

Edge Services & Messaging Systems

Java Accelerator 7

Lesson 5.2



The background is a dark charcoal gray with a series of parallel diagonal lines running from the top-left to the bottom-right. Overlaid on this are several teal-colored geometric shapes: a large central triangle pointing right, a smaller triangle to its left, and a square to its right. Scattered around these shapes are various white line-art symbols, including a plus sign, a minus sign, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a cross, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, a circle with a cross, a circle with a dot, a circle with a horizontal line, a circle with a vertical line, a circle with a diagonal line, and a circle with a cross.

WELCOME

Learning Outcomes

By the end of this lesson, you will be able to:

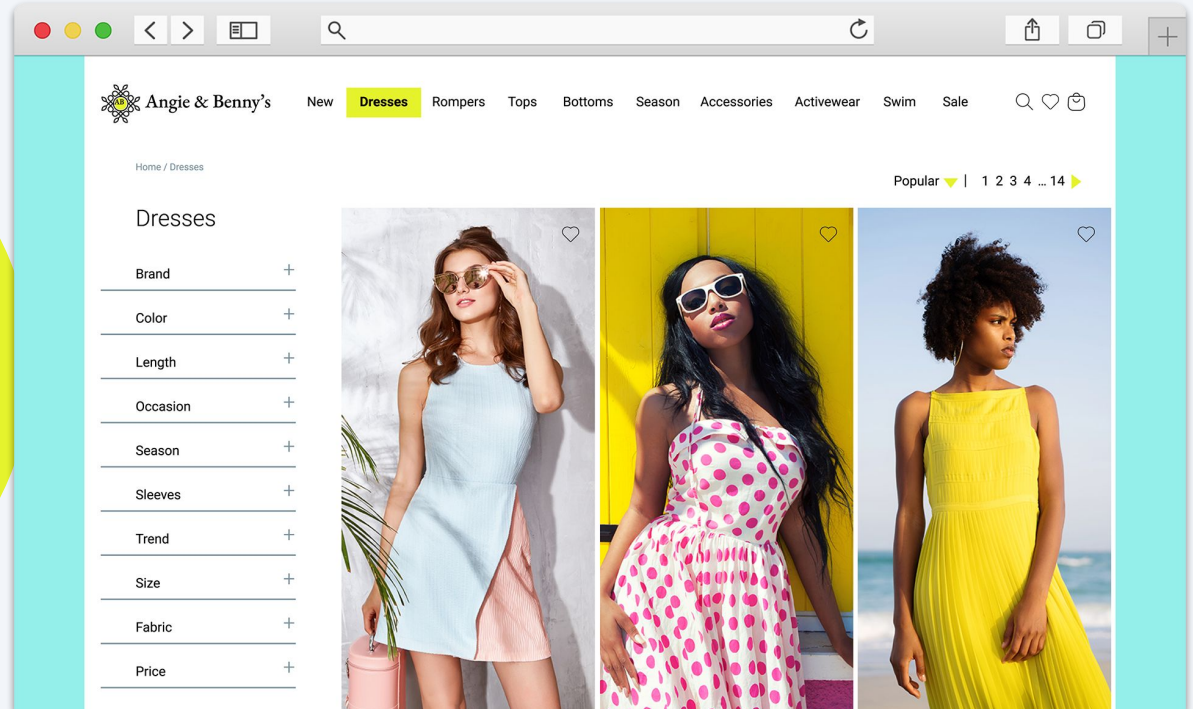
- Explain what an edge service is.
- Use an edge service in a microservices architecture.
- Explain how a queue works in a microservices architecture.
- Implement queues in a microservices architecture.
- Explain what a cache is.
- Compare and contrast buffering and caching.
- Use a cache in a microservices architecture.

Edge Services

Edge Services

A growing brick-and-mortar clothing retailer has hired us to break down their monolithic shopping app.

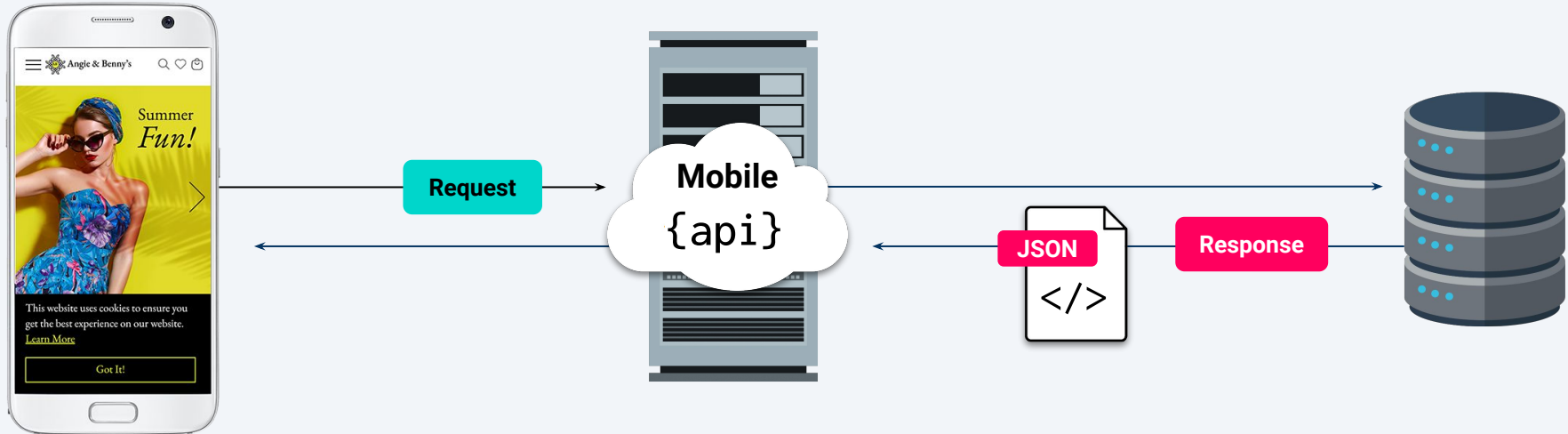
Not only is it a monolith, but it's only ever had a desktop interface. They want the application broken down into microservices.



Edge Services

They also want us to create APIs that will allow different clients to have their specific requests fulfilled.

Obviously a mobile interface is a must, but they want to be able to provide information to kiosks as well.



What Is an Edge Service?

An **edge service**:



Is an entry point into a service that sits in front of an API.



Is exposed to the public internet.



Acts as a gateway to other services.



Routes HTTP requests to the appropriate services.

Why Use an Edge Service?

Reasons to use an edge service:



Might improve security through authentication.



Might improve performance through load balancing.



Improves the user experience.



Can be tailored to specific clients.



Allows us to request from multiple services with a single request.

How Does an Edge Service Work?

01

A client makes a request.

02

The edge service looks for a microservice (or multiple microservices) to deliver the request.

03

Multiple microservices are invoked and aggregated.

04

The edge service returns the request to the caller.

Things to Consider when Using an Edge Service

Benefits

Encapsulates the internal structure of the app.

Each kind of client gets their own API.

Concerns

Another component to manage and update.

Feign REST Client Library



Feign is another library developed by Netflix for use in their platform and subsequently made open source.

Feign REST Client Library

01

The main purpose of the Feign REST client library is to make the creation of REST API clients easier and faster.

02

Creating a Feign client requires only the creation of an interface and the addition of annotations—similar to how JPA repositories are created.

03

Feign automatically integrates with the Eureka service registry/service discovery server.

Feign REST Client Library

The annotations used to create a client method to consume a REST endpoint are the same as those we use to create a REST endpoint with Spring:



`@RequestMapping`



`@PutMapping`



`@GetMapping`



`@DeleteMapping`



`@PostMapping`



`@PathVariable`



Time to Code

Hello Cloud v3

Suggested Time:

Queues

Grocery Store Line

We stand in lines (aka **queues**) all the time: grocery store lines, bathroom lines at concerts and sporting events, and so on.



Pizzeria

At the pizzeria, if there is only one phone line, only one call (**order**) can be processed at a time.

The solution? Add phone lines! When it gets really busy, we need someone from the kitchen to come up to the front to help answer the phones, but we don't need to hire someone solely to do that because it's an issue only during the dinner rush.



Pizzeria

When a huge batch of orders comes in, the result is a queue of order tickets. These are processed and may be routed—beyond just "Here's an order; make the whole order."

The salad might be made by one person, another person might operate the deep fryer for the corn dogs, and the person who was just answering the phone might come back and help make the pizzas.



How Does a Queue Work?

01

The producer creates a new entry/message.

02

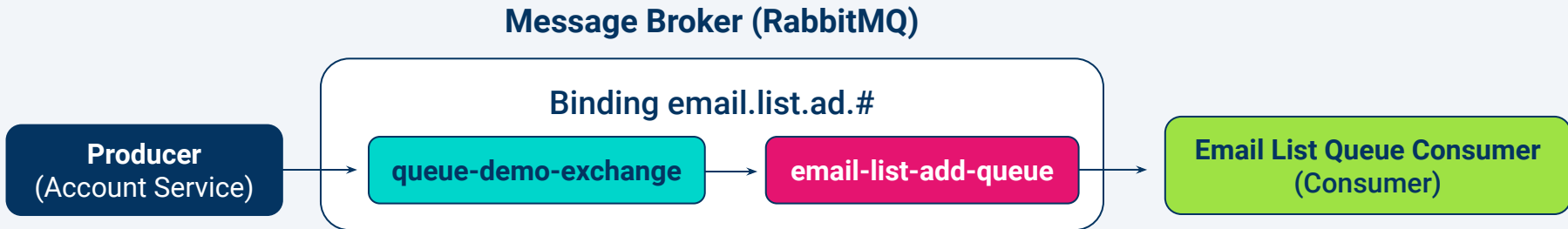
The entry/message includes a binding and a binding key (routing key).

03

The entry/message is sent to a message broker. such as RabbitMQ.

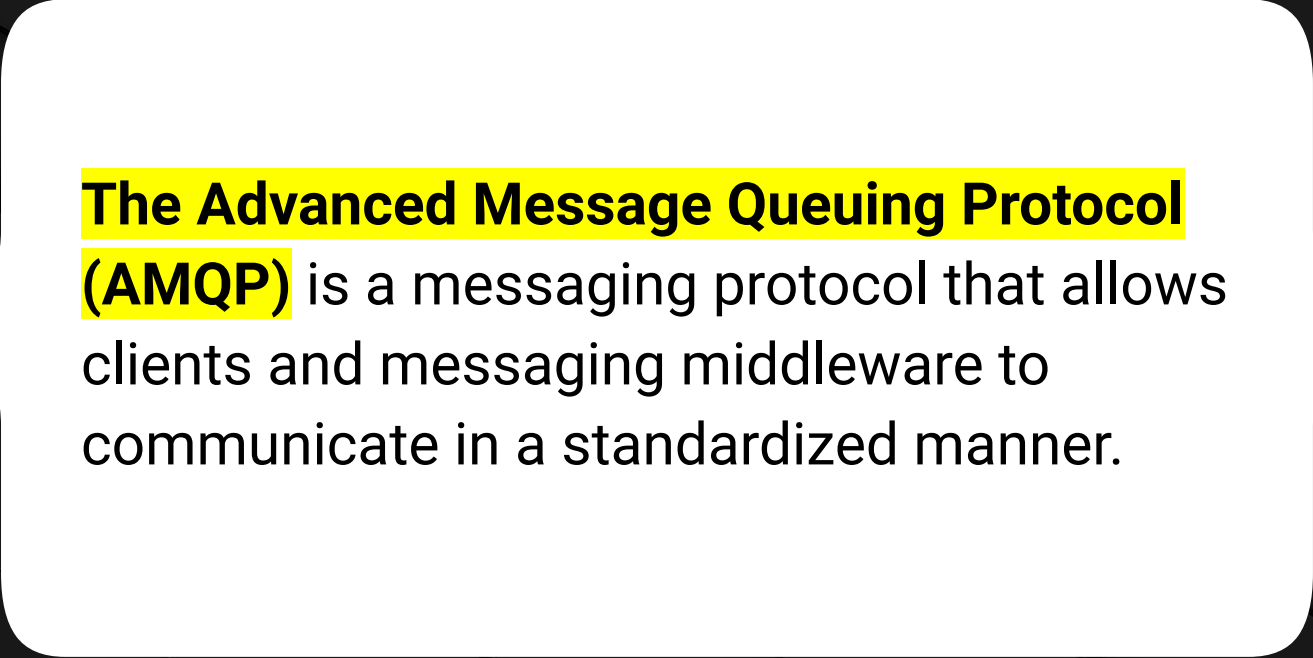
04

The entry/message is routed to the appropriate queue(s) based on the binding key and distribution protocols.





**There are a variety of rules for how an entry/message can be distributed.
For now, we will consider a one-to-one relationship: one message to one queue.**



The Advanced Message Queuing Protocol (AMQP) is a messaging protocol that allows clients and messaging middleware to communicate in a standardized manner.

The Advanced Message Queuing Protocol (AMQP)



RabbitMQ and the Spring client libraries conform to AMQP.



AMQP mandates the behavior between provider and client.



Think SMTP, HTTP, and FTP.



Why AMQP? Because interoperability matters!

Introduction to Queues

Queue

A queue is simply a place to store messages.

Producer

A producer is anything that creates new messages and places them in the queue for processing.

Consumer

A consumer is anything that processes, or consumes, messages from a queue.



Time to Code

Email List System

Suggested Time:

Caching

Caching

Our application contains data for movies: cast, crew, box office numbers, reviews, and so on. We're adding a new feature that allows people to find movies by searching two or more of the people starring in the film.

For example, the user would input Johnny Depp and Orlando Bloom, and the app would return *Pirates of the Caribbean*.

Title	Actors	Description	Trailer
<i>Pirates of the Caribbean</i>	Johnny Depp, Orlando Bloom, Geoffrey Rush, Keira Knightley	<i>Pirates of the Caribbean</i> is a series of fantasy swashbuckler films produced by Jerry Bruckheimer and based on Walt Disney's theme park attraction of the same name. The film series serves as a major component of the eponymous media franchise.	Pirates_Caribbean_Trailer.mp4



This new feature will require a bit of processing power to search through the entire database of films.



How can we help ensure that this new feature's need for processing doesn't increase the load time for common requests?

What Is a Cache?



Stores data for future reference.



A **cache** is a copy of the original data.



Previously accessed data is stored in a cache to make retrieval faster.



Example: A website might cache images to improve performance—specifically, images that might appear on multiple pages of the site.

Why Use a Cache?

01

It's less expensive:

- Saves money.
- Saves processing time.

02

Data transformation requires processing power. Once that data is transformed, caching the results saves the app from having to process it again.

03

Improved user experience.



Compare and Contrast Buffering and Caching



When a video app is loading an MP4 file, the file is **buffered** to match the transmission speed of the sender and the receiver.



Like caching, buffering is useful for improving the user experience.



Buffering preloads data from the original source.



Caching is accessing a copy of the data.

How Does a Cache Work?

01

Opening an app presents data that is typically specific to that user.

02

The app attempts to fetch the data from the cache.

03

If the data exists in the cache, it is presented.

04

If the data is not in the cache, the data is requested from the database.

05

This data can then be cached for faster retrieval when future requests are made.

Things to Consider When Using a Cache



Caches take up space (memory or disk).



Cached data might not be the most recent. Therefore, it might be inaccurate.



Caches are most useful with generic data (not user-specific data).

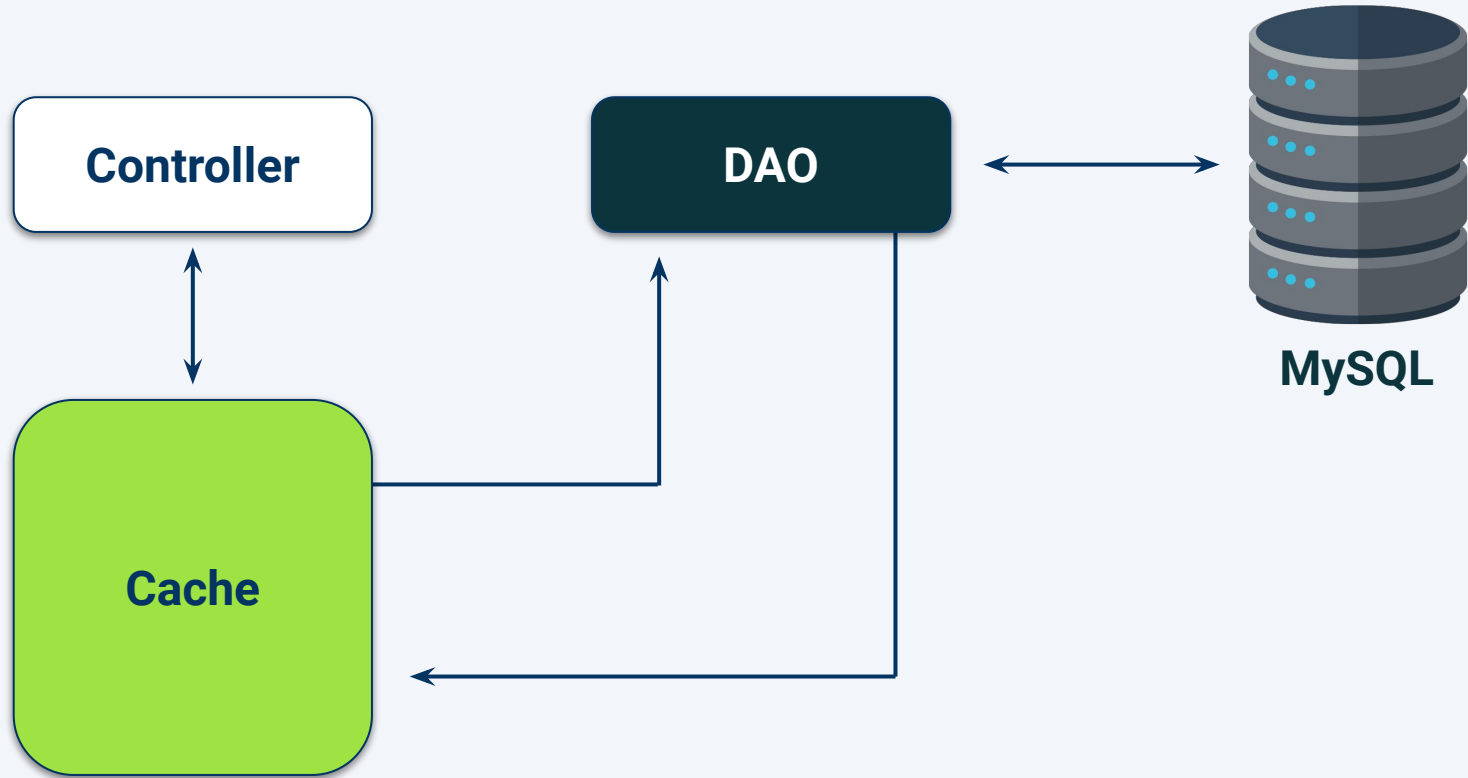


When data changes often, a cache is likely not very effective.



If the accuracy of the data is critical, a cache is likely not the best solution.

How Does a Cache Work?





Time to <code>