Understanding Pub-Sub and Request-Reply in Azure Service Bus

Introduction

Azure Service Bus is a cloud messaging service for queuing, publish-subscribe patterns, and complex integration patterns in cloud-native applications. It supports pub-sub and request-reply communication patterns.

Publish-Subscribe (Pub-Sub) Pattern

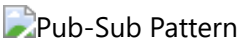
Overview

- **Definition**: A messaging paradigm where messages are published to a topic, and subscribers receive messages from topics they are subscribed to.
- **Azure Service Bus Implementation**: Uses topics and subscriptions.

Working Mechanism

1. **Publishing Messages**: Publishers send messages to a topic.
2. **Subscribing to Topics**: Subscribers create a subscription and can filter messages.
3. **Receiving Messages**: Subscribers receive messages from the topic.

Illustration


Pub-Sub Pattern

Implementation

Topics are there for one to many relationships, where the "one" is the topic and the "many" is the subscription to the topic. A possible implementation is to have application-wide topic to which all microservices can publish and then the microservices can consume this topic by having a generic subscription type.

A different implementation would be to have a topic per event and then let those topics be consumed in the various microservices that require the information. This would mean that every topic only has a single message type going through it.

Request-Reply Pattern

Overview

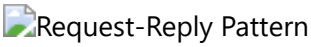
- **Definition**: A two-way communication pattern involving sending a message and receiving a reply.
- **Azure Service Bus Implementation**: Uses queues or a combination of queues and topics.

Working Mechanism

1. **Sending Requests**: A requestor sends a message to a queue or topic.
2. **Processing Requests**: A receiver processes the message and sends a reply.

3. **Receiving Replies**: The reply is sent back to the requestor.

Illustration


Request-Reply Pattern

Implementation

A possible way to set up request reply is to have a generic topic to which a request can be posted, but with each message, attach a session id that can then be used to send a reply to a specific queue.