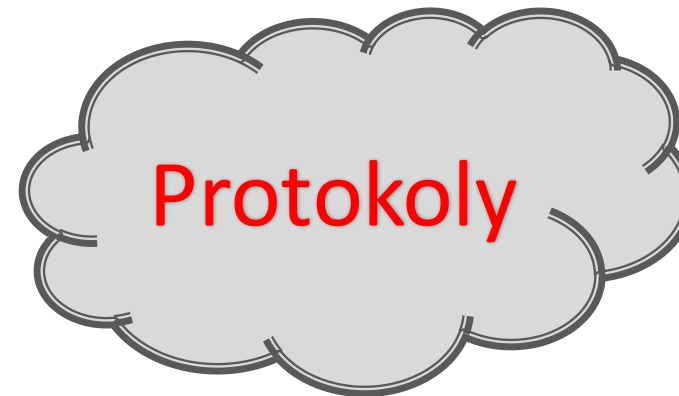
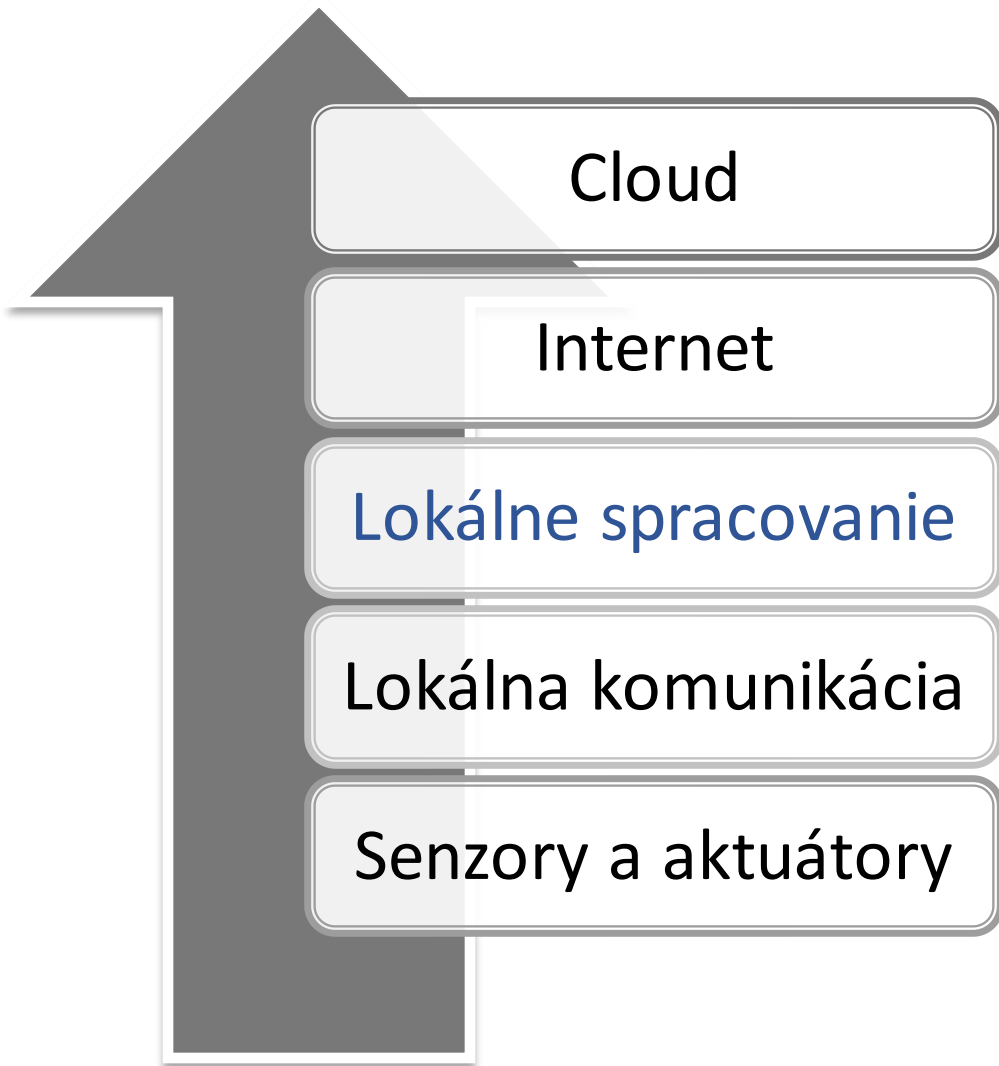


# Aplikačné protokoly IoT: MQTT

## Základy internetu vecí ÚINF PF UPJŠ

# IoT - komponenty



# IoT - komponenty

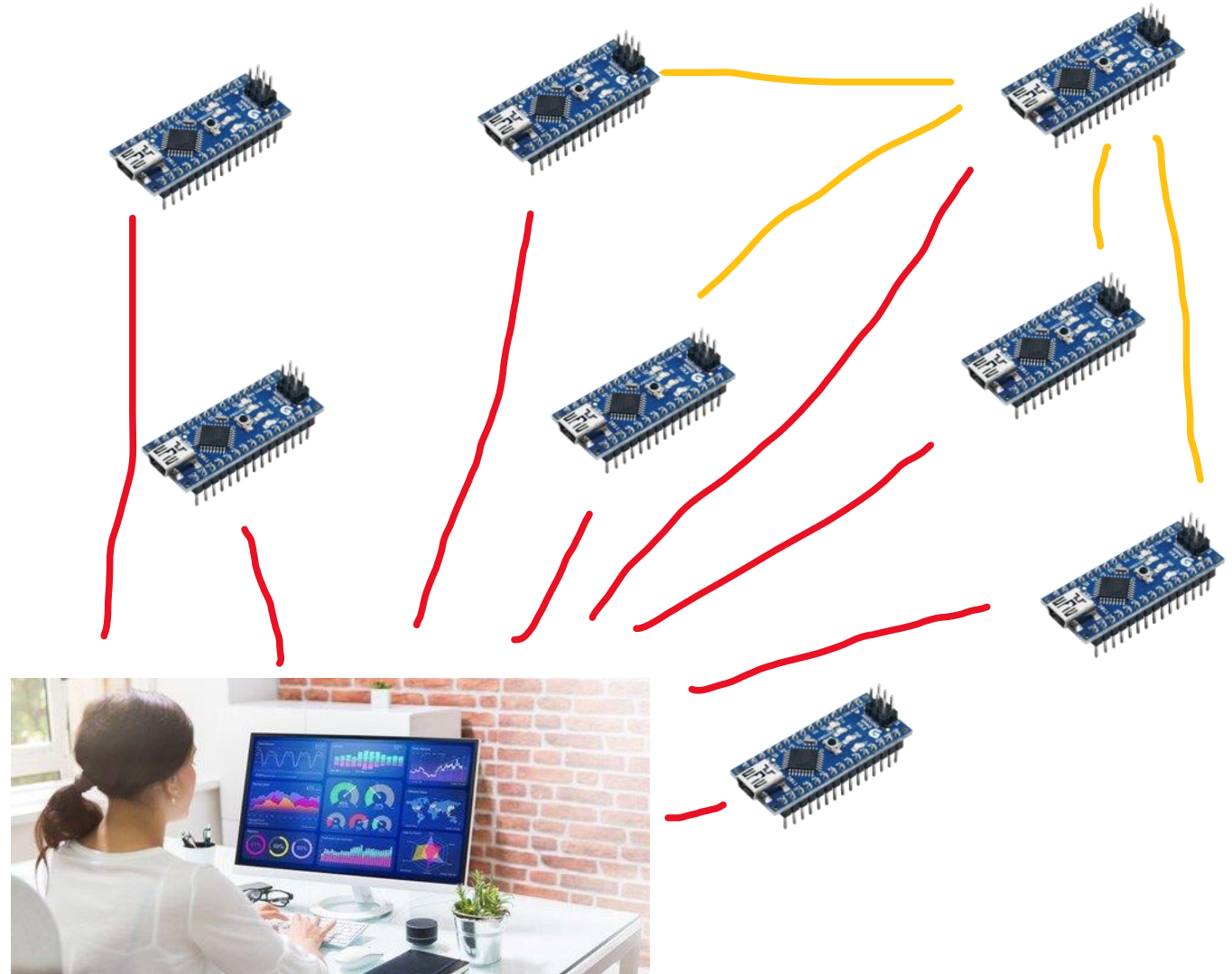
Cloud

Internet

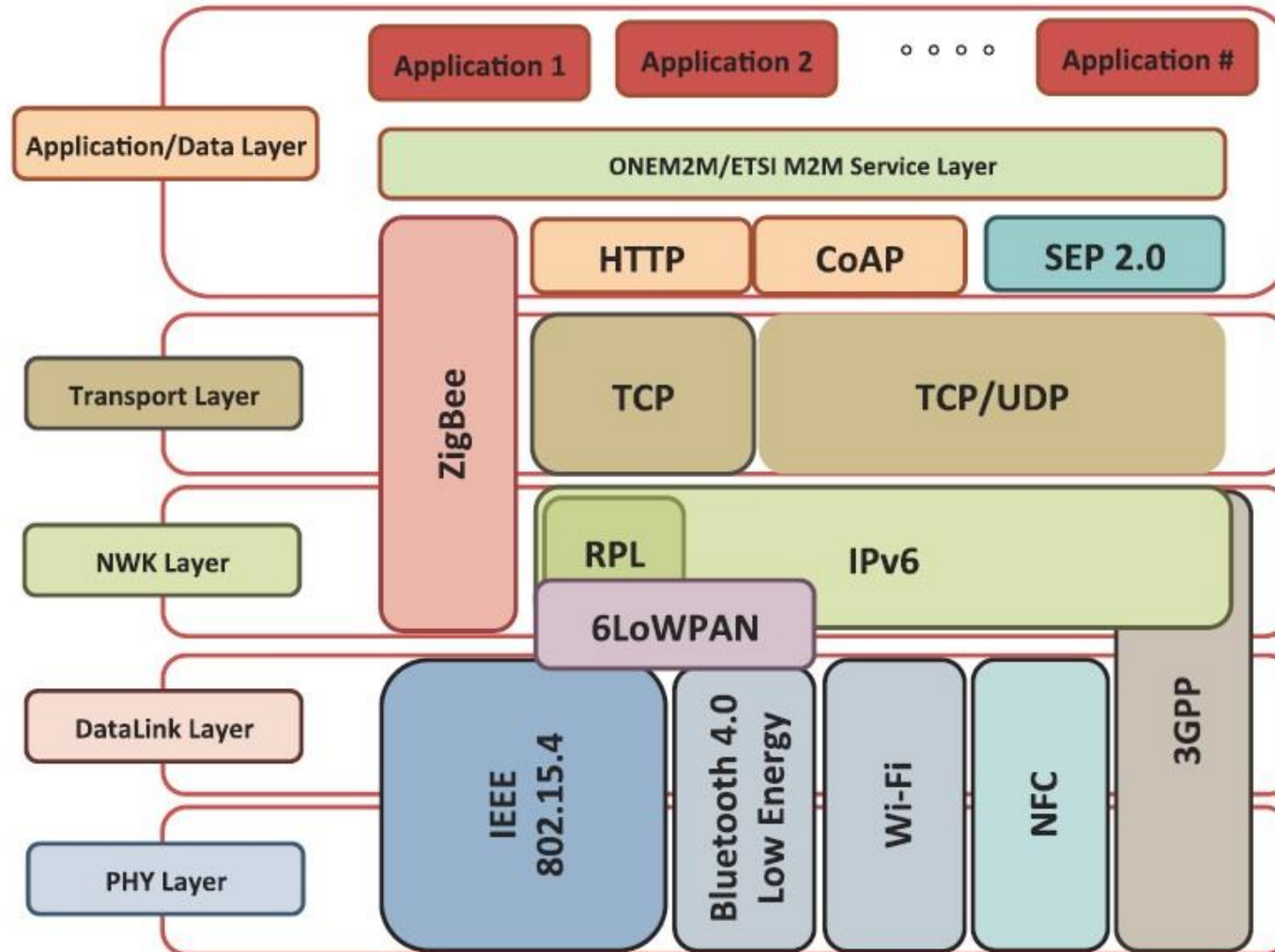
Lokálne spracovanie

Lokálna komunikácia

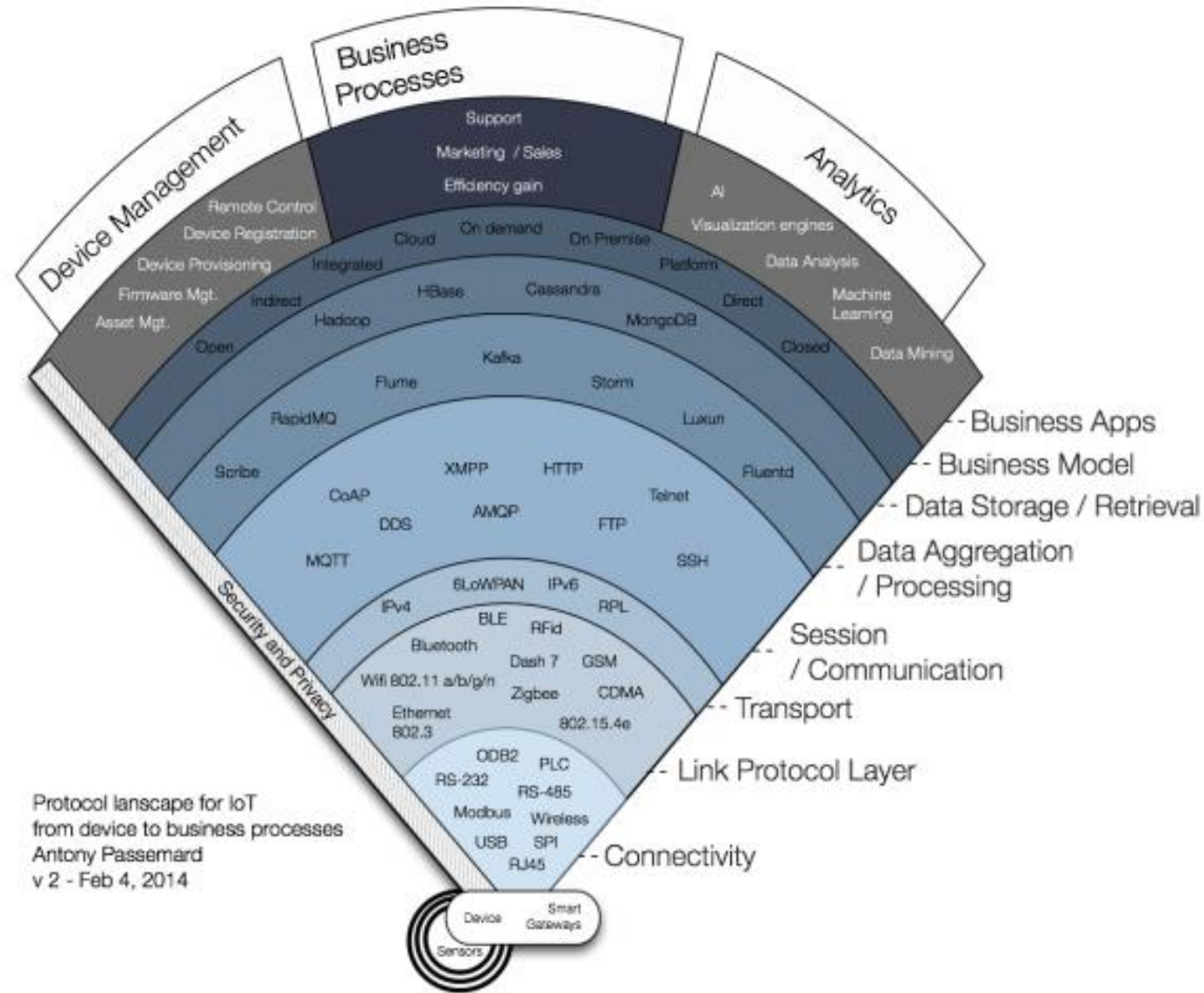
Senzory a aktuátory



# IoT protokoly

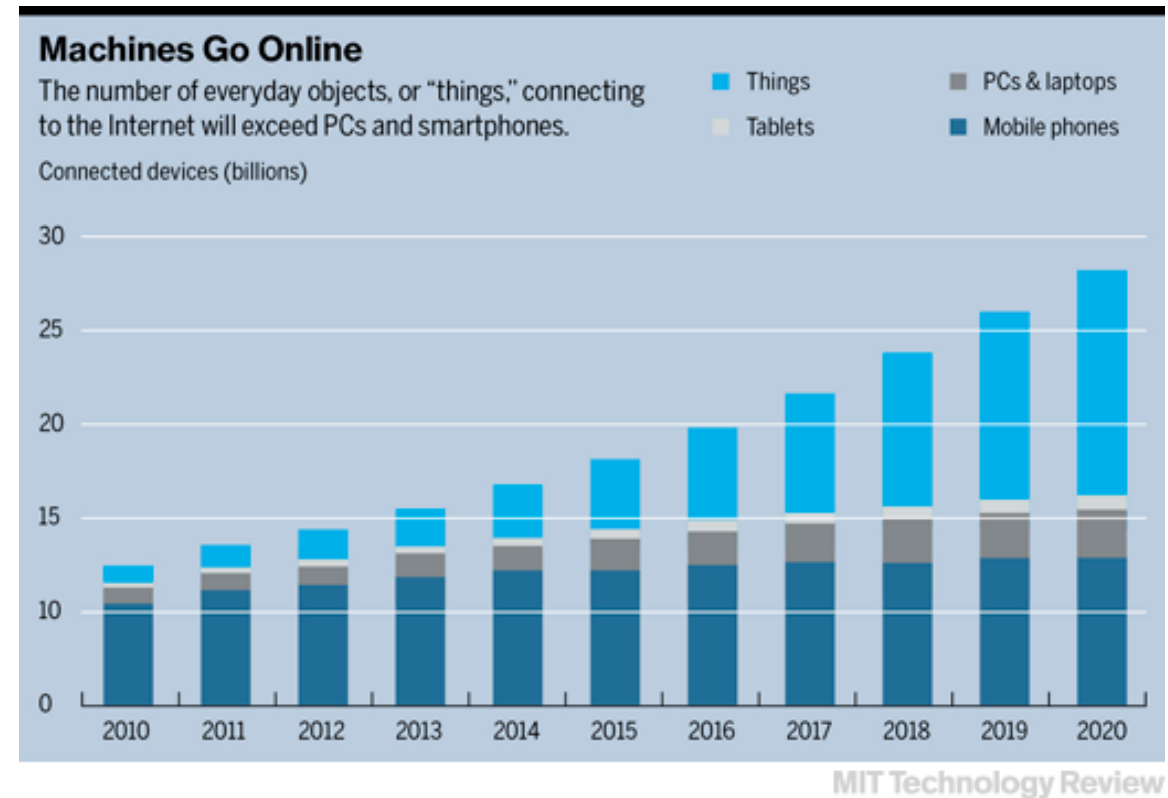


# Protokoly vo svete IoT

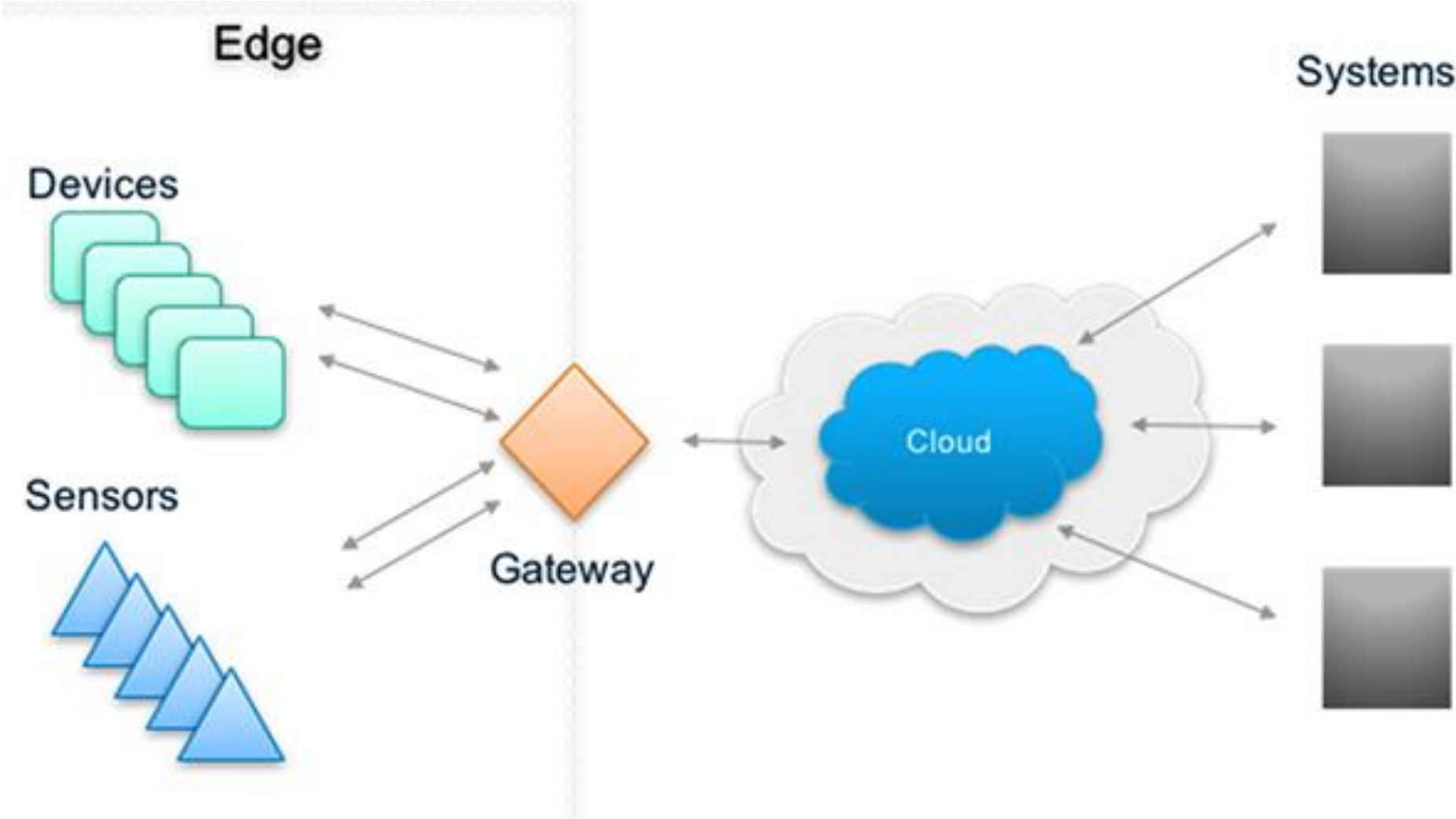


# Prečo nové protokoly?

- Masívny **nárast počtu** pripojených zariadení
- **Obmedzená** výpočtová sila zariadení
- Požiadavka na **nízku spotrebu** (napájanie batériami)
- Nespoľahlivé siete
- Nárast **M2M** (machine-to-machine)
- Požiadavka na lokálnu komunikáciu (edge-computing) nielen zariadenie ↔ cloud
- Internet bol vytvorený pre (osobné) počítače a servery



# Edge gateway



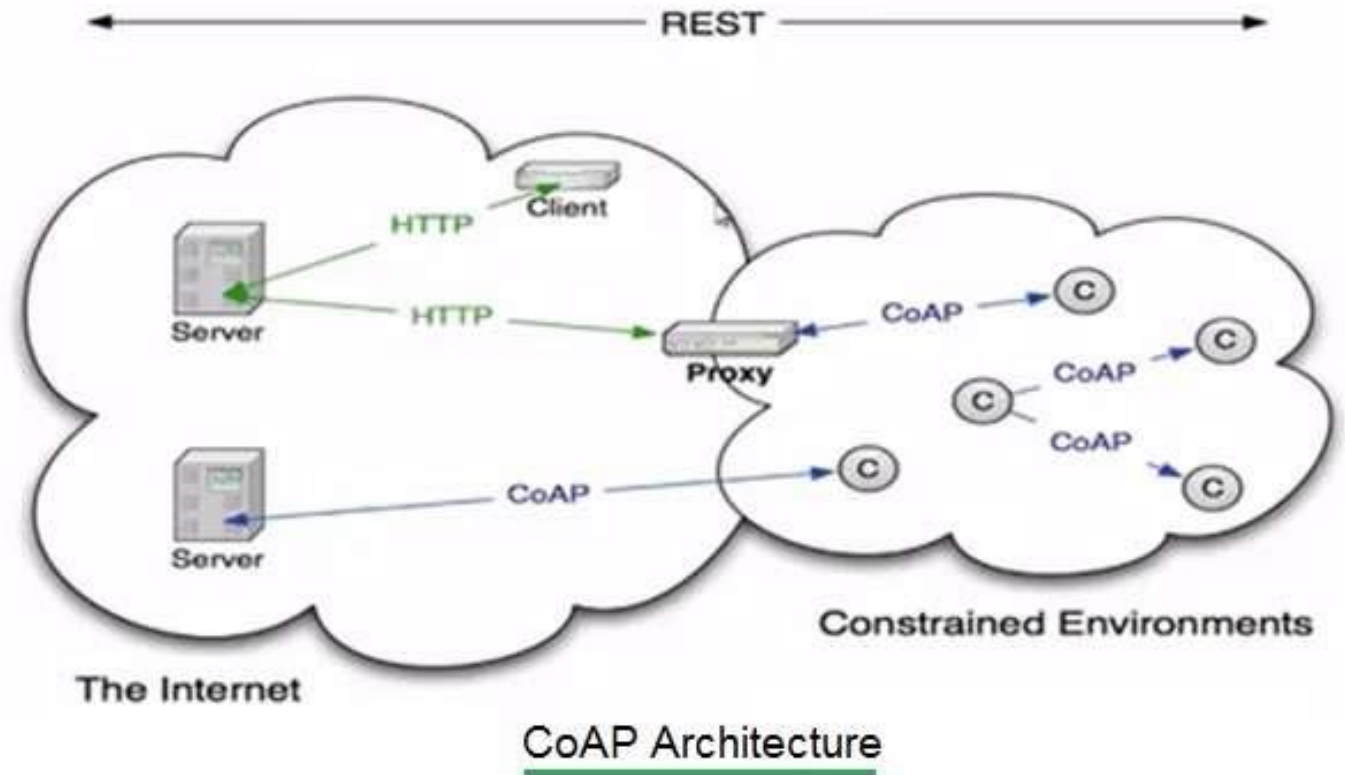
# Dátové (aplikačné) protokoly IoT

- **MQTT**
  - Message Queuing Telemetry Transport
- **CoAP**
  - Constrained Application Protocol
- **XMPP**
  - Extensible Messaging and Presence Protocol
  - správovo-orientovaný, založený na XML, klient-server, server-server
- **AMQP**
  - Advanced Message Queuing Protocol
  - komplexný messaging protokol podporujúci rôzne scenáre použitia
- a veľa ďalších

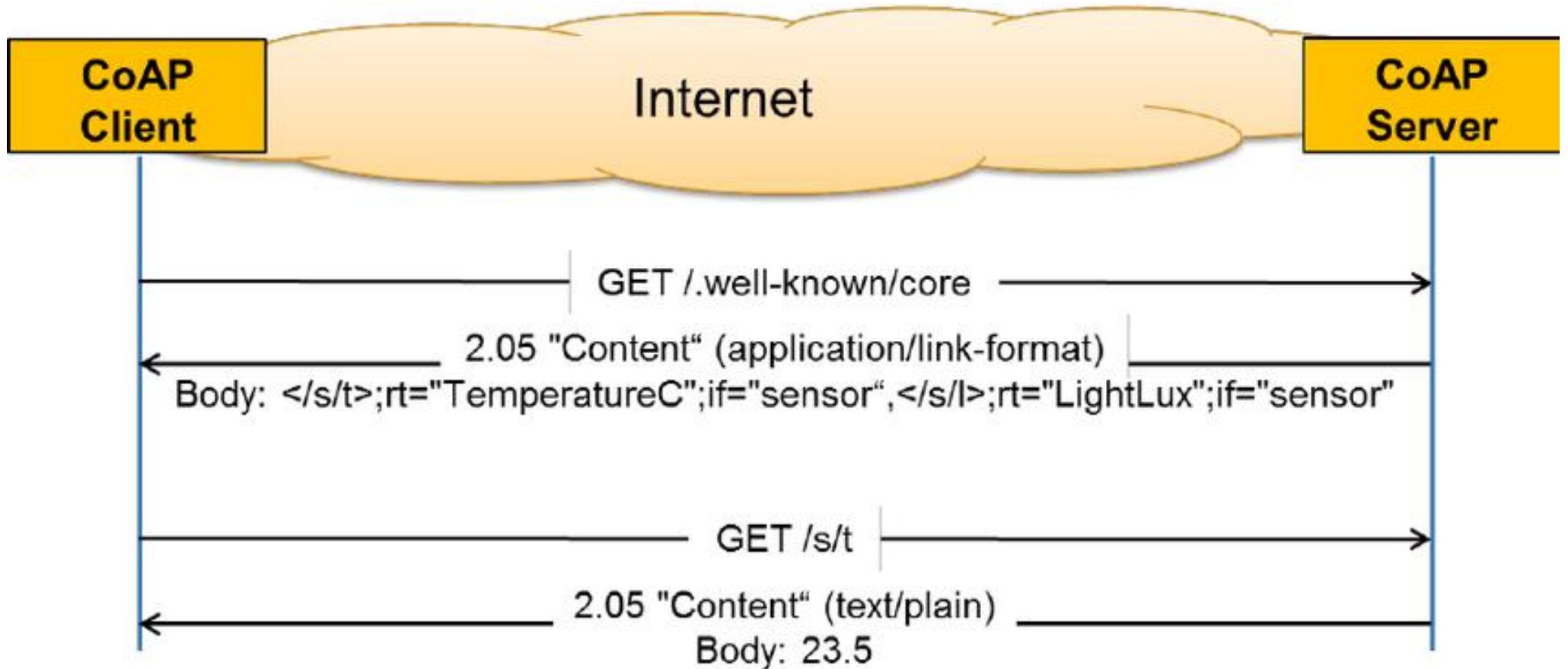


# CoAP

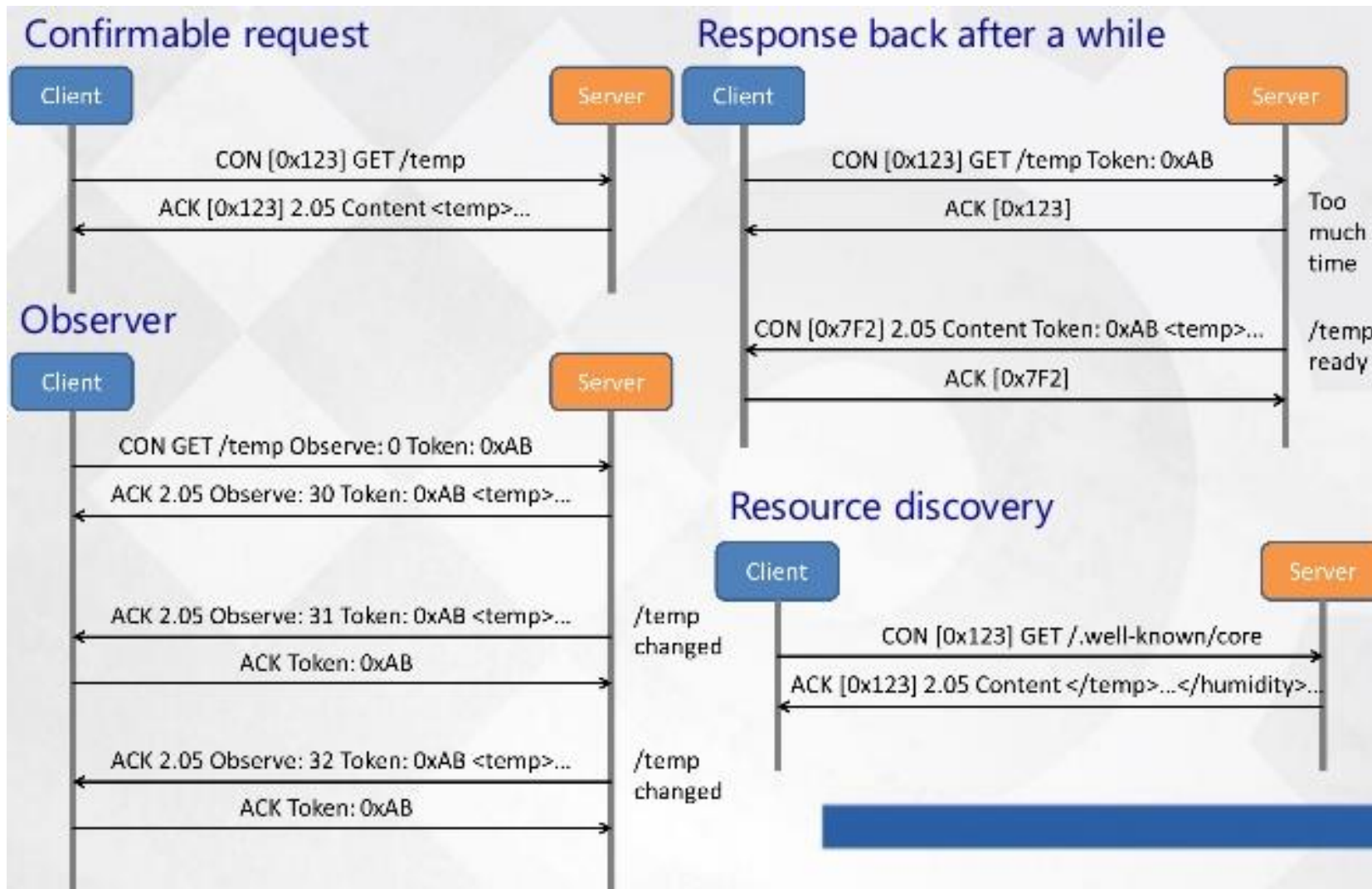
- „**REST**“ protokol pre zariadenia s obmedzenými zdrojmi
- **GET, POST, PUT, DELETE**
- Zjednodušené parsovanie
- 4B hlavička
- **UDP** protokol (NAT issue)
- Podpora **multicastu**
- „Subscription“ model
- Podpora cachovania
- IoT zariadenia vystupujú ako **servery**



# CoAP – komunikácia



# CoAP – komunikácia



# MQTT

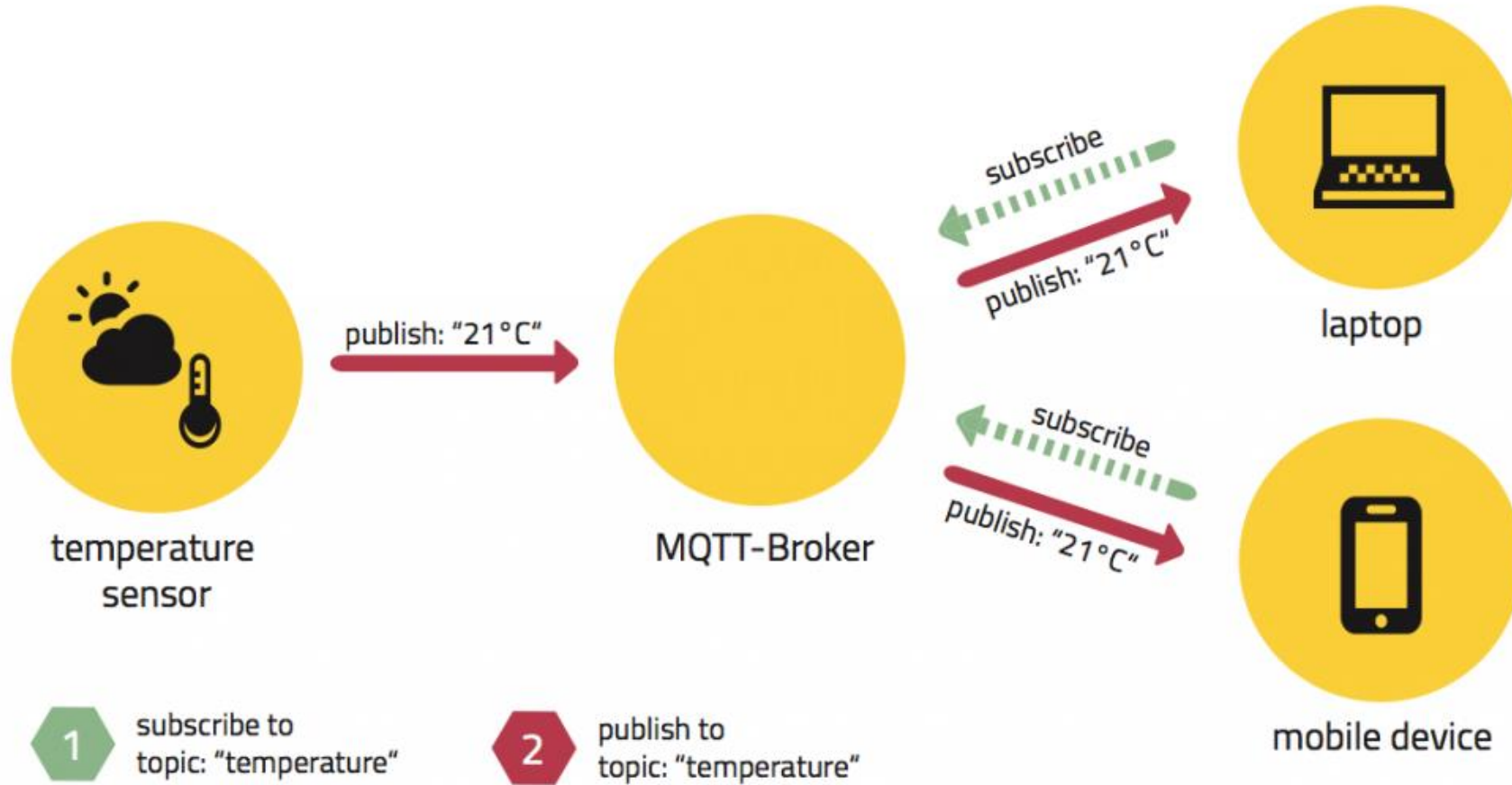
- **Client Server** **publish/subscribe** messaging transport protocol.
- Otvorený a štandardizovaný protokol
  - navrhnutý v IBM (1999)
  - najnovšia verzia 5.0
- Jednoduchý protokol
  - vhodný pre IoT a M2M
  - použiteľný na zariadeniach s obmedzenými zdrojmi
- TCP/SSL
  - „odľahčená“ verzia MQTT-SN (sensor networks) nie je viazaná na TCP stack, no je nevyhnutný gateway
- Početné implementácie klientov a serverov (brokerov)

# HTTP vs. MQTT

Characteristics		3G		WiFi	
		<i>HTTPS</i>	<i>MQTT</i>	<i>HTTPS</i>	<i>MQTT</i>
Receive Messages	<b>Messages / Hour</b>	1,708	160,278	3,628	263,314
	<b>Percent Battery / Hour</b>	18.43%	16.13%	3.45%	4.23%
	<b>Percent Battery / Message</b>	0.01709	0.00010	0.00095	0.00002
	<b>Messages Received (Note the losses)</b>	240 / 1024	1024 / 1024	524 / 1024	1024 / 1024
Send Messages	<b>Messages / Hour</b>	1,926	21,685	5,229	23,184
	<b>Percent Battery / Hour</b>	18.79%	17.80%	5.44%	3.66%
	<b>Percent Battery / Message</b>	0.00975	0.00082	0.00104	0.00016

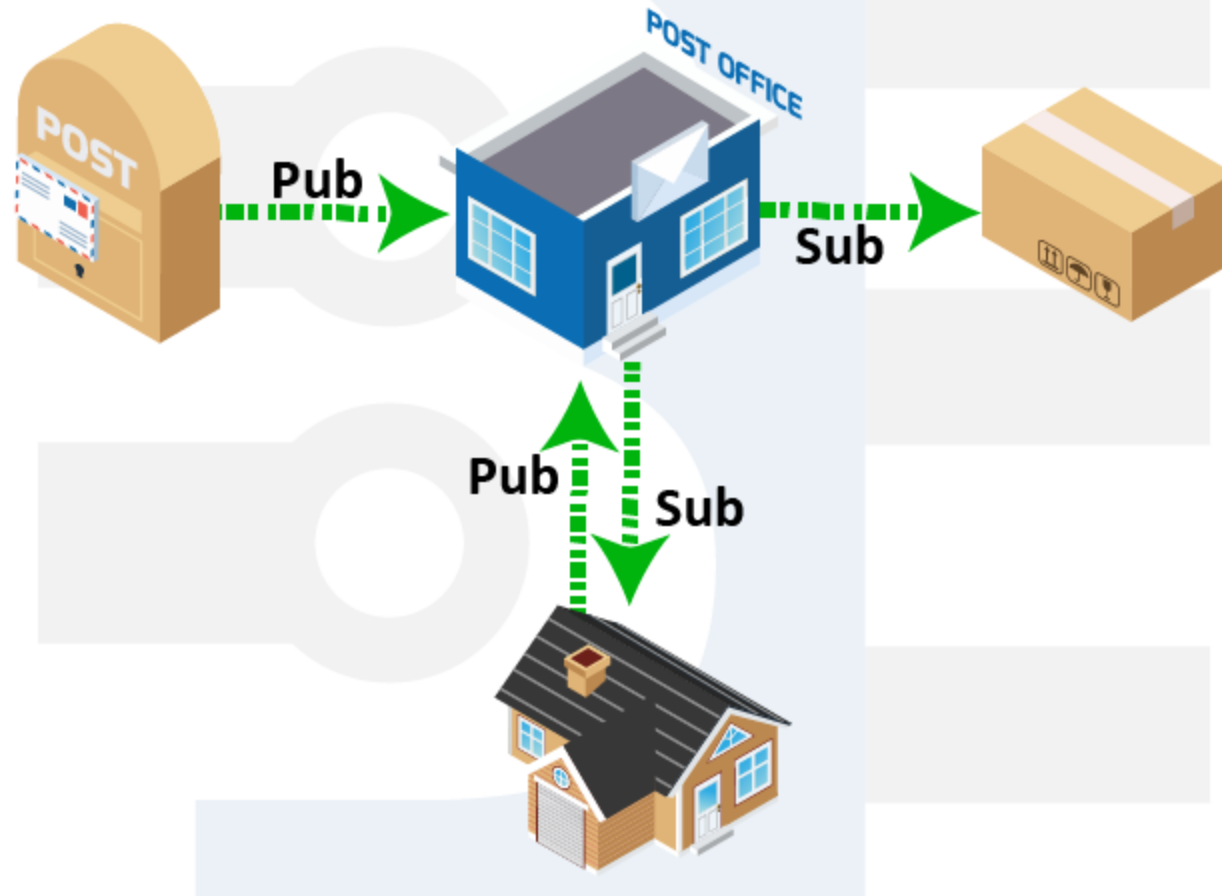
<http://stephendnicholas.com/posts/power-profiling-mqtt-vs-https>

# MQTT



# MQTT - analógia

mqtt: //broker/topic/message



# MQTT brokery

- **Služby:**

- [https://github.com/mqtt/mqtt.github.io/wiki/public\\_brokers](https://github.com/mqtt/mqtt.github.io/wiki/public_brokers)
- `test.mosquitto.org`
- `iot.eclipse.org`

- **Implementácie:**

- Mosquitto (C++)
- Moquette (Java)
- rôzne proprietárne implementované cloudovými službami
- väčšina brokerov podporuje aj pripojenie cez WebSockets





# MQTT klienti

- **Implementácie klientov:**
  - projekt Paho MQTT (C++, Java, JavaScript, Python, ...)
  - M2Mqtt (C#)
  - ...
- **GUI klienti**
  - MQTT explorer
  - MQTT Spy
  - HiveMQ cez websockety:  
<http://www.hivemq.com/demos/websocket-client/>
- Senzory/IoT zariadenia vystupujú ako klienti MQTT brokerov

# MQTT vytvorenie spojenia

MQTT-Packet:

## CONNECT



contains:

<code>clientId</code>	Example
<code>cleanSession</code>	<code>"client-1"</code>
<code>username</code> (optional)	<code>true</code>
<code>password</code> (optional)	<code>"hans"</code>
<code>lastWillTopic</code> (optional)	<code>"letmein"</code>
<code>lastWillQos</code> (optional)	<code>"/hans/will"</code>
<code>lastWillMessage</code> (optional)	<code>2</code>
<code>lastWillRetain</code> (optional)	<code>"unexpected exit"</code>
<code>keepAlive</code>	<code>false</code>
	<code>60</code>



MQTT Client



MQTT Broker

MQTT-Packet:

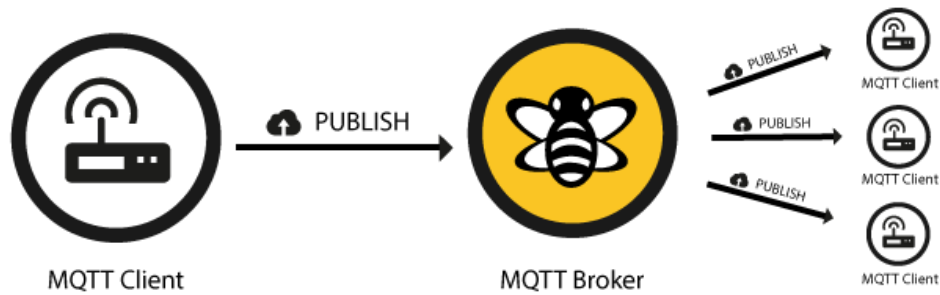
## CONNACK



contains:

<code>sessionPresent</code>	Example
<code>returnCode</code>	<code>true</code>
	<code>0</code>

# MQTT Publish



MQTT-Packet:

## PUBLISH



contains:

`packetId` (always 0 for qos 0)

`topicName`

`qos`

`retainFlag`

`payload`

`dupFlag`

Example

4314

"topic/1"

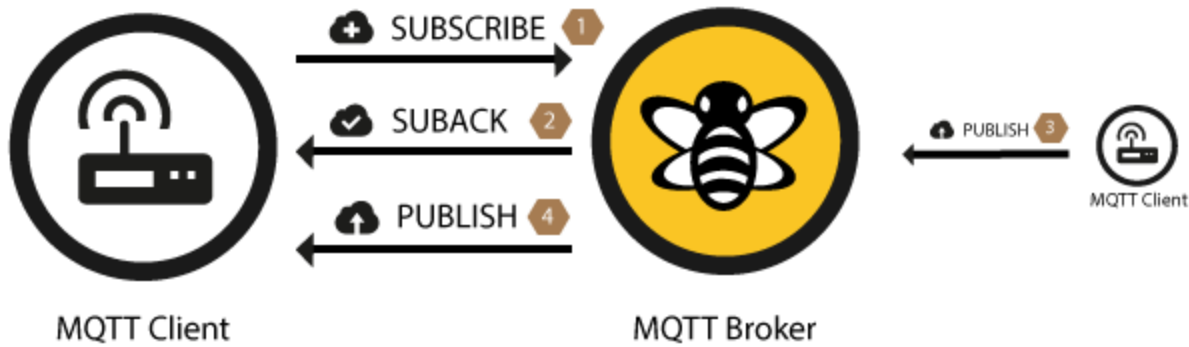
1

false

"temperature:32.5"


false

# MQTT Subscribe



„wildcards“ v topic-och pre subscribe:

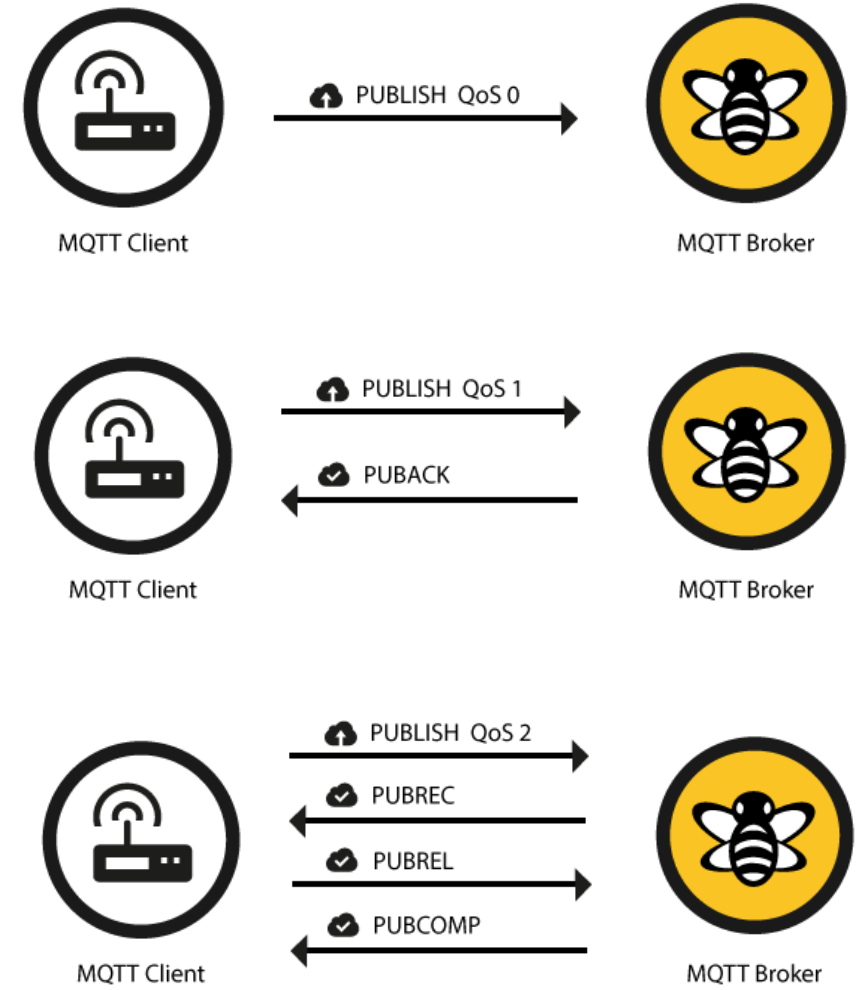
- **+** - zastupuje práve jeden level  
upjs/jesenna5/**+**/teplota
- **#** - zastupuje 0 a viac levelov  
- môže byť len na konci  
upjs/jesenna5/iotlab/**#**

MQTT-Packet: **SUBSCRIBE** 

contains:	Example
packetId	4312
qos1	1
topic1	"topic/1"
qos2	0
topic2	"topic/2"
...	...

# MQTT QoS

- **QoS = Quality of Service**
  - 0: At most once („fire and forget“)
  - 1: At least once
  - 2: Exactly once
- QoS pri Subscribe hovorí s akou QoS sa odosielajú správy klientovi
- Pri perzistentnej session sa uchovávajú pre klienta len správy s  $QoS > 0$



Ďakujem za pozornosť