



Architektur/QS Board – 20. Mai 2015

openK platform für openKONSEQUENZ

Jan Krüger BTC AG

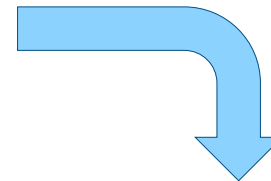
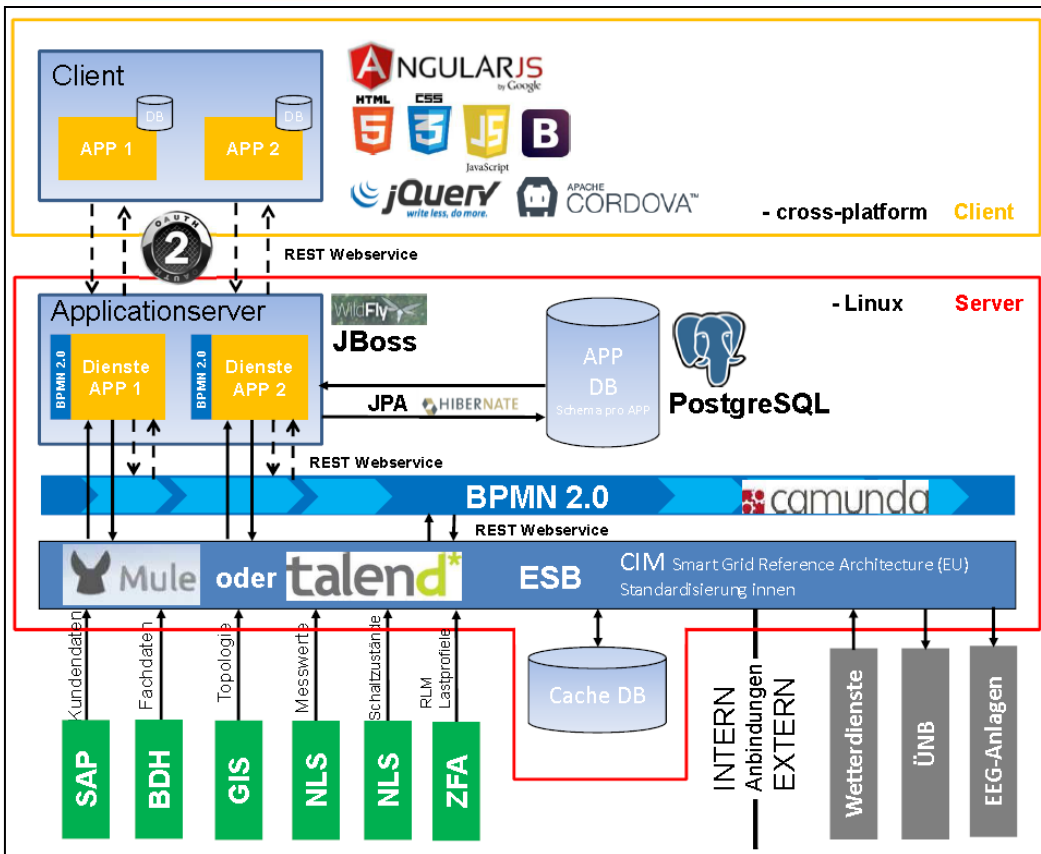
Seite 1



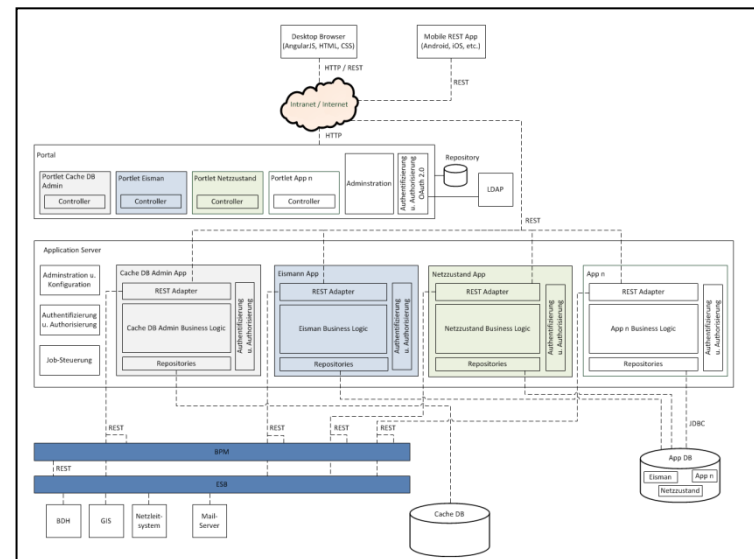
Agenda

1. Vorstellung Architektur – was hat sich getan?
2. Mandantenfähigkeit
3. Aufgaben Portal
4. CIM
5. UIs: Vorgehen und Implementierung
6. Build Process als Maven Projekt
7. Fragen & Antworten

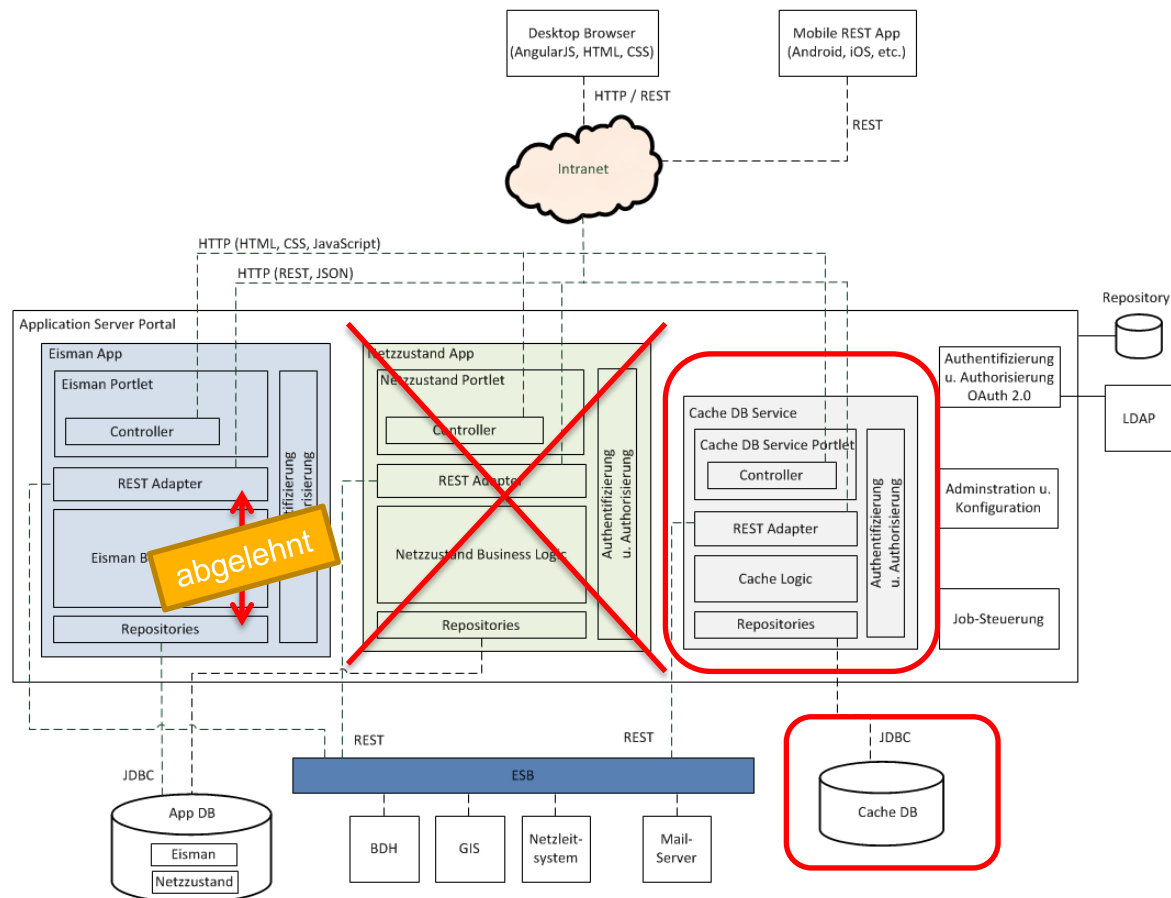
openKONSEQUENZ Architekturentwurf



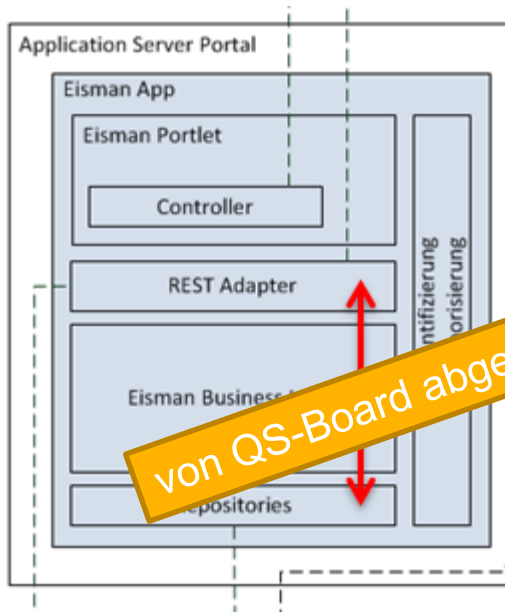
BTC Angebot



IT Architektur – was hat sich getan?



Anpassungen Architektur direkte Aufrufe REST-Adapter zu Repository



Aufruf eines Repository direkt aus dem REST Adapter unter „Umgehung“ der Business Logic Schicht.

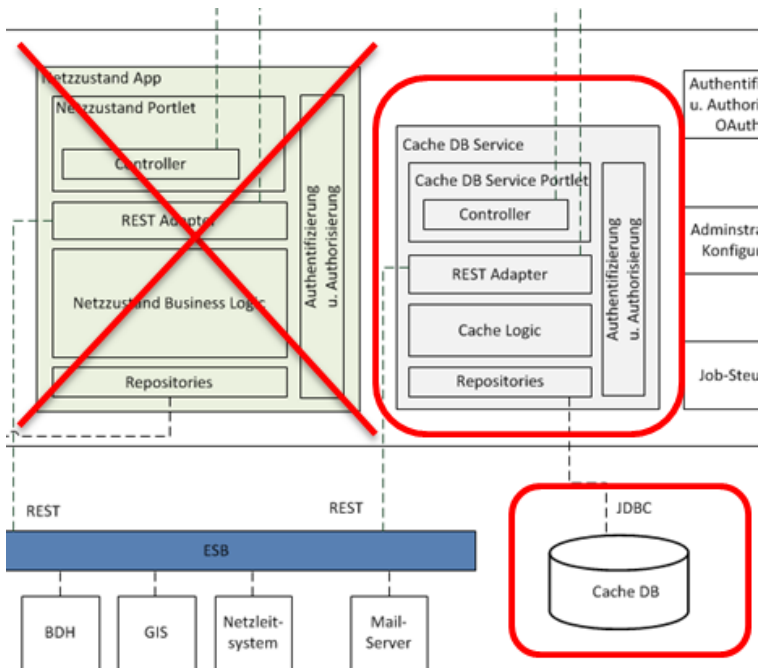
- REST Services ohne Business Logic rufen direkt ein Repository auf, um Daten in der Datenbank „direkt“ zu lesen oder zu schreiben.
 - keine Business Logic
- Ein Service in der Business Logic würde den Aufruf nur „durch-routen“



REST-Adapter können auch direkt ein Repository aufrufen

Anpassungen Architektur

Notwendigkeit Netzstatus und Funktionserweiterung Cache DB



Aufgaben Cache DB

- Kompensation der geringeren Verfügbarkeiten der Dritt-Systeme
- Erhöhung der Performance bzgl. Datenabfragen der Dritt-Systeme

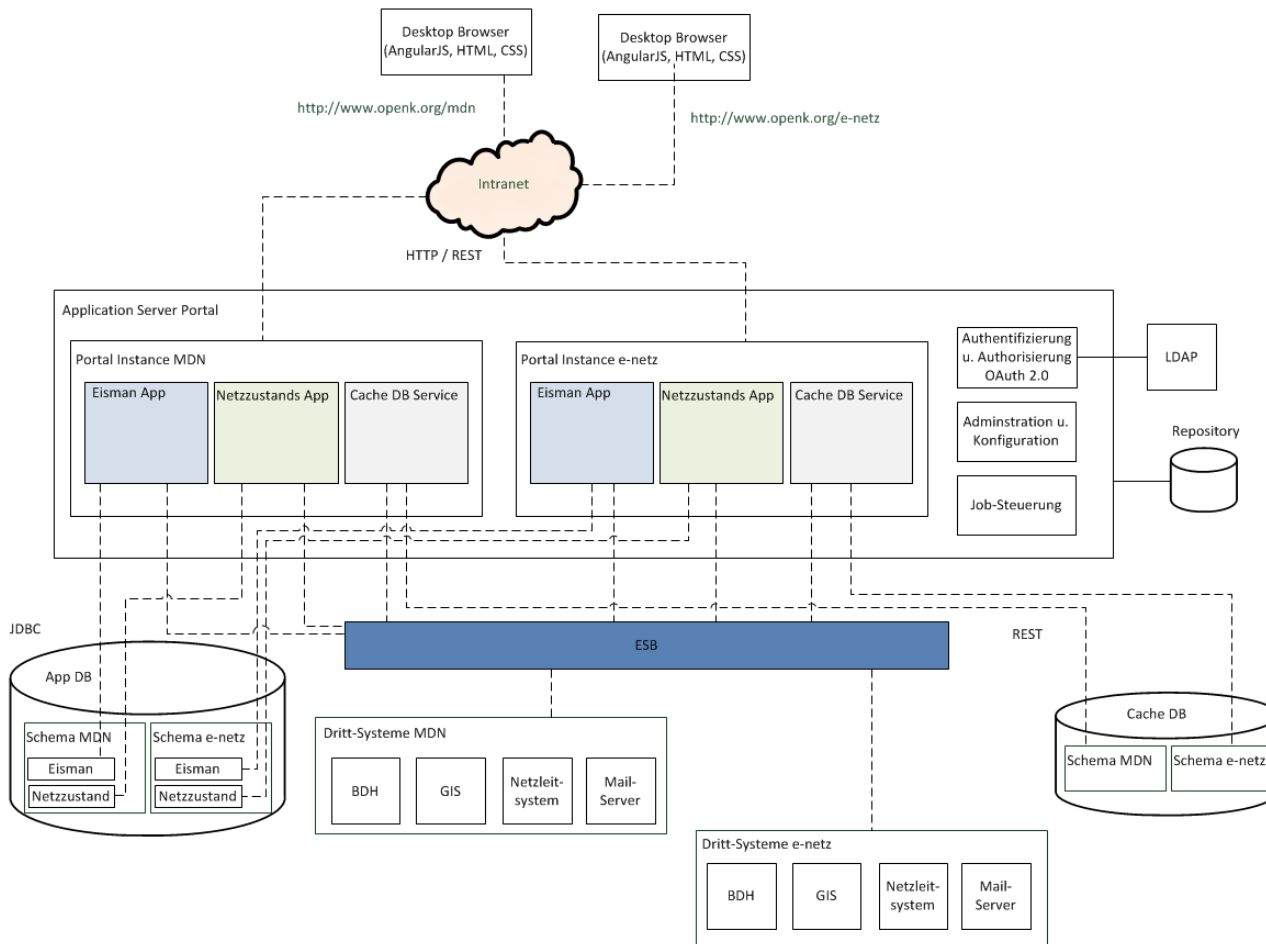
Cache DB beinhaltet mit CIM ein integriertes Datenmodell, das verschiedene Sichten auf die Daten ermöglicht

- Topologie
- Messwerte/Istwerte
- Stammdaten



Cache DB Service übernimmt Aufgaben der Netzzustand App und ist damit deutlich mehr als ein Cache

Mandantenfähigkeit mittels Portaleigenschaften



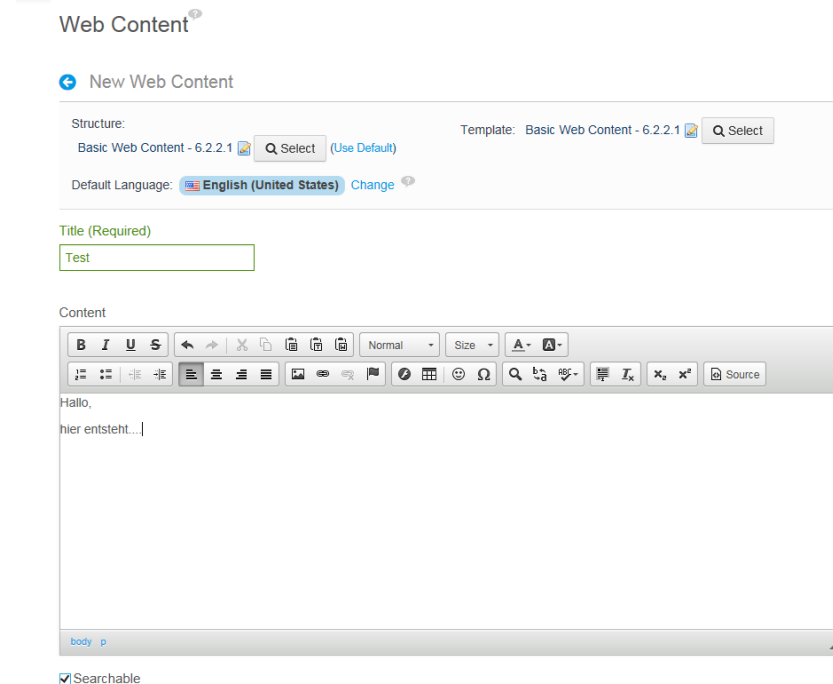
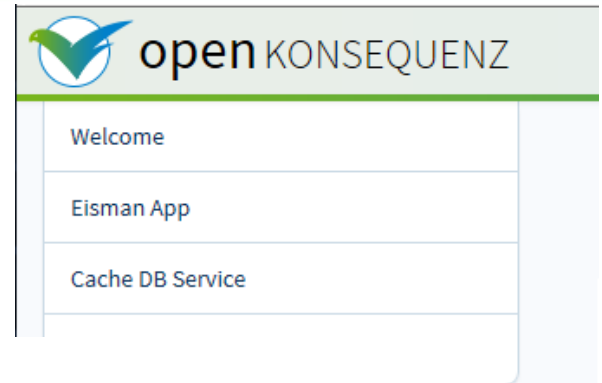
Erstellung einer
2. Portal-Instanz mit
einer „eigenen“ URL
Gleiche Portlets in jeder
Portal-Instanz

Portlet-Instanzen
verwenden eigene DB-
Schemas

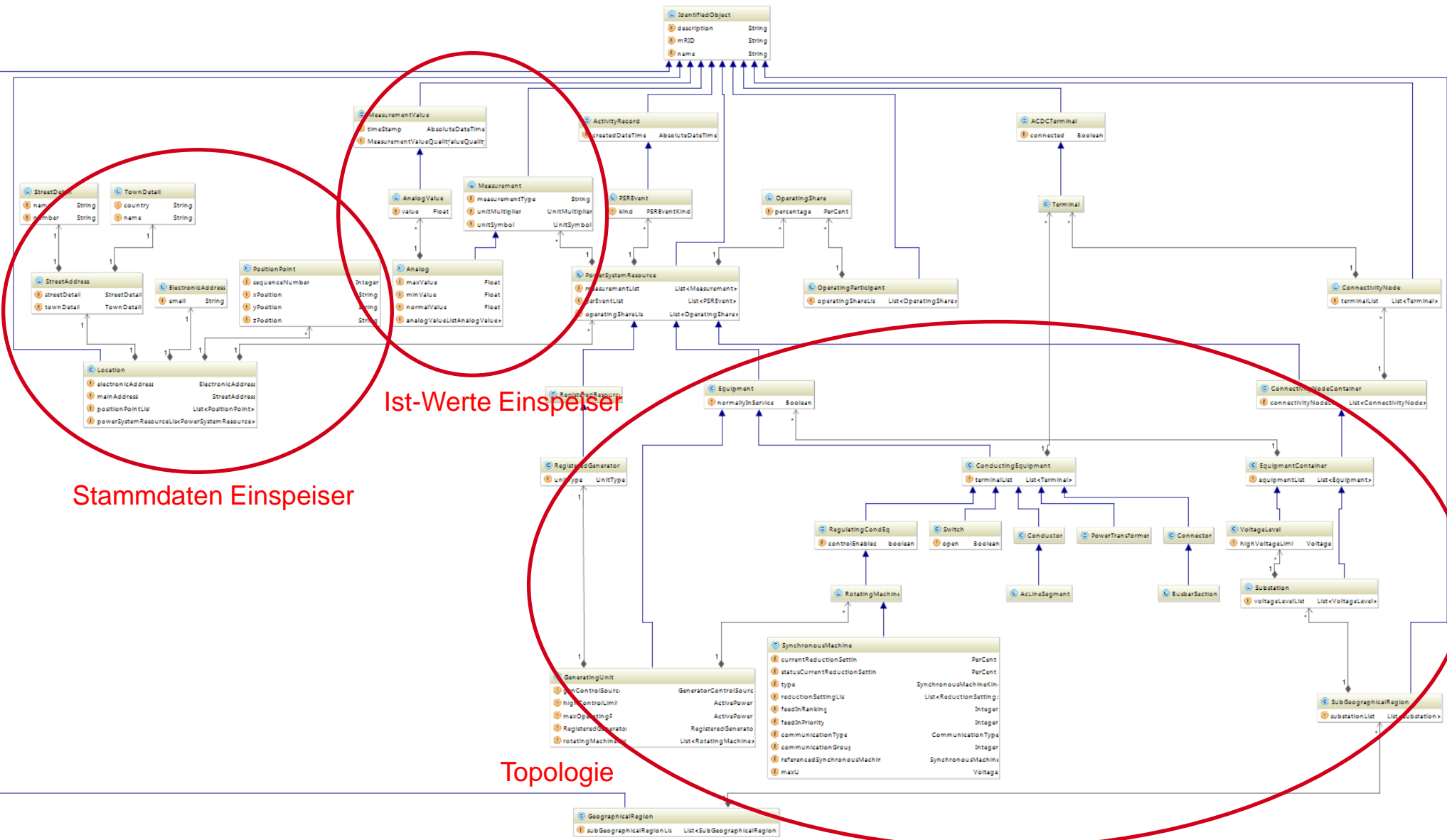
Ein ESB der verschiedene
Endpoints für die
Portlet-Instanzen
anbietet

Aufgaben Portal

- Dynamisches Menü, das mit Portalmitteln administriert wird
- Mandantenfähigkeit mittels Portal-Instanzen
- Mail-Template mit Liferay-CMS erstellen
- Autorisierung & Authentifizierung
- Verwendung „wie ein Application Server“



CIM – Modellausschnitt openk platform

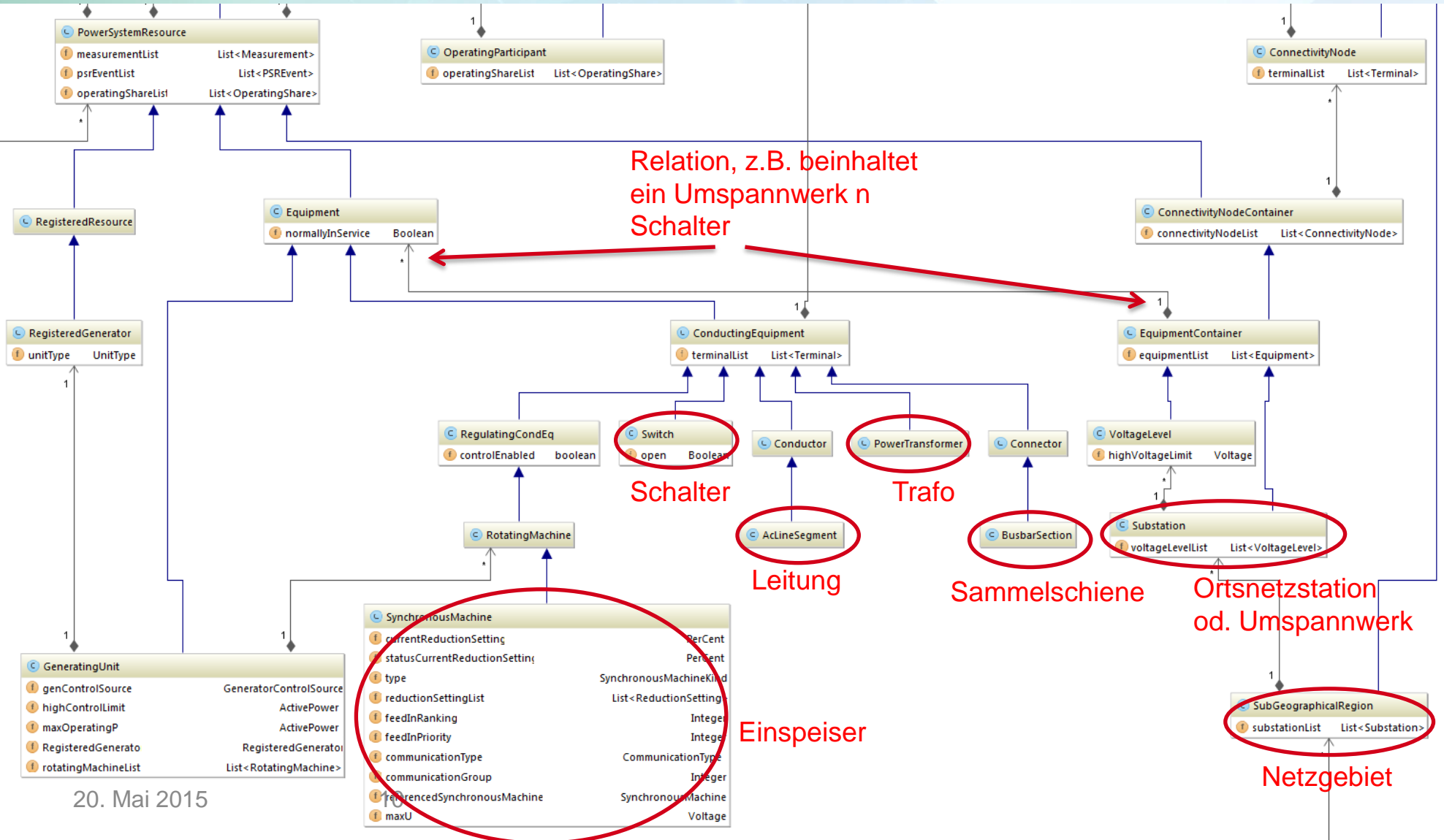


Ist-Werte Einspeiser

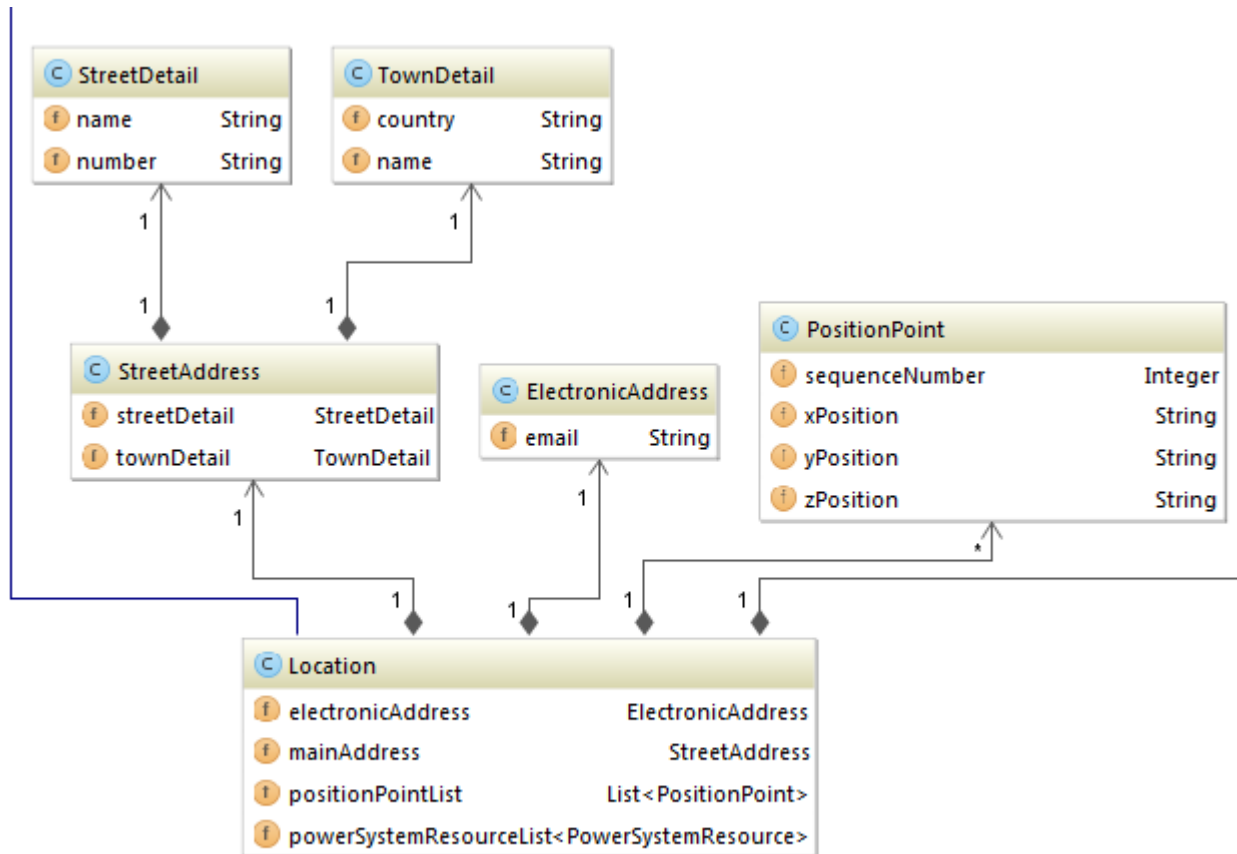
Stammdaten Einspeiser

Topologie

CIM - zentrale Klassen für das Einspeisemanagement



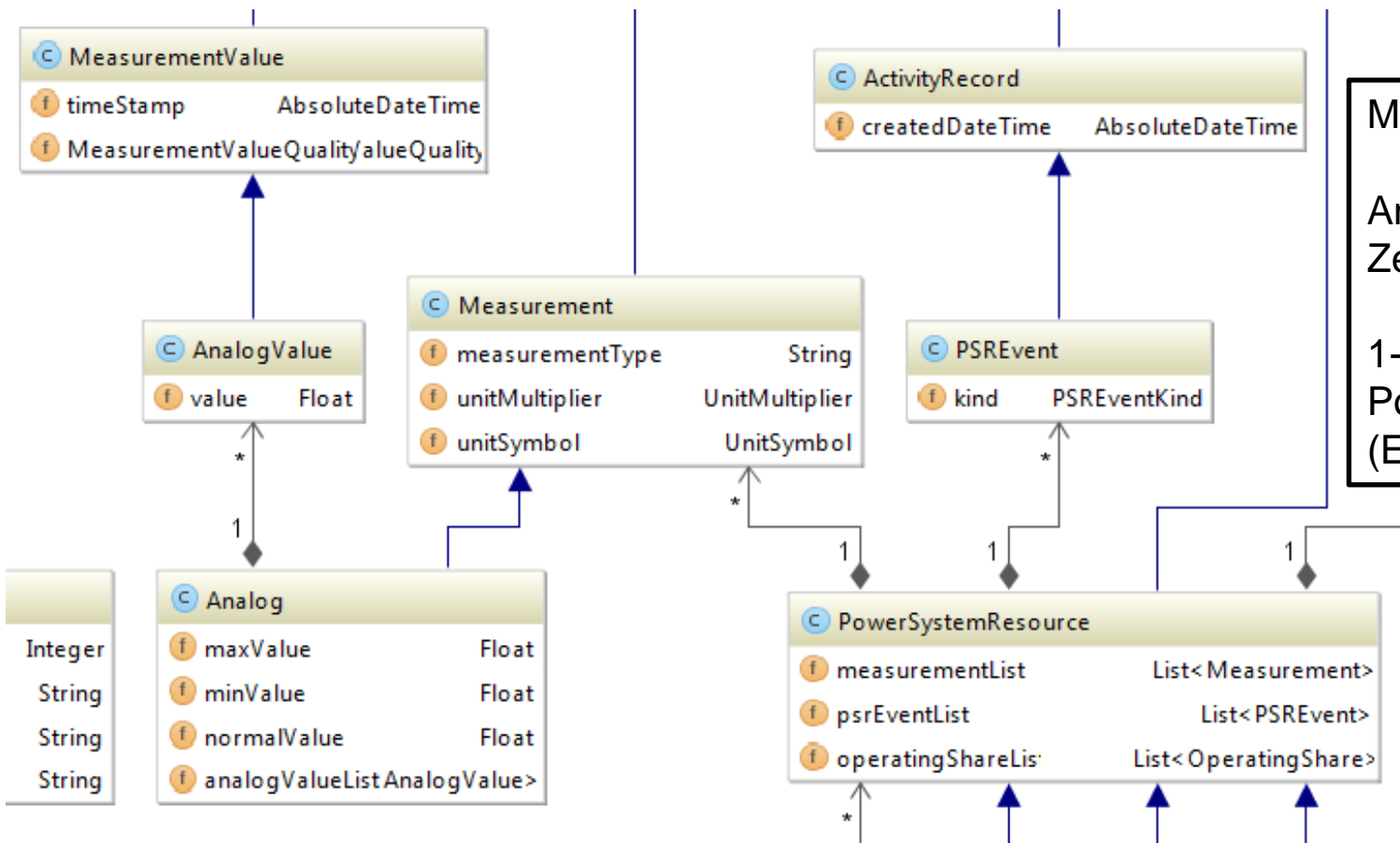
CIM – Adresdaten Betreiber/Partner



Strasse, Ort, PLZ, Mail, etc.

1-n Relation zu
PowerSystemResource
(Einspeisern)

CIM – Ist-Werte der Anlagen

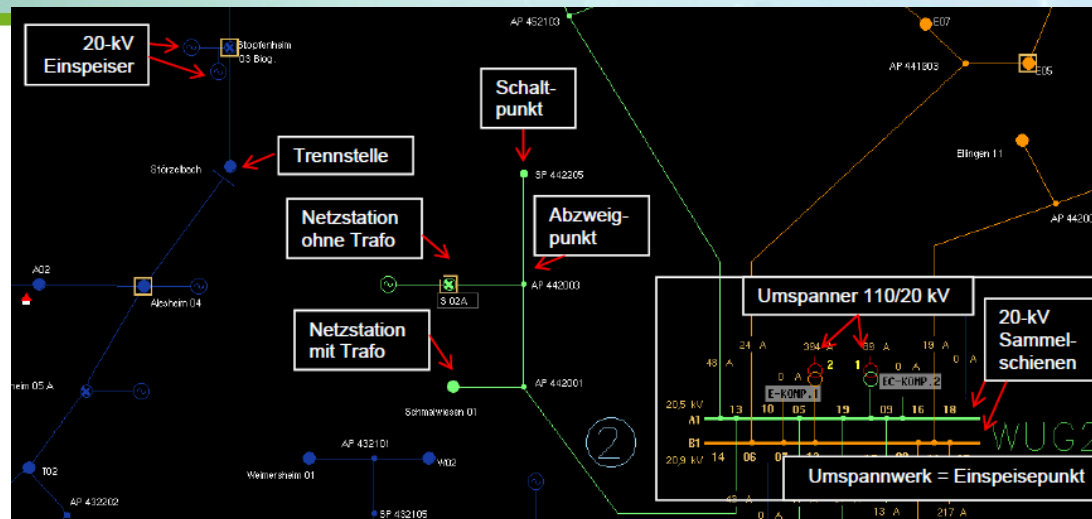


Measurement = Messreihe

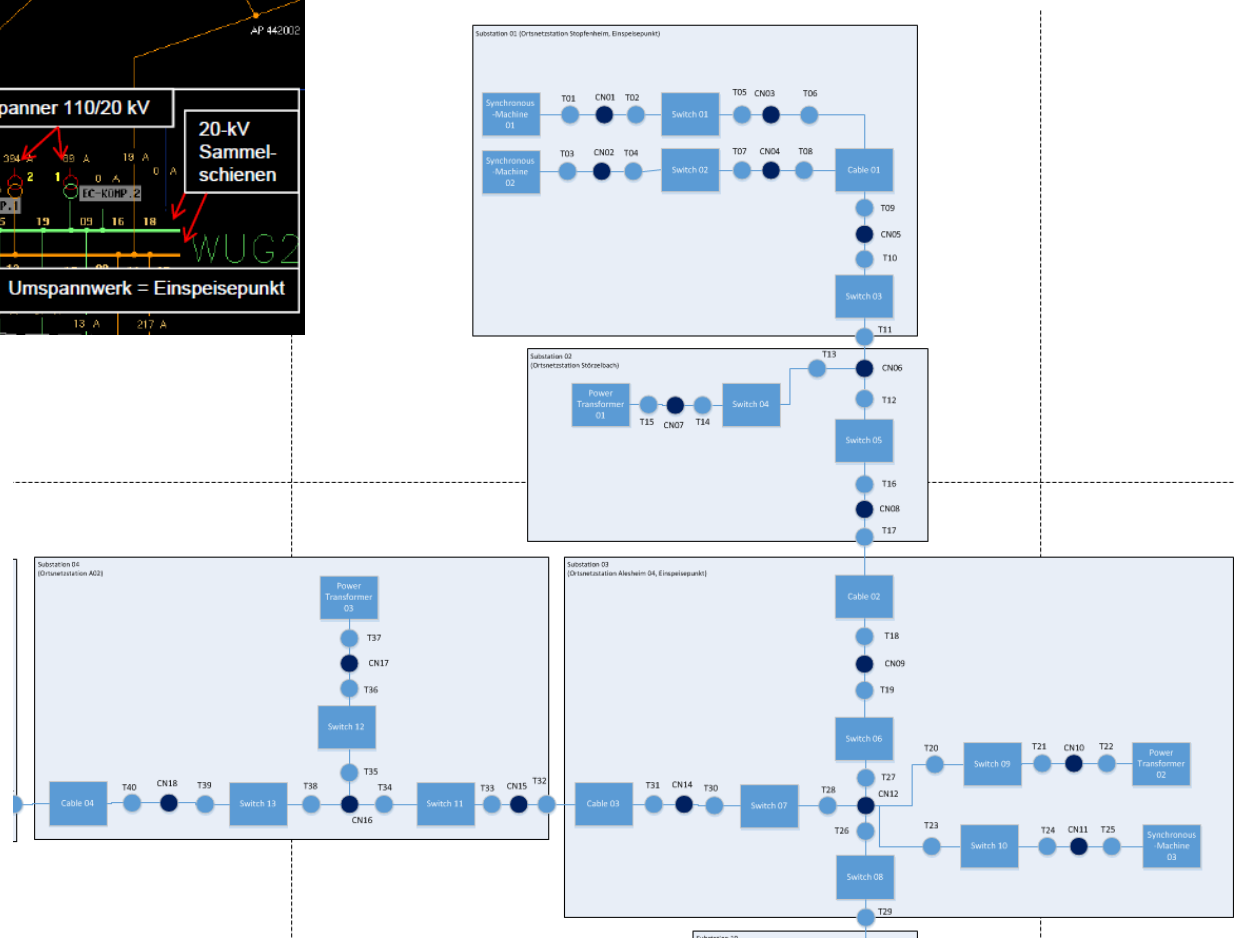
AnalogValue = Messwert mit Zeitstempel

1-n Relation Messreihe zu PowerSystemResource (Einspeisern)

CIM - Topologie



Terminals und
ConnectivityNodes
stellen die
Verbindungen der
Elemente eines
Stromnetzes dar



CIM - Erweiterungen für openk platform

neue Attribute in SynchronousMachine (Einspeiser)

- reductionSettingList (mögliche Absenkleistung einer Anlage)
- currentReductionSetting (aktuelle Absenkleistung)
- statusCurrentReductionSetting (Status)
- feedInRanking (Einspeiseranking nach BDEW)
- feedInPriority (Einspeisepriorität des Netzbetreibers)
- communicationType (FWT, EFR, ...)
- communicationGroup (z.B. Rundsteuergruppe mehrerer Anlagen, die nur gemeinsam geregelt werden können)
- referencedSynchronousMachine (Referenzanlage)

Attribute „**heruntergezogen**“ z.B. name und mRid

Relationen „**heruntergezogen**“

EquipmentContainer – Switch, SynchronousMachine, PowerTransformer,
BusbarSection u. AcLineSegment statt
EquipmentContainer - Equipment

JPA-Mapping &
Performance

Objekt-Relationales Mapping mit JPA

Inheritance Strategy

Inheritance Strategies beeinflussen Performance und Wartbarkeit!

Single-Table Strategy

- eine Tabelle für mehrere Klassen wobei Spalten nur von einer bestimmten Klasse verwendet werden
- DiscriminatorValue definiert die Klasse
- EquipmentContainer: Substation und VoltageLevel

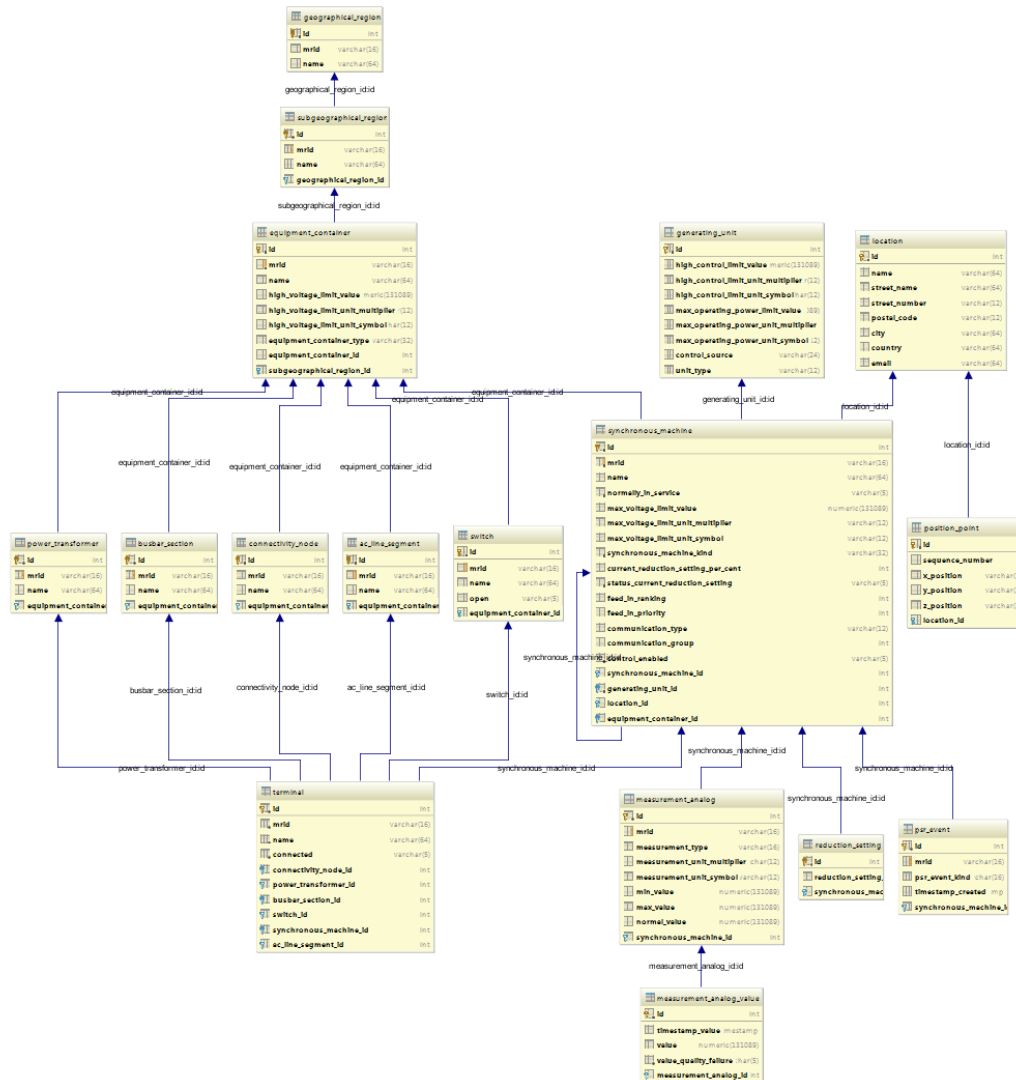
Joined-Table Strategy

- Jede Klasse der Hierarchie wird auf eine Tabelle „gemapped“, die dann mittels Joins verbunden sind
- kann zu „teuren“ Joins führen
- nicht angewendet in openk platform

Table-Per-Concrete-Class Strategy

- nur konkrete Klassen „erhalten“ eine Tabelle
- Anwendung in Equipment zu Switch, SynchronousMachine, AcLineSegment, etc

Datenbank Modell



Umsetzung der CIM-Klassen in ein relationales Datenbankmodell unter Verwendung der „Inheritance Strategies“

Anforderungen an Datenmodell:

- Performance
- Wartbarkeit
- Erweiterbarkeit

Entwicklung Uis - Vorgehen

Entwurf der Uis durch die BTC in Abstimmung mit Product Owner

Entwicklung Design durch die Firma Minnimedia

- Styleguide
- HTML-Seiten (inkl. Bootstrap und Javascript Artefakte)

The screenshot shows a web application interface for 'Benutzerangaben' (User Data) in a system called 'Eisman'. The interface is in German and shows a form for entering user information. The form is divided into sections for 'Voreinstellungen' (Default Settings), 'Benutzerangaben' (User Data), 'Abregelvorschlag' (Load Shedding Proposal), and 'Bestätigen' (Confirm). The 'Benutzerangaben' section is currently active and highlighted in green. It contains fields for 'Startdatum', 'Dauer', 'Grund der Abregelung' (set to 'Leitung überlastet'), 'Beforderte Gesamtregelleistung in MW' (set to '200'), 'Netzgebiet für Abregelung Gesamtnetz' (checkbox), 'Netzgebiet für Abregelung Netzgruppe' (set to 'Netzgruppe X'), and 'Netzgebiet für Abregelung Umspannwerk' (set to 'Umspannwerk X'). There are also checkboxes for 'Info an EEB Betreiber' and 'Info an nachgelagerte Netzbetreiber', and a 'Begründung' field. The 'Bestätigen' section has a 'Begründung' dropdown menu set to 'Eisman Eingriff'. At the bottom, there are radio buttons for 'Übung' (Yes/No) and 'Zurück' and 'Weiter' buttons.

Implementierung UIs

Menü

Breadcrumb

Tabelle: sortierbar, Paging, Anzahl der Zeilen konfigurierbar

ID	Von	Bis	Abgeregete Leistung	Begründung	Speisepunkte	Übung
1012	18.05.2015 04:10:35		22 MW	Inspection		Nein
1013	18.05.2015 04:10:35		2244 kW	Inspection		Nein
1014	18.05.2015 04:10:35		2244 kW	Inspection		Nein
1015	18.05.2015 04:10:35		2244 kW	Inspection		Nein
1016	18.05.2015 04:10:35		2244 kW	Inspection		Nein
1017	18.05.2015 04:10:35		2244 kW	Inspection		Nein
1018	18.05.2015 04:10:35		2244 kW	Inspection		Nein

Detailansicht

Abzuregelnde Leistung: 2244 kW

Erstellt am: NaN.NaN.NaN NaN:NaN:NaN

Startzeitpunkt: 18.05.2015 04:10:35

Begründung: Inspection

Neue Massnahme

Build Process als Maven Projekt

TestNG: Ausführung in IDE u. maven

```
@TransactionConfiguration(defaultRollback = false)
@Transactional(value = "importDbTransactionManager", rollbackFor = Throwable.class)
@TestExecutionListeners({TransactionalTestExecutionListener.class})
@ContextConfiguration(locations = {"classpath:spring-persistence-test.xml"})
public abstract class AbstractImportDbTransactionalBaseTest extends AbstractTransactionalBaseTest {
    @PersistenceContext(unitName = "importDbEntityManagerFactory")
    protected EntityManager entityManager;
}

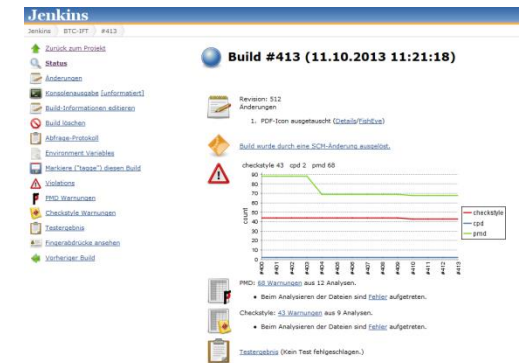
public abstract class AbstractTransactionalBaseTest extends AbstractTestNGSpringContextTests {
    ...
}
```



Integrationstest „gegen“ Datenbank und Tests für das Verarbeiten von REST-Services (JSON)

ToDo: Continuous Build

Checkstyle



Vielen Dank für Ihre Aufmerksamkeit!

Fragen & Antworten