Sicurezza j2ee con Acegi

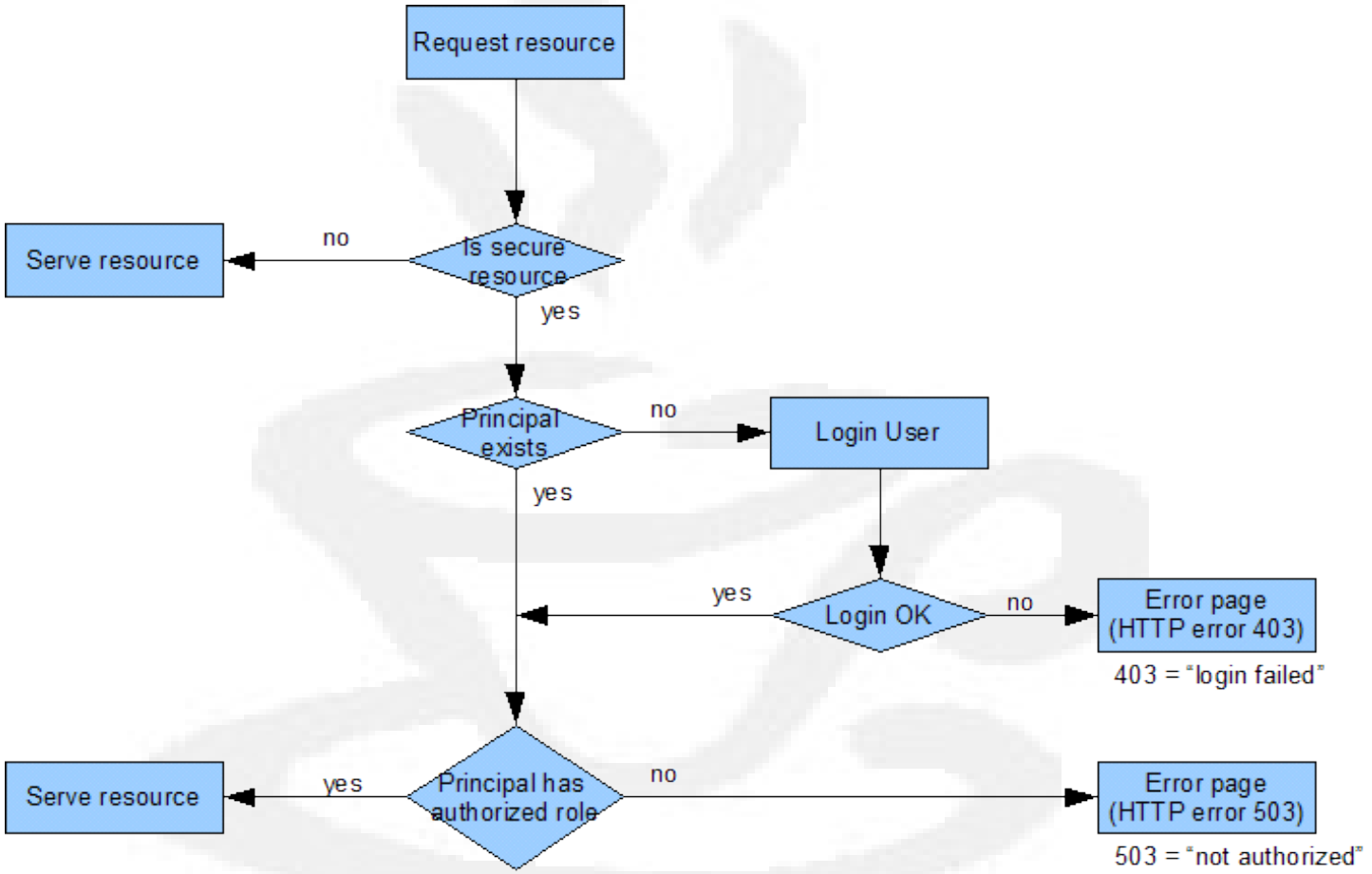
a cura di

Java User Group Padova

- Acegi fornisce servizi di sicurezza per applicazioni J2EE
- Sub project Spring framework
- Sito: http://www.acegisecurity.org
- Portabile, configurabile

- Autenticazione: il processo di identificare una persona. “Who are you?”
- Autorizzazione: il processo di determinare se un determinato utente ha il permesso di accedere alla risorsa che ha chiesto.

Accesso ad una risorsa protetta



- Fa utilizzo di filtri specializzati per la gestione di una specifica funzione (logout, autenticazione, etc.)
- Il web.xml dichiara un unico filtro, tramite FilterToBeanProxy delega ad una chain di Filter dichiarati nel context di spring
- In questo modo un filtro utilizza injection fornito dal contesto di spring
- Autorizzazione: web con filtri, AOP per metodi.
- Integrabile con molti sistemi di autenticazione: BASIC, Form-based, JA-SIG Central Authentication Service (CAS), LDAP.

```
<context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>
                        /WEB-INF/acegi-security-4.xml
                </param-value>
        </context-param>

    <filter>
        <filter-name>Acegi Filter Chain Proxy</filter-name>
        <filter-class>org.acegisecurity.util.FilterToBeanProxy</filter-class>
        <init-param>
            <param-name>targetClass</param-name>
            <param-value>org.acegisecurity.util.FilterChainProxy</param-value>
        </init-param>
    </filter>

    <filter-mapping>
      <filter-name>Acegi Filter Chain Proxy</filter-name>
      <url-pattern>/*</url-pattern>
    </filter-mapping>


   <listener>
        <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
   </listener>
```

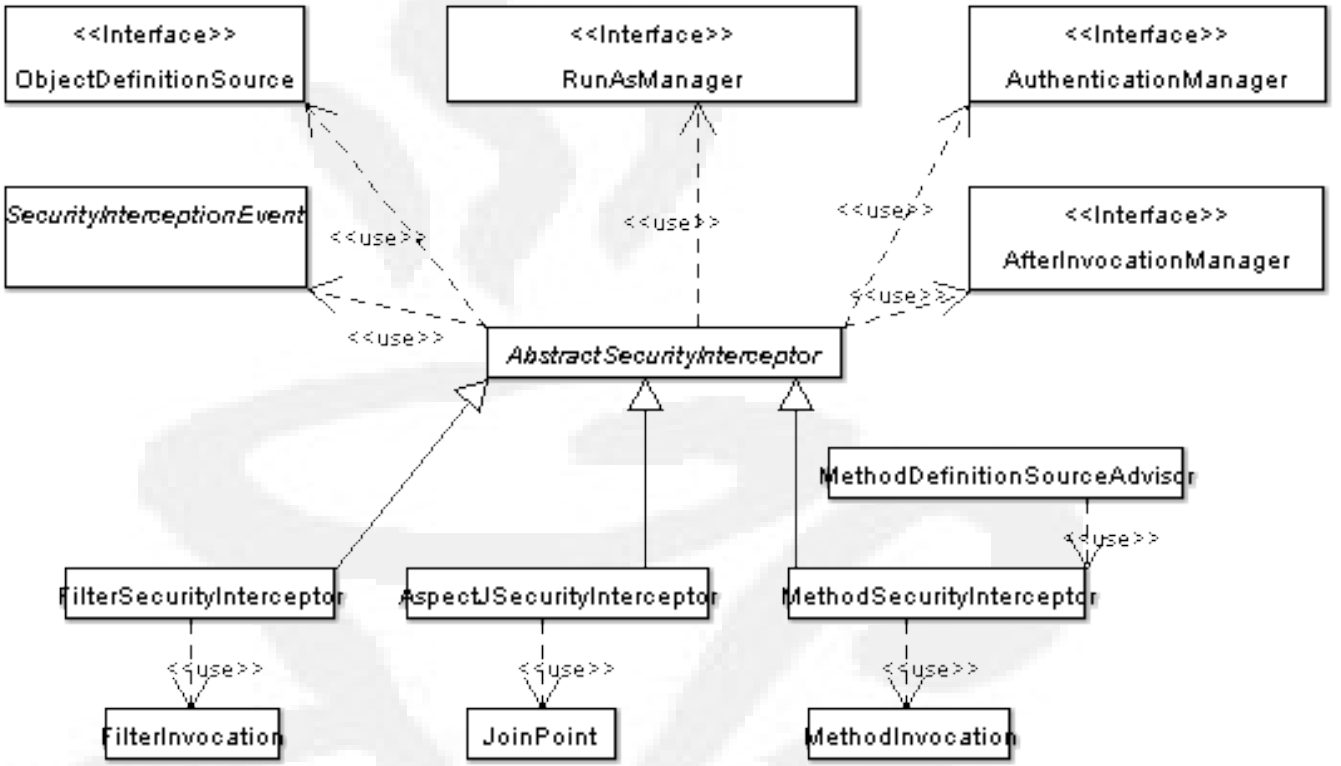
```
<!-- filter chain -->
	<bean id="filterChainProxy" class="org.acegisecurity.util.FilterChainProxy">
		<property name="filterInvocationDefinitionSource">
			<value>
				CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
				PATTERN_TYPE_APACHE_ANT
				/**=httpSessionContextIntegrationFilter,logoutFilter,
basicProcessingFilter,
securityContextHolderAwareRequestFilter,rememberMeProcessingFilter,
anonymousProcessingFilter,exceptionTranslationFilter,filterInvocationInterceptor
			</value>
		</property>
	</bean>
```

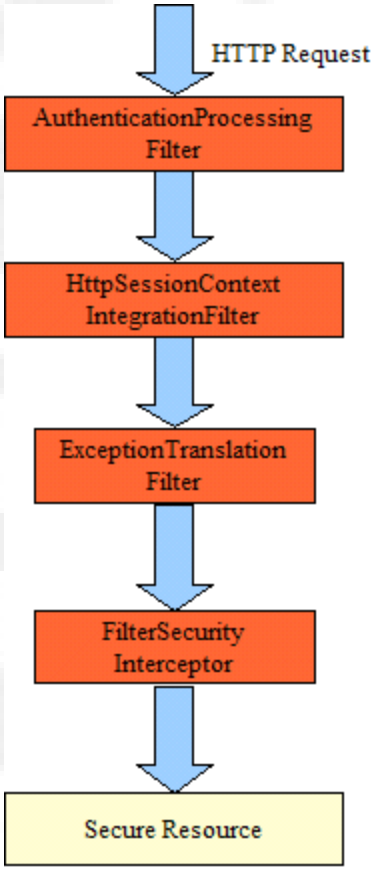
Associazione ruolo – risorsa (objectDefinitionSource)

```
<property name="objectDefinitionSource">
<value>
CONVERT_URL_TO_LOWERCASE_BEFORE_COMPARISON
PATTERN_TYPE_APACHE_ANT
/secure/extreme/**=ROLE_ADMIN
/secure/**=IS_AUTHENTICATED_REMEMBERED
/**=IS_AUTHENTICATED_ANONYMOUSLY
</value>
</property>
```

Acegi

- SecurityContextHolder: fornisce accesso con metodo statico a Security Context (utilizza ThreadLocal).
- SecurityContext: mantiene informazioni relative all' utente autenticato
- HttpSessionContextIntegrationFilter: associa Security Context alla sessione.
- Authentication: rappresenta un Principal.
- GrantedAuthority: permessi associati ad un Principal
- UserDetailsService: Il processo di recupero delle informazioni di sicurezza legate ad un user.


<<Interface>>
ObjectDefinitionSource
<<Interface>>
RunAsManager
<<Interface>>
AuthenticationManager
SecurityInterceptionEvent
<<Interface>>
AfterInvocationManager
<<use>>
<<use>>
<<use>>
<<use>>
<<use>>
AbstractSecurityInterceptor
MethodDefinitionSourceAdvisor
<<use>>
FilterSecurityInterceptor
AspectJSecurityInterceptor
MethodSecurityInterceptor
<<use>>
<<use>>
<<use>>
FilterInvocation
JoinPoint
MethodInvocation



- <%@ taglib prefix="authz" uri="http://acegisecurity.org/authz" %>
- authentication: visualizza informazioni utente nelle JSP
- authorization: permette di visualizzare/nascondere sezioni HTML a seconda se l'utente è in uno specifico ruolo (Ideali per la creazione di menu dinamici)

1. Autenticazione BASIC
2. Autenticazione con pagina di login
3. Dati utente memorizzati su DB MySQL
4. Protezione metodi (AOP)
5. Integrazione con il CAS (SSO)

- Parancoe: sviluppato dal JUG Padova (https://parancoe.dev.java.net)
- Principio: “Easy Going Web Application”
- Presentato a Jazoon 2007
- Fortemente basato su Spring, adotta i principi: DRY, SOC, Convention over Configuration
- Annotations vs XML
- Plugins

- Elimina la dichiarazione del filtro da web.xml
- Utilizza Spring Handler Interceptor
- Basta mettere un jar in WEB-INF/lib e ho la security.
- Meccanismo plugin
- Configurazione standard di sicurezza
- Applicativo: associazione URL/ruoli, accesso DB
- Crea le tabelle di security se non ci sono

- ReferenceDocumentation: “http://www.acegisecurity.org/docbook/acegi.html”
- Bart van Riel: “Spring Acegi Tutorial
- Matt Raible: “Acegi Security”

- email: enrico.giurin@jugpadova.it
- Sito Web:
  - `http://www.jugpadova.it`
- Mailing List:
  - `jugpadova@googlegroups.com`
- Persone di riferimento
  - Lucio Benfante (lucio.benfante@jugpadova.it)
  - Dario Santamaria (dario.santamaria@jugpadova.it)