Introduction to the Internet of Things

Session 03

Ulrich Norbisrath

Exchange Formats/Protocols How to speak IoT?

- In groups of 3-5 (10min)
  - Take your first from research records, check if you picked some exchange formats (or pick from below)
  - Research examples and properties (speed, readability, user friendliness)
  - Divide and conquer (each of you does 2 formats)
  - Pick two advantages and two disadvantages
- Present and discuss in group, refine advantages and disadvantages in comparison to others and your examples – pick the two best and two worst (10min)
- Open discussion (10min)
- Cover at least in team:
  - exchange formats:
  - Binary (also check out python's pickle and CBOR –> maybe relate to JSON)
  - Text
  - XML (focus on SOAP)
  - JSON
  - YAML

IoT-Communication

Text (source for gzipped-binary):

```
John,Smith,21 2nd Street,New
York,NY,10021,212 555-1234,646 555-4567
```

Binary (hex encoded):

```
0000000 8b1f 0008 c85a 599d 0300 caf3
0000014 c8cf 09d3 cdce c92c 31d0 5432
0000030 ca30 514b 2e08 4a29 2d4d f1d1
0000044 2d4b 8857 2fcc d6ca 8bf1 31d4
0000060 3034 3230 ca04 291b 9a98 ea9a
0000074 1a1a 9b19 98e8 9899 3981 a626
0000110 e666 005c 64a8 6397 0045 0000
```

XML:

```
<person>
  <firstName>John</firstName>
  <lastName>Smith</lastName>
  <address>
    <streetAddress>21 2nd Street</streetAddress>
    <city>New York</city>
    <state>NY</state>
    <postalCode>10021</postalCode>
  </address>
  <phoneNumber>
    <type>home</type>
    <number>212 555-1234</number>
  </phoneNumber>
  <phoneNumber>
    <type>fax</type>
    <number>646 555-4567</number>
  </phoneNumber>
</person>
```

JSON:

```
{ "firstName": "John",
  "lastName": "Smith",
  "address": {
    "streetAddress": "21 2nd Street",
    "city": "New York",
    "state": "NY",
    "postalCode": "10021" },
  "phoneNumber": [ {
      "type": "home",
      "number": "212 555-1234" }, {
      "type": "fax",
      "number": "646 555-4567" } ], }
```

YAML:

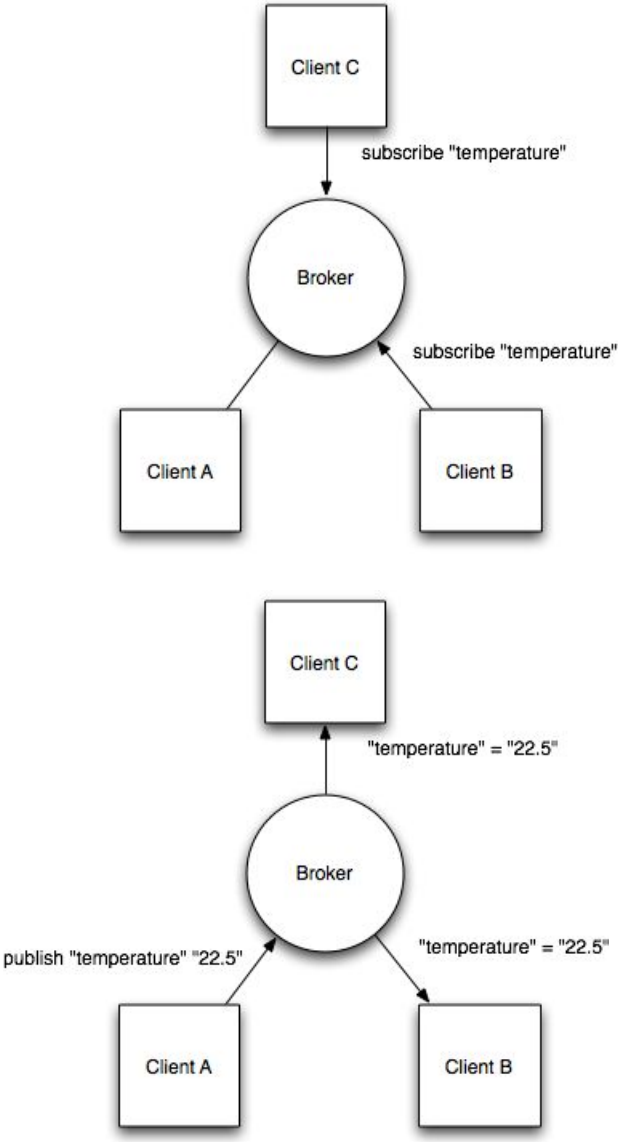
```
firstName: John
lastName: Smith
address:
  streetAddress: 21 2nd Street
  city: New York
  state: NY
  postalCode: '10021'
phoneNumber:
- type: home
  number: 212 555-1234
- type: fax
  number: 646 555-4567
```


Adapted from wikipedia JSON.

Publish Subscribe

- Listener
- Observer Pattern

- Why do we want this?



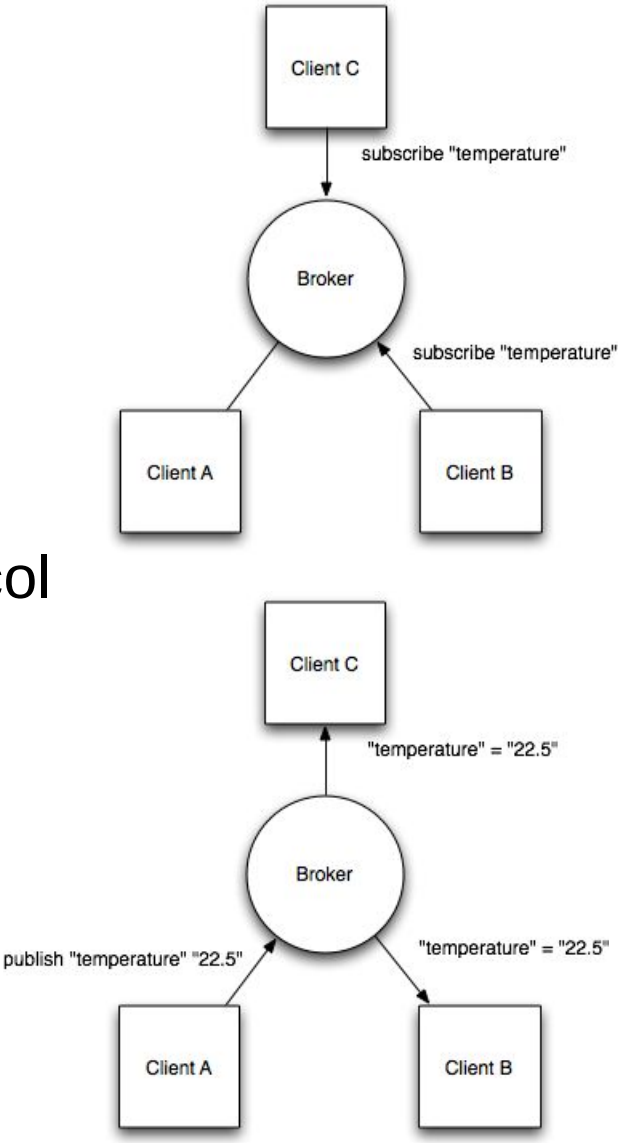
from https://eclipse.org/community/eclipse_newsletter/2014/february/article2.php

IoT Protocols

- Research in team (4-5) (10 min)
  - Example
  - Purpose/challenge
  - Software support
- Cover at least (each 1 – everybody should briefly look at mqtt and CoAP)
  - HTTP Post (everybody knows, so compare with this)
  - CoAP
  - MQTT
  - BLE
  - OSC

MQTT – M2M Communication

- MQ Telemetry Transport
  or Message Queue Telemetry Transport
- MQTT Gateway/Broker, star topology
- Publish Subscribe (Listener, Observer Pattern)
- ISO standard, Invented in 1999
- Runs over TCP/ any other stream-based protocol
- Very lightweight
  → runs even on slow Pis and routers
- Many implementations
- Built in security
  - Allows layered security/stacked gateways
  - User access management
  - End to end encryption possible



from https://eclipse.org/community/eclipse_newsletter/2014/february/article2.php

Lab 3

- COAP
  - Install simple COAP Arduino library example (use esp32 example, adjust led, and use ESP8266Wifi.h)
  - Explore and switch with Copper4Cr (Chrome developer extension – or old firefox <= 55 with copper extension)
  - Optional: use libcoap binary tools
- MQTT
  - Install mqtt.fx on your laptop
  - Send and receive with mqtt.fx messages to mosquitto (mqtt server) running on pi
  - Build simulators text or gui (use your preferred language and respective mqtt library or consider IoTempower's integriot for Python3 → installed and runs on pi in iot environment)
    - Sensor simulator: Temperature sensor simulator
      - Input start temp, end temp, time to linearly rise → then run this simulation
    - Relay simulator: Relay switch (like AC) simulator
      - Showing on and off → can be switched to theses states remotely
    - Integrator: switch at specific temperature (can be hardcoded) → needs to connect the upper two