

DB Transformation und Migration

Pedro Lafuente Blanco

tarent Solutions GmbH

Agenda

Problemstellung / Anforderungen

Kandidaten

- Liquibase
- flyway

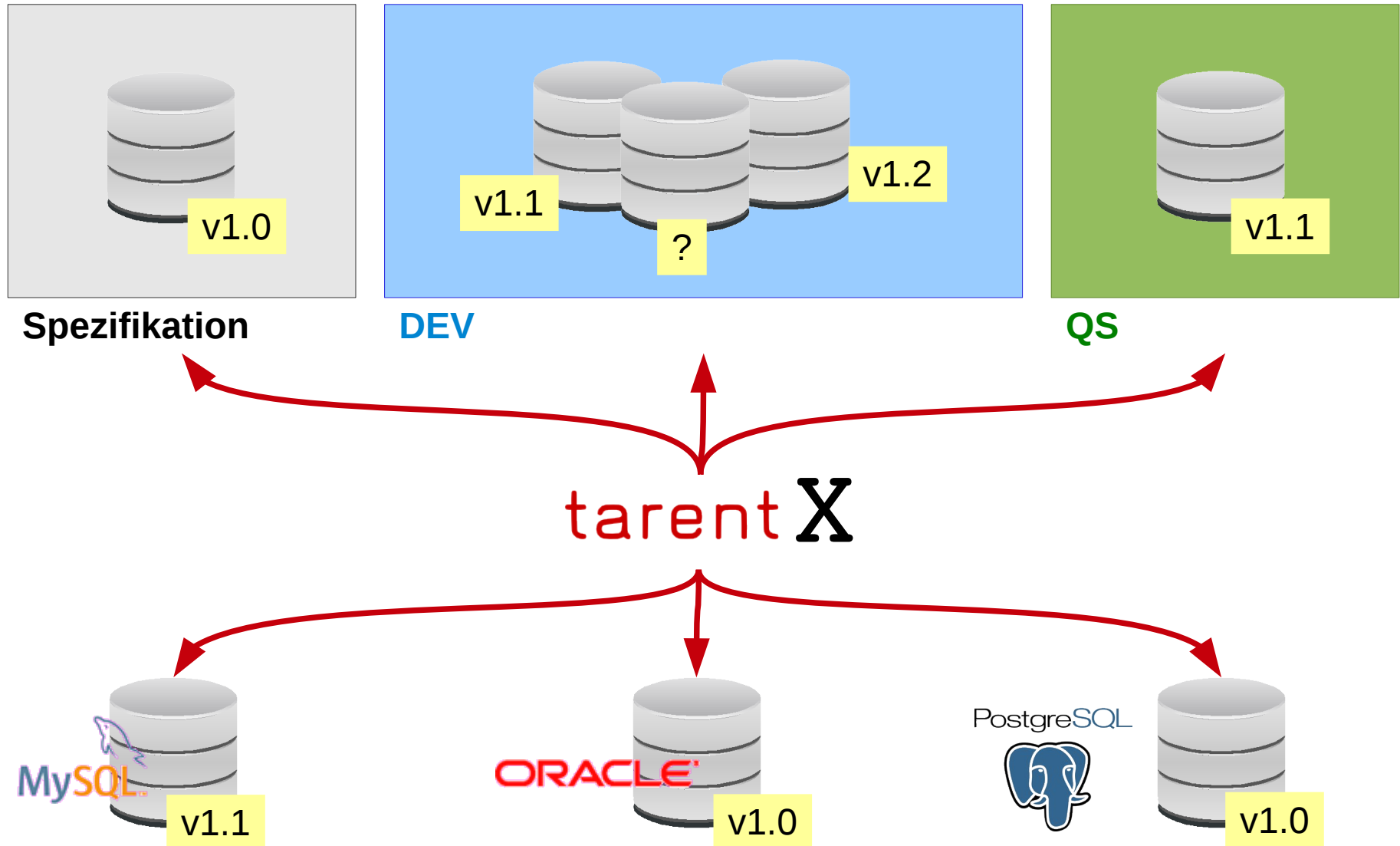
Tests/Use Cases

- ServletListener
- Spring

Empfehlung

Q&A

Problemstellung



Anforderungen

.log

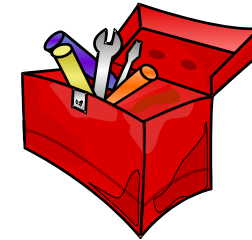
Versionsverwaltung
& Protokolle



Vereinfachung



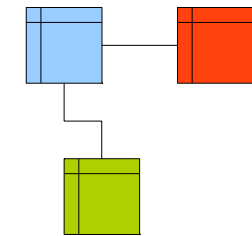
Breite DB
Unterstützung



Integration in
unsere Toolchain



Rollback Unterstützung
(unmittelbar und später)

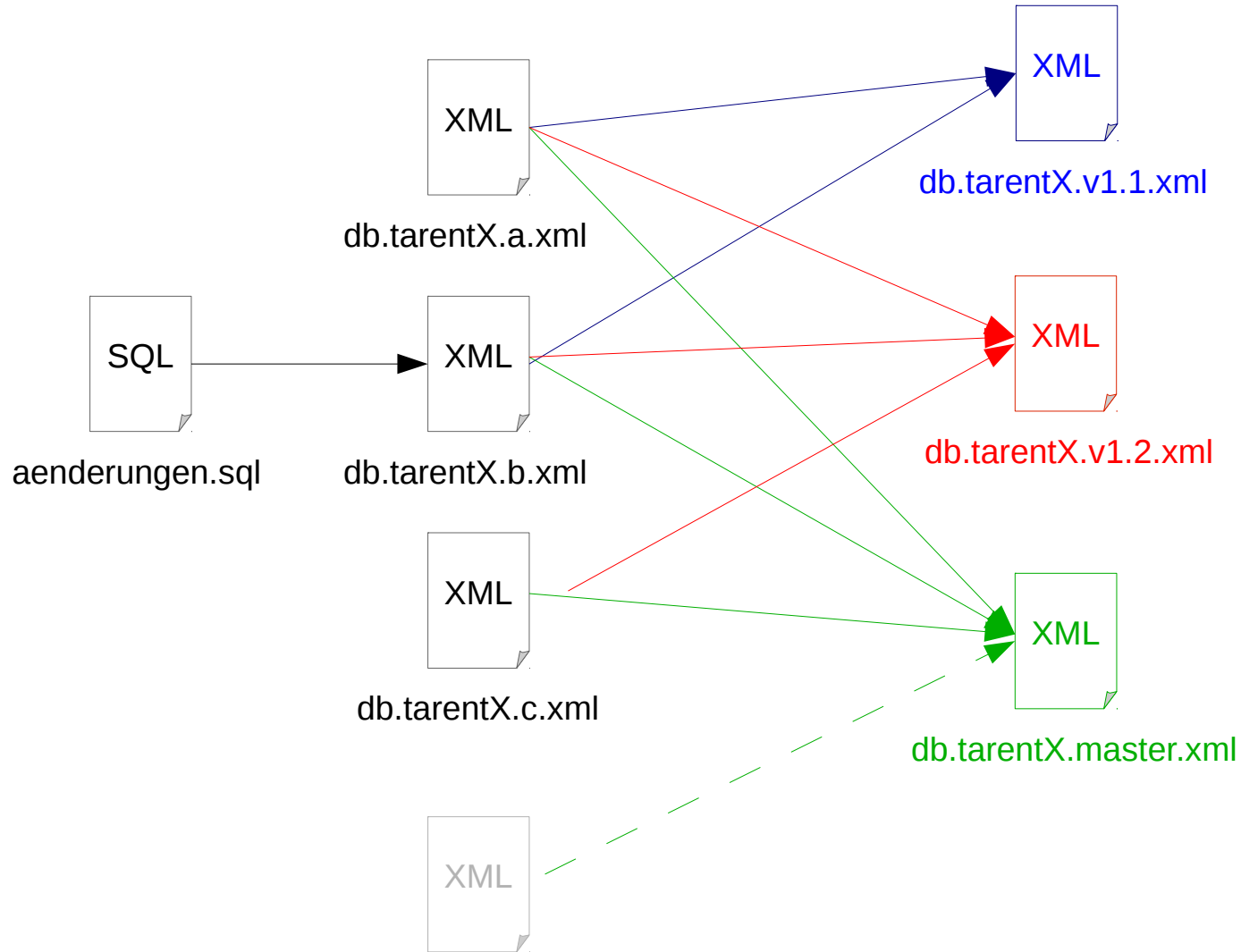


Graphische Modellierung
der Änderungen

Kandidaten

flyway
Tarantino
MIGRATEdb
c5-db-migration dbmaintain
migrate4j DBDeploy
mybatis
AutoPatch
solidbase liquibase

- XML Delta-Skripte (unterstützt auch SQL)
- JDBC-basiert
- Maven-Plugin, CLI, Java API, Spring, Grails
- Erzeugt Rollback-Skripte für viele SQL-Funktionen
- Änderungen und Version in Metadaten-tabelle abgebildet
- ServletListener Unterstützung
- Cluster-Safe
- Erweiterbar



Liquibase XML

```
<?xml version="1.0" encoding="UTF-8" ?>

<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog
    http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-2.0.xsd">

  <changeSet id="20120330-1" author="dbmgr">

    <createTable tableName="dummytable">
      <column name="id" type="int">
        <constraints primaryKey="true" nullable="false"/>
      </column>
      <column name="info" type="varchar(20)">
        <constraints nullable="false"/>
      </column>
      <column name="active" type="boolean" defaultValueBoolean="true"/>
    </createTable>

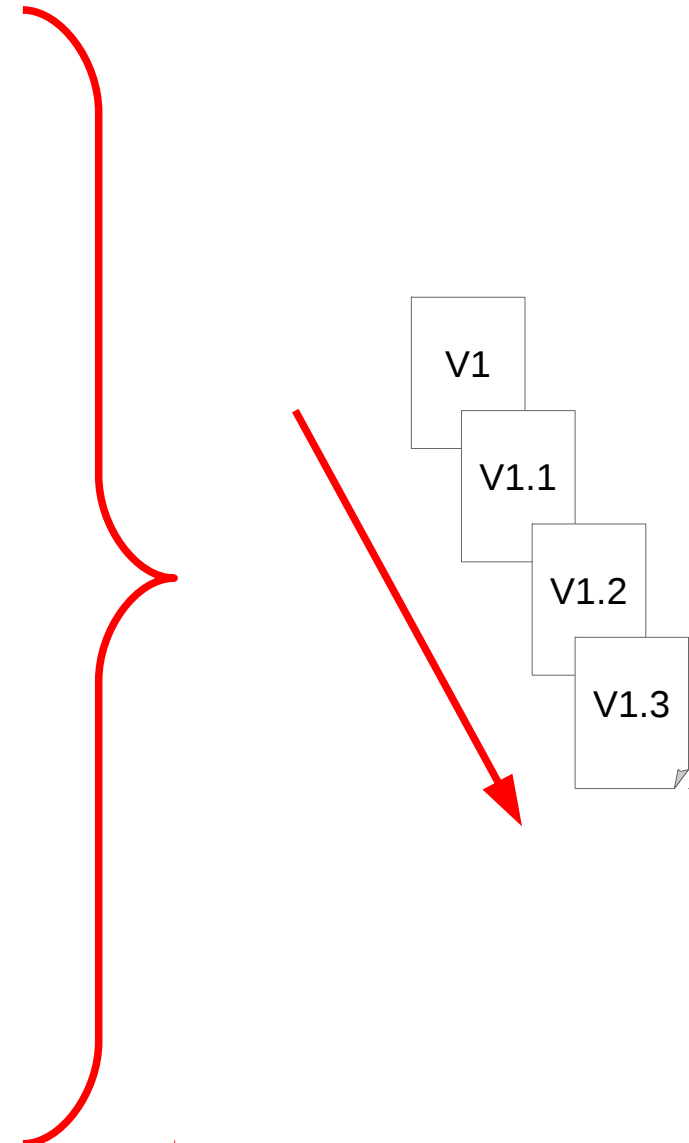
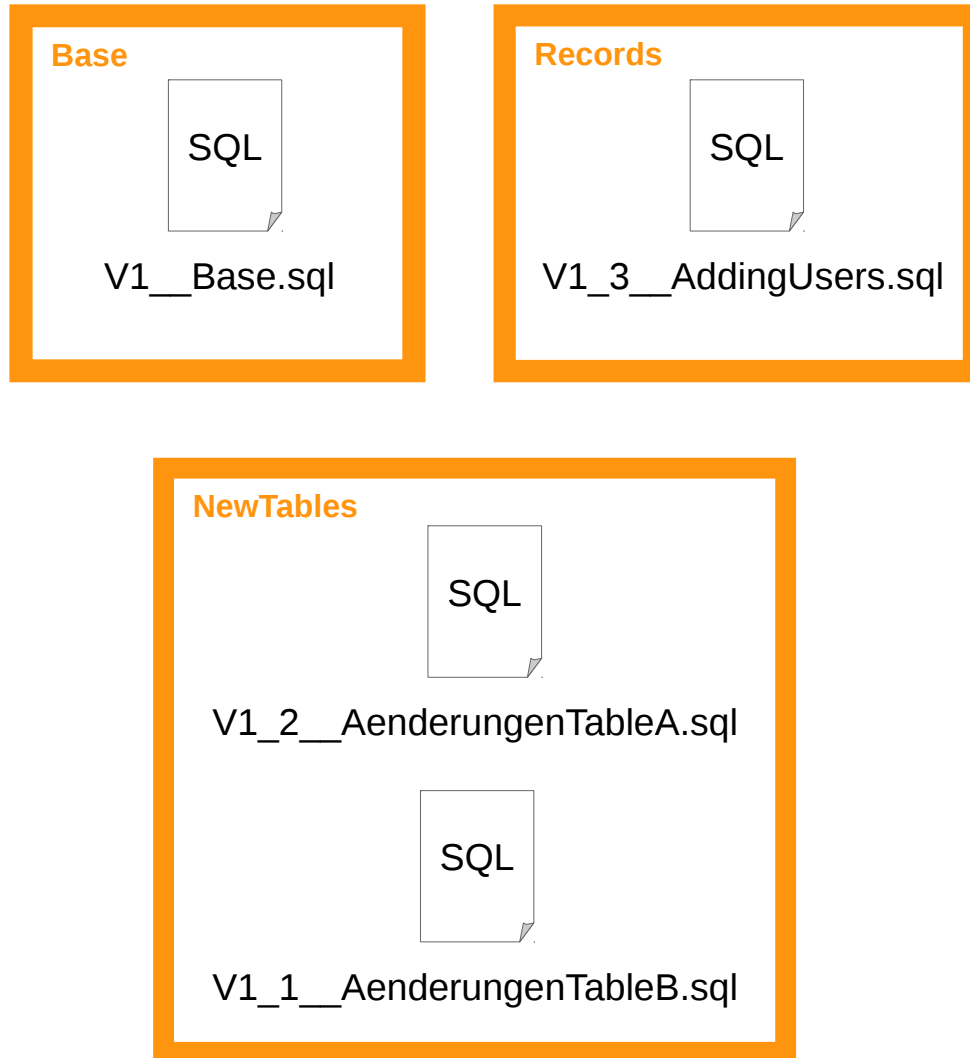
    <insert tableName="dummytable">
      <column name="id" valueNumeric="1"/>
      <column name="info" value="String1"/>
    </insert>
    <insert tableName="dummytable">
      <column name="id" valueNumeric="2"/>
      <column name="info" value="String2"/>
    </insert>
    <insert tableName="dummytable">
      <column name="id" valueNumeric="3"/>
      <column name="info" value="String3"/>
    </insert>

  </changeSet>

</databaseChangeLog>
```




- SQL-basiert
- JDBC-basiert
- Maven-Plugin, CLI, Java API
- Änderungen und Version in Metadattentabelle abgebildet
- Cluster-Safe
- „It just works“



Testphase

Typ	Beschreibung	Kurze B.
1	DB-Snapshot erzeugen	Snapshot Erzeugen
2	Feld employees.testfield (integer) hinzufügen.	Feld hinzufügen
...
5	Feld employees.emp_no entfernen	PK/FK entfernen
...
9	departments.dept_name tmanuel ändern, so dass ein Datensatz die 20char. überschreitet. departments.dept_name von varchar(40) in varchar(20) umwandeln	Datentyp ändern (notok)
...
13	dummytable.number von double in varchar(40) umwandeln	Datentyp ändern
14	Feld dummytable.fkemp hinzufügen, und eine Foreign-Key-Beschränkung auf employee.emp_no festlegen. Dabei Rollback-Script erzeugen. Rollback-Script ausführen.	Rollback

PostgreSQL



ServletListener (externes .war)

web.xml



context.xml



sql/xml + liquibase.jar
im classpath

```
<context-param>
  <param-name>liquibase.changelog</param-name>
  <param-value>de.tarent.tarentx.db-master.xml</param-value>
</context-param>
<context-param>
  <param-name>liquibase.datasource</param-name>
  <param-value>java:comp/env/liquibase/postgresdb</param-value>
</context-param>
<listener>
  <listener-class>
    liquibase.integration.servlet.LiquibaseServletListener
  </listener-class>
</listener>

<Context>
<Resource name="liquibase/postgresdb"
  auth="Container"
  type="javax.sql.DataSource"
  username="dbmgr"
  password="superSecret"
  driverClassName="org.postgresql.Driver"
  url="jdbc:postgresql://dbserver/tarentx"
  maxActive="8"
  maxIdle="4"/>
</Context>
```

```
▼ src
  ▼ resources
    ▼ db
      base.sql
      de.tarent.tarentx.db-master.xml
      de.tarent.tarentx.db-v1.0.xml
      de.tarent.tarentx.db-v1.1.xml
      de.tarent.tarentx.db-v1.2.xml
```



Spring *(im selben .war)*

applicationContext.xml



sql/xml + liquibase.jar
im classpath

```
<bean id="liquibase"  
      class="liquibase.integration.spring.SpringLiquibase">  
  <property name="dataSource" ref="dataSource" />  
  <property name="changeLog"  
            value="classpath:liquibase/db-master.xml" />  
</bean>
```

(dataSource wurde vordefiniert)

▼ src/main/resources
 ▼ liquibase
 base.sql
 de.tarent.tarentx.db-master.xml
 de.tarent.tarentx.db-v1.0.xml
 de.tarent.tarentx.db-v1.1.xml
 de.tarent.tarentx.db-v1.2.xml



SWOT

LIQUIBASE

Strengths

- ServletListener-Unterstützung
- Kein DBMS-spezifisches SQL
- Delta-Dateien können in verschiedenen „dependency trees“ benutzt werden
- Snapshots
- „Future rollback“ Skripte

Weaknesses

- <include>-Tags benötigt für „dependency tree“
- Proprietäres XML
- Fehlerhafte Updates werden nicht in Metadatentabelle abgebildet

- Automatisierung durch Hibernate-Integration
- Versionverzweigung

- Einige Bugs, obwohl Entwicklung ziemlich aktiv ist
- Exportiert nicht alle Objekte:
z.B: triggers, procedures und packages.
- Versioning für verschiedene DBMS mit nur einem Fileset könnte Fehler anfällig sein.

Opportunities

Threats

SWOT



Strengths

- Protokolliert auch nicht erfolgreiche Migrationen
- Flexibles „Version numbering“
- Abhängigkeiten werden automatisch erkannt

Weaknesses

- Keine Möglichkeit, verschiedene „dependency trees“ zu bauen, ohne Dateien zu duplizieren.
- DBMS spezifische SQL !
- Keine „future rollback“ Skripte
- Keine Snapshots

- Narrensichere Implementierung für kleinere Projekte

- Alle Deltas für verschiedene DBMS neuzuschreiben

Opportunities

Threats

Empfehlung



Könnte das Skript-per-DBMS
Problem lösen

Evtl. mehr Arbeit auf Grund des XML

Erleichtert Installation & Initialisierung,
auch bei Servlets

Besser für **größere Projekte**,
wo sich der Verwaltungsaufwand lohnt



Löst nicht das Skript-per-DBMS
Problem

Würde nicht bei Downgrades helfen

Kleiner Fortschritt im Vergleich zum
Status Quo

Besser für **kleinere Projekte**,
die DB-Versionsverwaltung nutzen wollen

Q&A



¡Muchas gracias!

p.lafuente@tarent.de