Yetkilendirme (Authorization)

harezmi bilişim çözümleri

Yetkilendirme Nedir?

- Kimliklendirmesi yapılmış herhangi bir **principal**'ın (kullanıcı veya harici sistem) uygulama içerisindeki **nesneler üzerinde hangi işlemleri** gerçekleştirebileceğinin **kontrolü**ne yetkilendirme denir
- Erişim denetimine tabi tutulan herhangi bir nesneye "**secure object**" veya "**secure resource**" adı verilir

Yetkilendirmenin Uygulandığı Resource Türleri

- Aşağıdakilerden herhangi birisi secure object olabilir
  - **Web istekleri**: web sayfalarına veya dosyalara erişim
  - **Metot çağrıları**: web servis metot çağrılarına veya servis katmanındaki nesnelerin metotlarına erişim
  - **Domain nesneleri**: veritabanındaki kayıtlara erişim, bu kayıtlar üzerinde herhangi bir işlem

Yetkisiz Erişim Örneği

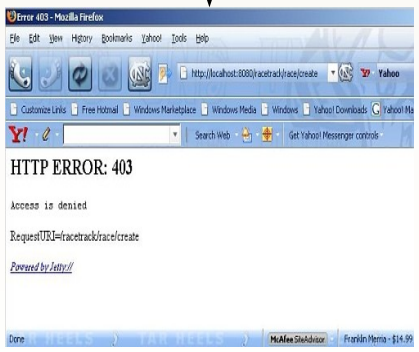
1:HTTP request from User with ROLE_READER to editor.jsp

Throws **AccessDenied Exception**

FilterSecurity Interceptor

ROLE_EDITOR

3:ExceptionTranlationFilter
Redirect to accessDenied page



accessDenied.jsp

AccessDecision Manager

AccessDecision Voter

Tek tek Voter nesnelere kullanıcının editor.jsp'ye secure resource'una erşip erişemeyeceği sorulur, **ACCESS_DENIED** cevabı Alındığında **AccessDeniedException** fırlatılır

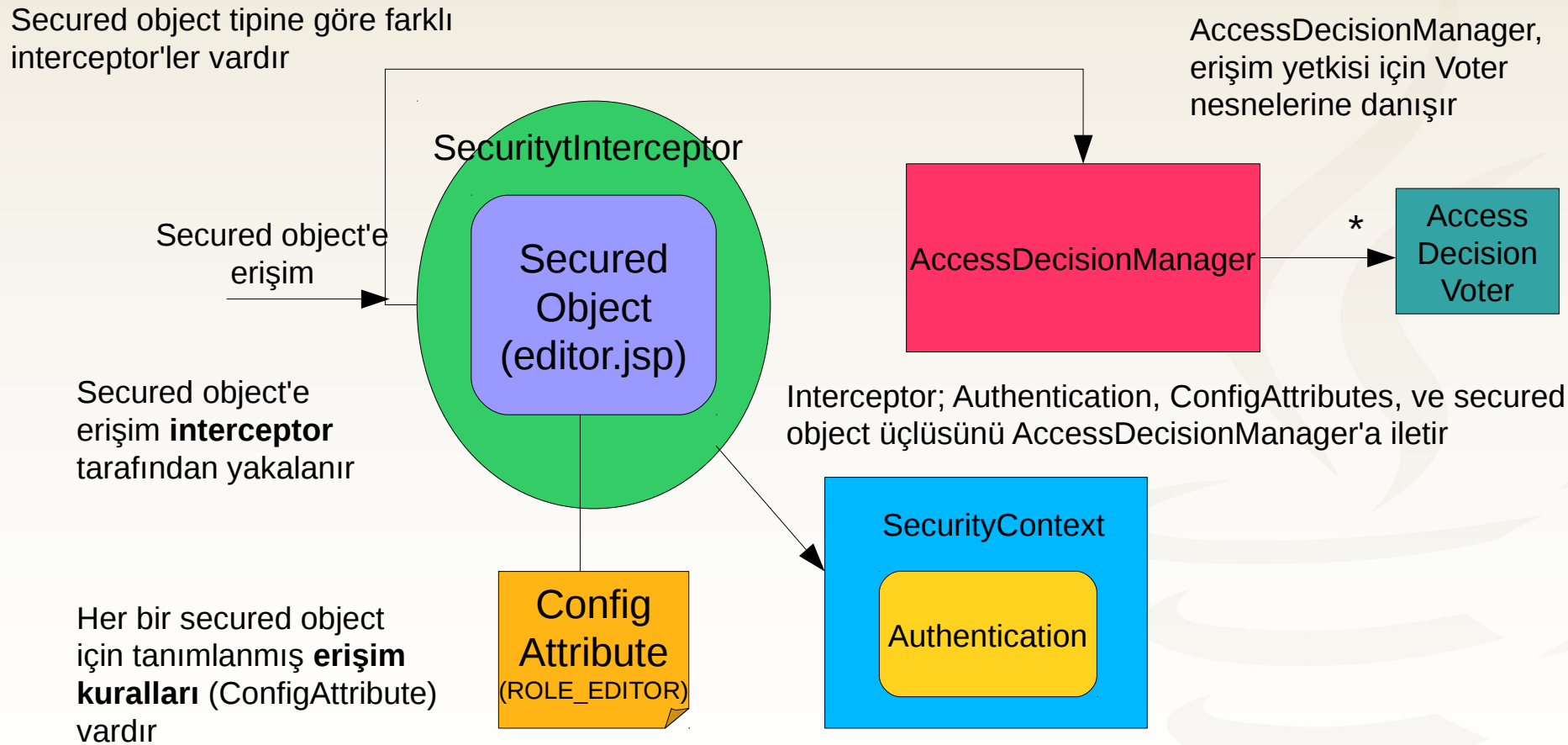
Web Kaynaklarının Yetkilendirilmesi

- `<security:http>` elemanı içerisinde tanımlanan `<security:intercept-url/>` elemanları ile hang **web kaynağı**na hangi **roller**in erişebileceği tanımlanır

Yetkisiz erişim durumunda gösterilecek olan hata sayfası

```
<security:http>
 <security:access-denied-handler error-page="/accessDenied.jsp"/>
 <security:intercept-url pattern="/editor.jsp"
                         access="hasRole('ROLE_EDITOR')" />
 <security:intercept-url pattern="/accessDenied.jsp" access="permitAll"/>
 <security:intercept-url pattern="/**" access="hasRole('ROLE_USER')"/>
</security:http>
```

Yetkilendirmenin İşleyişi



AccessDecisionVoter Nesneleri

- Secure nesne'ye erişim talebi **AccessDecisionVoter** nesneleri tarafından değerlendirilir
- Sistemde aynı anda **birden fazla voter** olabilir
- AccessDecisionVoter nesneleri **cevap olarak** aşağıdakilerden birisini dönerler
  - Çekimser ise :**ACCESS_ABSTAIN**
  - Erişimi onaylamıyor ise :**ACCESS_DENIED**
  - Erişime izin veriyor ise :**ACCESS_GRANTED**

AccessDecisionManager

- AccessDecisionVoter'ların kararlarına bakarak erişime karar veren son nokta ise **AccessDecisionManager** nesnesidir
- Üç farklı implementasyonu vardır
  - **AffirmativeBased**, en azından bir access_granted durumunda olumlu
    - Web ve metot yetkilendirmede **default** register edilen budur
  - **UnanimousBased**, en az bir access_denied varsa olumsuz
  - **ConsensusBased**, sum(access_granted)>sum(access_denied)

Expression Tabanlı Yetkilendirme

- Access attribute içerisindeki yetki tanımlarında **Spring EL** kullanılabilir
- Spring Security 4 ile birlikte expression tabanlı yetkilendirme **default yöntem**dir

```
<security:http use-expressions="true">

    <security:intercept-url pattern="/admin*"
          access="hasRole('ROLE_ADMIN') and
                  hasIpAddress('192.168.1.0/24')"/>

</security:http>
```

Expression Tabanlı Yetkilendirme

- Web ve metot yetkilendirme için bazı **built-in tanımlı** fonksiyonlar:
  - hasRole('ROLE_USER')
  - hasAnyRole('ROLE_USER','ROLE_EDITOR')
  - hasAuthority('ROLE_USER')
  - hasAnyAuthority('ROLE_USER','ROLE_EDITOR')
  - hasIpAddress('192.168.1.1')
  - isAnonymous(), isRememberMe(), isAuthenticated(), isFullyAuthenticated()

Expression Tabanlı Yetkilendirme

- Yetki ifadelerinde kullanılabilecek bazı placeholder değişkenler:
  - **principal**, **authentication**: aktif kullanıcı bilgisine erişmeyi sağlar
  - **permitAll**: her zaman için true değer üretir
  - **denyAll**: her zaman için false değer üretir

İletişim

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- http://www.java-egitimleri.com
- info@java-egitimleri.com