Miroslav Holec
Software & Cloud Architect
Microsoft MVP: Microsoft Azure
MCSD, MCSA, MTA

miroslavholec.cz
@miroslavholec

Azure Service Bus

HAVIT SOFTWARE DEVELOPMENT

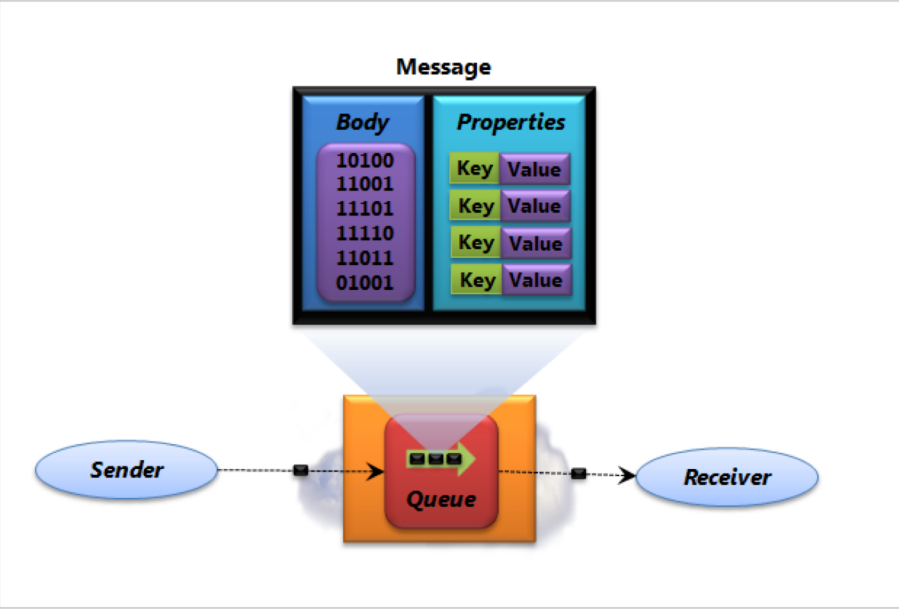
odkaz.me/cuni

Azure Service Bus

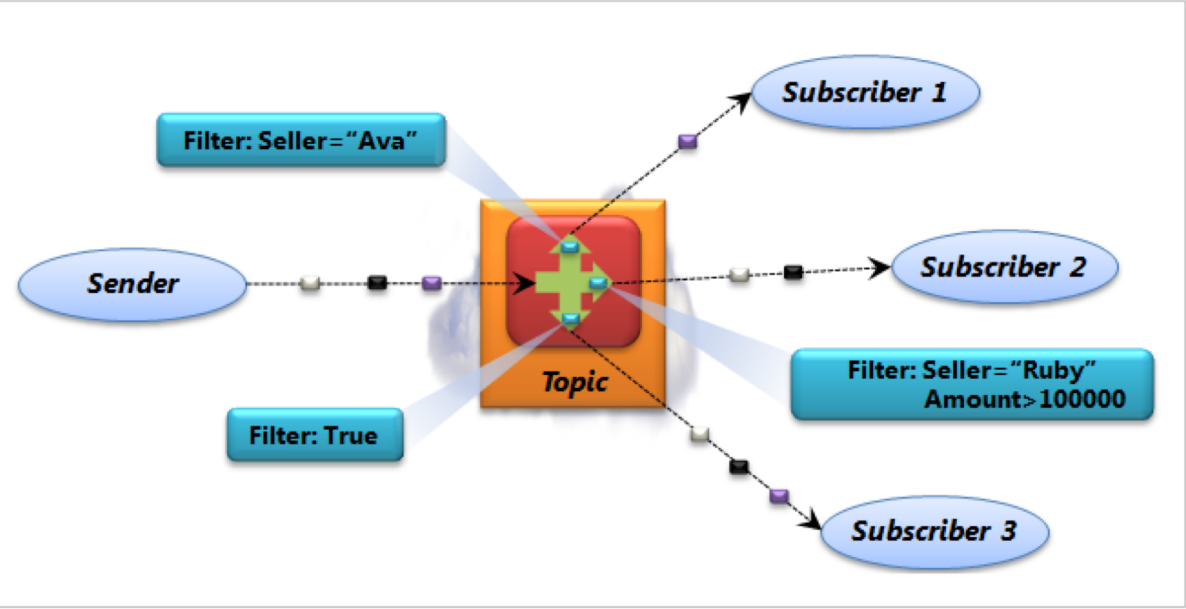
- **messaging** as a service in Microsoft Azure
- supports two features: **queues** and **topics**

not available in basic pricing tier

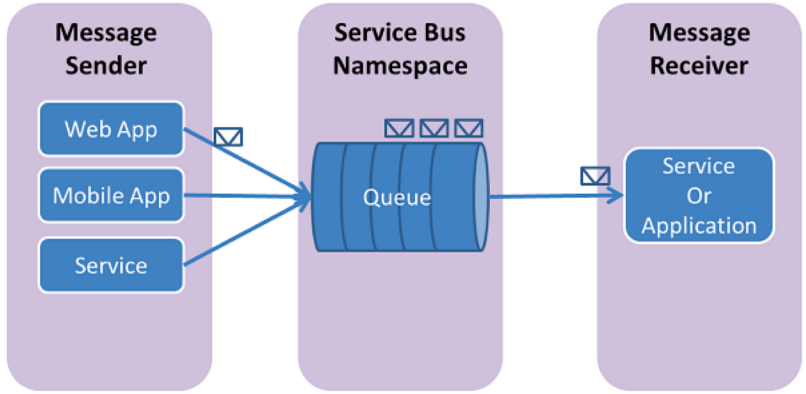
Service Bus Queues



Service Bus Topics



Queues

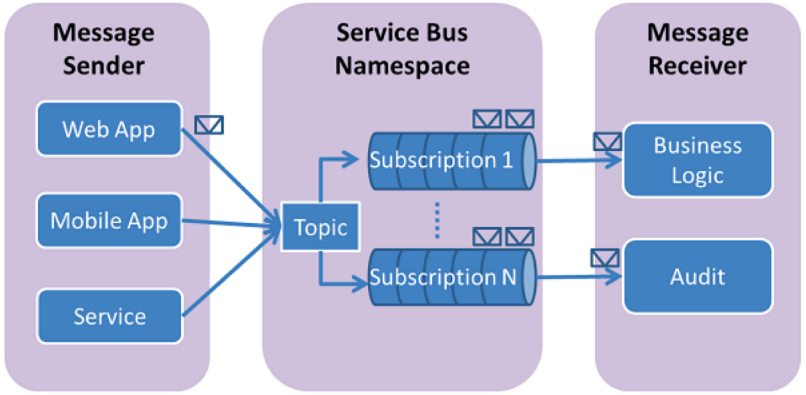


- messages are stored in **broker message queue**
- message is delivered **at least to one** (and optionally at most to one) **receiver**
- message has two parts: **properties** (key-value) and **payload** (binary, text, XML)
- receiver reads in **ReceiveAndDelete** or **PeekLock** (default) mode
- message is **locked during processing** and waits for Complete or Abandon op.
- there is a **FIFO mode** for Service Bus Queues (session ID needed / partitioning switch off)

Where is Azure Storage Queue stronger than Service Bus Queues?

- message queue is greater than 80 GB
- server side logging for all operations + you can see queues in storage explorer

Topics



- messages are sent to **topic** (same way as with queues)
- every topic can have up to 2000 **subscriptions**
- every client (app, service) is receiving messages from subscription
- subscription can have various **filters**
- ReceiveAndDelete / PeekLock behavior as with Queues
- difference: one message from topic can be **received by multiple subscriptions**

Another alternative to Service Bus Topics may be Redis (https://redis.io)

Get started with Queues

1. Create **Service Bus** service (namespace) in Azure
2. Create new service bus **Queue**
3. Download a compile demo application from GitHub
4. Code!
    1. Set your connection string
    2. Get NuGet package
    3. Create QueueClient
    4. Send or Receive messages using the QueueClient

Get started with Topics

1. Navigate to your **Service Bus** namespace in Azure
2. Create new service bus **Topic**
3. Create new service bus **Subscription** under your Topic
4. Code!
    1. Update connection strings to Service Bus
    2. Change Sender app to Publisher app using **ITopicClient**
    3. Change Receiver app to Subscriber app using **ISubscriptionClient**