Mobile Computing Sending Sensor Data to IoT Platforms

Slides: tmb.gr/mc-eco

Overview

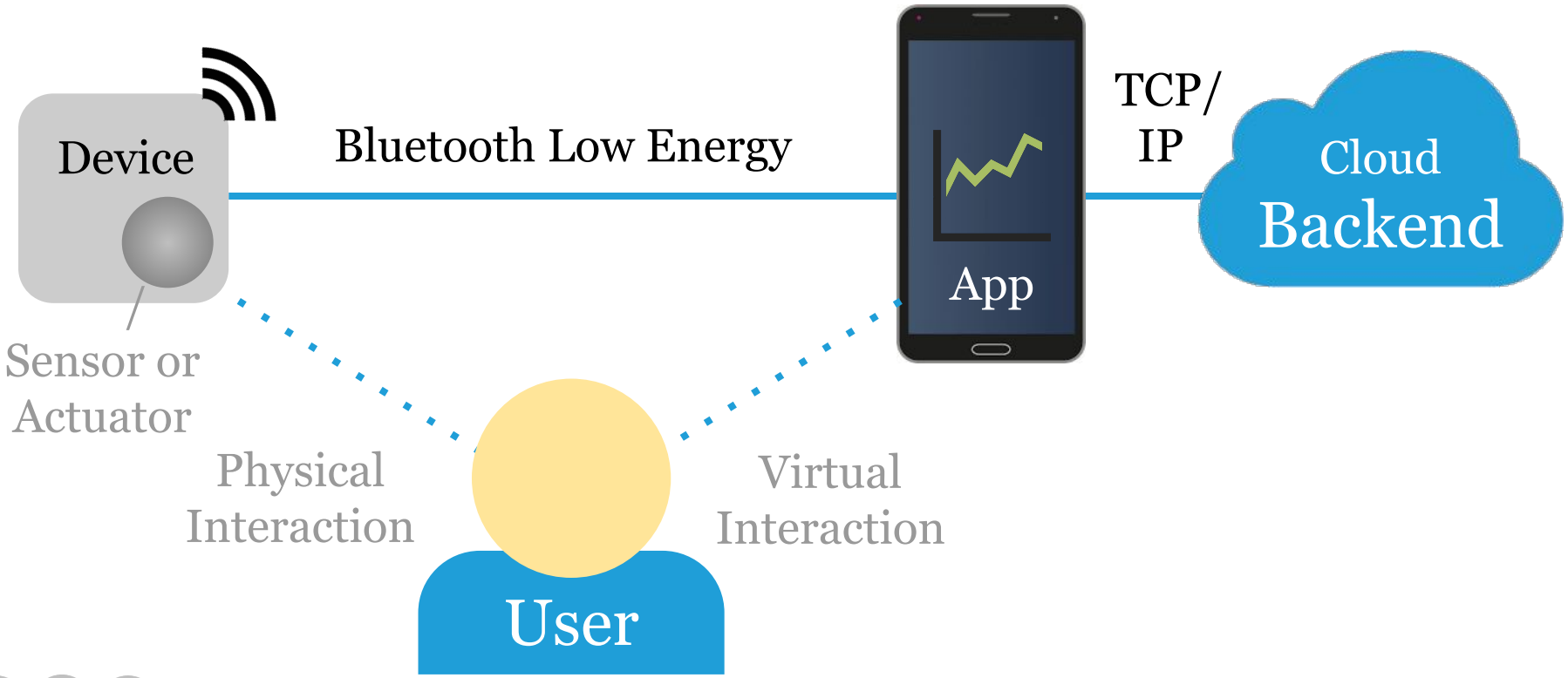
We will send data to an IoT platform or dashboard.

Using HTTP and MQTT, two major IoT protocols.

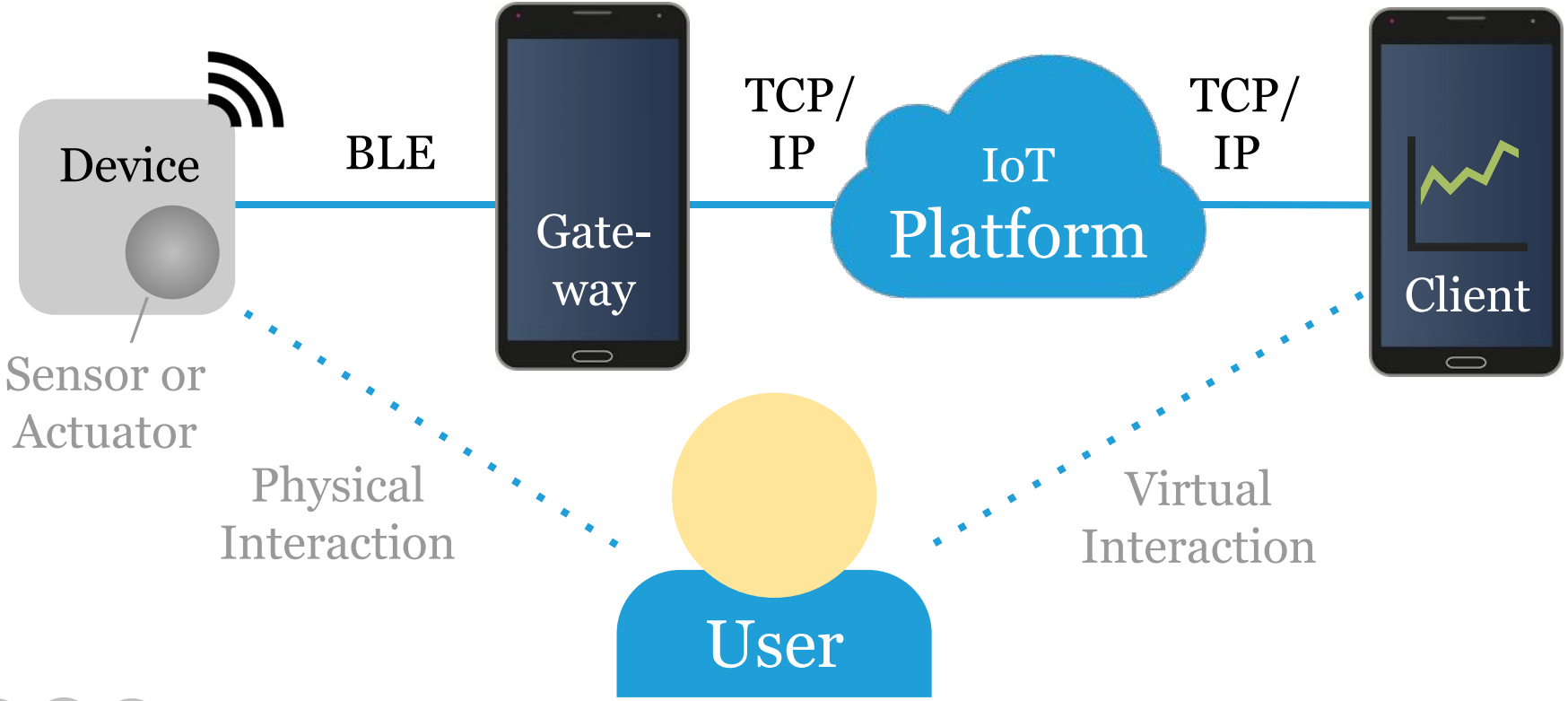
Debug API calls with simple command line tools.

Define API, data format and information model.

Reference model



App has two roles



IoT platforms

IoT platforms make data available at a central point.

There are many platforms, incl. AWS and Azure IoT.

We will look at a simple IoT platform as an example:

ThingSpeak provides data storage and visualisation.

It receives and provides data via HTTP or MQTT.

Internet protocol suite layers

RFC 1122 layers are loosely based on the OSI model:

Application layer, process to process, HTTP, MQTT, …

Transport layer, host to (remote) host, UDP or TCP.

Internet layer, inter-network addressing and routing.

Link layer, details of connecting hosts in a network.

Hypertext Transfer Protocol (HTTP)

HTTP, the "Web protocol", is specified in RFC 2616.

It uses TCP/IP as its transport, on port 80 and 443.

A *client* sends a *request*, the *server* sends a *response*.

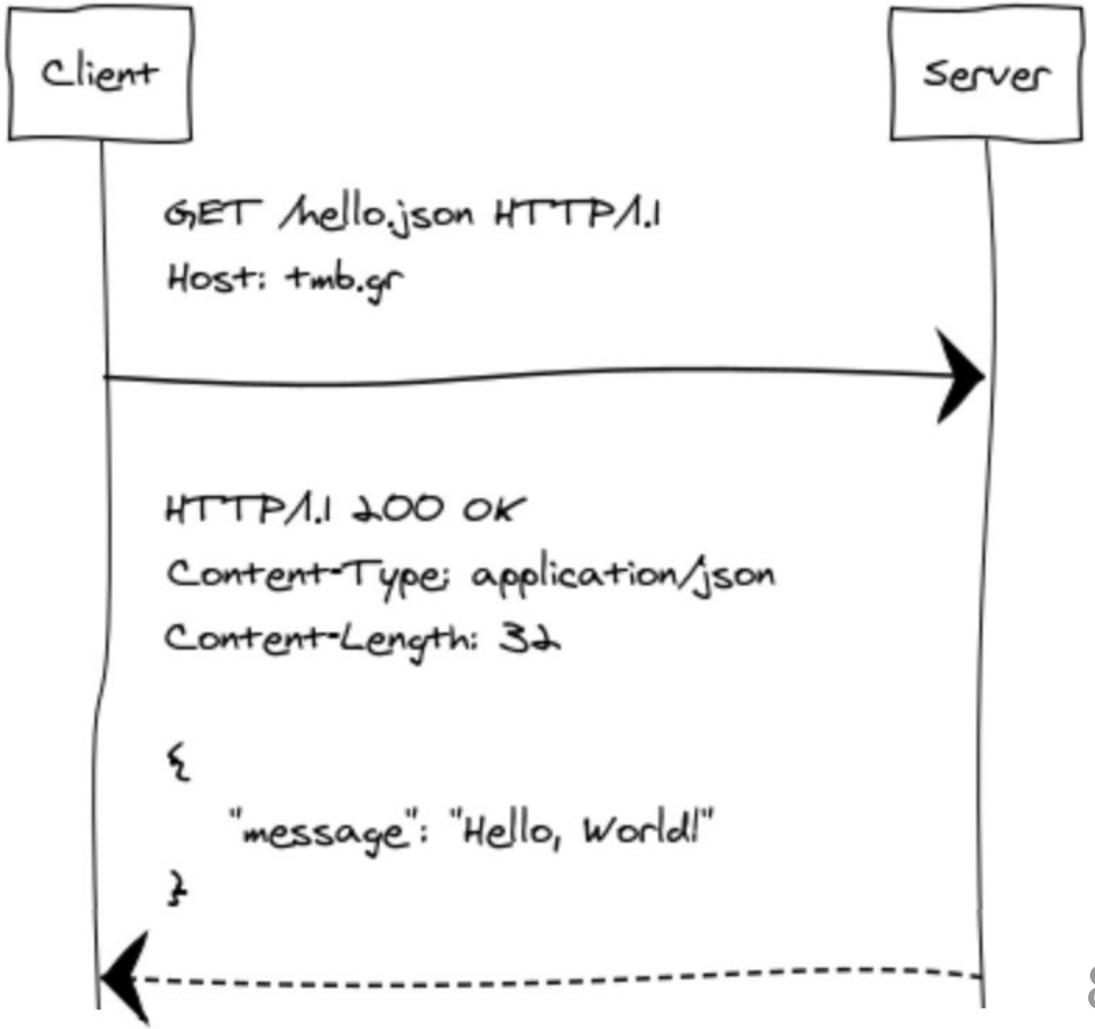
Request and response *headers* are encoded in ASCII.

The content type and length are declared in headers.

HTTP

Web request w/ host header.

Web response with headers and content.



Debugging HTTP with Curl and PostBin

Curl (https://curl.se/) is a command line Web client.

It's useful to test Web APIs, try this GET request:

```
$ curl -v http://tmb.gr/hello.json
```

Or create a PostBin and send a POST request with:

```
$ curl --data "temp:23" https://postb.in/…
```

Here's the manual and a book on Everything Curl.

ThingSpeak HTTP API

ThingSpeak has a device- and client-side HTTP API.

```
Host: api.thingspeak.com
Port: 80 or 443
```

```
POST /update?api_key=WRITE_API_KEY&field1=3
GET /channels/CHANNEL_ID/feed.json?
api_key=READ_API_KEY
```

See Wiki for ThingSpeak cURL examples.

Uniontown Weather Data - Thi

https://thingspeak.com/channels/3

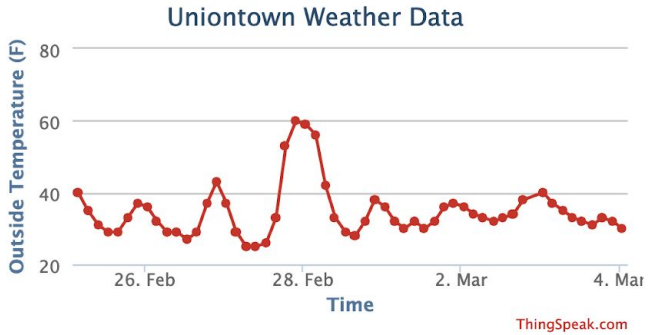
ThingSpeak™ Channels Apps Community Support How to Buy Sign In Sign Up

Uniontown Weather Data

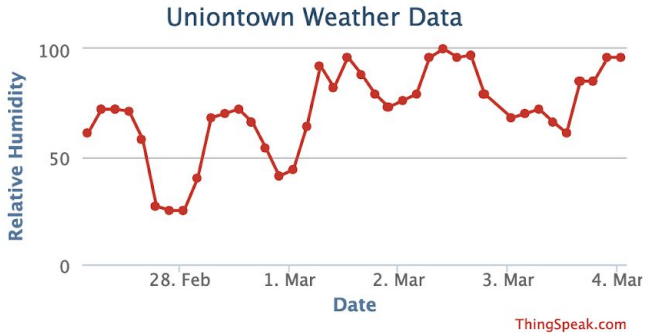
Channel ID: 3
Author: iothans
Access: Public

Weather data from Uniontown, PA
temperature, humidity, weather station, dew point, channel_3

Field 1 Chart



Field 2 Chart



MQTT

MQTT is a standard messaging protocol (v3.1.1, v5.0).

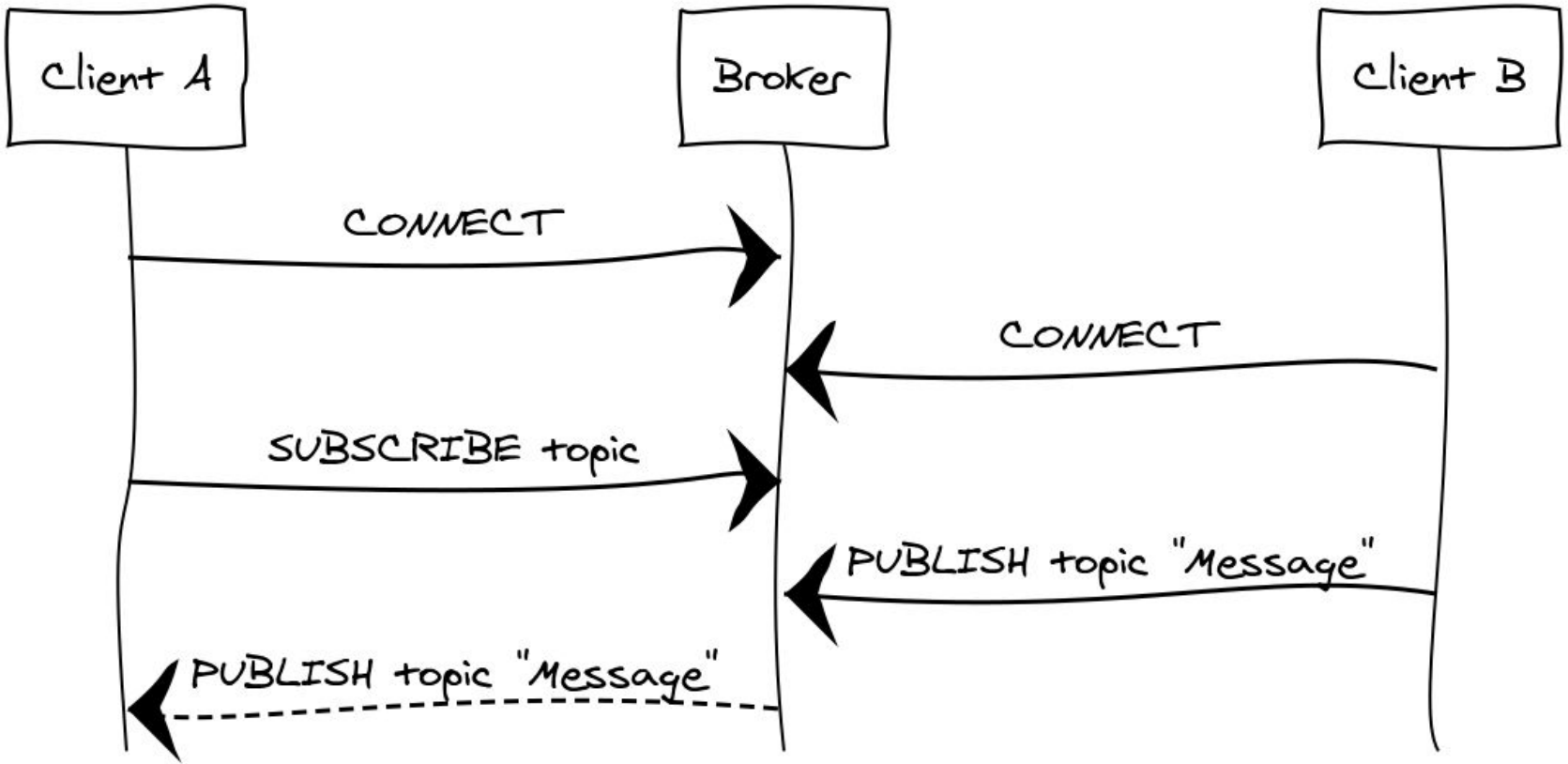
It uses TCP/IP as a transport, on port 1883 and 8883.

In MQTT, *clients* exchange *messages* via a *broker*.

Clients can be *publishers*, *subscribers* or both.

Brokers offer multiple channels, or *topics*.

MQTT


Client A
Broker
Client B
CONNECT
CONNECT
SUBSCRIBE topic
PUBLISH topic "Message"
PUBLISH topic "Message"

Debugging MQTT with Node.js *mqtt* CLI

Install the Node.js mqtt command line tool (CLI):

```
$ sudo npm install mqtt -g # adds tool to path
```

To publish/subscribe with the command line tool, try:

```
$ mqtt sub -t 'mytopic' -h 'test.mosquitto.org'
$ mqtt pub -t 'mytopic' \
-h 'test.mosquitto.org' \
-m 'Hello, world!'
```

ThingSpeak MQTT API

ThingSpeak has a device- and client-side MQTT API.

```
Host: mqtt.thingspeak.com
Port: 1883 or 8883 (or Websocket: 80, 443)

PUB -t 'channels/CHANNEL_ID/publish/\
WRITE_API_KEY' -m 'field1=42&field2=23'

SUB -t 'channels/CHANNEL_ID/subscribe/\
FORMAT/READ_API_KEY'
```

MQTT on Android

Eclipse Paho is an open source MQTT client library.

There are Android specific Java docs and examples.

An alternative is the HiveMQ MQTT client library.

Consider MQTT if there is no HTTP endpoint.

General definition of API

An API, or application programming interface, is a contract between clients and providers of a service.

Both parties have to agree on:

- How to access the service.
- How to submit data to it.
- How to get data out of it.

Good APIs are documented or self-explanatory.

Data formats

Two parties need to agree on what is valid content.

Parsing means reading individual "content tokens".

Record-based formats, e.g. CSV, are good for tables.

Text-based formats, e.g. JSON are easily readable.

Binary formats, e.g. Protobuf, are more compact.

CSV

Comma Separated Values (CSV), defined in RFC4180*.

```
file = record *(CRLF record) [CRLF];
record = field *(COMMA field);
field = *TEXTDATA;
CRLF = CR LF;
COMMA = %x2C; CR = %x0D; LF = %x0A;
TEXTDATA = %x20-21 / %x23-2B / %x2D-7E;
```

*Specified in EBNF, simplified for shortness.

JSON

JSON is a simple data format based on Unicode text:

```
{"temp": 23} // try ddg.co/?q=json+validator
```

On Arduino, use e.g. the Arduino_JSON library:

```
JSONVar obj = JSON.parse("{\"temp\": 23}");
String data = JSON.stringify(obj);
```

On Android, use e.g. json.org, Gson or Moshi.

```
JSONObject obj = new JSONObject(data);
String data = obj.toString();
```

Protobuf

Protocol Buffers (Protobuf) is a binary data format:

```
message Measurement {
  required int32 temp = 1;
  optional int32 humi = 2;
}
```

Message schemas are compiled to a target language, i.e. a parser is generated, to be called from your code.

Try this JSON to Protobuf schema converter.

Information model

The information model defines how data is structured.

It's the "common denominator" of all involved parties.

Data formats define how data is transported or stored.

An information model is more about data semantics*.

*E.g. ThingSpeak knows channels and fields.

Summary

IoT platforms receive and provide data via an API.

E.g. through HTTP requests or MQTT messages.

Data formats allow to write and read payloads.

The information model defines semantics.

Feedback or questions?

Join us on MSE TSM MobCom in MS Teams

Or email thomas.amberg@fhnw.ch

Thanks for your time.