✓ 100 XP ▶

Introduction

3 minutes

Modern applications frequently consist of multiple parts running on separate computers and devices, which are in locations around the world. Complex networks with varying reliability and speed exist between these components. A fundamental challenge with these distributed applications is how to communicate reliably between the components.

Suppose you are a cloud developer for Contoso Slices, a global pizza delivery chain. Your employer is upgrading their technology so that users can place orders from the web or their mobile apps. Those orders will be sent to the user's preferred storefront location, where employees will make the pizza. As the dough is rolled out, pizza put in oven, boxed, and put on a delivery vehicle, updates are sent to the user's mobile app. The users even receive location updates as the delivery driver heads toward them.

Contoso Slices previously created an online ordering system that immediately stored order data in a SQL Server database. Each store had to remember to manually refresh the "web orders" page to find out if they had new orders. In addition, during peak pizza times, such as televised sporting events, the system would frequently get deadlock exceptions and timeouts. Finally, the previous system lacked central payment processing or any kind of status updates for the user.

For this new, more ambitious project, Contoso has hired a cloud architect and plans to use a decoupled architecture.

In this module, we'll learn how Azure Service Bus can help build an application that stays reliable during peak demand. We'll also see how Azure Service Bus make it easy to add functionality to our applications. Along the way, we'll be writing the C# code necessary to put these lessons to work. Here, you will see how to use Azure Service Bus topics and queues in a distributed architecture to ensure reliable communications even at times of high demand. You will also write C# code that communicates through Service Bus.

Learning objectives

In this module, you will:

- Choose whether to use Service Bus queues, topics, or relays to communicate in a distributed application
- Configure an Azure Service Bus namespace in an Azure subscription

- Create a Service Bus **topic** and use it to send and receive messages
- Create a Service Bus **queue** and use it to send and receive messages

Prerequisites

None

Next unit: Choose a messaging platform

Continue >