spring I/O

BCN2017

Architecture Deep Dive in Spring Security

Joe Grandja
@joe_grandja
github.com/jgrandja

3 Key Areas in Security

- Authentication
- Authorization
- Exception Handling

User Database

Username	Password	Authorities
joe@example.com	password	ROLE_USER
rob@example.com	password	ROLE_USER
admin@example.com	password	ROLE_USER, ROLE_ADMIN

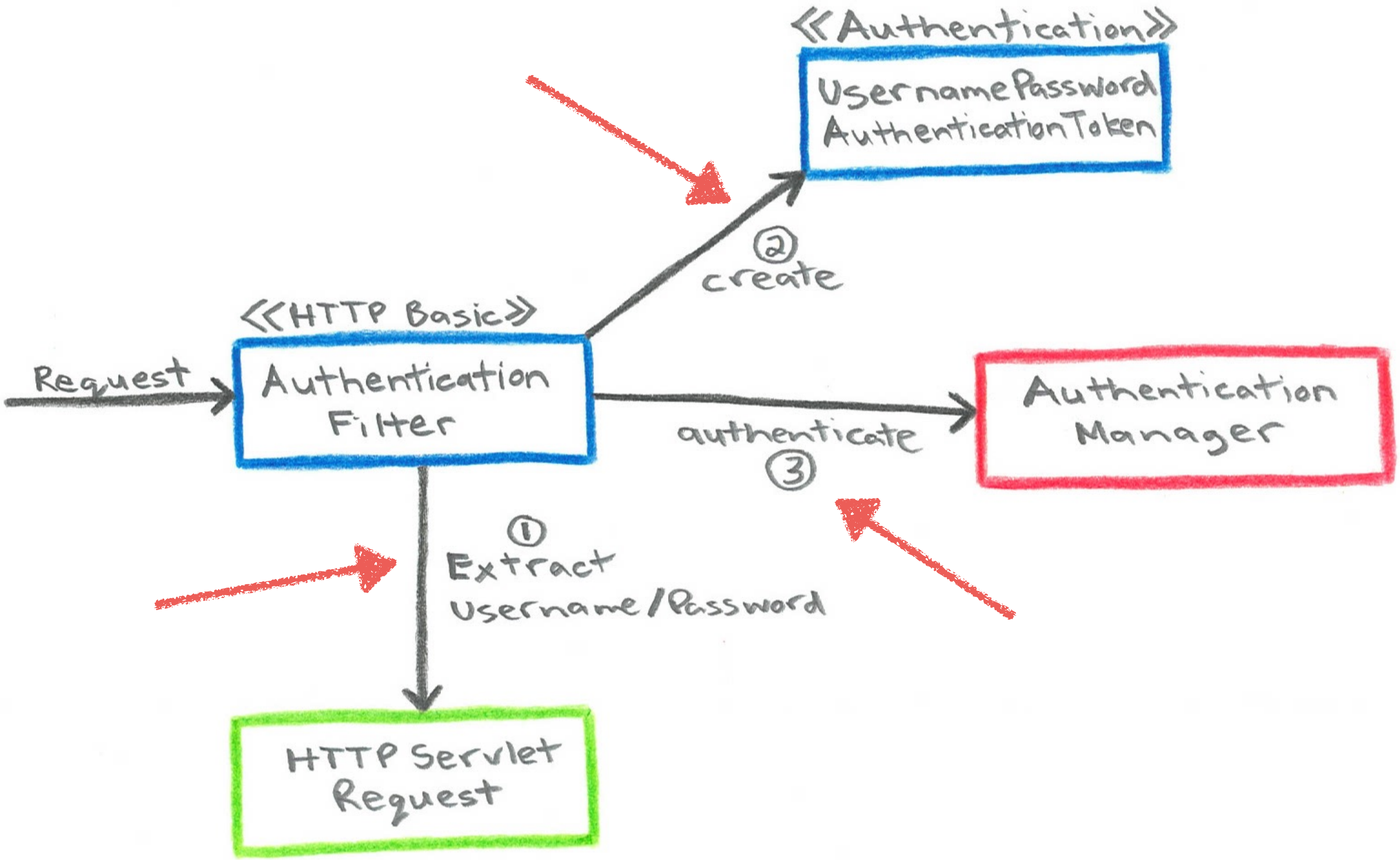
DEMO

AUTHENTICATION

AUTHORIZATION

EXCEPTION HANDLING

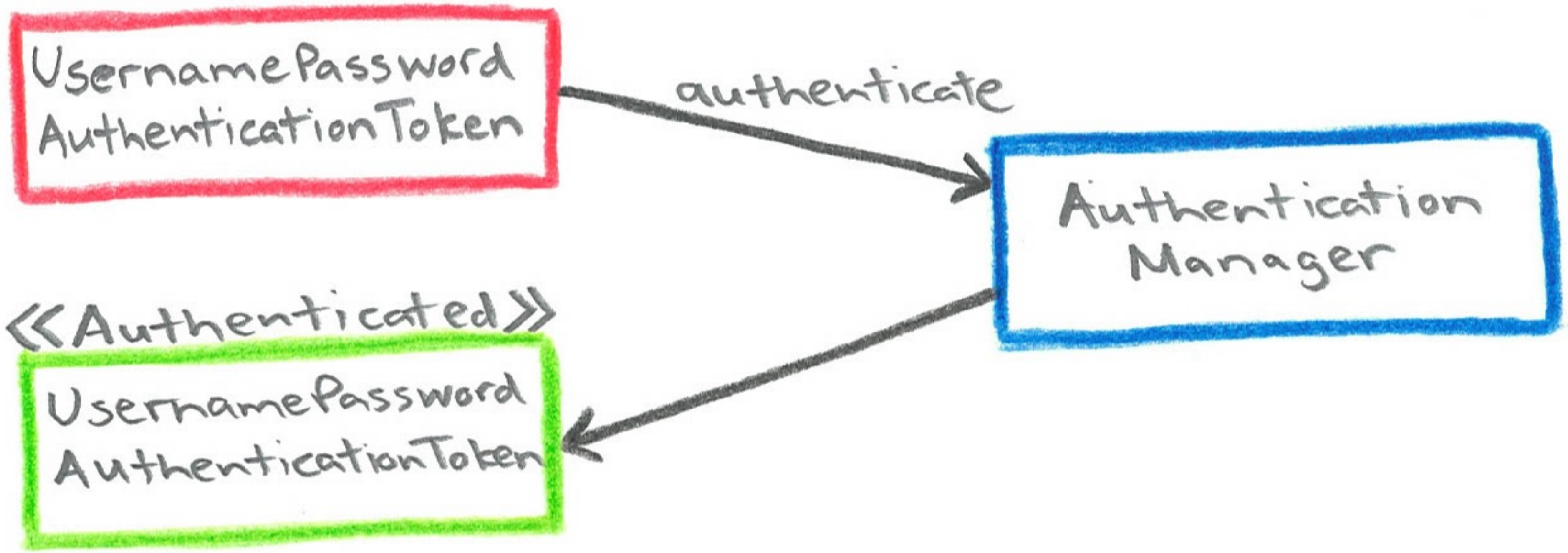
Authentication Filter



Authentication

Authentication	
Principal:	joe@example.com
Credentials:	password
Authorities:	——
Authenticated:	FALSE

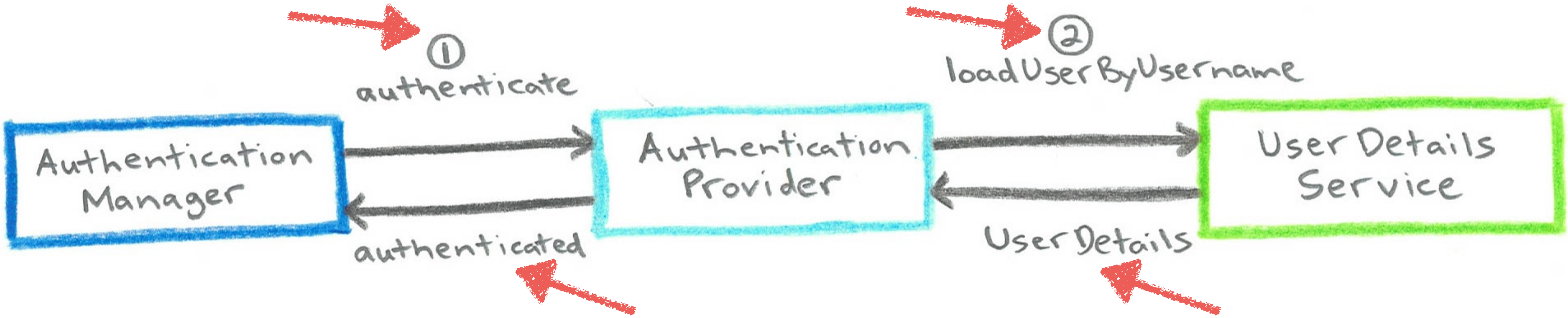
Authentication	
Principal:	UserDetails
Credentials:	——
Authorities:	ROLE_USER
Authenticated:	TRUE


UsernamePassword AuthenticationToken
authenticate
Authentication Manager
«Authenticated»
UsernamePassword AuthenticationToken

```java
public interface Authentication extends Principal, Serializable {

    Object getPrincipal();

    Object getCredentials();

    Collection<? extends GrantedAuthority> getAuthorities();

    . . .
}
```

UserDetails / Service

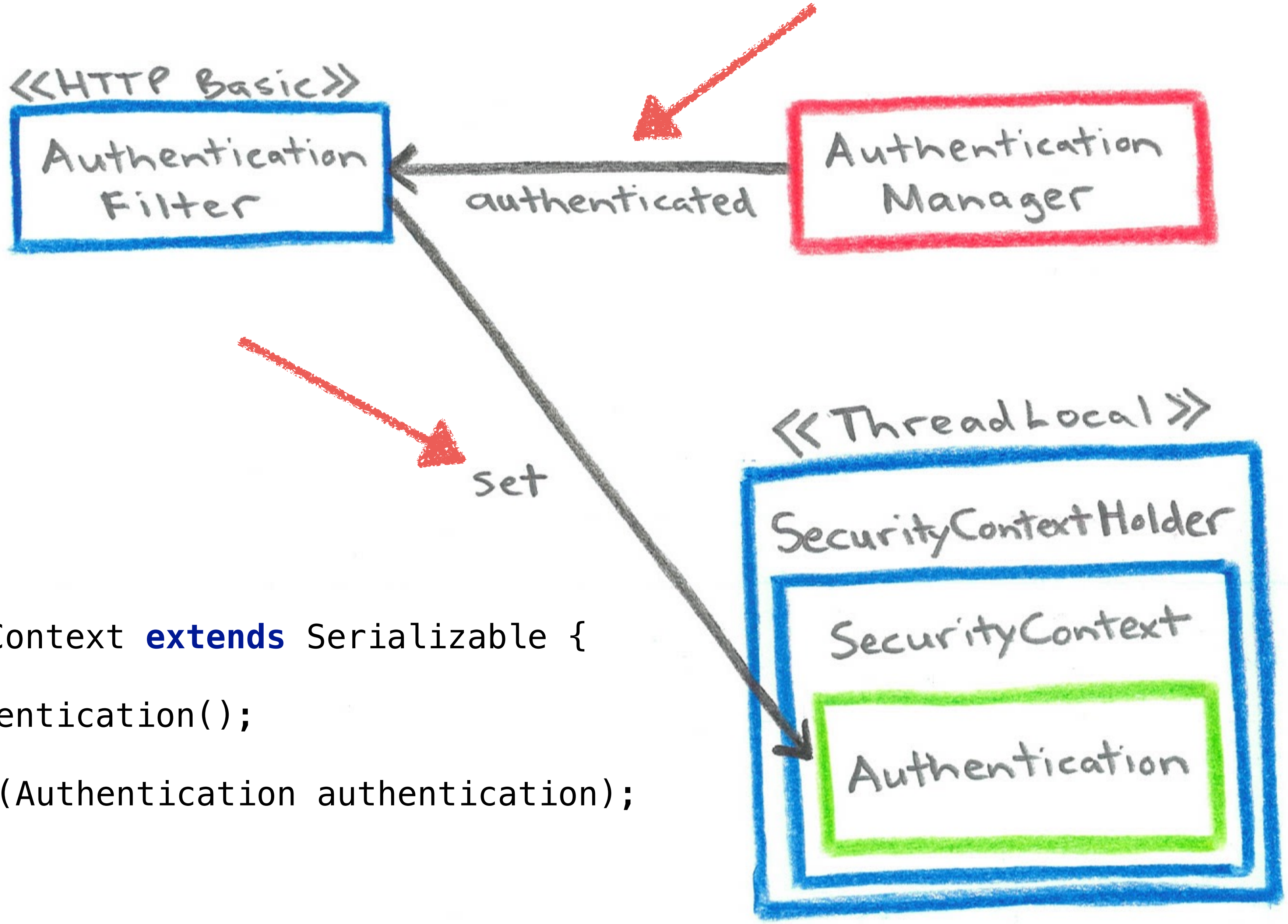
```
public interface UserDetailsService {

    UserDetails loadUserByUsername(String username)
          throws UsernameNotFoundException;

}
```



Authentication	
Principal:	UserDetails
Credentials:	——
Authorities:	ROLE_USER
Authenticated:	TRUE

```
public interface UserDetails extends Serializable {

    String getUsername();

    String getPassword();

    Collection<? extends GrantedAuthority> getAuthorities();

    . . .
}
```

Security Context



```
public interface SecurityContext extends Serializable {

    Authentication getAuthentication();

    void setAuthentication(Authentication authentication);

}
```

```
SecurityContextHolder.getContext().setAuthentication(authenticated);
```

Authentication Recap

- **Authentication Filter** creates an “*Authentication Request*" and passes it to the **Authentication Manager**
- Authentication Manager delegates to the **Authentication Provider**
- Authentication Provider uses a **UserDetailsService** to load the **UserDetails** and returns an “*Authenticated Principal*”
- Authentication Filter sets the **Authentication** in the **SecurityContext**

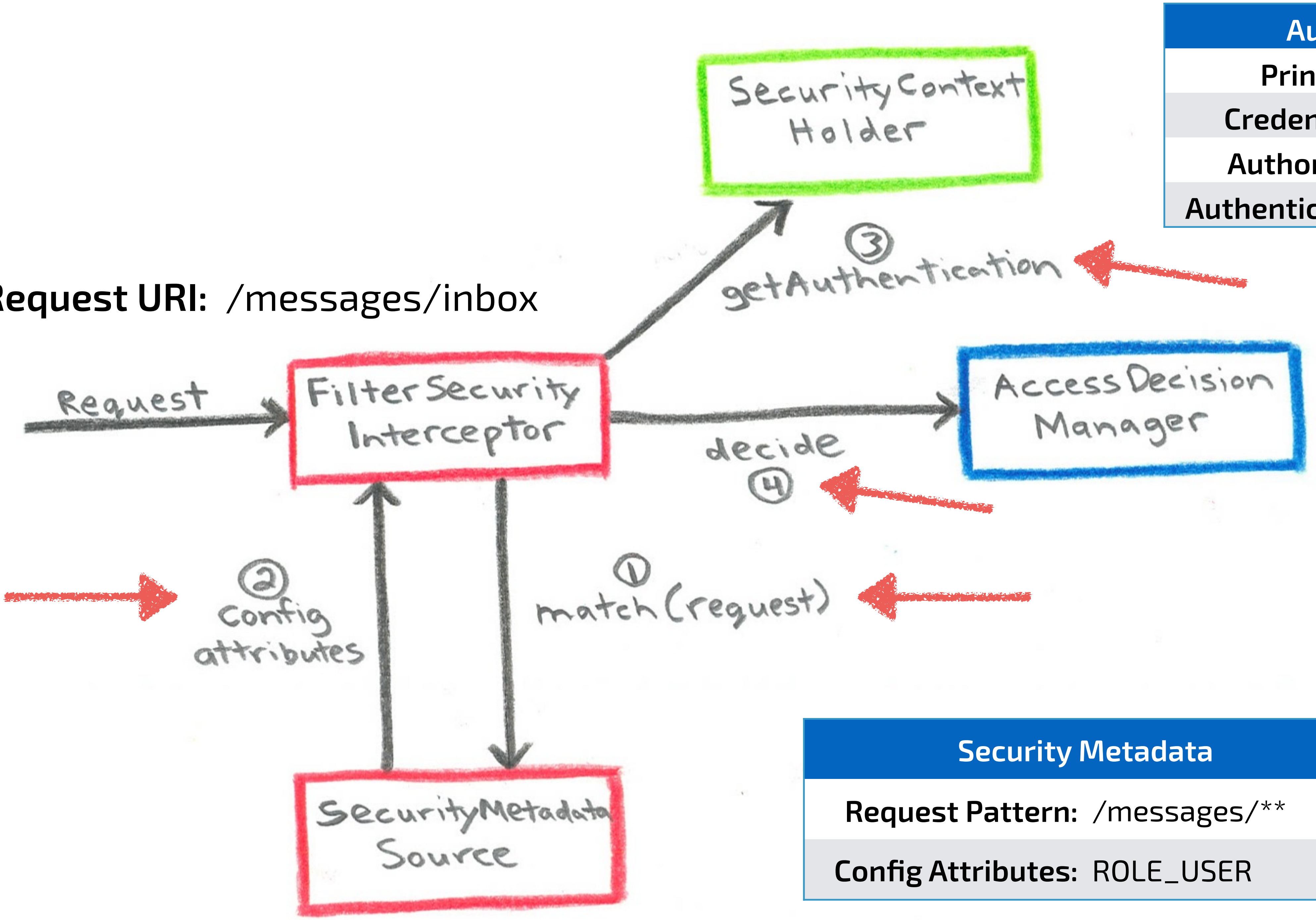
AUTHENTICATION

AUTHORIZATION

EXCEPTION HANDLING

Filter Security Interceptor

Request URI: /messages/inbox



Authentication	
Principal:	UserDetails
Credentials:	——
Authorities:	ROLE_USER
Authenticated:	TRUE

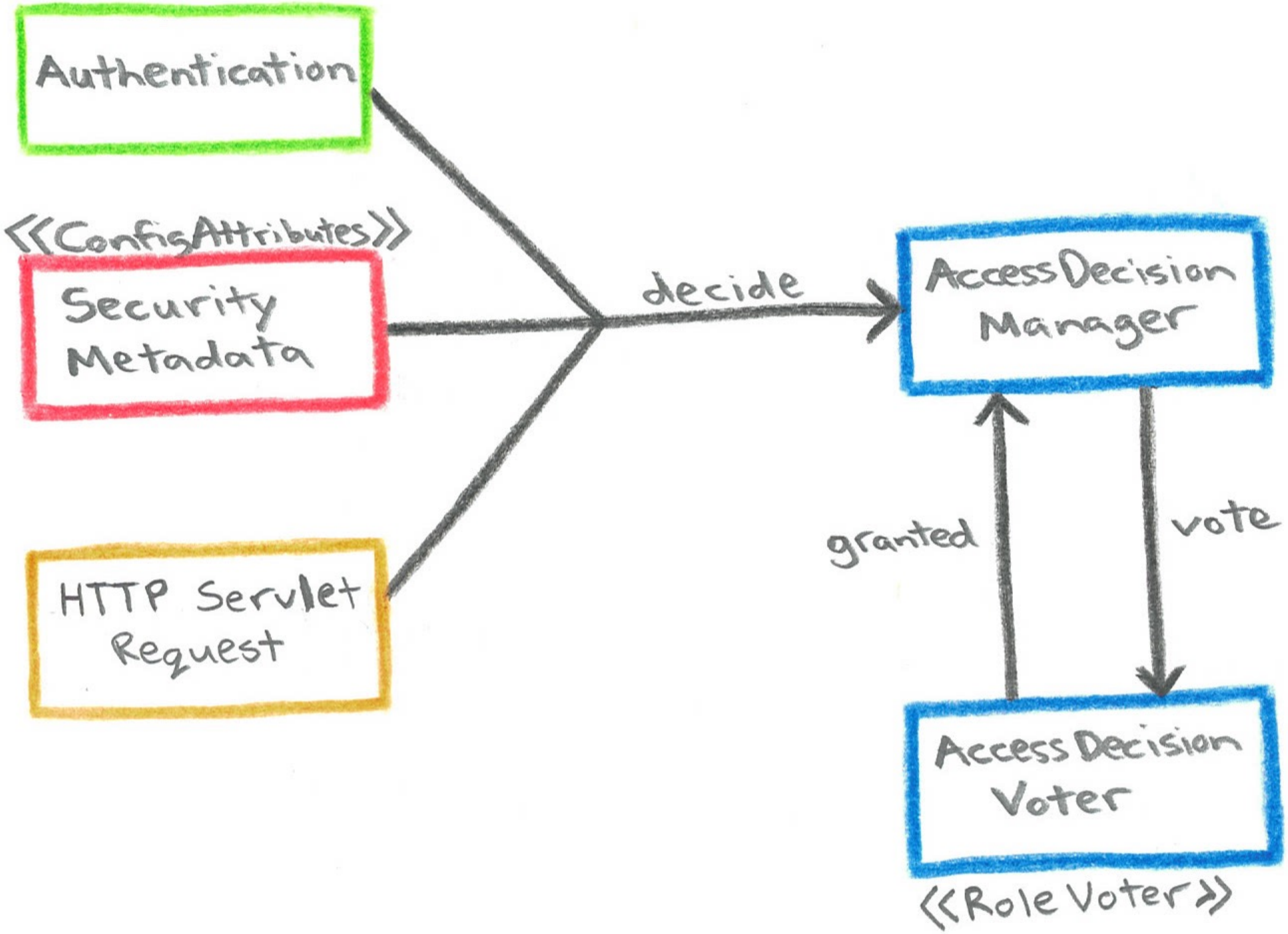
Security Metadata	
Request Pattern:	/messages/**
Config Attributes:	ROLE_USER

Access Decision

Authentication	
Principal:	UserDetails
Credentials:	——
Authorities:	ROLE_USER
Authenticated:	TRUE

Security Metadata	
Request Pattern:	/messages/**
Config Attributes:	ROLE_USER

Request URI: /messages/inbox



Authorization Recap

- **FilterSecurityInterceptor** obtains the "***Security Metadata***" by matching on the current request
- FilterSecurityInterceptor gets the current **Authentication**
- The Authentication, Security Metadata and Request is passed to the **AccessDecisionManager**
- The AccessDecisionManager delegates to it's **AccessDecisionVoter(s)** for decisioning

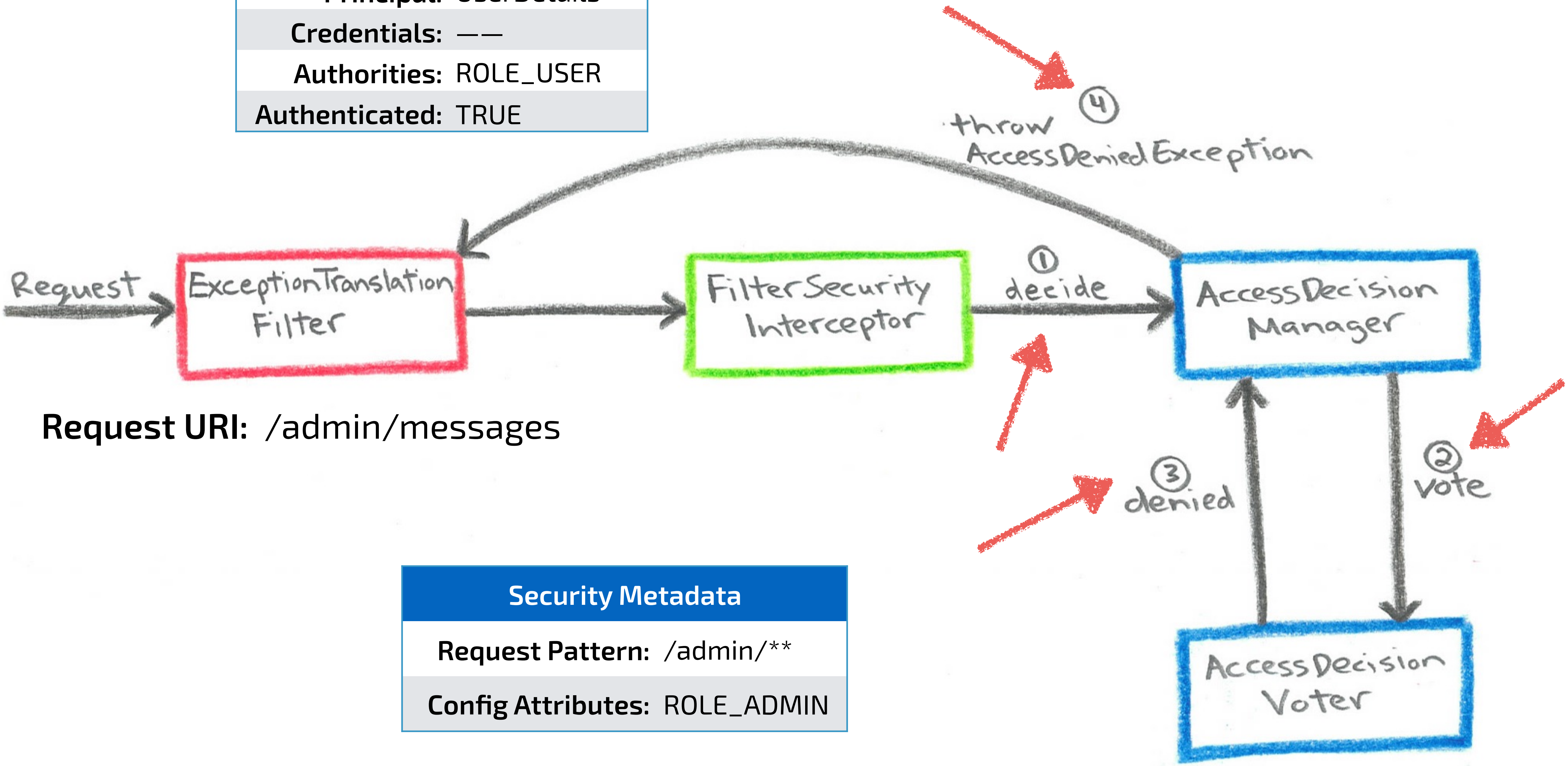
AUTHENTICATION

AUTHORIZATION

EXCEPTION HANDLING

Access Denied

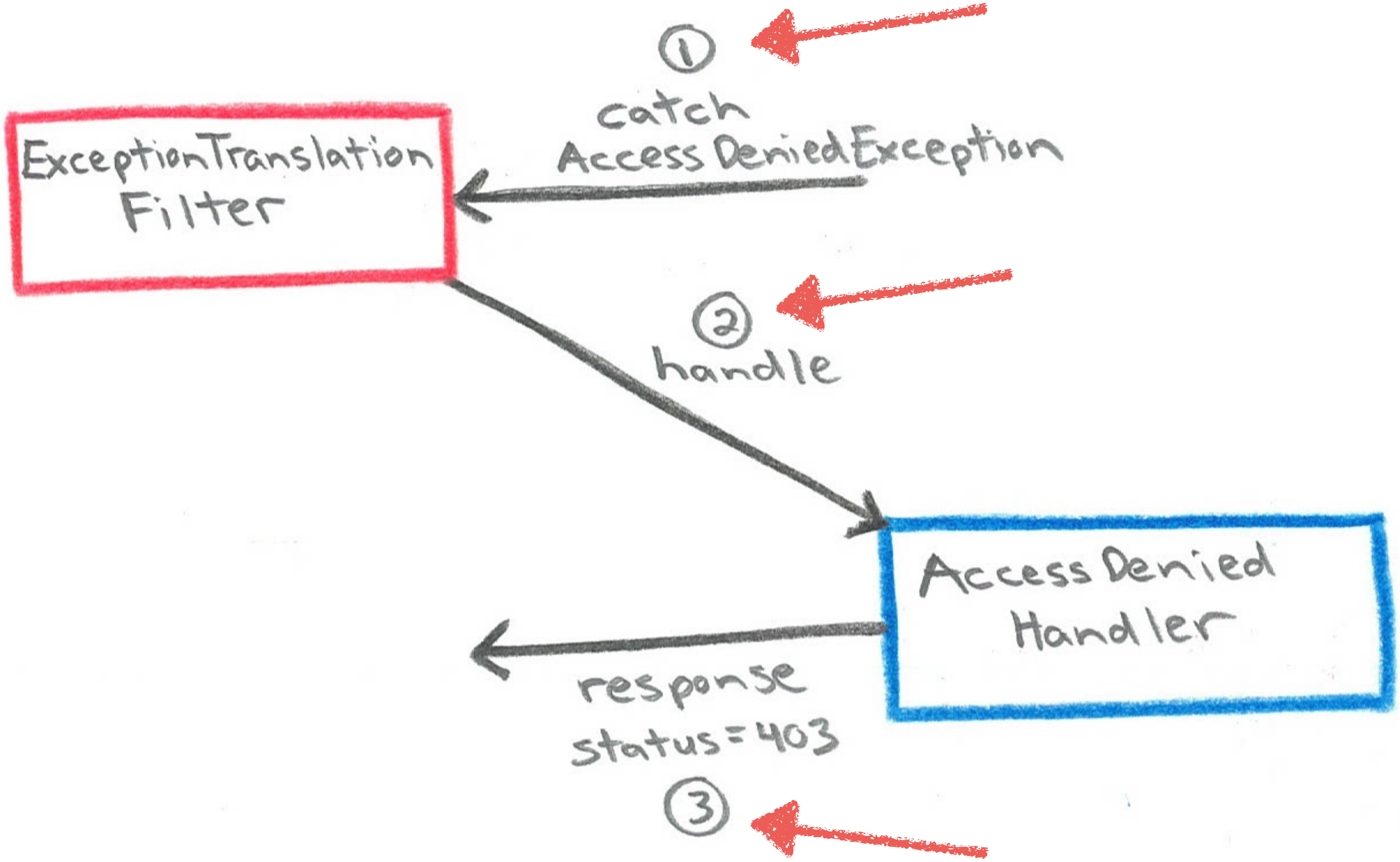
Authentication	
Principal:	UserDetails
Credentials:	——
Authorities:	ROLE_USER
Authenticated:	TRUE



Request URI: /admin/messages

Security Metadata	
Request Pattern:	/admin/**
Config Attributes:	ROLE_ADMIN

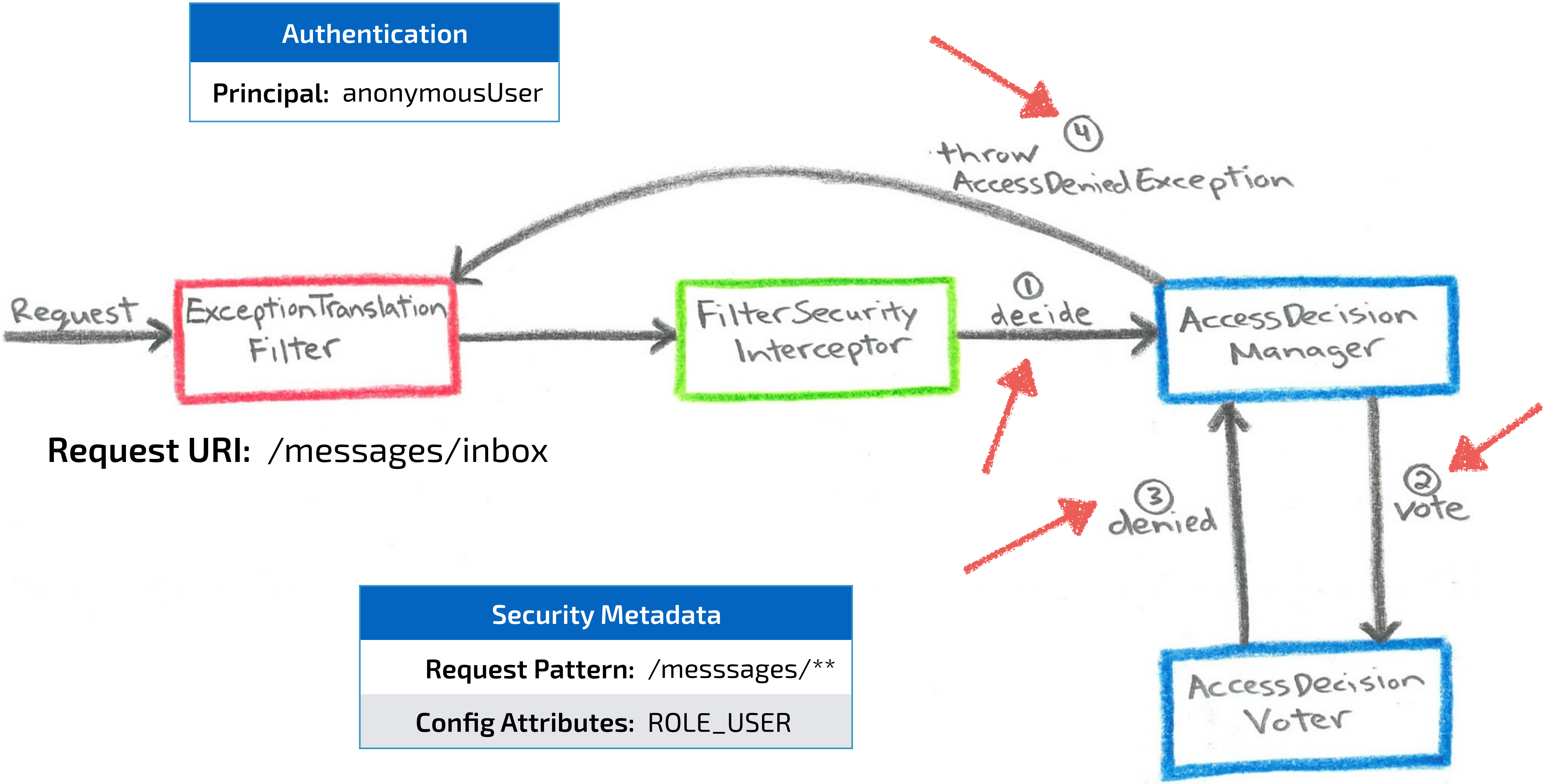
Access Denied Handler



```
public interface AccessDeniedHandler {

  void handle(HttpServletRequest request, HttpServletResponse response,
       AccessDeniedException accessDeniedException) throws IOException, ServletException;

}
```

“Unauthenticated”

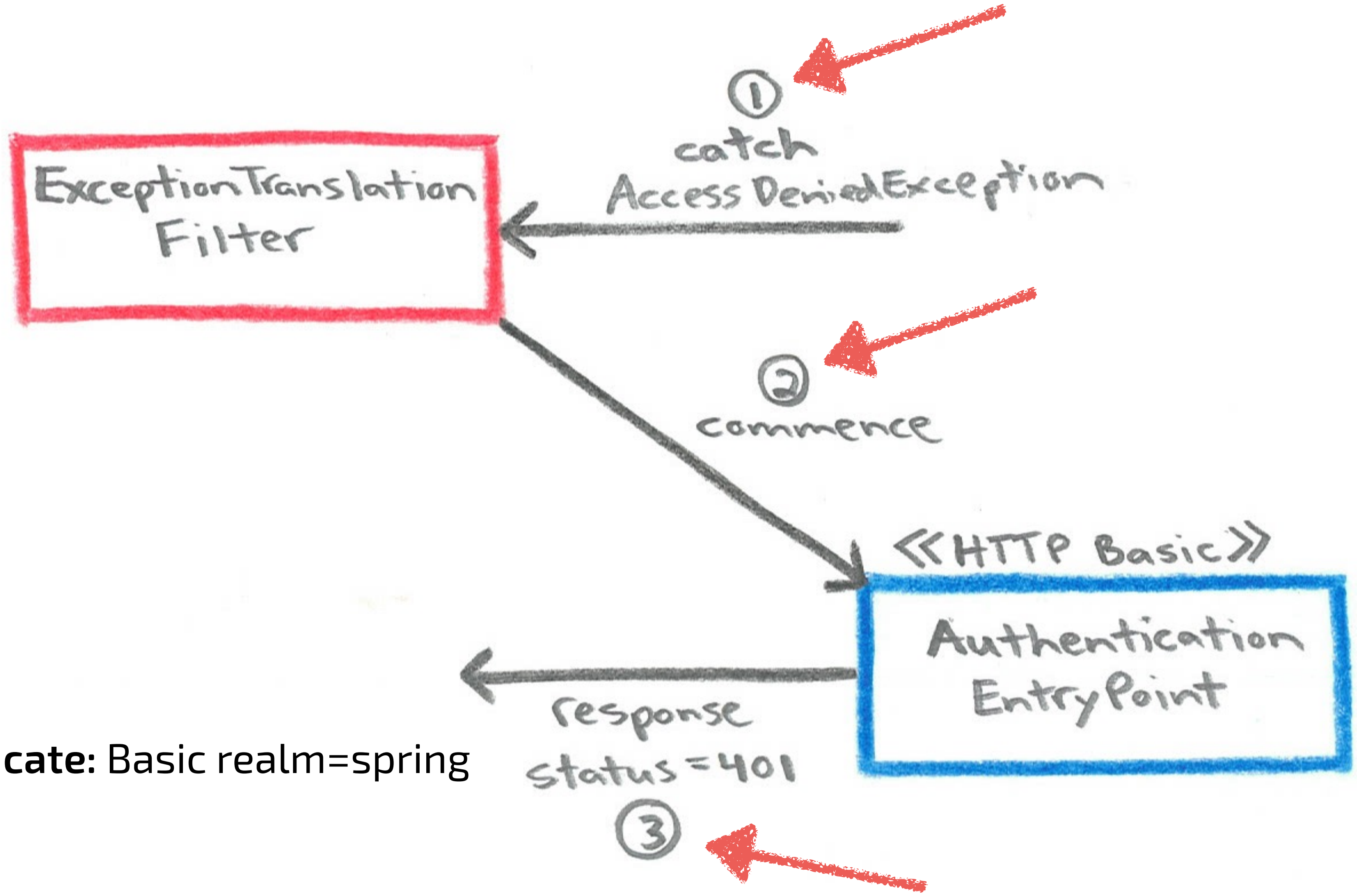
Authentication
Principal: anonymousUser



Request URI: /messages/inbox

Security Metadata	
Request Pattern:	/messsages/**
Config Attributes:	ROLE_USER

Start Authentication



WWW-Authenticate: Basic realm=spring

```
public interface AuthenticationEntryPoint {

   void commence(HttpServletRequest request, HttpServletResponse response,
           AuthenticationException authException) throws IOException, ServletException;

}
```

Exception Handling Recap

- When *"Access Denied"* for current Authentication, the **ExceptionTranslationFilter** delegates to the **AccessDeniedHandler**, which by default, returns a 403 Status.
- When current Authentication is *"Anonymous"*, the **ExceptionTranslationFilter** delegates to the **AuthenticationEntryPoint** to start the Authentication process.

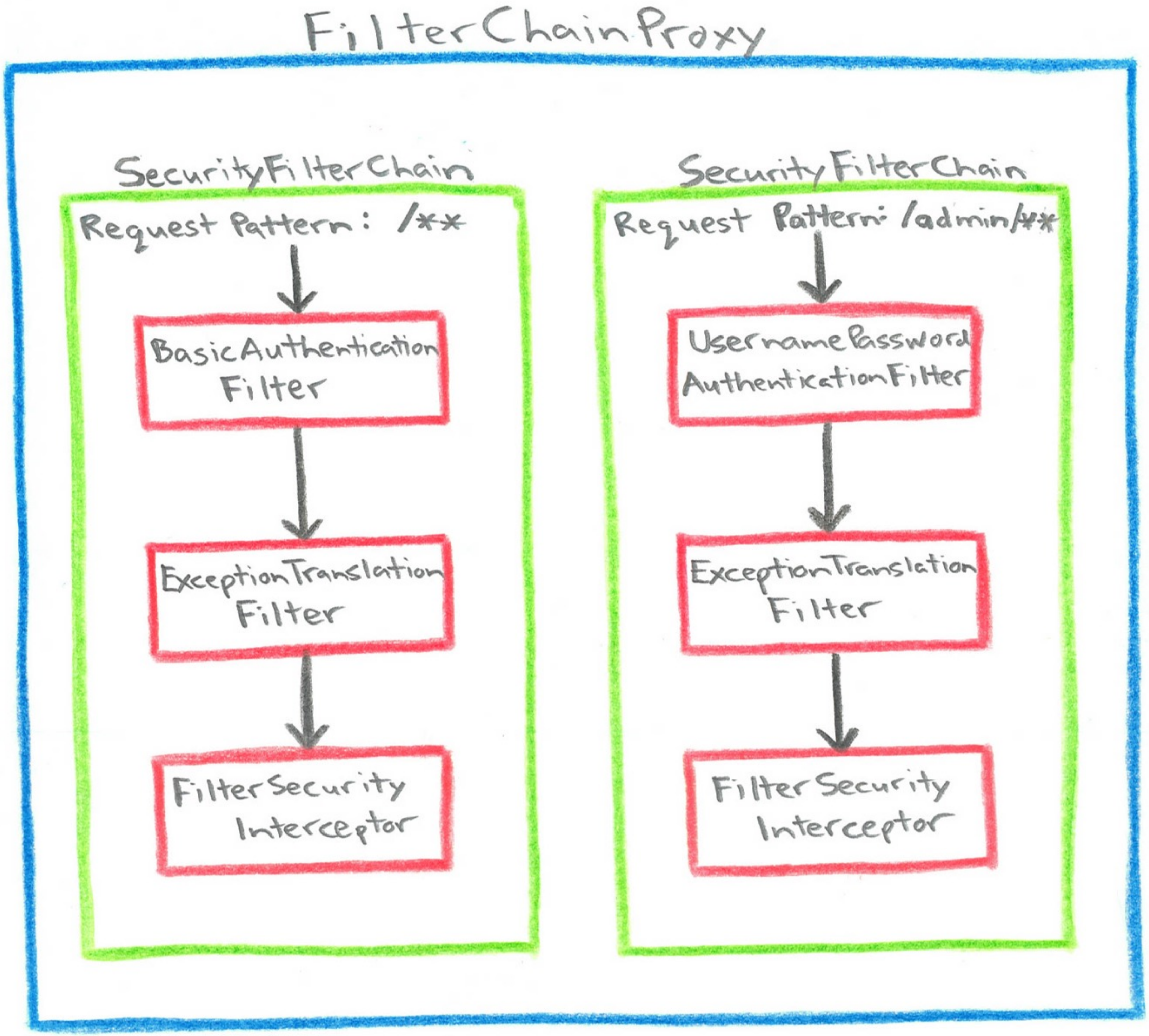
Summary

Authentication

Authorization

Exception Handling

Spring Security Filter Chain


FilterChainProxy
SecurityFilterChain
Request Pattern: /**
BasicAuthentication Filter
ExceptionTranslation Filter
FilterSecurity Interceptor
SecurityFilterChain
Request Pattern: /admin/**
UsernamePassword AuthenticationFilter
ExceptionTranslation Filter
FilterSecurity Interceptor

Q&A

github.com/jgrandja/messaging-sample