Microsoft Azure Service Bus In-Depth

Understanding the Azure Service Bus

Alan Smith

Active Solution

youtube.com/user/CloudCasts

Overview

Microsoft Azure Messaging Services

Asynchronous Messaging Scenarios

Microsoft Azure Service Bus

The Microsoft Azure Service Bus SDK

Demo: Simple Service Bus Messaging

Demo: Creating a Chat Applicartion with Publish-subscribe Messaging

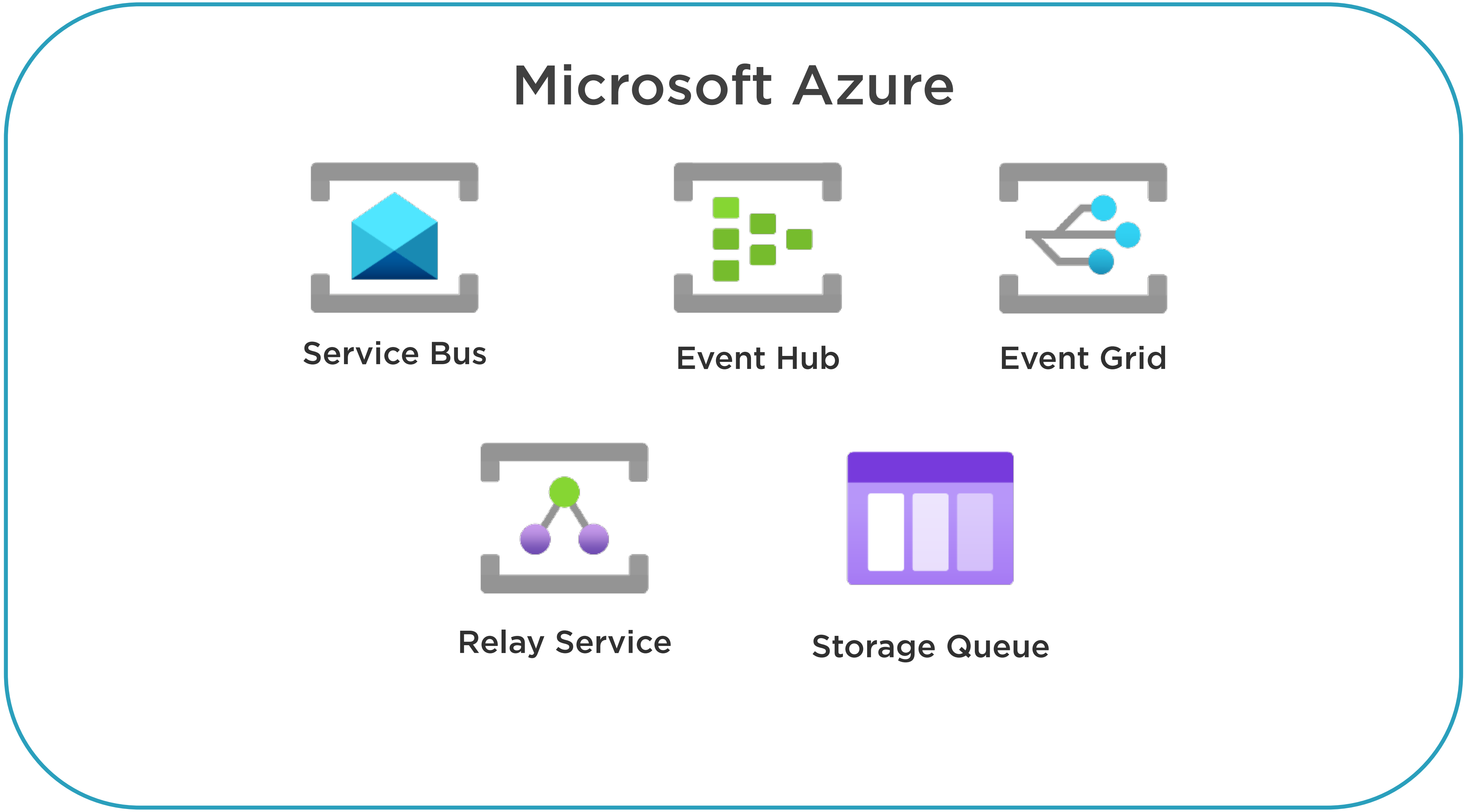
Overview

Service Bus Usage Scenarios

Service Bus Tiers & Pricing

Microsoft Azure Messaging Services

Microsoft Azure Messaging Services



Microsoft Azure

Service Bus

Event Hub

Event Grid

Relay Service

Storage Queue

Service Bus

Durable brokered messaging

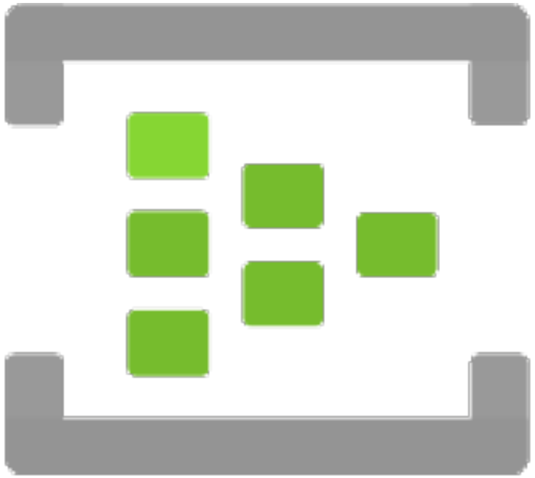
Point-to-point messaging

Publish subscribe messaging

Enterprise messaging functionality

Cost-efficient

Event Hub

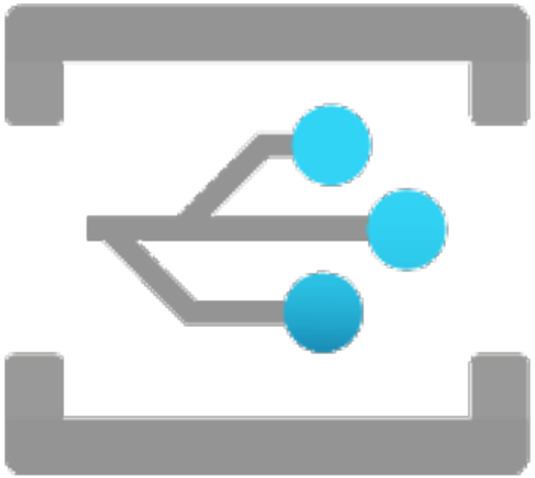


Large-scale telemetry ingestion

Buffered storage

Massively scalable

Event Grid



HTTP event routing and delivery

Near real-time notifications

Supported by many Azure services

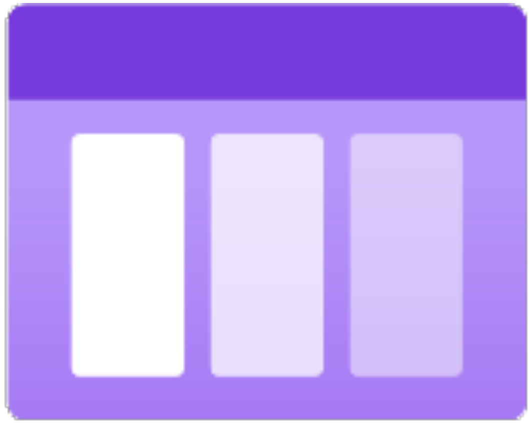
Relay Service

Securely exposes on-premises services

Endpoint “in the cloud”

Relays request and response calls

Storage Queues



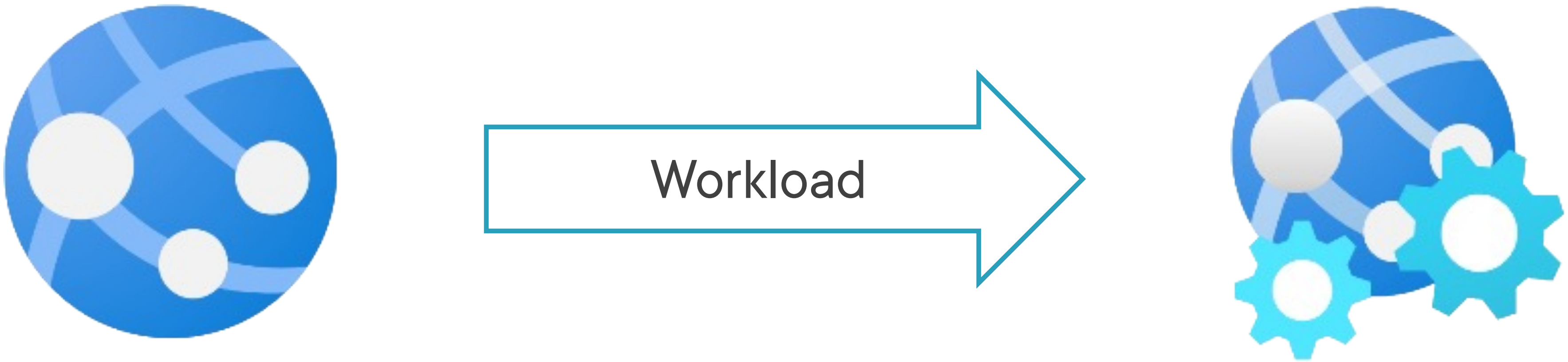
Simple point-to-point messaging

Very cost-effective

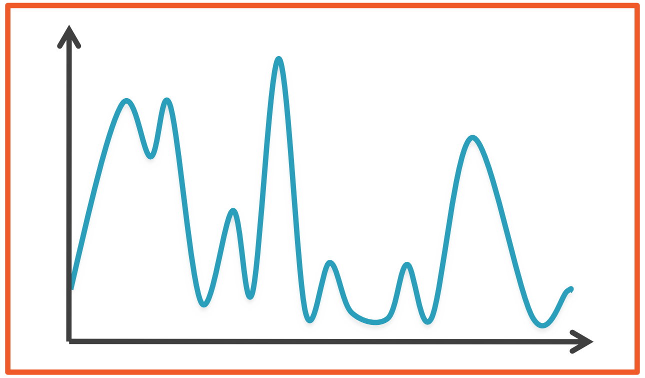
Limited functionality

Asynchronous Messaging Scenarios

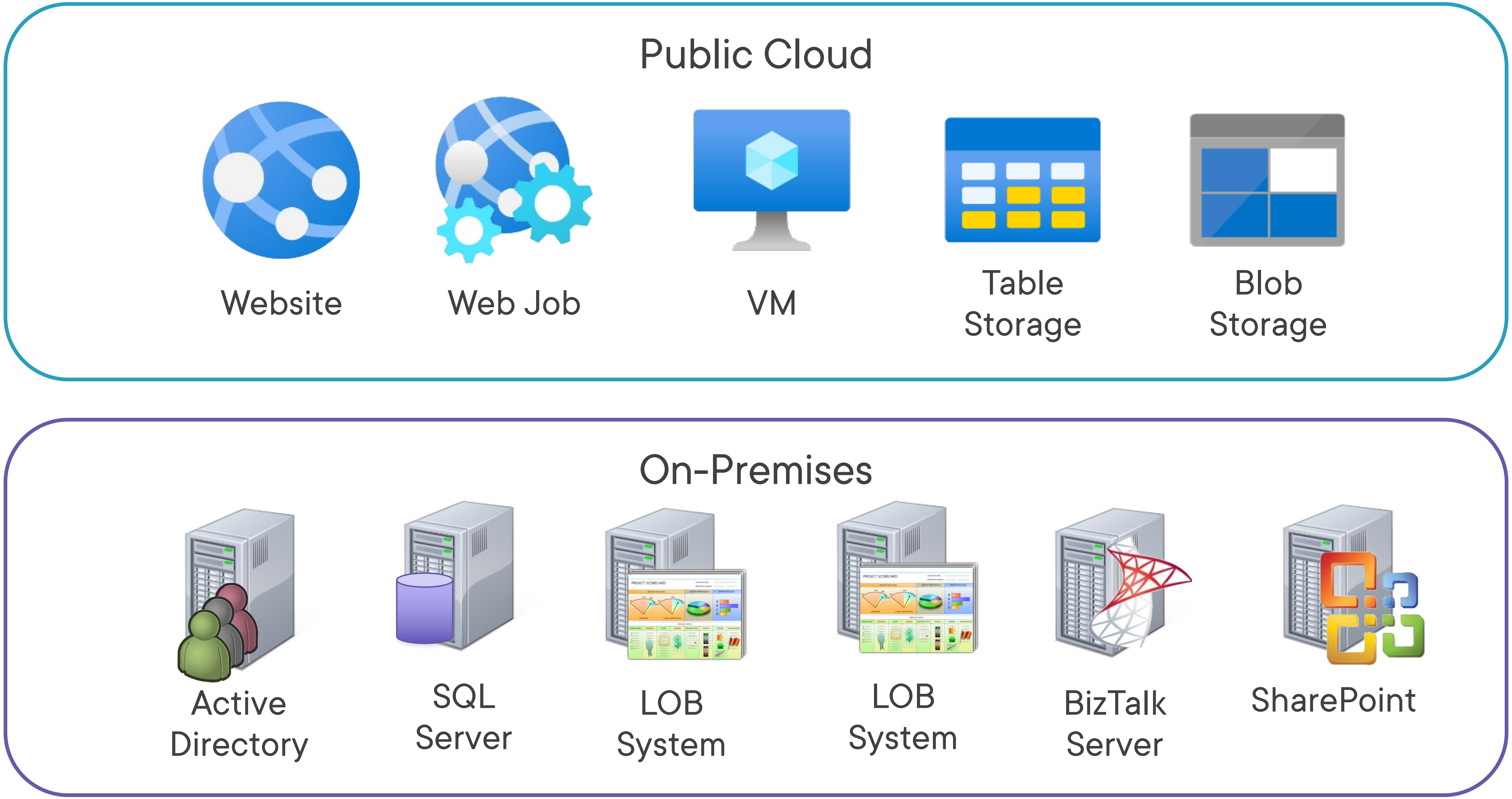
Connectivity Challenges – Asynchronous Processing



Workload



Connectivity Challenges – Hybrid Systems



Public Cloud

Website

Web Job

VM

Table Storage

Blob Storage

On-Premises

Active Directory

SQL Server

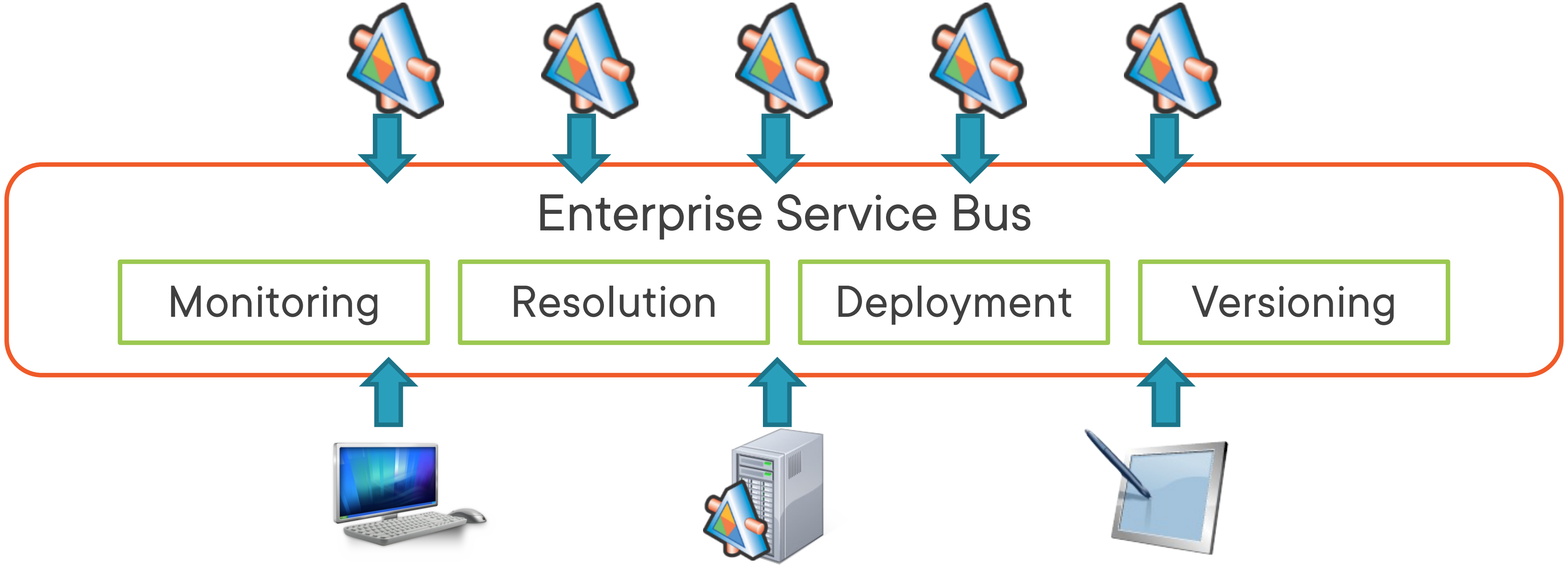
LOB System

LOB System

BizTalk Server

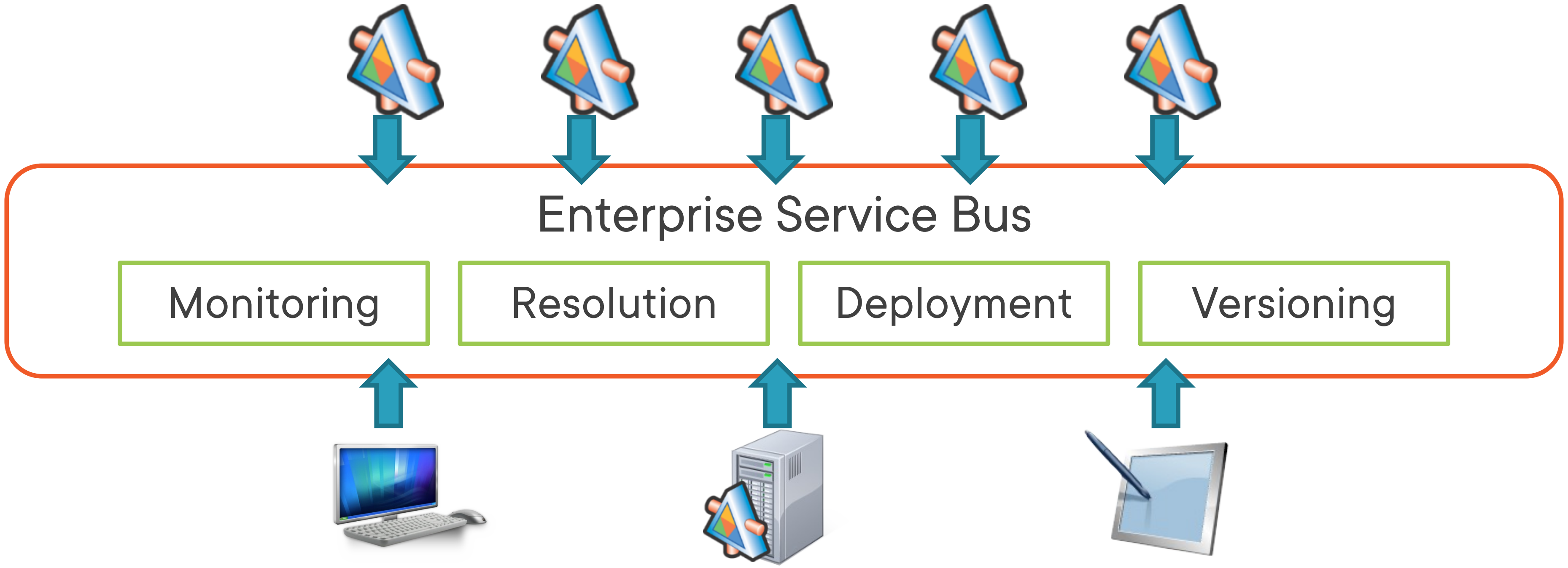
SharePoint

Enterprise Service Bus



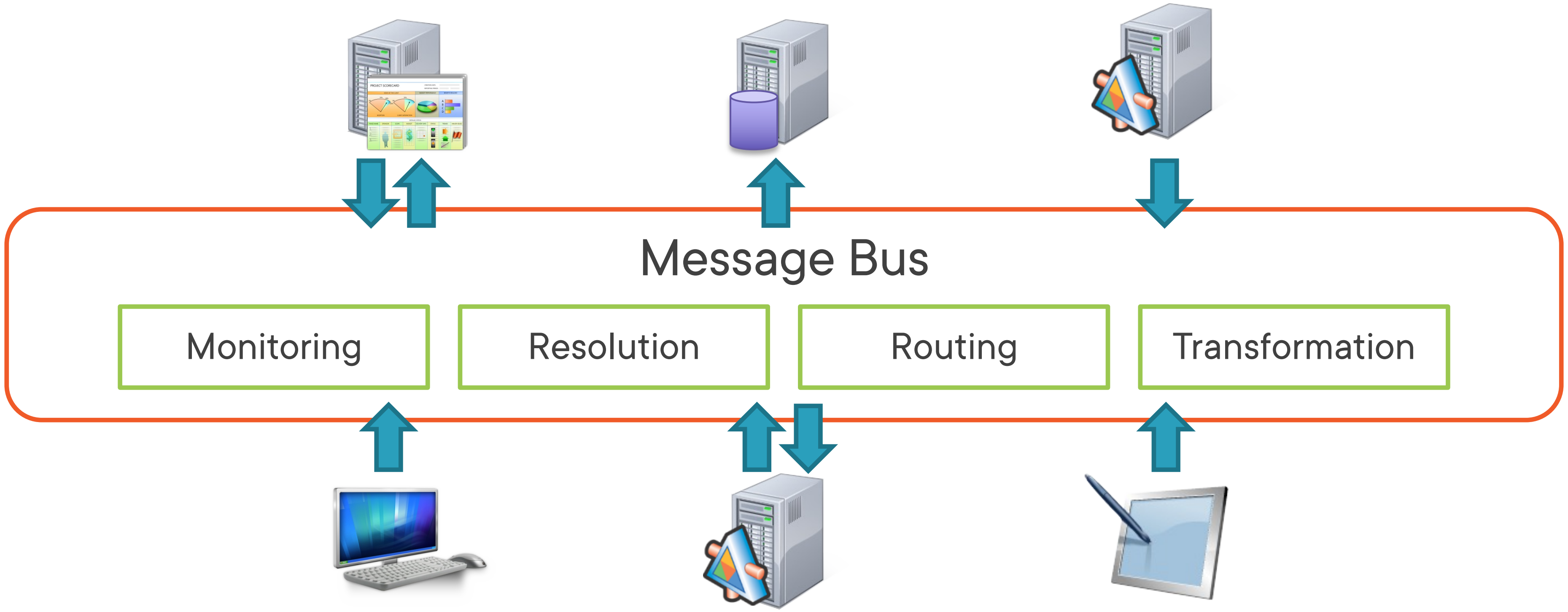
Enterprise Service Bus

"An enterprise service bus (ESB) is a software architecture model used for designing and implementing the interaction and communication between mutually interacting software applications in Service Oriented Architecture."
- Wikipedia



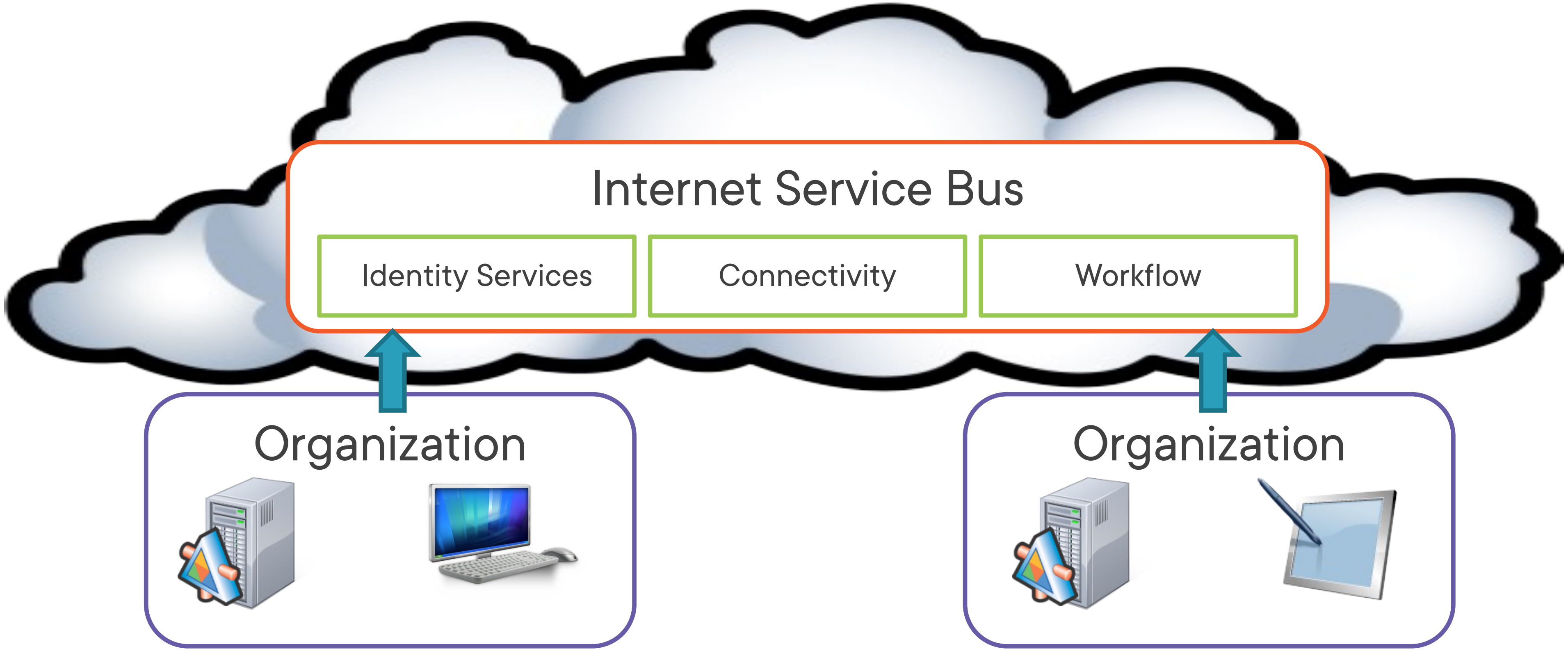
Message Bus

"A Message Bus is a combination of a common data model, a common command set, and a messaging infrastructure to allow different systems to communicate through a shared set of interfaces."
- Enterprise Integration Patterns



Internet Service Bus

“The ISB links devices to each others, devices to local servers, Web sites to Web sites, and ESBs to ESBs, and is itself an ESB. The ISB is a platform for “do-it-yourself” composite applications and business processes.
The ISB is also an example of Software as a Service (SaaS).”
- The Architecture Journal, October 2007

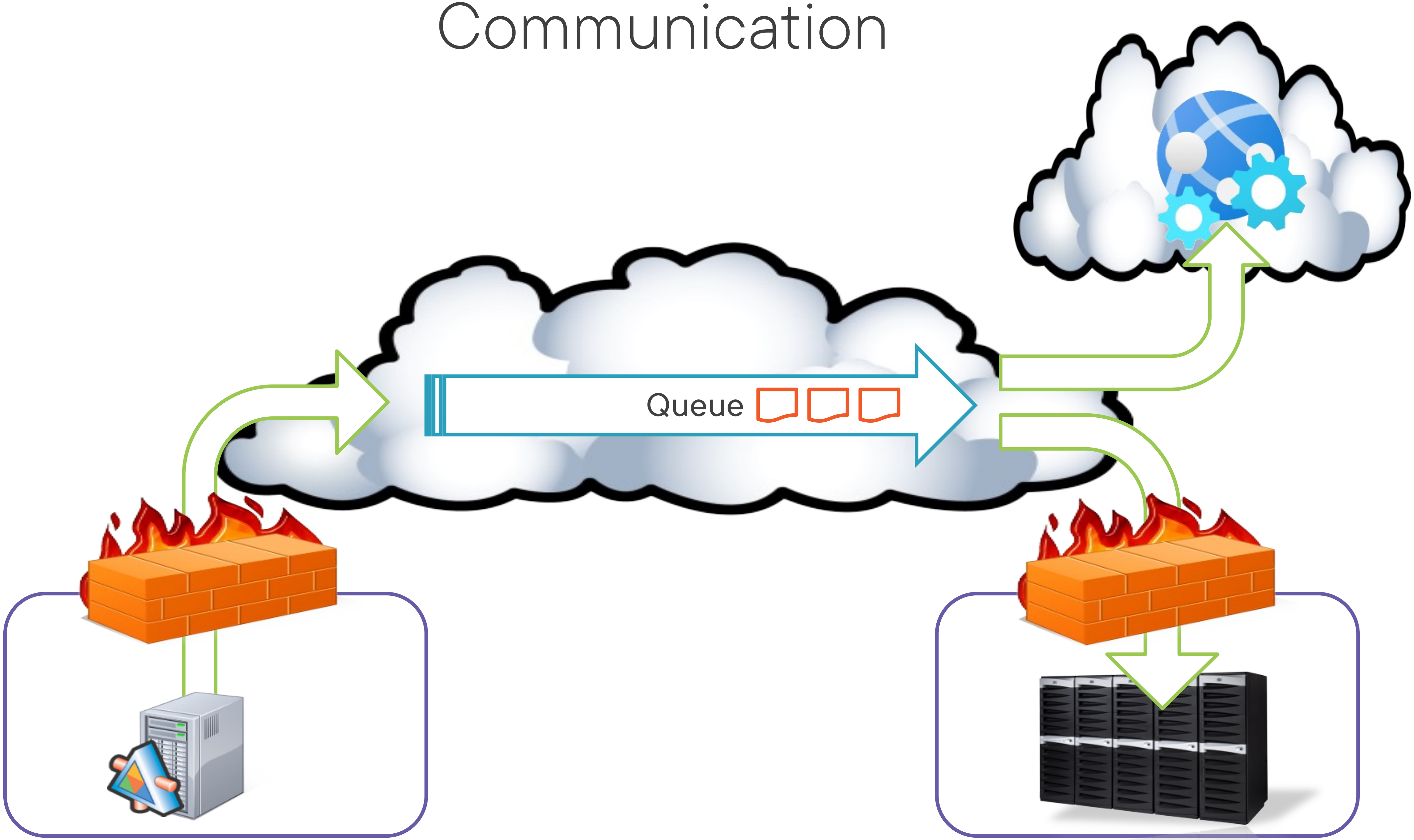


Extending to the Cloud

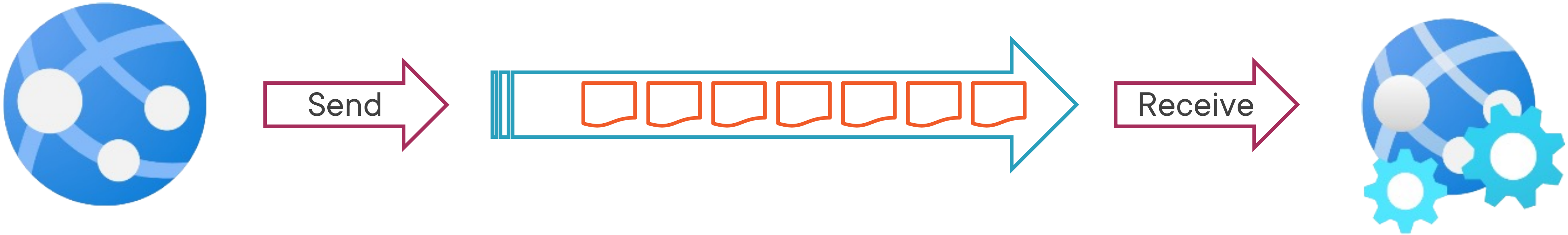
> “Leveraging the capabilities provided by cloud-based platforms to enhance or augment on-premise systems.”
> – Alan Smith

- Low entry cost
- Low risk
- Easy roll-back
- Enhance reach of existing systems

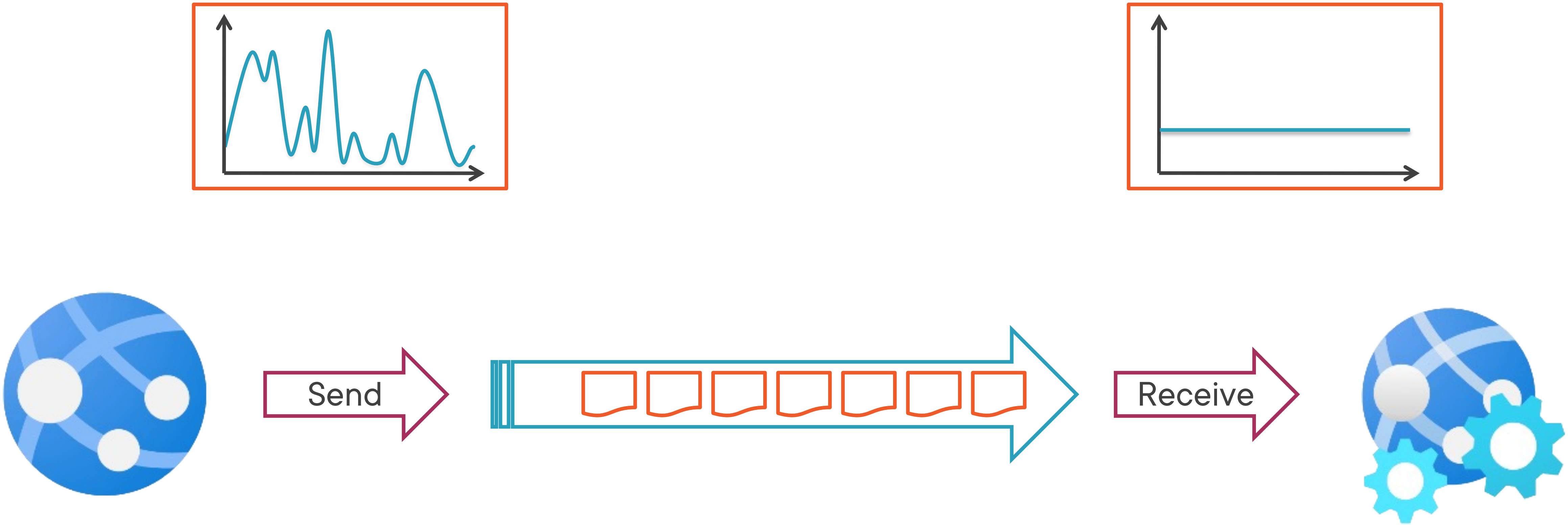
Communication



Load Leveling



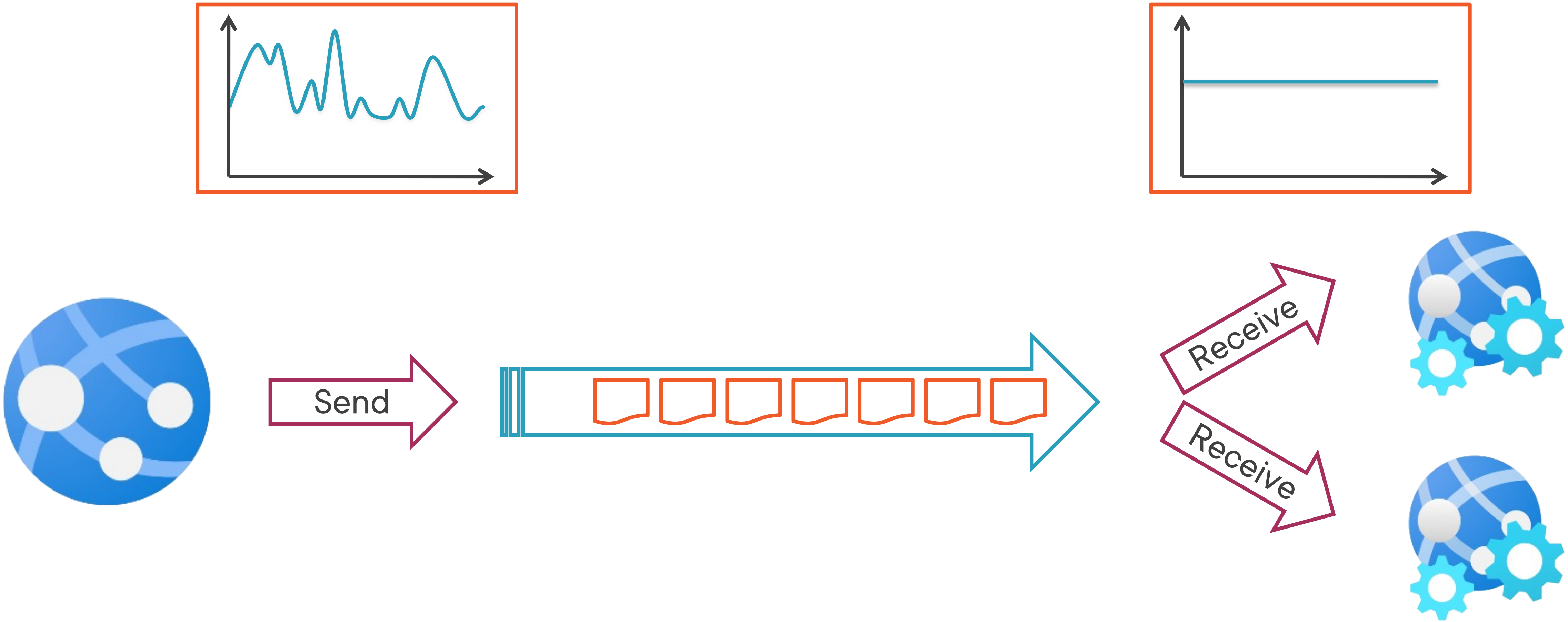
Load Leveling



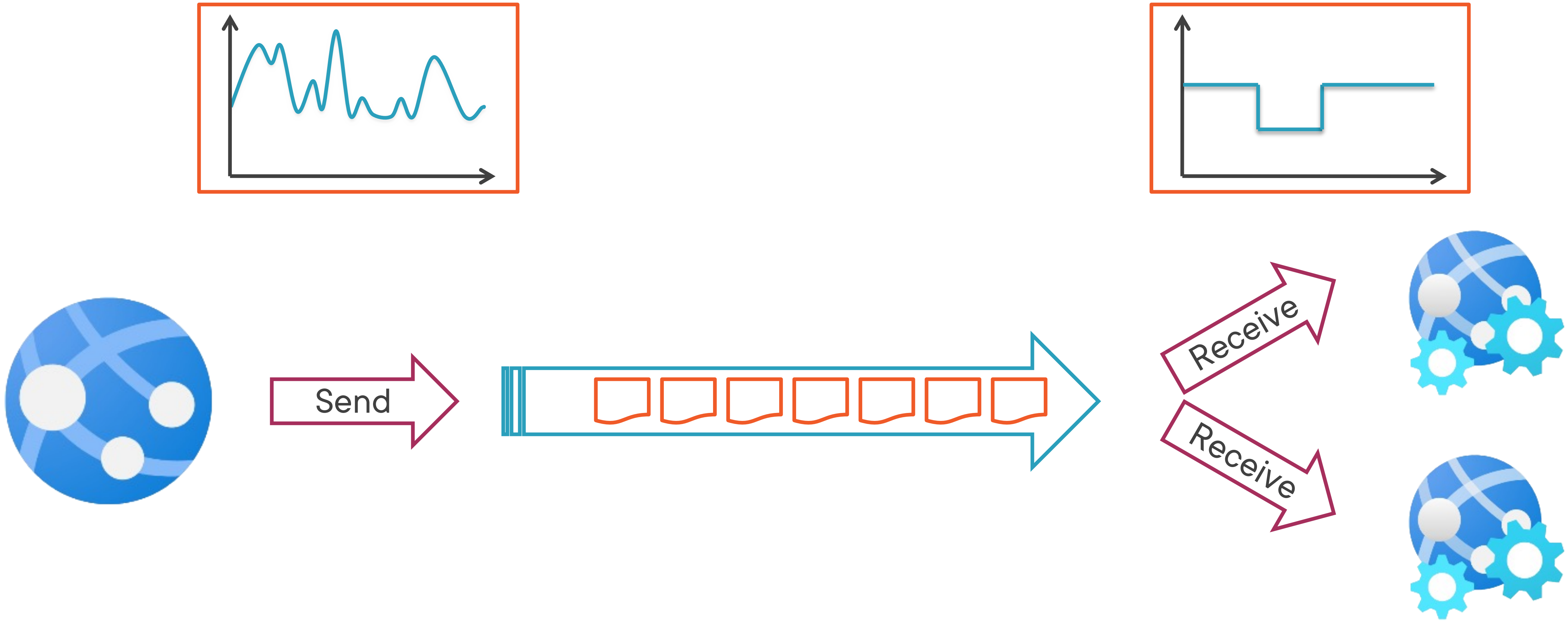
Send

Receive

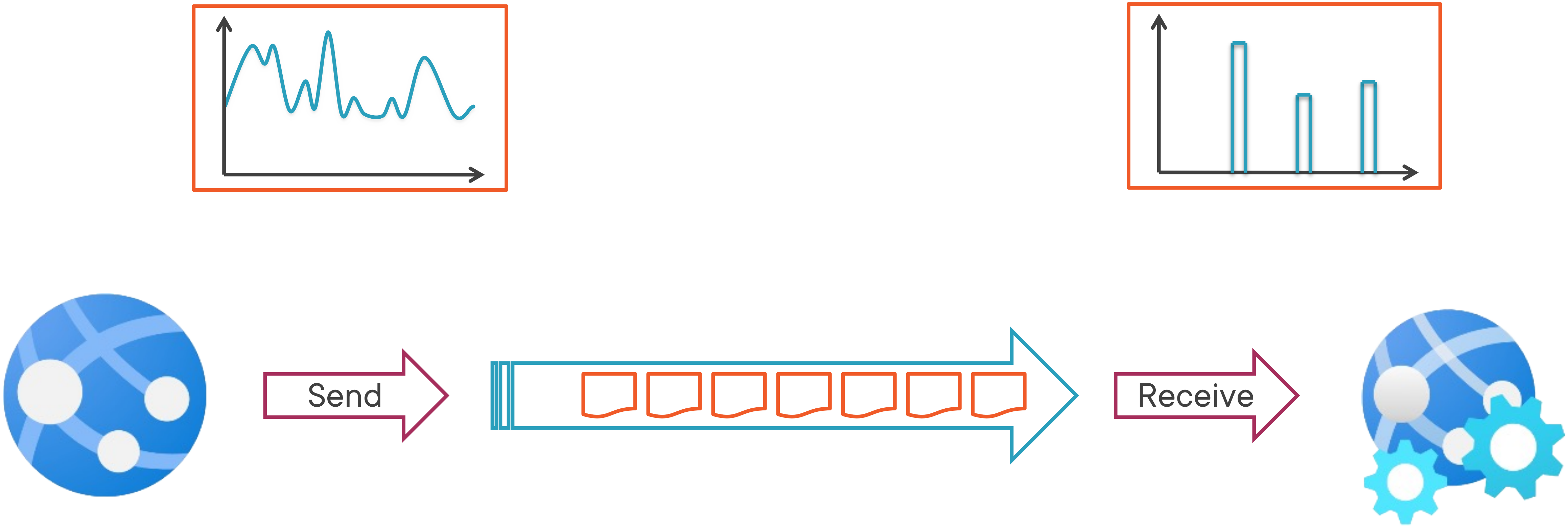
Load Balancing



High Availability

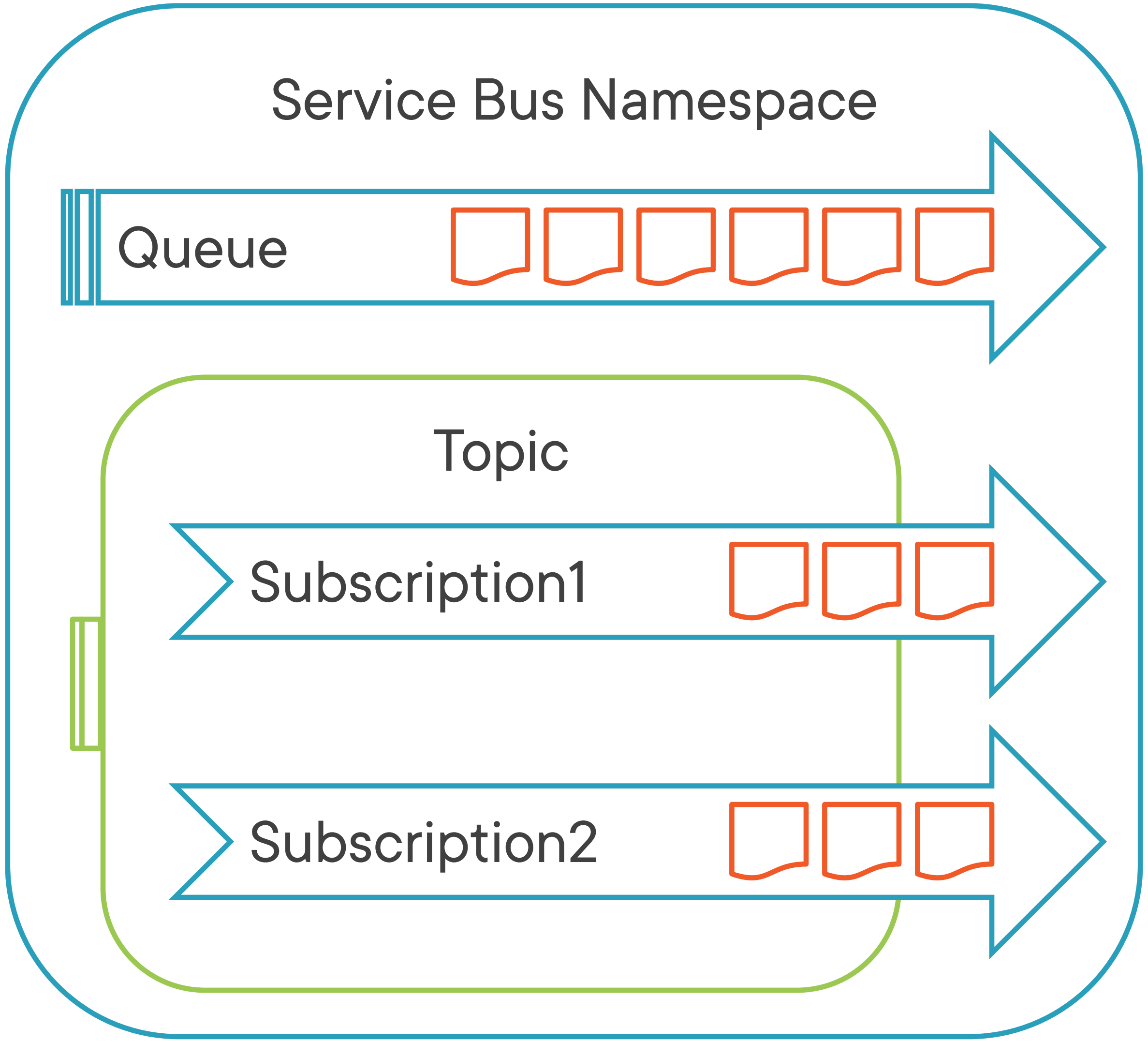


Temporal Decoupling



Microsoft Azure Service Bus

Messaging Entities



Namespace

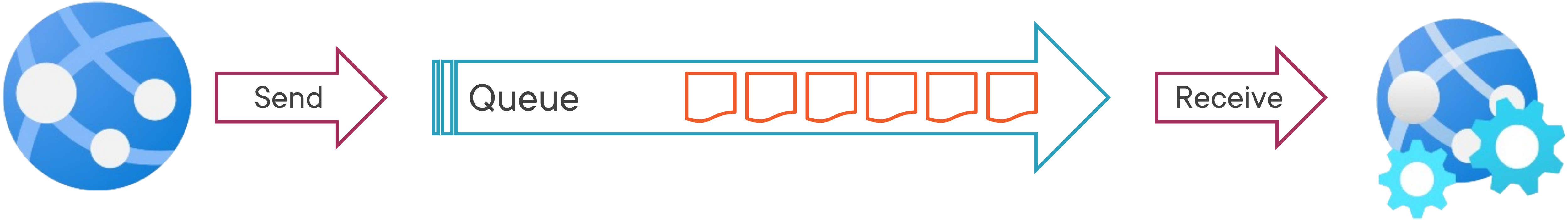
Queue

Topic

Subscription

Message

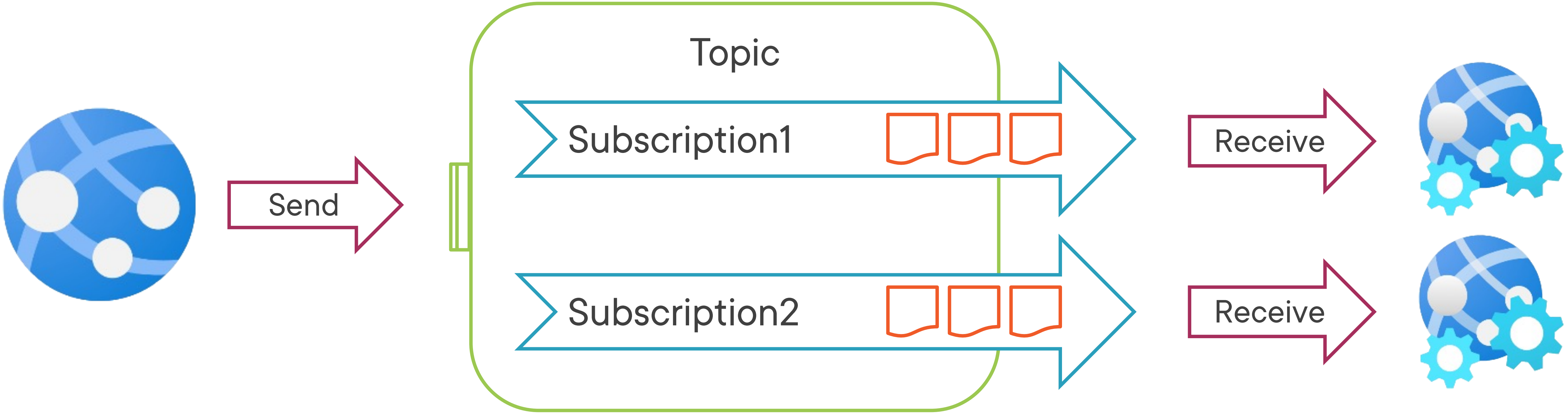
Queues



Point-to-point messaging

First-in first-out (FIFO) processing

Topics and Subscriptions



Publish-subscribe messaging

Messages are sent to topics

Messages are received from subscriptions

Filters can determine message subscription

Enterprise Messaging Capabilities

Capability	Description
Communication	How can applications in different environments communicate effectively and reliably with one another?
Security	How can the confidentiality and integrity of messages be maintained?
Reliable Delivery	How can the sending application be sure that the receiving application will receive all transmitted messages?
Low-Latency	How can the transmission and processing time of messages be kept as low as possible?
Availability	What level of uptime will a messaging system provide?
Scalability	How easily can a messaging system be upgraded to handle an increase in the message processing load?

Service Bus Messaging Features

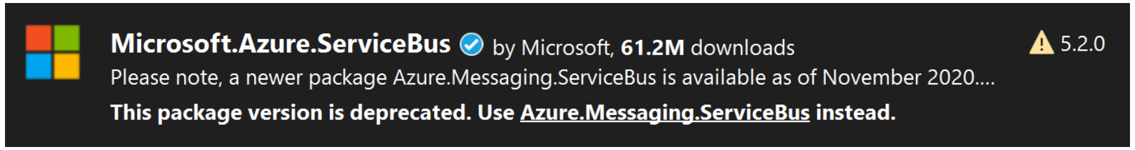
Feature	Description
Publish-subscribe	Messages can be broadcast to multiple receivers based on routing rules in the messaging entities.
Dead-lettering	Invalid or poison messages can be moved to a dead-letter queue.
Message Sessions	Related messages can be grouped together in sessions and processed together.
Request-response Correlation	Response messages can be correlated with the appropriate request messages to allow for asynchronous two-way communication.
Message Deferral	Messages can be preserved on a messaging entity and retrieved later for processing.
Scheduled Enqueue Time	Messages can be sent to a messaging entity and then enqueued at a specified time.
Duplicate Detection	Duplicate messages can be ignored by a messaging entity.
Message Expiration	Messages can be configured to expire after a specified duration.

Service Bus Protocols

Protocol	Usage
AMQP	Open messaging protocol. Default protocol used by Service Bus SDK. Supported by many applications and libraries.
HTTP	Provides communication where firewalls may limit connectivity on other protocols. Cross platform compatibility with many clients.

The Azure Service Bus SDK

Azure Service Bus NuGet Packages



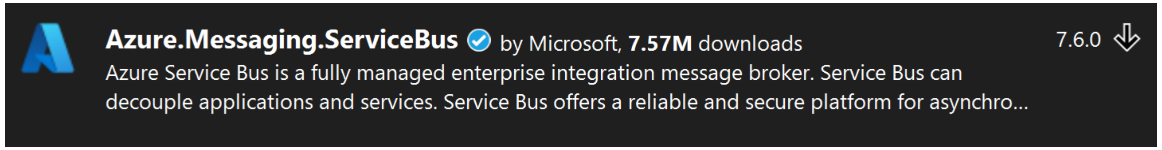
Microsoft.Azure.ServiceBus ✓ by Microsoft, **61.2M** downloads ⚠ 5.2.0

Please note, a newer package Azure.Messaging.ServiceBus is available as of November 2020....

This package version is deprecated. Use Azure.Messaging.ServiceBus instead.

Microsoft.Azure.ServiceBus

- Legacy Service Bus SDK
- Deprecated
- Not recommended for new projects
- Upgrading existing projects to new SDK requires significant code changes
- No compelling need to upgrade existing projects*



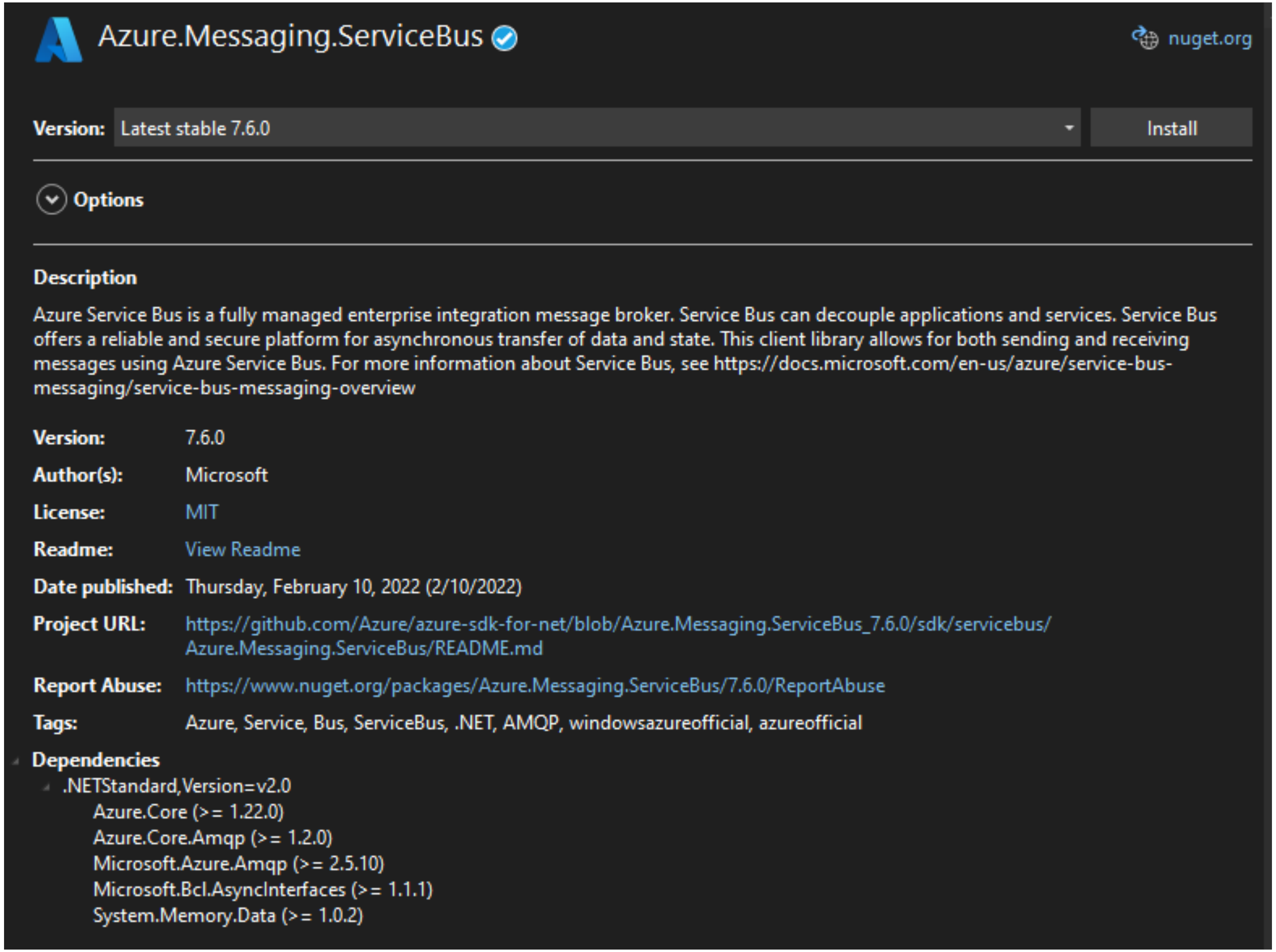
Azure.Messaging.ServiceBus ✓ by Microsoft, **7.57M** downloads 7.6.0

Azure Service Bus is a fully managed enterprise integration message broker. Service Bus can decouple applications and services. Service Bus offers a reliable and secure platform for asynchro...

Azure.Messaging.ServiceBus

- Latest Service Bus SDK
- Supports C# 8.0
- Improved message serialization
- Message serialization compatibility with legacy SDK
- Breaking code changes from legacy SDK

* In the author's opinion



Version: Latest stable 7.6.0

Description

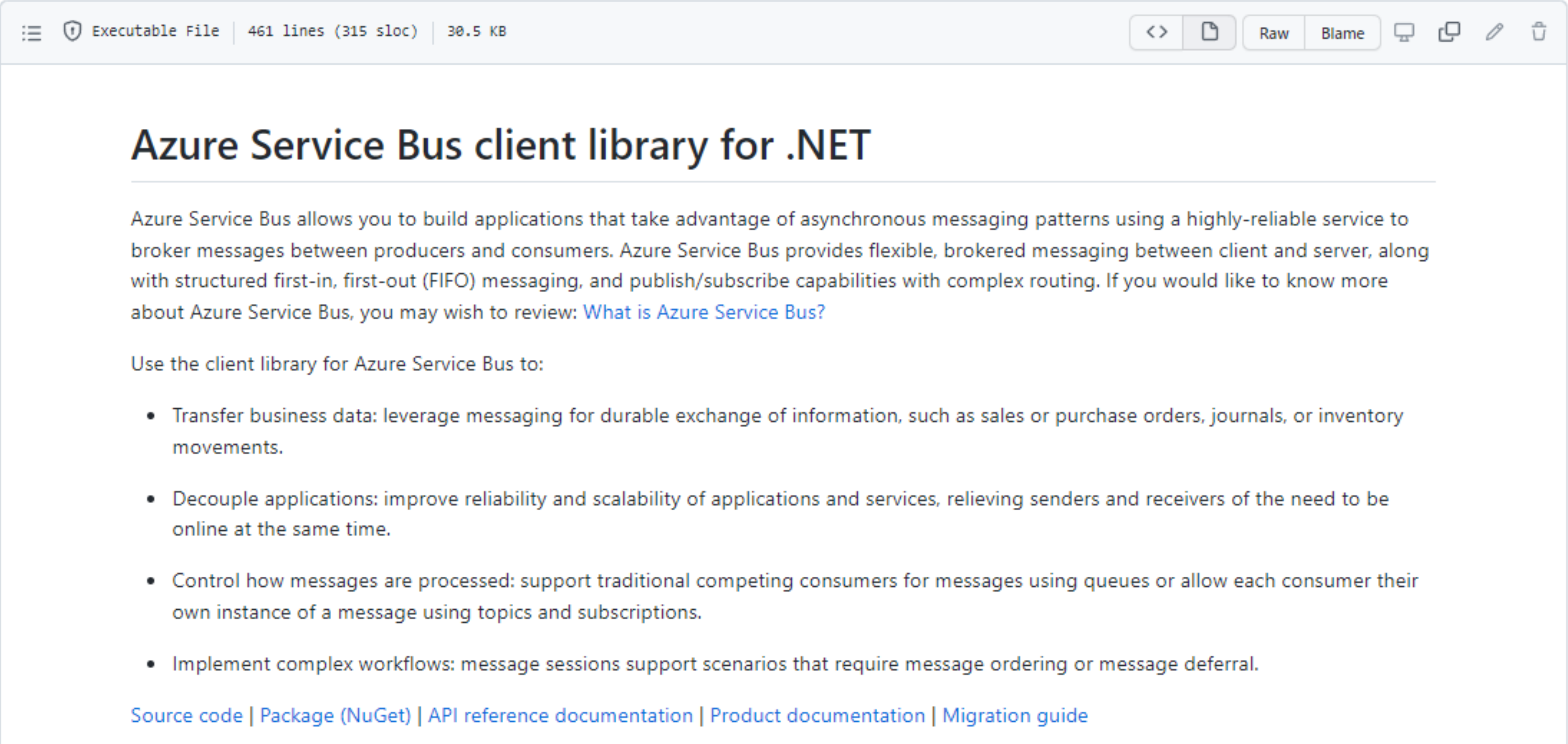
Azure Service Bus is a fully managed enterprise integration message broker. Service Bus can decouple applications and services. Service Bus offers a reliable and secure platform for asynchronous transfer of data and state. This client library allows for both sending and receiving messages using Azure Service Bus. For more information about Service Bus, see https://docs.microsoft.com/en-us/azure/service-bus-messaging/service-bus-messaging-overview

Version:	7.6.0
Author(s):	Microsoft
License:	MIT
Readme:	View Readme
Date published:	Thursday, February 10, 2022 (2/10/2022)
Project URL:	https://github.com/Azure/azure-sdk-for-net/blob/Azure.Messaging.ServiceBus_7.6.0/sdk/servicebus/Azure.Messaging.ServiceBus/README.md
Report Abuse:	https://www.nuget.org/packages/Azure.Messaging.ServiceBus/7.6.0/ReportAbuse
Tags:	Azure, Service, Bus, ServiceBus, .NET, AMQP, windowsazureofficial, azureofficial

Dependencies

```
.NETStandard,Version=v2.0
    Azure.Core (>= 1.22.0)
    Azure.Core.Amqp (>= 1.2.0)
    Microsoft.Azure.Amqp (>= 2.5.10)
    Microsoft.Bcl.AsyncInterfaces (>= 1.1.1)
    System.Memory.Data (>= 1.0.2)
```

```
PM> Install-Package Azure.Messaging.ServiceBus
```



Azure Service Bus client library for .NET

Azure Service Bus allows you to build applications that take advantage of asynchronous messaging patterns using a highly-reliable service to broker messages between producers and consumers. Azure Service Bus provides flexible, brokered messaging between client and server, along with structured first-in, first-out (FIFO) messaging, and publish/subscribe capabilities with complex routing. If you would like to know more about Azure Service Bus, you may wish to review: What is Azure Service Bus?

Use the client library for Azure Service Bus to:

- Transfer business data: leverage messaging for durable exchange of information, such as sales or purchase orders, journals, or inventory movements.
- Decouple applications: improve reliability and scalability of applications and services, relieving senders and receivers of the need to be online at the same time.
- Control how messages are processed: support traditional competing consumers for messages using queues or allow each consumer their own instance of a message using topics and subscriptions.
- Implement complex workflows: message sessions support scenarios that require message ordering or message deferral.

Source code | Package (NuGet) | API reference documentation | Product documentation | Migration guide

https://github.com/Azure/azure-sdk-for-net/tree/main/sdk/servicebus/Azure.Messaging.ServiceBus

Commonly Used Classes

Class	Description
ServiceBusAdministrationClient	Used to manage messaging entities within a service bus namespace.
ServiceBusClient	Top-level client through which all Service Bus entities can be interacted with.
ServiceBusSender	Used to send messages to queues and topics.
ServiceBusReceiver	Used to receive messages from queues and subscriptions.
ServiceBusProcessor	Provides using an event-based model for receiving and processing messages.
ServiceBusMessage	Used to represent a message to be sent to the service bus.

Demo

Simple Service Bus Messaging

- Creating a Service Bus Namespace
- Adding the Service Bus NuGet package
- Creating and sending messages
- Receiving and processing messages

Demo

Simple Chat Application

- Publish and subscribe messaging with topics and subscriptions

Summary

Azure Service Bus is one of a number of Azure messaging services

Azure Service Bus provides enterprise class messaging capabilities

Service Bus namespaces can contain Queues and Topics

Topics can contain Subscriptions

Queues provide point-to-point messaging

Topics and Subscriptions provide publish-subscribe messaging

Azure.Messaging.ServiceBus .NET SDK is available on NuGet and GitHub