# EMIT

## a system for monitoring networks of MQTT clients

Jérôme Rocheteau
AeLoS Seminar
2018-06-21

# EMIT

Overview

1. History
   - the « Code Vert » project
   - the MEASURE project
2. Functionalities
   - Prerequisites on MQTT
   - Configuring MQTT clients
   - Monitoring MQTT clients
3. Challengers
   - Eclipse MQTT Spy
   - IBM Watson IoT
4. Prospects

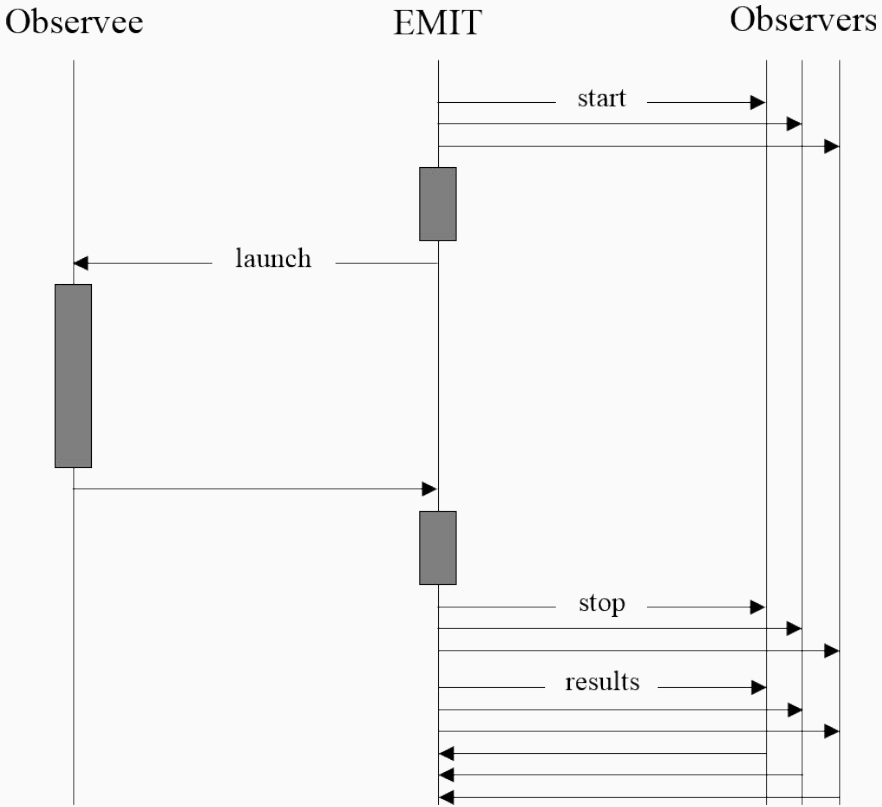
# Code Vert (2012-2014)

## Objectives & Means

- Energy Efficiency of Best Coding Practices
- Energy Consumption Difference (see [1])
  - Best Coding Practice
  - Not-that-Best Practice
- Fine-Grained Power Measurements

## Power Meters

- Slave Mode (On-Demand)
  - Start Measuring
  - Stop Measuring
  - Retrieve Measurements




[1] J. Rocheteau, V. Gaillard, L. Belhaj, How Green are Java best Coding Practices? SMARTGREENS, 2014.

# MEASURE (2016-2019)

## **EMIT** – **E**nergy **M**on**IT**oring (2017)

### Objectives & Means

- SEC Metrics in Production Phase
- Continuous Power Measurements

### Power Meters

- Connected Objects (MQTT clients)
- EMIT = MQTT client monitoring system

### Feedback

- Reliable and Flexible Architecture
- Generic vs Specific

SEC: Software Energy Consumption

# EMIT

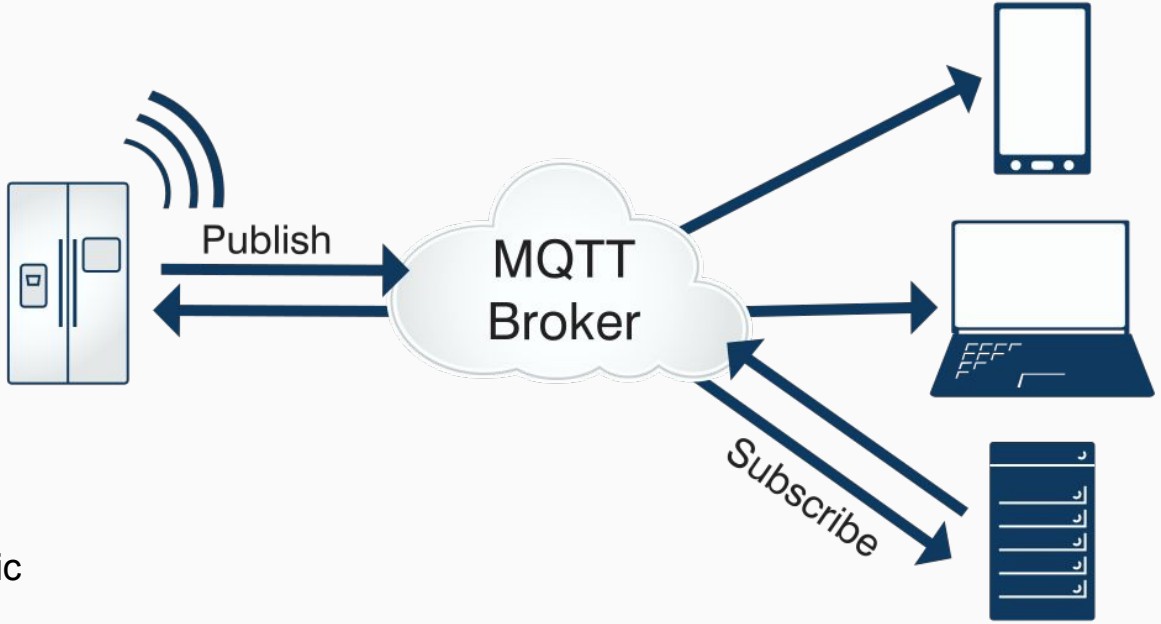
Functionalities

- MQTT
- Configuring
  - Brokers
  - Clients
  - Callbacks
- Monitoring
  - Controls
  - Records
  - Messages
- Dashboard

## MQTT

Message Queuing Telemetry Transport

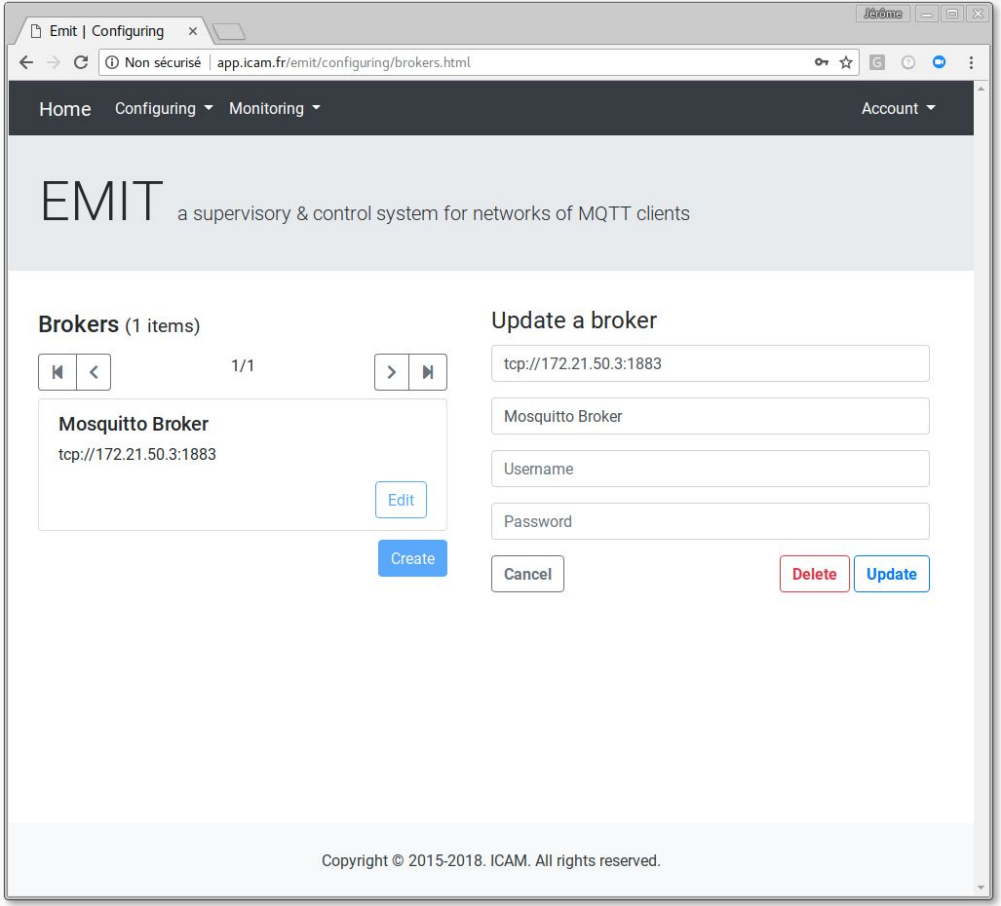
- open messaging protocol
- TCP/IP communication
- ISO standard
- publish / subscribe pattern




## MQTT Client Status

- connect (disconnect) to a broker
- subscribe (unsubscribe) to a topic
- publish a message to a topic

# EMIT




## Configuring

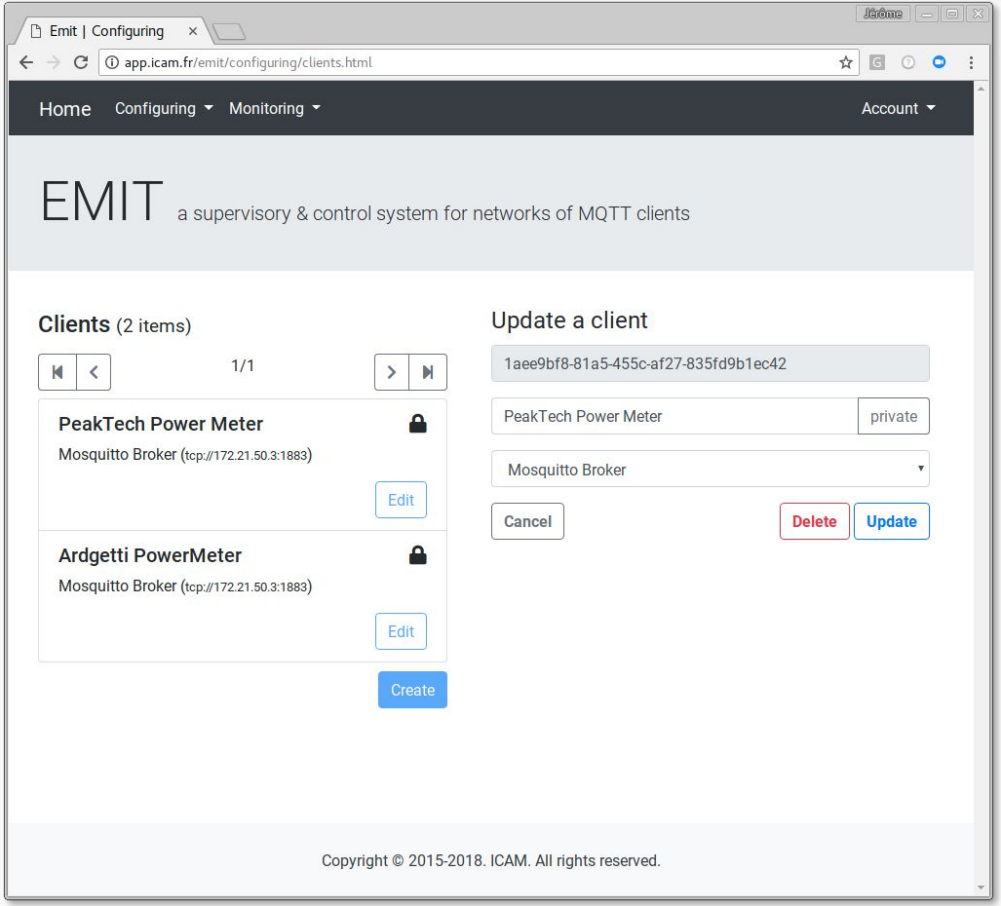
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

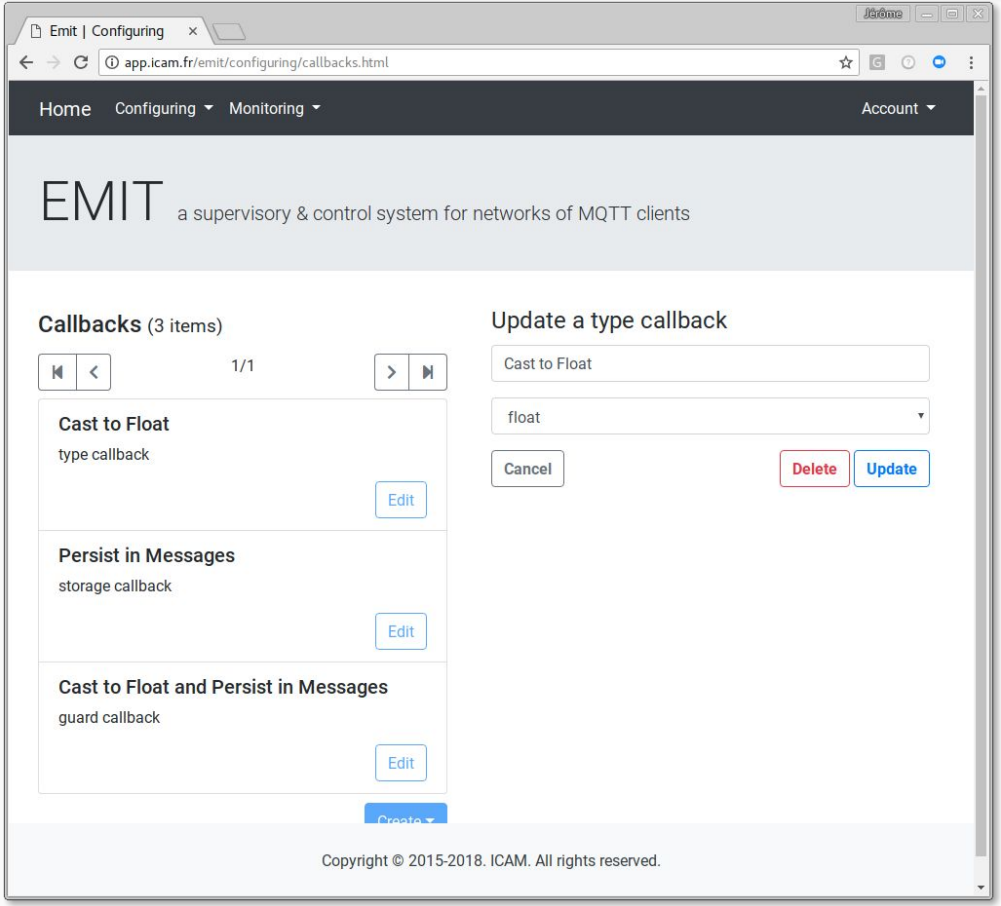
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

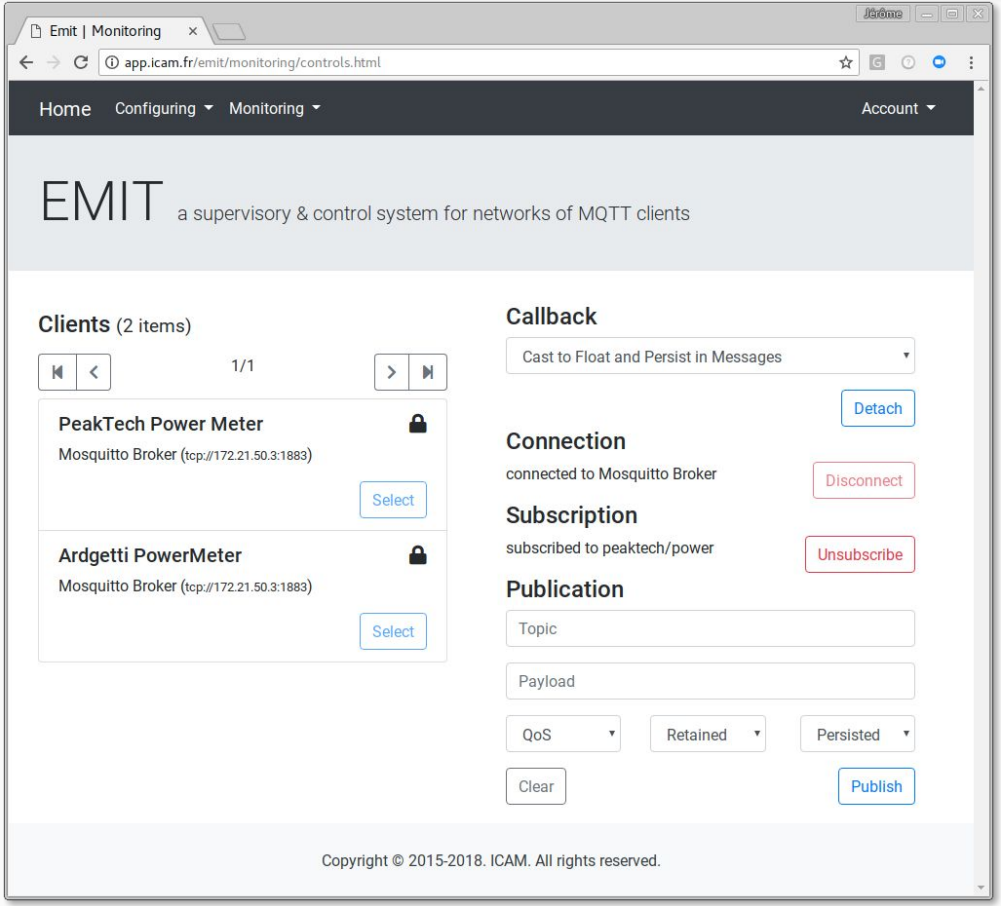
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

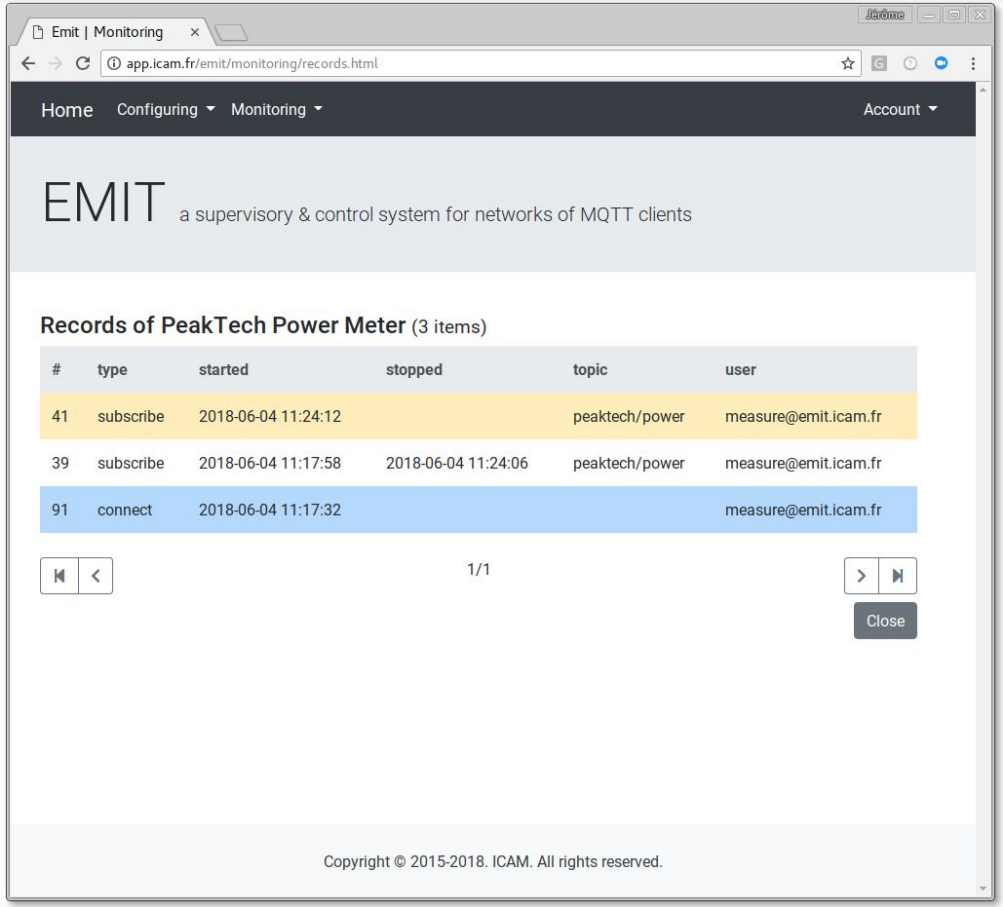
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

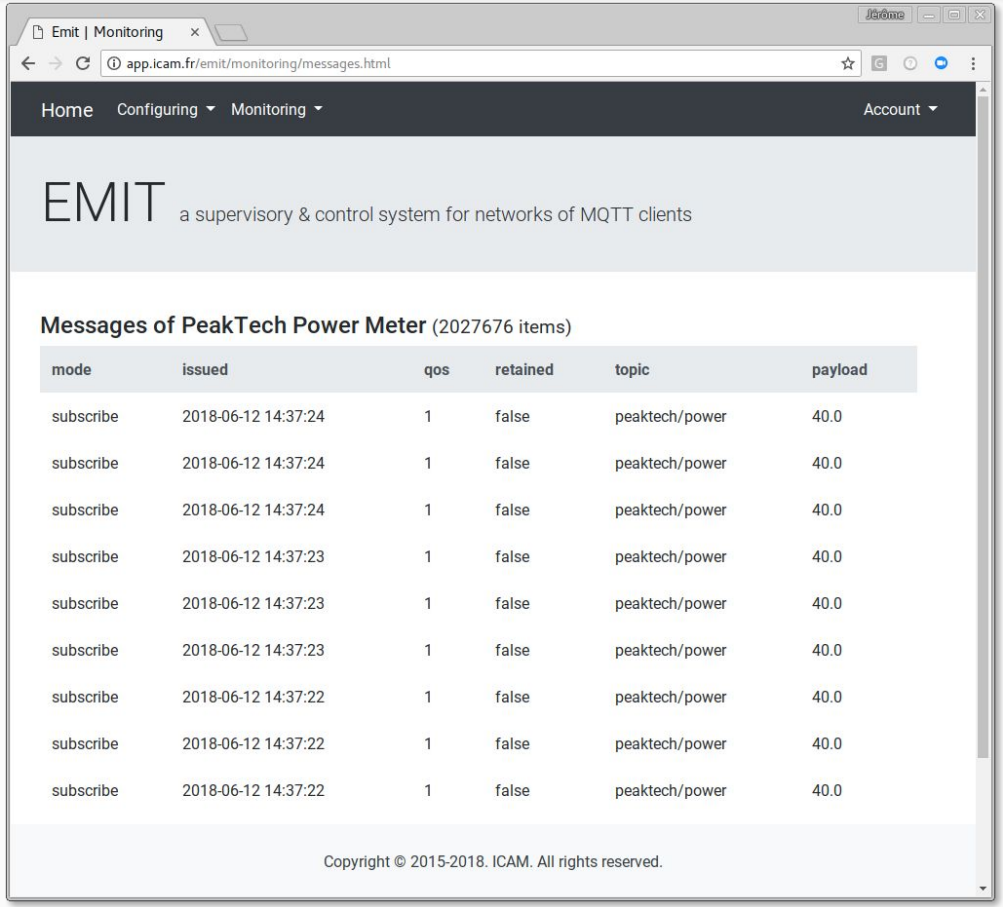
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

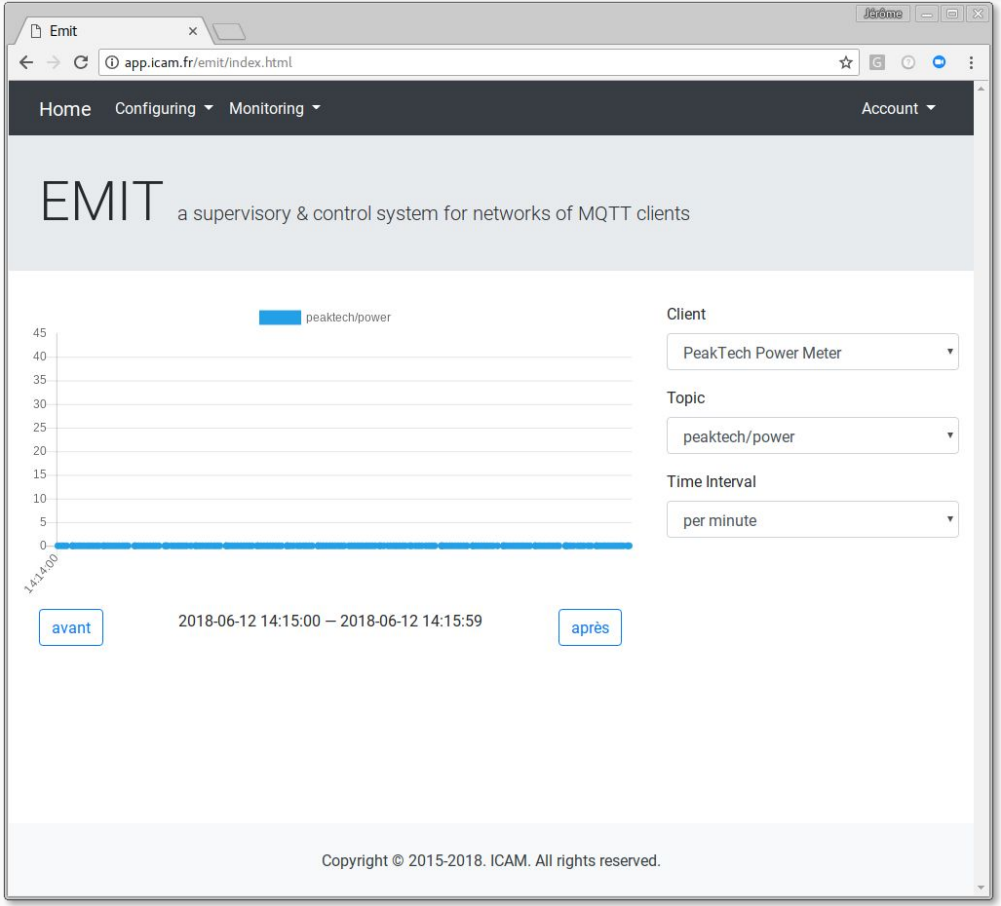
- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

## Dashboard

# EMIT




## Configuring

- Brokers
  *edit MQTT brokers*
- Clients
  *edit MQTT clients*
- Callbacks
  *edit message processing*

## Monitoring

- Controls
  *update MQTT client status*
- Records
  *view MQTT client status updates*
- Messages
  *view messages*

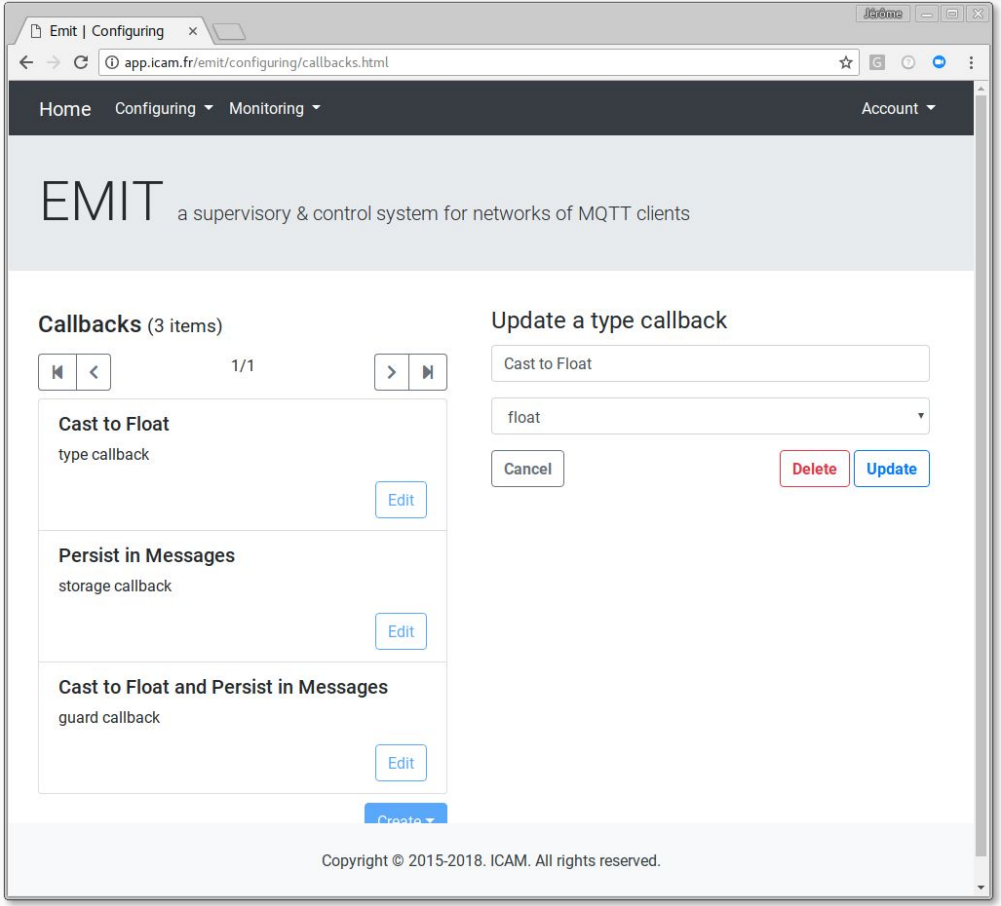
**Dashboard**

# EMIT

Callbacks

- Types
- Topics
- Storages
- Features
- Guards
- *Sequences*

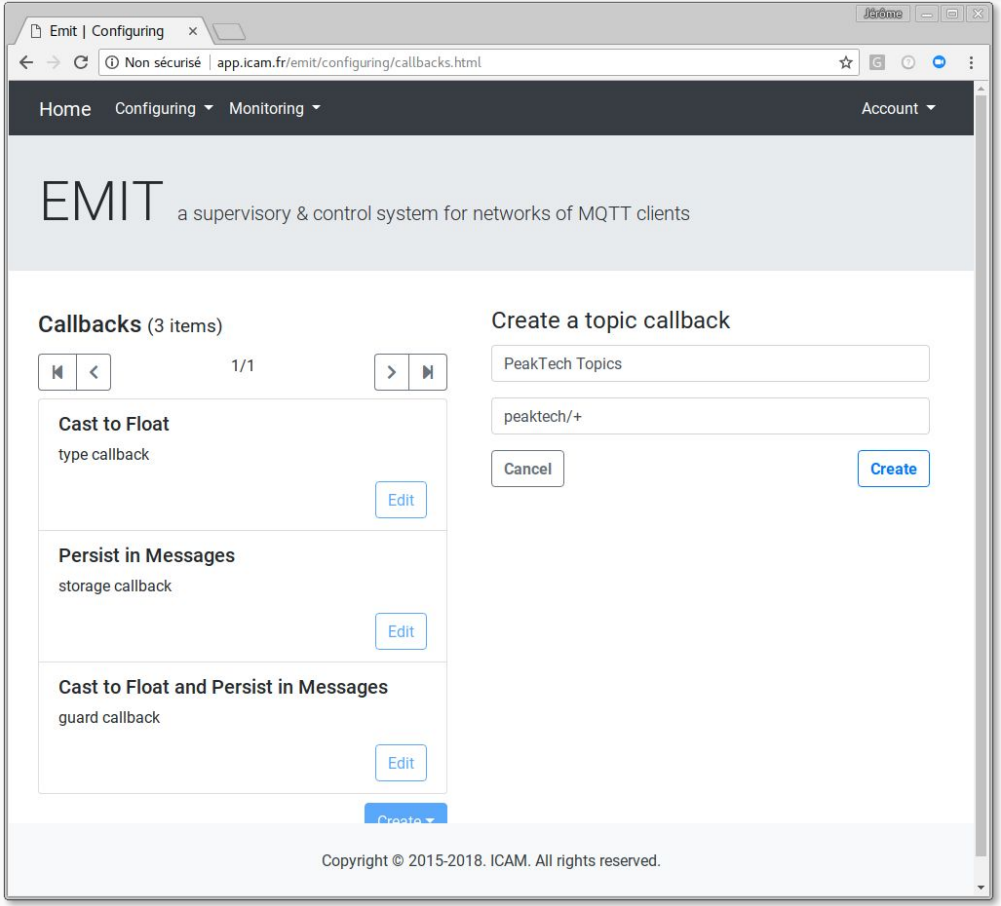
# EMIT




## Callbacks

- Types
- Topics
- Storages
- Features
- Guards
- *Sequences*

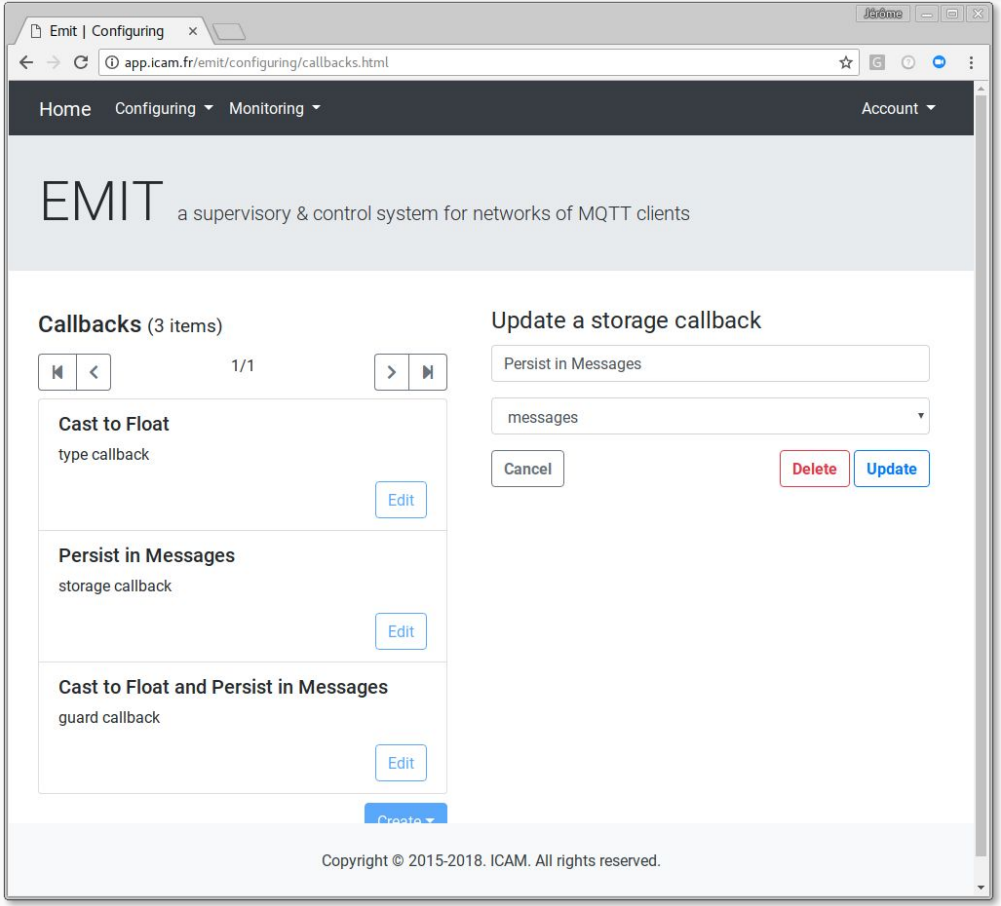
# EMIT




## Callbacks

- Types
- Topics
- Storages
- Features
- Guards
- *Sequences*

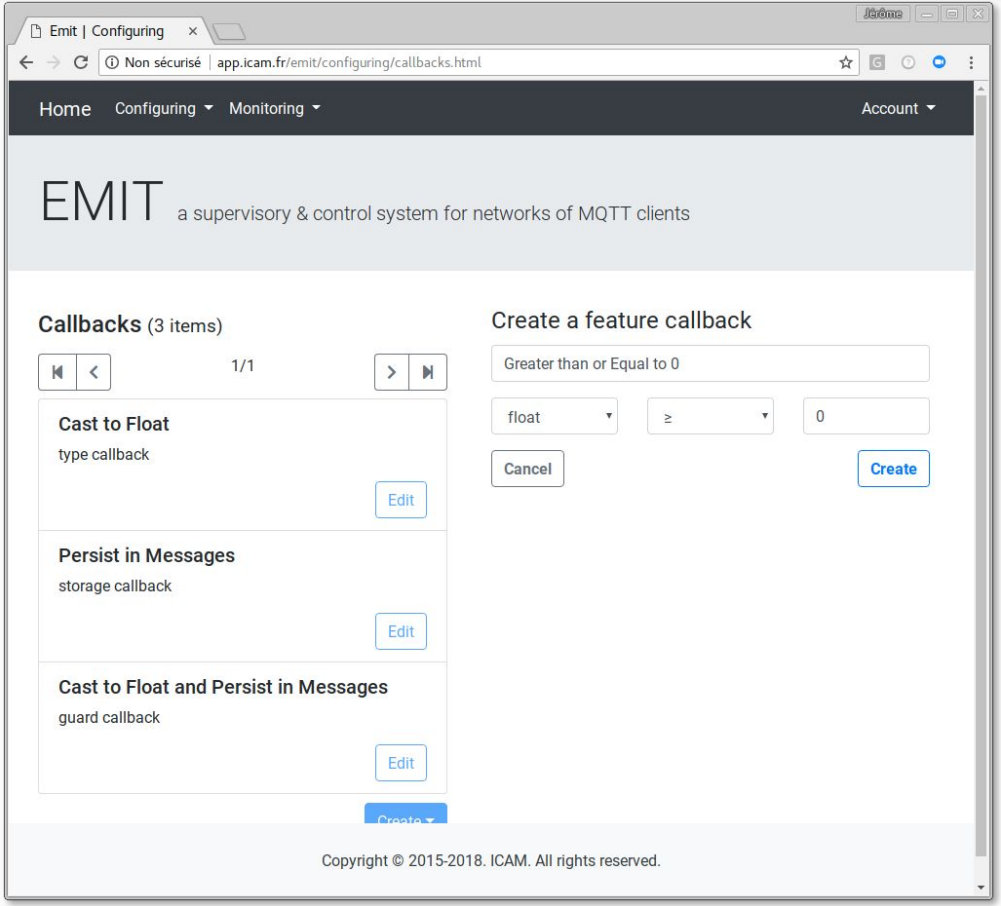
# EMIT




## Callbacks

- Types
- Topics
- Storages
- Features
- Guards
- *Sequences*

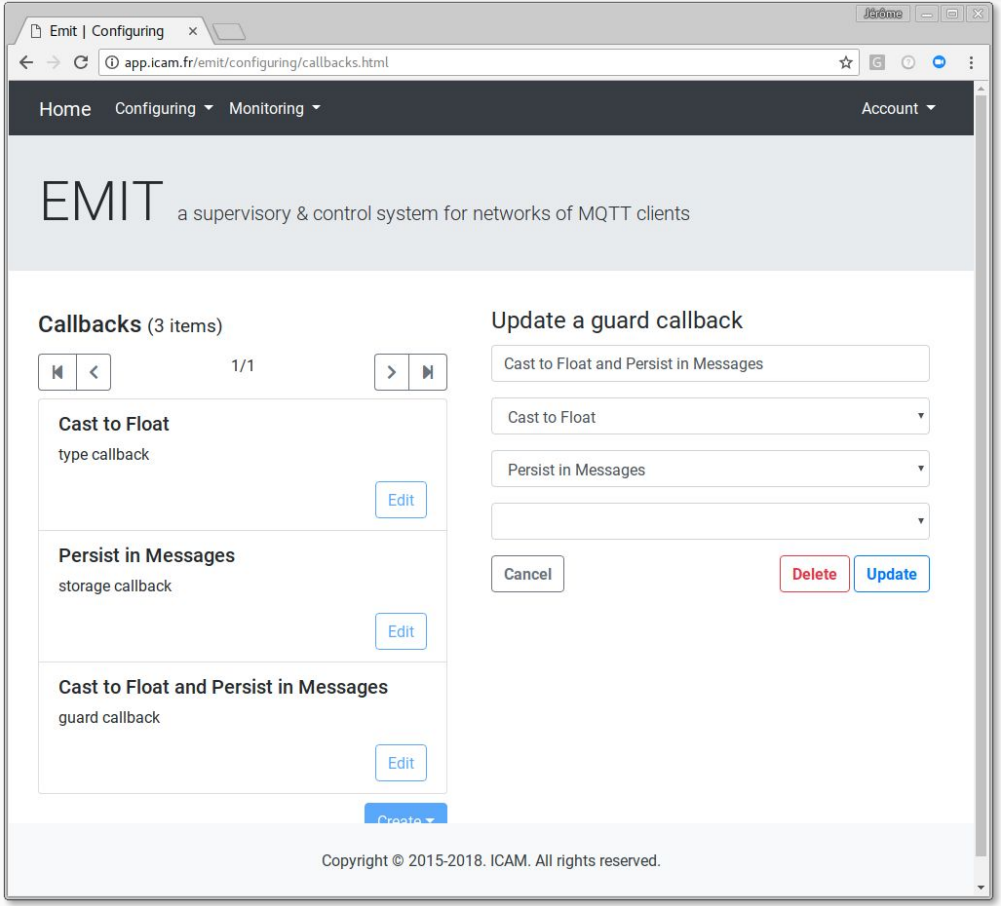
# EMIT




## Callbacks

- Types
- Topics
- Storages
- **Features**
- Guards
- *Sequences*

# EMIT




## Callbacks

- Types
- Topics
- Storages
- Features
- Guards
- *Sequences*

# EMIT

Challengers

- Evaluation Criterion
- Eclipse MQTT Spy
- IBM Watson IoT

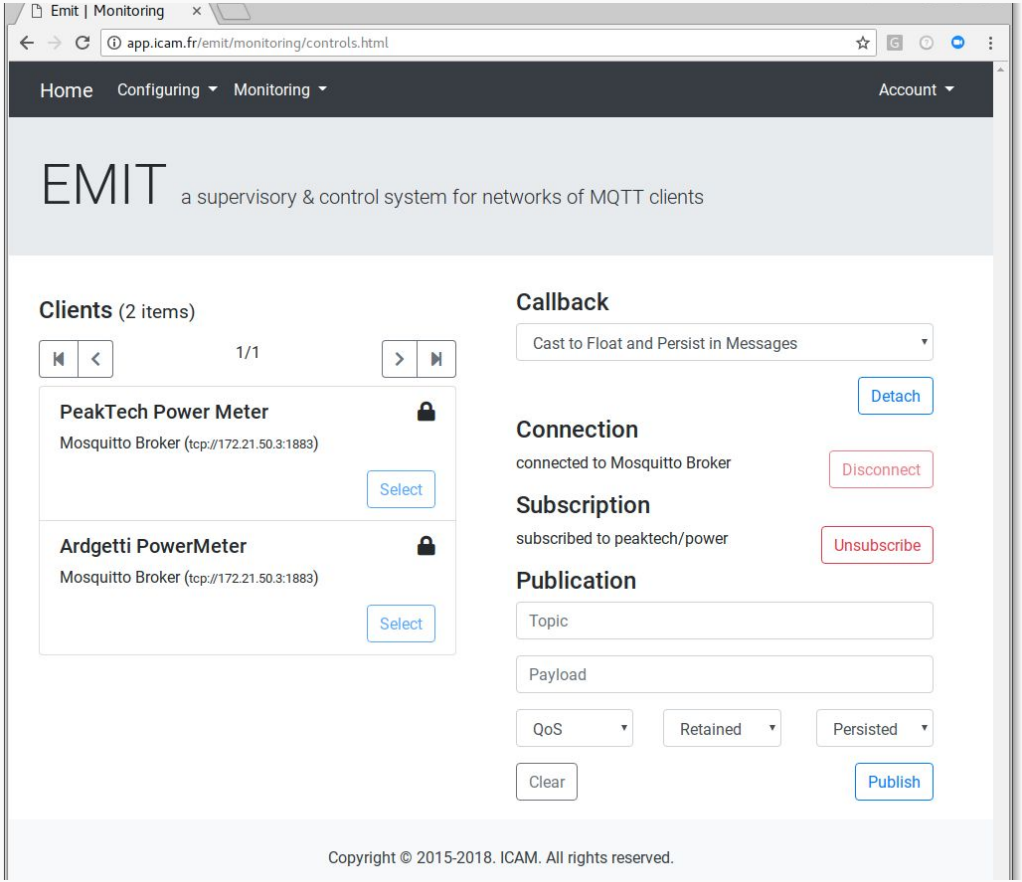
# EMIT

## Security
- ✔ Authentication
- ✘ SSL / TLS

## Communication
- ✔ Automatic Reconnect
- ✔ TCP
- ✘ Websocket

## Messages
- ✘ Filtering
- ✘ Statistics
- ✔ Processing

## Usage
- ✔ Application
- ✔ HTTP API



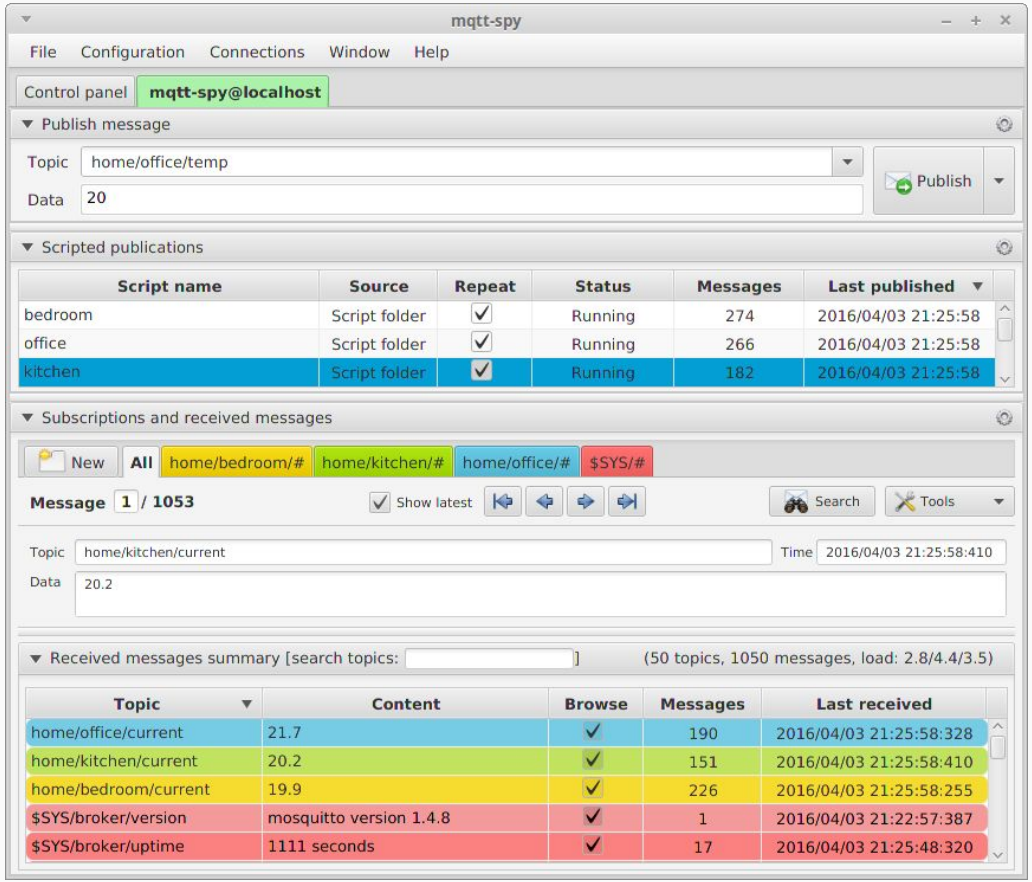
# Eclipse MQTT Spy

## Security
- ✔ Authentication
- ✔ SSL / TLS

## Communication
- ✔ Automatic Reconnect
- ✔ TCP
- ✔ Websocket

## Messages
- ✔ Filtering
- ✔ Statistics
- ✔ Processing

## Usage
- ✔ Application
- ✘ HTTP API



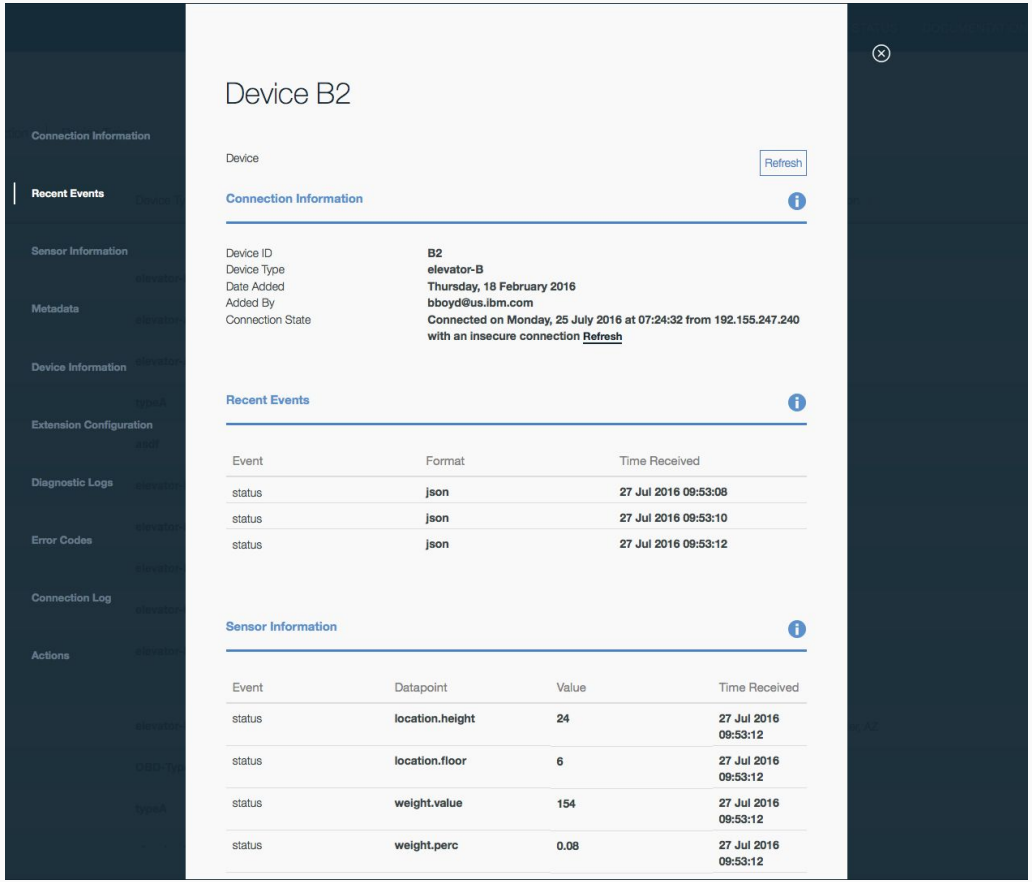
# IBM Watson IoT

## Security
- ✔ Authentication
- ✔ SSL / TLS

## Communication
- ✖ Automatic Reconnect
- ✔ TCP
- ✖ Websocket

## Messages
- ✔ Filtering
- ✔ Statistics
- ✖ Processing

## Usage
- ✔ Application
- ✔ HTTP API



# EMIT

Prospects

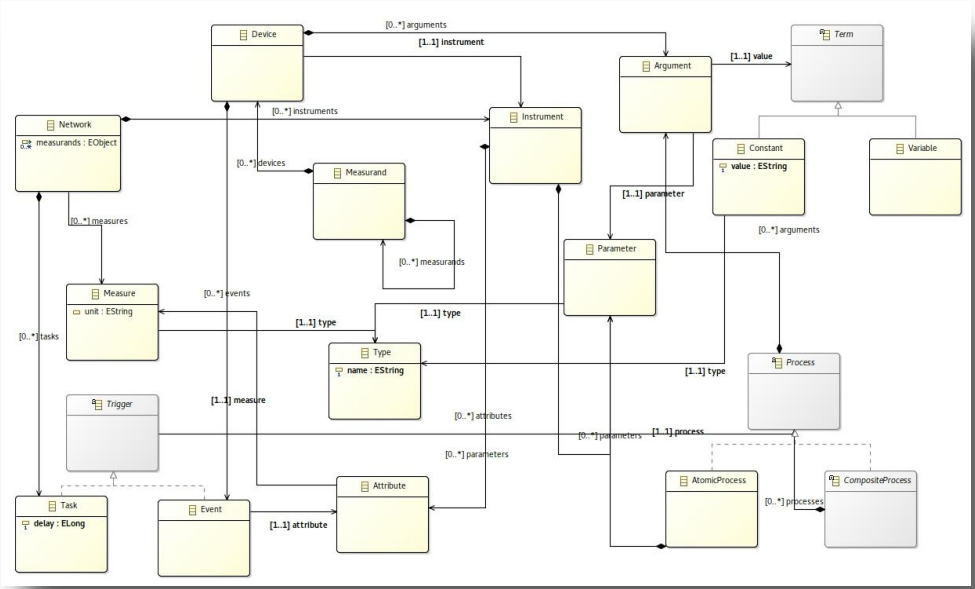
# Prospects

## Callback System

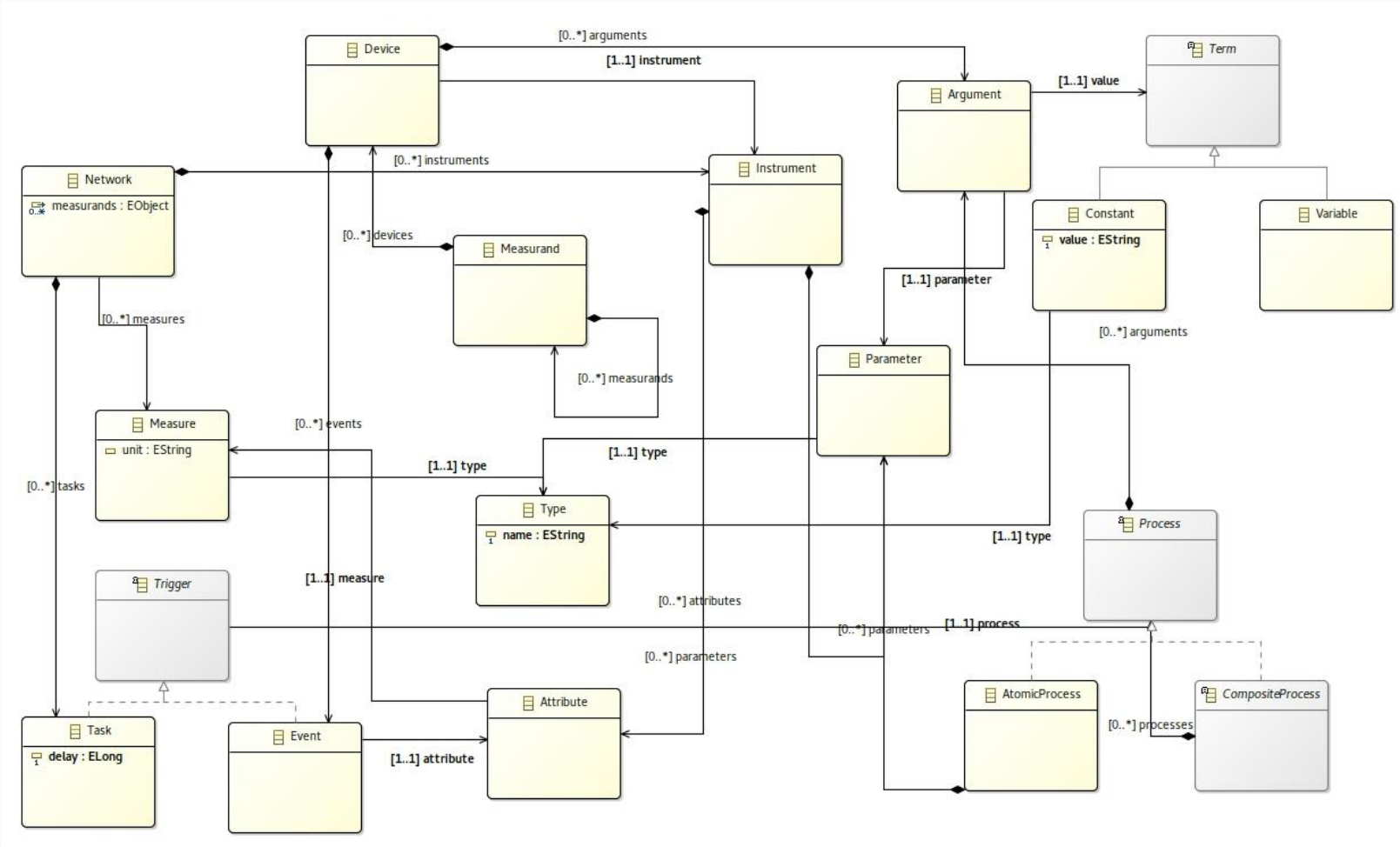
- Callback for Status Updates
  - connect / disconnect a client
  - subscribe / unsubscribe to a topic
  - publish a message to a topic
- Extensible Integration System

## Model-Driven IoT Engineering

- Models of IoT Systems
- Validation Support
  - Client and Callback Compliance
  - Requirements over Status Updates
- Configuration Generation
  - EMIT
  - IBM Watson IoT




**Thank you!**


Device
[0..*] arguments
[1..1] instrument
Argument
[1..1] value
Term
Network
measurands : EObject
[0..*] instruments
Instrument
Constant
value : EString
Variable
[0..*] devices
Measurand
[1..1] parameter
[0..*] measures
[0..*] arguments
[0..*] measurands
Parameter
Measure
unit : EString
[0..*] events
[1..1] type
[1..1] type
[0..*] tasks
Type
name : EString
Process
[1..1] type
Trigger
[1..1] measure
[0..*] attributes
[0..*] parameters
[1..1] process
[0..*] parameters
Attribute
AtomicProcess
CompositeProcess
Task
delay : ELong
Event
[1..1] attribute
[0..*] processes