

Mit Java entwickeln und liefern: besser ohne Build & Deploy

Oder: Es liegt nicht an der Sprache





zuviel einpacken, auspacken ...
zu weit weg von der Aus
zu viel code, zu
zu variabel, zu komplex, zu
zu wenig integriert, zu modular,



Scripts?



ABAP?



ABAP!

SAP ECC ~ $238 * 10^6$ LoC

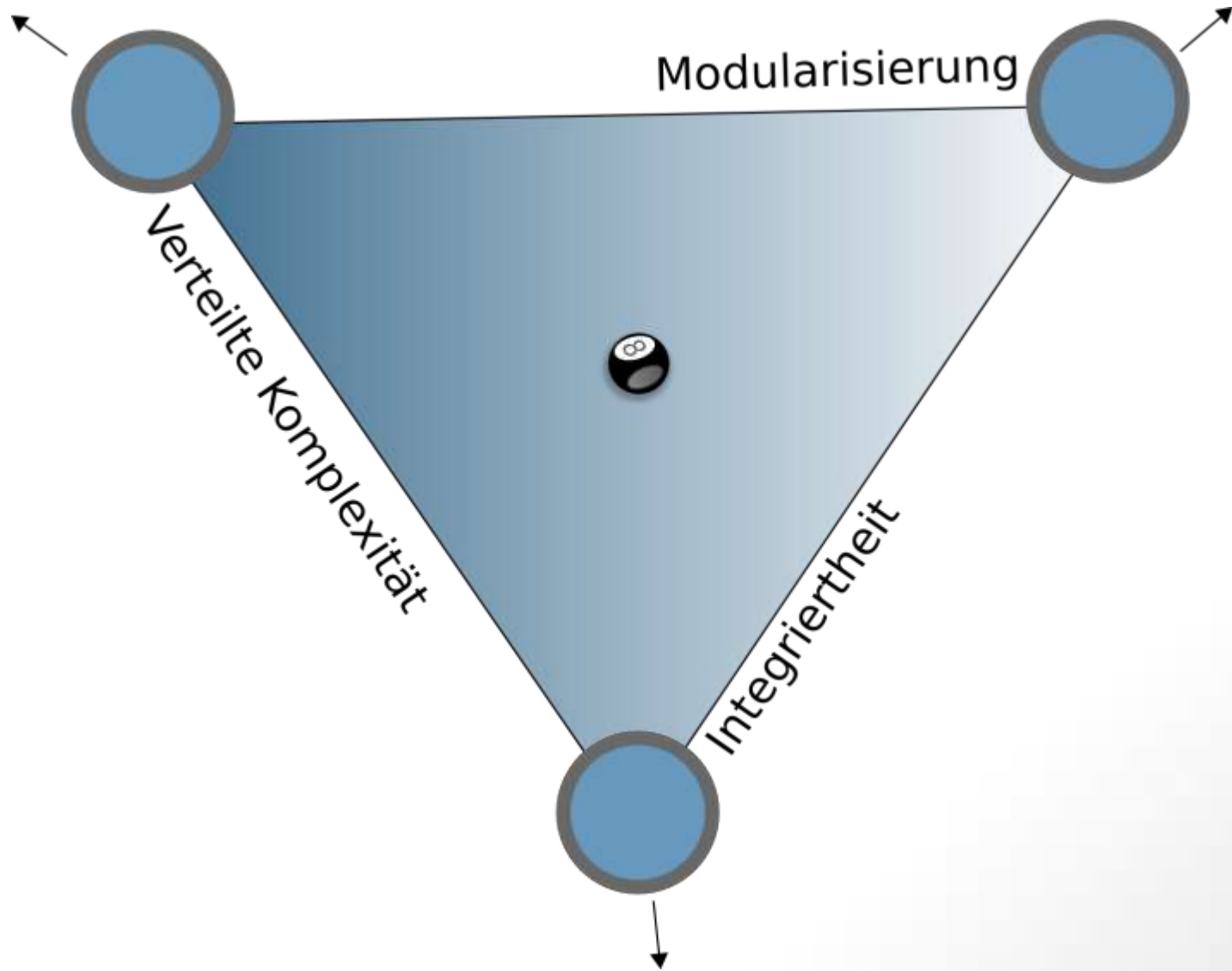
(http://de.wikipedia.org/wiki/Lines_of_Code)

Kundencode: Bis zu $6 * 10^6$ LoC

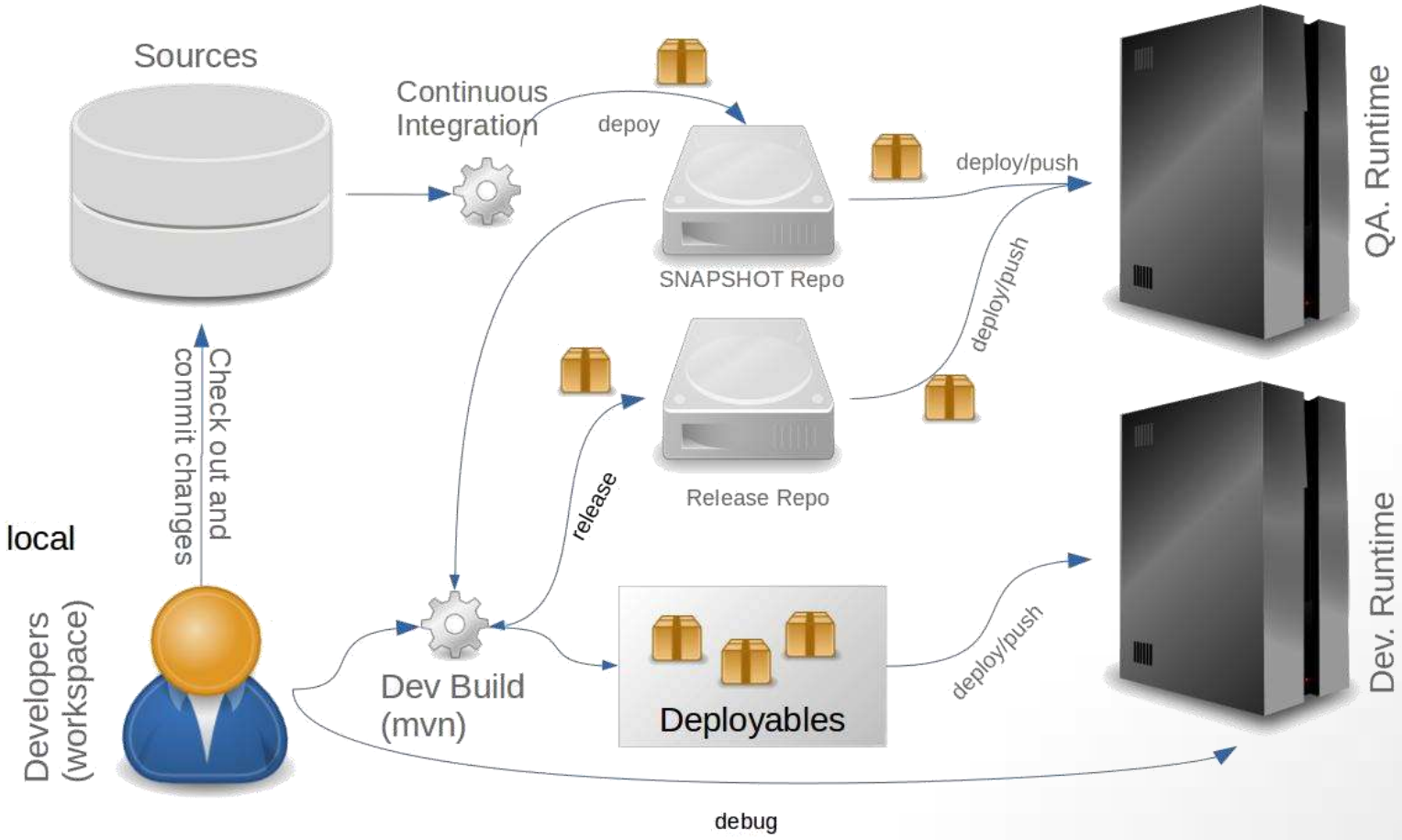
(<http://scn.sap.com/community/abap/blog/2009/01/20/how-many-lines-of-custom-abap-code-are-inside-your-system>)



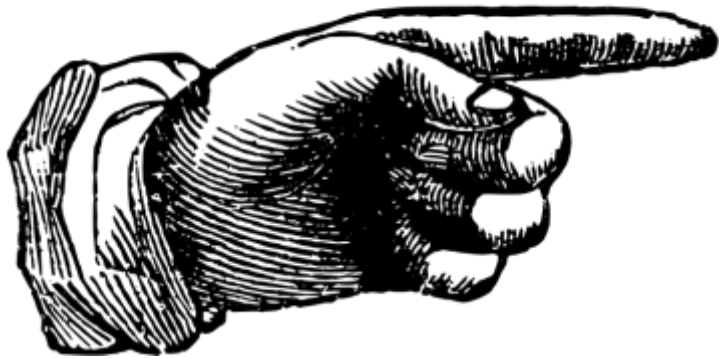
Was passiert hier?



Huh?



Please Notice This

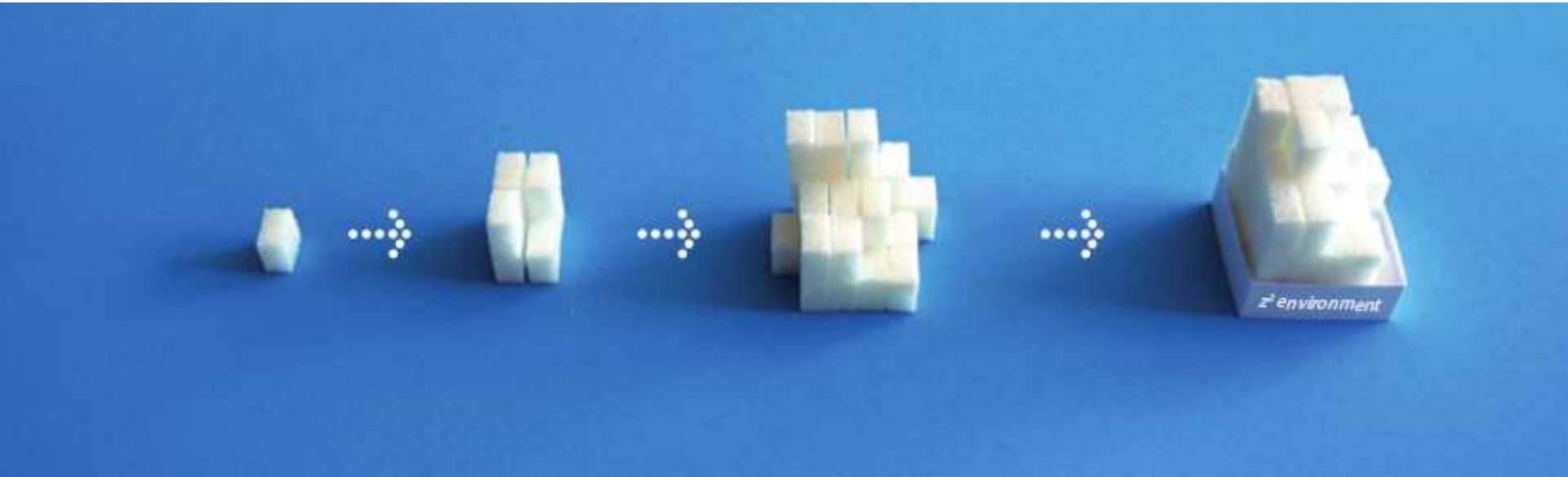


Kosten einer Änderung

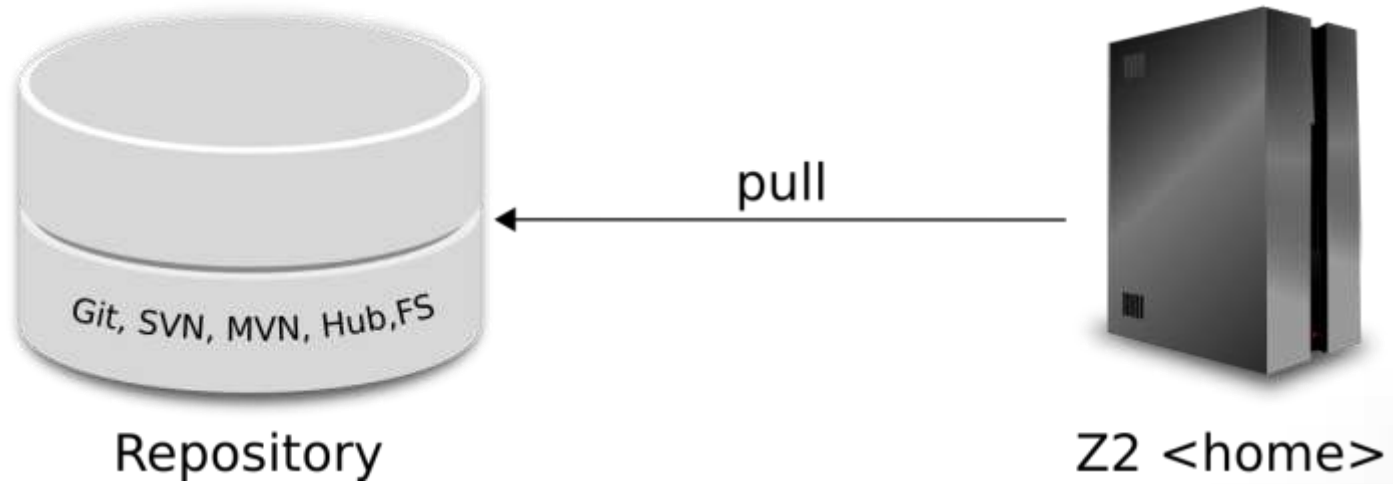
~

Anzahl der konfigurierbaren
oder manuellen Schritte

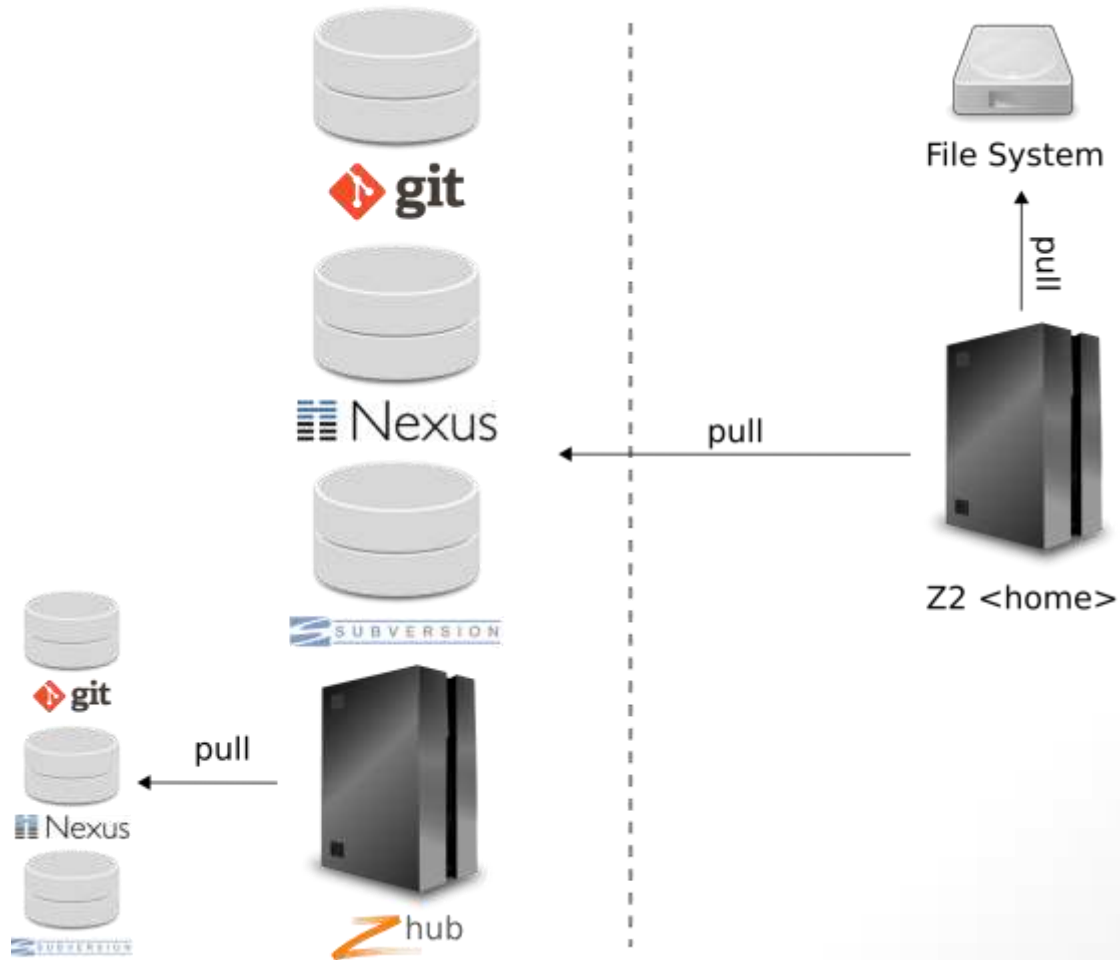
Gesucht: Produktiv bleiben bei steigender Komplexität und mit hinreichender Modularisierung



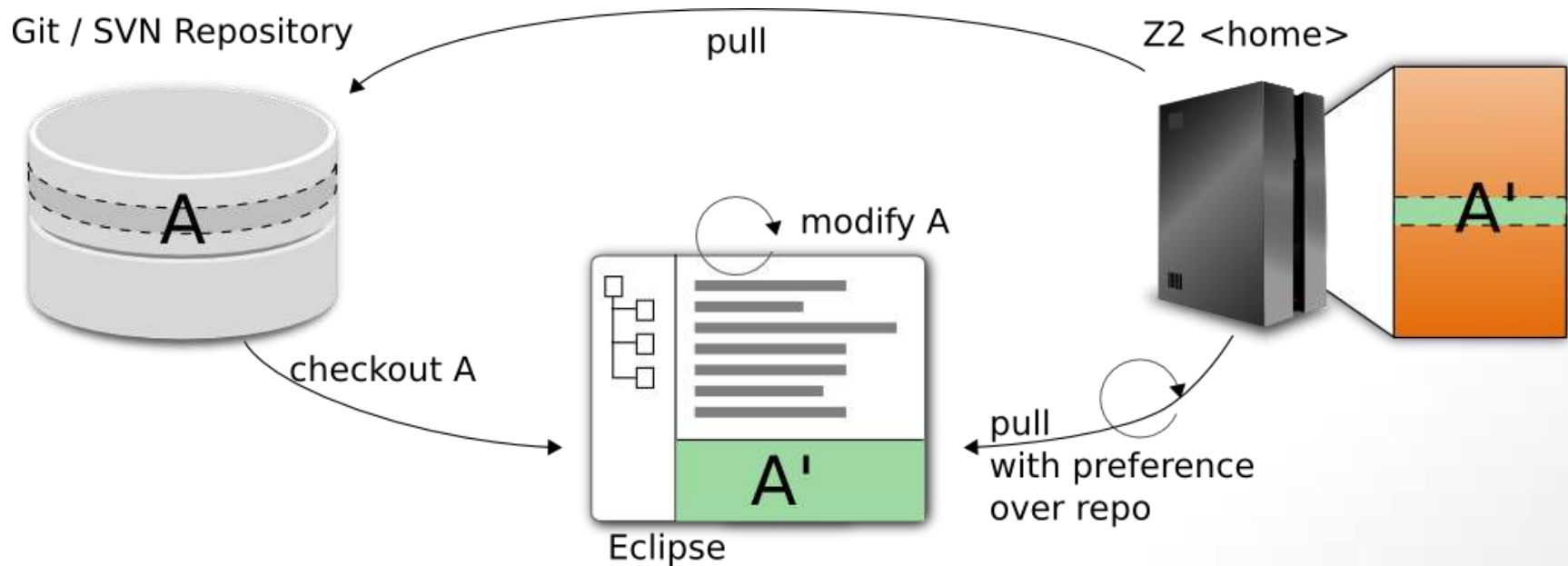
Z2 - Prinzip



Z2 - Quellen



Z2 - Entwicklung



Für

- Wachsende Lösungen / Websites
- Erweiterbarkeit wichtig
- Ständig gewartet
- Immer patchbar

Mit

- No-Deployment deployment pipeline
- Cloud Scale-Out
- Seamless Hadoop Map/Reduce

Keine Kompromisse

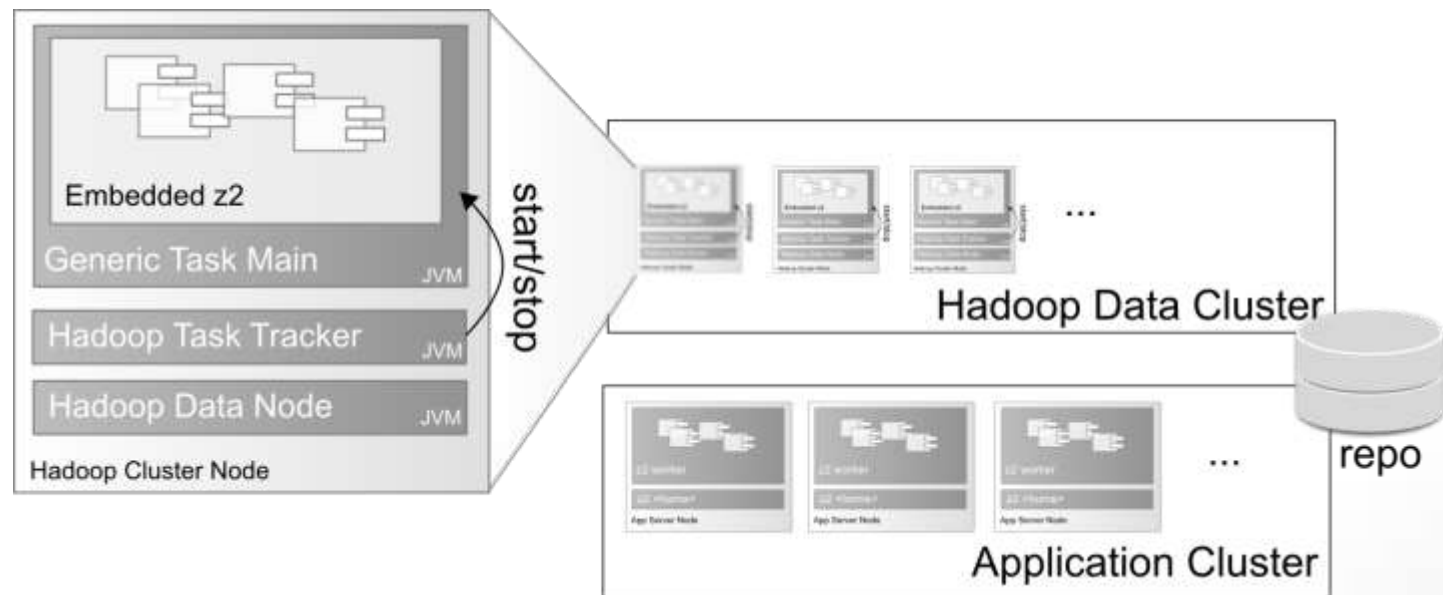


Bsp: Voll-integrierte Serverside UT

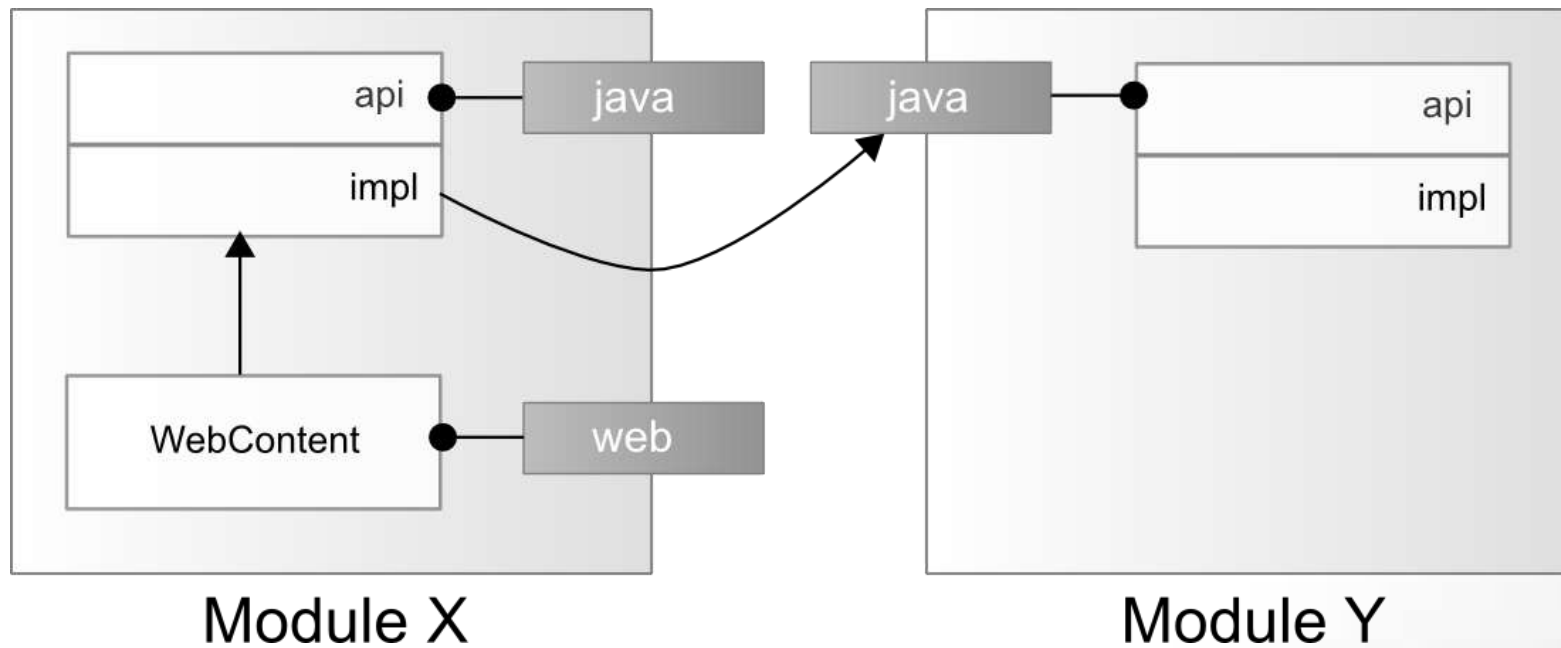
```
@RunWith(Z2UnitTestRunner.class)
@Z2UnitTest(
    componentName = "com.gi-de.tt.domain/java",
    dependencies = {"com.gi-de.tt.domain/applicationContext"}
)
@Configurable
@Transactional
public class SomeStuff {
    @Autowired private ProductRepository productRepository;

    @Test
    public void triggerDomainGC() {
        this.productRepository.purgeProductPropertyBinaryGarbage();
    }
}
```

“Operational” Map/Reduce



Service-Modularisierung



Z2 ist Open Source aus Baden

Mach doch mit!

<https://redmine.z2-environment.net>





ENTWICKLERTAG

meet the SPEAKER @speakerlounge



1. OG DIREKT ÜBER DEM EMPFANG