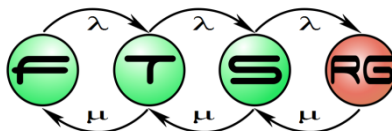


Eszközintegráció OSGi segítségével



Eszközintegráció

- **Feladat:** Összetett alkalmazás fejlesztés
 - (Általában) Nem zöldmezős beruházás
 - Részfeladatokra már vannak eszközeink
 - Ezeket közösen kezelnénk
- Hasonló problémakör
 - Fejlesztőeszközök integrációja
 - Eclipse környezet

OSGi

- „A dynamic module system for Java”
- OSGi szövetség (www.osgi.org)
 - Széleskörű tagság:
 - Adobe
 - IBM
 - Oracle
 - Red Hat
 - Siemens AG
 - VMware



- Közös problémák
 - Integráció
 - Verziófrissítés
 - Életciklus
- OSGi szabvány (specifikáció)
 - Komponens alapú
 - Közös integrációs „primitívekkel”
- Jelenleg: R5.0 (2012. június)

OSGi compendium

- OSGi core specifikáció kiegészítése
- Szolgáltatások jegyzéke
- Fontosabb szolgáltatás csoportok
 - Declarative Services: Deklaratív modell szolgáltatások kezelésére
 - Http Service: Http alapú szolgáltatások
 - Remote Services: Szolgáltatások használat különböző framework-ök között (Version 1.0)
 - Log Service: általános célú logolásra alkalmas szolgáltatások
 - stb...

**OSGi Service Platform
Service Compendium**
The OSGi Alliance

Release 4, Version 4.2
August 2009



OSGi implementációk

- Open Source
 - Eclipse Equinox (<http://www.eclipse.org/equinox/>)
 - Apache Felix (<http://felix.apache.org/>)
 - Knopflerfish (<http://www.knopflerfish.org/>)
 - ProSyst mBedded Server Equinox Edition (http://www.prosyst.com/products/osgi_se_equi_ed.html)
- Fizetős:
 - ProSyst (<http://www.prosyst.com/>)
 - Knopflerfish Pro (<http://www.makewave.com/>)

Hol használják

❖ Eclipse IDE és

eRCP
Nokia
Sprint

NASA
JPMorgan
Lotus
Jazz
SAS
Swiss Rail
Daimler

Rational Suite
Borland
BEA
Jazz

WAS
BEA
Jazz
Spring

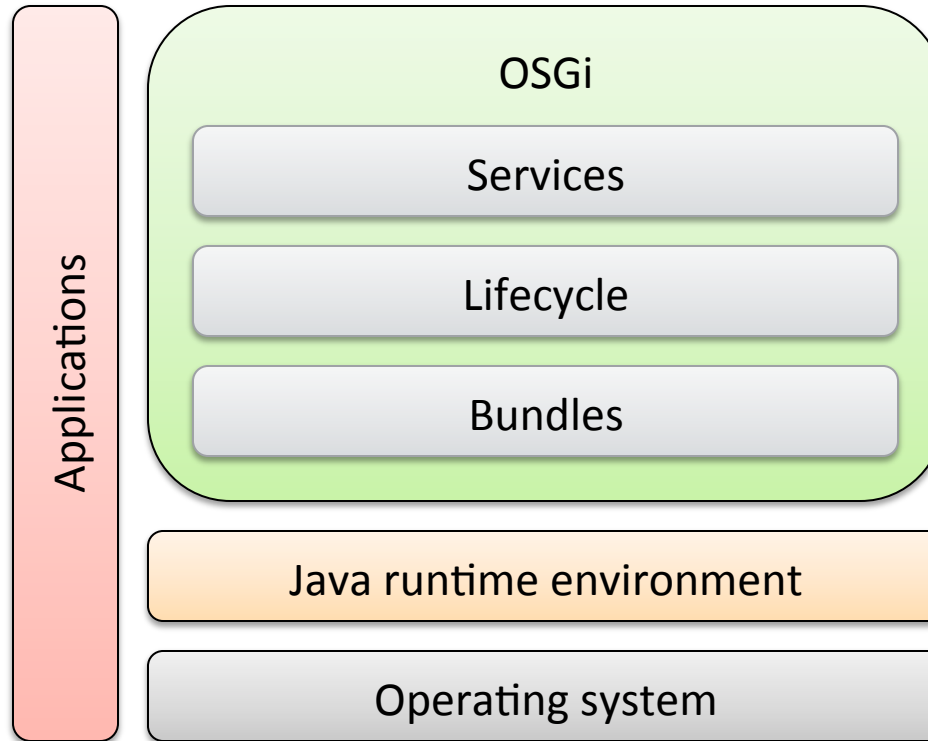
Embedded

Rich Client

Tooling

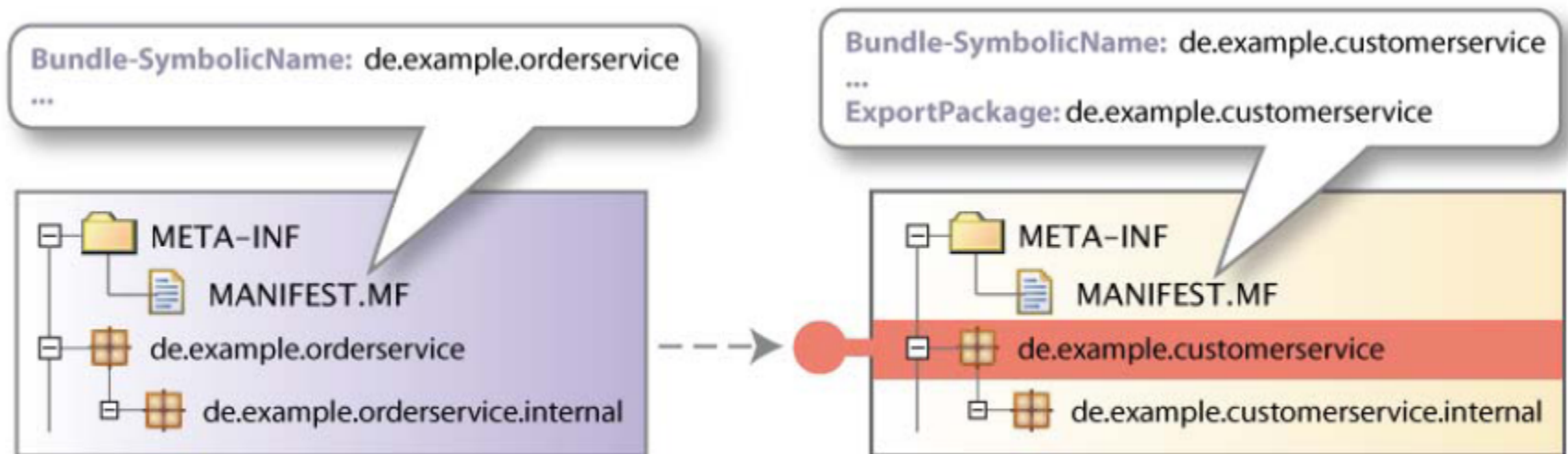
Server

OSGi



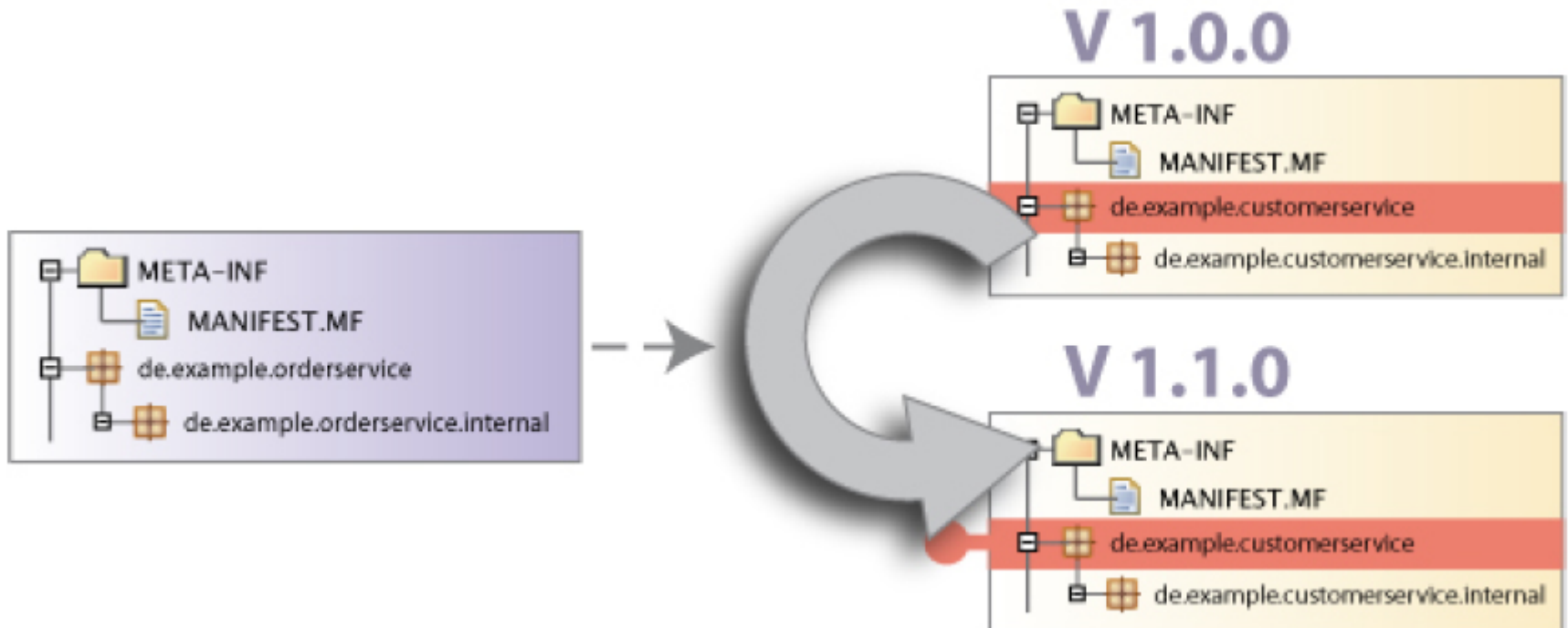
OSGi alapok: modulok

- Modulok (bundles)
 - Public és private API láthatósága
 - Függőség kezelés
 - Verziózás

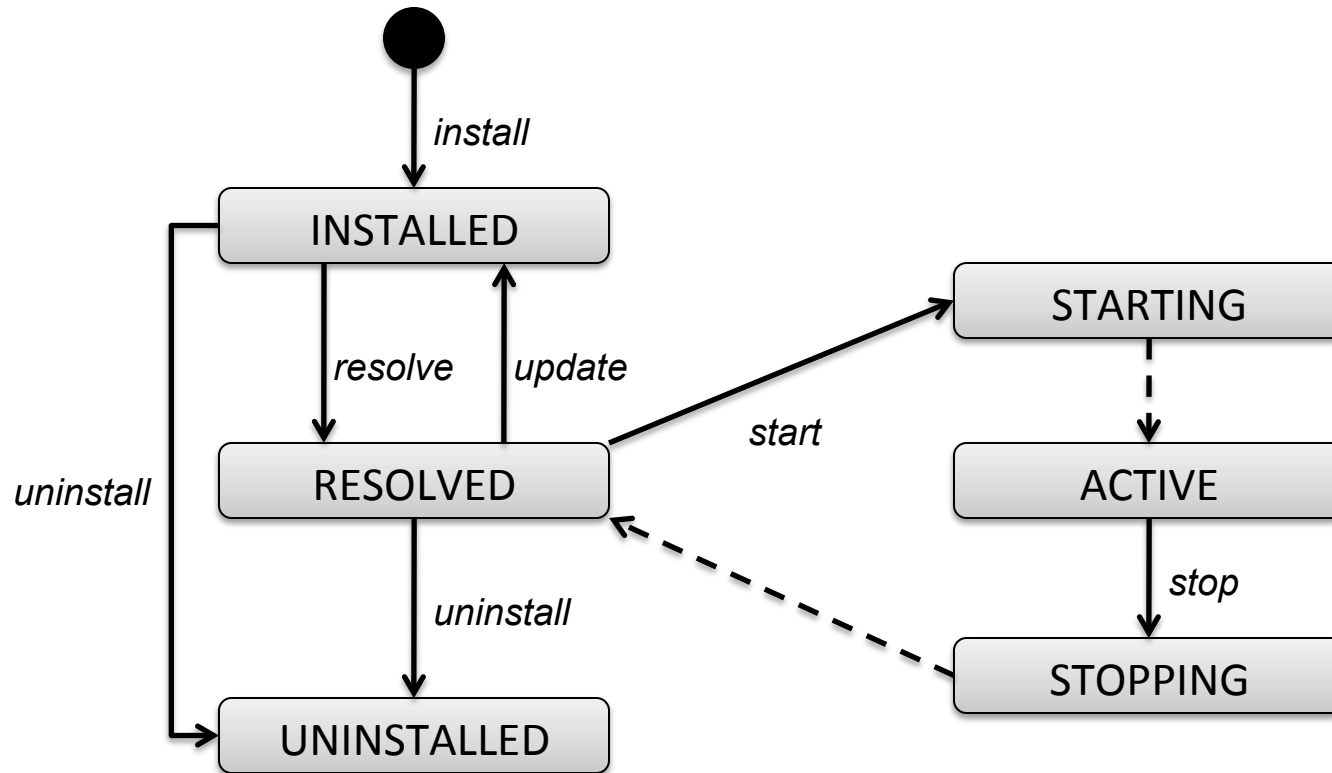


OSGi alapok: életciklus

- Életciklus (Life cycle)
- Dinamikus Bundle:
 - Betöltés (install)
 - Start/stop

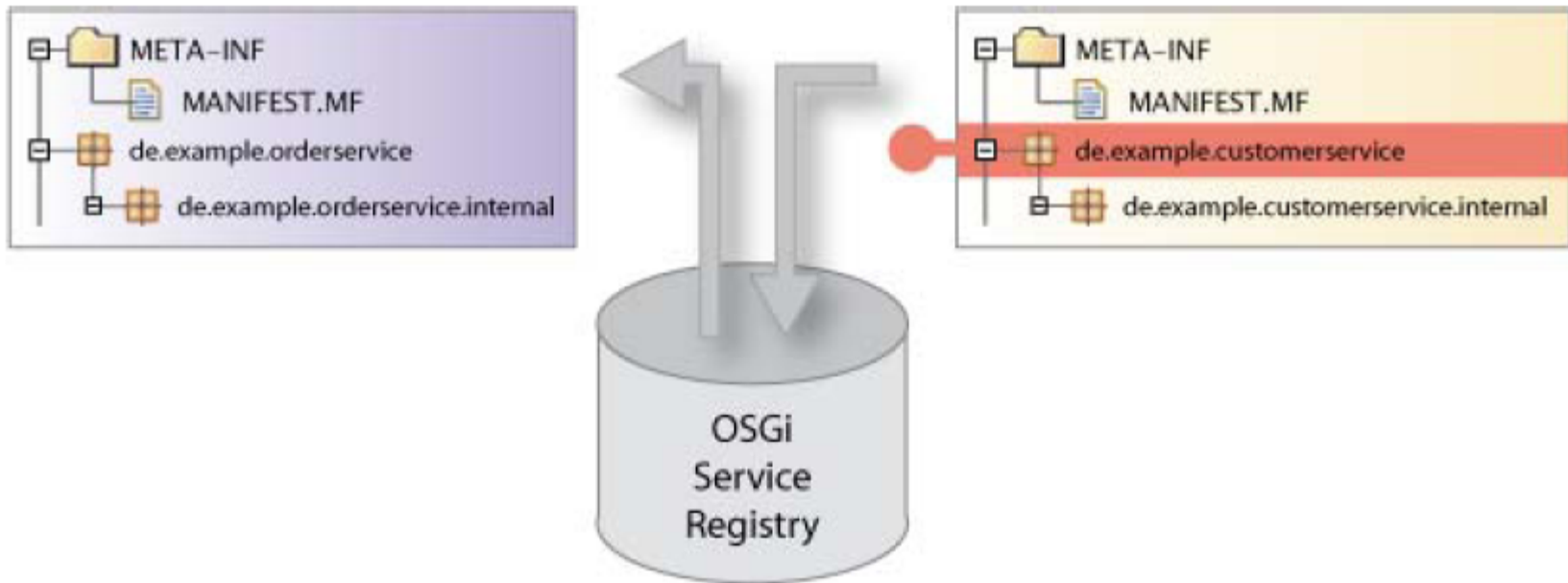


OSGi alapok: Bundle életciklus modell



OSGi alapok: szolgáltatások

- Szolgáltatás orientált (Services):
 - Modulok szolgáltatásokat ajánlhatnak ki
 - Kereshetőek (service registry)
 - Futás idő közben megjelenhetnek(/eltűnhetnek)!



OSGi Bundles

Modul réteg

- A modulok indíthatók, leállíthatók
- A futó bundle-k szolgáltatásai kiajánlásra kerülnek
- Fontos manifeszt adatok
 - Bundle-activator: az életciklus menedzselését végző osztály neve
 - Bundle-classpath: a bundle-n belüli classpath-ok listája. A default értéke: . (a bundle root)
 - Bundle-name: olvasható név
 - Bundle-SymbolicName: azonosító (egyedi)
 - Bundle-UpdateLocation: URL, ahonnan a frissítések letölthetők
 - Export-Package: a kiajánlott java csomagok listája
 - Import-Package: importált csomagok listája
 - Require-Bundle: szükséges modulok listája (import + függőségek)

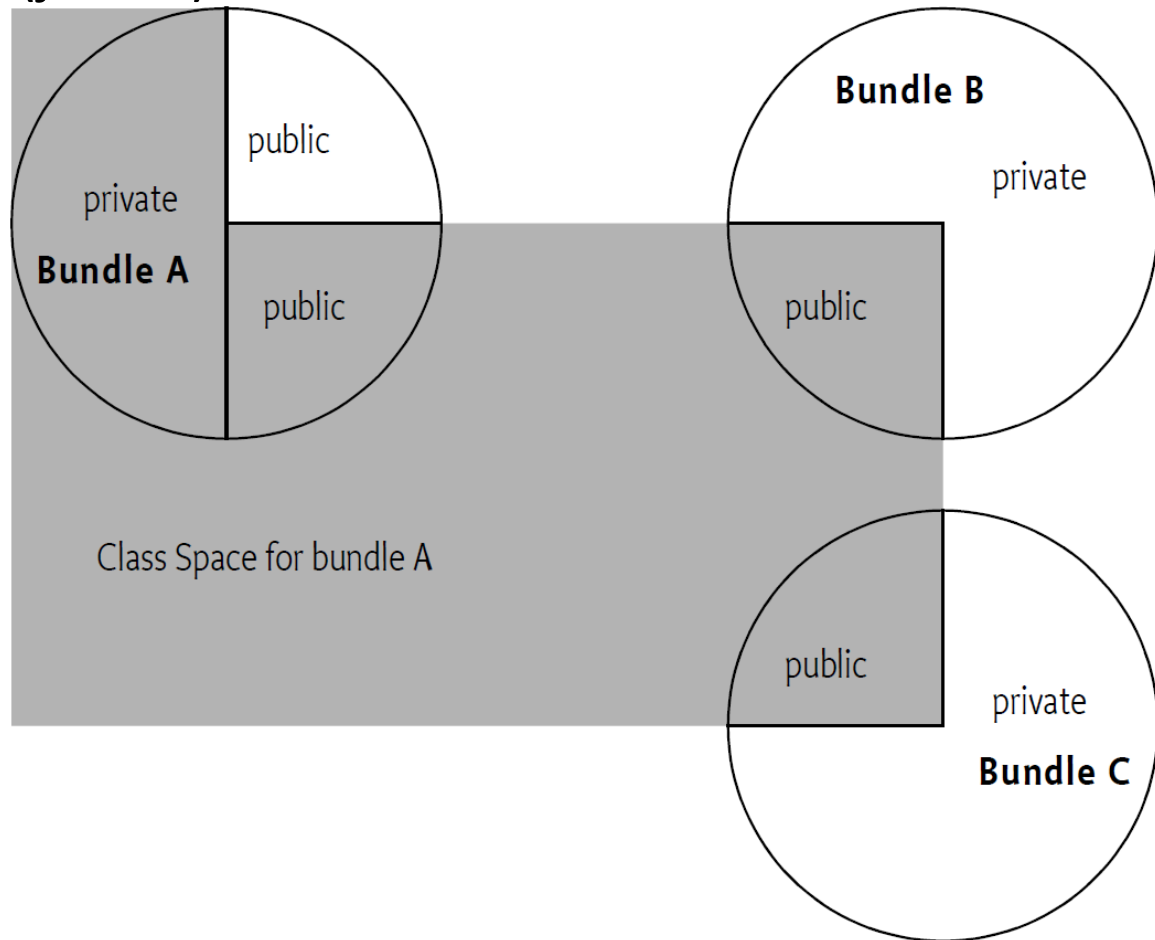
Class loading

- Minden bundle egy vm-en belül fut
- Minden bundle-nak saját class loadere van
 - 3 helyről tölthet be osztályokat/erőforrásokat
 - Boot class path: java.* csomagok és implementációik
 - Framework class path: a framework-nek saját class loadere van, amitől elkérhetőek az interfészek és implementáló osztályok
 - Bundle space: a bundle jar fájljai, valamint a hozzá kötődő egyéb jar-ok

Class loading – Class space

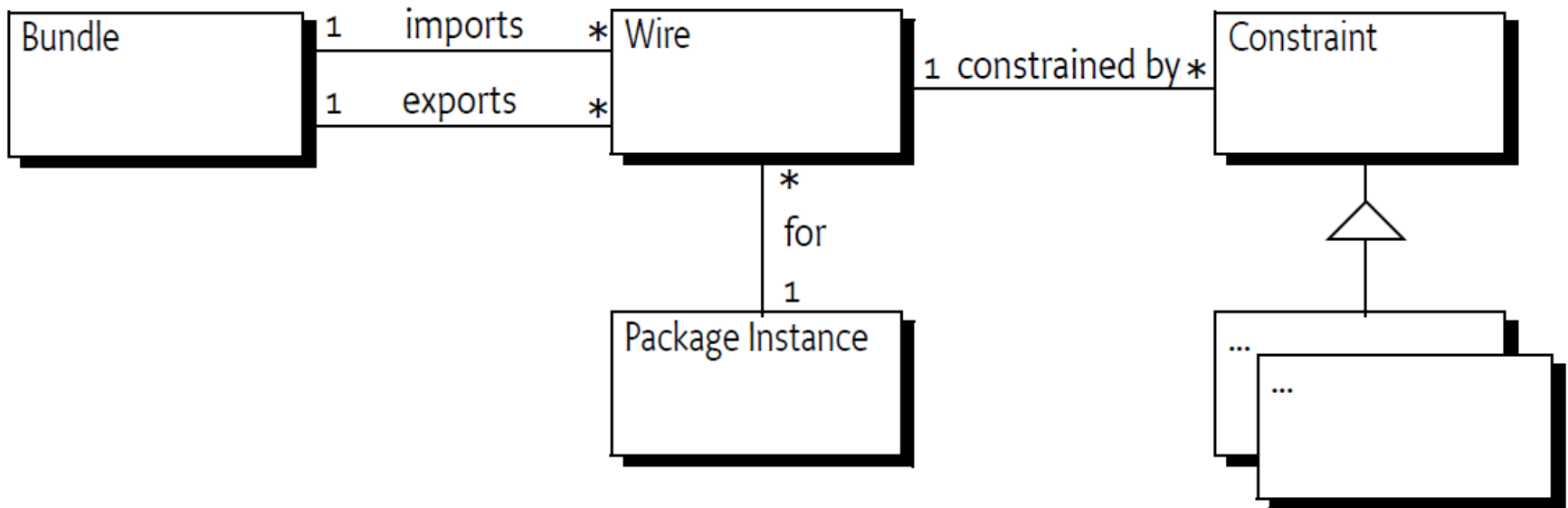
■ Egy bundle class space-e

- A szülő class loader-e (java.*)
- Importált csomagok
- Függőségek
- A bundle privát classpath-a
- Csatolt fragmensek



Bundle feloldás

- Feloldás: az importerek és exporterek összekötése
 - Kényszereknek megfelelően
- Vezeték (wire): összeköttetés importer és exporter között



Metaadat feloldás

■ Bundle-SymbolicName

- kötelező, egyedi azonosító
- Ha két egyező nevű és verziójú van, a második telepítése sikertelen
- Paraméterek
 - Singleton: csak egyetlen verziója lehet betöltve
 - Fragment-attached: definiálja, hogyan lehet fragmenseket hozzákapcsolni
 - Always: bármikor kapcsolódhat
 - Never: nem lehetséges
 - Resolve-time: csak a resolve fázisban

○ Példa: `Bundle-SymbolicName: com.acme.foo;singleton:=true`

■ Bundle-Version

- Meghatározott formátum: major.minor.micro.qualifier
- Összehasonlítás hierarchikus
 - Numerikusan, illetve a qualifier esetén String.compareTo
 - Két verzió akkor azonos, ha minden szegmensük egyezik

○ Példa: `Bundle-Version: 22.3.58.build-345678`

Metaadat feloldás

■ Imported-packages

- Importált csomagok listája
- Resolution – a csomagot fel kell oldani kötelező import esetében, ha ez sikertelen a bundle sem tölthető be
- Version – verzió intervallum a csomagot exportáló csomagra zárt [], nyitott (), pl. [1.0.0,2.0.0)
- Bundle-version: az exportáló bundle verziója
- Bundle-symbolic-name: az exportáló bundle neve
- Példa:

```
Import-Package: com.acme.foo;com.acme.bar;  
                version="[1.23,1.24]";  
                resolution:=mandatory
```

■ Exported-packages

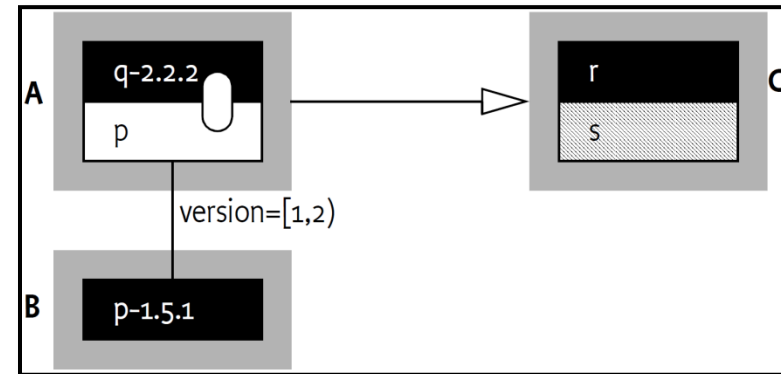
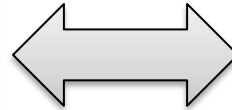
- Exportált csomagok listája
- Hasonlóan az Imported-packages-hez
- Példa:

```
Export-Package: com.acme.foo;com.acme.bar;version=1.23
```

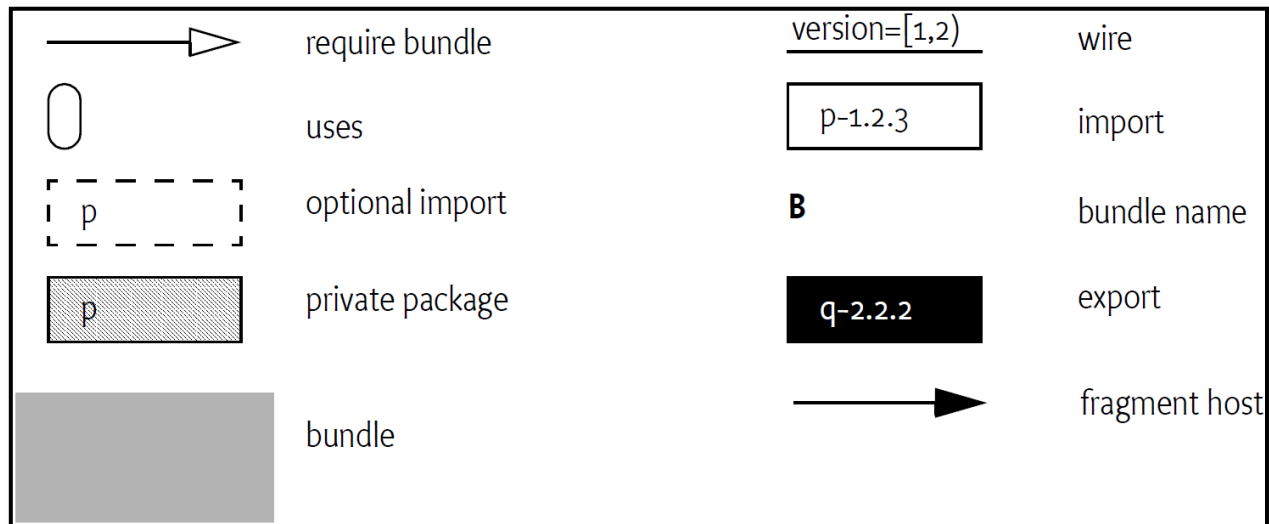
Bundle diagram

A: Import-Package: p; version="[1,2)"
 Export-Package: q; version=2.2.2; uses:=p
 Require-Bundle: C
 B: Export-Package: p; version=1.5.1
 C: Export-Package: r

Szöveges leírás



Grafikus ábrázolás



Jelölés

Újdonság: Capabilities

- Capability
 - Függőségi referenciák általánosítása
 - Tetszőleges, szöveges azonosítójú tulajdonság
 - Verziózható
- Require-Capability
 - Függőség jelzése képességre
- Provide-Capability
 - Képesség biztosításának jelzése

OSGi Services

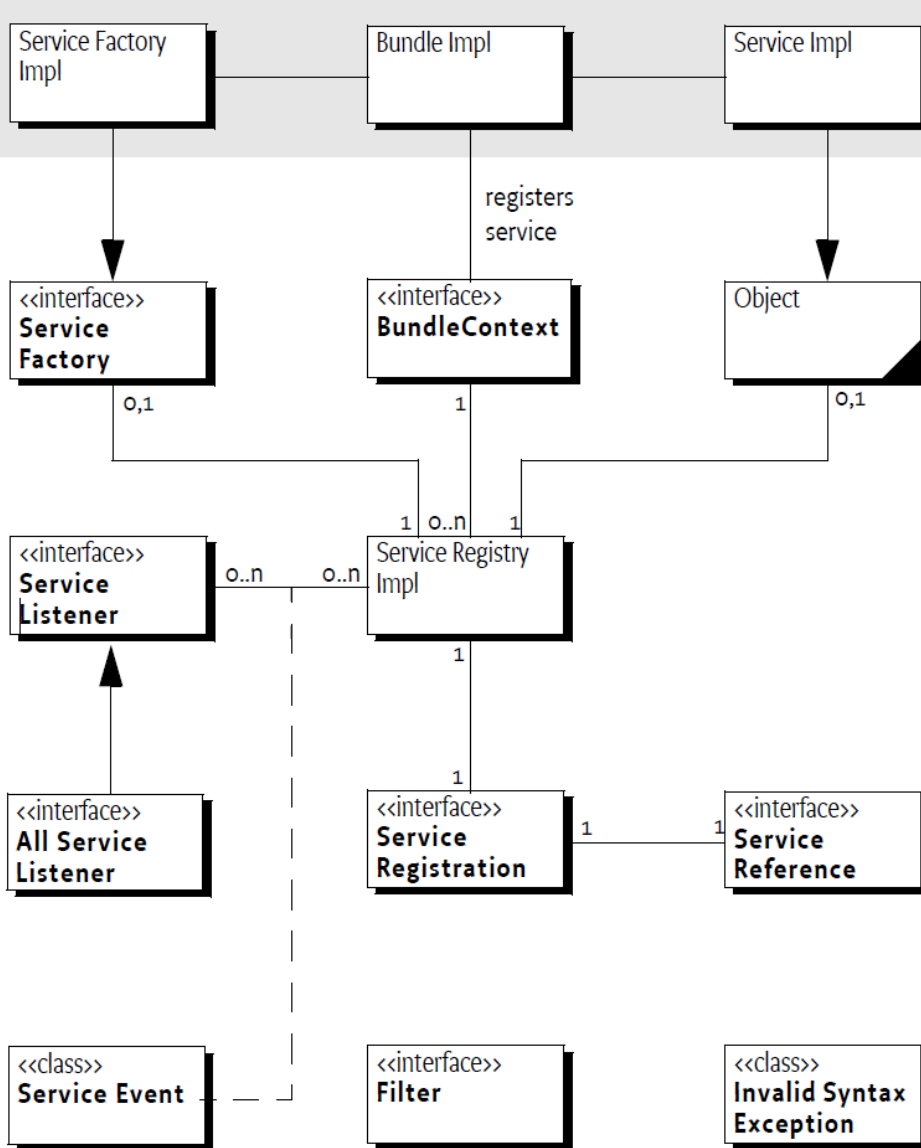
Szolgáltatási réteg

- Definiálja az együttműködési modellt
 - „Publish, find, and bind”
 - A szolgáltatás egy normál java objektum
 - Regisztrálódik egy vagy több java interfész alatt
- A bundle-k
 - Regisztrálhatnak
 - Kereshetnek
 - Használhatnak szolgáltatásokat
 - Illetve, ezekkel kapcsolatban eseményeket kezelhetnek

Szolgáltatási réteg - alaptulajdonságok

- Kollaboratív: bundle-k közötti együttműködés megvalósítása
- Dinamikus: futásidejű változások
 - Új szolgáltatások megjelenése
 - Régi szolgáltatások megszűnése
- Biztonságos: hozzáférés korlátozható
- Reflektív: teljes hozzáférés a réteg belső állapotához
- Verziókezelés: a szolgáltatások frissülhetnek
- Perzisztens id: framework indítások között is lehet a szolgáltatásokat követni

Szolgáltatási réteg elemei



Service

- Szolgáltatás implementációja

ServiceRegistry

- Beregisztrált szolgáltatások tárolója

ServiceReference

- Szolgáltatás tulajdonságai eléréséhez

ServiceRegistration

- Szolgáltatás tulajdonságainak leírása

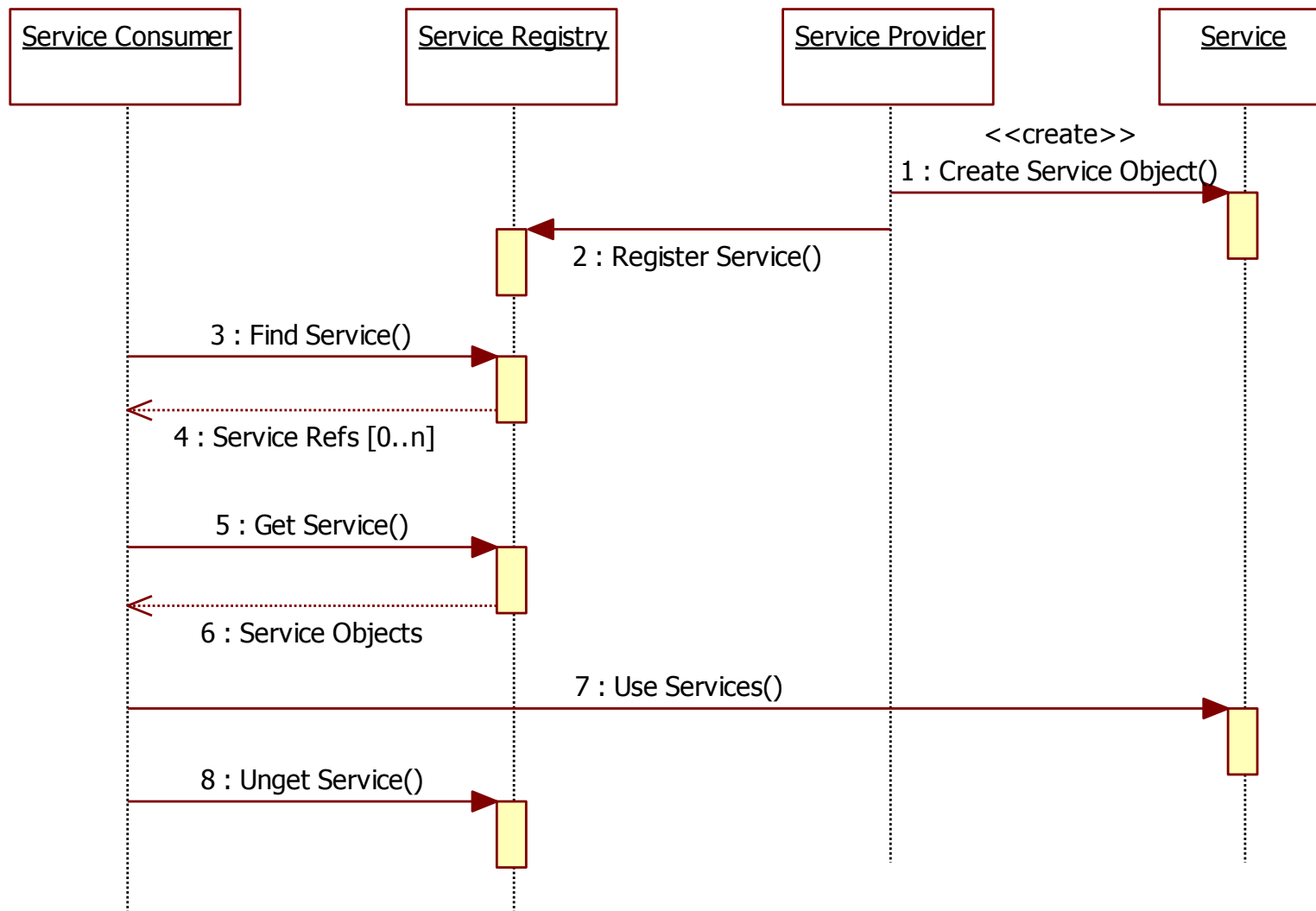
ServiceListener

- Pl. szolgáltatás megjelenés/eltűnés események

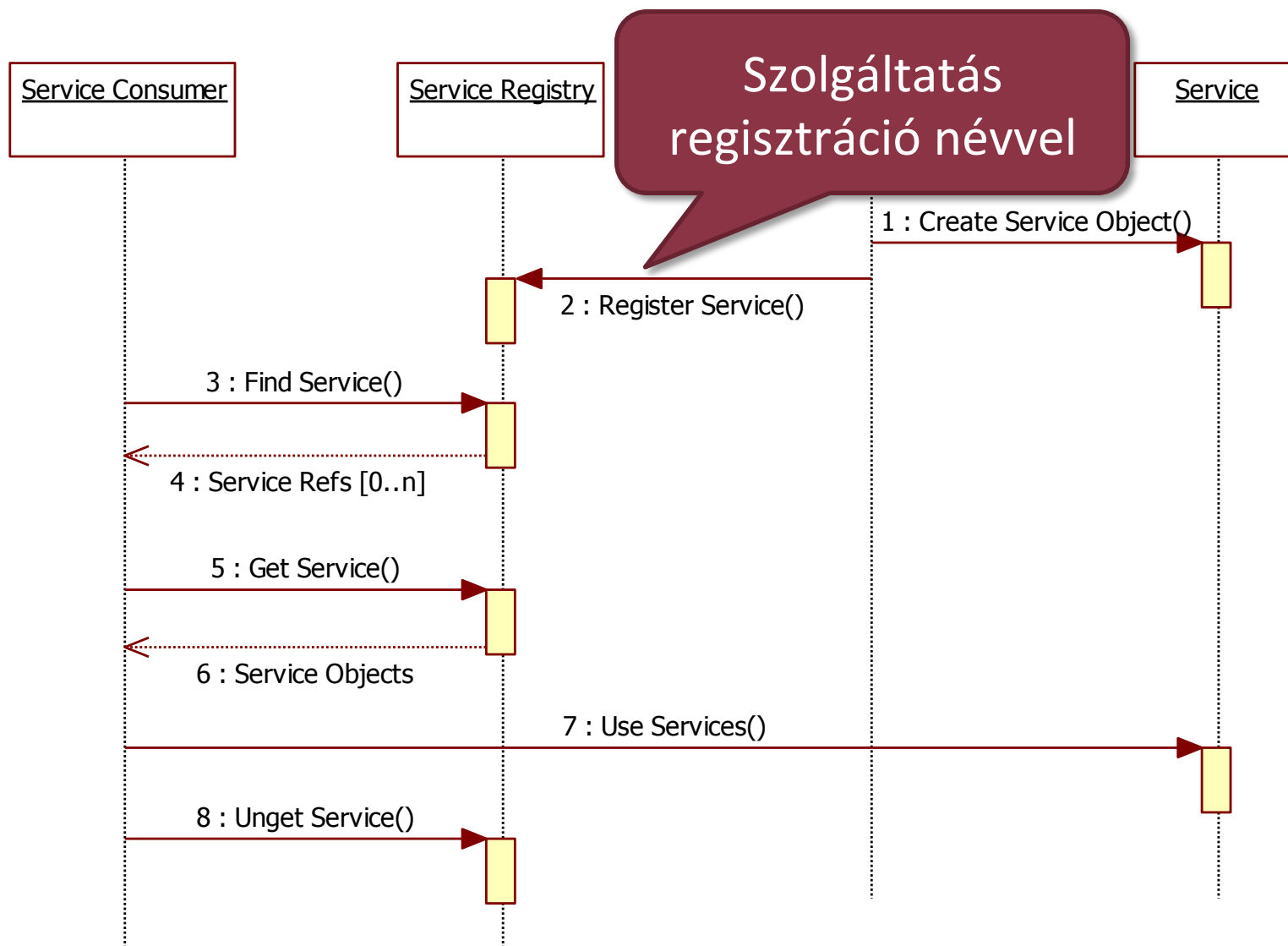
Filter

- Egyszerű attribútum alapú szűrő

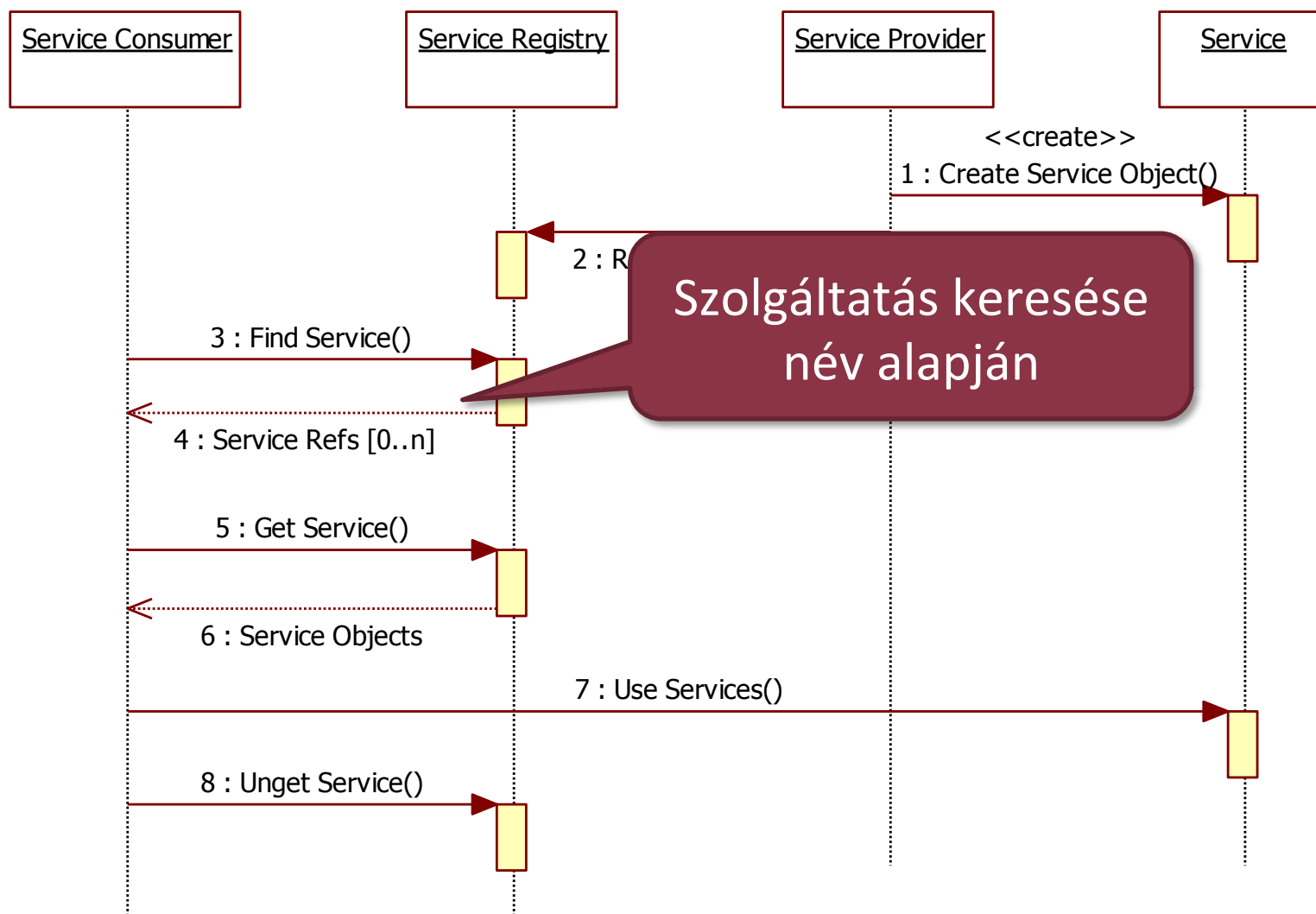
Szolgáltatások közvetlen elérése



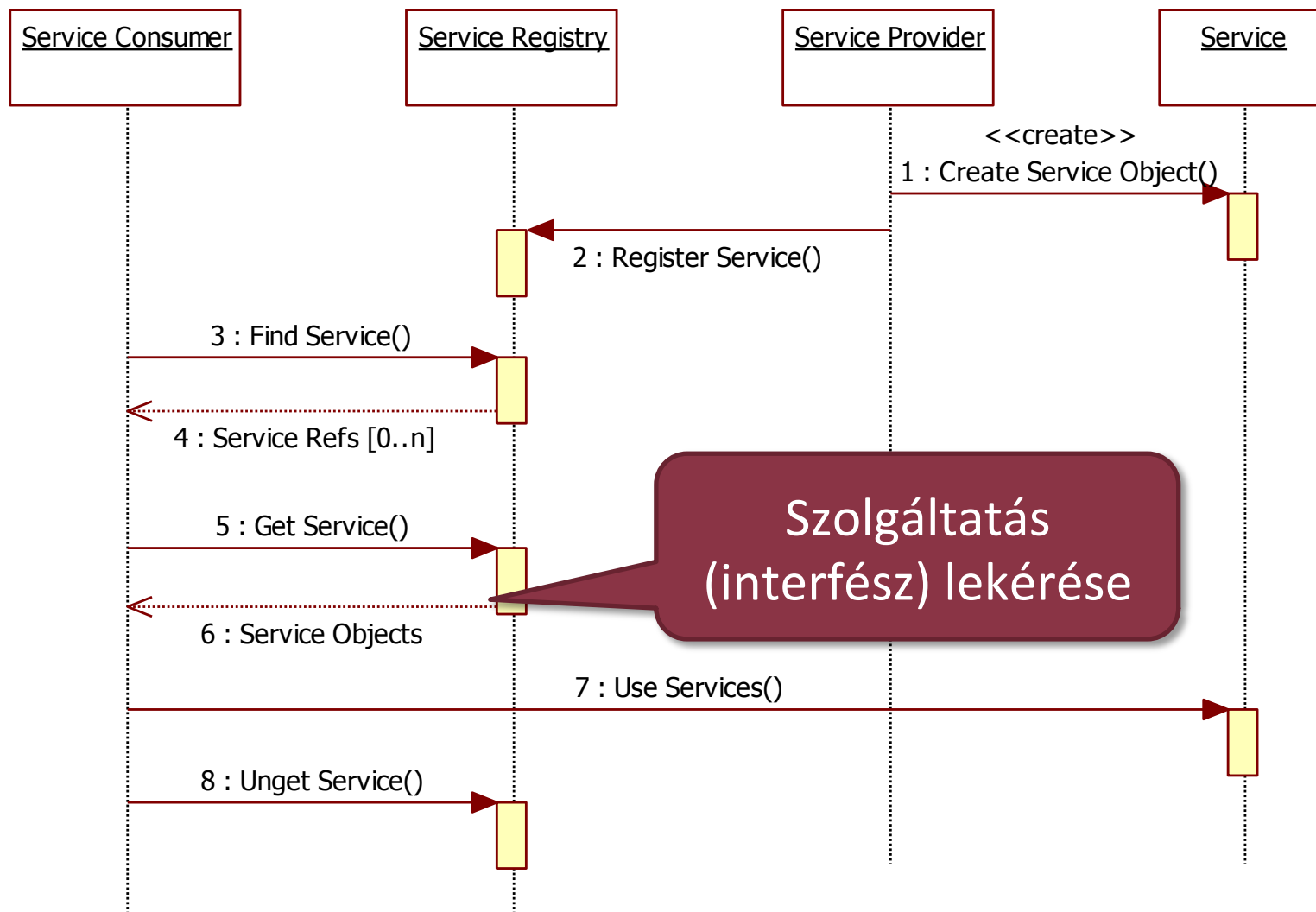
Szolgáltatások közvetlen elérése



Szolgáltatások közvetlen elérése



Szolgáltatások közvetlen elérése



Problémák

- Nincs értesítés szolgáltatás megszűnéséről
 - Szolgáltatás objektum tárolásához szükséges!
 - ServiceListener szóbajöhet, de könnyű hibázni
- Alacsony szintű API
 - Körülményes
 - Sok, ismétlődő Java kód
- Megoldás
 - ServiceTracker
 - Deklaratív szolgáltatások

Service Tracker

- Szolgáltatáshoz lehet regisztrálni
- Jelez, ha
 - Megjelenik egy, az adott típusú szolgáltatás
 - Eltűnik egy, az adott típusú szolgáltatás
 - Módosul egy, az adott típusú szolgáltatás

Declarative Services (OSGi compendium)

- Fejlesztő a szolgáltatások írására koncentrálhat
- Válasszuk szét a felelőségeket:
 - Implementáció: továbbra is bundle felelőssége
 - Regisztráció: Service Component Runtime (SCR)
- „Publish, find, and bind” deklaratív módon
 - XML leírók (komponens leíró)
- Szolgáltatások függőségeinek automatikus menedzselése
 - Dinamikus
 - „On demand” betöltés
 - Komponens elérhetetlen, ha a függő szolgáltatás megszűnik

DS – komponens leíró

- XML alapú
- Deklaratív módja a szervizek regisztrálásának, kötésének
- OSGI-INF könyvtárban
- Több komponens egy bundle-ban
- MANIFEST.MF-ban összeset fel kell venni
 - Service-Component

Példa komponensleíróra

```
<scr:component xmlns:scr="http://www.osgi.org/
  xmlns/scr/v1.1.0"
  name="sample.component">
  <implementation
class="org.sample.HelloServiceImpl"/>
  <service>
    <provide interface="org.sample.HelloService"/>
  </service>
  <reference
    bind="setService"
    unbind="unsetService"
    cardinality="0..1"
    interface="org.sample.ServiceForHello"
    name="„SERVICEFORHELLO"
    policy="dynamic"/>
</scr:component>
```

Példa komponensleíróra

```
<scr:component xmlns:scr="http://www.osgi.org/  
  xmlns/scr/v1.1.0"
```

```
  name="sample.component">
```

```
    <implementation
```

```
    class="org.sample.HelloServiceImpl"
```

```
    <service>
```

```
      <provide interface="org.sample.HelloService"/>
```

```
    </service>
```

```
  <reference
```

```
    bind="setService"
```

```
    unbind="unsetService"
```

```
    cardinality="0..1"
```

```
    interface="org.sample.ServiceForHello"
```

```
    name="SERVICEFORHELLO"
```

```
    policy="dynamic"/>
```

```
</scr:component>
```

Kiajánlott szolgáltatás
definíciója

Példa komponensleíróra

```
<scr:component xmlns:scr="http://www.osgi.org/
  xmlns/scr/v1.1.0"
  name="sample.component">
  <implementation
class="org.sample.HelloServiceImpl"/>
  <service>
    <provide interface="org.sample.HelloService"/>
  </service>
  <reference
    bind="setService"
    unbind="unsetService"
    cardinality="0..1"
    interface="org.sample.ServiceForHello"
    name="SERVICEFORHELLO"
    policy="dynamic"/>
</scr:component>
```

Függő szolgáltatás
megadása

Példa komponensleíróra

```
<scr:component xmlns:scr="http://www.osgi.org/
  xmlns/scr/v1.1.0"
  name="sample.component">
  <implementation
class="org.sample.HelloServiceImpl"/>
  <service>
    <provide interface="org.sample.HelloService"/>
  </service>
  <reference
    bind="setService"
    unbind="unsetService"
    cardinality="0..1"
    interface="org.sample.ServiceForHello"
    name="„SERVICEFORHELLO"
    policy="dynamic"/>
</scr:component>
```

DS – komponens leíró

- Eclipse támogatás: Declarative Service Tooling
 - Component Definition Editor

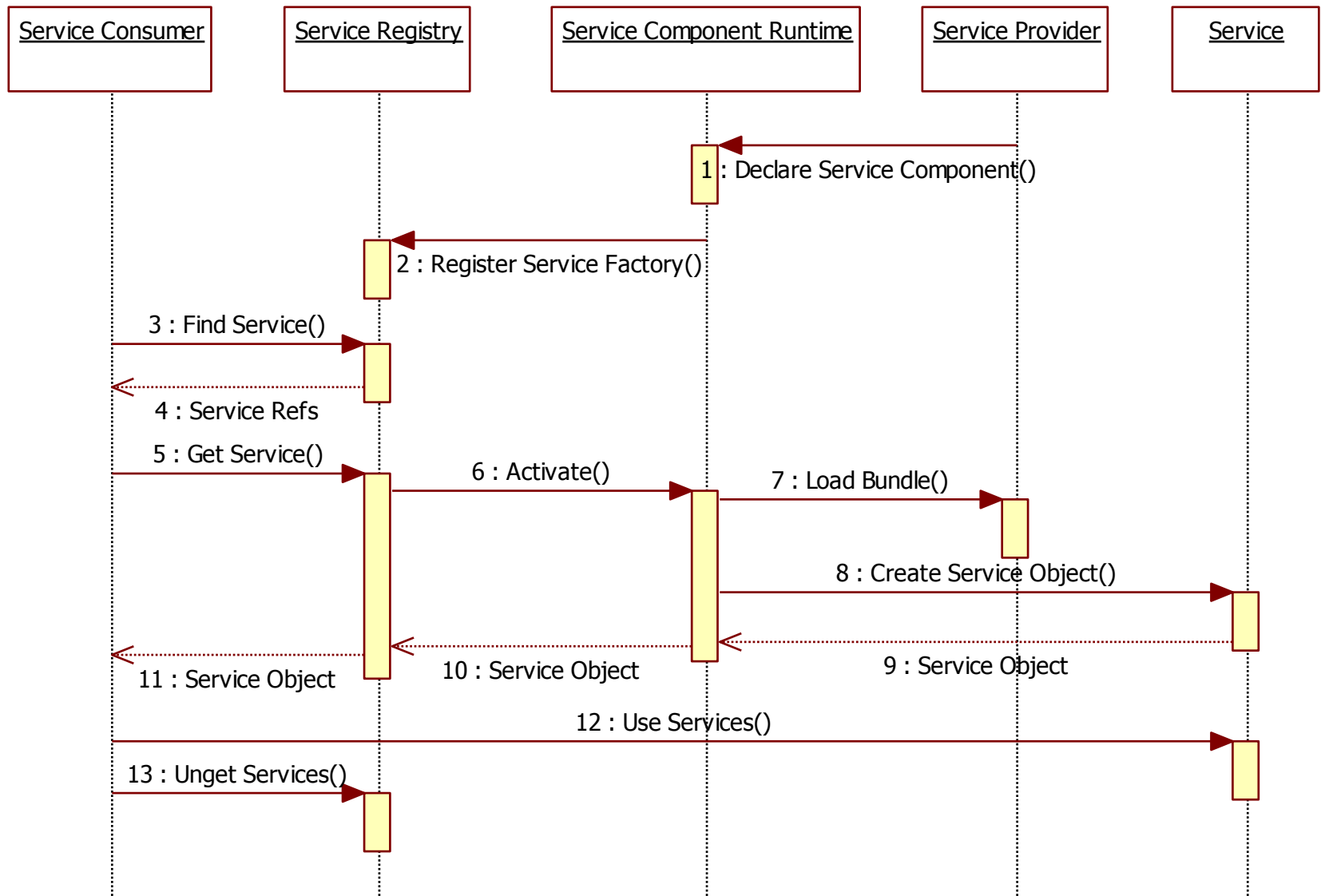
The screenshot shows the Eclipse IDE's Component Definition Editor with the 'Overview' tab selected. The window title is 'component.xml'. The 'Component' section contains fields for 'Name' (sample.comonent), 'Class*' (org.sample.HelloServiceImpl with a 'Browse...' button), 'Activate', 'Deactivate', and 'Modified'. The 'Options' section includes 'Factory ID', 'Configuration Policy' (a dropdown), and two checkboxes: 'This component is enabled when started' (checked) and 'This component is immediately activated'. The 'Properties (0)' section has a large empty text area and buttons for 'Add File...', 'Add Property...', 'Edit...', 'Remove', 'Up', and 'Down'. The bottom tab bar shows 'Overview', 'Services', and 'Source'.

The screenshot shows the Eclipse IDE's Component Definition Editor with the 'Services' tab selected. The window title is 'component.xml'. The 'Referenced Services (1)' section lists '0..1 SERVICEFORHELLO [setService,unsetService]' with buttons for 'Add...', 'Remove', 'Edit...', 'Up', and 'Down'. The 'Provided Services (1)' section lists 'org.sample.HelloService' with buttons for 'Add...', 'Remove', and 'Edit...'. The bottom tab bar shows 'Overview', 'Services', and 'Source'.

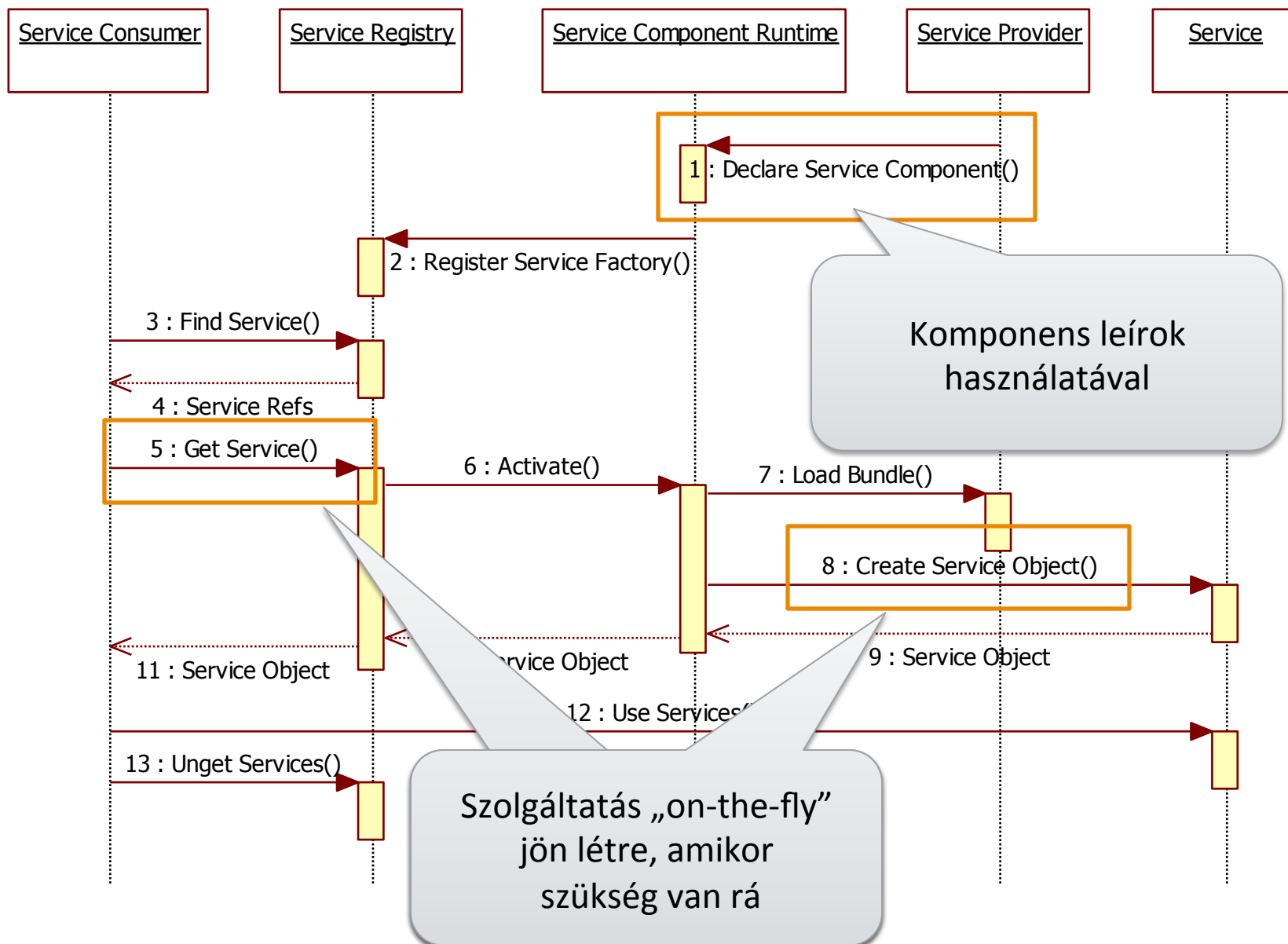
DS – komponensek összekapcsolása

- Szükséges adatok függő szolgáltatás megadására
 - Számosság
 - 0..1 → opcionális, egy értékű
 - 1..1 → kötelező, egy értékű
 - 1..n → kötelező, több értékű
 - 0..n → opcionális, több értékű
 - Policy
 - Dynamic: a szolgáltatás menet közben „kicserélhető”
 - Static: garantáltan egy szolgáltatás objektum lesz végig (pl. állapottal rendelkező szolgáltatás esetében)
 - Szolgáltatás interfész
 - Szolgáltatás név ebből származik!

DS – Életciklus menedzselés



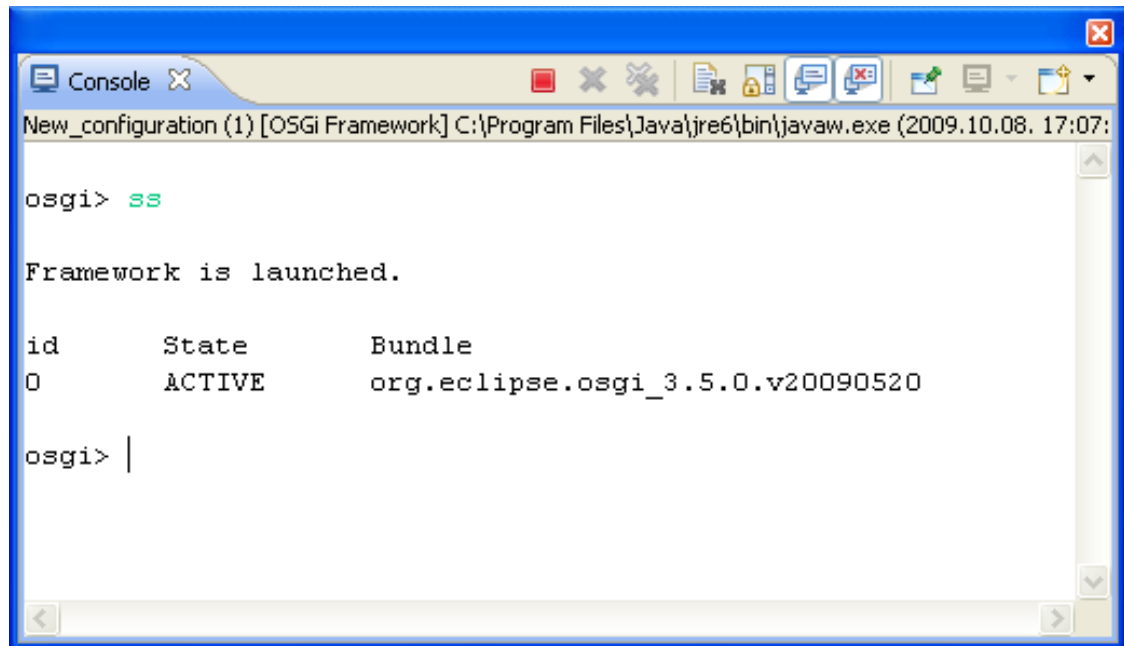
DS – Életciklus menedzselés



OSGi konzol

OSGi konzol

- OSGi prompt
 - Hasonló egy DOS v. Bash prompt-hoz
 - Általános parancskészlet
 - Bővítési lehetőségek
- Eclipse támogatás
 - Console view
 - Futó példányhoz is!
 - Highlighting



```
New_configuration (1) [OSGi Framework] C:\Program Files\Java\jre6\bin\javaw.exe (2009.10.08. 17:07:

osgi> ss

Framework is launched.

id      State      Bundle
0       ACTIVE    org.eclipse.osgi_3.5.0.v20090520

osgi> |
```

OSGi konzol - parancsok

■ Hasznos parancsok

- ss: kilistázza az telepített bundle-okat.
- start <id>: elindítja a megadott azonosítójú bundle-t
- stop <id>: leállítja a megadott azonosítójú bundle-t
- install file:<path>: telepíti a megadott „bundle”-t
- uninstall <id>: eltávolítja a megadott bundle-t
- update <id>: frissíti a megadott bundle-t
- services <filter>: kilistázza a futó szolgáltatásokat
pl.: `osgi> services (objectClass=*HelloService)`
- shutdown: a futó osgi framework leállítása
- close: shutdown és exit
- exit: ~ `System.exit`

Új parancs írása (GoGo Console)

- Parancsok megírása Java fájlban
 - Nincsen API, ami fölött fejleszteni kell!
 - `System.out.println()` is jól működik!
- Osztály kiajánlása szolgáltatásként
 - Szolgáltatás paraméterében felsorolni a parancsokat (metódusnevek)

Új parancs (Java osztály)

```
public class UserConsole {  
  
    public void createuser(  
  
        String username,  
  
        String password) {  
  
        User user = userManager.createUser(username, password);  
        if (user != null) {  
            System.out.println("A user '" + username + "' is created.");  
        } else {  
            System.err.println("ERROR: Command failed");  
        }  
    }  
    ...  
}
```

Új parancs (Java c

Parancs neve: createUser

```
public class UserConsole {
```

```
    public void createuser(
```

```
        String username,
```

```
        String password) {
```

```
    User user = userManager.createUser(username, password);
```

```
    if (user != null) {
```

```
        System.out.println("A user '" + username + "' is created.");
```

```
    } else {
```

```
        System.err.println("ERROR: Command failed");
```

```
    }
```

```
}
```

```
...
```

```
}
```

Új parancs (Java c

Parancs neve: createUser

```
public class UserConsole {
```

```
    public void createuser(
```

```
        String username,
```

```
        String password) {
```

```
    User user = userManager.createUser(username, password);
```

```
    if (user != null) {
```

```
        System.out.println("A user '" + username + "' is created.");
```

```
    } else {
```

```
        System.err.println("ERROR: Command failed");
```

```
    }
```

```
}
```

```
...
```

```
}
```

Két string paraméter

Új parancs (Java osztály)

```
public class UserConsole {
```

```
    public void createuser(
```

```
        String username,
```

```
        String password) {
```

```
        User user = userManager.createUser(username, password);
```

```
        if (user != null) {
```

```
            System.out.println("A user '" + username + "' is created.");
```

```
        } else {
```

```
            System.err.println("ERROR: Command failed");
```

```
        }
```

```
    }
```

```
    ...
```

```
}
```

Végrehajtott parancs –
Java/OSGi kód

Új parancs (Java osztály)

```
public class UserConsole {  
  
    public void createuser(  
  
        String username,  
  
        String password) {  
  
        User user = userManager.createUser(username, password);  
        if (user != null) {  
            System.out.println("A user '" + username + "' is created.");  
        } else {  
            System.err.println("ERROR: Command failed");  
        }  
    }  
    ...  
}
```

Visszajelzés standard
outputon

Új parancs (Java osztály)

```
public class UserConsole {
```

```
    public void createuser(
```

```
        String username,
```

```
        String password) {
```

```
        User user = userManager.createUser(username, password);
```

```
        if (user != null) {
```

```
            System.out.println("A user '" +
```

```
        } else {
```

```
            System.err.println("ERROR: Command failed");
```

```
        }
```

```
    }
```

```
    ...
```

```
}
```

Súgó szöveg megadható
@Descriptor annotációként

Új parancs (Java osztály)

```
public class UserConsole {
```

```
    @Descriptor(value = "Creates and saves a user.")
```

```
    public void createuser(
```

```
        @Descriptor(value = "name of the new user")
```

```
        String username,
```

```
        @Descriptor(value = "password of the new user")
```

```
        String password) {
```

```
    User user = userManager.createUser(username, password);
```

```
    if (user != null) {
```

```
        System.out.println("A user '" +
```

```
    } else {
```

```
        System.err.println("ERROR: Command failed");
```

```
    }
```

```
}
```

```
...
```

```
}
```

Súgó szöveg megadható
@Descriptor annotációként

Új parancs (Regisztráció)

```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/
v1.1.0" name="BookStore User Console">
  <implementation
class="hu.bme.mit.bookstore.console.UserConsole"/>
  <property name="osgi.command.scope" type="String"
value="bs"/>
  <property name="osgi.command.function" type="String">
listusers
createuser
deleteuser
</property>
<service>
  <provide
    interface="hu.bme.mit.bookstore.console.UserConsole"/>
</service>
</scr:component>
```

Új parancs (Regisztráció)

Implementációs osztály

```
<?xml version="1.0" encoding="UTF-8" ?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/v1.1.0" name="BookStore User Console">
  <implementation
class="hu.bme.mit.bookstore.console.UserConsole"/>
  <property name="osgi.command.scope" type="String"
value="bs"/>
  <property name="osgi.command.function" type="String">
listusers
createuser
deleteuser
</property>
<service>
  <provide
    interface="hu.bme.mit.bookstore.console.UserConsole"/>
</service>
</scr:component>
```

Új parancs (Regisztráció)

```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/
v1.1.0" name="BookStore User Console">
  <implementation
class="hu.bme.mit.bookstore.console"
  <property name="osgi.command.scope"
value="bs"/>
  <property name="osgi.command.function" type="String">
listusers
createuser
deleteuser
</property>
<service>
  <provide
    interface="hu.bme.mit.bookstore.console.UserConsole"/>
</service>
</scr:component>
```

Parancs prefix: bs

Új parancs (Regisztráció)

```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/
v1.1.0" name="BookStore User Console">
  <implementation
class="hu.bme.mit.bookstore.console.UserConsole"/>
  <property name="osgi.command.scope" type="String"
value="bs"/>
  <property name="osgi.command.function" type="String">
listusers
createuser
deleteuser
</property>
<service>
  <provide
    interface="hu.bme.mit.bookstore.console.UserConsole"/>
</service>
</scr:component>
```

Parancsok nevei

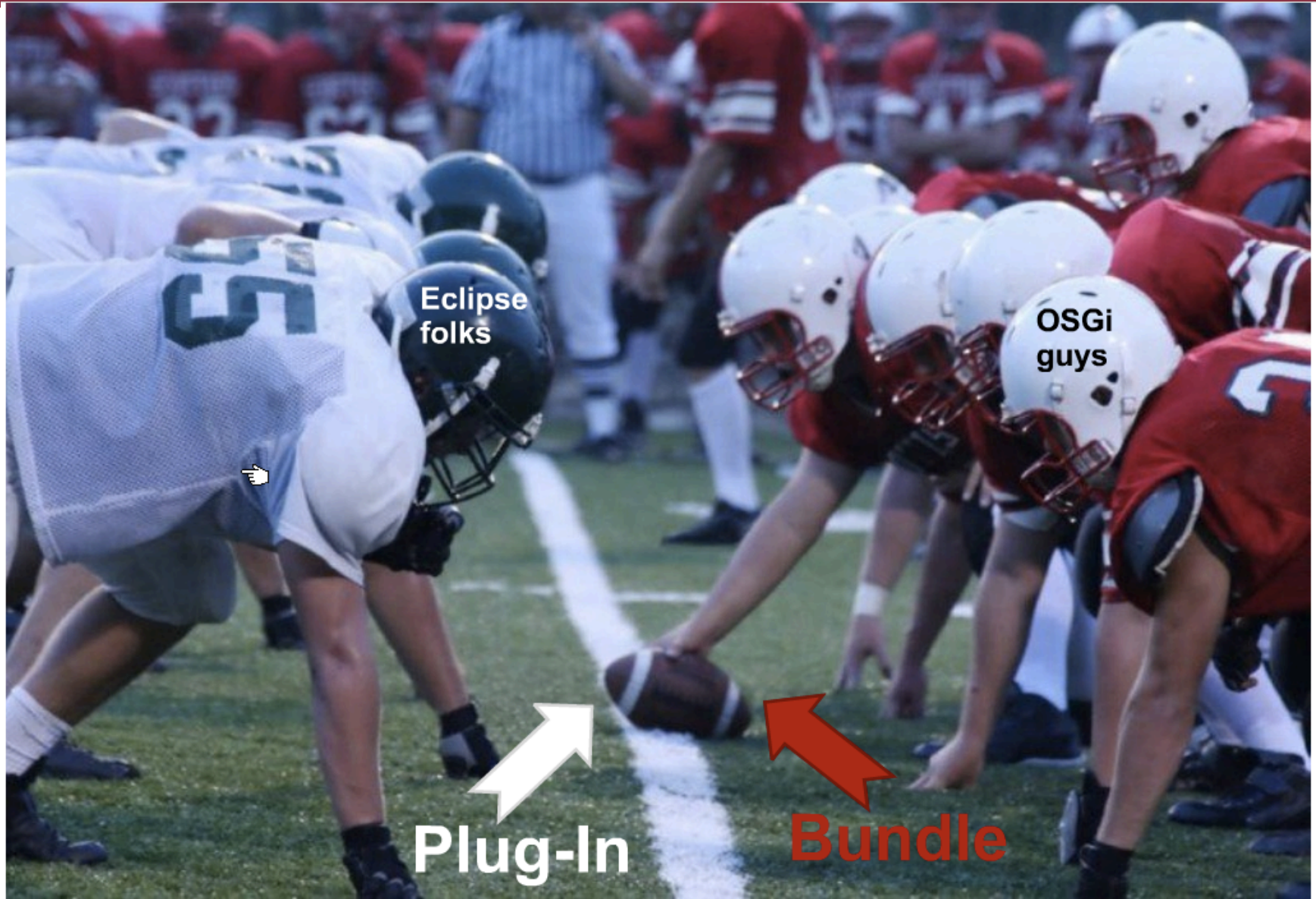
Új parancs (Regisztráció)

```
<?xml version="1.0" encoding="UTF-8"?>
<scr:component xmlns:scr="http://www.osgi.org/xmlns/scr/
v1.1.0" name="BookStore User Console">
  <implementation
class="hu.bme.mit.bookstore.console.UserConsole"/>
  <property name="osgi.command.scope" type="String"
value="bs"/>
  <property name="osgi.command.function" type="String">
listusers
createuser
deleteuser
</property>
<service>
  <provide
    interface="hu.bme.mit.bookstore.console.UserConsole"/>
</service>
</scr:component>
```

Szolgáltatás definíció, hogy
elinduljon a komponens

Eclipse vs OSGi

Bundle vs plug-in



Fontosabb fogalmak

■ OSGi

- Nyílt szabvány
- Komponens alapú leírások
- Egyre szélesebb alkalmazási kör (mobil, szerver, desktop, vállalati, beágyazott)

■ Eclipse

- RCP használata nagyon megnőtt
- Eclipse runtime lecserélése nyílt szabványra:
 - Eclipse 3.0 óta OSGi-ra épül

■ Equinox

- Eclipse OSGi implementációja (3.3 óta)
- OSGi 4.0 és 4.1 referencia implementáció

Eclipse Equinox

- OSGi referencia implementáció + Eclipse Extension!
 - Extension és extension point definíciók
 - Erős support SDK oldalon
 - Add-on:
 - Admin
 - Security
 - Application container
 - ...

Dinamikus bundle betöltés/eltávolítás

Plug-in Development – Eclipse – /Applications/Indigo.app/Contents/Resources/Java



Log Tasks Problems Console Properties Call Hierarchy History Coverage JUnit @ Javadoc

ce Log

ter text

Dinamikus bundle betöltés/eltávolítás

Plug-in Development – Eclipse – /Applications/Indigo.app/Contents/Resources/Java



Miért kell újraindítani
Eclipse frissítéskor?



Log Tasks Problems Console Properties Call Hierarchy History Coverage JUnit Javadoc

Log

ter text

Dinamikus bundle betöltés/módosítás

- Frissítés újraindítással
 - Történelmi okokból
- Egyszerűbb belső modell
 - Statikus szolgáltatáspéldányok
 - Függőségváltozás miatt újraindítani

Eclipse extensions vs. OSGi services

■ Eclipse kiterjesztés

- Egy kiterjesztési pont, amit bárki kiterjeszthet
- UI kiegészítések (túl kicsik OSGi szolgáltatáshoz)
- Nem kód jellegű kiegészítések
 - Témák megadása
 - Command framework jelentős része

■ OSGi szolgáltatás

- Szolgáltatásokat bárki definiálhat
- Szolgáltatásokat bárki használhat
- Nagyon dinamikus framework
- Laza csatolás

Eclipse extensions vs. OSGi services

	Extensions	Services	(Declarative Services)
Mit regisztrálunk	XML deklarációk	Java Objektumok	Java Objektumok (Proxy az első tényleges használatig)
Hogyan regisztrálunk	A plugin.xml összes extension -je <i>automatikusan</i>	A BundleContext API-ját használva <i>manuálisan</i>	A Service-Component leírókban lévő összes szerviz <i>automatikusan</i>
Hogyan használjuk fel	Lekérhetőek az Extension point ID-vel	Lekérhetőek az interface nevével és property filterekkel.	Ua., mint services esetében, de a szolgáltatást az SCR állítja be a megadott metódusokkal.
Milyen a kapcsolat számossága	<i>One-to-many</i> : egy Extension point -nak több Extension -je is lehet, de minden Extension -höz pont egy Extension point tartozhat	<i>Many-to-many</i> : egy szolgáltatást többen használhatnak és egy felhasználó több szolgáltatást használhat	Ua., mint services
Mikor töltődik be	Az Extension deklarációk indulásnál töltődnek be, de az ott szereplő osztályok <i>lazy-loading</i> -gal.	A megvalósító osztályt a betöltés előtt létre kell hozni.	A proxy elemnek köszönhetően a szolgáltatás csak akkor töltődik, amikor szükség van rá

Required-Bundle vs. Import-Package

■ Required-Bundle

- Eclipse felhasználók
- Minden package-t beimportál
 - amit a bundle kiajánl
 - re-exported package-ket is
- Erősebb csatolás

■ Import-Package

- OSGi felhasználók
- Csak a megadott package-ket importálja be
- Lazább csatolás
- Kevesebb látszik

Bnd és Bndtools

- OSGi köteg kezelő eszköz
 - Egyszerűsített leíró formátum
 - Szöveges konfigurációs állomány
 - Kötegeket automatikus generálja
- Szolgáltatások definíciója
 - Annotált Java osztályokkal
 - Deklaratív szolgáltatásokra épül

Bndtools

- Bnd frontend Eclipse környezetben
 - Automatikus fordítás
 - Form-alapú szerkesztő
 - Eclipse integráció
- DE: logikai eltérések PDE alapú fejlesztéstől
 - Egy projekt -> 1..n köteg
 - Képes szétosztani

Összefoglalás

OSGi alkalmazások

- Dinamikus modelarchitektúra
 - Moduláris fejlesztés
 - Szolgáltatásalapú
 - Modulok menet közben változhatnak
- Széles körű támogatás
 - Beágyazott rendszerek
 - Alkalmazások
 - Alkalmazásszerverek

További anyagok

- Neil Bartlett: OSGi and How it Got That Way
 - <http://njbartlett.name/2010/03/07/osgi-and-how-it-got-that-way.html>
 - Erősen ajánlott egyszer végignézni
- OSGi R5 újdonságok
 - <http://www.eclipsecon.org/2012/sessions/whats-new-osgi-enterprise-release-50>