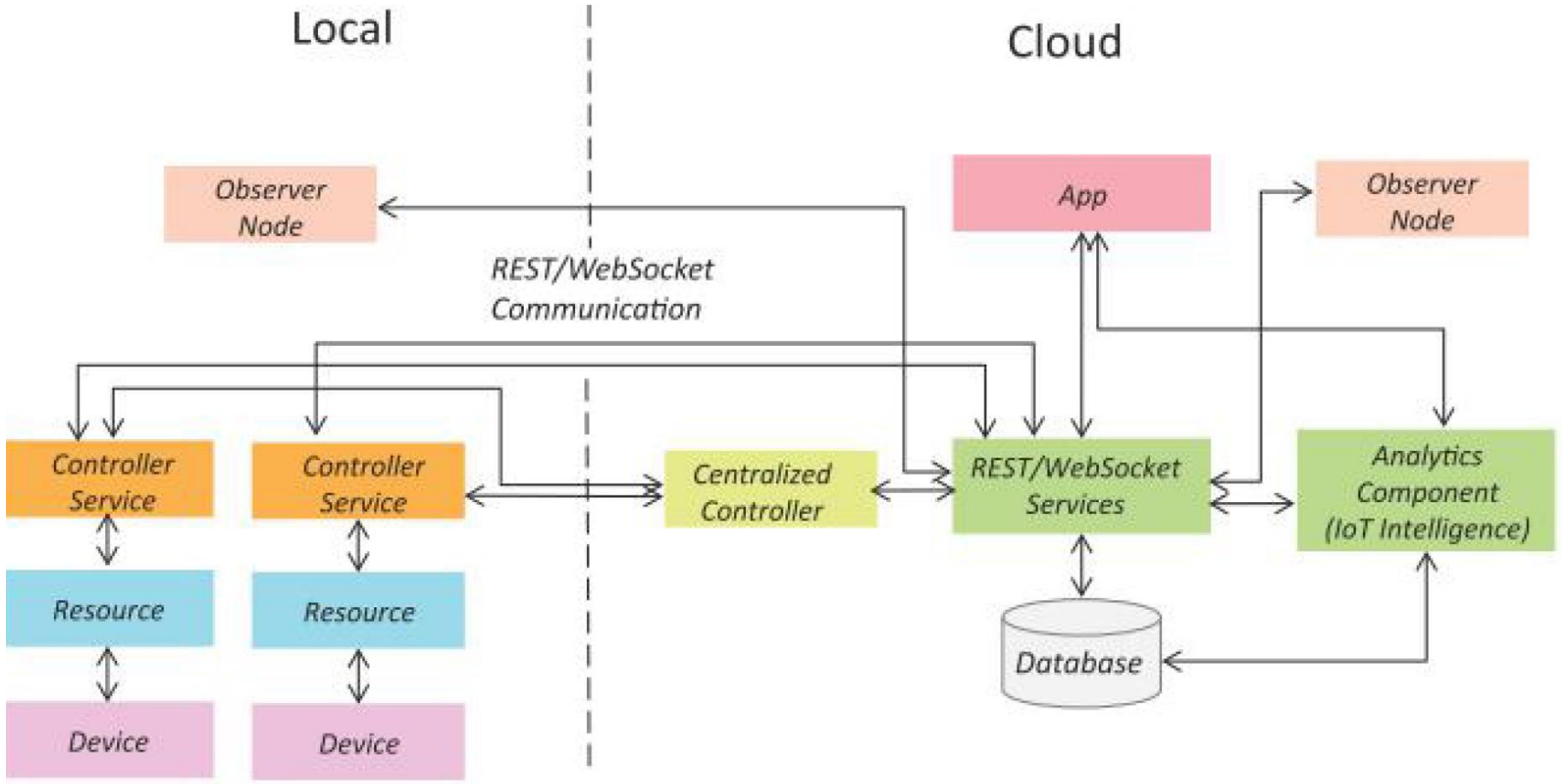
IoT and Cloud
Protocols and middleware

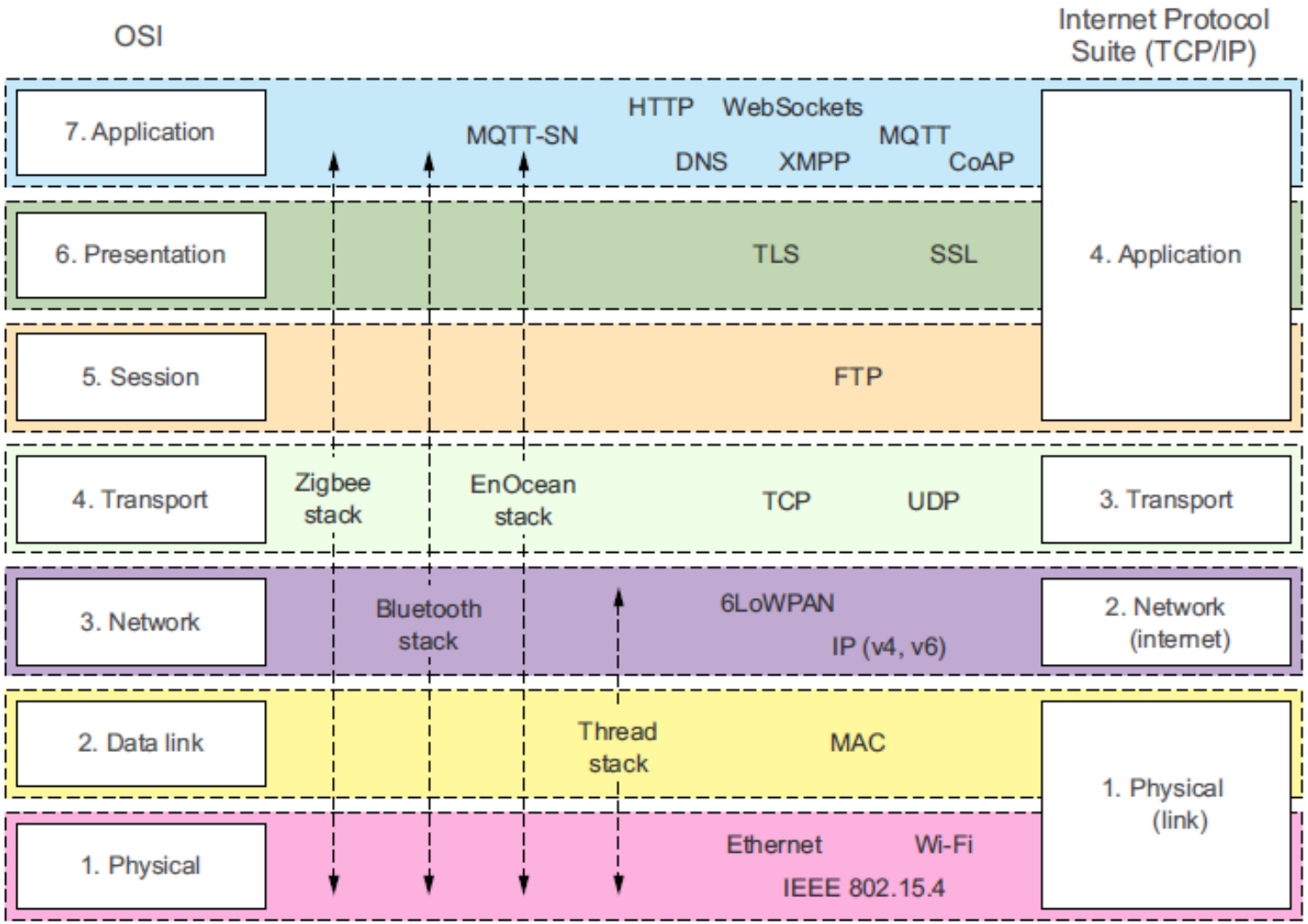
IoT system architecture

Software for web and cloud integration

Software components for sensor/actuator interaction and control



IoT networking protocols



IPv4 and IPv6

- **IoT will inevitable cause the number of connected devices to explode – and cause a shortage of IP addresses**

IPv4 address:

146.200.15.222

8 bits

4 x 8 bits = 32 bits

$= 2^{32} = \sim 4.3$ billion addresses

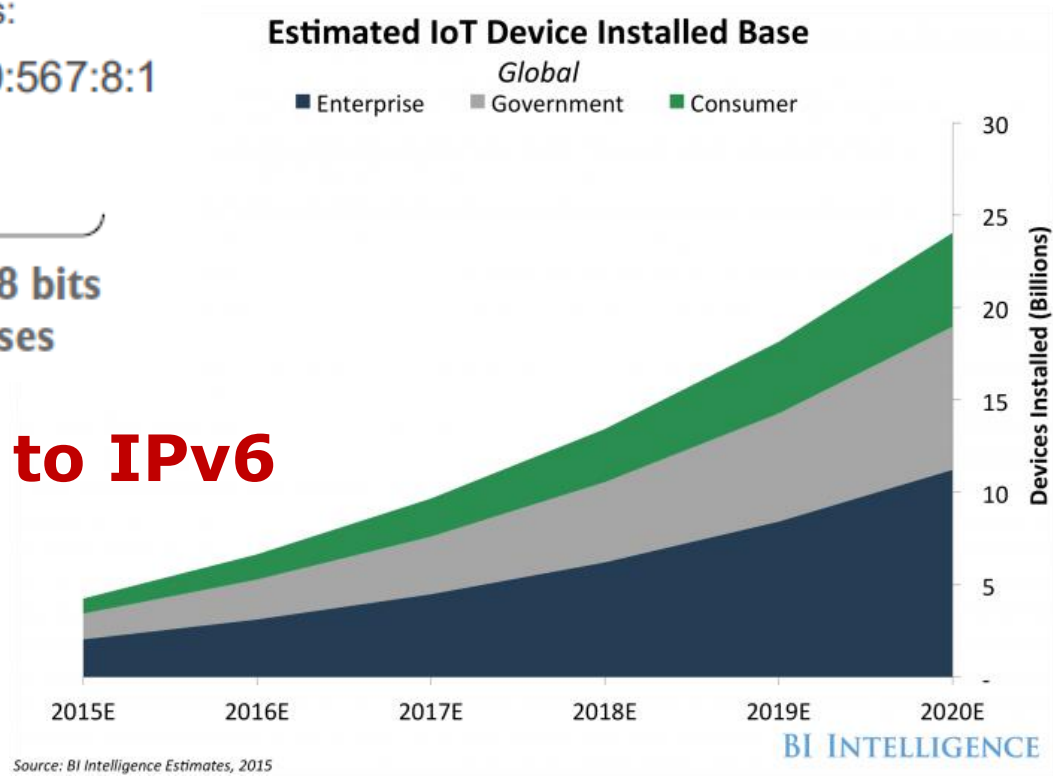
IPv6 address:

2001:db8:0:1234:0:567:8:1

16 bits

8 x 16 bits = 128 bits

$= 2^{128}$ addresses



- **Gradual transition from IPv4 to to IPv6**
  - **Larger address space**
  - **Improved security mechanisms**
  - **Improved auto-configuration**
  - **…**

Transport Protocols: UDP and TCP

- **Supports for end-to-end (host-to-host) communication**
  - An IP address and a port number identifies an endpoint
  - Based on the client-server architecture and supported by a socket API
  - Sockets are the endpoints used for sending and receiving data
- **UDP - User Datagram Protocol**
  - Connectionless protocol for datagrams
  - Can be used for unicast and multicast
  - Unreliable: duplication, loss, and reordering may occur
- **TCP - Transport Control Protocol**
  - Connection-oriented protocol sending/receiving a byte stream
  - Flow and congestion controls using sliding windows
  - Guarantees reliable and ordered delivery of a stream of bytes

Connecting IoT devices

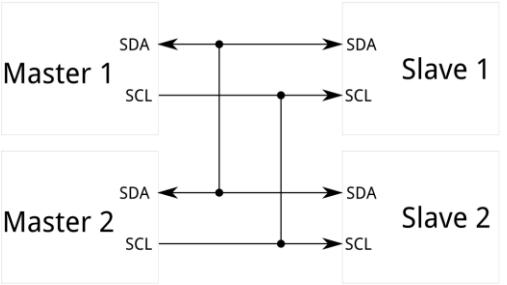
Arduino Uno - IoT device

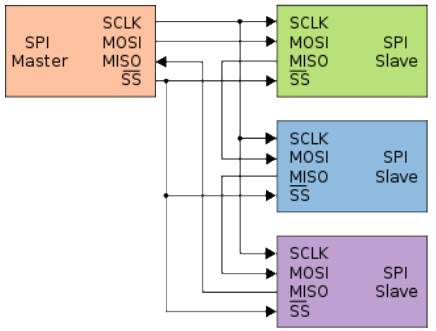
- **A network card / controller (link-layer) is required to for the device to get network (Internet) connectivity**

Microcontroller-based
ATmega128 microcontroller (8-bit – 16 MHz)
32Kb (Flash), 2kb (SRAM), EEPROM (1kb)
No operating system: application controls the hardware directly
USB programming interface
https://www.microchip.com/wwwproducts/en/ATmega328

Connecting microcontrollers

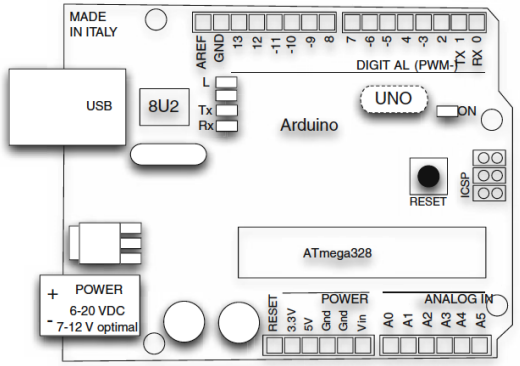
- **Inter-Integrated Circuit Protocol (I2C)**
  - Master/slave bus for IC-microcontroller communication
  - Masters coordinate the read/write bus transactions
  - Supported by the Wire library on the Arduino platform





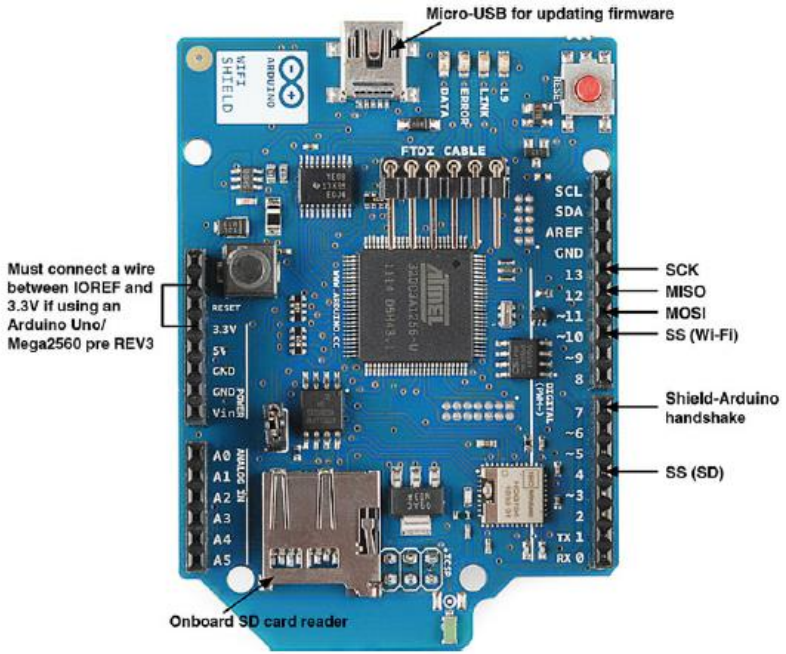
- **Serial Peripheral Interface Protocol (SPI)**
  - Data bus for synchronous serial communication
  - Support by the SPI library on the Arduino platform

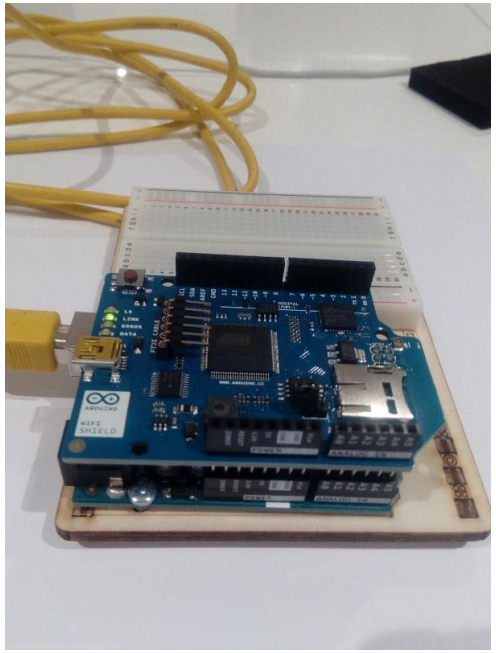
- **Controller Area Network (CAN)**
  - Automotive and industrial automation domain



Example: WiFi and Arduino

- **A daughter board (shield) is stacked on top of the main Arduino board and pins connecting according to protocol**





- **Software libraries (device drivers) are used to simplify configuration and support TCP/IP communication.**

Dekning

Nå er verdens raskeste mobilnett også optimalisert for ting og sensorer.

Abonnement

Våre IoT-abonnement dekker alle typer kommunikasjonsbehov.

NarrowBand

NarrowBand IoT gir dekning på tidligere utilgjengelige steder og dramatisk forlenget batteritid på sensorene.

Kjøp ferdige IoT-løsninger

I vår IoT-katalog finner du ferdige løsninger som er enkle å ta i bruk.

Utviklere

De beste idéene fortjener også det beste nettet, derfor ønsker vi utviklere velkommen til oss.

IoT-sikkerhet

IoT-utstyr har det samme høye sikkerhetsnivået på datatrafikken som vanlige mobiltelefoner i Telenors nett.

Basic Communication

Virtual IoT Devices

- **We do not have (sufficient) physical IoT devices (Arduinos) available for the course.**
- **For experimenting with IoT protocols and middleware we will use virtual sensors, actuators, and devices**
  - The reading/writing to sensors/actuators will be simulated in software (Java)
  - A virtual IoT device will use be associated with one or more virtual sensors/actuators and use TCP/UCP/… for communication.
- **Example system**
  - A virtual temperature sensor with readings following a sinus curve
  - A virtual display for printing message to the console
  - Relies on UDP for communication between the two virtual IoT devices
  - https://github.com/lmkr/dat159/tree/master/iotprotocols/virtualdevices

Example: UDP Communication

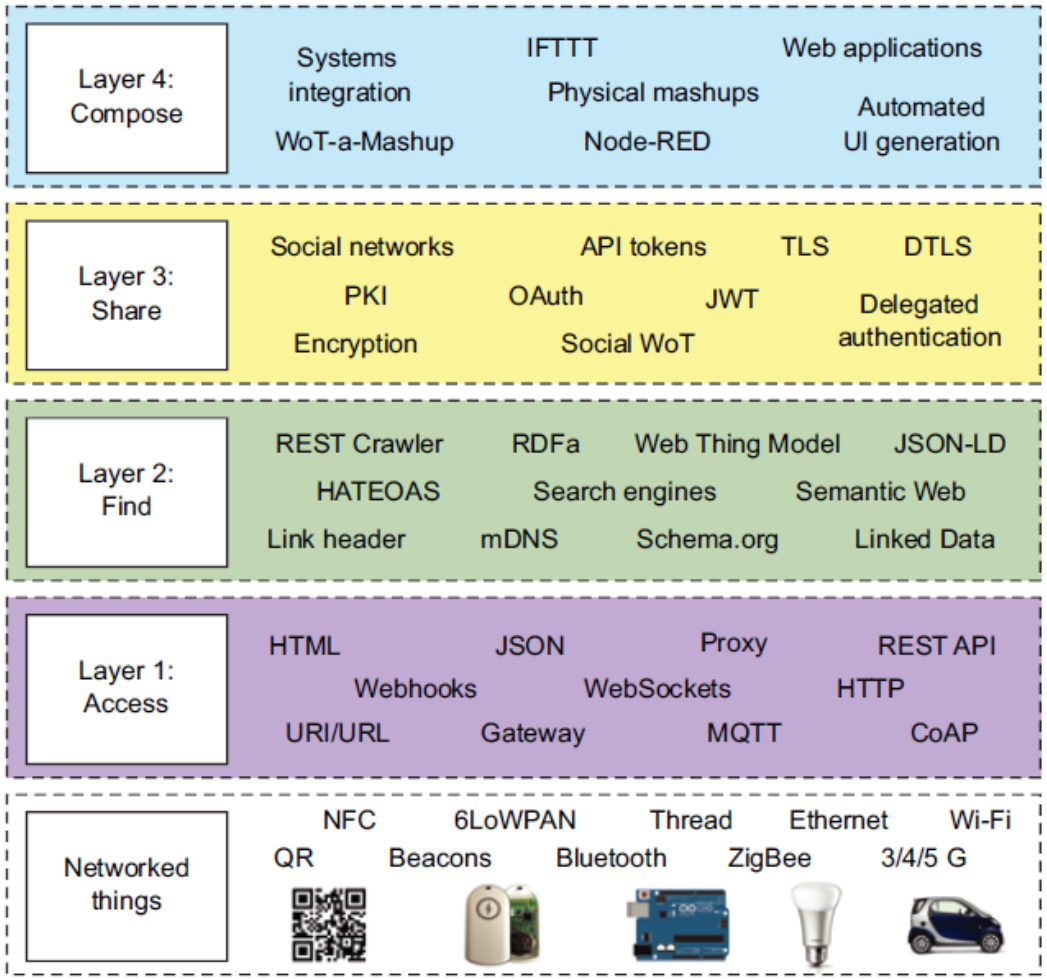
- **The IoT device with temperature sensor**
  - Uses UDP to regularly send the current temperature
  - Plays the roles of an UDP client
- **The IoT device with a display (actuator)**
  - Uses UDP to receive incoming temperatures
  - Plays the role of an UDP server

Web-of-things (WoT)

*«The **Internet of Things** is a system of physical objects that can be discovered, monitored, controlled, or interacted with by electronic devices that communicate over various networking interfaces and eventually can be connected to the wider internet».*

*«The **Web of Things** is a refinement of the Internet of Things by integrating smart things not only into the Internet (network), but into the Web Architecture (application)».*

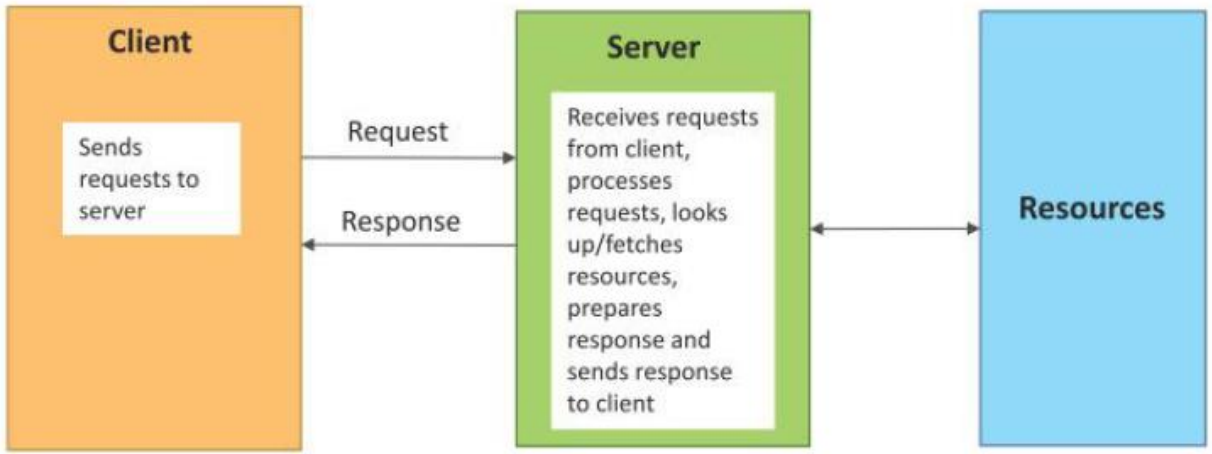
Web of Things (WoT)



- **4: Integration of data and services to build applications**
- **3: Efficient and secure sharing of data between things (big data)**
- **2: Semantic layer to discover and understand and what things are**
- **1: Connecting applications using APIs based on web-technology**
- **0: Internet-of-things – connecting devices**

HTTP (Hyper Text Transfer Protocol)

- **Transfer protocol following a request-response pattern and providing the foundation of the web**



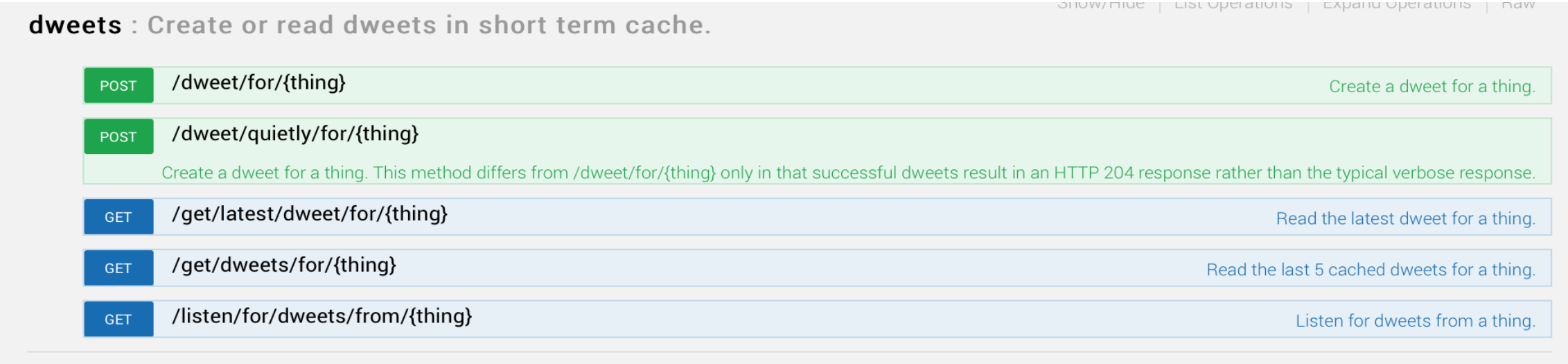
Operations - verbs/methods

- **GET: safe and idempotent operation to retrieve a resource**
- **POST: unsafe and non-idempotent operation to create a resource**
- **PUT: unsafe and idempotent operation to update a resource**
- **DELETE: unsafe and idempotent operation to delete a resource**
- **…**

- **IoT devices can play the role of both servers and clients - in many cases clients due to firewalls**

Example: dweet.io https://dweet.io/

- **An HTTP-based API for sending messages and receiving alerts for IoT using JSON**



- **Example: dat159-sensor (a thing) and a temperature in JSON:** `{ "Temperature": 30 }`

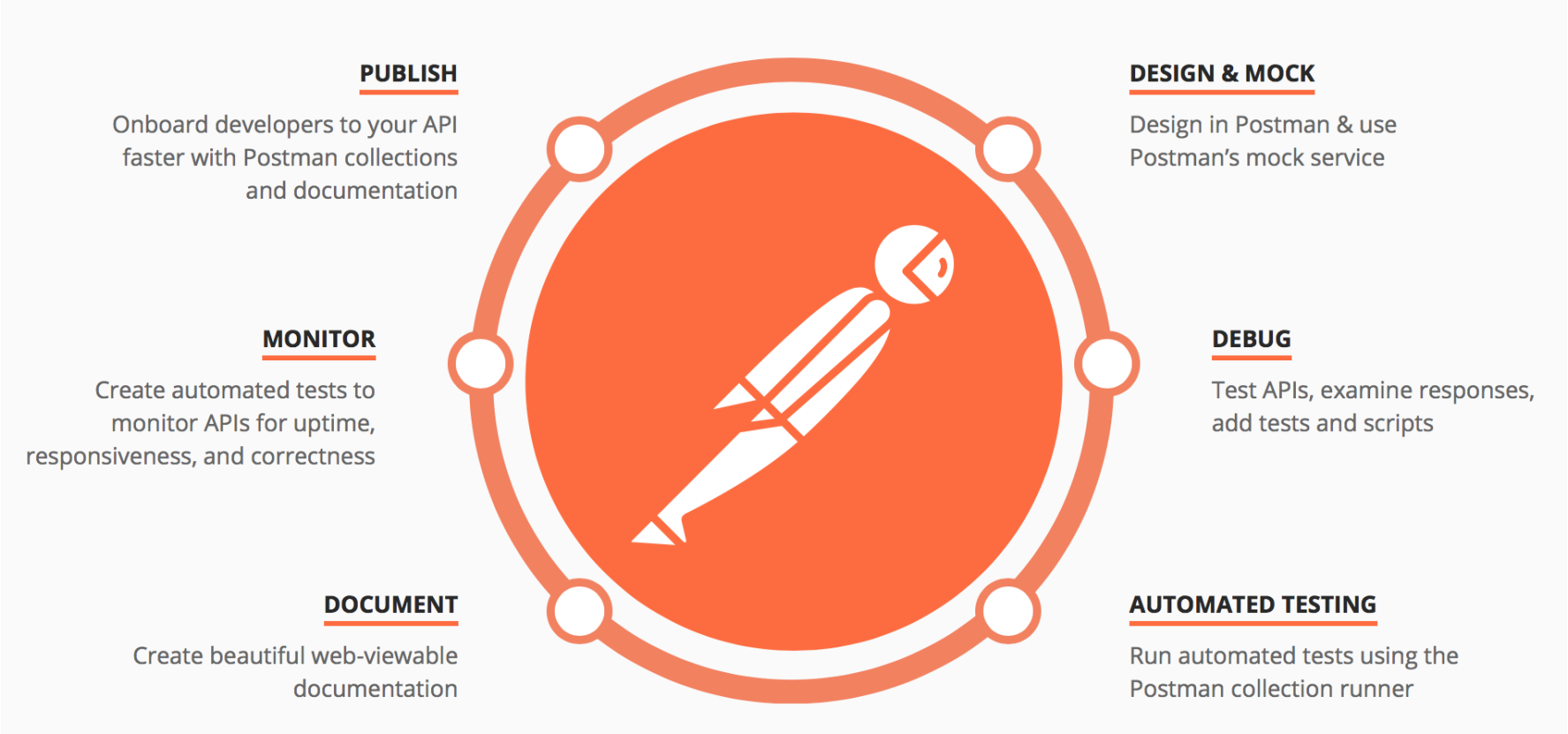
IoT Dweet - Java Example

- **Example code**
  - https://github.com/lmkr/dat159/tree/master/iotprotocols/dweet

- **Implementation**
  - A virtual temperature sensor and display (as before)
  - DweetClient.java for following the API for communication with dweet.io
  - IoTDweetThing.java for regularly reporting temperature to dweeo.io
  - IoTDweetDisplay.java for showing the recent temperature from dweeto.io

Postman https://www.getpostman.com

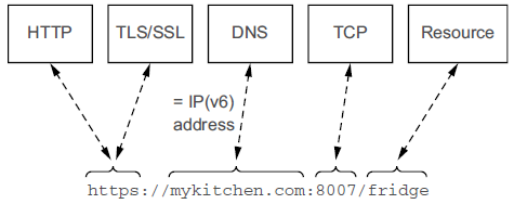
- **A tool for development with, and testing of, web APIs**



Beyond HTTP

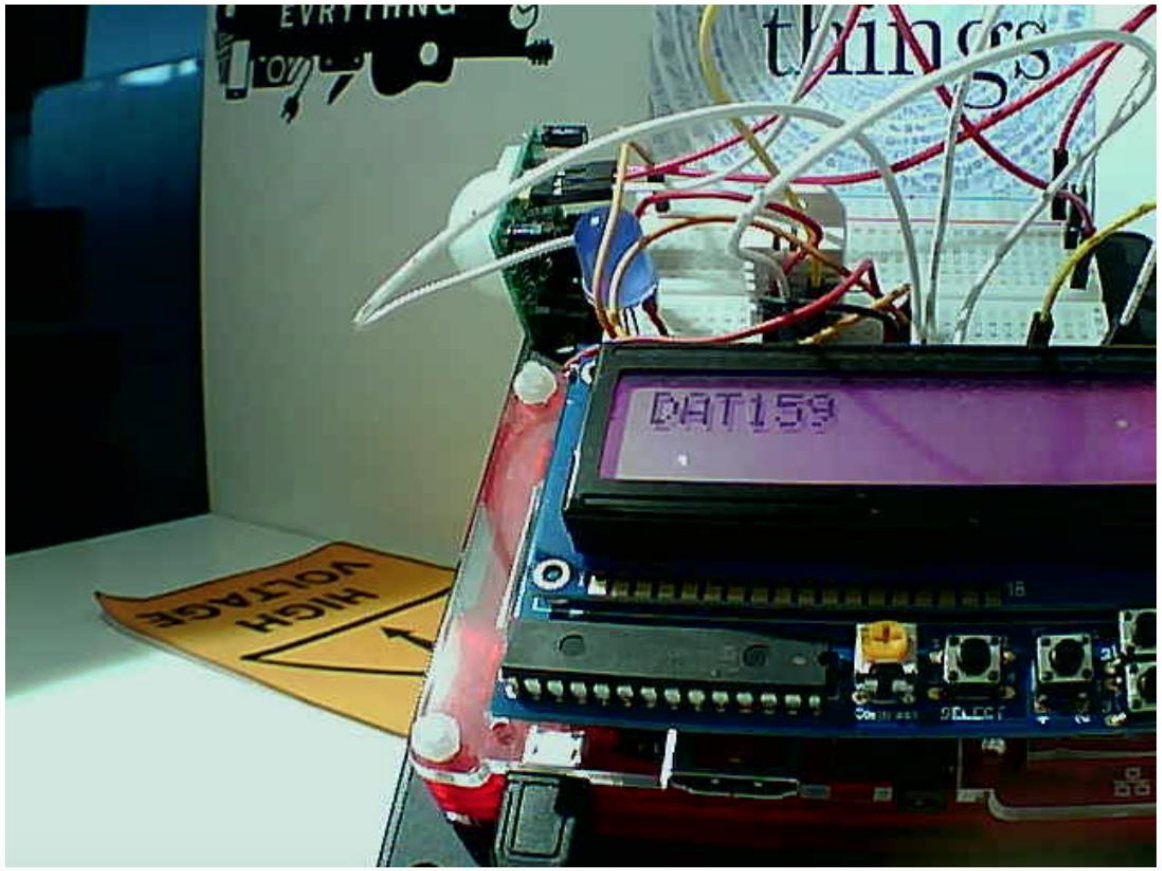
REST (representational state transfer)

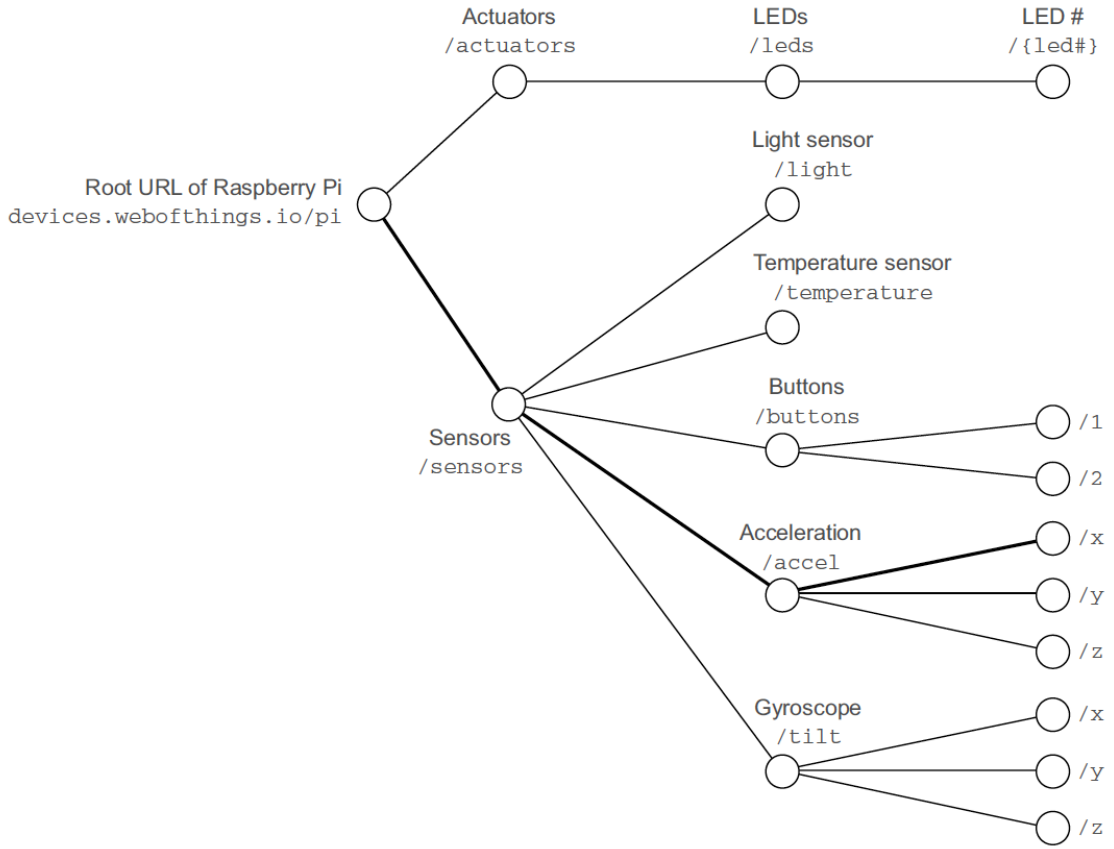
- **A general design principle for accessing Internet resources and providing web services and HTTP-based APIs**
- **The four design principles of REST**
  - **P1:** addressable resources based on URI/URLs (identity / location)
  - **P2:** manipulation of resources through (tangible) representations (XML,JSON,…)
  - **P3:** self-descriptive message using standardised operations (GET,POST,,…)
  - **P4:** the state of a resources can be obtained via linked sub-resources



- **REST applied to IoT for building the WoT**
  - IoT devices become navigateable resources accessible via URIs
  - IoT devices becomes accessible via HTTP-based REST APIs and web servers

WoT Example [http://devices.webofthings.io/camera/]



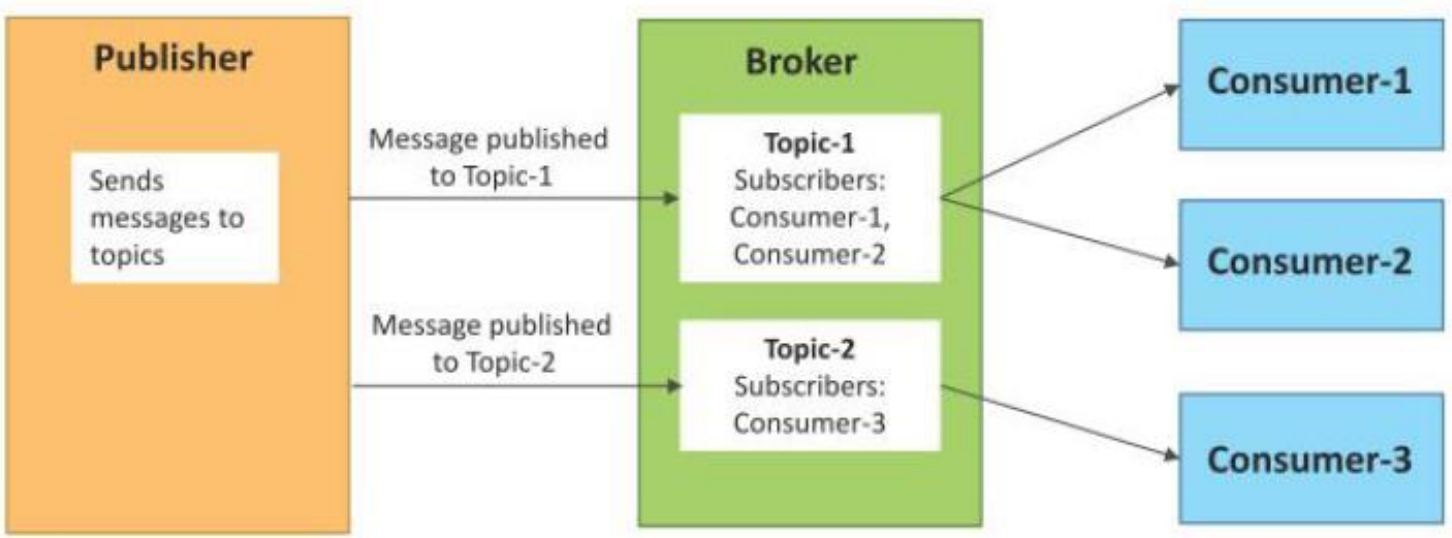


http://devices.webofthings.io/pi/

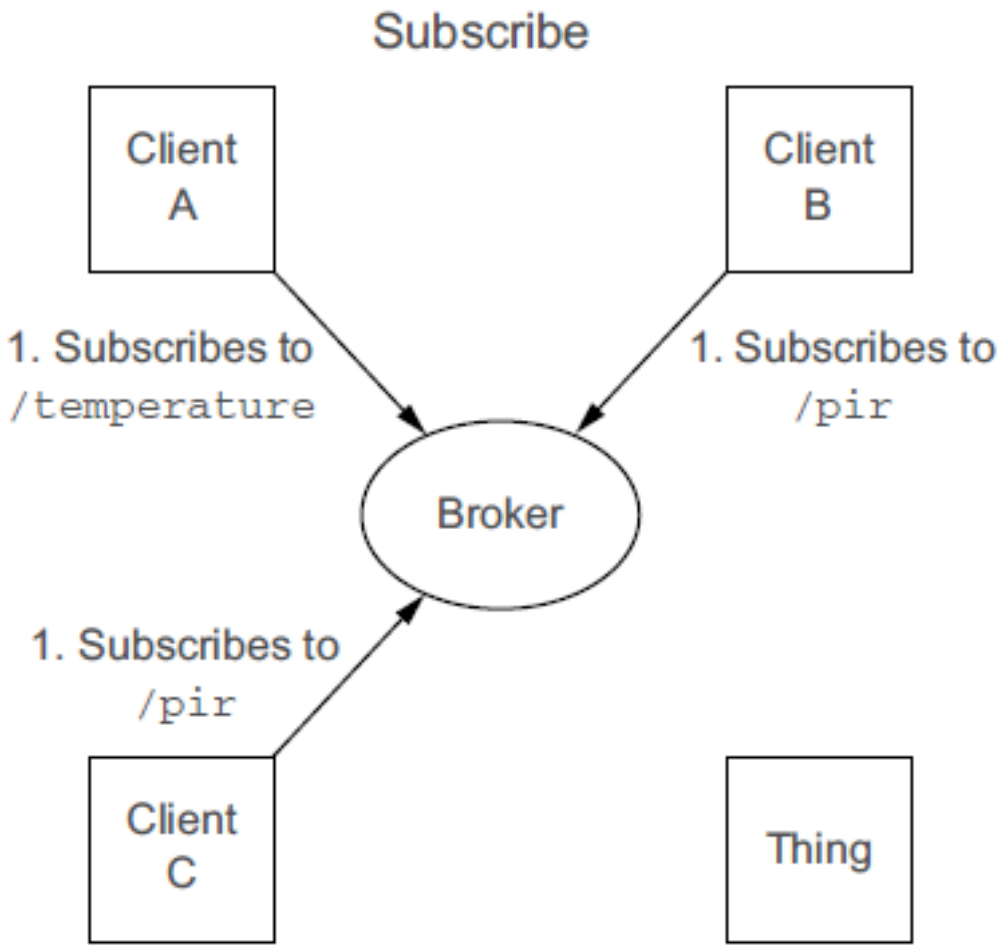
Beyond REST

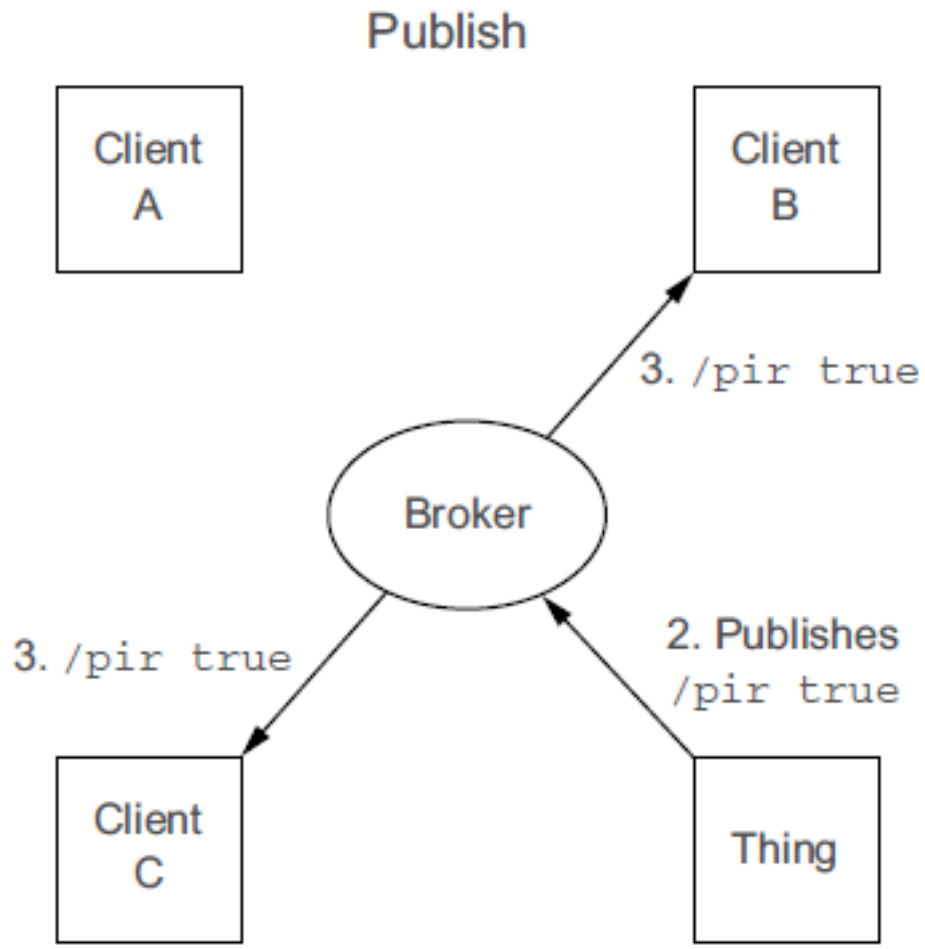
Publish-Subscribe Messaging

- **Software architectures for IoT/WoT are required to be reliable, scalable, and flexible.**
- **Messaging supports architectures consisting of loosely coupled communicating components.**
- **Based on an asynchronous communication model involving publishers, brokers, topics, and consumers.**



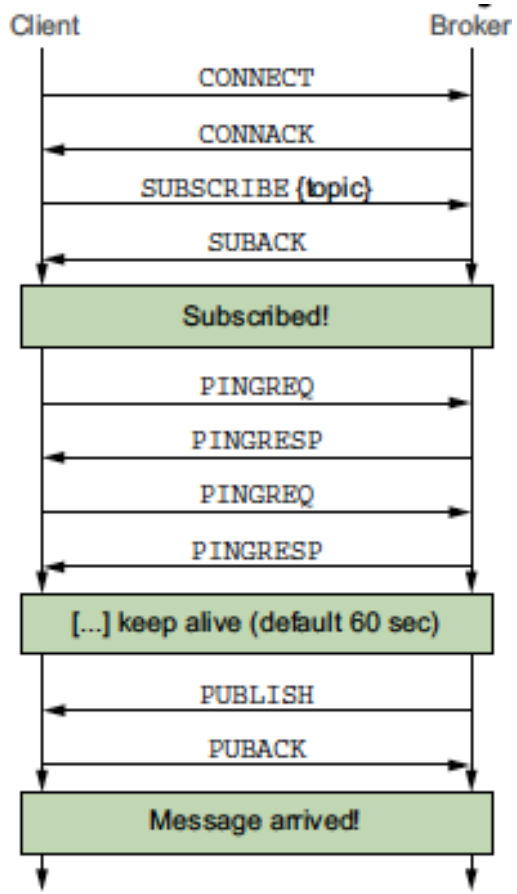
Publish-Subscribe WoT/IoT





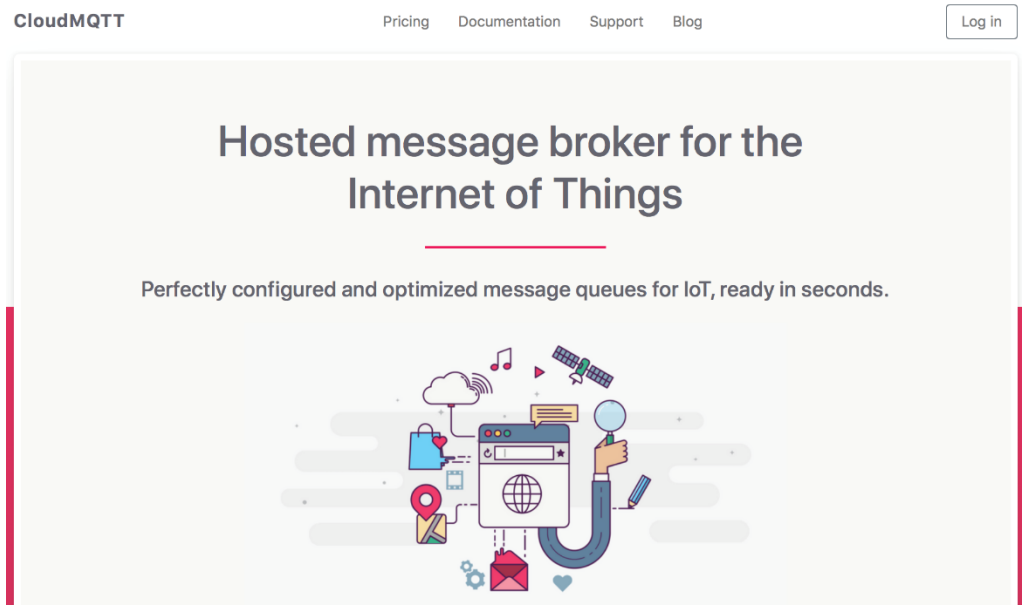
MQTT (Message Queuing Telemetry Transport Protocol)

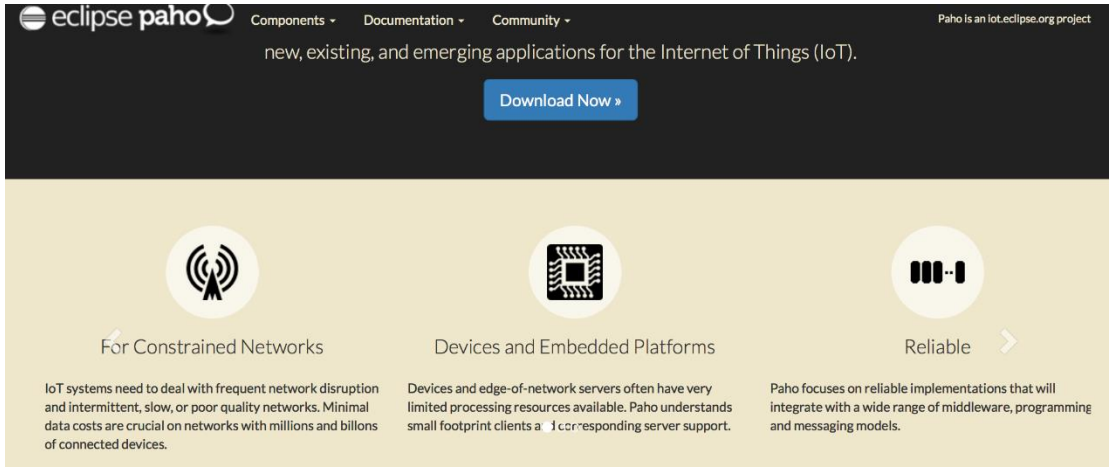
- **Publish-subscribe protocol for M2M communication between resource constrained devices.**
- **Designed for using TCP or UDP as the underlying transport protocols**
- **Three quality of service levels**
  - QoS 0 - no guarantee for message delivery
  - QoS 1 – deliver message at least once (idempotent messages / operations)
  - QoS 2 – deliver message exactly once (non-idempotent messages / operations)



Example: CloudMQTT

- **Cloud-hosted broker for MQTT: https://www.cloudmqtt.com**





- **Will used this together with the Eclipse Paho MQTT library in hands-on assignment part B https://www.eclipse.org/paho/**

Warm-up for hands-on B

- **Preliminary description is available via github**
  - https://github.com/lmkr/dat159/blob/master/week2.md
  - Start code is also available: will be discussed at the lab / lecture on friday
- **Implement an MQTT-based IoT system consisting of**
  - A room where we want to keep the temperature within certain limits
  - A controller service (software component) for controlling the temperature
  - IoT devices: A temperature sensor, a heating element, and a display.
- **Now – joint design exercise and discussion**
  - what topics are required?
  - what role should each IoT device have (publishers,subscribers)?
  - What role should the controller service have?

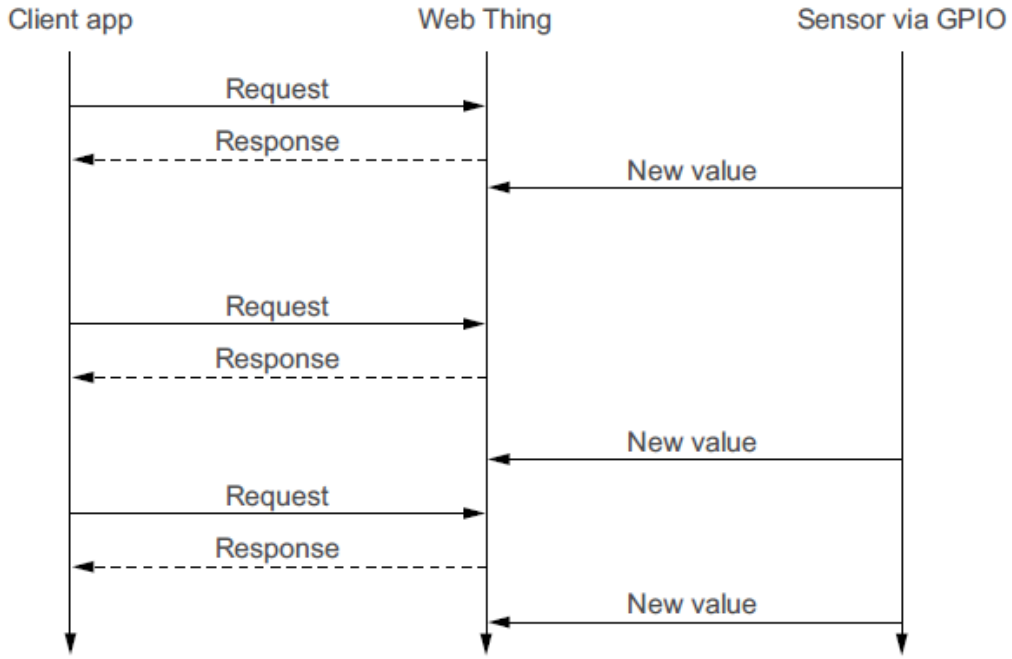
CoAP and Websockets

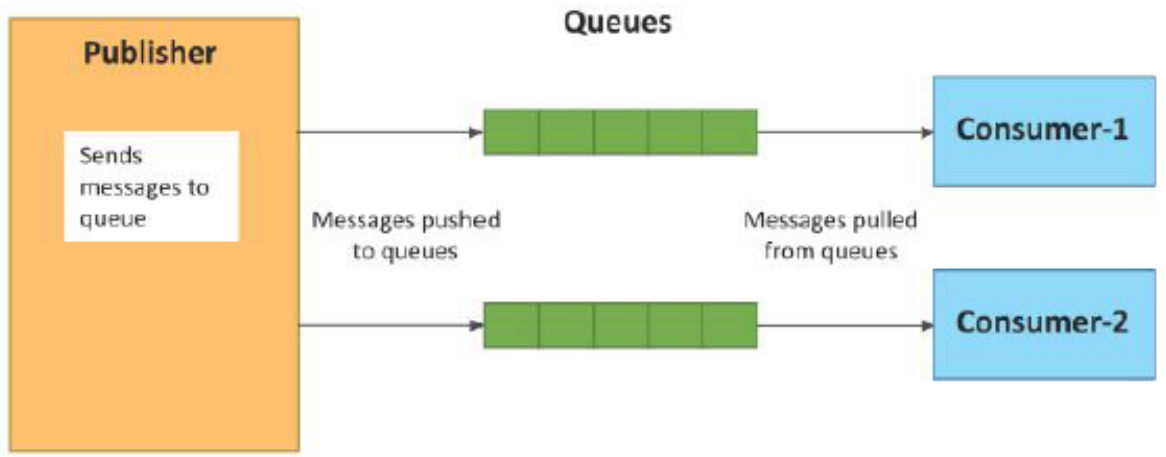
CoAP (Constrained Application Protocol - http://coap.technology)

- **HTTP requires opening and maintaining TCP connections – consumes resources: CPU, memory, and power**
- **For many application the reliability of TCP is not required – but the use of a REST-based API is desirable**
- **COAP is a request-response protocol that makes it possible to provide REST APIs over UDP**
  - Supports URIs and REST verbs / operations (GET, POST, PUT, and DELETE,…)
- **Messaging model**
  - Reliable message transport using confirmable messages and acknowledgments
  - Unreliable message transport using non-confirmable messages

HTTP Request-Response

- **The request-response pattern of HTTP has limitations in real-time and event-driven communication**

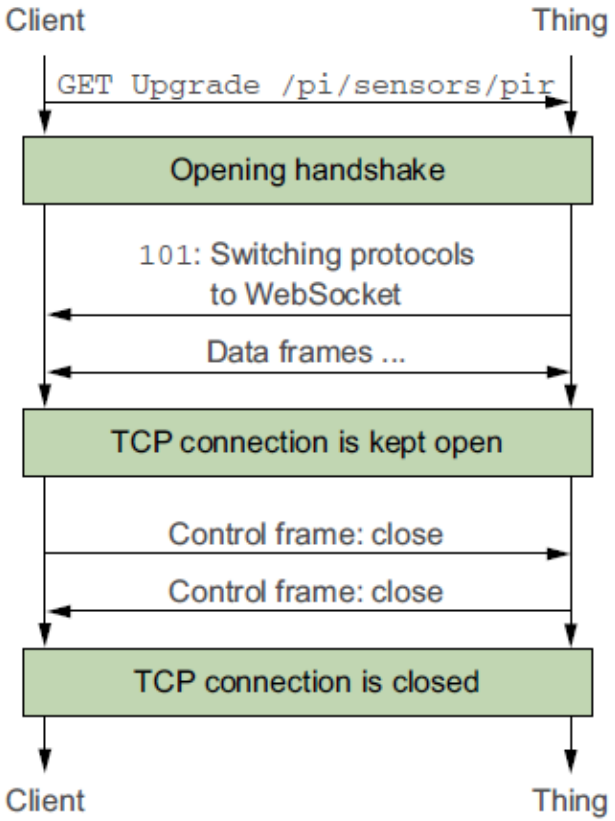




- **Often results in a push-pull communication pattern.**

WebSocket Protocol

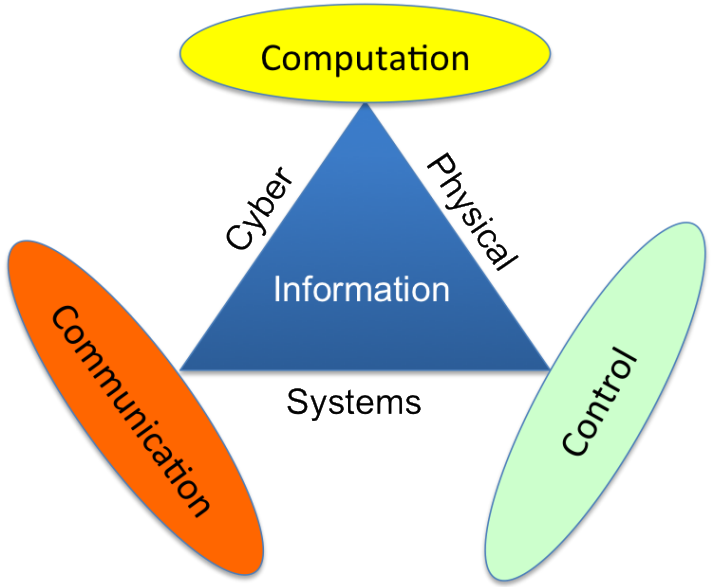
- **Allows an HTTP connection to be upgraded to full-duplex communication of data frames**

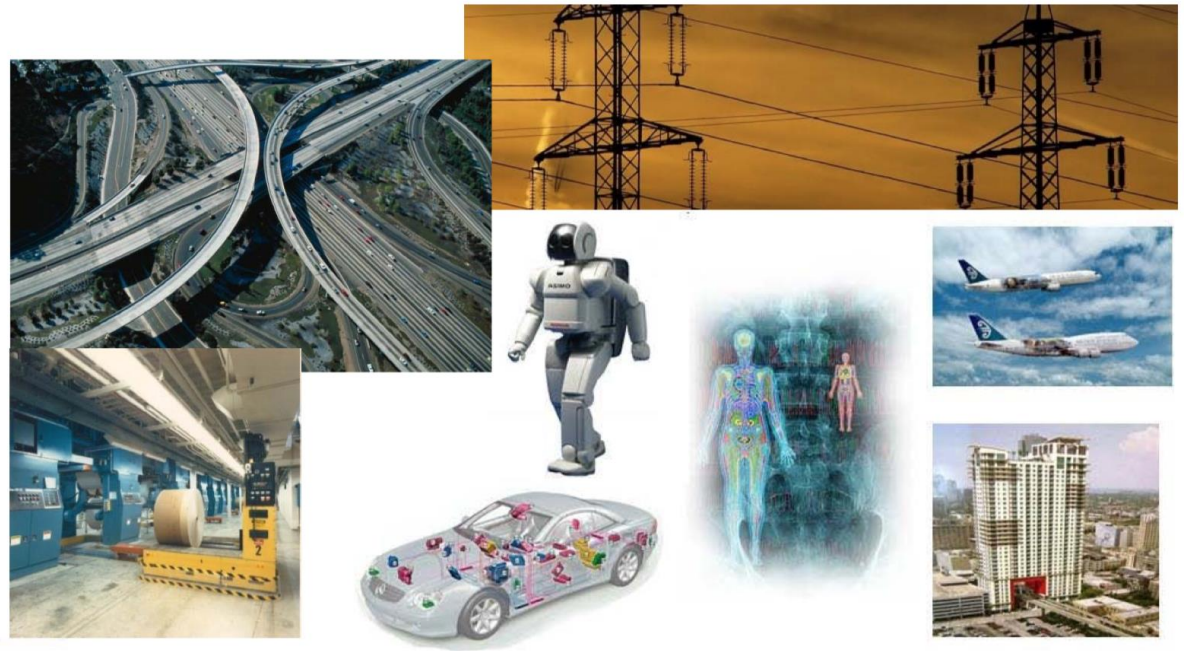


- **HTTP is used for opening the Websocket connection eliminating firewall issues.**
- **Application-specific protocols can be supported on top of HTTP/TCP.**
- **Low overhead per dataframe (2 bytes vs. ~900 bytes in HTTP).**
- **Websocket can also be used as transport protocol for other protocols such as MQTT.**

Cyber-Physical Systems (CPPs)

- **Integration computation, networking, and physical processes in a feedback control and monitoring loop.**
- **A coordinated network of embedded systems with physical input (sensors) and output (actuators).**





Industry 4.0

- **Combining CPS, robotics, industrial IoT, the Internet of systems, big data ,…**
- **Smart factory: web connected and autonomous machines**