



Konferenz für Java™, Enterprise Architekturen, SOA

C. Schädel, C. Schmidt-Casdorff | iks GmbH

Provisioning unter OSGi für Test und Betrieb

Einleitung

- Überblicksvortrag
- Beispiele „idealisiert“ – Mischformen existieren
- Problemstellung des Provisioning unter OSGi
- Lösungsansätze
 - exemplarisch anhand von Tools

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

- Deploybare Anwendung
 - Kann isoliert installiert / deinstalliert werden
 - Besteht aus einer/wenigen Datei(en)
 - Ist aus anderen Container-Konzepten bekannt
- OSGi kennt dieses Konzept nicht
 - Anwendung ist eine Menge an *Bundles*
 - *Bundles* interagieren untereinander
 - Anwendung hat keine Exklusivität an *Bundles*

Quintessenz

- Es wird eine Menge an *Bundles* installiert
 - Wie wird diese Menge zusammengestellt?
 - Wie wird diese Menge bereitgestellt?
 - Wie lässt sich die Reaktion auf die Installation abschätzen ?
 - Können die *Bundles* gestartet werden?
 - Wie werden *ImportPackages* aufgelöst?
- OSGi bietet keine *out-of-the-box* Lösung

⇒ Provisioning unter OSGi

- Bootstrapping
 - Zu installierende Bundles werden beim Startup des OSGi-Frameworks installiert und gestartet
 - Start-Level regeln die Reihenfolge
 - Neustart des OSGi-Frameworks notwendig
- Laufzeit - Provisioning
 - Bundles werden zur Laufzeit des OSGi-Frameworks installiert und gestartet
 - Kein Neustart des OSGi-Frameworks notwendig

Anforderungen

- Quality of Service
 - Automatisierbar
 - Zuverlässig
 - Deterministisch / wiederholbar
- . . . für Bootstrapping- und Laufzeitprovisioning

⇒ testbar

Bemerkungen

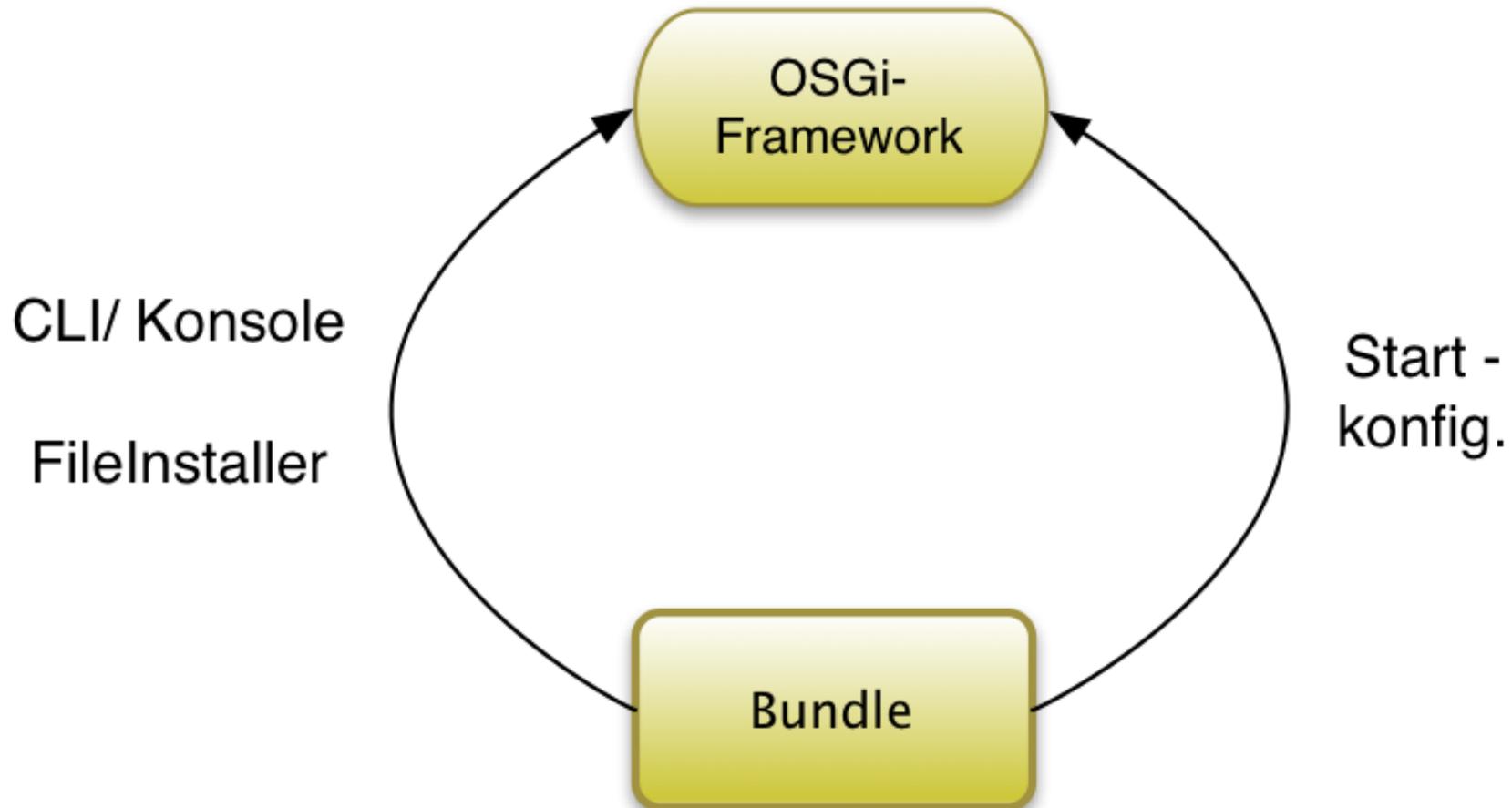
- Alle Provisioning-Verfahren
 - Definieren eine Menge an Bundles
 - Niemals die Abhängigkeiten der Bundles untereinander
 - Diese löst das OSGi-Framework auf
 - Daher Unterschiede bei z.B. Kompilieren
 - Bereitstellung : Bundle
 - Installation / Resolving : *Bundle / Package*
- Paradigmenwechsel
 - Wie unterstützt Provisioningprozess diesen?

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Repository Admin und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

- Installation beim Start des OSGi-Frameworks
 - Kurz: Bootstrapping
 - Startup-Konfiguration
 - Unterstützung durch *Start Level Service*
 - *auto-deploy* von Apache Felix
- Installation zur Laufzeit
 - CLI / Konsole
 - *FileInstaller Service* von Apache Felix

- Provisioning mit OSGi Bordmitteln
 - Händische Installation
 - Lage des Bundles im Filesystem muss bekannt sein
 - Anwendbar nur bei kleiner Menge an *Bundles*



Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

PAX Runner – Konzepte (1)

- *Bootstrapping*-Prozess für OSGi
- Konfiguration des zu startenden *Frameworks*
 - Unterstützt alle wichtigen Frameworks
- Zusammenstellung der zu installierenden *Bundles*
 - Unterstützt Start-Levels

PAX Runner – Konzepte (2)

- Bundles werden aus unterschiedlichen Quellen bereitgestellt
 - Filesystem, ZIP, *Maven Repository*, ...
 - Bundles werden in diese Quellen deployed
- Bundles können in *profiles* gruppiert werden
 - (transitive) Abhängigkeiten von *profiles* möglich
 - PAX Runner scannt die *profiles* und installiert *Bundles*
 - OSGi-Frameworks werden in *profiles* beschrieben

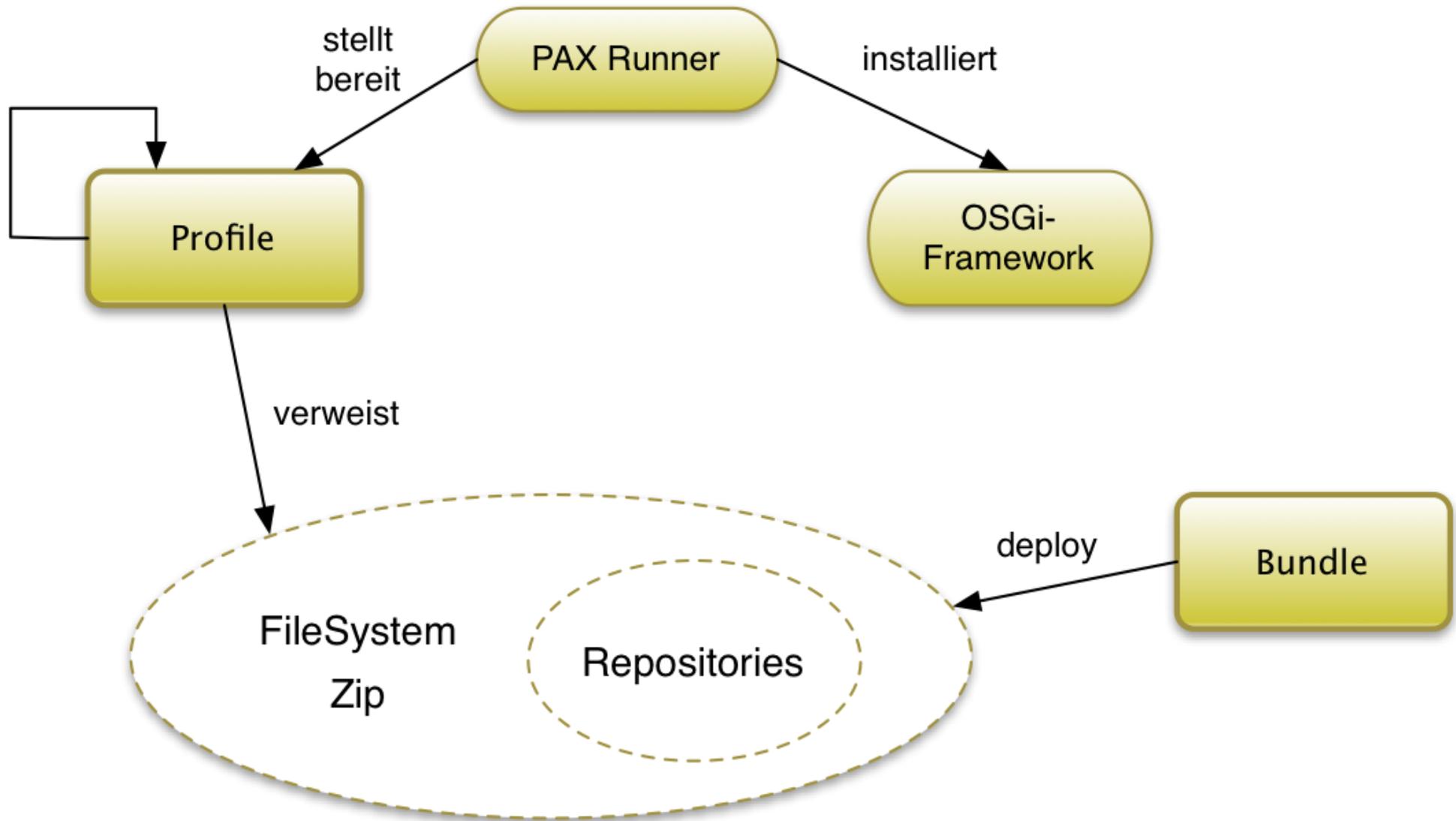
Installation via PAX Runner (3)

- PAX Runner liefert
 - einen deterministischen Provisioningprozess
 - Installation allerdings nur zum *Bootstrapping*
 - Aufsetzen eines initialen Zustands
 - Trennt Bereitstellung und Installation
 - Lässt eine spezifische Struktur der *Bundle*-Quellen zu

⇒ Das ist ein wichtiger Schritt

Installation via PAX Runner

(4)



Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

OSGi Bundle Repository (1)

- Kann ein Bundle installiert werden ?
 - Sind *require-bundle*, *ImportPackage* aufzulösen?
 - Sind die fehlenden Abh. dynamisch nachzuinstallieren?
 - Konzept zur Beschreibung von Anforderungen
 - Wenn ja, wo finde ich die Bundles?

OSGi Bundle Repository (2)

- OBR ist Spezifikation der OSGi Alliance
- Definiert Struktur von *Bundle-Repositories*
 - Meta-Informationen von *Bundles*
 - Abfragen gegen Repositories
- Definiert Zugriffsschnittstellen auf Repositories
 - Via *Repository Admin Service*
 - Abbildung auf unterschiedliche, physische Repositories
- Unterstützt einen proaktiven *resolving process*
 - Für Konflikte können Verfahren implementiert werden

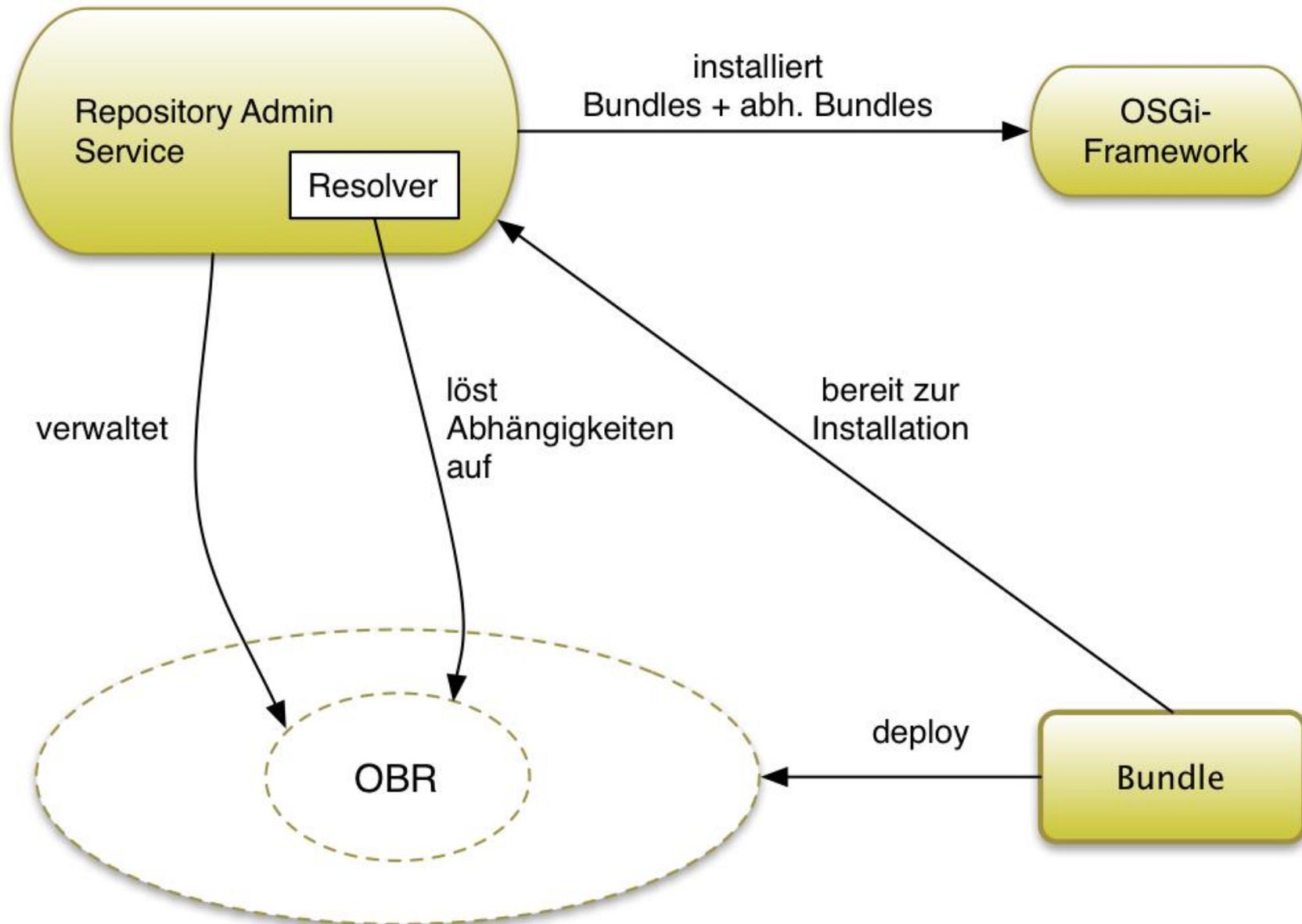
OSGi Bundle Repository (3)

- Herkömmliche Repositories (z.B. maven)
 - Beinhalten Artefakte
 - Abhängigkeiten zwischen den Artefakten
- OBR
 - Verwaltet Ressourcen
 - Trennt Metadaten und Ressourcen
 - Abhängigkeiten werden über *capabilities* abgebildet
 - Allgemeineres Konzept
 - Beinhaltet ImportPackages,
 - Abfragen über *capabilities*

- *Apache Karaf* bietet Schnittstelle
 - Um Bundles via OBR zu installieren
 - Inkl. erforderlicher Bundles
- Schnittstelle ist Bestandteil der *karaf OBR Shell*

OSGi Bundle Repository

(5)

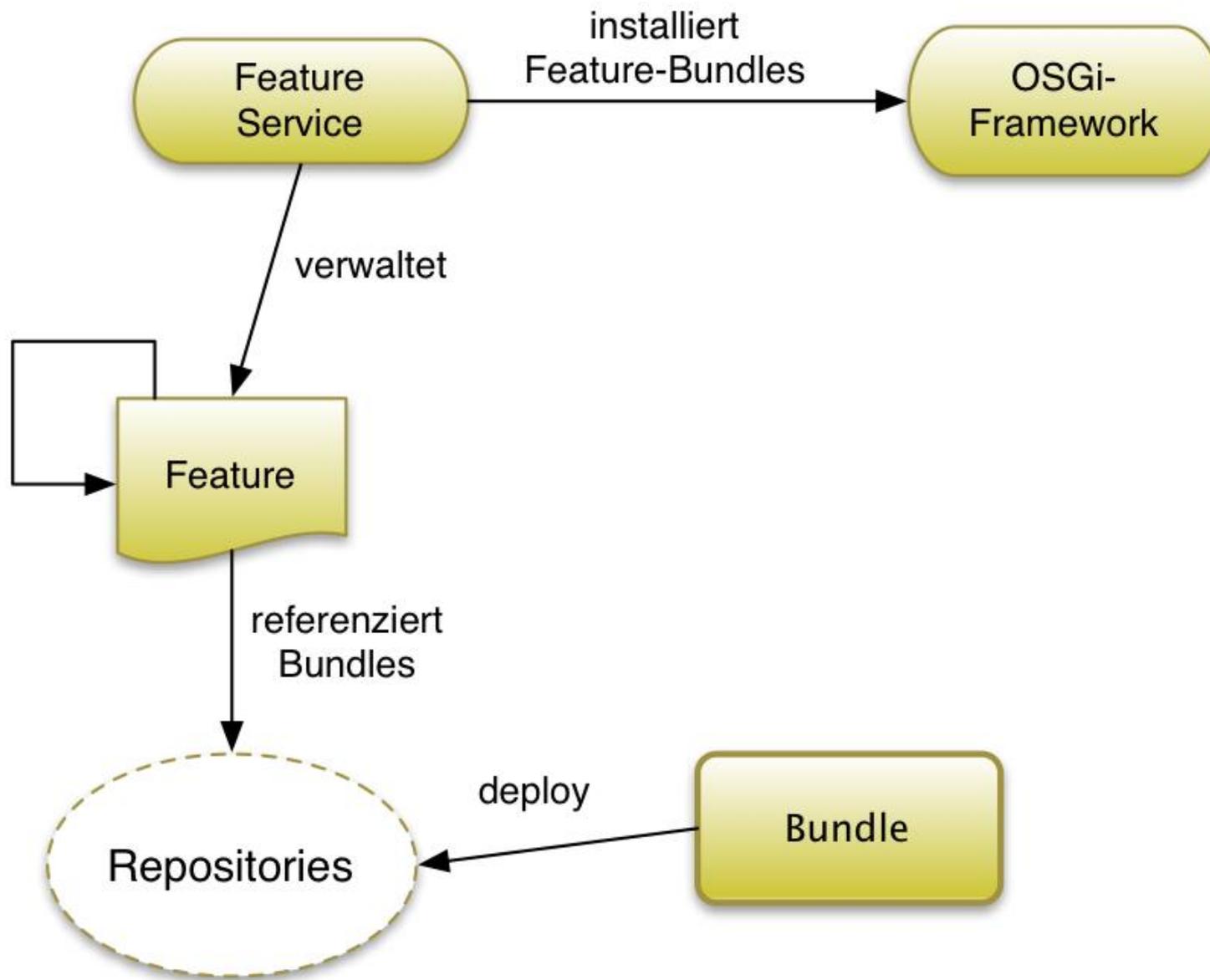


- Features fassen eine Menge an Bundles zusammen und unterstützen
 - abhängige *Features*
 - Konfiguration
 - *feature repositories*
- Sind ein Konzept von Apache
 - Integriert in *apache karaf*
- Werden durch den *Feature Service* verwaltet
 - Kein OSGi-Standard

- *Features* werden aus unterschiedlichen Quellen bereitgestellt
 - Filesystem, Http, *maven repository*, ...
- *Feature Service* hält nur *Features* konsistent
 - Verwaltet die Abhängigkeiten zwischen *Features*
 - Inkl. der Abhängigkeiten aller *feature bundles*
- Nutzt nicht OBR
- Bereitstellung muss für Konsistenz sorgen

Features

(3)



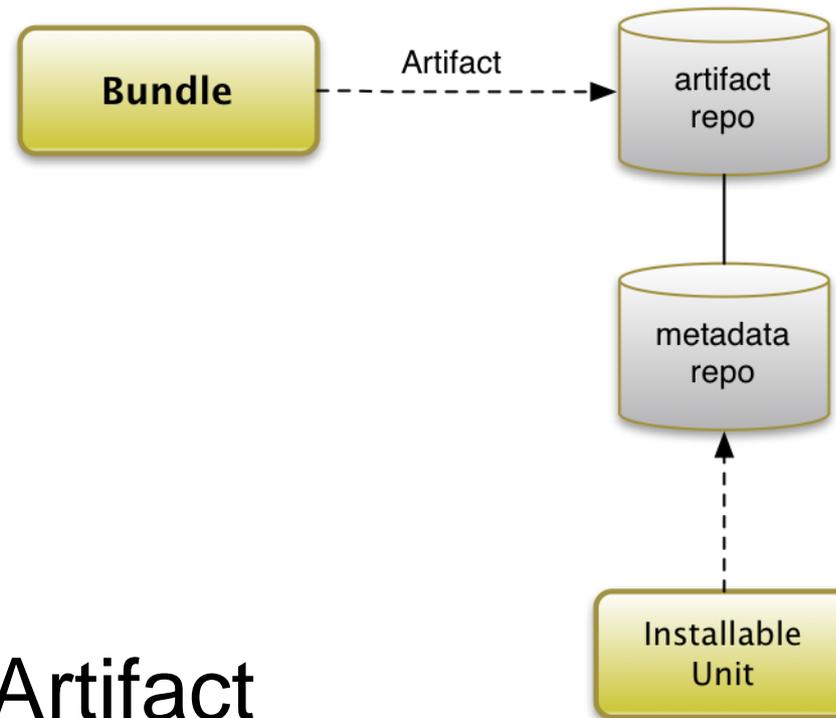
Zusammenfassung

- OSGi Bundle Repository
 - Unterstützt keine Bereitstellung
 - Ist Basistechnologie für einen Provisioningprozess
- Feature Service
 - Ermöglicht ein Provisioning zur Laufzeit
 - Verbindet Bereitstellung und Installation
 - Bereitstellung hat für die Konsistenz zu sorgen

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

- Equinox: OSGi-Framework
- P2: general provisioning system
 - Ersetzt den Update-Manager
 - Starke Unterstützung für RCP-Applikationen
 - Nicht nur für OSGi-Bundles



- Bundle = Artifact
- Installable Units (IU) sind Metadaten
- Artifacts liegen in ‚artifact repositories‘
- IUs liegen in ‚metadata repositories‘

- Konzepte:
 - „if it installs, it runs“
 - Bootstrapping Installation (Yoxos)
 - Installation zur Laufzeit
 - Bundle-Pooling
 - Paralleler Download von Bundles
 - Touchpoints

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

- Bisherige Provisioningprozesse
 - Keine Konfiguration von Installationen
 - Arbeiten gegen genau eine Instanz eines OSGi-Framework
 - Keine historisierten Installationen
 - Kein *Rollback* möglich
- ⇒ System zur Verteilung von Komponenten

- *software distribution framework*
 - Zentrale Konfiguration von Installationen
 - Versionierung
 - Zentrale Installation auf Ziel-OSGi-Frameworks
 - Historisierung
 - Rückkopplung der Veränderungen auf den Ziel-Frameworks
- <http://incubator.apache.org/ace/>

Ausblick

- Bisherige Provisioningprozesse ...
 - basieren alle auf Bundles
 - bieten keine weiteren Deployment- und Installationseinheiten
- Gibt es die deploybare OSGi-Anwendung?

⇒ **Plattform ARchive**

Spring Source DM Server

- Eclipse Equinox OSGi-Framework
- Spring DM (Dynamic Modules) Kernel
- provisioning service
 - Fehlende Abhängigkeiten werden automatisch 'on demand' installiert
 - dynamisches Deployment
- SpringSource Enterprise Bundle Repository

Spring Source DM Server - PAR

- Spring Source DM Server spezifisch
- OSGi-Applikation
- Kann Bundles oder Services außerhalb des PARs verwenden
- Ist in sich abgeschlossen
 - Keine Bundles oder Services des PAR können von ‚außen‘ genutzt werden
- Deployment erfolgt über Ablage in „pickup“ - Verzeichnis

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- **Repository Strategien**
- Provisioning für OSGi Integrationstests

- Ein Repository ist eine Quelle zur Bereitstellung
 - mvn, P2, OBR, aber auch ZIP, file
- Ein Provisioningprozess umfasst immer
 - Bereitstellung (der Bundles) -> Repository
 - Installation
- Ausliefern einer Anwendung bedeutet
 - Auslieferung eines Provisioningprozesses

⇒ Wie ist mit Repositories umzugehen?

- Integrationstests über Auslieferungsrepositories
 - Konsistenz des Repositories beeinflusst Provisioning
- Repository per Applikation vs. zentrales Repository
 - Dezierte Auflösung vs. universelle Auflösung
- Repository per Staging vs. zentrales Repository
 - Stabile Produktionsumgebung vs. produktionsgleiche Integrationstest

- Einsatz von Repository Manager
 - Siehe Artifactory, Nexus (Professional)
 - Unterstützen nicht alle Repository Strukturen
 - Verwaltung von Repositories
 - Transparenter Schwenk von Repositories
- OSGi-fizierte vs. herkömmliche Repositories
 - Hängt von Toolauswahl des Provisionings ab

Agenda

- Einführung
- Provisioning-Konzepte und Tools
 - OSGi Bordmittel
 - PAX Runner
 - OSGi Bundle Repository und Features
 - Equinox P2
 - Weitere ...
- Repository Strategien
- Provisioning für OSGi Integrationstests

Integrationstest unter OSGi (1)

Anforderungen

- Tests laufen in einem OSGi-Framework ab
 - Zugriff auf alle OSGi-Konzepte möglich
 - Definierte Umgebung pro Test
 - Jeder Test bekommt seine eigene Konfiguration
 - Spezifische Konfiguration der erforderlichen Bundles
 - Automatisierung / Wiederholbarkeit
- ⇒ benötigen Bootstrapping-Installationsprozess

Integrationstest unter OSGi (2)

Standardverfahren

- Es wird ein OSGi-Framework gestartet
- Es werden die benötigten Bundles bereitgestellt
- Es wird *on-the-fly* ein Testbundle erstellt/installiert
- Die Tests werden durchgeführt
- OSGi-Framework wird heruntergefahren

⇒ Welchen Umfang hat ein Test?

Integrationstest unter OSGi (3)

Varianten

- Umfang der Tests je gestartetem Framework
 - Pro Testmethode, Testklasse, zu testendem Bundle
 - konfigurierbar
- OSGi-Framework
 - embedded, extern
- Konfigurationsmöglichkeiten
 - Konfiguration per Programmierung
 - Konfiguration durch/in Buildprozess

Integrationstest unter OSGi (4)

Weitere Varianten

- Testframework
 - JUnit 3/4, TestNG
- Integration in IDE
- Ausführungsumgebung
 - Buildtool, IDE, händisch ...

PAX Exam (v 1.x)

- Umfang der Test pro Provisioning
 - Start eines Frameworks pro Testmethode
 - Viele Testmethoden bremsen
 - 1500 Testmethoden in 2 min (Erfahrung)
- Embedded oder externer OSGi-Framework
 - Externer OSGi-Framework
 - Kommunikation zwischen TestContainer und Framework via RMI
- Konfigurationsmöglichkeiten
 - Programmatische Konfiguration (inkl. Annotations)
 - Integration in maven-Buildprozess
 - Konfigurationsdatei wird durch *maven-plugins* erzeugt

PAX Exam (v 2)

- Löst sich von PAX Runner
 - Unterstützt mehrere TestContainer
 - Embedded, Remote, ACE-Client
- Konzept der Probes
 - Probes enthalten zu testenden Code
 - Normalerweise ‚under-the-hood‘
 - Ausführung via *extender pattern*
- *Restart Policies*
 - Welche Tests werden in einem Container ausgeführt

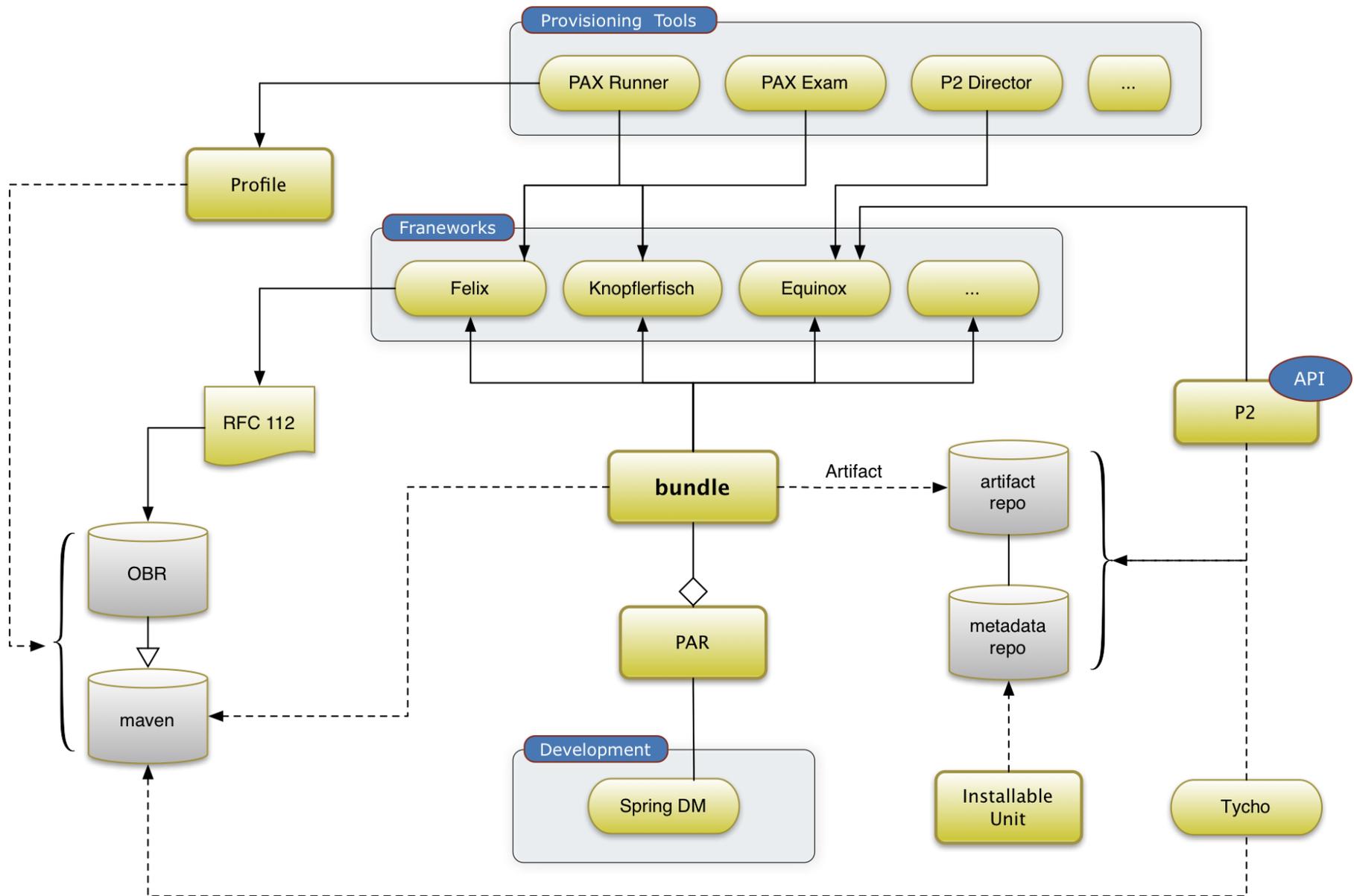
PAX Exam

- Vergleichbare Tools
 - Junit4osgi
 - <http://felix.apache.org/site/apache-felix-ipojo-junit4osgi.html>
 - Spring DM testsupport
 - <http://static.springsource.org/osgi/docs/1.2.1/reference/html/testing.html>
- Alle basieren auf Standardverfahren
 - Variieren gemäß „Integrationstest unter OSGi (4)“

Zusammenfassung

- Provisioningprozesse
 - Bootstrapping
 - Zur Laufzeit
- Provisioningprozesse im Integrationstest
- Provisioning unter OSGi ist noch nicht gelöst
 - Je nach Ausgangssituation und Anforderungen unterschiedliche Tools, Repositories und Konzepte

Übersicht



Fragen?



Gesellschaft für
Informations- und
Kommunikationssysteme mbH

www.iks-gmbh.com

c.schaedel@iks-gmbh.com

c.schmidt-casdorff@iks-gmbh.com