

# Class Loading in JBoss EAP 6+

...

Java Kafi, Puzzle ITC, 26. Mai 2016

Benjamin Bühlmann

# Übersicht

Class Loading?

Statische & dynamische Module

Module Dependencies

Konfiguration

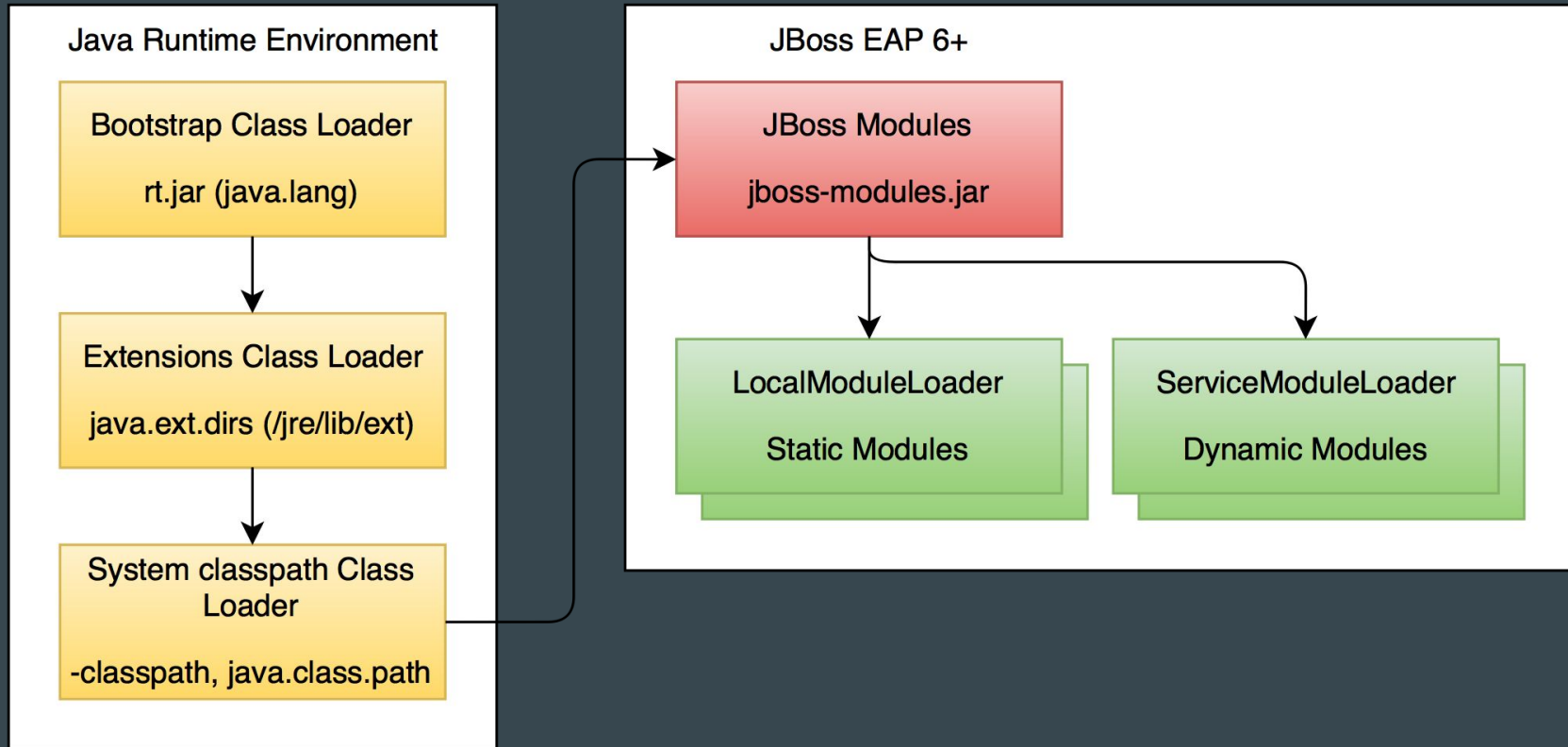
Demos

# Class Loading?

Ermöglicht Zugriff während Runtime auf Klassen (Bytecode) und Ressourcen

Dynamisches Nachladen von Bytecode in die VM

# Class Loading?



# Class Loading in JBoss Application Server 6+

Module-based Class Loading → Alles ist ein Modul

Implementation: JBoss Modules

1 Class Loader pro Modul

Isolation zwischen Modulen

Die Abhängigkeiten von Modulen kann konfiguriert werden

# Class Loading in JBoss Application Server 6+

Nur die benötigten Module sind im Classpath

Highly concurrent & Thread safe

Flexibilität durch Modularität

Gleiches Prinzip in EAP 7

# Static Modules

Mitgelieferte Module:

- **EAP\_HOME/modules/system**
- Enthält den kompletten AS (exkl. JBoss Modules)
- Supportlevel der Module
- Versionen der Module

Benutzerdefinierte Module:

- **EAP\_HOME/modules**
- Wiederverwendbarkeit  
→ *Beispiel PostgreSQL*

# Dynamic Modules

## WAR Deployment

- Ein Classloader für `WEB-INF/classes/*` und `WEB-INF/lib/*.jar`

## EAR Deployment

- Ein Classloader für das Parent-Module: `/lib`
- Ein Classloader pro Subdeployment
  - `application.ear:/*.jar`
  - `application.ear:/*.war`

→ *Beispiele in JConsole*



# Implicit Dependencies (System Dependencies)

Referenzieren ausschliesslich statische Module

Werden durch den Container hinzugefügt

Liste der impliziten Dependencies

Können in `jboss-deployment-structure.xml` ausgeschlossen werden

# Explicit Dependencies (User Dependencies)

Können zu statischen oder zu dynamischen Modulen definiert werden

Zwei Möglichkeiten zum Konfigurieren der expliziten Dependencies:

- `[WEB-INF | META-INF] / jboss-deployment-structure.xml`
- `META-INF / MANIFEST.MF`

# jboss-deployment-structure.xml

Export von transitiven Dependencies

Exclude von impliziten Dependencies

Exclude eines ganzen Subsystems

Subdeployment Isolation

# MANIFEST.MF

Manifest-Version: 1.0

Built-By: bbuehlmann

Build-Jdk: 1.8.0\_72-internal

Created-By: Maven Integration for Eclipse

Main-Class: org.jboss.as.quickstarts.appclient.acc.client.Main

Dependencies: org.apache.log4j, com.sun.jsf-impl:mojarra-2.1.19 export

# Class Loading Precedence

1. Implicit Dependencies (System Dependencies)
2. Explicit Dependencies (User Dependencies)
3. Local resources
4. Inter-deployment dependencies

# Demos

Austausch der JSF Implementation in JBoss EAP 6.4.0

Analyse des Class Loaders und der Konfiguration von Dependencies mittels Helper-JSP