NOVATEC | SPRING IO

Microservices Authentication & Authorization with Spring Security [WORKSHOP]

27.5.2022, Barcelona Spain

Andreas Falk

About Me

Andreas Falk

Novatec Consulting (Germany)

andreas.falk@novatec-gmbh.de

@andifalk

OWASP Open Web Application Security Project

OpenID

NOVATEC

Join **#theperfectflow** in Spain

Being part of novatec feels like...

- Educational Budget
- Transparent Salary Policy
- Fun Budget & Team Activities
- Home-office, Hybrid or Office
- Healthcare Insurance
- Gym Discounts
- Corporate Discounts
- Choose your devices

NOVATEC

Join **#theperfectflow** in Spain

Back-End Software Engineer

Java Kotlin	★★★★☆
spring spring boot	★★★★☆
PostgreSQL mongoDB	★★★★☆
JUnit WireMock	★★★★☆

Full-Stack Software Engineer

Java Kotlin	★★★★☆
Angular React Vue.js	★★★★☆
JUnit WireMock	★★★★☆

NOVATEC

Agenda

- Introduction & Organizational things
- Hands-On / Demo Part
  - OAuth2/OIDC Resource Server
  - Testing
  - Token Propagation
- Questions / Discussion / Feedback

Introduction & Organizational Stuff

Workshop Session Time:

27.5.2022: 11:30 – 13:20

Learning Targets of this Workshop (1)

- OAuth 2.0 & What‘s New in OAuth 2.1
- OpenID Connect (OIDC)
- How to use OAuth/OIDC in Practice
- How to use the new Spring Authorization Server
- Focus: Mainly on Server-Side (No JavaScript)
  - In „Spring Terms“: A Spring Boot Java/Kotlin Backend
  - In „OAuth/OIDC“ Terms: A Resource Server

Learning Targets of this Workshop (2)

- Workshop Steps:
  1. Change Authentication in existing Microservice
     - Basic Auth → JWT (retrieved from Authz Server)
     - Mapping options from JWT to Spring Security Authentication (Auth) & Authorization (Authz)
  2. Test the JWT Auth & Authz
  3. Propagate a JWT to call another Microservice

What we will use in this Workshop

- Spring Security 5.7.x
  https://docs.spring.io/spring-security/reference/index.html
- Spring Authorization Server
  https://docs.spring.io/spring-authorization-server/docs/current/reference/html
- Spring Boot 2.7.0
  - Spring Web
  - Spring Webflux (only for WebClient)
  - Spring Data JPA
  - H2 (In-Memory)
  - Actuator

Workshop Contents (1)

- Complete Source Code of Demos and Labs
  https://tinyurl.com/y3zrm86k

- The Workshop Tutorial (as GitBook)
  https://tinyurl.com/329raj96

Workshop Contents (2)

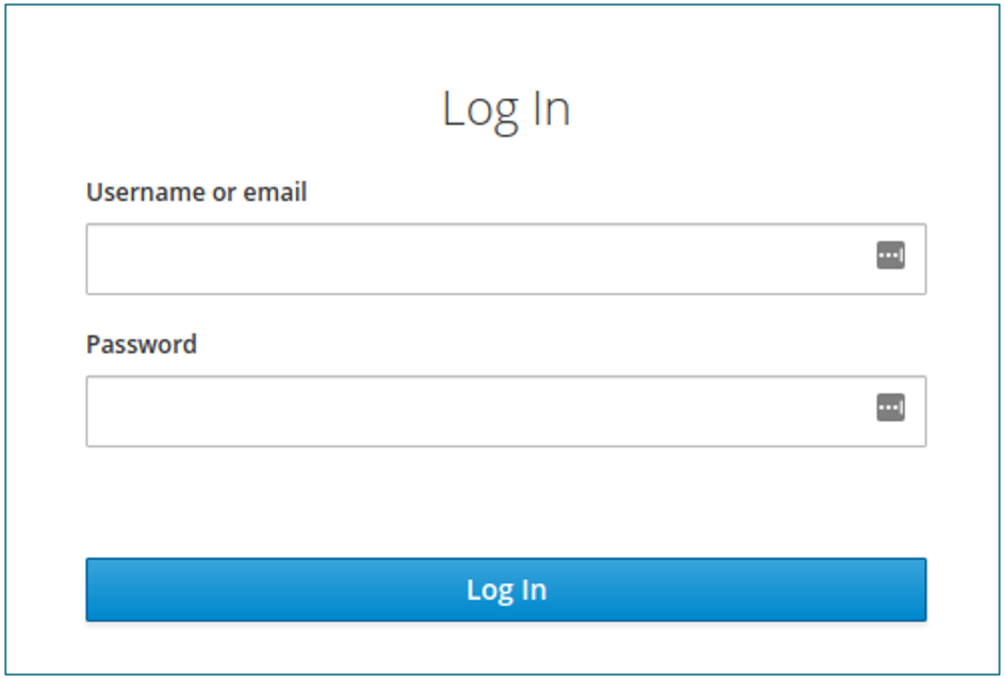
- Customized Spring Authorization Server
  https://tinyurl.com/39ukust9
- Based on Spring Authorization Server
  - Version 0.2.3
- Extended User Object + Predefined Users
- Customized ID+Access Token Contents
- Customized Userinfo Endpoint

OAuth 2.0 (RFC 6749)

https://tools.ietf.org/html/rfc6749

Do you like implementing your own Authentication?

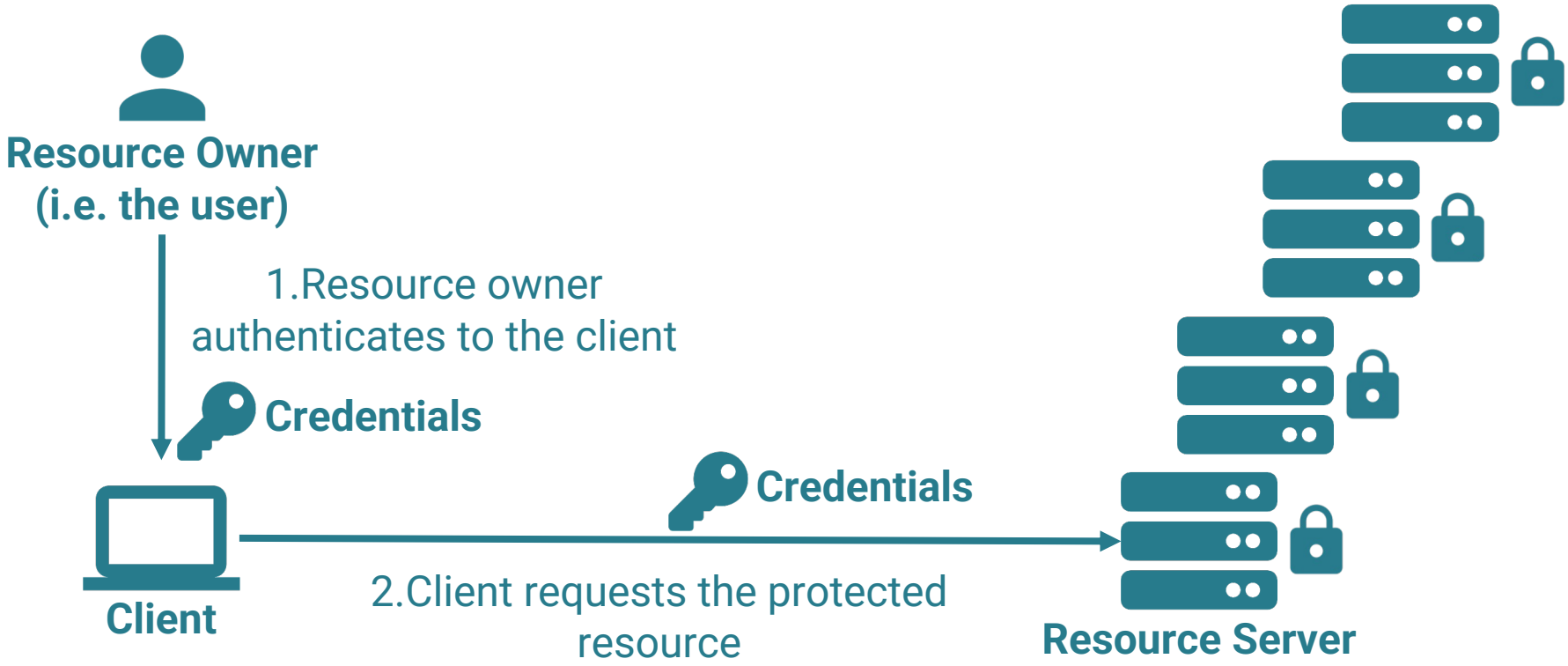
- Different Clients
- Brute Force Prevention
- Reset Password Process



- Password Policies
- MFA
- Secure Password Storage

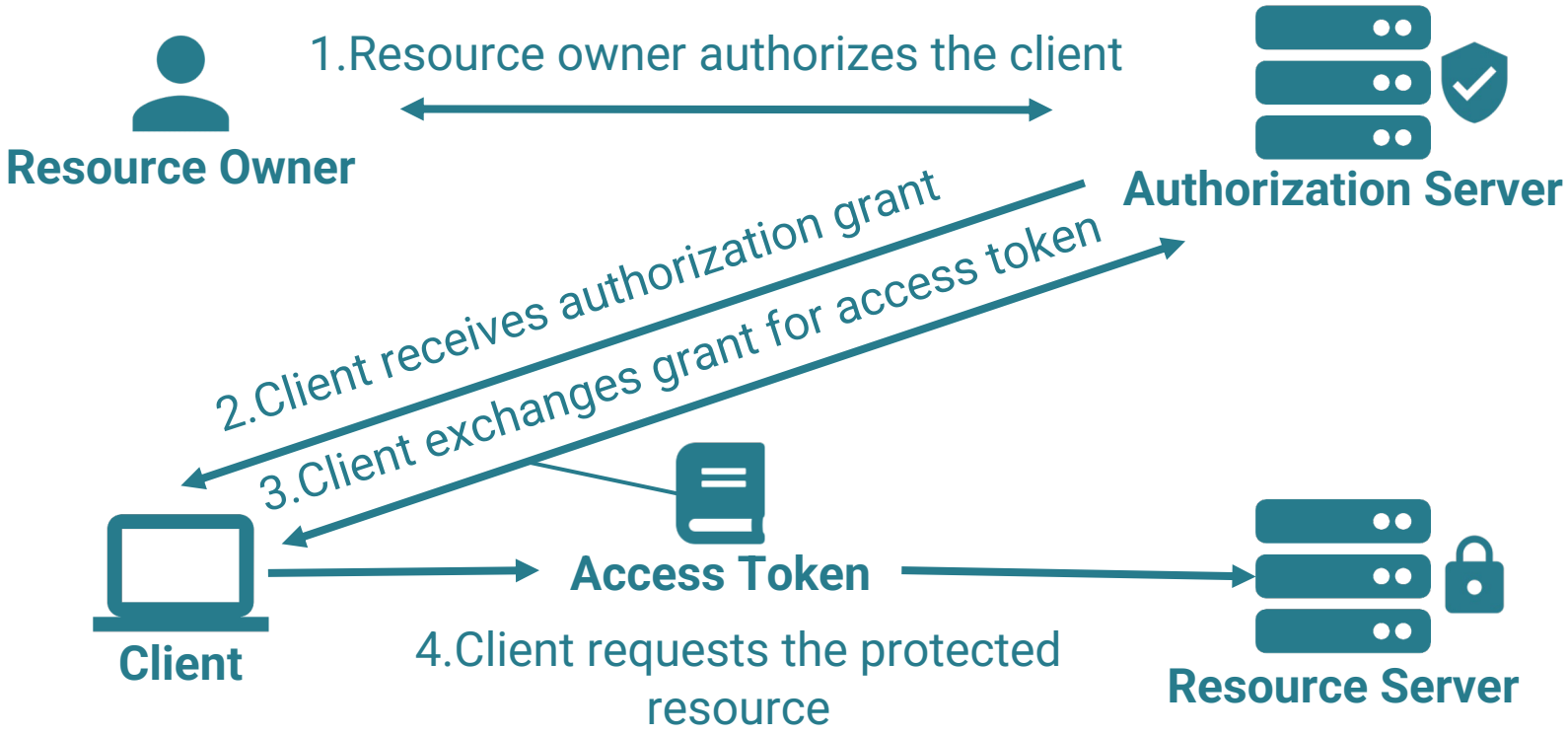
NOVATEC

Spreading Credentials before OAuth 2.0



NOVATEC

Basic OAuth 2.0 Protocol Flow



NOVATEC

OpenID Connect 1.0

https://openid.net/specs/openid-connect-core-1_0.html

NOVATEC

OpenID Connect 1.0 (OIDC)

Based on OAuth 2.0
Additions:

- ID Token (JWT format is mandatory)
- User Info Endpoint (Mandatory)
- Hybrid Grant Flow (Mandatory)
- OpenID Provider Configuration Information (Discovery, Optional)

https://openid.net/specs/openid-connect-core-1_0.html
https://openid.net/specs/openid-connect-registration-1_0.html
https://openid.net/specs/openid-connect-discovery-1_0.html

JSON Web Token (JWT)

JSON Web Tokens consist of three parts separated by dots (“.”), which are:

- Header
- Payload
- Signature

Each part is Base64Url encoded

Signature supports symmetric or asymmetric algorithms (e.g. HMAC or RSA)

Signature = HMACSHA256(
base64UrlEncode(header) + "." + base64UrlEncode(payload), secret)

https://tools.ietf.org/html/rfc7519
https://tools.ietf.org/html/draft-ietf-oauth-jwt-bcp
https://tools.ietf.org/html/draft-ietf-oauth-proof-of-possession

OAuth/OpenID Connect on the Client side

OAuth 2.1 Authorization Grant Types

Resource Owner	Client Type	Authorization Grant	Refresh Token
✓	Web Client (Confidential)	Authorization Code + PKCE	✓
✓	Mobile Client (Public)	Authorization Code + PKCE	✓
✓	SPA Client (Public)	Authorization Code + PKCE	✓
✗	Resource Owner = Client (Confidential)	Client Credentials	✗

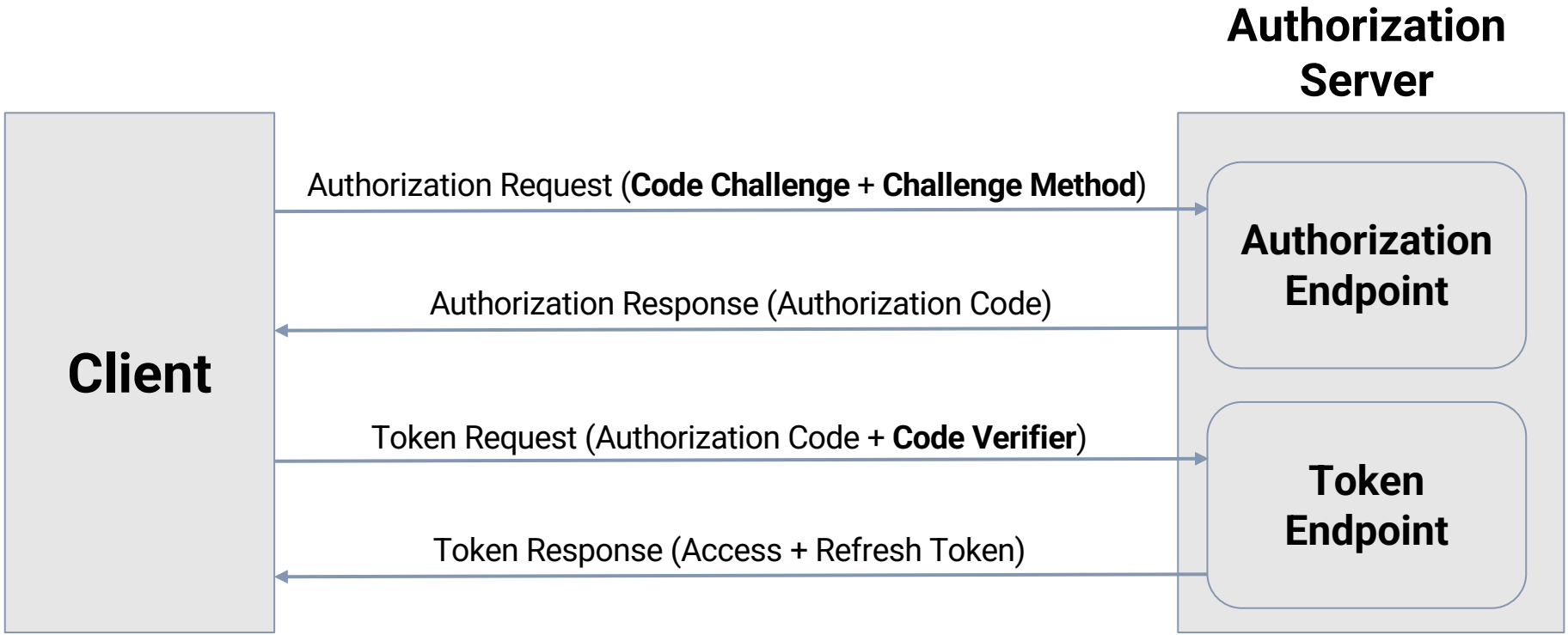
https://datatracker.ietf.org/doc/rfc6749
https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1

OAuth 2.1: Browser-Based Applications (SPA)

- OAuth 2.1: New recommendations for SPA
  - ✓ Do Not use Implicit flow any more
  - ✓ Instead use Authorization Code + PKCE
  - ✓ Restrict Redirect URIs (No Wildcards)
  - ✓ Use Secure Architecture Patterns
- Use OpenID Connect for Authentication
  - ✓ ID-Token (Client), Access-Token (Server)

OAuth Happy Hour - Authorization Code Injection Demo
https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1
https://datatracker.ietf.org/doc/draft-ietf-oauth-browser-based-apps
JSON Web Token Best Current Practices (RFC 8725)

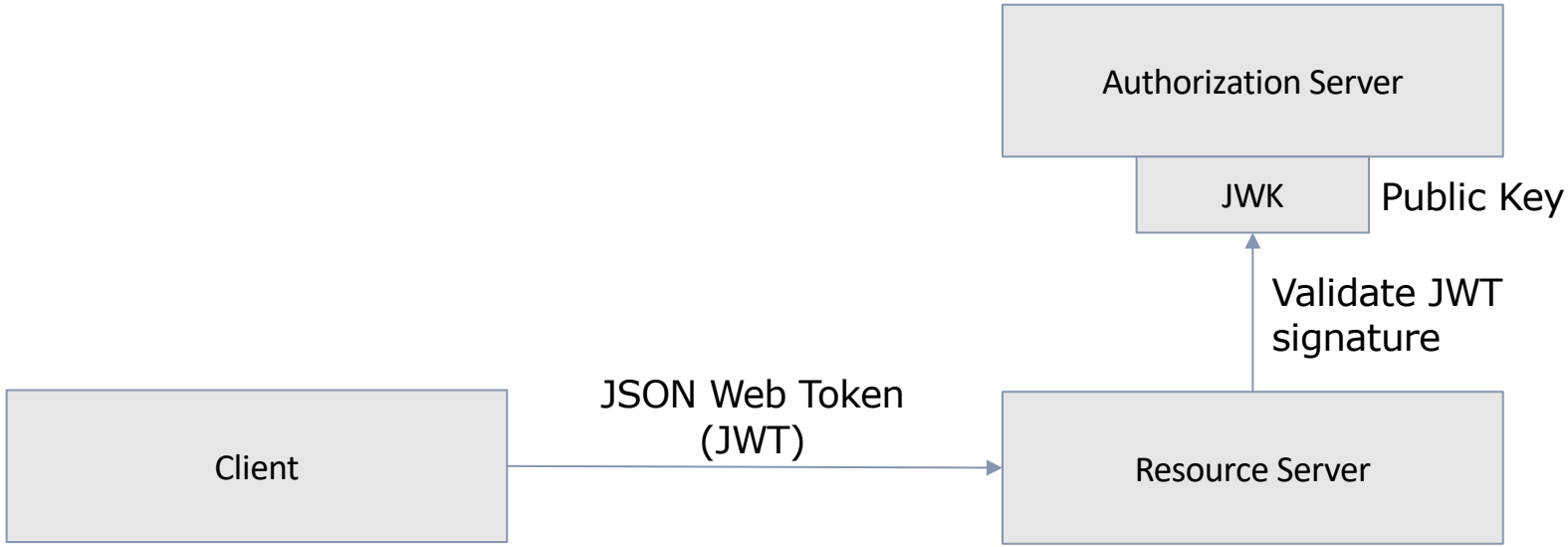
Authorization Code + PKCE



https://tools.ietf.org/html/rfc7636

OAuth/OpenID Connect on the Server side (Resource Server)

Validate Authentication in a Resource Server




```
GET / HTTP/1.1
Host: localhost:8080
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1N...
```

Spring Security Basics & Spring Authorization Server

https://spring.io/projects/spring-security
https://spring.io/projects/spring-authorization-server

Spring Security 5

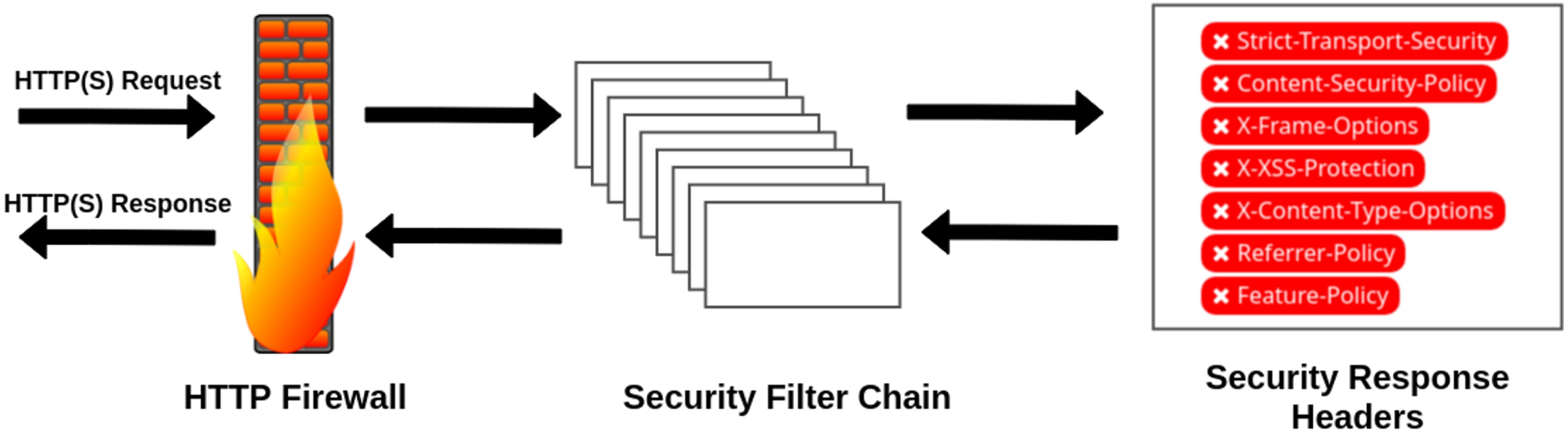
Authentication & Authorization
Password Encoding
Support for Servlet & Reactive Web Applications
Support for OAuth 2.0
Support for OpenID Connect 1.0, JWT and JOSE (JWS/JWK)
Testing Support for Auth/Authz/JWT/OAuth

https://docs.spring.io/spring-security/reference/index.html

Spring Security 5 - Secure by Default

Authentication required for all HTTP endpoints
Session Fixation Protection
Session Cookie (HttpOnly, Secure)
CSRF Protection
Security Response Header

Spring Security High Level View



Spring Authorization Server

- Based on Spring Security
- Implements Specifications of
  - OAuth 2.1 (no implicit and password ro grant flows)
  - OpenID Connect 1.0
- Token Support
  - JWT (with JWKS)
  - Opaque (with Introspection)
- Has a Reference Documentation since release 0.3.0
  https://docs.spring.io/spring-authorization-server/docs/current/reference/html

Let‘s start with Practicing

https://github.com/andifalk/microservices-auth-authz-spring-security
https://andifalk.gitbook.io/microservices-authentication-and-authorization