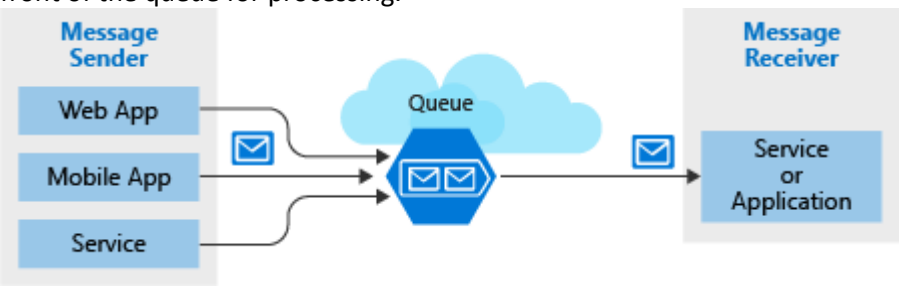
Storage queues and Service Bus queues

1. Introduction

 # Storage queues

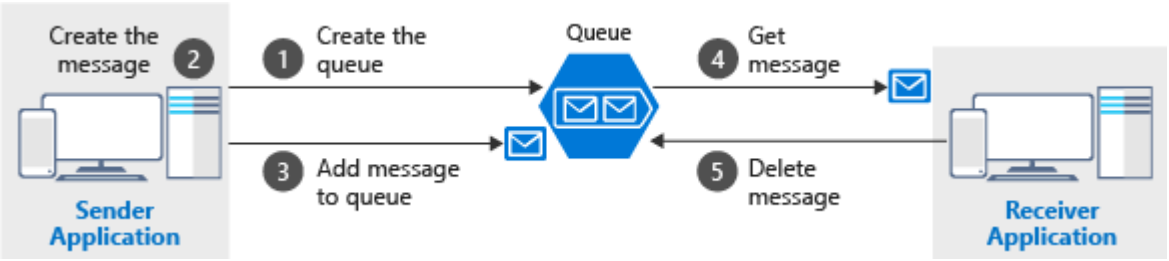
Azure Queue storage

• Azure Queue storage is an Azure service that implements cloud-based queues.
• Each queue maintains a list of messages.
• Application components access a queue using a REST API or an Azure-supplied client library.
• Typically, you will have one or more sender components and one or more receiver components.
• Sender components add messages to the queue. Receiver components retrieve messages from the front of the queue for processing.



Message

• A message in a queue is a byte array of up to 64 KB.
• Message contents are not interpreted at all by any Azure component.
• If you want to create a structured message, you could format the message content using XML or JSON.



 # Service Bus

• Azure Service Bus can exchange messages in three different ways: queues, topics, and relays.

Queue

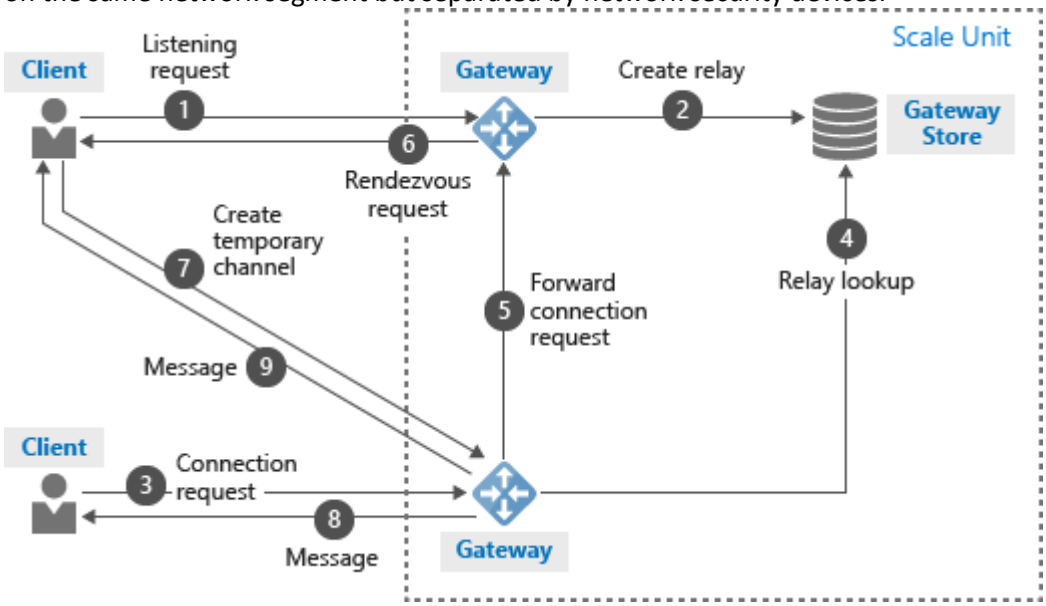
• A queue is a simple temporary storage location for messages.
• A sending component adds a message to the queue. A destination component picks up the message at the front of the queue.
• Under ordinary circumstances, each message is received by only one receiver.

Topic

• A topic is similar to a queue but can have multiple subscriptions.
• This means that multiple destination components can subscribe to a single topic, so each message is delivered to multiple receivers.

• Subscriptions can also filter the messages in the topic to receive only messages that are relevant. • Subscriptions provide the same decoupled communications as queues and respond to high demand in the same way.
• Use a topic if you want each message to be delivered to more than one destination component.
Note
• Topics are not supported in the Basic pricing tier.

Relay
• A relay is an object that performs synchronous, two-way communication between applications.
• Unlike queues and topics, it is not a temporary storage location for messages.
• Instead, it provides bidirectional, unbuffered connections across network boundaries such as firewalls.
• Use a relay when you want direct communications between components as if they were located on the same network segment but separated by network security devices.



Note
• An Azure Service Bus queue message must be larger than 64 KB but smaller than 256 KB

2. Advantages

 # Storage queues
Advantages of storage queues:
• Supports unlimited queue size (versus 80-GB limit for Service Bus queues)
• Maintains a log of all messages

 # Service Bus
Key advantages of Service Bus queues include:
• Supports larger messages sizes of 256 KB (standard tier) or 1MB (premium tier) per message versus 64 KB

• Supports both at-least-once and at-most-once delivery - choose between a very small chance that a message is lost or a very small chance it is handled twice
• Guarantees first-in-first-out (FIFO) order - messages are handled in the same order they are added (although FIFO is the normal operation of a queue, it is not guaranteed for every message)
• Can group multiple messages into a transaction - if one message in the transaction fails to be delivered, all messages in the transaction will not be delivered
• Supports role-based security
• Does not require destination components to continuously poll the queue

3. Comparison

 # Storage queues
Consider using Storage queues when:
• Your application must store over 80 GB of messages in a queue.
• Your application wants to track progress for processing a message inside of the queue. This is useful if the worker processing a message crashes. A subsequent worker can then use that information to continue from where the prior worker left off.
• You require server side logs of all of the transactions executed against your queues.

 # Service Bus
Consider using Service Bus queues when:
• Your solution must be able to receive messages without having to poll the queue. With Service Bus, this can be achieved through the use of the long-polling receive operation using the TCP-based protocols that Service Bus supports.
• Your solution requires the queue to provide a guaranteed first-in-first-out (FIFO) ordered delivery.
• Your application handles messages that can exceed 64 KB but will not likely approach the 256 KB limit.
• Your queue size will not grow larger than 80 GB.