**Pflichtenheft (2)**

|  |  |  |
| --- | --- | --- |
| Projektdefinition / Pflichtenheft: *Erstellung einer JRZ Demodatenbank (DemoDB)* | | |
| Auftraggeber:  FH Salzburg:  DI Eduard Hirsch  DI Fabian Knirsch, BSc | Datum: *4.4.2017* | Projektleiter/Mitarbeiter: Isidor Reimar Klammer, BSc  Christopher Wieland, BSc  Maximilian Unterrainer, BSc |
| 1. Zweck des Projektes    1. Das JRZ an der FH Salzburg hat Zugriff auf zahlreiche Quellen von Smart Meter Messdaten.   Unter anderem von seinem Partner Salzburg AG, im Labor selbst erfasste Daten und frei verfügbare Profile (zum Beispiel aus REDD, GREEND, UK-DALE und ADRES).  Diese Messdaten liegen je nach Quelle in unterschiedlichen Formaten und Qualitäten vor.   * 1. Für Analysen und die Weiterverarbeitung ist es zweckmäßig diese Messdaten in gemeinsames Format zu konvertieren.   In einer gemeinsamen Datenbank abgelegt, erfolgt der Zugriff kanalisiert und rollenbasiert über eine programmatische Schnittstelle (API). | | |
| 1. Zieldefinition 2. Musskriterien – Messbare Beschreibung der Ziele  * *Messdaten verschiedenster Qualitäten stehen nach dem Import in der JRZ-Datenbank für Abfragen zur Verfügung.* * *Schnittstellen (API) für den Zugriff auf die Datenbank sind definiert, ausprogrammiert und stehen zur Verfügung.* * *Funktionsdefinition und Rollenverwaltung der einzelnen Benutzer (Zugang einzelner User auf deren Daten, Universitäten auf grobe Übersichten, etc.) wurde implementiert*  1. Sollkriterien – was kann dazugehören  * *Komfortable Benutzerverwaltung*  1. Abgrenzung – was gehört nicht dazu  * *Keine Visualisierung* * *Selbsterfindungen der Anforderungen (wie und was soll ausgewertet werden)* * *Importmodule für weitere Messdatenformate* | | |
| 1. Lösungsansatz    1. Technische Ausgangssituation bzw. bestehende Teillösungen   zur Verfügung stehen:   * *Aus bestehenden Forschungsprojekten am JRZ: ERM (wird erweitert/angepasst),* * *Aus einem Visualisierungsprojekt: die Basis einer Zugriffsverwaltung über Rollen, wird analysiert, soweit möglich übernommen und ggf erweitert.* * *Messdatensätze in unterschiedlichen Formaten und unterschiedlicher Qualität (Auflösung, Anzahl der Attribute zu Messdaten)*   1. Umsetzung der Ziele * *Analyse des Datenmodells,* * *Analyse der Messdatenformate, festlegen der importier baren Messdatenformate,* * *Gemeinsames Format der Messdaten (Art der Werte, Timestamp, etc.), Metadaten* (*Quelle, Ort, etc.) des „vereinten Messdatensatzes“,* * *Klärung mit Auftraggeber und „Wissenden im JRZ“, welche Auswertungen gemacht werden sollen und welche Rollen es geben könnte.*  1. *Datenbankschema,* 2. *Schnittstellen (Implementieren, Testen, API zur Verfügung stellen, Musteraufrufe)*    1. Vergleich mit bestehenden Lösungen  *Modul 1 (*[*Bundesamt für Energie BFE*](http://www.bfe.admin.ch/index.html?lang=de)*, Projektbericht 501053 - Smart Meter Datenanalyse für automatisierte Energieberatung)* | | |
| 1. Marktrelevanz 2. Zielgruppen    * *In erster Linie JRZ und Salzburg AG, in der Folge eventuell weitere Forschungseinrichtungen oder Energieversorger/Netzbetreiber*    1. Anwendungsbereiche    * *komfortable Analyse von Smart-Meter-Messdaten über Datenquellen hinweg*    1. Markteinschätzung    * *Digital Europe [1] und universitäre Studien [2] beschäftigen sich mit der Analyse von Smartmeter Daten, und sehen einerseits Energiesparpotenzial, andererseits erwarten Businessleader [3] mögliche große Ertragschancen.* | | |
| 1. Entwicklungsumgebung / benötige Komponenten 2. Software  * *OS,* * *Datenbanksystem: RDBMS, Sternschema nach DWH, NoSQL (Keystore, zB. Cassandra, MongoDB),* * *Rollenverwaltung (Eigenentwicklung oder Open Source?),*      1. Hardware  * *Bladeserver am JRZ, installiertem Betriebssystem auf der VM*  1. Orgware  * *Anmeldedaten vom Bladeserver (Name, Benutzer) für das Projektteam eingerichtet* | | |
| 1. Produktspezifika 2. Funktionalität, Auflistung aller wesentlicher Funktionen bzw. Eigenschaften  * *ER: Whiteboxbeschreibung der Entitäten, Relationen.* * *API: Funktionsbeschreibung, Parameter, Rückgabewerte.* * *Rollenverwaltung* | | |
| 1. Testszenarien und Testfälle 2. Funktionstest  * *entlang des API Entwurfs und dem prototypischen Importprogramm erstellen wir die notwendigen Testfälle.* * *Testen der entwickelten Funktionalität durch Unit-tests.*  1. Integrationstest  * *Testen der Integration im Systemkontext (Installation auf der VM am JRZ)* * *API steht zur Verfügung, Einbinden in JRZ Applikationen* | | |
| 1. Sicherheitsanforderungen  * *Zugriffsschutz gewährleistet durch: - Benutzeranmeldung, - Rollenzuordnung, was eine rechtevergebende Instanz voraussetzt.* * *Diese soll getrennt von den API Verwendern sein.* * *Zugriff auf personenbezogene Daten erfolgt rollenbasiert,* * *als Datenimportierender besteht voller Zugriff auf die (eigenen) Daten,*      * *es ist zu gewährleisten, dass dieser Zugriff nach dem Import weiterhin besteht, jedoch auf Daten anderer Benutzer nur insoweit als dies die Summe der Rollen ermöglicht.* * *REDD Daten in hoher Auflösung erfordern viel Speicherplatz, der Importvorgang stellt vor dem Laden der Daten sicher, dass genug Platz zur Verfügung steht, und Importvorgänge sequentiell abgearbeitet werden.* * *Periodische Sicherungskopien sind mit geeigneten Tools anzulegen.* | | |
| 1. Rechtliche Rahmenbedingungen  * *Messdaten lassen Rückschlüsse auf das Verhalten der Benutzer zu,*      * *die Privatsphäre ist zu schützen, siehe Punkt 8. Sicherheitsanforderungen.* | | |
| 1. Szenario des Projektendes  * *Positiv: Abnahme erfolgt durch Prüfung der API Ergebnisse über die Unittests auf der Installation der Komponenten auf der VM des JRZ, Übergabe der Software, Handbuch, Installationsanleitung.*   *Alle Test Cases aus Punkt 7 sind erfüllt.*   * *Negativ: Die Ziele werden nicht erfüllt.* | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1. Ressourcenschätzung  |  |  |  | | --- | --- | --- | | **Meilenstein** | **Datum** | **Deliverable** | | Internes Meeting | Wöchentlich |  | | REST Schnittstelle Struktur festlegen |  | Report | | Datenbankzugriffe, Implementierung eines beispielhaften Zugriffs | 30.4.2017 | Demo | | Authentifizierung mit Benutzername und Passwort über LDAP | ?? |  | | Java Laufzeitumgebung festlegen und bereitstellen (Applikationsserver, Springboot,….) | 30.4.2017 | Demo | | LDAP Struktur festgelegt und Zugriff Demo | ?? | Report | | Implementierung restliche Zugriffe | ?? | Demo | | Installationsanleitung System | ?? | Report | | Dritter Statusreport | 30.4.2017 | **Statusreport 3** | | Vierter Statusreport | 15.6.2017 | **Statusreport 4** | | API Funktionsbeschreibung |  | Report | | Unterlagen und Ergebnisse |  | Report | | Ausblick, offenes | 31.8.2017 | **Meeting** | | Testkonzept |  | **!!!** | | Projektmarketing Unterlagen festlegen  Plakat |  | **Report** | | Präsentation | Ende September 2017 | **Powerpoint vs. LifeDemo?** | | Präsentationsinhalte festlegen |  | **Konsens auf Grund von Vorschlag** | |  |  |  | |  |  |  | |  |  |  | |  |  |  | |
|  |