

Sistemas do tipo "publicar-assinar":

1.

É um paradigma de comunicação onde os remetentes (publishers) enviam mensagens sem saber quem são os destinatários finais, e os destinatários (subscribers) recebem apenas as mensagens relevantes para eles, sem saber quem as enviou.

2.

Publisher (Editor): Envia mensagens para um tópico específico.

Broker (Intermediário): Recebe mensagens de publishers e encaminha para os subscribers adequados.

Subscriber (Assinante): Recebe mensagens de tópicos aos quais está inscrito.

3.

IoT (Internet das Coisas): Dispositivos IoT comunicam-se via MQTT para enviar dados para uma central de processamento.

Aplicações de mensagens em tempo real: Aplicações de chat, sistemas de notificações em tempo real, etc.

4.

Desacoplamento: Publishers e subscribers não precisam saber da existência um do outro.

Escalabilidade: Novos publishers e subscribers podem ser adicionados facilmente.

Flexibilidade: Os publishers não precisam saber quantos e quais subscribers existem, e vice-versa.

5.

Modelo de assinatura única: Cada mensagem é entregue a apenas um dos assinantes.

Modelo de assinatura múltipla: Cada mensagem é entregue a todos os assinantes.

Modelo de assinatura de tópico: Assinantes escolhem tópicos de interesse e recebem mensagens relacionadas a esses tópicos.

O MQTT se enquadra no modelo de assinatura de tópico, onde os assinantes se inscrevem em tópicos de seu interesse e recebem todas as mensagens publicadas nesses tópicos.

Microsserviços:

1.

Um sistema monolítico é uma aplicação em que todos os seus componentes estão interligados e empacotados juntos, geralmente como um único código de base e um único processo em execução.

2.

Uma aplicação microsserviço é composta por vários serviços independentes, cada um executando uma função específica e se comunicando através de protocolos como HTTP, gRPC ou, como mencionado, sistemas de mensageria publish-subscribe.

3.

Sim, se você criar vários serviços independentes, como uma WebAPI em Flask para expor informações e um sistema de mensageria publish-subscribe para comunicação assíncrona entre esses serviços, você estará seguindo uma abordagem microsserviços.