

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**: Isdor Reimar Klammer, BSc.

Maximilian Unterrainer, BSc.

Christopher Wieland, BSc.

Puch/Salzburg, 19.02.2017

Inhaltsverzeichnis

[1 Einleitung 1](#_Toc476131641)

[1.1 Problemstellung und Motivation 1](#_Toc476131642)

[1.2 Umgebung (oder Stand der Technik) 2](#_Toc476131643)

[2 Anforderungen 4](#_Toc476131644)

[2.1 Datenmodell 4](#_Toc476131645)

[2.1.1 Analyse bestehendes Datenmodell 5](#_Toc476131646)

[2.1.2 Anforderungen von Energieversorgern und Netzbetreibern 5](#_Toc476131647)

[2.1.3 Analyse der Usecases 6](#_Toc476131648)

[2.1.4 Weitere Datenmodelle 6](#_Toc476131649)

[2.1.5 Rechtliche Rahmenbedingungen und daraus resultierende Einschränkungen der möglichen Messwerte 7](#_Toc476131650)

[2.1.6 Einbeziehung zusätzlicher Domänen 8](#_Toc476131651)

[2.1.7 Das COSEM Modell 8](#_Toc476131652)

[2.1.8 ER-Modell Festlegung 8](#_Toc476131653)

[2.2 Anzahl erwarteter Datensätze 11](#_Toc476131654)

[2.3 Alternative Datenhaltung 11](#_Toc476131655)

[2.3.1 Analyse JRZ-DB 13](#_Toc476131656)

[2.3.2 Performanceanalyse 13](#_Toc476131657)

[2.3.3 Hadoop Tests 13](#_Toc476131658)

[2.3.4 MySQL Tests 14](#_Toc476131659)

[2.3.5 Testdaten 14](#_Toc476131660)

[2.3.6 MySQL Datenbank 14](#_Toc476131661)

[2.3.7 Hadoop Datenbank 16](#_Toc476131662)

[2.3.8 Zusammenfassung der Datenbankanalyse 17](#_Toc476131663)

[2.4 Rollenbasierter Zugriff 17](#_Toc476131664)

[2.5 Schnittstellen 19](#_Toc476131665)

[2.6 Systemarchitektur 19](#_Toc476131666)

[3 Literaturverzeichnis 23](#_Toc476131667)

[4 Anhang 25](#_Toc476131668)

[A.1 SQL Messungen 25](#_Toc476131669)

Abkürzungsverzeichnis

AMCS Advanced Meter Communication System

API Application Programming Interface

COSEM Companion Specification for Energy Metering

DAVID-VO Datenformat- und Verbrauchsinformationsdarstellungs Verordnung

DLSM Device Language Messaging Specification

ETSI European Telecommunication Standards Institute

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.

GDPR EU General Data Protection Regulation

IMA-VO Intelligente Messgeräte-AnforderungsVO 2011

MMS Manufacturing Messaging Specification

OBIS Object identification system, entsprechend der EN 62056-01

OSGP Open Smart Grid Protocol

RBAC Role Based Access Control

SmartValAPI Smart Meter Data Value API, Arbeitstitel des Projektes

SML Smart Message Language

SOAP Simple Object Access Protocol

DBM Datenbankmodell

Abbildungsverzeichnis

Abbildung 1: ER-Modell Entitäten 9

Abbildung 2: Datenbankbeziehungen 10

Abbildung 3: Abfrage Durchschnitt pro Tag 14

Abbildung 4: Dauer des Datenimports 15

Abbildung 5: Dauer Berechnung des Durchschnittsverbrauchs 16

Abbildung 6: Rollendefinition - UseCase Diagramm 19

Abbildung 7: Komponentenmodell SmartValAPI 22

Tabellenverzeichnis

Tabelle 1: Datenformat CSV 14

Tabelle 2: Dauer des Datenimports 15

Tabelle 3: Dauer Berechnung des Durchschnittsverbrauchs 16

# 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

Momentan wird auf der aktuellen Datenbank direkt auf die Messdaten zugegriffen. Dies verursacht unter Umständen doppelte Implementierungen sowie ein erhöhter Wartungsaufwand bei Datenbankerweiterungen. Ebenfalls sind zurzeit die Messdaten in unterschiedlicher Auflösung, welche genormt werden müssten.

Weitere Unterschiede bestehen in der Granularität der Daten, welche die Smart Meter Modelle zur Verfügung stellen. Ebenso ist die Frequenz, mit der Werte ausgelesen werden können, unterschiedlich. All diese Unterschiede erschweren vergleichende Auswertungen von Messdaten aus unterschiedlichen Quellen.

Dieses Projekt verfolgt vier Hauptziele:

* Schaffung einer erweiterbaren Programmierschnittstelle (SmartValAPI), welche 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 (oder Stand der Technik)

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 wird auf deren Eignung für die zu erwartenden Messdatenmengen evaluiert und stellt die Ausgangsbasis für mögliche, notwendige Erweiterungen dar.
* Importmodule [3] um Messdaten in der Datenbank abzulegen. Dieses Programmpaket ermöglicht es, Messwerte, die in den Formaten ADRES, GREEND, REDD und UK-DALE vorliegen zu importieren.
* Rollenbasierter Zugriff: OpenTC [4] TODO: noch nicht fertig! stellt ein Softwarepaket zur Verfügung, über das der rollenbasierte Zugriffsschutz realisiert wird, die Rollenverwaltung erfolgt über ein beliebiges LDAP-Administrationswerkzeug.

Weitere verwendete Softwarepakete werden im Abschnitt Systemarchitektur angeführt.

# Anforderungen

Dieser Abschnitt beleuchtet die Anforderungen, die sich aus den Projektvorgaben ableiten lassen.

## Datenmodell

***Ich habe das umbenannt, da ein ER eher zu einer RDB führt, das „Datenmodell“ lässt uns offen, dass es auch eine Mischung sein kann. Das Kapitel ER-Model überarbeiten, die Umfrage rein, und das fertige ER Modell dazu. Das ursprüngliche in den Anhang als Ausgangspunkt. Referenzieren als JRZ-DB, Fokus auf die zusätzlichen Werte.***

Das Datenmodell der JRZ-DB wird analysiert und auf Erweiterungsnotwendigkeiten untersucht. Die Analyse erstreckt sich auf die fachlichen Anforderungen durch das JRZ als Auftraggeber und berücksichtigt des Weiteren die möglichen Bedürfnisse von Energieversorgern und Netzbetreibern.

Zusätzlich ergibt sich die Notwendigkeit von technischen Erweiterungen durch die Verwaltung von Zugriffsrollen und Gruppen und Besitzern der Meterdaten. Des Weiteren wird Wert auf die Kompatibilität zu bestehenden Anwendungen gelegt.

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 bestehendes Datenmodell

In der JRZ-DB werden Messdaten als Tupel in einer Tabelle (meter\_data) abgelegt, je Messzeitpunkt werden folgende Werte, sofern vom Smart Meter zur Verfügung gestellt, gespeichert.

* Nutzdaten (Momentanwerte):   
  je Phase: aktuelle Leistung, aktueller Stromverbrauch (sofern vom Smart Meter übertragen in dieser Granularität zur Verfügung gestellt, sonst als Einzelwert in Phase1). 4 Werte (count\_register1 – count\_register4) die abhängig vom Smart Meter (meter\_type) belegt werden (Details dazu im Abschnitt „Importprogramme“),   
  Gesamtwert: Spannung, Frequenz, kumulierte Werte: Verbrauch kWh.
* Verwaltungsdaten (zur Identifikation):  
  meter\_id des Smart Meters: Fremdschlüssel zu meter\_management,  
  data\_id: eindeutiger Schlüssel des Messdaten-Tupels,  
  timestamp: Erstellungszeitpunkt zu dem die Nutzdaten aufgezeichnet werden.

### Anforderungen von Energieversorgern und Netzbetreibern

Fragenliste einfügen in Form einer Tabelle. Wenn nicht zitierfähig, dann erwähnen

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

[5] und des Lastenheftes [6] von Österreichs Energie. Die Usescases beschäftigen sich vor allem mit Daten bezüglich der Steuerung des Smart Meter als solches und nur am Rande mit der inhaltlichen Bedeutung der übertragenen Messdaten.

Neben den, in meter\_data bereits vorhandenen Datenfeldern, bietet das Lastenheft optional die Möglichkeit der Auslesung der Blindleistung (I.-IV. Quadrant), diese Werte werden übernommen, beziehungsweise sofern geliefert in meter\_data abgelegt.

Weitere Datenfelder, wie sie zum Beispiel das ETSI in der Definition des OSGP [7] vorschlägt, werden nicht in die Menge der gespeicherten Daten aufgenommen, da diese weder von den Energieversorgern noch von den Netzbetreibern gewünscht werden (Rückmeldungen, Lastenheft).

### Weitere Datenmodelle

Fusco et al. [8] schlagen einen dualen Betrieb von RDBMS und einer NoSQL Datenverwaltung vor. Warum wird ein dualer Betrieb vorgeschlagen? Warum wird es bei uns anders gemacht? Was sind die Vorteile mit unserem Weg? 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 [9]. Bezüglich der auslesbaren Daten legt die [6] 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.

### 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 und Wärme und Wasser. Technisch ist eine Erweiterung um jene Felder, die nach der rechtlichen Festlegung erfasst werden sollen, problemlos möglich, daher gehen wir in der Umsetzung dieses Projekts darauf nicht weiter ein.

### Das COSEM Modell

[2] versucht 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 [6] als Ausgangspunkt für die Festlegung der Datenbank herangezogen. Warum? Interessant ist doch für die DB in erster Linie welche Werte verspeicher werden sollen, und nicht das Protokoll mit dem der Meter ausgelesen wird.

### 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 1: 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. Hier wäre eine LDAPAnbindung (an den Server der OpenTC Gruppe) besser gewesen?

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 2: 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“).

## Anzahl erwarteter Datensätze

*Wir haben über die Anzahl folgender Datensätze gesprochen: Salzburg AG (mit 500000 Metern im Felde, 24 Stunden \* 4 (alle 15 Min einer) ergibt pro Monat 1.44 Mrd. Sätze, und im Jahr 17,28 Mrd. Datensätze Das hat Einfluss auf die Datenhaltung.*

## Alternative Datenhaltung

***Die im Abschnitt „Anzahl erwarteter Datensätze“ angeführten Zahlen führen insbesondere bei einem zentralen und längeren Betrieb der Datenbank zu einer Überforderung von RDBS, daher schauen wir uns Hadoop und evtl. MongoDB an. Legen wir den Fokus auf die Vorteile der Systeme, irgendwo hab ich diese schon herausgestrichen, das können wir noch verfeinern, dann passt hier auch der Verweis zu Fusco et al. Mit dem hybriden Ansatz mit meter\_data als NoSQL und dem Rest wie gehabt als MySQL.***

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.

### 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 3: Abfrage Durchschnitt pro Tag

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

Es haben sich bei den Tests folgende Ergebnisse errechnen lassen:

### MySQL Datenbank

Zuerst wird die Dauer des Datenimports gemessen.

Wie ist das Setup? Was war die Methode? Kann ein Netzwerkdelay ausgeschlossen/vernachlässigt werden? Wurde nur einmal gemessen oder ist das Ergebnis der Mittelwert/Median?

|  |  |  |
| --- | --- | --- |
| **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

Ist die Menge von ca. 9 Mio Datensätzen wirklich aussagekräftig, wenn allein der low frequency REDD Datensatz schon 52 Mio. hat.

Hier wäre eine Analyse interessant wann und warum (my)SQL einbricht in der „Linearität“

Abbildung 4: Dauer des Datenimports

Achsen benennen!

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 3 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 5: Dauer Berechnung des Durchschnittsverbrauchs

Achsen benennen!

Abbildung 5 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.

## Rollenbasierter Zugriff

***Rollendefinition: woraus ergibt sich diese und was bewirken diese Rollen, Definition auf ElWOG und DAVID für die Einschränkungen, zusätzlich die freien Daten wie z.B. REDD, UK-DALE u.ä. dürfen nach den Anforderungen des Auftraggebers mit einer Rolle drauf zugreifen. Ich bin LDAP-mäßig noch nicht ganz firm, habe aber gesehen, dass es da die Möglichkeit gibt, neben den Rollen, Gruppen und Benutzern auch Geräte zu definieren.***

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.
* Untersuchen der rechtlichen Rahmenbedingungen:
* die GDPR und das ElWOG §84 legt die Rahmenbedingungen für die Erfassung, die Weiterleitung und die Speicherung von Messdaten fest. Abs. (1) regelt die Erfassung von Viertelstundenwerten und den Zeitraum der Speicherung. Des Weiteren wird die Weitergabe des Tagesverbrauchs an den Netzbetreiber geregelt.
* 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. Außerdem ist in der DAVID-VO §4 der Zugriff für vom Endverbraucher bevollmächtigte Dritte auf die Daten im Kundenportal des Endverbrauchers geregelt.
* Überlegungen bezüglich der Administration führen zu einer Trennung von denjenigen Personen die Messdaten in die Datenbank importieren und jenen die die Administration des Gesamtsystems überhaben. Durch die Vergabe von beiden Rollen an eine Person ist die gesamte Administration durch eine Person, insbesondere in kleinen Installationen, möglich.

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 6: Rollendefinition - UseCase Diagramm

## Schnittstellen

*Zentrales ist der bisherige Klassenentwurf.*

## Systemarchitektur

***Deployen auf landsteiner.fh-salzburg.ac.at (193.170.119.66), die Ports sind nicht alle offen.***

Um die Systemumgebung festzulegen, und vor allem die Software passgenau in die Softwarelandschaft des JRZ einfügen zu können, werden die bestehenden weiterverwendenden Systeme und Importprogramme berücksichtigt. In die Überlegungen werden etwaige Kosten für Lizenzen und andererseits Sicherheitsaspekte einbezogen [10].

In den weiteren Punkten wird erläutert, um welches System es sich handelt und welche Software darauf installiert ist.

* Hardware: Für den Betrieb wurde vom JRZ eine virtuelle Maschine im Bladecenter zur Verfügung gestellt. Die Zugangsdaten hierzu sind: *landsteiner.fh-salzburg.ac.at (193.170.119.66)*
* Betriebssystem: Aufgrund der größeren Erfahrung der Entwickler mit der Administration erfolgte eine Installation eines Windows Systems. (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) wurde auch ein NoSQL Datenbankensystem in Betracht gezogen. Warum MySQL? Im Sinne der Integration in die bestehende Softwarelandschaft wird MySQL eingesetzt. Ziel des Projektes ist eine Integrationsdatenbank, welche dazu führt, dass sehr große Datenmengen verwaltet werden müssen, und der Zugriff in akzeptabler Zeit erfolgt. MySQL unterstützt lokale (partitioning) und verteilte (sharding) Fragmentierung. Nach [11] 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 Welchen Anforderungen? ergeben sich folgende Komponenten, deren Abhängigkeiten untereinander und Verbindungen zueinander, die im UML Diagramm (Abbildung 1) dargestellt werden:



Abbildung 7: Komponentenmodell SmartValAPI – Bild Neu Designen (Vektorgrafik)

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.

# 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] | A.-M. O. u. M. T. Carlo Bellucci, „Untersuchung unterschiedlicher Referenzdatensätze im Energiebereich,“ 2017. |
| [4] | W. F. T. H. Michael Egger, „Rollenbasierter LDAP Zugriff TODO TODO TODO,“ 2016. |
| [5] | 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]. |
| [6] | 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]. |
| [7] | 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]. |
| [8] | U. F. V. L. P. P. J.-B. F. B. C. Y. G. M. S. Francesco Fusco, „Data Management System for Energy Analytics and its Application to Forecasting,“ in *EDBT/ICDT Workshops*, Bordeaux, 2016. |
| [9] | 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. |
| [10] | *IT-Grundschutz-Profil für Open-Source-Software (GSProOSS),* 2010. |
| [11] | 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]. |
| [12] | 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]. |
| [13] | J. G. P. S. Phillipe Bonnet, „Towards sensor database systems,“ in *International Converence on Mobile Data Management*, Berlin, 2001. |
| [14] | 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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Messung | Anzahl Zeilen | Affected | Query | DauerNoIdx | Dauer Idx |
| 1 | 1.561.660 | 23 | A | 1,172 | 1,218 |
| 2 | 1.561.660 | 23 | A | 1,297 | 1,235 |
| 3 | 1.561.660 | 23 | A | 1,25 | 1,312 |
| 4 | 1.561.660 | 23 | A | 1,266 | 1,265 |
| 5 | 2.307.538 | 745.878 | B | 52,203 | 188 |
| 6 | 2.307.538 | 745.878 | C | 32,86 | 420 |
| 7 | 2.307.538 | 58 | A | 4,265 | 4,453 |
| 8 | 2.307.538 | 58 | A | 4,172 | 4,703 |
| 9 | 2.307.538 | 58 | A | 4,218 | 6,937 |
| 10 | 2.307.538 | 58 | A | 5,468 | 7,031 |
| 11 | 3.869.198 | 1.561.660 | B | 82 |  |
| 12 | 3.869.198 | 1.561.660 | C | 76 |  |
| 13 | 3.869.198 | 81 | A | 11,359 |  |
| 14 | 3.869.198 | 81 | A | 11,203 |  |
| 15 | 3.869.198 | 81 | A | 12,625 |  |
| 16 | 3.869.198 | 81 | A | 10,891 |  |
| 17 | 4.615.076 | 745.878 | B | 51,203 |  |
| 18 | 4.615.076 | 745.878 | C | 44,609 |  |
| 19 | 4.615.076 |  | A | 15,063 |  |
| 20 | 4.615.076 |  | A | 15,89 |  |
| 21 | 4.615.076 |  | A | 14,86 |  |
| 22 | 4.615.076 |  | A | 14,985 |  |
|  |  |  |  |  |  |
|  | Anzahl | Dauer |  |  |  |
|  | 1.561.660 | 1,24 |  |  |  |
|  | 2.307.538 | 4,53 |  |  |  |
|  | 3.869.198 | 11,52 |  |  |  |
|  | 4.615.076 | 15,2 |  |  |  |

|  |  |
| --- | --- |
| Query ID | Query |
| A | select avg(power) as power, day, month, meterId from redd group by  meterId asc, month asc, day asc; |
| B | update redd set datetime=FROM\_UNIXTIME(timestamp)  where meterId is null; |
| C | update redd set day=day(datetime), month=month(datetime),  meterId = 3 where meterId is null; |

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