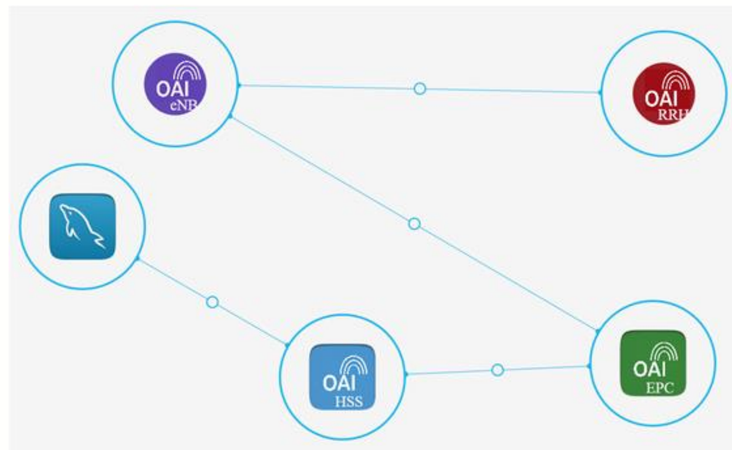
A leading institution at the heart of the digital society

EURECOM
Sophia Antipolis

MOSAIC-5G

OPEN AIR INTERFACE



Towards RAN Slicing in 5G

Navid Nikaein

Communication System Department, EURECOM

Coherent

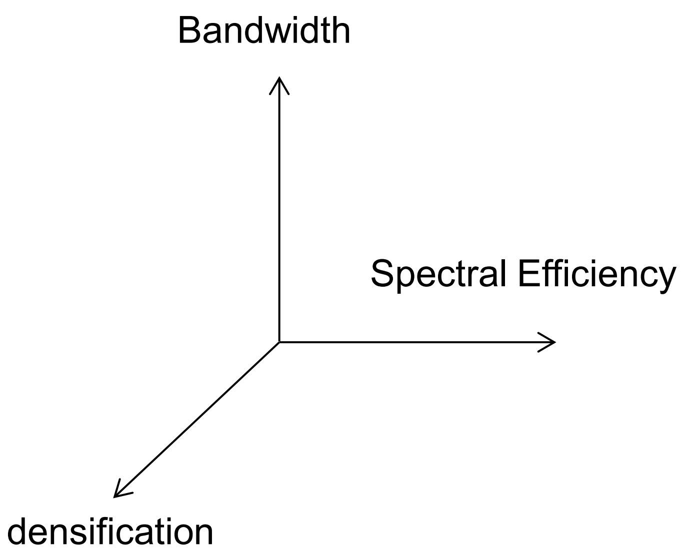
European Commission

5GOAI Workshop, 25 November, 2016

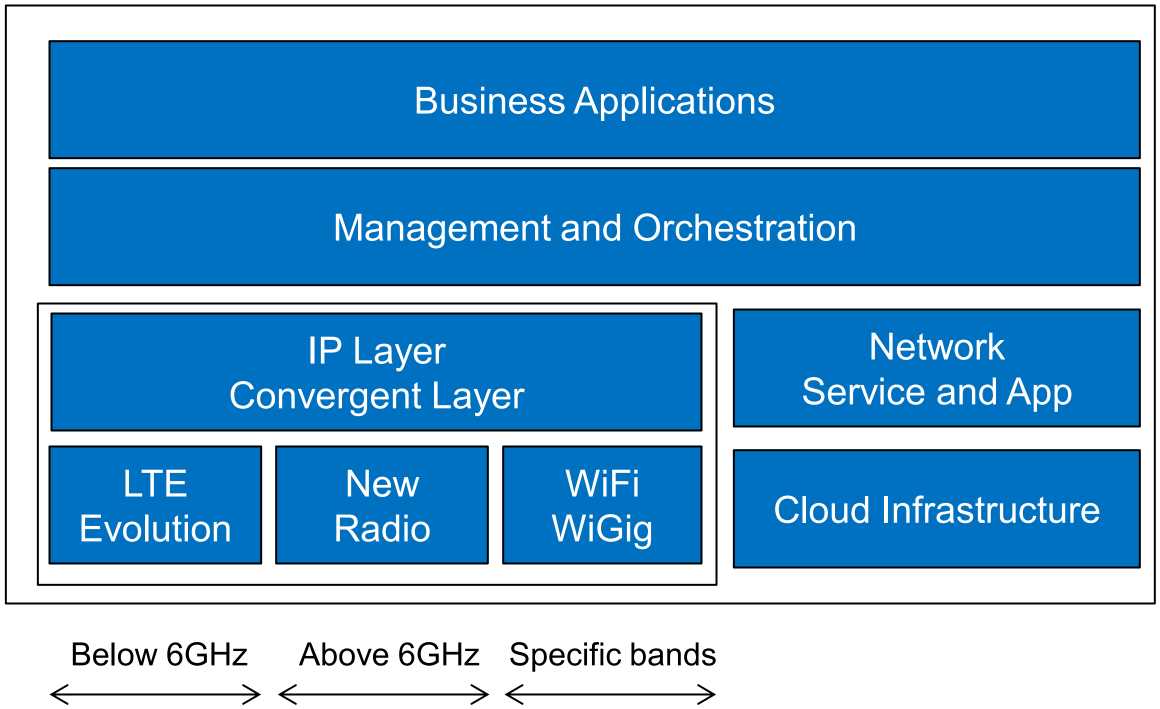
Outline

- 5G will be a paradigm shift
- What is software-defined 5G network
- Network Slicing Architecture
- NFV-SDN-MEC interplay
- RAN Slicing USE case
- Conclusion and Reflection

5G will be a paradigm shift



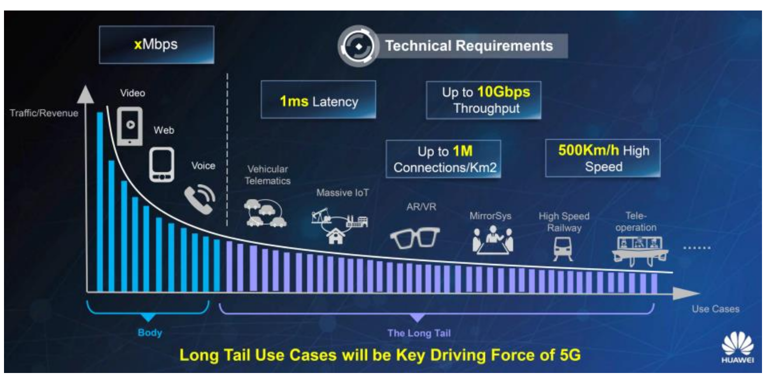
Overall 5G Solution

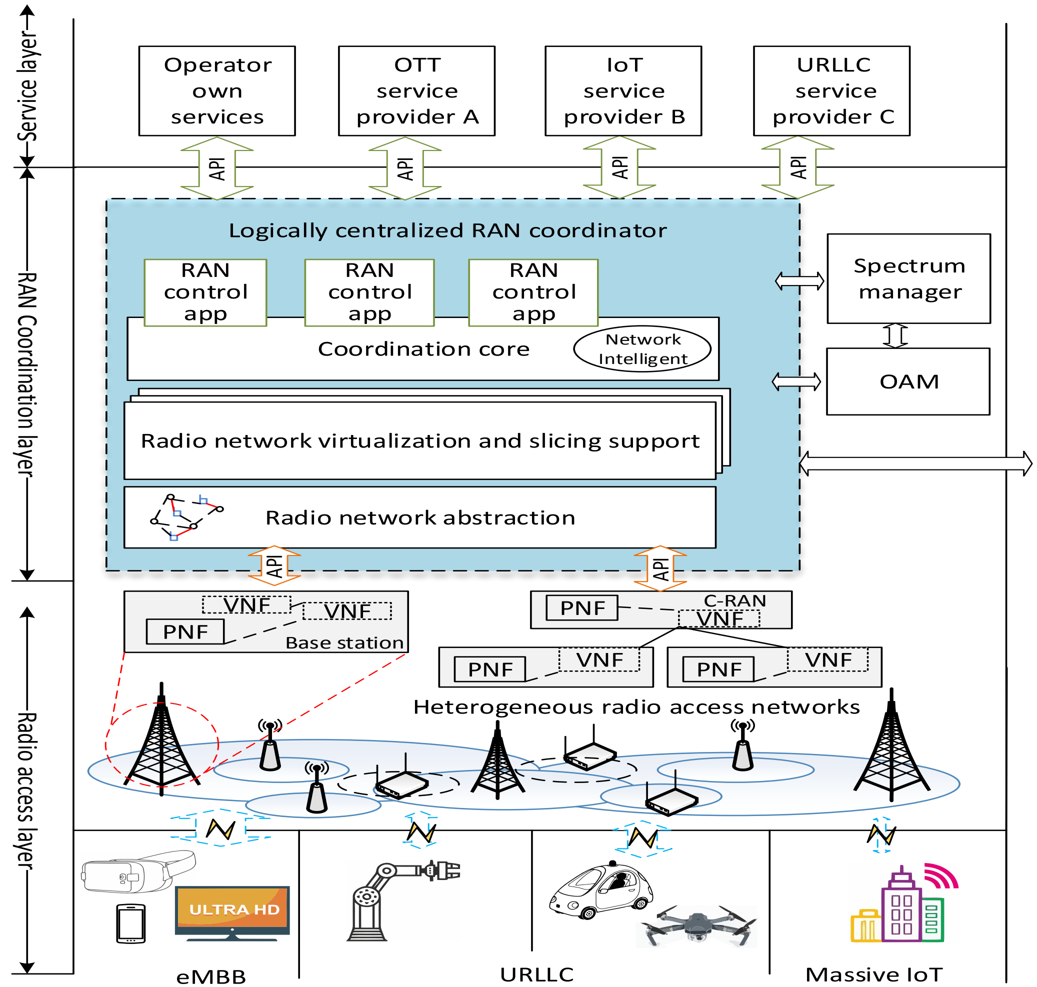


- **5G is not just a new radio/spectrum, but also a new architecture and business helper**

5G will be a paradigm shift
Long tail use cases to support verticals

- **Support of verticals will be key driving forces of 5G business to empower value creation**
- **Key capabilities**
  - eMBB, URLLC, mMTC





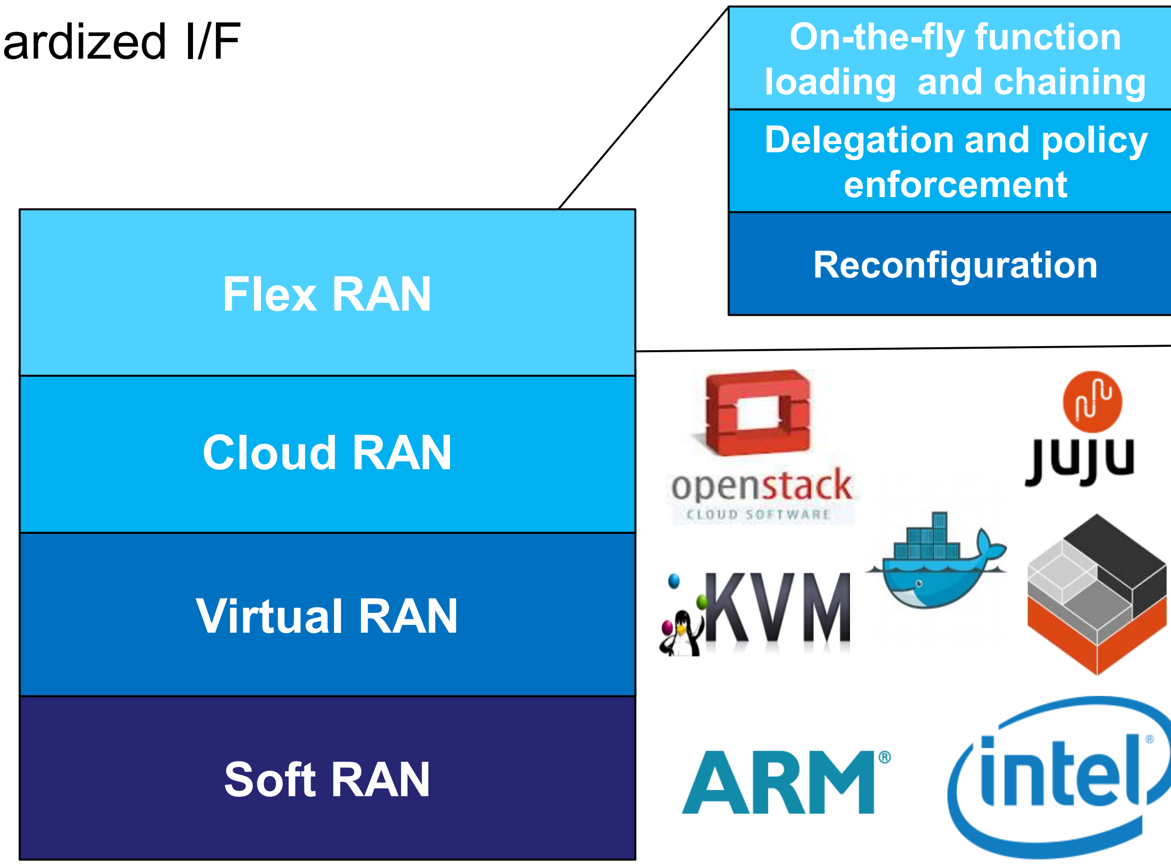
Source: Coherent Project

Economics of mobile are changing

- **Softwarization and Commoditization**
  - Software implementation of network functions on top of GPP with no or little dependency on a dedicated hardware
    - Full GPP vs. accelerated vs. system-on-chip
  - Programmable RF
- **Virtualization and Cloudification**
  - Execution of network functions on top of virtualized computing, storage, and networking resources controlled by a cloud OS.
  - Multi-tenancy, share resources among multiple guests
- **Abstraction and programmability**
  - Unified control-plane framework
  - Logical resources and network graphs
  - Data models, APIs and standardized I/F
- **Cognitive network control**
  - Logics for programmability in RAN and CN
    - Data plane and control-plane

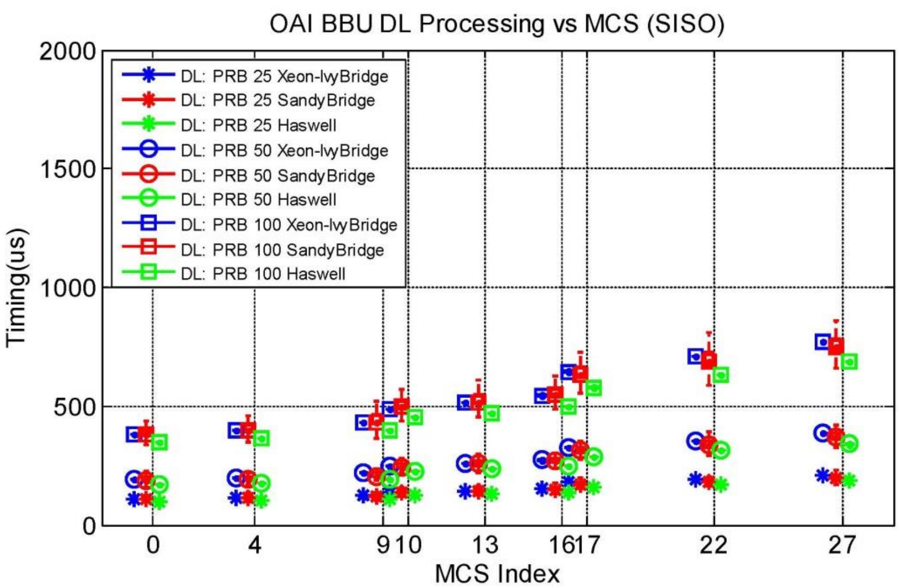
From SoftRAN to FlexRAN

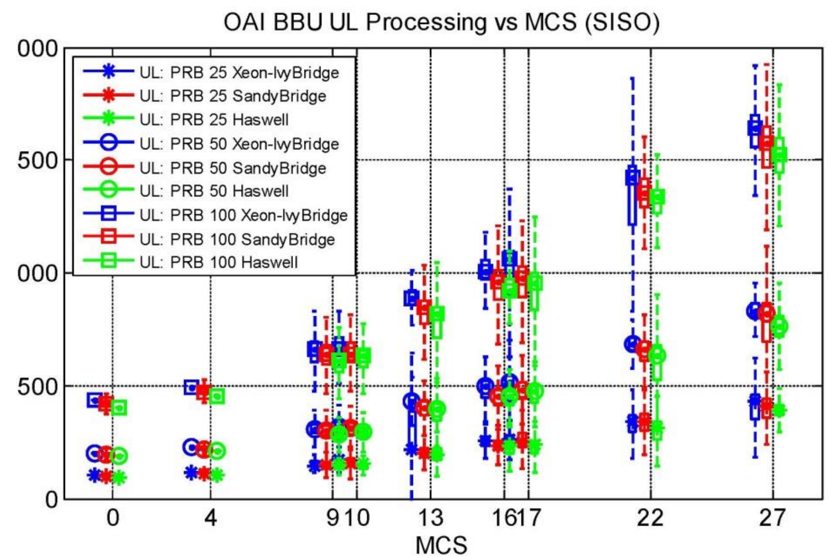
- **Emergence of rich ecosystem and opensource for telecom**
  - Cloud, NFV, SDN and MEC
  - Open APIs and standardized I/F



Softwarization and Commoditization

- **Leverage General Purpose Processors (x86)**
- **Today an eNB is approximately 1-2 x86 cores on Gen 3 Xeon silicon**
  - Perhaps more power efficient solutions from TI, Freescale or Qualcomm
  - But: lose commodity software environment and common HW platform to high-layer protocols and cloud





Softwarization and Commoditization Processing Budget

eNB Rx stats (1subframe)

- OFDM demod : 109.695927 us
- ULSCH demod: 198.603526 us
- ULSCH Decoding : 624.602407 us

➔ 931 us (<1 core)

eNB Tx stats (1 subframe)

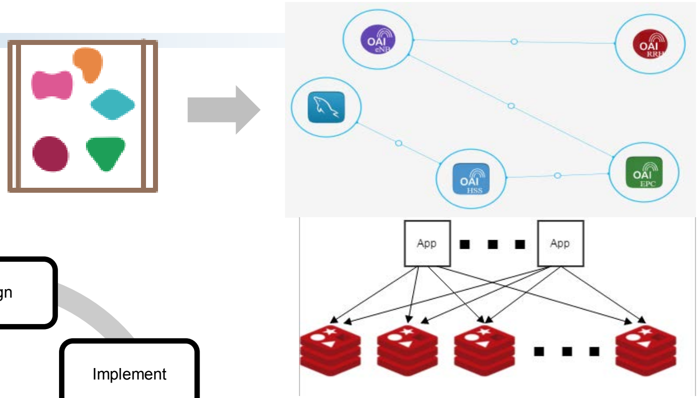
- OFDM mod : 108.308182 us
- DLSCH mod : 176.487999 us
- DLSCH scrambling : 123.744984 us
- DLSCH encoding : 323.395231 us

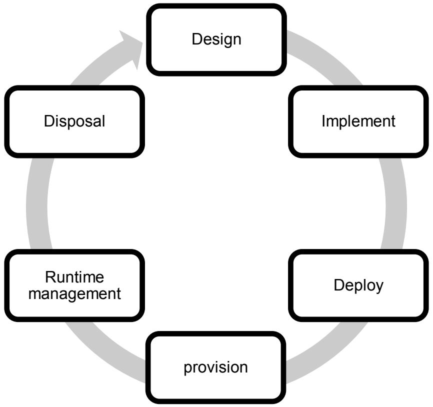
➔ 730 us (< 1core)

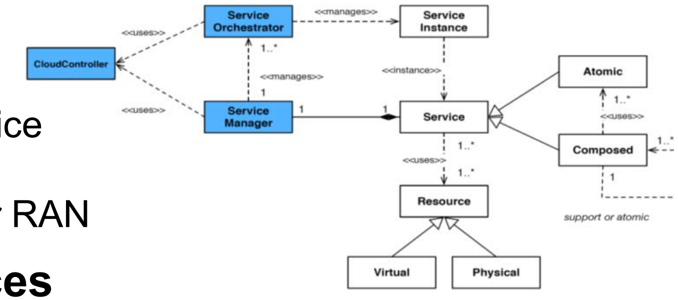
- **Efficient base band unit is challenging**
- **With AVX2 (256-bit SIMD), turbo decoding and FFT processing will be exactly twice as fast**
  - <1 core per eNB
  - .4 core per eNB without turbo en/decoder ← can this be exploited efficiently with HW acceleration? (Solution adopted in China Mobile CRAN project, offload of TC on Altera FPGA)
- **Configuration**
  - gcc 4.7.3, x86-64 (3 GHz Xeon E5-2690),
  - 20 MHz bandwidth (UL mcs16 – 16QAM, DL mcs 27 – 64QAM, transmission mode 1 - SISO)
  - 1000 frames, AWGN channel

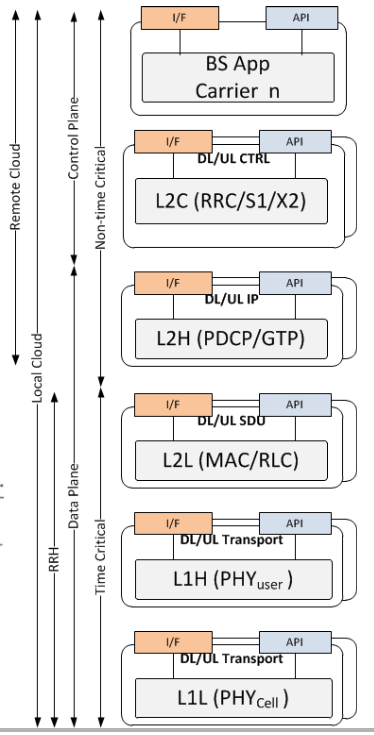
Virtualization and Cloudification

- **Mircoservice Architecture along with NFV**
  - Flexible Functional split
  - Move form monolitic to a composed and metered service
  - Stateless, composable, reusable
- **Scalability and metered service**
- **Reliability and resiliency**
  - Resource provisioning
  - Redundancy and stateless
  - Fast recovery
- **Multitenancy**
  - Share the resources
  - (spectrum, radio, and infrastructure)
- **Splitting, chaining, Placement**
  - Atomic, support, and composed service
  - Optimize the cost and performance
  - Supported Hardware, in particular for RAN
- **Realtime/lowlatency edge services**
  - Direct access to the radio information



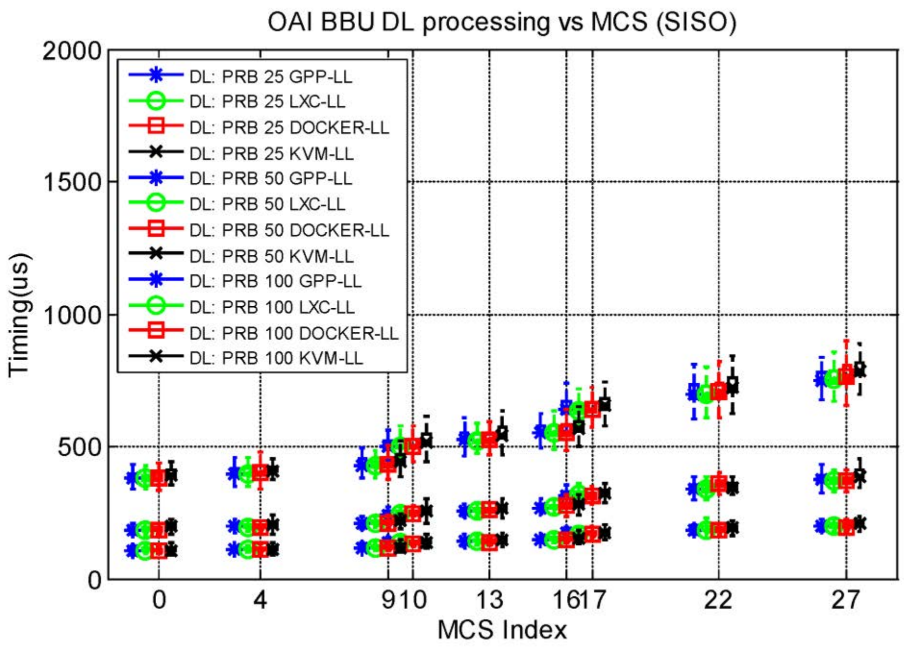


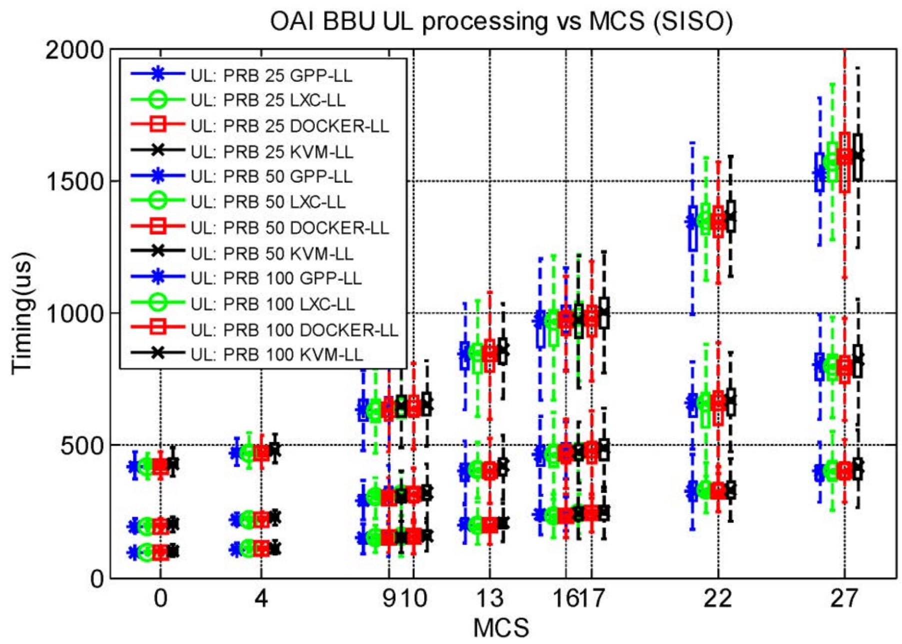




Virtualization and Cloudification

- **DL and UL BBU processing load for various MCS, PRB, and virtualization flavor**
  - Comparable BBU Processing time





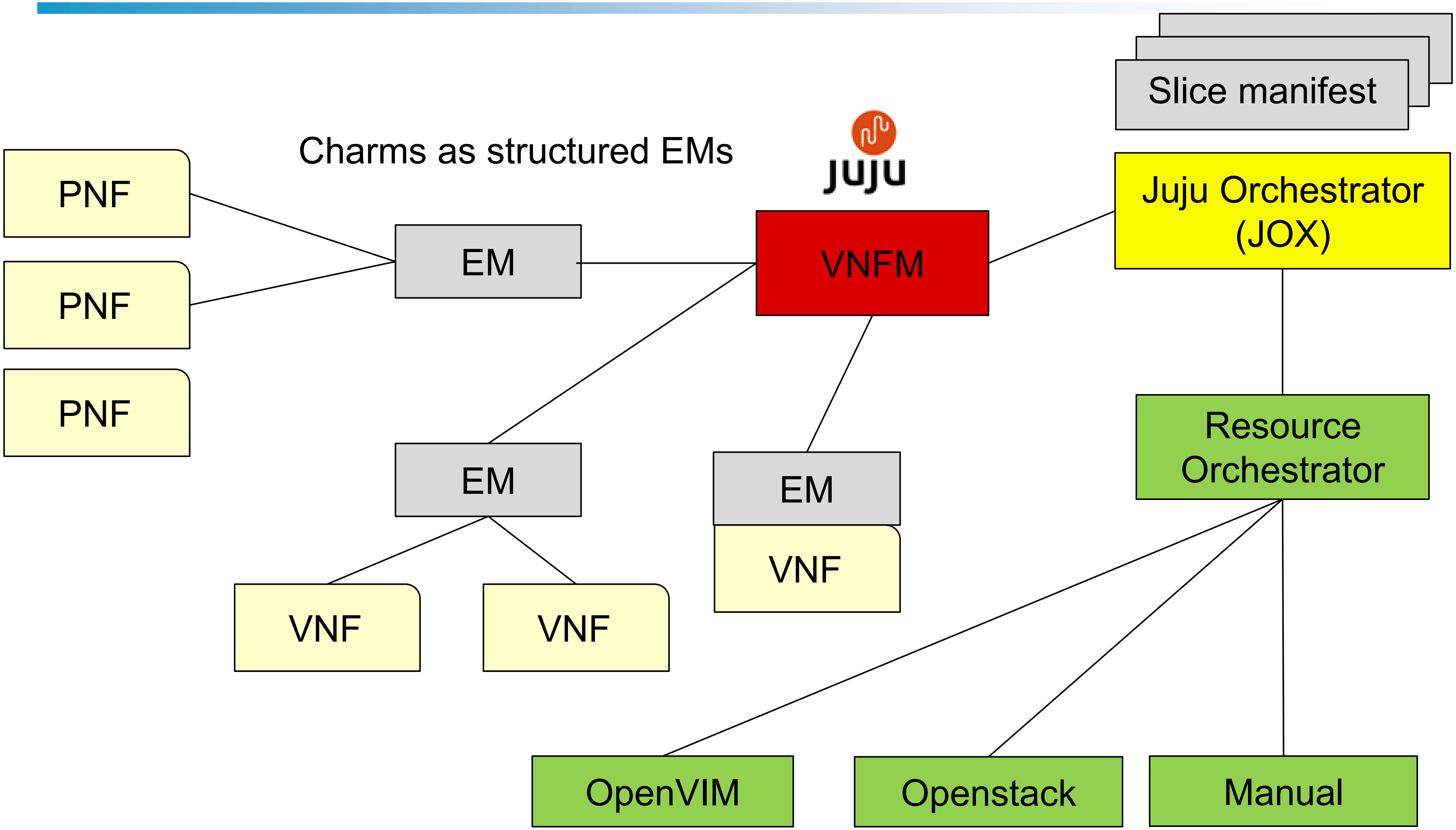
Virtualization and Cloudification
Service Management and Orchestration

- **Service level modeling**
  - design an abstract network slice for a particular use-case
  - Identify the data models and interfaces across the network functions
  - Standardize reference network slice templates
    - capex/opex considerations

- **Service layer encapsulates**
  - VNF image and descriptor
  - Configuration
  - Connection points
  - Two distinct lifecycles
    - Service
    - Relationships
  - Health and monitoring parameters
  - Resources and constraints
  - Upgrade

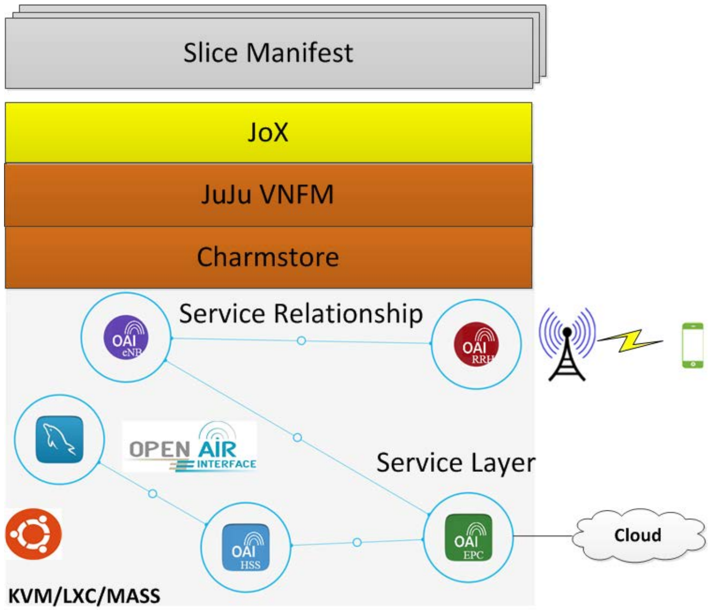
- **Service template defines**
  - Service descriptor
  - Input Parameters
  - Configuration primitives
  - Relationships/dependencies
  - Resources and constraints
  - Units (number of instances)
  - Machine (physical or virtual)
  - Domain

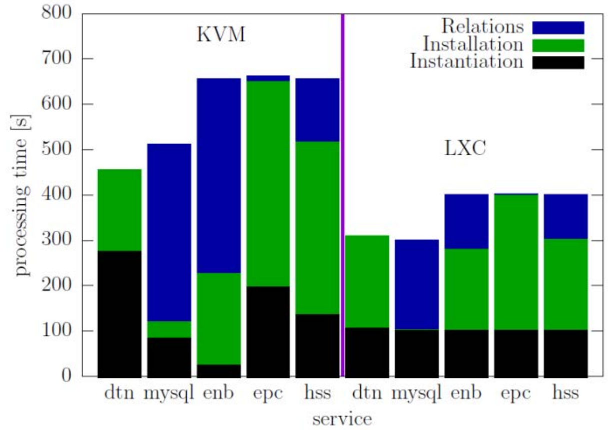
Virtualization and Cloudification NFV and OSM



Virtualization and Cloudification (JuJu/JoX) Architecture

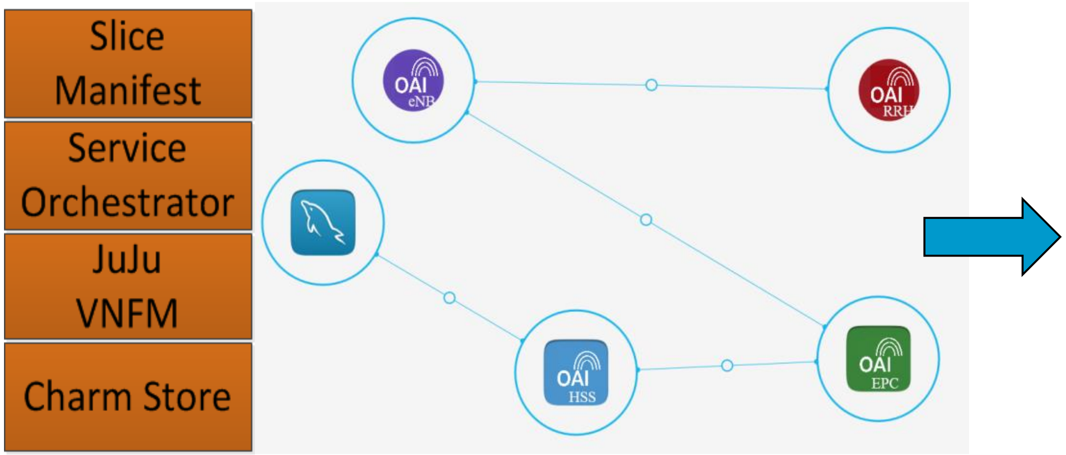
- **Slice manifest defining the service as a whole**
  - Service is a composition of VNFs spanning across a set of domains and machines
    - E.g.two units of this app with their respective configuration file
- **JoX orchestrates the E2E service lifecycle according to the slice manifest**
- **Juju manages the services over the infrastructure**
- **Charm acts a structured EM driven by juju**
  - Lifecycle
    - Install, update, and upgrade
  - configuration
  - Scale and elasticity
  - Integration
    - Relationship and interfaces, peers





Virtualization and Cloudification (JuJu)

Orchestration logic, Canonical Juju Example

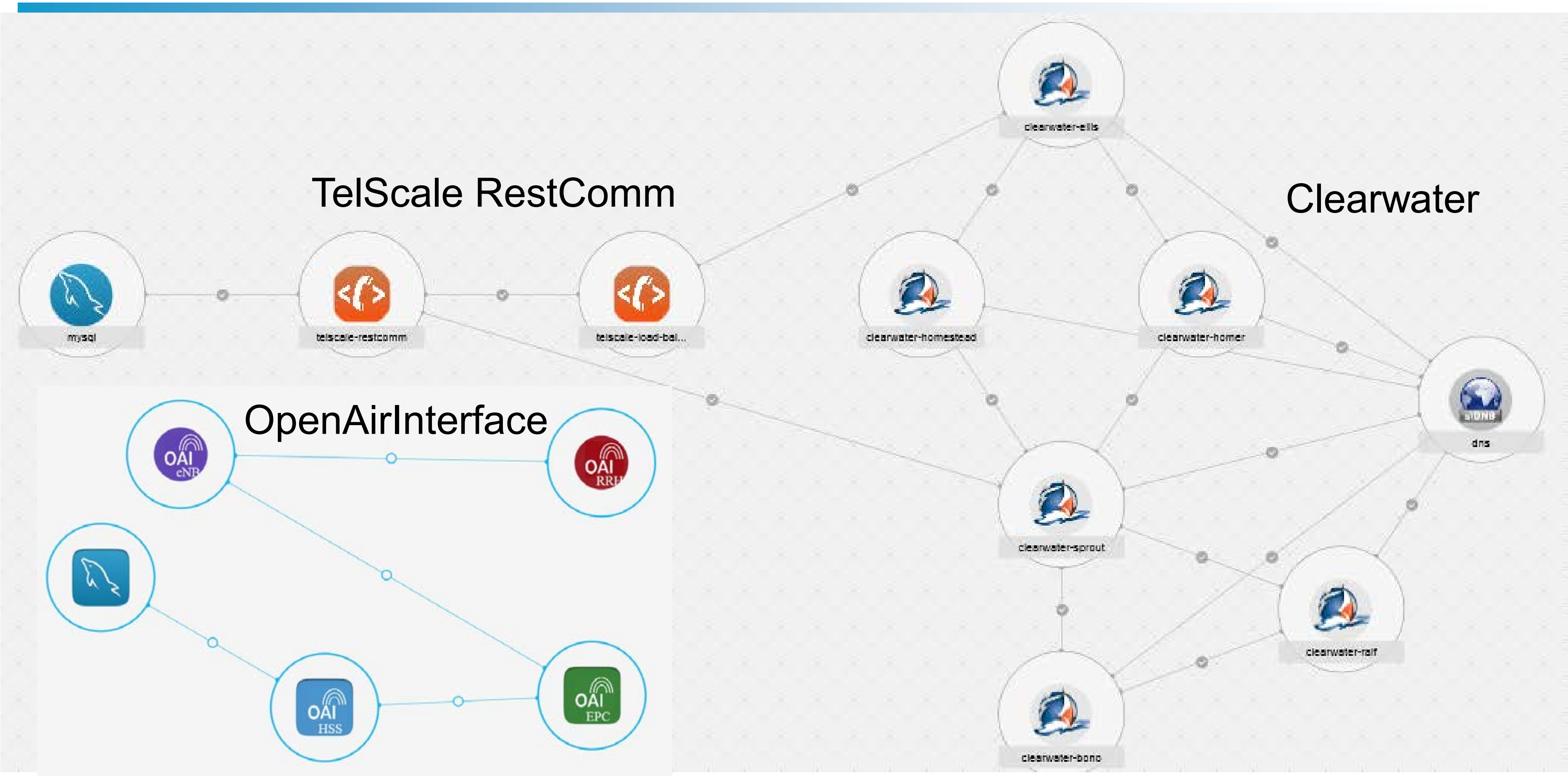


- **Template are built based on the slice manifest**
- **Orchestrator logic applied through a EM/charms able to change the service template definition on the fly**
  - Reliability and scalability
  - Single and multi-domain
- **Charms as structured element manager to drive the app lifecycle**
- **JUJU is a generic VNFM as well as store manager**
  - https://jujucharms.com/q/oai


```
series: trusty
services:
 "oai-enb":
  charm: "cs:trusty/oai-enb"
  num_units: 2
  options:
   N_RB_DL: 50
   downlink_frequency: 2680000000L
   eutra_band: 7
   rrh_active: "yes"
   uplink_frequency_offset: "-120000000"
  to:
   - "0:0"
 "oai-epc":
  charm: "cs:trusty/oai-epc"
  num_units: 1
  to:
   - “kvm:0:0"
relations:
 - - "oai-enb:epc"
  - "oai-epc:epc"
 - - "oai-hss:db"
  - "mysql:db"
 - - "oai-epc:hss"
  - "oai-hss:hss"
domain:
 “0”:
   provider:aws
   machines:
    "0":
      series: trusty
      constraints: "arch=amd64 cpu-cores=4
              mem=15951 root-disk=8192“
```

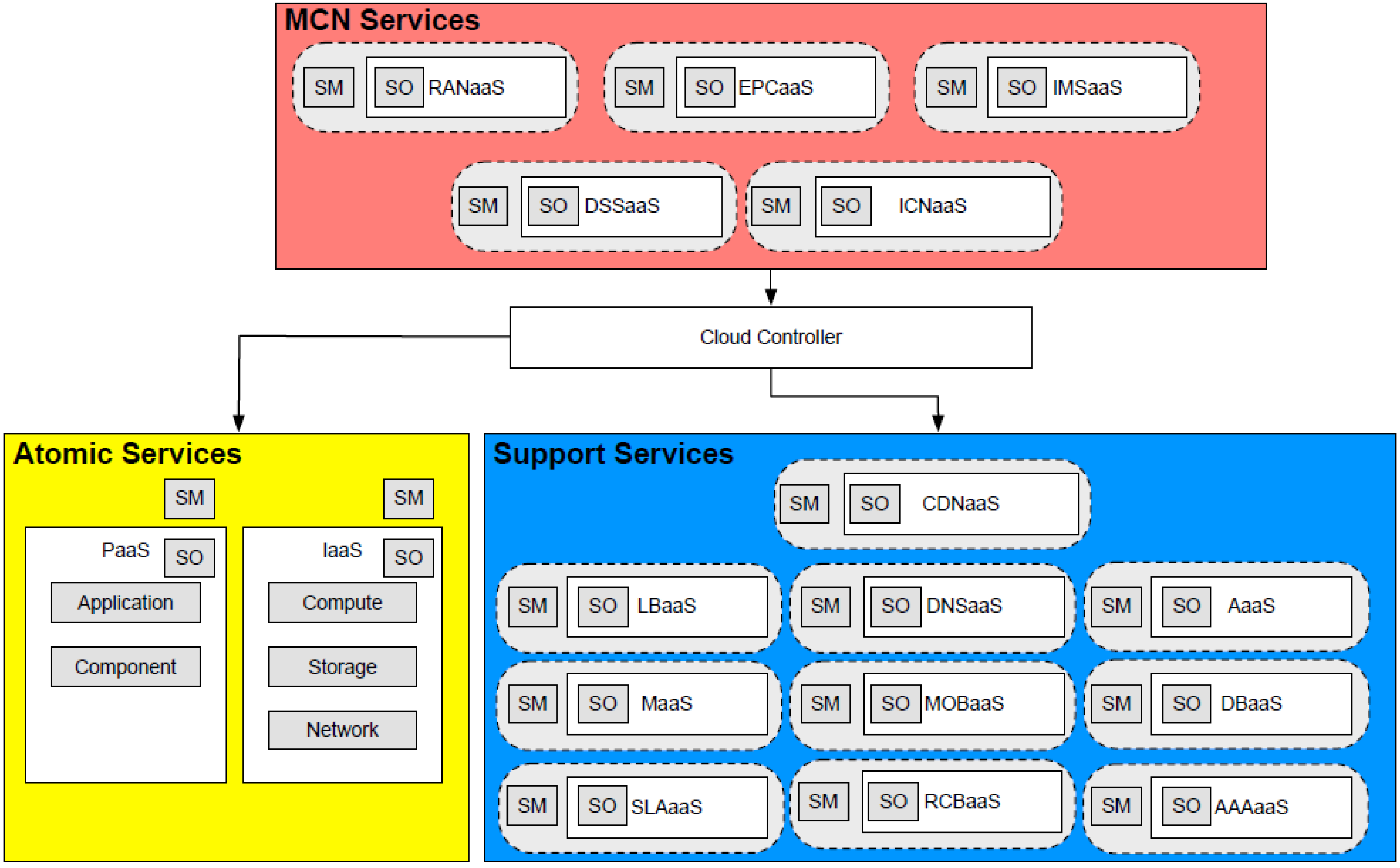
Virtualization and Cloudification (JuJu)

The need for flexible functional splitting, chaining, and placement



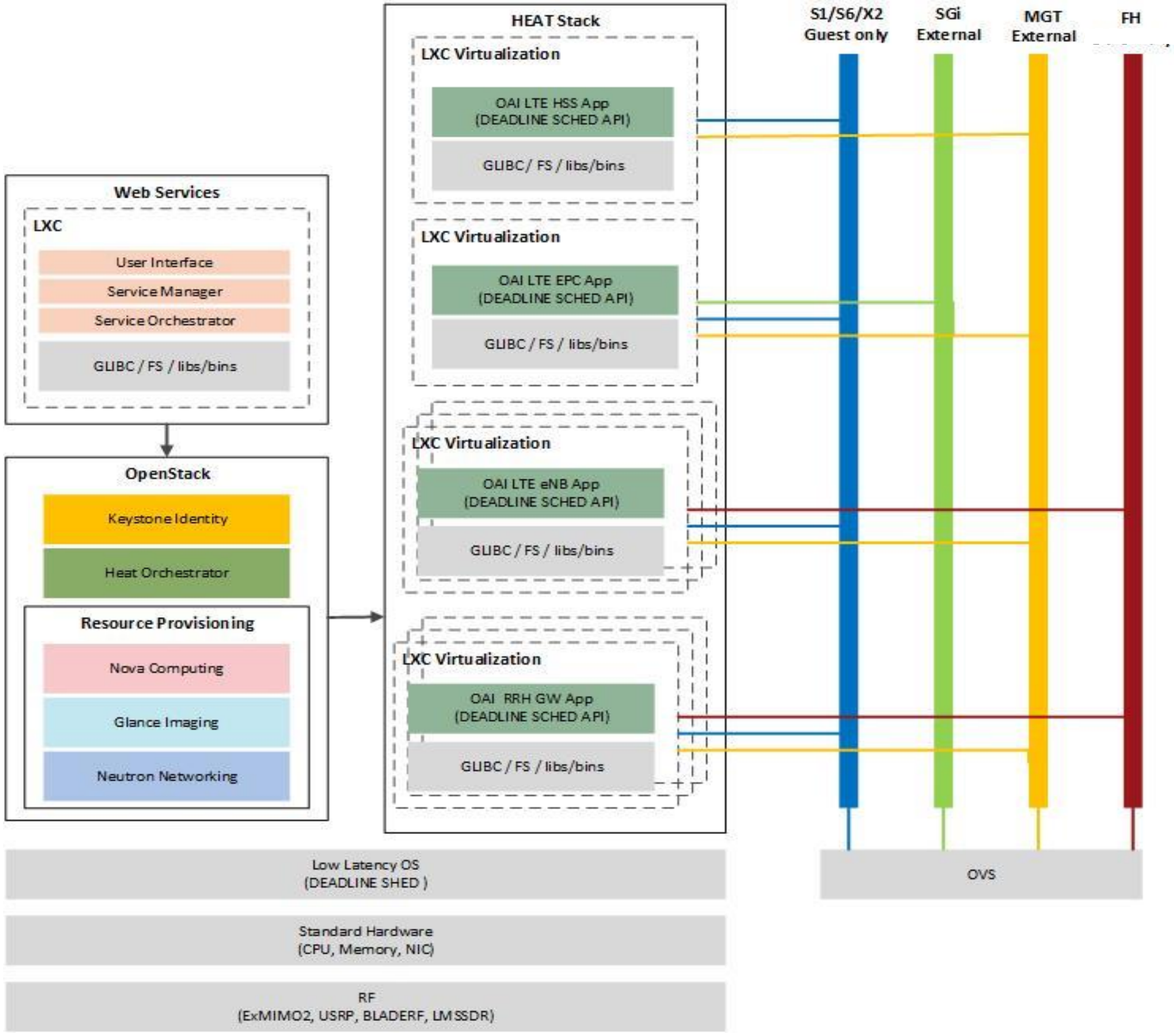
- **Rapidly build voice, video, WebRTC, USSD, SMS, fax and rich messaging applications over LTE**

Virtualization and Cloudification (OS,HEAT,OPS) Overall Service Chain for IMSaaS



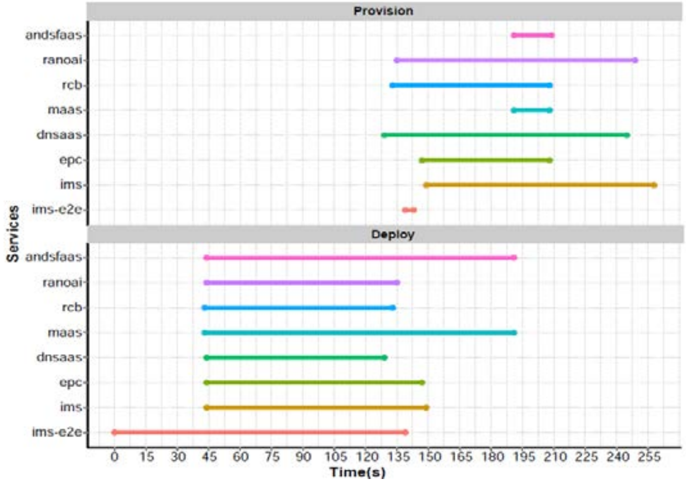
Virtualization and Cloudification (OS,HEAT,OPS) Overall Service Chain for LTEaaS

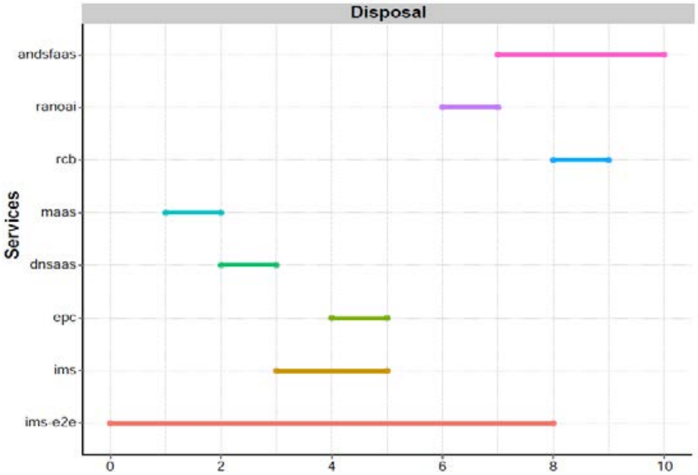
- Three components
  - web service
  - OpenStack
  - Heat stack
- Heat Template describes the virtual network deployment
  - Deployment Lifecycle
- Linux Container
- Open vSwitch
- Low latency kernel
- RF frontend HW



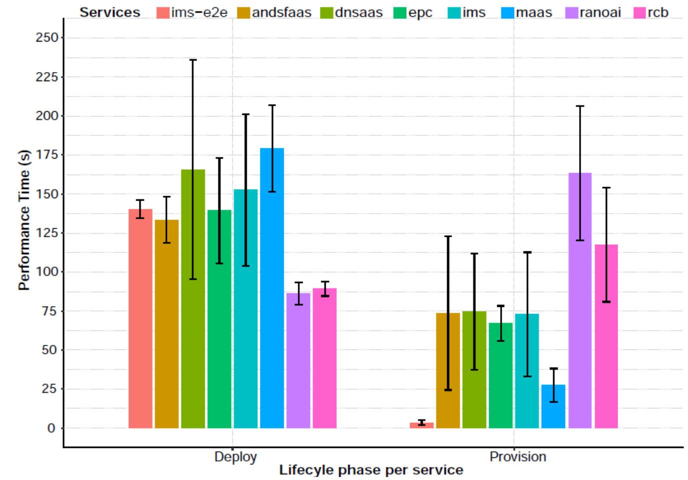
Virtualization and Cloudification (OS,HEAT,OPS) LTEaaS Perforamnce

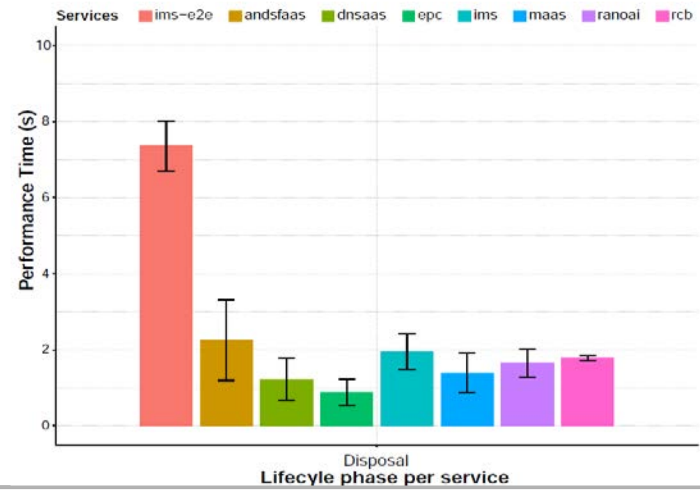
Lifecycle Sequence





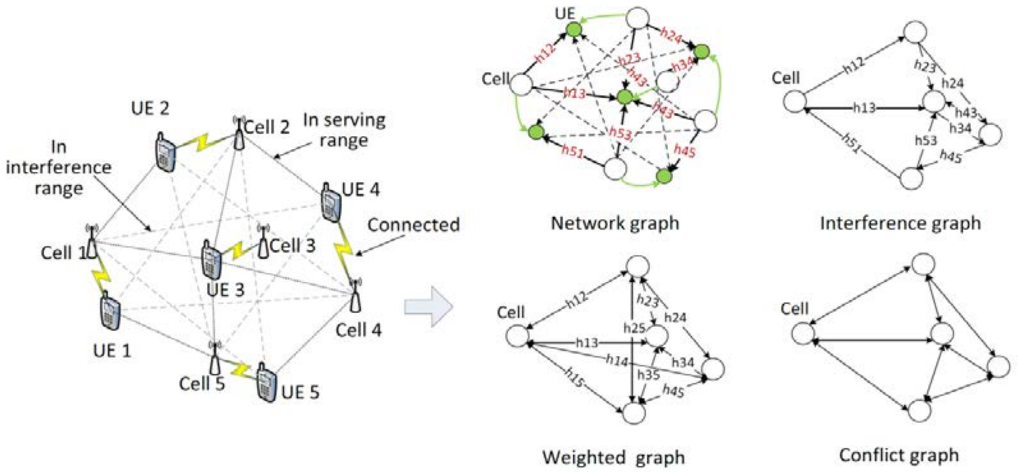
Lifecycle Time





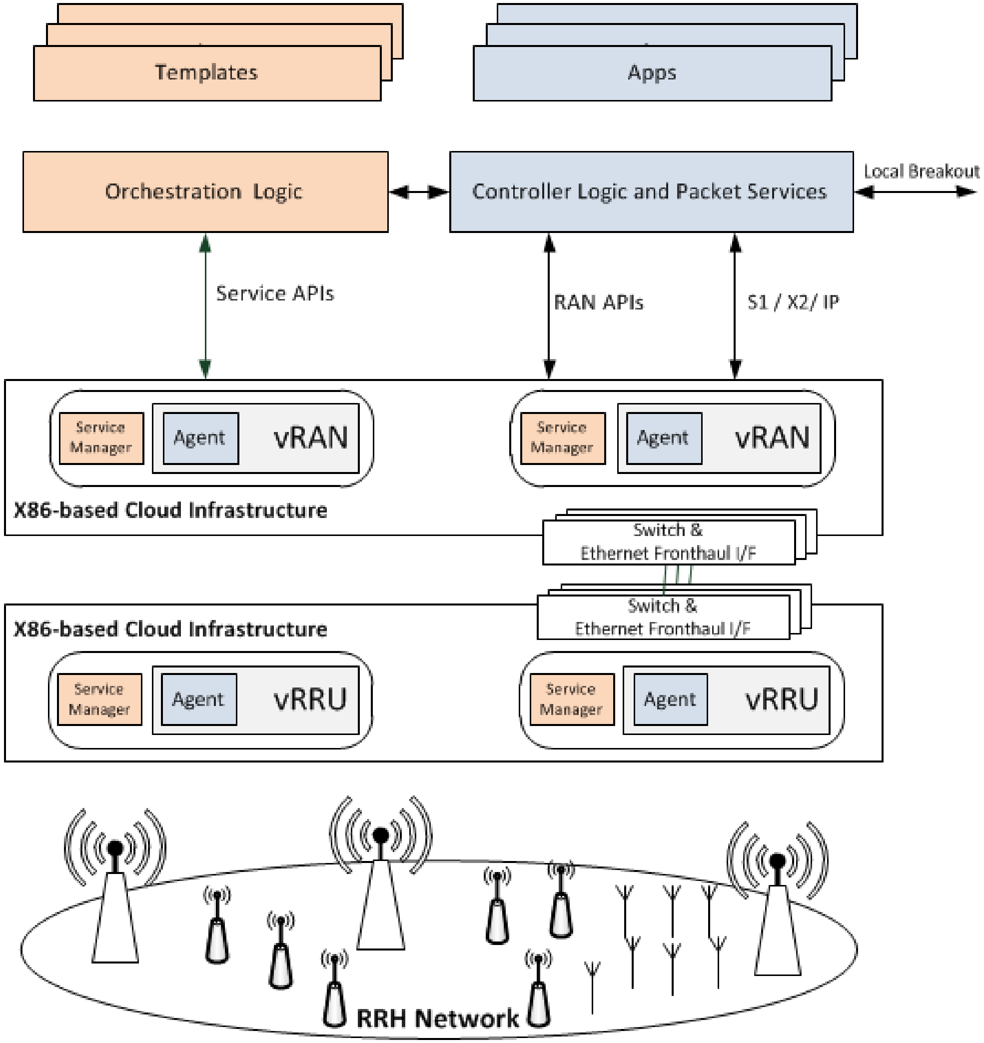
Abstraction and programmability

- **Control plane APIs allowing fine grain radio and core control and monitoring**
- **Effective representation of the network state at different network levels allowing**
  - fine-grained programmability, coordination and management of atomic or composed services across different domains/regions via
- **Network graphs can be separated based on**
  - Region, operator, cell, …
- **Encompass data models**
  - Time-frequency status and resources
  - Spatial capabilities
  - Key performance indicators



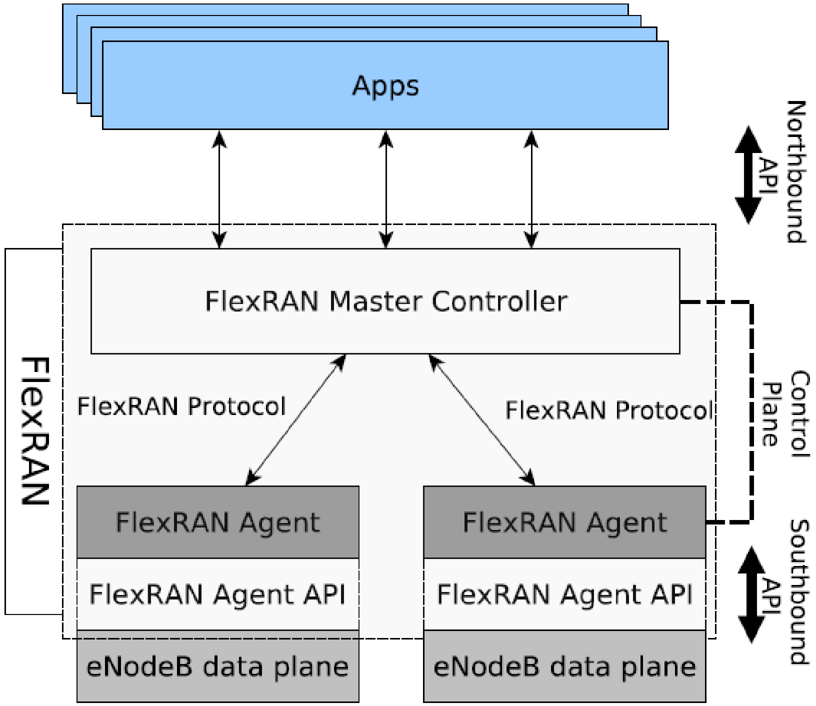
Abstraction and programmability

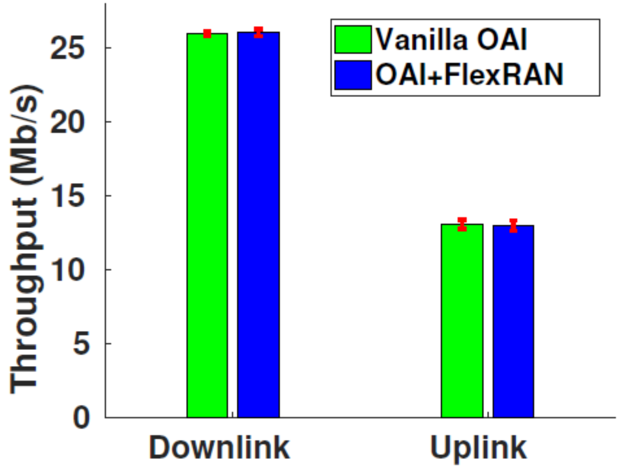
- **Data-plane and control plane programmability**
  - RAN API
  - Local breakout
- **Hierarchical controller logic managed by the orchestrator**
  - non-time critical → centralized entity
  - time critical → edge entity
  - May offloaded time critical operation to an agent acting as a local controller
- **Cognitive management, self-adaptive, and learning methods**
- **Northbound Application programming interface**



Abstraction and programmability RTC Design

- **Three subsystems and three time-scales**
- **Network app: eNB scheduler, and monitoring**

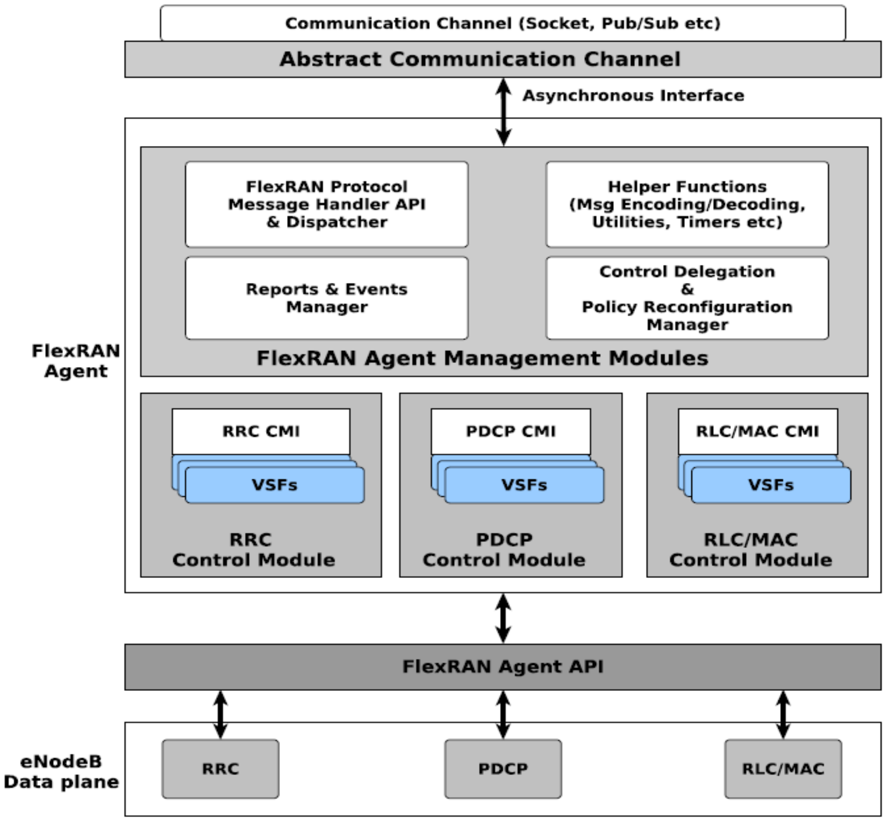




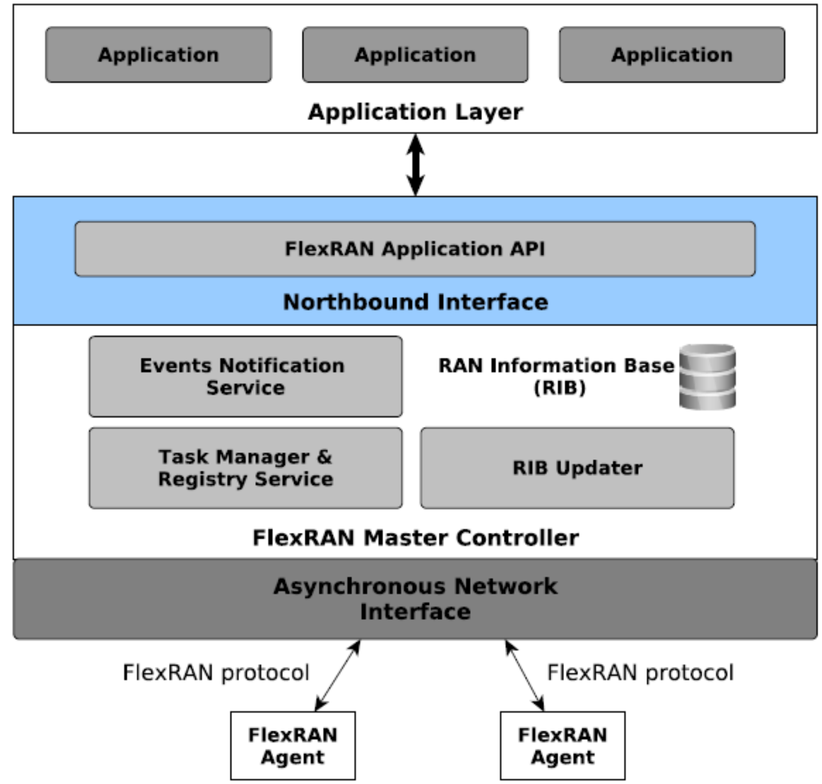
Message	Field	Usage
Configuration request	Configuration type	Type of configuration, either set or get
	Cell configuration flag	Bit map of the requested cell configuration
	Cell configuration list	List of cells (in IDs) to request configuration
	UE configuration flag	Bit map of the requested UE configuration
	UE configuration list	List of UEs (in IDs) to request configuration
Configuration reply	Cell configuration	Requested cell configuration report
	UE configuration	Requested UE configuration report
Status request	Status type	Can be periodical, one-shot, event-driven
	Status period	Period in Transmission Time Interval (TTI)
	Cell status flag	Bit map for the requested cell status
	Cell list	List of cells (in IDs) to request the status
	UE status flag	Bit map for the requested UE status
	UE status list	List of UEs (in IDs) to request the status
Status reply	Cell status	List of cell including the statistic reports
	UE status	List of UE including the statistic reports

Abstraction and programmability
Agent-Controller Design

Agent

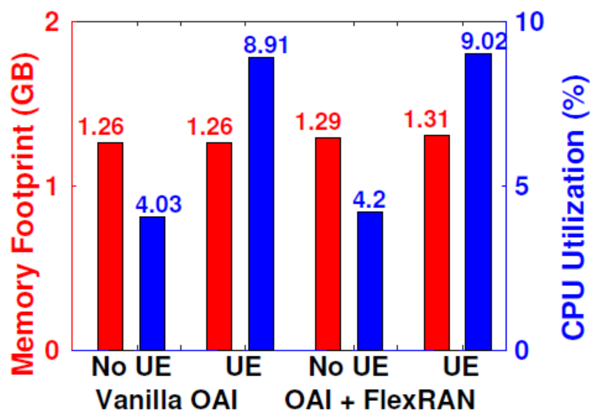


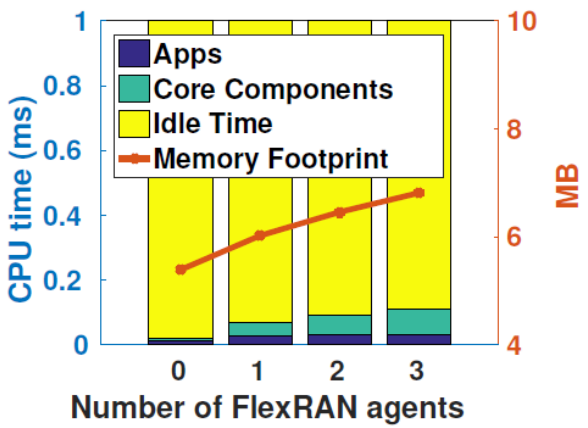
Controller



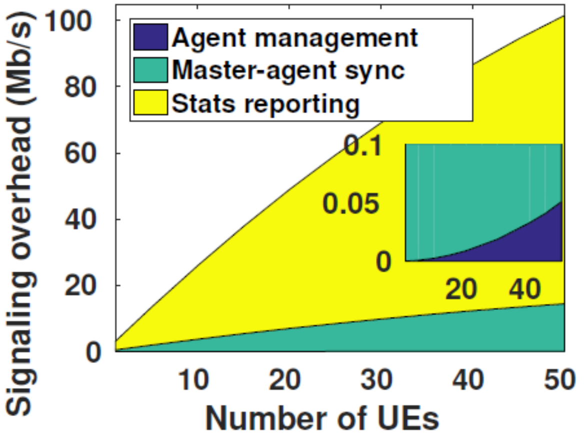
Abstraction and programmability
RTC Scalability

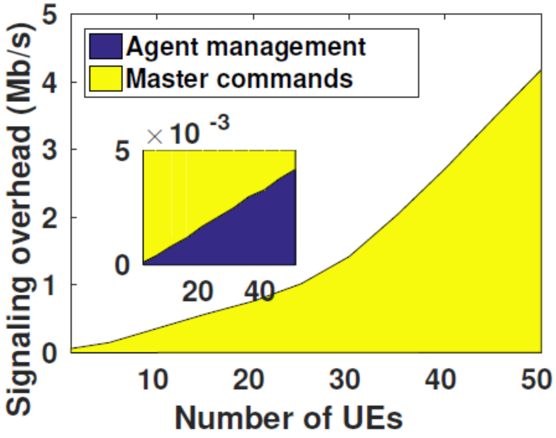
CPU Utilization and memory footprint





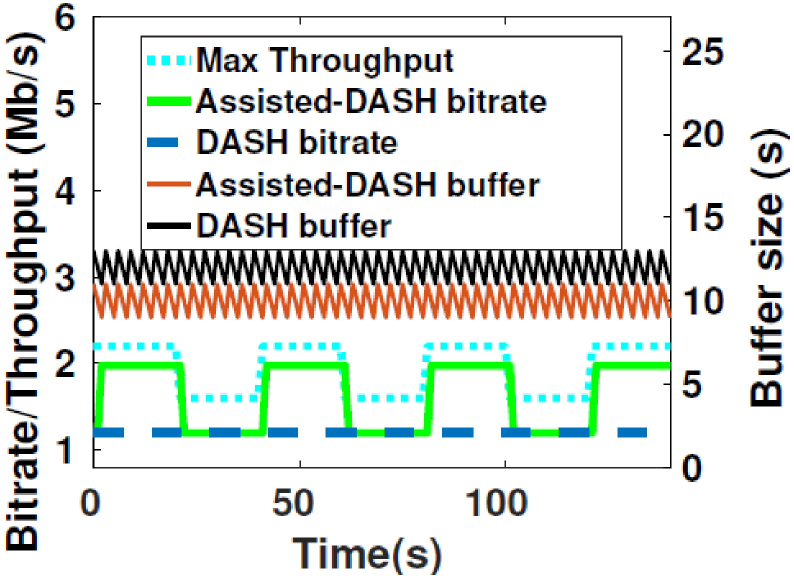
Agent-to-controller Controller-to-agent

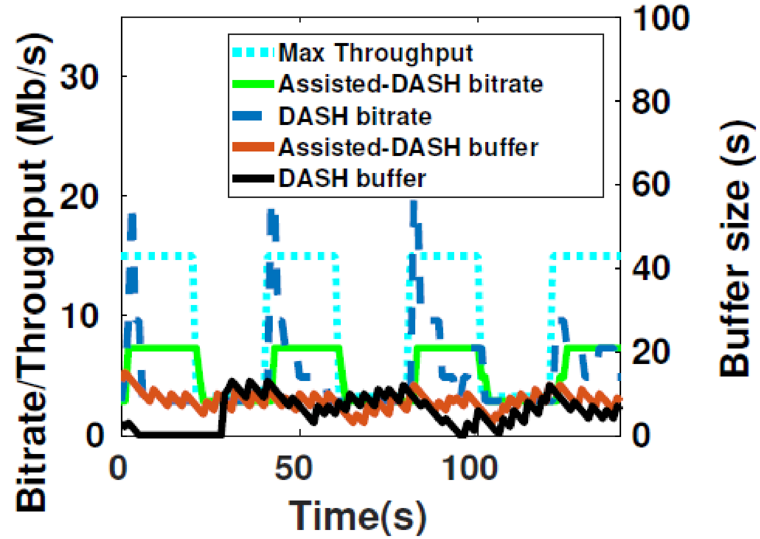




Abstraction and programmability
DASH Rate Adaptation with FLEXRAN

Low Variability	High Variability

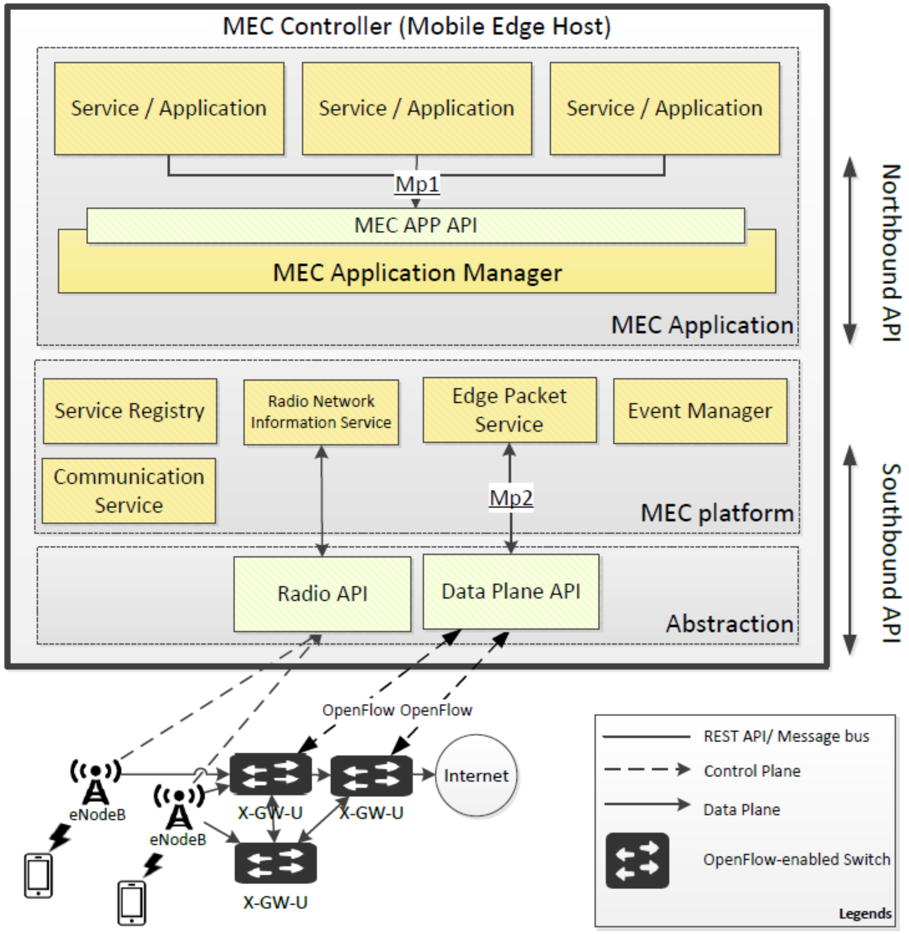




CQI	TCP Throughput (Mb/s)	Max sustainable bitrate (Mb/s)
2	1.63	1.4
3	2.2	2
4	3.3	2.9
10	15	7.3

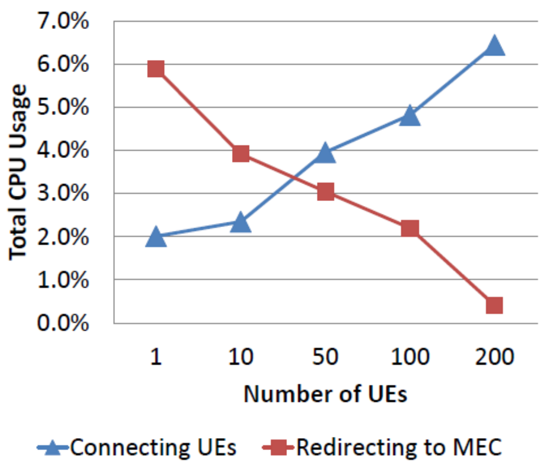
Abstraction and programmability LowLatency MEC

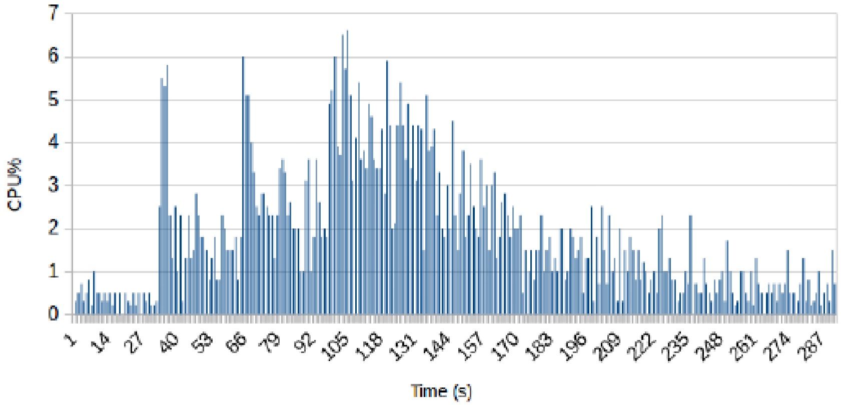
- **Flexibility data-plane programmability**
- **Compliant with ETIS MEC and 3GPP architecture**
- **Leverage SDN natively**
  - SPGW-C /MME as a MEC app
- **Support both RESTFULL and messagebus northbound**
- **MEC app and services**
  - Packet –in and out API
  - Redirect and Copy
- **Current testbed setup**
  - OVS+GTP patch, OF, OAI, FLEXRAN



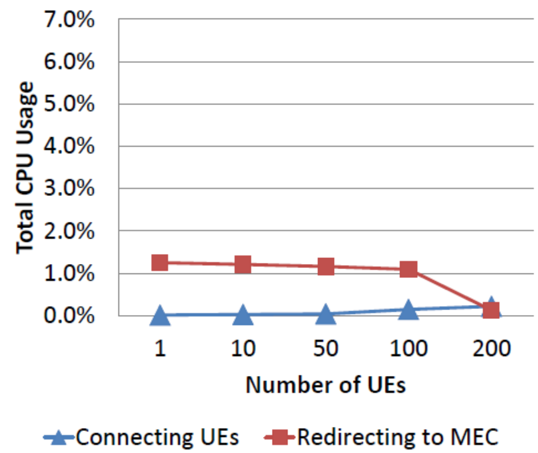
Abstraction and programmability
LL-MEC Scalability

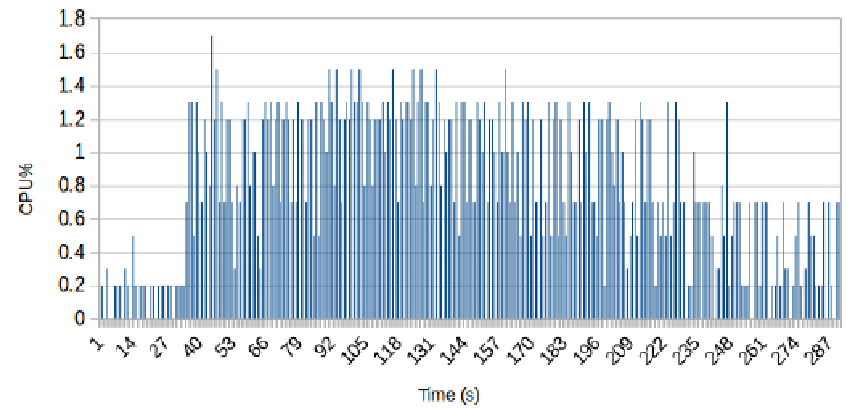
CPU Utilization and memory footprint





Agent-to-controller Controller-to-agent



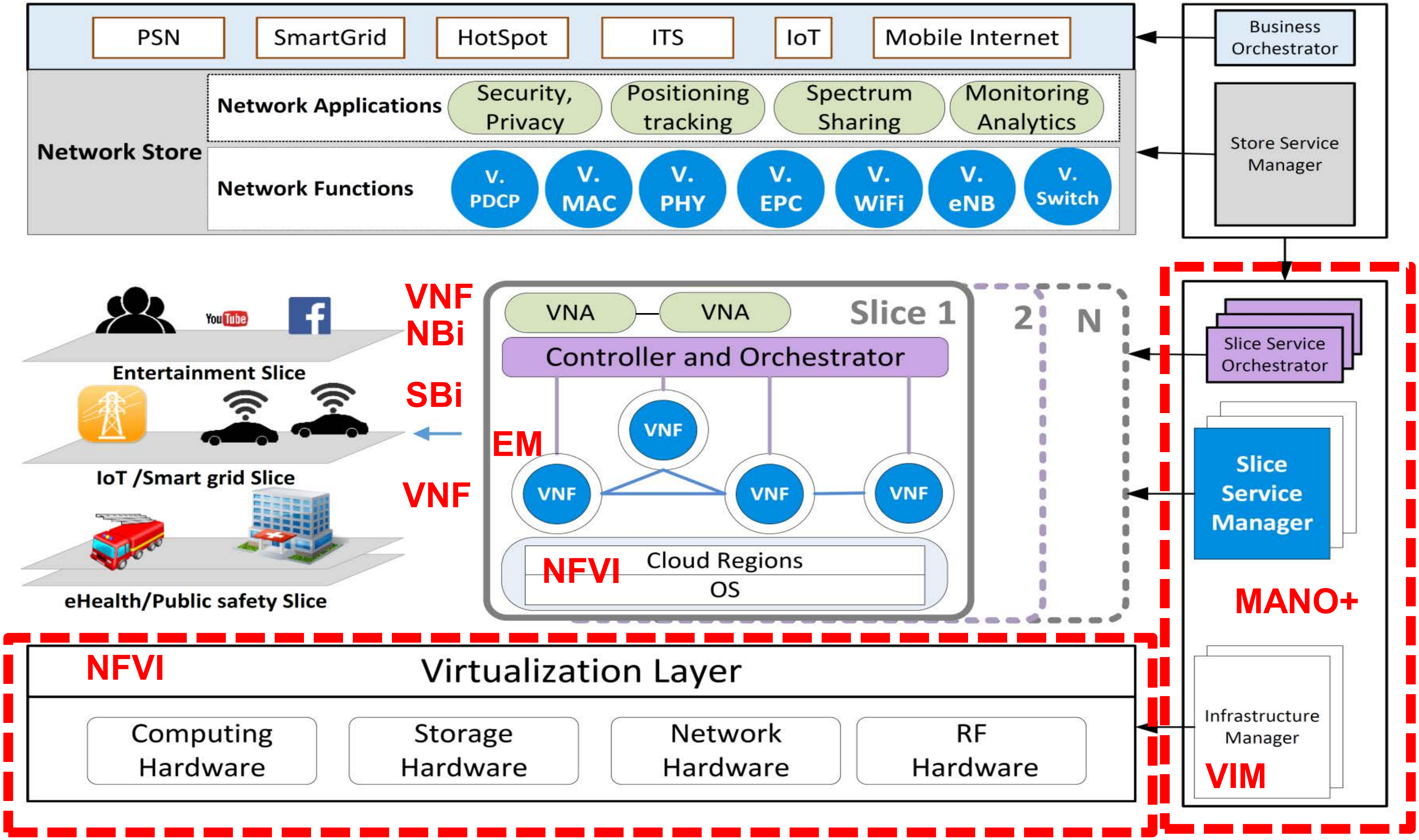


Network Slicing Architecture

Network slice and store concepts

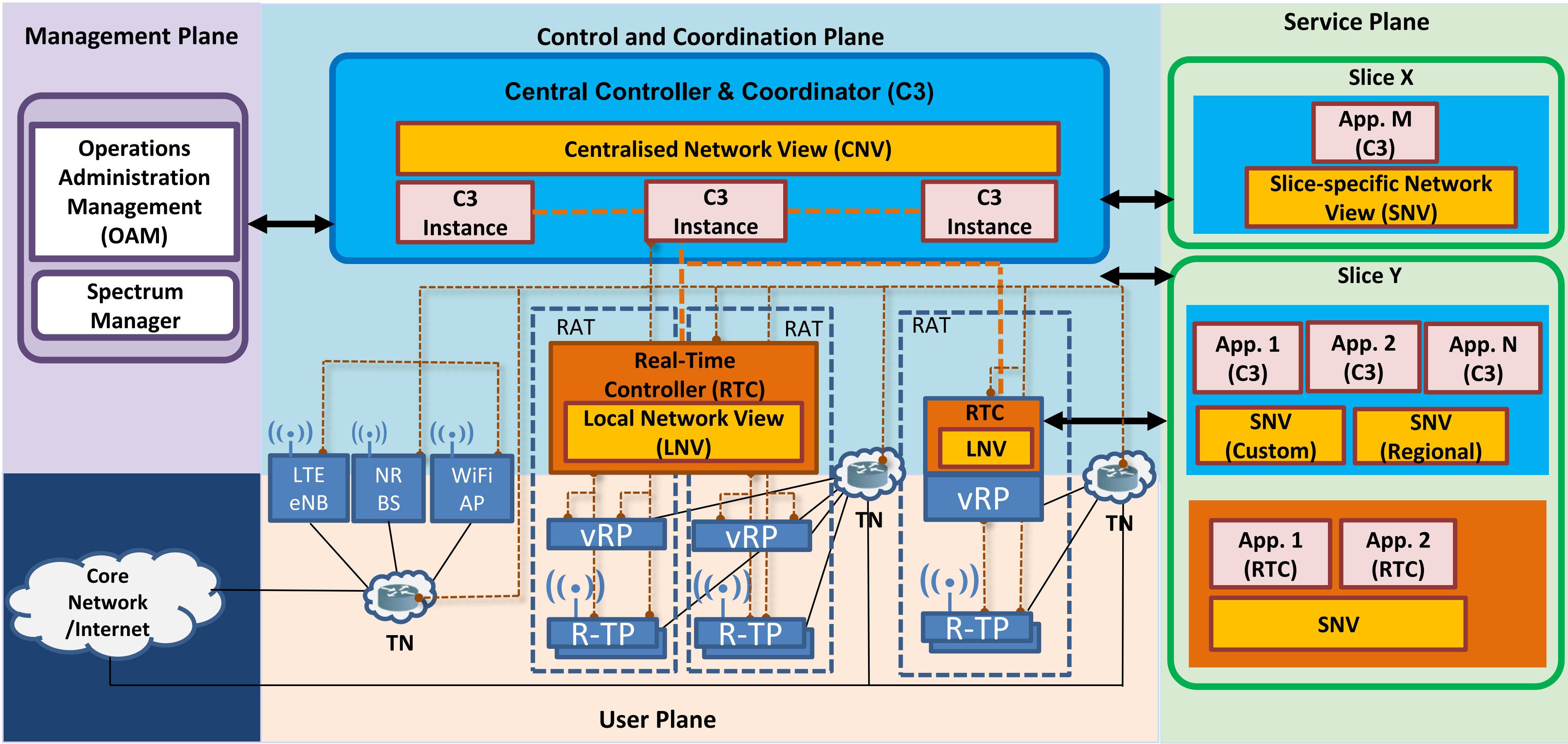
- **Slice manifest describes the business application across three planes**
  - Business, Service, Infrastructure
- **Network store allows creation of a slice for each virtual network through digital distribution platforms**
  - Network functions and network applications
- **A network slice is a virtual network that is instantiated on a common shared infrastructure (RAN, TN, CORE)**
  - Chain and compose adequately configured network functions, network applications, and underlying cloud infrastructures
  - Map and place them onto the infrastructure resources and assign target performance metrics
  - Program and scale them according to a particular business application scenario

Network Slicing Architecture
Network slice and store concept



Controller Architecture

RAN Coordination and Programmability



RP: Radio Processing
R- TP: Radio Transmission Points
TN: Transport Node

NFV-SDN-MEC Interplay

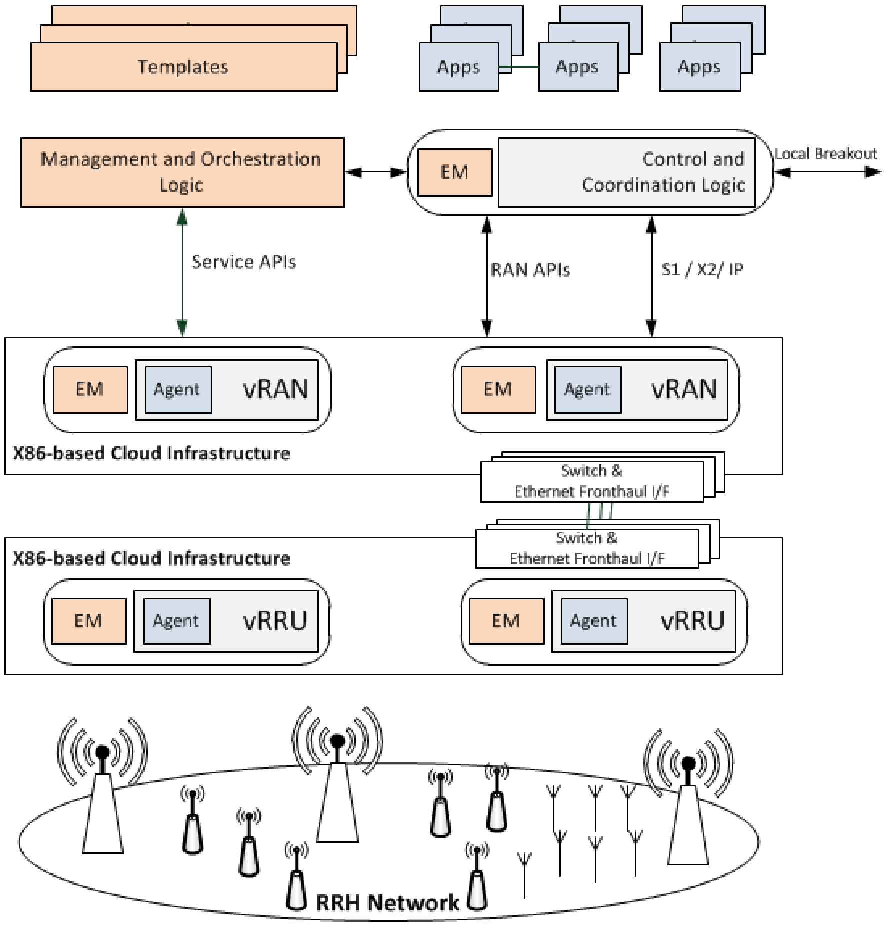
Need for Flexibility

- **Scaling capacity and managing a dense and potentially time-varying network require a tight coordination and programmability**
  - Obj: Decouple the control plane from the data plane
- **Flexibility to change the network service definition on-the-fly to deal with spatiotemporal network and traffic diversity**
  - Obj: abstraction and programmability of network functions
    - Control plane and data plane
- **Multi-service multi-tenant networks**
  - Obj: dynamic network service composition from reusable network functions
    - Nested chaining following micro-service design pattern
  - Obj: resource sharing (infra, radio, and spectrum)

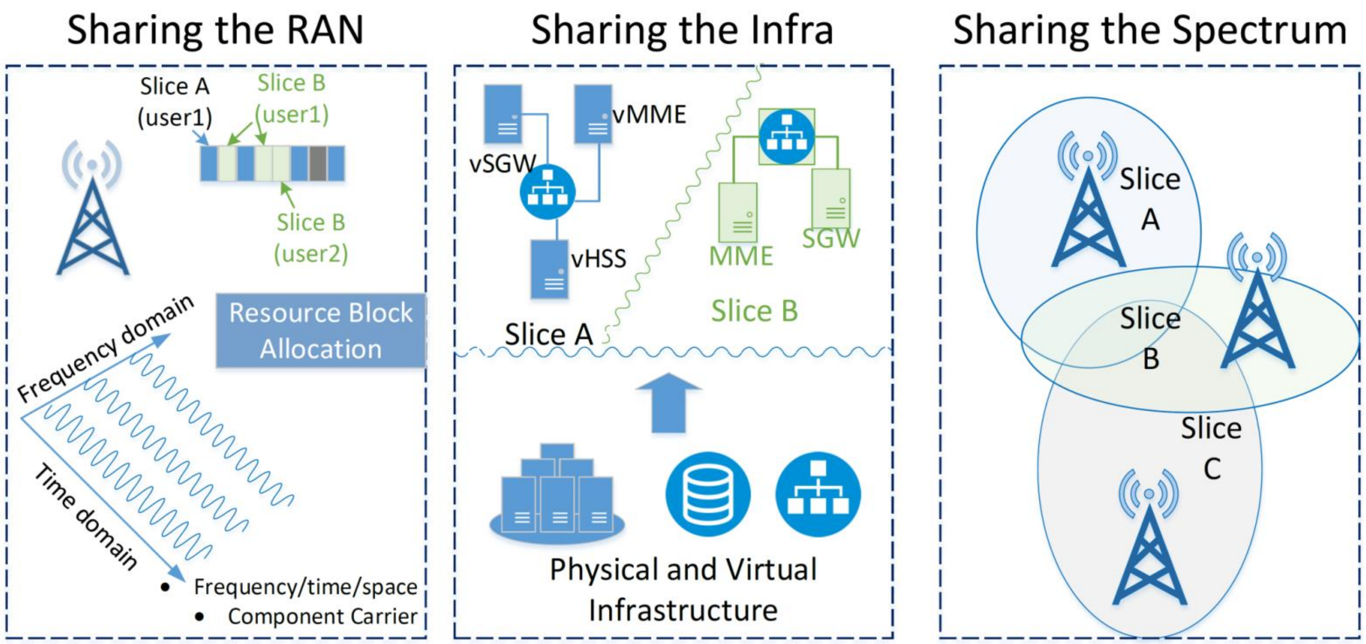
NFV-SDN-MEC Interplay

Need for a Flexibility

- **Data-plane and control plane programmability**
  - Flexible chaining and configuration
  - Traffic steering and local breakout
- **Hierarchical controller logic managed by the orchestrator**
  - C3: non-time critical → centralized entity
  - RTC: time critical → edge entity
  - Agent : offload a subset of time critical functions
- **Single-domain orchestrator**
  - Exploit EM-agent coupling
  - Controller interface



RAN Slicing and Sharing



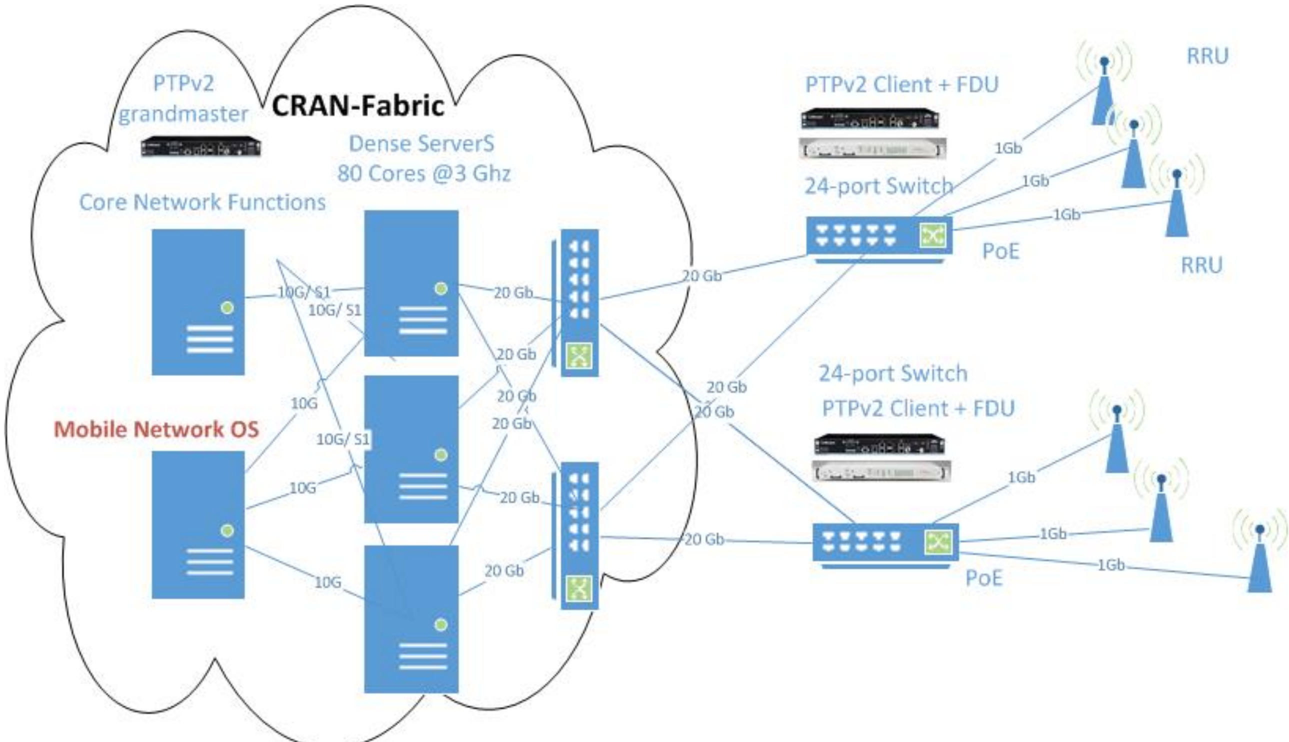
- **Sliceable elementary resources**
  - [RRU/Antenna, Fronthaul, CRAN, Backhaul]
  - [CPU/MEM/NET, Radio resources, spectrum]
  - [configuration, chain, placement]
- **Resource abstraction and network programmability is a key to achieve the required flexibility in slicing**

NEXT STEPS

MNOS and CRAN Playground

Sophia-Antipolis at Eurecom

- **Mobile Network OS consolidation**
  - Slicing, programmability, and APIs
  - Network application development and SDK
  - Support of vertical services



Conclusion

- **The exploitation of cloud technologies, SDN, NFV, and MEC can provide the necessary tools to**
  - Flexibly design, compose, chain, and place an E2E service
- **Network slices and stores are key to**
  - deliver differentiated network service offerings optimized for each and every use case, application and user
- **Gap between static and cognitive management and orchestration**
  - Exploit machine learning and data mining techniques

Conclusion

- Decision making
- Complex Event Processing
- ML and Analytics

- On-the-fly function loading and chaining
- Delegation and policy enforcement
- Reconfiguration

- Cognative RAN
- Flex RAN
- Cloud RAN
- Virtual RAN
- Soft RAN