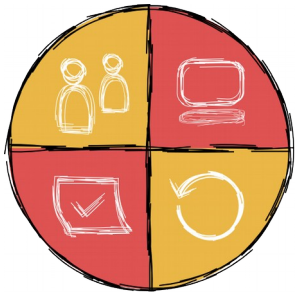
FroScon 2016

Continuous Database Integration mit Flyway

Sandra Parsick
mail@sandra-parsick.de
@SandraParsick

Zur meiner Person

- Freiberufliche Softwareentwickler und Consultant im Java-Umfeld
- Schwerpunkte:
  - Java Enterprise Anwendungen
  - Agile Methoden
  - Software Craftmanship
  - Automatisierung von Entwicklungsprozessen
- Softwerkskammer Dortmund



Agenda

- Continuous Database Integration (CDBI)
- Flyway
- Flyway Demo
- Fallstricke

Continuous Database Integration

- Definition
- Motivation
- Aufbau

Definition

„Continuous Database Integration (CDBI) is the process of rebuilding your database and test data any time a change is applied to a project's version control repository“

(aus Continuous Integration by Paul M. Duvall, Steve Matyas und Andrew Glover)

Motivation

- Alle Entwickler teilen sich eine Testdatenbank.
- Keiner weiß, welche Datenbankskripte auf welchen Datenbankinstanzen ausgeführt worden.
- Testdatenbank unterscheidet sich von der Produktionsdatenbank.
- Datenbankmigrationsskripte verteilen sich auf Emails, Release Notes, Ticketsysteme, etc.

Aufbau

- Behandle den Datenbank-Code wie einen ganz normalen Source-Code
  - ➔ Alle Datenbank Artefakte (DDL, DML, Konfigurationen, Testdaten, Stored Procedures, Functions etc) gehören ins VCS.
  - ➔ Jede Änderung an den DB Artefakten wird getestet.
- Jeder Entwickler hat seine eigene Datenbank / Testdatenbanken ähneln den Produktionsdatenbanken.
  - ➔ Automatisiertes Aufsetzen der Datenbank.
- Änderungen an der Datenbank sind nachvollziehbar.
  - ➔ Historie der Änderungen

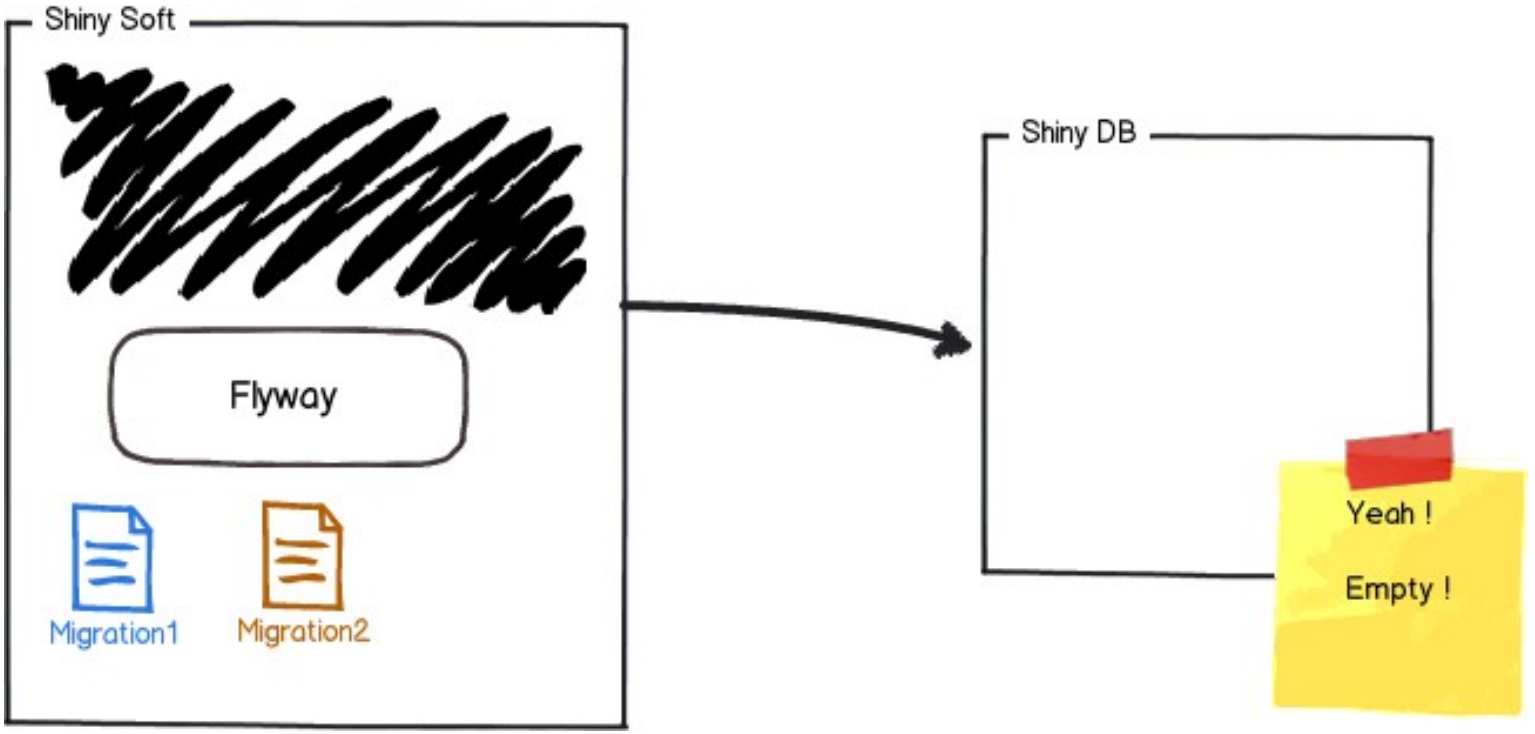
Flyway

- Was ist Flyway?
- Wie funktioniert Flyway?
- Wie werden Migrationsskripte für Flyway geschrieben?
- Was kann Flyway nicht?
- Wie kann Flyway benutzt werden?

Was ist Flyway?

- Migration Framework für Relationale Datenbanken basierend auf Java
- Erstellt eine Datenbank „from scratch“
- Verwaltet den Stand der Datenbank
- Zwei Migrationsmodi:
    - SQL Migration
    - Java Migration
- Aktuelle Version: 4.0.3
- Homepage: http://flywaydb.org/
- Twitter: @flywaydb

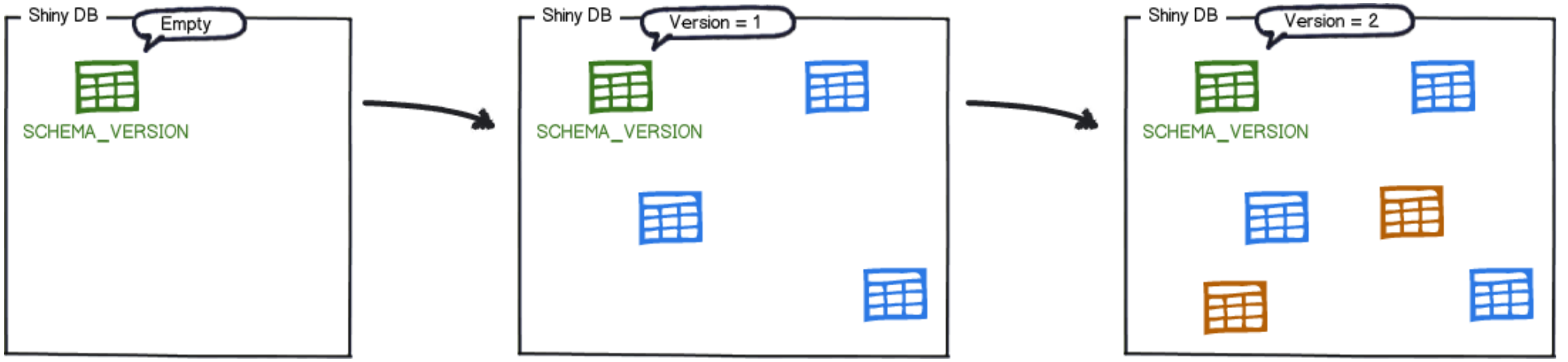
Wie funktioniert Flyway?



Reference: flywaydb.org

Wie funktioniert Flyway?

migrate



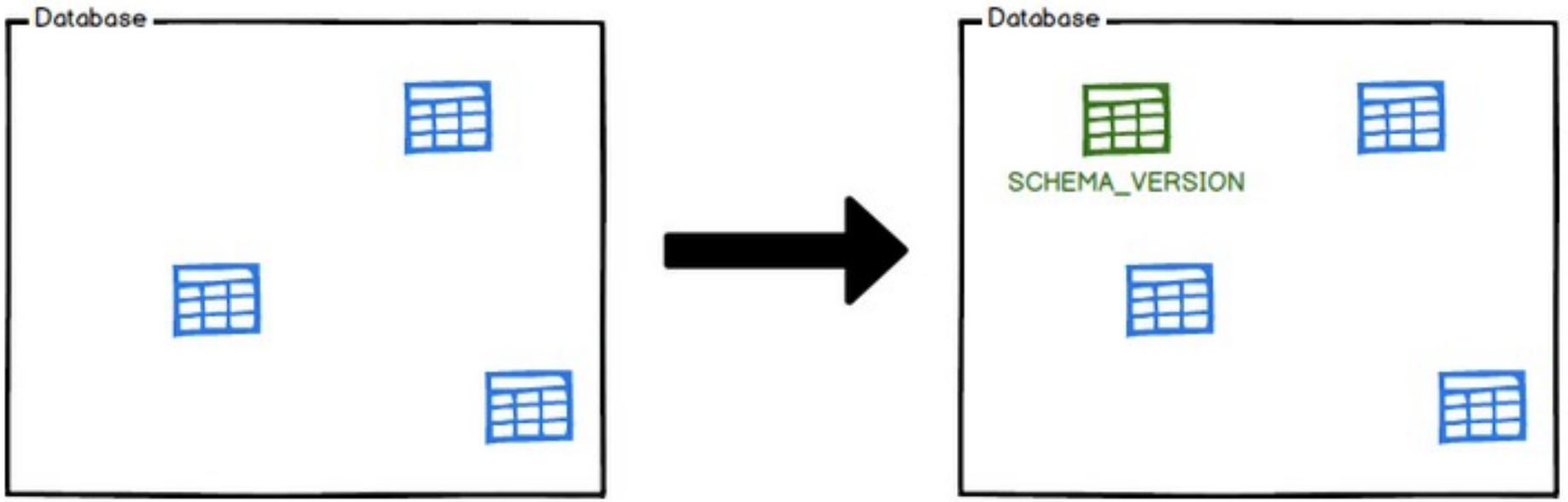
Reference: flywaydb.org

schema_version

installed_rank	version	description	type	script	checksum	installed_by	installed_on	execution_time	success
1	1	Initial Setup	SQL	V1__Initial_Setup.sql	1996767037	axel	2016-02-04 22:23:00.0	546	true
2	2	First Changes	SQL	V2__First_Changes.sql	1279644856	axel	2016-02-06 09:18:00.0	127	true

Wie funktioniert Flyway?

baseline



Reference: flywaydb.org

Migrationsskripte

- Vier Möglichkeiten

	Versioniert	Wiederholbar
SQL-basiert	✓	✓
Java-basiert	✓	✓

Versionierte Migration

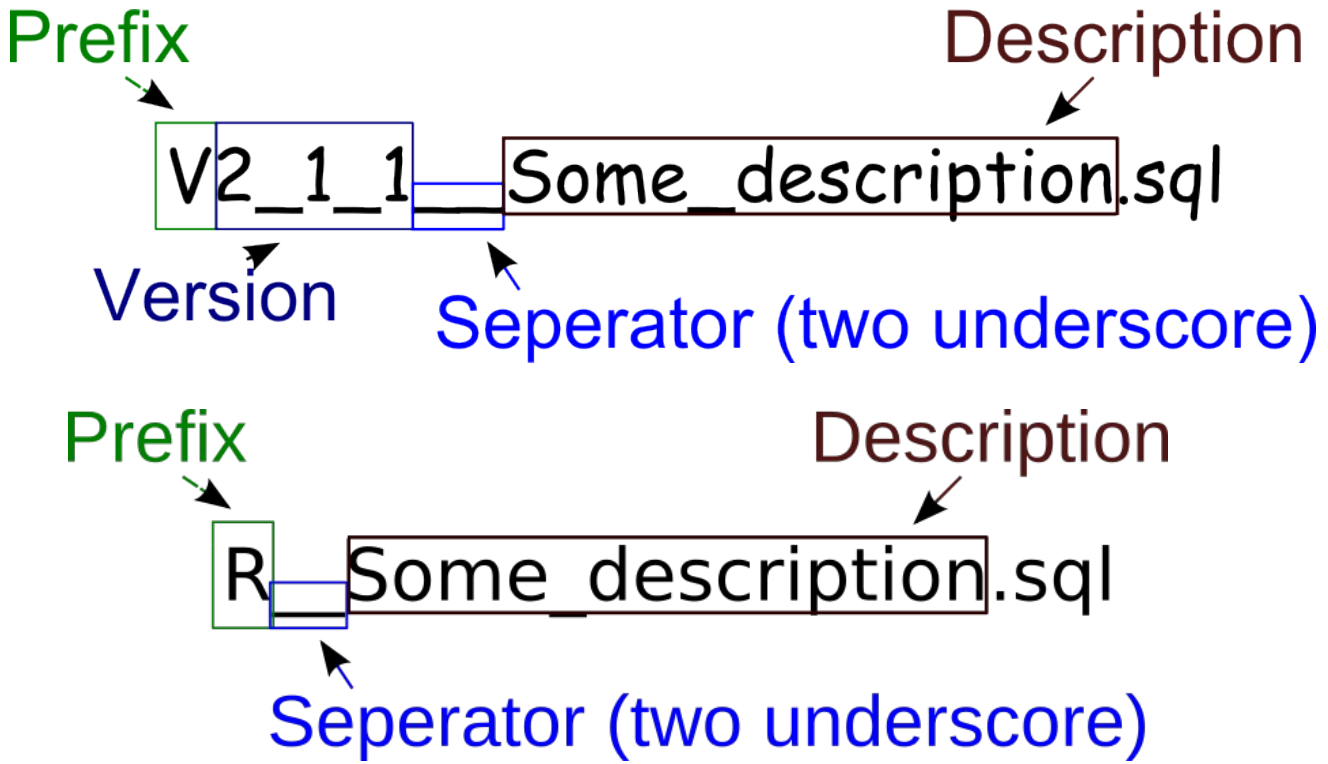
- **Eigenschaften**
  - Skripte haben eine eindeutige Version
  - Werden genau einmal ausgeführt
- **Typische Anwendungsfälle**
  - DDL Änderungen (CREATE/ALTER/DROP für TABLES,INDEXES,FOREIGN KEYS,...)
  - Einfache Datenänderungen

Wiederholbare Migration

- **Eigenschaften**
  - Skripte haben keine Versionsnummer
  - Werden immer dann ausgeführt, wenn sich ihre Checksumme ändert
  - Werden immer dann ausgeführt, nachdem alle versionierte Skripte ausgeführt wurden
- **Typische Anwendungsfälle**
  - (Wieder-) Erstellung von views / procedures / functions / packages / ...
  - Massenreimport von Stammdaten

SQL Migration

- **Typische Anwendungsfälle**
  - DDL Änderungen (CREATE/ALTER/DROP für TABLES,VIEWS,TRIGGERS,SEQUENCES,...)
  - Einfache Datenänderungen
- **Benamung der Skripte**



SQL Migration

- **Syntax**
  - Statement kann über mehrere Zeile gehen
  - Platzhaltersupport
  - Kommentare: Single (–) oder Multi-Line (/* */)
  - Datenbank-spezifische SQL Syntax
- **Beispiel**

```
1   /* Create a table for person */
2
3   Create table person (
4       first_name varchar(128),
5       last_name varchar(128)
6   );
```

Unterstützte Datenbanken

Oracle
10g and later, all editions
(incl. Amazon RDS)

SQL Server
2008 and later
(incl. Amazon RDS)

SQL Azure
latest

MySQL
5.1 and later
(incl. Amazon RDS & Google Cloud SQL)

MariaDB
10.0 and later
(incl. Amazon RDS)

Phoenix
4.2.2 and later

PostgreSQL
9.0 and later
(incl. Heroku & Amazon RDS)

Vertica
6.5 and later

AWS Redshift
latest

DB2
9.7 and later

DB2 z/OS
9.1 and later

Derby
10.8.2.2 and later

H2
1.2.137 and later

Hsql
1.8 and later

SQLite
3.7.2 and later

SAP HANA
latest

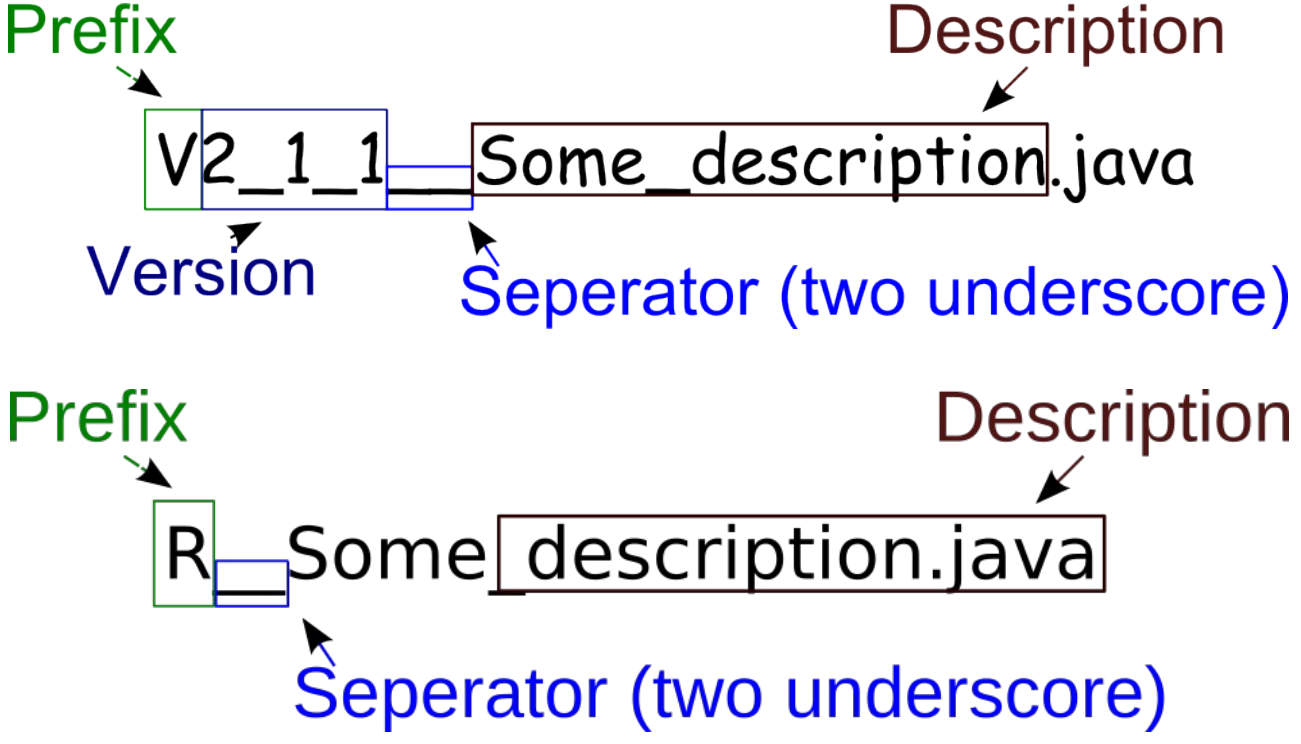
solidDB
6.5 and later

Sybase ASE
12.5 and later

Reference: flywaydb.org

Java Migration

- **Typische Anwendungsfälle**
  - BLOB & CLOB Änderungen
  - Fortgeschrittene Änderungen an Massendaten (Neuberechnungen, fortgeschrittene Formatsänderungen, …)
- **Benamung der Java Klassen**



Java Migration

Beispiel

```java
package db.migration;

import java.sql.Connection;
import java.sql.Statement;
import org.flywaydb.core.api.migration.jdbc.JdbcMigration;


public class V1_1_0__Insert_Data implements JdbcMigration {

    @Override
    public void migrate(Connection connection) throws Exception {
        try (Statement statement = connection.createStatement()) {
            statement.execute("Insert into person (first_name, last_name) Values ('Alice', 'Bob')");
        }

    }

}

```

Java Migration

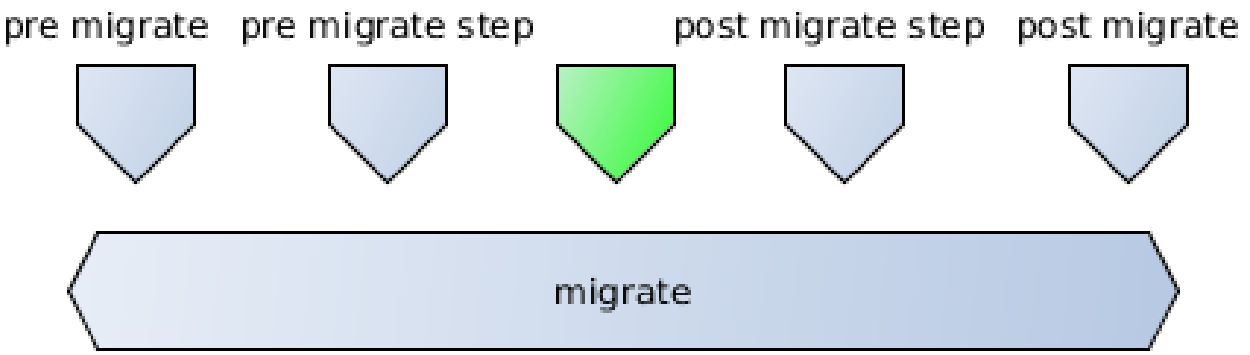
Beispiel Spring Support

```java
package db.migration;

import org.flywaydb.core.api.migration.spring.SpringJdbcMigration;
import org.springframework.jdbc.core.JdbcTemplate;


public class V1_2_0__Create_Table_With_Spring_Support implements SpringJdbcMigration {

    @Override
    public void migrate(JdbcTemplate jdbcTemplate) throws Exception {
       jdbcTemplate.execute("Create table address (street Varchar(128), place Varchar(128))");
    }

}

```

Migration für Fortgeschrittene - Callbacks

- **Typische Anwendungsfälle**
  - Stored Procedure Kompilierung
  - Materialized View Update

- **Flyway Lifecycle (Beispiel migrate)**



SQL Callbacks

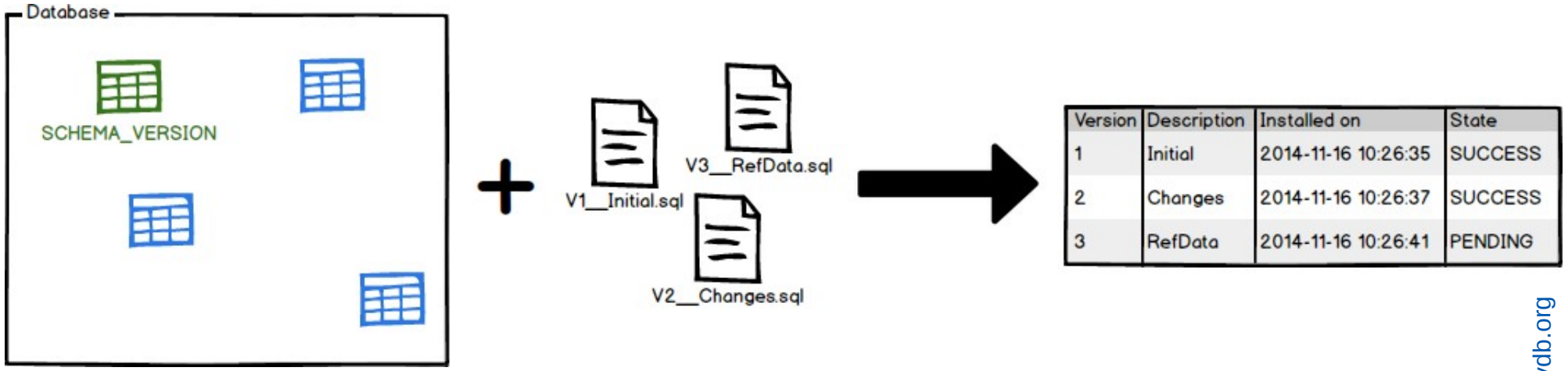
- **Beispiel migrate-Lifecycle:**
  - SQL Callback Skripte werden anhand deren Namen erkannt:
    - BeforeMigrate.sql
    - BeforeEachMigrate.sql
    - AfterEachMigrate.sql
    - AfterMigrate.sql

Java Callbacks

```
public interface FlywayCallback {
    /**
     * Runs before the clean task executes.
     *
     * @param connection A valid connection to the database.
     */
    void beforeClean(Connection connection);

    /**
     * Runs after the clean task executes.
     *
     * @param connection A valid connection to the database.
     */
    void afterClean(Connection connection);

    /**
     * Runs before the migrate task executes.
     *
     * @param connection A valid connection to the database.
     */
    void beforeMigrate(Connection connection);
```

Weitere Flyway Befehle

info

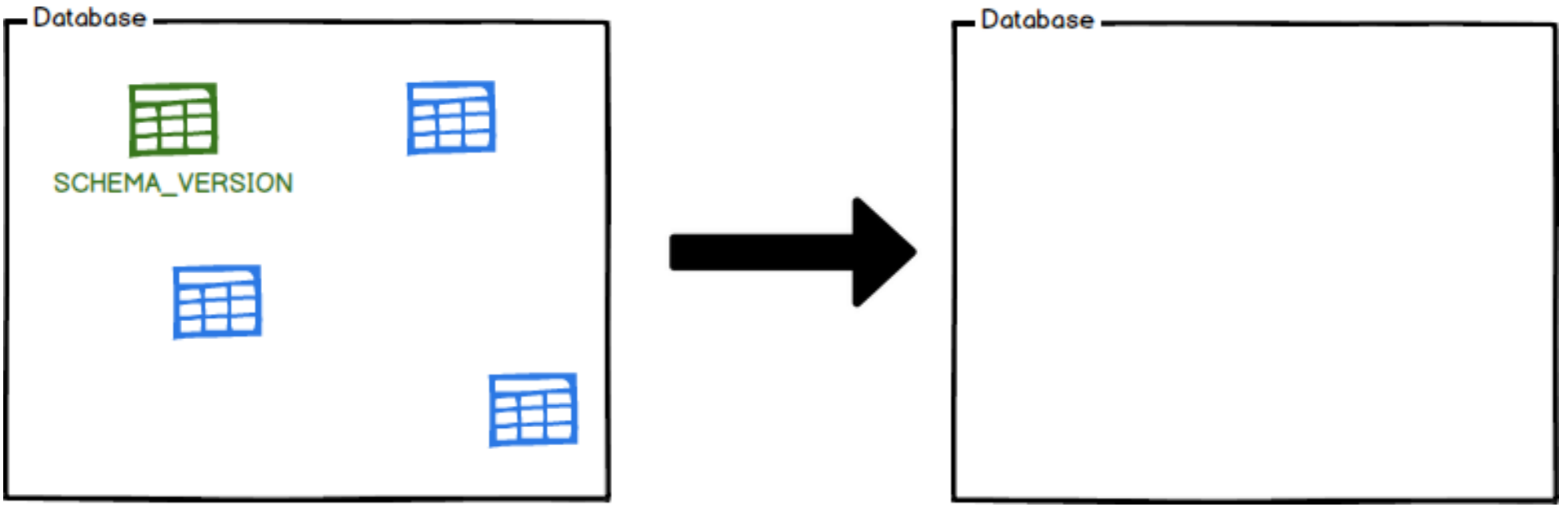


Version	Description	Installed on	State
1	Initial	2014-11-16 10:26:35	SUCCESS
2	Changes	2014-11-16 10:26:37	SUCCESS
3	RefData	2014-11-16 10:26:41	PENDING

Reference: flywaydb.org

Weitere Flyway Befehle

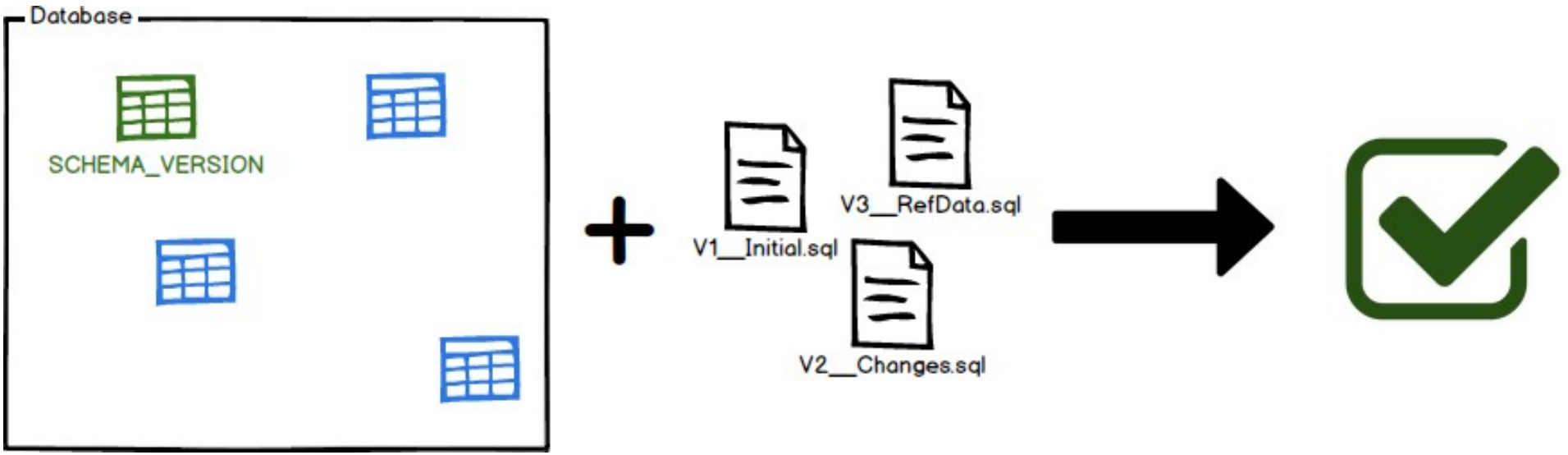
clean



Reference: flywaydb.org

Weitere Flyway Befehle

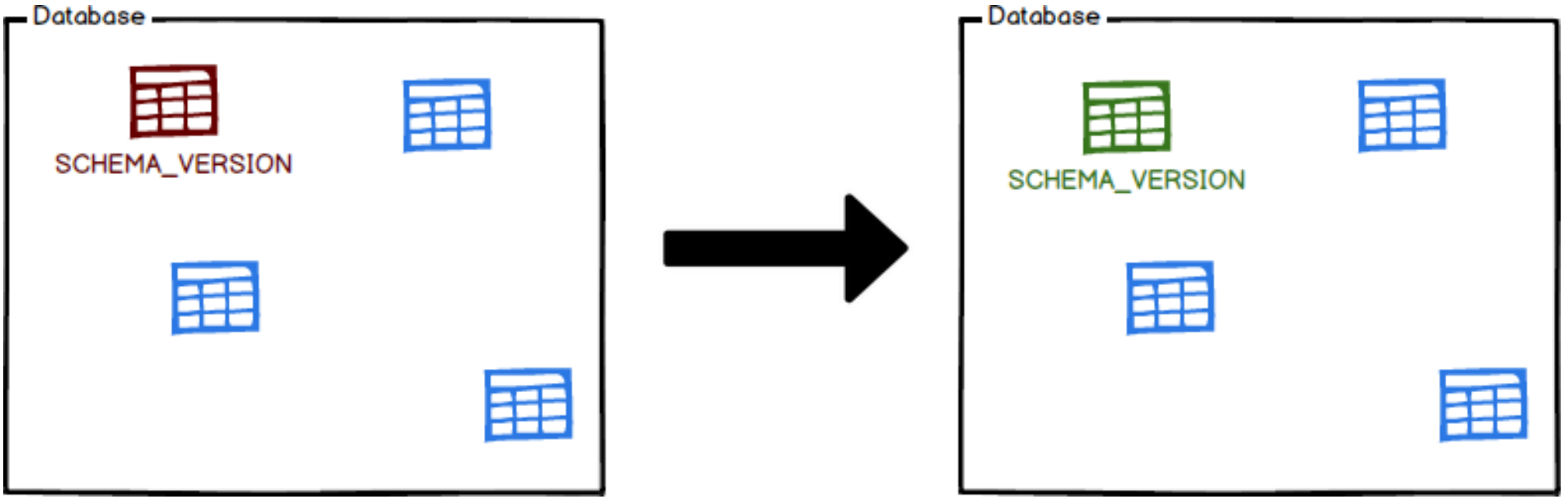
validate



Reference: flywaydb.org

Weitere Flyway Befehle

repair



Reference: flywaydb.org

Was kann Flyway nicht?

- Rollback Skripte aufrufen
- „Write once, run on many database vendors“

Wie kann Flyway benutzt werden?

- **Flyway Clients:**
  - Java API
  - Maven Plugin
  - Command-line Tool
  - Gradle Plugin
  - SBT Plugin
  - Ant task

Demo

Aufbau CDBI

- Behandle den Datenbank-Code wie einen ganz normalen Source-Code
  - ➔ Alle Datenbank Artefakte (DDL, DML, Konfigurationen, Testdaten, Stored Procedures, Functions etc) gehören ins VCS. 
  - ➔ Jede Änderung an den DB Artefakten wird getestet. 
- Jeder Entwickler hat seine eigene Datenbank / Testdatenbanken ähneln den Produktionsdatenbanken.
  - ➔ Automatisiertes Aufsetzen der Datenbank.
- Änderungen an der Datenbank sind nachvollziehbar.
  - ➔ Historie der Änderungen 

Fallstricke

Keine Instanz-spezifischen Daten

Beispiel

```
1
2   GRANT SELECT, INSERT ON usermgm.* TO
3   `technical-user`@'192.168.33.10' IDENTIFIED BY 'pA$$w0rt';
4
```

Keine Instanz-spezifischen Daten

Möglicher Lösungsansatz:

```
1
2     GRANT SELECT, INSERT ON usermgm.* TO
3     `technical-user`@'*' IDENTIFIED BY 'pA$$w0rt';
4
```

- Zugriffskontrolle über eine Firewalls (iptables)

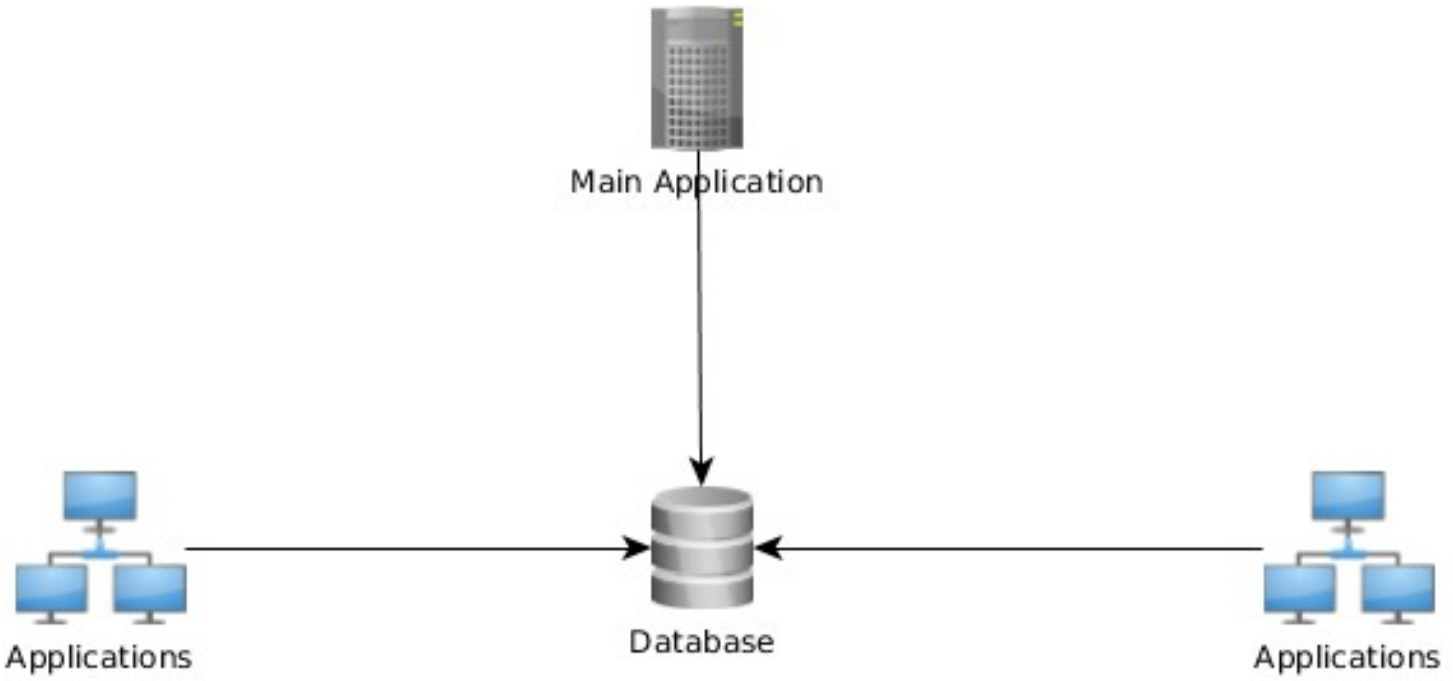
Keine Instanz-spezifischen Daten

Möglicher Lösungsansatz:

```
1    GRANT SELECT, INSERT ON usermgnt.* TO
2    'technical-user' @ '${address}' By '${password}';
3
4
```

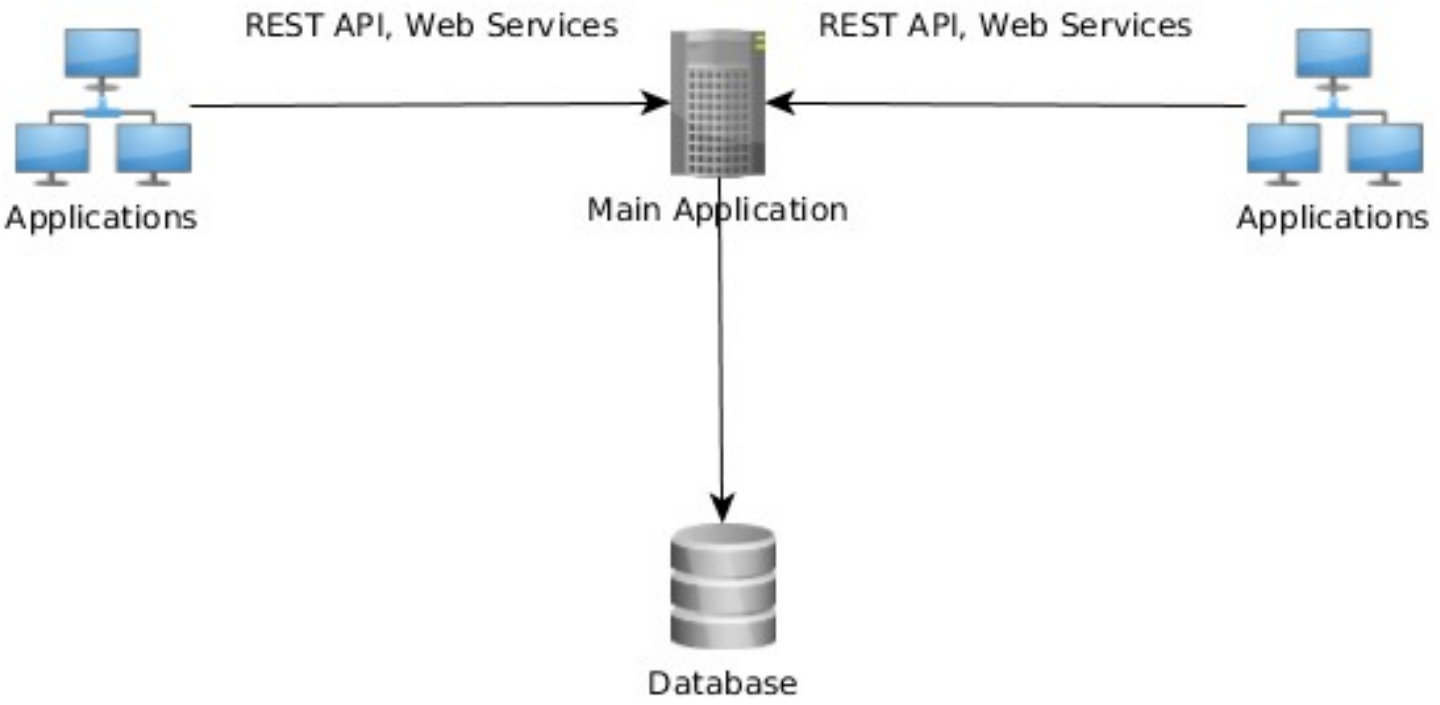
Datenbank wird von mehreren Applikationen benutzt

Ausgangslage :



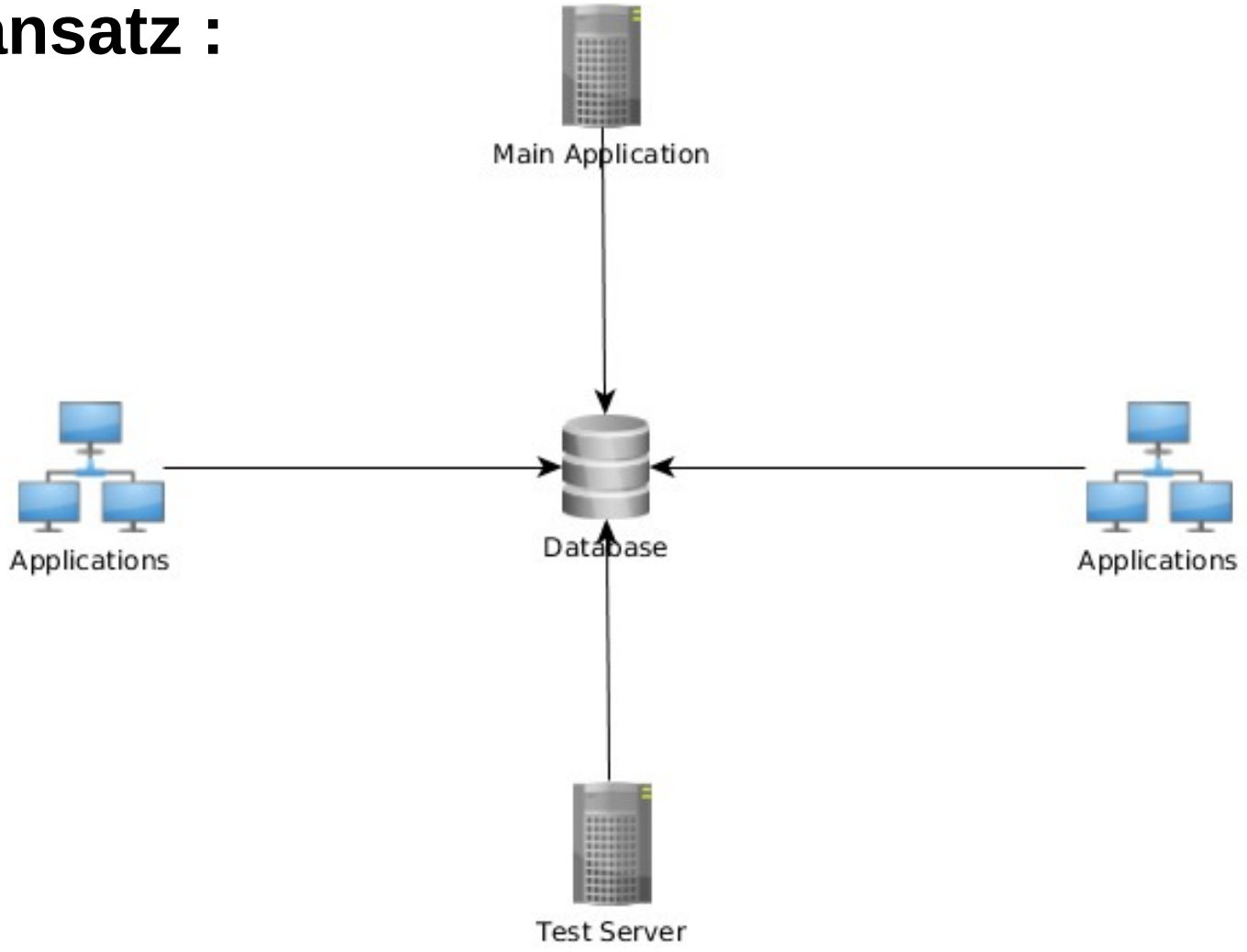
Datenbank wird von mehreren Applikationen benutzt

Lösungsansatz :

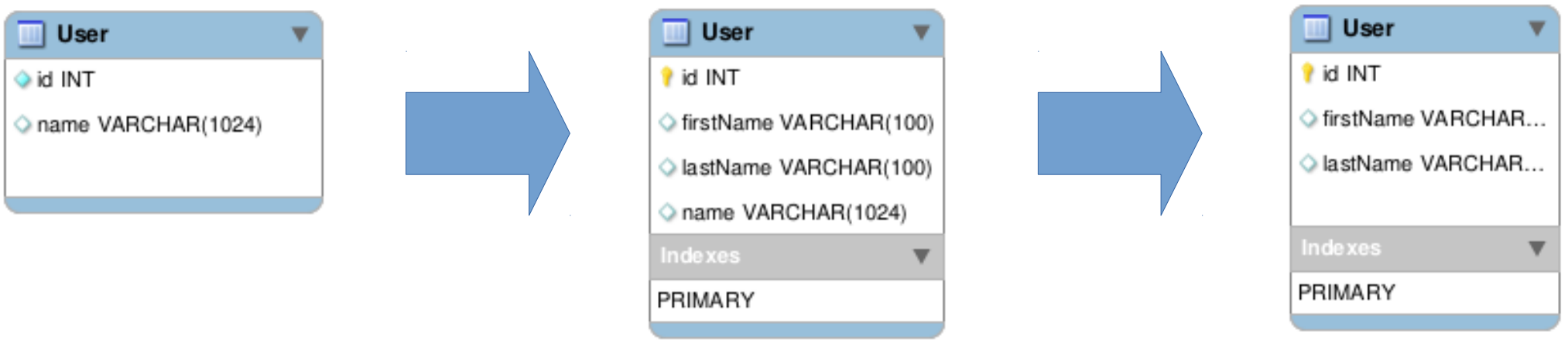


Datenbank wird von mehreren Applikationen benutzt

Lösungsansatz :



Datenbank wird von mehreren Applikationen benutzt



Weitere Fallstricke (Auszug)

- Datenänderung dauern zu lange
- Datenlöschung
- Faktor Mensch
- ...

Weitere Informationen

- Continuous Integration von Paul M. Duvall, Steve Matyas und Andrew Glover
- Refactoring Databases: Evolutionary Database Design von Scott J. Ambler und Pramodkumar J. Sadalage
- Flyway Documentation
  http://flywaydb.org/documentation/migration/
  http://flywaydb.org/getstarted/

Fragen?

mail@sandra-parsick.de
@SandraParsick