

MQTT

The solution for our real time data transfer needs ?

Why ?

- Intermagnet september 2014 :
DD 26 Rapid Transport of real-time data from IMO to institute. (0.0)
- Can we leverage immediate transmission of second data with minimal latency ?
- Possibilities proposed :
 - Seismic solution :
 - Seedlink that plugs in antelope and earthworm
 - Other approaches
 - WebSocket ?

Why ?

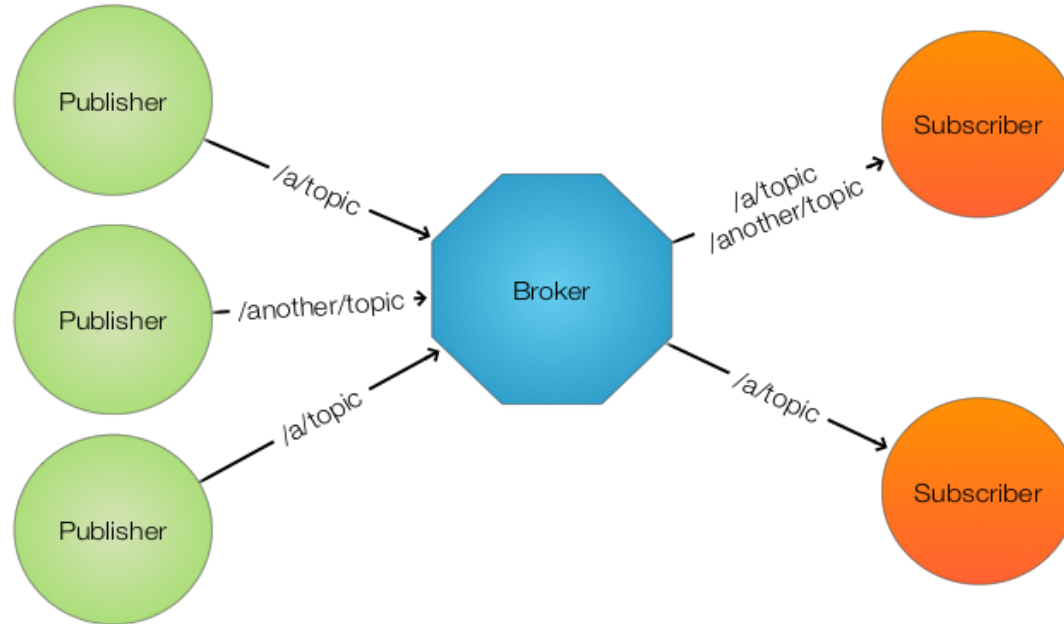
- Decouple own developed software from my current database
 - Autodif , variometers, protonmagnetometers ...

What about MQTT

MQTT stands for MQ Telemetry Transport. It is a publish/subscribe, **extremely simple and lightweight messaging protocol**, designed for constrained devices and **low-bandwidth, high-latency or unreliable networks**. The design principles are to minimize network bandwidth and device resource requirements whilst also attempting to ensure reliability and **some degree of assurance of delivery**. These principles also turn out to make the protocol ideal of the emerging “machine-to-machine” (M2M) or “Internet of Things” world of connected devices, and for mobile applications where bandwidth and battery power are at a premium.

What is MQTT

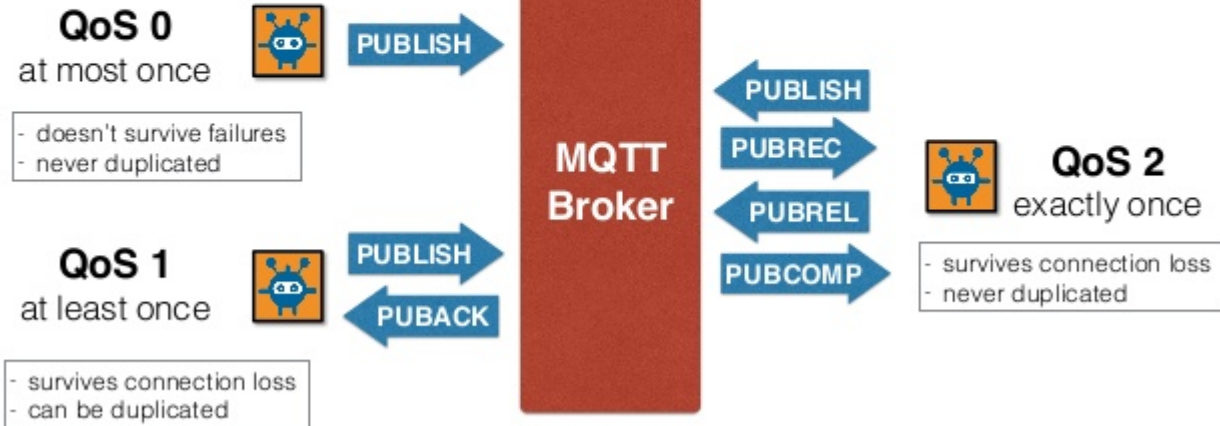
MQTT Diagram



Quality Of Service

MQTT

Quality of Service for **reliable messaging**



Persistent Sessions and Queuing

- Durable subscriptions : CleanSession false
- Broker stores
 - Existence of a session
 - All subscriptions
 - All messages in QoS 1 or 2 flow, which are not confirmed by the broker
 - All new QoS 1 or 2 messages, which the subscriber missed while it was offline
 - All received QoS 2 messages, which are not yet confirmed to the publisher

TOPICS

- Topics are UTF-8 strings : myhome/floor/room/temp
- For my purpose for the moment I used :
 - IAGACODE/INSTRUMENTID/SAMPLERATE : dou/lemi0001/sec
- Subscriptions can be for a specific topic or with wildcards :
 - DOU/# all information from DOU
 - DOU/+/sec all second data from all instruments in Dourbes

MQTT in action

```
var mqtt = require('mqtt');
var client = mqtt.connect('mqtt://test.mosquitto.org');

client.on('connect',

    function () { client.subscribe('hello/world', { "qos":1});

        client.publish('hello/world', 'Hello mqtt', { "qos":1});});

client.on('message', function (topic, message)

{ console.log(message.toString());

  client.end();});
```

MQTT main principals

- Async protocol
 - No guarantee on order of delivery
- Binary protocol
- TCP level with specific port
- Need of at least one Broker (on side of institution)
 - Open source Mosquitto can be used (all popular OS also Raspberry PI)
 - Needs monitoring and management

Experiences after 1 year : Antarctica

- Set up :
 - Each instruments sends seconds and minute data directly after availability
 - QOS = 1
 - On connection loss keep sec data of last 4 hours in memory , keep minute data for 1 day
 - Keepalive : 10 seconds
 - One Broker configured in Dourbes
- Experience :
 - All data arrives at Dourbes with a delay that is +/- 300 ms (due to satellite links)
 - Connections get lost +/- 10 times a month, the protocol reconnects automatically
 - Other data that was not send needs to be transferred via another channel
 - Their are limits on what a topic can store

Other Experiences

- Autodif software adapted to send on MQTT his measurement.
- Test Field installation (three months) of completely automated observatory worked with small satellite



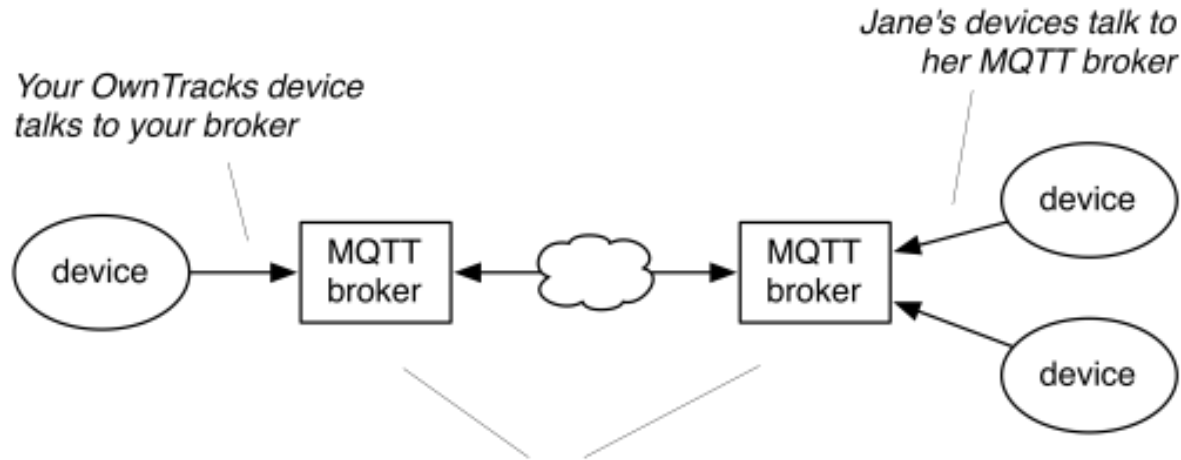
The Solution ?

- In terms of realtime it works :
 - Lots of client libraries in different languages (seedlink ?)
- When connection lost :
 - Need of fallback solution to guarantee completeness of dataset
- The solution to what ?
 - Space weather : K-indexes real time adaptation

Questions ?



No experience : Bridging



By configuring a bridge, both brokers can talk to each other; ACLs ensure they exchange only data you both agree on.

Why practically MQTT

- Open Sourced and accepted as OASIS standard (version 3.1.1)
- Clients are lightweight and a lot of client libraries available
- Run easily on small embedded devices
- Can run over websockets if needed (port 80)
- Can use ssl layer for full encryption

What next

- Installation of gyrodif with automated calculation of baseline points
- Investigate the use of MQTT for sensor networks (even more lightweight no tcp stack necessary)

Why Technically MQTT

- Simple binary protocol , small footprint
- Fast communication (minimize latency on network)
- Low power consumption
 - Compared with http (on 3G network android)
 - 93x faster throughput
 - 11.89x less battery to send
 - 170.9x less battery to receive
 - 1/2 power to keep connection open
 - 8x less network overhead
- Three levels of Quality of service :
 - 0 at most one
 - 1 at least one (needs to receive an ack)
 - 2 exactly one (extra level of acknowledgment)