MOSAIC-5G.io
A community led consortium

Leveraging an Ecosystem of 5G services

What is LL-MEC?

A Low Latency Multi-access Edge Computing Platform for Software-Defined Mobile Network

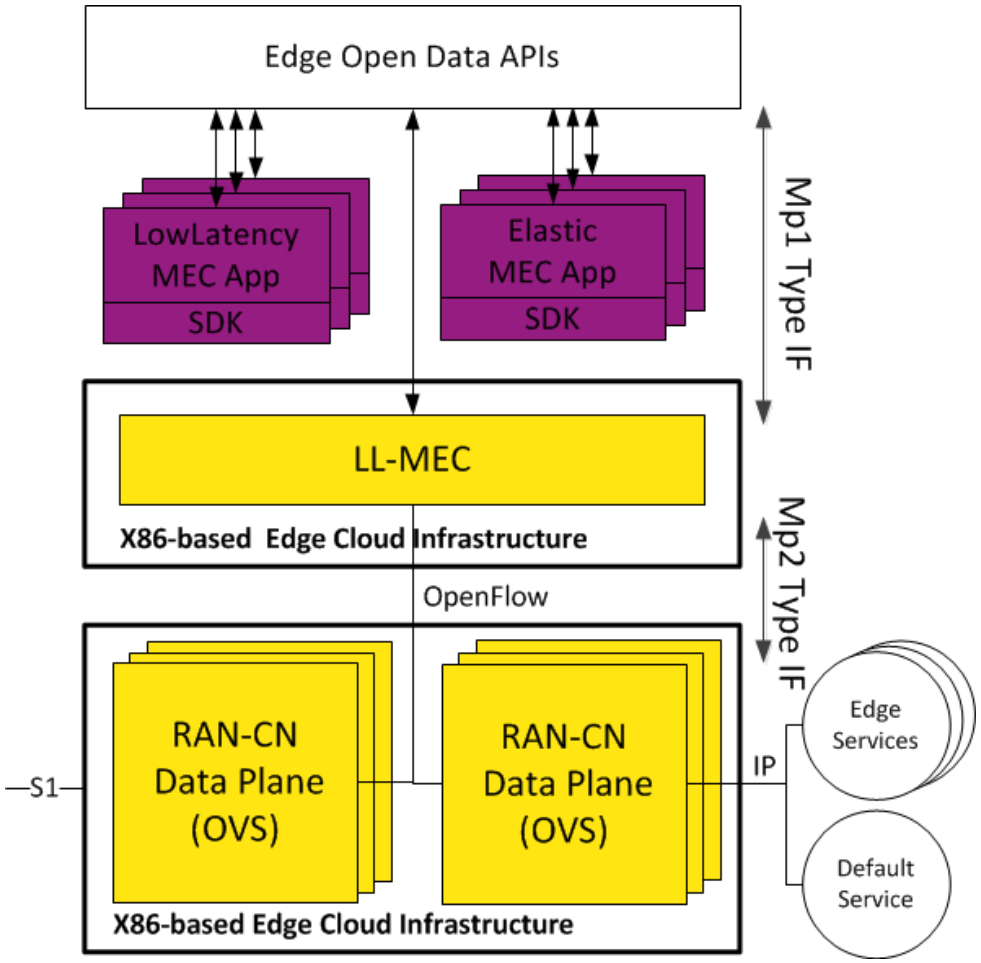
LL-MEC objectives

- Abstraction layer between RAN and CN data plane and LL-MEC platform
- Monitor and control the traffic in the network
- IP-service endpoint and real-time RAN information on per user/service basis
- Low latency applications policies for data plan setup based on traffic statistics

LL-MEC features

- CN Control and Data plane separation (SDN-based MEC)
  - Leveraging OpenVirtualSwitch with GTP support and OpenFlow protocol
  - Verticalization of core network at network edge
  - Shared and dedicated core network
- Abstraction and Programmability
  - Flexibility and programmability to the underlying RAN and CN data plane
    - Abstraction of underlying data paths with OpenFlow APIs
  - Traffic rules are automatically generated and passed to the associated OpenVirtualSwitch
- Application SDK
  - Flexible application programming framework
  - Network abstraction via well-defined northbound interfaces

LL-MEC schema



- **Application manager (mp1)**
  - Low-latency: CoreAPI, MBus
  - Elastic: RestAPI, MBus
- **Platform (mp2)**
  - Edge packet service
    - Multi OF libs, OVS
    - Static and dynamic rules
  - Radio network information
    - Real-time control and monitoring
  - Event manager
- **Abstraction**
  - Data plane APIs: OF Agent
  - C-plane Radio API: FlexRAN Agent

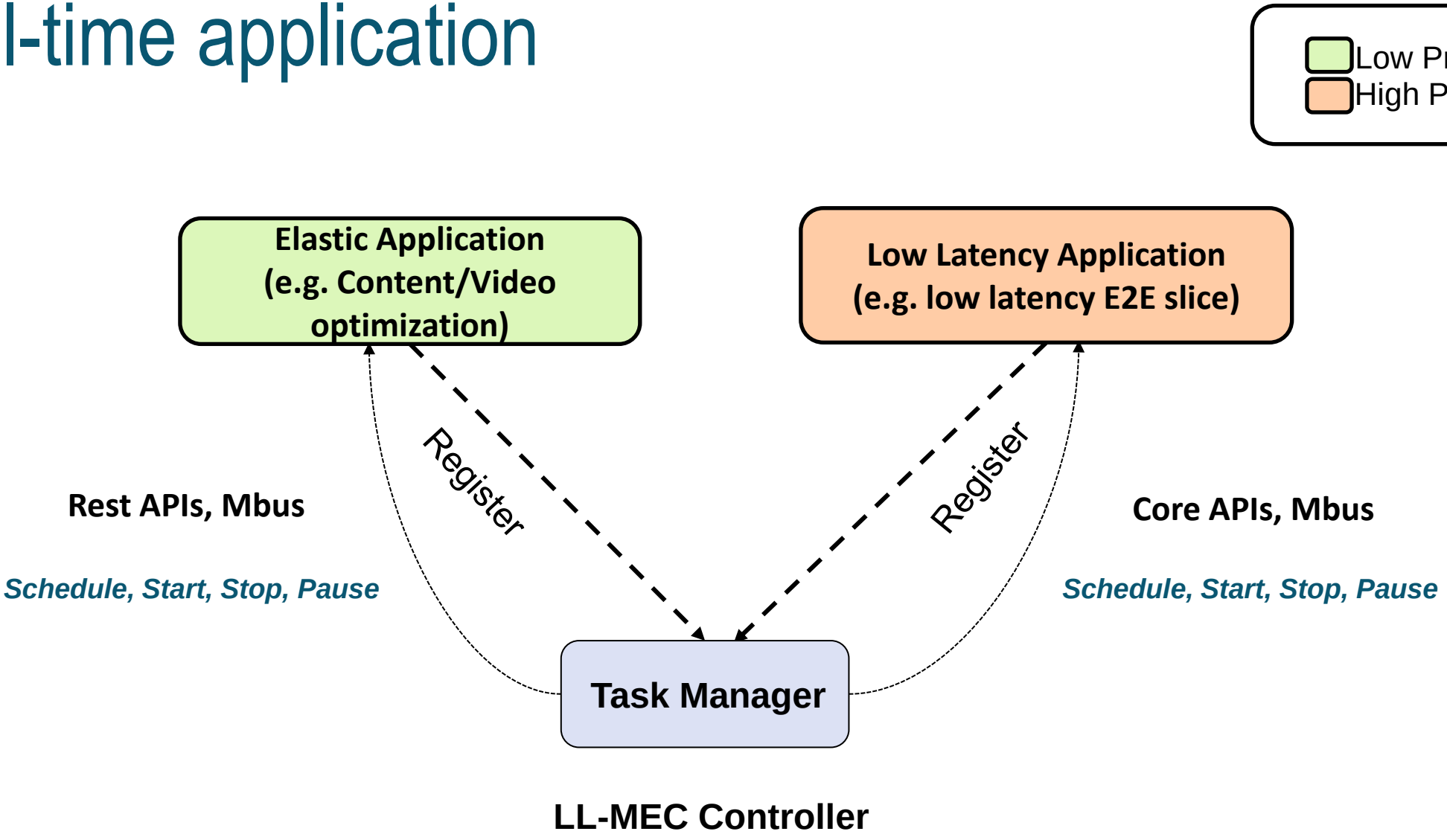
LL-MEC implementation

- LL-MEC controller
  - From scratch in C++ and Python
  - x64 Linux support
  - Core network programmability coordinated with RAN real-time operation
  - Flexible application programming environment at the network edge
  - Apache V2.0 license
  - OpenFlow rules support
- GTP enabled OVS
  - OVS 2.7.0 applied with GTP patch
  - x64 Linux support (4.7 kernel)
  - Apache V2.0 license

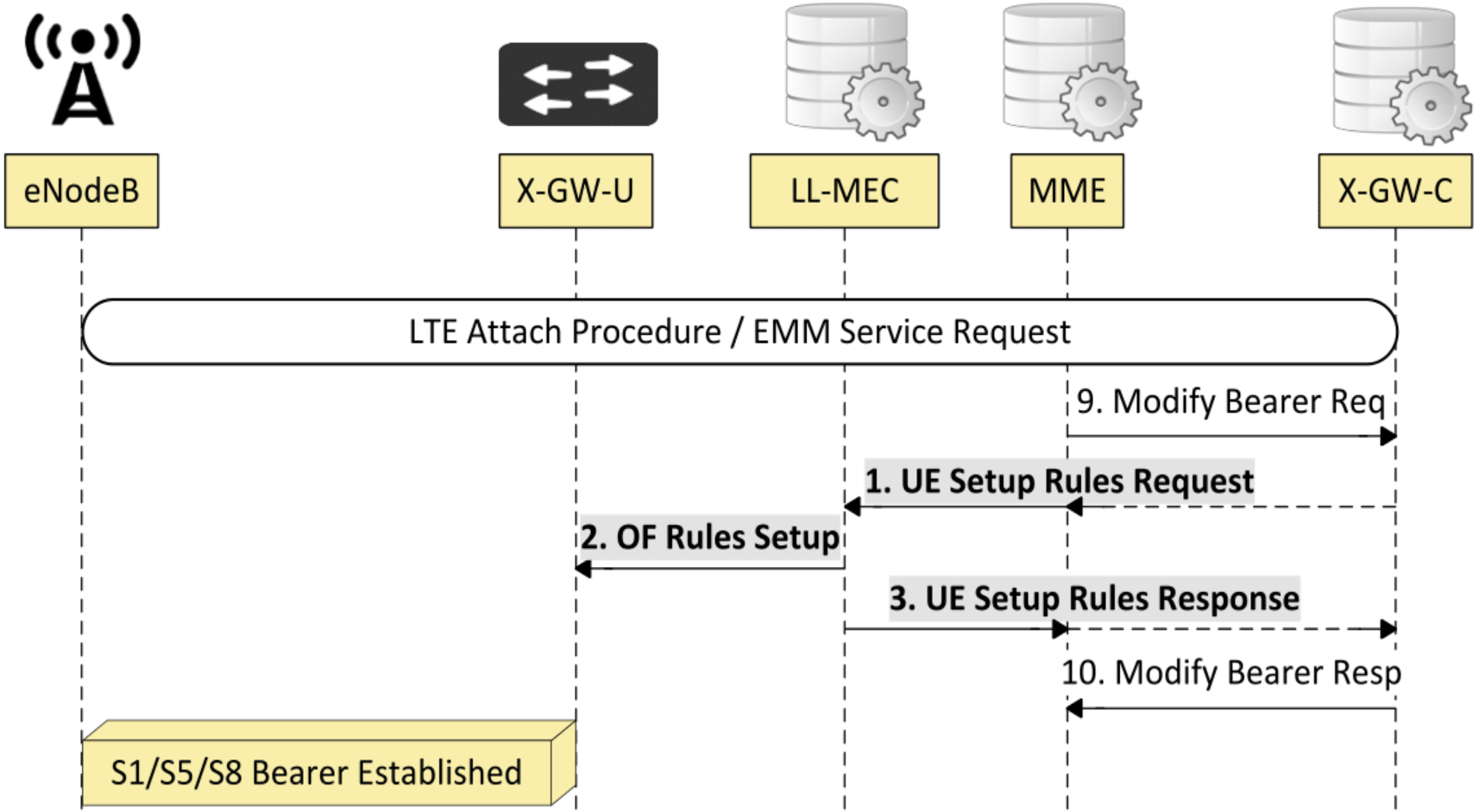
LL-MEC API

API	Target	Direction	Example	Applications
Add UE / Slice	UE, Slice	EPC → Slice	• Create session request • SGi endpoint update • Slice create	• Core network CU • S-GW-C
Delete UE / Slice	UE, Slice	EPC → Slice	• Release access bearer request • SGi endpoint update • Slice destroy	• Core network CU • S-GW-C
Get UE / Slice info	UE, Slice	EPC → Slice	• Scheduling decisions • Coordinated slicing	• Monitoring
Redirect UE / Slice	UE, Slice	EPC → Slice	• Video optimization • IoT gateway	• Control actions • Programmability
Get flow statistic	Stats	EPC → Slice	• Traffic shaping	• Monitoring

Real-time application



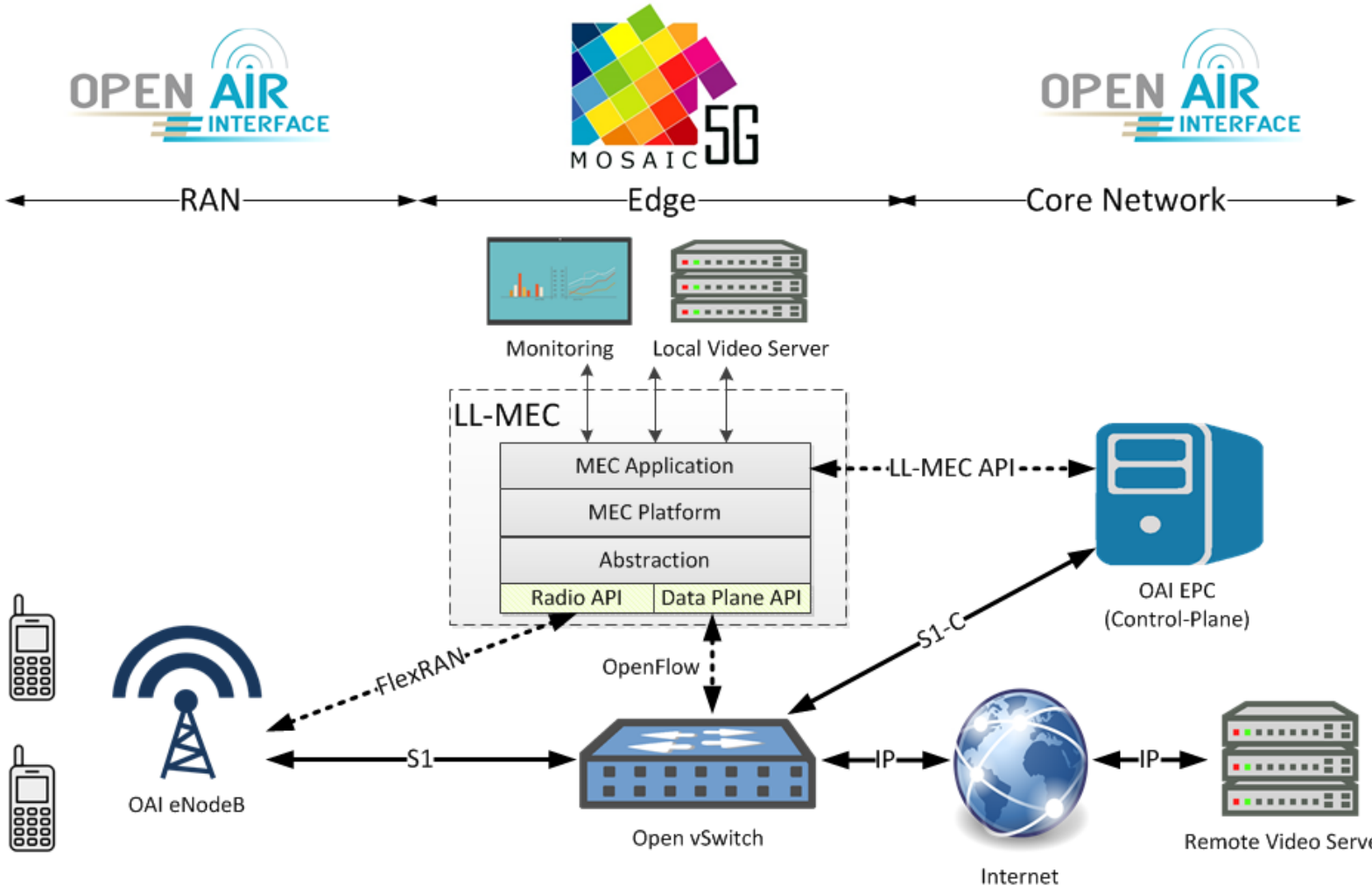
Sequence diagram of S1/S5/S8 bearer setup



LL-MEC apps

- Video optimization
- Load balancing
- Content caching
- Recommendation system
- Traffic steering
- E2E network slicing
- Dedicated core network
- IoT Gateway

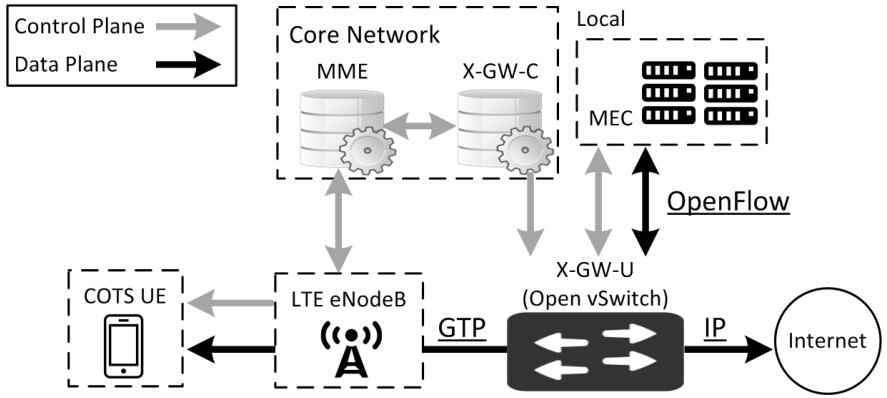
Video Streaming Optimization



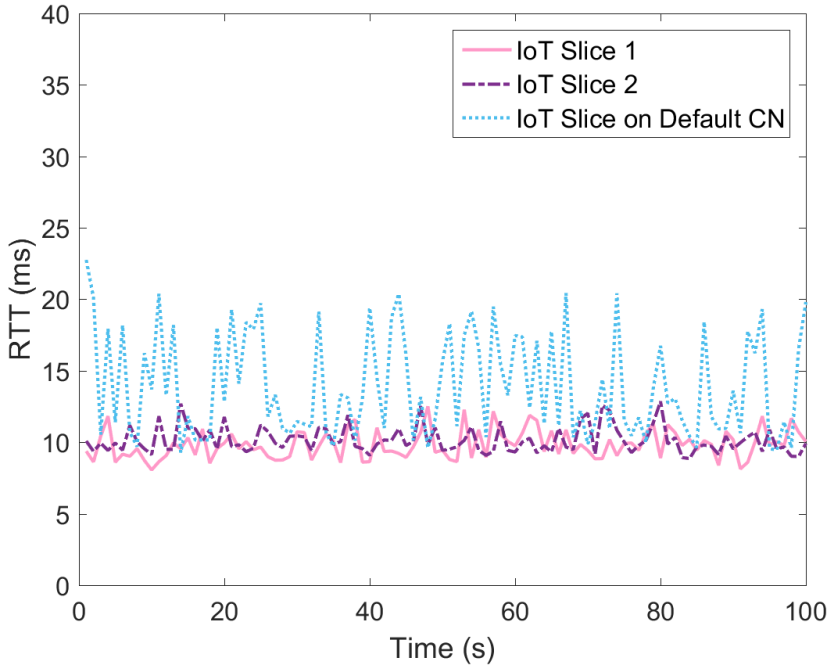
IoT Gateway: X-GW-U as a service

- Shared vs. Dedicated data plane to support isolation
  - network slicing

Network setup with a total of 2000 UEs



Latency of isolated / non-isolated slices



Useful links

- Slicing and orchestration in service-oriented 5G architecture
- LL-MEC: An SDN-based MEC Platform
- LL-MEC platform
- How to enable DP programmability in mobile network?

Mosaic5G-Contact

E-mail: llmec_users@lists.eurecom.fr

Website: mosaic-5g.io/ll-mec

Twitter: @mosaic5g