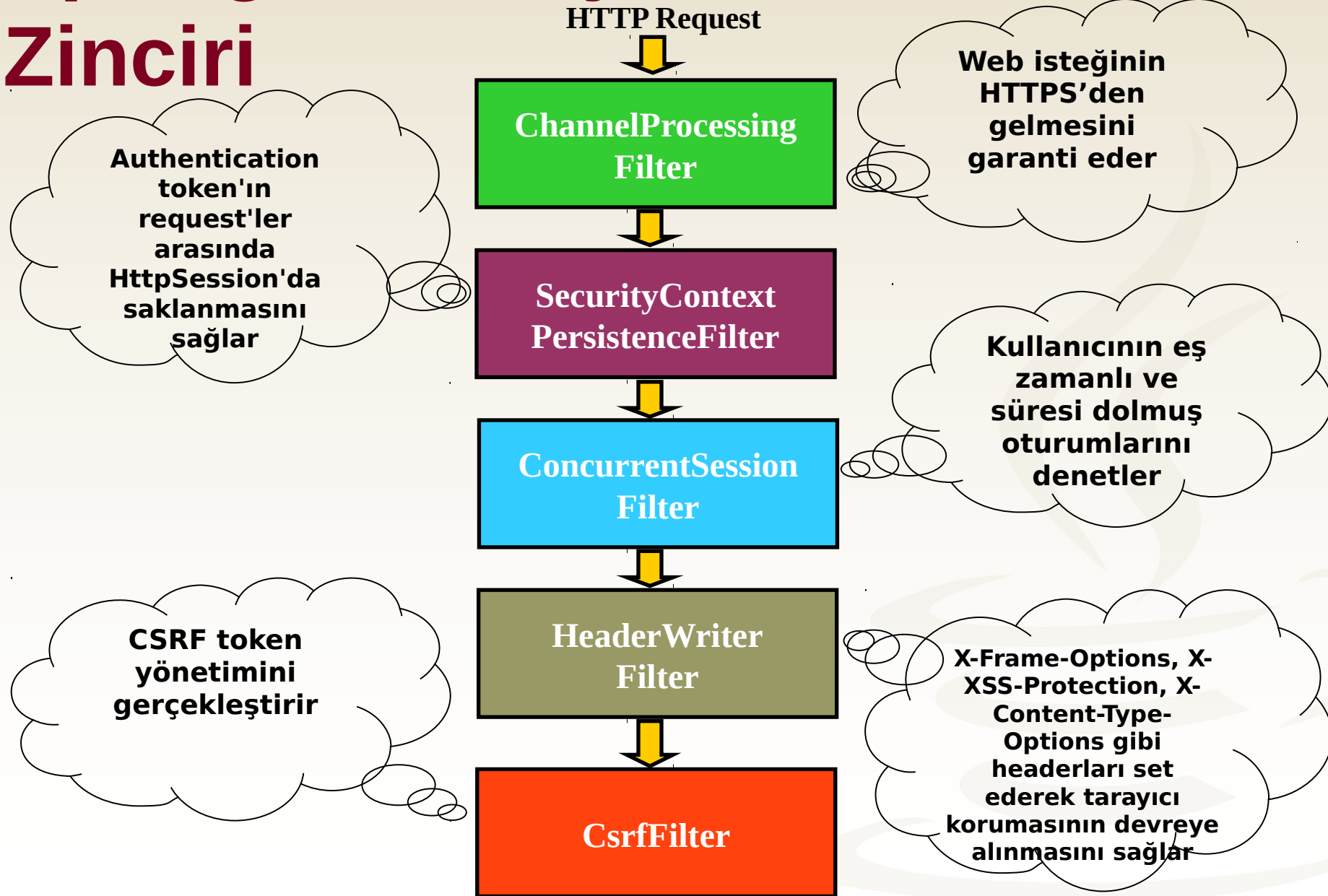


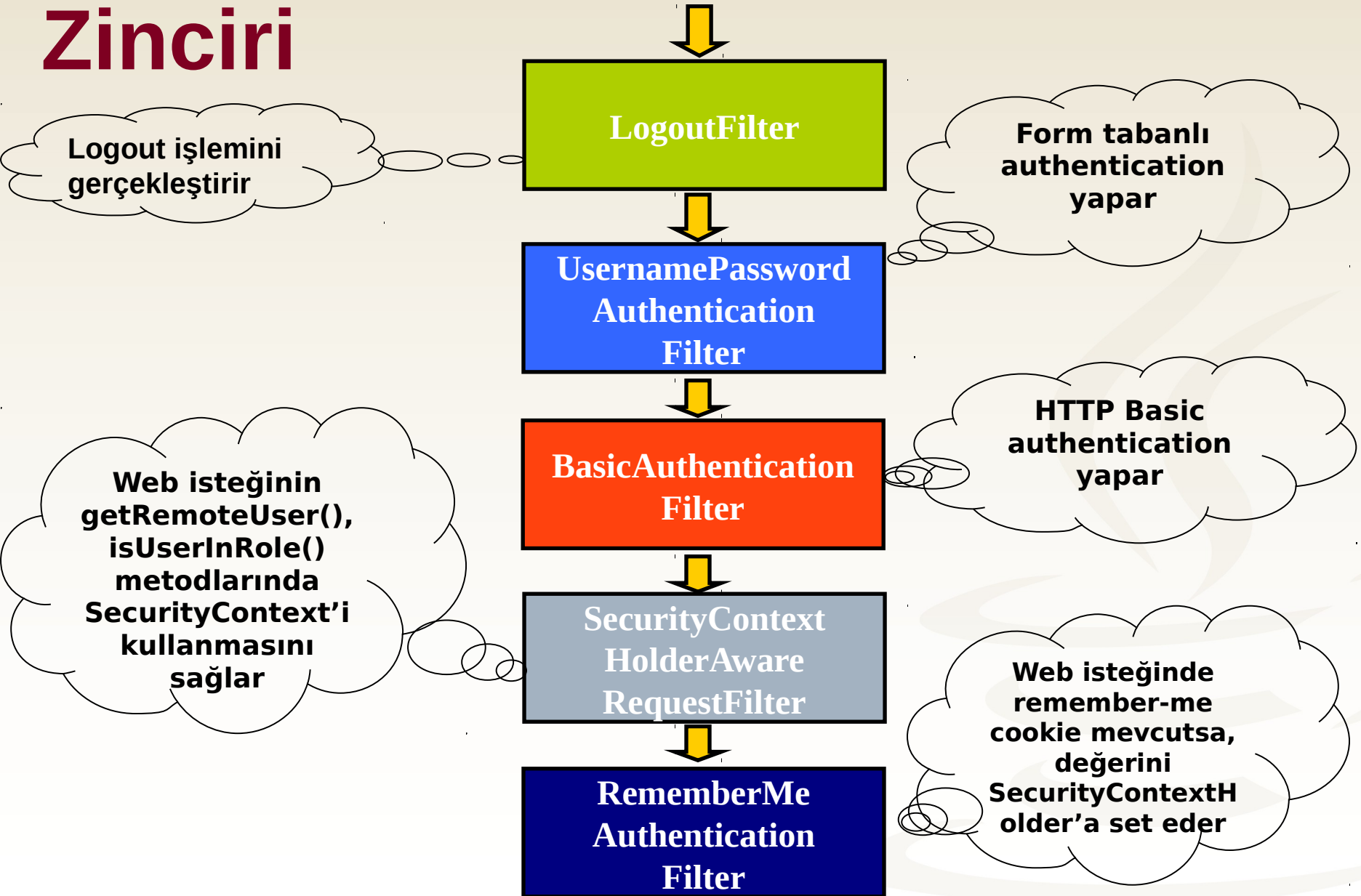
# Spring Security Filter Zinciri ve Filter Mimarisi (İleri Düzey)



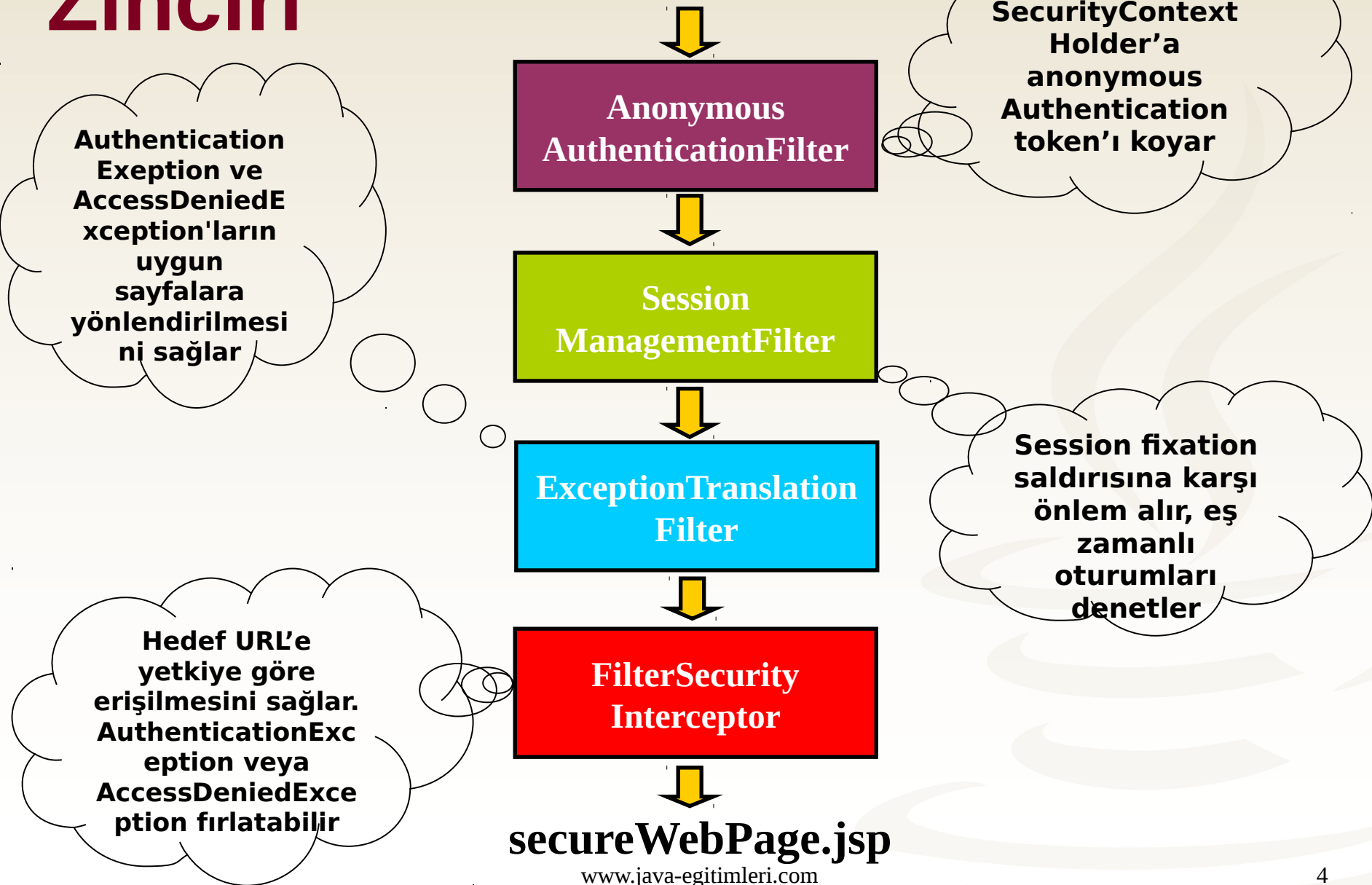
# Spring Security Filter Zinciri



# Spring Security Filter Zinciri



# Spring Security Filter Zinciri



# Container'da Birden Fazla <security:http> Elemanı

- Spring konfigurasyon dosyasında aynı anda birden fazla `<security:http>` elemanı tanımlanabilir
- Bunun amacı uygulama içerisinde **farklı URI pattern'larını farklı kimliklendirme metotları** ile ele almaktır
- Ya da her bir URI pattern'ında **farklı güvenlik kontrollerini** devreye sokmaktır

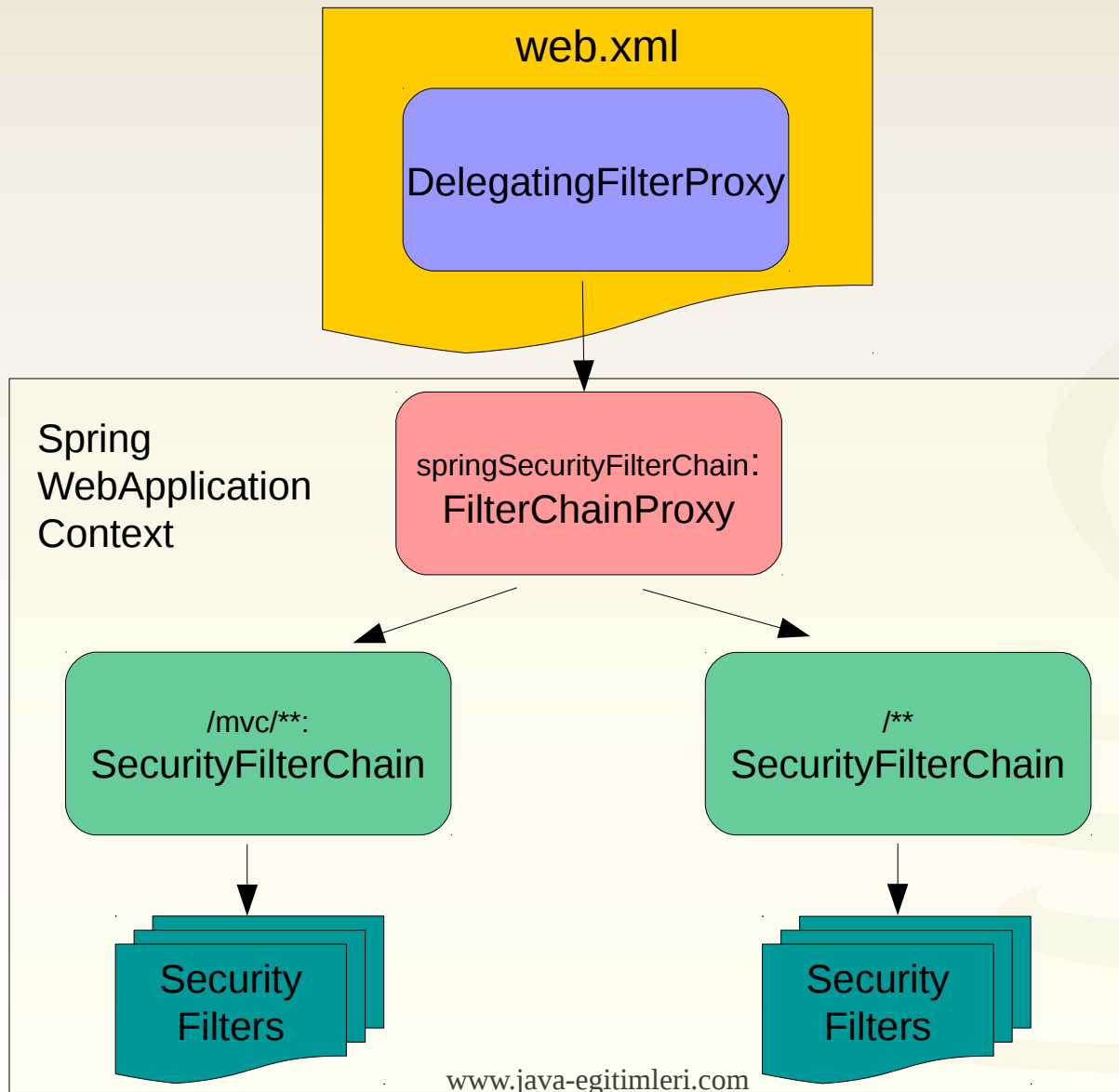
# Container'da Birden Fazla <security:http> Elemanı

```
<beans...>
  <security:http pattern="/mvc/**" create-session="stateless">
    <security:http-basic />
    <security:intercept-url pattern="/**" access="hasRole('ROLE_USER')"/>
  </security:http>

  <security:http pattern="/**" auto-config="true">
    <security:access-denied-handler
      error-page="/accessDenied.jsp"/>

    <security:form-login login-page="/login.jsp"
      authentication-failure-url="/login.jsp?error=true" />
    <security:intercept-url pattern="/login.jsp"
      access="isAnonymous()" />
    <security:intercept-url pattern="/editor.jsp"
      access="hasRole('ROLE_EDITOR')" />
    <security:intercept-url pattern="/**"
      access="hasRole('ROLE_USER')" />
  </security:http>
</beans>
```

# Spring Security Filter Mimarisi



# Spring Security Filter Mimarisi

- Her bir http elemanı ayrı bir **SecurityFilterChain** tipinde bean'e karşılık gelir
- Ayrıca herhangi bir http elemanının tanımlanması `ApplicationContext`'de **FilterChainProxy** tipinde **tek bir bean** tanımlanmasını da tetikler
- `SecurityFilterChain` bean'leri bu **FilterChainProxy bean'i içerisinde** yer alırlar



# Spring Security Filter Mimarisi

- FilterChainProxy tipindeki bean'in ismi “**springSecurityFilterChain**” dir ve değiştirilemez
- web.xml'de tanımlı **DelegatingFilterProxy** tipindeki filter ApplicationContext'de bu isimde bir bean'i arar
- Fakat her bir **SecurityFilterChain** bean'ine **http elemanı üzerinden ayrı bir bean ismi** atanabilir

# Spring Security

## Debug Info

`<security:debug/>`

- **Debug** altyapısını aktive eder
- Security Filter'lara gelen request'leri okunur biçimde **console**'a basar
- **Bu request'lerdeki** header, request param, request path gibi önemli **bilgilerde** console'da yer alır
- Filter zincirindeki **Filter'ları listeler**
- Sadece **development ortamı** içindir

# Spring Security Debug Info

Request received for POST '/j\_spring\_security\_check':

(POST /petclinic/j\_spring\_security\_check)@1819769884 org.eclipse.jetty.server.Request@6c777c1c

servletPath:/j\_spring\_security\_check

pathInfo:null

headers:

Cookie: JSESSIONID=1q3txc33ic4nylg4ht2n29wrlv; locale=en

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8

User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:32.0) Gecko/20100101 Firefox/32.0

Referer: http://localhost:8080/petclinic/spring\_security\_login

Connection: keep-alive

Host: localhost:8080

Accept-Language: tr,en-us;q=0.7,en;q=0.3

Accept-Encoding: gzip, deflate

Content-Length: 44

Content-Type: application/x-www-form-urlencoded

Security filter chain: [

SecurityContextPersistenceFilter

WebAsyncManagerIntegrationFilter

LogoutFilter

UsernamePasswordAuthenticationFilter

DefaultLoginPageGeneratingFilter

BasicAuthenticationFilter

RequestCacheAwareFilter

SecurityContextHolderAwareRequestFilter

AnonymousAuthenticationFilter

SessionManagementFilter

ExceptionTranslationFilter

FilterSecurityInterceptor

]

# İletişim

- **Harezmi** Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

