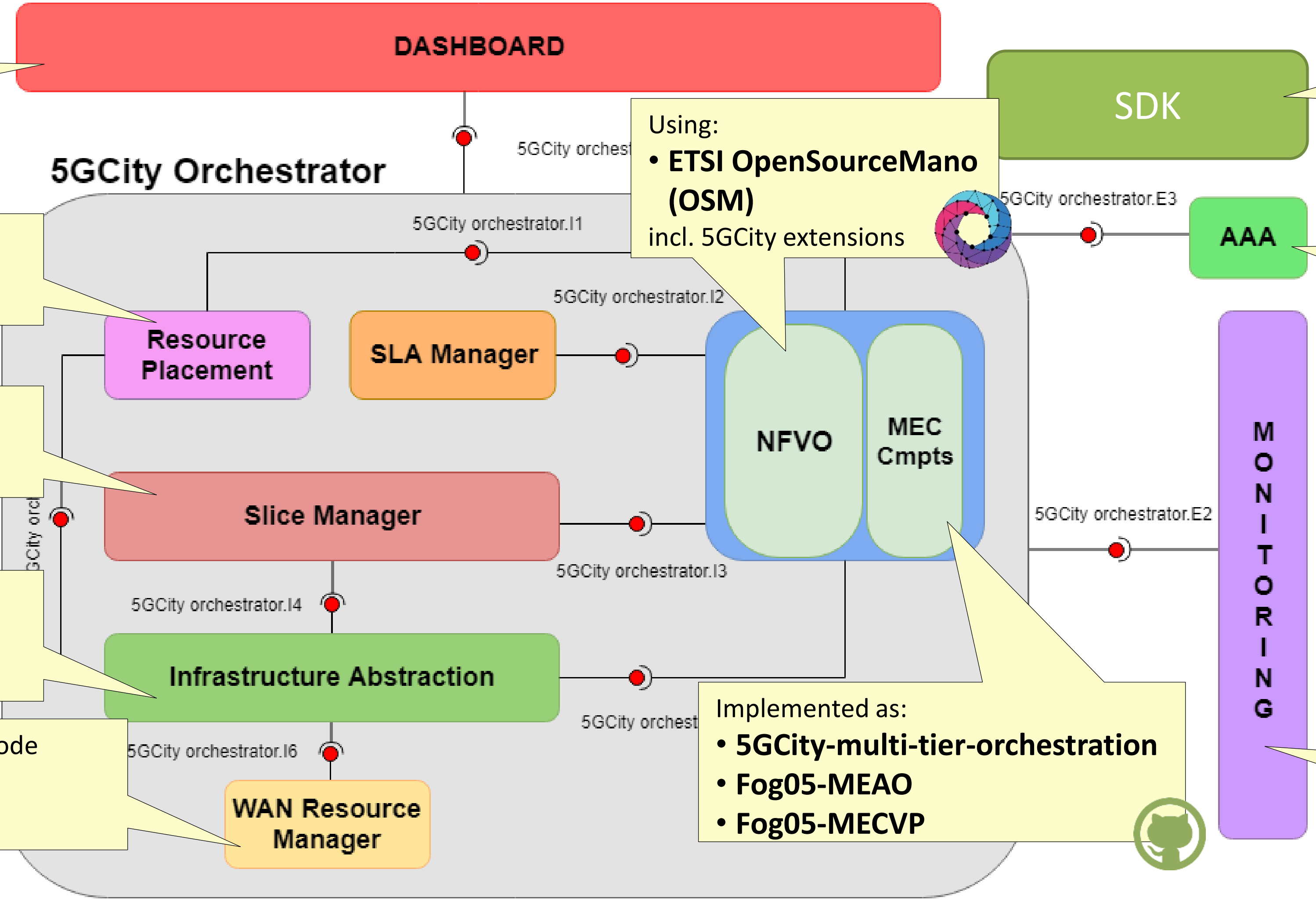
incl. 5GCity extensions

Implemented as:
- **5GCity-SDK**

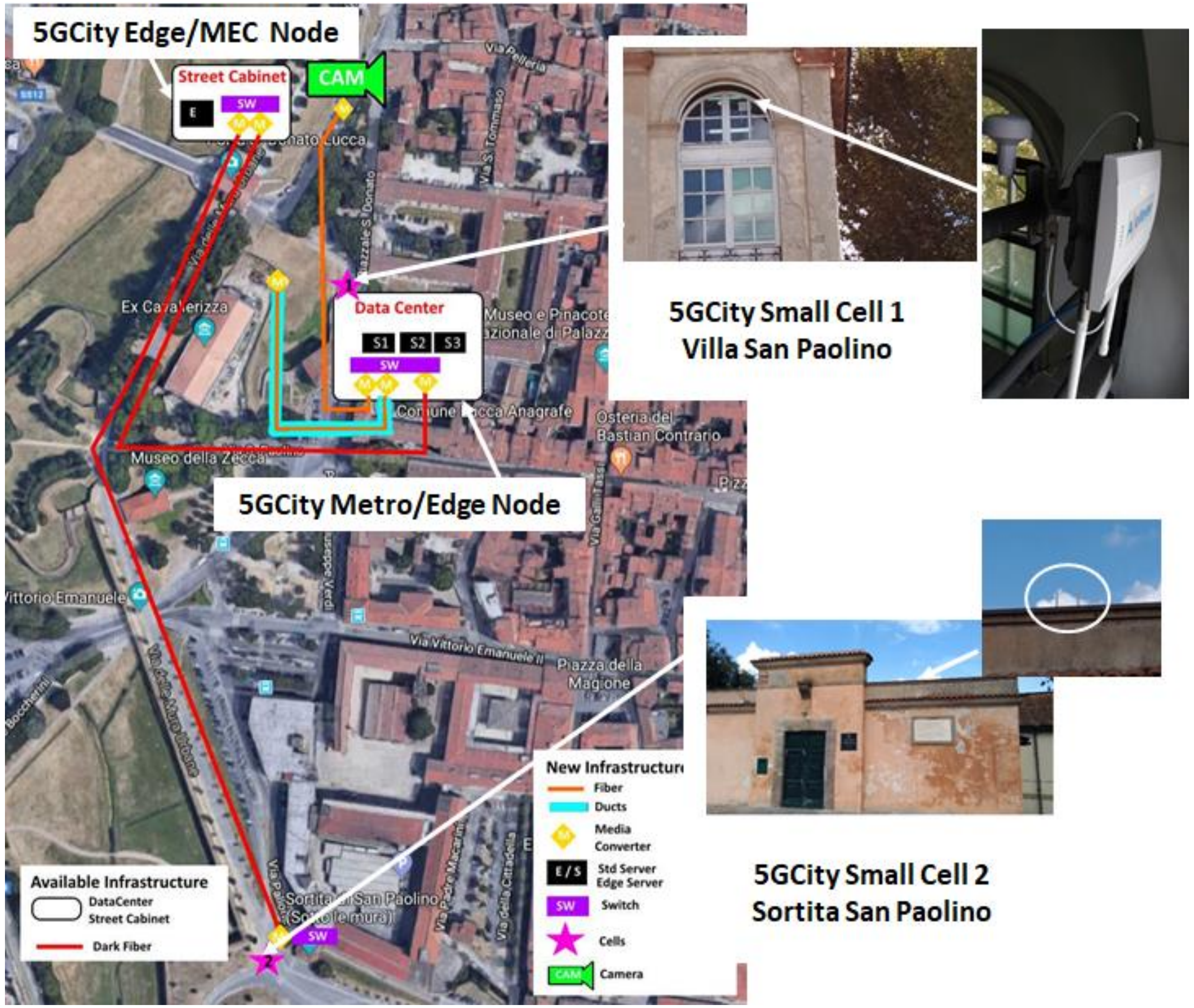
Implemented as:
- **5GCity-AAA**

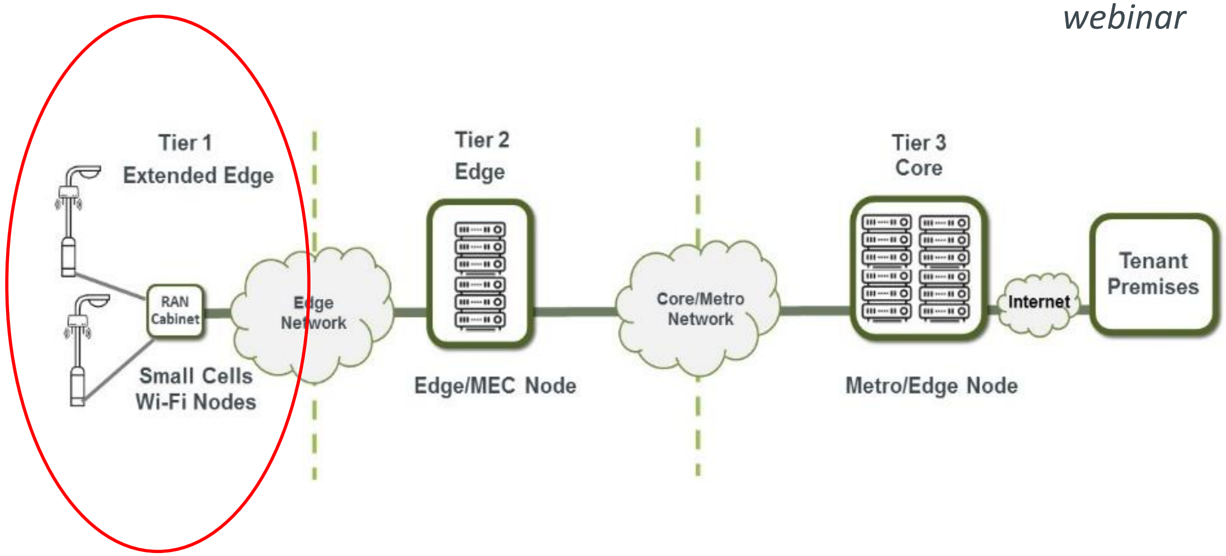
Implemented as:
- **5GCity-multi-tier-orchestration**
- **Fog05-MEAO**
- **Fog05-MECVP**

Implemented as:
- **5GCity-monitoring**

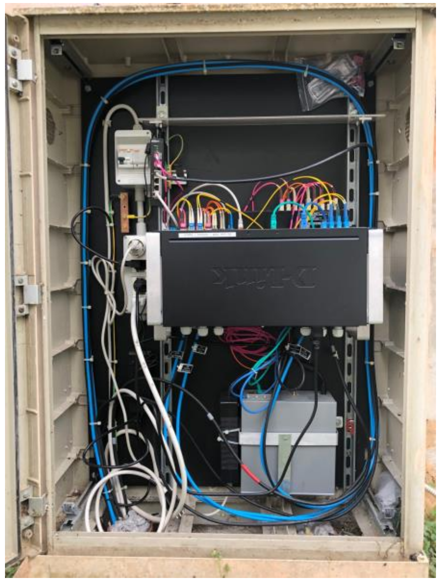


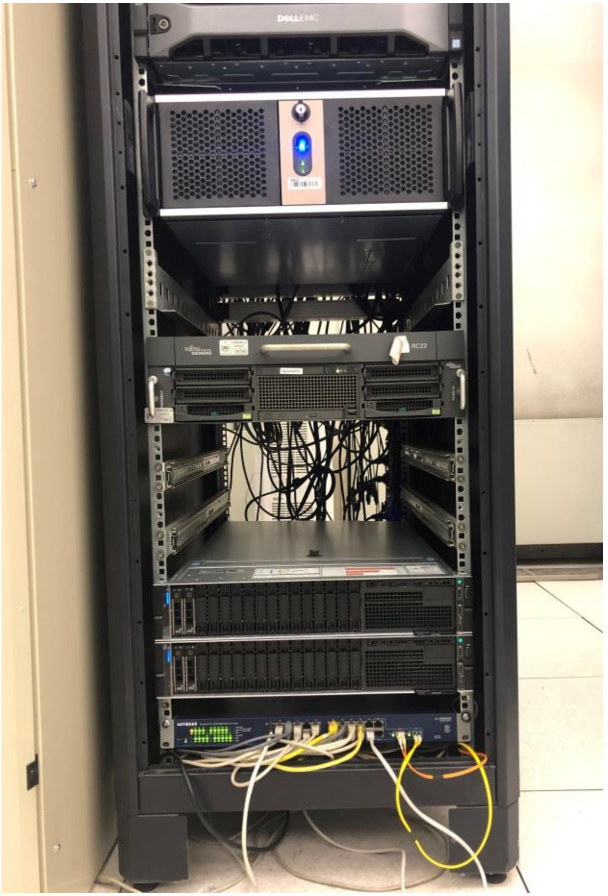
Deployments – City of Lucca

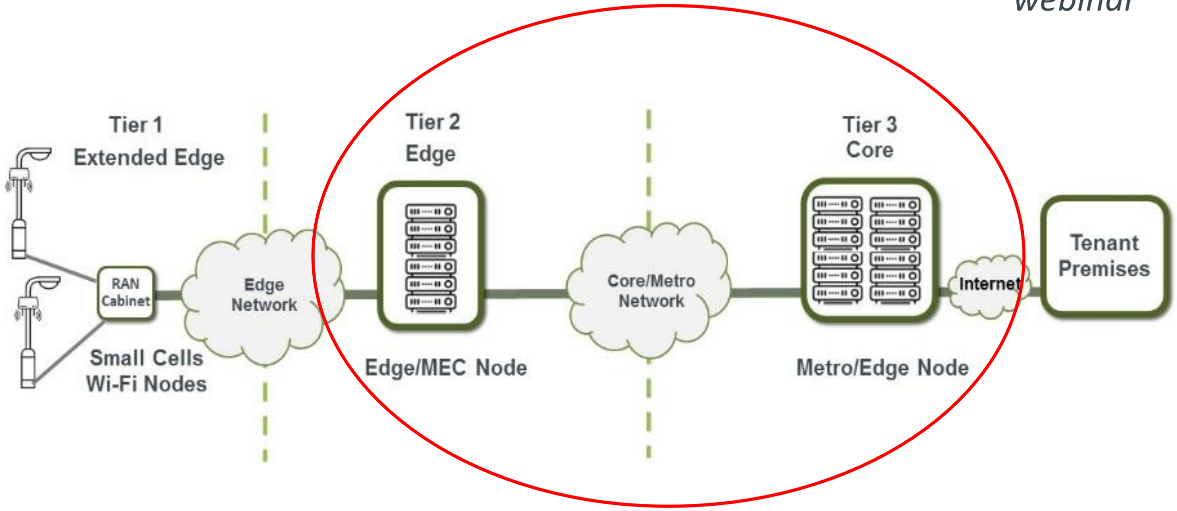




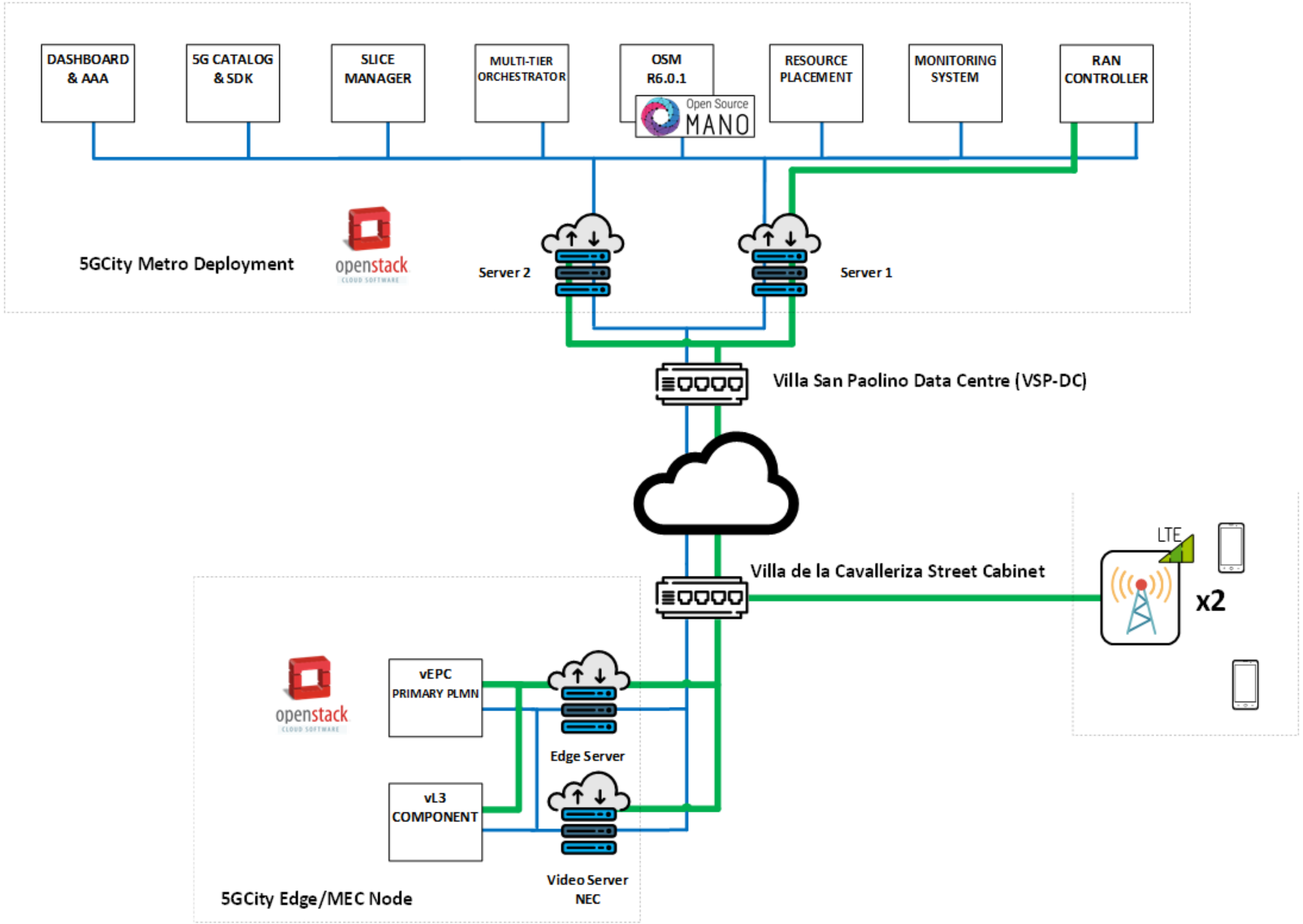
Deployments – City of Lucca



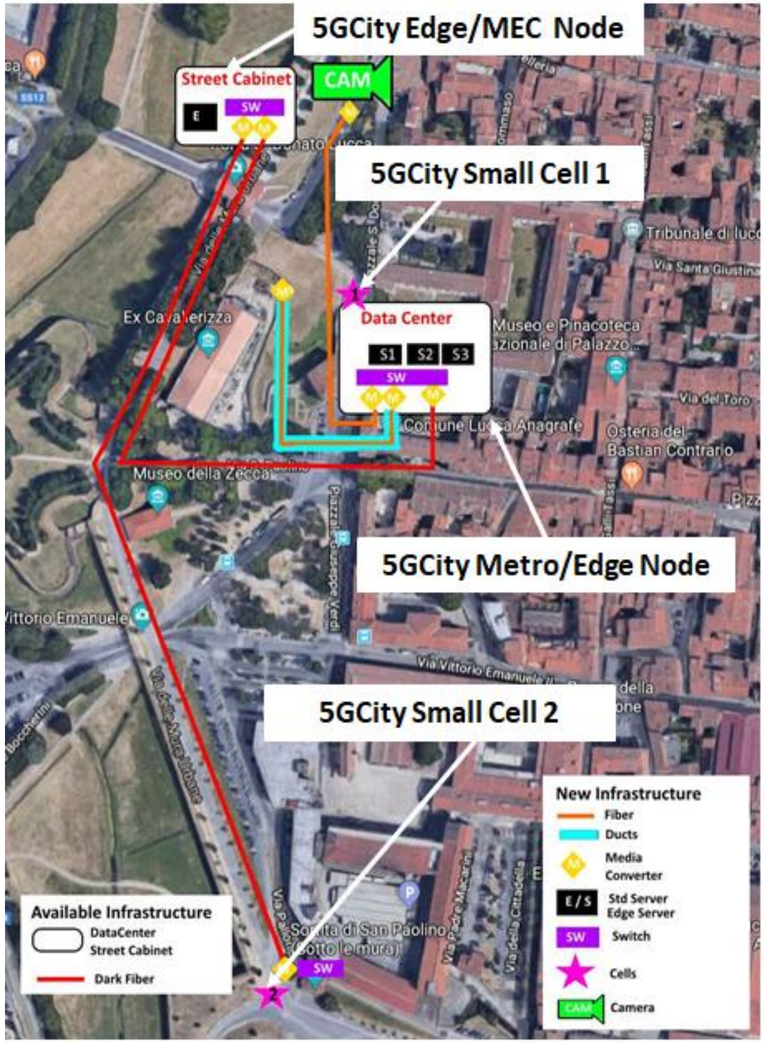


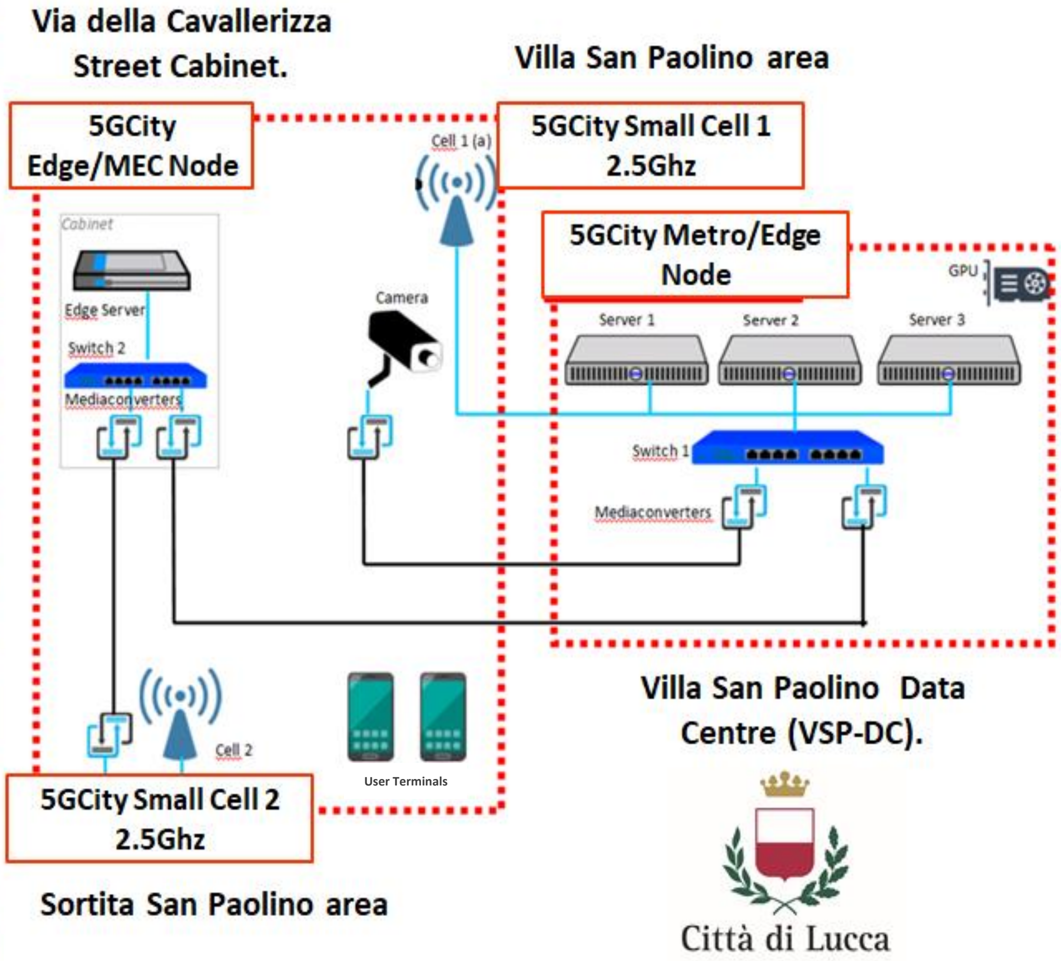


5GCity Platform in Lucca pilot



5GCity Lucca – physical deployment




Via della Cavallerizza Street Cabinet.

5GCity Edge/MEC Node

Villa San Paolino area

5GCity Small Cell 1 2.5Ghz

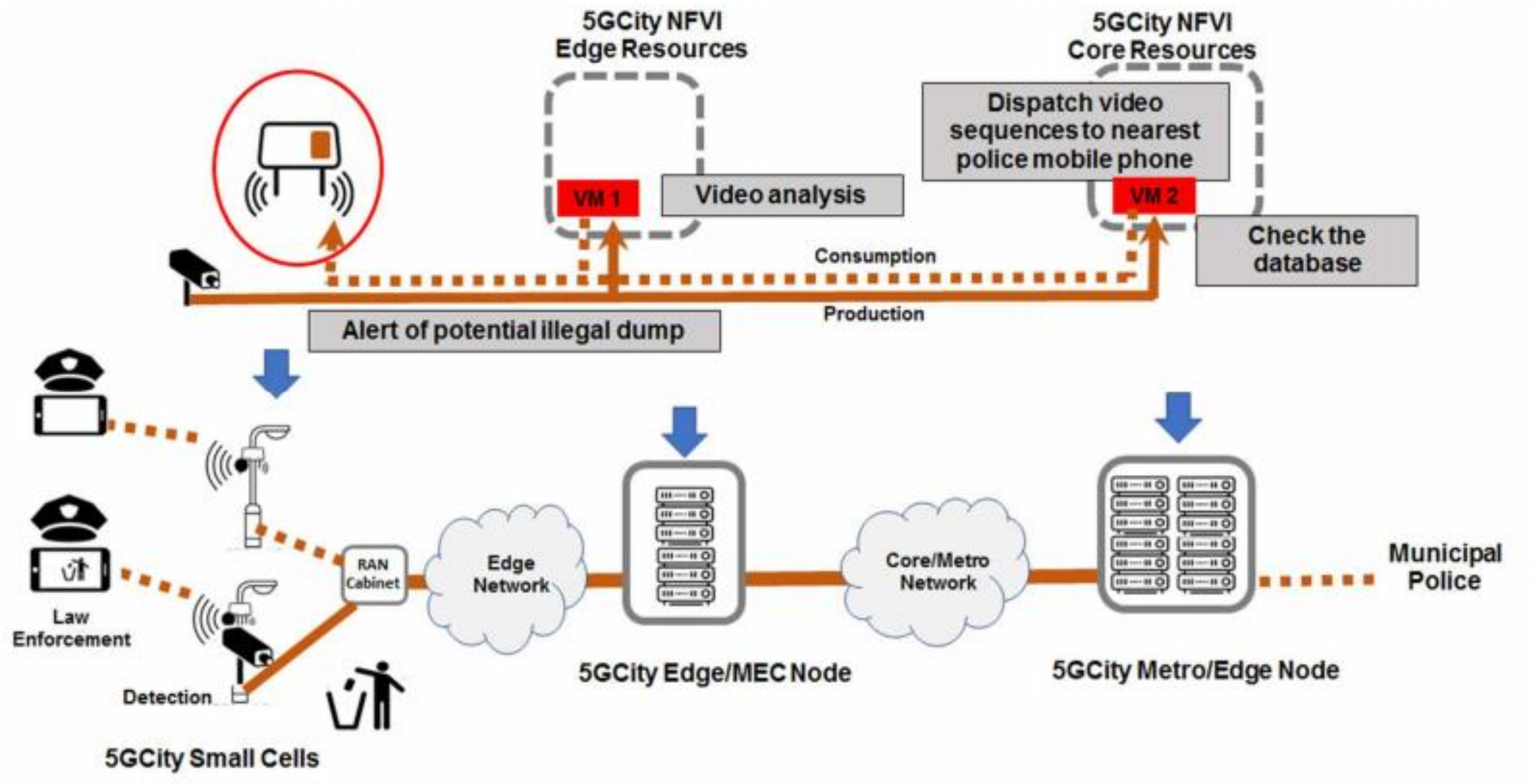
5GCity Metro/Edge Node

Villa San Paolino Data Centre (VSP-DC).

5GCity Small Cell 2 2.5Ghz

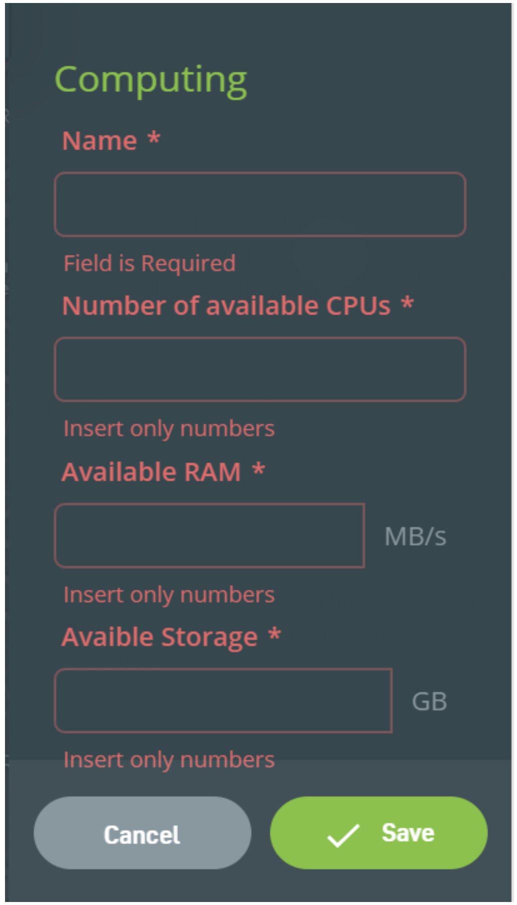
Sortita San Paolino area

Example Use Case (Waste Dumping)



5GCity platform main operations

Infrastructure registration



Computing

Name *

Field is Required

Number of available CPUs *

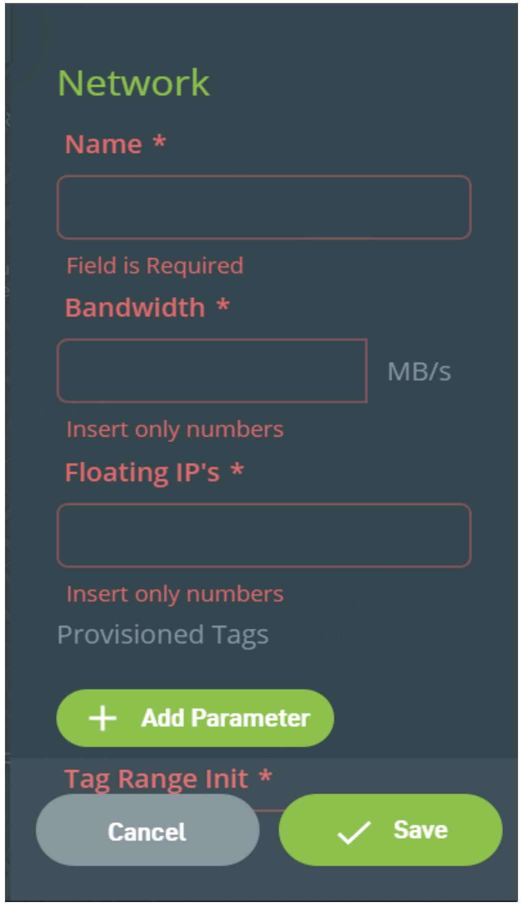
Insert only numbers

Available RAM * MB/s

Insert only numbers

Avaible Storage * GB

Insert only numbers



Network

Name *

Field is Required

Bandwidth * MB/s

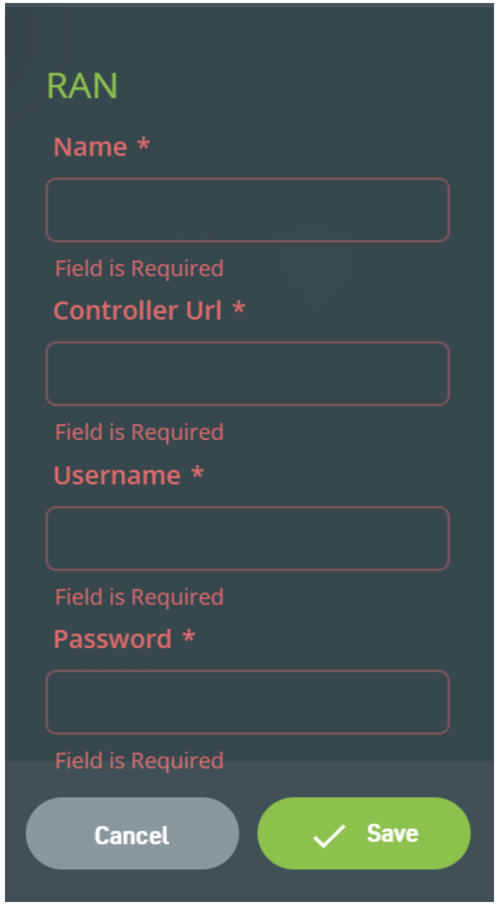
Insert only numbers

Floating IP's *

Insert only numbers

Provisioned Tags

Tag Range Init *



RAN

Name *

Field is Required

Controller Url *

Field is Required

Username *

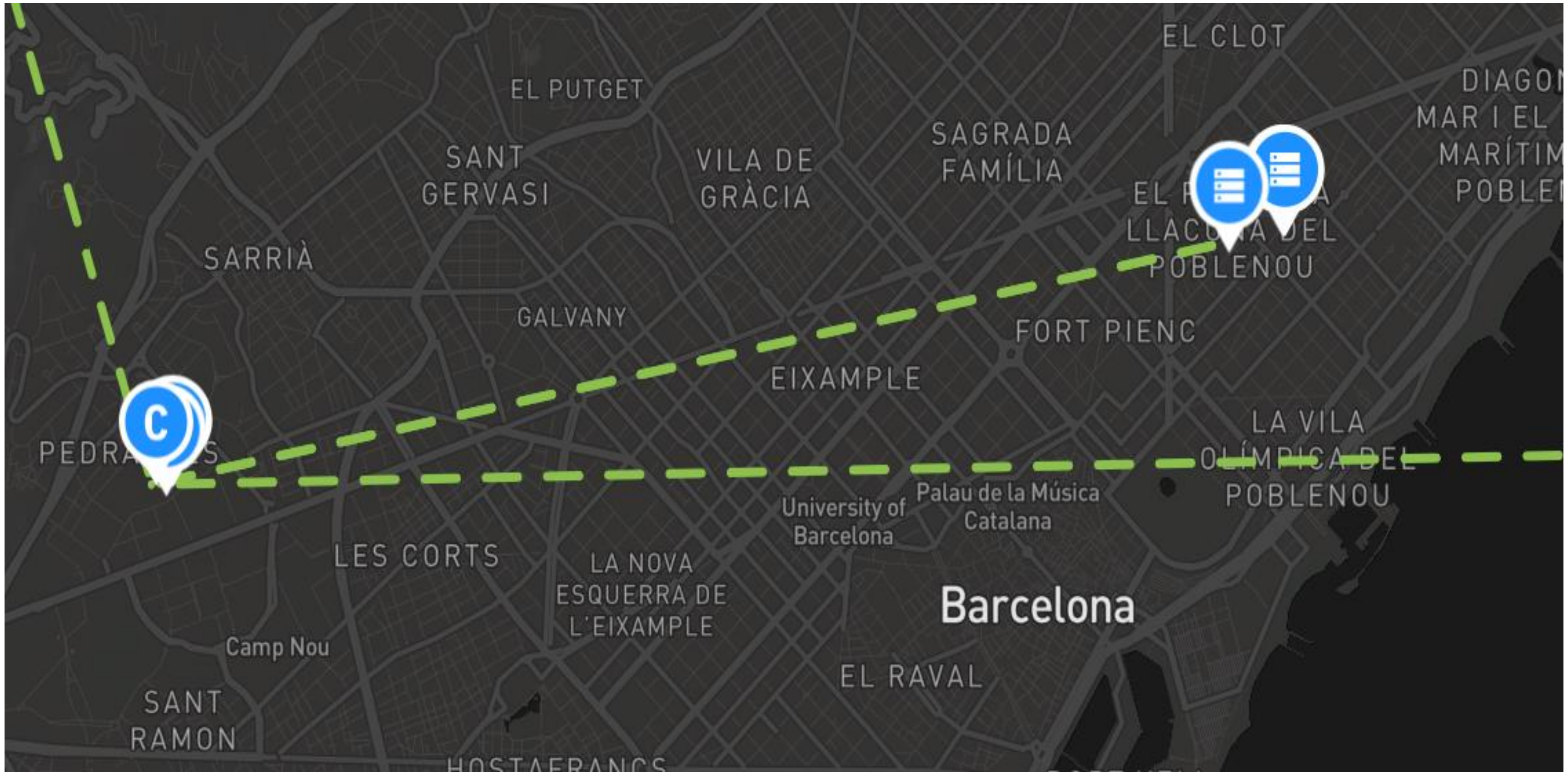
Field is Required

Password *

Field is Required

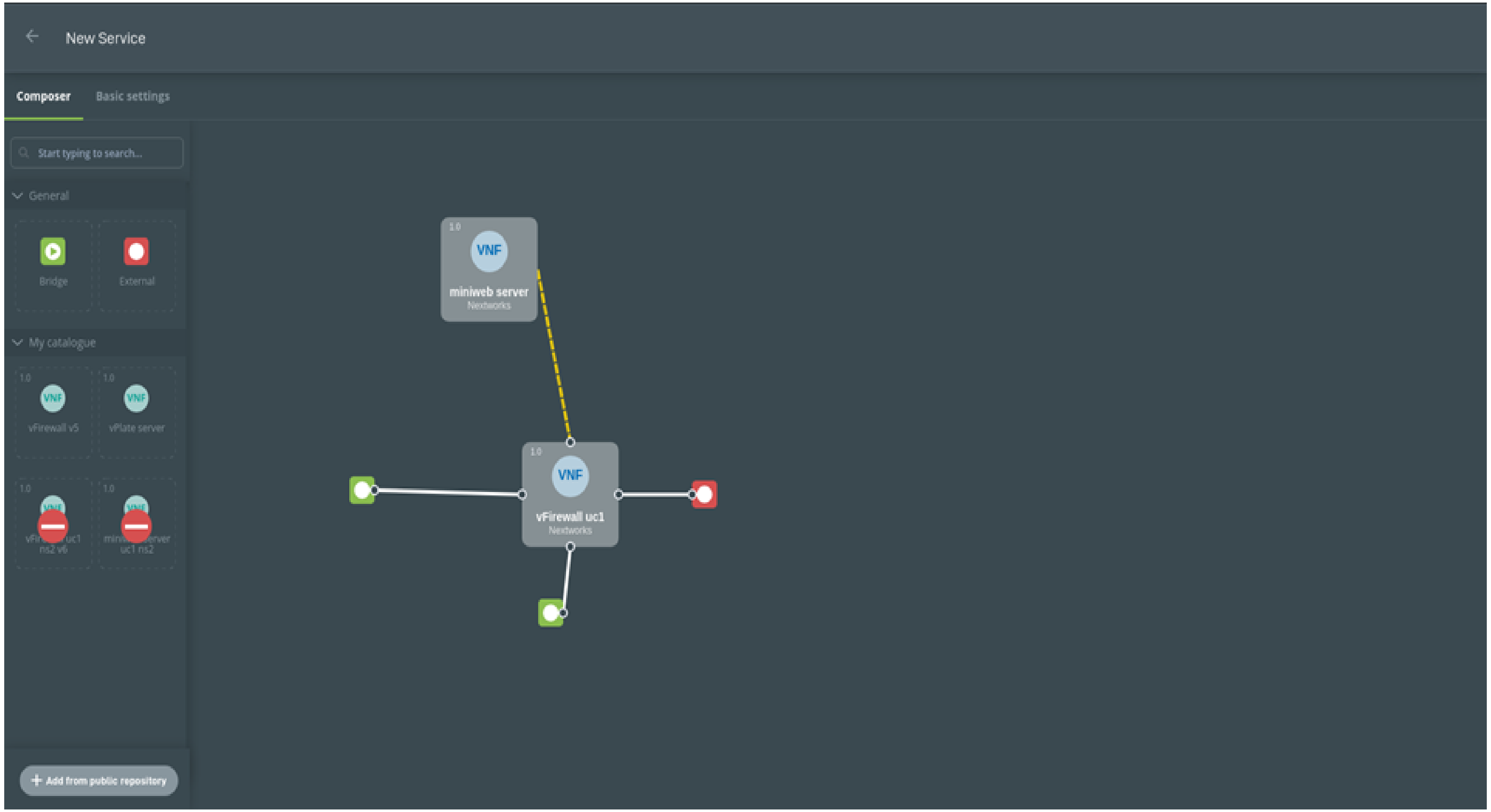
5GCity platform main operations

Slice creation and activation



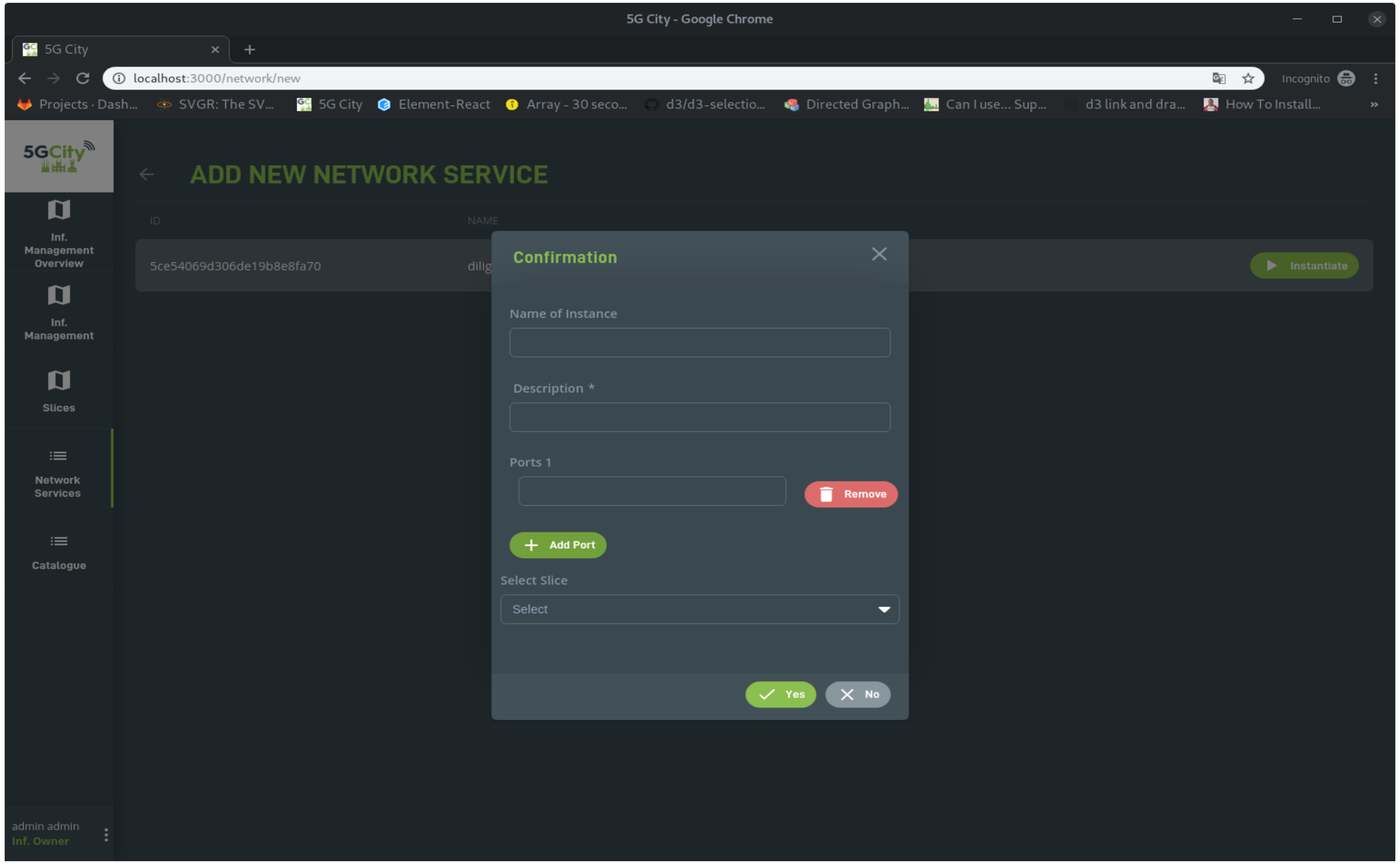
5GCity platform main operations

Service creation



5GCity platform main operations

Service creation



ADD NEW NETWORK SERVICE

ID	NAME
5ce54069d306de19b8e8fa70	dillg

Confirmation

Name of Instance

Description *

Ports 1

Remove

Add Port

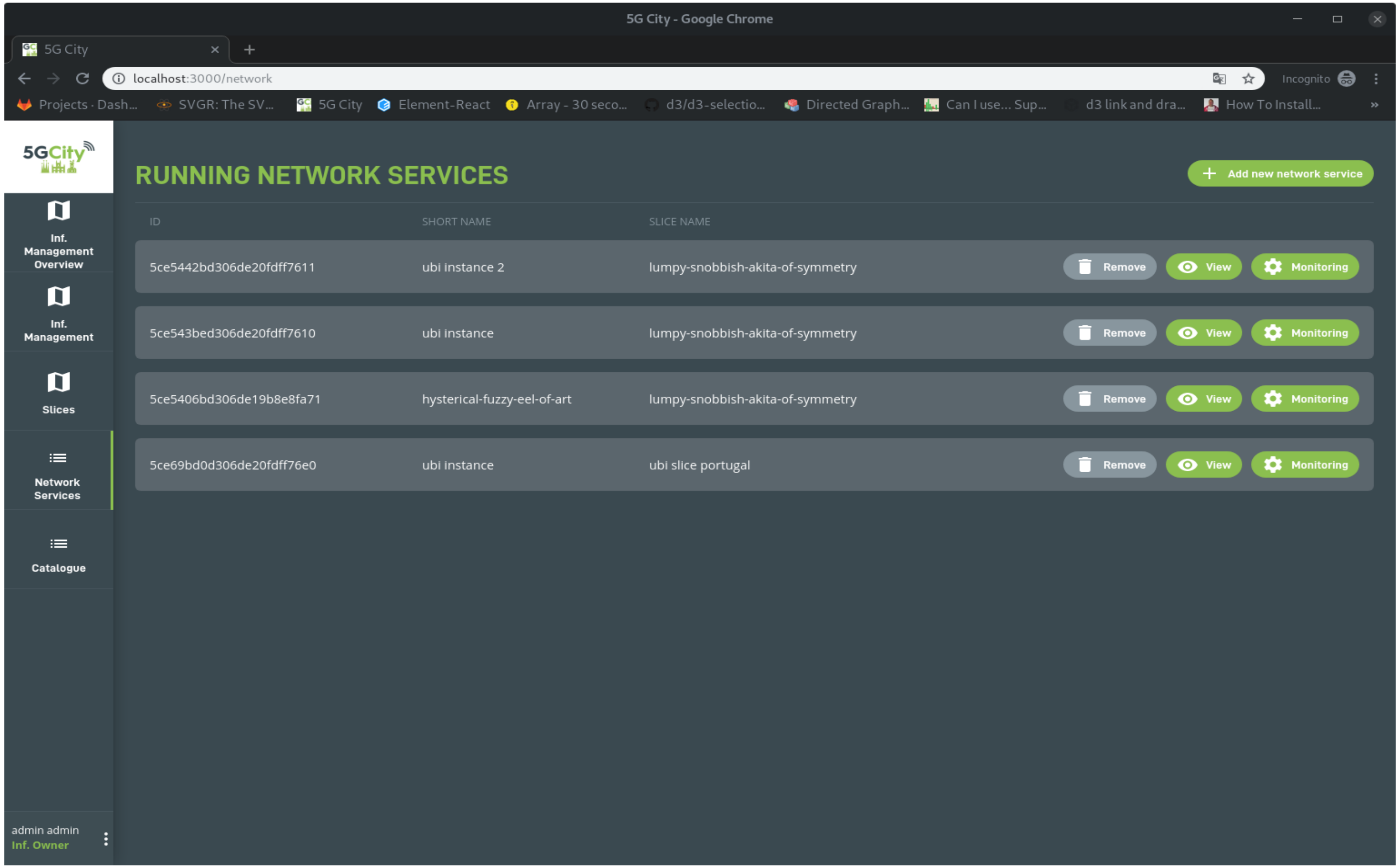
Select Slice

Select

Yes No

5GCity platform main operations

Service instantiation



RUNNING NETWORK SERVICES

ID	SHORT NAME	SLICE NAME	
5ce5442bd306de20fdff7611	ubi instance 2	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce543bed306de20fdff7610	ubi instance	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce5406bd306de19b8e8fa71	hysterical-fuzzy-eel-of-art	lumpy-snobbish-akita-of-symmetry	Remove View Monitoring
5ce69bd0d306de20fdff76e0	ubi instance	ubi slice portugal	Remove View Monitoring

Service design with the 5GCity SDK

Leonardo Agueci

from **Nextworks**

5GCity SDK software team

5GCity SDK

A **graphical environment** to compose functions in end-to-end services

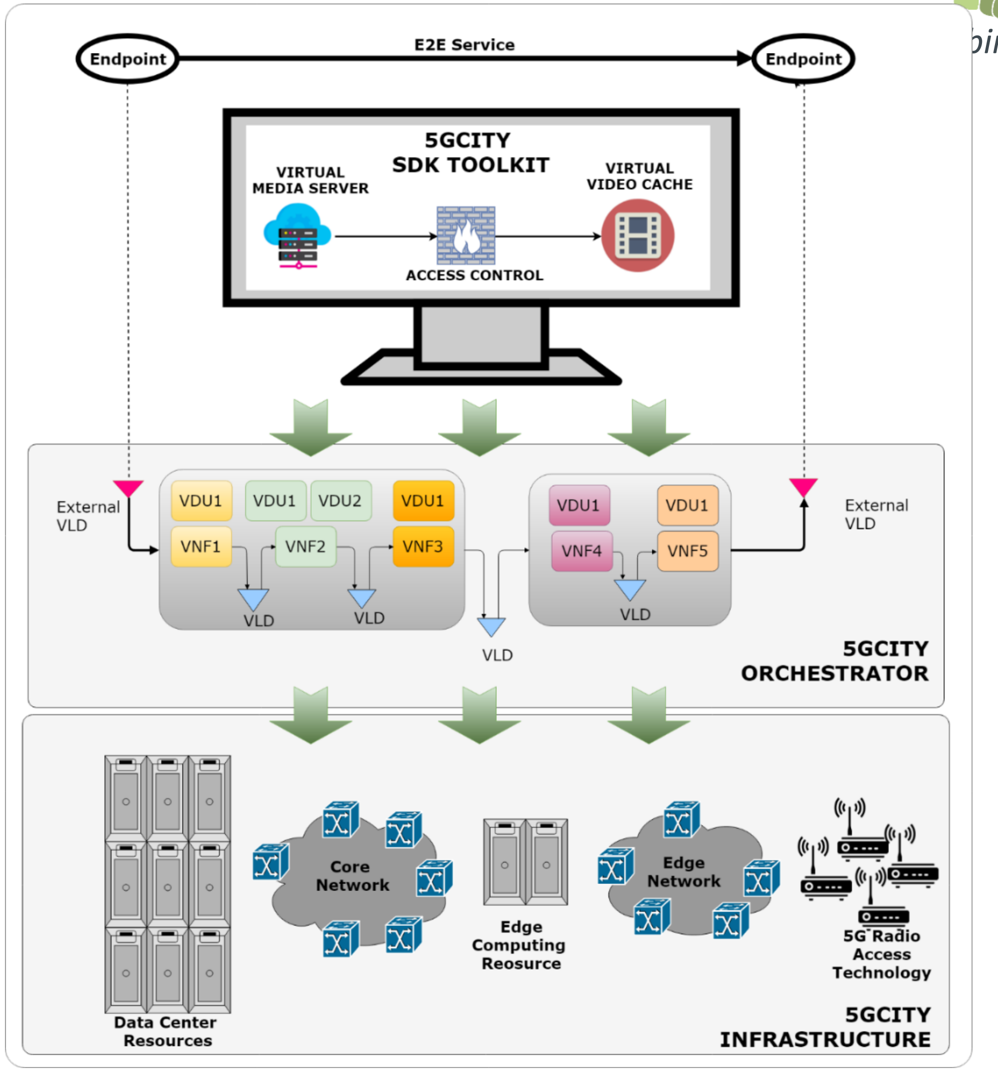
- Wire, in an arbitrary sequence, a pre-defined set of functions defining a logical service chain
- Create & reuse service templates, ready to be deployed in 5GCity infrastructure

An **adaptation layer**

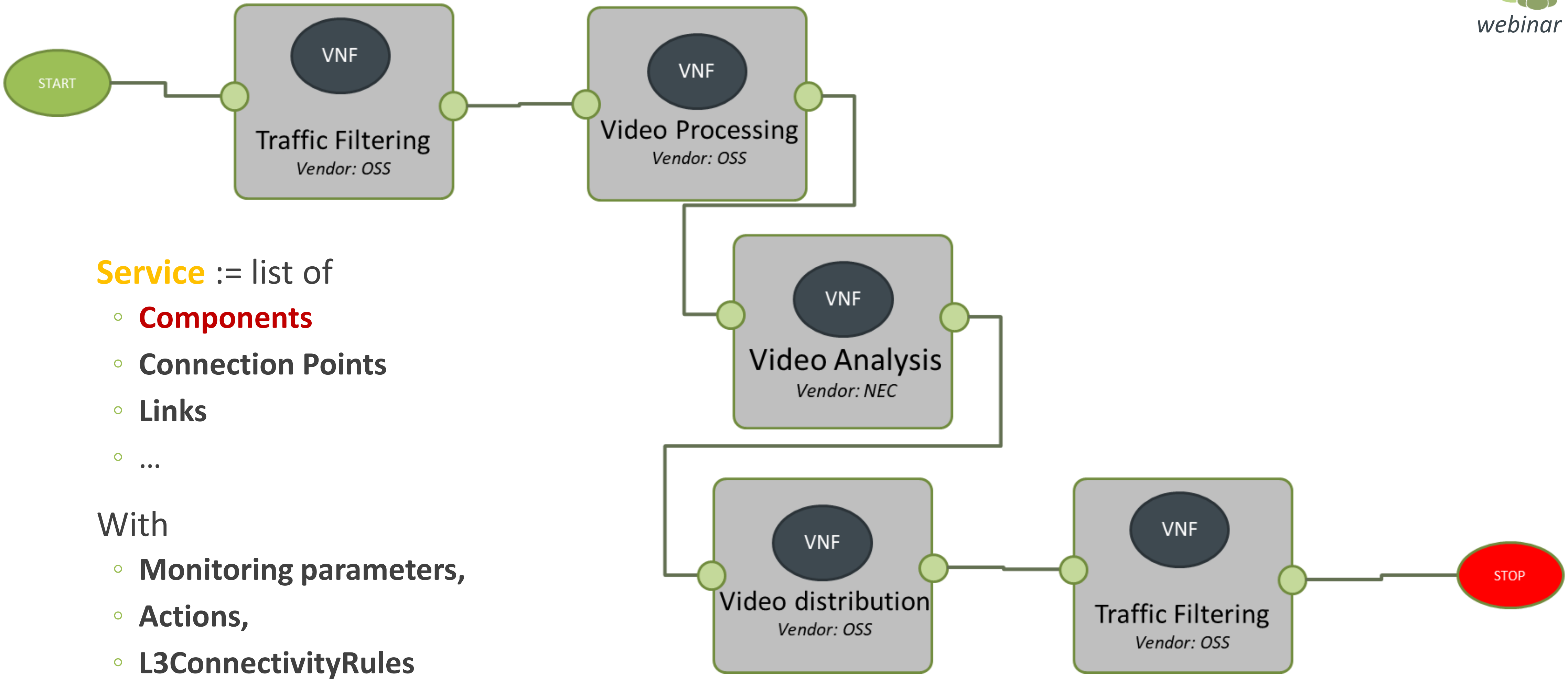
- to hide the complexity of the infrastructure and automatically translate functional components and business requirements into an operational service

A **validation module**

- to perform the formal validation of the service templates and verify the consistency of the end-to-end service pattern



SDK model



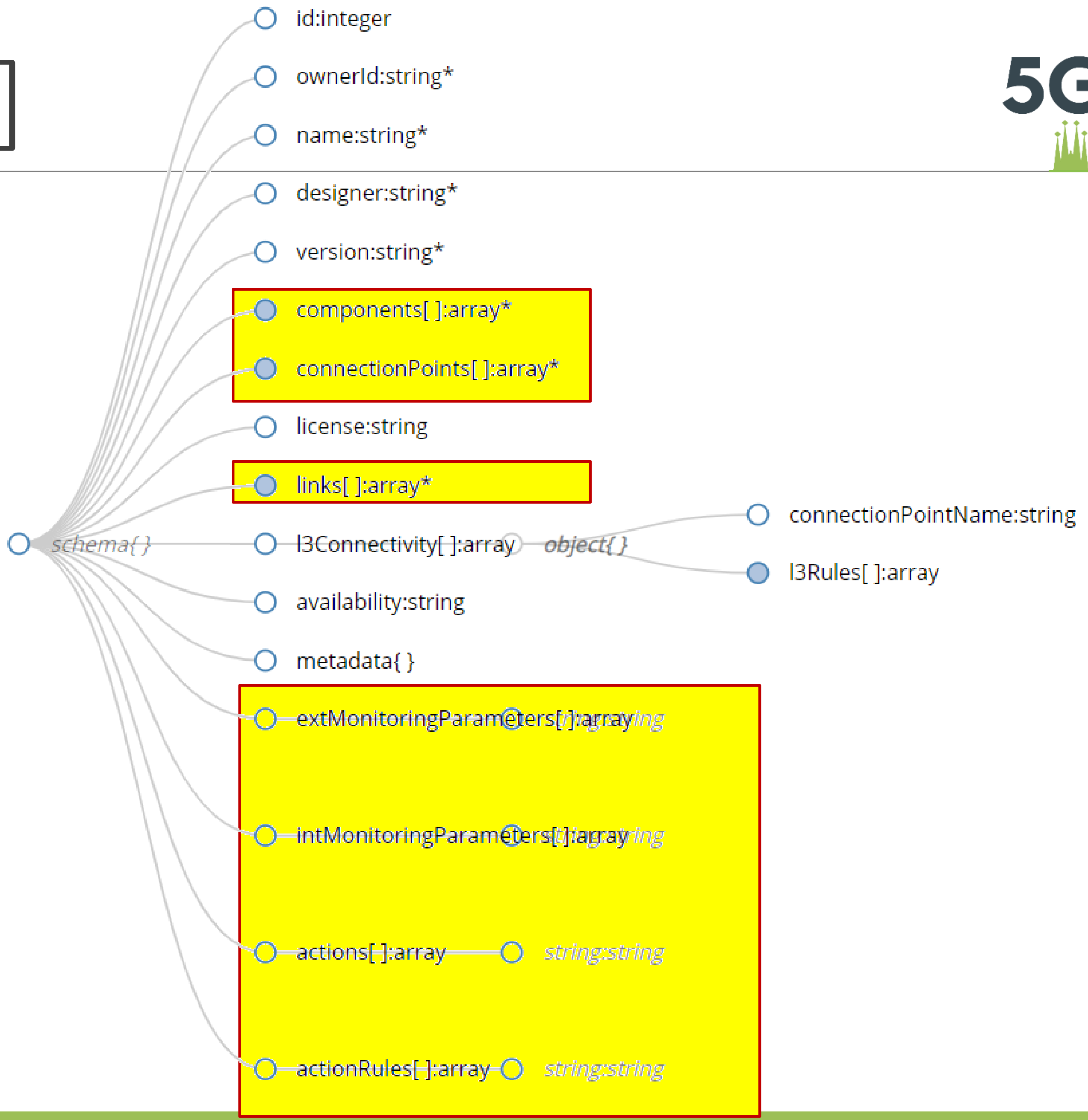
Service := list of

- **Components**
- **Connection Points**
- **Links**
- …

With

- **Monitoring parameters,**
- **Actions,**
- **L3ConnectivityRules**
- …

SDK model [2]


id:integer
ownerId:string*
name:string*
designer:string*
version:string*
components[ ]:array*
connectionPoints[ ]:array*
license:string
links[ ]:array*
schema{ }
l3Connectivity[ ]:array
object{ }
connectionPointName:string
l3Rules[ ]:array
availability:string
metadata{ }
extMonitoringParameters[ ]:array
string:string
intMonitoringParameters[ ]:array
string:string
actions[ ]:array
string:string
actionRules[ ]:array
string:string

5GCity SDK high-level architecture

5GCity Dashboard

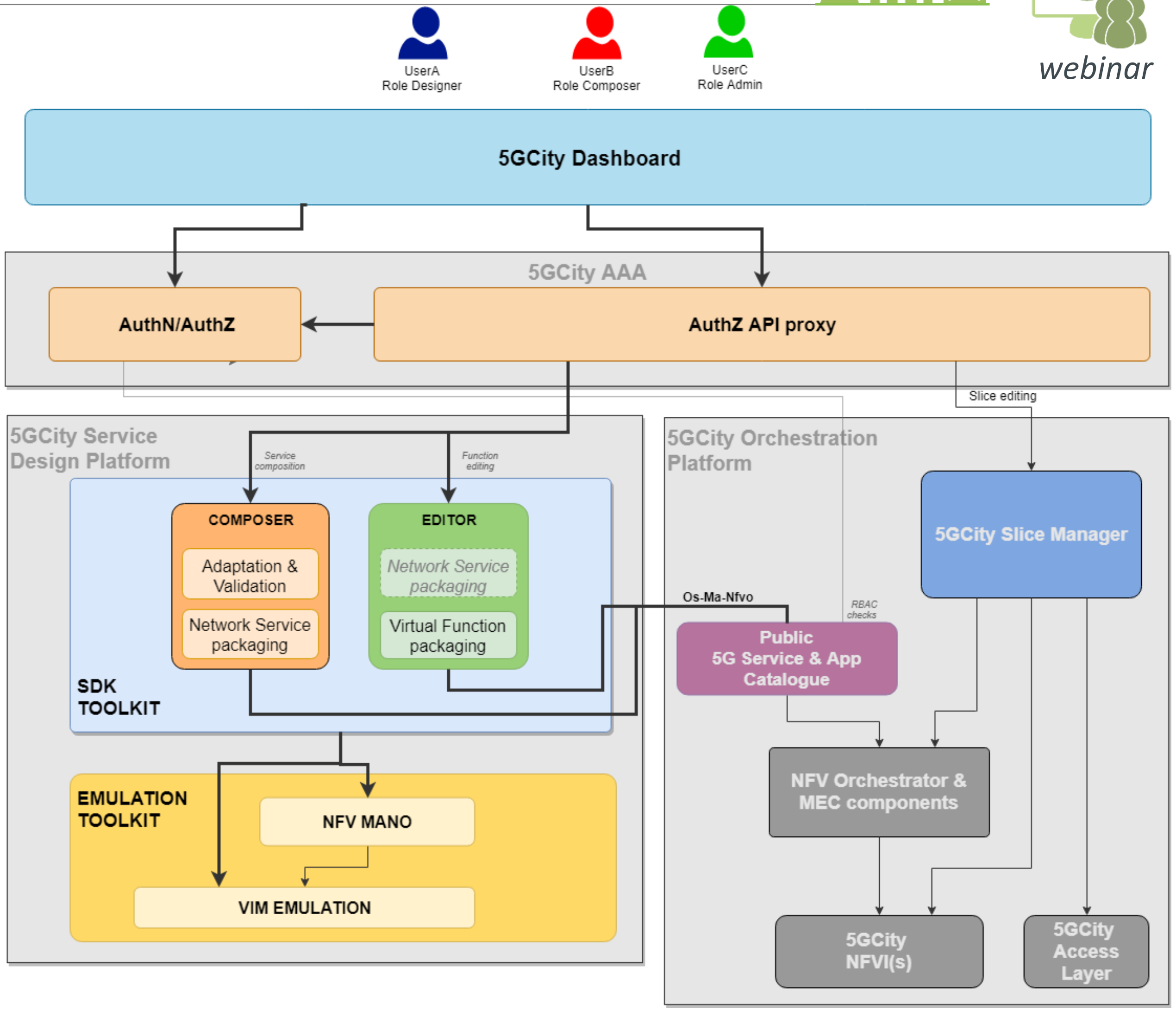
- entry point to the SDK platform, allowing different categories of users to define new 5G services based on different levels of abstraction

SDK Composer

- allows the Vertical user to compose services and perform CRUD operations on services
- Possibility to publish the service into a public 5G Service and Application Catalogue, after an internal process of validation and translation into the ETSI TOSCA model
- The 5G Service and Application Catalogue is part of the MANO infrastructure, and will be used to onboard ETSI compliant Network Services and VNF package descriptors into the underlying NFV Orchestrator

SDK Editor

- allows the DevOps/admin user to define new functions that are used by the vertical user to create new services or update already created ones
- A SDK function is an abstraction of NFV parameters better known from a DevOps user when he/she intends to create a vertical service

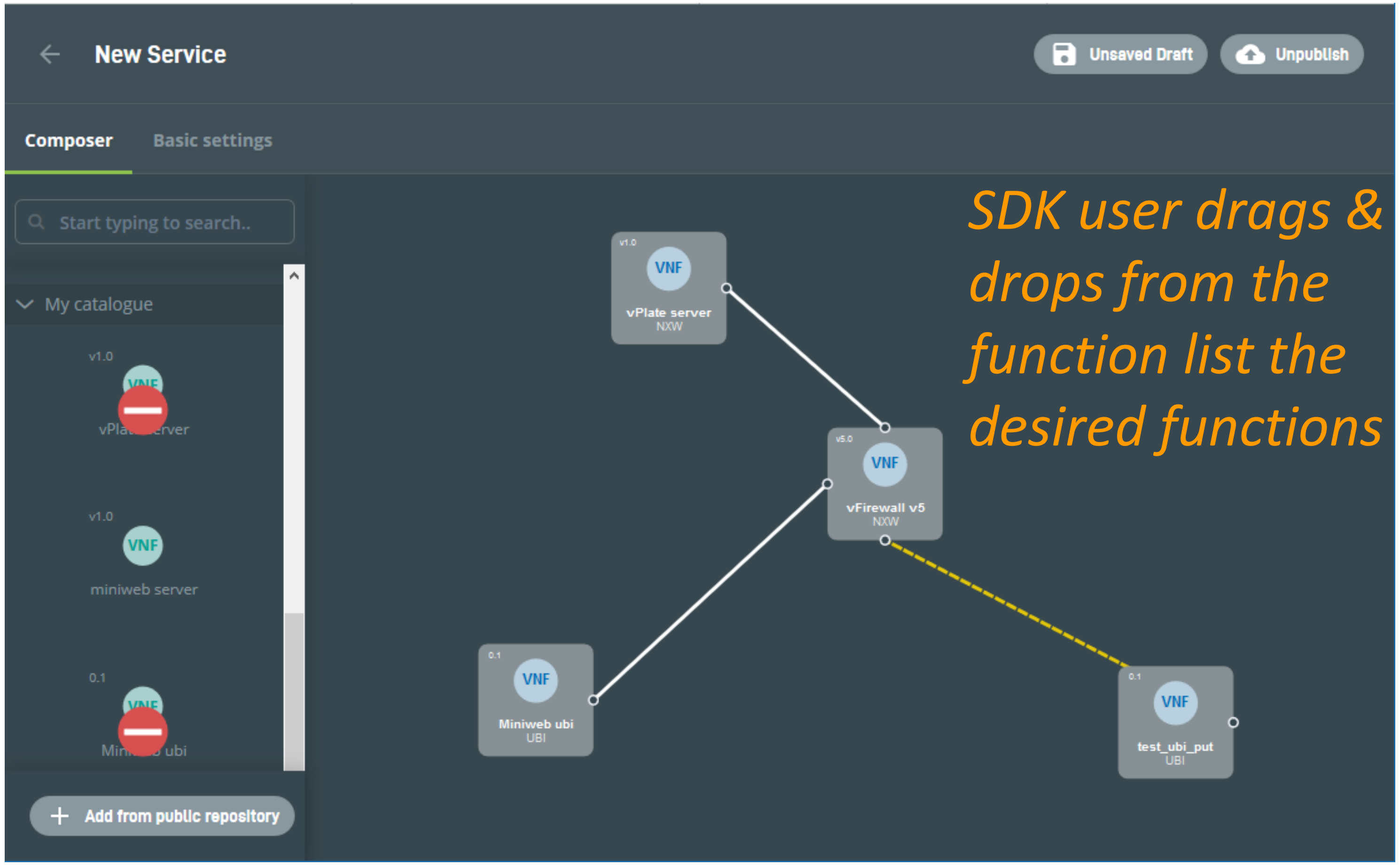


5GCity SDK GUI

@Composer

- Design a new service
- Modify an existing one
- Publish a service to the catalogue

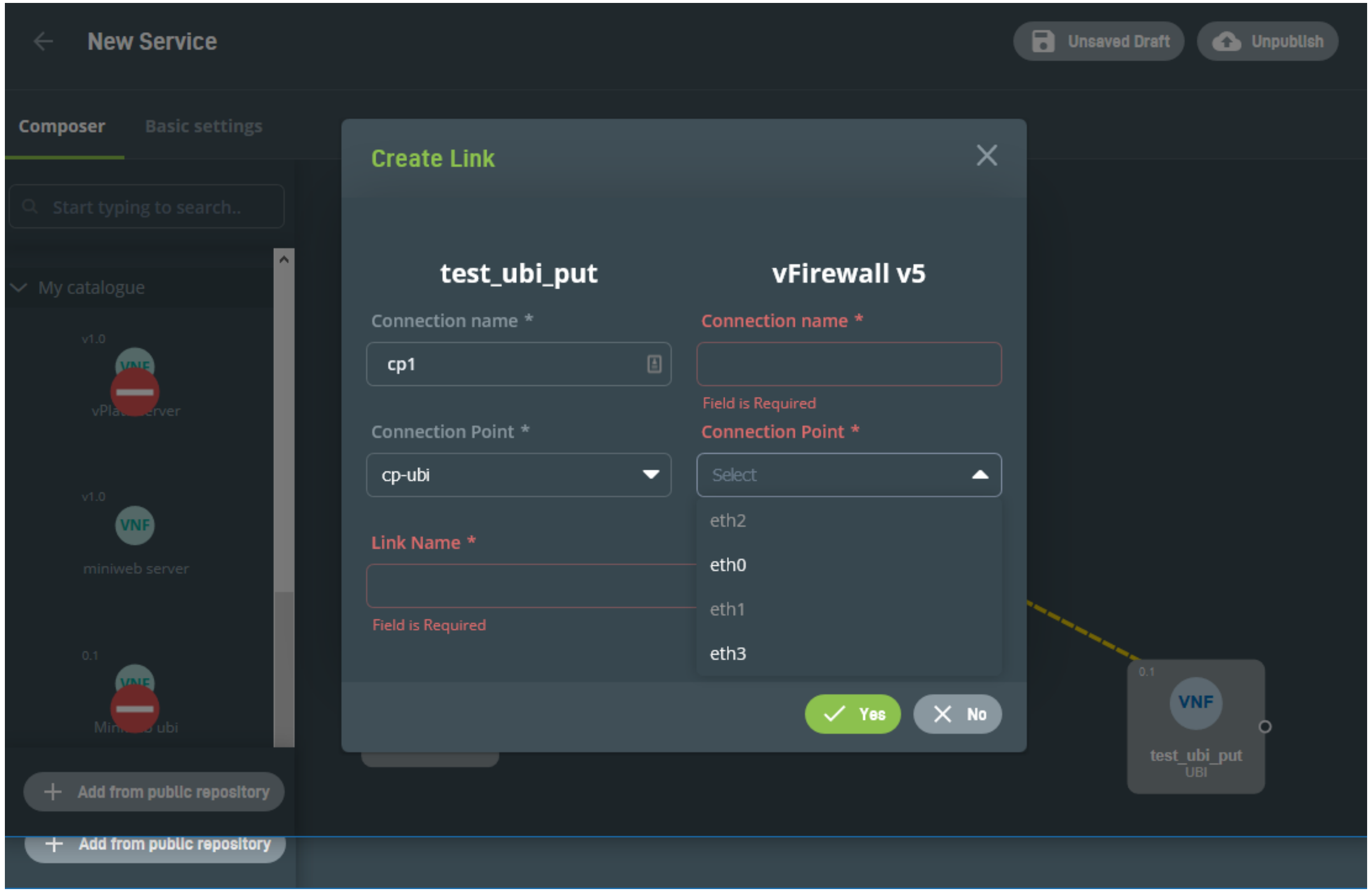
SDK functions from Editor should be already present



SDK user drags & drops from the function list the desired functions

5GCity SDK GUI (2)

Creating a new service



New Service

Create Link

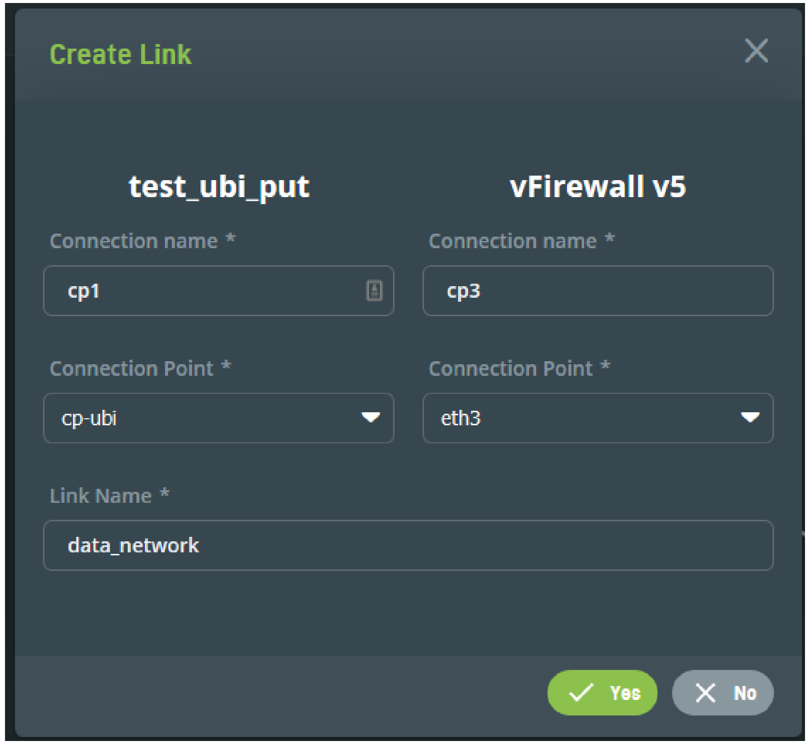
test_ubi_put

- Connection name *: cp1
- Connection Point *: cp-ubi
- Link Name *: Field is Required

vFirewall v5

- Connection name *: Field is Required
- Connection Point *: Select

User creates a connection (link) between two functions



Create Link

test_ubi_put

- Connection name *: cp1
- Connection Point *: cp-ubi

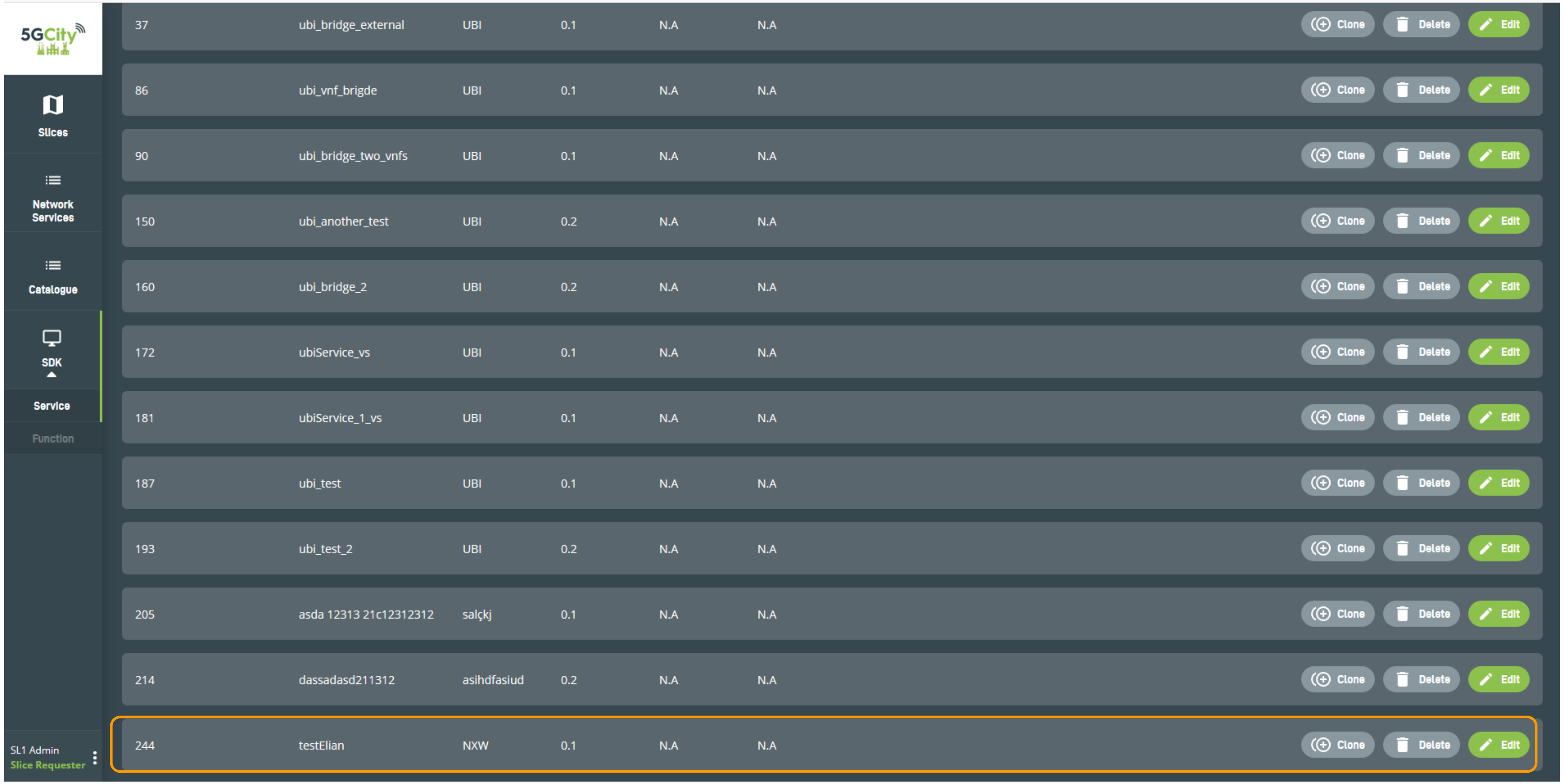
vFirewall v5

- Connection name *: cp3
- Connection Point *: eth3

Link Name *: data_network

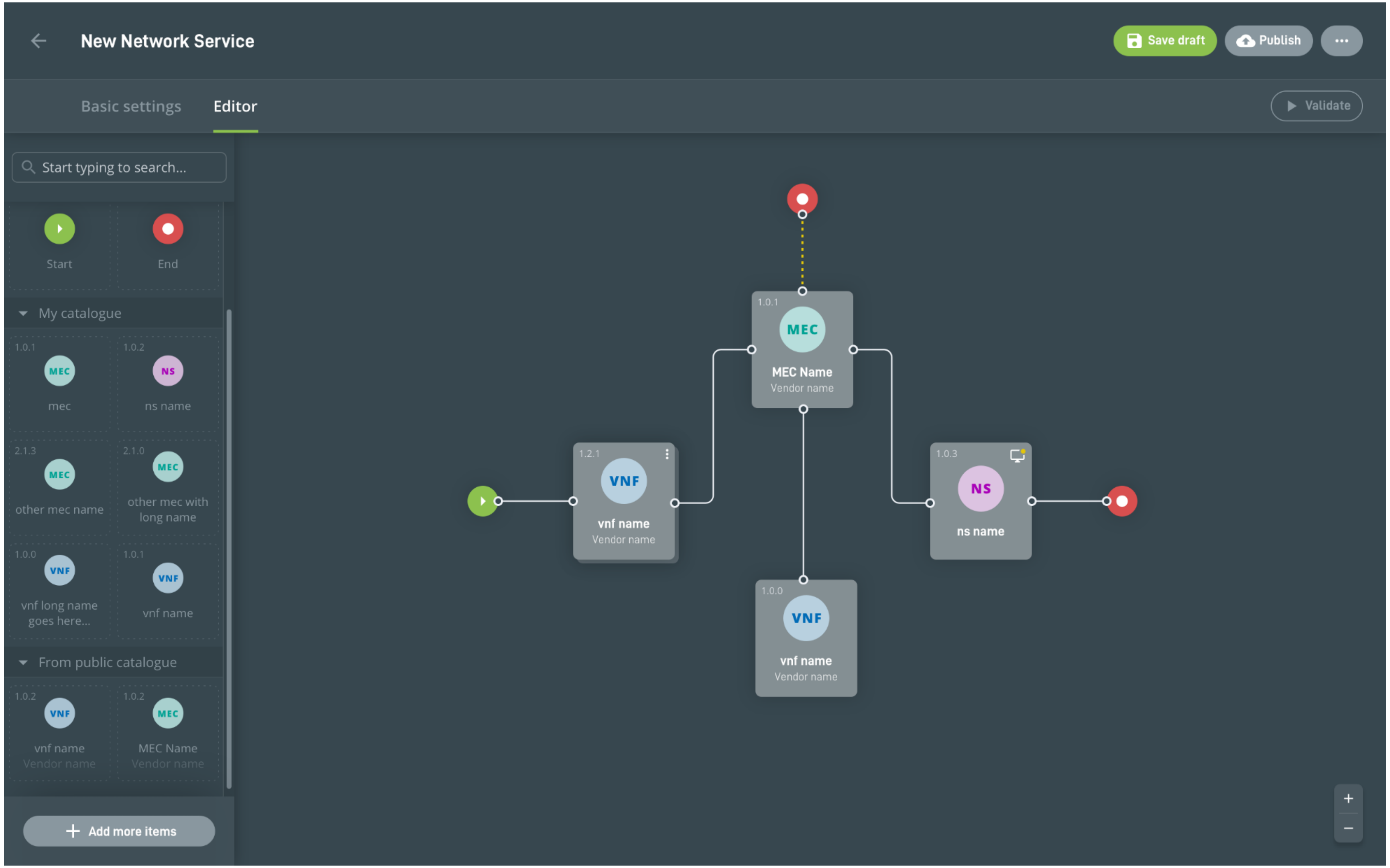
5GCity SDK GUI (3)

New service is available and ready to be published to the catalogue



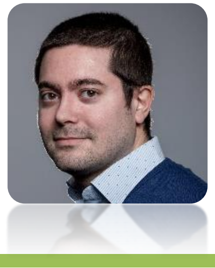
37	ubi_bridge_external	UBI	0.1	N.A	N.A	Clone	Delete	Edit
86	ubi_vnf_brigde	UBI	0.1	N.A	N.A	Clone	Delete	Edit
90	ubi_bridge_two_vnfs	UBI	0.1	N.A	N.A	Clone	Delete	Edit
150	ubi_another_test	UBI	0.2	N.A	N.A	Clone	Delete	Edit
160	ubi_bridge_2	UBI	0.2	N.A	N.A	Clone	Delete	Edit
172	ubiService_vs	UBI	0.1	N.A	N.A	Clone	Delete	Edit
181	ubiService_1_vs	UBI	0.1	N.A	N.A	Clone	Delete	Edit
187	ubi_test	UBI	0.1	N.A	N.A	Clone	Delete	Edit
193	ubi_test_2	UBI	0.2	N.A	N.A	Clone	Delete	Edit
205	asda 12313 21c12312312	salçkj	0.1	N.A	N.A	Clone	Delete	Edit
214	dassadasd211312	asihdfasiud	0.2	N.A	N.A	Clone	Delete	Edit
244	testElian	NXW	0.1	N.A	N.A	Clone	Delete	Edit

5GCity SDK GUI (4)



Unikraft

EXTREME SPECIALIZATION, SECURITY AND PERFORMANCE



Santiago Pagani

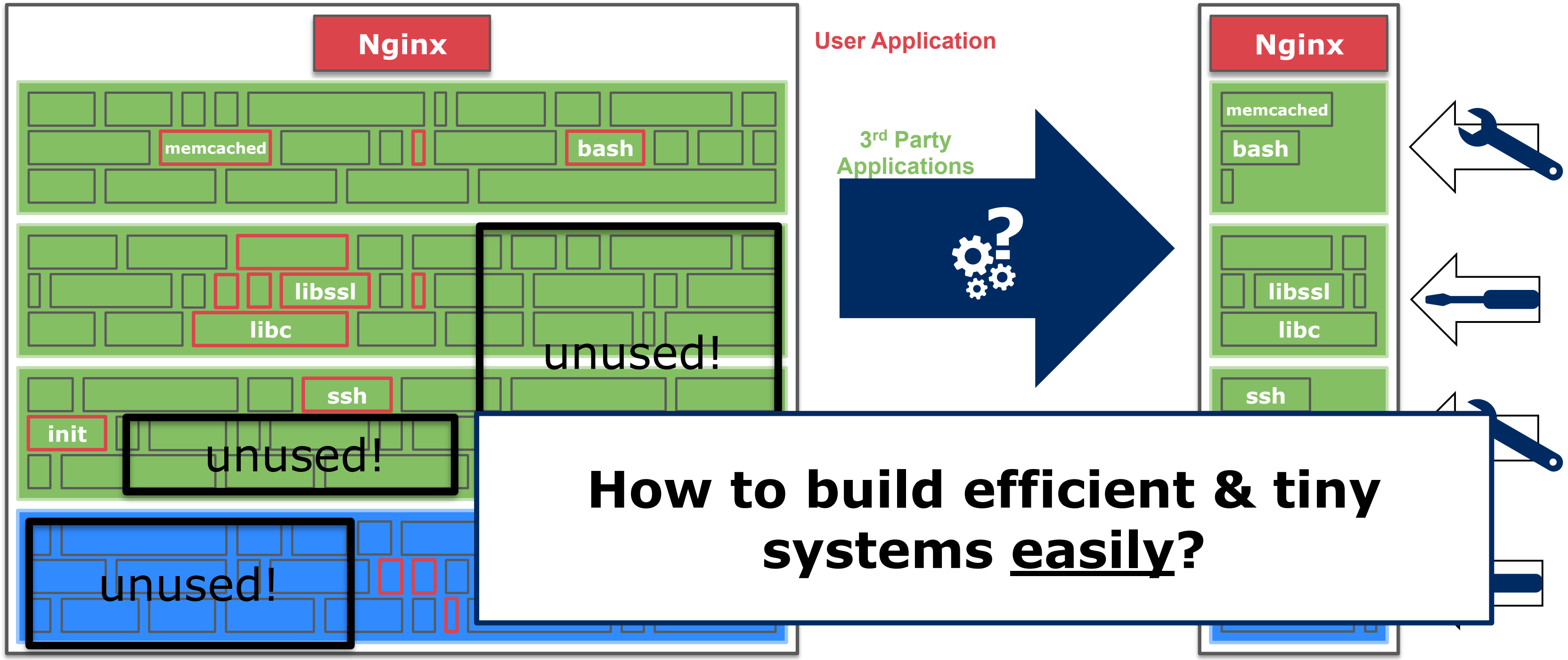
from **NEC Europe**

Unikernel/Unikraft team

Background: Specialization in Practice

- Today OS/VM/container: ***lots*** of unnecessary code = ***lots*** of overhead, ***big*** attack vector

- Specialized System: only what's needed is there ***but lots*** of development time! (has to be done **manually,** may require changing code)



How to build efficient & tiny systems <u>easily</u>?

Application on General Purpose OS

Specialized Unikernel

The Potential of Unikernels

webinar

Fast instantiation, destruction and migration time

- 10s of milliseconds or less (and as little as 2.3ms)
  (LigthVM [Manco SOSP 2017], Jitsu [Madhvapeddy, NSDI 2015])

Low memory footprint

- Few MBs of RAM or less *(ClickOS [Martins NSDI 2014])*

High density

- 8k guests on a singlex86 server *(LigthVM [Manco SOSP 2017])*

High Performance

- 10-40Gbit/s throughput with a single guest CPU
  (ClickOS [Martins NSDI 2014], Elastic CDNs [Kuenzer VEE 2017])

Reduced attack surface

- Small trusted compute base
- Strong isolation by hypervisor

Unikraft: Unikernel Framework

Motivation

- Support wide range of use cases
- Simplify building and optimizing
- Simplify porting of existing applications
- Common and shared code base for Unikernel projects: “win-win”
- Support for many hypervisors, bare-metal nodes, and CPU architectures

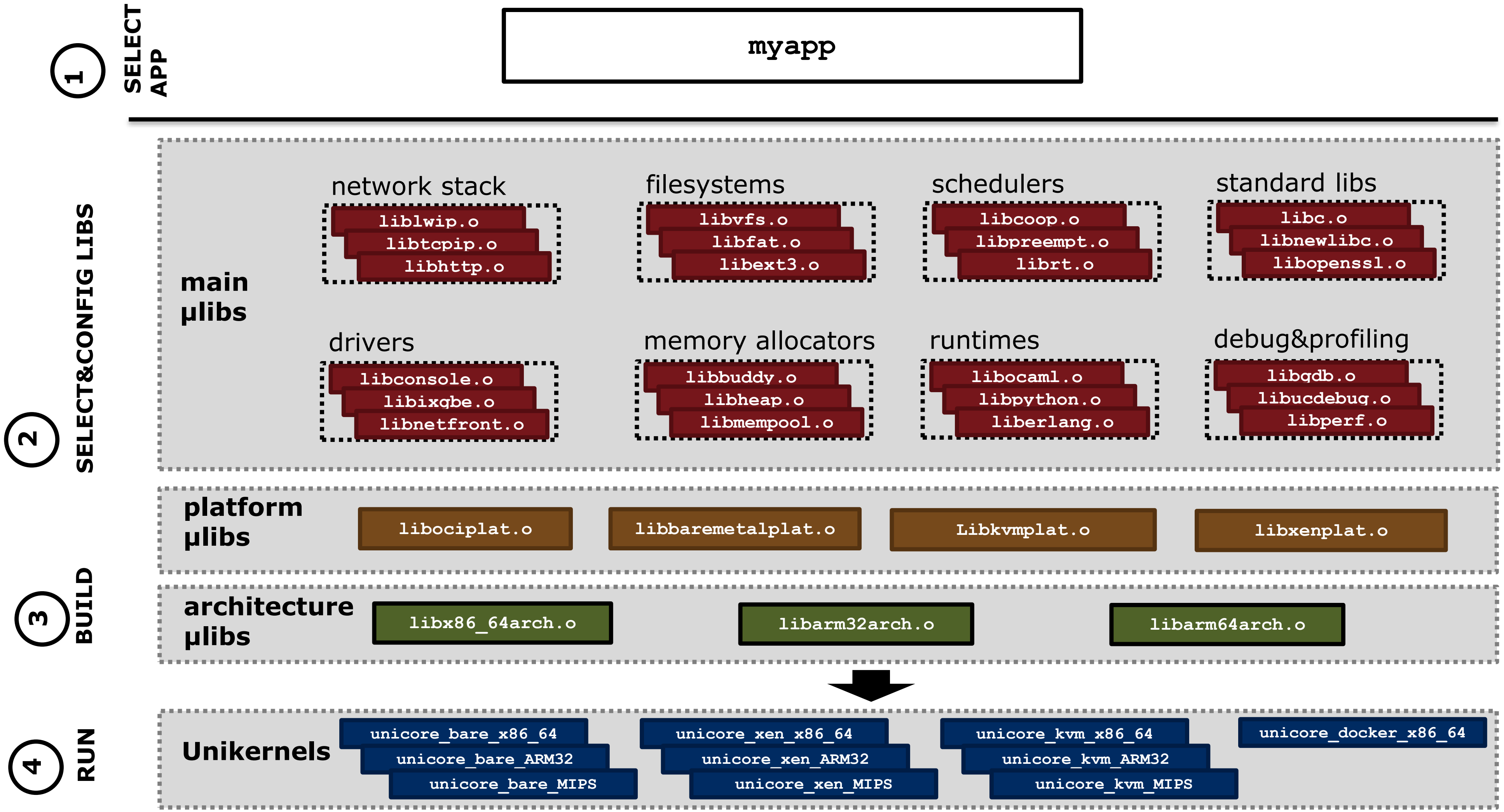
Concept: **“Everything is a library”**

- Decomposed OS functionality

Two components:

- Library Pool
- Build Tool

Unikraft Overview – Everything as a Microlibrary



1 SELECT APP

myapp

2 SELECT&CONFIG LIBS

main µlibs

network stack: liblwip.o, libtcpip.o, libhttp.o

filesystems: libvfs.o, libfat.o, libext3.o

schedulers: libcoop.o, libpreempt.o, librt.o

standard libs: libc.o, libnewlibc.o, libopenssl.o

drivers: libconsole.o, libixgbe.o, libnetfront.o

memory allocators: libbuddy.o, libheap.o, libmempool.o

runtimes: libocaml.o, libpython.o, liberlang.o

debug&profiling: libgdb.o, libucdebug.o, libperf.o

platform µlibs: libociplat.o, libbaremetalplat.o, Libkvmplat.o, libxenplat.o

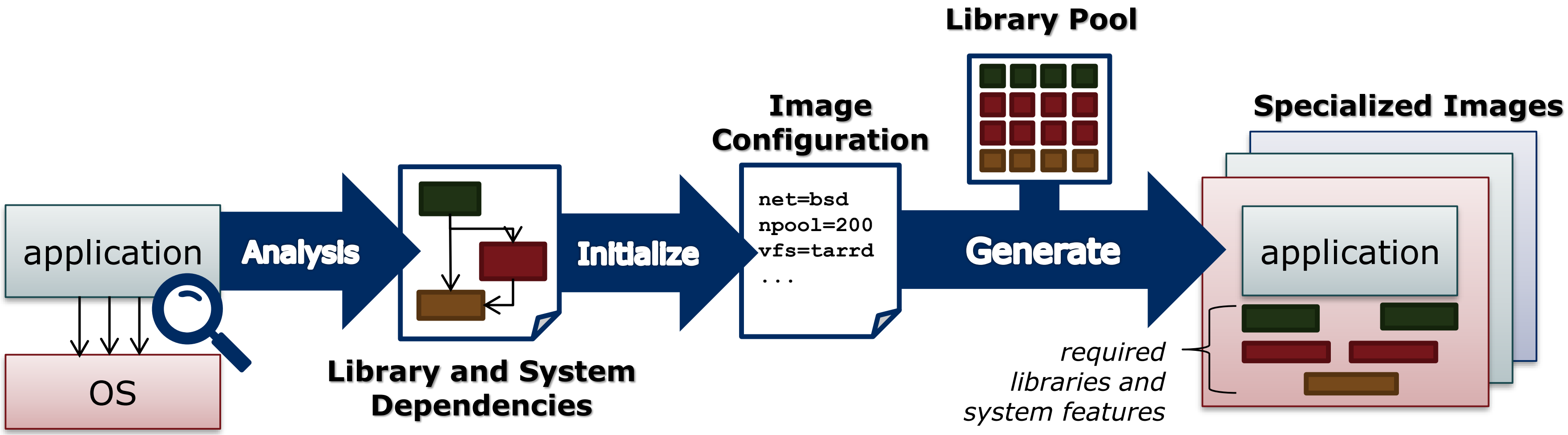
3 BUILD

architecture µlibs: libx86_64arch.o, libarm32arch.o, libarm64arch.o

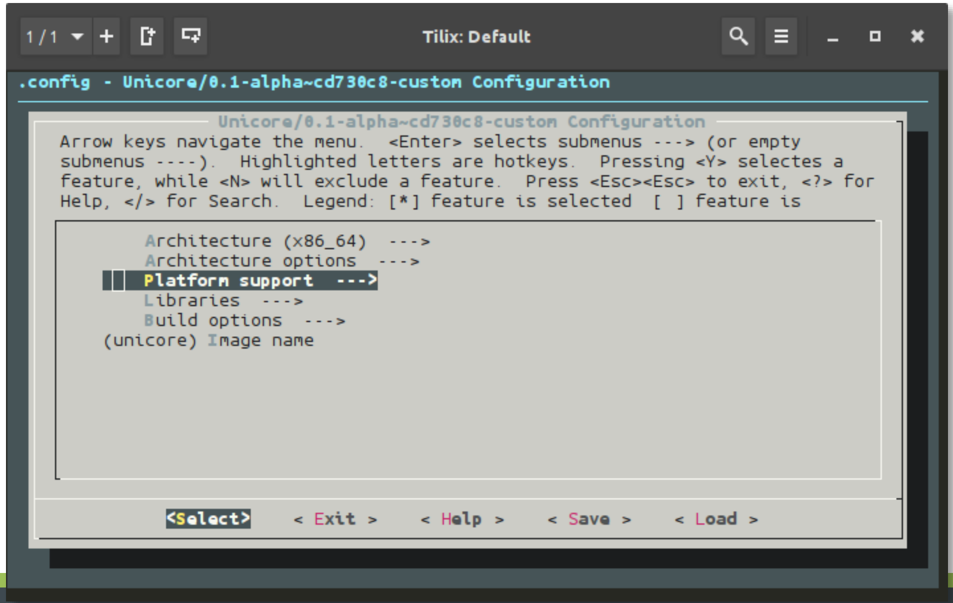
4 RUN

Unikernels: unicore_bare_x86_64, unicore_bare_ARM32, unicore_bare_MIPS, unicore_xen_x86_64, unicore_xen_ARM32, unicore_xen_MIPS, unicore_kvm_x86_64, unicore_kvm_ARM32, unicore_kvm_MIPS, unicore_docker_x86_64

Technology Approach 2: Build Tool, Automated Opts

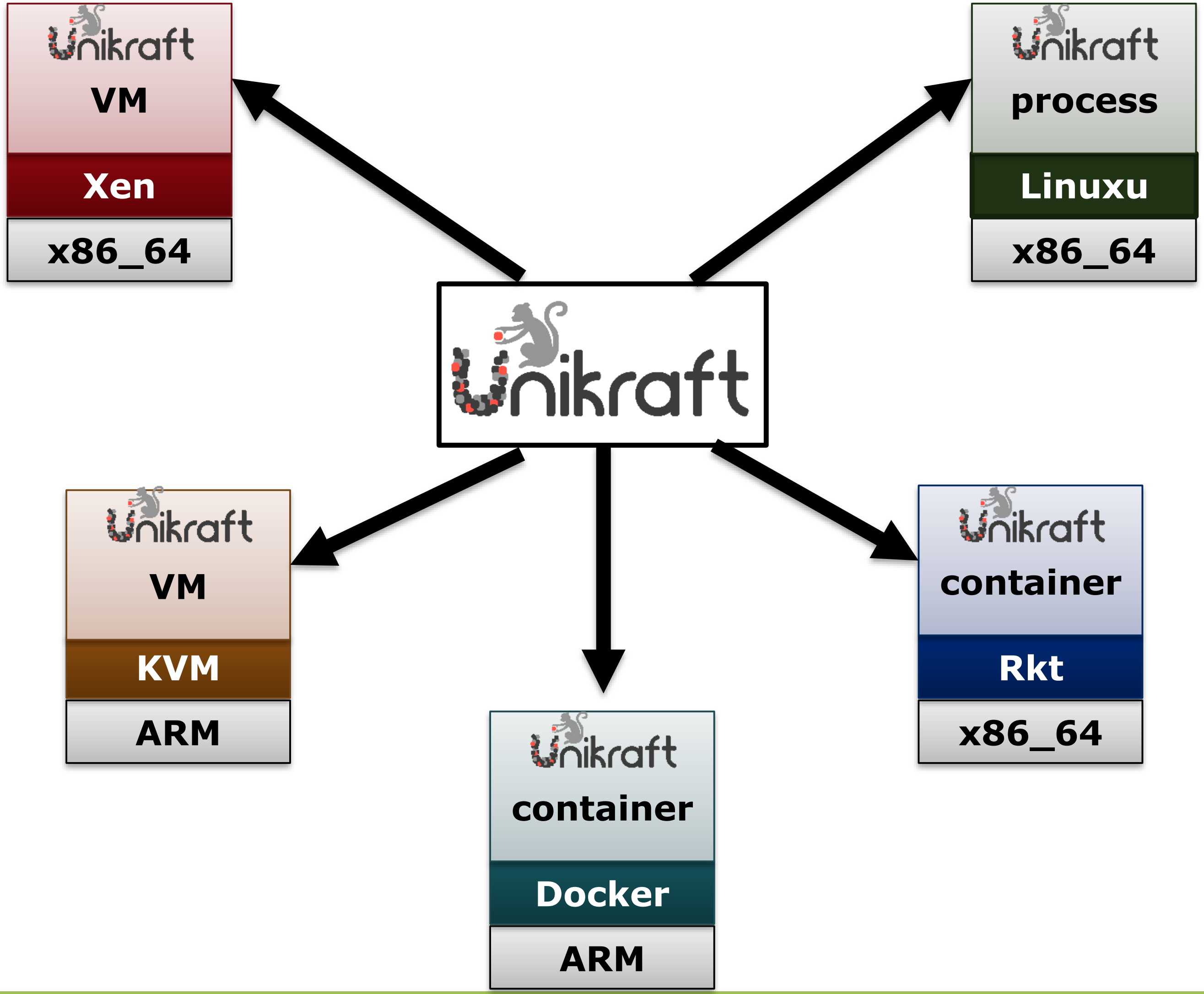


```
net=bsd
npool=200
vfs=tarrd
...
```



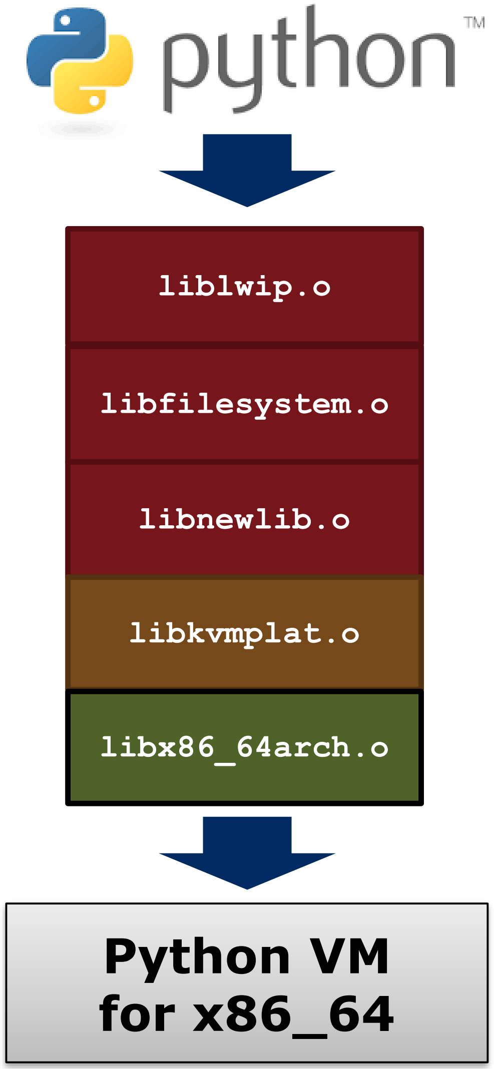
```
.config - Unicore/0.1-alpha~cd730c8-custom Configuration

        Unicore/0.1-alpha~cd730c8-custom Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> selects a
  feature, while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
  Help, </> for Search.  Legend: [*] feature is selected  [ ] feature is

          Architecture (x86_64)  --->
          Architecture options  --->
          Platform support  --->
          Libraries  --->
          Build options  --->
      (unicore) Image name

          <Select>    < Exit >    < Help >    < Save >    < Load >
```

One tool, Lots of Platforms



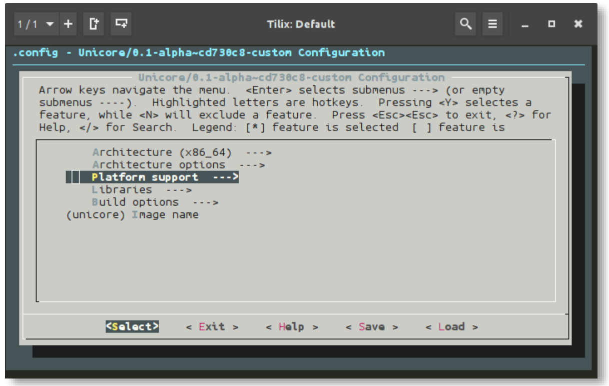
Unikraft in Action: Python

Micropython KVM VM: 731KB!



KVM

x86



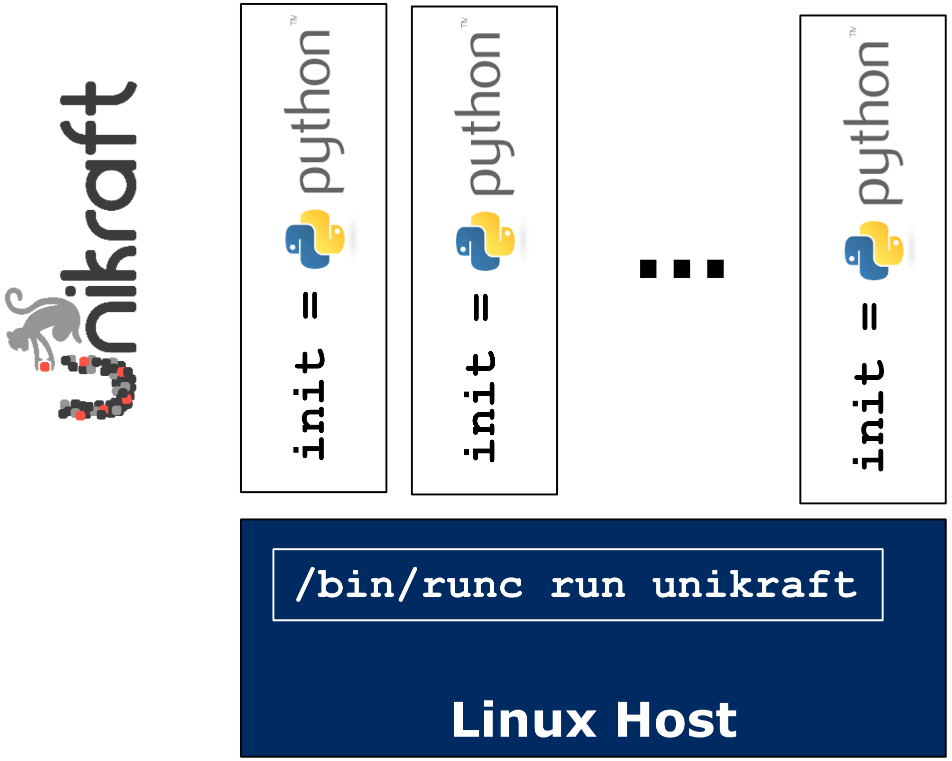
```
.config - Unicore/0.1-alpha~cd730c8-custom Configuration
 ─────────── Unicore/0.1-alpha~cd730c8-custom Configuration ───────────
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty
  submenus ----).  Highlighted letters are hotkeys.  Pressing <Y> selectes a
  feature, while <N> will exclude a feature.  Press <Esc><Esc> to exit, <?> for
  Help, </> for Search.  Legend: [*] feature is selected  [ ] feature is

          Architecture (x86_64)  --->
          Architecture options  --->
          Platform support  --->
          Libraries  --->
          Build options  --->
      (unicore) Image name

         <Select>    < Exit >    < Help >    < Save >    < Load >
```

The Unikraft Container

OCI-compatible Unikraft target produces

- **Root-file system**: contains the application and necessary libs
- **Container config file**: config.json (networking, application command line, etc)
- **OCI-compatible runtime** (runc): creates container environment based on config.json

Micropython OCI container: 576KB!



Key Asset: The Team

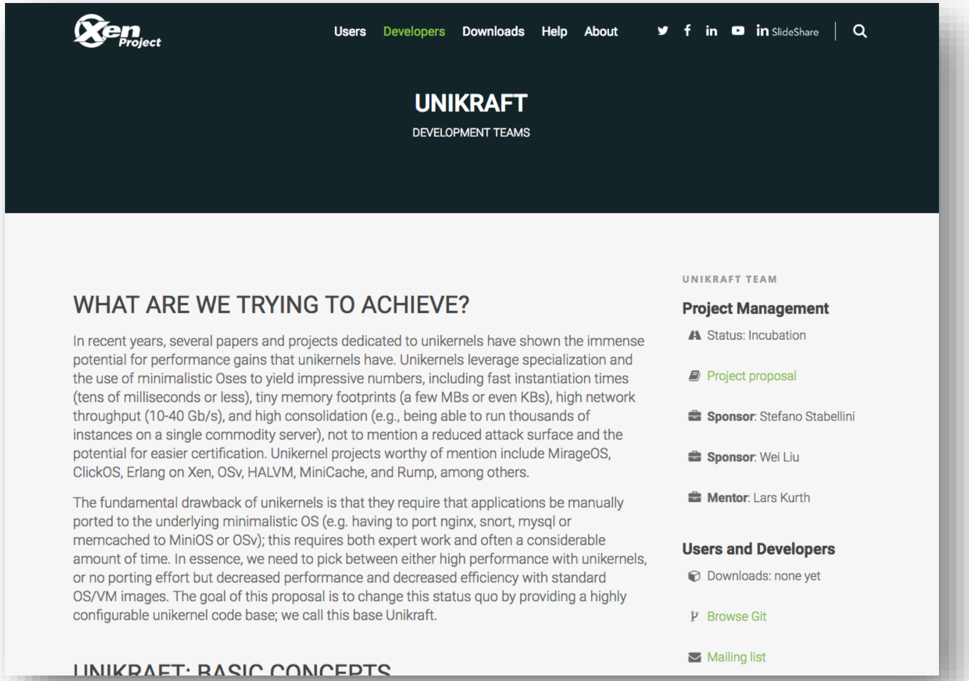
A Linux Foundation project since December 2017

Core members

- NLE core team: ~5 PY
- NLE students: ~ 1-2 PY
- U. of Bucharest: ~ 15 PY
- U. of Liege ~ 2 PY
- NTI: ~1 PY

Stakeholders/contributors

- ARM: ~5 PY
- Accelleran (sw RAN) ~ 1 PY
- Ekinops (NFV) ~ 2 PY
- Nextworks (Smart home/IoT) ~ 1 PY
- Orange Romania (NFV) ~ 1 PY
- CSUC (serverless) ~ 1 PY
- EPFL (Smart contracts) ~ 2 PY
- U. of Amsterdam ~ 1 PY



UNIKRAFT

DEVELOPMENT TEAMS

WHAT ARE WE TRYING TO ACHIEVE?

In recent years, several papers and projects dedicated to unikernels have shown the immense potential for performance gains that unikernels have. Unikernels leverage specialization and the use of minimalistic Oses to yield impressive numbers, including fast instantiation times (tens of milliseconds or less), tiny memory footprints (a few MBs or even KBs), high network throughput (10-40 Gb/s), and high consolidation (e.g., being able to run thousands of instances on a single commodity server), not to mention a reduced attack surface and the potential for easier certification. Unikernel projects worthy of mention include MirageOS, ClickOS, Erlang on Xen, OSv, HALVM, MiniCache, and Rump, among others.

The fundamental drawback of unikernels is that they require that applications be manually ported to the underlying minimalistic OS (e.g. having to port nginx, snort, mysql or memcached to MiniOS or OSv); this requires both expert work and often a considerable amount of time. In essence, we need to pick between either high performance with unikernels, or no porting effort but decreased performance and decreased efficiency with standard OS/VM images. The goal of this proposal is to change this status quo by providing a highly configurable unikernel code base; we call this base Unikraft.

UNIKRAFT: BASIC CONCEPTS

UNIKRAFT TEAM

Project Management

Status: Incubation

Project proposal

Sponsor: Stefano Stabellini

Sponsor: Wei Liu

Mentor: Lars Kurth

Users and Developers

Downloads: none yet

Browse Git

Mailing list

Supported Functionalities

webinar

LANGUAGES

- C
- C++
- Python
- Ruby (ongoing)
- Javascript/V8 (ongoing)
- Rust (ongoing)
- Go (ongoing)
- Java/OpenJDK (ongoing)
- Micropython (ongoing)
- Web assembly (WASM)

CPU ARCHITECTURES

- X86_64
- ARM32
- ARM64 (ongoing)

PLATFORMS

- Xen
- KVM
- OCI containers
- Solo5
- Firecracker (boot only)
- Linux user-space

CORE FUNCTIONALITY

- Memory allocator
- Co-operative scheduler
- Pre-emptive scheduler (ongoing)

NETWORKING

- API for high-speed NW
- Virtio driver
- Tap driver
- Network stack (lwip)

FS/BLOCK

- VFS API and layer
- API and driver for block I/O
- 9PFS support
- In-RAM filesystem

Supported Functionalities (2)

STANDARD LIBRARIES

- newlib
- musl (ongoing)
- intel-intrinsics
- libunwind
- libuuid
- pthread-embedded
- compiler-rt
- eigen
- fp16
- fxdiv
- pthreadpool
- Mkl-dnn (ongoing)
- Qnnpack (ongoing)
- libuv (ongoing)
- zlib (ongoing)
- openssl (ongoing)
- libaxtls (ongoing)

FRAMEWORKS

- Intel DPDK (ongoing)
- PyTorch (ongoing)
- Node.js (ongoing)
- Sol (ongoing)

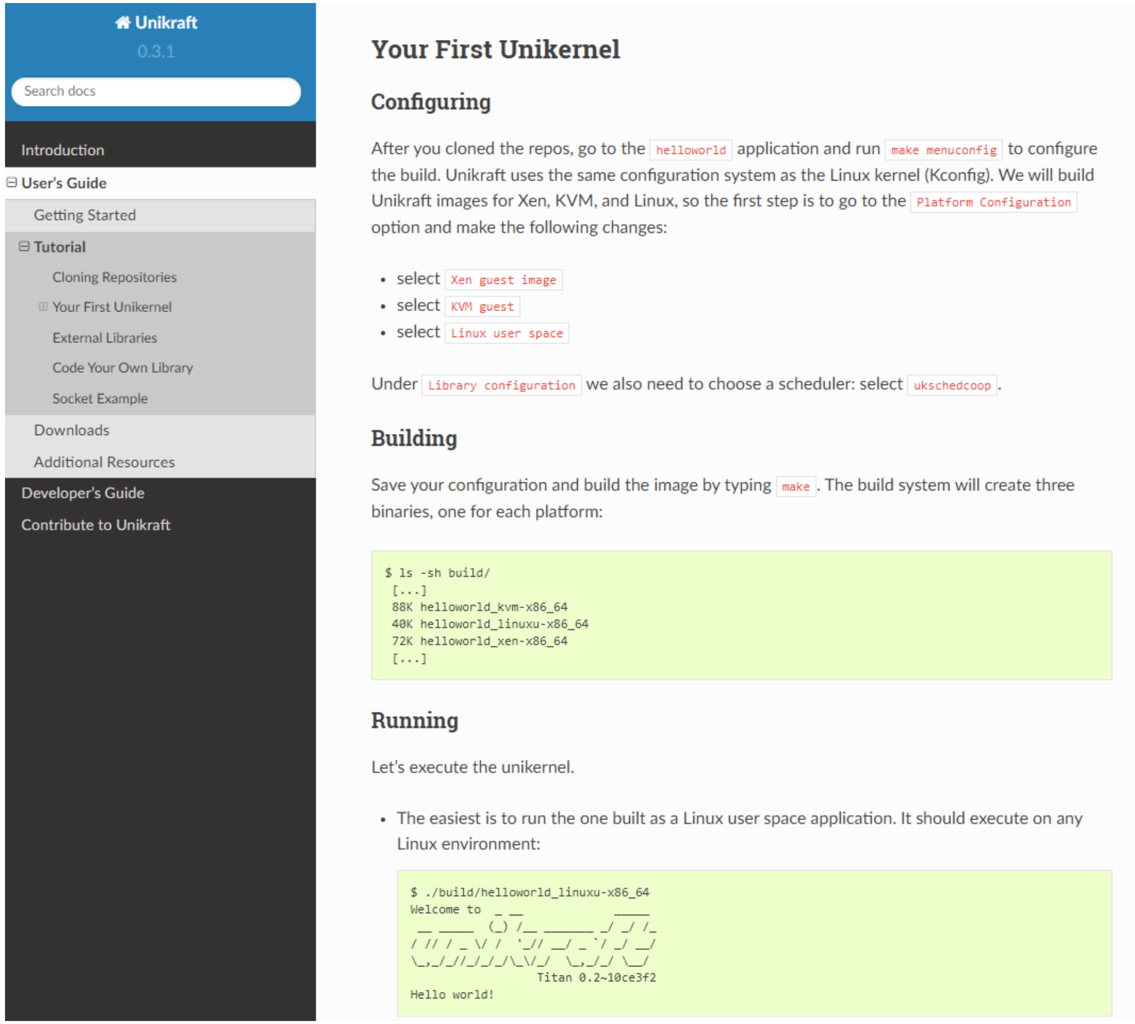
APPLICATIONS

- Click modular router
- nginx (ongoing)
- SQLite (ongoing)
- Redis (ongoing)
- memcached (ongoing)
- lighttpd (ongoing)

CLOUD PLATFORMS

- Amazon AWS
- Google Cloud Platform
- Digital Ocean (ongoing)

Build your first unikernel with Unikraft



Your First Unikernel

Configuring

After you cloned the repos, go to the `helloworld` application and run `make menuconfig` to configure the build. Unikraft uses the same configuration system as the Linux kernel (Kconfig). We will build Unikraft images for Xen, KVM, and Linux, so the first step is to go to the `Platform Configuration` option and make the following changes:

- select `Xen guest image`
- select `KVM guest`
- select `Linux user space`

Under `Library configuration` we also need to choose a scheduler: select `ukschedcoop`.

Building

Save your configuration and build the image by typing `make`. The build system will create three binaries, one for each platform:

```
$ ls -sh build/
 [...]
 88K helloworld_kvm-x86_64
 40K helloworld_linuxu-x86_64
 72K helloworld_xen-x86_64
 [...]
```

Running

Let's execute the unikernel.

- The easiest is to run the one built as a Linux user space application. It should execute on any Linux environment:

```
$ ./build/helloworld_linuxu-x86_64
Welcome to  _ __             _____
 __ _____  (_) /__ _______ _/ _/ /_
/ // / _ \/ /  '_// __/ _ `/ _/ __/
\_,_/_//_/_/_/\_\/_/  \_,_/_/ \__/
                  Titan 0.2~10ce3f2
Hello world!
```

Follow the Unikraft's Tutorial

- http://docs.unikraft.org/users-tutorial.html

Xen Project Website

- https://xenproject.org/developers/teams/unikraft/

Unikraft's Documentation

- http://docs.unikraft.org/

References and additional material

webinar

[1] NFV architecture
- https://www.slideshare.net/rmijumbi/nfv-tutorial-61544473

[2] NFV orchestration tool OSM
- https://osm.etsi.org/wikipub/index.php/OSM_Release_SIX

[3] 5GCity Orchestrator design, service programming, and ML models
- https://zenodo.org/record/2558306

[4] Unikraft‘s Tutorial
- http://docs.unikraft.org/users-tutorial.html

[5] 5GCity website
- https://www.5gcity.eu/

Q&A

SEE YOU SOON AT

5GCity

HACKATHON LUCCA

21-22 NOVEMBER 2019

Project Funded

5G PPP
PUBLIC-PRIVATE PARTNERSHIP
5g-ppp.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 761508

5GCity

www.5gcity.eu

@5gcity

HACK THE 5GCITY: A NEW ERA OF CITY SERVICES ON 5G

CO-LOCATED WITH
ETSI-OSM#8 HACKFEST
18-22 NOVEMBER 2019

5GCity Consortium

Accelleran, ADLINK Technology Inc., Ajuntament de Barcelona, betevé, cellnex, Città di Lucca, Comunicare Digitale, i2cat, inCITES Consulting S.A.R.L, ITALTEL, mOG, NEC, NEXTWORKS Engineering Forward, Rai, ubiwhere, University of BRISTOL Smart Internet Lab, Virtual Open Systems, WIND 3

Thank you!

www.5gcity.eu info@5gcity.com

@5GCity https://www.linkedin.com/groups/8607896

https://www.youtube.com/channel/UCqakofF2vjTdyEihpadfvLQ

5GCity is a project partially funded by the European Commission Horizon 2020 5G-PPP Programme under Grant Agreement number 761508