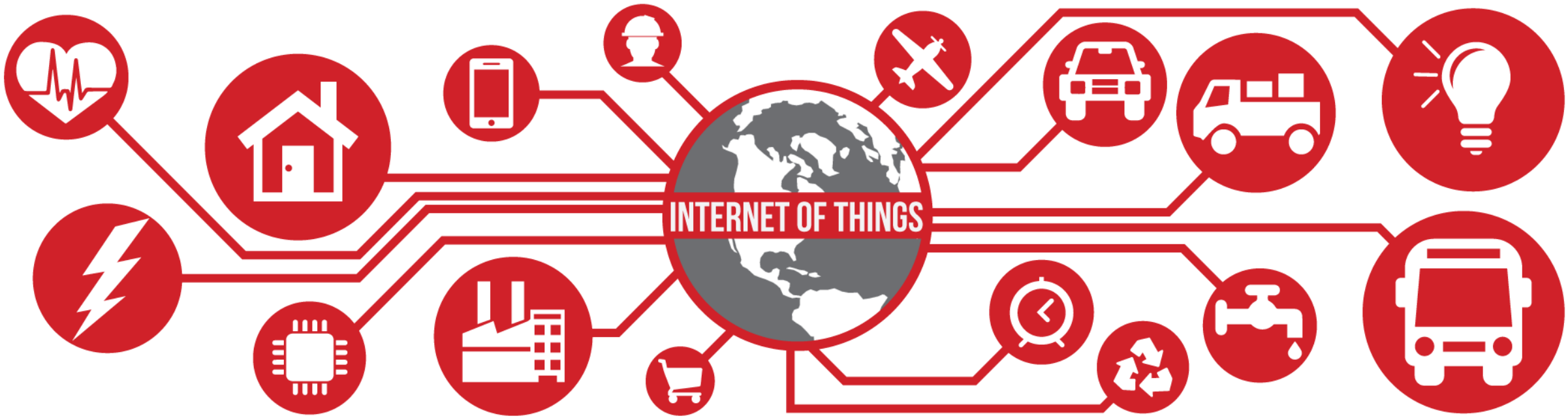

INTERNET OF THINGS

Acerca de mi

- Ingeniería de Sistemas - Universidad EAFIT
- Ingeniero de software en Tecnologías MARTE S.A.S.
- Experiencia en Python, Java, Django, C/C++, microcontroladores y sensores

Acerca de Tecnologías MARTE

- Tecnologías contra minas y artefactos explosivos, enfocada en su desactivación mecánica y electromagnética.
- Protección personal y de vehículos.
- Análisis y protección de señales, sistemas de comunicación.
- Consultoría especializada: asesoría en seguridad para el sector público y privado, asesoría en compras de equipos de seguridad.
- Desarrollos a la medida.

Censado de ambientes complejos

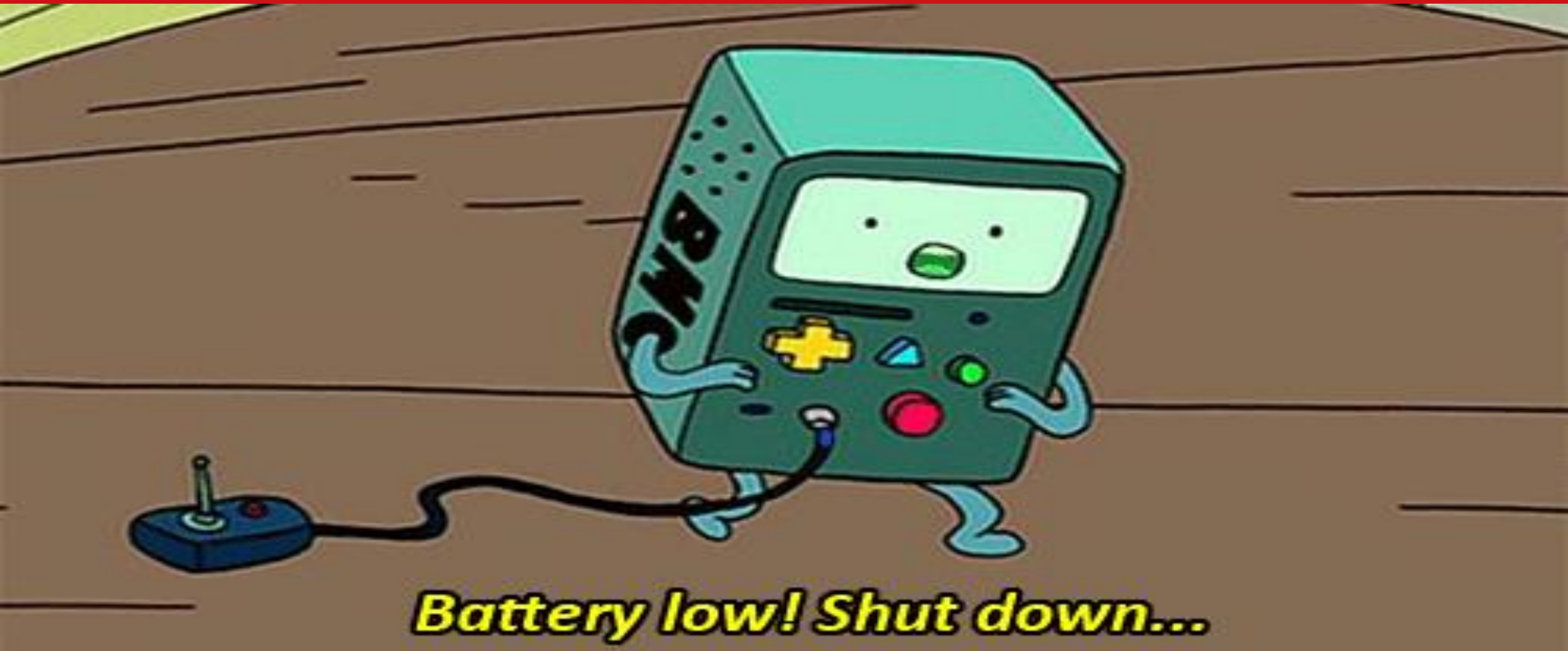
Maneras innovadoras de percibir y enviar información desde el mundo físico hasta la nube

Conectividad

Variedad de estándares de conectividad alambricos e inalambricosque son necesarios para habilitar las diferentes demandas de las aplicaciones

La energía es critica

Muchas aplicaciones de IoT deben correr con baterías por años o en entornos con limitaciones de consumo de energía

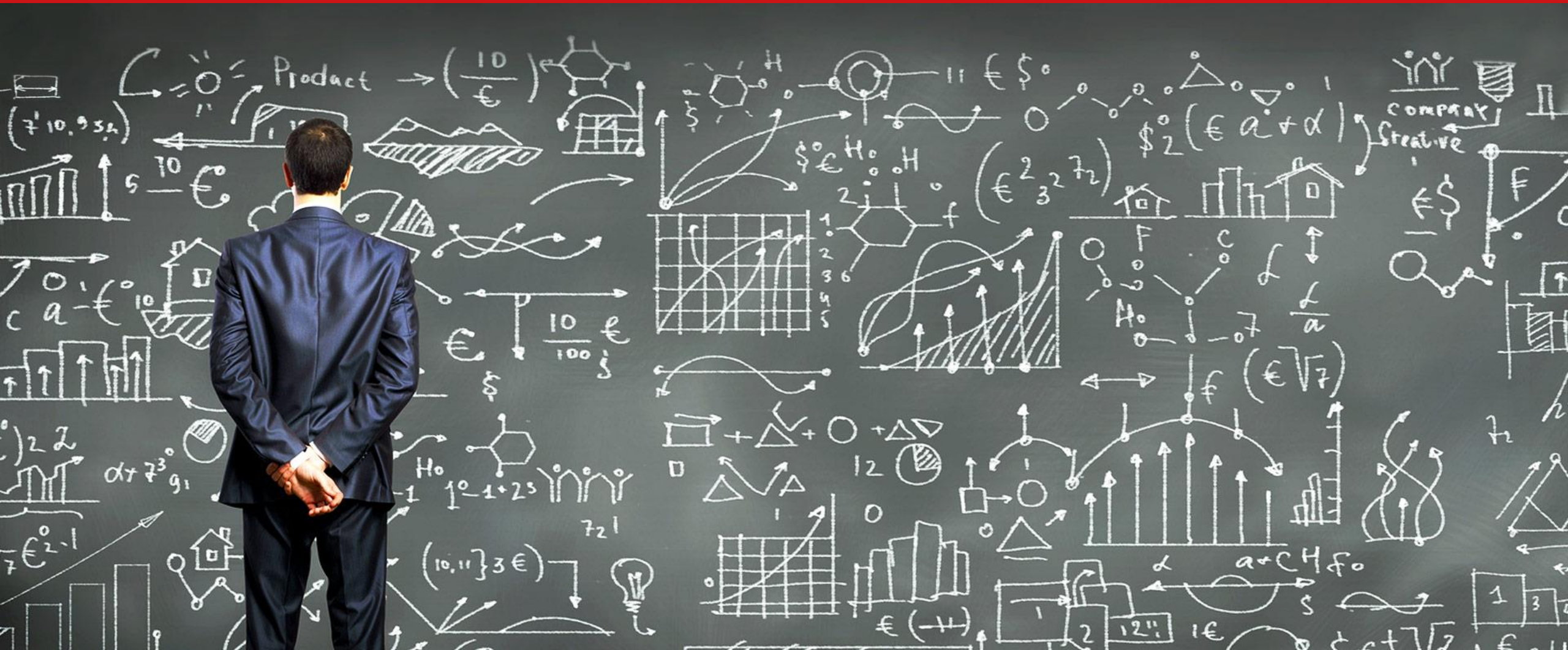


La seguridad es vital

Proteger la privacidad de los usuario detectando y bloqueando actividad maliciosa

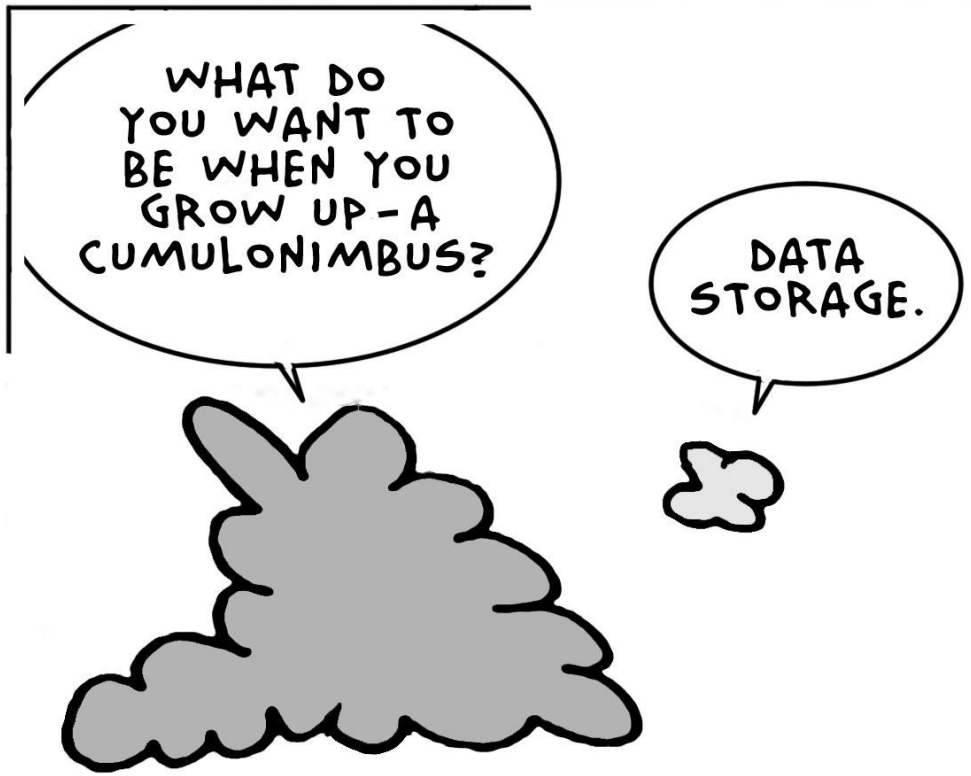
IoT es complejo

El desarrollo de aplicaciones necesita ser fácil para todos los desarrolladores, no solo para los expertos



La nube es importante

Las aplicaciones de IoT requieren soluciones de punto a punto, incluyendo la nube dentro de estos



Soluciones

Veloz, liviano y estándar. MQTT resuelve muchos de nuestros problemas del IoT

Batería

Uso de batería en establecer la conexión inicial

% Battery Used			
3G		Wifi	
HTTPS	MQTT	HTTPS	MQTT
0.02972	0.04563	**0.00228**	0.00276

Batería

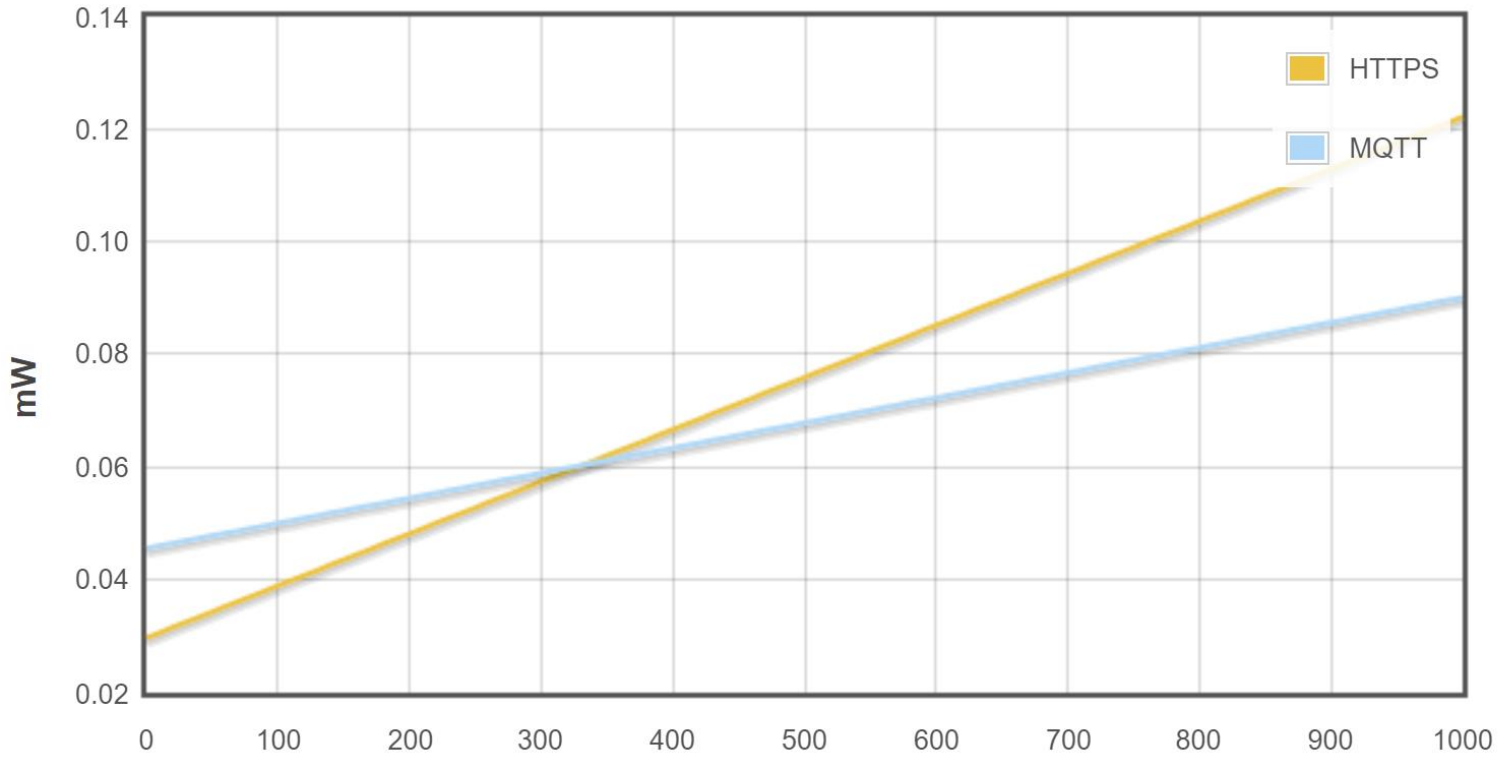
Uso de batería al mantener la conexión

	% Battery / Hour			
	3G		Wifi	
Keep Alive (Seconds)	HTTPS	MQTT	HTTPS	MQTT
60	1.11553	**0.72465**	0.15839	**0.01055**
120	0.48697	**0.32041**	0.08774	**0.00478**
240	0.33277	**0.16027**	0.02897	**0.00230**
480	0.08263	**0.07991**	0.00824	**0.00112**

Batería

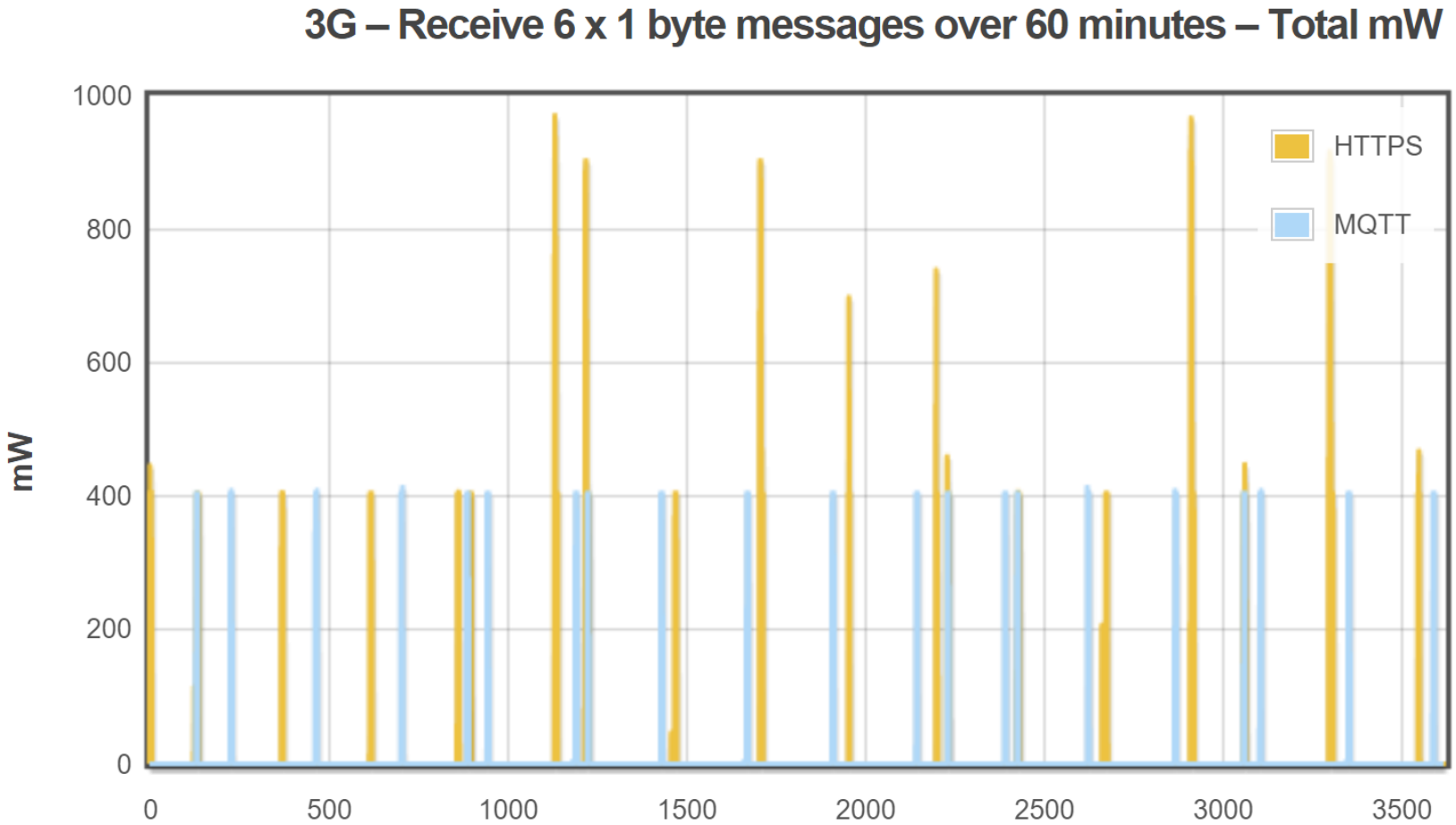
Uso de batería al mantener la conexión

3G – 240s Keep Alive – % Battery Used Creating and Maintaining a Connection



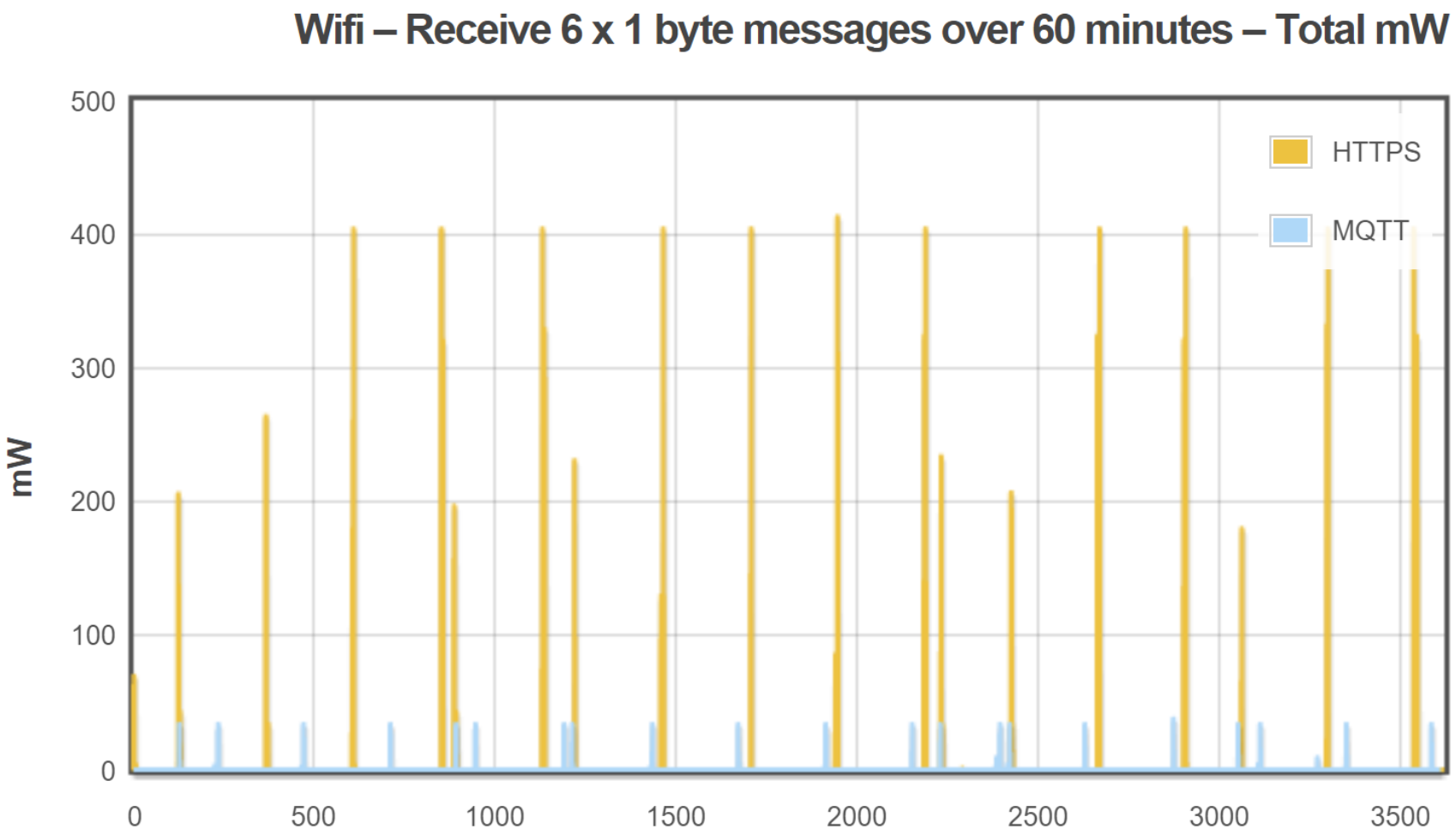
Batería

Uso por 60 minutos, HTTP Vs. MQTT



Batería

Uso por 60 minutos, HTTP Vs. MQTT



Batería

Uso por 60 minutos, HTTP Vs. MQTT

Mensajes

Recibiendo tan rápido como nos sea posible

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.43%	**16.13%**	**3.45%**	4.23%
Messages / Hour	1708	**160278**	3628	**263314**
% Battery / Message *	0.01709	**0.00010**	0.00095	**0.00002**
Messages Received	240 / 1024	**1024 / 1024**	524 / 1024	**1024 / 1024**

Mensajes

Mandar tan rápido como nos sea posible

	3G		Wifi	
	HTTPS	MQTT	HTTPS	MQTT
% Battery / Hour	18.79%	**17.80%**	5.44%	**3.66%**
Messages / Hour	1926	**21685**	5229	**23184**
% Battery / Message *	0.00975	**0.00082**	0.00104	**0.00016**

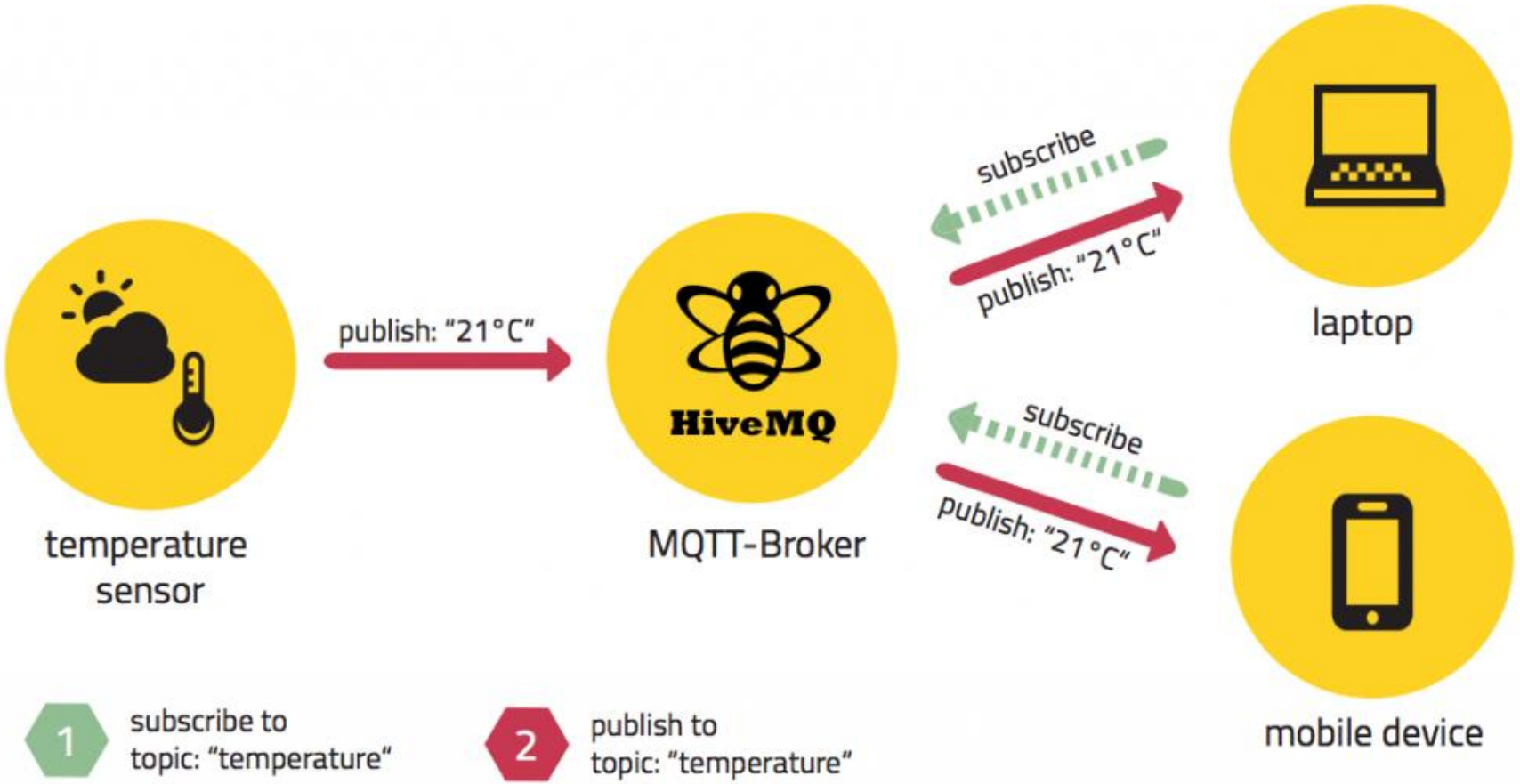
Mensajes

Conclusión

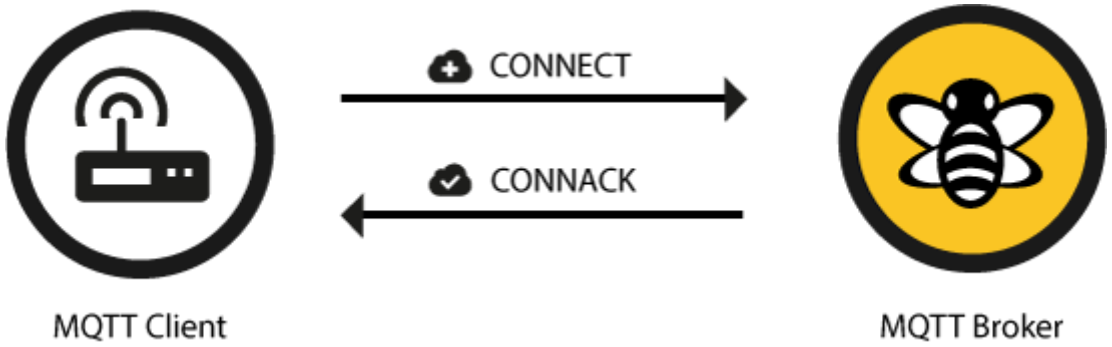
MQTT
Fundamentos

- Simple de implementar
- Proporciona calidad de envio y entrega de datos
- Liviano y eficiente en ancho de banda
- Agnóstico de tipo de información
- Consciencia continua de las cesiones

MQTT
Arquitectura



MQTT
Conexión



MQTT-Packet:

CONNECT

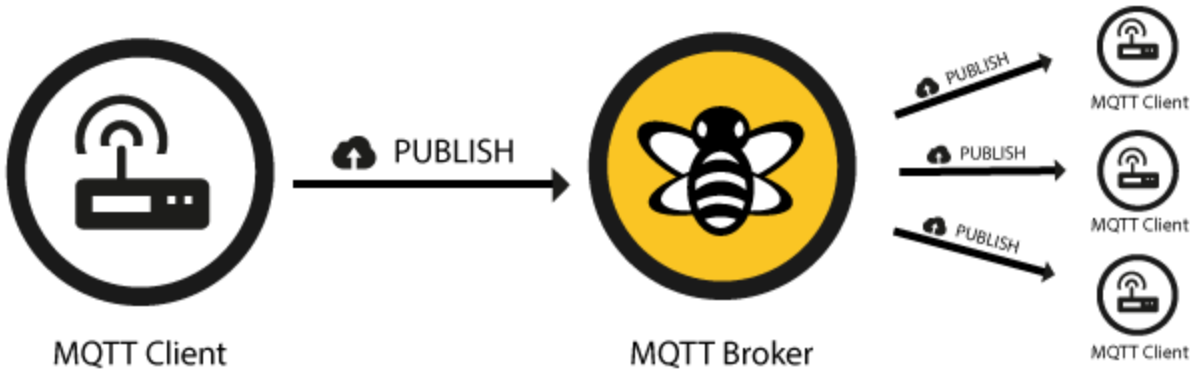
contains:	Example
`clientId`	`"client-1"`
`cleanSession`	`true`
`username` (optional)	`"hans"`
`password` (optional)	`"letmein"`
`lastWillTopic` (optional)	`"/hans/will"`
`lastWillQos` (optional)	`2`
`lastWillMessage` (optional)	`"unexpected exit"`
`keepAlive`	`60`

MQTT-Packet:

CONNACK

contains:	Example
`sessionPresent`	`true`
`returnCode`	`0`

MQTT
Publicación

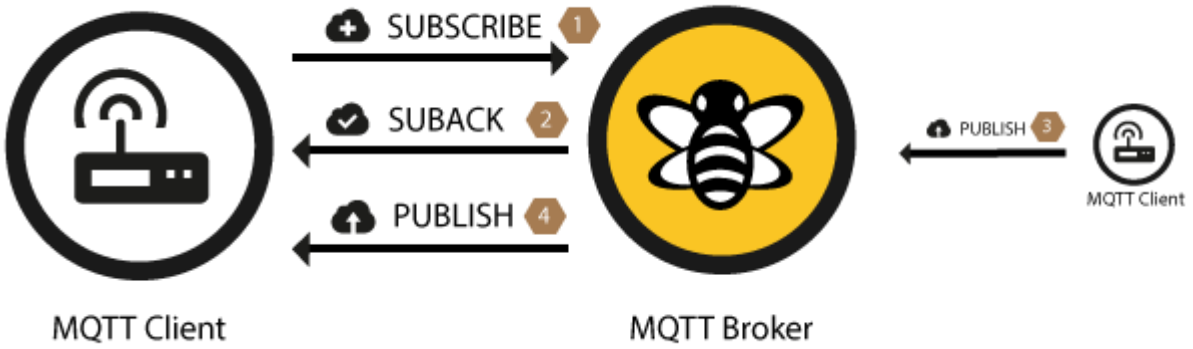


MQTT-Packet:

SUBSCRIBE

contains:		Example
packetId		4312
qos1	} (list of topic + qos)	1
topic1		"topic/1"
qos2	}	0
topic2		"topic/1"
...		...

MQTT
Suscripción



MQTT-Packet:

SUBSCRIBE

contains:		Example
packetId		4312
qos1	} (list of topic + qos)	1
topic1	} (list of topic + qos)	"topic/1"
qos2	}	0
topic2	}	"topic/1"
...		...

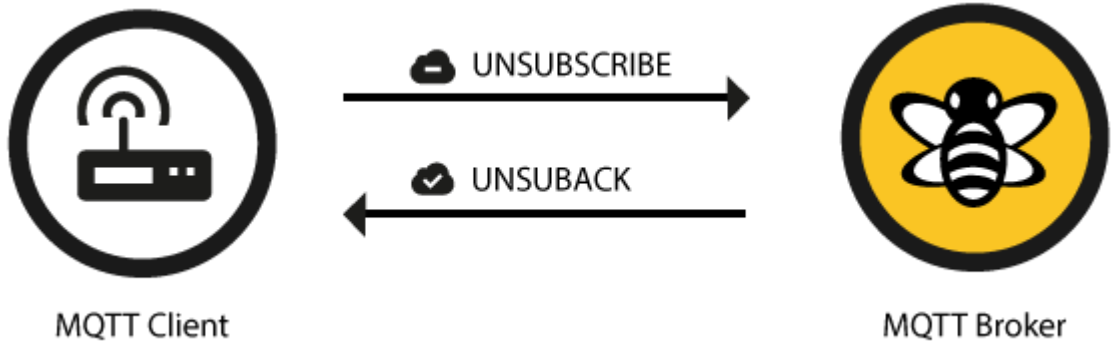
MQTT-Packet:

SUBACK

contains:		Example
packetId		4313
returnCode 1	(one returnCode for each topic from SUBSCRIBE, in the same order)	2
returnCode 2		0
...		...

MQTT

Deshacer suscripción



MQTT-Packet:

UNSUBSCRIBE

contains:

```
packetId
topic1
topic2
...
```

(list of topics)

Example

```
4315
"topic/1"
"topic/2"
...
```

MQTT-Packet:

UNSUBACK

contains:

```
packetId
```

Example

```
4316
```

Tópicos

Estructura de un tópico

- No utilizar un slash al principio
- No utilizar espacios en un tópico
- Los tópicos deben ser cortos y concisos
- Utilizar solo caracteres ASCII, evitar los caracteres difíciles de imprimir

Tópicos

Los comodines

single-level wildcard

myhome / groundfloor / + / temperature

only one level

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature
- ✗ myhome / groundfloor / kitchen / fridge / temperature

multi-level wildcard

myhome / groundfloor / #

only at the end

multiple topic levels

- ✓ myhome / groundfloor / livingroom / temperature
- ✓ myhome / groundfloor / kitchen / temperature
- ✓ myhome / groundfloor / kitchen / brightness
- ✗ myhome / firstfloor / kitchen / temperature

NO suscribirse a

Tópicos

¡Simplemente NO!

Tecnologías para la codificación en vivo

Si algo puede salir mal cuando haga código lo hará

Mosquitto

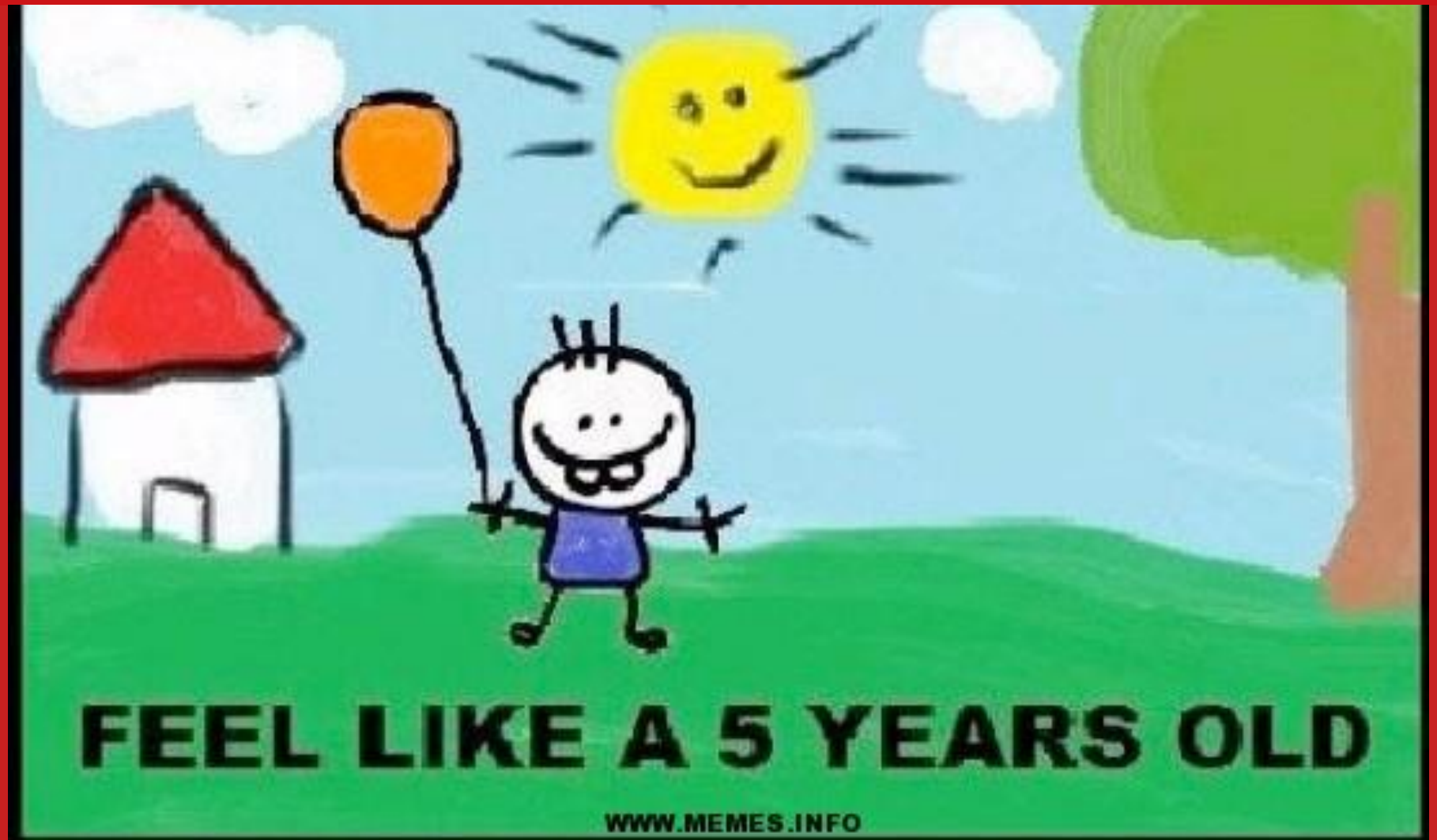
An Open Source MQTT v3.1 Broker

paho

amazon web services™

Tecnologías para la codificación en vivo

Si algo puede salir mal cuando haga código lo hará



Protocol Buffers

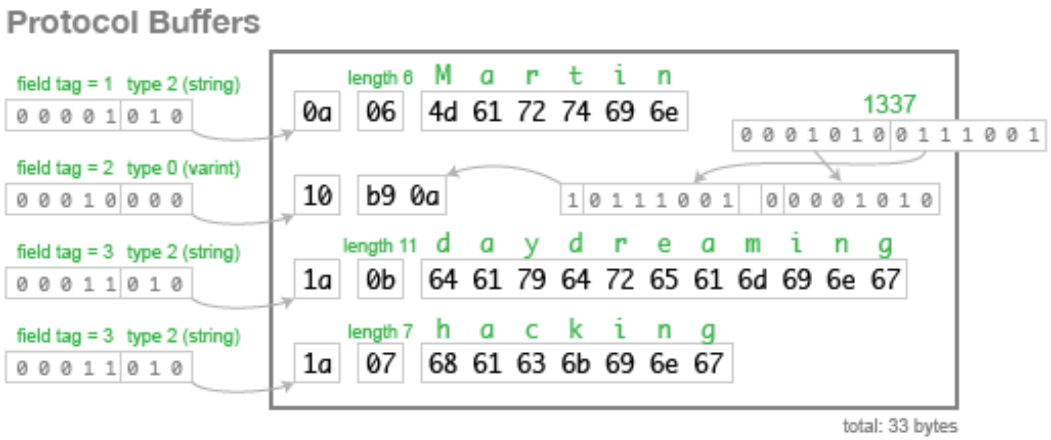
¿Por qué protocol buffers?

- Utiliza esquemas
- Tiene compatibilidad con versiones anteriores
- Menos código repetitivo
- Validación y extensibilidad
- Fácil de migrar de lenguaje a lenguaje

Protocol Buffers

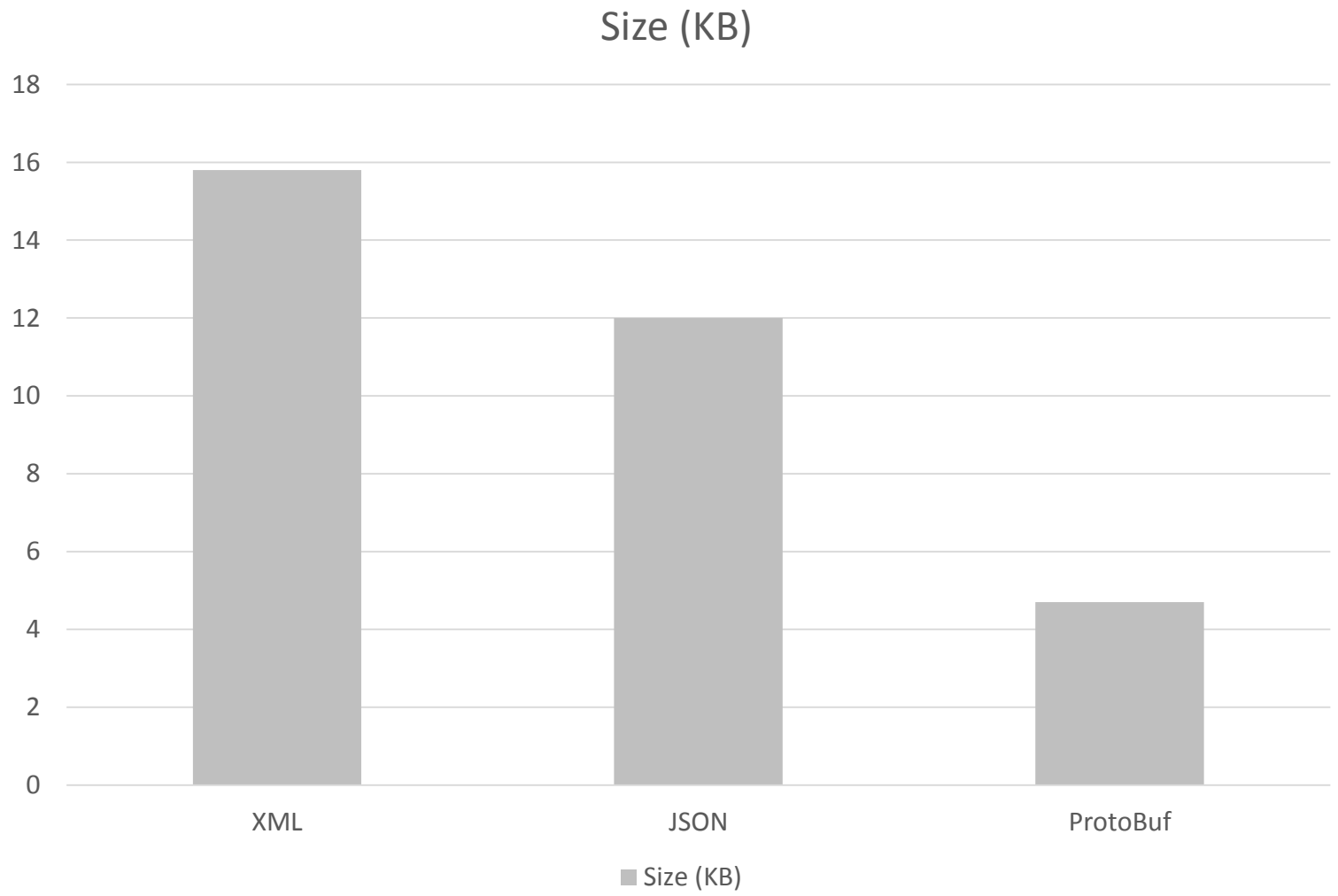
¿Cómo funciona?

Instancia	Descripción
`{ "userName": "Martin", "favouriteNumber": 1337, "interests": ["daydreaming", "hacking"] }`	`message Person { required string user_name = 1; optional int64 favourite_number = 2; repeated string interests = 3; }`



Protocol Buffers

¿Y los números qué?



Protocol Buffers

¿Y los números qué?