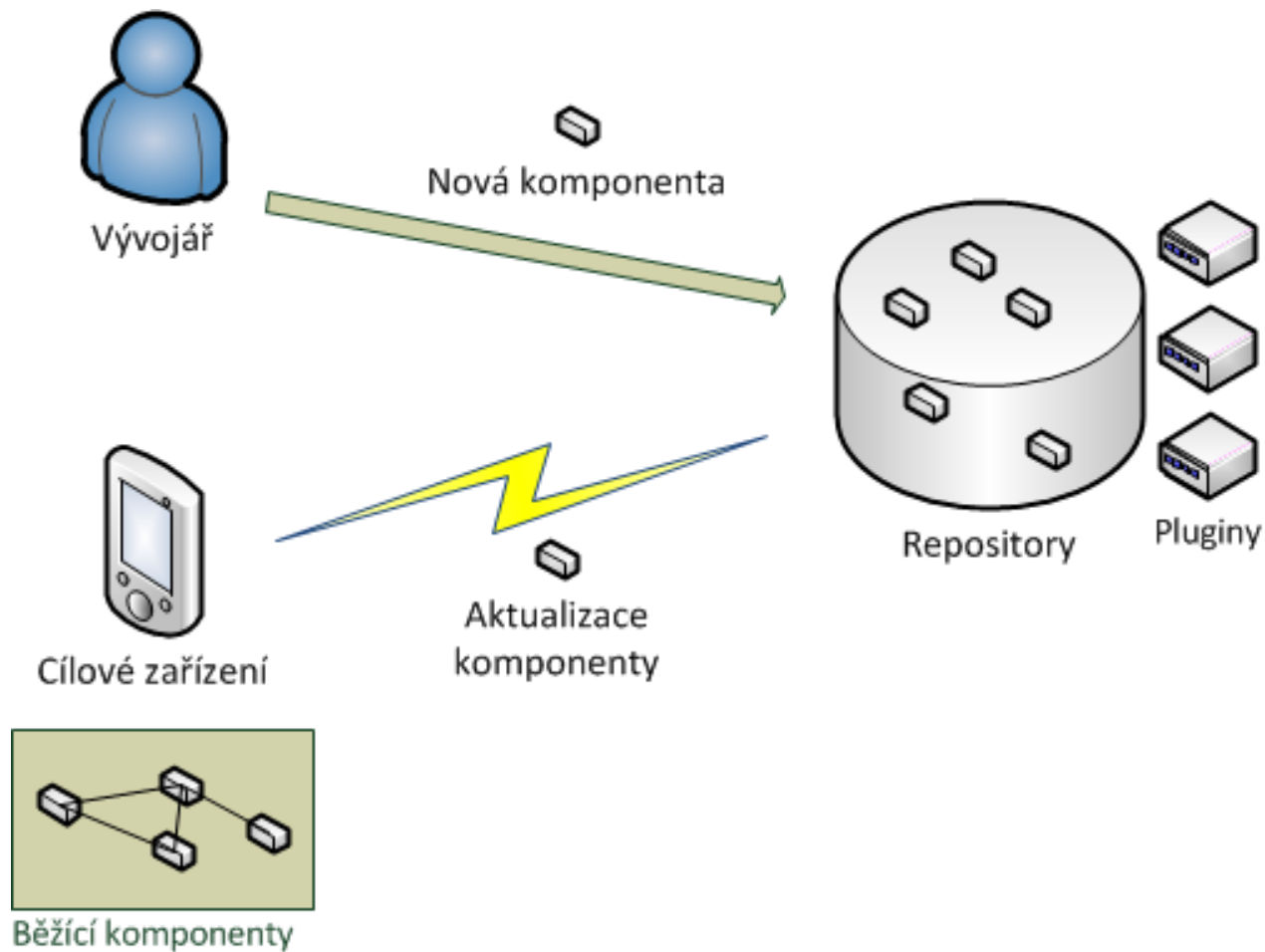


Úložiště komponent s podporou kontroly kompatibility

Jiří Kučera
kalwi@students.zcu.cz

Úložiště komponent



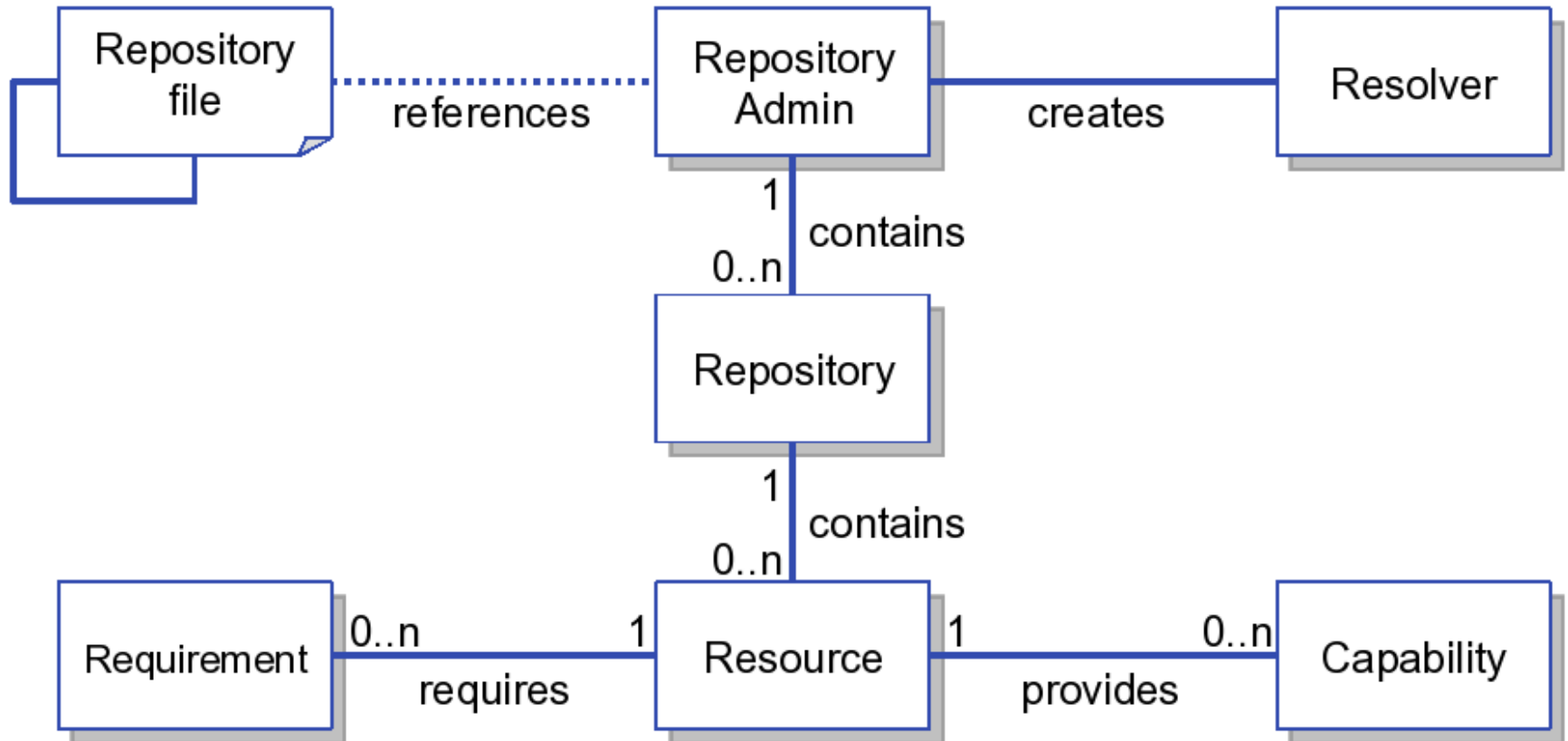
Požadavky na úložiště

- ▶ Vývojář komponent
 - Uložit komponentu
 - Spustit testy nad komponentou
- ▶ Rozšíření úložiště (pluginy)
 - Běh testů nad komponentou
 - Generování metadat
- ▶ Uživatel komponent, cílové zařízení
 - Získat seznam dostupných komponent
 - Zjistit nejnovější verzi komponenty
 - Získat popisná metadata komponenty
 - Porovnat kompatibilitu
 - Stáhnout komponentu

OSGi Bundle Repository (OBR)

- ▶ RFC-0112 Bundle Repository (draft)
- ▶ Snaha vytvořit centrální repository
 - OBR vs. Maven repository
- ▶ Definuje
 - XML formát pro popis repository a jeho obsahu
 - Službu pro přístup k repository
- ▶ Umožňuje nebo usnadňuje
 - Popis vlastností bundlů
 - Vyhledávání dostupných bundlů
 - Popis a řešení závislostí mezi bundly
 - Deployment bundlů do kontejneru

Entity OBR repository



Repository

- ▶ Artefakty uloženy v adresářové struktuře
- ▶ Obsah repository popisuje XML soubor
 - repository.xml
 - Uložen v kořenovém adresáři
- ▶ Repository lze seskupovat do stromové struktury

```
<repository name='Vzorové repository' time='20110131133541.062'>  
  <resource ... />  
  <resource ... />  
  ...  
  <referral depth='1' url='http://vnořené.repository/repository.xml' />  
</repository>
```

Resource

- ▶ Popisuje artefakt, jeho vlastnosti a závislosti
- ▶ ID artefaktu – symbolicname + version
- ▶ URI – cesta k artefaktu
 - Absolutní – teoreticky i mimo strukturu repository
 - Relativní – vůči adresáři s repository.xml
- ▶ Další proměnné (properties)
 - presentationname, description, size, license...
 - Uživatelsky definovatelné
- ▶ Kategorie
 - Usnadňují vyhledávání artefaktů

Resource (XML)

```
<resource id='Bundle1/1.0.0' symbolicname='Bundle1' version='1.0.0'  
  uri='cz/zcu/kiv/Bundle1.jar' presentationname='Bundle č. 1'>  
  <description>Ukázkový bundle č. 1</description>  
  <size>6477</size>  
  <category id='osgi' />  
  <category id='spring-dm' />  
  ...  
  <capability ... />  
  <capability ... />  
  ...  
  <require ... />  
  ...  
</resource>
```

Capability

- ▶ Cokoliv, co lze popsat množinou proměnných
 - package, service, execution environment apod.
- ▶ Má název
 - Není to ale ID
 - Na *capability* se mapují *requirements* se stejným názvem
- ▶ Proměnné (properties)
 - Název
 - Hodnota
 - Typ
 - string (výchozí), version, uri, long, double, set

Capability (XML)

```
<capability name='package'>  
  <p n='package' v='cz.zcu.kiv.jacc' />  
  <p n='version' t='version' v='1.0.0' />  
  <p n='uses:' v='org.osgi.framework' />  
</capability>
```

```
<capability name='example'>  
  <p n='some-string' v='this is a string' />  
  <p n='some-version' t='version' v='1.2.3' />  
  <p n='some-uri' t='uri' v='this:/is.an.uri/' />  
  <p n='some-set' t='set' v='values,separated,by,commas' />  
  <p n='some-long' t='long' v='12345' />  
  <p n='some-double' t='double' v='123.45' />  
</capability>
```

Requirement

- ▶ Požadavek na *capability* se shodným názvem
- ▶ Vyjádřen jako OSGi (LDAP) filtr – RFC 1960
 - Např. (&(package=cz.zcu.kiv.jacc)(version>=1.0.0))
- ▶ Proměnné *capability* musí odpovídat filtru
- ▶ Další možnosti požadavku
 - optional – požadavek je pouze volitelný
 - multiple – může být použito více řešení požadavku
 - extend – obrací model, volí artefakty, pro které může být naopak užitečný, např. jako plugin
 - Popis vysvětlující důvod požadavku

Requirement (XML)

```
<require name='package' optional='false' multiple='false' extend='false'  
  filter='(&(package=cz.zcu.kiv.jacc)(version>=1.0.0))'>  
  Import package cz.zcu.kiv.kacc; version is equal or greater than 1.0.0  
</require>
```

Pozn.: Znaky &, >, <, ' a " je třeba nahradit XML entitami.

```
<require name='example' optional='true' multiple='true' extend='true'  
  filter='(!(some-long=12345))'>  
  Optional extension for all example capabilities whose some-long  
  value is not equal to 12345  
</require>
```

Příklad repository

```
<repository name='Příklad' lastmodified='20110131135021.012'>
```

```
  <resource id='Bundle1/1.0.0' symbolicname='Bundle1' version='1.0.0' uri='Bundle1.jar'>
```

```
    <capability name='vlastnost'>
```

```
      <p n='promenna' t='double' v='123.45'>
```

```
      <p n='verze' t='version' v='1.2.3'>
```

```
    </capability>
```

```
  </resource>
```

```
  <resource id='Bundle2/1.2.0' symbolicname='Bundle2' version='1.2.0' uri='Bundle2.jar'>
```

```
    <require filter='(&#x26;#x26;(promenna&#x26;#x26;=100.00)(verze=1.2.3))'>
```

```
      name='vlastnost' extend='false' optional='false' multiple='false'>
```

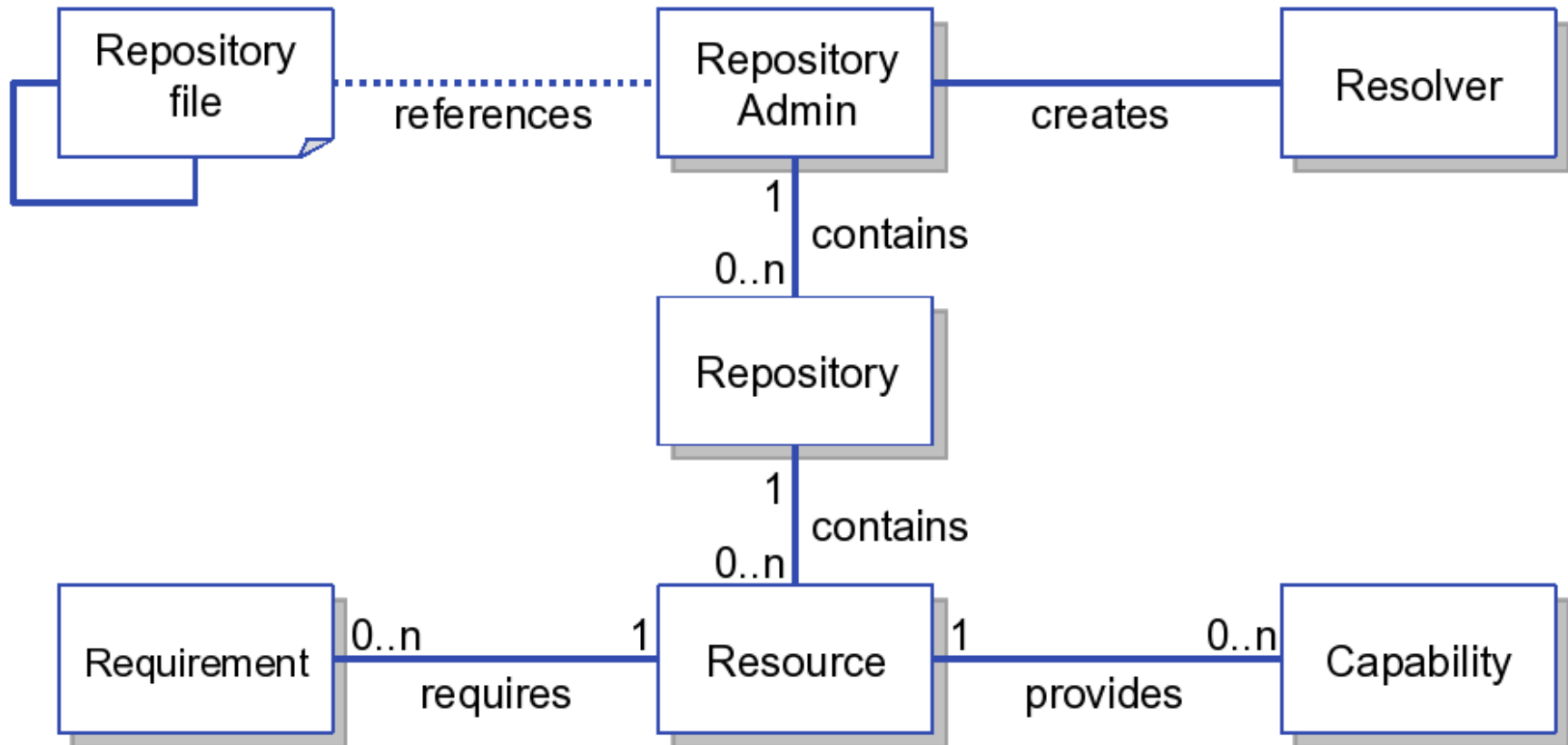
```
    </require>
```

```
  </resource>
```

```
</repository>
```



Entity ještě jednou



Resolver

- ▶ Analyzuje závislosti předem určených artefaktů
 - Zjistí *requirements*, které vůbec žádný artefakt nesplňuje
 - Zjistí *requirements*, kvůli kterým je vyžadován nějaký artefakt z repository
 - Zjistí artefakt, který má nespílitelné *requirements*
 - Zjistí artefakty, které se nacházejí v repository a jsou potřeba k vyřešení závislostí
- ▶ Výsledkem je určení, zda jsou požadavky artefaktů splněny
- ▶ Provede *deploy* zvolených artefaktů do kontejneru
 - Pokud jsou vyřešeny všechny závislosti

Resolver – příklad

```
RepositoryAdmin repoAdmin = ... // Get repo admin service
```

```
String filter="(&(symbolicname=Bundle1)(version>2.0.0))";
```

```
Resource resource = repoAdmin.discoverResources(filter);
```

```
Resolver resolver = repoAdmin.resolver();
```

```
resolver.add(resource);
```

```
if (resolver.resolve()) {  
    resolver.deploy(true);
```

```
} else {
```

```
    Requirement[] reqs = resolver.getUnsatisfiedRequirements();
```

```
    for (Requirement req : reqs) {
```

```
        System.out.println("Unable to resolve: " + req);
```

```
    }
```

```
}
```

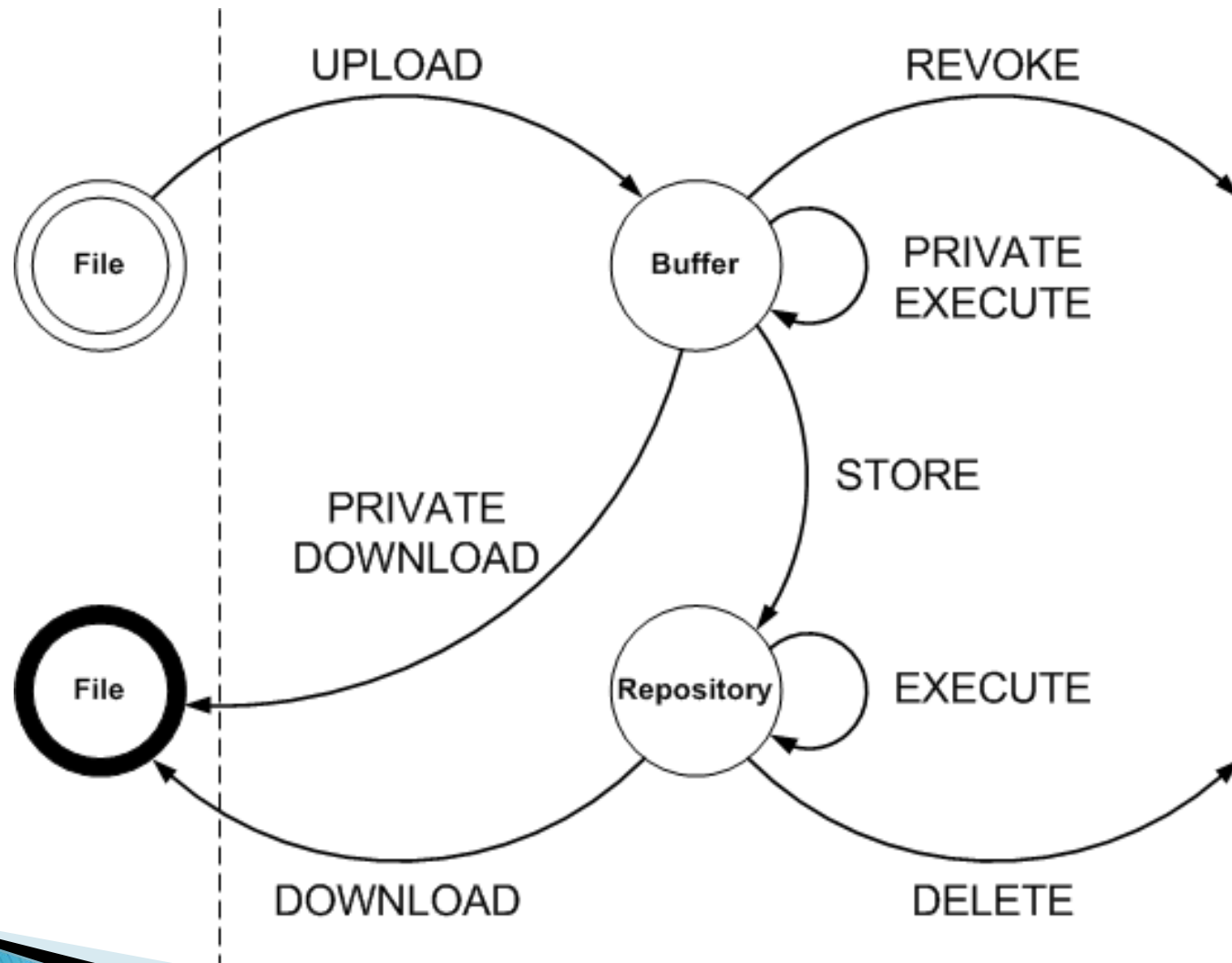
Úložiště komponent

- ▶ Integrace nástrojů pro kontrolu kompatibility
 - Verzování bundlů
 - Mimofunkční charakteristiky
 - Simulace běhu
- ▶ Možná integrace s Apache ACE
 - Framework pro centrální správu komponent, konfigurace a jiných artefaktů
 - Distribuce artefaktů na cílová zařízení
 - Definiuje množinu artefaktů (distribuce), cílová zařízení sama kontrolují nové verze ke stažení

Úložiště komponent

- ▶ Modulární architektura
 - OSGi
 - Apache Felix Dependency Manager
- ▶ Využívá OBR
 - Re prezentace vložených artefaktů jako OBR Resource
 - Neomezuje použití OBR pouze na OSGi bundly
 - Do OBR ukládá i výsledky běhu testů
- ▶ Rozšiřitelné prostřednictvím pluginů
 - Modifikovaný Chain of Responsibility
 - Výběr pluginů dle priority a informací v OBR

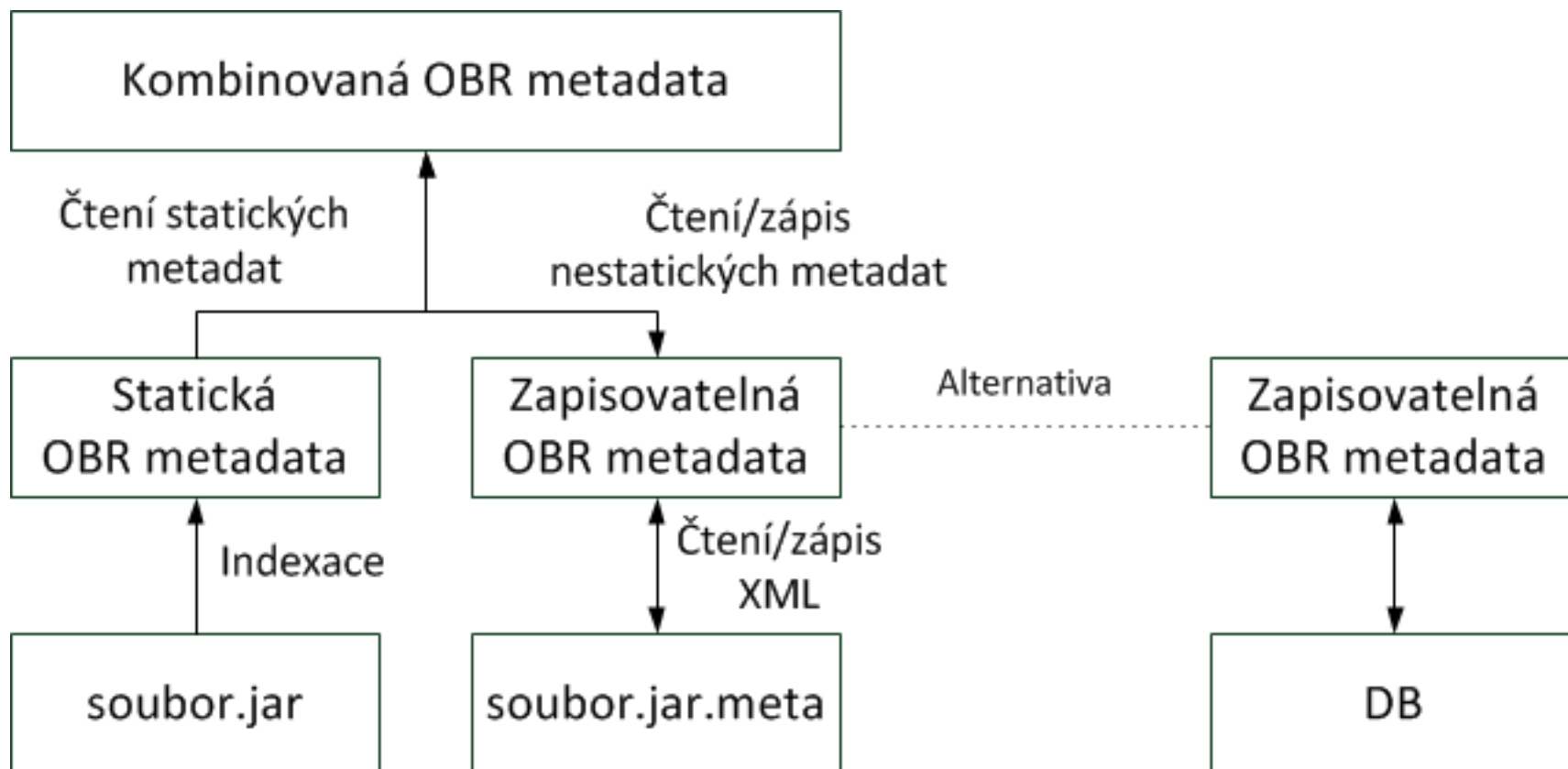
Lifecycle artefaktu v repository



Plugins – listeners

- ▶ Volání se váže na události lifecycle
- ▶ Paralelní běh pluginů
 - Výhradní běh pluginů modifikujících artefakt
- ▶ Vzájemné závislosti pluginů → pořadí volání
 - Nejsou přímo vyjádřené
 - Spuštění možné po splnění určitých *requirements*
 - Volaný plugin modifikuje OBR metadata, umožní tím spuštění jiného pluginu
 - V případně nerozhodnosti rozhodne priorita

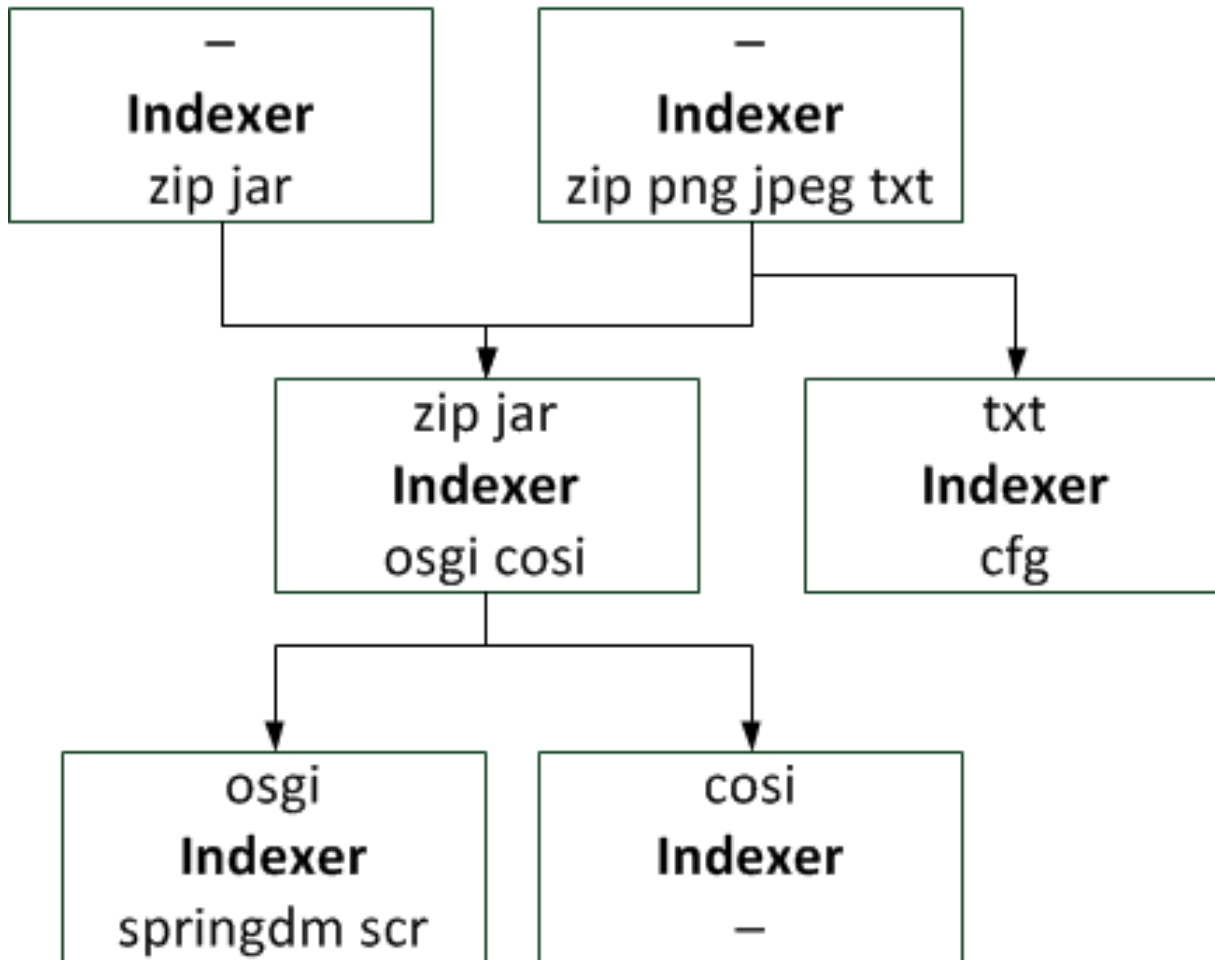
Pluginy – DAO pro metadata



Indexace statických metadat

- ▶ Statická metadata – neměnitelná
 - Manifest, Declarative Services, Spring DM...
- ▶ Rychlé zjištění statických metadat artefaktu
 - Požadavek na rychlou odezvu UI
- ▶ Pro urychlení možnost hierarchizace indexerů
 - Využití kategorií v OBR Resource
 - Indexer definuje kategorie, které umí nastavit
 - ...a kategorie podmiňující pro daný indexer
 - Stromová struktura (nebo podobná)
- ▶ Další možné indexery
 - CoSi komponenty, konfigurační soubory, apod.

Hierarchizace indexerů



Diskuse

- ▶ Vhodnost OBR formátu
- ▶ Požadavky
 - na lifecycle
 - na vnitřní API
 - na vnější API
 - na rozhraní
- ▶ Možná rozšíření (semestrální práce)
 - Indexace OSGi Declarative Services a Spring DM
 - Indexace CoSi komponent
 - Ukládání OBR metadat do DB
 - Webové rozhraní

Zdroje

- ▶ OSGi RFC-0112 Bundle Repository
 - http://www2.osgi.org/download/rfc-0112_BundleRepository.pdf
- ▶ Apache Felix OSGi Bundle Repository (OBR)
 - <http://felix.apache.org/site/apache-felix-osgi-bundle-repository.html>
- ▶ OSGi BIndex
 - <http://www.osgi.org/Repository/BIndex>
- ▶ RFC 1960 LDAP Search Filters
 - <http://www.ietf.org/rfc/rfc1960.txt>
- ▶ Apache ACE
 - <http://incubator.apache.org/ace/>
- ▶ SVN
 - <https://kalwi.eu/svn/crce>