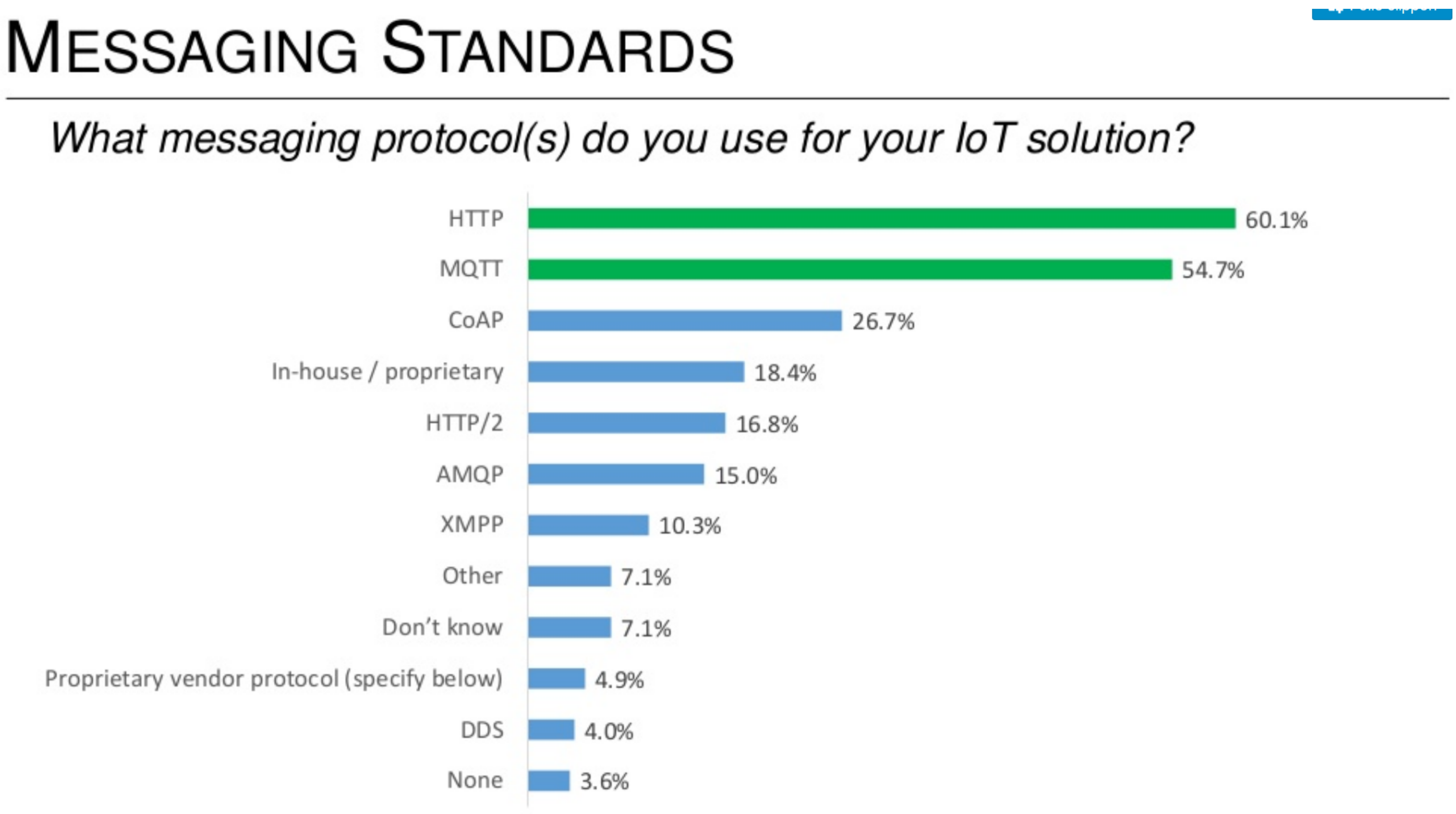
Towards Implementing a MQTT-Gateway for Semi-Constrained Devices

Gabriel Nikol

Content

- Motivation
- Related Work
  - HTTP, MQTT, MQTT-SN
  - Device Types
  - MQTT-SN Architecture
  - MQTT-SN Client Life Cycle
  - Transmission Protocols
- MQTT-SN Gateway Implementation
  - Core Components
  - Linux Gateway Implementation
  - Unit and Regression Tests
  - BLESocket
- Conclusion
- Future Work

Most used Messaging Standards

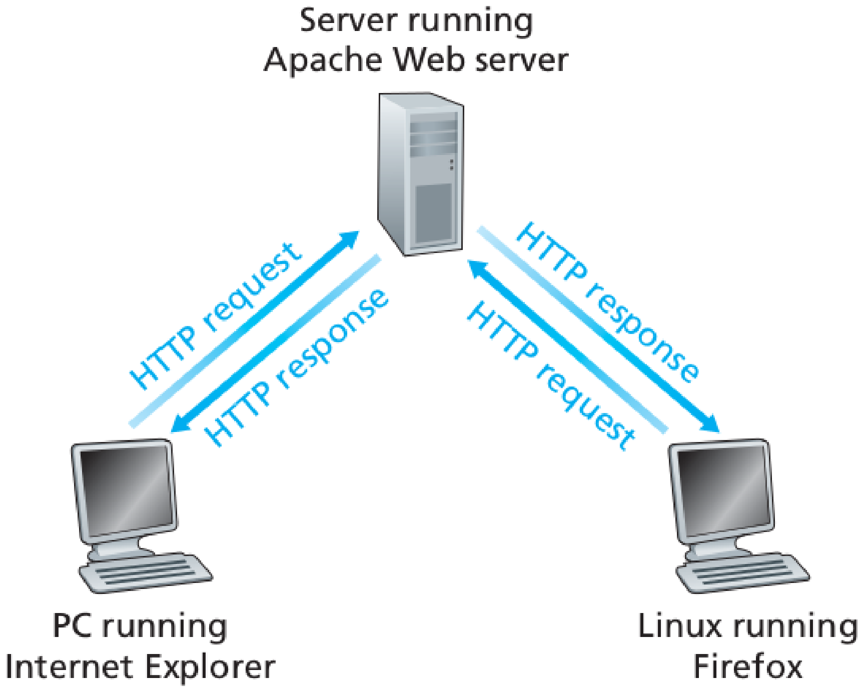


MESSAGING STANDARDS

What messaging protocol(s) do you use for your IoT solution?

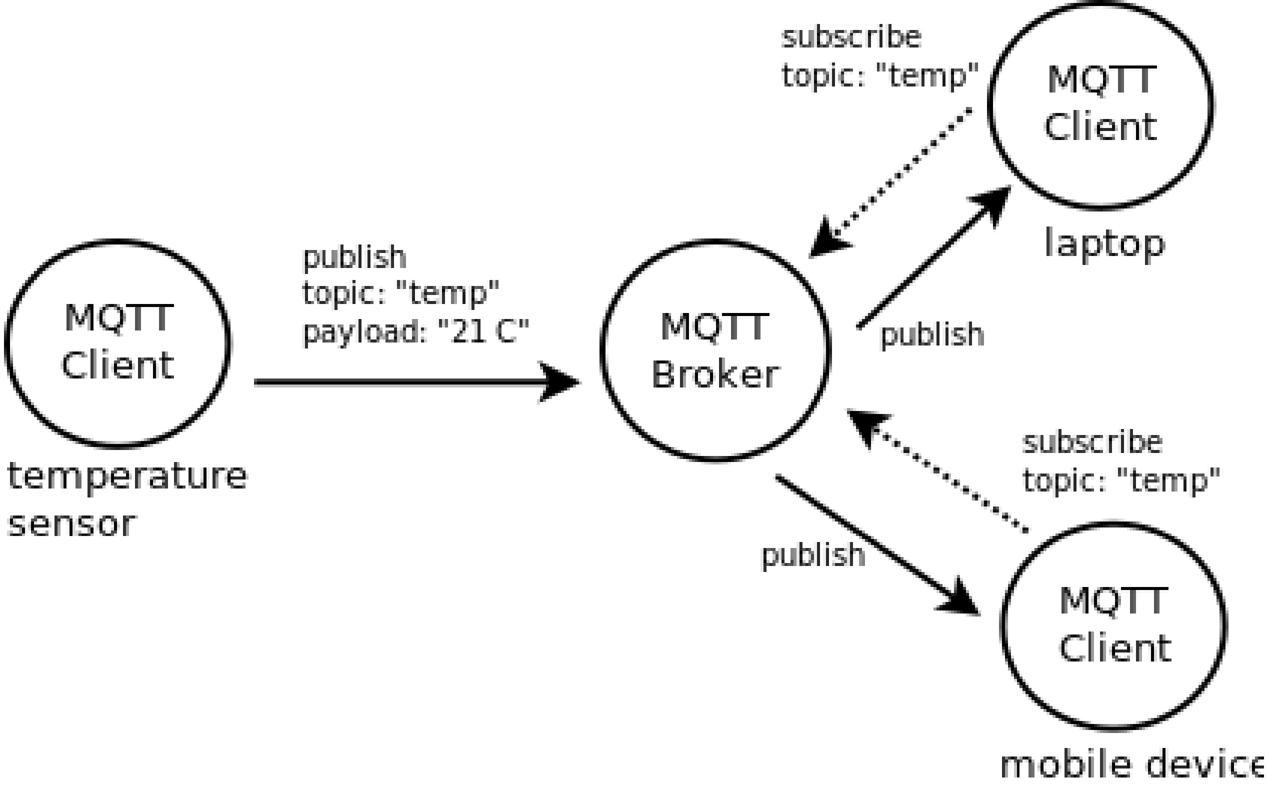
HTTP

- Client-Server paradigm
- Addressable Resources (URI)



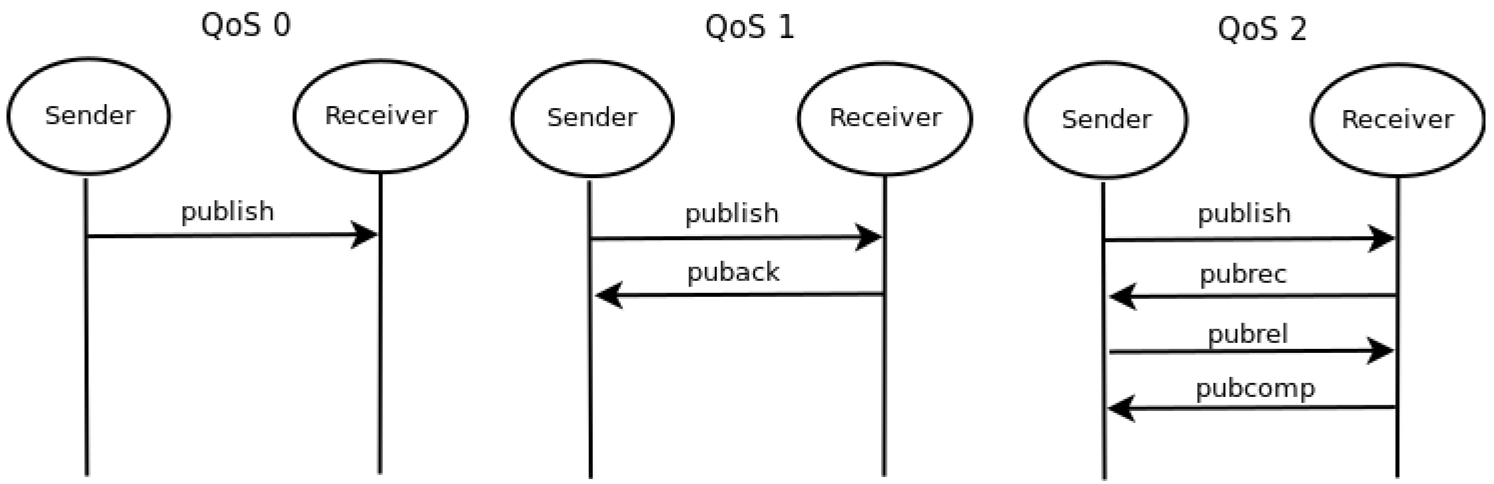
MQTT

- Publish-subscribe paradigm
- Data centric approach



MQTT QoS

- QoS 0 (at most once)
- QoS 1 (at least once)
- QoS 2 (exactly once)

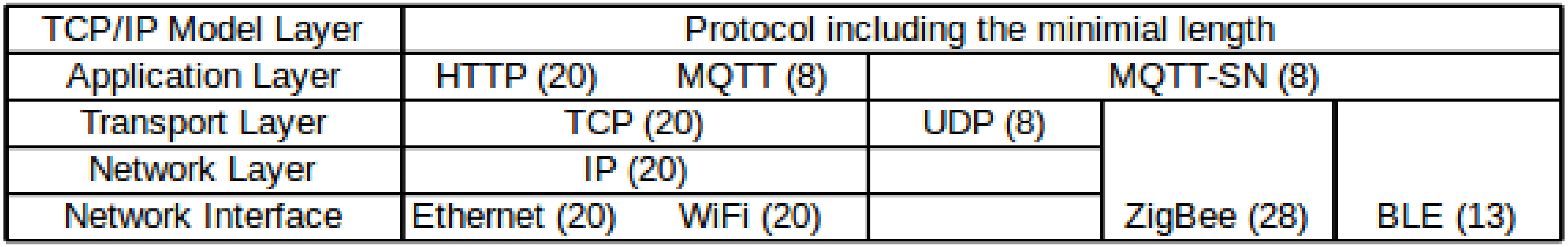


MQTT-SN

- Version of MQTT
- Not connection oriented
- Supports short topic names + pre-defined topic
- QoS -1 like QoS 0 but without connect
- Supports sleeping clients

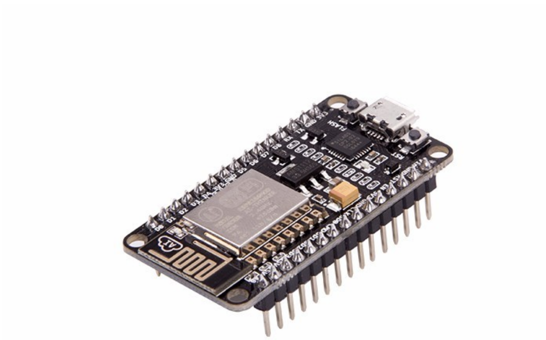
Comparing HTTP, MQTT and MQTT-SN

- Comparing packet sizes in TCP/IP Model



TCP/IP Model Layer	Protocol including the minimial length				
Application Layer	HTTP (20)	MQTT (8)	MQTT-SN (8)		
Transport Layer	TCP (20)		UDP (8)	ZigBee (28)	BLE (13)
Network Layer	IP (20)				
Network Interface	Ethernet (20)	WiFi (20)			

IoT Devices



docker

Microsoft Azure

amazon web services

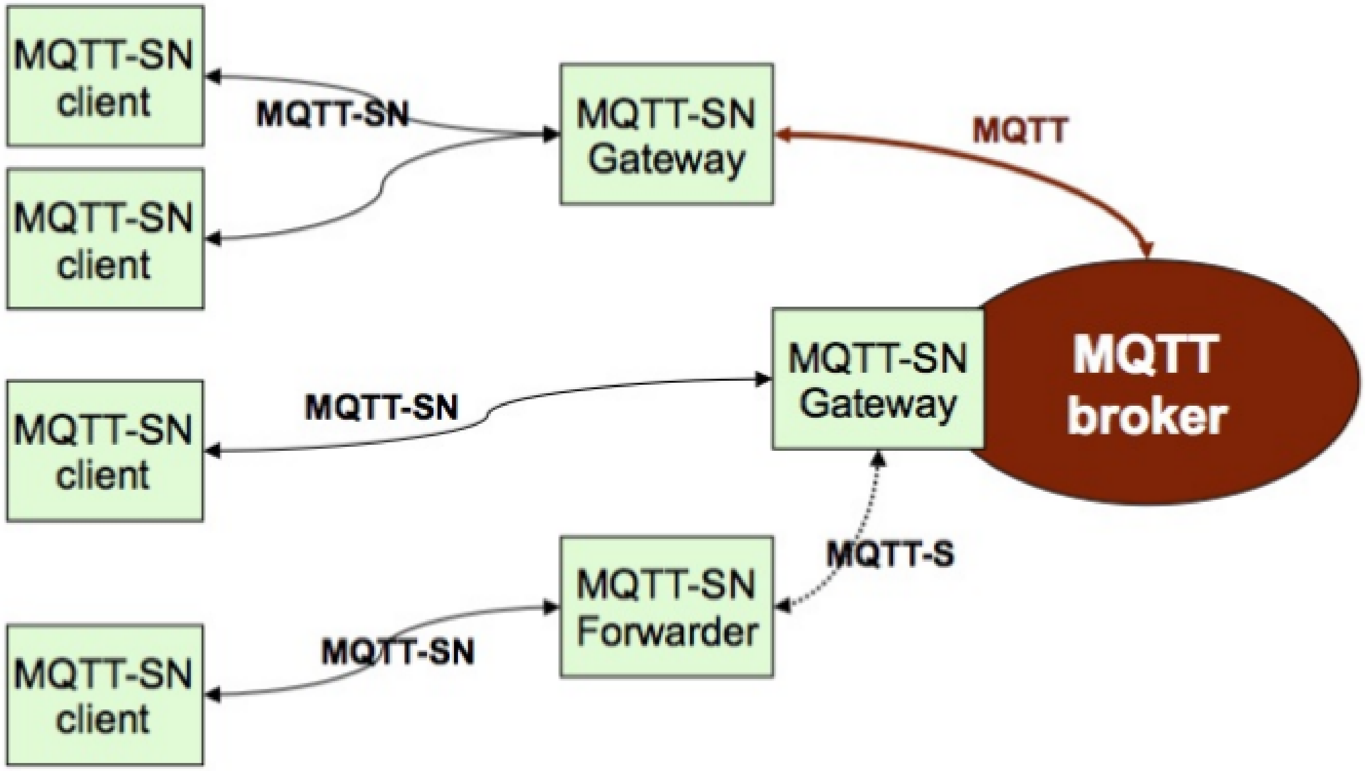
Device Types

- Constrained Devices
  - Limited resources (FLASH, CPU, RAM, Energy)
- Semi-Constrained Devices
  - Limited resource (FLASH, CPU, RAM, not Energy)
- Unconstrained Devices
  - Nealy unlimited resources (HDD, CPU, RAM)

MQTT-SN Architecture

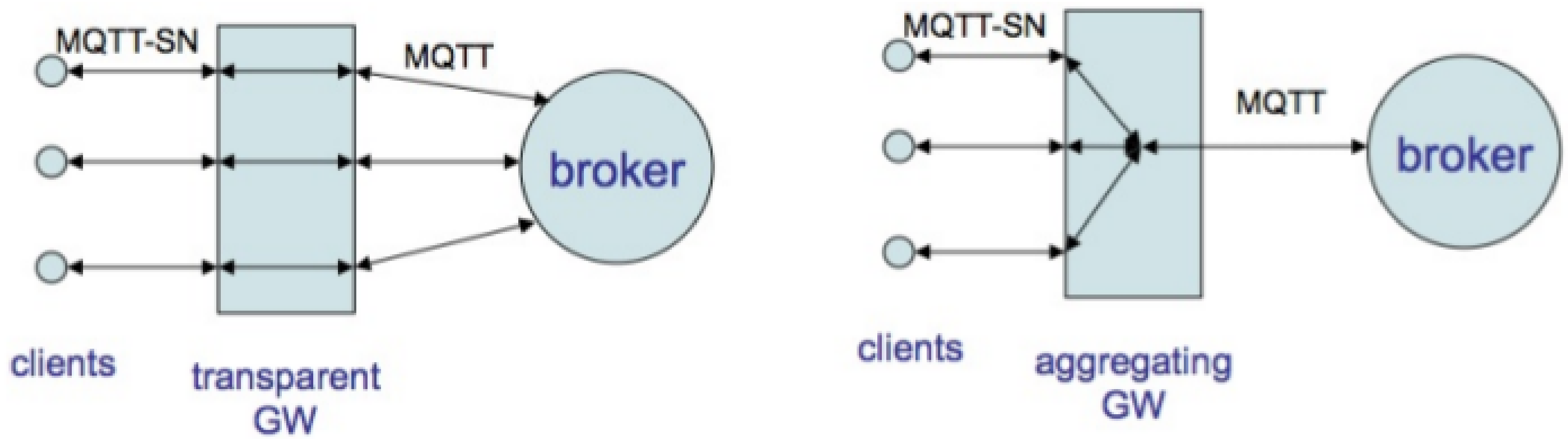
- Three MQTT-SN components:

  client, gateway, forwarder



MQTT-SN Architecture

- Two kinds of gateways:
  - transparent and aggregating

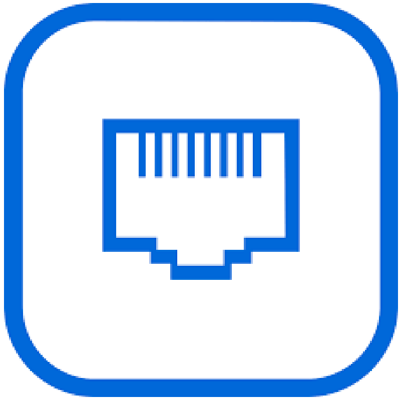


MQTT-SN Client Life Cycle

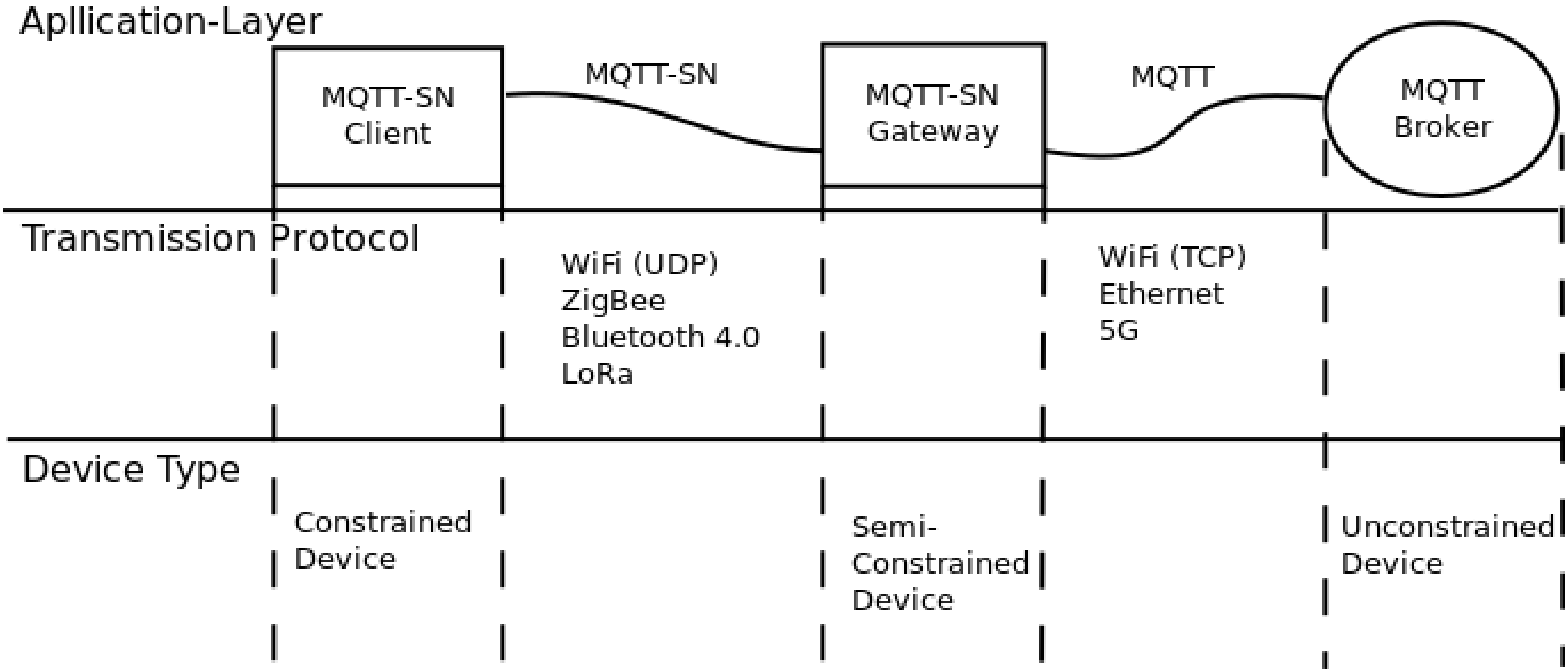
- MQTT-SN client is a constrained device
  - find a MQTT-SN Gateway via advertisement or searching a gateway
  - connect to the MQTT-SN gateway with a will message
  - register topics
  - subscribe to topics
  - send and receive publishes
  - unsubscribe from topics
  - sleep
  - wake up and collect queued publishes
  - sleep and wake up frequently
  - power source is empty - will message is published

Transmission Protocols

Bluetooth™
4.0



IoT Overview



Apllication-Layer

MQTT-SN Client

MQTT-SN

MQTT-SN Gateway

MQTT

MQTT Broker

Transmission Protocol

WiFi (UDP)
ZigBee
Bluetooth 4.0
LoRa

WiFi (TCP)
Ethernet
5G

Device Type

Constrained Device

Semi-Constrained Device

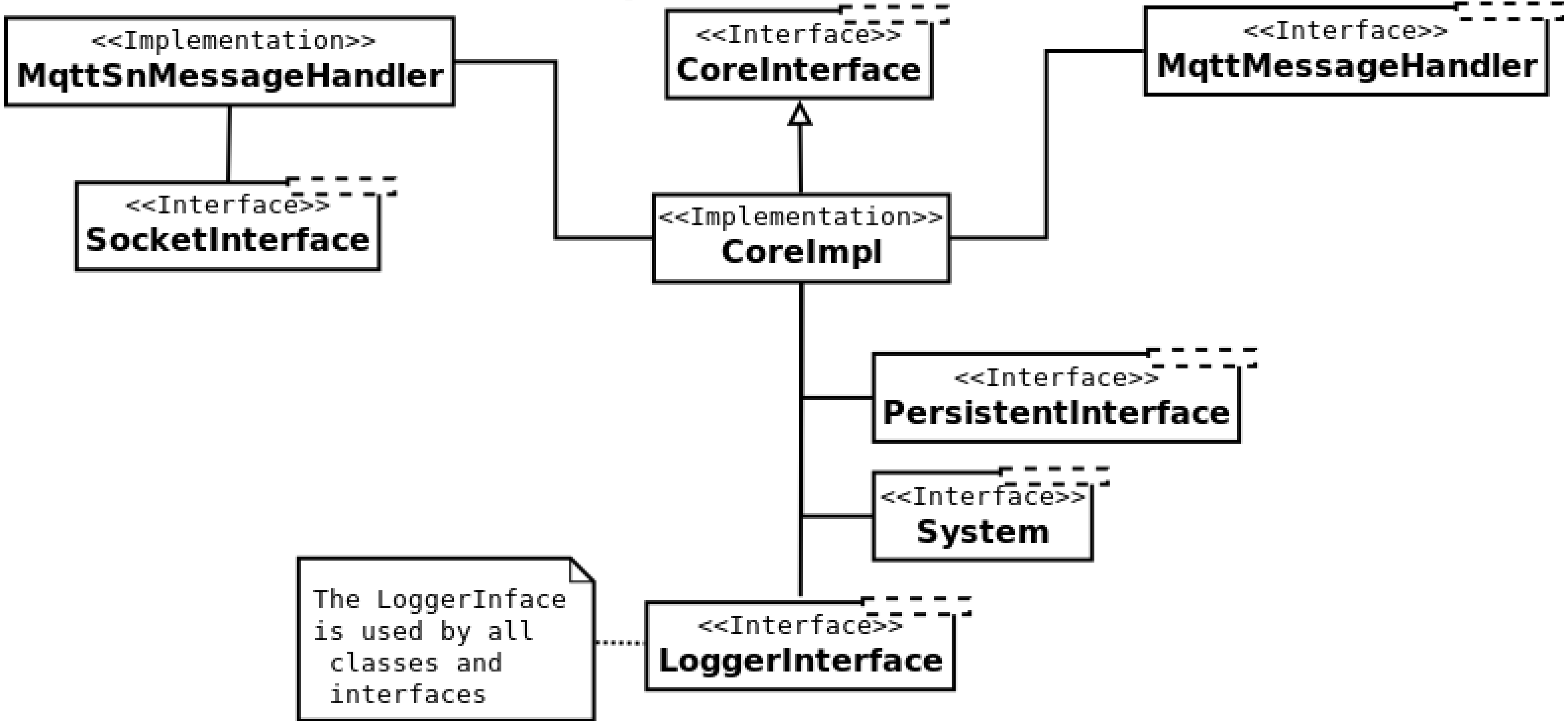
Unconstrained Device

MQTT-SN Gateway Implementation

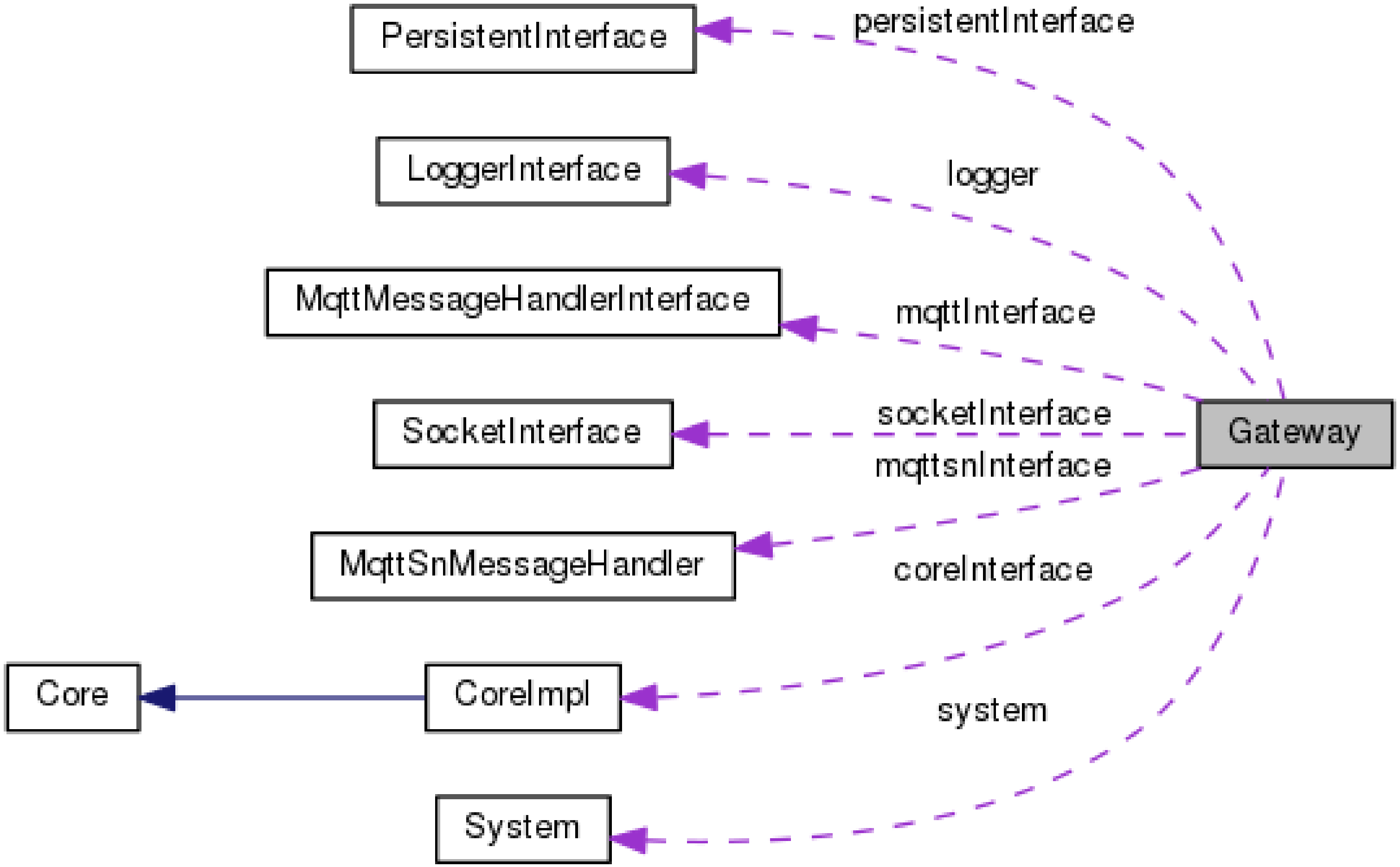
Core Components

- MqttSnMessageHandler + CoreInterface

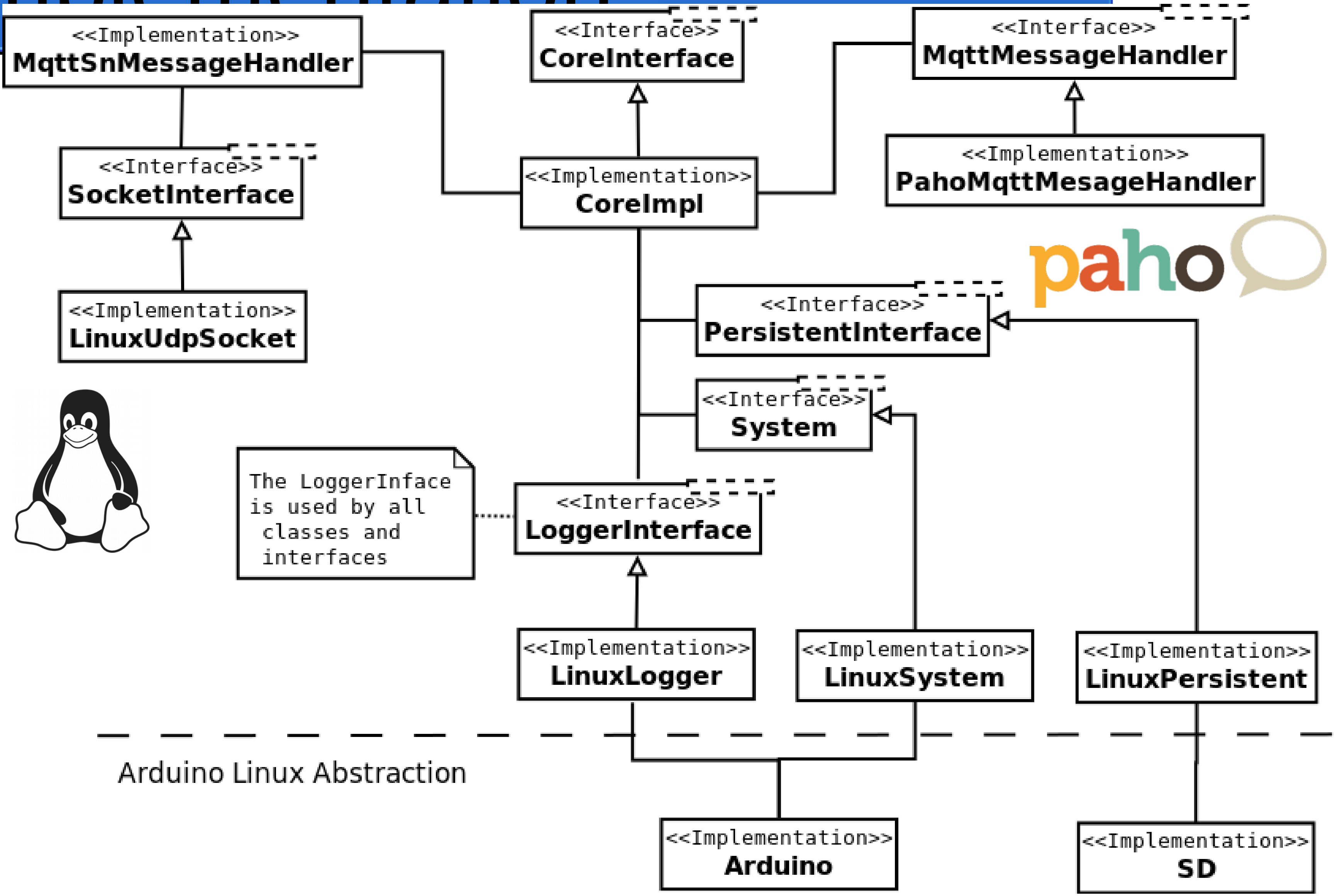
Environment Independent



Gateway class



Linux Gateway Implementation



<<Implementation>>
MqttSnMessageHandler

<<Interface>>
CoreInterface

<<Interface>>
MqttMessageHandler

<<Interface>>
SocketInterface

<<Implementation>>
CoreImpl

<<Implementation>>
PahoMqttMesageHandler

<<Implementation>>
LinuxUdpSocket

<<Interface>>
PersistentInterface

<<Interface>>
System

The LoggerInface is used by all classes and interfaces

<<Interface>>
LoggerInterface

<<Implementation>>
LinuxLogger

<<Implementation>>
LinuxSystem

<<Implementation>>
LinuxPersistent

Arduino Linux Abstraction

<<Implementation>>
Arduino

<<Implementation>>
SD

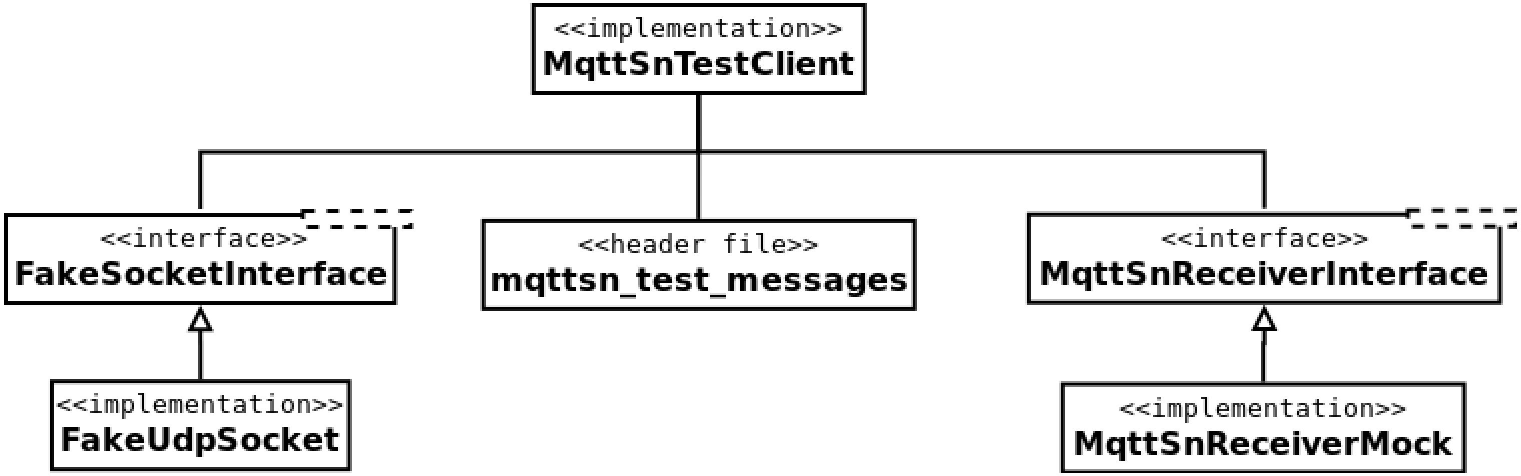
Automated Unit and Regression Testing

- Using GoogleTest and GoogleMo
- Starting tests inside IDE CLion
- Writing a MqttSnTestClient + PahoMqttTestMessageHandler

googletest

Google C++ Testing Framework

Test Clients



<<implementation>>
MqttSnTestClient

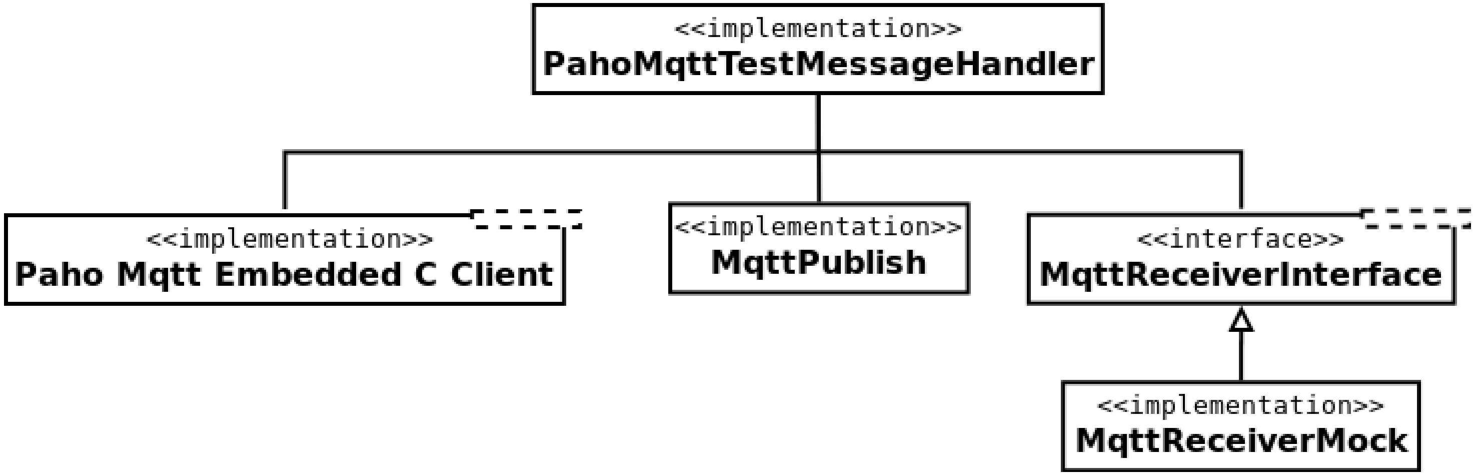
<<interface>>
FakeSocketInterface

<<header file>>
mqttsn_test_messages

<<interface>>
MqttSnReceiverInterface

<<implementation>>
FakeUdpSocket

<<implementation>>
MqttSnReceiverMock



<<implementation>>
PahoMqttTestMessageHandler

<<implementation>>
Paho Mqtt Embedded C Client

<<implementation>>
MqttPublish

<<interface>>
MqttReceiverInterface

<<implementation>>
MqttReceiverMock

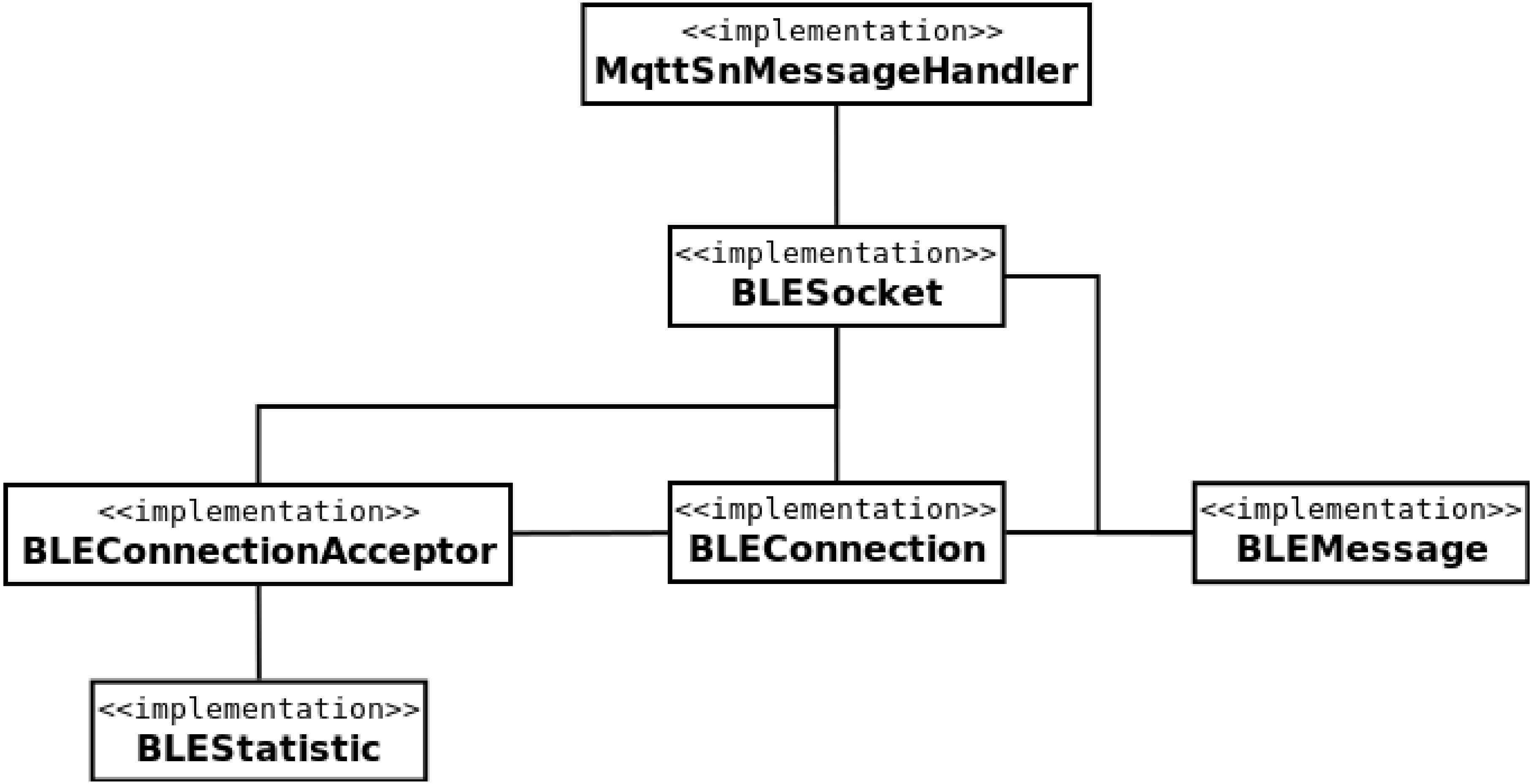
Test Results

- Total: 96 unit tests
- 61 test pass
- 35 test fail

- But:
  - Important functionality is tested and working

BLESocket

- Drop in replacement for LinuxUdpSocket



Conclusion

- Project successfull:
- implemented a MQTT-SN Gateway Prototype
- Runs on Linux and with little changes on Arduino/ESP8266
- Partially tested
- SocketInterface can be exchanged

Future Work

- Implement more transmission procotols: ZigBee, LoRa
- Support more platforms: Mbed, RTOS
- Implement: QoS 2 and will update
- Enhance unit tests
  - More tests
  - Fix broken parts
- Add stress tests and learn the limits of the design

Thank you for your Attention

Ideas or Suggestions?