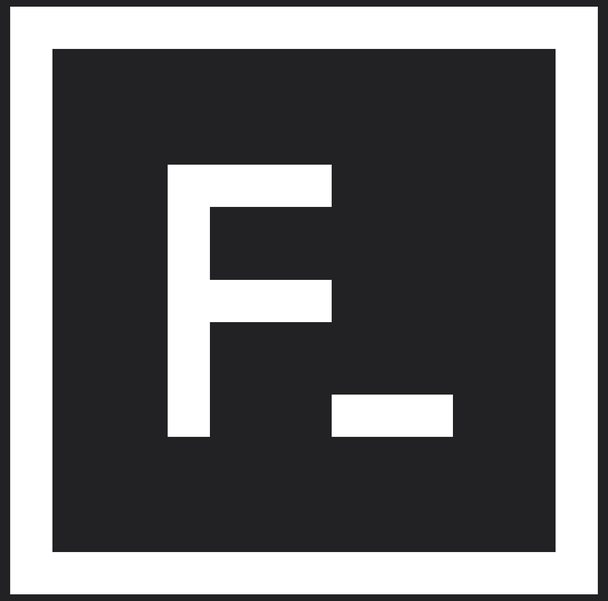
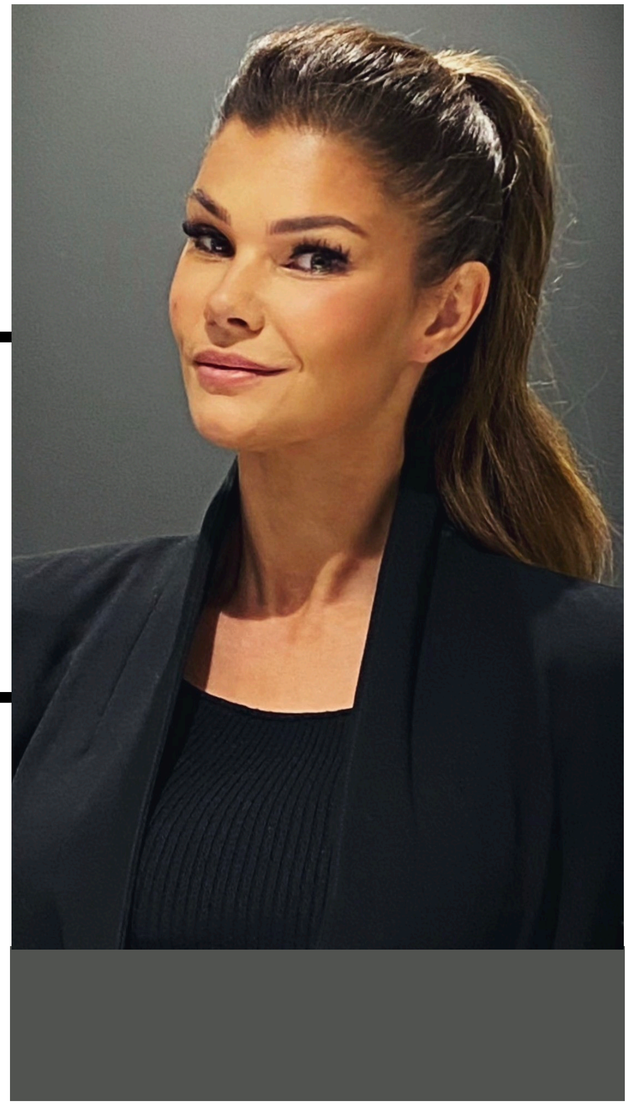
.NET & AZURE COMMUNITY - FORTE DIGITAL
PRESENTATION BY MONICA BEATE TVEDT, DIRECTOR OF TECHNOLOGY





Monica Beate Tvedt

Teknologidirektør

Forte_ Digital

TIDLIGERE

- Agency Director - Head of Microsoft Development, Mixed Reality & Microservices at Sopra Steria
- Head of UMS Innovation Center at Unified Messaging Systems
- Global Head of SaaS Development at Unified Messaging Systems
- Senior Software Engineer Consultant, Webstep @ Sparebanken Vest
- Software Engineer, CellVision
- Gründer

PROSJEKT 2020

- Kunde: **ASKO**
  Rolle: Arkitekt og Front-end lead
- Kunde: **Kværner**
  Rolle: Arkitekt og Mobilspesialist
- Kunde: **COVID-19 Digital Feberpoliklinikk**
  Rolle**:** Løsningsarkitekt

FOREDRAG 2020

Oslo Business Forum 2020, Relevans 2020, Global AI on Tour 2020, Women in Tech 2020, Lørn.Tech.

DIVERSE NTERESSER

Alpint, tennis, programmering, tegne, lese bøker

0.0	Recap: Azure Functions, SignalR, APIM
1.0	Azure Messaging Models
2.0	Azure Service Bus
3.0	Azure Queue Storage
4.0	Azure Event Hubs
-	-
5.0	Self Study

0.0 - Recap

Demo: Let’s take a quick look at what we learned at our last meetup..

1.0
Azure Messaging Models.
Choosing the right one.

Distributed Applications

When you have an application that consists of components running on different *computers*, *servers*, and *mobile devices*, better known as **loosely coupled architecture**, reliable communications **between** those components can be difficult and unreliable.

Azure provides several technologies that you can use to communicate more reliably, including:

- **Storage Queues**
- **Event Hubs**
- **Event Grid**
- **Service Bus**

Introducing a whole new kind of show

Spotify®

Forte Music Sharing App

Musicians can upload music they create to your platform by using a **web front** end or a **mobile app**. They can listen to and comment on other members' work. The application consists of a website that runs at *your ISP*, a mobile app that runs on *users' mobile devices*, a web API that runs in *Azure*, and both an **Azure SQL Database & Azure Blob Storage** where data is stored.

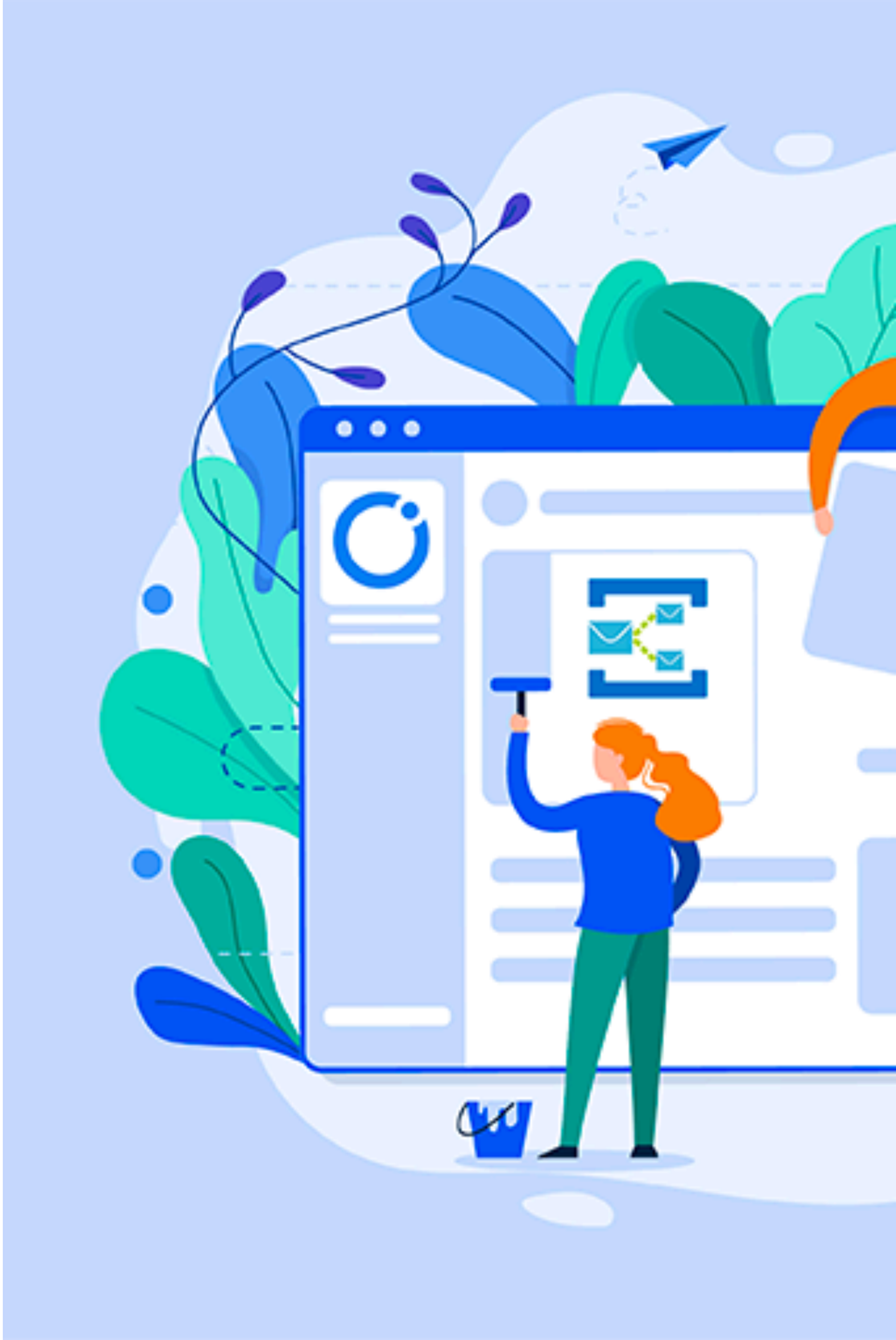
Identified Challenges

On high demand:

- Music files are not successfully uploaded
- Comments are not posted

Main cause:

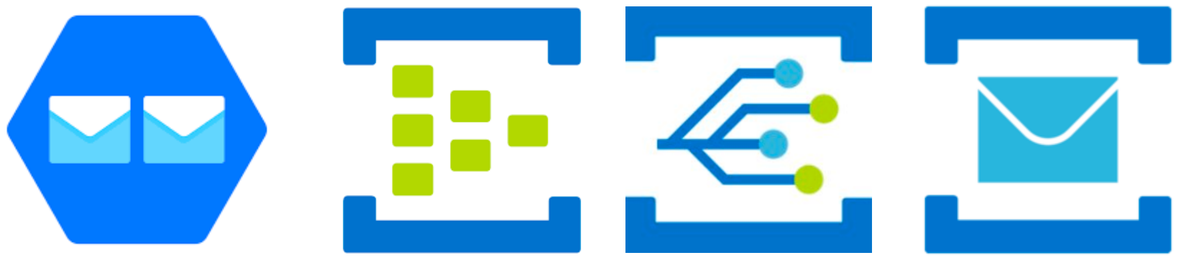
- Dropped messages between front-end components and the web API running in Azure



Our Task at Hand

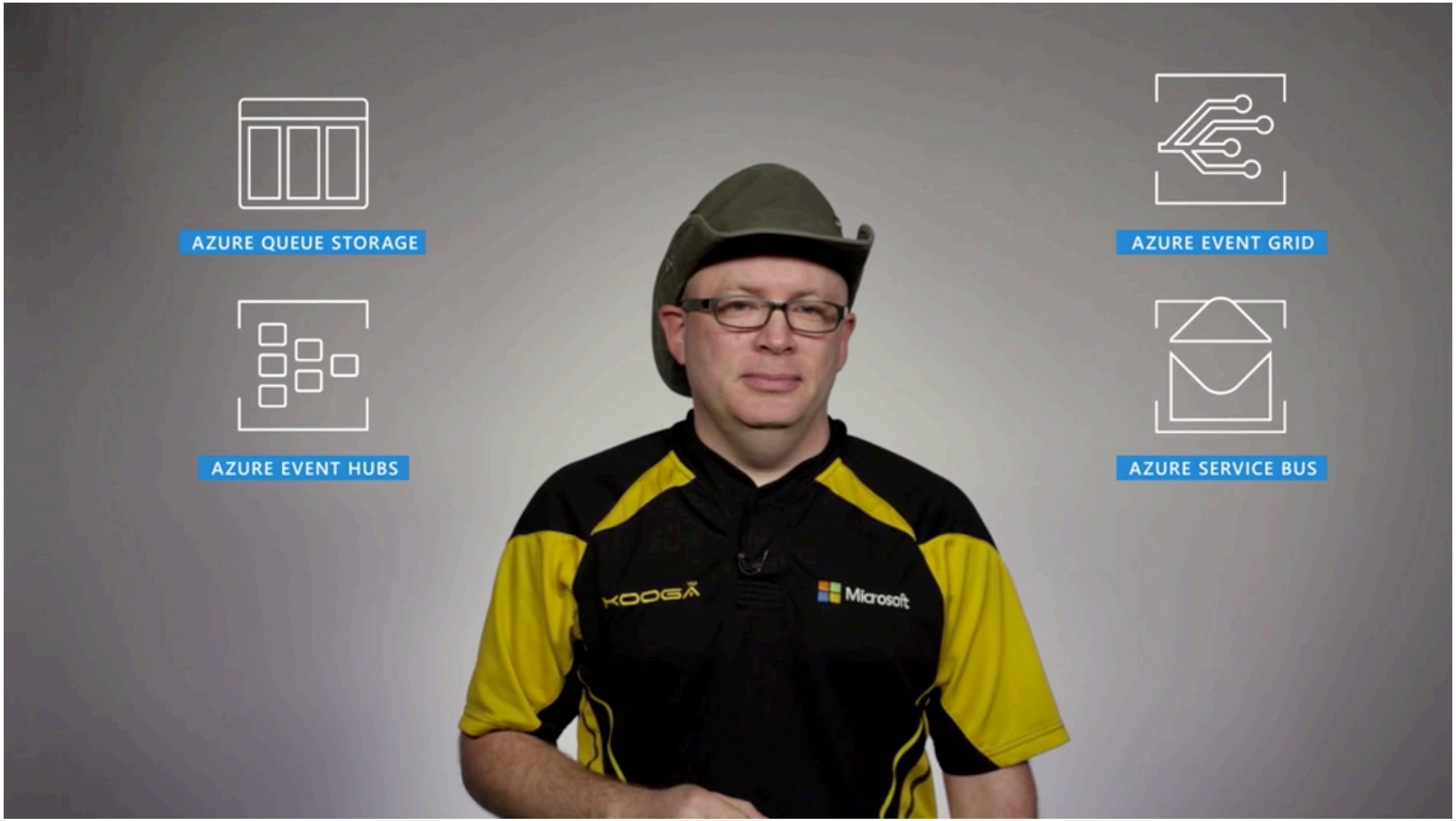
We plan to solve these issues by using one or more of the following technologies:

- **Azure Storage queues**
- **Azure Event Hubs**
- **Azure Event Grid**
- **Azure Service Bus**



Numero Uno: Messages or Events

Before we start, we need to understand *each individual communication* that the components of the application exchange, and whether it sends **messages** or **events.**


AZURE QUEUE STORAGE
AZURE EVENT GRID
AZURE EVENT HUBS
AZURE SERVICE BUS
KOOGA
Microsoft

Messages

In the terminology of distributed applications, messages have the following characteristics:

- **Contains Raw Data**
- **Produced by the Sender & Consumed by the Receiver**
- **Contains the message itself, not just a reference**
- **The sending component expects the message content to be processed in a certain way by the destination component**

Events

Events are lighter weight than messages, and are most often used for broadcast communications. The components sending the event are known as **publishers**, and receivers are known as **subscribers**.

- **Lightweight notification that indicates that something happened.**
- **Sent to multiple receivers, or to none at all.**
- **Intended to have a large number of subscribers for each publisher.**
- **The publisher of the event has no expectation about the action a receiving component takes**

Knowledge Check

Suppose you have a distributed application with a web service that **authenticates** users. When a user logs on, the web service notifies all the client applications so they can **display that user's status as "Online"**.

Is the login notification an example of a message or an event?

Knowledge Check

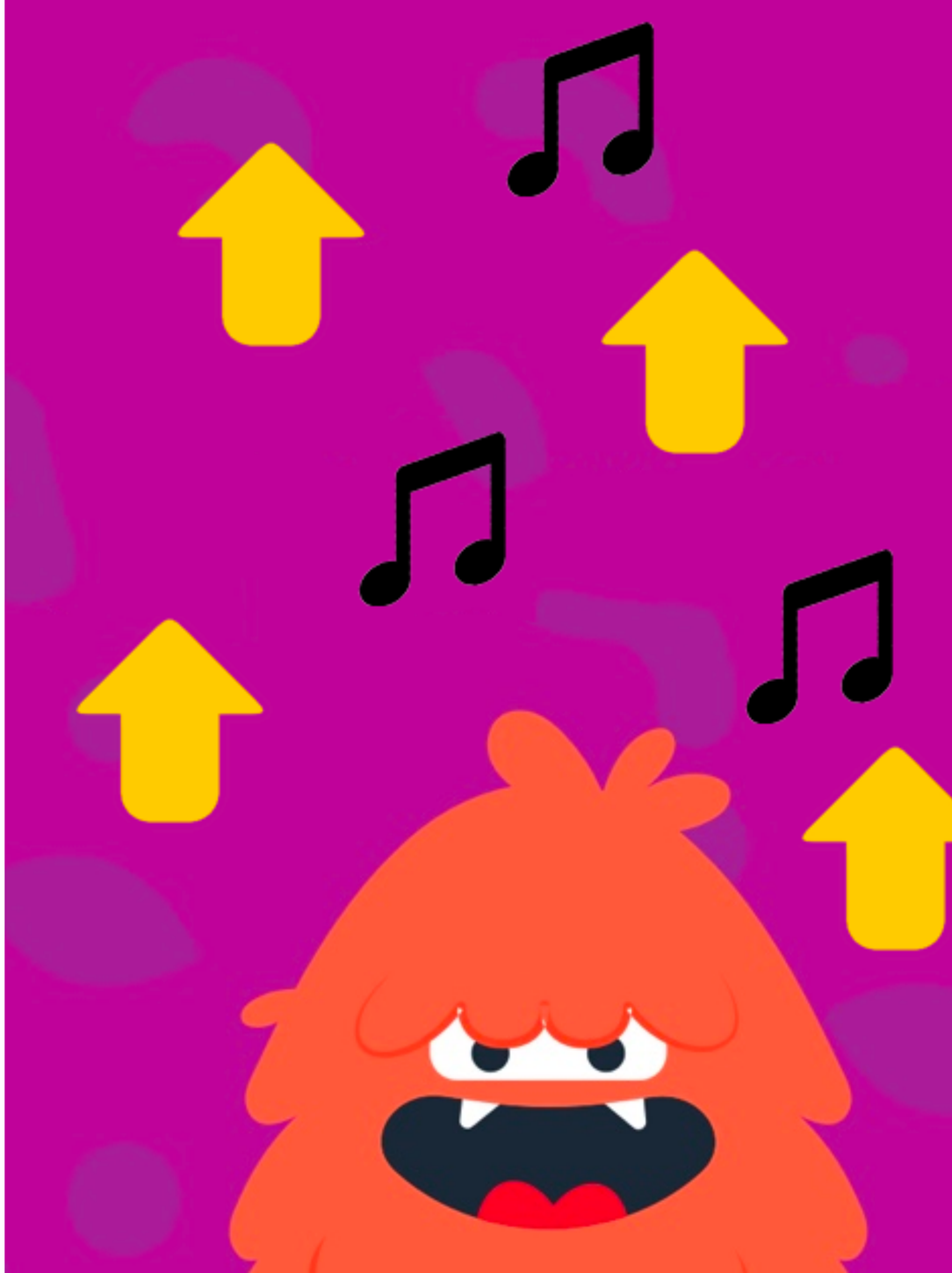
Suppose you have a distributed application with a web service that lets users manage their account.
Users can sign up, edit their profile, and **delete their account**.
When a user deletes their account, your web service notifies your data layer so the user's **data will be removed from the database**.

Is the delete-account notification an example of a message or an event?

Forte Music -
Messages or Events

- Ensure that music files are uploaded to our web API reliably from the mobile app.

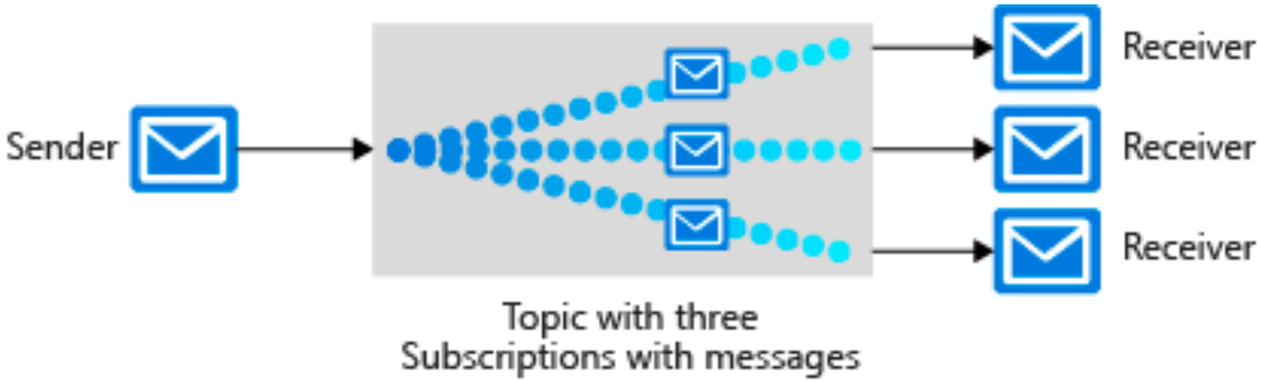
Answer:

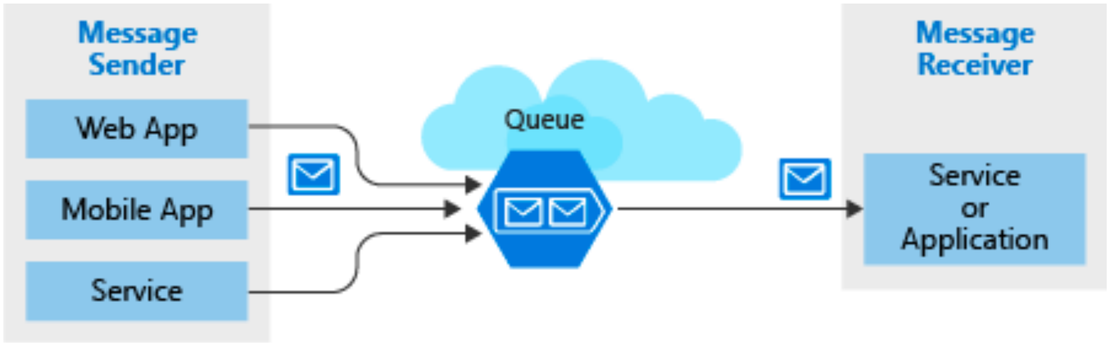


What is a Queue?

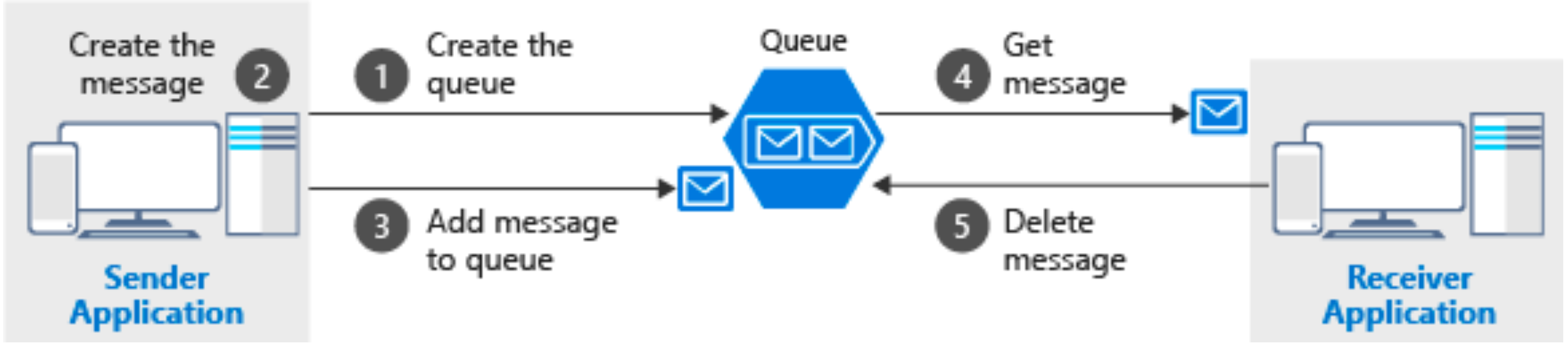
A **queue** is a simple temporary storage location for messages. A sending component adds a message to the queue. A destination component picks up the message at the front of the queue.

During peak times, messages may come in faster than destination components can handle them. Because source components have no direct connection to the destination, the source is unaffected and the queue will grow. Destination components will remove messages from the queue as they are able to handle them. When demand drops, destination components can catch up and the queue shortens.





The **sender** creates the queue and adds a message. The **receiver** retrieves a message, processes it, and then deletes the message from the queue. The following illustration shows a typical flow of this process.



```
//Service Bus Queues - send a message to the Queue;

using System.Threading;
using System.Threading.Tasks;
using Microsoft.Azure.ServiceBus;

var queueClient = new QueueClient(TextAppConnectionString, “PrivateMessageQueue”);

string message = "Sure would like a large pepperoni!”;
var encodedMessage = new Message(Encoding.UTF8.GetBytes(message));

await queueClient.SendAsync(encodedMessage);
```

//Service Bus Queues - receive messages from the Queue;

```
using System.Threading;
using System.Threading.Tasks;
using Microsoft.Azure.ServiceBus;
…
queueClient.RegisterMessageHandler(MessageHandler, messageHandlerOptions);

static async Task MessageHandler(Message message, CancellationToken token){

  Console.WriteLine($"Received message: SequenceNumber:
  {message.SystemProperties.SequenceNumber} Body:{Encoding.UTF8.GetString(message.Body)}");

  // Complete the message so that it is not received again
  await queueClient.CompleteAsync(message.SystemProperties.LockToken);
}
```

Which Messaging-based Service should we choose?

Use Azure Service Bus Queues if you:

- Need an At-Most-Once delivery guarantee.
- Need a FIFO guarantee.
- Need to group messages into transactions (order, payment)
- Want to receive messages without polling the queue.
- Need to provide a role-based access model to the queues.
- Need to handle messages larger than 64 KB but less than 256 KB.
- Queue size will not grow larger than 80 GB.
- Want to publish and consume batches of messages.

Use Azure Storage Queues if you:

- Need an audit trail (logs) of all messages that pass through the queue.
- Expect the queue to exceed 80 GB in size.
- Want to track progress for processing a message inside of the queue.

Use **Storage queues** when you want a **simple** and easy-to-code queue system.

For more advanced needs as Enterprise applications with higher security requirements, use Service Bus queues.

If you have *multiple destinations* for a single message, but need queue-like behavior, *use Service Bus topics*.

Knowledge Check

Which of the following queues should you use if you need first-in-first-out order and support for transactions?

- Azure Storage Queues
- Azure Service Bus Queues

Knowledge Check

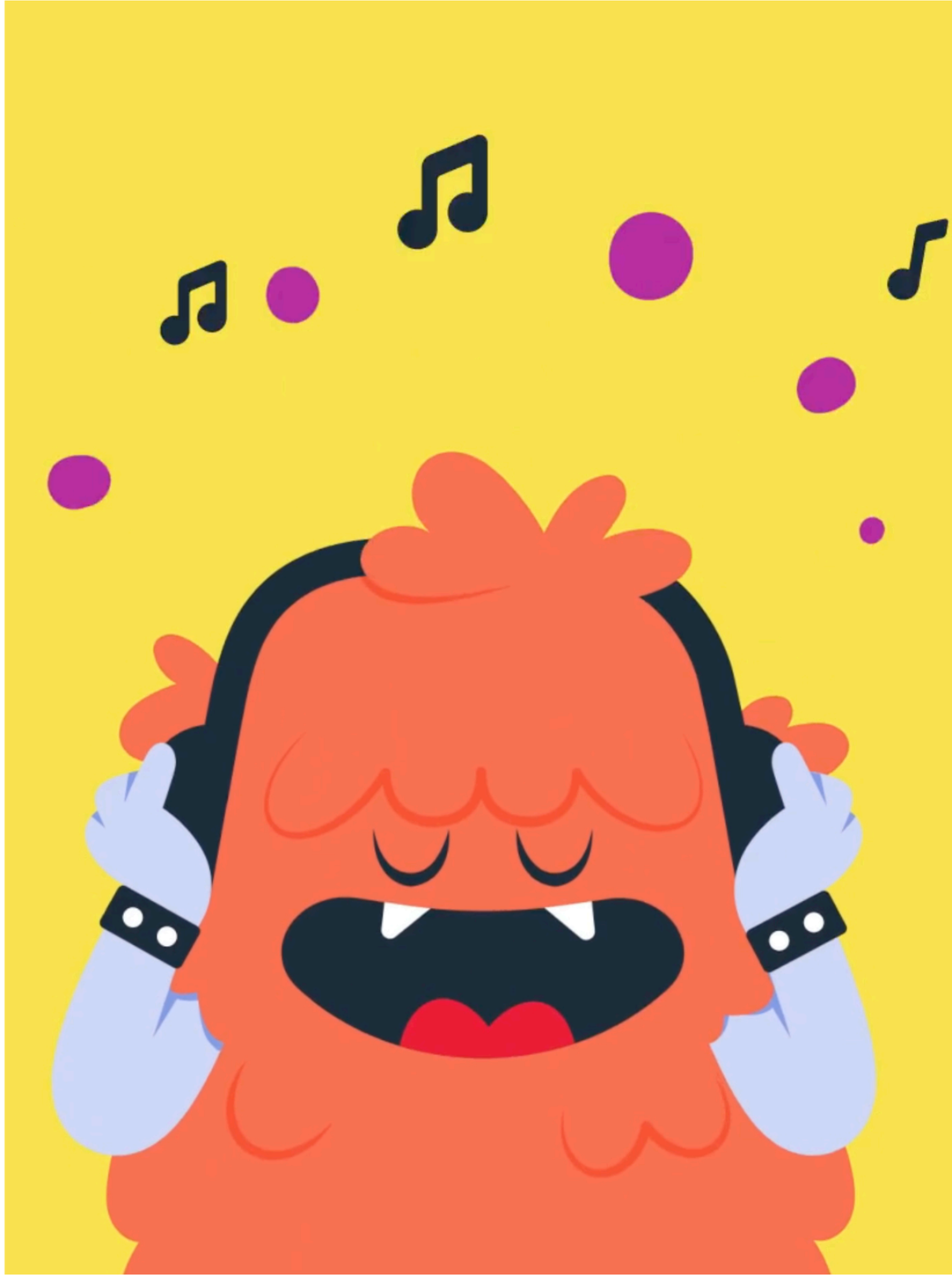
You can add a message to an Azure Service Bus queue that is 2 MB in size

True or false?

Forte Music -
Messages or Events

- When a user uploads a new song, we need to notify all the mobile apps installed on user devices around the world who are interested in that genre.

Answer:



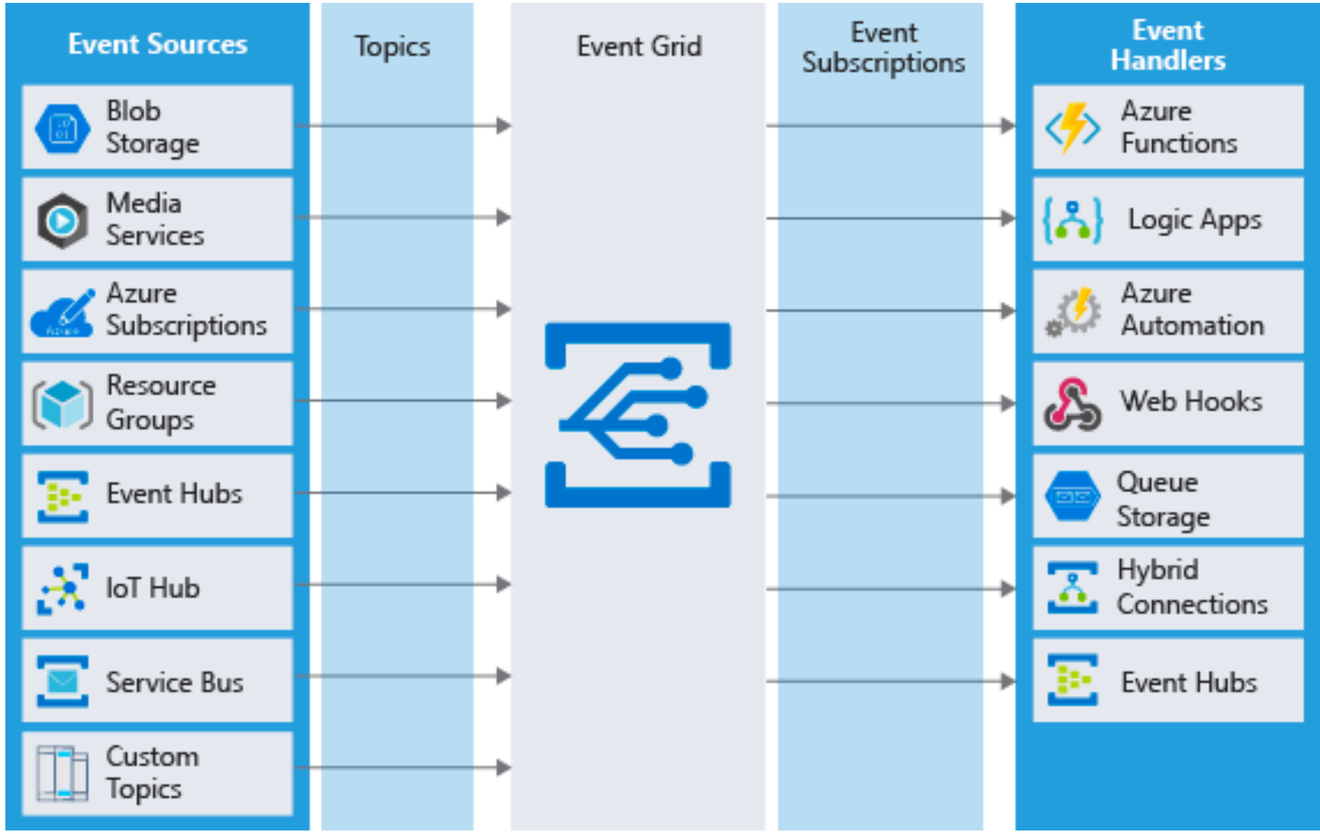
Choosing Event Grid

The publisher of the sound file doesn't need to know about any of the subscribers interested in the shared music. Also, we want to have a one-to-many relationship where we can have multiple subscribers who can optionally decide whether they are interested in this new song.

What is Event grid

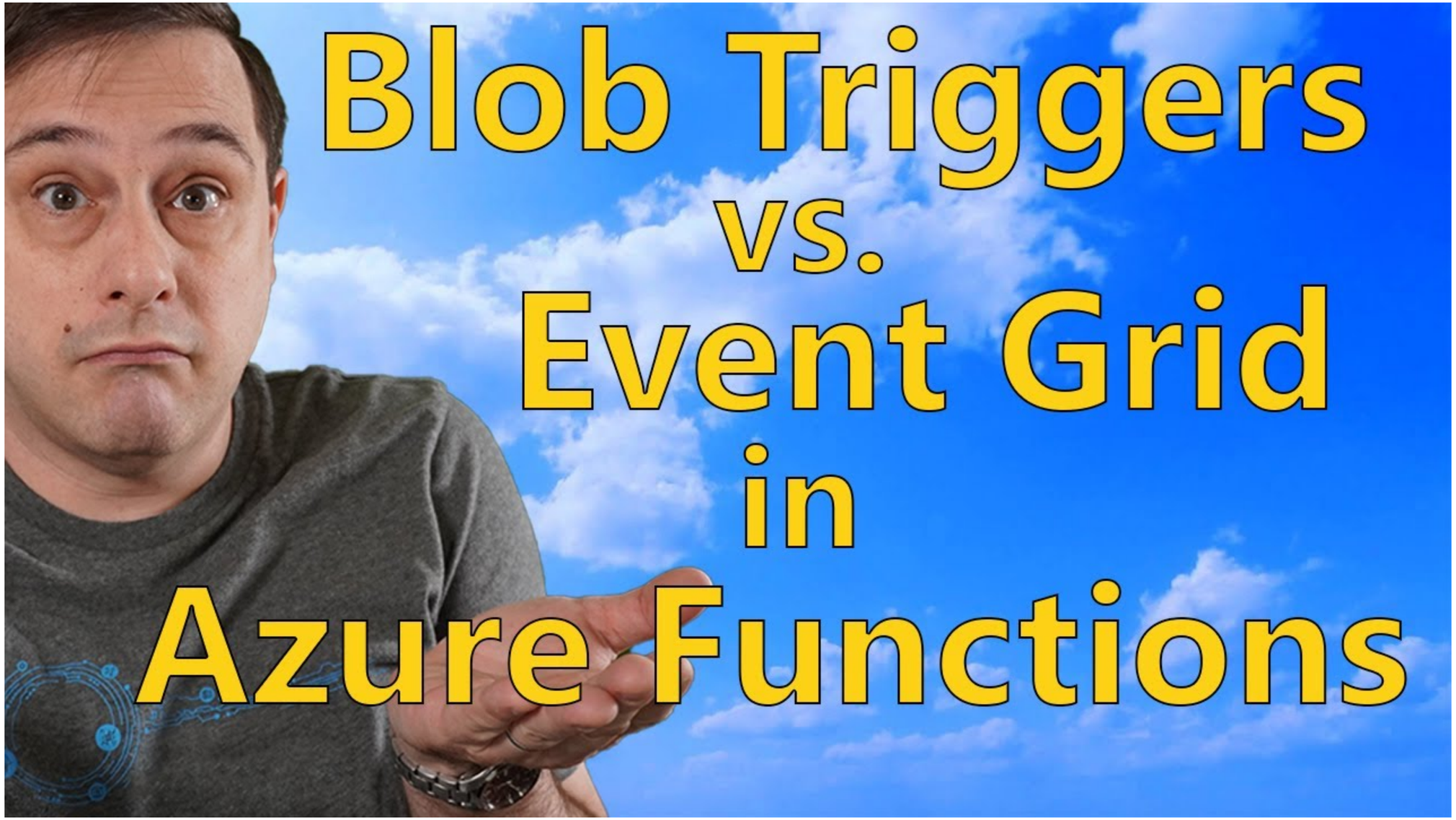
Azure Event Grid is a routing service running on top of *Azure Service Fabric*. Event Grid distributes *events* from different sources, to different handlers. There are several concepts in Azure Event Grid that connect a source to a subscriber:

- **Events:** What happened.
- **Event sources:** Where the event took place.
- **Topics:** The endpoint where publishers send events.
- **Event subscriptions:** The endpoint or built-in mechanism to route events, sometimes to multiple handlers. Subscriptions are also used by handlers to filter incoming events intelligently.
- **Event handlers:** The app or service reacting to the event.

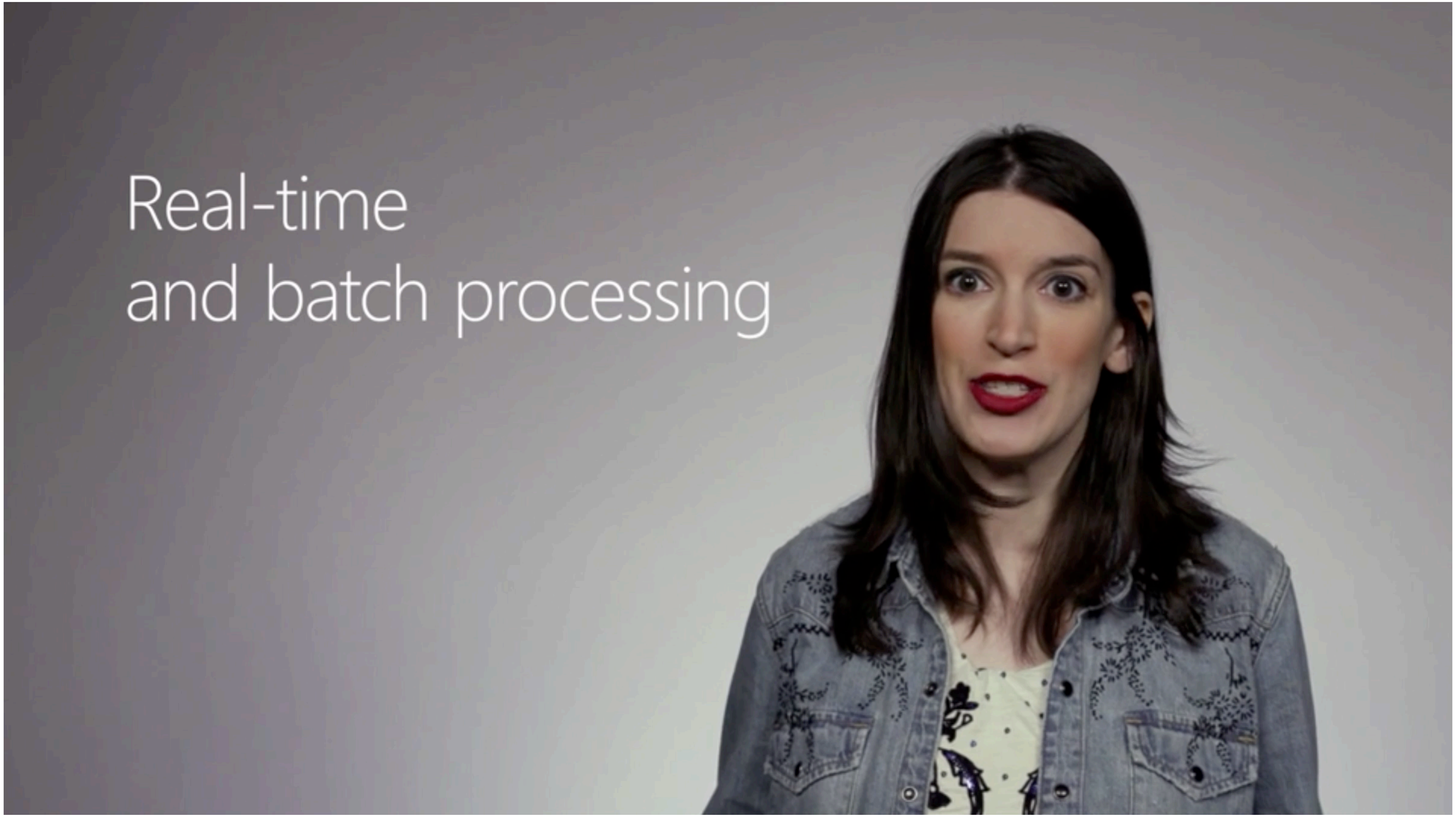


For example, when a new song is uploaded, status changed event could be sent to **Azure Event Grid**, then on to **Azure Functions**, and to **Azure Notification Hubs** - notifying all the mobile apps

Hmm.. Azure Functions can execute code in response to the new song being added to the Blob storage account… But why not just use BlobTriggers?


Blob Triggers
vs.
Event Grid
in
Azure Functions

4.0
Azure Event Hubs.
High-flow Stream Analytics.


Real-time
and batch processing

Which Event-based Service should we choose?

Use Azure Event Grid if you need these features:

- **Simplicity:** It is straightforward to connect sources to subscribers in Event Grid. One event at a time.
- **Advanced filtering:** Subscriptions have close control over the events they receive from a topic.
- **Fan-out:** You can subscribe to an unlimited number of endpoints to the same events and topics.
- **Reliability:** Event Grid retries event delivery for up to 24 hours for each subscription.
- **Pay-per-event:** Pay only for the number of events that you transmit.

Which Event-based Service should we choose?

Use Azure Event Hub if you:

- You need to save a stream of events to Data Lake or Blob storage.
- You need to support authenticating a large number of publishers.
- You need aggregation or analytics on your event stream.
- You need reliable messaging or resiliency.
  inside of the queue.

5.0
Self Study

Microsoft Learn - Connect your services together

https://docs.microsoft.com/en-us/learn/paths/connect-your-services-together/

Thank you.