Ohne Build geht's besser:

Makeloses Java mit dem z²-Environment

Henning Blohm

5.7.2012

JAVAFORUM stuttgart 2012

ZFABRIK be powered!

Z2 ist ein radikal neuer* Ansatz für System Life-Cycle Management in Java

* jedenfalls für Java

Ein Builtool?

Ein Deploy-Tool?

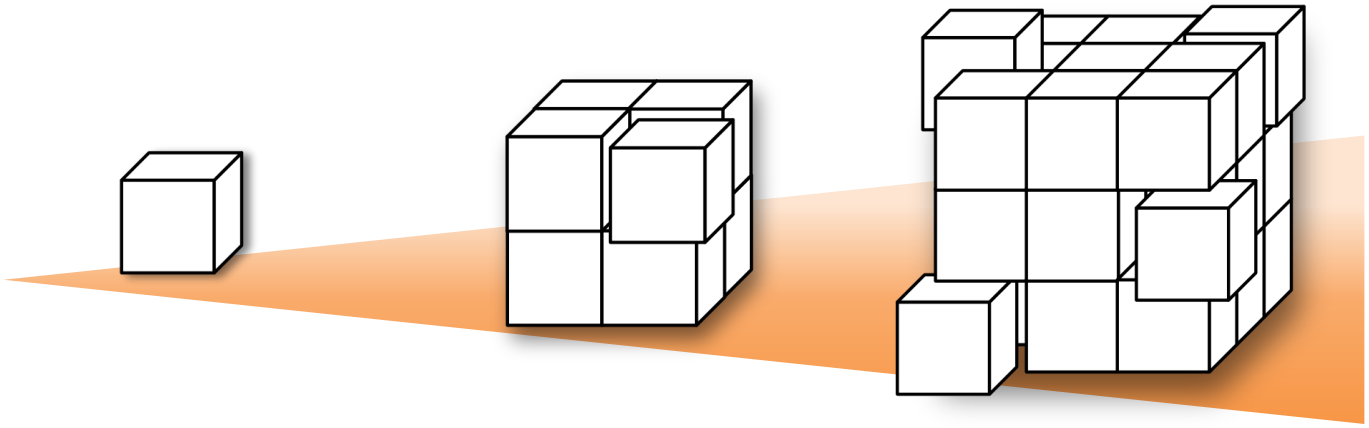
Ein Byte Code Replacement Tool?

Nein!

Viel einfacher:

Laufzeit trifft Source Verwaltung

Warum Z2?

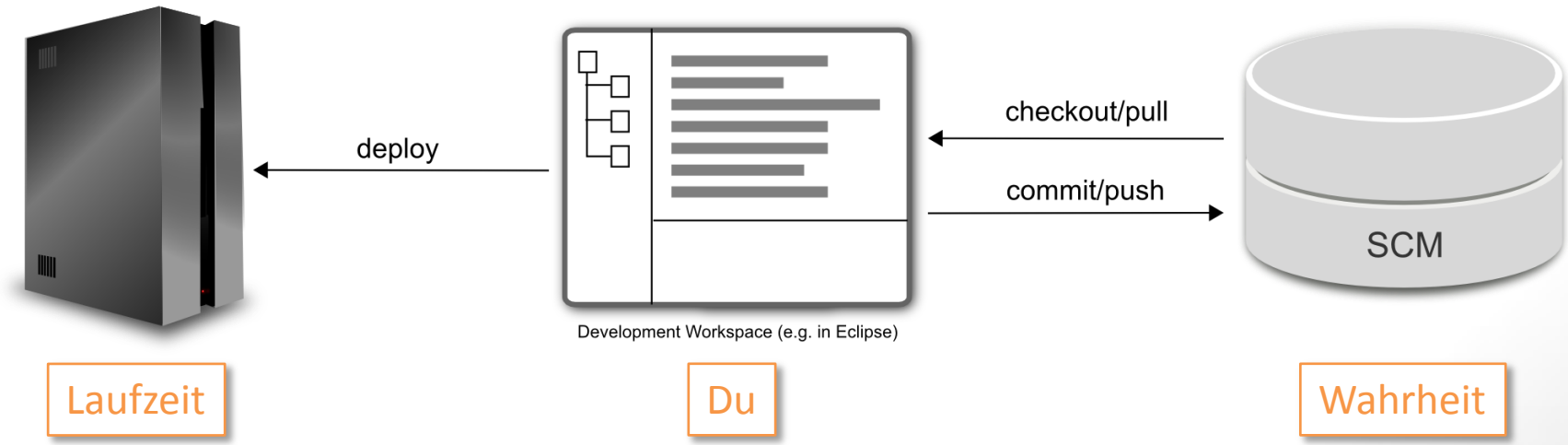


- Aus einfach und klein wird groß und modular
- Wir wollen modular sein, um zu skalieren
- Lösung!=Infrastruktur: verteilte Komplexität
- Lokale updates sind mühsam

► Es gibt ein ständiges Integrationsproblem

ZFABRIK be powered!

Das Integrationsleck



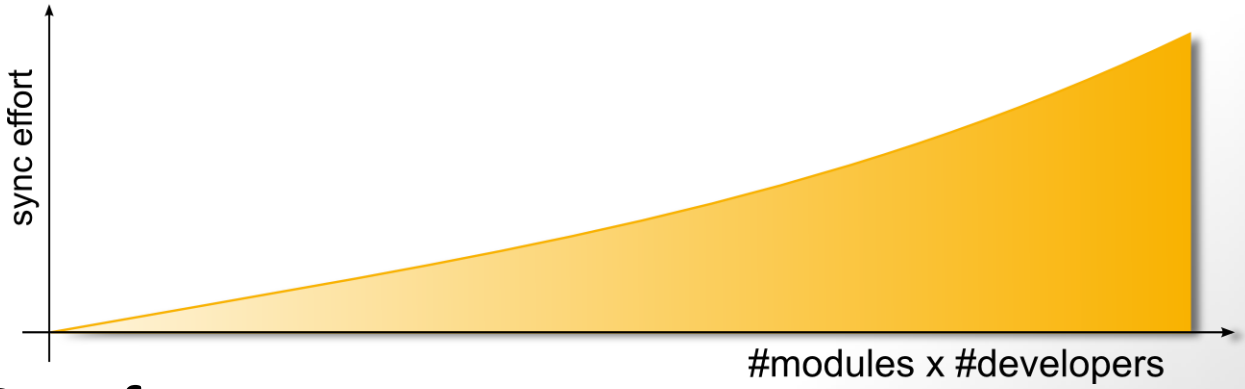
- Laufzeit!=Wahrheit => Integration erforderlich
- Modularität => Push Deployment wird immer komplizierter
- Schlechte Integration + Komplexe Lösung + Team => R.I.P
- Integrationsleck liegt in der Entwicklungsumgebung

► Was tun?

FABRIK be powered!

Was tun?

1. Gegen anpumpen:
   - Infrastruktur zum Nachbauen und Testen aufstellen
   - Z. B. Continuous Integration + ANT / Maven
   - Es sieht aber eher so aus:

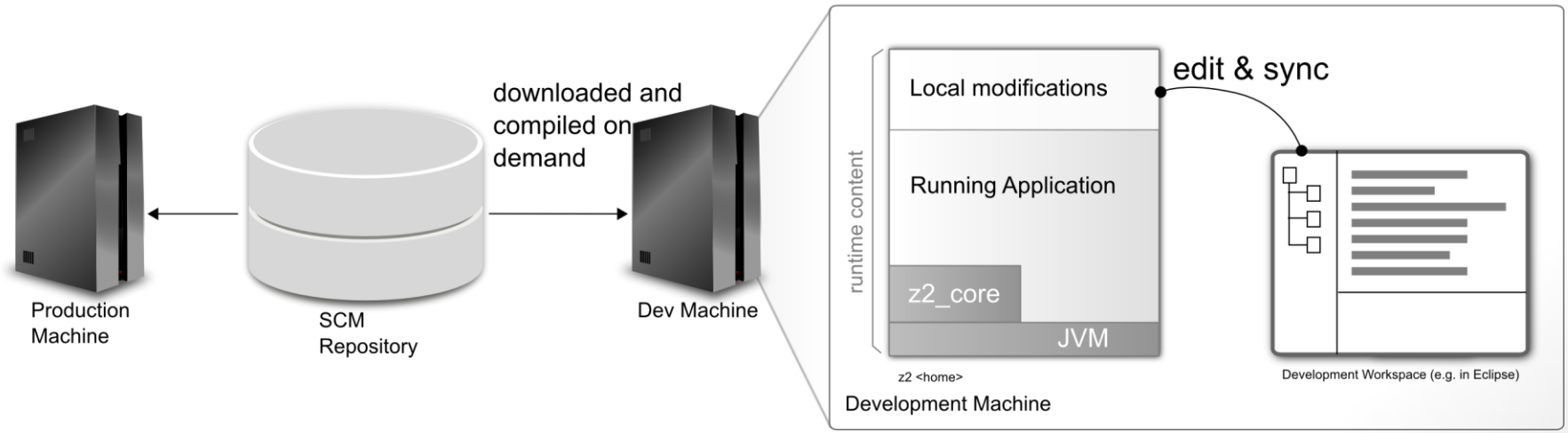


2. Leck Stopfen:

▶ Z2-Environment

FABRIK be powered!

Wie funktioniert Z2?

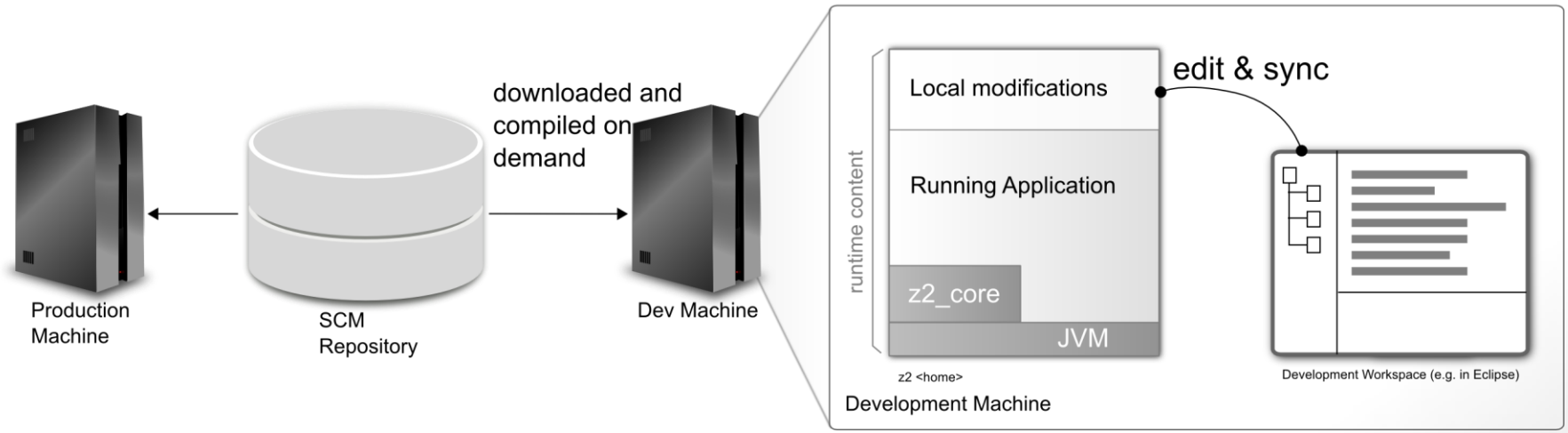


- Laufzeit lädt updates nach Bedarf aus SCM
- Kompilation (Java und mehr) nach Bedarf in der Laufzeit
- Lokales Overlay zum Entwickeln „as fits“
- Eine Struktur, die das System vollständig definiert

► Sytemzentrischer Ansatz

FABRIK be powered!

Oder anders gesagt:



- Kein Deployment: Yippie!
- Kein Build: Hurra!
- Nur auschecken woran man arbeitet: Oh ja!
- Systemstruktur im Ganzen verstehen: Danke!

► Die Quelle ist das System

FABRIK be powered!

Was bringt das?

- Einfachstes Aufsetzen der Dev-Umgebung
- Keine Build-Umgebung, keine Build-skripte
- Schnelle Roundtrips
- Systemskalierung mit minimalem Aufwand
- Jedes System ist immer patchbar
- Keine Unklarheiten, welcher Code wo läuft

► Weniger Infrastruktur und weniger „Waste“

FABRIK be powered!

- Aufsetzen und Starten (ca. 250kLoC)
- Modifikation und Synchronisation

DEMO

FABRIK be powered!

Was ist drin in Z2?

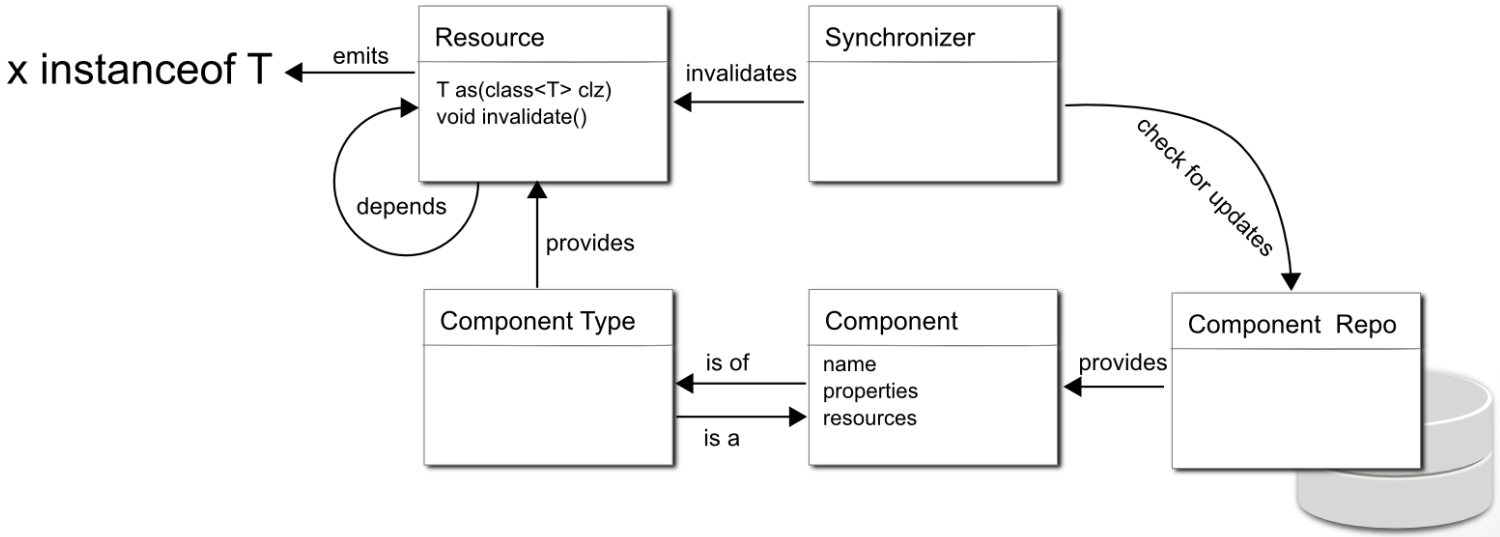
- z2@base: Z2 Basis – nur das Essenzielle
  - Jetty als Web Container
- z2@spring: Spring für Z2
  - Spring 3.0.5 und integrationen
- z2@hadoop: (noch) nicht OSS
  - Hadoop config und Map/Reduce Programmierung

FABRIK be powered!

ARCHITEKTUR

ZFABRIK be powered!

Core

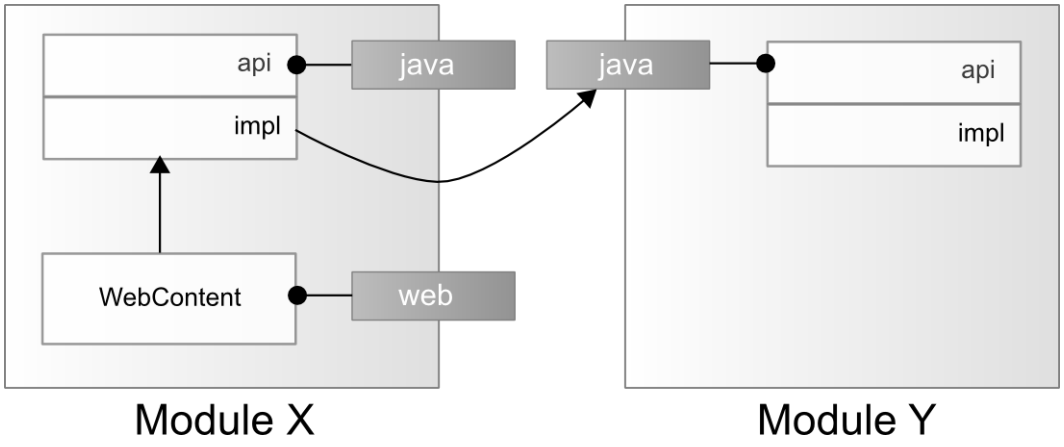


- Z2 core kennt wenige grundlegende Konzepte
- Alles weitere sind spezifische Komponententypen
- Komponententypen sind Grundlage der Erweiterbarkeit
- Bsp: Java, Worker Processes, Data Sources, Web Apps, ….

►Siehe Component Type Reference in Dokumentation

FABRIK be powered!

Modularisierung

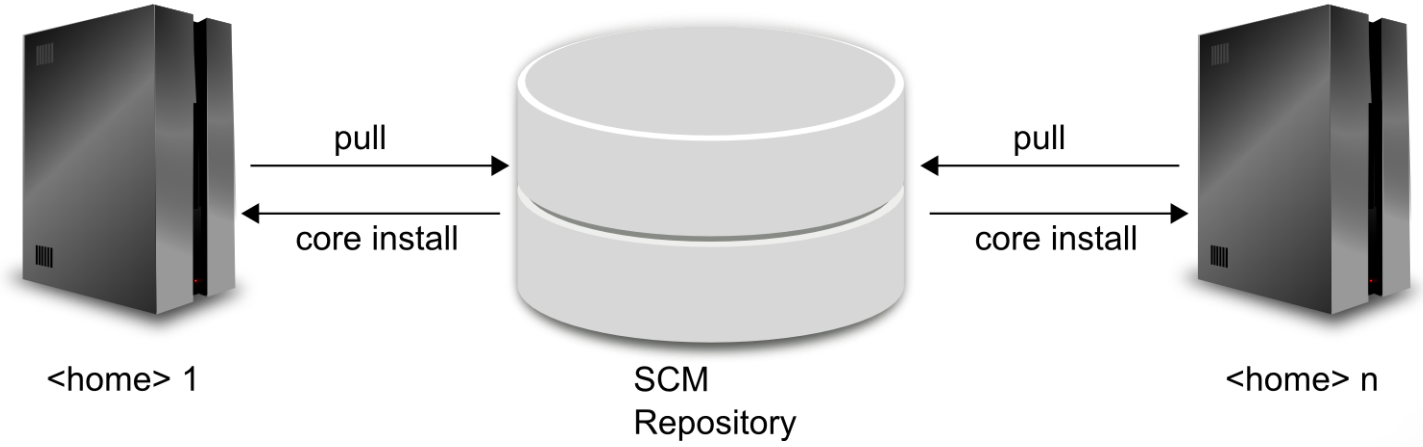


- Konvention: <modul>/<name> == Component Name
- Konvention: Mein Java ist <modul>/java
- Java: Separation in API und IMPL class loading + refs
- Modul ist Hülle für mehr als Java: Web App, Data Source,…

►Einfache aber effektive Isolation, gezieltes Sharing

FABRIK be powered!

Betrieb

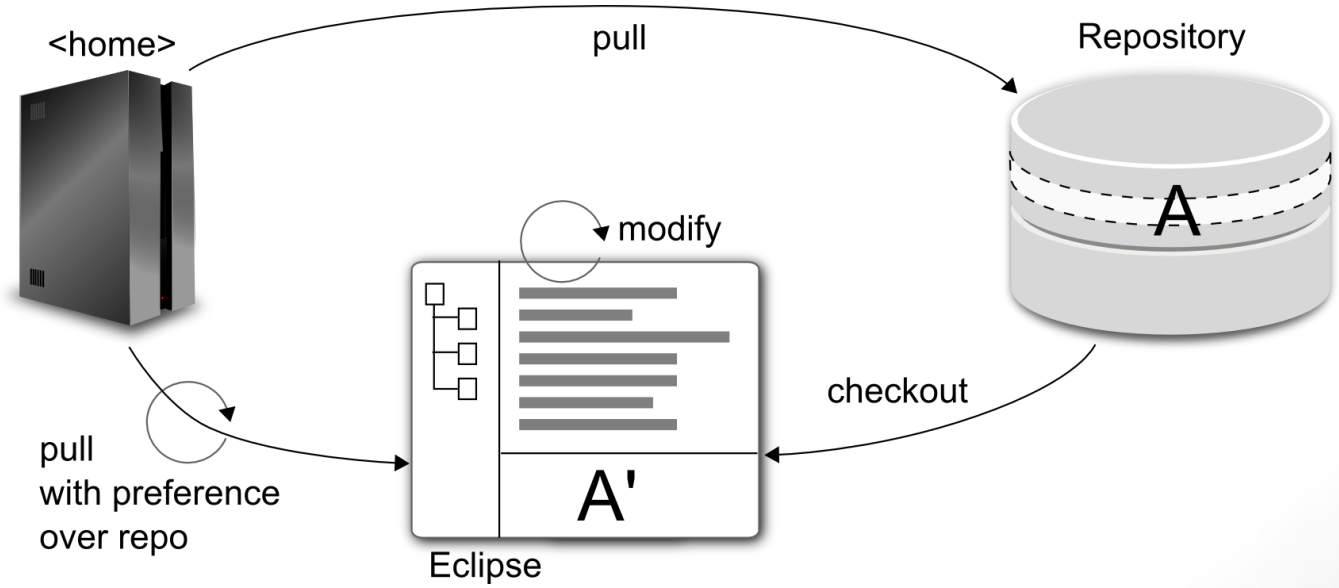


- Ein Z2 <home> entsteht durch check out des Z2 core
- Alles andere ergibt sich daraus
- Core updates (selten) per „svn up“ o.ä
- Z2 kann im server mode oder embedded benutzt werden

►Einfaches Ausskalierung

FABRIK be powered!

Entwicklung

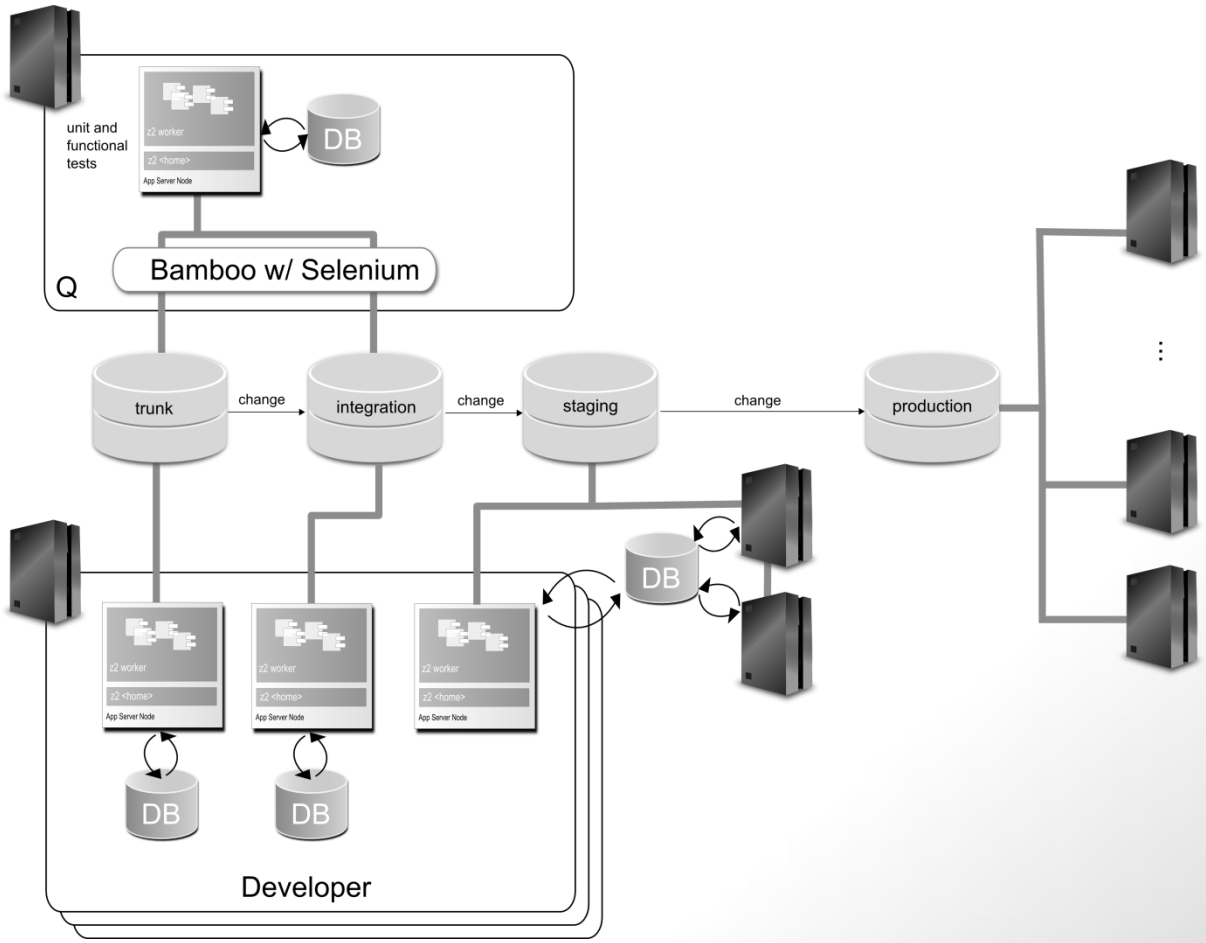


- Projekte (==Modul) in IDE auschecken und „armen“
- Eclipsoid plugin löst classpath von lokalem <home> auf
- Lokales Projekt hat Präferenz gegenüber Repo

►Entwicklung auf Modulebene mit sofortiger Integration

FABRIK be powered!

Beispiel: Echte Landschaft

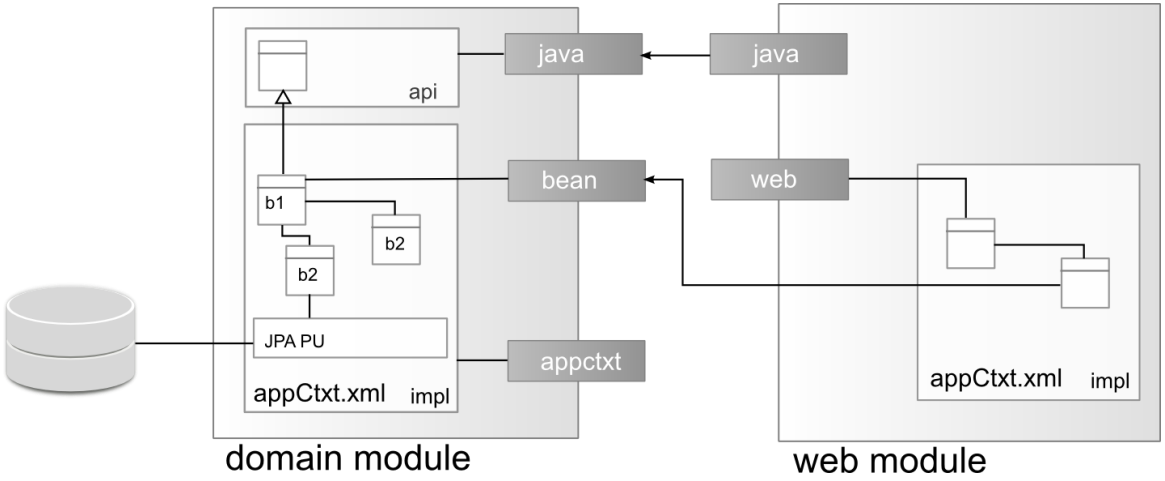


FABRIK be powered!

INTEGRATIONEN & FEATURES

FABRIK be powered!

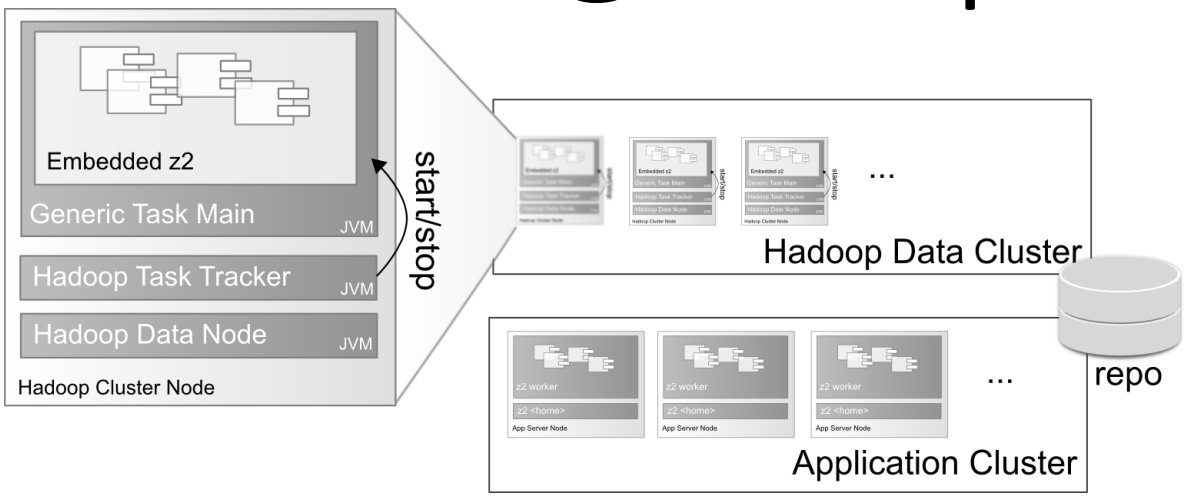
z2@spring



- Spring Application Contexts und Beans als Component Type
- Voller support von Annotation-based config und Spring/AspectJ
- Z2 verlängert Spring wiring über Modulgrenze
- Spring beans konsumieren oder exponieren services der Lösung

► Viel besser als Java EE, viel offener als SCA jemals wäre

z2@hadoop



- Map/Reduce Job als Component Type
- Ausführung per embedded mode als Hadoop task
- M/R Job benutzt identische Modularisierung wie alles andere
- Einfachstes Job scheduling, ohne build und programmatisch

►Hadoop wird natürlicher Teil der Lösung

UND SCHLIEßLICH…

FABRIK be powered!

Was ist Z2 nicht?

- Z2 ist primär kein Buildtool
  - Man kann sich aber Binaries abholen
  - Nützlich für automatisierte Tests etc.
  - Deployer als Sync-Extension angedacht
- Z2 ist ganz anders als Maven
  - Hier Systeme, dort Communities

FABRIK be powered!

Wo geht‘s weiter?

WWW	http://www.z2-environment.eu
Doku	http://www.z2-environment.eu/v20doc
Blog	http://www.z2-environment.net/blog
Forum	In Arbeit.
Kontakt	henning.blohm@zfabrik.de

ZFABRIK be powered!

© 2012 ZFabrik Software KG

HERZLICHEN DANK!

ZFABRIK be powered!