UMONS
Université de Mons

Hybrid communication architecture for time sensitive smart-metering

WSPLC 2016 EDF-X-RENESAS

Student Challenge

Ir Aurélien Van Laere, Ir Gaston Bayot Katumba

David Hauweele, Jérémy Dubrulle

(firstname.name@umons.ac.be)

Faculté des Sciences

POLYTECH MONS

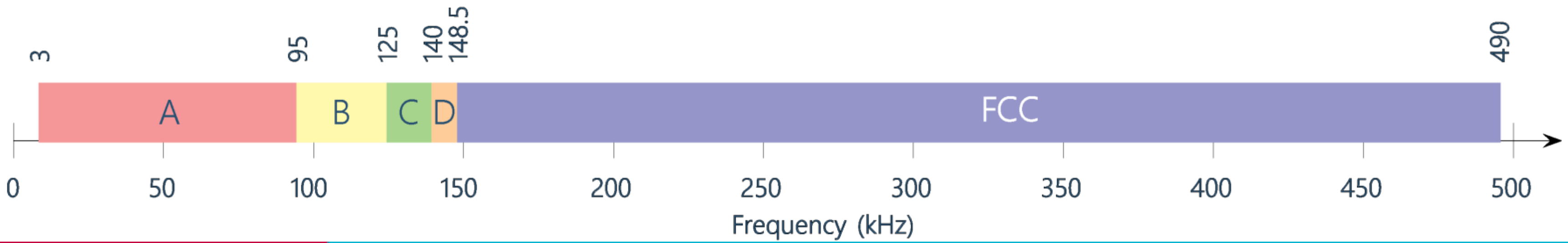
Service d'Electromagnétisme et de Télécommunications

Outline

- Hybrid architecture
  - Introduction
  - Implementation
- Smart-metering application
  - Introduction
  - Implementation
- Demo
- Conclusion

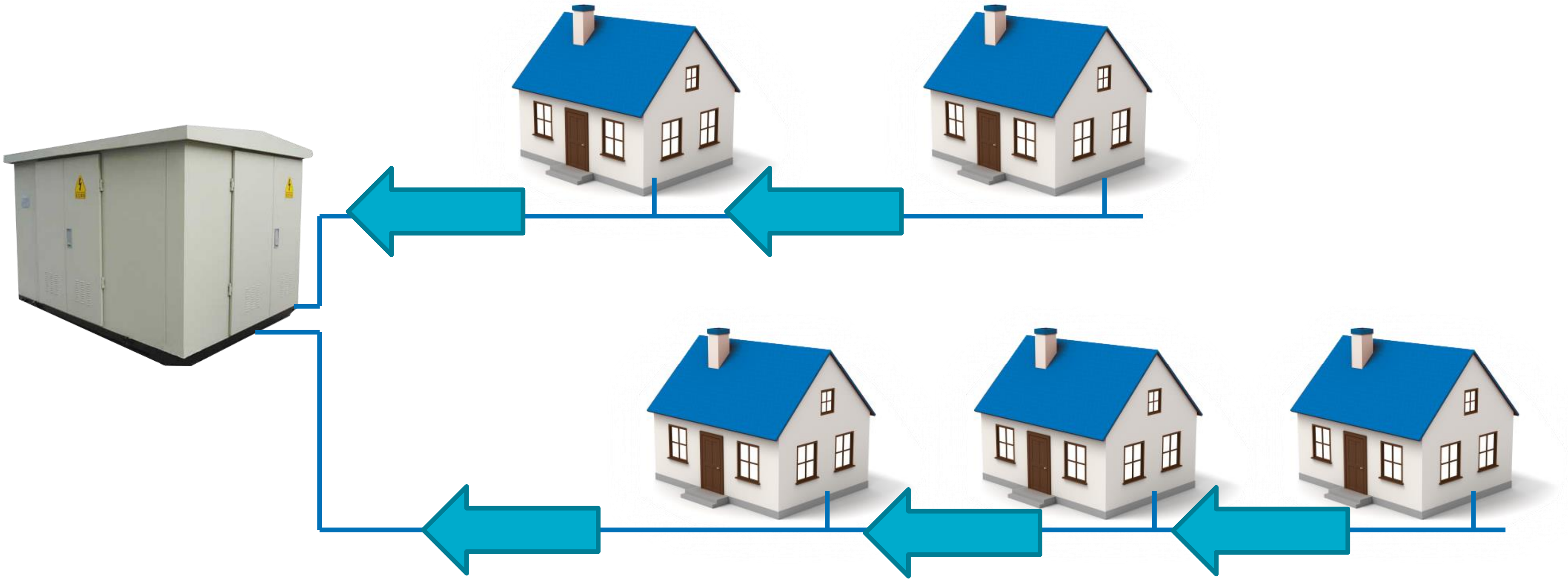
PLC is a technology usable for Smart-Metering

- Smart-metering → snapshot of the distribution grid for DSO's
  - Prevents power outage
  - Plans future grid investment
  - Gives better access and feedback to their clients
- Different frequency bands
  - Mainly G3-PLC
  - Depending on the country, multiple frequency bands are defined for smart-metering use:
    - CENELEC-A (Europe): 3-95 kHz
    - FCC (USA): 100-500 kHz



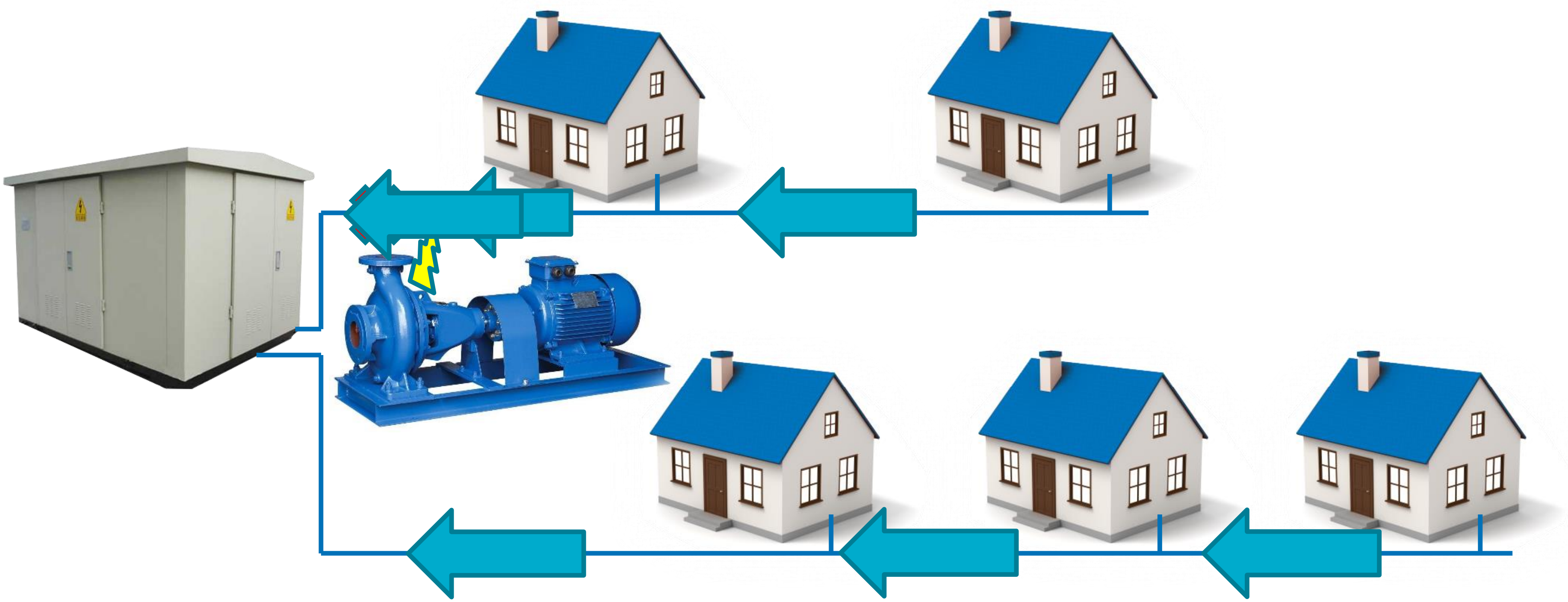
Smart-metering with G3-PLC

- Data concentrator placed in the MV/LV transformer cabinet :
  → control a subnet of G3-PLC devices (PAN)
- PAN coordinator :
  - Administer every power meter
  - Gateway to DSO private IP network
- Each power meter can act as a repeater
  → Forward packets to coordinator



End-to-end connectivity in G3-PLC

- Every node is a repeater
- Routes are dynamically reconfigured
- Link quality → adapts modulation robustness
- Retransmissions possible at every hop



Problem with this approach :

This behavior is robust but can be very slow

- Smart-metering aims to provide quasi-realtime distribution grid status to DSO’s, but :
  - problematic links → long delays
  - inconsistencies in short term grid state estimation

Why it matters ?

- New time-sensitive jobs could be added in the future :
  - Control of household appliances could be done remotely
  - Coordination of islanded grid operation
  - …

Solution : hybrid communication scheme

- Provides more reliable timings
- Uses a second wireless interface and some small modifications to the G3-PLC network
  - Each modem is equipped with a LoRa interface
  - Reduced max. retransmissions for packets

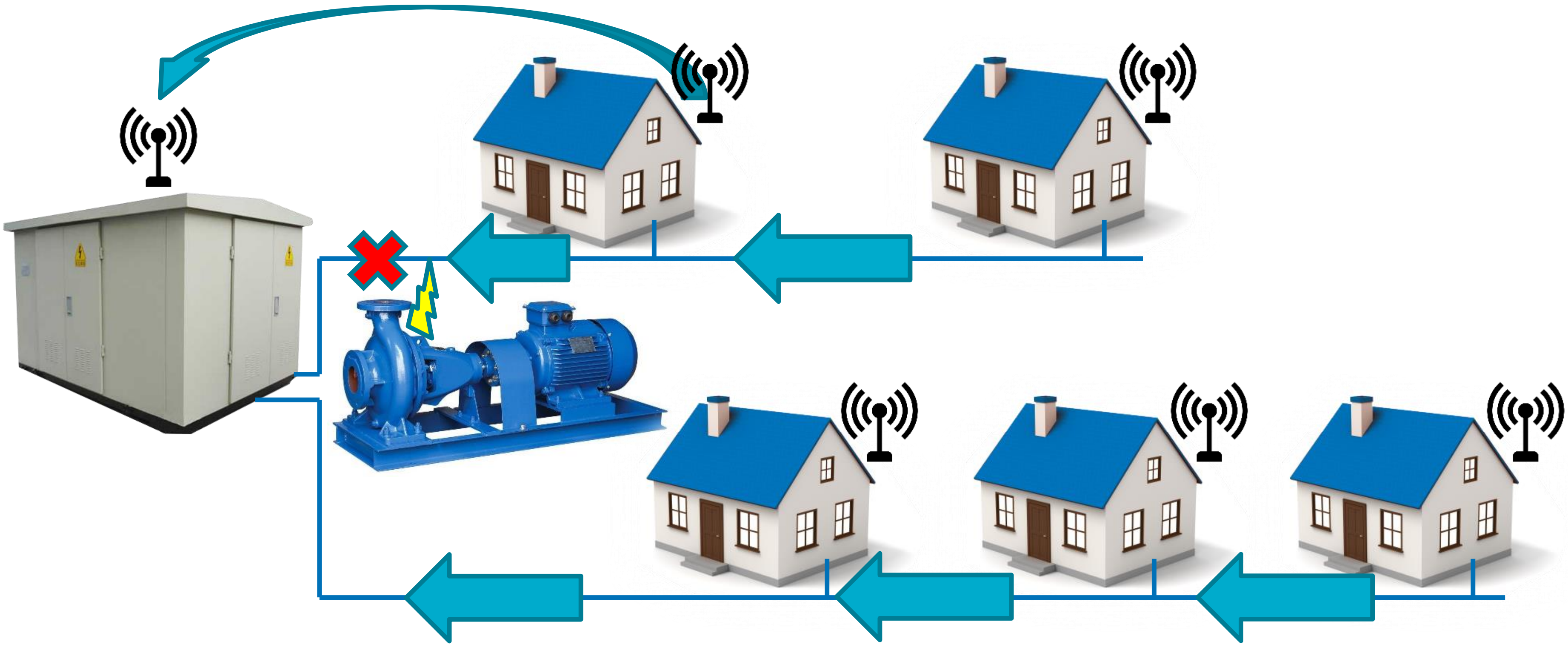
- G3-PLC still used as the default interface but:
  - After the maximum number of retransmission is reached, the packet is forwarded using the second interface to minimize the overall delay

Solution : hybrid communication scheme

- G3-PLC :
  - Higher datarate
  - No channel occupation time limitation
  - Multi-hop for farthest nodes

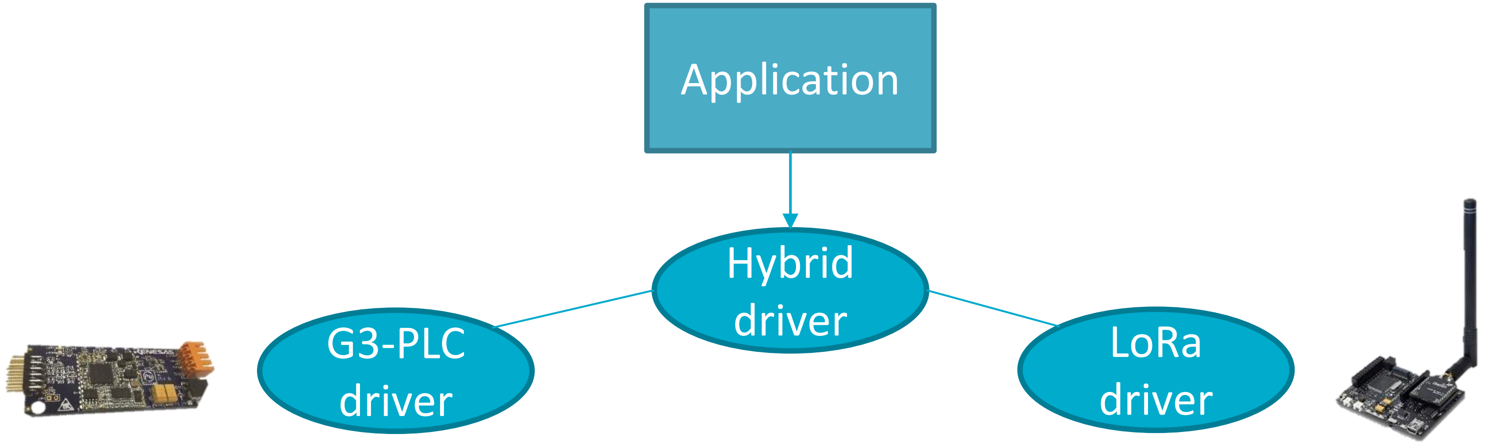
- LoRa :
  - Low datarate
  - Long range of operation and robustness
  - Limited channel occupation time
  - Single-hop communication (no route)

Solution : hybrid communication scheme



Proposed implementation

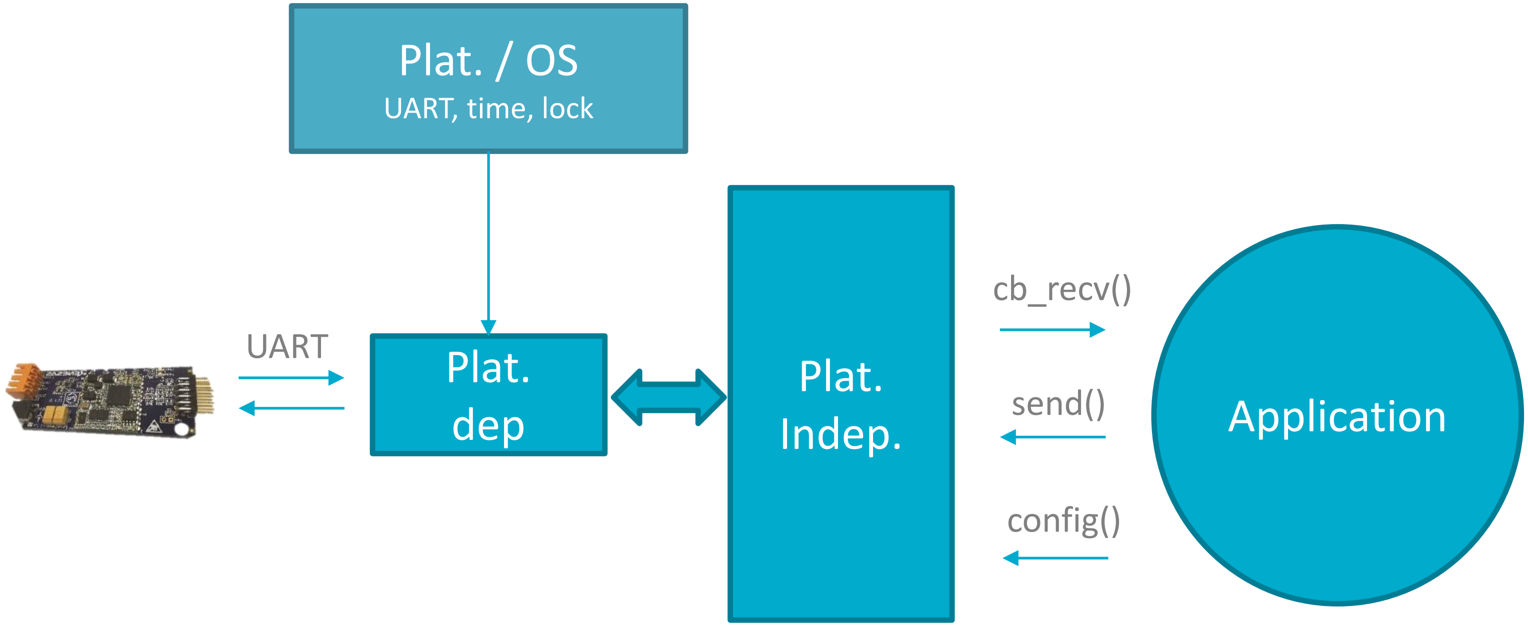
- One driver for each interface
  - Suppose a fixed MTU of 63 bytes
- Hybrid driver
  - On top of real interfaces
  - Handle fallback to second interface
  - Transparent to application layer



Driver implementation

- We want to use our implementation on other platforms
  - Renesas, RPi (Linux), our own machine, …
  - More about this later …
- Driver divided in platform-dependent/independent parts
- Platform independent :
  - Pure C
  - Interface for application layer
  - Contains all logic of driver (+ MAC logic)
- Platform dependent :
  - Low-level interface to the platform
  - Basic UART, timers, locking mechanisms, …
  - Mostly interfacing (small fraction of the code)

Driver implementation

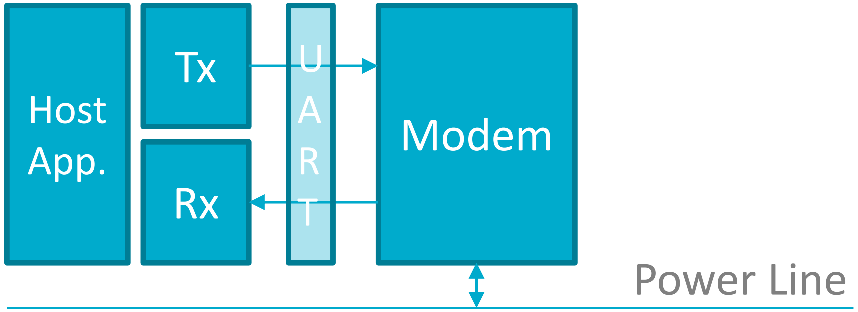


Advantages :

- Interoperable between platforms
- Provide more structure to the code
- Offline testing

Driver implementation : G3-PLC

- Use any UART
- Modem flashed at each start
  - Reset pin must be held down during 30ms
  - Reset and upload handled by the driver
- All primitives necessary to control the modem are implemented inside the driver
  - Modem messages are handled by an independent thread
  - The communication mode used by the modem must be set statically in the source code (MAC mode in our tests)



Driver implementation : LoRa

Similar to G3-PLC but :

- LoRaMAC modules are broadcast-only
- Does not meet the requirements of our architecture

MAC layer :

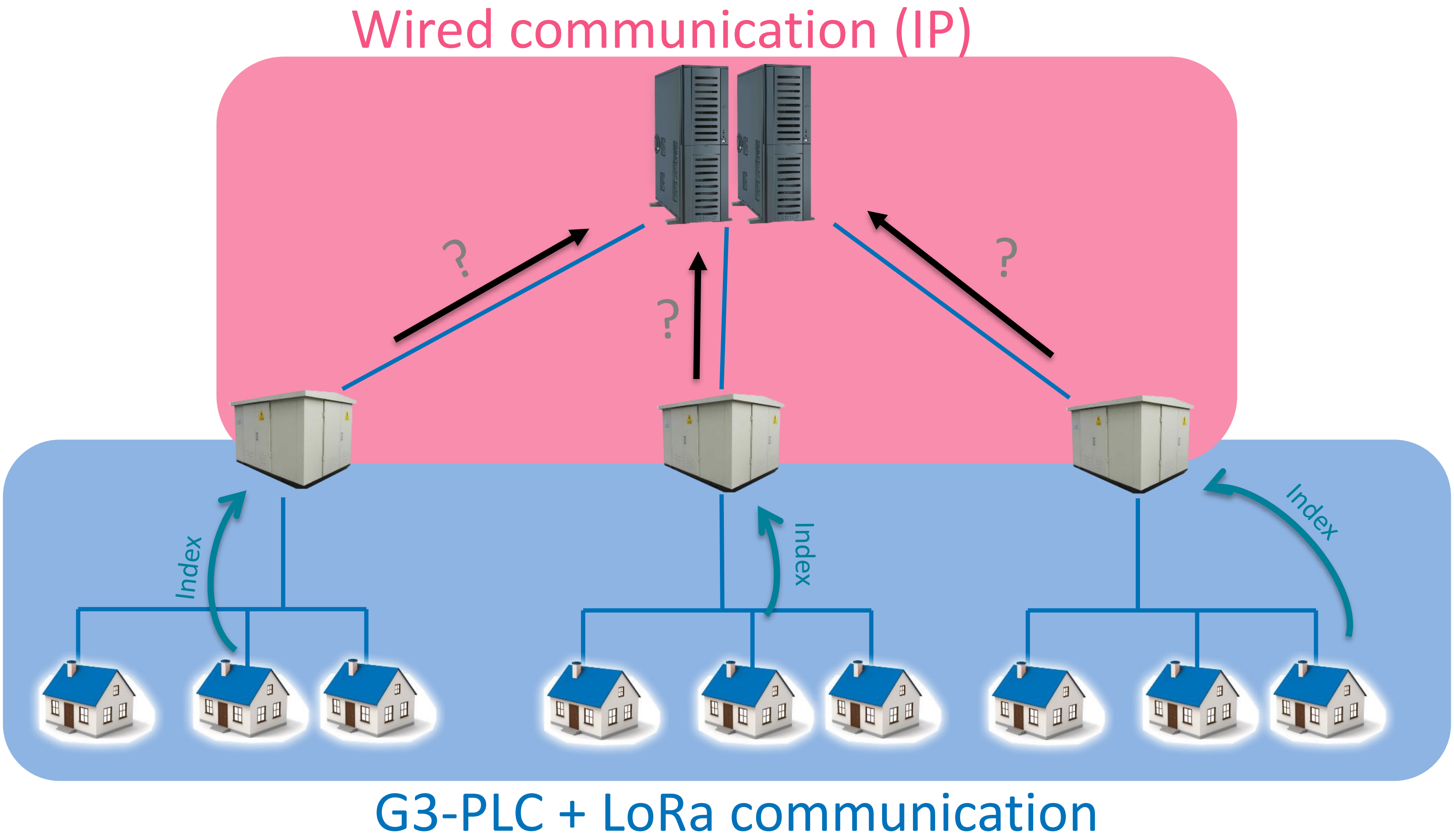
- Unicast communication
- Short MAC addresses
- Frame acknowledgment with retransmissions
- CRC

Smart-metering Application

- Data collection
  - How to achieve quasi-realtime ?
  - Which protocol ?

- Data analysis
  - How to provide access to data ?
  - How to allow offline & realtime analysis ?

Smart-metering : Data Collection



MQTT

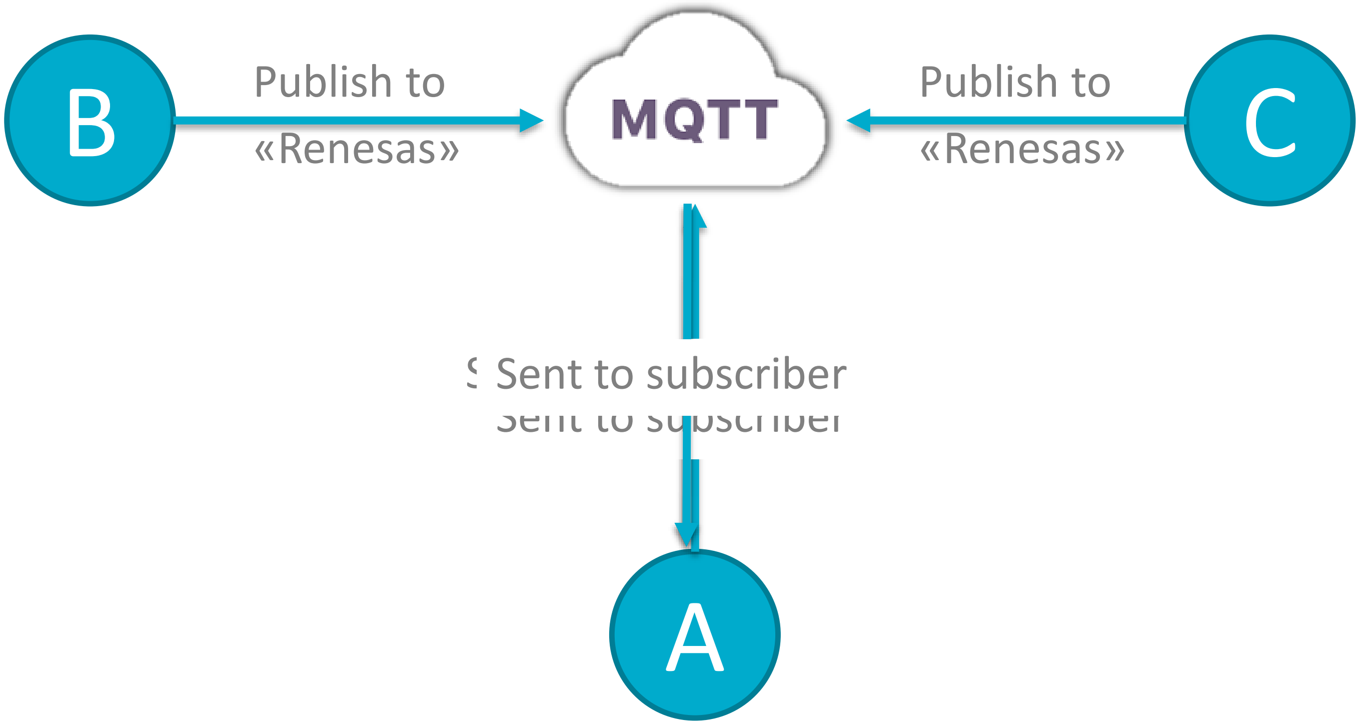
Message queuing and transport protocol

- Lightweight
- Open-source
- Suited for transport of telemetry data
- Publish / Subscribe model

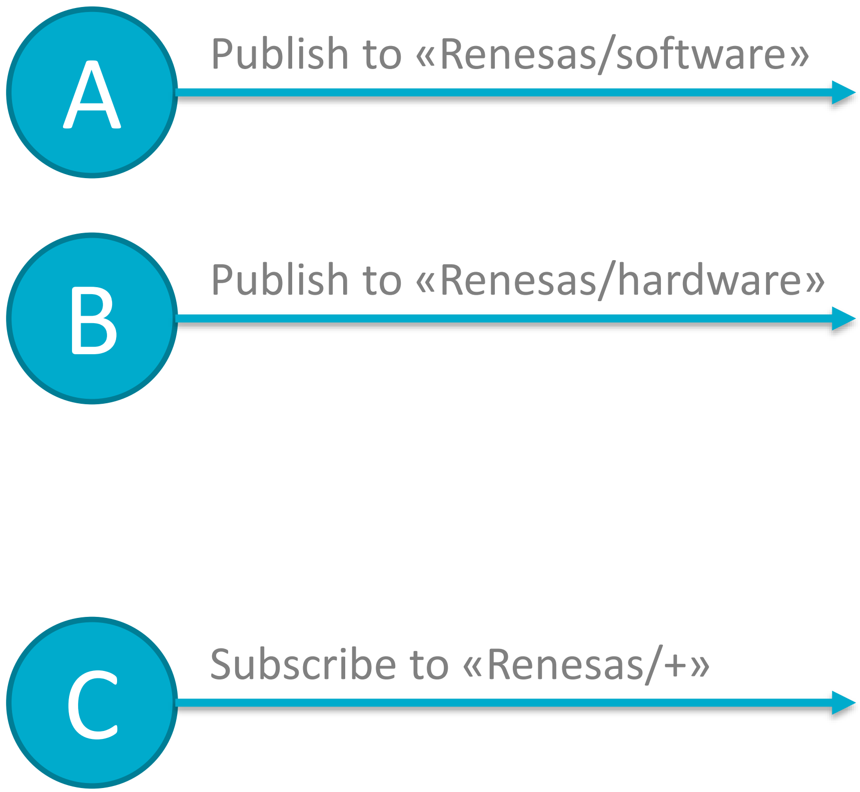
0 1 2 3	4	5 6	7
Message Type	UDP	QoS Level	Retain
Remaining Length (1~4 bytes)			
Variable Length Header (Optional)			
Variable Length Message Payload (Optional)			

MQTT : Publish / Subscribe model

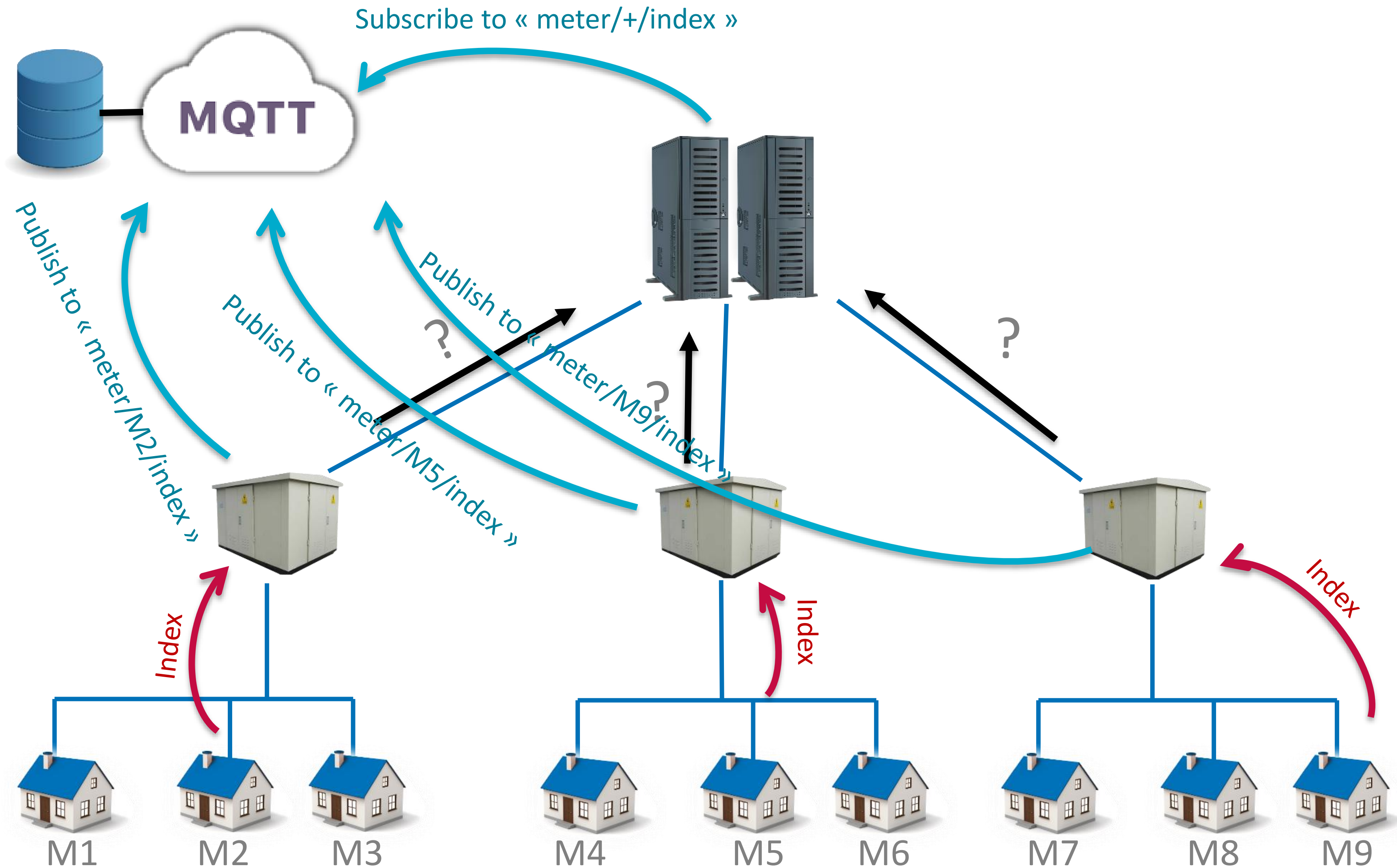
- Message published under a specific topic
- Interested nodes need to subscribe to the topic



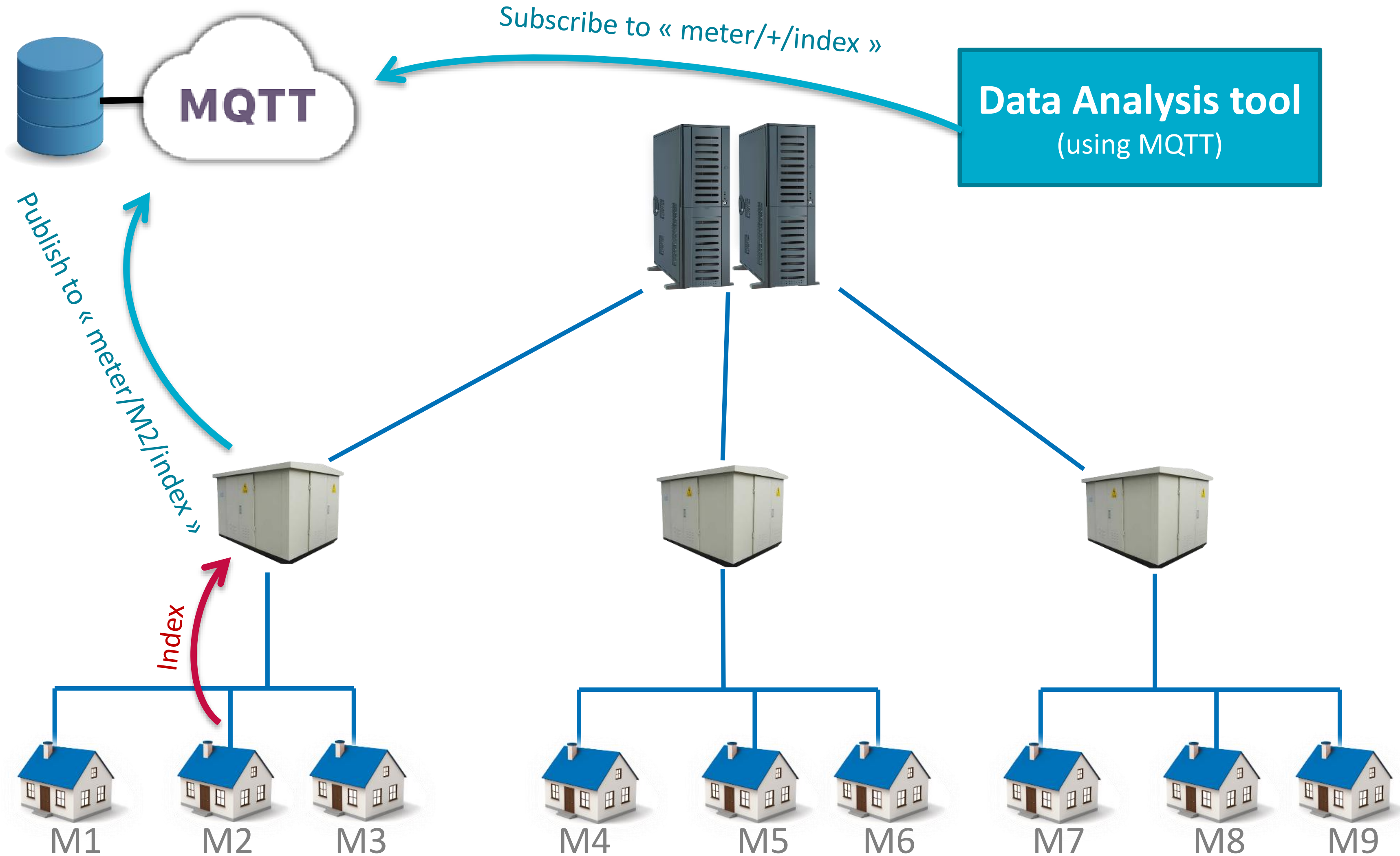
MQTT : Topics hierarchy & Wildcards



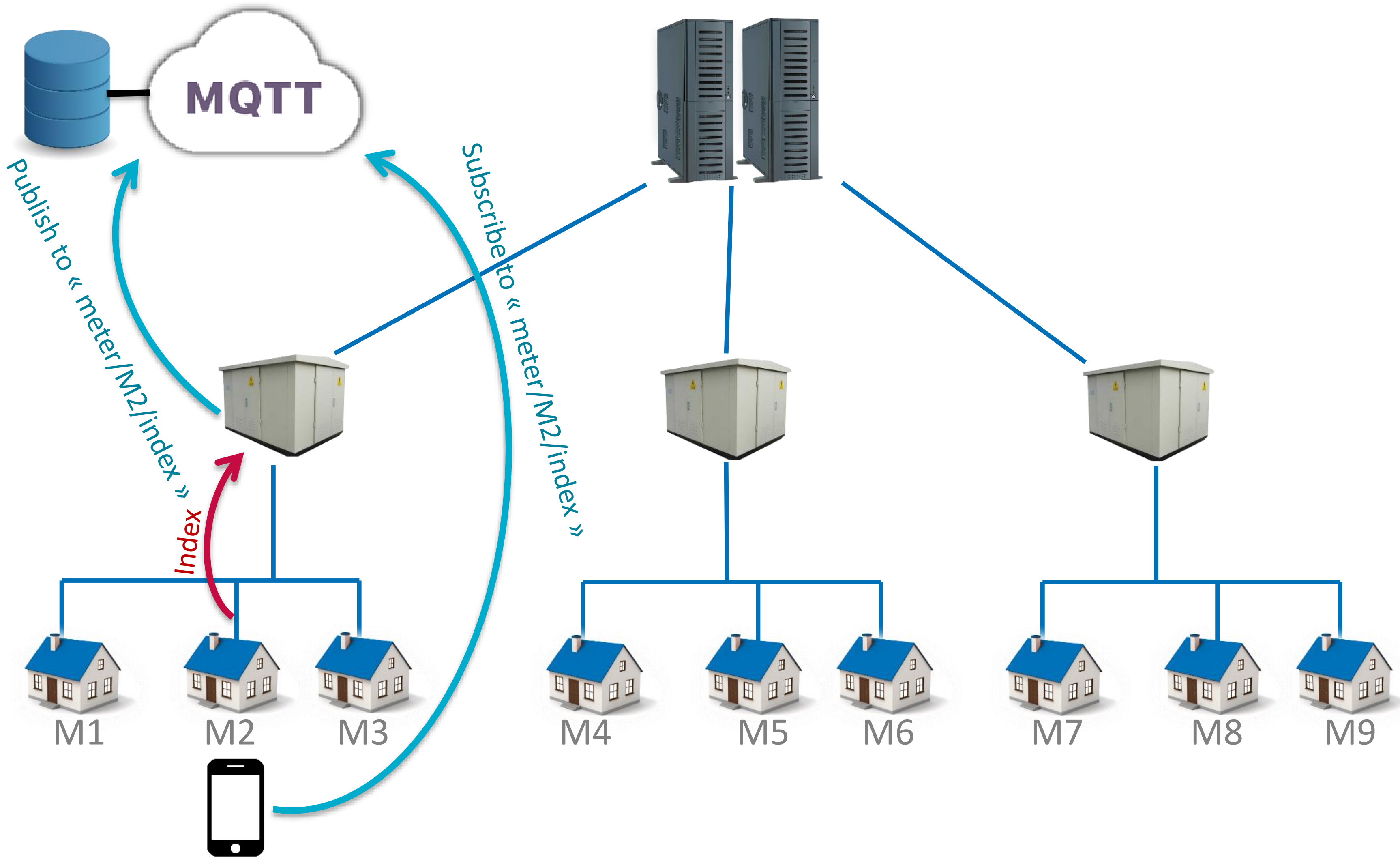
Smart-metering : Data Collection



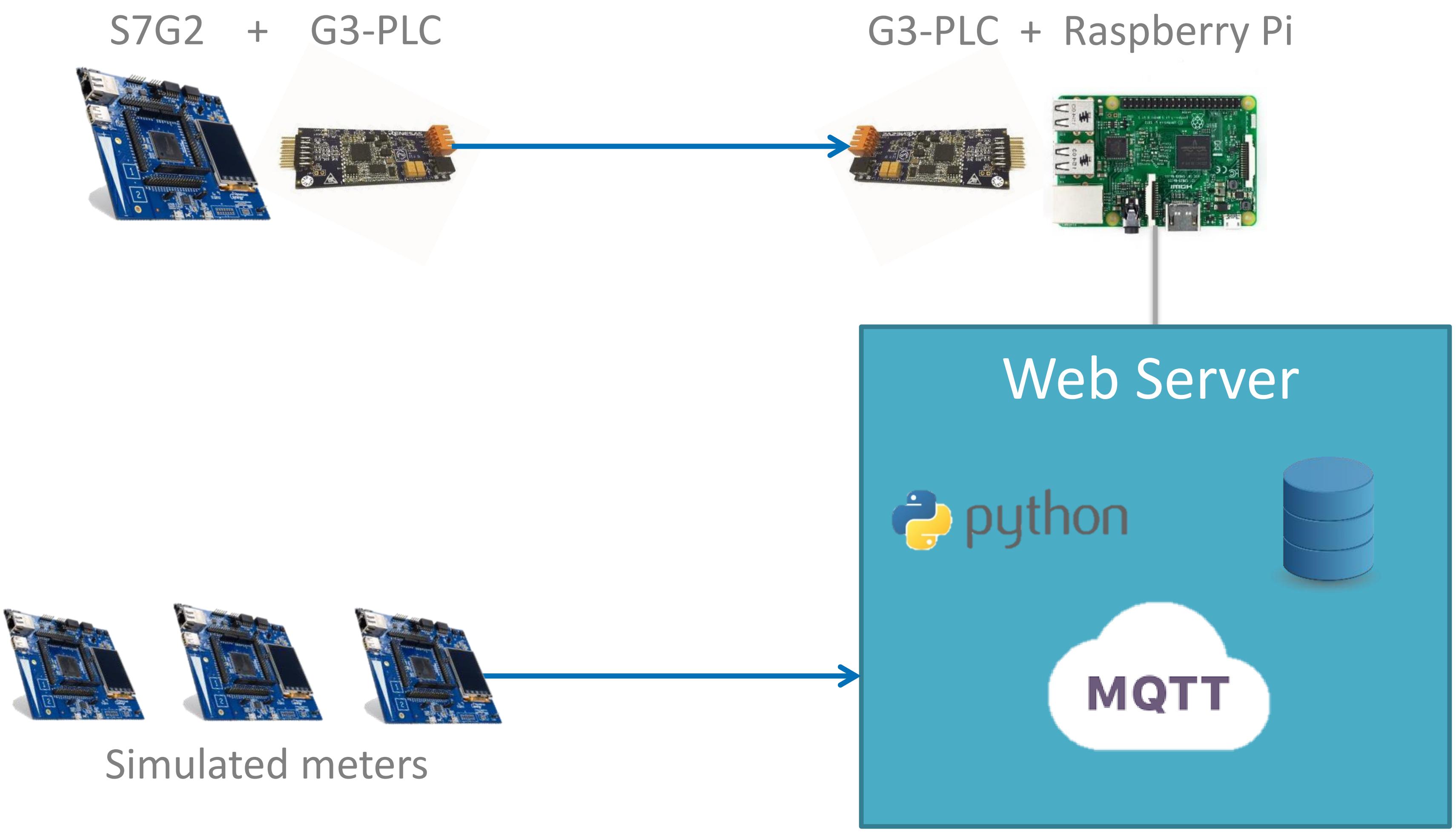
Smart-metering : Data Analysis



Smart-metering : Data Analysis



Demo Setup


S7G2 + G3-PLC
G3-PLC + Raspberry Pi
Web Server
python
MQTT
Simulated meters

Conclusion

Hybrid communication scheme

- Link problem with G3-PLC → longer delays / packet drop
- Use LoRa as fallback for degraded links
- Hybrid interface over G3-PLC/LoRa
- Portable solution to multiple platforms

Smart-metering application :

- Integration with MQTT
- Quasi-realtime visualization of the distribution grid status

Conclusion

Current work

- G3-PLC driver tested and working on RPi and Renesas S7G2
- LoRa driver + MAC layer implemented and tested offline
- MQTT & Visualisation frontend implemented

Future work

- Fix LoRa driver platform-dependent issues
- Hybrid driver
- Small improvements in the application part
- Integrate MQTT authentication & encryption features