

# Spring security

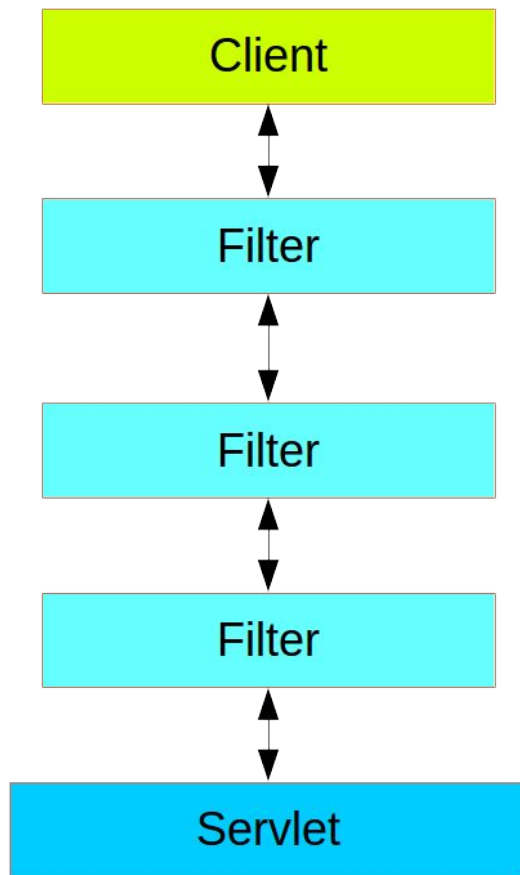
# Spring Security Architecture

<https://spring.io/guides/topicals/spring-security-architecture/>

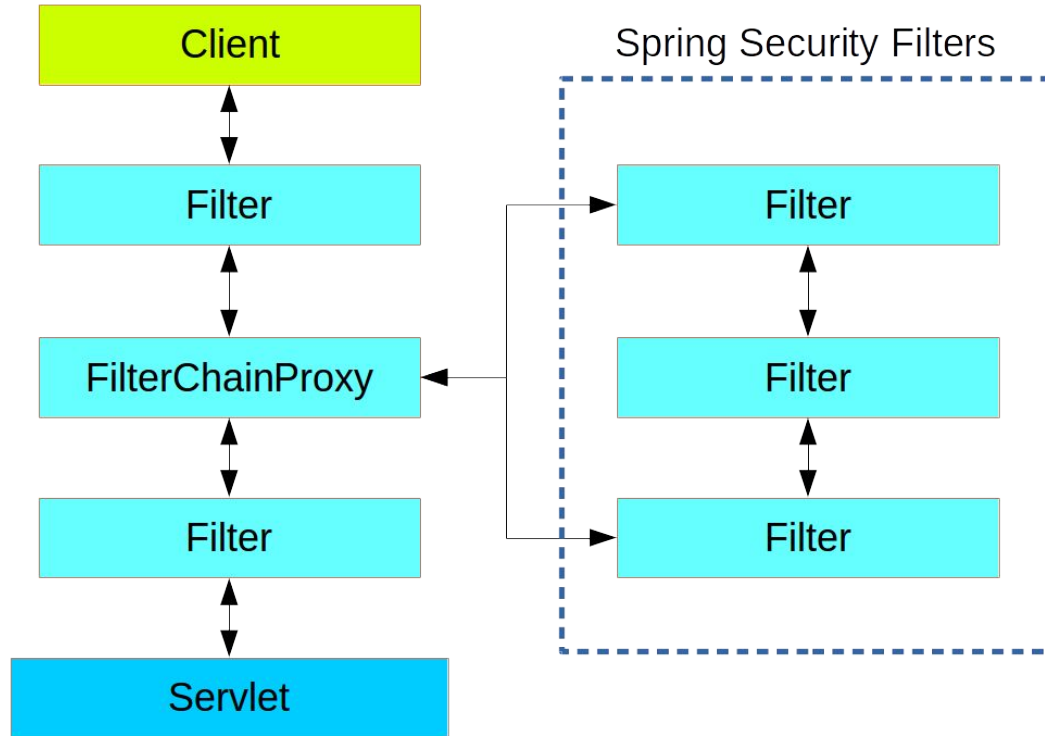
## Authentication and Access Control

1. Authentication (who are you?)
2. Authorization (what are you allowed to do?)

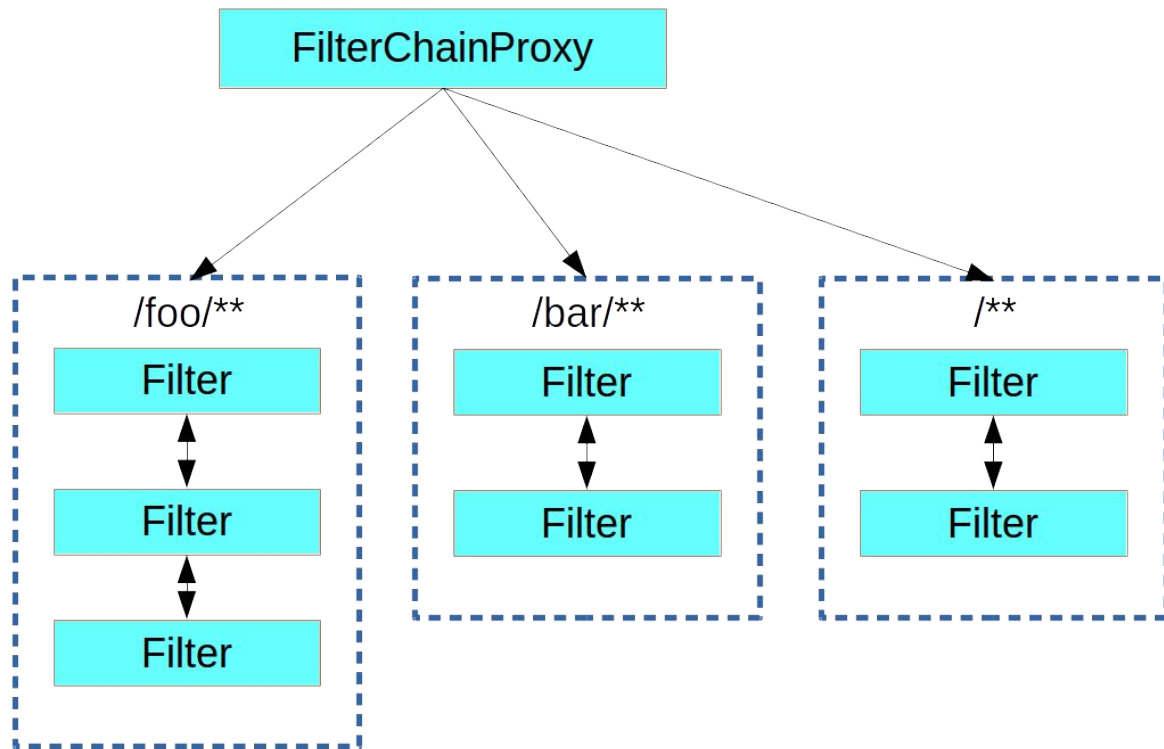
# Web Security



# Spring Security Filter



## Filter Chain Proxy



# Matching roles

---

```
@Configuration
@Order(SecurityProperties.BASIC_AUTH_ORDER - 10)
public class ApplicationConfigurerAdapter extends WebSecurityConfigurerAdapter {
    @Override
    protected void configure(HttpSecurity http) throws Exception {
        http.antMatcher("/match1/**")
            .authorizeRequests()
                .antMatchers("/match1/user").hasRole("USER")
                .antMatchers("/match1/spam").hasRole("SPAM")
                .anyRequest().isAuthenticated();
    }
}
```

# Method Security

```
@SpringBootApplication

@EnableGlobalMethodSecurity(securedEnabled = true)

public class SampleSecureApplication {

}

@Service
public class MyService {

    @Secured("ROLE_USER")
    public String secure() {
        return "Hello Security";
    }

}
```

# Get user in our code

```
SecurityContext context = SecurityContextHolder.getContext();  
  
Authentication authentication = context.getAuthentication();  
  
assert(authentication.isAuthenticated);
```

```
@RequestMapping("/foo")
```

```
public String foo(@AuthenticationPrincipal User user) {  
  
    ... // do stuff with user  
  
}
```

```
@RequestMapping("/foo")  
public String foo(Principal principal) {  
    Authentication authentication = (Authentication) principal;  
    User = (User) authentication.getPrincipal();  
    ... // do stuff with user  
}
```



# Processing Secure Methods Asynchronously

```
@Configuration
```

```
public class ApplicationConfiguration extends AsyncConfigurerSupport {
```

```
    @Override
```

```
    public Executor getAsyncExecutor() {
```

```
        return new DelegatingSecurityContextExecutorService(Executors.newFixedThreadPool(5));
```

```
    }
```

```
}
```