Authenticating and Authorizing Microservices

Mark Heath
CLOUD ARCHITECT

@mark_heath www.markheath.net

Overview

Authentication and authorization

Defense in depth

Authenticating end users

- API gateway
- Identity server

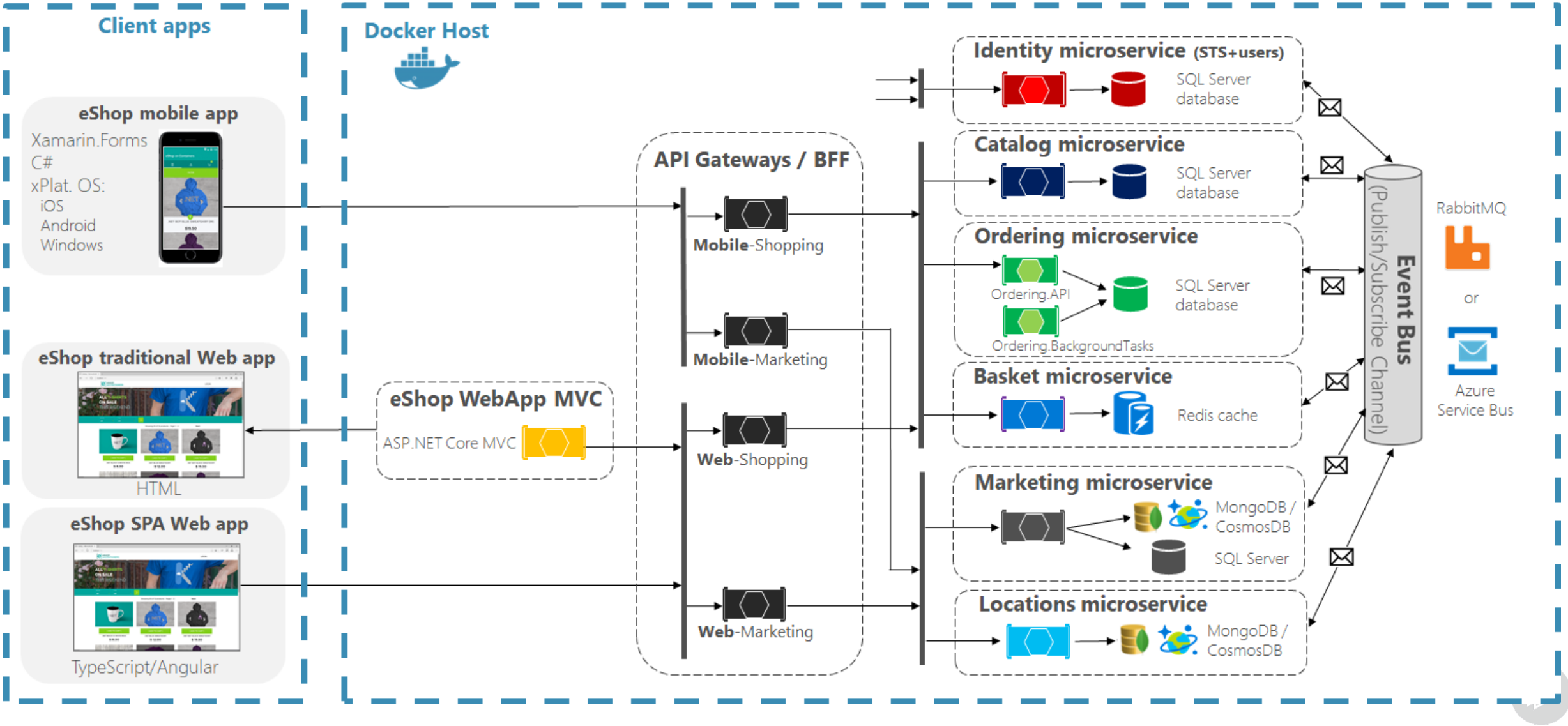
OAuth 2.0 and OpenID Connect

Authenticating between microservices

- “On behalf of” calls

Authorization

eShopOnContainers Architecture



Authentication and Authorization

Authentication

- **Who you are**
  - Allows us to request "my" data
- **HTTP requests**
  - Anonymous by default
- **Pass user information in HTTP headers:**
  - Username and password
  - API key
  - Bearer token

Authorization

- **What you are allowed to do**
- **e.g. Who can issue a refund?**
  - Only an employee
- **e.g. Can I update a shipping address?**
  - Only for my orders
- **Authorization based on:**
  - Who I am
  - What action I am performing
  - What data I am accessing

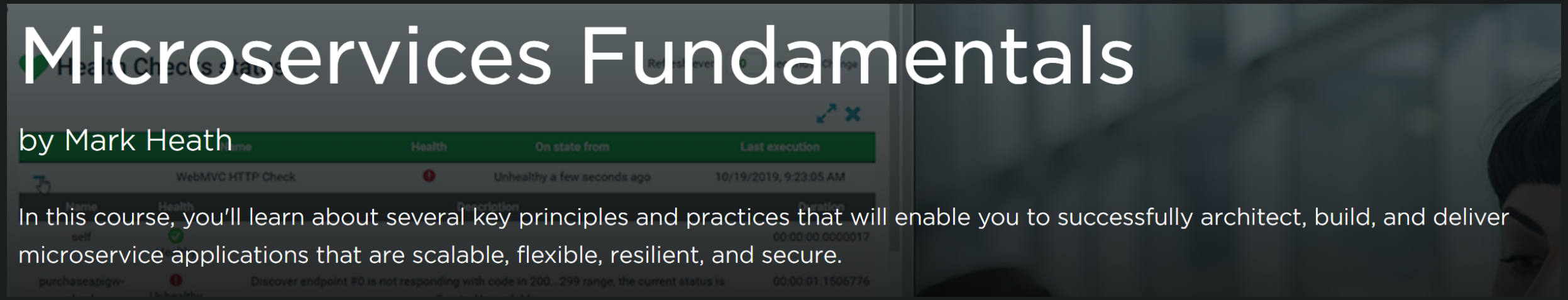
Defense in Depth

Authentication and authorization should form part of a wider security strategy

Defense in Depth

Don’t rely on a single security mechanism. Apply multiple layers of protection.

More on Defense in Depth



app.pluralsight.com/library/courses/microservices-fundamentals

Multiple Layers of Security

Encryption

In transit

Industry standard algorithms

At rest

Encrypted disks

Network

Not all endpoints should be publicly accessible

IP whitelisting, firewalls & VNets

API gateways

Monitoring

Regular security reviews

Security audits

Scan for known vulnerabilities

Penetration testing

Authenticating End Users

End User Login Process

1

User visits login page

Enters username and password

2

Username and password are sent to server

Encrypted in transit (HTTPS)

“Authorization” header

```
Basic dXNlcjpleG1wbGU=
```

Basic Access Authentication

3

Server validates the password

Database lookup

Passwords are stored as hashes

4

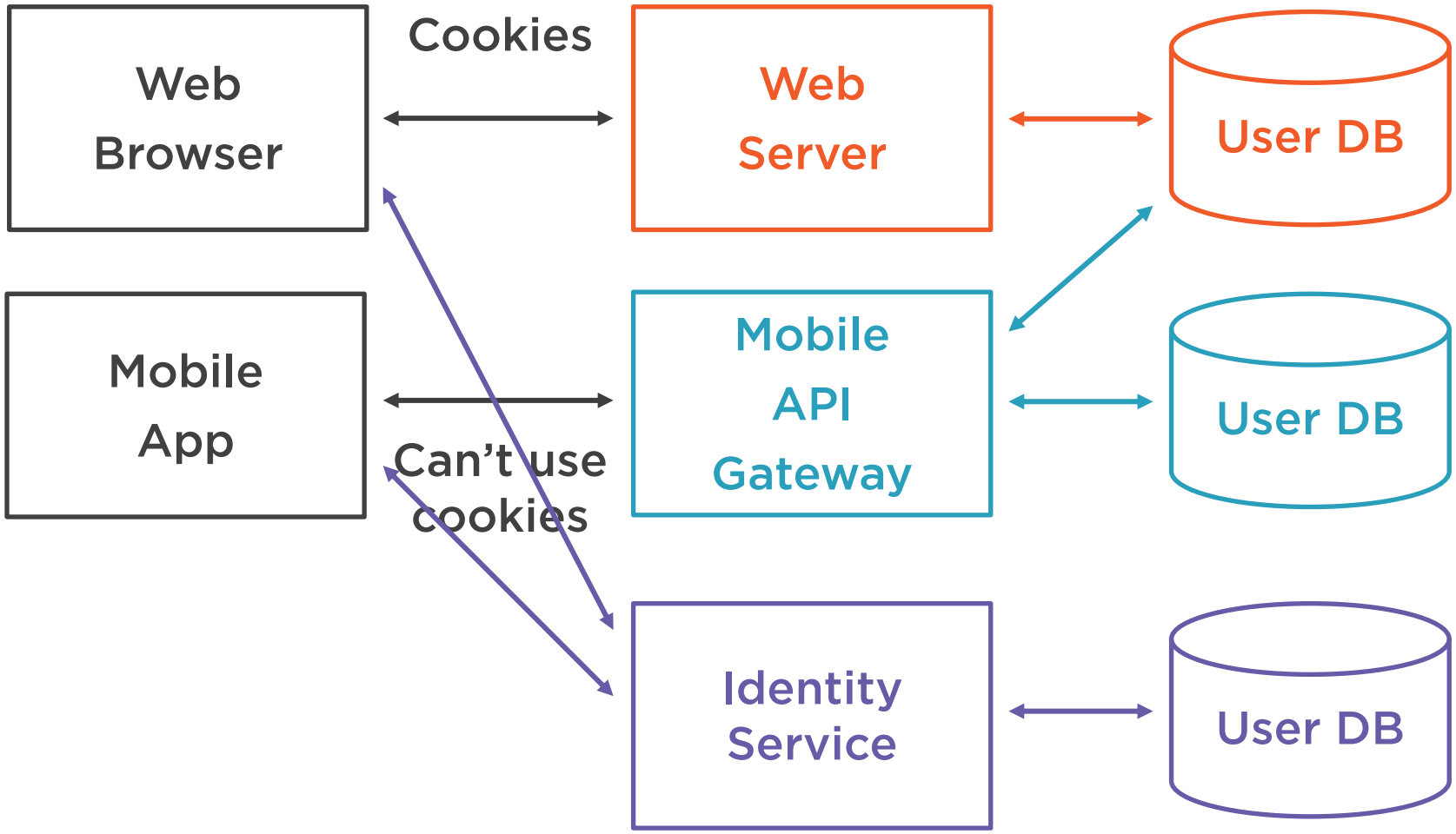
Web server issues a cookie

Stored in the browser

Sent with every subsequent HTTP request

Server validates the cookie

Problems with Basic Authentication



- Password salting and hashing
- User registration
- Password reset
- Brute force attack detection
- Two factor authentication

Basic Authentication

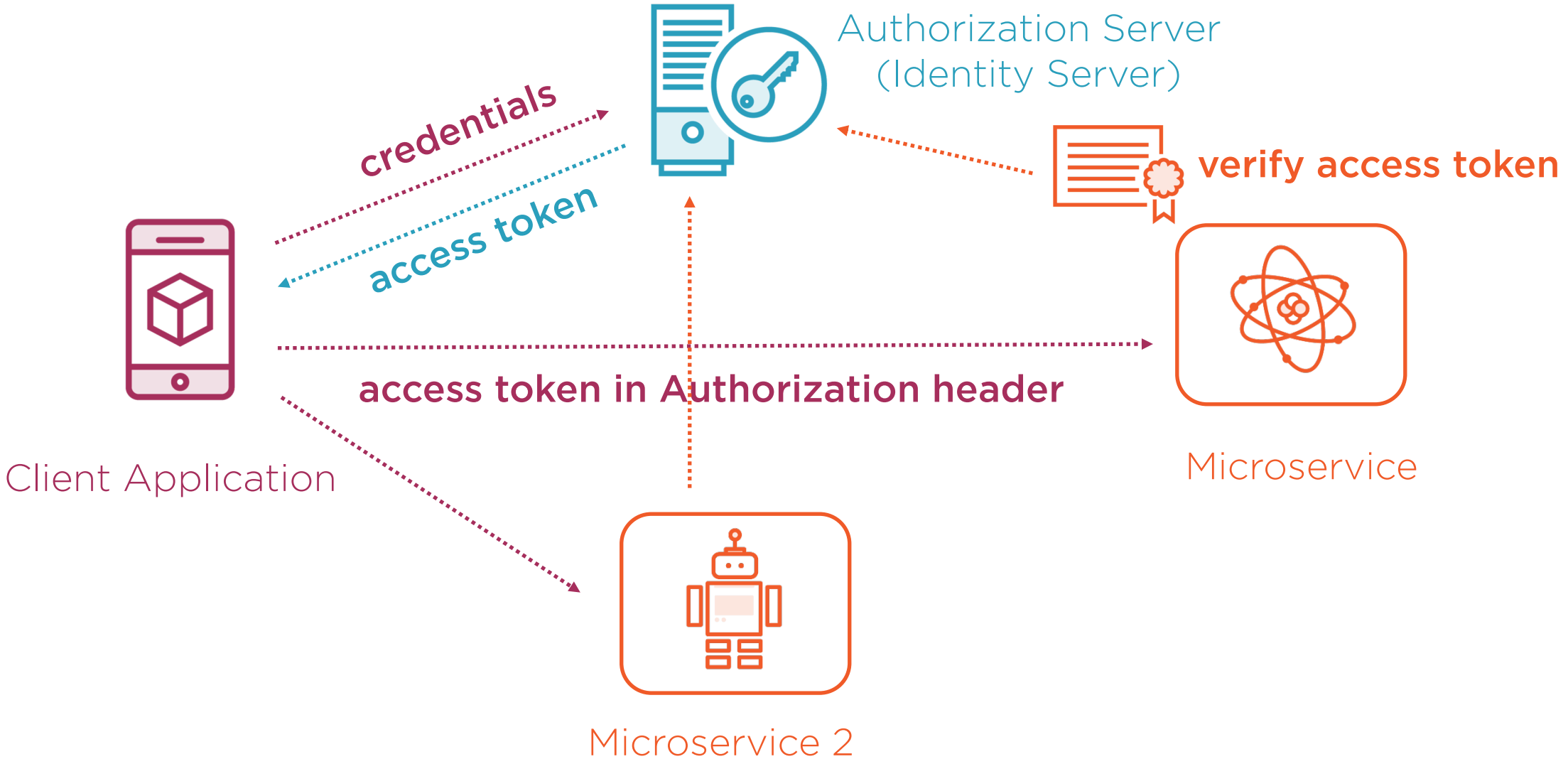
Username and password are passed in a HTTP header

Challenges:

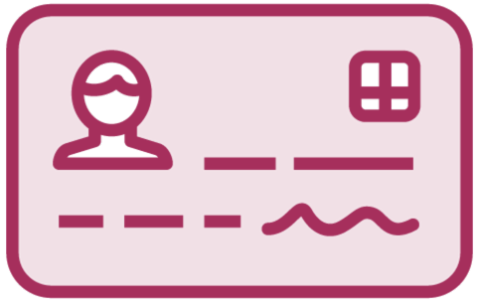
Managing user credentials is complex

Multiple microservices need authentication

Using an Authorization Server



OAuth 2.0 and OpenID Connect



OAuth 2.0

- Delegated authorization

OpenID Connect

- Identity (ID) token
- JSON web token (JWT)

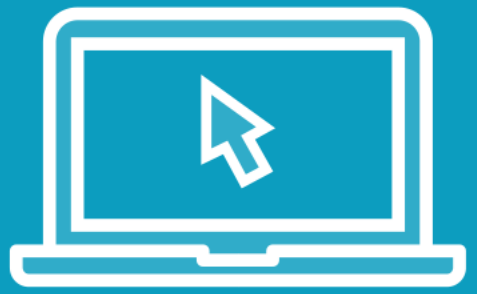
OAuth "flows"

- Interaction between client application, authorization server and microservice

Third party identity services

- E.g. Facebook, Google, Azure AD

Demo

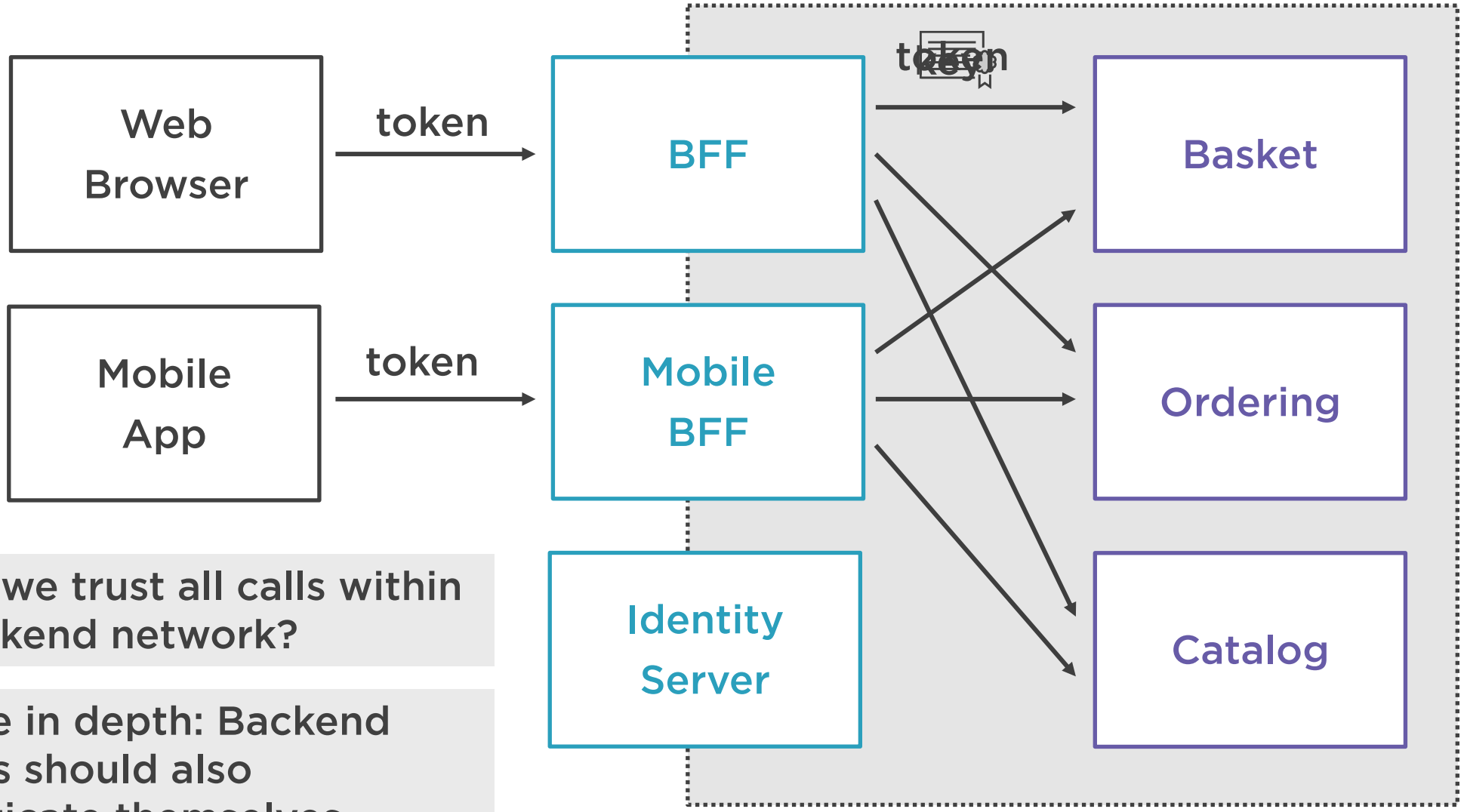


eShopOnContainers authentication

Identity microservice

- Identity Server 4
- https://identityserver.io/
- Runs as a container
- Is publicly accessible

Authenticating Between Microservices



Should we trust all calls within the backend network?

Defense in depth: Backend services should also authenticate themselves

Authorization

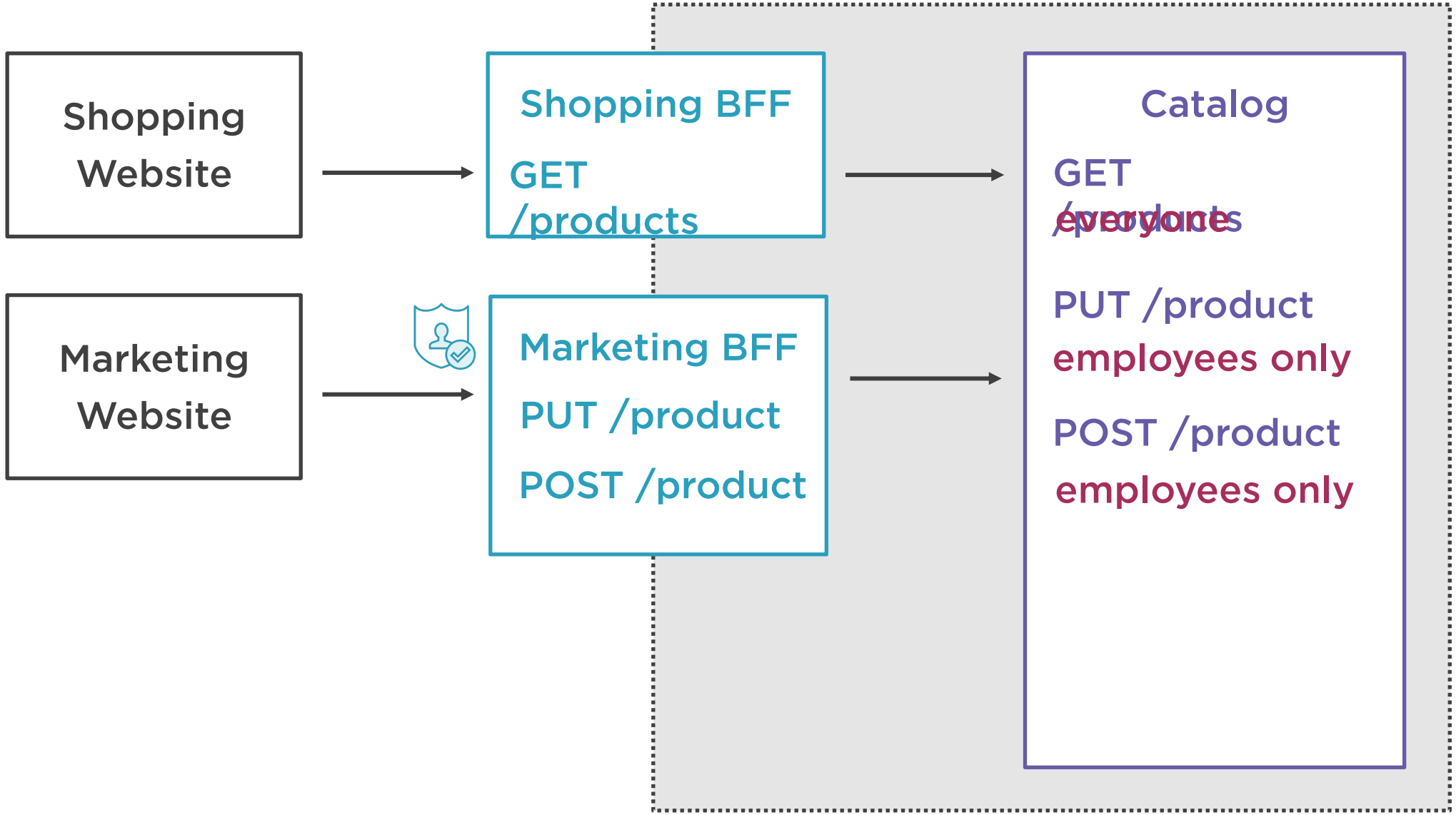
Authentication

Who you are

Authorization

What you are allowed to do

Authorization Example: Catalog API



Sometimes knowing your identity and role is not enough for an authorization decision.

Authorization often also depends on what data you are attempting to access.

Authorization Example: Shipping Address


```
PUT /order/{orderid}

{
   address: {
      street: "12 West St",
      city: "Gullbrough",
      postalCode: "GU1 3DA"
   }
}
```

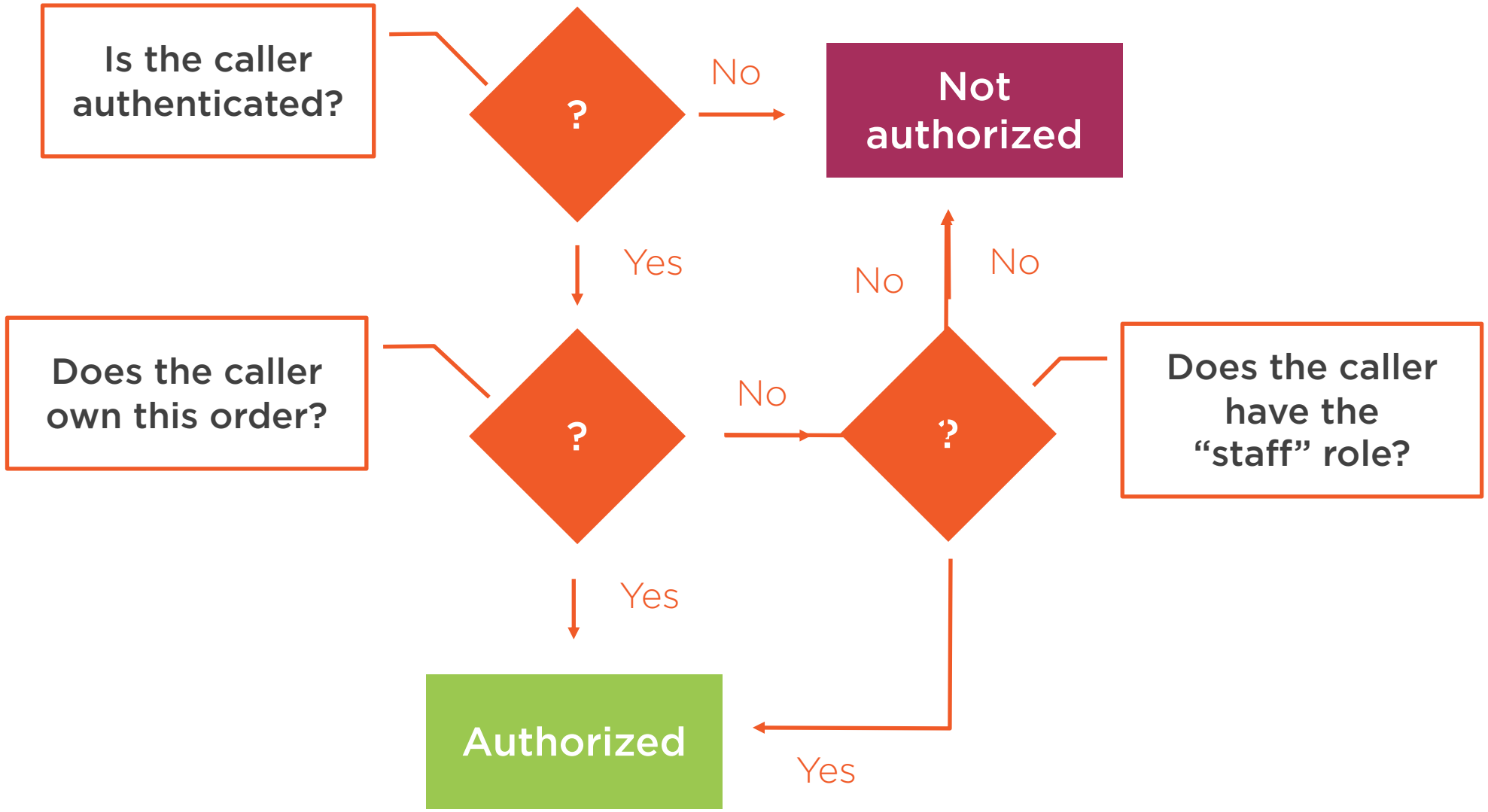
How can we make an authorization decision?

1. **Who is making the request?**

   Identity token can provide this information

2. **Does the order belong to this customer?**

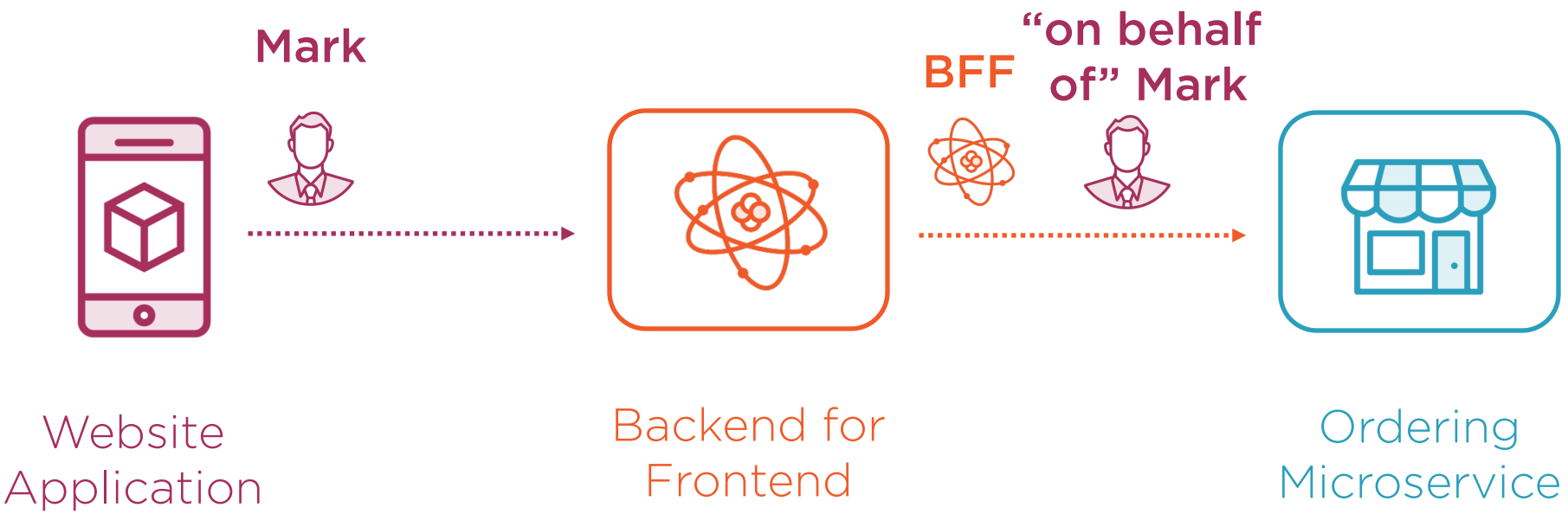
Authorization Logic



Review each API endpoint to ensure sufficient authorization is performed

“On Behalf Of” Requests

Example: Update Shipping Address



"Confused Deputy"

Tricks a microservice into performing an unauthorized action by passing the request through a deputy, losing identity information along the way

Authorizing Messages

Messaging benefits

- Loose coupling, scalability, retries, etc
- Outbound connection only

Who posted this message?

- Carefully guard event bus credentials
- Include identity of end user in the message

Summary

- Authentication: who you are
- Authorization: what you can do
- Basic authentication
- OAuth 2.0 and OpenID Connect
- Identity Server 4
- External identity providers
- Microservice to microservice communications
- “On behalf of” requests

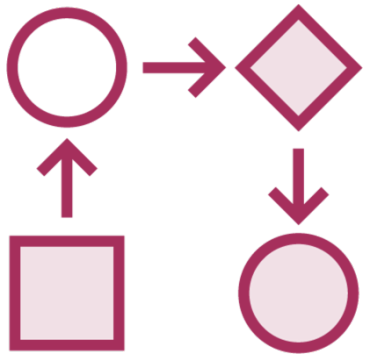
Summary

Authorization decisions

- Who you are
- What roles you have
- What data you are accessing

Defense in depth

Learning More



Structuring Domain Logic

Domain-driven design (DDD)

SOLID principles

Design patterns

Testing Microservices

Unit testing

Mocking frameworks

Authorization and Authentication

OAuth 2.0 and OpenID Connect

OWASP top 10

Thanks for watching