Reporting in a Microservice Architecture

Konzept für die Erstellung von Berichten in einer Microservice Architektur

Bachelorarbeit

Studiengang Informatik

Zürcher Hochschule für angewandte Wissenschaften

eingereicht bei

Beat Seeliger, Fachdozent für Software-Entwicklung

Oktober 2015

Eingereicht von

Severin Andrew Müller

Hauptstrasse 39

CH-8224 Löhningen SH

[muelles5@students.zhaw.ch](mailto:muelles5@students.zhaw.ch)

Inhaltsverzeichnis

[1 Einleitung 5](#_Toc424291454)

[1.1 Ausgangslage 5](#_Toc424291455)

[1.2 Ziele der Arbeit 6](#_Toc424291456)

[1.3 Aufgabenstellung 6](#_Toc424291457)

[1.3.1 Ist-Analyse 6](#_Toc424291458)

[1.3.2 Anforderungsanalyse 6](#_Toc424291459)

[1.3.3 Recherche 6](#_Toc424291460)

[1.3.4 Konzept 6](#_Toc424291461)

[1.3.5 Proof of Concept 6](#_Toc424291462)

[1.3.6 Testing 6](#_Toc424291463)

[1.3.7 Abschluss 6](#_Toc424291464)

[2 Bisherige Situation (Ist-Analyse) 7](#_Toc424291465)

[2.1 Einleitung 7](#_Toc424291466)

[2.1.1 Reports in der Applikation r5 8](#_Toc424291467)

[2.1.2 Reports auf der MRE 8](#_Toc424291468)

[2.1.3 Reports auf der Web-Plattform 11](#_Toc424291469)

[2.2 Datenquellen 11](#_Toc424291470)

[2.3 Datenstrukturen 11](#_Toc424291471)

[2.4 Report Templates 11](#_Toc424291472)

[2.5 Schwächen 12](#_Toc424291473)

[3 Anforderungsanalyse 13](#_Toc424291474)

[3.1 Funktionale Anforderungen 13](#_Toc424291475)

[3.1.1 Stakeholder 13](#_Toc424291476)

[3.1.2 Allgemeines 13](#_Toc424291477)

[3.1.3 Automatisierte Reports 13](#_Toc424291478)

[3.1.4 Report-Eigenschaften 14](#_Toc424291479)

[3.1.5 Datensicherheit 15](#_Toc424291480)

[3.1.6 Plattform-Eigenschaften 15](#_Toc424291481)

[3.2 Nicht-Funktionale Anforderungen 15](#_Toc424291482)

[3.2.1 Performance 15](#_Toc424291483)

[3.2.2 Verfügbarkeit 15](#_Toc424291484)

[3.2.3 Optik 15](#_Toc424291485)

[3.3 Technische Anforderungen 16](#_Toc424291486)

[3.3.1 Scheduler 16](#_Toc424291487)

[3.3.2 Daten 16](#_Toc424291488)

[3.3.3 Datenobjekte 16](#_Toc424291489)

[3.3.4 Logging 16](#_Toc424291490)

[3.3.5 Migration 16](#_Toc424291491)

[4 Recherche 17](#_Toc424291492)

[4.1 Benötigte Werkzeuge 17](#_Toc424291493)

[4.1.1 Scheduling 17](#_Toc424291494)

[4.1.2 Rendering 17](#_Toc424291495)

[4.2 Evaluation der Werkzeuge 17](#_Toc424291496)

[4.3 Ergebnisse 17](#_Toc424291497)

[5. Konzept 18](#_Toc424291498)

[5.1 Request-Format 18](#_Toc424291499)

[5.2. Schnittstellen 18](#_Toc424291500)

[5.3 Datenaufbereitung 18](#_Toc424291501)

[5.5 Layout-Bearbeitung 18](#_Toc424291502)

[5.6 Rendering 18](#_Toc424291503)

[5.7 Deployment 18](#_Toc424291504)

[6 Proof of Concept 19](#_Toc424291505)

[6.1 Bau eines Prototypen 19](#_Toc424291506)

[6.2 Definition Beispiel-Reports 19](#_Toc424291507)

[6.3 Datenaufbereitung für Beispiel-Reports 19](#_Toc424291508)

[6.4 Ergebnisse der Beispiele 19](#_Toc424291509)

[7 Testing 20](#_Toc424291510)

[7.1 Testkonzept 20](#_Toc424291511)

[7.2 Unit Tests 20](#_Toc424291512)

[7.3 Integrationstest 20](#_Toc424291513)

[7.4 Regression Test 20](#_Toc424291514)

[7.5 Acceptance Test 20](#_Toc424291515)

[8 Fazit und Erkenntisse 21](#_Toc424291516)

[8.1 Zusammenfassung 21](#_Toc424291517)

[8.2 Fazit 21](#_Toc424291518)

[8.3 Ausblick 21](#_Toc424291519)

[9 Anhang 22](#_Toc424291520)

[9.1 Anhang A: Bilderverzeichnis 22](#_Toc424291521)

[9.2 Anhang B: Tabellenverzeichnis 22](#_Toc424291522)

[9.3 Anhang C: Abkürzungsverzeichnis 22](#_Toc424291523)

[9.3.Anhang D: Literaturverzeichnis 22](#_Toc424291524)

# 1 Einleitung

## 1.1 Ausgangslage

Wir leben in einer Zeit in der Information von enorm grosser Bedeutung ist. Seit das Internet seinen Siegeszug Mitte der 1990er Jahre begonnen hatte ist die weltweite Datenmenge enorm gestiegen. Aus einer grossen Flut von Daten die nützlichen Informationen empfängergerecht und anschaulich zu repräsentieren kann sich als schwierig gestalten.

Mit Berichten, sogenannten Reports kann man Informationen rasch und einfach bereitstellen. Solche Berichte können in unterschiedlichen Formen auftreten – von einfachen Datenauszügen (-> *Dumps*) bis hin zu personalisierbaren Dashboards ist alles dabei.

Der Autor der vorliegenden Arbeit ist bei seinem Arbeitgeber unter anderem zuständig für die Erstellung und Wartung von Kundenberichten. Diese Kundenberichte sind einerseits in der Applikation, die der Arbeitgeber des Autors vertreibt, laufen sowie Reports die auf einer eigenen Plattform betrieben werden.

Die Applikation „r5“ der MESPAS AG ist eine Komplettlösung für Reedereien, deren Schiffe und Lieferanten. Dieses Legacy System läuft seit über 10 Jahren auf einem Tomcat Server mit einer zentralen MySQL Datenbank. Fat Clients verbinden sich mit dem Zentralserver um damit zu arbeiten. Mit diesem Client hat der Kunde die Möglichkeit für fast jeden Bereich einen Bericht zu erstellen, sogenannte Reports. In den meisten Fällen soll ein solcher Report die Information, die auf dem Screen zu sehen ist, in einer PDF-Datei ausgegeben werden. Im Einkaufsmodul (Procurement) werden zudem auch für alle benötigten Formulare wie Einkaufsanfrage, Bestellungen, etc. entsprechende Reports generiert. Dazu läuft auf demselben Server eine Open-Source Lösung von BIRT (Business Intelligence Report Tools). Auf Basis eines sogenannten Report-Templates werden die Daten aufbereitet und die Ausgabedatei generiert.

Zusätzlich zu den Reports in der Applikation r5 bietet die MESPAS AG eine entkoppelte Plattform an, genannte MESPAS Reporting Engine (kurz MRE). Diese ermöglicht es den Kunden kompliziertere Reports unabhängig von der Applikation zu generieren und beinhalten meist komplexere Aggregationen als jede in der Applikation selbst. Zudem können diese MRE-Reports auch terminiert werden. Das heisst, dass ein Benutzer sich einen Report regelmässig ausführen und sich per E-Mail schicken lassen kann. Die Implementierung der Reports folgen dem gleichen Muster (Report Templates mit BIRT).

Seit 2014 ist eine neue, web-basierte Lösung der Applikation r5 in Form einer Micro-Service Architektur in Entwicklung. Jede Tätigkeit läuft in einem eigenen Micro-Service in einer Datenbank mit entsprechenden zugehörigen Schemas. Nun soll für diese neue Plattform ein Konzept für die Integration von Kunden-Reports entwickelt werden. Diese soll ebenfalls als Micro-Service implementiert werden. Mit dieser neuen Lösung soll auch die MRE abgelöst werden können. Die Herausforderung ist dabei in erster Linie die Datenaufbereitung. Da die Daten bisher in einer zentralen Datenbank verwaltet wurde mussten diese nur von dort geholt werden. Für die neue Datenbank muss ein Konzept geschaffen werden, dass diese Daten unter Berücksichtigung von Performance und Aspekten der Datensicherheit aus den verschiedenen Datenbanken geladen und für die Erstellung dieser Reports aufbereitet werden können. Zu beachten ist dabei, dass jeder Microservice bei Änderungen individuell "deploybar" ist. Deshalb soll die Definition des Reports beim zugehörigen Microservice liegen und der Reporting-Microservice muss bei solchen Änderungen die neue Definition verarbeiten können, ohne dass ein neues Deployment nötig wird.

## 1.2 Ziele der Arbeit

Das Ziel der Arbeit ist das fertige Konzept für die Integration der Reports. Dabei sollen zunächst mögliche Werkzeuge für das Rendering evaluiert werden (z.B. ob BIRT Templates weiterhin unterstützt werden können). Danach soll analysiert werden, wie mit einem Request in JSON Form alle notwendigen Daten aufbereitet und zurückgegeben werden können. Die neue Reporting-Lösung soll es zudem ermöglichen, dass die Reports terminiert und automatisiert per E-Mail versendet werden können.

Bis anhin wurden die Reports von Hand getestet. Es soll daher eine Möglichkeit für automatisierte Unit Tests geschaffen werden. Zuletzt soll anhand eines Prototyps / Proof of Concept die Lauffähigkeit dieses Konzepts dargelegt werden.

## 1.3 Aufgabenstellung

Aus der Ausgangslage und den Zielen der Arbeit ergibt sich folgende Aufgabenstellung:

### 1.3.1 Ist-Analyse

* Analyse der bisherigen Datenstrukturen
* Analyse der bisherigen Werkzeuge

### 1.3.2 Anforderungsanalyse

* Erarbeitung der Anforderungen an das neue Reporting-Konzept
* Anforderungen an das Datenmodell beschreiben

### 1.3.3 Recherche

* Evaluation der Werkzeuge durchführen
* Entscheidung für ein Tool treffen

### 1.3.4 Konzept

* Definition des Request-Formates
* Beschreibung der Schnittstellen
* Beschreibung der Datenaufbereitung
* Beschreibung des Test-Konzeptes

### 1.3.5 Proof of Concept

* Prototyp des Micro-Service entwickeln
* Proof of Concept anhand Beispiel-Reports beschreiben

### 1.3.6 Testing

* Definition von Testfällen
* Durchführung der Tests

### 1.3.7 Abschluss

* Fazit und Erkenntnisse

# 2 Bisherige Situation (Ist-Analyse)

## 2.1 Einleitung

Die bisherige Lösung steht auf drei Säulen: die Reports in der Applikation r5, die Reports auf der MRE und die Reports auf der neuen Web-Plattform. Sämtliche Reports werden mit BIRT (Business Intelligence Reporting Tools) erstellt.



Abb. : Bestehende Lösung – Übersicht

Auf Abb. 1 wird eine Übersicht über die Reporting-Architektur in r5 und MRE gezeigt. Dabei fällt auf, dass die MRE und der r5 Office Client dieselbe Datenbank verwenden. Der r5 Offshore Client wird auf Schiffen installiert. Da Schiffe über keine Permanente Internetverbindung verfügen wird hier eine lokale H2 Datenbank verwendet. Die Daten werden dann jeweils über einen Synchronisationsmechanismus ausgetauscht.



Abb. : Reports in MESPAS Web (vereinfacht)

Abb. 3 zeigt schematisch, wie die Erstellung eines Reports grundsätzlich abläuft.



Abb. : Generierung von Reports

### 2.1.1 Reports in der Applikation r5

In der Applikation r5 befinden sich Reports die in der Regel die Informationen, die sich auf dem Screen befinden, für die selektierten Elemente abbilden sollen. Das folgende Beispiel soll dieses Prinzip näher erläutern. Im r5 gibt es den Screen „*Task List*“. Dieser Liste bildet ab, welche Wartungsarbeiten auf einem Schiff erledigt wurden oder zu erledigen sind:



Abb. : Task Liste in r5

Nun kann man die Zeilen selektieren für die man den Report erhalten möchte und den Report mittels Klick auf den Button ausführen. Für die selektierte Zeile erhält man nun folgenden Report im PDF Format:



Abb. : Beispiel Report Task Liste

In einigen Fällen gibt es auf einem Screen einen Übersichts- und einen Detailreport. Ein Übersichtreport bildet im Wesentlichen das ab, was auf dem Screen zu sehen ist während ein Detailreport genauere Information über die ausgewählte Zeile liefert.



Abb. : Report-Button

Zusätzlich kann zwischen PDF, Excel, Word und HTML als Ausgabeformat gewählt werden.



Abb. : Ausgabeformat

### 2.1.2 Reports auf der MRE

Wie bereits erwähnt ist die MRE von der Applikation entkoppelt. Die Reports werden auf einem eigenen Server, genannt iServer betrieben. Dabei wird zwischen Standard Reports, die für alle Benutzer verfügbar sind, und kundenspezifischen Reports, die nur für Benutzer der entsprechenden Firma sichtbar sind, unterschieden.



Abb. : Dokumentenstruktur in der MRE

MRE Reports haben eine andere Aufgabe als jene in der Applikation r5. Die MRE aggregiert und berechnet Zahlen und Fakten, die für das Zielpublikum von Interesse sind. Besonders im Einkaufsbereich, genannt „*Procurement*“, sind Übersichten über Bestellungen wie z.B. Volumen, oder Bestellungen nach Kategorie, etc. wichtig für die Reedereien. Auch im Bereich Wartung („*Planned Maintenance System*“, PMS) sind Reports die einen zusammenfassenden Charakter aufweisen von grossem Interesse.

Um einen Report auf der MRE auszuführen muss sich der Benutzer zunächst einloggen. Dann kann er über den Baum (siehe Abb. 5) zum entsprechenden Verzeichnis gelangen und den gewünschten Report ausführen. Beispiel:



Abb. : Auswahl Beispiel-Report: Übersicht über die Wartungen der Motoren eines Schiffes

In diesem Beispiel erhält der Benutzer eine Übersicht über die Wartungsarbeiten die an den Motoren eines Schiffes durchgeführt wurden. Da dieser Report einen Parameter in Form einer Eingabe erwartet (in diesem Fall die Auswahl des Schiffes) wird der Benutzer danach gefragt:



Abb. : Auswahl eines erforderlichen Parameters für einen MRE Report

Das Resultat sieht wie folgt aus:



Abb. : Resultat Beispiel-Report: Übersicht über die Wartungen der Motoren eines Schiffes

Das Resultat zeigt die Art der Aktivität, die Beschreibung der Komponente, das Wartungsintervall in Stunden, die Einheiten, sowie Daten der Wartung wie Durchführungsdatum, Laufzeit der Komponente, etc.

Ein wichtiger Unterschied zwischen der MRE und den Reports im r5 ist, dass die MRE fähig ist, die Ausführung von Reports zu terminieren. Viele Benutzer wünschen sich in regelmässigen Abständen den gleichen Report mit den jeweils aktuellen Daten und sie möchten diesen auch automatisch erhalten. Der Benutzer kann diese Terminierung selbst vornehmen. Bei jedem Report findet sich ein Symbol in Form einer Uhr:



Abb. : Symbol zur Terminierung von MRE Reports

Wählt man dieses Symbol aus gelangt man zum Fenster zur Terminierung von Reports. Darin kann man die benötigten Informationen hinterlegen:



Abb. : Terminierung von Reports (Einstellungen)

Im Bereich **Schedule** hinterlegt man die allgemeinen Informationen wie Jobname und die Ausführungsdaten. Dabei kann man zwischen drei Optionen auswählen:

* Einmalige, sofortige Ausführung des Reports („*Right now*“)
* Einmalige Ausführung des Reports an einem bestimmten Datum (*„Once“*)
* Wiederkehrende Ausführung im angegeben Intervall (*„Recurring“*)

Im Bereich **Parameter** kann, falls erforderlich, der entsprechende Parameter der für den Report angegeben wurde hinterlegt werden.

Im Bereich **Save As** werden die Information hinterlegt, die für den Report nach der Ausführung benötigt werden. Dabei kann angegeben wo das Output-Dokument abgelegt werden und ob der Report per E-Mail verschickt werden soll.



Abb. : Terminierung von Reports - Optionen für den Output

Die entsprechende E-Mail Adresse ist im Profil des Benutzers hinterlegt.

### 2.1.3 Reports auf der Web-Plattform

Wie in Kapitel 1.1 erwähnt ist die Web-Plattform seit 2014 in Entwicklung. Im August 2015 wurde die erste Version produktiv geschaltet. Es handelt sich dabei um einen Client für die Lieferanten von Ersatzteilen und Materialien. Für diesen Client wurde bereits eine Reporting-Lösung implementiert. Ein grosser Unterschied zu den Reports in r5 ist die Auswahl der Elemente. In r5 können die Elemente die im Report erscheinen sollen ausgewählt werden. Auf der Web-Plattform (MW) werden Reports nur für alle angezeigten Elemente ausgeführt. Beispiel:



Abb. : Übersicht der Offerten

In Abb. 12 ist die Übersicht der Offerten, die der Benutzer erstellt hat, zu sehen. Möchte der Benutzer nun eine Übersicht dieser Offerten in einem Report aufbereitet haben braucht er nun nur den Button „PDF Report“ zu bestätigen und er hält sofort den gewünschten Report:

## 2.2 Datenquellen

Wie in Abb 3. zu sehen ist wird nach dem Laden des Report Templates die Datenquelle gelesen. Die Datenquelle ist ein Element in BIRT das festlegt, auf welche Art und Weise die Daten aufbereitet werden. Dabei unterscheidet man grob zwischen zwei Kategorien: Scripted Data Sources und Datenbanken. Die Scripted Data Sources verwendet man dann, wenn man die Daten nicht mittels SQL holen möchte, sondern mittels Java aufbereitet.

## 2.3 Datenstrukturen

Mit Datenquellen alleine können noch keine Resultate abgebildet werden. Dazu benötigt man sogenannte Datasets. Ein Dataset ist im Grunde nichts anderes als ein relationales Modell einer Untermenge von Daten. Diese Daten können im Report verwendet werden. Um ein Dataset zu erzeugen kann man eine SQL Abfrage schreiben (für Datasets die als Basis eine Datenbank haben) oder diese programmatisch abfragen.

## 2.4 Report Templates

BIRT Reports werden als Template mittels dem BIRT Report Designer erstellt. Die BIRT Runtime ist grundsätzlich abwärtskompatibel jedoch kann eine Runtime keinen Report ausführen der mit einem neueren BIRT Report Designer erstellt wurde. Nachfolgend ist eine Übersicht der Runtime und der dazu benötigten Ausgabe des BIRT Report Designers:

|  |  |  |
| --- | --- | --- |
| **Plattform** | **Runtime Version** | **BIRT Designer Version** |
| r5 | 2.5.2 | 2.5.1 |
| MRE | 2.6.2 | 2.6.2 |
| MESPAS Web | 3.2.23 | 3.2.23 |

Tabelle : Verwendete BIRT Version

## 2.5 Schwächen

Das aktuelle System funktioniert in allen Belangen fehlerfrei. Da jedoch in der neuen Lösung keine einzelne, zentrale Datenbank zur Verfügung muss die Verarbeitung neu entworfen werden. Zudem ist die proprietäre Lösung für die MRE sehr kostspielig, daher ist eine eigene Lösung nur schon aus diesem Grund begrüssenswert.

# 3 Anforderungsanalyse

Um die Anforderungen an die neue Lösung festzuhalten werden User Stories verwendet. Dabei unterscheidet man zwischen funktionalen und nicht-funktionalen Anforderungen. In diesem Kapitel werden die Anforderungen ermittelt und entsprechend dokumentiert.

## 3.1 Stakeholder

Folgende Stakeholder sind in der neuen Lösung involviert:

* Benutzer: regulärer Benutzer des Reporting.
* Administrator: administriert die Umgebung
* Projektleiter: ist zuständig für die Kunden. Sie teilen der Entwicklung mit, ob ein Kunde eine Reporting-Lösung braucht.
* Entwickler: zuständig für die Entwicklung und Bereitstellung von Reports
* System Administrator: zuständig für die Bereitstellung der Hardware und den Betrieb

## 3.2 Use Case

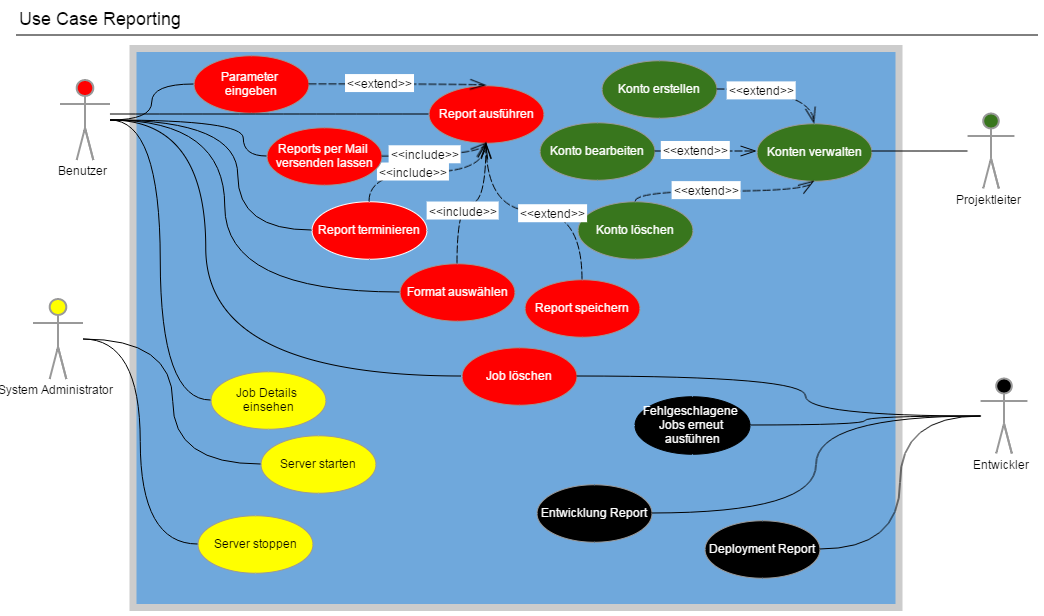


Abb. : Use Case Diagramm

## 3.3 User Stories

### 3.3.1 Allgemeines

In der neuen Lösung sollen die Applikationsreport und die funktionalen Eigenschaften der MRE berücksichtig werden. Dabei sollen die Bereiche der neuen Web-Plattform die einen Report benötigen, einen solchen erhalten und die MRE-Funktionalität soll in einem eigenen Microservice als Modul berücksichtigt werden. Da die Implementierung der Applikationsreport bereits vorhanden ist sind die User Stories nur für die MRE Funktionalitäten.

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-01** | **Priorität:** mittel |
| Titel | Kundenspezifische Reports | |
| Beschreibung | Als Benutzer möchte ich Bereich haben in dem meine kundenspezifischen Reports abgelegt sind. | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Die kundenspezifischen Reports sind von den Standard Reports getrennt | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-02** | **Priorität:** mittel |
| Titel | Report-Kategorien | |
| Beschreibung | Als Benutzer möchte ich, dass Report nach Kategorien gruppiert werden. | |
| Bemerkungen | Die Applikation kennt verschiedene Module und die Reports sollen in Kategorien, die sich an diesen Modulen orientieren, eingeteilt werden. | |
| Akzeptanzkriterien | Die Reports sind korrekt kategorisiert.  Reports die keiner bestimmten Kategorie zugeordnet werden können sind in einer allgemeinen Kategorie eingeteilt. | |

### 3.3.2 Automatisierte Reports

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-03** | **Priorität:** hoch |
| Titel | Terminierung von Reports | |
| Beschreibung | Als Benutzer möchte ich einen Report terminieren können | |
| Bemerkungen | Ein Benutzer soll diesen Schedule selbst setzen können. | |
| Akzeptanzkriterien | Ein terminierter Report wird im korrekten Intervall ausgeführt  Ein terminiert ausgeführter Report enthält die korrekten Daten. | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-04** | **Priorität:** hoch |
| Titel | Löschen von Schedules | |
| Beschreibung | Als Benutzer möchte ich die Ausführung von terminierten Reports stoppen können. | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Ein terminierter Report ist nach der Lösung nicht mehr sichtbar. | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-05** | **Priorität:** hoch |
| Titel | Mail-Versand | |
| Beschreibung | Als Benutzer möchte ich, dass ein Report automatisch per Mail an mich versendet wird. | |
| Bemerkungen | Abhängigkeit von Story S3 | |
| Akzeptanzkriterien | Ein terminierter Report wird im korrekten Intervall ausgeführt  Ein terminiert ausgeführter Report enthält die korrekten Daten. | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-06** | **Priorität:** niedrig |
| Titel | Status von terminierten Reports | |
| Beschreibung | Als Benutzer möchte ich, dass ich eine Übersicht über meine terminierten Reports mit deren Status einsehen kann. | |
| Bemerkungen | Die Übersicht soll alle terminierten Reports anzeigen, unabhängig von dessen Status | |
| Akzeptanzkriterien | Der Reporting-Service enthält einen Bereich mit einer Status-Übersicht  Die Statusmeldungen liefern Informationen über den Job. | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-07** | **Priorität:** mittel |
| Titel | Erneutes Ausführen fehlgeschlagener Jobs | |
| Beschreibung | Als Benutzer möchte ich, dass, wenn ein Ausführungsjob fehlschlägt ein Administrator (Entwickler) diesen manuell erneut ausführen kann. | |
| Bemerkungen | Ein Entwickler ist bei der MESPAS für das Nachsendern von Reports verantwortlich und nicht der Support. | |
| Akzeptanzkriterien | Ein fehlgeschlagener Ausführungsjob kann durch einen Entwickler mit einem Klick erneut ausgeführt werden. | |

### 3.3.3 Report-Eigenschaften

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-08** | **Priorität:** mittel |
| Titel | Formate | |
| Beschreibung | Als Benutzer möchte ich, dass verschiedene Output-Formate unterstützt werden. | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Bei der Ausführung eines Report kann der Benutzer zwischen folgenden Formaten als Output auswählen: PDF, Excel, Word Dokument | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-09** | **Priorität:** hoch |
| Titel | Report-Parameter | |
| Beschreibung | Als Benutzer möchte ich, dass Reports Parameter entgegen nehmen können | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Parameter wird korrekt verarbeitet. | |

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-10** | **Priorität:** hoch |
| Titel | Diagramme | |
| Beschreibung | Als Benutzer möchte ich, dass Reports mittels Charts und Diagrammen Informationen kompakt darstellen können. | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Reports unterstützen die gängigen Chart-Format (Säulen-, Kuchen-, Linienchart, etc.) | |

### 3.3.4 Datensicherheit

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-11** | **Priorität:** hoch |
| Titel | Mandantentrennung | |
| Beschreibung | Als Benutzer möchte ich, dass meine Daten nicht für Benutzer anderer Firmen sichtbar sind. | |
| Bemerkungen |  | |
| Akzeptanzkriterien | Der Reporting-Service stellt sicher, dass die Mandantentrennung gewährleistet ist. | |

### 3.3.5 Plattform-Eigenschaften

|  |  |  |
| --- | --- | --- |
| **Story Nr.** | **US-12** | **Priorität:** mittel |
| Titel | Archiv-Funktion | |
| Beschreibung | Als Benutzer möchte ich, dass ich meine ausgeführten Reports archivieren kann | |
| Bemerkungen | Standard-Reports werden nicht archiviert. Diese Story kann daher nur für kundenspezifische Reports umgesetzt werden. | |
| Akzeptanzkriterien | Die Plattform erlaubt es, einen Report so einzustellen, dass ein Report archiviert werden kann. | |

## 3.4 Funktionale Anforderungen

Zu den im vorherigen Kapitel erarbeiteten Stories müssen nun die Anforderungen entwickelt werden.

### 3.4.1 Anforderungen Allgemeines

|  |  |  |
| --- | --- | --- |
| **Anforderung Nr.** | R-002 | **Priorität:** mittel |
| **Story** | US-01 |  |
| Titel | Datenbank | |
| Beschreibung | Die kundenspezifischen Reports sollen von den Standardreport getrennt sein. Um dies zu bewerkstelligen soll das Datenbankschema entsprechend entwickelt werden. | |
| Bemerkungen | Blockiert von: R-015 (Datenbank) | |
| Akzeptanzkriterien | Benutzer A kann die kundenspezifischen Reports (KSR) von Kunde B nicht sehen.  Benutzer B kann die KSR von Benutzer A nicht sehen.  Benutzer A und B können sämtliche Standard-Reports sehen. | |

|  |  |  |
| --- | --- | --- |
| **Anforderung Nr.** | R-001 | **Priorität:** mittel |
| **Story** | US-01 |  |
| Titel | Report-Bereiche | |
| Beschreibung | Das Datenbankschema ist so zu gestalten, dass kundenspezifische von den Standard Reports getrennt werden können. Dies kann zum Beispiel über einen entsprechenden Schlüssel im Dokument gesteuert werden. | |
| Bemerkungen | Möglichkeit: | |
| Akzeptanzkriterien | Die kundenspezifischen Reports sind von den Standard Reports getrennt | |

## 3.5 Nicht-Funktionale Anforderungen

Neben den funktionalen Anforderungen gibt es einige nicht-funktionale Anforderungen die die neue Lösungen erfüllt. Diese sind wie folgt definiert:

### 3.4.1 Performance

Die Plattform soll eine akzeptable Performance liefern können. In Zahlen ausgedrückt heisst das konkret, dass keine Ausführung länger als 60 Sekunden dauern soll.

### 3.4.2 Verfügbarkeit

Der Reporting-Microservice ist nur dann verfügbar, wenn die ganze Plattform verfügbar ist. Diese Verantwortlichkeit liegt beim Operations-Team, das für die Infrastruktur verantwortlich ist.

### 3.4.3 Optik

Die MESPAS AG verfügt über interne Richtlinien über das Corporate Design. Die Reports haben sich bezüglich Farben, Schriftart und Logos an diesen Richtlinien zu orientieren.

## 3.5 Technische Anforderungen

Da die neue Plattform über einen komplett anderen Aufbau als die bestehende MRE verfügt gibt es einige technische Anforderungen, die erfasst werden müssen.

### 3.5.1 Datenbank

### 3.5.2 Scheduler

Da mit der Story Nr. S3 Reports terminiert werden können sollen benötigt die Lösung einen Scheduler. Dieser solle Benutzer Schedules regelmässig prüfen und wenn eine Intervallsgrenze erreicht den entsprechenden Report oder die gewünschte Aktion ausführen. Zunächst sind bestehende Lösungen zu evaluieren.

### 3.5.3 Daten

In der bisherigen Lösung sind sämtliche Daten in einer zentralen MySQL Datenbank abgelegt. In MESPAS Web verfügt jedoch jeder Microservice über eine eigene Datenbank. Daraus ergibt sich die technische Anforderung, dass die Lösung in der Lage sein muss die Daten aus verschiedenen Datenbanken zu holen. Es ist daher ein Lösungsweg zu entwickeln wie die bewerkstelligt werden kann.

### 3.5.4 Datenobjekte

Die Daten sämtlicher Reports sollen als Java Objekt aufbereitet werden. Wenn Bibliotheken benutzt werden ist dies zu berücksichtigen.

### 3.5.5 Logging

In der MRE findet in der heutigen Form kein echtes Logging statt. Wenn ein Fehler auftaucht erhält der Benutzer zwar eine Fehlermeldung aber ein Stacktrace wird nicht geloggt. In der neuen Lösung ist daher darauf zu achten, dass allfällige Exceptions korrekt geloggt werden.

### 3.3.6 Migration

Aus Gründen der Einfachheit der Migration ist sicherzustellen, dass die bestehenden BIRT Templates wiederverwendet werden können. Da die Firma MESPAS AG im Jahre 2015 ein Rebranding durchgeführt hat ist zu klären, ob die bisherigen Reports dem neuen Corporate Design entsprechend umgestaltet werden sollen.

# 4 Recherche

Nachdem die Anforderungen erfasst wurden geht es nun darum, dass die benötigten Werkzeuge evaluiert werden.

## 4.1 Benötigte Werkzeuge

### 4.1.1 Scheduling

Um Reports terminieren zu können wird ein Werkzeug benötigt, dass in der Lage ist, Jobs in regelmässigen Abständen durchzuführen. Es ist darauf zu achten, dass der Benutzer kein Fachwissen benötigt um diese Schedules zu setzen.

### 4.1.2 Rendering

## 4.2 Evaluation der Werkzeuge

## 4.3 Ergebnisse

# 5. Konzept

## 5.1 Request-Format

## 5.2. Schnittstellen

## 5.3 Datenaufbereitung

## 5.5 Layout-Bearbeitung

## 5.6 Rendering

## 5.7 Deployment

# 6 Proof of Concept

## 6.1 Bau eines Prototypen

## 6.2 Definition Beispiel-Reports

## 6.3 Datenaufbereitung für Beispiel-Reports

## 6.4 Ergebnisse der Beispiele

# 7 Testing

## 7.1 Testkonzept

## 7.2 Unit Tests

## 7.3 Integrationstest

## 7.4 Regression Test

## 7.5 Acceptance Test

# 8 Fazit und Erkenntisse

## 8.1 Zusammenfassung

## 8.2 Fazit

## 8.3 Ausblick

# 9 Anhang

## 9.1 Anhang A: Bilderverzeichnis

## 9.2 Anhang B: Tabellenverzeichnis

## 9.3 Anhang C: Abkürzungsverzeichnis

|  |  |
| --- | --- |
| BIRT | Business Intelligence Reporting Tools. Tool von Actuate für die Erstellung von Reports |
| KSR | Kundenspezifischer Report |
| MRE | MESPAS Reporting Engine |
| MW | MESPAS Web (Webplattform der MESPAS AG) |
| PMS | Planned Maintenance System (System für geplante Wartungen) |

## 9.3.Anhang D: Literaturverzeichnis