

# **SOFTWARETECHNIK PRAKTIKUM**

**Entwurfsbeschreibung**

**Sommersemester 2017**

Projekt: ENA-17: Big Data & KINECT - Eingangsassistent

Vladimir Moroz, Christian Fuß, Marco Neumann, Patrick Oswald,

Sven Oswald, Chiara Hergl

## Inhaltsverzeichnis

<b>1</b>	<b>Allgemeines</b>	<b>2</b>
<b>2</b>	<b>Produktübersicht</b>	<b>2</b>
<b>3</b>	<b>Grundsätzliche Struktur- und Entwurfsprinzipien</b>	<b>2</b>
<b>4</b>	<b>Struktur- und Entwurfsprinzipien einzelner Pakete</b>	<b>3</b>
4.1	Gestenerkennung und Datenbank . . . . .	3
4.2	Web GUI und Registry Datenbank . . . . .	3
4.3	Server-Client C# . . . . .	3
4.4	C# Kinect Anbindung und GUI . . . . .	3
4.5	Speech-to-text Modul . . . . .	4
4.6	Text-to-speech Modul . . . . .	5
4.7	Text-to-Visualization . . . . .	5
4.8	Guidepost Modul . . . . .	6
4.9	Wetter Klasse . . . . .	7
4.10	Reaktions Klasse . . . . .	8
4.11	Email Benachrichtigung . . . . .	8
4.12	Datenbank . . . . .	8
<b>5</b>	<b>Datenmodell</b>	<b>9</b>
<b>6</b>	<b>Glossar</b>	<b>12</b>

## 1 ALLGEMEINES

---

### 1 Allgemeines

Dieses Dokument soll die Designentscheidungen für den zu entwickelnden Eingangsassistenten darstellen und eine Übersicht über die notwendigen Ressourcen der Pakete geben.

### 2 Produktübersicht

Der Eingangsassistent wird verschiedene Funktionen erfüllen können. Er soll einem Gastnutzer über Mensch - Maschinen Interaktion, durch einen Kinectsensor, die Möglichkeit geben Informationen zu erlangen. Weiterhin soll der Eingangsassistent dem Systemnutzer ermöglichen, über den Kinectsensor, gewonnene Informationen zur Auswertung zu sammeln und auszugeben.

### 3 Grundsätzliche Struktur- und Entwurfsprinzipien

Der Eingangsassistent besteht aus mehreren Elementen, welche auf einzelnen Hardware-systemen gehostet oder auf einem Gesamtsystem genutzt werden können. Hierbei kann die Unterteilung in folgende Elemente erfolgen. Der Kinectsensor und die Gastnutzer GUI stellen ein Element des Eingangsassistenten dar. Als weiteres Element steuert der Webserver die Kommunikation zwischen Maschine und Gastnutzer und kann von dem Systemnutzer mithilfe einer Web GUI verwaltet werden. Als letztes Element wird eine Datenbank zur Speicherung der gewonnen Daten bereitgestellt. Alle Elemente kommunizieren über einen Netzmerkmanager miteinander. Der Gastnutzer hat die Möglichkeit per Gesten und Spracherkennung mit dem Eingangsassistenten zu kommunizieren. Diese Kommunikation ist in den Webserver integriert. Der Systemnutzer kann die Reaktionen des Assistenten sowie die gesammelten Informationen über die Web GUI einsehen. Die Softwareimplementation erfolgt über Pakete um Wiederverwendbarkeit zu ermöglichen, die unter Punkt 4 besprochen werden.

## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

### 4 Struktur- und Entwurfsprinzipien einzelner Pakete

#### 4.1 Gestenerkennung und Datenbank

Die Gestenerkennung ist über C# und das Kinect SDK 2.0 implementiert. Um Gesten nutzen zu können muss eine Gestendatenbank importiert werden, welche die gewünschten Gesten enthält. Diese Datenbank kann über das Kinect Tool „Visual Gesture Builder“, welches Bestandteil der Kinect SDK 2.0 ist, erstellt werden. Hierzu nimmt man verschiedene Videos mit der Geste auf und markiert der Maschine wann die Geste stattfindet. Man sollte dann mehrere Tests erstellen um zu prüfen ob die Geste richtig erkannt wird.

#### 4.2 Web GUI und Registry Datenbank

Die Web GUI wurde mit Hilfe der Bootstrap Engine unter Anwendung des AdminLTE Designs erstellt. Diese dient dem externem Zugriff auf die Daten des Systems und unter Rücksichtnahme der richtigen Privilegien und Rechte auch dem Steuern einzelner Komponenten. Diese Rechte werden von Administratoren vergeben und werden in einer lokalen MySQL Datenbank verwaltet. Diese Datenbank wird über phpMyAdmin unter den entsprechenden Benutzerkonten geführt. Die Anmeldung erfolgt ebenfalls über die Web GUI in den entsprechenden Formularen.

#### 4.3 Server-Client C#

Die Kommunikation zwischen den Java Komponenten und den C# Komponenten erfolgt über eine einfache Server-Client Kommunikation. Java stellt den Server zur Verfügung der nur auf Daten wartet, die vom C# Client gesendet werden. Die Daten bestehen zum einen aus Kommandos die eine Reaktion des Servers auslösen und zum anderen über Informationen die von dem Kinectsensoren gesammelt wurden.

#### 4.4 C# Kinect Anbindung und GUI

Die initiale Datenverarbeitung der Kinect v2 geschieht über die C# Kinect Bibliotheken und die selbsterstellte C# GUI. Die Daten werden als Bitstream von der Kinect v2 aus über eine USB 3.0 Schnittstelle bereitgestellt. Um diese Verbindung herzustellen, werden

## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

über die Klasse `harambeGUI` die Kinect v2 Sensoren aktiviert, der Bitstream verarbeitet, Frames über eine Bitmaske von dem Bitstream erstellt und verarbeitet. Die Aktivierung der Sensoren geschieht in der Methode `getResources`. Hier wird zudem auch der `MultiSourceFrameReader` initialisiert, welcher für die Übertragung des Bitstreams von Nöten ist. In dieser Methode werden auch neue Methoden für die Erstellung, Analyse und Darstellung der Framedaten gecalled. Die `BodyFrameReader_FrameArrived` Methode handelt dabei die Übertragung der Körpermasken auf den Frame und die Auswertung der erkannten Körperdaten. Dabei wird ein Skelett der Kinect Body Bibliothek auf den Frame gemapped, wenn dieser einen Körper mit mindestens dem Oberkörper eines Menschen enthält. Alle initialisierten Ressourcen werden abschließend über die `disposeResources` Methode wieder geschlossen und die Sensoren der Kinect v2 deaktiviert.

### 4.5 Speech-to-text Modul

Externe Produkte verwendet:

CMU Sphinx4 – Spracherkennungsbibliothek von Carnegie Mellon University German  
Voxforge acoustic model – akustisches Modell für CMU pocketsphinx von Guenter Bartsch, veröffentlicht unter GNU GPL.

Dieser Modul gibt die Möglichkeit Sprachbefehle für die Programmsteuerung zu benutzen. Modul bietet eine einfache API um die Integration in andere Teile des Programms zu erleichtern.

Speech-to-text Modul ist in Java geschrieben und besteht aus folgenden Klassen:

- `Recognizer` – bietet API für die Erkennung von aufgenommenen WAV-Dateien
- `LiveRecognizer` – bietet API für die Erkennung der Sprache direkt aus dem Mikrophon-Signal
- `AudioRecorder` – wird intern benutzt für die Aufnahme von WAV-Dateien

Im Modul ist die Spracherkennung für Deutsch und Englisch implementiert. Es ist auch die Möglichkeit vorgesehen unterschiedliche, unter anderem für bestimmte Projektteile maßgeschneiderte Wörterbücher, zu benutzen. Im Allgemeinen, um die Fehlerrate der Erkennung so niedrig wie möglich zu halten, wird es empfohlen in Programmkomponenten nur die verkürzten Wörterbücher zu benutzen.

## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

Bei der Erstellung vom Modul wurde Source Code von mehreren Klassen aus CMU Sphinx4 verändert, nämlich `edu.cmu.sphinx.frontend.endpoint.SpeechMarker` – zusätzlicher Flag übergeben (als statische Eigenschaft `nonTerminationFlag`, die dann in der Methode `getData()` weiter benutzt wird) um zu vermeiden, dass man das Programm nicht schließen kann weil sie auf Input wartet `edu.cmu.sphinx.api.Microphone` – in die Methode `stopRecording()` wurde Zeile `line.close()` hinzugefügt um Mikrofon Ressourcen freizusetzen, was im weiteren die Instanziierung von `LiveRecognizer` ermöglicht `edu.cmu.sphinx.recognizer.Recognizer` bei der Methode `deallocate()` wurde noch zusätzliche Überprüfung des Zustandes eingefügt um zu vermeiden, dass Sphinx4 geschlossen wird bevor der Spracherkennungsalgorithmus abgearbeitet wurde `edu.cmu.sphinx.frontend.feature.DeltasFeatureExtractor` – zusätzliche Fehlerbehandlung in die Methode `computeNextFeature()` hinzugefügt um die Änderungen in die Klasse `SpeechMarker` zu ermöglichen Folgende Änderungen waren notwendig, weil Endbenutzer API von CMU Sphinx4 manche Features, die für unser Projekt notwendig sind nicht aufweist, unter anderem gibt es mehrere Probleme beim schließen von Sphinx4.

### 4.6 Text-to-speech Modul

Externe Produkte verwendet:

MaryTTS – Open-Source Software für Sprachsynthese erstellt von DFKI GmbH.

Dieser Modul gibt die Möglichkeit zusammen mit dem Speech-to-text Modul Kommunikation mit dem Endbenutzer des Systems zu ermöglichen.

Speech-to-text Modul ist in Java geschrieben und besteht aus folgenden Klassen: Synthesizer – bietet API um die Anfragen an den MaryTTS HTTP Server zu machen, Ergebnis über das Netzwerk erhalten und wiedergeben.

Sprachsynthese ist möglich sowohl in Deutsch, als auch in Englisch. Für die Benutzung des Moduls soll ein MaryTTS HTTP Server gestartet werden.

### 4.7 Text-to-Visualization

Um die Sprachausgabe anschaulich darzustellen, wurden ein pixelart Bilder erstellt. Dabei handelt es sich bisher um 2 Bilder, welche aufeinander abgestimmt wurden. Auf diesen befindet sich der Kopf eines Gorillas, welcher auf dem einen Bild den Mund offen, auf

## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

dem Anderen diesen geschlossen, hat. Für die Sprachvisualisierung wurden diese Bilder hintereinander geschaltet, sodass die Illusion des Sprechens angedeutet wird. Im nächsten Schritt wird von der Text to Speech Klasse eine Benachrichtigung geschickt, welche benachrichtigt, wann angefangen und wann aufgehört wird zu sprechen.

### 4.8 Guidepost Modul

Die Guidepost Klasse dient primär als Wegweiser für den Nutzer. Die Hauptfunktionalität besteht darin, dass der Nutzer aus einer Liste verfügbarer Orte auswählen kann, um sich deren Adresse anzeigen zu lassen, und so beispielsweise eine öffentliche Toilette, den nächstgelegenen Arzt oder eine Touristenattraktion zu finden. Im Laufe der Zeit wurden auch Informationen zu Öffnungszeiten einiger Institutionen sowie eine Wettervorhersage eingebaut, da diese leicht in die schon bestehende Programmstruktur zu integrieren waren.

Das Herzstück des Programms sind eben diese abrufbaren Informationen. Sie werden in mehrere Kategorien unterteilt, wobei jede Kategorie aus einer HashMap besteht, welche wiederum einen Integer als Key, und eine ArrayList als Wert annimmt. Der Integer Wert dient der Nummerierung der verfügbaren Elemente.

Jeder Eintrag der HashMap repräsentiert einen bestimmten Ort. Die ArrayList besteht jeweils aus zwei Strings, wobei der erste den Namen des Ortes angibt, der zweite - je nach Kategorie - die Adresse/Öffnungszeiten/Wettervorhersage für diesen Ort.

Um diese verschiedenen Kategorien in einer Datenstruktur zu vereinigen, haben wir eine weitere HashMap, die wir knowledgeBase nennen, erstellt. Diese besitzt als Key ebenfalls einen Integer, der diesmal die Kategorienummer angibt. Als Wert wird jeweils eine der HashMaps zugewiesen, die eine eigene Kategorie darstellen.

Weiterhin benötigen wir eine HashMap categoriesBase, die als Key die Kategorienummer als Integer besitzt, und als Wert den Kategorienamen als String. Wichtig ist, dass die Kategorienummer einer Kategorie für knowledgeBase und categoriesBase übereinstimmen muss! Die Entscheidung unsere Daten in einer HashMap zu speichern, soll vor allem die leichte Erweiterbarkeit um neue Kategorien ermöglichen. So können neue Informationen einfach durch das Hinzufügen weiterer Einträge in eine schon bestehende HashMap, oder durch das Hinzufügen einer HashMap 'neueKategorie' in die HashMap knowledgeBase,

## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

sowie dem Eintragen von Kategoriennummer- und -namen in `categoriesBase` realisiert werden.

Bei der Konsolenausgabe werden automatisch alle Kategorien aus `categoriesBase` aufgelistet. Für die ausgewählte Kategorie werden anschließend alle Elemente angezeigt, die in der `knowledgeBase` für die entsprechende Kategorie vorhanden sind. Der Nutzer kann dann entweder per Spracheingabe oder alternativ per Tastatur Kategorie und Element auswählen, und bekommt die ausgewählten Informationen als Konsolenausgabe angezeigt.

Für die Spracheingabe gibt es zwei verschiedene Einstellungen: Den Live-Modus und den WAV-File-Modus. Bei der Livemethode wird ein `LiveRecognizer` verwendet, der die Sprache in Echtzeit verarbeiten kann, u.U. aber eine etwas schlechtere Erkennung bereitstellt. Beim WAV-File-Modus wird ein `.wav` File mit einer im Code definierten Länge (z.B. 2000 ms) aufgezeichnet, und anschließend auf Sprache analysiert. Das kann eine leichte Zeitverzögerung mit sich bringen, auf schwachen Systemen aber die Erkennung verbessern.

Die Einstellung ob man Spracherkennung, und falls ja auch die Live-Erkennung nutzen möchte, wird beim Erstellen einer `Guidepost` Instanz durch die zwei booleans `voiceInputEnabled` bzw. `liveVoiceInputEnabled` festgelegt. Es kann pro Instanz nachträglich nicht mehr zwischen den Modi gewechselt werden, da ein `Recognizer` und ein `LiveRecognizer` nicht gleichzeitig laufen können, und es sonst zu Kollisionen käme.

Zur Sprachausgabe wird das `MaryTTS` Framework benutzt. Dazu wird ein lokaler Server gestartet, zu dem sich der `MaryTTS` Client verbinden kann. Der Port der Verbindung wird im `Configuration File` des Projekts bestimmt. Die `Guidepost` Klasse startet den Server bereits automatisch, sodass keine weiteren Aktionen nötig sind.

### 4.9 Wetter Klasse

Die `Wetter` Klasse ist ein Teil des `Guideposts` und dient dazu eine Übersicht der vorhergesagten Temperaturen für die nächsten drei Tage auszugeben. Dazu wurde eine `OpenWeatherMap` benutzt. Dafür wird die API für `OpenWeatherMap.org`, sowie die Einheit (unit für `kg, deg`), in der sich die Daten befinden sollen, übergeben. Die `OpenWeatherMap` wird für eine eingegebene Stadt erstellt.. Hierbei wird sich auf Deutsche Städte beschränkt. Über die einzelnen täglichen `Wettervorhersagen` für die kommenden 3 Tage wird



## 4 STRUKTUR- UND ENTWURFSPRINZIPIEN EINZELNER PAKETE

---

die Vorhersage erstellt. Dafür wird eine Forecast Instanz erstellt, für die die Temperatur abgerufen werden kann.

### 4.10 Reaktions Klasse

Die Reaction Klasse definiert die Text- bzw. Sprachreaktionen, welche durchgeführt werden, um einen Person zu bergüßen. Dazu wurde eine ArrayList angelegt, welche standardmäßige Reaktionen enthält. Diese können durch einen externen Aufruf verändert oder gelöscht werden. Außerdem können weitere Reaktionen über add-Funktionen hinzugefügt werden. Um die Begrüßung zu variieren wird bei Aufruf dieser eine zufällige Reaktion aus der vorhandenen ArrayList ausgewählt und ausgegeben. Für die Ausgabe wird ein String zurück gegeben, welcher dann bearbeitet werden kann. Die Ausgabe wird dann entweder von einer Konsolenausgabe oder aber von einer Speechnausgabe übernommen.

### 4.11 Email Benachrichtigung

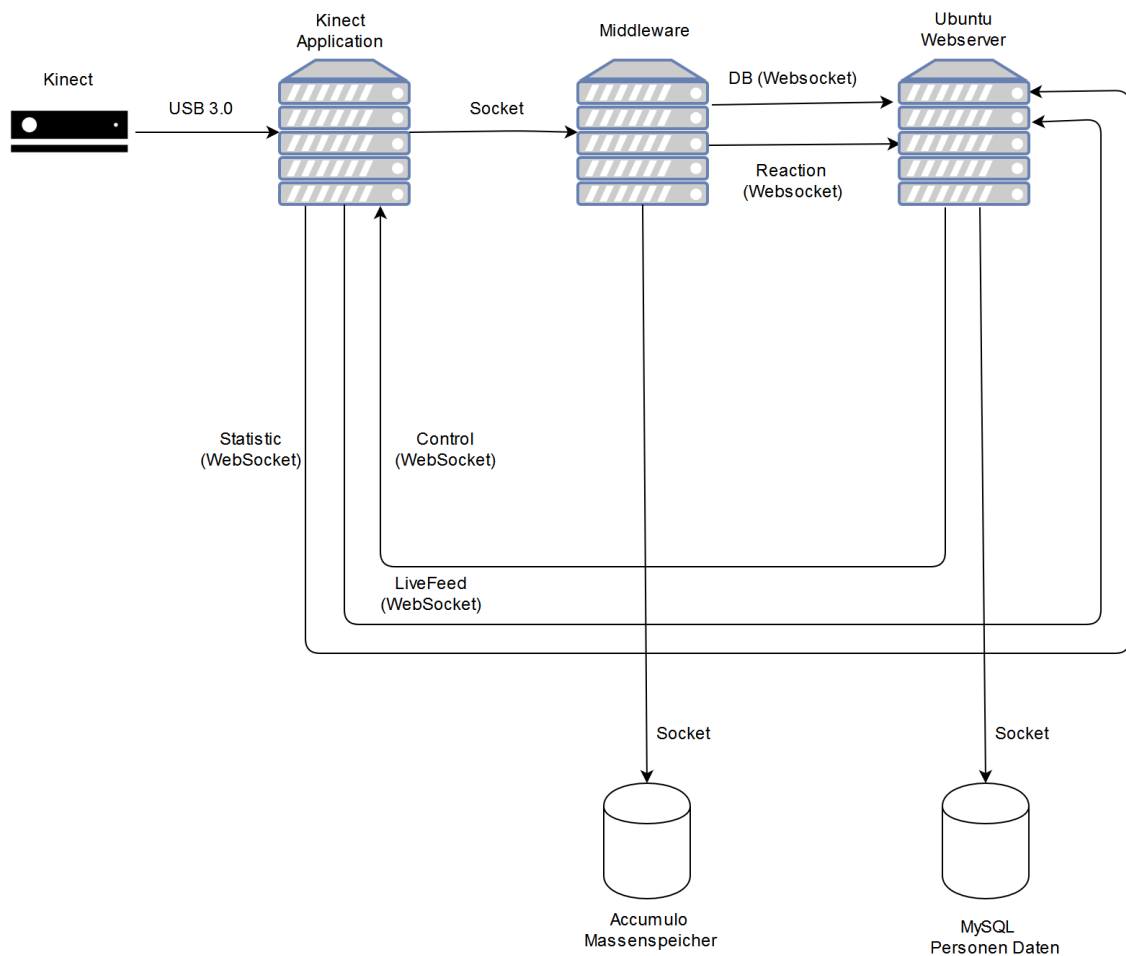
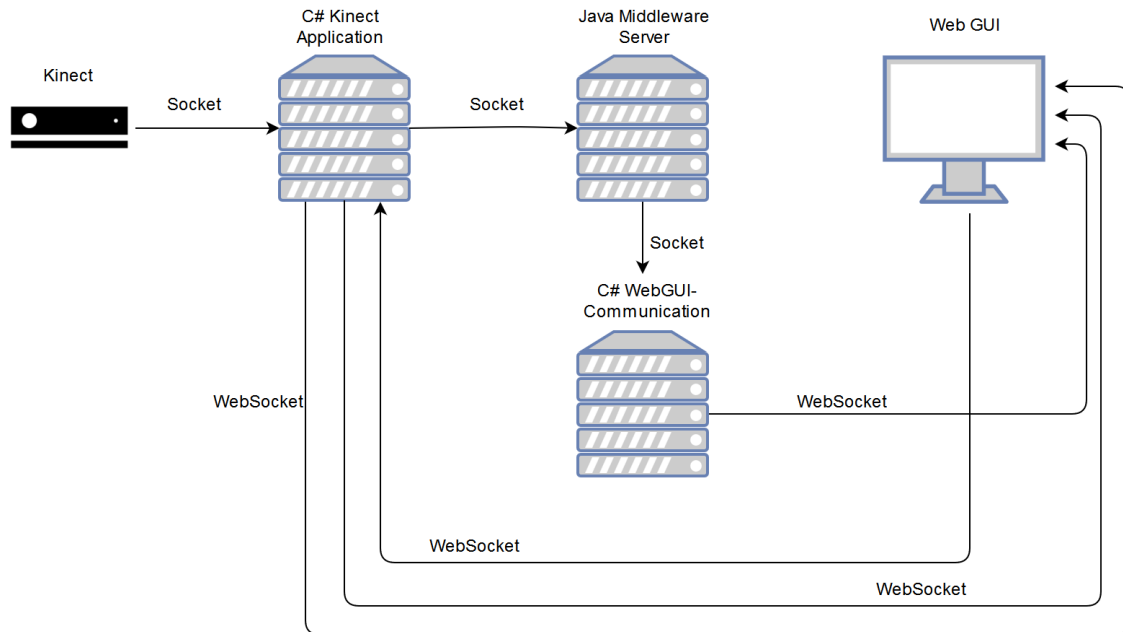
Die Klasse Send\_Mail dient dem Benachrichtigen der jeweiligen Administratoren oder Service User über das jetzige Geschehen. Es werden aktuelle Daten versandt mit angehängten Frames des eingestellten Cases. Diese Cases können variieren und die gelieferten Daten somit auch. Das versenden der Email ist automatisiert und wird zusätzlich noch an den Service Mail Account des Projekts verschickt, als Kopie und Geschehenslog. Der Code wurde auf C# geschrieben, da die Umsetzung in Java ohne die Enterprise Edition zu aufwändig wäre.

### 4.12 Datenbank

Durch die Kinect werden Bild- und Videoaufnahmen angefertigt, welche gespeichert werden sollen. Durch den Umfang der Daten wird eine Datenbank benötigt, welche große Datenmengen verarbeiten, speichern und schnell abrufen kann. Für diesen Zweck wird Accumulo verwendet. Weiterhin werden einfache Tabellen auch in Accumulo abgelegt, damit nicht mehrere Datenbanken nebeneinander gepflegt werden müssen.

5 DATENMODELL

5 Datenmodell



## 5 DATENMODELL

Tabelle 1: Implementierter Zustand

Row ID	Column			Value
	Family	Qualifier	Visibility	
[timestamp Aufnahme]	picture			[picture value]

Tabelle 2: Zielmodell, Implentierungsvariante 1

Row ID	Column			Value
	Family	Qualifier	Visibility	
[timestamp Aufnahme]	picture			[picture value]
[timestamp Aufnahme]	meta	brille		[0 1]
[timestamp Aufnahme]	meta	gesicht		[0 1]

**Daten**

Da accumulo eine key/value Datenbank ist haben wir folgende Ansätze um eine Tabelle mit den von uns genutzten Daten aufzubauen.

\*Bitfeld besteht zu Beginn aus 2 Bits die folgende Bedeutung haben und beliebig in ein Richtung erweitert werden können.

Bedeutung:

01 – Gesicht

10 – Brille

Kombinationen sind zugelassen die Reihenfolge muss nur beim schreiben in die Datenbank und beim lesen beachtet werden.

In Abbildung 1 werden für alle Metadaten eigene rows angelegt, was zu einer hohen Splitting der Daten führt und man viele Zugriffe benötigt um die gewünschten Informationen zu erhalten.

Abbildung 2 hingegen verschlüsselt Metadaten, welche mit booleschen Werten dargestellt

Tabelle 3: Zielmodell, Implentierungsvariante 2

Row ID	Column			Value
	Family	Qualifier	Visibility	
[timestamp Aufnahme]	picture			[picture value]
[timestamp Aufnahme]	meta			[Bitfeld]

## 5 DATENMODELL

---

werden können in einem Bitfeld. So ist nur ein Zugriff erforderlich um alle Daten zu erhalten. Hierbei muss man jedoch genau darauf achten, dass das Bitfeld immer gleich aufgebaut und gelesen wird, da es sonst zu korrupten Informationen kommt.

In unserer Anwendung nutzen wir Abbildung 2, da diese für unsere Zwecke ausreichend ist.

## 6 GLOSSAR

---

### 6 Glossar

Kinect for Windows SDK	Das <i>Kinect for Windows SDK</i> enthält alles, was zum Entwickeln von Kinect-Anwendungen benötigt wird. Dank bestimmter Windows-Treiber kann der Motion-Sensor der Xbox 360 auch am PC genutzt werden. Spiele und Programme, die eine Bewegungs- oder Spracherkennung per „Kinect“ unterstützen, lassen sich durch Gesten oder gesprochene Anweisungen bedienen.
MaryTTS	Es handelt sich bei <i>MaryTTS</i> um ein Open Source TTS (Text-to-Speech) Programm, welches in Java geschrieben wurde. Es unterstützt verschiedene Sprachen, sowie Sprachstile.
FreeTTS	Es handelt sich bei <i>FreeTTS</i> um ein Open Source TTS (Text-to-Speech) Programm, welches in Java geschrieben wurde. Es unterstützt verschiedene Sprachstile. Die deutsche Sprache wird nicht unterstützt.
Xbox 360	Die Xbox 360 ist eine stationäre Spielekonsole der 7. Konsolengeneration, welche im Jahr 2005 von Microsoft erstmals verkauft wurde. Sie ist der Nachfolger der Xbox (2002) und der Vorgänger der Xbox one (2013).
Kinect	Bei einer Kinect handelt es sich um eine Hardware zur Steuerung der Xbox 360. Diese wird über Körperbewegungen, sowie Sprache gesteuert. Die Eingabe wird dabei von der Kinect über Prime Sense-Tiefensensoren, 3D-Mikrofone und Farbkameras erkannt und von der entsprechenden Software verarbeitet.
Xbox 360 Development Kit	Ein Xbox 360 Development Kit (kurz: DevKit) ist ein Entwicklungswerkzeug der Spieleentwicklung. Es wird außerdem zum Testen, sowie Debuggen benutzt.

## 6 GLOSSAR

---

Big Data	Big Data wird als Sammelbegriff für digitale Technologien verwendet, bei denen große digitale Datenmengen, sowie deren Analyse, Sammlung, Verwendung und Vermarktung im Mittelpunkt stehen. Dabei werden 5 Big Data Challenges gestellt, welche sich in 2 Teilbereiche aufteilen. Es handelt sich um volume (bis zu viele Peta- und Exabyte an Speicher), velocity (sofortige Analyse dynamischer Datenströme) sowie variety (hohe Qualität und Glaubwürdigkeit der Daten), welche unter den Begriff 'Big' fallen, sowie value (Gewinnung nützlicher Informationen) und variety (heterogene strukturierte, teil- und unstrukturierte Daten), welche für die Sicherstellung der Daten verantwortlich sind.
Webserver	Ein Server ist ein zentraler Computer, mit dem sich andere Computer verbinden können, um bereitgestellte Dienste/Daten zu erhalten. Bei Webservern handelt es sich um Server, die entweder für die Öffentlichkeit bestimmt sind und als WWW-Dienst eingesetzt werden oder sie werden lokal im internen Netz eingesetzt.
Server-Client Struktur	Die Server-Client Struktur ist eine Möglichkeit der Aufgabenverteilung im Netzwerk. Dabei stellt der Server Daten/Dienste bereit und wartet auf Anfragen, um diese zu verschicken/leisten. Der Client hingegen verbindet sich mit dem Server, um Dienste anzufragen.
Front-End	Front-End ist ein Begriff der Schichteneinteilung aus dem Bereich der Informationstechnik. Es handelt sich dabei um den Teil der Software (hat jede Software), die von dem Nutzer gesehen wird und mit der dieser agieren kann, wie zum Beispiel die Nutzeroberfläche einer Webseite.

## 6 GLOSSAR

---

Back-End	Back-End ist ein Begriff der Schichteneinteilung aus dem Bereich der Informationstechnik. Es handelt sich dabei um den Teil der Software, in dem die Funktionen dieser festgelegt werden. Das Front-End ist abhängig von diesem Bereich, da die Definition dieses in jenem Bereich erfolgt. Dabei ist das Back-End für den Nutzer nicht sichtbar.
Softwarelizenz	Programme, welche im öffentlichen Rahmen ausgeführt werden, benötigen eine Nutzungserlaubnis. Diese Nutzungserlaubnisse werden urheberrechtliche Softwarelizenzen genannt.
J4K	<b>Java for Kinect</b> ist eine OpenSource Java Bibliothek, welche die Programmierung der Microsoft Kinect zu ermöglichen. Sie ist kompatibel mit allen Kinectgeräten. Dafür wird eine Verbindung zu den Sensoren der Kinect aufgebaut und die Signale ausgewertet.
JNI	Java ist keine plattformunabhängige Programmiersprache. Um plattform-spezifische Funktionen aufzurufen, wird die standardisierte Anwendungsprogrammierschnittstelle <b>Java Native Interface</b> benutzt.

## 6 GLOSSAR

---

### Java Maven

Maven ist ein Tool, um die Wiederverwendung von Java-Bausteinen zu fördern und die Entwicklung zu vereinfachen. Es steht dabei in direkter Konkurrenz zu Ant, hat jedoch weitaus mehr Funktionen, als das einfache Build-Tool. Es dient zum einen dazu einen Build-Prozess zu erstellen, der die Erstellung der Executables automatisiert und vor allem bei größeren Projekten vereinfacht. Maven bietet dabei die Möglichkeit einer einfachen Einbindung von eigenen oder fremden Bibliotheken. Build-Tools im allgemeinen, jedoch Maven im speziellen können folgende Aufgaben erfüllen:

- Downloaden von Abhängigkeiten (dependencies)
- Compilen des Sourcecodes zu Binaries
- Packaging der Binaries
- Tests ausführen
- Tests automatisieren
- Deployen der Binaries als produktives/lauffähiges Gesamtsystem

### TDD

**Test Driven Development** ist eine Methode der Programmierung, welche bei der Entwicklung von Software eingesetzt wird. Dabei werden die Tests für die Software vor der eigentlichen Software geschrieben. Dies hat den Vorteil, dass es nicht, wie bei anderen Methoden, zu mangelnder Testbarkeit kommt oder das Programm aus Zeitgründen nicht ausführlich getestet werden kann, sowie kommt es nicht so leicht zu Redundanzen.



## 6 GLOSSAR

---

Java JUnit	Um die Funktionalität einer Software oder einem Teil davon sicherzustellen, bieten sich unit tests an. Dabei handelt es sich um kleine Programme, welche ohne Benutzerkontrolle den Code automatisch auf bestimmte, vorher definierte Regeln zu testen. JUnit stellt dabei ein solches Testframework dar.
Bootstrap - adminLTE	AdminLTE basiert auf Bootstrap, wobei Bootstrap ein Open Source Framework ist, welches auf allen mobilen Geräten läuft und zugleich sehr anpassungsfähig ist. Dabei liefert Bootstrap eine umfangreiche Dokumentation für HTML, sowie CSS, sowie JavaScript Erweiterungen.