

Os desafios/problemas para interoperabilidade aplicada à Cidades Inteligentes

Grupo - 9 IoT

Integrantes:

- MAURICIO GOMES ROCHA – 20210703
- ONORTON ALMEIDA PONTES – 202203526
- RAFAEL RIBEIRO M. PARREIRA – 202107035
- ROGÉRIO RODRIGUES ROCHA – 202203530
- VICTOR HUGO BENATTI FRANCO - 202203532

Os desafios para interoperabilidade aplicada à Cidades Inteligentes

A medida que a adoção da IoT aumenta, também surgem desafios significativos relacionados à comunicação entre esses dispositivos. Muitos deles utilizam protocolos de comunicação diferentes, o que dificulta a criação de soluções abrangentes e eficazes.

A falta de padronização compromete a capacidade de diferentes dispositivos se comunicarem de maneira eficiente, prejudicando a criação de ecossistemas verdadeiramente integrados.



Os desafios para interoperabilidade aplicada à Cidades Inteligentes

Com base nisso, uma possível sugestão para tal problema é a concepção de um padrão de intercomunicação destinado a integrar os variados sistemas dos dispositivos IoT em uma Cidade Inteligente.

Para alcançar esse objetivo, propõe-se a exploração de frameworks que promovam a padronização e facilitem a interação com os diversos tipos de sistemas, possibilitando a troca de dados e informações.



Protocolos de Comunicação na IoT

Com base no contexto apresentado, para alcançar a interoperabilidade entre dispositivos IoT, é vital compreender os diferentes protocolos de comunicação utilizados.

Temos como exemplos de protocolos de comunicação na IoT:

- MQTT
- CoAP
- HTTP
- AMQP
- Wi-fi.

Protocolos de Comunicação na IoT

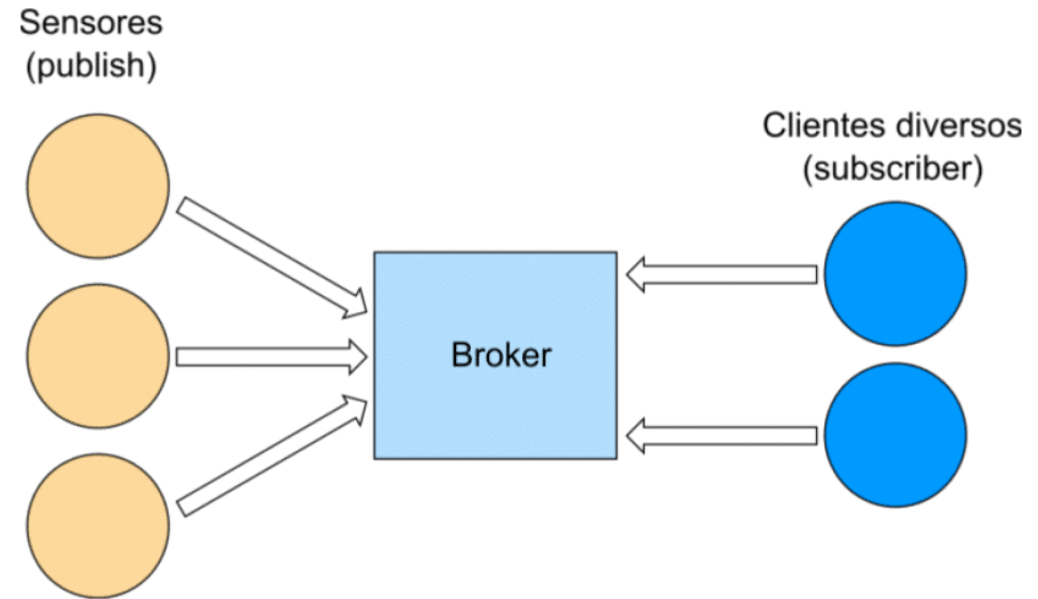
MQTT:

Vantagens:

- Baixo consumo de largura de banda.
- Adequado para dispositivos com recursos limitados.
- Suporte a notificações push.

Limitações:

- Falta de mecanismos integrados de segurança, podendo depender de camadas adicionais.



Protocolos de Comunicação na IoT

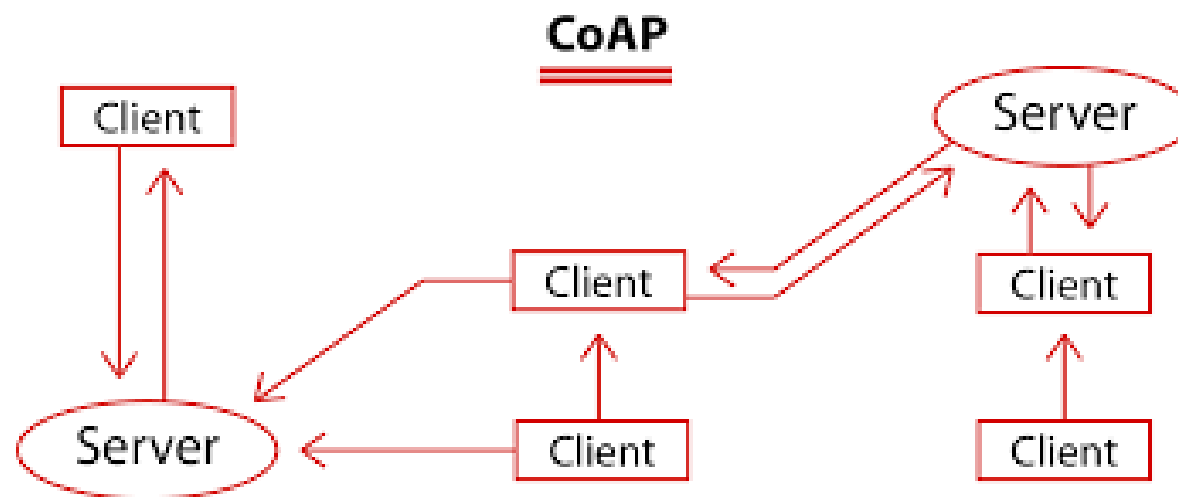
CoAP:

Vantagens:

- Consumo eficiente de recursos
- Menor sobrecarga em comparação com o HTTP.

Limitações:

- Menos difundido que o HTTP, o que pode limitar a interoperabilidade em alguns casos.



Protocolos de Comunicação na IoT

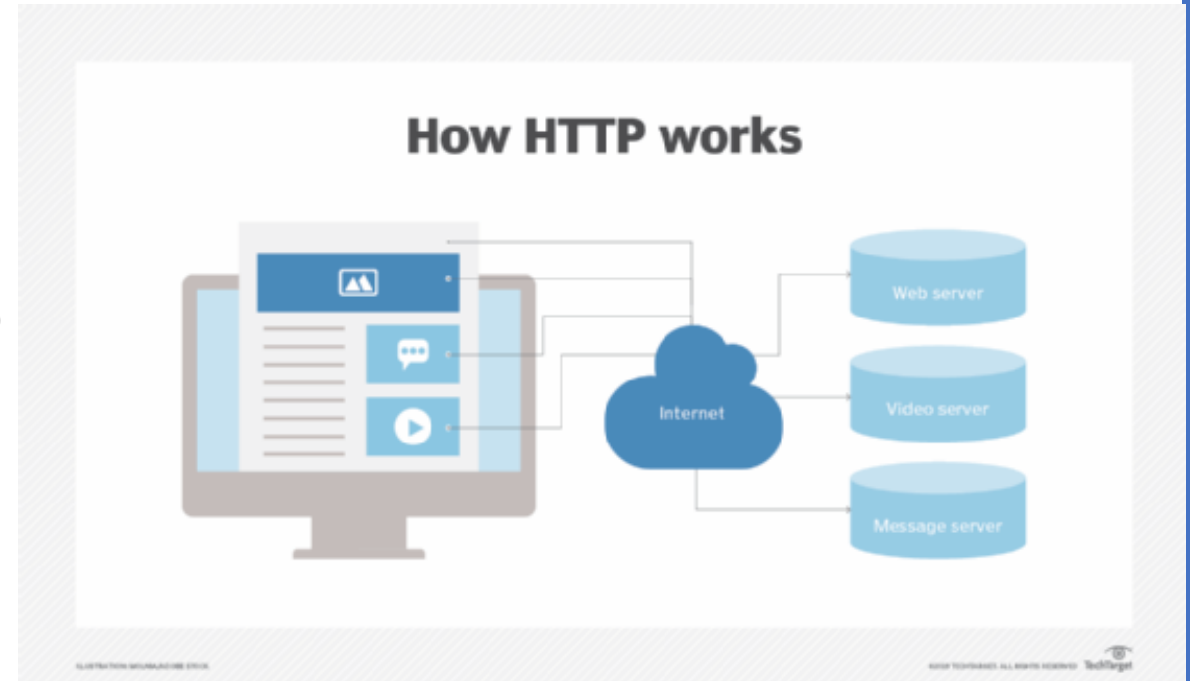
HTTP:

Vantagens:

- Fácil integração com a infraestrutura web existente
- Suporte generalizado em dispositivos e plataformas
- Compatibilidade com firewalls e proxies.

Limitações:

- Maior overhead em comparação com protocolos mais leves como MQTT e CoAP.
- Menor eficiência em ambientes com restrições severas de recursos.



Protocolos de Comunicação na IoT

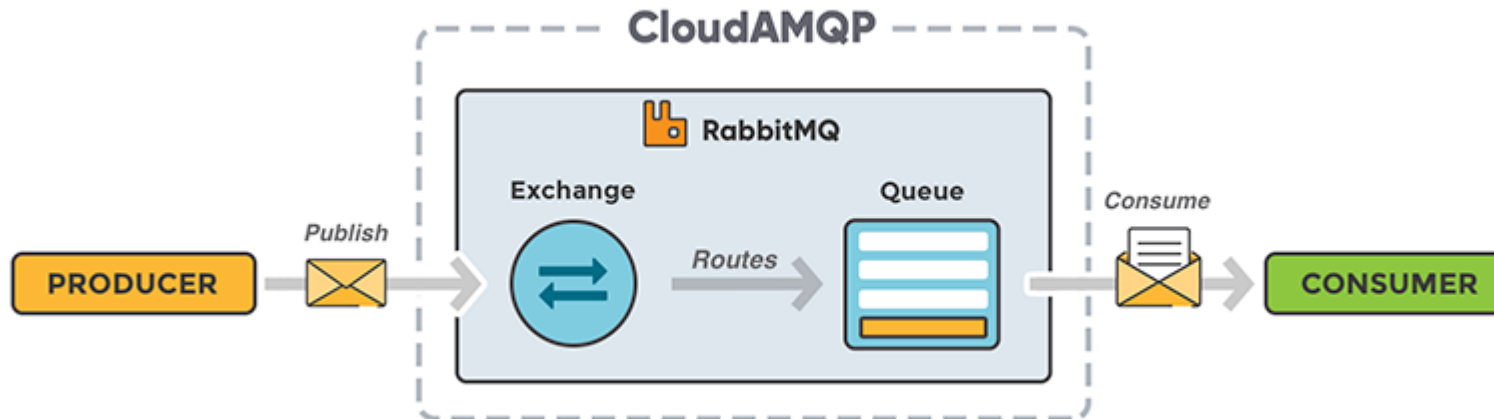
AMQP:

Vantagens:

- Flexibilidade na implementação de padrões avançados
- Suporte a cenários de mensageria assíncrona e distribuída.

Limitações:

- Pode ser mais complexo do que protocolos como MQTT para casos de uso simples.
- Maior consumo de recursos em comparação com protocolos mais leves.



Protocolos de Comunicação na IoT

Wi-fi:



Vantagens:

- Alta taxa de transferência de dados.
- Ampla disponibilidade e compatibilidade.

Limitações:

- Consumo mais elevado de energia em comparação com tecnologias de baixo consumo.
- Alcance limitado em comparação com tecnologias de longo alcance como LoRa.

Frameworks para Padronização na IoT

Diante da busca a padronização na intercomunicação dos dispositivos IoT em uma Cidade Inteligente, é necessário comparar os principais frameworks relacionados a IoT para se observar quais seus pontos fortes e suas limitações, principalmente, em relação a interoperabilidade.

Dentre eles, temos:

- Arduino
- Raspberry Pi
- IoTivity
- Node-RED
- AWS IoT Core
- Microsoft Azure IoT
- Google Cloud IoT
- Eclipse IoT

Frameworks para Padronização na IoT

Arduino e Raspberry Pi:

- Plataformas de Hardware: Essas não são plataformas de software, mas placas de desenvolvimento de hardware. Elas podem alcançar interoperabilidade por meio de:
- Protocolos padrão: Ambas suportam protocolos comuns de comunicação, como WiFi, Bluetooth e MQTT, permitindo a comunicação com outros dispositivos que utilizam os mesmos protocolos.
- Bibliotecas e Frameworks: Bibliotecas de código aberto como ArduinoJSON e bibliotecas MQTT para Raspberry Pi facilitam a troca de dados com diversos dispositivos.
- Limitações: Projetadas principalmente para o desenvolvimento de um único dispositivo, exigindo esforço adicional para cenários complexos de interoperabilidade.

Frameworks para Padronização na IoT

IoTivity:

- Framework de Código Aberto: Foca na interoperabilidade de dispositivo para dispositivo.
- Modelo de Dados Padronizado: Promove a descoberta de dispositivos e comunicação usando o protocolo CoAP leve e um modelo de dados comum, permitindo interação fácil entre dispositivos, independentemente do fornecedor ou plataforma.
- Adoção: Ainda está ganhando popularidade, exigindo dispositivos e bibliotecas compatíveis para implementação.

Frameworks para Padronização na IoT

AWS IoT Core, Microsoft Azure IoT, Google Cloud IoT:

- Plataformas de IoT Baseadas na Nuvem: Projetadas para gerenciar e conectar implantações de IoT em larga escala.
- APIs e Protocolos Proprietários: Conectam principalmente a dispositivos dentro de seus ecossistemas respectivos, limitando a interoperabilidade nativa com outras plataformas.
- Soluções de Interoperabilidade: Oferecem gateways e serviços de integração para conectar-se a dispositivos e plataformas externas, embora com aumento de complexidade e potencial bloqueio do fornecedor.

Frameworks para Padronização na IoT

Eclipse IoT:

- Ecosistema de Projetos de Código Aberto: Fornece diversas ferramentas e frameworks para diversas necessidades de IoT, incluindo soluções de interoperabilidade como Eclipse Kura e Eclipse Leshan.
- Protocolos e APIs Padronizados: Promove padrões e protocolos abertos como MQTT e LwM2M, facilitando a comunicação com diversas plataformas e dispositivos.
- Complexidade: Requer escolher e integrar ferramentas adequadas do ecossistema Eclipse IoT, com diferentes níveis de suporte à interoperabilidade.

Plataformas

Dessa forma, priorizando a interoperabilidade e outras características essenciais para uma cidade inteligente (tais como escalabilidade e flexibilidade), foram escolhidas as seguintes plataformas:

Plataformas - Comunicação forte de dispositivo para dispositivo

IoTivity: Possibilita interação perfeita entre dispositivos diversos, independentemente do fornecedor ou plataforma, sendo ideal para redes de sensores e aplicações distribuídas.



Plataformas - Gerenciamento baseado em nuvem e escalabilidade

Microsoft Azure IoT: Oferece capacidades robustas para implantações em larga escala, integra-se bem com os serviços da Microsoft e apresenta recursos de segurança sólidos.



Microsoft Azure
IoT Platform

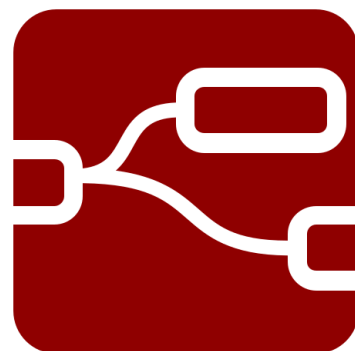
Plataformas - Flexibilidade e integração de código aberto

Eclipse IoT: Utiliza várias ferramentas como Kura e Leshan para interoperabilidade, suporta padrões abertos, permitindo a conexão com diferentes plataformas e ecossistemas.

Com base nas necessidades é possível combinar diferentes opções: IoTivity + Azure IoT: Aproveita a comunicação de dispositivo para dispositivo para processamento na borda e conecta perfeitamente à nuvem para gerenciamento centralizado.

Eclipse Kura + Google Cloud IoT: Utiliza o Eclipse Kura para interoperabilidade de código aberto e conecta-se ao Google Cloud IoT para monitoramento e análise em nuvem escaláveis.

Node-RED + AWS IoT Core: Utiliza o fluxo visual do Node-RED para processamento flexível de dados e conecta-se ao AWS IoT Core para gerenciamento de dispositivos e serviços em nuvem dentro do ecossistema da AWS.



Referências

- [1] D. L. Dantas, L. V. L. Filgueiras, and A. A. F. Brandão, "Portability and interoperability in IoT Platforms application layer portabilidade e interoperabilidade na camada de aplicação de plataformas de internet das coisas," in Anais, 2019.
- [2] S. Andrade and D. Luque. "Interoperabilidade de Sistemas aplicados às Cidades Inteligentes: Um Estudo de Mapeamento Sistemático", in Anais do X Workshop de Computação Aplicada em Governo Eletrônico, Niterói, 2022, pp. 97-108, doi: <https://doi.org/10.5753/wcge.2022.222970>.
- [3] D. V. A. Silveira et al., "Proposta de padronização de comunicação para dispositivos IoT," in Congresso Brasileiro de Automática-CBA, vol. 1, no. 1, 2019, doi: <https://doi.org/10.20906/CBA2022/511>.
- [4] Microsoft, Arquitetura do Connected Customer Service com o Hub IoT: <https://learn.microsoft.com/pt-br/dynamics365/customer-service/administer/cs-iot-connected-customer-service-architecture>

Obrigado Pela
Atenção!