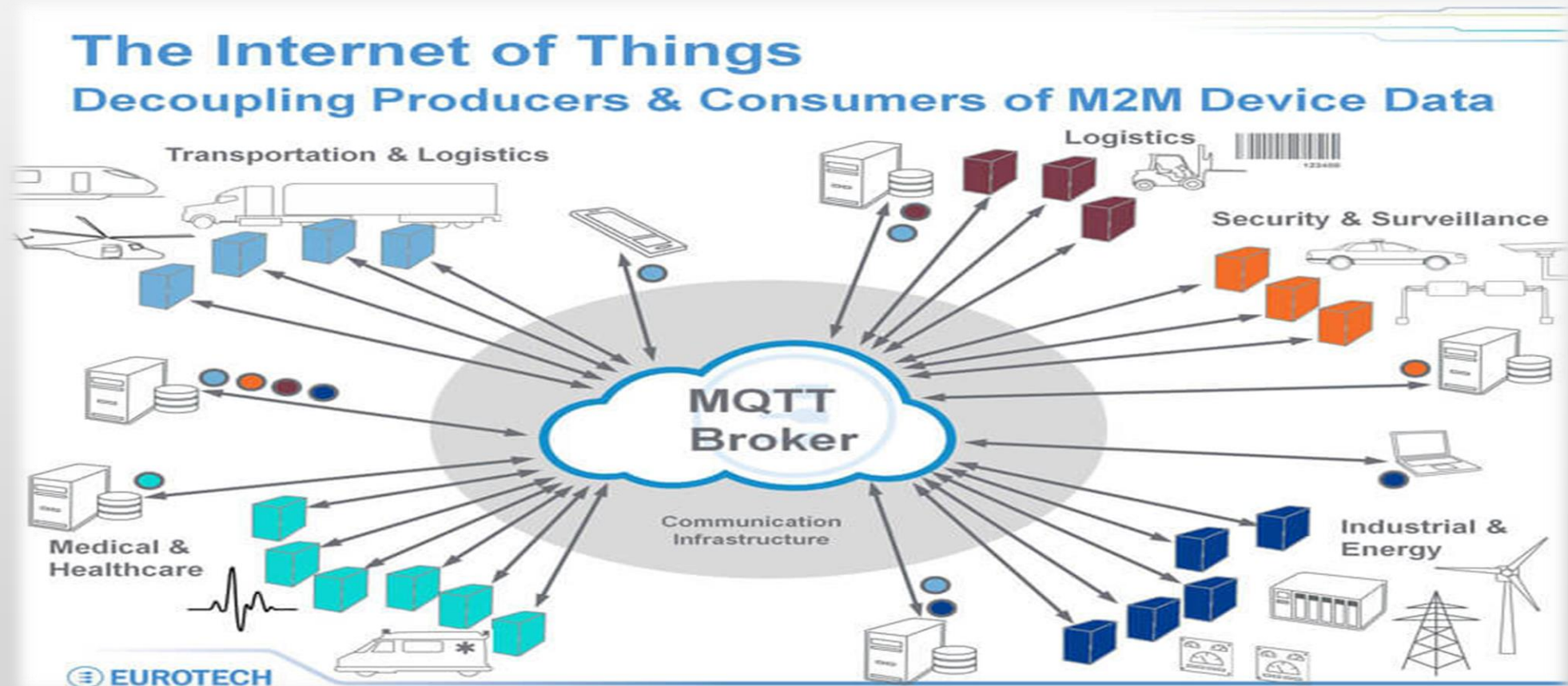


# MQTT y Arduino

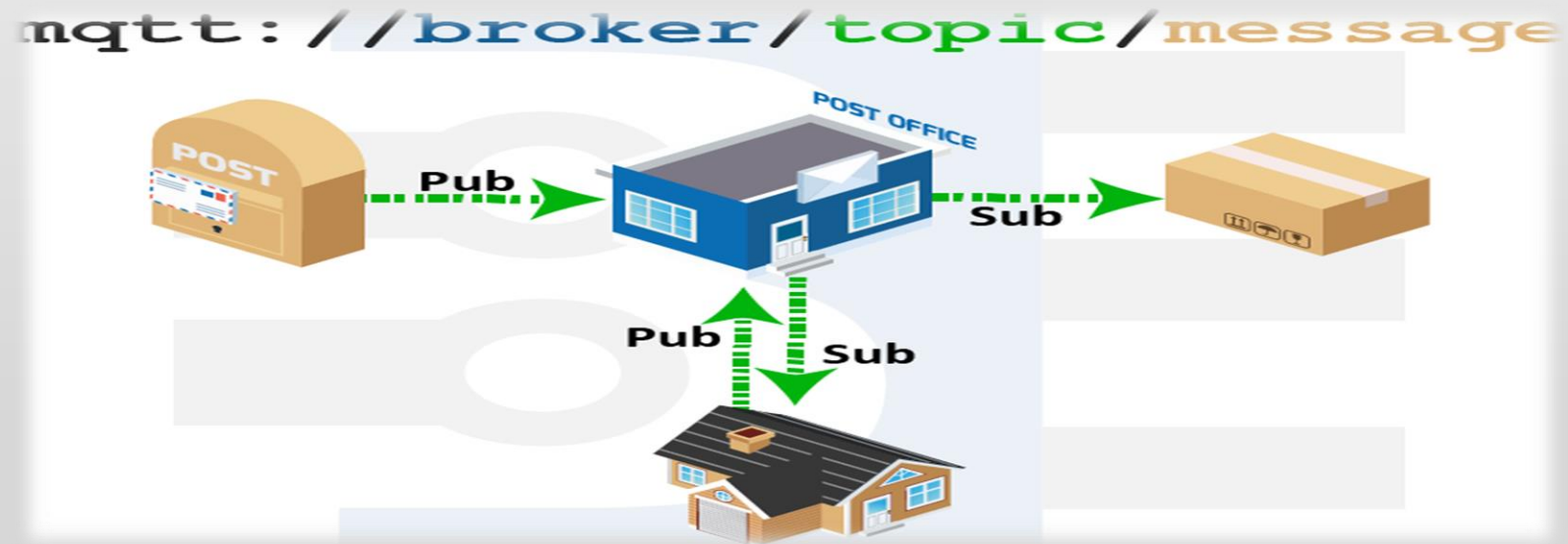
- ✓ MQTT son las siglas de Message Queue Telemetry Transport es un protocolo ideado por IBM enfocado a la conectividad Machine-to-Machine (M2M).
- ✓ Está enfocado al envío de datos en aplicaciones donde se requiere muy poco ancho de banda.> Consumo realmente bajo. Pocos recursos
- ✓ Protocolo muy empleado en la comunicación de sensores y, consecuentemente, dentro del Internet de las Cosas.
- ✓ Es precisamente el broker el elemento encargado de gestionar la red y transmitir los mensajes.

# MQTT y Arduino



## MQTT y Arduino

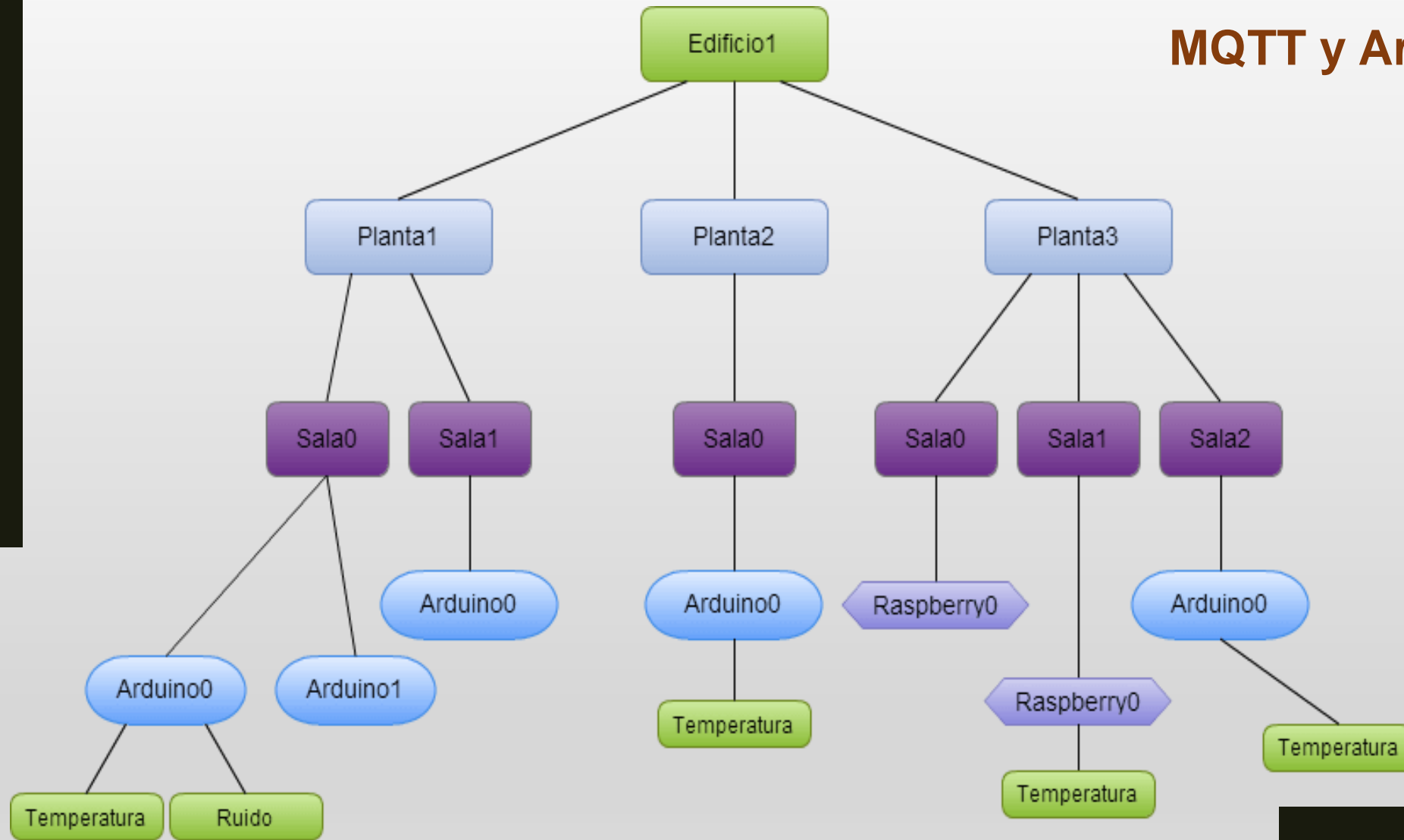
- ✓ Dentro de la arquitectura de MQTT, es muy importante el concepto “topic” o “tema” en español ya que a través de estos “topics” se articula la comunicación puesto que emisores y receptores deben estar suscritos a un “topic” común para poder entablar la comunicación. Este concepto es prácticamente el mismo que se emplea en colas.



## MQTT y Arduino

- Este tipo de arquitecturas, lleva asociada otra interesante característica: la comunicación puede ser de uno a uno o de uno a muchos.
- Un “topic” se representa mediante una cadena y tiene una estructura jerárquica. Cada jerarquía se separa con ‘/’. Por ejemplo, **“edificio1/planta5/sala1/raspberry2/temperatura”** o **“/edificio3/planta0/sala3/arduino4/ruido”**. De esta forma se pueden crear jerarquías de clientes que publican y reciben datos, como podemos ver en la imagen:

## MQTT y Arduino



## MQTT y Arduino

- ✓ De esta forma un nodo puede subscribirse a un “topic” concreto (“edificio1/planta2/sala0/arduino0/temperatura”) o a varios (“edificio1/planta2/#”).
- ✓ MQTT me permite gran escalabilidad. Añadir un nuevo Arduino o un suscriptor es muy sencillo dentro de la jerarquía vista
- ✓ Un cliente MQTT sencillo (y con limitaciones) para instalar en nuestro PC es MQTT Lens. Uso de MQTT lens: <http://www.hivemq.com/blog/mqtt-toolbox-mqtt-lens>
- ✓ Como habrás imaginado, para conseguir una comunicación MQTT, emplearemos una librería. Existen muchas disponibles gracias a la gran (tanto en tamaño como en calidad) comunidad que existe alrededor de Arduino.

## MQTT y Arduino

Una de las librerías más conocidas y la más estable y flexible es Arduino Client for MQTT <http://pubsubclient.knolleary.net/> que nos provee de un sencillo cliente que nos permite tanto subscribirnos como publicar contenido usando MQTT. Internamente, usa la API de Arduino Ethernet Client lo que lo hace compatible con un gran número de shields y placas

Otras Librerías MQTT para Arduino:

- <https://github.com/i-n-g-o/esp-mqtt-arduino>
- <http://pubsubclient.knolleary.net/>
- [https://github.com/adafruit/Adafruit\\_MQTT\\_Library](https://github.com/adafruit/Adafruit_MQTT_Library)

## Ejemplo práctico

Pasos previos:

- Darse de alta en la plataforma: test.mosquitto.org y suscribirse con un cliente de MQTT como MQTT Dashboard para móvil o alguno de los muchos para escritorio.
- No es necesario crear feeds puesto que se crean automáticamente al generarlos desde el programa de Arduino al suscribirse
- Instalar la librería MQTT: <https://pubsubclient.knolleary.net/>
- Carga en Arduino el sketch que se encuentra en el campus virtual
- Modificarlo para acceder con una dirección IP estática y cambiar el Topic a “/master”.
- Una vez hecho suscribirse al feed correspondiente para realizar las lecturas.



# MQTT y Arduino

## Ejemplo práctico

- Ahora vamos a publicar en Thingspeak usando Arduino.
- Para ello seguimos los pasos en:  
<https://www.mathworks.com/help/thingspeak/use-arduino-client-to-publish-to-a-channel.html>
- Vamos a crear un fichero con los datos que tenemos publicando en ThingSpeak. Se trata de un fichero de Excel con los nombres de las variables y los valores publicados en todos los channels de ThingSpeak.

## Ejemplo práctico

- Para ello emplearemos el formato:
- | Date       | Hour  | Temperatura del sensor del Arduino |
|------------|-------|------------------------------------|
| 16/04/2018 | 03:45 | 34                                 |
- Este fichero lo generaremos de la siguiente manera:
  - a) Se escriben los datos en ThingsPeak empleando MQTT desde el Arduino
  - b) Se leen mediante Matlab desde ThingsPeak
  - c) Se va guardando cada lectura en el fichero usando 'csvwrite' (se adjunta información de las órdenes de manejo de ficheros en Matlab).