

Daten für Performance-Tests

C1 SetCon Performance Day
5. Oktober 2010 13:00 - 13:45

Update vom 2. Juli 2011

Volker Bergmann

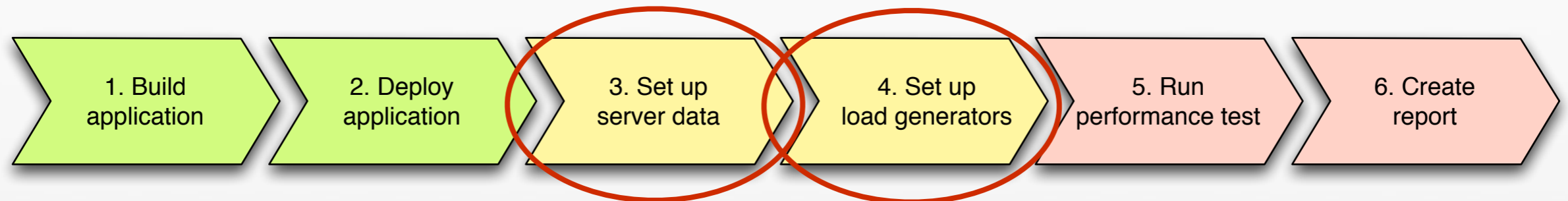
Kennen Sie den?

- 13 Jahre professionelle Software-Entwicklung
- zahlreiche Großprojekte
- Analyse, Design, Implementation, Test
- Bestrebt, Software-Qualität schon in frühen Projektphasen zu sichern
- Performance-Fokus, besonders for J2EE Server-Software
- Entwicklung von Open Source Testwerkzeugen
- Er heißt Volker Bergmann



Performance Testing Process

- Diese Schritte sind Bestandteil fast jeden Performance-Tests



- Datenaspekte (Schritte 3 und 4)
 - oft vernachlässigt
 - essentiell für die Aussagekraft des Tests
 - Thema des Vortrags

Performanz-Prognose

- Eine möglichst genaue Prognose der Produktionsperformance erfordert den Test mit
 - produktionsähnlicher Hardware
 - produktionsähnlichem Client-Verhalten
 - produktionsähnlichen Daten
 - Menge
 - Struktur
 - **aber: bessere Qualität**

Produktionsdaten in Performance-Tests

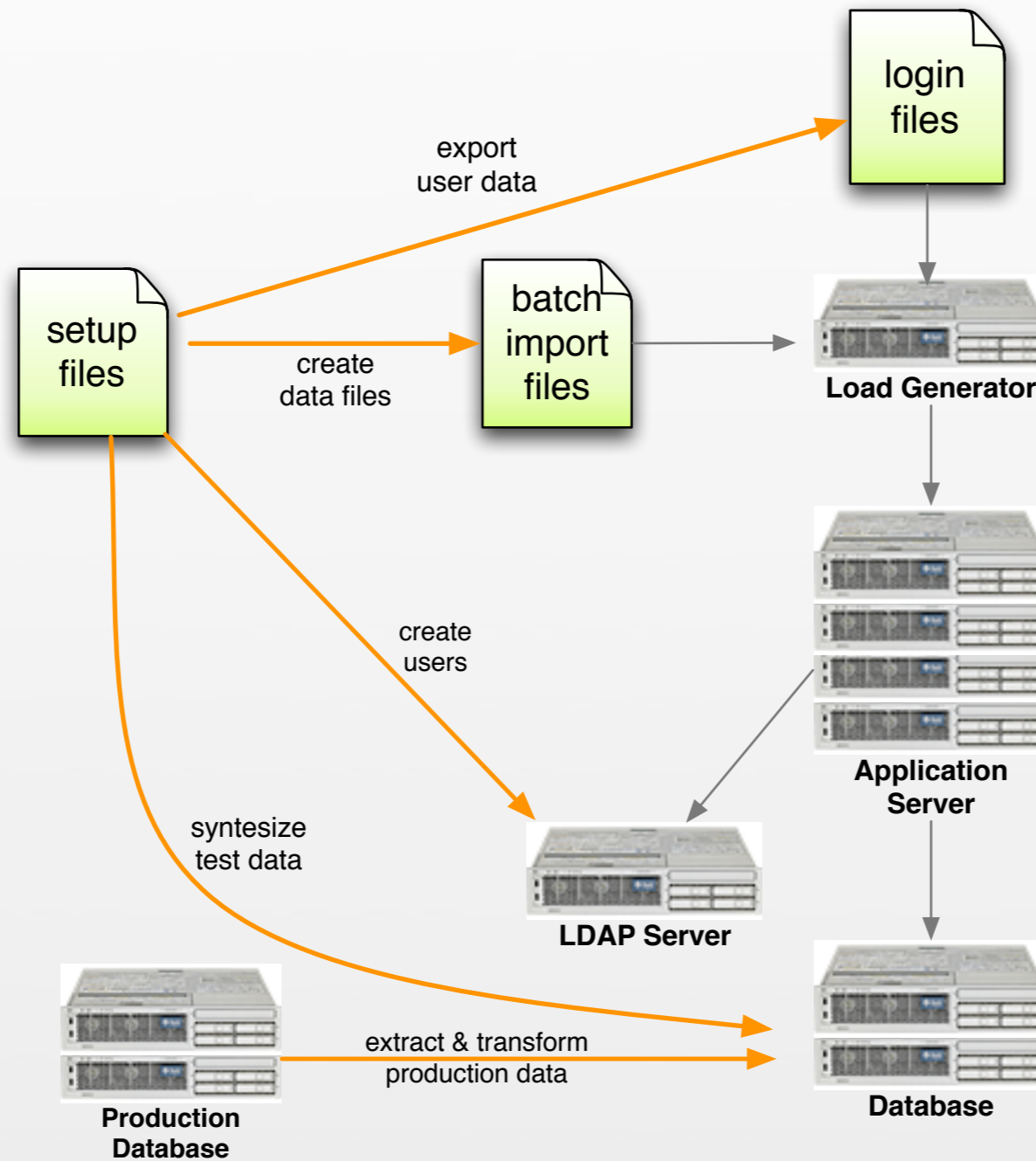
- **Sind Produktionsdaten**
 - **verfügbar?** (Outsourcing, erste Version, sensible Daten, Urlaub, Migration, ...)
 - **anonymisiert?** (Sensitivität der Daten, Kompetenz der Verantwortlichen, Meldepflichten...)
 - **valide?** (10-30% fehlerhafte Altdaten üblich durch Abbruch von Prozessen, Abschalten von Constraint-Validierung bei Migration, ...)
 - **anwendbar?** (obsoletere Daten, neuer Release erfordert neue Daten, geändertes Nutzerverhalten, mehr Nutzer in der Zukunft...)

Produktionsdaten-Fazit

- Produktionsdaten sind der perfekte Test-Maßstab
...für die Vergangenheit
- Im Regelfall müssen zusätzlich zur Extraktion von Produktionsdaten...
 - Datenvalidität geprüft werden
 - Untermengen extrahiert werden
 - Daten hinzugeneriert werden
- Extraktion und Anonymisierung von Produktionsdaten stellen nur einen Teil der Datenaufbereitung für Performance-Tests dar

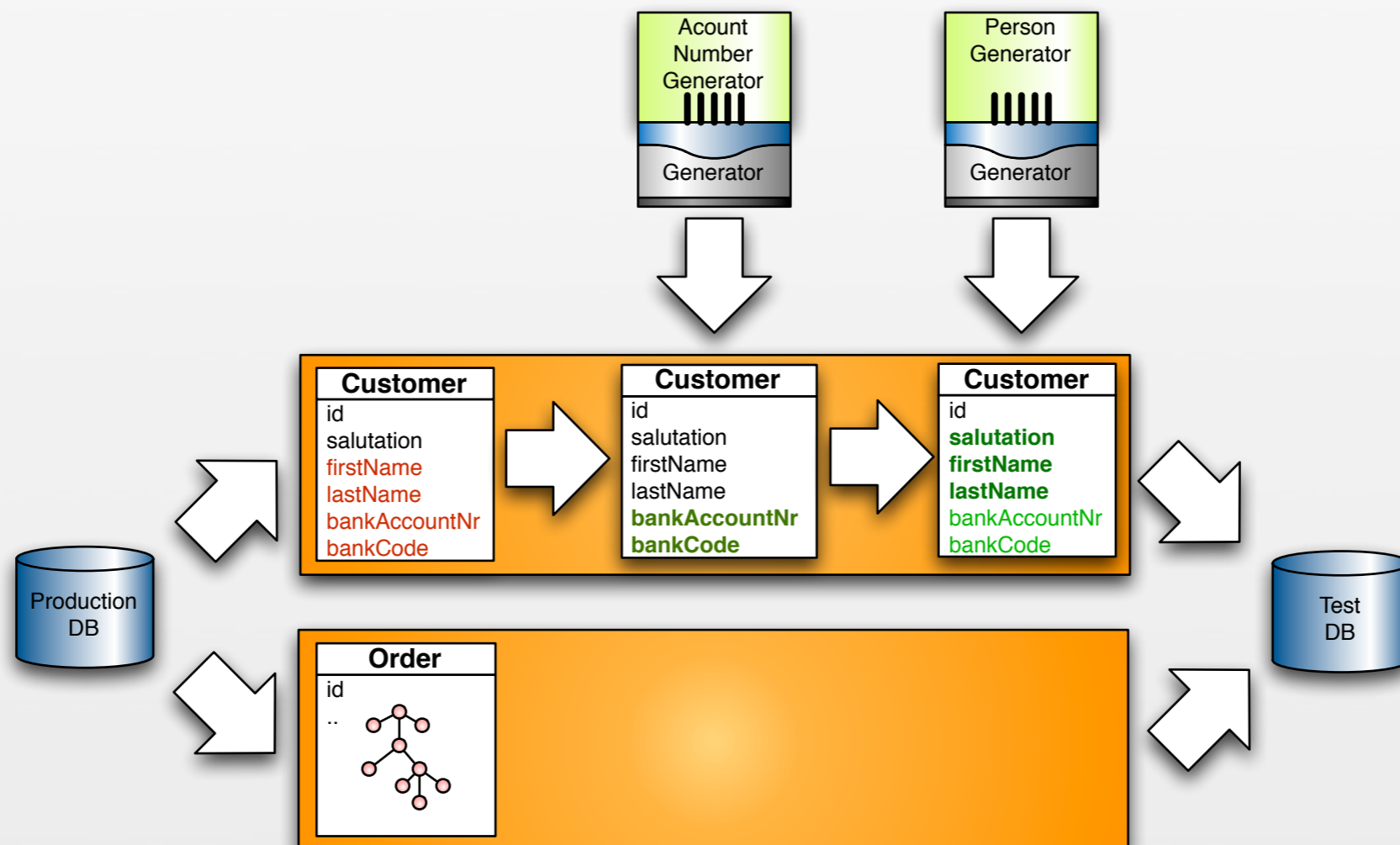


Datenaspekte im Performance-Test



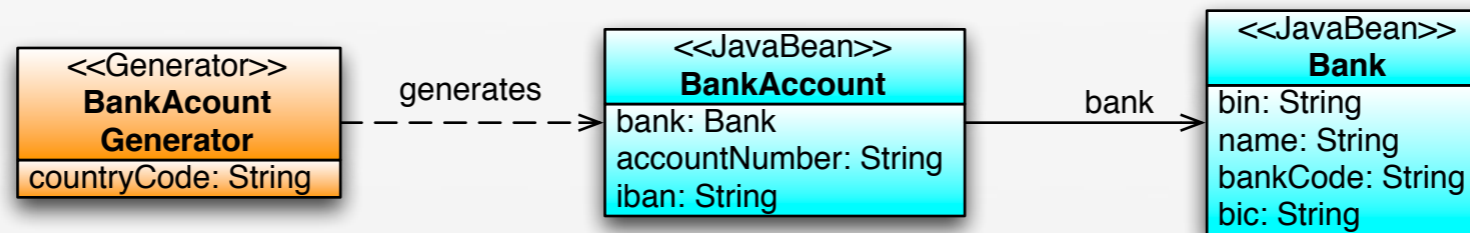
Extraktion & Anonymisierung

- Oft können Datenstrukturen 1:1 in das Testsystem übertragen werden
- Nur vertrauliche Daten müssen anonymisiert werden



Anonymisierung mit Benerator

- Generatormodule erzeugen Objekte oder konsistente Objektgraphen, deren Komponenten auf das Zieldatenmodell abgebildet werden



```

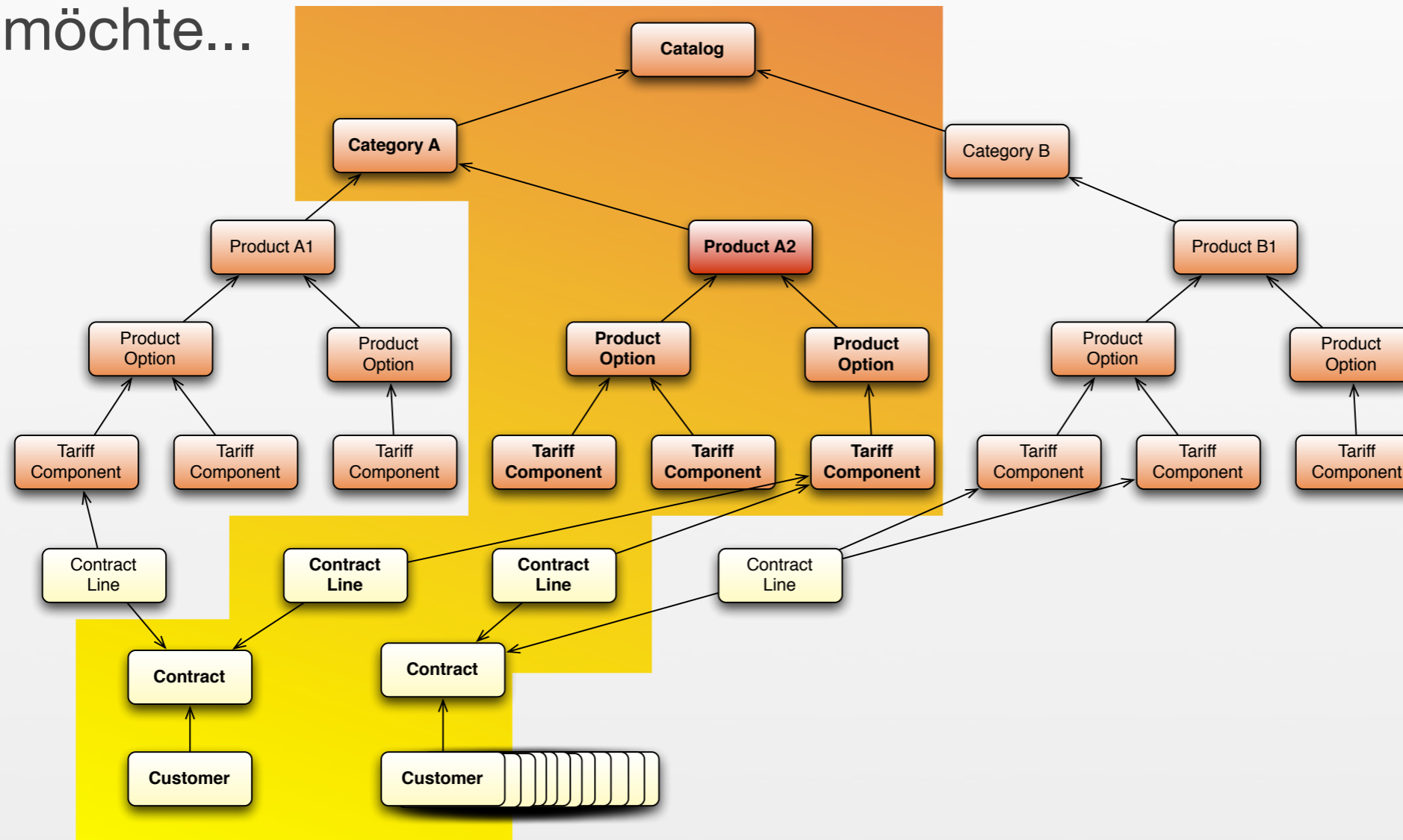
<database id="prod" readOnly="true" ... />
<database id="testdb" ... />

<iterate source="prod" type="CUSTOMER" consumer="testdb">
  <variable name="acct" generator="BankAccountGenerator" />
  <attribute name="BANK_ACCOUNT_NR" script="acct.accountNumber" />
  <attribute name="BANK_CODE" script="acct.bank.bankCode" />
</iterate>

<iterate source="prod" type="ORDER" consumer="testdb" />
  
```

Datenbank-Subsetting

- Wenn man z.B. den Produktkatalog auf ‚Product A2‘ beschränken möchte...



- ...wie wahrt man dann referentielle Integrität?

Jailer Jailer

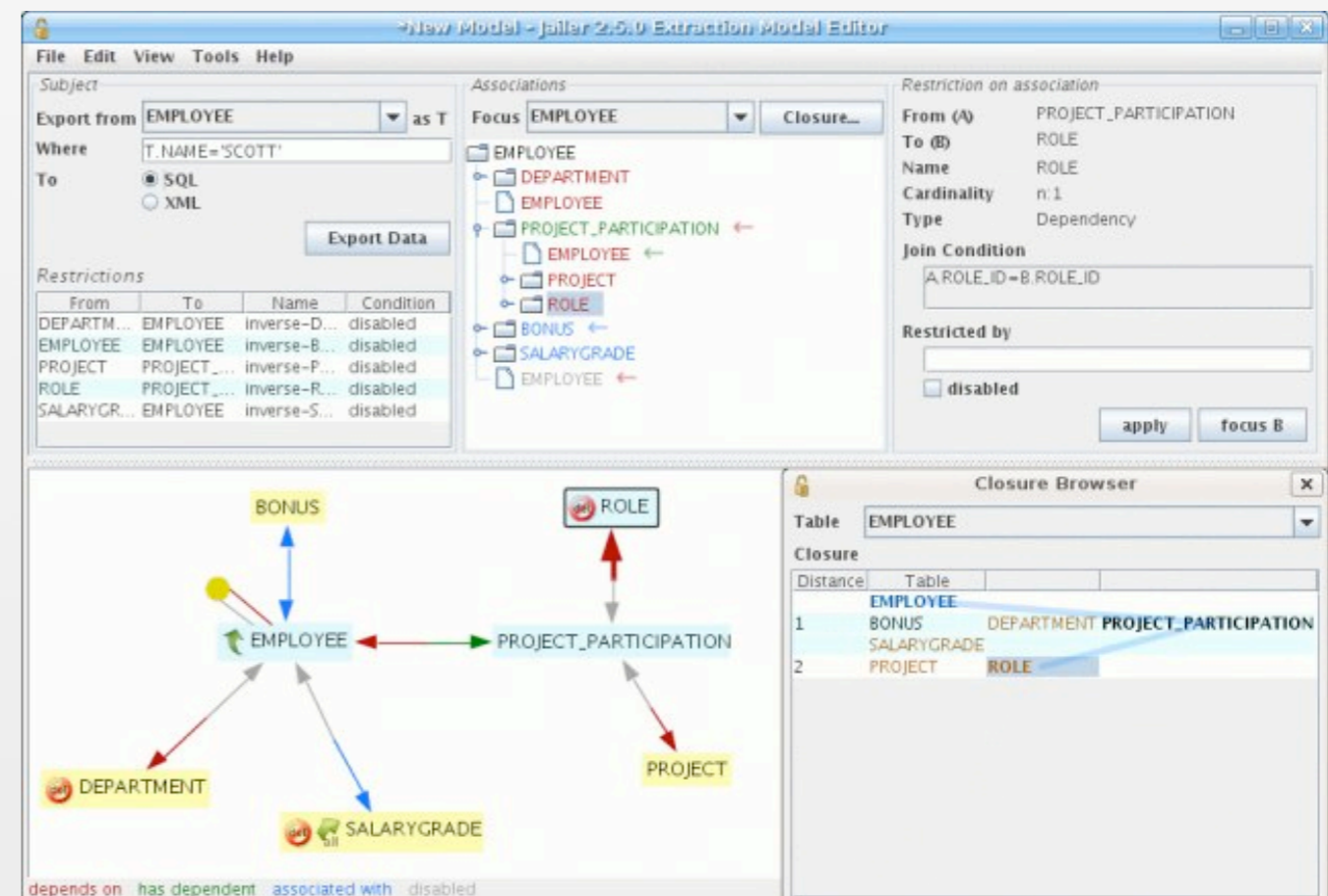
- Datenbank-Subsetting
- Open Source, aktives Projekt, Aktuelle Version: 3.4.7 (2010)
- Erkennt explizite Foreign Key-Beziehungen in der Datenbank

- Konfiguration:

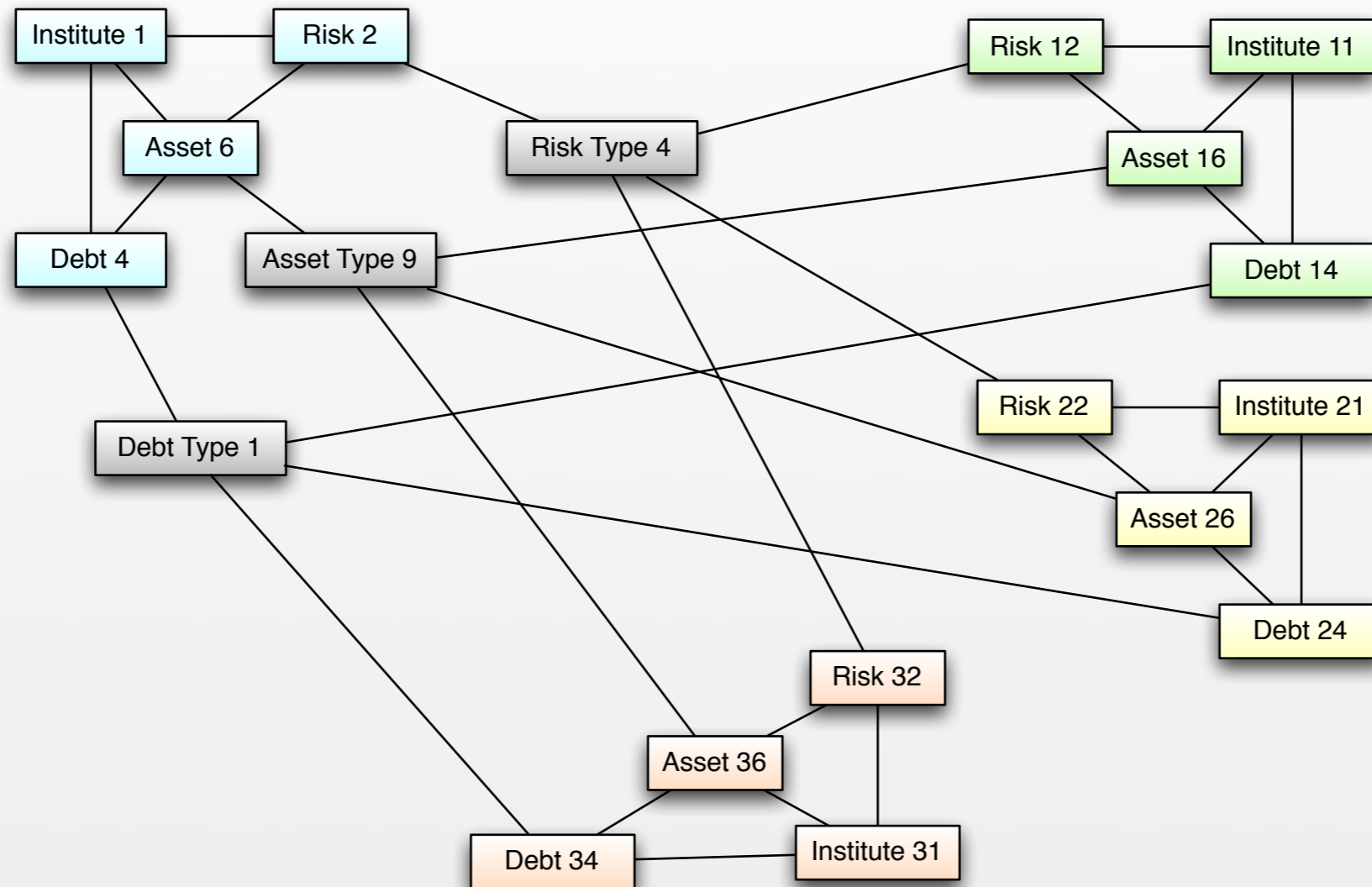
- implizite Foreign Keys
- Aggregation/Referenz

- Export in

- SQL
- XML
- DbUnit

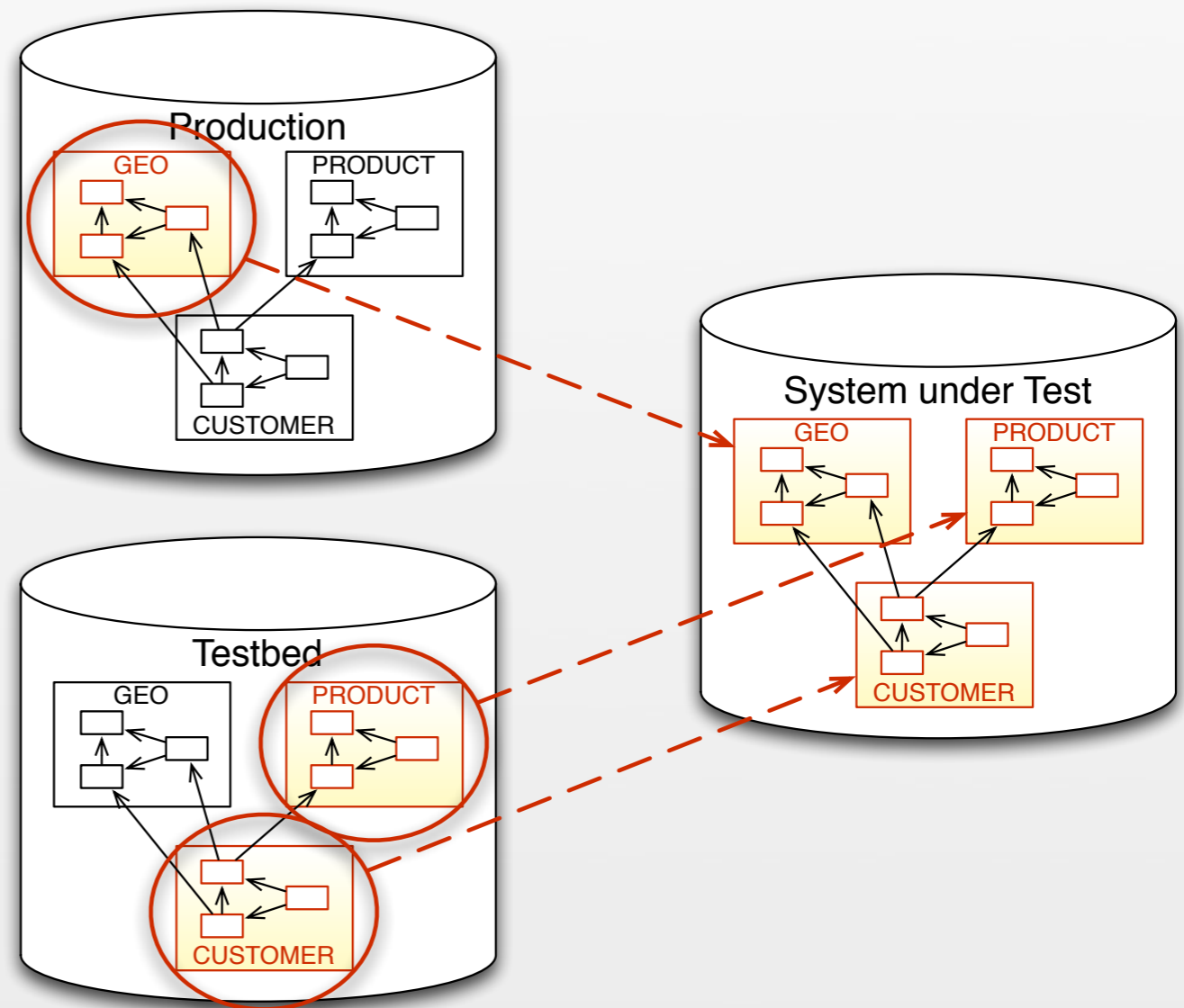


Datenreplikation



Zusammenführen von Datenquellen

- Hauptaufgaben:
 - Vergabe neuer Primärschlüssel
 - Ersetzen von Referenzen
 - Konsistenzprüfung
- ‚Transcoding‘ Feature in Benerator
- Wird durch kein weiteres Standardwerkzeug unterstützt



Datengenerierung

- Synthetische Erzeugung von Daten
 - ETL-Funktionalität: Extraktion und Transformation von Daten aus Datenbanken oder Dateien
 - Generator-Funktionalität: Generierung von Zufallsdaten / zufälligen Kombinationen von Basisdaten
- Kernproblem: Explizite und implizite Bedingungen in
 - Datenbank
 - Applikation
- Für die meisten Performance-Tests müssen zumindest Teile der Daten generiert werden
- Erfahrung: Typischerweise unbrauchbare Zufallsdaten

Wenn man gute Daten generieren könnte...

- **Test von Entwicklungsszenarien:**
 - Mobilfunk-Beispiel: Wie wäre die Systemperformance in einem Jahr, wenn wir durch uns unseren neuen Flatrate-Tarif 30% Neukunden gewinnen und alle Kunden 50% länger telefonieren würden?
- **Test auf Spezialprobleme:**
 - Beispiel Adressvalidierung: Gibt es bei der automatischen Korrektur von Postadressen Spezialkonstellationen, die problematisch sind („Killeradressen“?)
 - Ihr Beispiel: Der typische Prio 1-Performance-Incident nach 2 Monaten Produktion
 - Lösungsansatz: Generative Abdeckung aller Kombinations-/ Fehlermöglichkeiten (deckt u.U. auch funktionale Fehler auf)

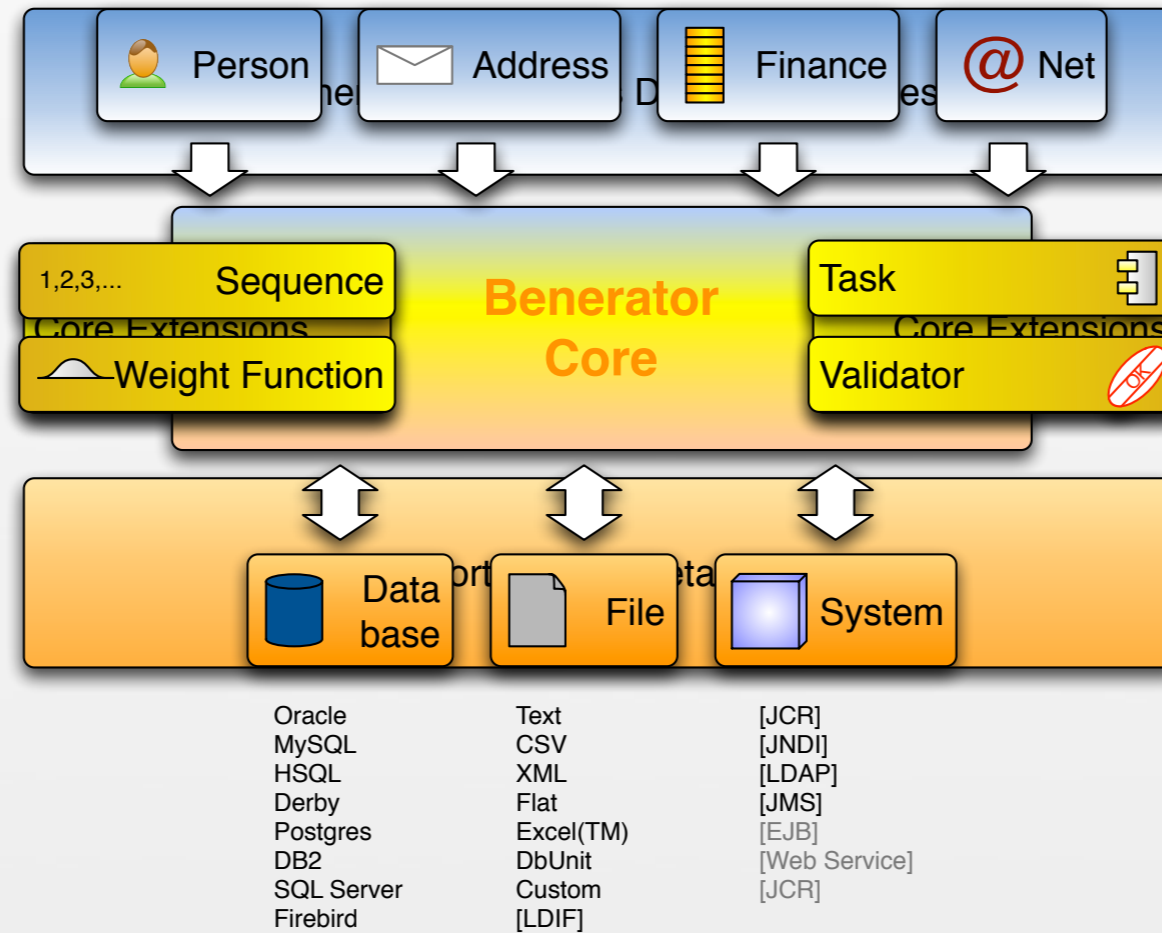
Wenn man gute Daten generieren könnte...

- **Performance-Test in frühen Projektphasen**
 - Start mit ‚Smoke‘-Daten, die sehr schnell konfiguriert sind
 - Verfeinerung der Generierung bei Implementationsfortschritt
- **Continuous Performance Testing**
- **Show-Case-Erstellung** und Schulungsdaten als Nebenprodukt

Das kann man: Benerator...

- entstand, um für Performance Tests Produktionsdaten simulieren zu können
- hat zum Hauptziel, valide Daten in großer Menge zu erzeugen und Produktionsdaten zu anonymisieren
- erlaubt die Generierung extrem komplexer Strukturen
- ist Open Source
- ist unabhängig vom Betriebssystem
- ist unabhängig von der Zielplattform, RDBMS, XML, CSV, XLS, ...
- erlaubt die Definition fachlicher Generierungskomponenten
- ist das Standardtool zur Testdatengenerierung

Benerator-Architektur



Benerator-Anwender



bislang 10.000 Downloads



Datengenerierung mit Benerator

Definiere eine Datenbank ,db' -->

```
<database id="db" ... />
```

Führe DDL/SQL-Skripte aus -->

```
<execute uri="create_tables.sql" target="db" />
```

Importiere Daten aus DbUnit-Dateien -->

```
<iterate source="products.dbunit.xml" consumer="db" />
```

Generiere valide Daten -->

```
<generate type="db_order" consumer="db">  
  <id name="id"  
    generator="new DBSequenceGenerator('SEQ_ORDER',db)"/>  
  <reference name="customer" targetType="db_customer"  
    distribution="random" />  
  <attribute name="created_at"  
    generator="CurrentDateGenerator" />  
  <attribute name="created_by" script="this.customer" />  
</generate>
```



Zusammenführung mit Benerator

Definiere zwei Quelldatenbanken ,db1' -->
... und ,db2' -->
... und eine Zieldatenbank -->

Führe DDL/SQL-Skripte aus -->

Importiere Daten aus DbUnit-Dateien -->

Generiere valide Daten -->

```
<database id="source1" ... />  
<database id="source2" ... />  
<database id="target" ... />  
  
<transcodingTask target="target" identity="id.xml" />  
  <transcode source="source1" table="COUNTRY">  
    <cascade ref="STATE.COUNTRY_REF" />  
  </transcode>  
  <transcode source="source2" table="CTR_CURRENCY" />  
</generate>
```

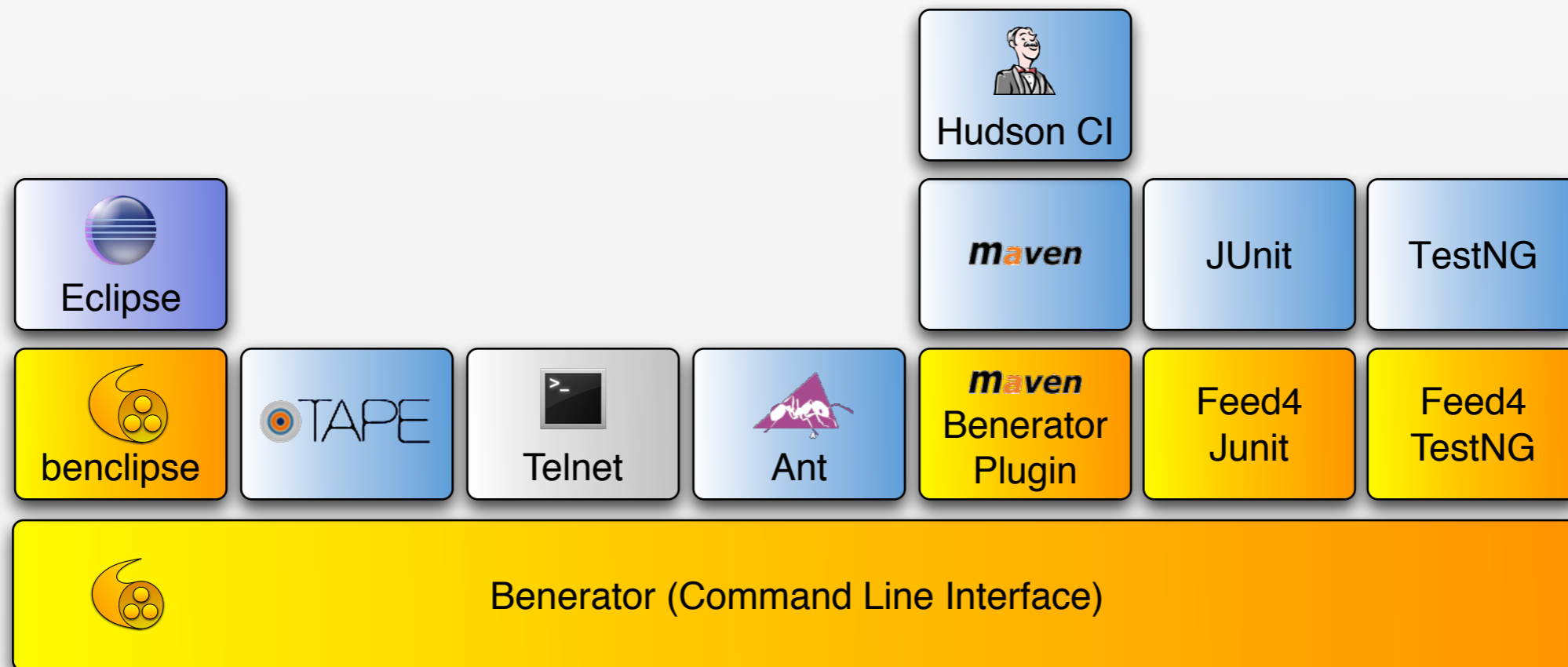
Die Datei id.xml:

Identität der Tabelle COUNTRY... -->
...durch die Spalte NAME -->

Identität der Tabelle STATE... -->
...als Sub-Identität von COUNTRY... -->
...anhand des STATE-Namens -->

```
<dbsanity>  
  
  <identity table="COUNTRY"  
    type="unique-key" columns="NAME" />  
  
  <identity table="STATE" >  
    type="sub-nk-pk-query" parents="COUNTRY">  
      select NAME, ID from STATE  
      where COUNTRY_FK = ?  
    </identity>  
  
</dbsanity>
```

Benerator-Anbindungen



Demo: Spring Pet Clinic

Pet Clinic A Spring Framework Demonstration

Spring

Willkommen


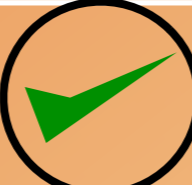
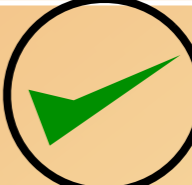















- [Find owner](#)
- [Display all veterinarians](#)
- [Tutorial](#)



[Home](#)

Sponsored by SpringSource 

Datenbezogene Werkzeuge

	Benerator	 Jailer	Altova MapForce
Anonymization			
Datenbank Subsetting			
Replikation			
Datengenerierung			
Zusammenführung			
*ix OS			

Danke für Ihre Aufmerksamkeit



Volker Bergmann
volker@databene.org

...noch Fragen?