

Diving Under the Hood of Spring Security Authentication



Wojciech Lesniak

AUTHOR

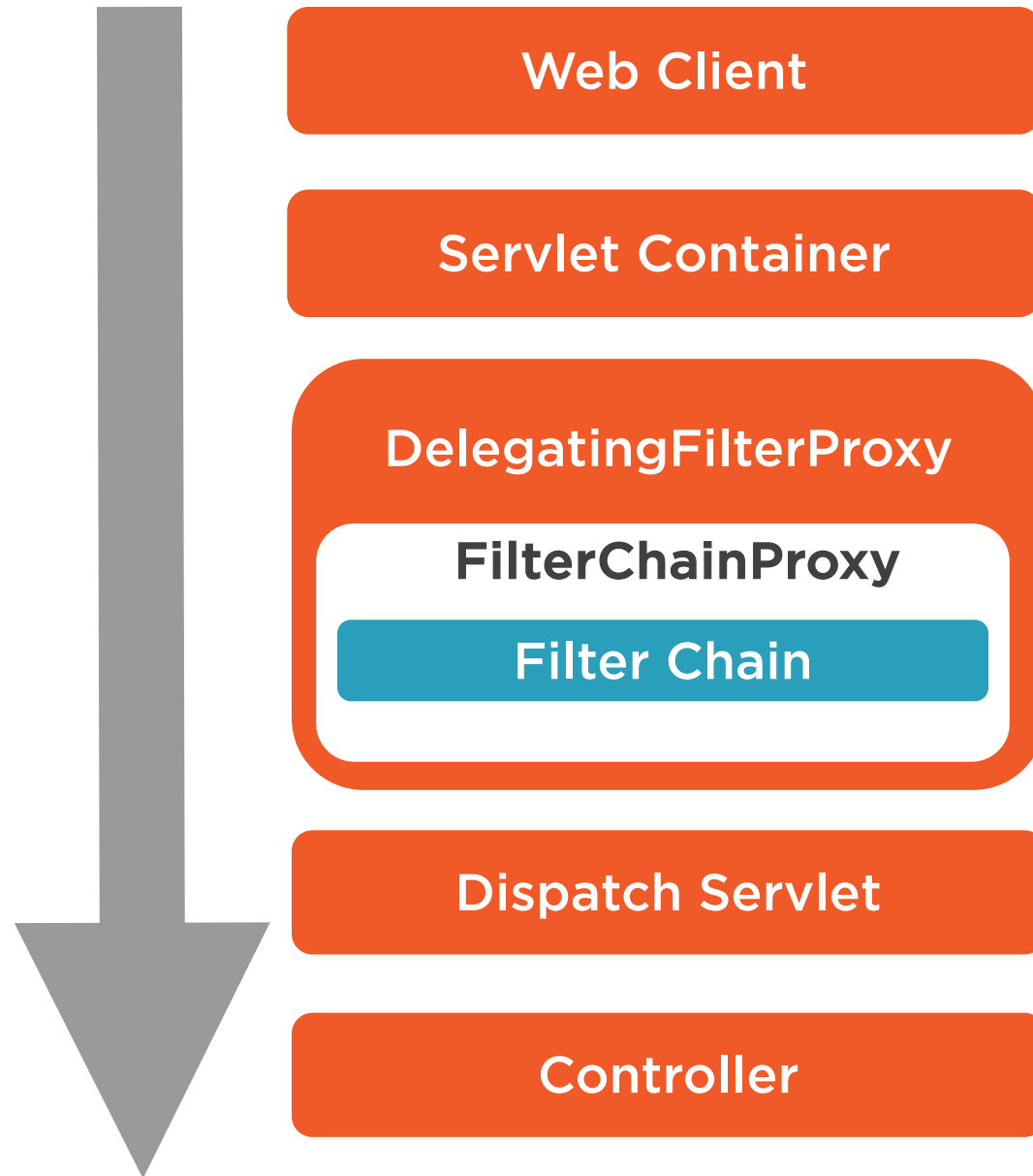
@voit3k



Authentication

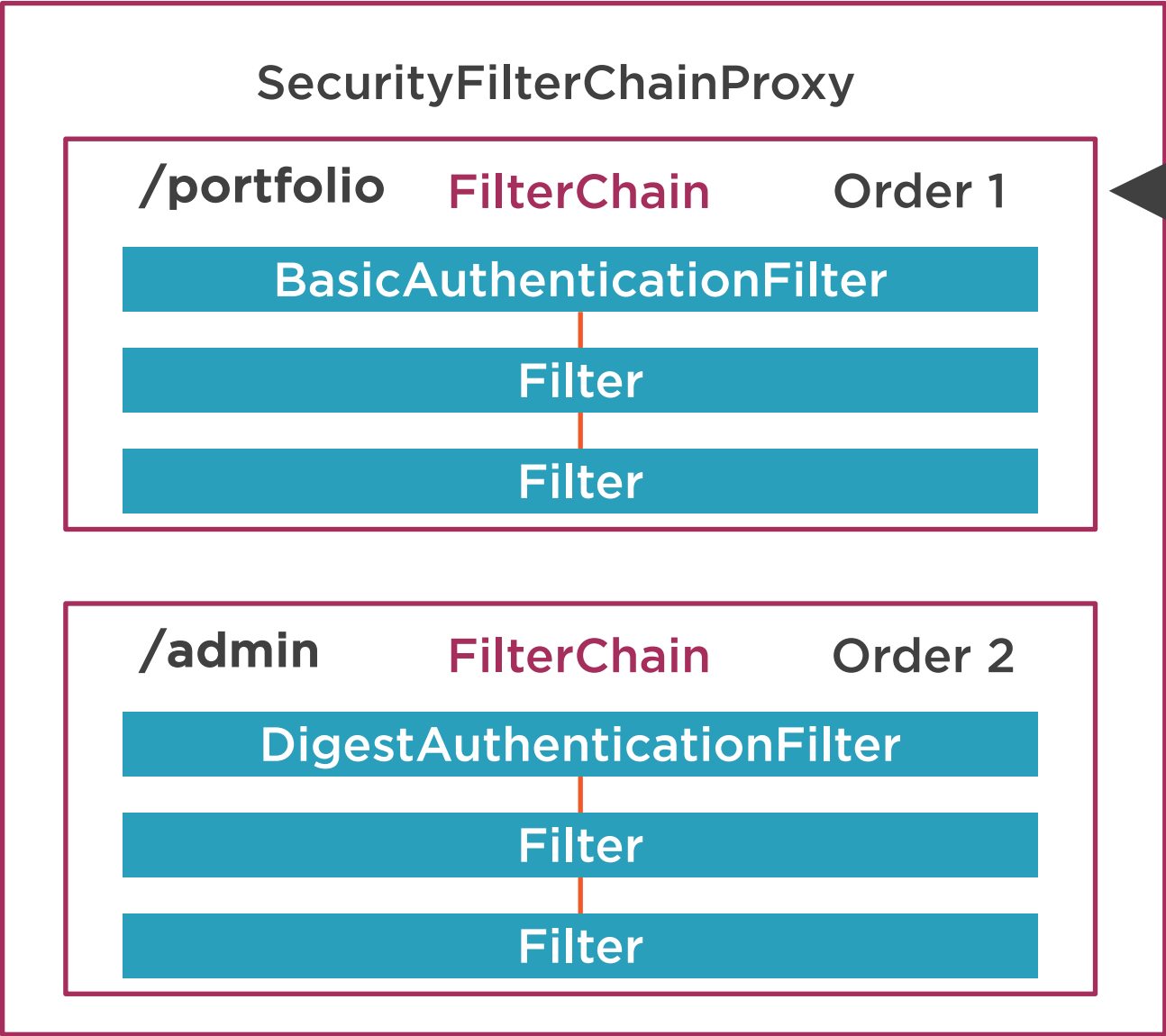


Request

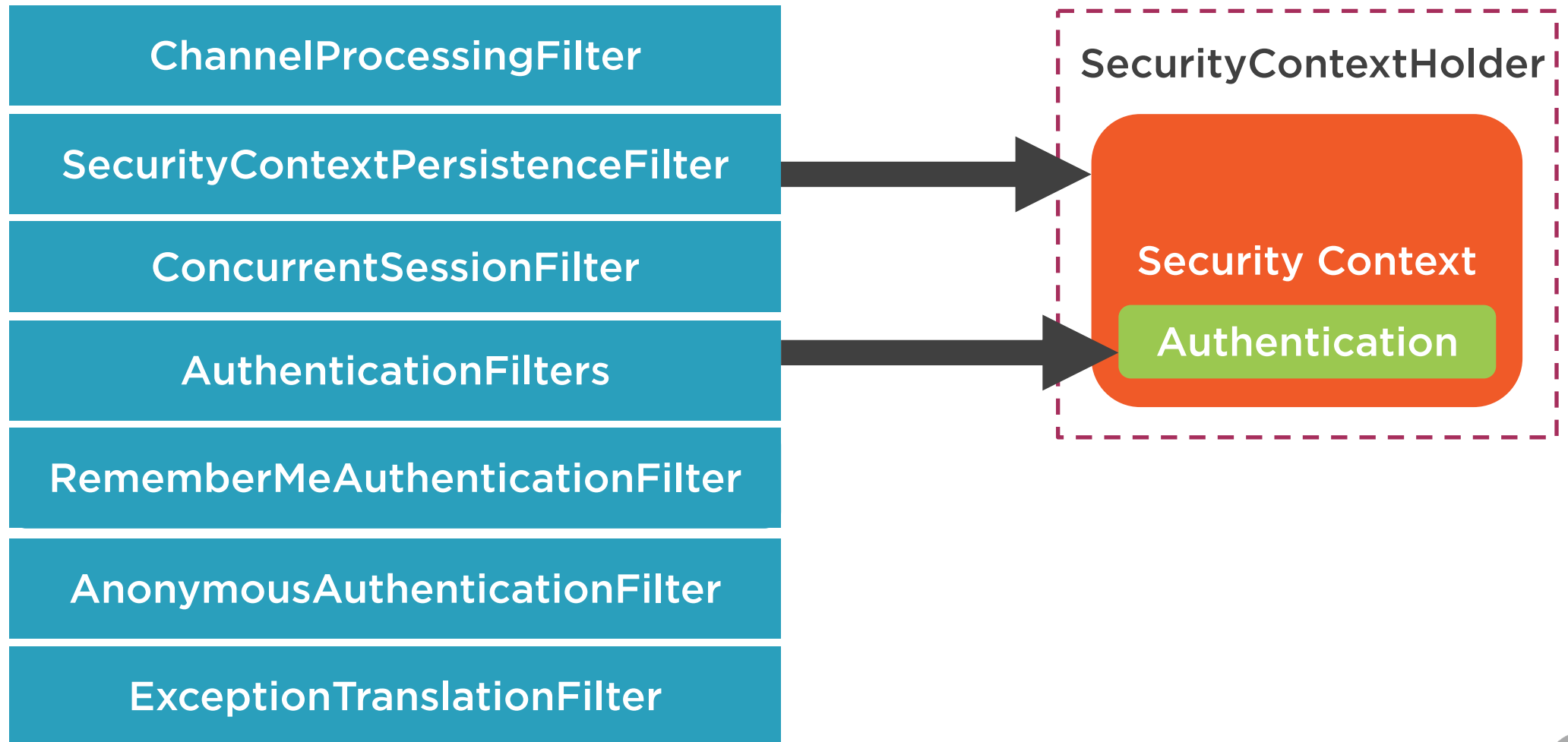


Security Filter Chain Proxy

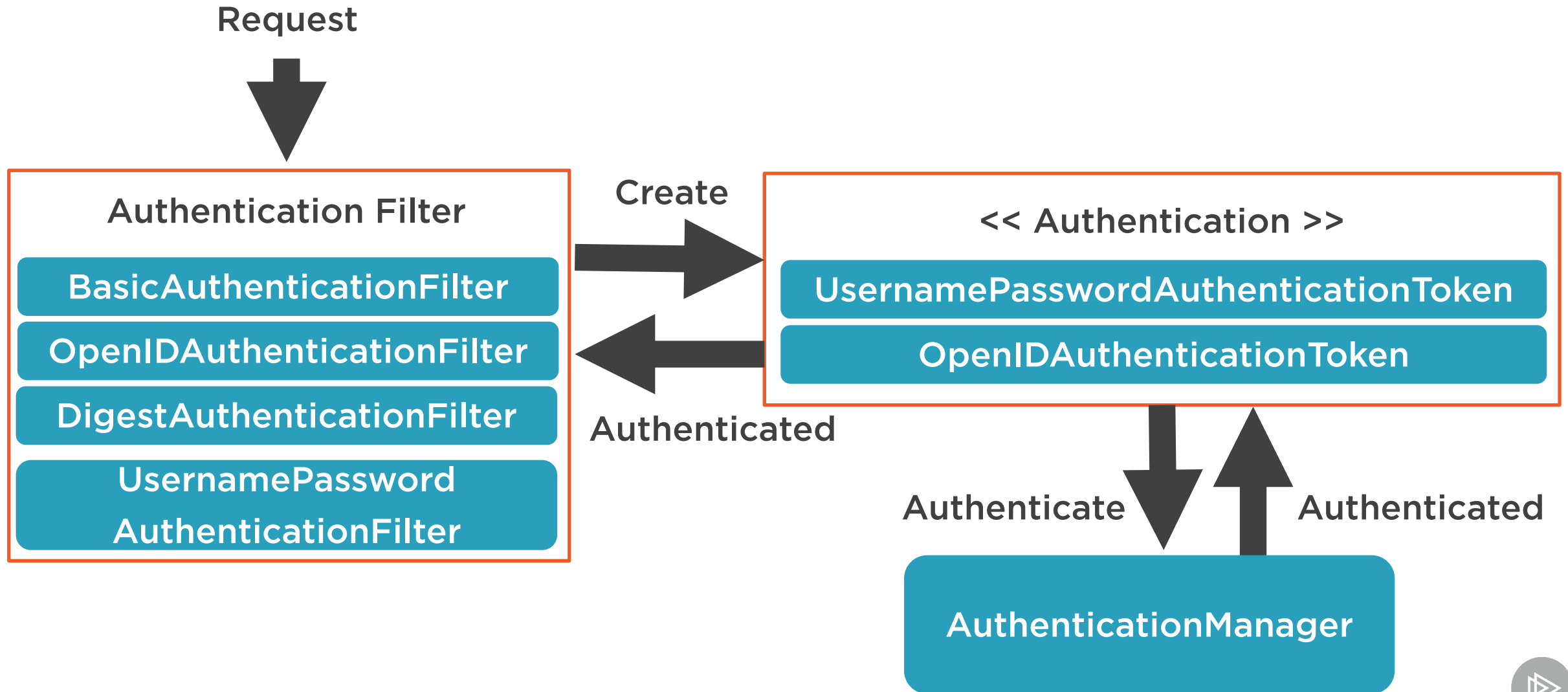




Filter Chain



Authentication



AuthenticationManager Interface

```
public interface AuthenticationManager {  
  
    Authentication authenticate(Authentication authentication)  
        throws AuthenticationException;  
  
}
```



Authentication Interface

```
import java.security.Principal

public interface Authentication extends Principal, Serializable {
    Collection<? extends GrantedAuthority> getAuthorities();
    Object getCredentials();
    Object getPrincipal();
}
```



<< Authentication >>

Authentication request

Authenticated: False

Principle: Username

Credentials: Password

Authorities:

Authenticated principle

Authenticated: True

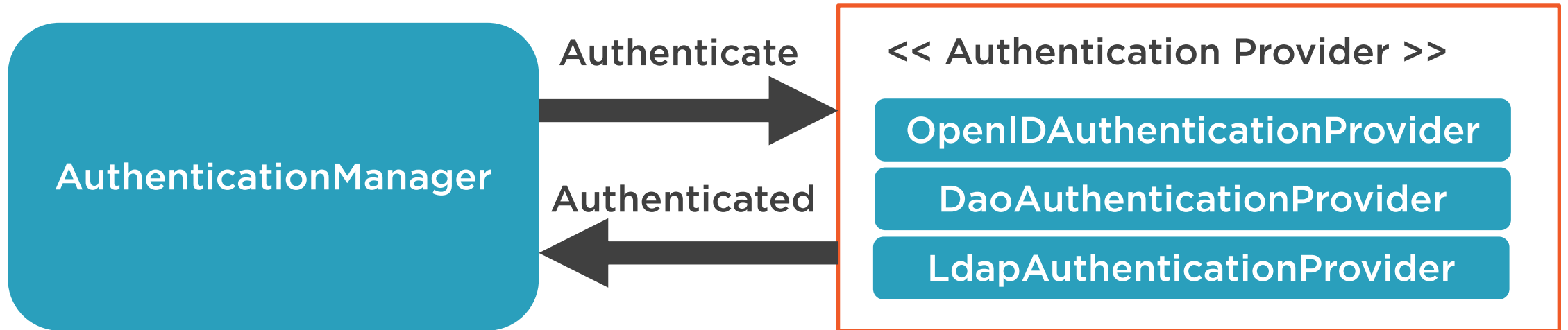
Principle: UserDetails

Credentials:

Authorities: Roles e.g. ADMIN



Authentication

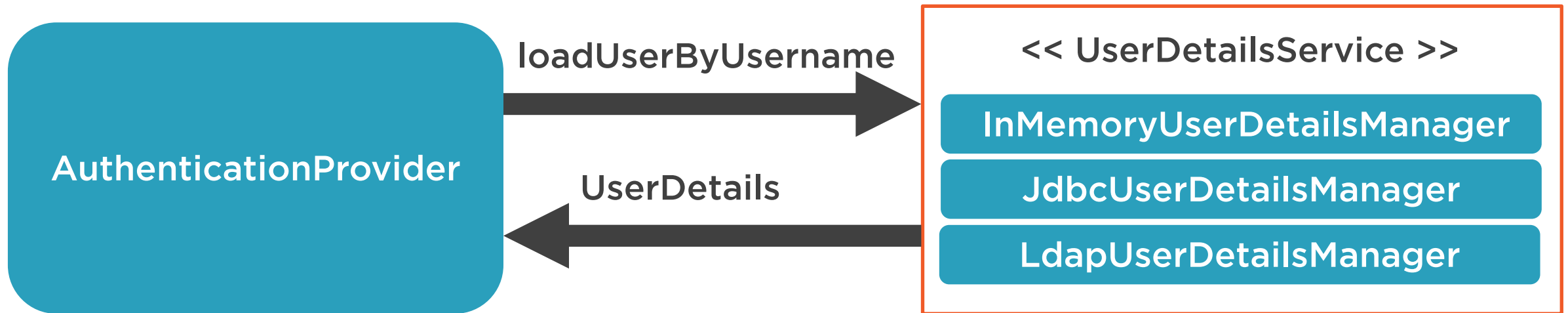


AuthProvider Interface

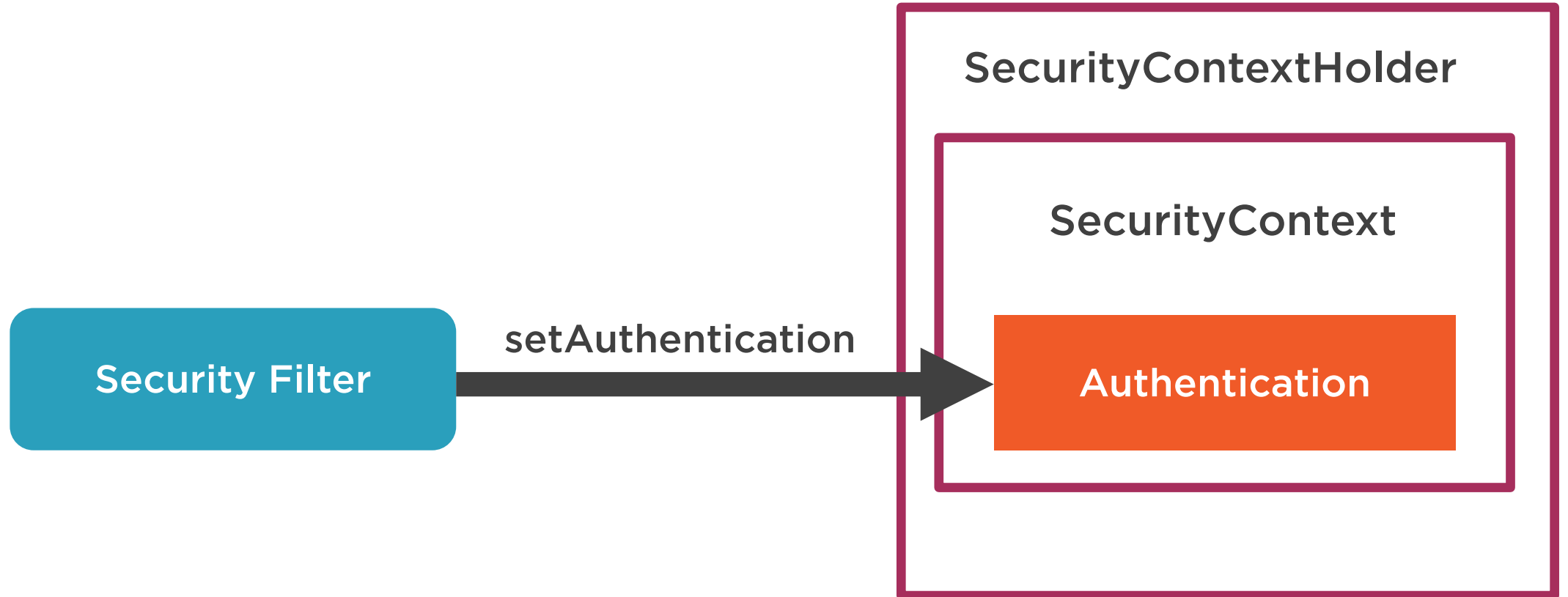
```
public interface AuthProvider {  
  
    Authentication authenticate(Authentication authentication)  
        throws AuthenticationException;  
  
    boolean supports(Class<?> authentication);  
}
```



Authentication Provider



Security Context Holder



SecurityContext

```
public interface SecurityContext extends Serializable {  
  
    void setAuthentication(Authentication authentication);  
    Authentication getAuthentication();  
}
```



Summary

An *Authentication Filter* generates an *Authentication Request* and calls the *Authentication Manager* to authenticate

The *Authentication Manager* delegates authentication to the *Authentication Provider(s)*

The *Authentication Provider* requests the *UserDetails* from the *UserDetailsService* and adds it the *Authentication*

Then the *Authentication Principle* is added to the *SecurityContext* by the *Authentication Filter*



Spring Security with Spring Boot



Version management



Automatically configures security filter chain



Handles a lot of the unnecessary boilerplate code and configuration



Demo



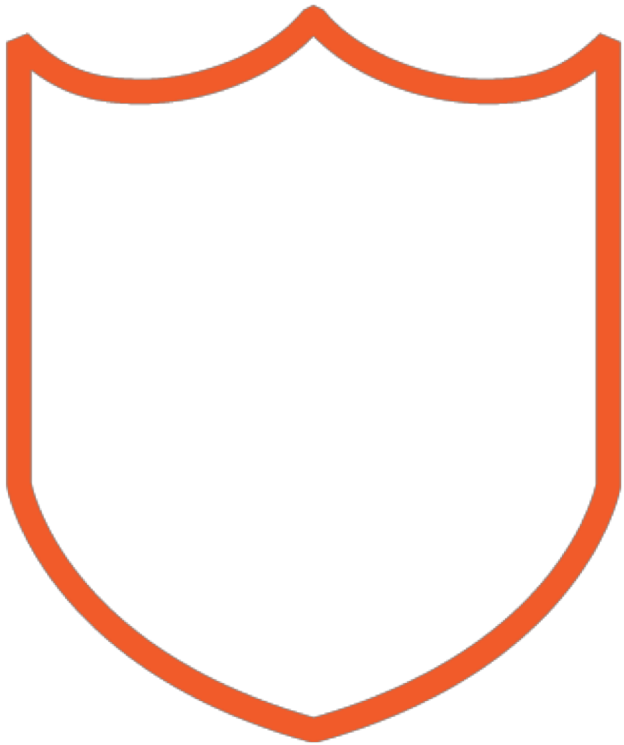
Configuring Spring Security with Spring Boot

Default form login



Basic Authentication





Used with stateless clients

Not secure

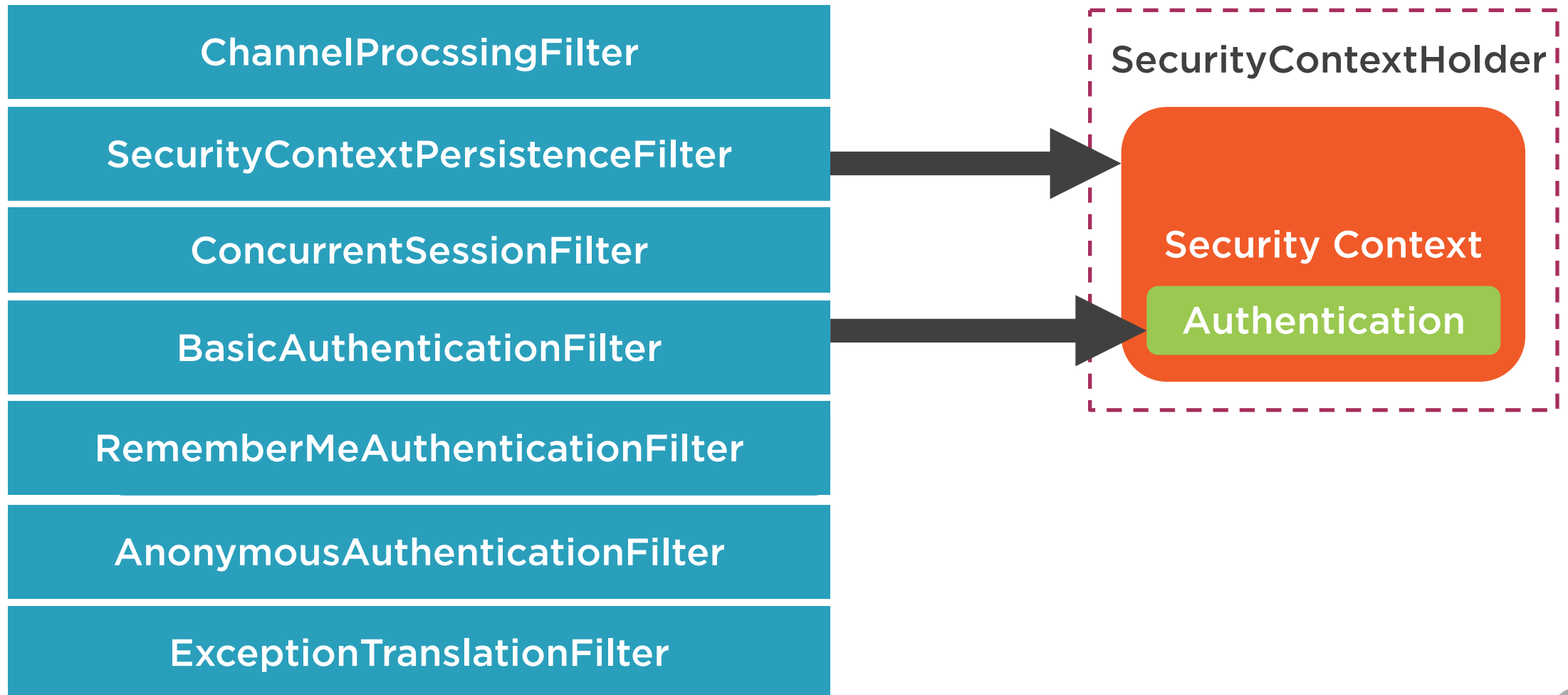
- Username / password transmitted in plain text

Spring implementation compatible with RFC 1945, Section 11

Credentials transmitted in the header

- Header name: Authentication
- Header value: Basic + Base64(username:password)
- E.g. Authorization: Basic Ym9iOnBhc3N3b3Jk

Security Filters



Demo



Configure Basic Authentication



Digest Authentication





A hash of the credentials is transmitted

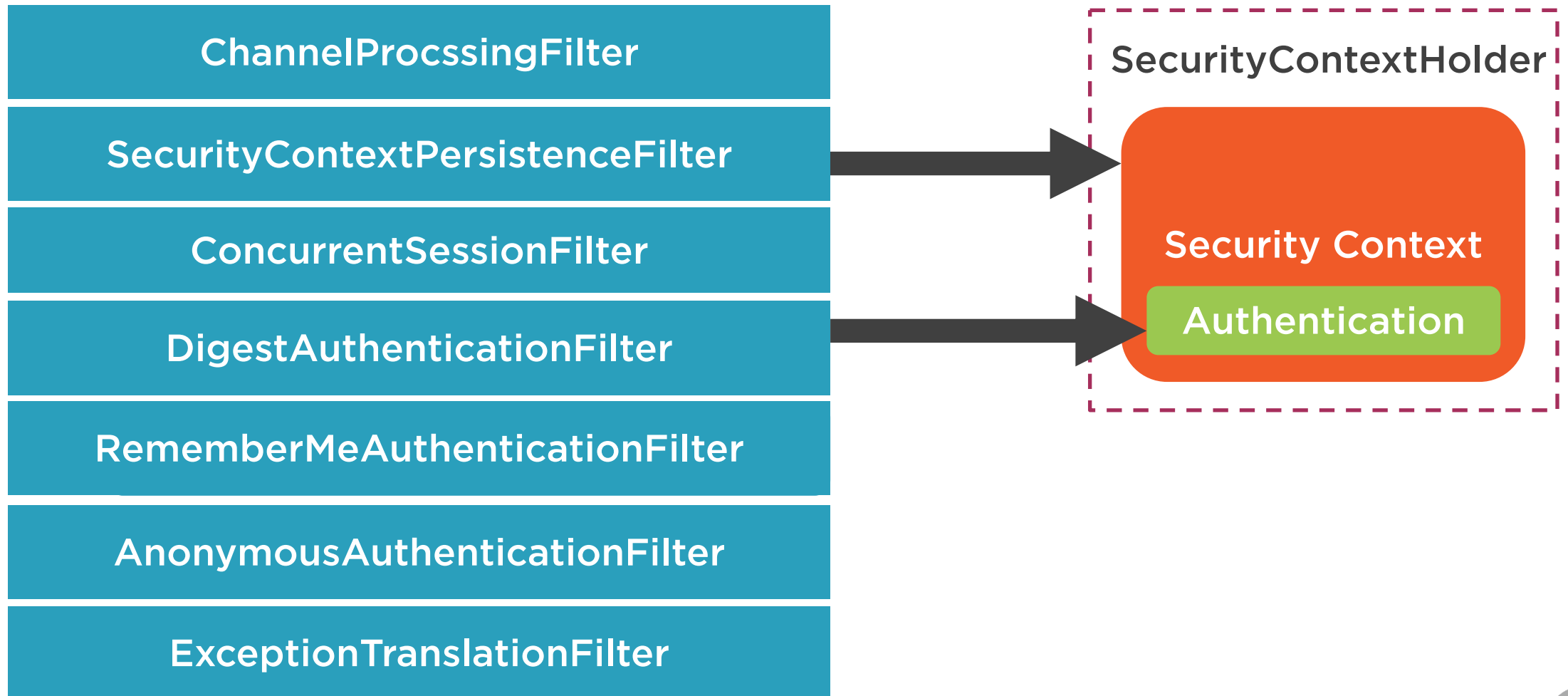
Spring implementation compatible with RFC 2617 is backward compatible with RFC 2069

Prevents phishing

Susceptible



Security Filters



Demo



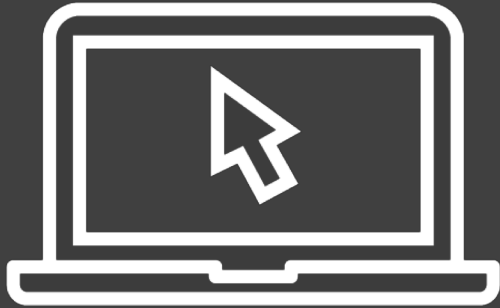
Configure Digest Authentication



You should not use Digest in modern applications because it is not considered secure. The most obvious problem is that you must store your passwords in plaintext, encrypted, or an MD5 format. All of these storage formats are considered insecure. Instead, you should use a one way adaptive password hash (i.e. bCrypt, PBKDF2, SCrypt, etc).



Demo



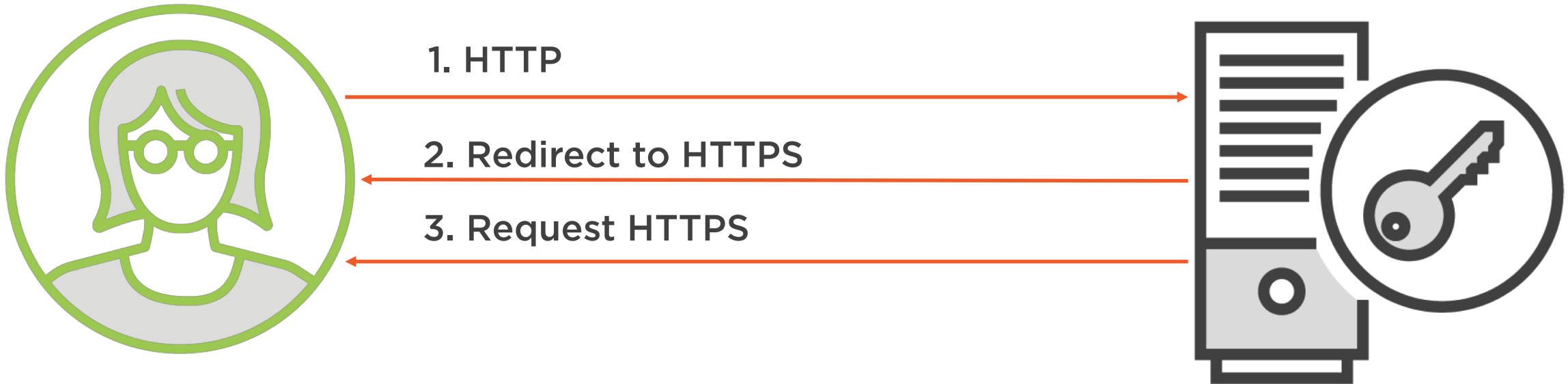
Cracking Digest
Authentication



HTTP Strict Transport Security (HSTS)



HSTS Flow




Demo




Configure HTTPS and HSTS



Troy







What Every Developer Must Know About HTTPS

Products Resource Plans Contact Us Login Sign Up

by Troy Hunt

HTTPS is an essential component of any software running on the web. This course teaches developers how to get their apps talking securely over the web, while avoiding the common pitfalls so many sites fall victim to.

 Resume Course

 Bookmark




 Add to Channel

Table of contents Description Transcript Exercise files Discussion Learning Check Recommended

Expand all


 Course Overview


✓



1m 50s


▼


 The HTTPS Value Proposition



38m 1s

▼

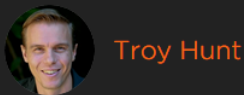
 HTTPS Fundamentals



28m 12s

▼

Course author



Troy Hunt is a Microsoft Regional Director and MVP for Developer Security. He's a regular conference speaker, frequent blogger at troyhunt.com and is the creator of the data breach notification...

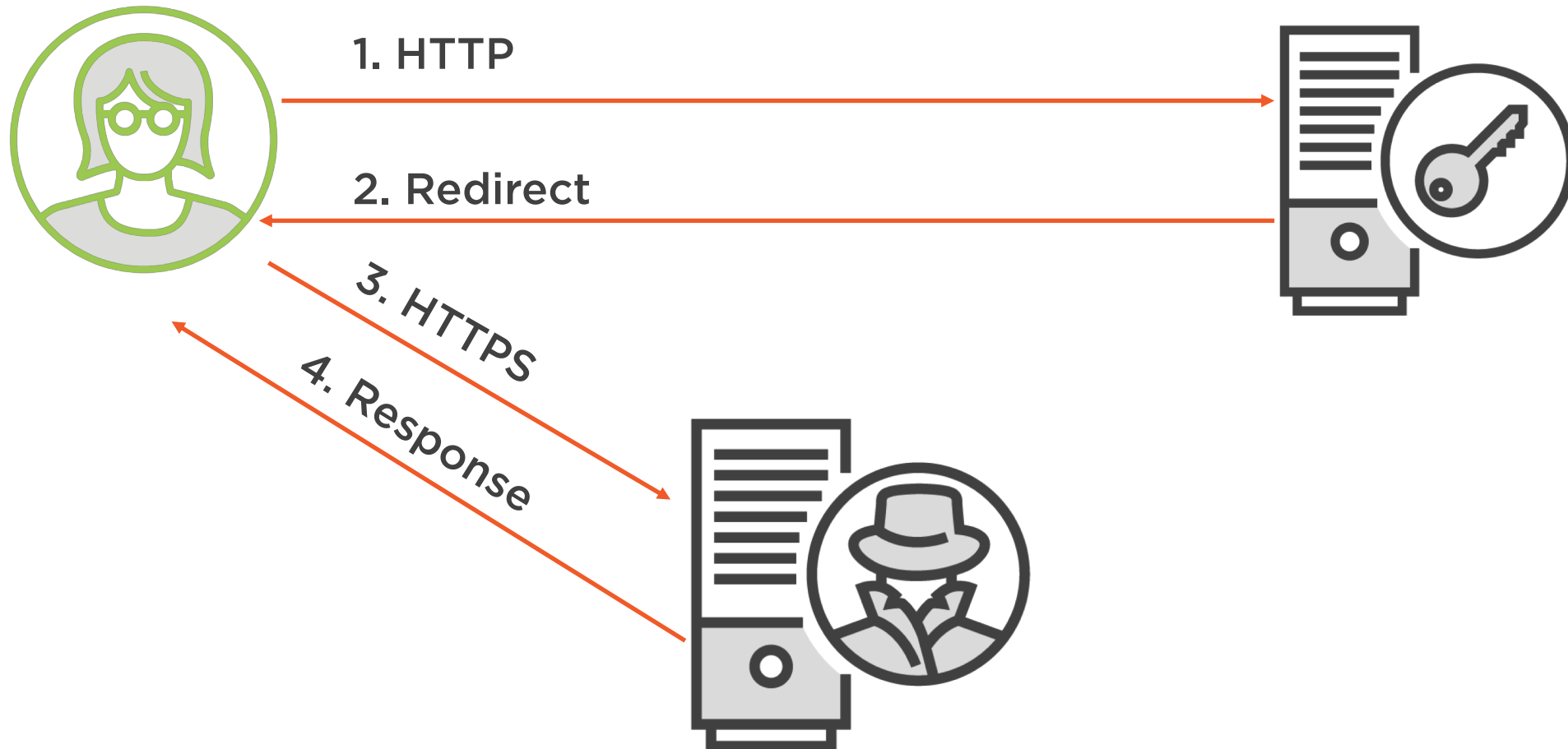
Course info

Level	Beginner
Rating	★★★★★ (204)
My rating	★★★★★
Duration	3h 24m
Released	12 Apr 2017

Share course



Man-in-the-Middle (MITM)



Summary



Authentication

- Filters
- Authentication manager.
- UserDetailsService / UserDetails
- SecurityContext

Basic and Digest Authentication

Spring Boot

HSTS, HTTPS

