

R&D Projekt

Semesterendbericht

**Bezeichnung**: Erstellung einer JRZ Demodatenbank (DemoDB)

**Projektschlüssel:** RD16-03

**Betreuer**: DI Eduard Hirsch, DI Fabian Knirsch, BSc

**Kurzbeschreibung**: Konvertierung verschiedener Smartmeter Messdaten und Ablage in einer gemeinsamen Datenbank mit rollenbasiertem Zugriff.

**Beteiligte** **Firma**: Salzburg AG

**Studenten**: Isidor Reimar Klammer, BSc.

Maximilian Unterrainer, BSc.

Christopher Wieland, BSc.

Puch/Salzburg, 07.03.2017

Inhaltsverzeichnis

[1 Einleitung 1](#_Toc476497444)

[1.1 Problemstellung und Motivation 1](#_Toc476497445)

[1.2 Umgebung 2](#_Toc476497446)

[2 Anforderungsanalyse 3](#_Toc476497447)

[2.1 Datenmodell 3](#_Toc476497448)

[2.1.1 Analyse der JRZ-DB 3](#_Toc476497449)

[2.1.2 Anforderungen von Energieversorgern und Netzdienstleistern 4](#_Toc476497450)

[2.1.3 Anforderungsprofil „Lastenheft Österreichs Energie“ 5](#_Toc476497451)

[2.1.4 Rechtliche Rahmenbedingungen 6](#_Toc476497452)

[2.1.5 Anpassung des Datenmodells 6](#_Toc476497453)

[2.2 Rollenbasierter Zugriff 7](#_Toc476497454)

[2.2.1 Rollenidentifikation 7](#_Toc476497455)

[2.2.1.1 Messdaten aus dem Echtbetrieb 7](#_Toc476497456)

[2.2.1.2 Anonymisierte Messdaten aus dem akademischen oder Forschungsbereich 8](#_Toc476497457)

[2.2.2 Einbeziehung zusätzlicher Domänen 8](#_Toc476497458)

[2.2.3 Rollendefinition 9](#_Toc476497459)

[2.2.4 Verbindung zur Rollenverwaltung 9](#_Toc476497460)

[2.3 Datenbankanforderungen 10](#_Toc476497461)

[2.3.1 Testdaten 11](#_Toc476497462)

[2.3.2 Messung 11](#_Toc476497463)

[2.3.3 Erwartete Datenmengen 13](#_Toc476497464)

[2.4 Schnittstellen 14](#_Toc476497465)

[2.5 Systemarchitektur 16](#_Toc476497466)

[3 Ausblick 17](#_Toc476497467)

[3.1 Alternative Datenhaltung 17](#_Toc476497468)

[4 Recherche 19](#_Toc476497469)

[4.1 ER-Modell 19](#_Toc476497470)

[4.2 Analyse von Datenbankanforderungen 19](#_Toc476497471)

[4.2.1 Analyse JRZ-DB 20](#_Toc476497472)

[4.2.2 Performanceanalyse 21](#_Toc476497473)

[4.3 Analyse vorhandener Software 21](#_Toc476497474)

[4.4 Rollendefinition 21](#_Toc476497475)

[5 Umsetzung 22](#_Toc476497476)

[5.1 Systemarchitektur 22](#_Toc476497477)

[5.2 ER-Modell 22](#_Toc476497478)

[5.3 Analyse der Datenmodelle 23](#_Toc476497479)

[5.3.1 Hadoop Tests 23](#_Toc476497480)

[5.3.2 MySQL Tests 23](#_Toc476497481)

[5.3.3 Testdaten 23](#_Toc476497482)

[6 Ergebnisse 25](#_Toc476497483)

[6.1 Systemarchitektur 25](#_Toc476497484)

[6.2 ER Modell 27](#_Toc476497485)

[6.2.1 Analyse bestehendes Datenmodell 27](#_Toc476497486)

[6.2.2 Anforderungen von Energieversorgern und Netzbetreibern 27](#_Toc476497487)

[6.2.3 Analyse der Usecases 28](#_Toc476497488)

[6.2.4 Weitere Datenmodelle 28](#_Toc476497489)

[6.2.5 Rechtliche Rahmenbedingungen und daraus resultierende Einschränkungen der möglichen Messwerte 29](#_Toc476497490)

[6.2.6 Das COSEM Modell 29](#_Toc476497491)

[6.2.7 ER-Modell Festlegung 30](#_Toc476497492)

[6.3 Analyse möglicher Datensätze 32](#_Toc476497493)

[6.3.1 MySQL Datenbank 32](#_Toc476497494)

[6.3.2 Hadoop Datenbank 34](#_Toc476497495)

[6.3.3 Zusammenfassung der Datenbankanalyse 34](#_Toc476497496)

[6.4 Analyse vorhandener Software 35](#_Toc476497497)

[6.5 Rollendefinition 36](#_Toc476497498)

[7 Weitere Schritte 37](#_Toc476497499)

[Literaturverzeichnis 38](#_Toc476497500)

[Anhang 41](#_Toc476497501)

[SQL Messungen 41](#_Toc476497502)

Abkürzungsverzeichnis

AMCS Advanced Meter Communication System

API Application Programming Interface

COSEM Companion Specification for Energy Metering

DAVID-VO Datenformat- und Verbrauchsinformationsdarstellungs Verordnung

DBM Datenbankmodell

DLSM Device Language Messaging Specification

ETSI European Telecommunication Standards Institute

ElWOG Elektrizitätswirtschafts- und -organisationsgesetz

GDPR EU General Data Protection Regulation

IMA-VO Intelligente Messgeräte-AnforderungsVO 2011

JRZ Josef Ressel Zentrum für Anwenderorientierte Smart Grid Privacy, Sicherheit und Steuerung

JRZ-DB Datenbank, die im JRZ eingesetzt wird und auf deren Basis die gemeinsame Datenplattform entwickelt wird.

MMS Manufacturing Messaging Specification

OBIS Object identification system, entsprechend der EN 62056-01

OSGP Open Smart Grid Protocol

RBAC Role Based Access Control

REST Representational State Transfer

SmartValAPI Smart Meter Data Value API, Arbeitstitel des Projektes

SML Smart Message Language

SOAP Simple Object Access Protocol

JSON JavaScript Object Notation

XML Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: Datenmodell JRZ-DB 4

Abbildung 2: erweiterte Tabelle meter\_data 7

Abbildung 3: Tabelle REDD mit Testdaten 11

Abbildung 4: Dauer Berechnung des Mittelwerts auf der MySQL Datenbank 12

Abbildung 5: Dauer des Einfügens in die MySQL Datenbank 13

Abbildung 1: Abfrage Durchschnitt pro Tag 25

Abbildung 2: Komponentenmodell SmartValAPI 29

Abbildung 3: ER-Modell Entitäten 32

Abbildung 4: Datenbankbeziehungen 33

Abbildung 5: Dauer des Datenimports 35

Abbildung 6: Dauer Berechnung des Durchschnittsverbrauchs 36

Abbildung 7: Rollendefinition - UseCase Diagramm 38

Tabellenverzeichnis

Tabelle 1: Rücklauf der Anfragen bei Energieversorgern 5

Tabelle 2: maximale Auflösung auf Grund einer zugeteilten Rolle 9

Tabelle 3: Zugriff auf anonymisierte Messdaten aus Forschungsquellen 9

Tabelle 1: Datenformat CSV 16

Tabelle 2: Dauer des Datenimports 26

Tabelle 3: Dauer Berechnung des Durchschnittsverbrauchs 27

Listing Verzeichnis

[Listing 1: CSV Format der REDD Daten 11](file:///C:\Users\maximilian\Githubs\R&D\Projektplanung\Zwischenbericht_Februar%20-%20nachbearbeitet%20-%20gemerged%20am%205.3..docx#_Toc476493977)

[Listing 2: Berechnung des Durchschnittsverbrauchs pro Meter, Tag und Monat 12](file:///C:\Users\maximilian\Githubs\R&D\Projektplanung\Zwischenbericht_Februar%20-%20nachbearbeitet%20-%20gemerged%20am%205.3..docx#_Toc476493978)

[Listing 3: Klasse für den Datenzugriff 15](file:///C:\Users\maximilian\Githubs\R&D\Projektplanung\Zwischenbericht_Februar%20-%20nachbearbeitet%20-%20gemerged%20am%205.3..docx#_Toc476493979)

[Listing 4: Basisklasse um Abfragen zu definieren 16](file:///C:\Users\maximilian\Githubs\R&D\Projektplanung\Zwischenbericht_Februar%20-%20nachbearbeitet%20-%20gemerged%20am%205.3..docx#_Toc476493980)

# Einleitung

Durch die Verabschiedung der Richtlinie 2009/72/EC [1] sind die Mitgliedsstaaten der EU aufgefordert, deren Inhalte in nationales Recht umzusetzen. Thema dieser Richtlinie ist es, die vorhandenen analogen Stromzähler durch digitale Smart Meter zu ersetzen. Mit der flächendeckenden Installation stehen sowohl den Netzbetreibern als auch den Energieproduzenten und den Verbrauchern Möglichkeiten das Netz optimal zu nützen, Energie zu günstigen Preisen zu erwerben und Energieverschwendung zu verringern [2]. Um diese Vorteile zu nützen, ist Kommunikation bezüglich des aktuellen Verbrauchs, der Netzbelastung und der im Netz vorhandenen Energie notwendig.

Über Kommunikationsprotokolle tauschen Verteilstationen, Energieeinspeiser und Smart Meter beim Endkunden Daten bezüglich des Verbrauchs aus. Der Preis für diese Vorteile ist die notwendige, zumindest teilweise Offenlegung des Energieverbrauchs des Endkunden.

Im Spannungsfeld von Schutz der Privatsphäre einerseits, und maschineller Messdatenauswertung im Rahmen des Erlaubten andererseits sollen die Ergebnisse dieses Projekts für Komfortverbesserung sorgen.

## Problemstellung und Motivation

Für die Ablage und Verwaltung von Smart Meter Messdaten setzt das JRZ eine hausintern geschriebene Datenbankapplikation (JRZ-DB) ein. Es bestehen bereits einige Importprogramme die Messdaten aus unterschiedlichen Formaten, zum Beispiel REDD [3], UK-DALE [4], ADRES [5] und GREEND [6], in das Datenbankformat konvertieren. Hingegen existiert kein zentrales Zugriffsmodul, jede Anwendung, die auf diese Daten zugreift, benötigt das Wissen über den Tabellenaufbau, die Zugriffspfade sind je nach Anforderung neu zu implementieren und in der Folge zu warten. Nach der Umsetzung dieses Projekts steht existierenden und zukünftigen Applikationen ein vereinheitlichter und damit vereinfachter Zugriff zur Verfügung.

Die Speicherung der Messdaten aus unterschiedlichen Quellen in einer Datenbankinstanz kann in Verbindung mit uneingeschränktem Zugriff datenschutzrechtliche Probleme aufwerfen, andererseits ist der gleichzeitige Zugriff für vergleichende Auswertungen notwendig. Die Einbindung einer externen Komponente zur rollenbasierten Zugriffregelung löst diese Anforderung, der Zugriff auf Messdaten im Allgemeinen und auf bestimmte Auflösungen im Speziellen wird über Berechtigungen des Benutzers gesteuert.

Dieses Projekt verfolgt vier Hauptziele:

* Schaffung einer erweiterbaren Programmierschnittstelle (SmartValAPI), die einen geregelten Zugriff auf Smartmeterdaten ermöglicht
* Einbindung und gegebenenfalls Erweiterung der im JRZ eingesetzten Datenbank (JRZ-DB, Details siehe Abschnitt Anforderungen an das ER-Modell) als einheitliche Datenplattform für bereits existierende Anwendungen
* Evaluierung alternativer Datenbanksysteme zur Ablage der Messdaten
* Einbindung einer rollenbasierten Zugriffsverwaltung

Nach der erfolgreichen Umsetzung des Projektes steht der Zugriff auf alle gespeicherten Messdaten, den Berechtigungen entsprechend, für programmtechnische Auswertungen in vereinheitlichter Form zur Verfügung.

## Umgebung

Die Umsetzung des Projektes erfolgt unter zu Hilfenahme von bereits im Umfeld des Josef Ressel Zentrums für Anwenderorientierte Smart Grid Privacy, Sicherheit und Steuerung an der Fachhochschule Salzburg durchgeführten Projekte. Im Detail sind dies:

* Datenmodell: die JRZ-DB stellt die Ausgangsbasis für mögliche, notwendige Erweiterungen dar und wird auf deren Eignung für die zu erwartenden Messdatenmengen evaluiert.
* Importmodule [7], um Messdaten in der Datenbank abzulegen. Dieses Programmpaket ermöglicht es, Messwerte, die in den Formaten ADRES, GREEND, REDD und UK-DALE vorliegen zu importieren.
* OpenTC [8] stellt eine rollenbasierte Authentifizierung und Autorisierung zur Verfügung, über die der Zugriffsschutz realisiert wird, die Rollenverwaltung erfolgt über ein beliebiges LDAP-Administrationswerkzeug.

Weitere verwendete Softwarepakete werden im Abschnitt Systemarchitektur angeführt.

# Anforderungsanalyse

Aus dem Projektauftrag können Anforderungen abgeleitet werden, die Details dieser Anforderungen und den Weg zur jeweiligen Entscheidungsfindung beleuchtet dieser Abschnitt.

## Datenmodell

Ausgangsbasis für die Anforderungen ist die bereits eingesetzte JRZ-DB, analysiert wird die Eignung für die Datenhaltung. Anforderungen von Energieversorgern und Netzdienstleistern fließen in die Untersuchung mit ein. Nicht zuletzt wird die Kompatibilität zu existierenden Anwendungen erhalten.

## Analyse der JRZ-DB

Die Tabellenwelt der JRZ-DB kann in zwei Gruppen eingeteilt werden: einerseits Stammdaten, wie zum Beispiel meter\_management, meter\_type oder customer\_data und anderseits Bewegungsdaten: meter\_data. Letztere Tabelle ist über einen Fremdschlüssel mit meter\_management verbunden. Die Welt der Stammdaten wird ohne Änderungen übernommen.

Alle Messdaten, die von einem Smart Meter zu einem Zeitpunkt ausgelesen werden, speichert die JRZ-DB als Tupel in der Tabelle meter\_data. Abgelegt. Nicht jedes Smart Meter stellt alle Werte zur Verfügung, in der JRZ-DB werden nicht vorhandene als NULL-Wert gekennzeichnet. Je Messzeitpunkt können folgende Werte abgelegt werden.

* Nutzdaten (Momentanwerte):   
  je Phase, aktuelle Leistung (power\_p1, power\_p2, power\_p3), aktueller Stromverbrauch (work \_p1, work \_p2, work\_p3),

4 freie Werte (count\_register1 – count\_register4), die abhängig vom Typ des Smart Meter (meter\_type) belegt werden,   
Gesamtwerte für Spannung (voltage) und Frequenz (frequency),  
kumulierter Verbrauch (count\_total).

* Verwaltungsdaten (zur Identifikation):  
  Identifikationsnummer des Smart Meter (meter\_id), Fremdschlüssel zu meter\_management,  
  eindeutiger Schlüssel des Messdaten-Tupels (data\_id),  
  Erstellungszeitpunkt zu dem die Nutzdaten aufgezeichnet werden (timestamp).

Das bestehende Modell der JRZ-DB gibt Abbildung 1 wieder. [neu zeichnen]



Abbildung 1: Datenmodell JRZ-DB

## Anforderungen von Energieversorgern und Netzdienstleistern

Neben den Anforderungen des JRZ, als Auftraggeber, werden die möglichen Bedürfnisse von Energieversorgern und Netzbetreibern ermittelt, dazu wurde ein Fragenkatalog erstellt und an folgende 10 in diesem Feld tätigen Unternehmen übermittelt: Ebner Strom GmbH, Energie AG, Energie Steiermark, EVN AG und Netz Niederösterreich GmbH, Linz AG, Salzburg Netz GmbH, TINETZ – Tiroler Netze GmbH, Vorarlberger Energienetze GmbH und Wien Energie. Diese Fragen wurden gestellt:

* + Welche Messwerte, abgesehen von Spannung, Strom und Wirkleistung, jeweils phasengetrennt sind für Sie als Energieversorger/Netzbetreiber von Bedeutung?
  + In welcher Granularität (einzelne Smart Meter/Gruppen zB Trafostation) sollen diese Messdaten zur Verfügung stehen?
  + Mit welcher/n Abtastrate/en soll/en diese Messwerte zur Verfügung stehen?
  + Welche Werte, auf Basis der in der JRZ-DB sind verzichtbar?
  + Weitere freie Hinweise.

Das Ergebnis zeigt Tabelle 1, auf Grund der Forderung einiger Betriebe, in anonymisierter Form zusammengefasst.

|  |  |
| --- | --- |
| **Zusätzliche Messwerte** | |
| zusätzliche Werte (Phasenlage) | 2 |
| Kein Interesse an Erweiterung bzw. Verweis auf das ElWOG [9] | 2 |
| Keine Auskunft | 6 |
| **Granularität** | |
| Gruppierung | 1 |
| Keine Auskunft | 9 |
| **Auflösung** | |
| Messwerte in der vorliegenden, dem ElWOG [9] entsprechenden, Auflösung sind ausreichend | 4 |
| Keine Auskunft | 6 |
| **Verzichtbare Werte** | |
| Netzfrequenz | 2 |
| Keiner der Werte soll weggelassen werden | 5 |
| Keine Auskunft bzw. keine Antwort | 3 |
| **Weitere Hinweise** | |
| Datenschutzkritische Anwendung | 3 |
| Verweis auf ElWOG, daher gilt „interessant“ nicht | 1 |

Tabelle 1: Rücklauf der Anfragen bei Energieversorgern

## Anforderungsprofil „Lastenheft Österreichs Energie“

Österreichs Energie, als Interessensgemeinschaft der österreichischen Netzbetreiber und einem Großteil der österreichischen Energieversorger, fasst im Zuge des Vergabeverfahrens der Smart Meter Einführung die zu erfüllenden Anwendungsfälle in [10] und der Arbeitsunterlage zur Erstellung eines Lastenheftes in [11] zusammen. Diese Anwendungsfälle beschreiben vor allem die Anforderungen an die Fernsteuerung von bereits beim Kunden installierten Smart Meter Geräten und nur am Rande mit der inhaltlichen Bedeutung der übertragenen Messdaten. Neben den, in der JRZ-DB in *meter\_data* bereits vorhandenen Datenfeldern, bietet das Lastenheft optional die Möglichkeit der Auslesung der Blindleistung.

## Rechtliche Rahmenbedingungen

Die Erfassung, Übertragung und Speicherung von Smart Meter Messdaten wird in vier Richtlinien geregelt:

* ElWOG [9]
* GDPR [12]
* IMA-VO [13]
* DAVID-VO [14]

Geregelt werden einerseits Mindestanforderungen an Smart Meter, andererseits die Inhalte und die Frequenzen, mit denen die Werte ausgelesen werden dürfen. Nach §84 ElWOG müssen dem Verbraucher die Daten bezüglich des „Verbrauchs der über ein intelligentes Messgerät gemessen wird“ zeitnah zur Verfügung zu gestellt werden. Es erfolgt keine genauere Definition, welche Daten dies im Detail sind, lediglich die Frequenzen, mit denen ausgelesen wird, sind festgelegt. Details dazu, finden sich im Abschnitt Rollenidentifikation. Bezüglich der auslesbaren Daten legt das Lastenheft von Österreichs Energie [11] jene Messwerte fest, die von einem Smartmeter übertragen werden müssen. Die IMA-VO befasst sich mit der Einführung der der intelligenten Zähler. die GDPR regelt die kontrollierte Löschung nach der Verarbeitung von Messdaten als Form der personenbezogenen Daten. Beide haben daher keinen Einfluss auf das Datenmodell. Die Weitergabe der erfassten Daten an berechtigte Dritte wird von der DAVID-VO geregelt, diese Verordnung hat keinen Einfluss auf das Datenmodell, wohl aber auf die maximale Frequenz mit der Messdaten ausgelesen werden können. Die Details dazu werden daher im Abschnitt Rollenidentifikation angeführt.

## Anpassung des Datenmodells

Auf Grund der Rückmeldungen der Energiedienstleister kommt es, in Übereinstimmung mit den Usecases von Österreichs Energie zu folgender Erweiterung von meter\_data:

[hier fehlt die Tabelle meter\_data wie sie nach der Erweiterung ausschaut + Tabelle #]

Weitere Datenfelder, wie sie zum Beispiel das ETSI in der Definition des OSGP [15] vorschlägt, werden nicht in die Menge der gespeicherten Daten aufgenommen, da diese weder von den Energieversorgern in den Rückmeldungen noch von den Netzbetreibern im Lastenheft gewünscht werden.

Um die Vorgabe der Kompatibilität zu erfüllen werden keine Felder aus der Tabelle meter\_data entfernt, auch wenn zum Beispiel auf die Netzfrequenz aus Sicht der Netzbetreiber verzichtet werden könnte. Abbildung 2 zeigt die erweiterte Tabelle meter\_data. [neu zeichnen]

Abbildung 2: erweiterte Tabelle meter\_data

## Rollenbasierter Zugriff

Werden Messdaten aus unterschiedlichsten Quellen in einer gemeinsamen Datenbank abgelegt, darf der Zugriff darauf nur jenen Benutzern gewährt werden, die dazu berechtigt sind. Neben der Verhinderung von nicht autorisierten Zugriffen auf fremde Messdaten muss eine weitere Anforderung erfüllt werden: die Wartung des Zugriffs über Rollen. Bei einem rollenbasierten Zugriffsmodell werden einzelne Berechtigungen an Rollen zugeteilt und nicht an Benutzer vergeben. Das vereinfacht die Administration und ermöglicht eine flexible Anpassung der Berechtigungen [16].

## Rollenidentifikation

Die Messdatenquellen können auf Grund der Analyse in zwei Gruppen, die unterschiedliche Anforderungen an das Zugriffsystem stellen, eingeteilt werden.

## Messdaten aus dem Echtbetrieb

Für die Erfassung und damit einhergehend dem Zugriff auf Messdaten aus dem Echtbetrieb gelten im Wesentlichen zwei Verordnungen: das ElWOG [9] und die DAVID-VO [14]. Das ElWOG legt in §84 die Rahmenbedingungen für die Erfassung von Messdaten zum Zwecke der Verrechnung, der Kundeninformation, der Energieeffizienz, der Energiestatistik und der Aufrechterhaltung eines sicheren und effizienten Netzbetriebes fest. Abs. (1) legt Periode der Erfassung auf 15 Minuten fest. Die Weitergabe der Viertelstundenwerte ist nur bei ausdrücklicher Zustimmung des Endverbrauchers möglich (Abs. (2)), in begründeten lokalen Einzelfällen, zur Aufrechterhaltung eines sicheren Netzbetriebes, ist die Weitergabe von Viertelstundenwerten ohne ausdrückliche Erlaubnis möglich. Ohne Zustimmung zur Weitergabe steht ein Messwert je 24 Stunden zur Verfügung.

Die DAVID-VO §4 regelt die Übertragung der Messdaten und in der Folge die Bereitstellung für den Kunden über eine Website. Auf Kundenwunsch wird hier die kleinstmögliche Auflösung dargestellt, ohne diese Zustimmung wird ein Wert je 24 Stunden dargestellt. Die Weitergabe erfolgt an den Energielieferanten und optional an einen berechtigten Energieberater.

Messdaten werden in Verbindung mit Kundendaten nach [17] zu einer Ausprägung von personenbezogenen Daten. Es ist daher der Artikel 17 der GDPR [12] anzuwenden. Es wird darin ausgesagt, dass Daten, welche für die Zwecke, für die sie erhoben wurden, nicht mehr benötigt werden, zu löschen sind. Da einerseits der Datenimport keinen Teil dieses Projektes darstellt und andererseits davon ausgegangen wird, dass zur Verfügung gestellte Messdaten anonymisiert wurden, finden die beiden Regelwerke keine Anwendung. Es wird daher im Zuge dieses Projektes in der Folge nicht weiter auf die Löschung von Messdaten eingegangen.

## Anonymisierte Messdaten aus dem akademischen oder Forschungsbereich

Bezüglich der anonymisierten Messwerte gelten diese rechtlichen Einschränkungen nicht. Eine Beschränkung des Zugriffs auf Daten aus Quellen wie zum Beispiel REDD, UK-DALE, ADRES oder GREEND ist aus diesem Blickwinkel nicht notwendig. Manche der anonymisierten Datensätze, beispielsweise REDD, stehen nur mit Benutzername/Passwort-Zugriff zur Verfügung, sie werden daher auch nicht jedem Benutzer des API zugänglich gemacht. Gekennzeichnet werden Messwerte aus anonymisierten Quellen über ein LDAP-Attribut, Details dazu folgen im Abschnitt 2.3.3 .

### Einbeziehung zusätzlicher Domänen

Aktuell wurde die die Richtlinie 2006/3 2/EG in Österreich für die Datenerfassung und Kommunikation von Messgeräten für elektrische Energie umgesetzt, derzeit gibt es in Österreich keine äquivalenten Grundlagen für Gas, Wärme und Wasser. Technisch ist eine Erweiterung um jene Felder, die nach der rechtlichen Festlegung erfasst werden sollen, problemlos möglich, daher wird ein Einsatz in weiteren im Zuge der Umsetzung dieses Projekts nicht weiter in Betracht gezogen.

## Rollendefinition

Aus den in den Abschnitten 2.3.1.1 und 2.3.1.2 angestellten Überlegungen werden folgende Rollen mit den ihnen zugeordneten Auflösungen abgeleitet:

|  |  |
| --- | --- |
| **Rolle** | **maximale Auflösung** |
| Smart Meter Besitzer | 15-Minuten |
| Netzbetreiber, Energieversorger | 24-Stunden |
| Energieberater | 15-Minuten |

Tabelle 2: maximale Auflösung auf Grund einer zugeteilten Rolle

|  |  |
| --- | --- |
| **Rolle** | **Zugriff** |
| Akademischer oder Forschungsbenutzer | Höchste vorhandene Auflösung |

Tabelle 3: Zugriff auf anonymisierte Messdaten aus Forschungsquellen

In Tabelle 2 und Tabelle 3 werden die Zugriffsberechtigungen mit den dafür notwendigen Rollen in Verbindung gebracht und dargestellt. Anzumerken ist, dass für einen Zugriff auf nicht anonymisierte Daten nicht nur die jeweilige Rolle vonnöten ist. Zusätzlich wird eine Zuordnung zwischen Benutzer und Smart Meter hergestellt. Diese grundlegende Berechtigung wird in der Implementierung als Attribut zum Benutzer abgelegt.

## Verbindung zur Rollenverwaltung

Für die Administration der Zugriffsberechtigungen wird die Komponente Open-TC [8] eingesetzt. Alle rollenrelevanten Attribute werden hier abgelegt. Eine Übersicht der dieser Attribute zeigt Tabelle 4.

|  |  |
| --- | --- |
| **Attribut** | **Mögliche Werte** |
| Rolle | Besitzer, Energieversorger, Energieberater, Forscher |
| Kunde | customer\_id |

Tabelle 4: LDAP-Attribute zu Benutzer-Objekten

Um im bestehenden Datenmodell keine Erweiterungen bezüglich des Zugriff machen zu müssen, werden Smart Meter, die Messdaten aus akademischen oder Forschungsquellen enthalten werden mit der objectClass Computer abgelegt, das identifizierende Attribut verweist, wie in Tabelle 5 zu sehen, auf die meter\_id.

|  |  |
| --- | --- |
| **Attribut** | **Mögliche Werte** |
| cn | meter\_management.meter\_id |
| MeterType | open |

Tabelle 5: LDAP-Objekt für anonymisierte Messdatenquellen

## Datenbankanforderungen

Ziel des Projekts bezüglich der verwendeten Datenbank sind die Weiterverwendung des bestehenden Datenmodells und die Recherche nach möglichen Alternativen.

Für die Weiterverwendung des bestehenden Modells spricht, dass es bereits mehrere [Gruppen| Komponenten] gibt, die dieses System verwenden [7], [18]. Für die Ablösung des bestehenden Modells spricht eine eventuell bessere Performance alternativer Datenbanken bei großen Datenmengen.

Da die Weiterverwendung gewünscht wird, werden zuerst Performancemessungen auf einem Relationalen Datenbanksystem durchgeführt. Ist die Performance ausreichend, wird diese Datenbank verwendet.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Eigene Vorkenntnisse | Popularität | Open Source (Kostenlos) |
| MySQL | 3 | 3 | 1 |
| PostreSQL | 1 | 2 | 1 |
| MariaDB | 0 | 1 | 1 |

Tabelle 6: Auswahlkriterien Relationale Datenbank

Für die Auswahl der Datenbank wurden die Vorkenntnisse und freie Verfügbarkeit als Kriterien definiert. Da die Erfahrung mit Datenbanken in der Projektgruppe allgemein eher gering ist, ist die erwartete Unterstützung aus der Community ein weiteres Kriterium. Dafür wurde die Anzahl der Fragen innerhalb einer Woche auf Stack Overflow verwendet. Nach den Kriterien aus Tabelle 6 ist die Wahl auf MySQL gefallen.

Zum Zeitpunkt der Messung sind wir von ungefähr 250.000.000 Datensätzen innerhalb eines Jahres ausgegangen. Das ergibt sich aus 500.000 Smartmetern der Salzburg AG (1 Wert pro Tag) und ungefähr 60.000.000 Messwerten aus den REDD Daten.

## Testdaten

Als Testdaten wurden die ‚low\_freq‘ Daten aus dem REDD Datensatz verwendet. Die Daten liegen in pro Haus in einem eigenen Ordner und dort pro Kanal in einem eigenen File. Die Files sind ‚channel\_X.dat‘ benannt wobei X eine fortlaufende Nummer ist. Parallel zu den Messwerten liegt eine Datei ‚labels.dat‘ in welche die Zuordnung der einzelnen Kanäle zu den Verbrauchern ermöglicht. Die Messwerte sind, wie in Listing 1 dargestellt, in einem CSV Format abgespeichert, wobei das Trennzeichen ein Leerzeichen ist.

Listing 1: CSV Format der REDD Daten

timestamp\_power

## Messung

Für die Performance Messungen wurde eine C# Applikation geschrieben, welche sämtliche ‚low\_freq‘ REDD Daten in eine MySQL Datenbank importiert. Dafür wurden die Datensätze pro Kanal geparst und in 100.000er Schritten in die Datenbank importiert. Aus dem Zeitstempel im Unix-Epoch-Format wurde lokal der Tag und der Monat ausgerechnet um später danach gruppieren zu können. Zusätzlich wurde pro Kanal eine fortlaufende Id, in Abbildung 3 als ‚meterId‘ dargestellt, vergeben. [Zeichung konsistent machen]



Abbildung 3: Tabelle REDD mit Testdaten

Damit das Einfügen von 100.000 Datensätzen auf einmal funktioniert wurde die maximal erlaubte Paketgröße auf 160 MB erhöht [https://dev.mysql.com/doc/refman/5.7/en/packet-too-large.html]. Die Datenbank läuft auf dem lokalen Testsystem (i5 4690K @ 3.5GHz, 16GB Arbeitsspeicher, Windows 10 Pro) um einen möglichen Delay über das Netzwerk ausschließen zu können.

Nach dem Hochladen eines jeden Datensatzes wurde der Durchschnittsverbrauch pro Id, Monat und Tag abgefragt (Listing 2).

**select** **SQL\_NO\_CACHE** **avg**(**power**) **as** **power**, **day**, **month**, meterId

**from** redd

**group** **by** meterId **asc**, **month** **asc**, **day** **asc**;

Listing 2: Berechnung des Durchschnittsverbrauchs pro Meter, Tag und Monat

Durch die SQL\_NO\_CACHE Anweisung wird verhindert das das Ergebnis der Abfragen aus dem Cache zurückgeliefert werden, was einem realistischen Szenario entspricht. Die Abfrage wurde fünf Mal wiederholt ausgeführt und die jeweilige Zeit mittels der ‚System.Diagnostics.Stopwatch‘ Klasse gemessen. Die Messungen wurden in einem XML Dokument abgespeichert um dann mit Excel weiterverarbeitet werden zu können. Aus den 5 Messungen wurde der Median und Mittelwert berechnet, welche dieselben Schwankungen zeigen. Daher wurde auf die Darstellung des Mittelwerts verzichtet.

Abbildung 4: Dauer Berechnung des Mittelwerts auf der MySQL Datenbank

Abbildung 4 zeigt, dass die Dauer der Berechnung des Durchschnittsverbrauches in linearem Zusammenhang zur Anzahl der Datensätze in der Tabelle ist. Die teilweise starken Schwankungen des Medianes lassen sich durch Hintergrundprozesse wie Virenscanner und verschiedene Updatedienste erklären. Die Formel der Regressionsgerade liefert im Gegensatz zur y-Achse eine erwartete Dauer, bei der Abfrage auf X Datensätzen, in Millisekunden.

Abbildung 5: Dauer des Einfügens in die MySQL Datenbank

Auch das Einfügen neuer Datensätze steht, wie in Abbildung 5 zu sehen in linearem Zusammenhang mit der Anzahl der neu eingefügten Datensätze. Die Tabelle mit den Messungen ist im Anhang zu finden.

## Erwartete Datenmengen

Die Datenbank soll eine Sammlung von Messdaten aus verschiedenen Quellen sein. Im Moment sind vier verschiedene Datenquellen bekannt (Tabelle 7).

|  |  |  |
| --- | --- | --- |
| **Name** | **Anzahl** | **Kommentar** |
| REDD | 56M | bisher nur low\_freq |
| ADRES | 36M | 30 Haushalte je 1 Sensor, 2 Wochen, 1Hz |
| GREEND | 2.270M | 8 Häuser je 9 Sensoren, 1 Jahr, 1Hz |
| Salzburg AG | 182M / Jahr | 500.000 Haushalte 1 Messung pro Tag |
| **Summe** | 2.544M |  |

Tabelle 7: Erwartete Anzahl an Datensätzen

Diese Datensätze zusammen gerechnet beinhalten ungefähr 2.544 M Einträge. Rechnerisch würde eine Abfrage des Durchschnittswertes auf dem Testsystem ungefähr 40 Minuten dauern. Da moderne Datenbankserver allerdings um ein vielfaches performanter als ein Heimcomputer sind, ist dieser Wert für ein Livesystem nicht aussagekräftig. Es zeigt sich ebenfalls, dass der derzeitig verfügbare Testserver (Xeon E5-2620 @ 2GHz, 4GB Arbeitsspeicher), für einen Betrieb mit allen Datensätzen nicht geeignet ist. Mögliche Ansätze zur Erhöhung der Performance wären horizontale Skalierung mit einem SQL System oder vertikale Skalierung mit einem NoSQL System. Eine Lösung dieses Problems ist für das nächste Semester geplant.

## Schnittstellen

Um die Daten abfragen zu können wird eine REST Schnittstelle definiert. Das Datenaustauschformat JSON oder XML kann wie von einer REST Schnittstelle erwartet in der Abfrage eingestellt werden. Derzeit ist noch kein Framework für die REST API definiert. In der Schnittstelle ist die zentrale Klasse DataAccess (Listing 3).

*/\*\*  
 \* The central class to query the Database  
 \*/***public class** DataAccess {  
  
 **private final** ILdapPermissionManager **\_permissionManager**;  
 **private final** IDatabaseAccess **\_databaseAccess**;  
  
  
 **public** DataAccess(ILdapPermissionManager permissionManager,

DatabaseAccess databaseAccess){  
 **\_permissionManager** = permissionManager;  
 **\_databaseAccess** = databaseAccess;  
 }  
  
 */\*\*  
 \* Query the Database  
 \** ***@param query*** *the query to execute  
 \** ***@return*** *a QueryResult with either Data or a ErrorMessage.  
 \*/* **public** QueryResult QueryDataSource(QueryBase query){  
 **if**(!**\_permissionManager**.IsAllowedToAccess(  
 query.getUserContext(),  
 query.getDataSourceContext())){  
 **return** createNotAllowedResult();  
 }  
 **return \_databaseAccess**.QueryDatabase(query);  
 }  
  
 **private** QueryResult createNotAllowedResult(){  
 **return new** QueryResult(**false**,  
 **"Not allowed to access datasource"**,  
 QueryStatusCode.***Error***) {  
 @Override  
 **public** Object GetData() {  
 **return null**;  
 }  
 };  
 }  
}

Listing 3: Klasse für den Datenzugriff

Kommt eine Anfrage aus der REST Schnittstelle wird zuerst über den ILdapPermissionManager überprüft ob der User anhand seiner Rolle berechtigt ist auf die gewünschten Daten zuzugreifen. Ist das erlaubt wird die IDatabaseAccess aufgerufen, welche dann die Abfrage an die Datenbank stellt und ein QueryResult zurückliefert. Der Vorteil dieser Implementierung ist, dass der Datenbankzugriff völlig unabhängig vom Berechtigungssystem ist, und beliebig ausgetauscht werden kann.

Um eine neue Abfrage hinzuzufügen muss lediglich die Methode getQuery() der Klasse QueryBase implementiert werden (Listing 4).

Listing 4: Basisklasse um Abfragen zu definieren

*/\*\*  
 \* A Baseclass to query the database  
 \*/***public abstract class** QueryBase {  
 **private** IUserContext **\_userContext**;  
 **private** IDataSourceContext **\_dataSourceContext**;  
  
  
 **public** QueryBase(IUserContext userContext,

IDataSourceContext dataSourceContext){  
 **\_userContext** = userContext;  
 **\_dataSourceContext** = dataSourceContext;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *the user context of the query  
 \*/* **public** IUserContext getUserContext(){  
 **return \_userContext**;  
 }  
  
 */\*\*  
 \*  
 \** ***@return*** *the dataSourceContext of the query  
 \*/* **public** IDataSourceContext getDataSourceContext(){  
 **return \_dataSourceContext**;  
 }  
  
 */\*\*  
 \* Implement this method for the query.  
 \* It will get  
 \** ***@return*** *\*/* **public abstract** String getQuery();  
}

Die Schnittstellen und Klassen stellen ein Gerüst für die Implementierung dar und können sich im Projektverlauf noch ändern. Auf Darstellung der REST Schnittstelle wurde derzeit noch verzichtet, da noch nicht klar ist, wie genau die Authentifizierung darüber funktioniert und in welcher Form die Abfragen darüber ausgewählt werden können.

## Systemarchitektur

Um die Systemumgebung festzulegen, und vor allem die Software passgenau in die Softwarelandschaft des JRZ einfügen zu können, werden die Schnittstellen von Open-TC [7] und das Modell der bestehenden Datenbank, siehe Abschnitt 2.1.1, verwendet. Mit der Ausnahme des Betriebssystems werden Open Source Komponenten eingesetzt. Neben der freien Verfügbarkeit und dem Betrieb ohne Lizenzkosten sprechen weitere Aspekte, wie zum Beispiel Unabhängigkeit, Einsatz offener Standards und oft eine schnellere Behebung von Sicherheitslücken [18]. Die eingesetzten Komponenten sind wie angeführt:

* Hardware: für den Betrieb ist keine explizite Hardware vonnöten, vom JRZ wurde eine virtuelle Maschine im Bladecenter zur Verfügung gestellt: 2 Prozessoren Intel Xeon E5-2620 2GHz, 4 GB Hauptspeicher und 80 GB Plattenplatz. Die Identifikationsdaten des Produktivsystems sind:  
  Hostname: landsteiner.fh-salzburg.ac.at  
  IP-Adresse: 193.170.119.66  
  zugängliche Ports: 80 (?)
* Betriebssystem: auf Grund der größeren Erfahrung der Entwickler mit der Administration erfolgt der Betrieb auf einem Windows System (Windows Server 2012 R2).
* Java als Programmiersprache ist eine Vorgabe, da bestehende Software im Umfeld des JRZ bereits damit umgesetzt wurde.
* Die Entwicklung der Software erfolgt mit zwei unterschiedlichen Entwicklungsumgebungen: Eclipse Neon oder IntelliJ IDEA. Beide Entwicklungsumgebungen unterstützen die Entwicklung von Java Programmen, der jeweilige Einsatz erfolgt auf Grund der persönlichen Präferenzen der Entwickler.
* Als RDBMS wird MySQL eingesetzt, nach „Guide to Scaling Web Databases with MySQL Cluster“ [19] kann der zu erwartenden Menge von Schreibzugriffen (~500k/d) durch Verteilung auf einzelne Nodes begegnet werden. Sollte ein Cluster mehrerer Nodes zu Performanceeinbußen führen können einem Cluster einfach weitere Nodes hinzugefügt werden, eine Anpassung der Applikation ist in diesem Fall nicht notwendig.
* Als Datenbankdesigntool wird, passend zur Datenbank die MySQL Workbench 6.3.9 eingesetzt
* Der Entwurf des API sieht die Übergabe eines generischen Benutzerkontextes vor, damit ist die Anbindung von Open-TC oder einer anderen Benutzerverwaltung ohne Änderungen am API möglich.
* Die Schnittstelle zu Importmodulen ist einzig und allein die Datenbank, da hier das Schema vorgeschrieben ist, sind die Anforderungen an die Imoportmodule klar definiert und es bestehen keine programmseitigen Abhängigkeiten.
* Die öffentlichen Schnittstellen sind im Abschnitt 2.4 detailliert dargestellt.
* Aus den Anforderungen ergeben sich folgende Komponenten, deren Abhängigkeiten untereinander und Verbindungen zueinander, die im UML Diagramm (Abbildung 1) dargestellt werden:   
    
  Abbildung1 : Komponentenmodell

Der Übersicht halber wird die Komponente „Logger“, die jede der angeführten Komponenten verwendet, nicht in der Übersicht dargestellt.

# Ausblick

## Alternative Datenhaltung

Die JRZ-DB ist für aktive Anwendungen wie zum Beipiel Open-TC auf Basis einer MySQL Datenbank im Einsatz. Bestehende Anwendungen, wie zum Beispiel Smart Vis [20] zeigen, dass die Verarbeitungsgeschwindigkeit für Messdatenmengen in der Größe von akademischen und Forschungsquellen ausreichend ist. Die im Abschnitt 2.3.3 angeführten Zahlen führen, insbesondere bei einem zentralen und dauerhaften Betrieb, zu Datensatzmengen, die Überlegungen bezüglich einer alternativen Datenhaltung notwendig machen. Auch wenn es keine einheitliche Definition für den Begriff Big Data gibt [21], ergibt die Anzahl der Messdatensätze eine Datenmenge, die im praktischen Betrieb möglicherweise zu verlängerten Antwortzeiten führen wird.

Die vorhandene Hardware und die bisher durchgeführten Tests lassen noch keine konkreten Aussagen über die Verarbeitungsgeschwindigkeit von Messdatenmengen in der Größenordnung von ca. 500.000 Datensätze pro Tag, wie sie von zum Beispiel der Salzburg AG [21] zu erwarten sind, zu. Abhängig von den Reaktionszeiten werden folgende Vorgehensweisen in Betracht gezogen:

* MySQL unterstützt die Fragmentierung einzelner Tabellen, sowohl lokal (partitioning) als auch verteilt (sharding). Nach „Guide to Scaling Web Databases with MySQL Cluster“ [20] kann der zu erwartenden Menge von Schreibzugriffen durch Verteilung auf einzelne Nodes begegnet werden. Sollte ein Cluster mehrerer Nodes zu Performanceeinbußen führen können einem Cluster einfach weitere Nodes hinzugefügt werden, eine Anpassung der Applikation ist in diesem Fall nicht notwendig, über die Administration des MySQL Clusters wird das Partitionierung transparent für die Anwendung durchgeführt.
* Fusco et al. schlagen in [21] einen dualen Betrieb von RDBMS und einer NoSQL Datenverwaltung vor. Die Messwertedatenpakete werden in unterschiedlicher Granularität, zum Beispiel Rohdaten, Messdaten aggregiert nach Smartmeter, Zeitraum und vorverarbeitet, zum Beispiel Durchschnittsverbrauch über einen bestimmten Zeitraum, abgelegt. Ziel dieses Ansatzes ist es auch Vorhersagen über den zukünftigen Verbrauch zu tätigen. Jene Daten die Messwerte betreffen werden in einer separaten Komponente verwaltet, dadurch soll ermöglicht werden die Datenhaltung mittelfristig ersetzen zu können. Details siehe sin im Abschnitt „Systemarchitektur“ angeführt.

Ein dezentraler Ansatz, wie zum Beispiel das COUGAR Sensornetzwerk [Referenz Cougar], als Alternative zu einer zentralen Datenbank bietet zwar den Vorteil, den einen, zentralen Angriffspunkt zu vermeiden, Messdaten hingegen ausschließlich ad hoc auszulesen widerspricht den Regelungen des ElWOG und scheidet daher aus den Alternativen aus.

-------------------------------------------------------------------------------------------------------------------- alles nach dieser Linie, bis zum Anhang gehört in die obere Struktur gepackt. ---------------------------------------------------------------------------------------------------------------------

# Recherche

Die folgenden Recherchen sind notwendiger Teil des Projektes für die Informationsbeschaffung sowie der Prävention von eventuell doppelt gemachter Arbeit.

## ER-Modell

Um ein geeignetes Datenbankmodell zur Verfügung stellen zu können, müssen vorhandene Datenbankmodelle genauer betrachtet werden.

Das Datenmodell der JRZ-DB wird analysiert und auf Erweiterungsnotwendigkeiten untersucht.

## Analyse von Datenbankanforderungen

Es gilt herauszufinden, welches Datenbankmodell (DBM) für die Umsetzung des Projektes ideal ist. Um ein geeignetes DBM zu finden, wurden verschiedene Typen wie RDBMS, NoSQL und Hadoop genauer betrachtet. Weitere Datenbankmodelle ansehen und auflisten. Für die Ablage und den Zugriff auf Messdaten bietet jede dieser Architekturen Vorteile, die gegeneinander abgewogen werden. RDBMS stellen mit SQL eine Programmiersprache der vierten Generation zur Abfrage zur Verfügung, NoSQL Datenbanken, wie zum Beispiel MongoDB[nicht weil die flexibilität auf Kosten der Kompatibilität geht], sind flexibel bei der Erweiterung um zusätzliche Messwertarten und das Hadoop Rahmenwerk verwaltet Daten im Bereich der zu erwartenden Messdatenmengen.

Für die Analyse wird folgendermaßen Vorgegangen:

* Analyse des bestehenden Datenbankmodells (JRZ-DB)
* Performanceanalyse der zu testenden DBM sowie
* Einlesen in relevante Produkte

*Beim Meeting mit der Projektbetreuung am 27.1.2017 wurde der Hinweis gegeben, dass Hadoop für dieses Projekt interessant sein könnte und wir uns das näher ansehen sollten. Zudem wurde erwähnt, dass es von Hortonworks eine Sandbox gibt, auf der ein fertig konfiguriertes Hadoop System mit unterschiedlichsten Tools verfügbar ist.*

*Ein großer Teil dieses Arbeitspaketes bestand darin, sich in Hadoop einzuarbeiten, Tutorials durchzumachen und erste Erfahrungen mit Big Data Systemen zu machen.*

*Bei den Gesprächen mit der Projektbetreuung hat sich allerdings auch herausgestellt, dass ein Weiterverwenden der JRZ-DB wünschenswert ist, da es bereits einiges an Software dafür gibt.*

Des Weiteren spricht dafür, dass es eine BAC1 Gruppe gibt, welche sich mit dem Datenimport von frei verfügbaren Smartmeter Datensätzen beschäftigt hat. Der aktuelle Stand dieser Arbeit ist nach Aussage der Gruppe soweit, dass Daten normiert importiert werden können, die Performance allerdings noch nicht optimal ist. *Das ist zum aktuellen Zeitpunkt noch nicht verifiziert, allerdings gehen wir davon aus, dass dieses Projekt nach Performanceoptimierung einsatzbereit ist. Da die verwendetete Programmiersprache allerdings C# ist, ist auf jeden Fall eine Portierung notwendig, da wir uns für das Projekt auf Java geeinigt haben.*

### Analyse JRZ-DB

Auf Grund der Analyse des ER-Modells der JRZ-DB können die Tabellen in zwei Gruppen eingeteilt werden: Stammdaten, wie zum Beispiel customer\_data oder meter\_management, deren Anzahl verwalteter Datensätze von einem RDBMS ohne Performanceeinbußen verwaltet werden kann. Zum Beispiel verwaltet der Projektpartner Salzburg AG ca. 500.000 intelligente Zähler [Referenz: Salzburg AG Geschäftsbericht 2015 https://www.google.at/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwioxJnvzK7SAhXiCJoKHfubAM0QFggcMAA&url=https%3A%2F%2Fwww.salzburg-ag.at%2F%3FeID%3Ddownload%26uid%3D1825&usg=AFQjCNHc-oFEdGUSo3qC\_JXquXgB6QzpVg&cad=rja]. Andererseits werden in der Tabelle ‚meter\_data‘ die Bewegungsdaten abgelegt, hier sind durch die viertelstündliche Auslesung von jedem dieser Smart Meter täglich 96 Tupel abzulegen. Hochgerechnet auf die ca. 500.000 Smart Meter, welche durch die Salzburg AG versorgt werden, ergeben sich pro Monat 1,44 Milliarden Sätze. Das Datenmodell der JRZ-DB gilt als zu erfüllende Vorgabe, daher wird dieses in der Grundform eingesetzt, bezüglich der Abbildung der performancekritischen Tabelle meter\_data erfolgen weitere Untersuchungen mit dem Hadoop Rahmenwerk.

### Performanceanalyse

Für erste Tests wurde ein Teil der REDD ‚low\_freq‘ Daten verwendet. Diese wurde in verschiedene Datenbanken importiert und es wurden darauf Abfragen ausgeführt. Ziel dieser Analyse war es ein Gefühl zu bekommen, wie sich die Performance mit Zunahme an Daten verhält und ob eine SQL Datenbank überhaupt in Frage kommen kann.

## Analyse vorhandener Software

Da das Projekt ein Teilprojekt eines Größeren ist, galt es festzustellen, ob schon bestehende Software für das Projekt verwendet werden kann. Somit wird redundante Arbeit vermieden und der Fokus auf die noch benötigen Dinge gesetzt.

## Rollendefinition

Mit SmartValAPI wird der Zugriff auf sensible Daten verwaltet, daher ist die Schutz des Zugriffs unerlässlich. Auf Aspekte der Datensicherheit wie physischer Zugang zum Datenbank beziehungsweise Applikationsserver geht der Abschnitt „Installation“ näher ein, dieser Abschnitt beleuchtet den Zugriff über Rollen und legt die Rollendefinition fest.

Die Kernaufgaben sind wie folgt:

* Rollen identifizieren und definieren.
* Unterschiede in fachlichen und technischen Anforderungen in notwendigen Rollen überleiten.
* Verbindung zum Code, beziehungsweise Funktionen herstellen.
* Rechtliche Umgebung einbeziehen.

# Umsetzung

In dieser Sektion wird beschrieben, wie bisher die Pakete Systemarchitektur, ER-Modell und die Rollendefinition umgesetzt wird.

## Systemarchitektur

Um die Systemumgebung festzulegen, und vor allem die Software passgenau in die Softwarelandschaft des JRZ einfügen zu können, werden die bestehenden System wie zum Beispiel Smart Viz, die in Entstehung befindlichen Zugriffsysteme (BAC1 Gruppe – OpenTC) und Importprogramme (BAC1 Gruppe Oberluggauer und Co). Was war mit Open-Nes? In die Überlegungen werden weitrs etwaige Kosten für Lizenzen und andererseits Sicherheitsaspekte einbezogen [18]. Funktionen nach Aufruf und Zusammengehörigkeit gruppieren.

## ER-Modell

~~Beim Festschreiben des geeigneten Datenbankmodells wurde wie unterhalb beschrieben vorgegangen:~~

* ~~Recherche nach Datenmodellen, die bereits abseits der JRZ-DB im Einsatz sind, wie zum Beispiel COSEM.~~
* ~~Feststellen der Wertemenge die SmartMeter zur Verfügung stellen und herausarbeiten welche davon gespeichert werden.~~
* ~~Kontaktaufnahme mit österreichischen Energieversorgern bezüglich der Werte von Interesse.~~
* ~~Analyse der Usecases von Österreichs Energie, ergibt vor allem Daten bezüglich der Steuerung des Smart Meter als solches, und nicht der inhaltlichen Bedeutung der übertragenen Messdaten.~~
* ~~Erhebung der rechtlichen Rahmenbedingungen und sich daraus ergebende Einschränkungen.~~
* ~~Prüfung auf Verwendbarkeit des Datenmodells in weiteren Domänen wie zum Beispiel: Gas, Wärme und Wasser.~~
* ~~Festschreiben des Datenmodells.~~

## Analyse der Datenmodelle

### Hadoop Tests

Für die Tests wurde eine virtuelle Maschine mit der ‚Hortonworks Hadoop Sandbox‘ aufgesetzt. Der Maschine wurden alle Cores des Hosts[[1]](#footnote-1) sowie 8GB Arbeitsspeicher zur Verfügung gestellt.

### MySQL Tests

Für die Tests wurde ein MySQL Server auf dem o.g. Testsystem aufgesetzt. Im Gegensatz zu den Hadoop Tests allerdings direkt auf dem Host-Betriebssystem.

### Testdaten

Als Testdaten wurden REDD Daten eines Hauses importiert. Die Abfrage, die darauf abgesetzt wurde, ergibt den Durchschnittsverbrauch pro Tag pro Kanal.

Die Daten liegen in CSV Dateien mit folgendem Format vor.

|  |  |
| --- | --- |
| timestamp | power |
| int | double |

Tabelle 1: Datenformat CSV

Der Timestamp lässt sich mit FROM\_UNIXTIME(timestamp) in ein Datum umwandeln womit gerechnet werden kann.

Folgende Abfrage liefert den Durchschnittsverbrauch gruppiert nach Kanal, Monat und Tag. Die Dauer ist unser Performance Index.



Abbildung 1: Abfrage Durchschnitt pro Tag

Es wurden Schritt für Schritt Kanäle hinzugefügt um einen wachsenden Datensatz zu simulieren.

# Ergebnisse

Bisherige Ergebnisse in Bezug Systemarchitektur, ER- Modell, Analyse vorhandener Datensätze sowie vorhandener Software, Recherche noch benötigter Software beziehungsweise Funktion und momentaner Rollendefinition werden nun vorgestellt.

## Systemarchitektur

* Hardware: für den Betrieb ist keine explizite Hardware vonnöten, vom JRZ wurde eine virtuelle Maschine im Bladecenter zur Verfügung gestellt.
* Betriebssystem: auf Grund der größeren Erfahrung der Entwickler mit der Administration erfolgt die Implementierung auf einem Windows System (Windows Server 2012 R2),
* Programmiersprache: bestehende Software wurde im Umfeld des JRZ in Java implementiert, ebenso werden laufenden Projekte in Java programmiert, die Erfahrung der Umsetzenden reicht aus, um die Anforderungen umzusetzen. Im Sinne einer einfachen Übergabe, Weiterführung und Wartung fällt die Entscheidung, dieses Projekt in Java zu implementieren.
* Datenbank: neben unterschiedlichen RDBMS (Oracle Database Server, MySQL) wurden NoSQL Datenbankensysteme untersucht (MongoDB, Hadoop). Im Sinne der Integration in die bestehende Softwarelandschaft wird MySQL eingesetzt. Ziel des Projektes ist eine Integrationsdatenbank, was dazu führt, dass sehr große Datenmengen verwaltet werden können müssen, und der Zugriff in akzeptabler Zeit erfolgt. MySQL unterstützt lokale (partitioning) und verteilte (sharding) Fragmentierung. Nach dem „Guide to Scaling Web Databases with MySQL Cluster“ [19] kann der zu erwartenden Menge von Schreibzugriffen (~500k/d) durch Verteilung auf einzelne Nodes begegnet werden. Sollte ein Cluster mehrerer Nodes zu Performanceeinbußen führen können einem Cluster einfach weitere Nodes hinzugefügt werden, eine Anpassung der Applikation ist in diesem Fall nicht notwendig, über die Administration des MySQL Clusters wird das Partitionierung transparent für die Anwendung durchgeführt. Details dazu sind im Abschnitt „Datenbankzugriff“ erläutert.
* Datenbankdesigntool: untersucht wurden Oracle SQL Developer Data Modeler 4.1.5 und die MySQL Workbench 6.3.9. Für den graphischen Entwurf des Entity-Relation-Modells bieten beide Tools Unterstützung. Modeler wie Workbench generieren aus dem ER-Modell sowohl die graphische Übersicht der Tabellen und Schlüssel, als auch Generation der Skriptdateien zur Anlage der Tabellen, Indizes, Einschränkungen bezüglich referentieller Integrität und die automatische Vergabe eindeutiger Schlüssel. Da als Datenbanksystem MySQL eingesetzt wird, liegt es nahe das Designtool vom gleichen Hersteller einzusetzen und damit Kompatibilitätsprobleme beziehungsweise Nachbearbeitungen zu vermeiden.
* Schnittstellen zu BAC1 (Autorisierung): der bisherige Entwurf des API ist in einer generischen Form implementiert, sodass einerseits ein Austausch des Autorisierungsmoduls möglich ist (zum Beispiel in einem ersten Schritt realisiert als Benutzername/Passwort Zugang). Eine Abhängigkeit zum Datenmodell wurde identifiziert: die Form der Benutzerkennung des externen Rollensystems muss in den Tabellen untergebracht werden könne (Details siehe Abschnitt Datenmodell).
* Schnittstellen zu BAC1 (Importmodule): gemeinsame Basis für die API und Importmodule ist das Datenmodell (Details zu notwendigen Erweiterungen siehe im Abschnitt Datenmodell), programmtechnisch wurden keine Überschneidungen identifiziert.
* Die Anbindung und die Veröffentlichung der Schnittstellen: <wir haben noch nicht festgelegt wie das API aufgerufen werden kann> Webservice? Datenstrukturen?
* Aus den Anforderungen ergeben sich folgende Komponenten, deren Abhängigkeiten untereinander und Verbindungen zueinander, die im UML Diagramm (Abbildung 1) dargestellt werden:



Abbildung 2: Komponentenmodell SmartValAPI

Die Details zu den Komponenten und die exportierten Methoden werden im Abschnitt „API Funktionen“ beschrieben. Der Übersicht halber wird die Komponente „Logger“, die jede der angeführten Komponenten verwendet, nicht in der Übersicht dargestellt.

## ER Modell

Hier werden die Ergebnisse des zu verwendenden Datenbankmodells gezeigt.

### Analyse bestehendes Datenmodell

### Anforderungen von Energieversorgern und Netzbetreibern

Anfragen an Ebner Strom GmbH, Energie AG, Energie Steiermark, EVN AG und Netz Niederösterreich GmbH, Linz AG, Salzburg Netz GmbH, TINETZ – Tiroler Netze GmbH, Vorarlberger Energienetze GmbH und Wien Energie ergeben:

* Verweis auf das ElWOG: die erheb baren Daten sind klar definiert eine darüberhinausgehende Erfassung von Messwerten ist nicht zulässig.
* Verweis auf die laufende Ausschreibung der Smart Meter Geräte, hier werden die Messwerte in Pflicht- und optionale Werte eingeteilt.
* Die im Datenmodell vorhandene Netzfrequenz ist auf Grund der Rückmeldungen nicht auswertungsrelevant auf Basis einzelner Smart Meter nicht interessant.
* Keine Rückmeldung, beziehungsweise nur telefonisch, wobei sich aus letzterer keine weiteren Datenfelder ergeben.
* Hervorzuheben sind die Antworten von Salzburg Netz GmbH und Vorarlberger Energienetze GmbH, bezüglich der Messwerte steht für den Netzbetreiber, neben den bereits vorhandenen Werten stehen der Winkel zwischen Spannung und Strom, und die daraus resultierende Blindleistung im Fokus.

Im Gegenzug dazu kann auf die Frequenz verzichtet werden, dieses Datenfeld verbleibt auf Kompatibilitätsgründen im ERM.

### Analyse der Usecases

### Weitere Datenmodelle

Fusco et al. [20] schlagen einen dualen Betrieb von RDBMS und einer NoSQL Datenverwaltung vor. Die Messwertedatenpakete werden in unterschiedlicher Granularität, zum Beispiel Rohdaten, Messdaten aggregiert nach Smartmeter, Zeitraum und vorverarbeitet zum Beispiel Durchschnittsverbrauch über einen bestimmten Zeitraum abgelegt.

Ziel dieses Ansatzes ist es auch Vorhersagen über den zukünftigen Verbrauch zu tätigen. Jene Daten die Messwerte betreffen werden in einer separaten Komponente verwaltet, dadurch soll ermöglicht werden die Datenhaltung mittelfristig ersetzen zu können (Details siehe Abschnitt Systemarchitektur).

Ein dezentraler Ansatz, wie zum Beispiel das COUGAR Sensornetzwerk [Referenz Cougar], als Alternative zu einer zentralen Datenbank bietet zwar den Vorteil, einen zentralen Angriffspunkt zu vermeiden, Messdaten hingegen ausschließlich ad hoc auszulesen widerspricht den Regelungen des ElWOG und scheidet damit aus.

### Rechtliche Rahmenbedingungen und daraus resultierende Einschränkungen der möglichen Messwerte

Die Erfassung, Übertragung und Speicherung von Smart Meter Messdaten wird in vier Richtlinien geregelt:

* ElWOG
* GDPR
* IMA-VO
* DAVID-VO

Geregelt werden einerseits Mindestanforderungen an Smart Meter, andererseits die Inhalte und die Frequenzen, mit denen die Werte ausgelesen werden dürfen. Im ElWOG werden in §84 dem Verbraucher die Daten bezüglich des „Verbrauchs der über ein intelligentes Messgerät gemessen wird“ zeitnah zur Verfügung zu stellen. Es erfolgt keine genauere Definition, welche Daten das im Detail sind, lediglich die Frequenzen, mit denen ausgelesen wird, werden festgelegt, Details dazu, siehe Abschnitt Rollendefinitionen.

Einer der Hauptgründe für die Reglementierung ist der Schutz der Privatsphäre [21]. Bezüglich der auslesbaren Daten legt die [11] jene Daten fest, die von einem Smartmeter übertragen werden müssen. Die IMA-VO und DAVID-VO befassen sich mit der Einführung der der intelligenten Zähler und der Weitergabe der ausgelesenen Daten und haben keinen Einfluss auf das Datenmodell.

### Das COSEM Modell

[2] versucht hier einen Standard zu etablieren. Nachdem die Salzburg AG als Partner des Projektes in den Ausschreibungen für Smart Meter die Kommunikation über DLSM/COSEM als Muss-Kriterium festlegt, wird in der Folge dieses Protokoll in Verbindung mit den Anforderungen des Lastenheftes von Österreichs Energie [11] als Ausgangspunkt für die Festlegung der Datenbank herangezogen.

### ER-Modell Festlegung

Ausgangsbasis ist die JRZ-DB, bei der Erweiterung wurde auf die Kompatibilität zu bestehenden Programmen geachtet, um diese Applikationen ohne Anpassung auch weiterhin gegen die gleiche Datenbank betreiben zu können. Folgendes Diagramm gibt einen Überblick über die Entitäten:



Abbildung 3: ER-Modell Entitäten

Aus dem existierenden Modell werden die bestehenden Entitäten übernommen, für die zusätzlichen Anforderungen ergeben sich folgende Erweiterungen:

Zusätzliche Entitäten:

* userAdmin: die Benutzerverwaltung wird als eigene Komponente eingebunden, um die Kopplung lose ausführen zu können wird im System lediglich der Schlüssel (der Benutzername, eine LDAP ID, ein etwaiges anders identifizierendes Merkmal) hinterlegt.

Zusätzliche Beziehungen:

* istSystembenutzer (userAdmin – customer): optionale Erweiterung zum customer, ermöglicht die Ablage einer externen Benutzerkennung.
* berät\_oder\_verwaltet (customer – meter\_management): verbindet m customer mit n meter\_management, Zweck ist die Abbildung von Verbinungen wie zu, Beispiel: Netzbetreiber versorgt Meter, Energieberater berät Eigentümer von Meter.
* ist\_Kunde\_von: ermöglicht die hierachische Verbindung von Kunden, zum Beispiel Energeversorger mit Kunden.

Zusätzliche Attribute:

* in meter\_data: reactive\_P1, reactive\_P2, reative\_P3: Blindleistungsanteil aufgeteilt nach Phase, sofern nur gesamt übermittelt in P1, wenn nicht ausgelesen: 0.
* phi\_P1, phi\_P2, phi\_P3: Phasenwinkel von Strom und Spannung, je Phase, wenn ausgelesen,

Aus den Typen der Beziehungen ergeben sich folgende Tabellen:



Abbildung 4: Datenbankbeziehungen

Alternative für meter\_data: da die Tabelle das maximale Set an Daten abbilden kann entstehen eventuell einige Tupel mit Null-Werten, sofern Smart Meter nicht alle Felder auslesen können. Die Alternative besteht in der Definition einer Tabelle die je Tupel einen Messwerttyp (bedingt eine weitere Definitionstabelle für die zugelassenen Werttypen) und einen Messwert.

Der Messwerttyp ist Teil des primären Schlüssels. Die Vorteile der Vermeidung von Nullwerten und der flexiblen Einführung neuer Werttypen, zum Beispiel auch Gas, Wasser oder Wärme, werden durch die Vervielfachung der Einträge einerseits und des Verlustes der Kompatibilität mit den bestehenden Softwarepaketen erkauft.

Formate der Datenfelder:

Aus der Tabellendefinition können die Formate der jeweiligen Felder ermittelt werden, inhaltlich wird festgelegt, dass in **meter\_management:**  meterId entsprechend der OBIS Identifikation befüllt wird (Details siehe Abschnitt „API Funktionen“).

## Analyse möglicher Datensätze

Es haben sich bei den Tests folgende Ergebnisse errechnen lassen:

### MySQL Datenbank

Zuerst wird die Dauer des Datenimports gemessen.

|  |  |  |
| --- | --- | --- |
| **Import Dauer** | | |
| **Vorhanden** | **Hinzugefügt** | **Dauer [s]** |
| - | 1.561.660 | 27 |
| 1.561.660 | 1.561.660 | 28 |
| 3.123.320 | 745.878 | 10 |
| 3.869.198 | 745.878 | 11 |
| 4.615.076 | 745.878 | 10 |
| 5.360.954 | 3.729.390 | 62 |
| 9.090.344 |  |  |

Tabelle 2: Dauer des Datenimports

Abbildung 5: Dauer des Datenimports

Hierbei ist zu sehen, dass die Dauer unabhängig von den bereits vorhandenen Datensätzen linear abhängig zu der Menge der Importierten Datensätze ist.

Als nächsten Schritt wird die Dauer des Query aus *Abbildung 1: Abfrage Durchschnitt pro Tag*gemessen.

|  |  |
| --- | --- |
| **Berechnung des Durchschnitts** | |
| **Anzahl Zeilen** | **Dauer** |
| 1.561.660 | 6 |
| 3.123.320 | 16 |
| 3.869.198 | 19 |
| 4.615.076 | 23 |
| 5.360.954 | 27 |
| 9.090.344 | 42 |

Tabelle 3: Dauer Berechnung des Durchschnittsverbrauchs

Abbildung 6: Dauer Berechnung des Durchschnittsverbrauchs

Abbildung 6 zeigt, dass die auch die Dauer der Abfrage linear abhängig zur Anzahl der vorhandenen Datensätze ist.

Abschließend wurden Tag und Monat aus dem Timestamp fix in die Tabelle geschrieben und mit einem Index versehen. Dadurch konnte die Berechnungsdauer bei ~ 10 Millionen Datensätzen von 42 Sekunden auf 8 reduziert werden.

### Hadoop Datenbank

Erste Tests haben gezeigt, dass die Abfragen mit HIVE annähernd gleich funktionieren wie mit SQL, die Performance allerdings auf dem Testsystem schlechter ist. Aufgrund der zufriedenstellenden Ergebnisse mit SQL wurde auf ebenso detaillierte Tests verzichtet. Die Performance lässt sich durch Hinzufügen von mehreren Nodes nahezu beliebig steigern.

### Zusammenfassung der Datenbankanalyse

Beim Import der Datensätze fällt auf, dass die Anzahl der bereits vorhandenen Daten keine Auswirkung auf die Dauer des Vorgangs hat, was uns entgegenkommt.

Die Dauer einer Berechnung des Durchschnitts nimmt proportional mit der Anzahl der Datensätze zu. Der REDD Datensatz mit niedriger Auflösung beinhaltet ungefähr 52 Millionen Datensätze.

Hier würde die Abfrage hochgerechnet 210 Sekunden dauern. Es hat sich aber gezeigt, dass sich durch die geschickte Verwendung von Indizes, die Performance des getesteten MySQL System sehr stark steigern lässt. Aus diesem Grund, und auch um die Kompatibilität zu den bestehenden Applikationen zu wahren, wird für dieses Projekt bis auf weiteres ein Relationales Datenbanksystem verwendet.

Falls sich im weiteren Verlauf des Projekts herausstellen sollte, dass eine SQL Datenbank nicht ausreichend ist, kann eine Hybrid Lösung angestrebt werden.

Hier wäre vorstellbar, dass die ganze Meterverwaltung weiterhin in einer SQL Datenbank verbleibt, die Messdaten allerdings in Hadoop abgespeichert werden.

## Analyse vorhandener Software

Für die Analyse bereits vorhandener Software wurde folgendermaßen vorgegangen:

* Abfragen der einzelnen Gruppen, welche Projekte verfolgt werden und welche Dokumente oder Projekte denen zur Verfügung stehen.
* Einschätzen, ob eines dieser Programme oder Projekte verwendet werden kann.
* Einholung der Dokumentation und des Projektes bei Interesse an einem Projekt.

Abfragen der einzelnen Gruppen, welche Projekte verfolgt werden und welche Dokumente oder Projekte denen zur Verfügung stehen.

* + Eine BAC Gruppe hat ein vielversprechendes Projekt (OpenTC). Hierbei handelt es sich um eine Software, die Rollenbasiertes Zugreifen auf Dokumente zulässt.
  + Eine weitere Gruppe wurde ausgeforscht, welche interessante Dokumentation hat. (BAC 1 Gruppe Oberluggauer und Co)

Einschätzen, ob eines dieser Programme oder Projekte verwendet werden kann:

* + Das BAC Projekt OpenTC könnte für unseren rollenbasierten Zugriff auf die Datenbank umgeschrieben werden.
  + Informationen der weiteren Gruppe (BAC1 Gruppe Oberluggauer und Co).

Einholung der Dokumentation und des Projektes bei Interesse an einem Projekt:

* + Die Dokumentation sowie das ganze Projekt konnte organisiert werden für Recherche.
  + Die Informationen der weiteren Gruppe. (BAC1 Gruppe Oberluggauer und Co)

## Rollendefinition

Folgende Rollen wurden identifiziert und mit den im Use-Case-Diagramm angeführten Anwendungsfällen verbunden, die in Klammern angeführten Attribute geben die grundlegende Ansiedlung der Personen, die die Rollen bekleiden.

Administrator (technisch): Stammdatenwartung (Meter Anlage und abhängige Daten),

Energieversorger (fachlich): jener, der den Verbraucher mit Energie versorgt.

Energieberater (fachlich): über die ausdrückliche Zustimmung des Eigentümers befugte Person.

Eigentümer (fachlich): derjenige dessen Verbrauch mit dem Smartmeter gemessen wird.

Importeur (technisch): Meterdaten in die Datenbank importieren, das Smart Meter muss bereits im System bestehen.



Abbildung 7: Rollendefinition - UseCase Diagramm

# Weitere Schritte

# Literaturverzeichnis

|  |  |
| --- | --- |
| [1] | Das Europäische Parlament und der Rat der Europäischen Union, „Richtlinie 2009/72/EG des Europäischen Parlaments und des Rates vom 13. Juli 2009,“ 13 Juli 2009. [Online]. Available: http: //publications.europa.eu/resource/celex/32009R0713. [Zugriff am 19 Februar 2017]. |
| [2] | K. D. Craemer und G. Deconinck, „Analysis of state-of-the-art smart metering communication standards,“ in *Proceedings of the 5th young researchers symposium*, 2010. |
| [3] | J. Z. Kolter und J. Johnson, „REDD: A public data set for energy disaggregation research,“ in *Workshop on Data Mining Applications in Sustainability (SIGKDD),*, San Diego, CA, 2011. |
| [4] | W. K. Jack Kelly, „UK-DALE: A dataset recording UK Domestic Appliance-Level Electricity demand and whole-house demand,“ 2014. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.750.4515&rep=rep1&type=pdf. [Zugriff am 01 03 2017]. |
| [5] | Alfred Einfalt, et al., „ADRES-Concept: Konzeptentwicklung für ADRES-Autonome Dezentrale Regenerative EnergieSysteme,“ TU Wien and Austiran Institute of Technology and Austian Power Grid, Wien, 2012. |
| [6] | Andreas Monacchi et al., „GREEND: An energy consumption dataset of households in Italy and Austria,“ in *IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2014. |
| [7] | C. Bellucci, A.-M. Oberluggauer und M. Tschuchnig, „Untersuchung unterschiedlicher Referenzdatensätze im Energiebereich,“ 2017. |
| [8] | W. F. T. H. Michael Egger, „Rollenbasierter LDAP Zugriff TODO TODO TODO,“ 2016. |
| [9] | *Elektrizitätswirtschafts- und -organisationsgesetz, Fassung vom 01.03.2017,* Wien, 2017. |
| [10] | Oesterreichs Energie, „Smart Metering Use-Cases,“ 14 Dezember 2015. [Online]. Available: http://oesterreichsenergie.at/branche/stromnetze/smart-meter-use-cases.html?file=files/oesterreichsenergie.at/Downloads%20Netze/Smart%20Meter/Oesterreich%20Use%20Cases%20Smart%20Metering\_14122015\_Version\_1-1.pdf. [Zugriff am 19 Februar 2017]. |
| [11] | Oesterreichs Energie, „Lastenheft Smart Meter,“ 01 Juli 2013. [Online]. Available: http://oesterreichsenergie.at/branche/stromnetze/lastenheft-smart-meter.html?file=files/oesterreichsenergie.at/Downloads%20Netze/Smart%20Meter/Lastenheft\_SmartMeter\_1\_0.pdf. [Zugriff am 19 Februar 2017]. |
| [12] | Das Europäische Parlament und der Rat der Europäischen Union, „Verordnung des Europäischen Parlaments und zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten und zum freien Datenverkehr,“ 11 06 2015. [Online]. Available: http://data.consilium.europa.eu/doc/document/ST-9565-2015-INIT/de/pdf. [Zugriff am 28 02 2017]. |
| [13] | „Intelligente Messgeräte-AnforderungsVO,“ 25 10 2011. [Online]. Available: https://www.e-control.at/documents/20903/-/-/20a992e6-d11f-48b8-aef9-8e5d66f284c1. [Zugriff am 16 02 2017]. |
| [14] | „Datenformat- und VerbrauchsinformationsdarstellungsVO,“ 2012. [Online]. Available: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=20007999. [Zugriff am 14 02 2017]. |
| [15] | ETSI, „Open Smart Grid Protocol (OSGP),“ 01 Januar 2012. [Online]. Available: http://www.etsi.org/deliver/etsi\_gs/OSG/001\_099/001/01.01.01\_60/gs\_osg001v010101p.pdf. [Zugriff am 19 Februar 2017]. |
| [16] | Ferraiolo, David and Cugini, Janet and Kuhn, D Richard, „Role-based access control (RBAC): Features and motivations,“ in *Proceedings of 11th annual computer security application conference*, New Orleans, 1995. |
| [17] | „Gesamte Rechtsvorschrift für Datenschutzgesetz 2000,“ [Online]. Available: https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10001597. |
| [18] | *IT-Grundschutz-Profil für Open-Source-Software (GSProOSS),* 2010. |
| [19] | Oracle, „Guide to Scaling Web Databases with MySQL Cluster,“ 1970 Januar 01. [Online]. Available: https://www.mysql.de/why-mysql/white-papers/guide-to-scaling-web-databases-with-mysql-cluster/. [Zugriff am 19 Februar 2017]. |
| [20] | F. Fusco, U. Fischer, V. Lonij, P. Pompey, J.-B. Fiot, B. Chen, Y. Gkoufas und M. Sinn, „Data Management System for Energy Analytics and its Application to Forecasting,“ in *EDBT/ICDT Workshops*, Bordeaux, 2016. |
| [21] | S. W. M. Lisovich, „Privacy concerns in upcoming residental and commercial demand-response systems,“ in *Proc. of the Clemson University Power Systems Converence*, Clemson, SC, 2008. |
| [22] | B. f. S. i. d. IT, „BSI TR-03109 Technische Vorgaben für intelligente Messsysteme und deren sicherer Betrieb,“ 18 März 2013. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/TR03109/TR-03109-1\_Anlage\_Feinspezifikation\_Drahtgebundene\_LMN-Schnittstelle\_Teilb.pdf?\_\_blob=publicationFile. [Zugriff am 19 Februar 2017]. |
| [23] | J. G. P. S. Phillipe Bonnet, „Towards sensor database systems,“ in *International Converence on Mobile Data Management*, Berlin, 2001. |
| [24] | Das Europäische Parlament und der Rat der Europäischen Union, „Energieeffizenz und Energiedienstleistungen und zur Aufhebung der Richtlinie 93/76/EWG des Rates,“ 05 April 2006. [Online]. Available: eur-lex.europa.eu/legal-content/DE/ALL/?uri=CELEX%3A32006L0032. [Zugriff am 19 Februar 2017]. |

# Anhang

## SQL Messungen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Anzahl Datensätze Gesamt** | **Anzahl Datensätze Neu** | **Median Query** | **Mittelwert Dauer Query** | **Mittelwerte Einfügen** |
| 1.561.660 | 1.561.660 | 3.046 | 2.912 | 29.269 |
| 3.123.320 | 1.561.660 | 2.786 | 4.355 | 30.764 |
| 3.869.198 | 745.878 | 3.449 | 4.797 | 15.450 |
| 4.615.076 | 745.878 | 4.448 | 7.836 | 15.960 |
| 5.360.954 | 745.878 | 5.009 | 6.686 | 17.114 |
| 6.106.832 | 745.878 | 9.382 | 8.209 | 14.166 |
| 6.852.710 | 745.878 | 8.752 | 8.691 | 14.245 |
| 7.598.588 | 745.878 | 7.561 | 8.604 | 14.967 |
| 8.344.466 | 745.878 | 7.788 | 9.909 | 15.903 |
| 9.090.344 | 745.878 | 8.386 | 10.261 | 17.873 |
| 9.836.222 | 745.878 | 11.197 | 10.706 | 14.746 |
| 10.582.100 | 745.878 | 13.738 | 12.065 | 14.338 |
| 11.327.978 | 745.878 | 15.181 | 13.176 | 15.373 |
| 12.073.856 | 745.878 | 15.079 | 13.701 | 15.152 |
| 12.819.734 | 745.878 | 15.215 | 14.287 | 14.555 |
| 13.565.612 | 745.878 | 12.712 | 14.293 | 13.679 |
| 14.311.490 | 745.878 | 14.933 | 14.804 | 13.346 |
| 15.057.368 | 745.878 | 17.127 | 16.247 | 13.601 |
| 15.803.246 | 745.878 | 16.913 | 17.009 | 14.104 |
| 16.549.124 | 745.878 | 17.541 | 17.568 | 13.036 |
| 17.747.658 | 1.198.534 | 18.325 | 18.629 | 21.674 |
| 18.946.192 | 1.198.534 | 20.384 | 21.117 | 22.140 |
| 19.264.951 | 318.759 | 19.933 | 21.712 | 5.870 |
| 19.583.710 | 318.759 | 20.170 | 20.349 | 6.154 |
| 19.902.469 | 318.759 | 20.369 | 21.567 | 5.721 |
| 20.221.228 | 318.759 | 20.404 | 21.537 | 5.413 |
| 20.539.987 | 318.759 | 21.737 | 22.309 | 5.923 |
| 20.858.746 | 318.759 | 21.821 | 21.447 | 5.997 |
| 21.177.505 | 318.759 | 21.820 | 22.041 | 5.338 |
| 21.496.264 | 318.759 | 21.683 | 22.011 | 5.354 |
| 21.815.023 | 318.759 | 22.716 | 23.106 | 5.481 |
| 23.242.307 | 1.427.284 | 22.585 | 22.967 | 26.950 |
| 24.669.591 | 1.427.284 | 24.229 | 24.617 | 25.510 |
| 25.073.698 | 404.107 | 25.250 | 25.136 | 7.577 |
| 25.477.805 | 404.107 | 24.785 | 24.726 | 6.572 |
| 25.881.912 | 404.107 | 25.809 | 25.772 | 6.810 |
| 26.286.019 | 404.107 | 24.898 | 25.111 | 6.633 |
| 26.690.126 | 404.107 | 25.312 | 25.339 | 6.739 |
| 27.094.233 | 404.107 | 27.168 | 27.323 | 7.631 |
| 27.498.340 | 404.107 | 27.355 | 27.535 | 6.823 |
| 27.902.447 | 404.107 | 27.442 | 27.533 | 7.317 |
| 28.306.554 | 404.107 | 28.522 | 28.555 | 6.955 |
| 28.710.661 | 404.107 | 29.089 | 29.080 | 7.985 |
| 29.114.768 | 404.107 | 29.270 | 29.166 | 7.483 |
| 29.518.875 | 404.107 | 29.098 | 29.153 | 7.224 |
| 29.922.982 | 404.107 | 32.988 | 32.276 | 7.922 |
| 30.327.089 | 404.107 | 32.010 | 32.721 | 7.356 |
| 30.731.196 | 404.107 | 28.320 | 28.772 | 7.681 |
| 31.135.303 | 404.107 | 29.645 | 29.863 | 7.436 |
| 31.539.410 | 404.107 | 32.354 | 32.011 | 6.638 |
| 31.943.517 | 404.107 | 29.732 | 30.124 | 7.352 |
| 32.347.624 | 404.107 | 30.250 | 30.300 | 6.979 |
| 32.751.731 | 404.107 | 30.106 | 31.241 | 7.395 |
| 34.431.570 | 1.679.839 | 31.689 | 32.122 | 33.013 |
| 36.111.409 | 1.679.839 | 32.697 | 32.854 | 30.951 |
| 36.681.772 | 570.363 | 32.073 | 32.377 | 10.693 |
| 37.252.135 | 570.363 | 33.639 | 33.956 | 10.918 |
| 37.822.498 | 570.363 | 34.188 | 34.188 | 9.101 |
| 38.392.861 | 570.363 | 35.098 | 35.026 | 10.033 |
| 38.963.224 | 570.363 | 35.123 | 35.631 | 9.734 |
| 39.533.587 | 570.363 | 34.343 | 35.308 | 10.195 |
| 40.103.950 | 570.363 | 35.954 | 36.484 | 10.146 |
| 40.674.313 | 570.363 | 37.084 | 37.248 | 9.601 |
| 41.244.676 | 570.363 | 38.779 | 39.015 | 10.745 |
| 41.815.039 | 570.363 | 39.344 | 39.206 | 10.896 |
| 42.385.402 | 570.363 | 38.134 | 37.900 | 11.496 |
| 42.955.765 | 570.363 | 37.281 | 37.783 | 9.998 |
| 43.526.128 | 570.363 | 38.608 | 39.094 | 10.064 |
| 44.096.491 | 570.363 | 39.215 | 39.504 | 10.215 |
| 44.666.854 | 570.363 | 40.336 | 40.338 | 9.114 |
| 45.237.217 | 570.363 | 40.198 | 40.472 | 10.629 |
| 45.807.580 | 570.363 | 40.761 | 40.711 | 9.836 |
| 46.377.943 | 570.363 | 40.756 | 41.062 | 9.825 |
| 46.680.065 | 302.122 | 41.908 | 42.431 | 5.216 |
| 46.982.187 | 302.122 | 40.767 | 41.030 | 5.221 |
| 47.062.604 | 80.417 | 40.688 | 40.915 | 858 |
| 47.143.021 | 80.417 | 40.119 | 40.690 | 870 |
| 47.223.438 | 80.417 | 41.244 | 41.275 | 1.001 |
| 47.303.855 | 80.417 | 40.245 | 40.402 | 826 |
| 47.384.272 | 80.417 | 42.313 | 42.747 | 904 |
| 47.464.689 | 80.417 | 41.341 | 41.376 | 959 |
| 47.545.106 | 80.417 | 41.157 | 41.489 | 759 |
| 47.625.523 | 80.417 | 40.780 | 40.795 | 879 |
| 47.705.940 | 80.417 | 41.429 | 41.580 | 847 |
| 47.786.357 | 80.417 | 41.859 | 41.926 | 1.098 |
| 47.866.774 | 80.417 | 41.344 | 41.339 | 801 |
| 47.947.191 | 80.417 | 41.318 | 41.392 | 810 |
| 48.027.608 | 80.417 | 41.913 | 41.707 | 880 |
| 48.108.025 | 80.417 | 42.380 | 42.136 | 914 |
| 48.188.442 | 80.417 | 42.311 | 42.161 | 824 |
| 48.268.859 | 80.417 | 42.430 | 42.674 | 787 |
| 48.349.276 | 80.417 | 42.722 | 42.128 | 798 |
| 48.429.693 | 80.417 | 47.812 | 50.891 | 852 |
| 48.510.110 | 80.417 | 47.744 | 47.971 | 1.184 |
| 48.590.527 | 80.417 | 47.434 | 47.925 | 1.149 |
| 48.670.944 | 80.417 | 48.016 | 48.488 | 1.060 |
| 48.751.361 | 80.417 | 48.165 | 50.397 | 994 |
| 48.831.778 | 80.417 | 49.126 | 49.097 | 940 |
| 48.912.195 | 80.417 | 50.028 | 50.125 | 1.018 |
| 49.799.652 | 887.457 | 48.200 | 48.851 | 18.768 |
| 50.687.109 | 887.457 | 48.321 | 48.778 | 18.176 |
| 51.064.077 | 376.968 | 48.709 | 52.024 | 7.211 |
| 51.441.045 | 376.968 | 50.506 | 51.418 | 7.506 |
| 51.818.013 | 376.968 | 54.101 | 55.036 | 8.319 |
| 52.194.981 | 376.968 | 50.304 | 50.361 | 7.627 |
| 52.571.949 | 376.968 | 52.504 | 53.179 | 7.436 |
| 52.948.917 | 376.968 | 52.966 | 53.548 | 7.142 |
| 53.325.885 | 376.968 | 53.259 | 54.507 | 7.273 |
| 53.702.853 | 376.968 | 56.174 | 56.052 | 7.514 |
| 54.079.821 | 376.968 | 56.656 | 56.808 | 7.516 |
| 54.456.789 | 376.968 | 54.351 | 54.845 | 7.066 |
| 54.833.757 | 376.968 | 56.484 | 57.023 | 7.532 |
| 55.210.725 | 376.968 | 53.445 | 55.320 | 6.812 |
| 55.587.693 | 376.968 | 58.440 | 60.198 | 7.319 |
| 55.964.661 | 376.968 | 55.223 | 55.174 | 7.117 |
| 56.341.629 | 376.968 | 56.523 | 56.217 | 7.584 |

1. Intel Core i5-4690K @ 3.5Ghz [↑](#footnote-ref-1)