Architekturbeschreibung

MI Praktikumsgruppe 3

Hamburg University of Applied Sciences
lucasandreas.jenss@haw-hamburg.de

1. Architektur

Die Architektur des Prototyps besteht aus drei Hauptkomponenten: Data Extraction, Data Analysis und Visualization, welche der internen Aufteilung der Gruppe entsprechen. Diese Aufteilung ermöglicht den Teams die Komponenten der jeweils anderen Teams als Blackbox zu betrachten, sodass eine Abstimmung lediglich über die Funktionalität (was wird umgesetzt) und die Schnittstellen untereinander, nicht aber über die konkrete Implementation der Komponenten (wie wird es umgesetzt) erfolgen muss.

Im Folgenden wird zuerst die Schnittstelle zwischen den Komponenten und anschließend jede Komponente einzeln beschrieben (Übersicht in Abbildung 3).

1.1. Komponentenschnittstelle

Für den Prototyp des Systems wird als zentrale Schnittstelle zwischen den Komponenten eine einfach relationale Datenbank genutzt (MySQL). Diese Designentscheidung wurde getroffen, um eine möglichst einfache Integration der Komponenten zu ermöglichen, und somit den Fokus auf die für den Prototypen relevanteren Themen legen zu können, wobei die Komplexität innerhalb der Komponenten liegt. Aus der Verwendung einer Datenbank als Schnittstelle ergibt sich der Nachteil, dass die Analyse- sowie die Visualisierungskomponente die Datenbank konstant nach neuen Daten Fragen muss (polling) um diese zeitnah an den Benutzer auszuliefern. Als Ergänzung zur Datenbank könnte man hier eine Message Queue und einen Cache für Abfragen einsetzen.

Diese zentrale Datenbank wird hier als Datawarehouse angesehen, auf dem alle Komponenten operieren. Dabei ist die *Data Extraction* für das sammeln der Basisdaten und Metadaten zuständig. Mittels ETL[1] sind die gesammelten Daten dann in das Datawarehouse Modell zu übertragen (siehe Abbildung 1). In der *Data Anlysis* Komponente werden dann mit OLAP und Data-Mining aus den Daten wertvolle Informationen extrahiert. Diese können dann in der *Visualization* abgefragt und dargestellt werden.

Abbildung 1: Datawarehouse Dimensionen in einem Stern-Schema

1.2. Data Extraction

Die Data Extraction Komponente bildet die Schnittstelle zwischen den sozialen Netzwerken, unserer primäre Datenquelle, und dem Rest des Systems. Für die einzelnen Netzwerke wird hier ein entsprechender Adapter verwendet, welcher generische Daten extrahiert. Außerdem werden Netzwerkspezifische Daten (bei Twitter bspw. Retweets oder Geolocation) gesondert ablegt, damit diese für die Analyse nicht verloren gehen. Die Adapter persistieren die Daten über das `IPersistence-`

[1] Extract, Transform, Load

Manager Interface des PersistenceManagers. Dieser dient als Fassade vor dem Datenbank-Interface, damit die Adapter unabhängig von der genutzten Datenbank sind. Neben den Basisdaten werden noch weitere Metadaten gesammelt, die in die Analyse einbezogen werden können. Hier haben wir z. B. eine Städteliste mit den zugehörigen Georeferenzen.

1.3. Data Analysis

In der Data Analysis Komponente werden die extrahierten Daten zu Informationen konsolidiert. Hier gibt es wie bei der *Data Extraction* für jedes angebundene soziale Netz eine Analyser Komponente, welche die Daten in ein für den InformationExtractor nutzbares Format transformiert, damit diese über die IDataSource Schnittstelle abgefragt werden können. Der InformationExtractor verwendet anschließend weitere Analyse-interne Komponenten um Informationen aus den Daten zu gewinnen. Für den Prototyp sind hier ein KeywordFinder und Categorizer angedacht. Der KeywordFinder soll dabei möglichst alle wichtigen Schlüsselworte identifizieren können. Der Categorizer soll die Daten in verschiedenen Kategorien, wie bspw. Hilfegesuch, Pressemitteilung, etc. einteilen.

1.4. Visualization

Die Visualization Komponente ist dafür zuständig die von der Analyse gewonnen Informationen verständlich und interaktiv darzustellen. Ziel ist es, den in den Katastrophenschutz involvierten Behörden vor allem einen Überblick über die aktuelle Situation zu verschaffen, wie sie sich in den sozialen Netzen wiederspiegelt.

Die Visualisierung besteht aus zwei Komponenten, jene die im Browser ausgeführt wird (Dashboard Client) und jene die auf dem Server ausgeführt wird (Dashboard Server). Um möglichst einfachen Zugriff auf die Benutzeroberfläche zu ermöglichen läuft diese als Webanwendung, womit sie auch über Smartphones und Tablets aufgerufen werden kann. Der Dashboard Client soll die verschiedenen Informationen, wie in dabei in einzelnen Ansichten darstellen (Abbildung 2). Dabei liegt der Fokus darauf die Frage "**Wann** ist **was wo** passiert?" zu beantworten.

- Das "wann" wird mithilfe einer Auswahl des Benutzers getroffen. Der Benutzer wählt dafür einen bestimmten Zeitpunkt bzw. einen Zeitraum (bspw. heute /jetzt, gestern, letzte Woche, heute 12:00–13:30 Uhr) aus. Durch die Auswahl werden dann die durch die Analyse gewonnen Daten visualisiert.
- "Was" zum ausgewählten Zeitraum passiert wird durch die kategorisierten Daten in einem Streamgraphen dargestellt.
- Weiterhin wird mithilfe einer interaktiven Karte der geographische Bezug der Daten visualisiert, um das "wo" zu beantworten.

Durch diese miteinander verbundenen Ansichten soll damit ein Lagebild zu einem bestimmten Kontext aus verschiedenen Sichtweisen schnell vermittelbar sein.

Der Dashboard Server stellt die Webanwendung bereit und liefert die für den Client erforderlichen Daten aus. Dabei greift dieser auf die Datenbank zu, welche durch die Data Analysis Komponente erzeugt wurde.

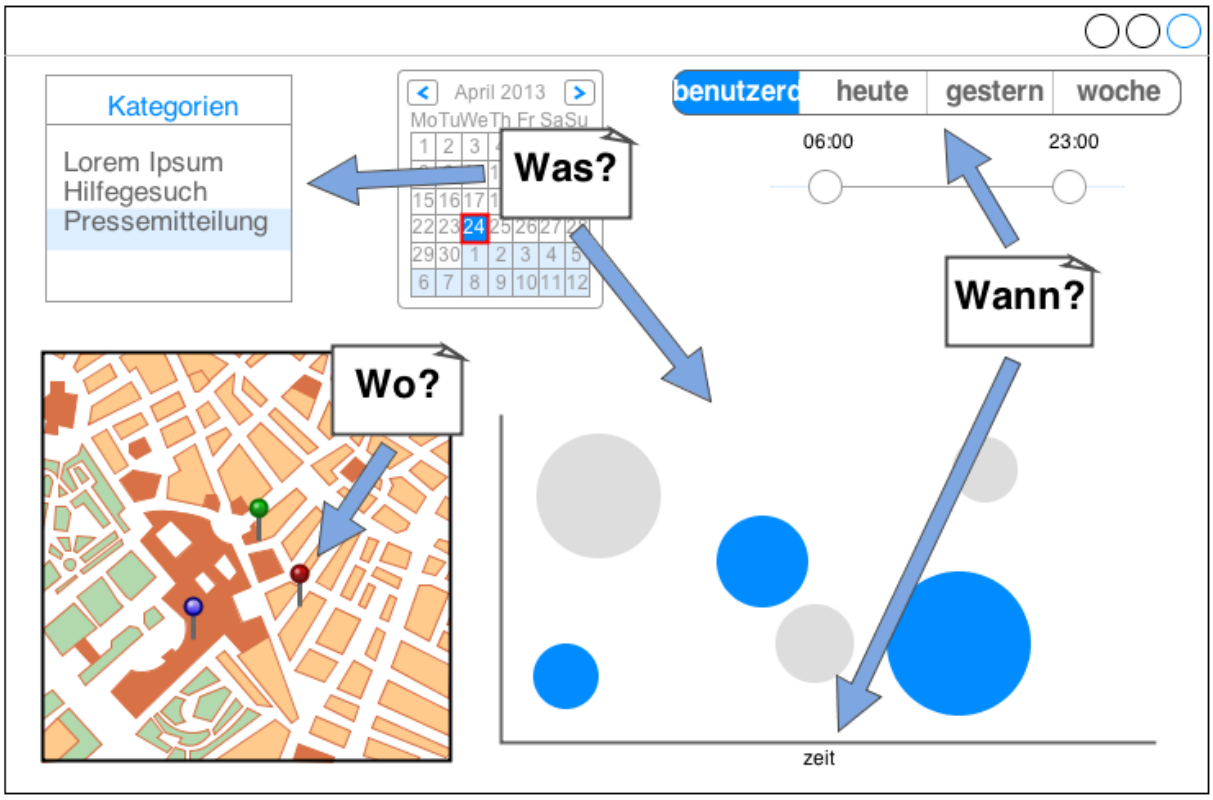

Abbildung 2: Mockup zur Visualisierung der Informationen in einem Dashboard

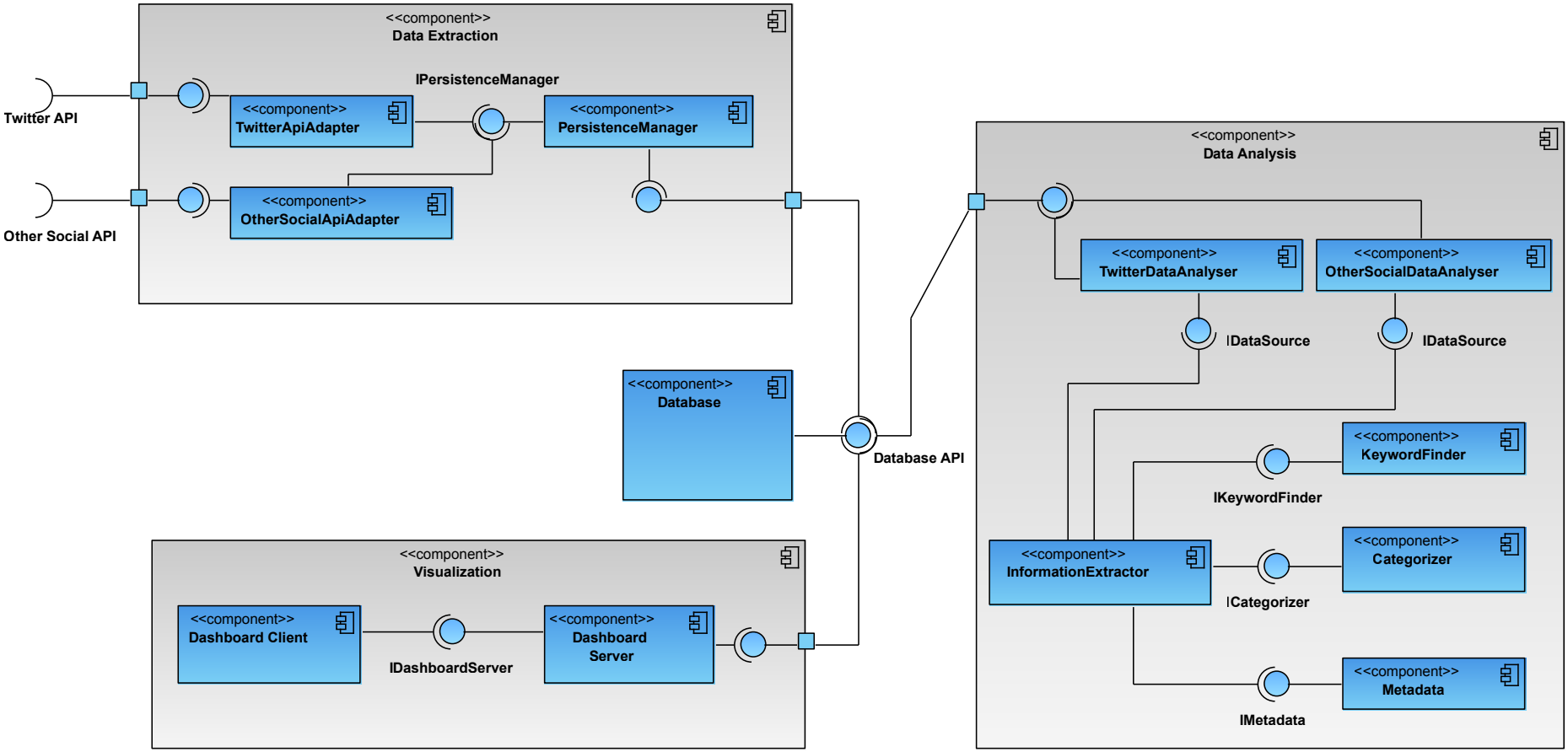

Abbildung 3: UML Komponentensicht auf die Architektur

2. Datenmodell

Das vorliegende Modell beschreibt die aus unserer Sicht vorhandenen fachlichen Daten. Diese sind grob in drei Kategorien zu unterteilen: Rohdaten, verarbeitete Daten, sowie applikationsspezifische Daten. Da das Ziel zunächst ein Prototyp ist, fließt dieses in die Entwicklung des Modells mit ein.

Die Rohdaten sind die in der Datenextraktion gesammelten Daten aus den Sozialen Netzwerken, sowie potenziell weitere Informationsquellen (z.B. Pegelstände, Trenderkennung etc.). Die Gesamthierarchie der Daten setzt sich daher aus verschiedenen Ebenen zusammen: Grundsätzlich besitzen Daten einen eindeutigen Identifikator und eine Iteration, welche es uns ermöglicht die Datensätze grob nach Erfassungsdatum zu selektieren. Daten aus den Sozialen Medien besitzen im Modell zusätzlich einen Autor, dieser ist nur ein Text welcher einen Namen darstellt (diese Einschränkung ist nötig um nicht direkt personenbezogene Daten zu sammeln). Des Weiteren besitzt jeder Datensatz der sozialen Medien ein spezifisches Veröffentlichungsdatum.

Da sich unser Prototyp zunächst auf Twitter als Plattform bezieht, nutzen wir im ersten Schritt nur Tweets als Informationsquelle. Diese enthalten im Datenmodell den eigentlichen Inhalt, die Anzahl der aktuellen Retweets und Follower (des Autors), sowie die Geolocation und einen vom Absender definierten Wohnort, sowie eine Referenz auf einen möglichen ursprünglichen Tweet. Diese Angaben sind allerdings nur dann vorhanden, wenn sie vom Verfasser des Tweets zur Verfügung gestellt wurden. Darüber hinaus verfügt jeder Tweet über beliebig viele Hashtags, welche ihm zugeordnet sind.

Die verarbeiteten Daten sind zunächst eine reine Kopie der Rohdaten und werden durch die Analysegruppe um gewonnene Informationen erweitert. Für das erste Anwendungsszenario soll daher zunächst jedem Tweet eine Kategorie zugewiesen werden. Diese soll grob den Inhalt des Tweets wiedergeben. Weitere Informationen sind ebenfalls geplant. Die dritte Kategorie, welche im Modell genannt wird, gehört zur eigentlichen Applikation. Diese enthält ein Dashboard mit Informationen, welche in beliebig vielen Widgets dargestellt werden. Darüber hinaus sind für die Applikation Nutzer, samt eines rudimentären Authentifikationsverfahrens anhand einer E-Mail Adresse und eines Passworts, vorgesehen.

Für den bisherigen Prototypen fallen alle genannten Informationen auf einer Ebene zusammen, sodass die Tabellen in der Datenbank dementsprechend gestaltet ist.

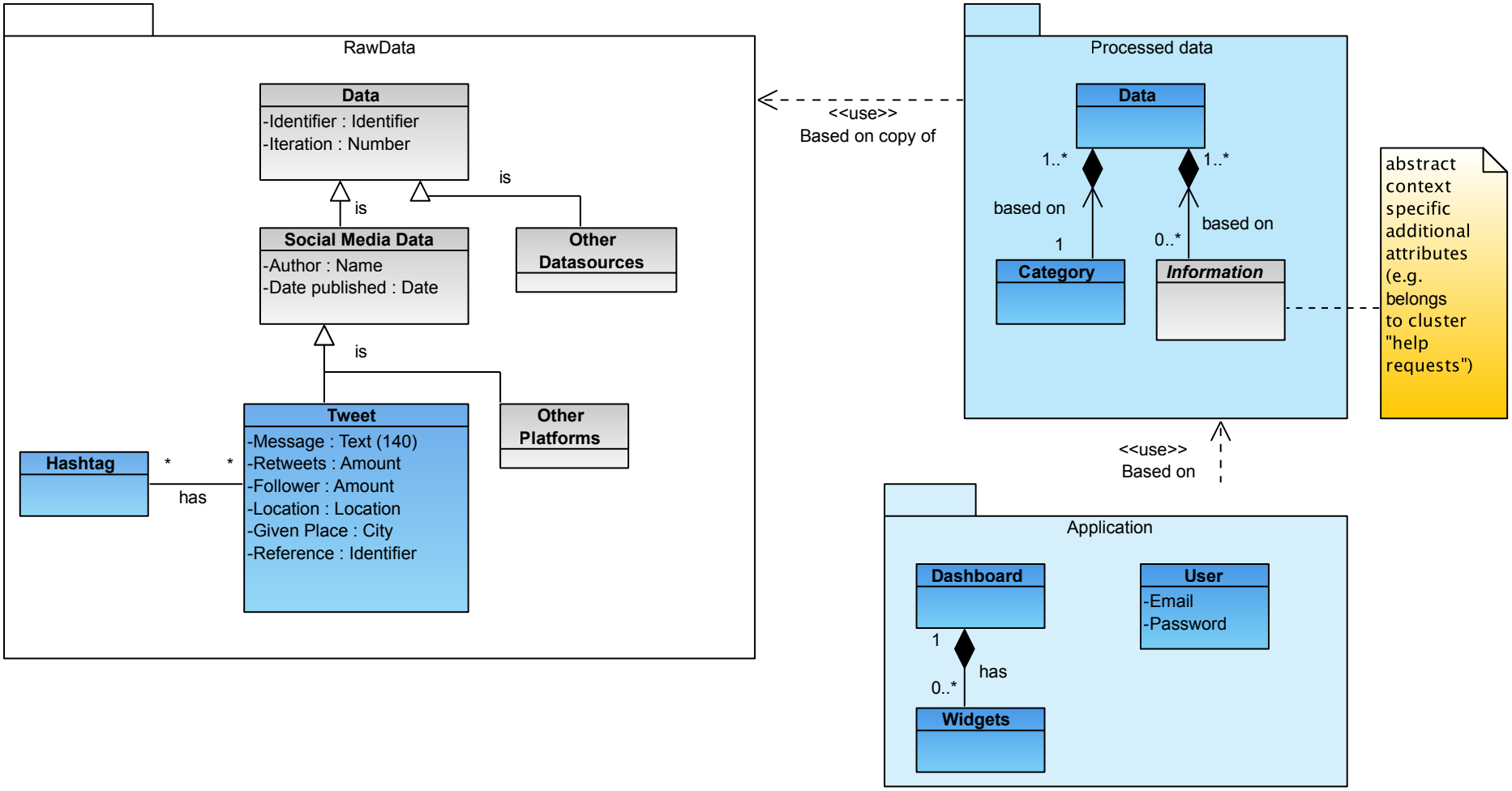

Abbildung 4: Fachliches Datenmodell