

# UniFlex

## A Framework for Simplifying Wireless Network Control

**P. Gawłowicz**, A. Zubow, M.Chwalisz, A. Wolisz



**TKN**

**Telecommunication  
Networks Group**

# Motivation

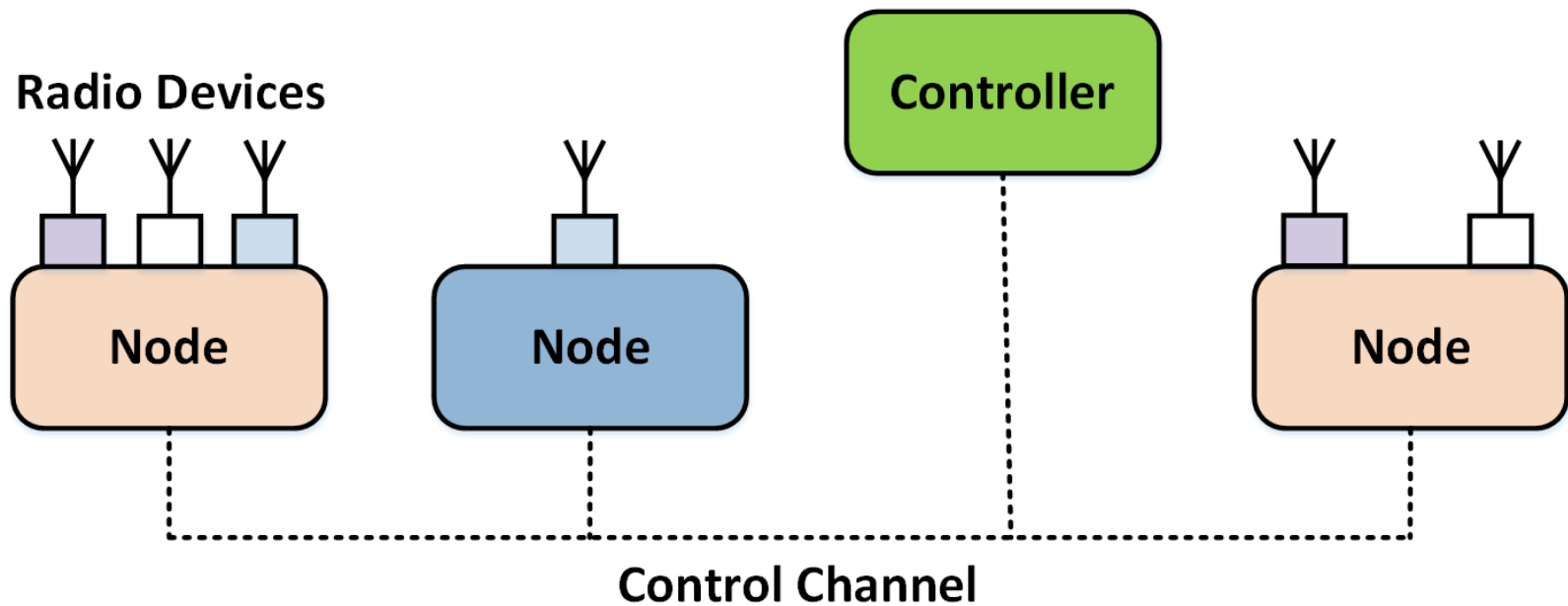
- Wireless network evolved to be highly complex system
- Constantly changing radio environment requires frequent tuning of parameters
- Control applications are ususally implemented case-by-case

# Objectives

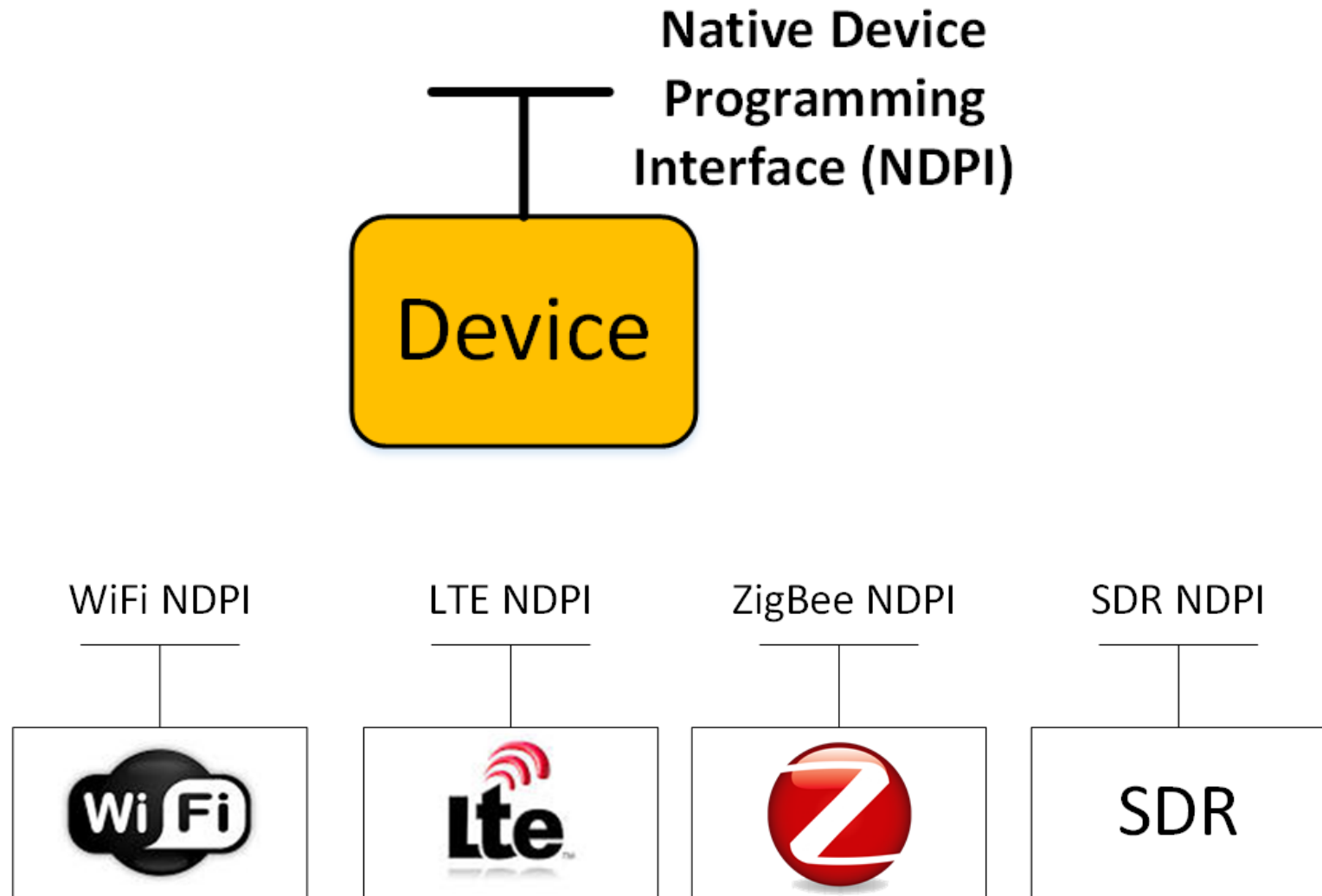
- Reduce threshold for experimentation and shorten time of prototyping
- Provide framework for development of novel control and management solutions wireless networks
- Improve performance and efficiency of wireless networks

# System Model

## Wireless Network



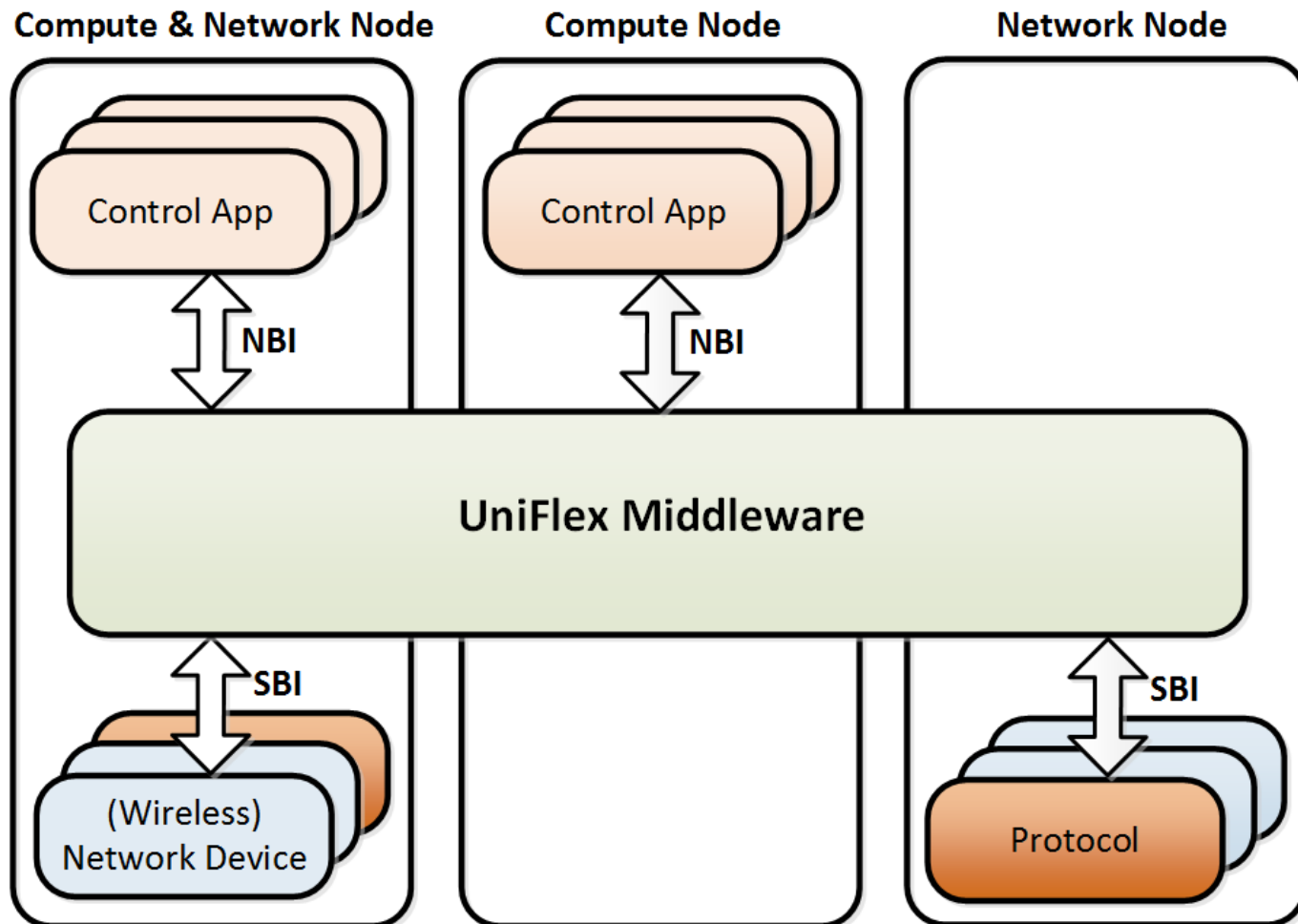
# Device Programming Interface



# Requirements

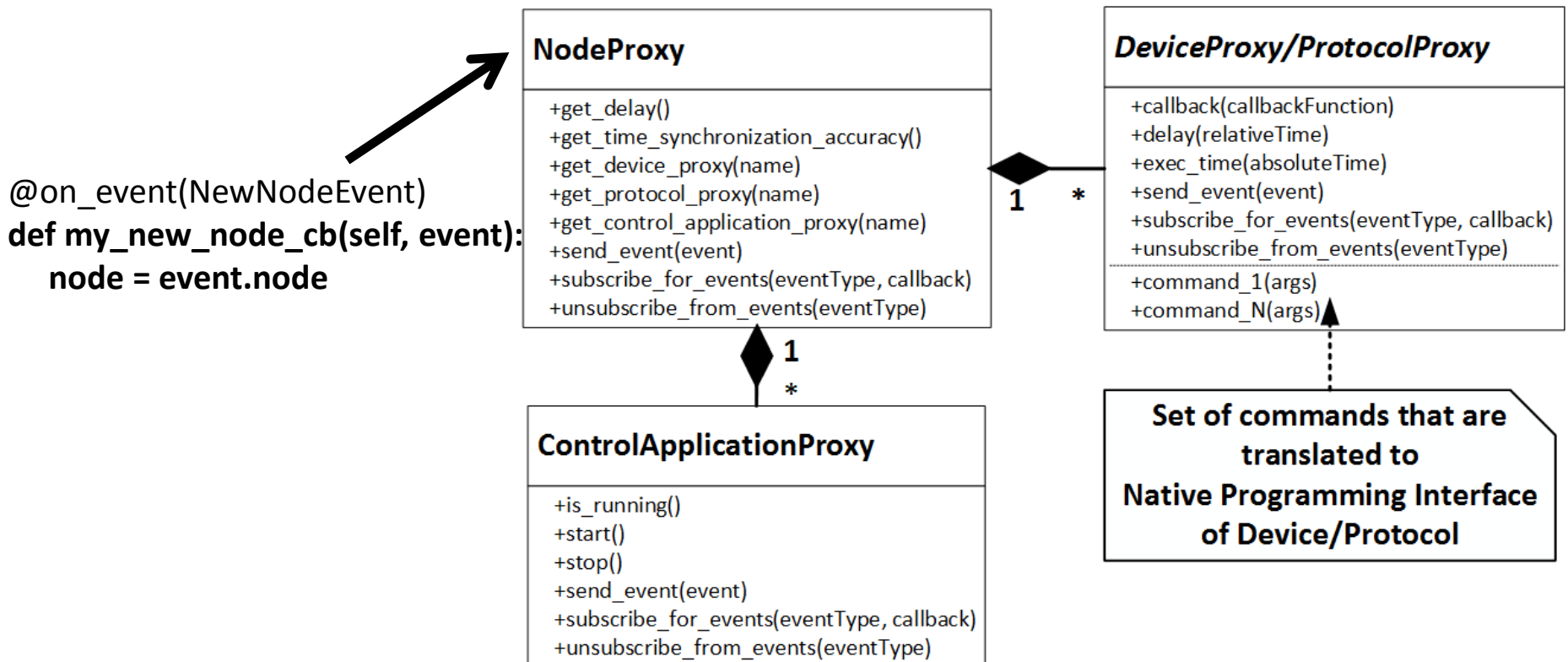
- Coordinated collection of information from and execution of commands on different:
  - protocol layers (**cross-layer**),
  - heterogeneous devices (**cross-technology**)
  - multiple nodes (**cross-node**) within network
- Possibility to implement logically centralized and physically distributed control applications
- Support for multiple levels of control for scalability reasons
- Support for proactive and reactive control schemes
- A high-level API for control of operation of individual wireless devices and groups of devices
- Unification of different NDPI

# UniFlex Architecture



# Northbound Interface

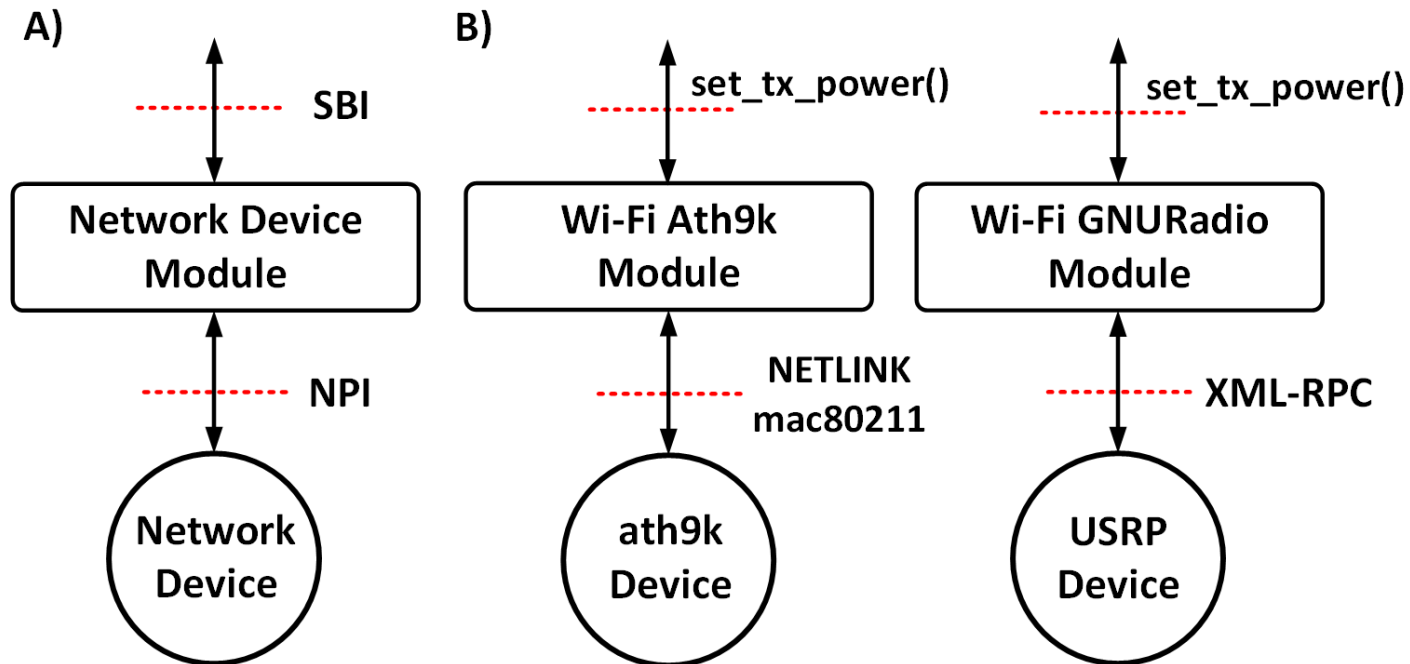
- Based on **event** exchange mechanism + Remote Procedure Calls





# Southbound Interface

- translates function calls from control applications into NDPI
- unifies different NDPIs



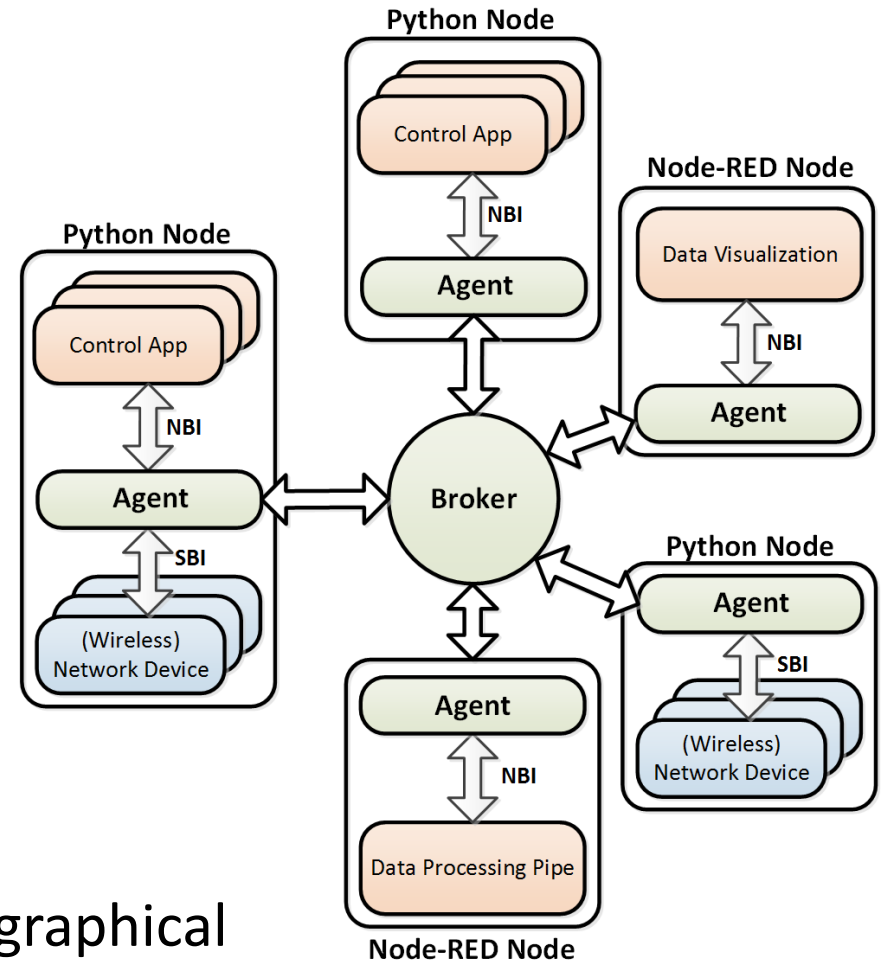
# Distributed Middleware

- The framework is an distributed middle-ware platform that:
  - provides communication channel for control applications
  - exposes NDPI of network devices to control applications
- It takes care about maintaing up-to-date information, including:
  - node discovery and monitoring connection between all nodes
  - notifying control applications about changes

# Implementation



- **Event** delivery mechanisms implemented using PUB/SUB sockets from ZMQ
- **RPC** implemented on top of unicast event mechanism
- Integration with Node-RED – a graphical language for pipeline data processing



# Calling Examples

@on\_event(PacketLostEvent) ← subscribe for PacketLostEvent

**def my\_pkt\_lost\_cb(self, event):**

# get device proxy from node proxy

device = event.device

# execution of blocking call

pwr = device.get\_tx\_power()

# delay execution of call by 3 seconds

device.delay(3).set\_tx\_power(pwr+2)

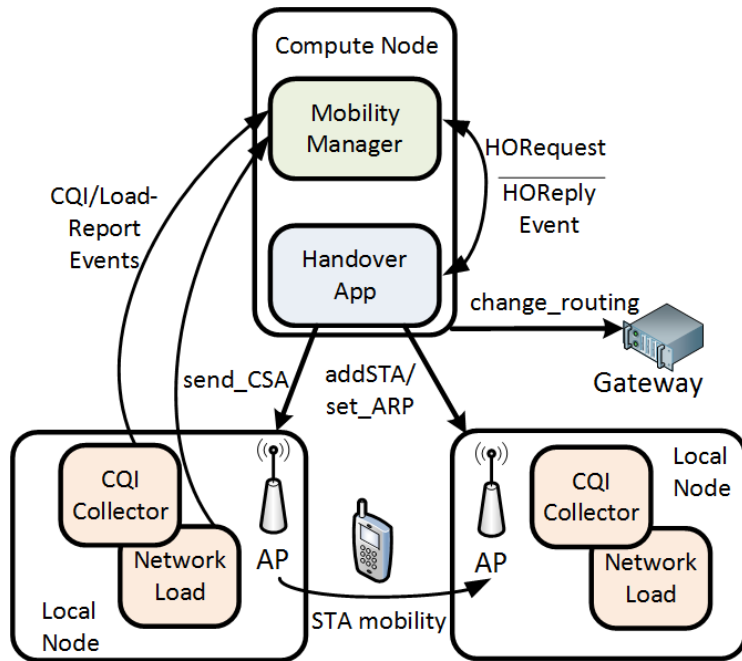
# schedule execution of non-blocking call

t = datetime.now() + timedelta(seconds=6)

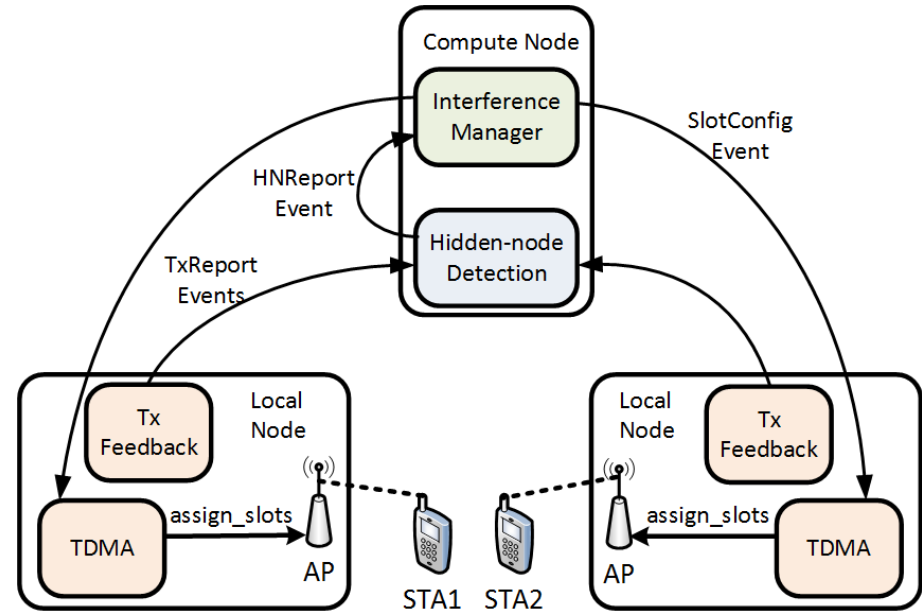
device.exec\_time(t).callback(my\_get\_power\_cb).get\_tx\_power()

# Example Applications

## Mobility Management – Handover



## Interference management through air-time management



# Conclusions

- UniFlex is a framework that simplifies prototyping of novel wireless solutions
- It provides rich API for control and management of network entities
- It allows to implement local, central and hierarchical control planes.
- It's usability was proved in several implemented use-cases.

**Thank you!**