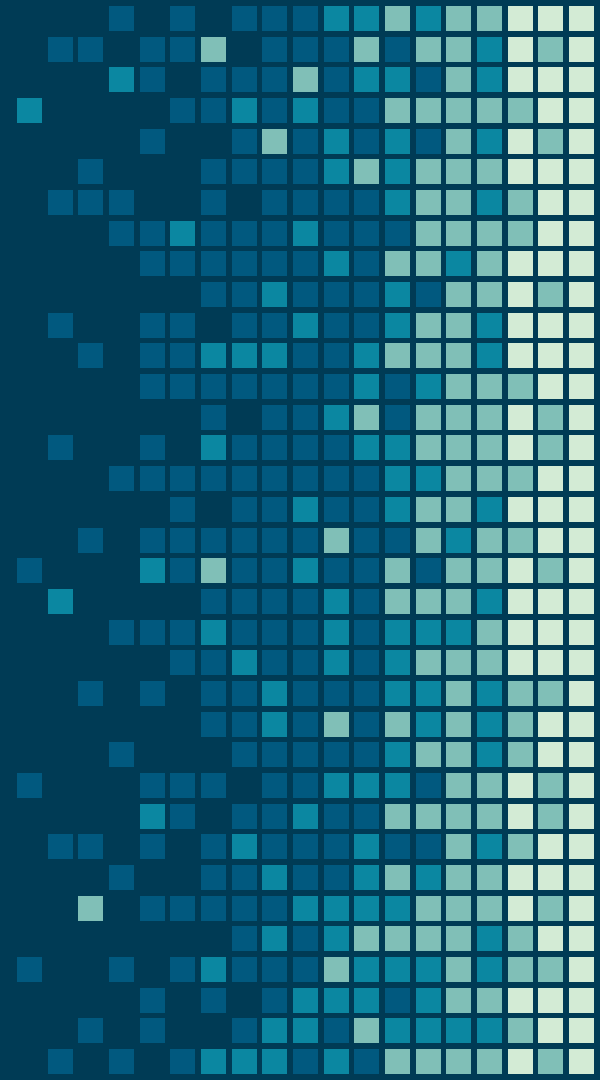


# MQTT

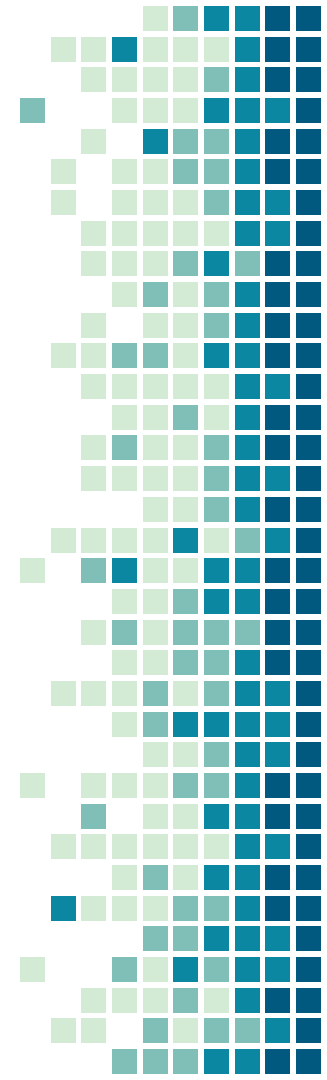
## Message Queuing Telemetry Transport

Aluno: Pedro Tavares  
Professor: Adenauer Yamin

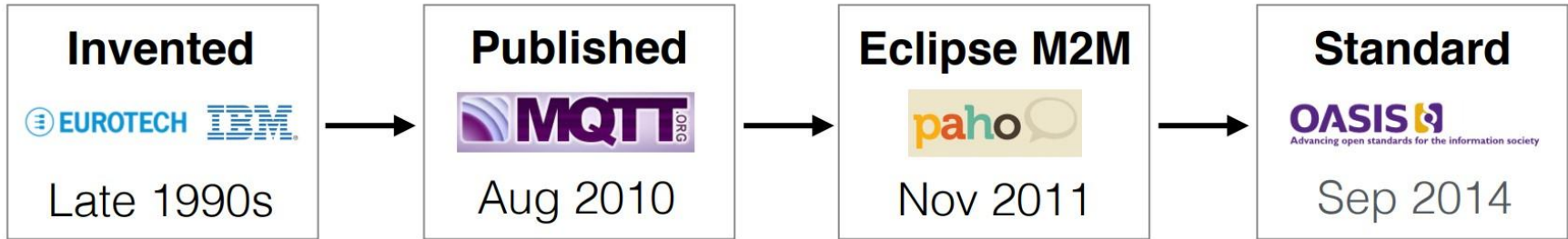


# Introdução

- O MQTT foi criado pela IBM em conjunto com a Eurotech em 1999, para um case onde deveriam se conectar ao sistema de telemetria de oleodutos através de uma conexão via satélite totalmente precária e insegura.
- Isso os motivou a criar um protocolo assíncrono eficiente, com baixo consumo, com formato de dados livre e com suporte a múltiplos níveis de QoS.
- Esse conjunto de características tornam o MQTT um dos protocolos mais utilizados em sistemas M2M e de IoT.

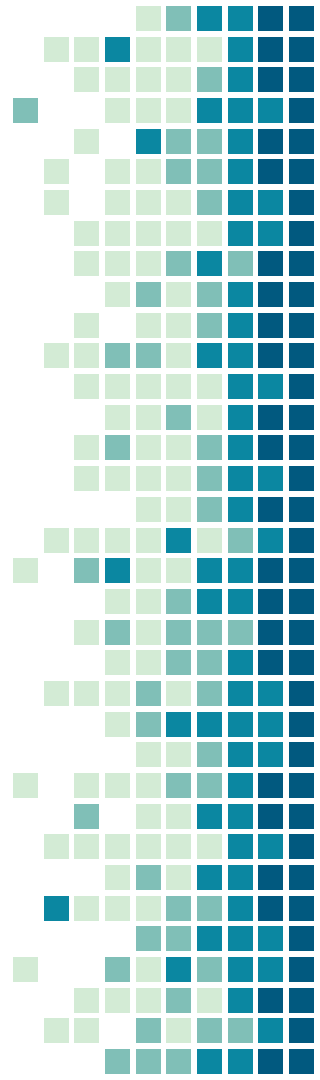


# Histórico



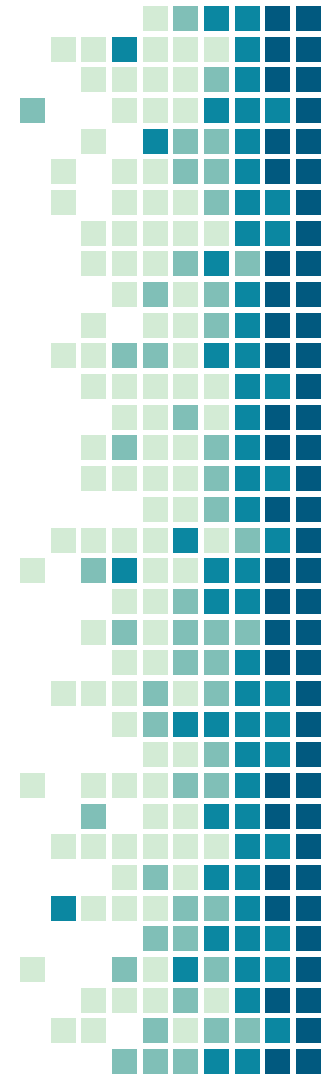
# Características

- Simples implementação e utilização (spec. um pouco mais de 60 páginas)
- Código aberto: +40 standard client implementations
- Consume pouca banda (roda sobre camada TCP, mas tem baixo payload comparado com HTTP)
- Formato de dados livre



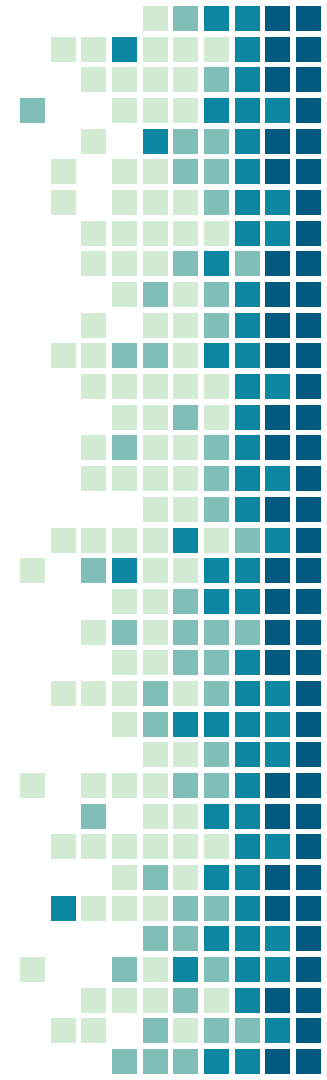
# Características

- Fornece diferentes níveis de QoS para aquisição dos dados: agrega confiabilidade em redes não-confiáveis
- Continuous session awareness
- Extremamente leve: clientes poucos kb
- Comunicação assíncrona bidirecional

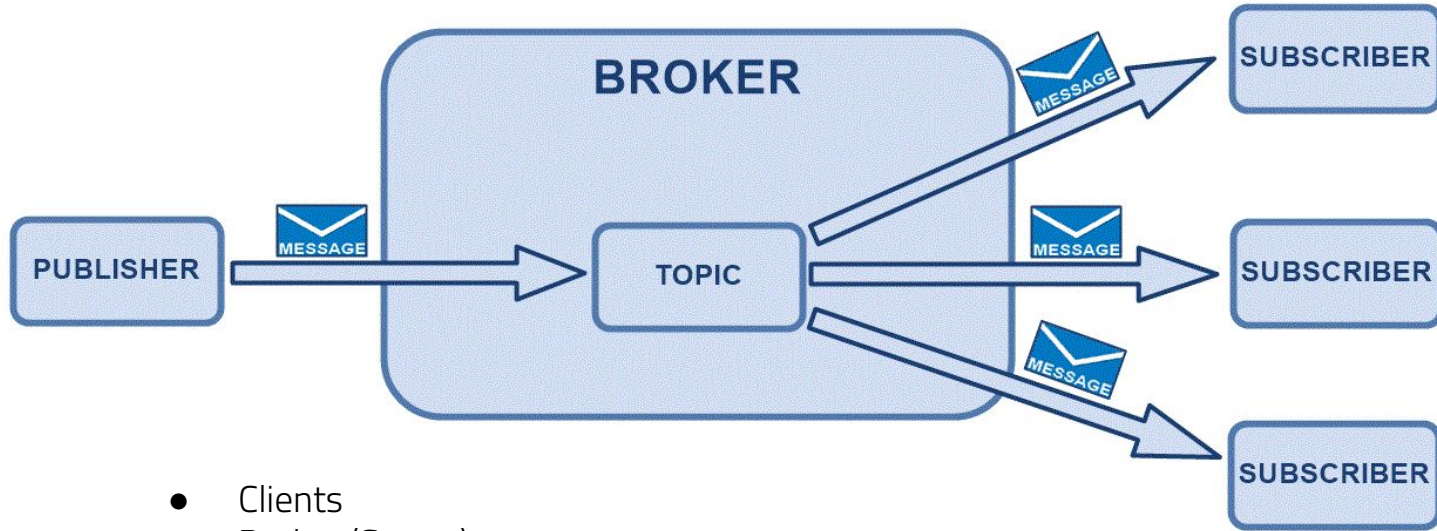


# Características

- Utilização de um broker desassocia os clientes entre si
- Username/Pwd para estabelecimento de conexão;
- Brokers fornecem suporte para SSL/TSL: mensagens criptografadas;



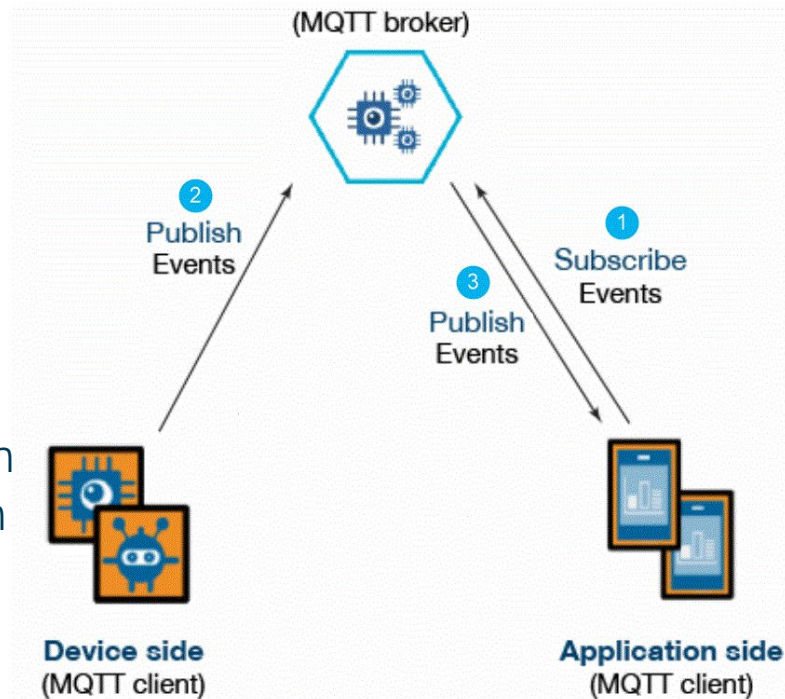
# Publish/Subscribe



- Clients
- Broker (Server)
- Message
- Topic

# Publish/Subscribe

1. Clientes se conectam a um broker e **assinam** os tópicos que estão interessados;
2. Clientes se conectam ao broker e **publicam** mensagens nos tópicos;
3. Broker distribui (publica) a mensagem para todos os clientes que assinaram o tópico;

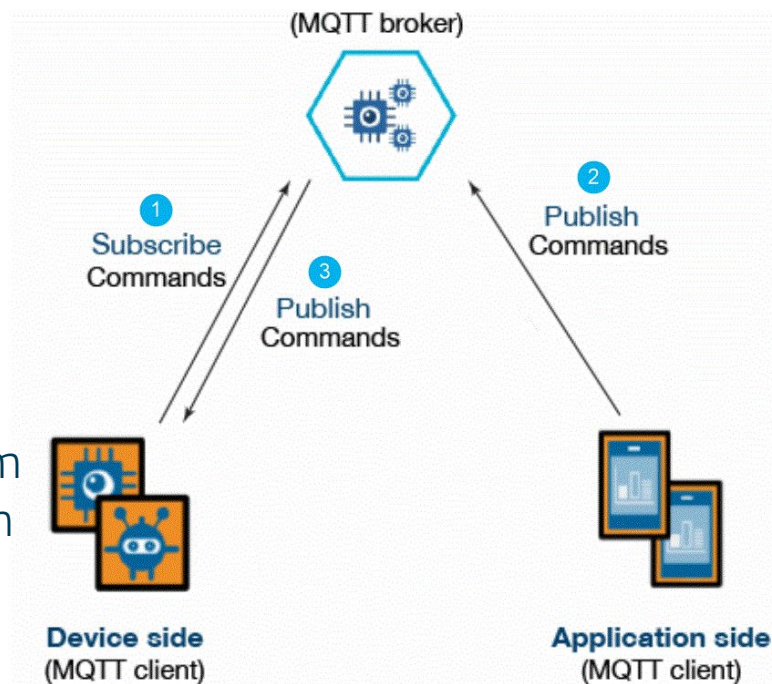


Ex: Leitura de sensor



# Publish/Subscribe

1. Clientes se conectam a um broker e **assinam** os tópicos que estão interessados;
2. Clientes se conectam ao broker e **publicam** mensagens nos tópicos;
3. Broker distribui (publica) a mensagem para todos os clientes que assinaram o tópico;



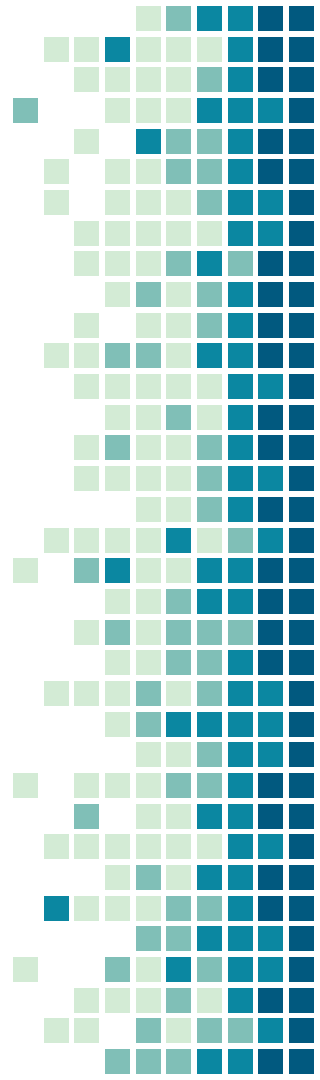
**Ex: Configurando um dispositivo**

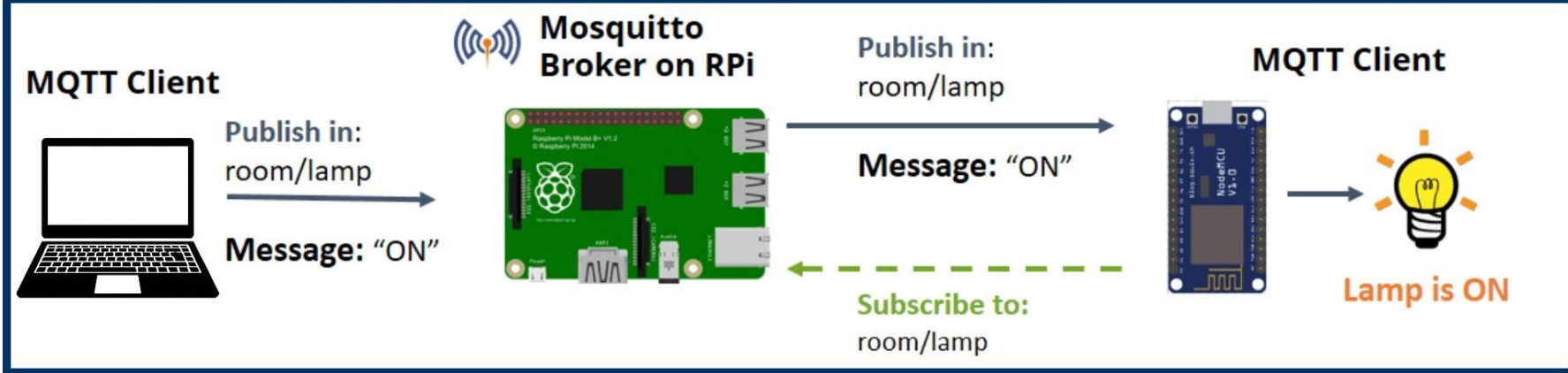
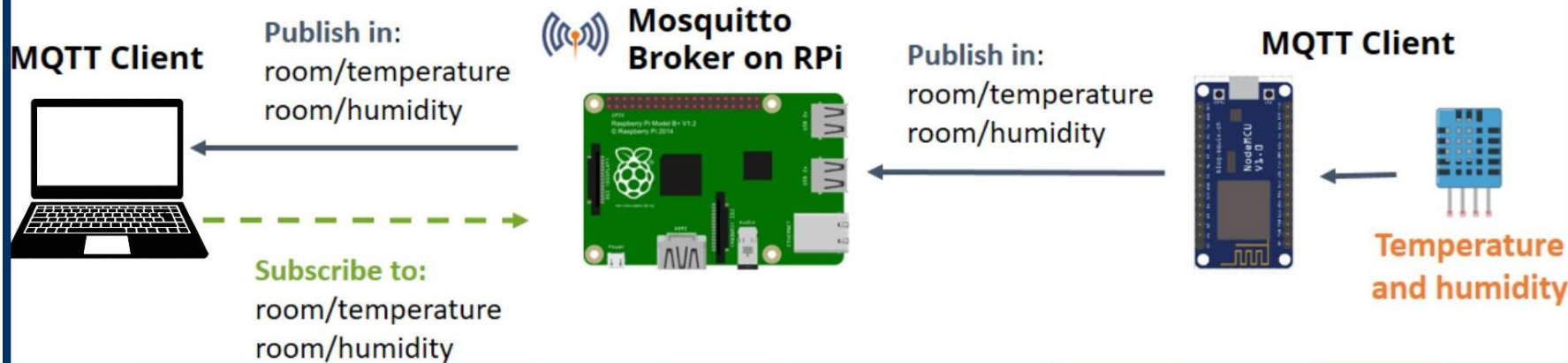
# Tópicos

Tópicos são estruturas às quais os clientes se conectam.  
São strings hierárquicas simples delimitadas pelo caractere "/"

```
1 area/ID_da_area/sensor/ID_do_sensor/temperatura
2 area/ID_da_area/sensor/ID_do_sensor/umidade
```

```
1 area/10/sensor/5000/temperatura
2 area/10/sensor/5000/umidade
3 area/10/sensor/5001/temperatura
4 area/10/sensor/5001/umidade
5 area/20/sensor/4000/temperatura
6 area/20/sensor/4000/umidade
7 area/20/sensor/4001/temperatura
8 area/20/sensor/4001/umidade
```



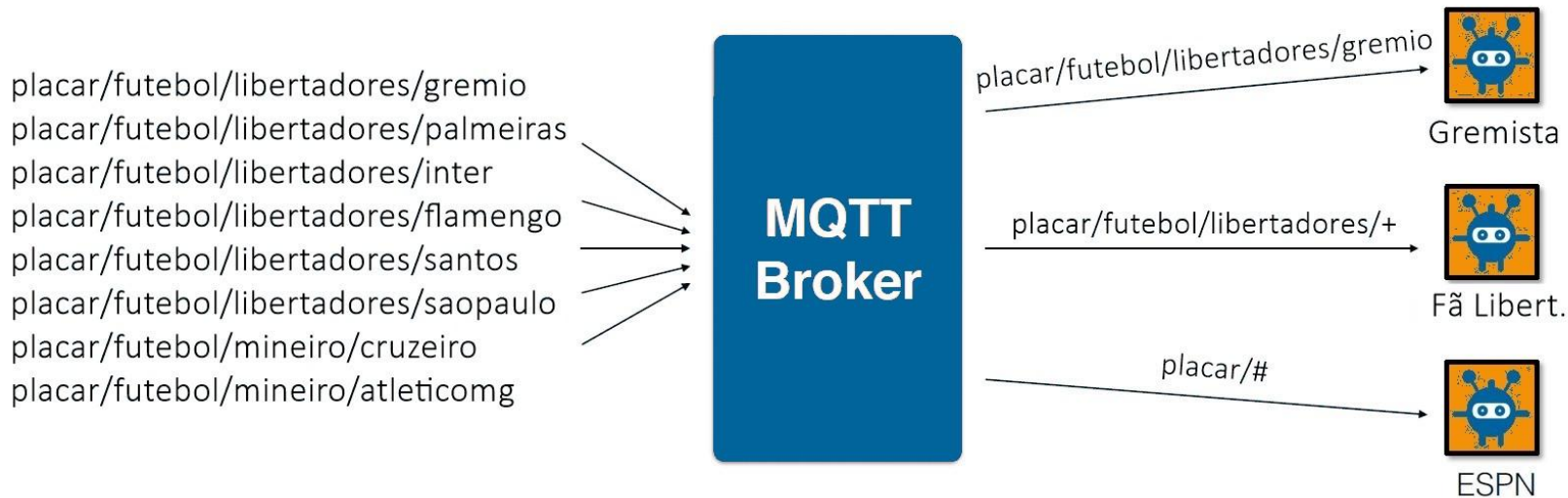


# Comandos

- Connect
- Subscribe
- Publish
- Unsubscribe
- Disconnect

| Name        | Value | Direction of flow                       | Description                              |
|-------------|-------|-----------------------------------------|------------------------------------------|
| Reserved    | 0     | Forbidden                               | Reserved                                 |
| CONNECT     | 1     | Client to Server                        | Connection request                       |
| CONNACK     | 2     | Server to Client                        | Connect acknowledgment                   |
| PUBLISH     | 3     | Client to Server or<br>Server to Client | Publish message                          |
| PUBACK      | 4     | Client to Server or<br>Server to Client | Publish acknowledgment (QoS 1)           |
| PUBREC      | 5     | Client to Server or<br>Server to Client | Publish received (QoS 2 delivery part 1) |
| PUBREL      | 6     | Client to Server or<br>Server to Client | Publish release (QoS 2 delivery part 2)  |
| PUBCOMP     | 7     | Client to Server or<br>Server to Client | Publish complete (QoS 2 delivery part 3) |
| SUBSCRIBE   | 8     | Client to Server                        | Subscribe request                        |
| SUBACK      | 9     | Server to Client                        | Subscribe acknowledgment                 |
| UNSUBSCRIBE | 10    | Client to Server                        | Unsubscribe request                      |
| UNSUBACK    | 11    | Server to Client                        | Unsubscribe acknowledgment               |
| PINGREQ     | 12    | Client to Server                        | PING request                             |
| PINGRESP    | 13    | Server to Client                        | PING response                            |
| DISCONNECT  | 14    | Client to Server or<br>Server to Client | Disconnect notification                  |
| AUTH        | 15    | Client to Server or<br>Server to Client | Authentication exchange                  |

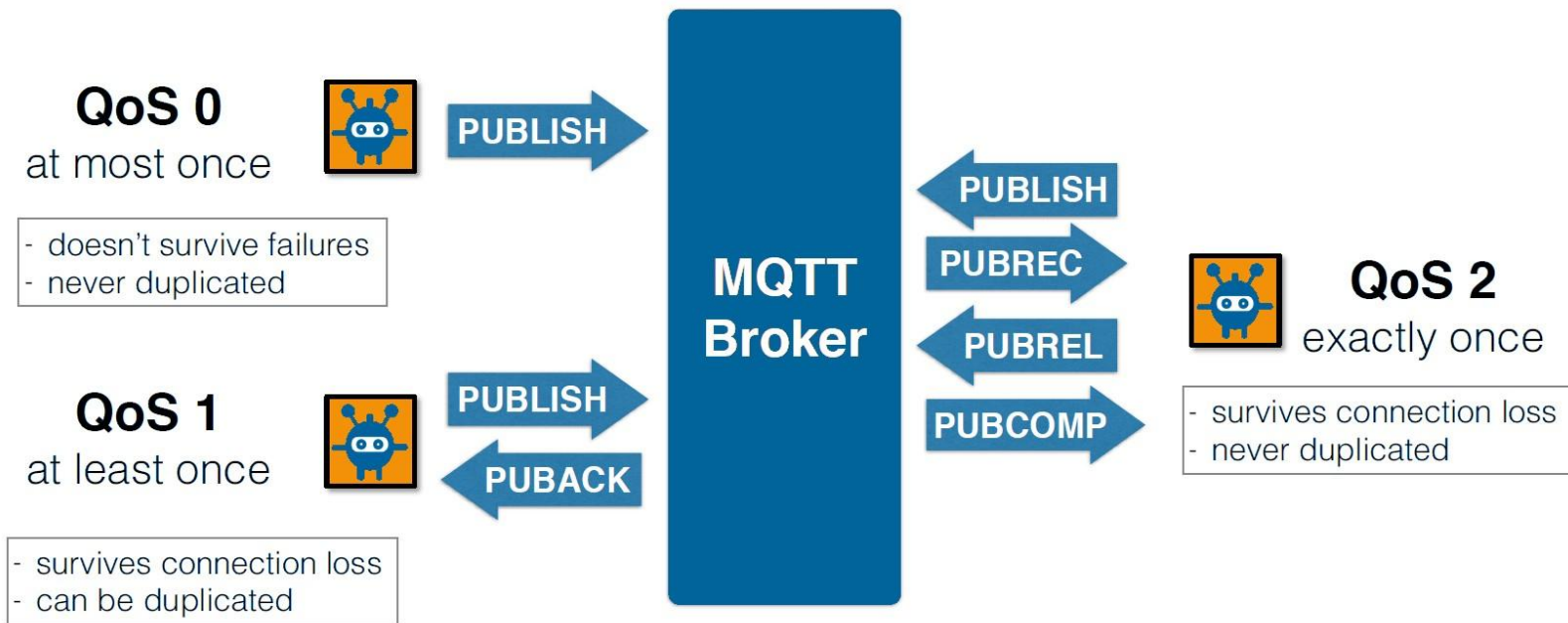
# Wildcards



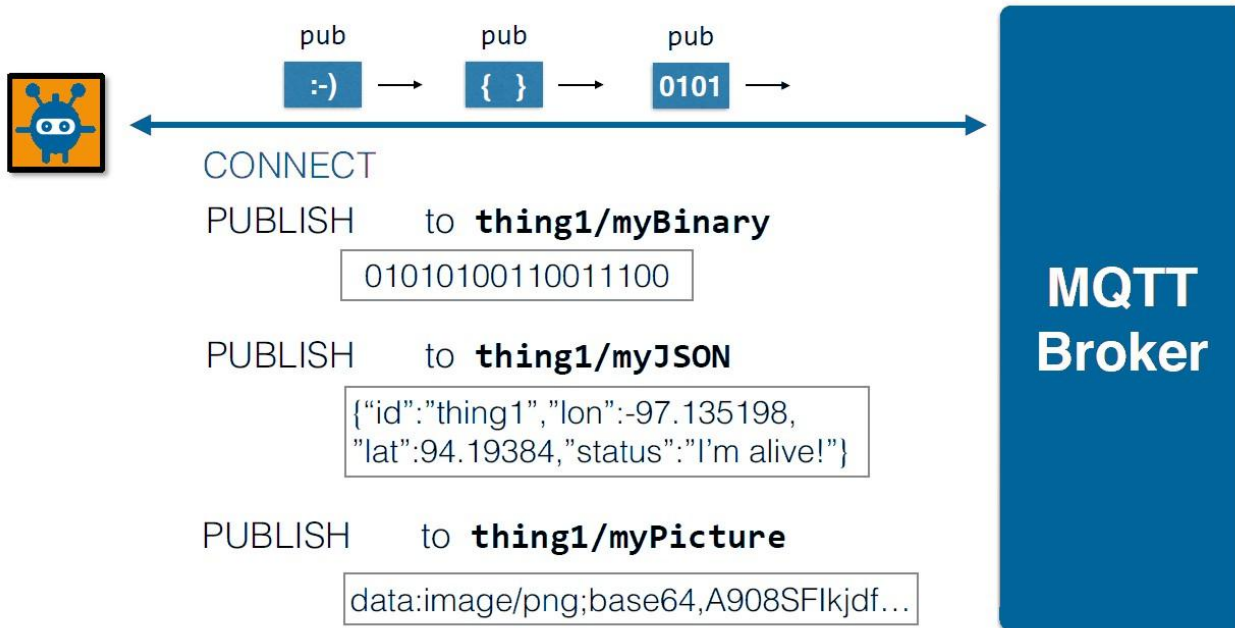
single level wildcard: **+**

multi-level wildcard: **#**

# QoS



# Agnostic payload





# Referências

- IBM  
<https://developer.ibm.com/technologies/iot/>
- MQTT.org  
<http://mqtt.org/>
- Eclipse.org  
<https://iot.eclipse.org> e <http://mosquitto.org/>

