

# PROJEKTREFERENZ



Entwicklung eines Master-Daten-Management-Systems zur Konsolidierung von Kundendatensätzen

## PROJEKTAUFTRAG

Die DEVBRAIN IT Solutions GmbH wurde von einem deutschen Reiseunternehmen beauftragt in Kooperation mit anderen Entwicklern aus anderen Firmen, eine Master-Data-Management-Lösung (MDM) bereitzustellen, dessen Ziel es ist, Kundendaten aus verschiedenen anderen Datenbanken des Unternehmens zu konsolidieren, zu bereinigen und zu harmonisieren. Außerdem sollen Kunden-Werbeeinstellungen im Rahmen der aktuellen Datenschutzgrundverordnung mit den Kundendatensätzen ebenfalls verknüpft sein.

Wichtigstes Ziel ist es dabei, eine Vielzahl an gleichzeitigen Anfragen zu bearbeiten und den harmonisierten (goldenen) Kundendatensatz an angeschlossene Unternehmensbereiche zurück zu propagieren. Nebenauftrag ist ein Webinterface zur Verfügung zu stellen, um späteren Datenexperten (Data-Stewarts) die Möglichkeit zu bieten, konsolidierte Kundendaten zu überprüfen und ggf. zu korrigieren, um eine transparente Nachverfolgung zu generieren, die beschreibt, wie ein goldener Kundendatensatz zustande gekommen ist. Es ist außerdem gewünscht, die Werbeeinstellungen eines Kunden aktualisieren zu können; auch dies soll transparent protokolliert sein.

## UMSETZUNG

Ein anfragendes System eines Subunternehmens sendet einen Kundensatz an das neue MDM-System per REST. Das MDM-System speichert den Datensatz unverändert in der Datenbank zur Reproduktion aller Referenzen zu einem generierten goldenen Kundendatensatz. Anschließend werden andere Kundendatensätze per Elasticsearch nach ähnlichen Kunden in AWS (Amazon Web Services) durchsucht. Ein Schwellwert für den Qualitätswert aus Elasticsearch zu jedem Suchergebnis erlaubt es, einen Kunden hinter einem Suchergebnis entweder automatisch oder durch einen Data-Stewart mit dem eingehenden Kundendatensatz in Verbindung zu bringen. Aufgabe des Data-Stewarts ist es dann, die Ähnlichkeit zu belegen und somit das Suchergebnis als gleichen Kunden zu bestätigen oder abzulehnen. Ist dagegen der Schwellwert überschritten, kann der eingehende Kunde hinter dem Suchergebnis automatisch dem eingehenden Kundendatensatz zugeordnet werden. Eine Gemeinsamkeit zweier oder mehrerer Kunden bedeutet, dass diese Kunden den exakt gleichen Kunden repräsentieren. Ihre Angaben aus verschiedenen Systemen können zusammengeführt werden. Für die Suche und die Indizierung wurden die Daten bereinigt, d.h. von textuellen Fehlern, wie Buchstabendrehern befreit.

Bei der Zusammenführung der Daten bleiben die Originaldaten vorhanden, jedoch verschwinden unvollständige Abbilder aus dem System und sind wiederzufinden in dem neuen goldenen Datensatz des Kunden, dessen Gemeinsamkeiten auf die bisherigen originalen Kundendaten verweisen.

Alle Prozesse sind zu jeder Zeit transparent abrufbar und reproduzierbar. Gemeinsamkeiten können wieder aufgelöst werden und Kundendaten können, gemäß der DSGVO, jederzeit gelöscht werden (Recht auf Vergessen).

Damit ein Data-Stewart Suchergebnisse / erkannte Gemeinsamkeiten auswerten kann, ist eine grafische Benutzeroberfläche nötig. Diese wurde mittels Angular und HTML5 als Webinterface realisiert.

In der MDM-Lösung sind explizit Micro-Services gefordert für das MDM-System, die Suche, Permissions (DSGVO Einstellungen), und den Zusammenführungsprozess. Diese wurde ebenfalls für AWS bereitgestellt. Hintergrund ist, dass einzelnen Funktionen voneinander unabhängig skaliert werden können, um späteren Belastungen standhalten zu können.

Um einen reibungslosen Ablauf der anfragenden Schnittstelle zu ermöglichen, werden einige Services per Queuing arbeiten. Das bedeutet, dass es ebenso eine Rest-Schnittstelle gibt, womit sich der Stand und das Ergebnis anfragen lassen. Ist ein Prozess aus dem Queuing beendet, so werden die Ergebnisse an die entsprechenden Stellen zurückgeliefert. Für das Queuing sind separate Systeme erstellt worden, die nur zur Aufgabe haben, aus ihren Queues neue Aufträge abzuholen, zu bearbeiten und nach Abschluss in das anfragende System zurückzuliefern.

Die Arbeiten am Quellcode werden mittels IntelliJ und Eclipse in Java erstellt. Die Kundendatenbank wird per MariaDB (kostenfreie Open-Source Variante basierend auf MySQL) bereitgestellt. Per Git via CodeComit in AWS wird der Source-Code persistiert. Es wurde eine Build-Pipeline in AWS aufgebaut, die rudimentäre Iterationsprozesse startet, sobald ein neuer Code eingeht.

In AWS sind zwei Bereiche eingerichtet. Bereich für Testzwecke: dorthin werden alle gerade entwickelten Codestände publiziert, die nicht für den Produktivbereich verwendet werden und von außen unzugänglich sind. Der zweite Bereich ist der Öffentliche Zugang zum Produkt: dorthin werden fertige Versionen des Testbereichs manuell publiziert. Auf diese Art findet die Iteration des Produkts statt, unabhängig von der aktuellen Entwicklung.

## TECHNISCHES RADAR

|                     |  |
|---------------------|--|
| Tools & Technologie | Bootstrap, HTML5, AJAX, log4j, Jira, Scrum, Confluence, Amazon Web Services (elastic search, code commit), SOA (Service orientierte Architektur), CSS3, IntelliJ, Eclipse, MariaDB, Angular4, Git, JUnit, Integrationstest |
| Methodik            | Continuous Integration, Agile Vorgehensweise, Scrum Board  |
| UX                  | Responsive Design, HTML5 und CSS3  |
| QA                  | UAT (User Acceptance Tests) und Tracking von Defects mittels Jira. Dokumentation per Confluence.   |

## SCREENSHOTS

*Aus Datenschutzrechtlichen Gründen sind Screenshots der Weboberfläche nicht gestattet, da eventuelle Kundendaten, wenn auch Testdaten ersichtlich wären.*