Instituto Federal de Educação, Ciência e Tecnologia do Paraná - IFPR

disciplina de

SERVIÇOS PARA INTEGRAÇÃO DE SISTEMAS

Especialização em Internet das Coisas

PROTOCOLOS PARA IOT

Professor Jefferson de Oliveira Chaves
jefferson.chaves@ifpr.edu.br

Agenda

- Protocolos para contexto de IoT
  - Protocolos Camadas Física e de Enlace : Zigbee
  - Camada de Aplicação: Matter
  - Camadas de Rede e Aplicação: MQTT e HTTP
- Características dos protocolos (uso, disponibilidade, segurança, etc);
- Laboratórios de testes;

Protocolos

- Os sistemas de IoT (Internet das Coisas) dependem de protocolos que permitem a comunicação, troca de dados e interoperabilidade entre dispositivos e serviços;
- Esses protocolos atuam em diferentes camadas e com diferentes finalidades;
- Os protocolos variam de acordo com as necessidades:
  - Tempo de resposta exigido (tempo real ou não);
  - Distância entre dispositivos;
  - Quantidade de dispositivos;
  - Ambiente (ruído, vibração, Electromagnetic Compatibility);
  - Integração com sistemas e nuvem;
  - etc;

Aplicações

- **IIoT:** Controle e monitoramento de equipamentos (condições de temperatura, vibração, pressão, etc.), monitoramento de consumo energético, segurança, monitoramento de condições ambientais (gases tóxicos, temperatura extrema), controle de qualidade do produto ou serviço em tempo real;
- Protocolos:
  - Modbus: Serial/TCP (ponto a ponto) - Simples e amplamente usado, suporte via RS-232, RS-485 ou TCP/IP
  - PROFIBUS: Fieldbus (serial) - Comunicação rápida entre CLPs e sensores/atuadores em campo
  - PROFINET: Ethernet Industrial - Tempo real com alta precisão, usado em sistemas de controle de movimento;

Aplicações

- **Prédios e condomínios:** Controle automatizado de iluminação, temperatura, monitoramento da qualidade do ar interno (níveis de CO2, poluentes, umidade), gestão da água, monitoramento e manutenção preditiva, monitoramento da integridade estrutural do prédio (como vibrações, fissuras e umidade), etc.;
- Protocolos:
  - KNX: Protocolo internacional para automação predial e residencial.
  - BACnet: Usado em sistemas de automação predial (BMS); Ideal para controle de HVAC (climatização), iluminação, sensores, alarmes.
  - Zigbee e Z-Wave: Protocolos sem fio, muito usados em automação residencial e retrofit de prédios; Comunicação entre sensores, lâmpadas, tomadas, fechaduras inteligentes.
  - Matter (em expansão): Protocolo recente de interoperabilidade entre dispositivos IoT; vem ganhando espaço em residências e edifícios inteligentes por unir grandes marcas (Apple, Google, Amazon);
  - Etc.

Aplicações

- **Domótica ou IoT Residencial:** controle e gerenciamento de dispositivos com assistente de voz. Controle de iluminação, controle de segurança doméstica (câmeras, fechaduras, sensores), controle de climatização, controle de sistemas de entretenimiento (televisão, video games), gestão de energia, etc.
- Protocolos:
  - Zigbee, Z-Wave;
  - Wi-Fi;
  - Matter (novo padrão unificador);
  - Bluetooth;
  - Proprietários (menos recomendados): não são protocolos, mas cabe a menção as plataformas Tuya, eWeLink, etc

E a conexão com a web?

Independentemente da aplicação, quando a comunicação com a web é necessária, normalmente são utilizados os protocolos:

- MQTT - modelo (publish/subscribe) - Leve, ideal para integração de sensores com nuvem e sistemas ou sistemas;
- HTTP: modelo (cliente-servidor) - Integração com APIs, dashboards, e sistemas web
- AMQP - Fila de mensagens - Mais pesado que MQTT, usado quando há necessidade de confiabilidade elevada
- etc.;

Domótica ou IoT residencial

*“Domótica é a automatização e o controle aplicados à residência, realizados mediante o uso de equipamentos com capacidade para se **comunicar interativamente entre eles** e com capacidade de **seguir as instruções de um programa** previamente estabelecido pelo usuário da residência e com possibilidades de alterações conforme seus interesses.*

Asociación Española de Domótica, 2018

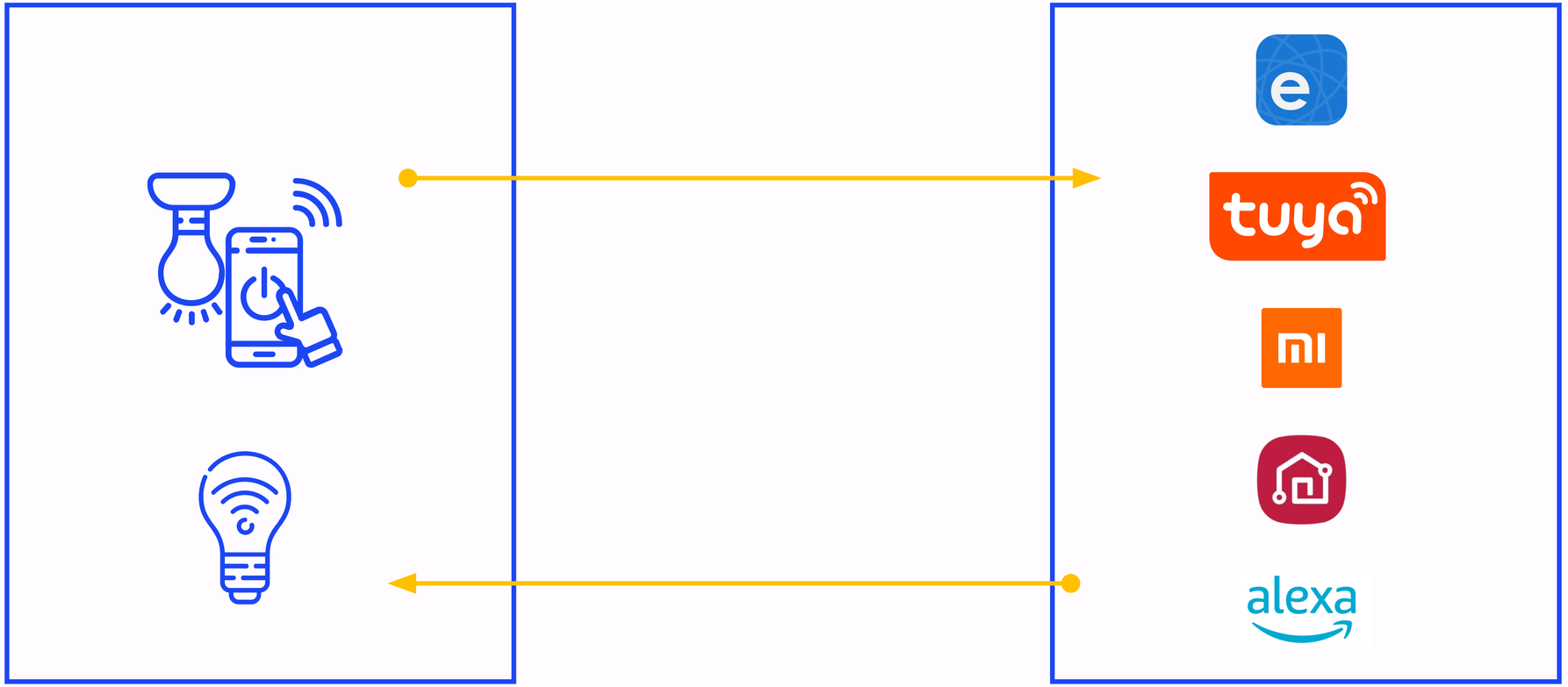
Automação Residencial

- Sistemas elétricos: iluminação, persianas e cortinas automatizadas, gestão de energia e outros;
- Sistema de segurança: alarmes de intrusão, alarmes técnicos (fumaça, vazamento de gás, inundação), circuito fechado de TV, monitoramento, controle de acesso;
- Sistemas multimídia: áudio e vídeo, som ambiente, jogos eletrônicos, além de vídeos, imagens e sons sob demanda;
- Sistemas de comunicações: telefonia e interfonia, redes domésticas;
- Utilidades: irrigação, piscina, climatização, aquecimento de água, bombas e outros.

Plataformas de Automação

- As principais empresas de IOT hoje, possuem plataformas para gerenciamento de seus dispositivos;
- O acesso aos dispositivos é realizado pela plataforma (comandos de automação são processados pelos servidores de cada plataforma);
- pouca coisa é feita localmente;
- Gatilhos de automação dessas plataformas realizam uma requisição que vai pela internet até os servidores da empresa e volta até a sua casa para o dispositivo;
- É necessário (na maioria dos casos) conexão com a internet para o dispositivo ser acionado e ocorrer a automação.

Plataformas de Automação



Plataformas de Automação

- Produtos “White Label”:
  - modelo de negócio em que um produto ou serviço desenvolvido por determinada empresa pode ser revendido por outras empresas ou pessoas físicas sem divulgação dos direitos autorais.
- Esse modelo é comum no Brasil: Positivo, NovaDigital, Kabum, Intelbras, Multilaser, Ekaza, i2go, Geonav, AGL, Elgin, Smarteck, RSmart, Líder, Lis, GAYA e muitas outras utilizam a plataforma Tuya.
- Com a compra dos direitos, são adquiridos também o acesso aos servidores, aplicativos e etc.

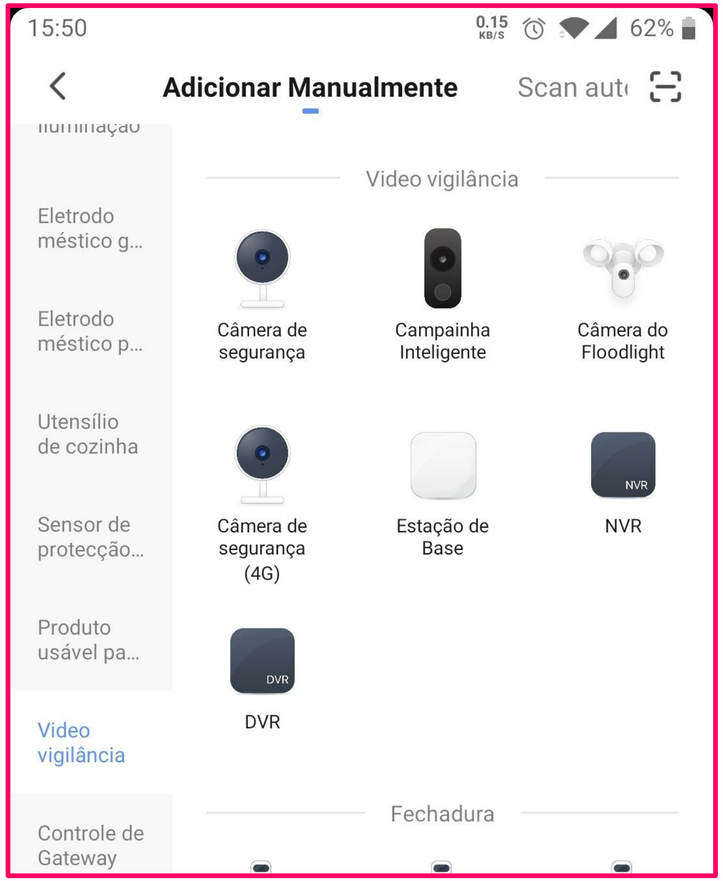
Plataformas de Automação

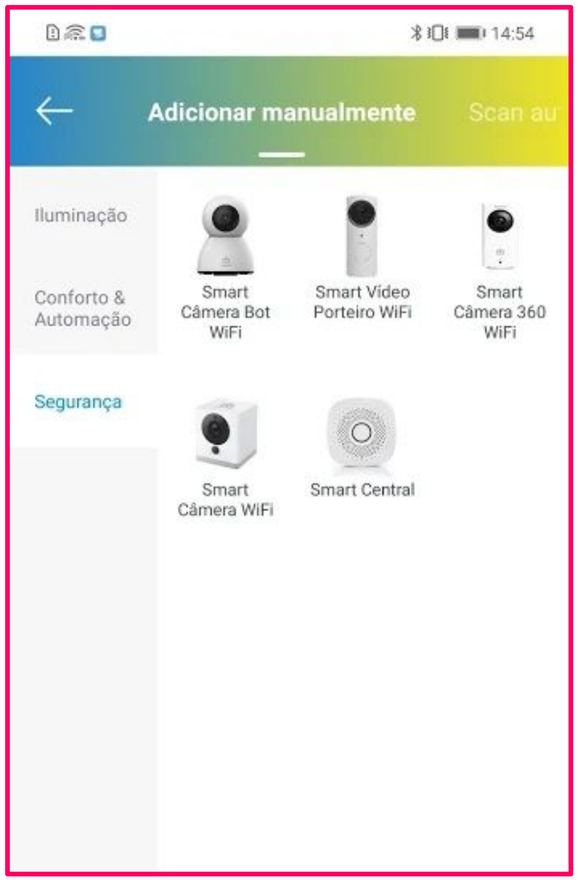
Hub Zigbee white Label

Hub Zigbee Blitzwolf

Hub Zigbee Intelbras

Plataformas de Automação

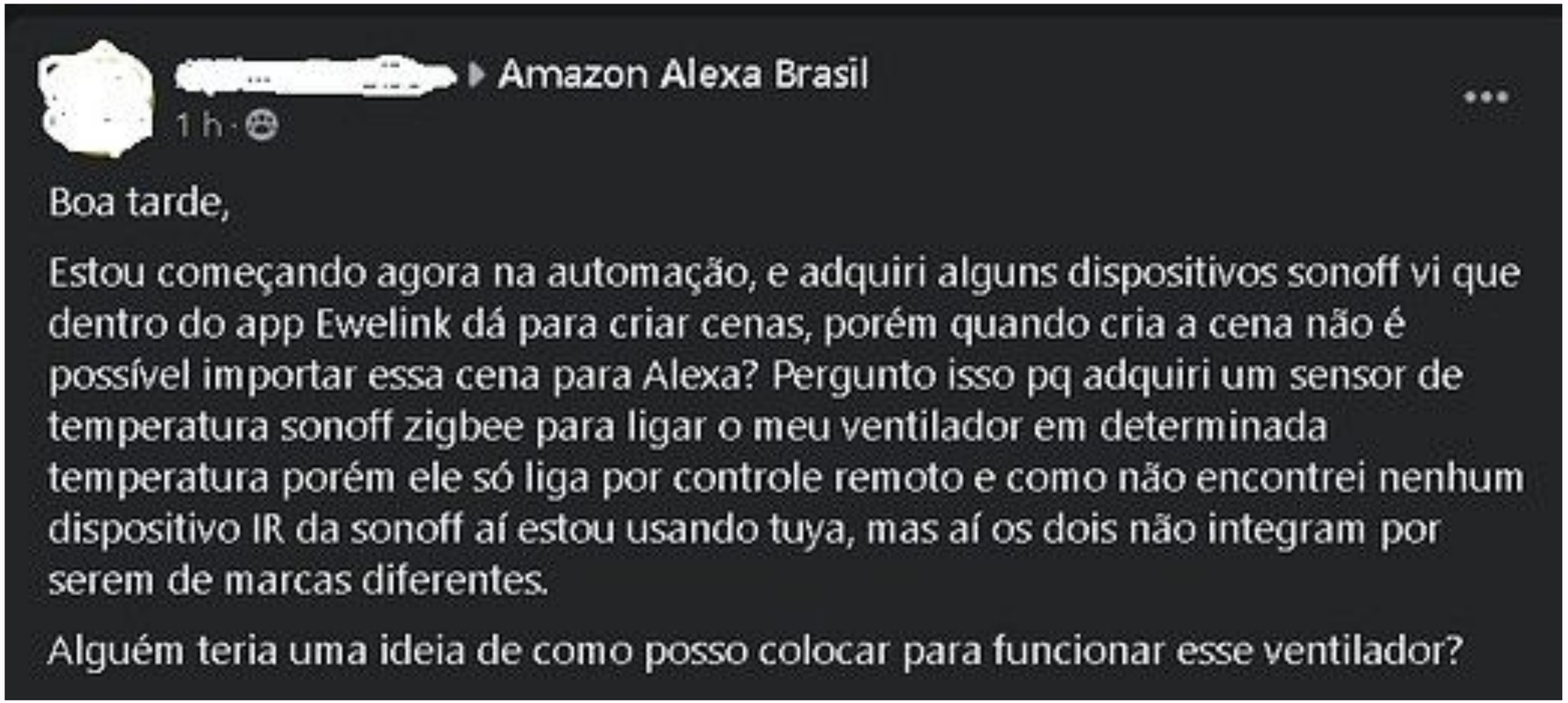




Plataformas de Automação

- Produtos de plataformas diferentes, normalmente, não trocam dados entre para criarem uma automação;
- De forma geral, não é possível adicionar um produto de uma plataforma em outra;
  - Exemplo: sensores de porta de uma plataforma não se comunicam com uma lâmpadas de outra, então não é possível criar uma automação simples que ao abrir uma porta, acenda uma lâmpada.

Plataformas de Automação



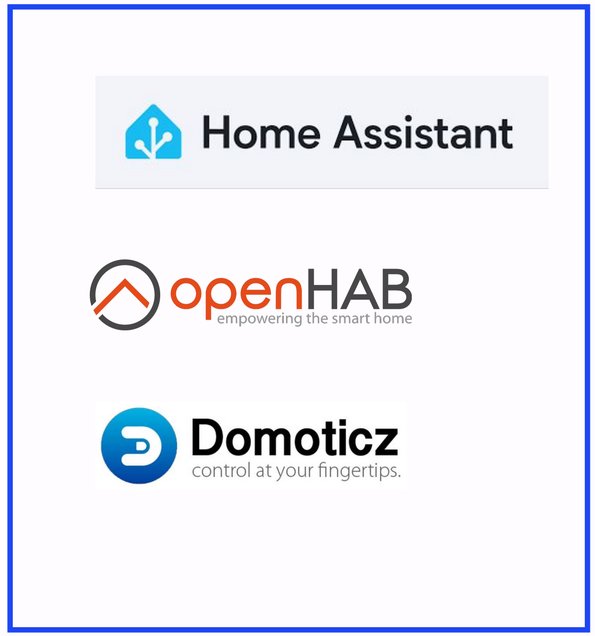
Integração entre dispositivos

Hubs de Integração

- Hubs de integração são dispositivo, serviços ou plataformas que realiza a interconexão entre as diversas plataformas, tecnologias e protocolos;
- Com tais hubs é possível que um sensor de porta da plataforma Tuya ligue uma luz da plataforma Ewelink, por exemplo.

Hubs de Integração

- Existem vários hubs de integração, dentre os quais destacam-se:



super integradores

Hubs de Integração

- O Home Assistant é um software de automação residencial gratuito e de código aberto com foco no controle local e na privacidade;
- Trata-se de um “sistema operacional” open source baseado em Linux;
- É recomendado que sua instalação seja realizada em um hardware dedicado, tal como um mini pc ou um Raspberry Pi;

Hubs de Integração

- Possui integração com + de 2000 plataformas diferentes, dentre as quais:
  - Xiaomi, Tuya, Ewelink, Broadlink, IFTTT, Smartthings, Hubitat, Philips Hue, Amazon Alexa, Google Home, Apple Homekit, Magic Home, LG SmartThinQ, LG webOS, Samsung TV Tizen, GoogleTV/AndroidTV, Amazon FireTV, Roku, Telegram, Traccar, ESPHome, Tasmota, Sonos, Shelly, Xbox, Ps4, Tesla, Nissan Leaf, EPSON, HP, PLEX, Twitch, Youtube, Spotify, ClaroTV e etc.

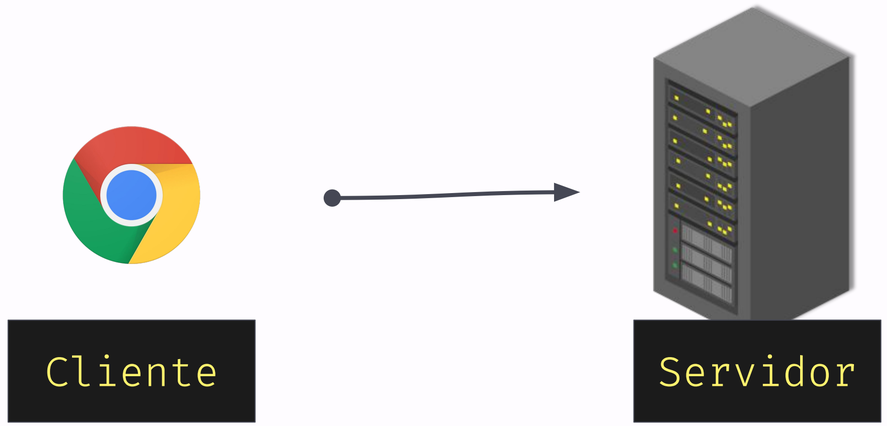
Hubs de Integração

- Automações complexas podem ser facilmente criadas, tanto por meio do próprio Home Assistant (automações, cenas e scripts), quanto por outras integrações, tais como NODE RED.

Modelo
Cliente /Servidor

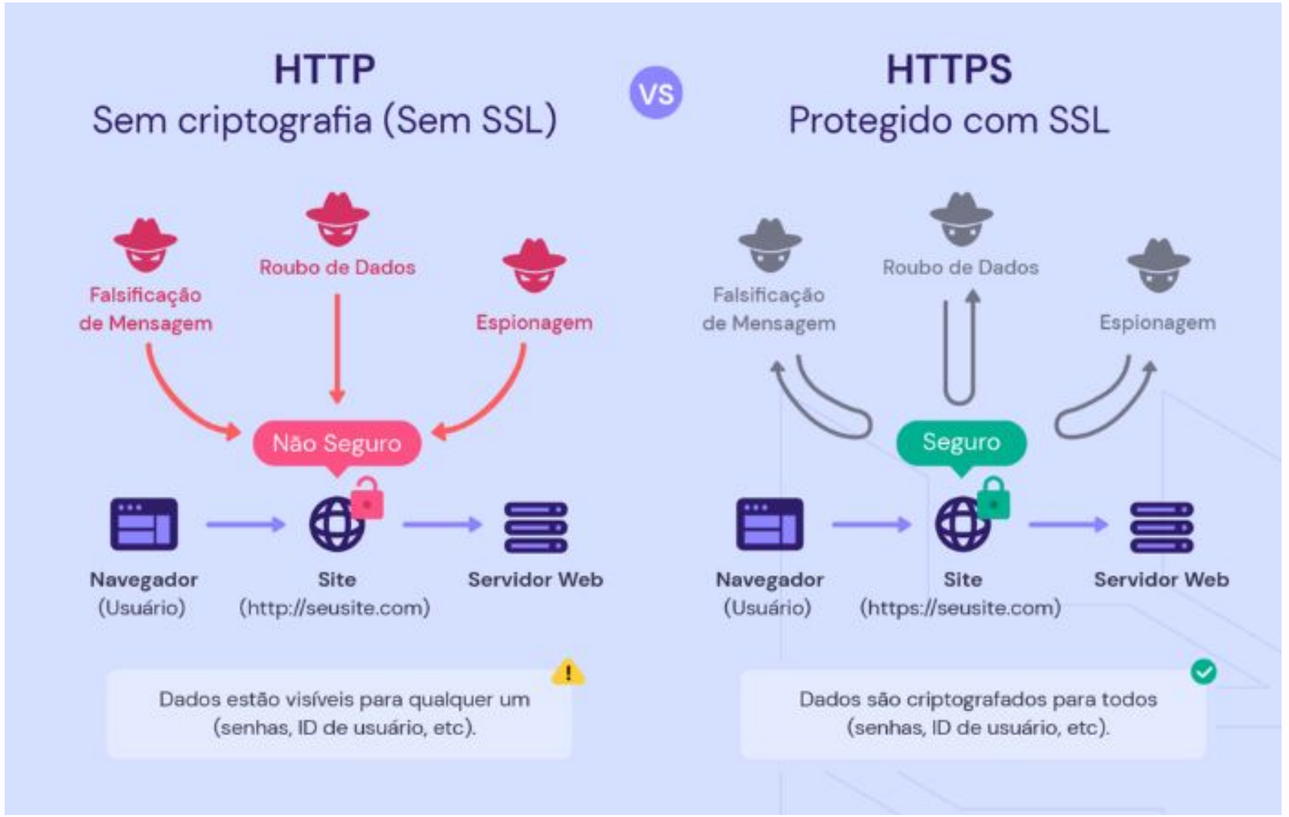
Modelo Cliente / Servidor

- Assimetria de poder computacional;
- Centralização de dados;
- Comunicação unidirecional;
- Processo inicia-se pelo cliente;
- Relativamente seguro;
- Se adequada bem as necessidades de IoT
  - Servidor no cliente?



Modelo Cliente / Servidor

- Principais representantes:
  - HTTP;
  - HTTPS;
  - CoAP.



Modelo Cliente / Servidor

- Uma forma análoga de se pensar no modelo cliente servidor como um pedido do IFood:
  - O IFood não te enviará um lanche, sem que você tenha feito um pedido;
  - O IFood não pede lanches a você! Ele é o “servidor” de lanches. Você é apenas um cliente;
  - Se quiser outro lanche, deverá fazer outro pedido, ainda que o lanche possa ser o mesmo;
  - Apenas o IFood tem ciência do seu pedido.

HTTP - Hyper Text Transfer Protocol

- O HyperText Transfer Protocol (HTTP) é um protocolo de aplicação responsável pelo tratamento de pedidos e respostas entre cliente e servidor na World Wide Web;
- Surgiu da necessidade de distribuir informações pela Internet;
- Com isso, o protocolo HTTP passou a ser utilizado para a comunicação entre computadores na Internet e a especificar como seriam realizadas as transações entre clientes e servidores, por meio do uso de “regras” protocolares.

HTTP - Hyper Text Transfer Protocol

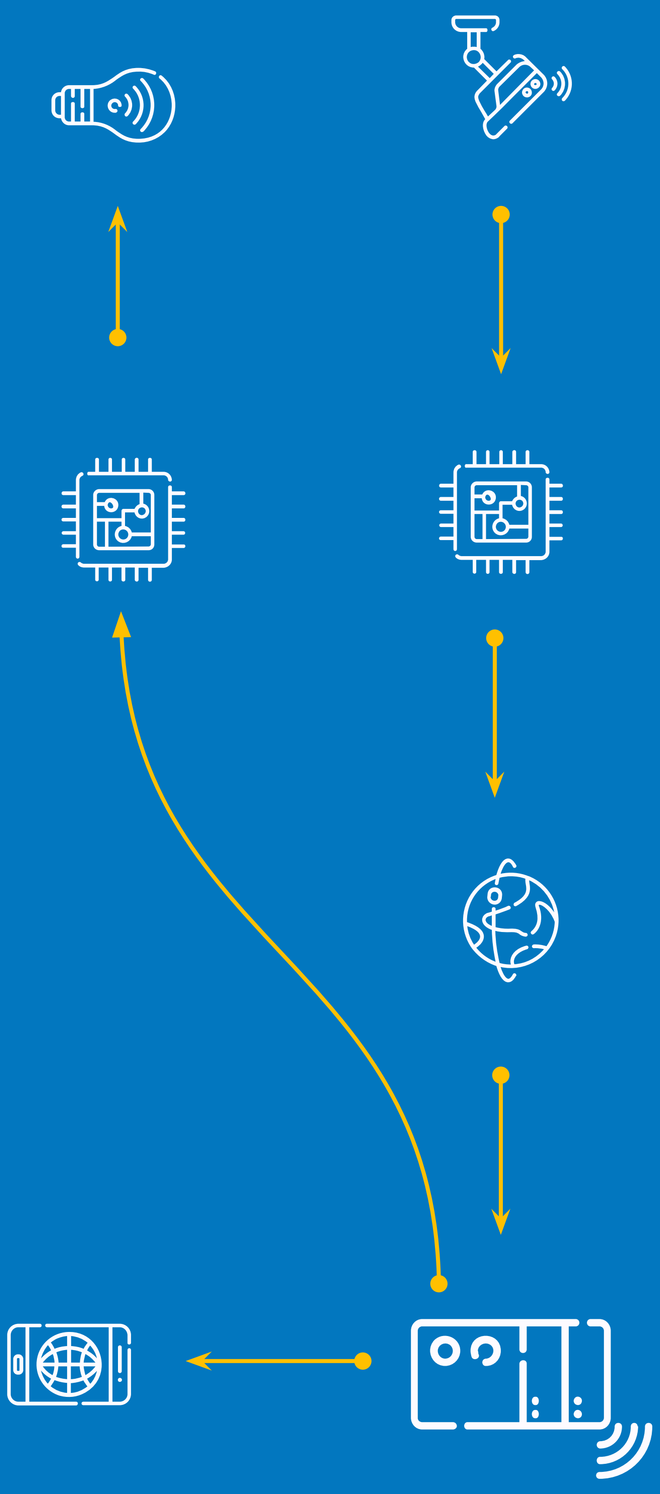
- Modelo cliente-servidor;
- As requisições são iniciadas por um cliente;
- Servidor é a entidade FIM (Endpoint);
- Permite a obtenção de recursos, da web;
- Síncrono;
- Stateless.

HTTP - Hyper Text Transfer Protocol

- Possui uma interface uniforme de requisição, resposta;
  - Requisições: GET, POST, PUT, DELETE, PATCH, OPTIONS, ETC.
  - Respostas:
    - Respostas de informação (100-199);
    - Respostas de sucesso (200-299);
    - Redirecionamentos (300-399);
    - Erros do cliente (400-499);
    - Erros do servidor (500-599).
- “Arquitetura” relativamente leve e bem definida (RestFull);
- Orientado a recursos: “/spot-1265/light”;
- Recursos devem ser implementados.

Podemos considerar o HTTP adequado para IoT?

Cenário: detecção de imagens



SERVIDOR NÃO ENVIA DADOS?

- O modelo cliente servidor não permite* ao servidor enviar dados ao cliente sem uma requisição explícita;
- Para contornar esse problema, os desenvolvedores usam várias técnicas:
  - Executar ping no servidor periodicamente por meio das APIs Ajax;
  - Fetch, usando a WebSockets API ou protocolos semelhantes.
  - Pool de espera;

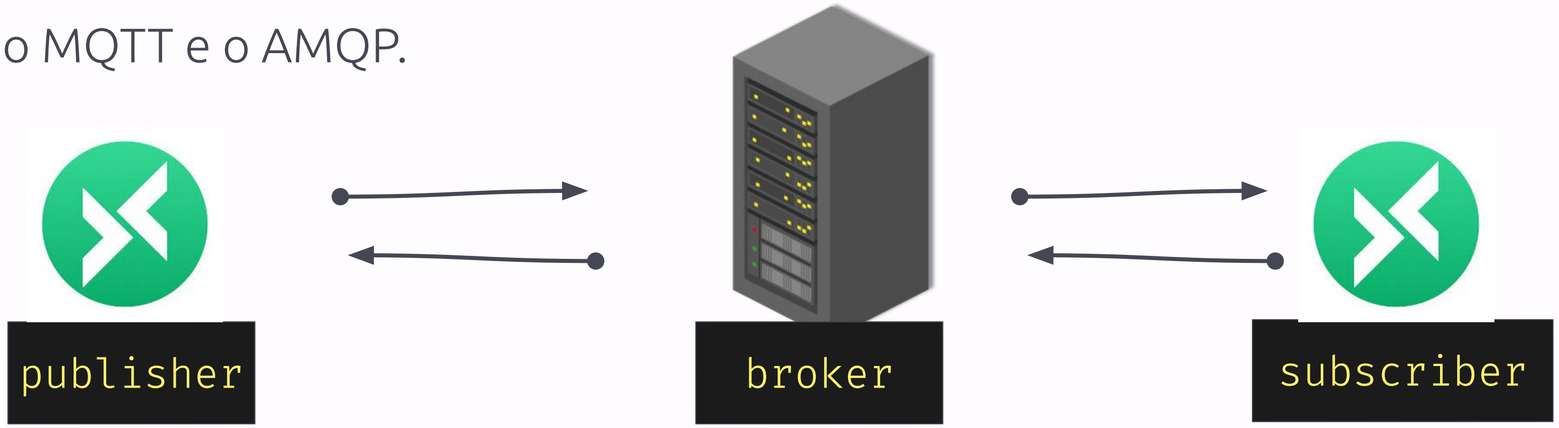
Links Interessantes

- https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview
- https://kinsta.com/pt/blog/http3/

Modelo Publisher / Subscriber

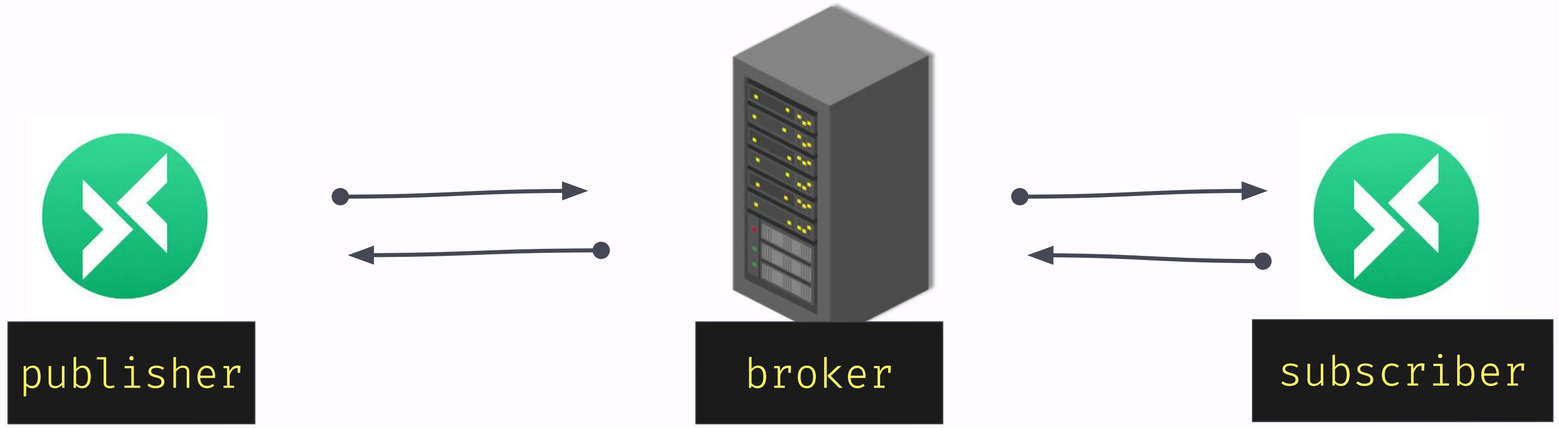
Modelo Publisher / Subscriber

- Comunicação bidirecional;
- Baseia-se no conceito de tópicos e assinaturas;
- Baixo Overhead de Rede;
- Qualidade de Serviço (QoS);
- Relativamente seguro;
- Se adequada bem as necessidades de IoT;
- São exemplos o MQTT e o AMQP.



Modelo Publisher / Subscriber

- Uma analogia possível de se fazer é comparar o modelo Pub/Sub com o sistema de transmissão de rádio;
  - Uma emissora transmite uma programação, sem se preocupar com quem irá consumir esse conteúdo;
  - O cliente por sua vez deve sintonizar o canal ou estação para podem ter acesso a esse conteúdo;
  - Nesta analogia, um cliente também pode produzir conteúdo de rádio.



MQTT - Message Queuing Telemetry Protocol

- Protocolo de comunicação M2M (Machine to Machine);
- Desenvolvida em 1999 por Andy Stanford-Clark (IBM) e Arlen Nipper;
- Tinha como objetivo inicial conectar **sistemas de telemetria de oleodutos via satélite**;
- Liberada para uso gratuito em 2010, padrão OASIS em 29/10/2014;
- OASIS – Organization for the Advancement of Structured Information Standards.

MQTT - Message Queuing Telemetry Protocol

- Hoje, é definida em diversos livros como “o protocolo para IoT” (polêmico);
- Quando comparado ao HTTP, apresenta algumas vantagens em termos de IoT:
  - Conceito de distribuição Publisher/Subscriber;
  - Vários níveis de serviço, com garantia de entrega;
  - Foco no transporte e não no conteúdo, maior nível de segurança;
  - Envio de dados 1-0, 1-1, 1-N
  - Desenvolvido para infraestruturas de comunicação instáveis;

MQTT - Message Queuing Telemetry Protocol

- Adequado à relevância de cada mensagem;
- Controle de desconexão com notificação;
- Alta segurança;
- Criptografia da conexão e dos dados é possível;

MQTT - Message Queuing Telemetry Protocol

- Publisher (Publicador): Cliente que envía pacotes de dados;
- Subscriber (Assinante): Cliente que recebe pacotes de dados;
- Broker (Agente): Recebe dados do Publisher, armazena e distribui aos Subscribers;
- **Cliente pode ser Publisher e Subscriber simultaneamente**;
- “Não existe obrigatoriedade de identificação do Cliente perante o Broker”, seja para publicar ou assinar;
  - Obrigatório para sessão persistentes, aplicações com autenticação ou quando existe a necessidade de gerenciamento de conexões..

MQTT - Message Queuing Telemetry Protocol

- Message (Mensagem) / Payload (Conteúdo): Pacote de dados trafegados entre Clientes e Broker;
- Topic (Tópico): Repositório de dados único que recebe a Mensagem, acessível aos Clientes, instantâneo e sem histórico;
- Publish (Publicar): Ação de enviar ao Broker uma Mensagem para um determinado Tópico
- Subscribe (Assinar): Ação de solicitar receber Mensagens de um determinado Tópico
- Unsubscribe (Deixar de Assinar): Ação de solicitar não receber mais Mensagens de um determinado Tópico.

MQTT - Message Queuing Telemetry Protocol

- QoS (Quality of Service): nível de serviço ao qual o broker deve seguir ao transmitir uma mensagem;
  - Nível 0 - at most once (no máximo uma vez);
  - Nível 1 - at least once (pelo menos uma vez);
  - Nível 2 - exactly once (exatamente uma vez);
- Will message / Last will (ultima vontade)
  - Mensagem a ser enviada pelo broker para os subscribers quando o cliente perde a conexão (deixa de enviar PINGREQ);

MQTT - Message Queuing Telemetry Protocol

- PINGREQ e PINGRESP: teste periódico de conexão entre cliente e servidor;
- Session: nível de serviço ao qual o broker deve seguir ao transmitir uma mensagem;
- Clean session:
  - indica se a sessão deve ser persistida;
  - Para sessões persistidas, mantem-se:
    - tópicos assinados;
    - mensagens recebidas enquanto o cliente está desconectado;
  - Ao iniciar uma sessão limpa, os dados da sessão anterior serão descartados;
  - Retain: indica se a mensagem deve ser mantida pelo broker até uma que uma nova mensagem seja recebida.

Demonstração prática

Resumo

- Modelo Publish/Subscribe
- Alternativa ao tradicional modelo Client/Server;
- Não há conexão direta do Client ao Endpoint;
- Desvincula o Cliente Origem (Publisher) do Cliente Destino (Subscriber);
- Clientes não têm contato entre si, nem tomam conhecimento um do outro;
- Cliente atua como Publisher, Subscriber ou ambos;
- Comunicação centrada no Broker.

Para consultas futuras:

Um pouco mais do modelo Publisher / Subscriber

*número indeterminado de publishers e subscribers

publisher/subscriber

subscriber

publisher/subscriber

Broker

subscriber

publishers

publisher/subscriber

*número indeterminado de publishers e subscribers

publisher/subscriber

publish A=33
publish B=true

publisher/subscriber

subscribe A e B
publish B=hello

Broker

subscriber

subscribe B

subscriber

subscribe A

Tópico A	Tópico B	Tópico C
999	world	true

publishers

publish A=999

publisher/subscriber

subscribe A
publish B=world

Processo de Conexão ao Broker

- Processo de Conexão ao Broker Igual para Publisher e Subscriber;
  - Papel não é definido no processo de Conexão
- Dados informados:
  - ID do Cliente – opcional (Obrigatório para sessão com retenção)
  - Flag Clean Session – obrigatório (Indica sessão com ou sem retenção)
  - Usuário e Senha – opcional (conforme política de segurança do Broker)

Processo de Conexão ao Broker

- Dados informados
- Last Will – opcional
  - Dados para definir a Last Will Message
  - Tópico, Mensagem, QoS, Retain
- Keep Alive – obrigatório
  - Intervalo máximo sem comunicação
  - Cliente deve enviar Ping antes de atingir esse intervalo

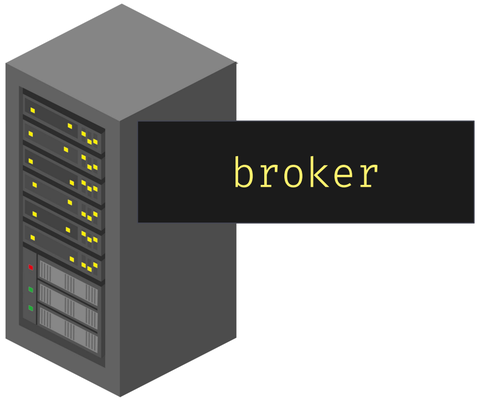
Publicando

- Qualquer cliente, qualquer conteúdo em qualquer Tópico;
- Desde que permitido pelas políticas de Segurança;
- Dados informados:
- ID da Mensagem – opcional (Obrigatório para QoS diferente de 0 (At Most Once));
- Tópico – obrigatório (nome ou caminho do Tópico para publicação);
- QoS – obrigatório (Nível de Serviço).

Publicando

- Dados informados:
  - lag Retain – obrigatório: Indicação para retenção da Mensagem
  - Flag Duplicate – obrigatório Indica que a Mensagem está sendo reenviada
    - Relevante para QoS diferente de 0 (At Most Once)
- Payload – obrigatório
  - Conteúdo da Mensagem;
  - Qualquer conteúdo é válido
- MQTT é agnóstico ao conteúdo da Mensagem

publisher



subscriber

1) Conecta

2) Publica Mensagem

8) Publica Mensagem

3) Recebe Mensagem

4) Algum Assinante?

5) Não, descarta Mensagem

9) Recebe Mensagem

10) Algum Assinante?

11) Sim, distribui Mensagem

6) Conecta

7) Assina Tópico

12) Recebe Mensagem

Tópicos

- Local de armazenamento da Mensagem no Broker
- Identificador textual
- Normalmente com significado em algum idioma
- Diferencia letras maiúsculas e minúsculas

Arquitetura baseada em tópicos

- Organização hierárquica em múltiplos níveis: “Níveis do Tópico”;
- Exemplos:
  - ifpr/sala/temperatura
  - ifpr/sala/umidade
  - ifpr/sala/arcondicionado/modo
  - ifpr/labmaker/luz_impressora/estado
  - ifpr/labmaker/tomada/estado

Arquitetura baseada em tópicos

- Separador nível – “/” (barra);
- Cada nível deve possuir ao menos um caractere;
- Um Separador sozinho é considerado um nível;
- **Coringas (**válidos na assinatura de tópicos**):**
  - Nível único – “+” (sinal de adição);
  - Níveis múltiplos – “#” (cerquilha).

Arquitetura baseada em tópicos {coringas}

- Nível único – “+”;
- Pode ser utilizado em qualquer nível;
- Corresponde a qualquer conteúdo naquele nível;
- Exemplo: assinar tópico **casa/+/temperatura**;
  - casa/sala/temperatura – correto;
  - casa/suite1/temperatura – correto;
  - casa/cozinha – incorreto, apenas dois níveis;
  - casa/estufa/temperatura/interna – incorreto, nível adicional.

Arquitetura baseada em tópicos {coringas}

- Níveis Múltiplos – “#”
- Pode ser utilizado apenas no final
- Qualquer conteúdo e quantidade de níveis
- Exemplo: assinar tópico **casa/#**;
  - casa/sala/temperatura – correto
  - casa/suite1/temperatura – correto
  - casa/cozinha – correto
  - garagem – tópico diferente

Arquitetura baseada em tópicos

- Convencionou-se que tópicos iniciando com “$SYS/” fornecem estatísticas internas do Broker
  - $SYS/broker/clients/total
  - $SYS/broker/messages/sent
  - $SYS/broker/uptime
- Desabilitado na maior parte dos Brokers públicos.

Boas práticas com tópicos

- Não assine o tópico “#”: isso fará com o cliente receba todas as mensagens enviadas pelo Broker;
- Planeje a estutura de tópicos;
- Inclua a identificação do cliente no tópico (auxilia nas politicas de segurança);
- Crie tópicos específicos e especializados:
  - casa/temp-umi - incorreto
  - casa/temperatura - correto
  - casa/umidade - correto

QoS (Quality of Service)

- Umas das principais característica do protocolo;
- Define o nível de garantia de entrega de uma Mensagem ao Subscriber;
- Impacta diretamente no fluxo de tráfego;
- **O QoS esperado é definido pelo Publisher!**
  - Nível máximo de execução do serviço;
- O QoS no recebimento é definido pelo Subscriber;
  - Nível real de execução do serviço;
  - Pode ser inferior ao definido pelo Publisher.

Níveis de QoS

- 0 – At most once (No máximo uma vez)
  - Mensagem será entregue até uma única vez
  - Ou nenhuma!
- 1 – At least once (Pelo menos uma vez)
  - Mensagem será entregue ao menos uma vez
  - Pode ocorrer entrega duplicada, triplicada...
- 2 – Exactly once (Exatamente uma vez)
  - Mensagem será entregue exatamente uma vez

QoS – Qual Nível Utilizar e Quando

0 – At most once

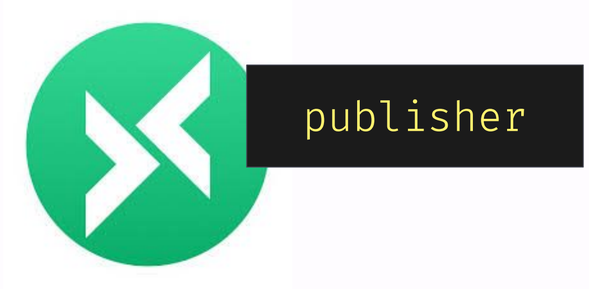
- As conexões são estáveis
- A perda de alguma mensagem é aceitável
- As mensagens não precisam ser persistidas

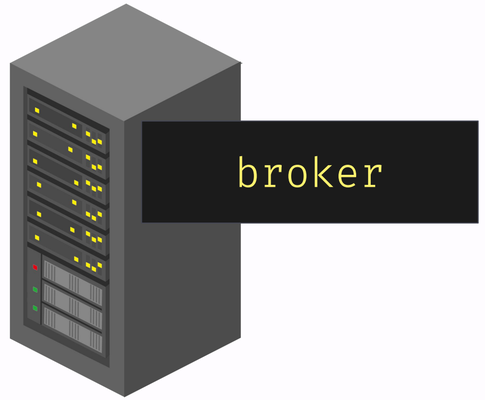
1 – At least once

- Todas as mensagens precisam ser recebidas, mesmo que eventualmente duplicadas
- O tráfego adicional gerado por eventuais mensagens duplicadas é
- mais aceitável do que a tráfego gerado pelo QoS 2

2 – Exactly once

- A precisão no recebimento da mensagem é crítico
- A sobrecarga de tráfego e processamento é justificável





0) Publica Mensagem (QOS 0 No máximo uma vez)

1) Publica Mensagem (QOS 1 Pelo menos uma vez)

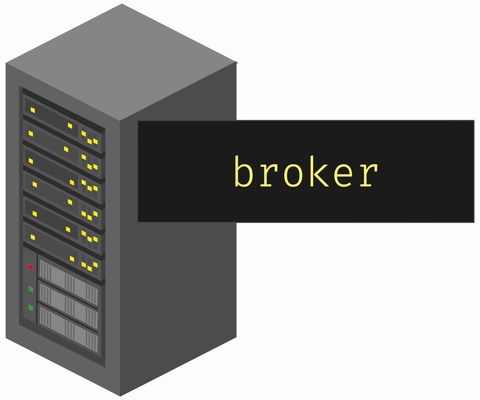
1) Publicação recebida (PUBACK)

2) Publica Mensagem (QOS 2 Exatamente uma vez)

2) Publicação recebida

2) Publicação encerrada

2) Publicação completada



0) Envia Mensagem (QOS 0 No máximo uma vez)

1) Envia Mensagem (QOS 1 Pelo menos uma vez)

1) Envio recebido

2) Envia Mensagem (QOS 2 Exatamente uma vez)

2) Envio recebido

2) Publicação encerrada

2) Publicação completada

HTTP vs MQTT

Comparações de características

	MQTT	HTTP
Arquitetura	Publicar assinatura (o MQTT também possui um modo de solicitação/resposta)	Solicitar resposta
Alvos de comando	Tópicos	URIs
Protocolo subjacente	TCP/IP	TCP/IP
Conexões seguras	TLS + nome de usuário/senha (possível suporte a SASL)	TLS + nome de usuário/senha (possível suporte a SASL)
Observabilidade do cliente	Status de conexão conhecido (será mensagens)	Status de conexão desconhecido
Modo de mensagens	Assíncrono, baseado em eventos	Síncrono
Enfileiramento de mensagens	O broker pode enfileirar mensagens para assinantes desconectados	O aplicativo precisa ser implementado

Comparações de características

	MQTT	HTTP
Sobrecarga da mensagem	mínimo de 2 bytes. Os dados do cabeçalho podem ser binários	Mínimo de 8 bytes (os dados do cabeçalho são texto - compactação possível)
Tamanho da mensagem	Máximo de 256 MB	Sem limite, mas 256 MB está além dos casos de uso normais de qualquer maneira.
Tipo de conteúdo	Qualquer (binário)	Texto (codificação Base64 para binário)
Distribuição de mensagens	Um para muitos	Um a um
Confiabilidade	Três qualidades de serviço: **0** - disparar e esquecer, **1** - pelo menos uma vez, **2** - uma vez e apenas uma vez	Tem que ser implementado no aplicativo

Comparações de características

	MQTT	HTTP
Sobrecarga da mensagem	mínimo de 2 bytes. Os dados do cabeçalho podem ser binários	Mínimo de 8 bytes (os dados do cabeçalho são texto - compactação possível)
Tamanho da mensagem	Máximo de 256 MB	Sem limite, mas 256 MB está além dos casos de uso normais de qualquer maneira.
Tipo de conteúdo	Qualquer (binário)	Texto (codificação Base64 para binário)
Distribuição de mensagens	Um para muitos	Um a um
Confiabilidade	Três qualidades de serviço: **0** - disparar e esquecer, **1** - pelo menos uma vez, **2** - uma vez e apenas uma vez	Tem que ser implementado no aplicativo

Comparações de quantidades de bytes

	Bytes MQTT	Bytes HTTP
Estabelecer conexão	5572	2261
desconectar	376 (opcional)	0
Para cada mensagem publicada	388	3285
Soma para 1 mensagem	6336	5546
Soma para 10 mensagens	9829	55.460
Soma para 100 mensagens	44.748	554.600

Comparações de tempo médio

Nº de mensagens em um ciclo de conexão para MQTT	Média MQTT tempo de resposta por mensagem (ms) (QoS 1)	média HTTP tempo de resposta por mensagem (ms)
1	113	289
100	47	289
1000	43	289

Obrigado