IEEE Wireless Communications and Networking Conference (WCNC 2019)

Model-Driven Framework to Speed up Design and Exploitation of Sensor Networks

presented by : Rémy KOCIK

Abdenour Kifouche, Redha Hamouche, Remy Kocik, Abderrezak Rachedi, Genevieve Baudoin

Université Paris-Est, ESYCOM (EA2552) , ESIEE-Paris, France

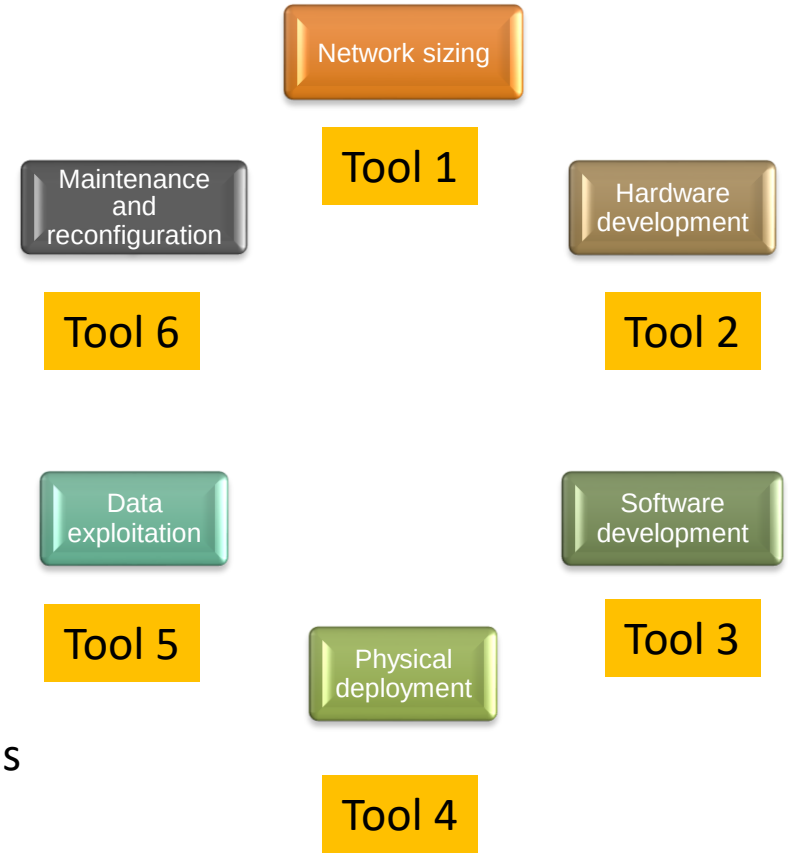
(abdenour.kifouche, genevieve.baudoin, redha.hamouche, remy.kocik)@esiee.fr

Outline

- INTRODUCTION
- TOWARDS A NEW FRAMEWORK
- THE FRAMEWORK THROUGH A USE-CASE
- CONCLUSION AND FUTURE WORK

Context

- Sensors Network Implementation
- Size may be important
- Lifecycle made of design steps and exploitation steps
- Dedicated tools/languages used in each step



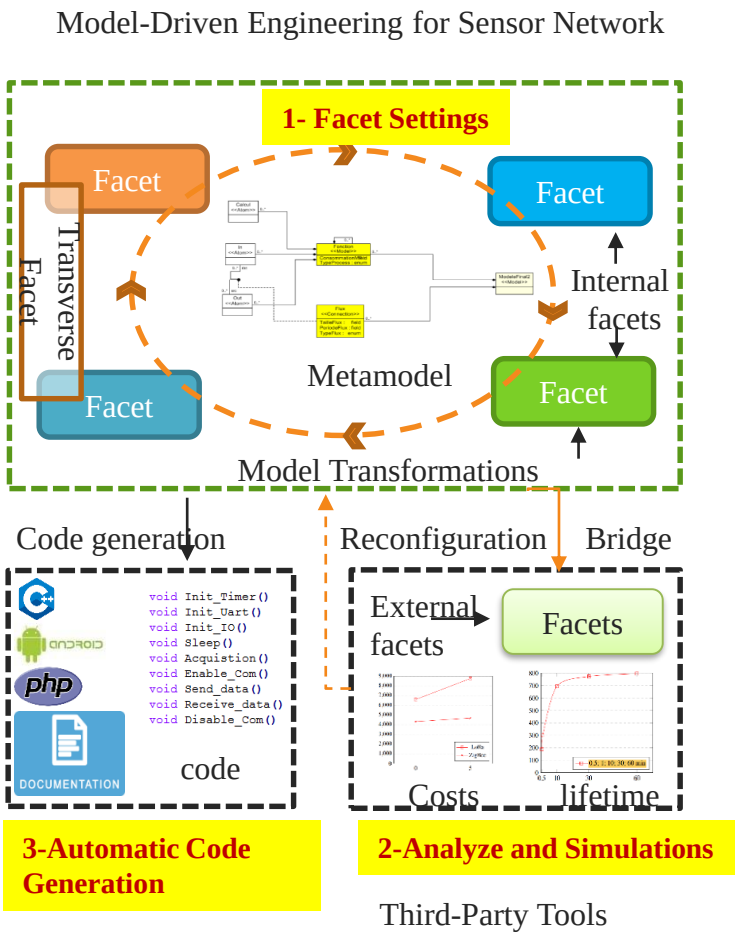
Sensor network lifecycle

Challenges

- Manage complexity in each step
- Reduce time spent to describe data in each tools
- Ensure Consistency between tools/steps

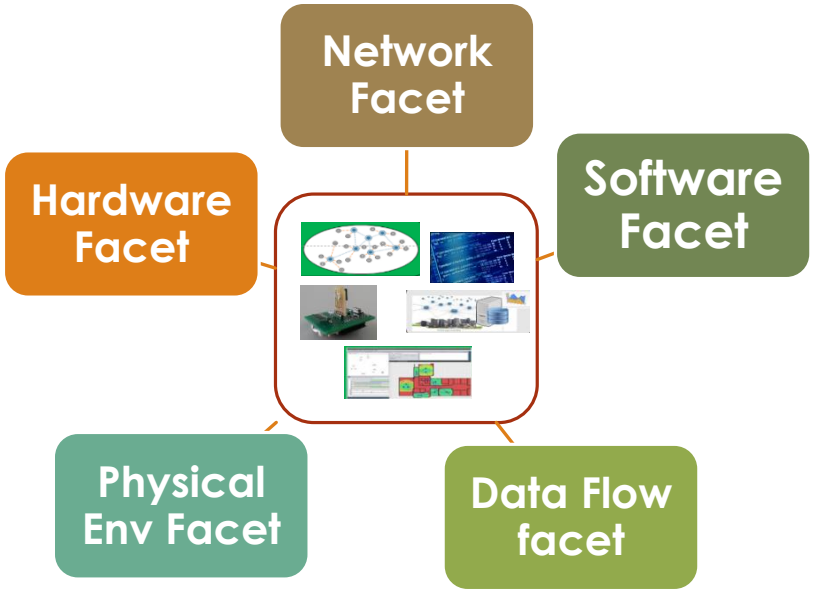
Proposed Solution

- A methodology based on MDE
  - high level graphical models to specify
  - implementation obtained by code generation
- Define a Metamodel
  - model description with separation of concern
  - generic and extendable
- Multi-facet approach
  - domain-specific modeling language (DSML)
  - Focus on a subset of the metamodel to reduce complexity
  - External tools seen as external facets
  - Automatic Gateway/link between facets provided (M2M)



A Framework to speed-up SN design and exploitation

- Implements our multi-facet MDE methodology
  - Our metamodel
  - 5 facets
- Tools
  - Based on Eclipse Workspace
  - Extended Modeling Framework (EMF)
    For metamodel description
  - Acceleo
    For code generation



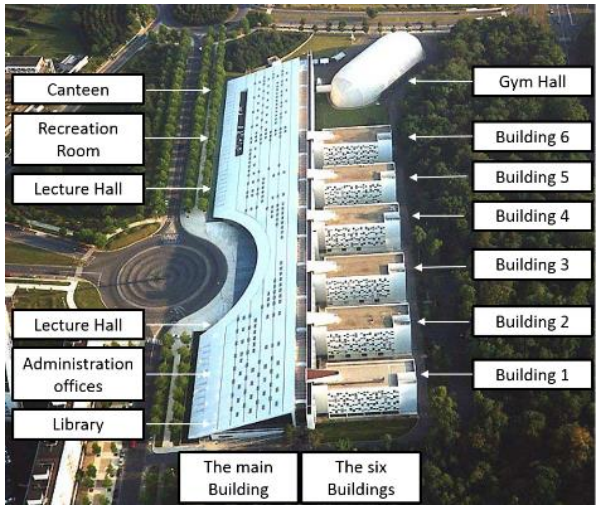
The Real Use Case

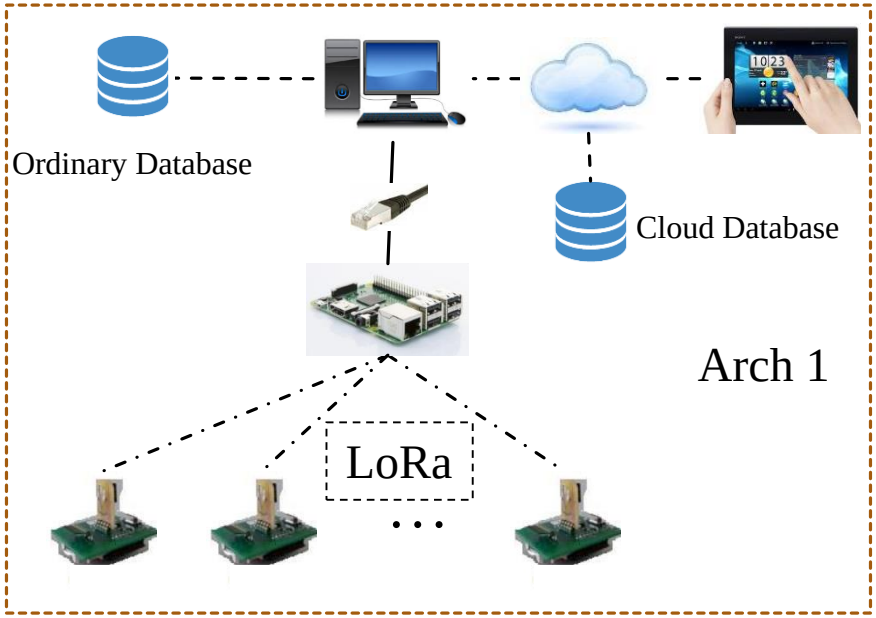
Building Humidity, Light & Temperature Monitoring

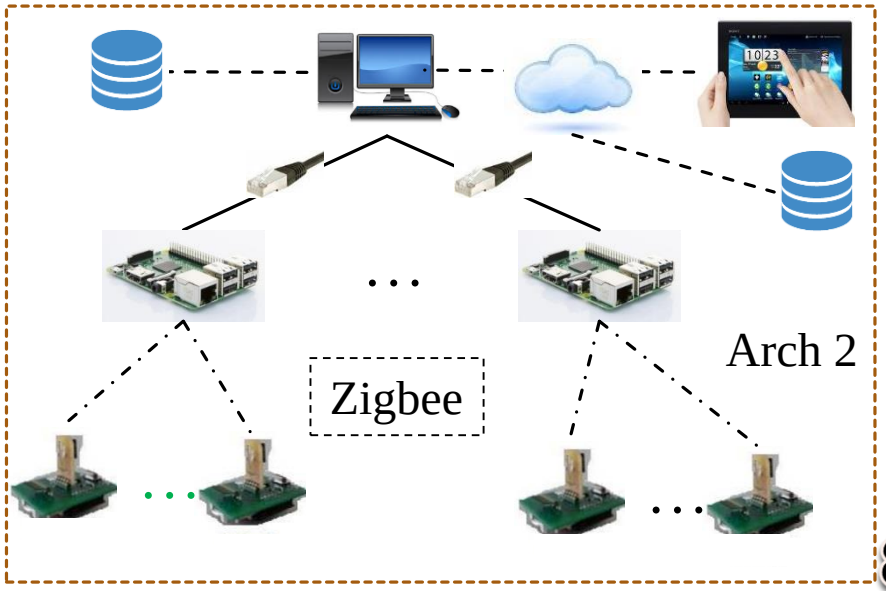
Our school is the physical environment for the sensor network deployment.

- 6 Buildings + main building,
- 5 floors/building,
- ~ 5 sensors per floor,

=> ~ 200 wireless battery powered sensors





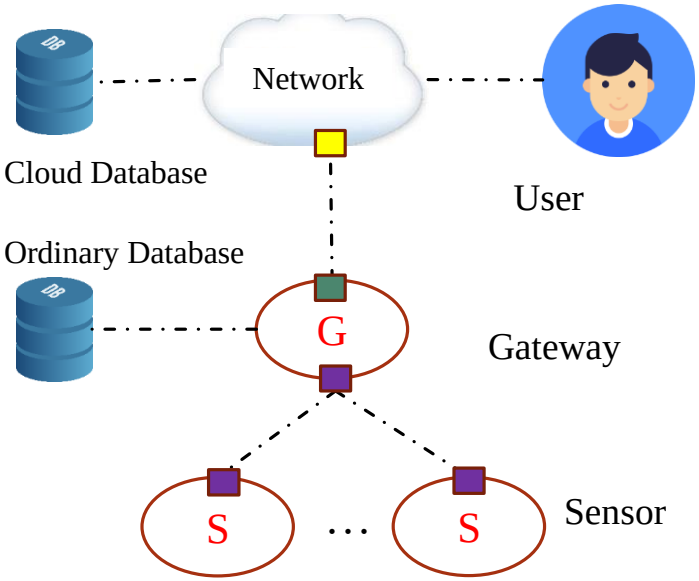


Framework Facet's

The Network Facet

Objectives :

- Describe the connectivity between nodes
- Check the end-to-end connections



Items of the model:

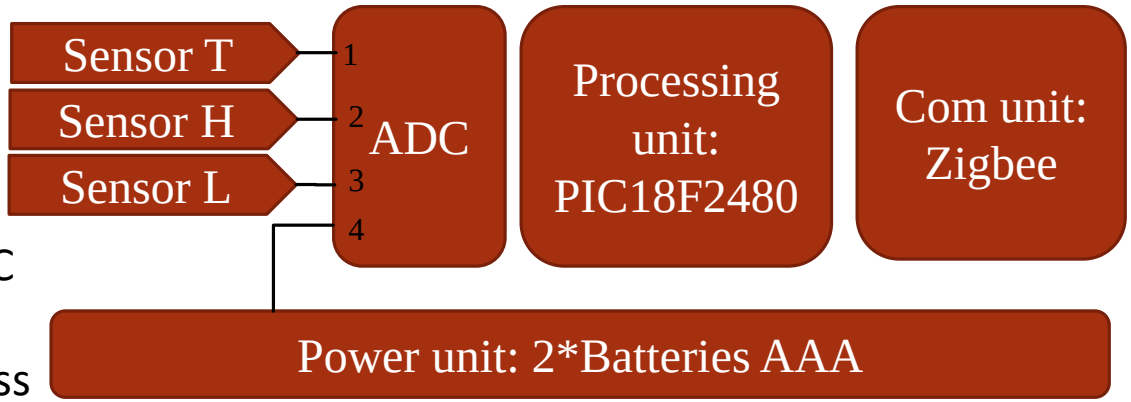
- Nodes (2...*)
- Database (0...*)
- User (0...*)
- External network infrastructure (0...*)
- Communications link (1...*)

Framework Facet's

The Hardware Facet

Objectives :

- Describe the hardware architecture of nodes
- Specify parameters for configuration, simulation....



Items of the model:

- Acquisition unit: Sensor (0...*), ADC
- Control unit: Actuator (0...*), DAC
- Communication unit: wired/wireless
- Processing unit
- Storage unit
- Power supply unit


Capteur

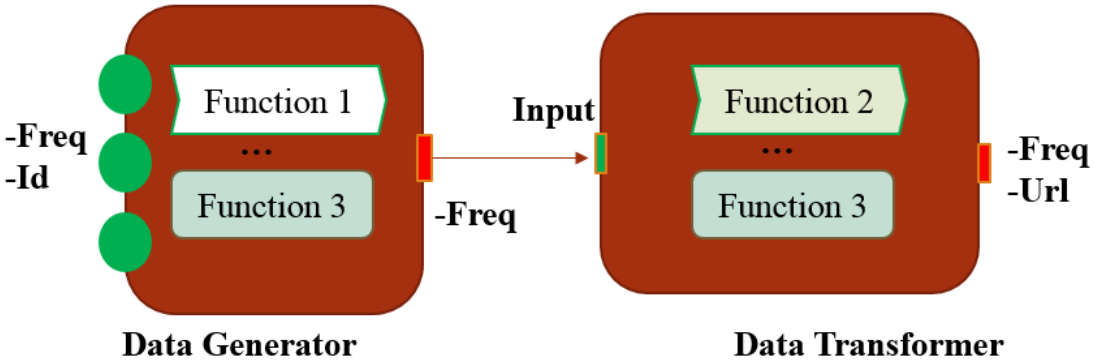
In each unit, information is specified for code generation and also for analysis and simulation (e.g. processing unit, power supply unit).

Framework Facet's

The Data Flow Facet

Objectives:

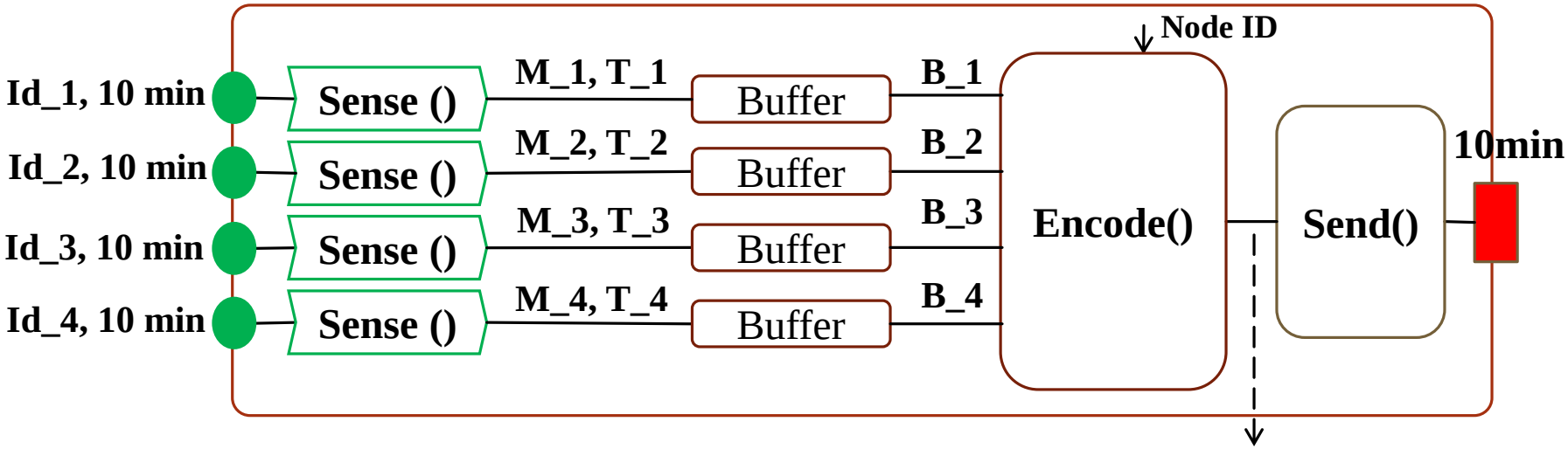
- Describes the flow of data
- Describes the format of the data exchanged



Elements of the model

- Data Generator (Data Generator)
- Data Transformer (Data Transformer)

A node can be formulated using a block diagram composed of functions



Framework Facet's

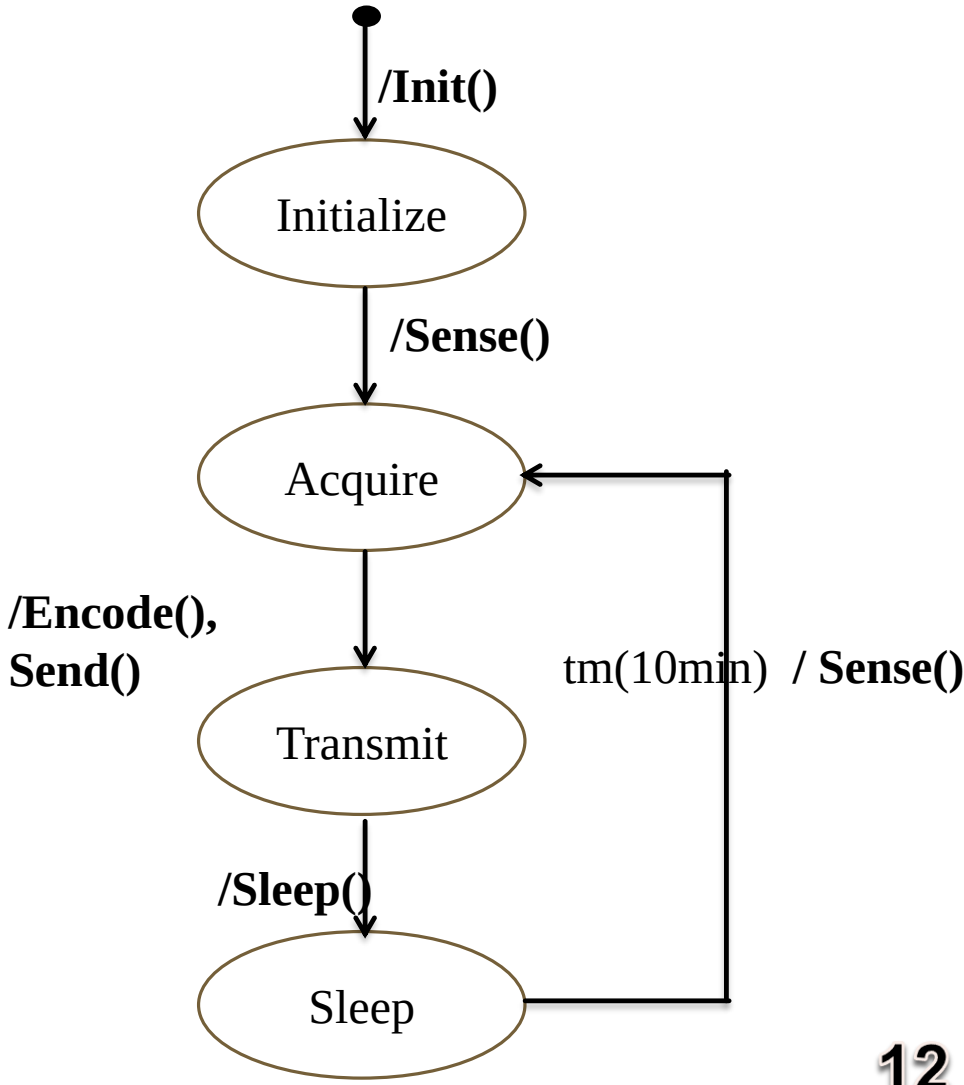
The Software Facet

Objectives :

- Describes the dynamical behavior of a node
- Complement the data flow view

Items of the model:

- States
- Transitions between states with associated conditions.



Framework Facet's

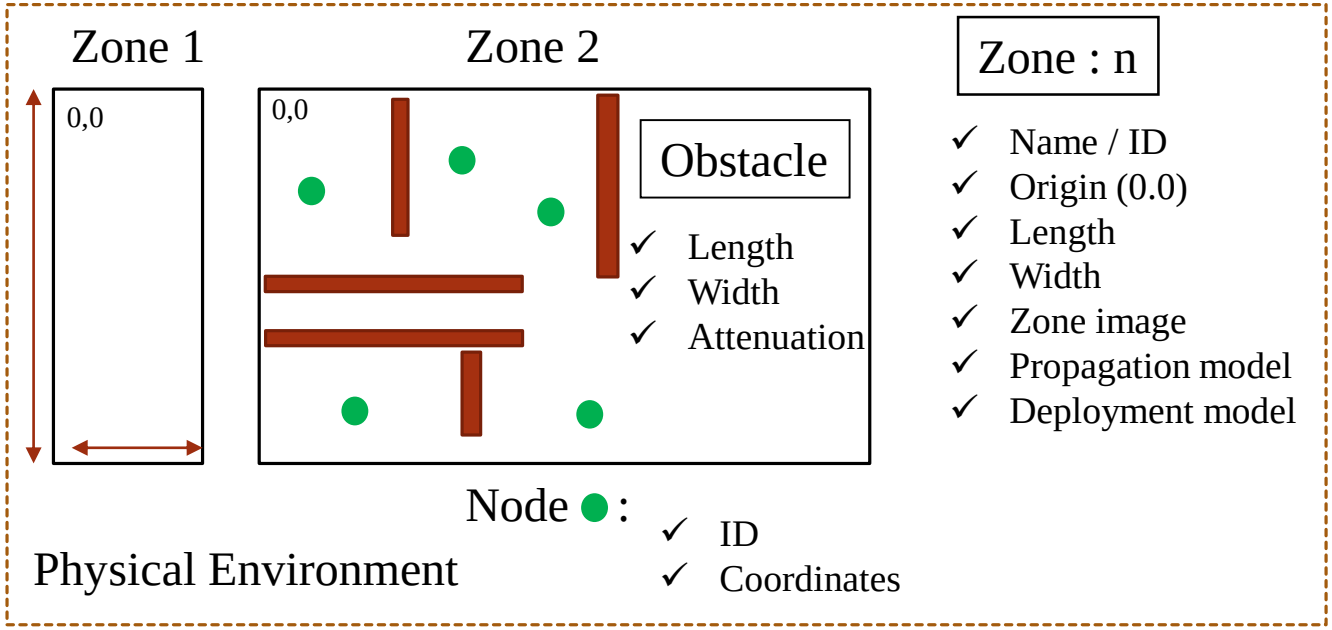
The Physical Environment Facet

Objectives:

- Describe the deployment environment
- Locate the nodes in the environment
- Study the influence of the environment on the network (obstacles, propagation model...)
- Correlate the data with the environment

Items of the model:

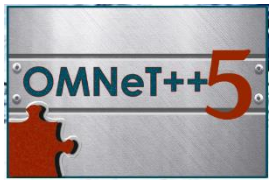
- Physical areas,
- obstacles,
- node instances



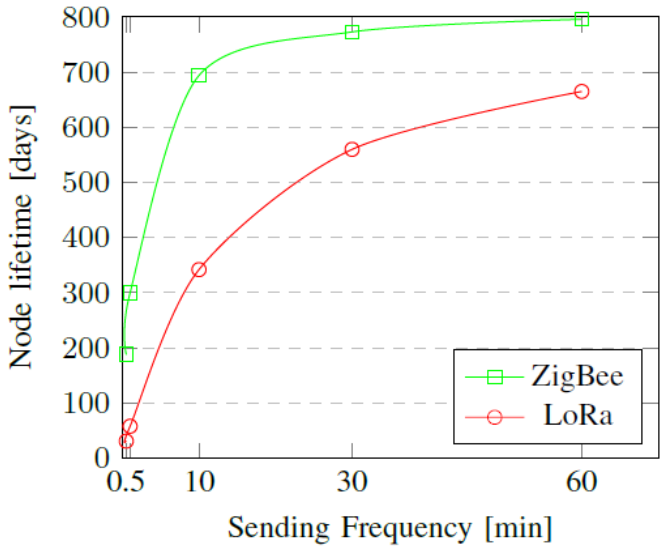
Network Sizing : Simulation

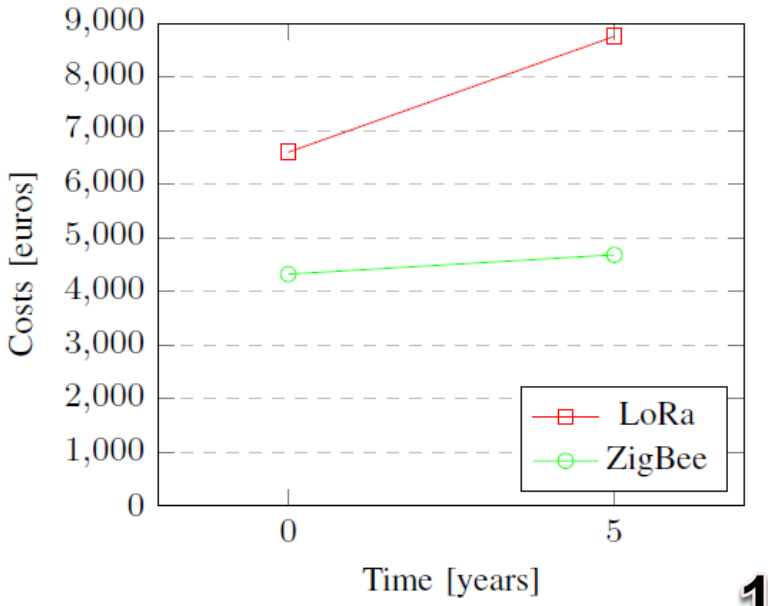
Our framework generates the Omnet++ simulation files from the facets:

- NED file for topology
- C++ file for behavior
- File omnet.init

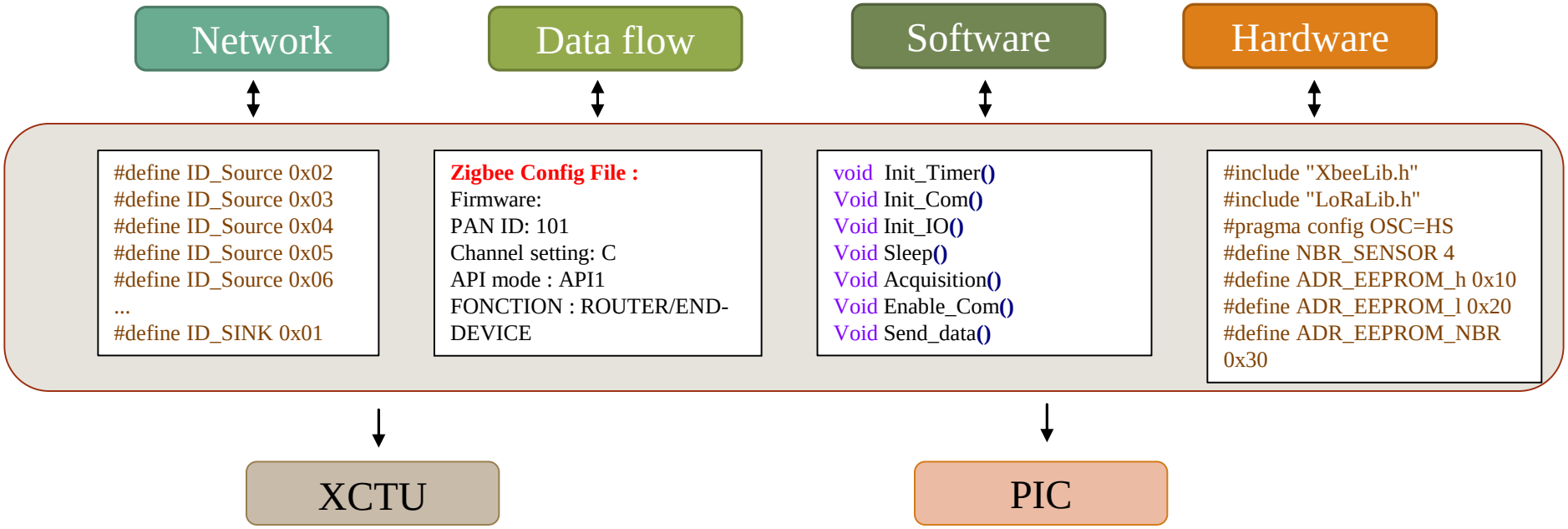



```
volatile double sTimeSleeping @unit(s) = default(60s);
volatile double sTimeSensing @unit(s) = default(2s);
volatile double sTimeSending @unit(s) = default(0.12s);

volatile double BatterySize = default(1200);       // (mAh)

volatile double consoSleeping = default(0.08);    // (mA)
volatile double consoSensing = default(1.98);     // (mA)
volatile double consoSending = default(40);       // (mA)
```



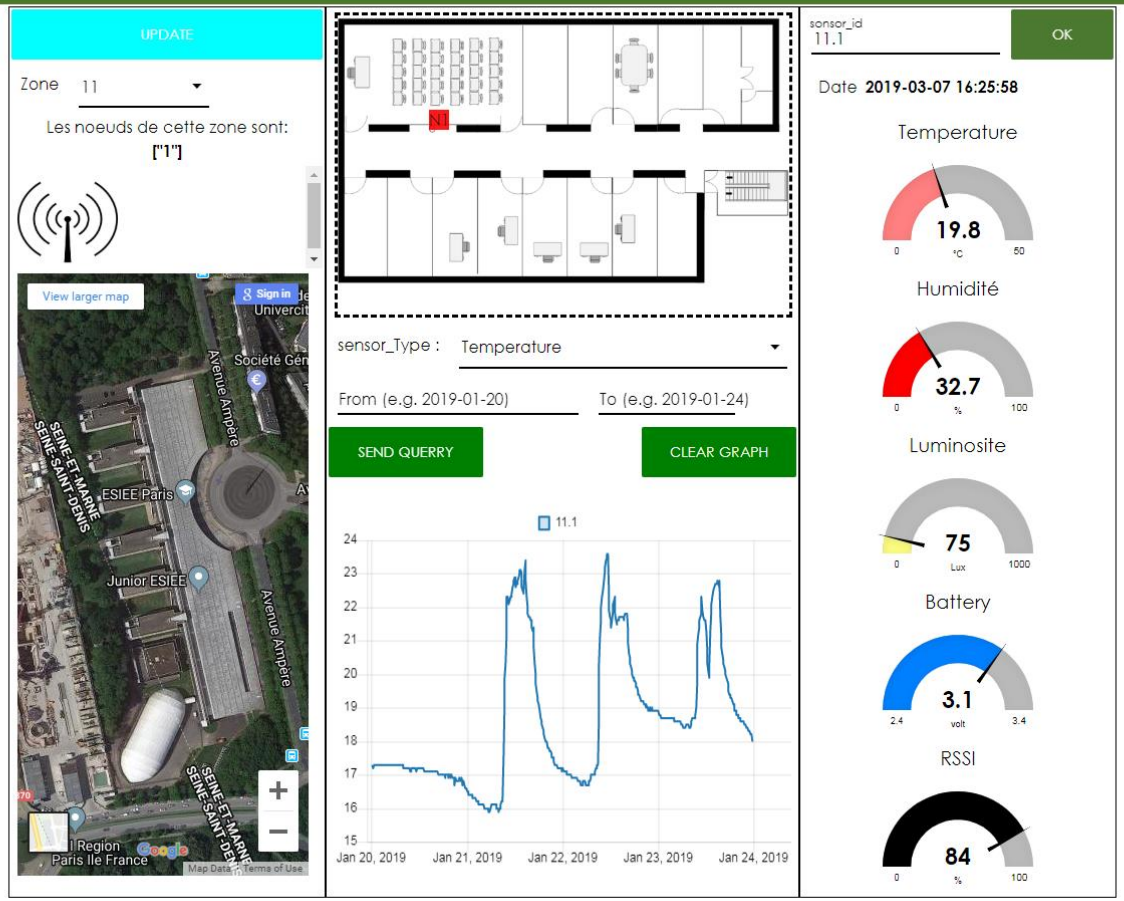


Code Generation



Data Exploitation

- **User interface** for data exploitation is based on PHP scripts for data visualization.
- Generate node-red script



Conclusion & Perspectives

- We provide framework for network sensor design/exploitation
- Validated on real case study
- Need to be enhance :
  - more facets : maintenance, security aspects,…
  - more targets for code generator
  - database generation

- Need to handle interoperability of sensors networks from different suppliers
- More case study : outdoor,…