Software Projekt: Projektbeschrieb

Student:	Raphael Matile Wingertweg 12 7015 Tamins Tel: +41 79 785 54 53 E-Mail: `raphael.matile@uzh.ch` Matrikel-Nr: `12-711-222`
Arbeitgeber:	RaiseNow c/o getunik AG Hardturmstrasse 101 8005 Zürich Tel: +41 44 533 56 20
Verantwortliche Kontaktperson:	Tsuyoshi Ito Technical Director E-Mail: `tsuy.ito@raisenow.com`
Beginn und Enddatum des Praktikums:	1. Juli 2015 - 14. August 2015 42 Stunden pro Woche

Kontext

Seit 2008 bietet RaiseNow AG Dienstleistungen im Bereich Digital Fundraising an. RaiseNow ist inzwischen ein führender Anbieter von Online-Fundraising-Lösungen. Der Fokus liegt auf leistungsstarken, benutzerfreundlichen Online-Fundraising-Tools, die einfach zu bedienen und anzuwenden sind. Die Philosophie: Hervorragendes intuitives Nutzererlebnis auch bei starken Features. RaiseNow gestaltet damit das Online-Fundraising effizient und gleichzeitig bequem und bietet eine umfassende Lösung, die auf die Bedürfnisse von (Nonprofit) Organisationen angepasst ist.

RaiseNow entwickelt ein einheitliches API für die verschiedensten Zahlungsmethoden, unter anderem Kreditkarten, SMS-Zahlungen sowie PayPal. Dieses API vereinfacht die Implementierung von Zahlungsschnittstellen bei Kunden erheblich, da sich die dortigen Entwickler nicht mehr mit den Finessen der einzelnen Methoden auseinandersetzen müssen. Inzwischen greifen um die 150 Kunden auf diese Zahlungsschnittstelle zu. Dabei handelt es sich vor allem um Nonprofit-Organisationen, welche dadurch ihre Spenden entgegennehmen. Für sie gibt es entweder Webwidgets, welche aus einer Zeile Javascript-Code bestehen oder Custom-Implementierungen, welche via REST-API auf diese Zahlungsschnittstelle zugreifen.

Aufgabenstellung

Die RaiseNow Produktpalette besteht aus verschiedenen Applikationen, welche dedizierte Aufgaben übernehmen und somit eine Software-as-a-Service Plattform darstellen. Damit die Authentifizierung zwischen diesen Applikationen sichergestellt werden kann, wird das standardisierte Authorisierungsframework OAuth2 eingesetzt. Die eigentliche Kommunikation zwischen den Applikationen findet entweder über spezielle Events (WebHooks) statt oder kann über das Nachrichten-orientierte Protokoll AMQP erfolgen.

Als zentrale Administrationsoberfläche dient der *RaiseNow Manager*, in welchem jeder Kunde Statistiken über einen Zahlungszeitraum, Details zu einzelnen Transaktionen sowie wiederkehrenden Zahlungen einsehen kann. Des Weiteren ist es ihm ermöglicht, einzelne Applikationen zu konfigurieren.

Weiterhin bietet RaiseNow auch eine Lösung für Crowdfunding für Nonprofit-Organisationen an, die *RaiseNow P2P Plattform*. Sie ermöglicht es jedem, eine eigene Spendenkampagne für einen bestimmten Zweck zu erstellen und ein dafür eigenes SMS Kennwort zu generieren, welches eine Spende an die gewünschte Kampagne sicherstellt. Der Kunde als Kampagnenersteller hat dann die Möglichkeit die einkommenden Zahlungen zu überwachen und weitere Informationen über die Kampagne zu verbreiten.

Durch die grosse Anzahl von Kunden und mehreren tausend Transaktionen ergibt sich eine grössere Menge an Daten, welche in gewissen Applikationen durchsucht werden müssen. Zurzeit werden die Datensätze mittels SQL auf relationalen Datenbanken durchsucht. Durch die Komplexität sowie der Grösse der Datenmenge dauern einzelne Anfragen sehr lange.

Die Aufgaben besteht nun, einen einheitlichen Such-Service zur Verfügung zu stellen, um Daten sowie Objekte zu indexieren und sogenannte Facettenfilter bereitzustellen. Dieser Service sollte später nicht nur den hauseigenen Applikationen zur Verfügung stehen, sondern auch 3rd Party Apps.

Einzusetzende Technologie und grober Projektablauf

Um die Datensätze im Vornherein zu indexieren soll der auf Lucene basierende Suchserver *Elasticsearch* zum Einsatz kommen. Zusätzlich zu der Einarbeitung in diese Technologie und die dazu einzusetzenden Frameworks, soll auch eine Evaluation über die Übermittlung der Daten an den Such-Service (*d.h.* ORM-Events oder AMQP basierte Technologie) vorgenommen werden.

Dazu sollen die folgenden Meilensteine als Übersicht dienen:

1. Meilenstein: Überblick der aktuellen Situation (1 Woche)

Wissensanreicherung über die eingesetzte Elasticsearch Version (1.4.4). Da des Weiteren bereits eine interne Such-API existiert (Filtern von Transaktionen) muss auch deren Architektur und Funktionsweise erarbeitet werden.

2. Meilenstein: Spezifikation der Such-API (2 Wochen)

Das Ziel dieses Meilensteins besteht darin, die bestehende Such-API zu erweitern, so dass Daten von bereits existierenden Tabellen (Entitäten) auf ein Mapping in Elasticsearch transferiert werden können. Die Spezifikation soll ein Vorgehen definieren, anhand welchem eine generische Transformation einer bestehenden Entität vorgenommen wird. Zusätzlich soll diese Transformation konfigurierbar sein (zum Beispiel für Parent-Child-Relations).

Fragestellungen welche in diesem Schritt gelöst werden müssen sind - nicht abschliessend - hier aufgelistet:

- Sind bereits Vorgehen für diese Problemstellung etabliert?
- Wie können Änderungen am primary Datastore (MySQL Datenbank) in Elasticsearch abgebildet werden?
- Welche Strategien beinhalten möglichst wenig Reindexierungen bereits bestehender Datensätze (Documents)?
- Soll ein Custom Such-Format verwendet werden oder soll direkt die API-Spezifikation von Elasticsearch für Suchanfragen übernommen werden?
- Sollen Dritt-Entwickler auch die API-Spezifikation von Elasticsearch verwenden müssen? Wäre eine Abstraktion sinnvoller?
- GET Request mit Query Parameter vs. POST Request mit Body (maximale URL Länge in verschiedenen Browsern)

Verschiedene Ansätze für die RESTful-API sollen erarbeitet und im Bezug auf die Browser-Kompatibilität (Chaos in der Spezifikation von Query-Strings), den Aufwand für Drittentwickler (z.B. ob ein Custom-Format erlernt werden muss), das Ausmass für die internen Änderungen an der bestehenden API sowie auf eine Struktur, welche erweiterbar ist und sich generisch darstellen lässt, evaluiert werden. Schliesslich sollen in einer Teamsitzung die resultierenden Vorgehen nochmals besprochen und ein Favorit ausgewählt werden, sodass die Implementation im darauffolgenden Meilenstein stattfinden kann.

3. Meilenstein: Implementierung der Such-API (2 Wochen)

Anhand der erstellten Spezifikation soll die bestehende API angepasst sowie der Zugang für Dritt-Entwickler bereitgestellt werden. Zusätzlich soll der oben genannte Entity-Mapper implementiert werden.

4. Meilenstein: Dokumentation (1 Woche)

Schlussendlich soll die angepasste Such-API dokumentiert werden: Falls die Entscheidung auf verschiedene Abstraktionen für die API gefallen ist, anhand mehreren Dokumentationen für die interne Entwicklung sowie für Dritt-Entwickler. Auch werden die getroffenen Architektur-Entscheidungen für die Zukunft festgehalten.

Optional: 5. Meilenstein: Client-Referenz Implementierung anhand der RaiseNow P2P Plattform

Im Falle eines raschen Vorankommens kann die restliche Zeit dazu genutzt werden, eine Implementation für die RaiseNow P2P Plattform zu entwickeln. Andernfalls wird dieser Meilenstein während der regulären Arbeitszeit bearbeitet.