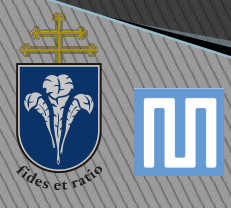
Security

Alapfogalmak

- Autentikáció: Az autentikáció, más néven *partner hitelesítés* vagy *biztonságos azonosítás* azt jelenti, valamilyen biztonságos módon megbizonyosodunk róla, hogy azzal kommunikálunk–e, akivel szeretnénk.
- Autorizáció: felhatalmazás, engedélyezés. Megadott erőforrásokhoz (adatállományokhoz, valamilyen rendszer meghatározott szolgáltatásaihoz) való hozzáférés biztoítása
- UserPrincipal: bejelentkezett felhasználó

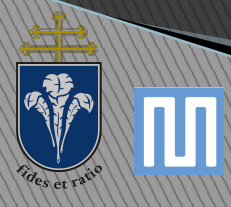
Bevezetés

- Ami nem publikus tartalom, azt védeni kell
- Védhetünk függvényeket
- Védhetünk xhtml felületeket tartalmazó foldereket



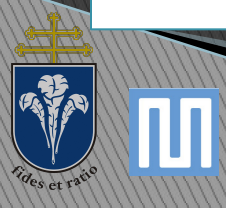
EJB Security

- A konténer nyilvántartja a bejelentkezett felhasználókat, azoknak szerepköreit
- Szabályozható, melyik szerepkör milyen oldalakat láthat, milyen függvényeket hívhat meg
- Hogy melyik EJB függvényt ki hívhatja meg, annotációkkal tudjuk szabályozni

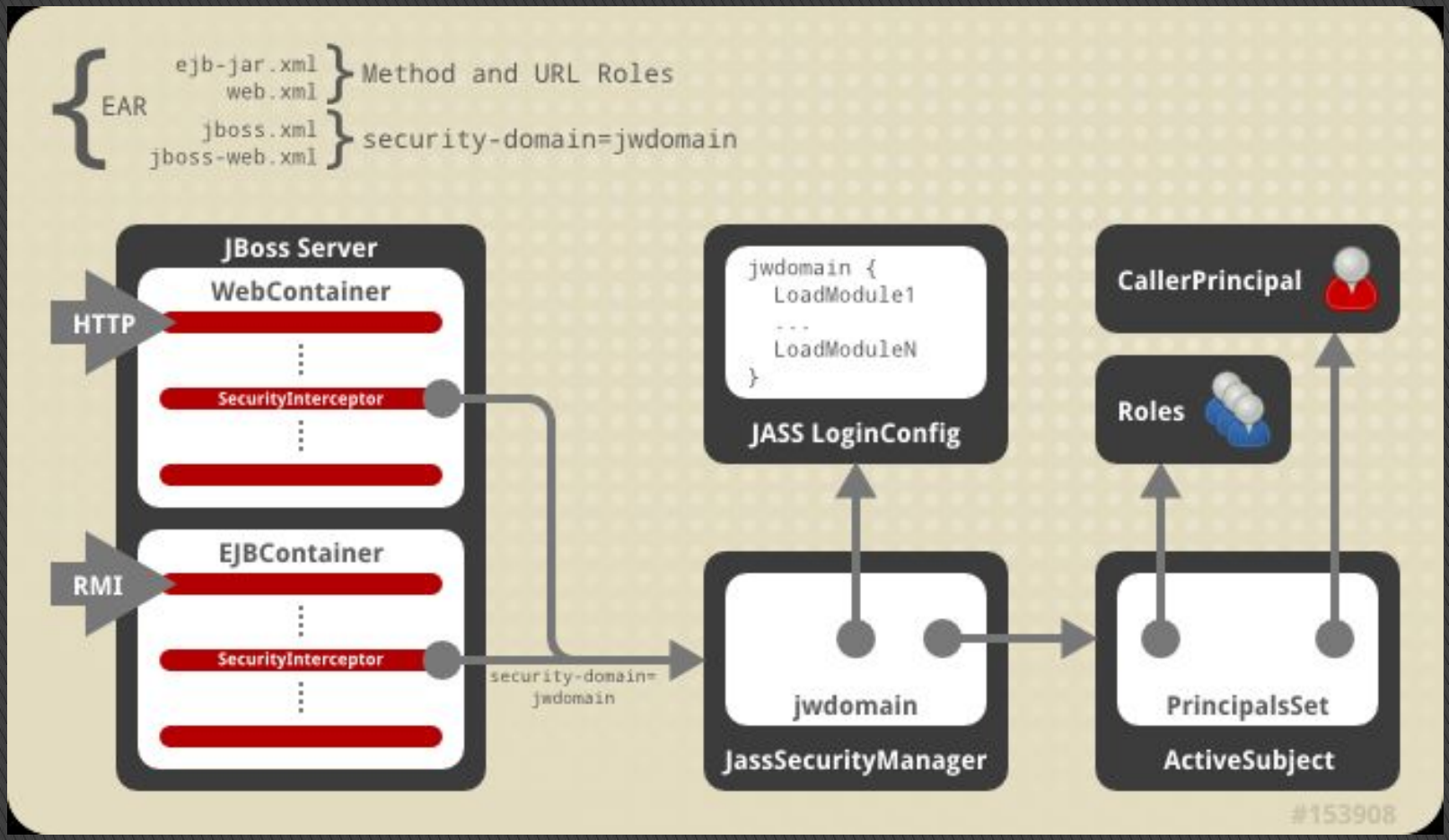


EJB Security példa

```
@Stateless
@RolesAllowed({“ADMIN”, “SUPERUSER”})
public class SampleEJB{

    @PermitAll
    public void sampleFunction(){
    }

    @DenyAll
    public void sampleFunction2(){
    }

    @RolesAllowed({“ADMIN”})
    public void sampleFunction3(){
    }


}
```

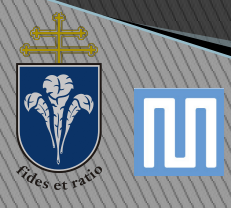


JBOSS Security



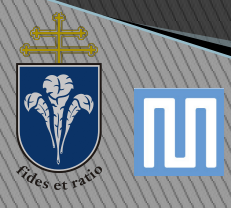
Konfigurációs beállítások

- standalone.xml –> security subsystem, autentikáció
- jboss–web.xml –> security domain
- web.xml –> autorizáció
- Kód, kód strukturálása –> autorizáció



A biztonsági konfiguráció lépései

- Security domain deklarálása (jboss-web.xml)
- Server konfigurálása (standalone.xml)
- Alkalmazás oldali login oldal létrehozása
- Alkalmazás konfigurálása (web.xml): login, jogosultságok

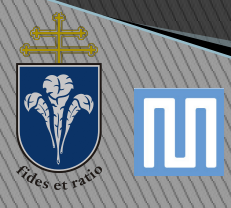


Security domain

- **Security Domain** – A containerbe deployolandó alkalmazások által használható biztonsági beállításgyűjtemény.
  - Többet is meghatározhatunk rendszerenként, az alkalmazás dönti el, melyiket használja.
  - Többek között a használható bejelentkeztetési módokat és ezek beállításait tartalmazza (pl. LDAP, Database, Kerberos…)
- **`web.xml`** – a JavaEE webalkalmazás deployment-descriptora.
- **`jboss-web.xml`** – a JavaEE webalkalmazás JBoss-specifikus beállításai (web.xml kiegészítése)
  - Itt kell megadni, hogy a container által kínált Securiy Domainek közül melyiket (milyen nevűt) használja az alkalmazásunk

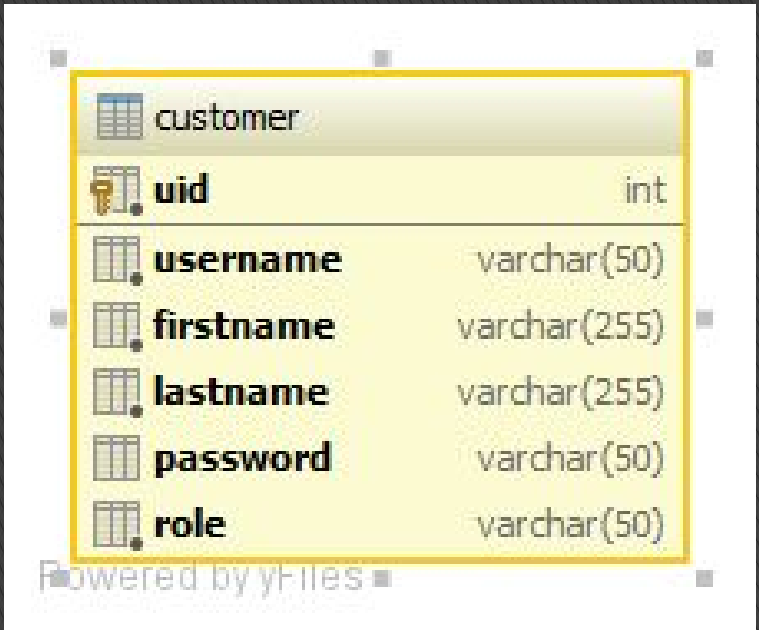
Jboss-web.xml: Security domain

```
<?xml version="1.0" encoding="UTF-8"?>
<jboss-web>
    <security-domain>hj_security</security-    domain>
    <context-root>/lab_5</context-root>
</jboss-web>
```



Adatbázis séma a biztonsági beállításokhoz

- Adatmodell



- SQL select-ek:
  - PrincipalsQuery: select password from customer where username = ?
  - RolesQuery: select role,
    'Roles' from customer where username=?

Szerver oldali konfiguráció - Standalone.xml

```
 <subsystem xmlns="urn:jboss:domain:security:1.2">
 <security-domains>
<security-domain name="hj_security">
  <authentication>
    <login-module code="Database" flag="required">
       <module-option name=„GUEST"
                      value="vendeg"/>
       <module-option name="dsJndiName"
                      value="java:/laborDS"/>
       <module-option name="principalsQuery"
              value="SELECT password FROM customer WHERE
username=?"/>
       <module-option name="rolesQuery"
              value="select role, 'Roles' from customer
                     where username=?"/>
    </login-module>
  </authentication>
</security-domain>
 </security-domains>
</subsystem>
```

Beléptetés a konténerbe

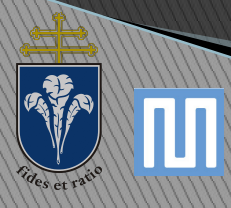
- Form alapú (j_security_check)
- programozott

Beléptetés a konténerbe (j_security_check)

- **login.xhtml:**

```
<form method="post" action="j_security_check">
    [...]
    <h:outputLabel for="j_username" value="Username" />
    <h:inputText id="j_username" />

    <h:outputLabel for="j_password" value="Password" />
    <h:inputSecret id="j_password" />
    [...]
    <h:commandButton value="Login" />
    [...]
</form>
```

- Egy sima(!) html <form>, ahol
- action="j_security_check". Ez a JAAS szabvány szerint submitkor meghívja a konténerben beállított login-hívást.
- A két JSF-mezőt `id="j_username"` illetve `id="j_password"` teszi ennek paramétereivé.



Beléptetés a konténerbe (programozott)

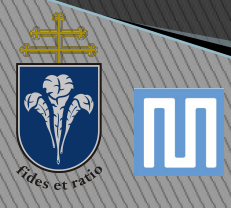
```
ExternalContext extc =
            FacesContext
                .getCurrentInstance()
                   .getExternalContext();
 final HttpSession httpSession =
    (HttpSession) extc.getSession(false /* don't create */);

HttpServletRequest request =
       (HttpServletRequest) extc.getRequest();

request.login(String username, String password); /* programozott
  beléptetés*/

request.isUserInRole(String role); /* felhasználónak van-e joga
  az adott szerepkörhöz - Expression Language-ből is elérhető*/

request.getUserPrincipal(); /* visszaadja a bejelentkezett
  felhasználó objektumot*/
      request.getUserPrincipal().getName(); /* visszaadja a
      bejelentkezett felhasználónevet*/
```

Alkalmazás-oldal: web.xml

- **`Web.xml`** – JavaEE standard deployment descriptor. Fontosabb elemek:
  - login-config (bejelentkezés módja)
  - security-constraint (melyik web-resource-hoz mi alapján lehet hozzáférni)
  - security-role
  - filter, filter-mapping…
- login-config példa:

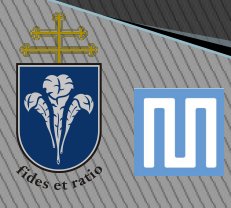
```
<login-config>
 <auth-method>FORM</auth-method>
     <realm-name>hj_security</realm-name>
      <form-login-config>
    <form-login-page>/login/login.xhtml</form-login-page>
    <form-error-page>/login/loginError.xhtml</form-error-page>
  </form-login-config>
</login-config>
```

- **`auth-method`** típusai:
  - `BASIC`: HTTP-alapú auth, böngésző login-formja.
  - `FORM`: én mondom meg, milyen form léptet be.
  - `CLIENT-CERT`: HTTPS kliens-alapú authentikáció, digitális certificate használatával



Jogosultságok beállítása

```
<security-constraint>
   <display-name>AdminConstraint</display-name>
   <web-resource-collection>
    <web-resource-name></web-resource-name>
    <url-pattern>/customer/*</url-pattern>
     <url-pattern>/product/*</url-pattern>
   </web-resource-collection>
   <auth-constraint>
     <role-name> ADMIN </role-name>
   </auth-constraint>
</security-constraint>

<security-role>
   <description>Admin pages</description>
   <role-name>ADMIN</role-name>
</security-role>
```



Logout

```
public void logout() {

  // Elkérjük a HttpSessiont.
  ExternalContext extc =
              FacesContext
               .getCurrentInstance()
                  .getExternalContext();

  final HttpSession httpSession =
    (HttpSession) extc.getSession(false /* don't create */);

  // Maga a kiléptetés.
  httpSession.invalidate();

  // Küldjük az főoldalra.
  extc.redirect(extc.getRequestContextPath()
                    + "/index.xhtml");
}
```