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)

**Danksagung**

Ich möchte mich an dieser Stelle bedanken, die mich beim Erstellen dieser Arbeit in irgendeiner Weise unterstützt haben. Besonders hervorheben möchte ich:

Simon Kunz, MSc Statistics ETH für die Betreuung

Simon Regenscheit, dipl. el. Ing. ETH für die Unterstützung in punkto Arbeitszeit

Lis Steiger, lic. phil., für das Korrekturlesen

Christian Vetterlein, Marketingmitarbeiter, für das Korrekturlesen

Inhaltsverzeichnis

[1 Einleitung 5](#_Toc423339658)

[1.1 Ausgangslage 5](#_Toc423339659)

[1.2 Ziele der Arbeit 6](#_Toc423339660)

[1.3 Aufgabenstellung 6](#_Toc423339661)

[1.3.1 Ist-Analyse 6](#_Toc423339662)

[1.3.2 Anforderungsanalyse 6](#_Toc423339663)

[1.3.3 Recherche 6](#_Toc423339664)

[1.3.4 Konzept 6](#_Toc423339665)

[1.3.5 Proof of Concept 6](#_Toc423339666)

[1.3.6 Testing 6](#_Toc423339667)

[1.3.7 Abschluss 6](#_Toc423339668)

[2 Bisherige Situation (Ist-Analyse) 7](#_Toc423339669)

[2.1 Einleitung 7](#_Toc423339670)

[2.1.1 Reports in der Applikation r5 7](#_Toc423339671)

[2.1.2 Reports auf der MRE 8](#_Toc423339672)

[2.2 Datenquellen 8](#_Toc423339673)

[2.3 Datenstrukturen 8](#_Toc423339674)

[2.4 Implementierung 8](#_Toc423339675)

[2.5 Werkzeuge 8](#_Toc423339676)

[3 Anforderungsanalyse 9](#_Toc423339677)

[3.1 Funktionale Anforderungen 9](#_Toc423339678)

[3.2 Nicht-Funktionale Anforderungen 9](#_Toc423339679)

[3.3 Ergebnisse 9](#_Toc423339680)

[4 Recherche 10](#_Toc423339681)

[4.1 Verfügbare Werkzeuge 10](#_Toc423339682)

[4.2 Evaluation der Werkzeuge 10](#_Toc423339683)

[4.3 Ergebnisse 10](#_Toc423339684)

[5. Konzept 11](#_Toc423339685)

[5.1 Request-Format 11](#_Toc423339686)

[5.2. Schnittstellen 11](#_Toc423339687)

[5.3 Datenaufbereitung 11](#_Toc423339688)

[5.5 Layout-Bearbeitung 11](#_Toc423339689)

[5.6 Rendering 11](#_Toc423339690)

[5.7 Deployment 11](#_Toc423339691)

[6 Proof of Concept 12](#_Toc423339692)

[6.1 Bau eines Prototypen 12](#_Toc423339693)

[6.2 Definition Beispiel-Reports 12](#_Toc423339694)

[6.3 Datenaufbereitung für Beispiel-Reports 12](#_Toc423339695)

[6.4 Ergebnisse der Beispiele 12](#_Toc423339696)

[7 Testing 13](#_Toc423339697)

[7.1 Testkonzept 13](#_Toc423339698)

[7.2 Unit Tests 13](#_Toc423339699)

[7.3 Integrationstest 13](#_Toc423339700)

[7.4 Regression Test 13](#_Toc423339701)

[7.5 Acceptance Test 13](#_Toc423339702)

[8 Fazit und Erkenntisse 14](#_Toc423339703)

[8.1 Zusammenfassung 14](#_Toc423339704)

[8.2 Fazit 14](#_Toc423339705)

[8.3 Ausblick 14](#_Toc423339706)

[9 Anhang 15](#_Toc423339707)

[9.1 Anhang A: Bilderverzeichnis 15](#_Toc423339708)

[9.2 Anhang B: Tabellenverzeichnis 15](#_Toc423339709)

[9.3 Anhang C: Abkürzungsverzeichnis 15](#_Toc423339710)

[9.3.Anhang D: Literaturverzeichnis 15](#_Toc423339711)

# 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 wurden mit BIRT (Business Intelligence Reporting Tools) erstellt.

### 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 Übersicht- 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

## 2.3 Datenstrukturen

### 2.3.1 SQL

### 2.3.2 Scripted Reports

## 2.4 Implementierung

## 2.5 Werkzeuge

# 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 Anforderung ermittelt und entsprechend dokumentiert.

## 3.1 Funktionale Anforderungen

## 3.2 Nicht-Funktionale Anforderungen

## 3.3 Ergebnisse

# 4 Recherche

## 4.1 Verfügbare Werkzeuge

## 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 Erstellungn von Reports |
| MRE | MESPAS Reporting Engine |
| MW | MESPAS Web (Webplattform der MESPAS AG) |
| PMS | Planned Maintenance System (System für geplante Wartungen) |

## 9.3.Anhang D: Literaturverzeichnis