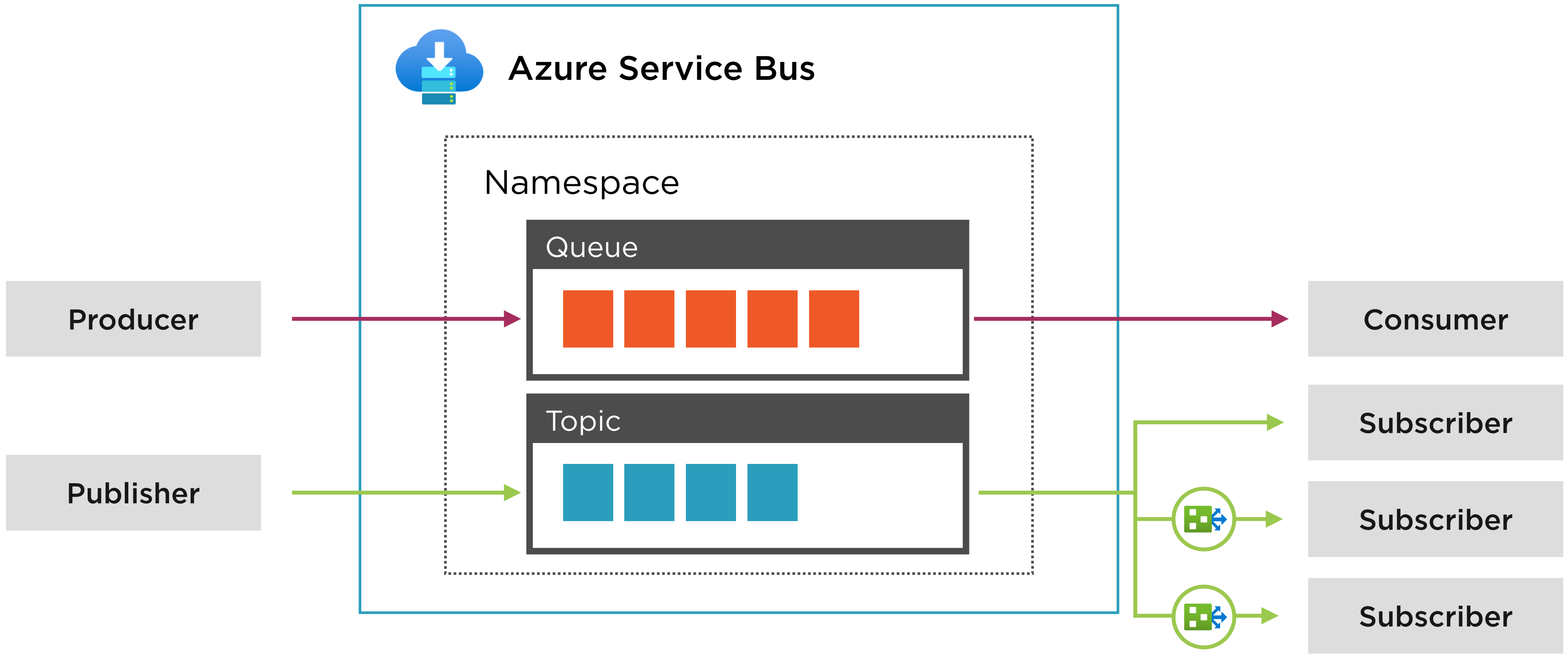
Azure Service Bus

David Tucker
TECHNICAL ARCHITECT & CTO CONSULTANT
@_davidtucker_   davidtucker.net

Azure Service Bus

Fully-managed enterprise message broker service that enables multiple modes of messaging with integrations for common messaging systems including Java Message Service (JMS).

Organization of Azure Service Bus



Azure Service Bus

Supports both HTTP/HTTPS and AMQP protocols

Includes messaging for both queues and topics

Supports three different performance tiers: basic, standard, and premium

Supports advanced configurability:

- Ordering
- Batching
- DLQ
- Duplicate detection

Basic Tier

The basic tier of Azure Service Bus only supports queues (and not topics). To fully utilize the functionality of this service, it is recommended to use the Standard or Premium tier.

Comparing Service Tiers

Standard Tier	Premium Tier
Pricing is “pay as you go”	Pricing is fixed based on messaging units
Throughput is variable and has variable latency	Throughput is fixed based on messaging units
Utilizes shared resources	Utilizes dedicated resources
Provides automatic scaling	Requires configuration of scaling rules
Supports messages up to 256 KB	Supports messages up to 1 MB
Does not support geo-disaster recovery or availability zones	Supports geo-disaster recovery and availability zones

Service Bus URL Structure

$$\text{https://}\underbrace{\text{pluralsight.}}_{\text{namespace}}\text{servicebus.windows.net/}\underbrace{\text{testqueue}}_{\text{queue or topic name}}$$

Message Ordering

Azure Service Bus supports message FIFO (first in first out) **ordering by leveraging sessions. This is supported in queues and topics, but must be enabled on the queue or topic.**

Scaling Azure Service Bus

Standard tier namespaces support partitioning of queues and topics

Partitioning is not supported for premium tier namespaces

Partitioning enables separate messaging stores and brokers for a single entity

Partitioned queues and topics can use a partition key to determine the partition

Without a partition key, a round-robin algorithm is leveraged by Azure

Transactions on a partitioned entity must use a partition key

Dead-letter Queue (DLQ)

Azure Service Bus supports a DLQ for a queue or topic. This enables you to capture messages that were not processed during their lifetime, and act accordingly with those messages.

Selecting a Messaging Solution

Azure Queue Solutions

Azure Queue Storage

Azure Service Bus Queue

Azure Queue Storage Use Cases

Total storage for queue needs to be over 80 GB

Logs needed for all transactions executed against queue

Need to track progress of message processing

Azure Service Bus Use Cases

Need support for receiving messages without polling (with AMQP 1.0)

There is a need to guarantee message processing order (FIFO)

There is a need to detect duplicate messages

You need to support messages up to 256 KB

You may need to support topic based notifications (one to many)

You need to support publishing and consuming in batches

Creating an Azure Service Bus Queue

```
# create a queue
az servicebus queue create --namespace-name pluralsight
--name testqueue --resource-group pluralsight

# delete a queue
az servicebus queue delete --namespace-name pluralsight
--name testqueue
```

Interacting with Service Bus Queues using the CLI

Azure CLI

Demo

Creating an Azure Service Bus namespace

Utilizing the portal to create a queue

Leveraging the portal to send a message to a queue

Using a Service Bus Queue with the SDK

Demo

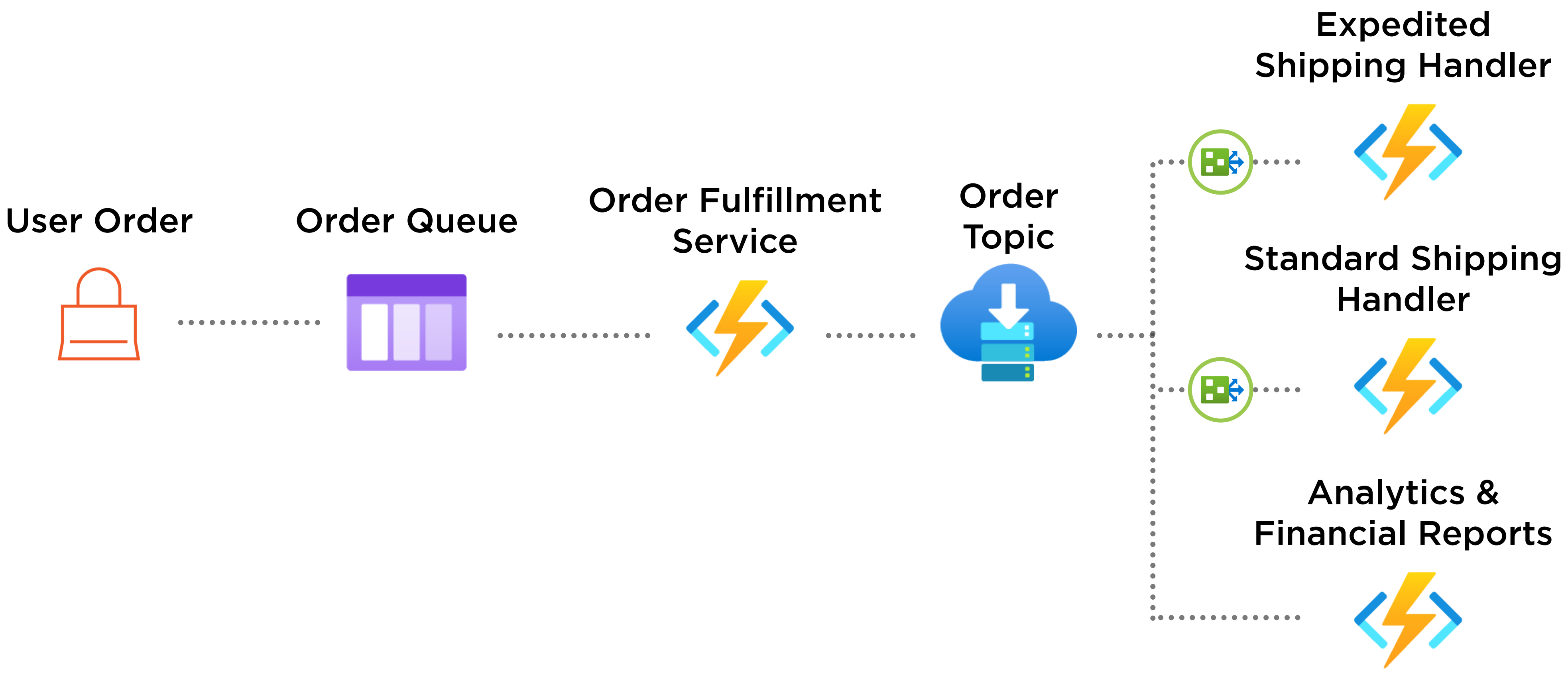
Creating a shared access policy for an Azure Service Bus namespace

Utilizing the SDK to produce messages for a queue

Receiving messages from a queue using the SDK

Utilizing Azure Service Bus Topics

Utilizing a Message-based Architecture


User Order
Order Queue
Order Fulfillment Service
Order Topic
Expedited Shipping Handler
Standard Shipping Handler
Analytics & Financial Reports

Azure Service Bus Topics

Enables a one-to-many relationship between messages and consumers

A consumer creates a subscription to a topic

Subscriptions act as dedicated queues for a subscriber with configuration options

Topic filters can be specified as:

- Boolean filters
- SQL filters
- Correlation filters

Creating an Azure Service Bus Topic

```
# create a topic
az servicebus topic create --namespace-name pluralsight
--name testtopic --resource-group pluralsight

# delete a topic
az servicebus topic delete --namespace-name pluralsight
--name testtopic

# create a subscription
az servicebus topic subscription create --namespace-name pluralsight
--name testsub --topic-name testtopic
```

Interacting with Service Bus Topics using the CLI

Azure CLI

Demo

Creating an Azure Service Bus topic

Creating an Azure Service Bus topic subscription

Sending and receiving messages for a topic in the portal

Using a Service Bus Topic with the SDK

Demo

Publishing messages to a Service Bus topic using the SDK

Receiving messages from a topic using the SDK

Configuring filters for a Service Bus topic