Fakultät für Informatik

Telefon: +49 821 5586-3450

Fax: +49 821 5586-3499

Verfasser der Diplomarbeit:

Andreas Hofmann

Lilienweg 1

86559 Adelzhausen

Telefon: +49 178 7131560

andreas.hofmann@live.de

|  |
| --- |
| Bachelorarbeit |
|  |
| Studienrichtung  Informatik    **Andreas Hofmann**  **Schaffung einer fusionierten Datenbasis zur Immobilienbewertung in Microsoft Azure**    Prüfer: Dipl.-Inf. (FH), Dipl.-De Erich Seifert  Zweitprüfer: Prof. Dr. Rolf Winter  Abgabe der Arbeit am: 20.08.2021  In Kooperation mit der digatus software GmbH |

Fakultät für Informatik

**Inhaltsverzeichnis**

[1 Einleitung 2](#_Toc76919022)

[1.1 2](#_Toc76919023)

[1.2 Stand der Wissenschaft 2](#_Toc76919024)

[1.2.1 Cloud Computing 2](#_Toc76919025)

[1.2.2 Cloud Computing Modelle 2](#_Toc76919026)

[1.2.3 Servicemodelle 2](#_Toc76919027)

[1.3 Motivation 2](#_Toc76919028)

[1.4 Ausgangslage 2](#_Toc76919029)

[1.5 Zielsetzung 3](#_Toc76919030)

[1.6 Unternehmen digatus it group AG 3](#_Toc76919031)

[1.6.1 Unternehmensdaten 3](#_Toc76919032)

[1.6.2 It-consulting GmbH 4](#_Toc76919033)

[1.6.3 software GmbH 4](#_Toc76919034)

[1.6.4 it-services GmbH 4](#_Toc76919035)

[1.6.5 digital GmbH 4](#_Toc76919036)

[2 Grundlagen 5](#_Toc76919037)

[2.1 Definitionen 5](#_Toc76919038)

[2.1.1 Authentisierung, Authentifizierung und Autorisation 5](#_Toc76919039)

[2.2 Umsetzungskonzept 6](#_Toc76919040)

[2.3 Verwendete Methoden 6](#_Toc76919041)

[2.3.1 SCRUM 6](#_Toc76919042)

[3 Entwicklung 7](#_Toc76919043)

[3.1 Tools 7](#_Toc76919044)

[3.1.1 Visual Studio 7](#_Toc76919045)

[3.2 DataGrip 7](#_Toc76919046)

[3.2.1 Docker 8](#_Toc76919047)

[3.3 Kubernetes 8](#_Toc76919048)

[3.4 Azure Cosmos Emulator 8](#_Toc76919049)

[3.4.1 Postman 9](#_Toc76919050)

[3.4.2 9](#_Toc76919051)

[3.5 Entwicklung des Webservices 9](#_Toc76919052)

[3.5.1 Anforderungen 9](#_Toc76919053)

[3.5.2 Implementierung 9](#_Toc76919054)

[3.6 Datenbanken 12](#_Toc76919055)

[3.7 Synchronisation 12](#_Toc76919056)

[3.7.1 Anforderungen 12](#_Toc76919057)

[4 Diskussion 12](#_Toc76919058)

[5 Zusammenfassung 12](#_Toc76919059)

[6 Anhang 12](#_Toc76919060)

[7 Abkürzungsverzeichnis 13](#_Toc76919061)

[8 Literaturverzeichnis 14](#_Toc76919062)

[9 Abbildungsverzeichnis 15](#_Toc76919063)

**Abstract**

## Einleitung

* 2. Stand der Wissenschaft

In diesem Kapitel wird auf den Stand der Wissenschaft oder Technik eingegangen, und eine Übersicht der verwendeten Techniken geboten.

* + 1. Cloud Computing

Cloud Computing (deutsch Rechnerwolke oder Datenwolke ...) beschreibt ein Modell, das bei Bedarf – meist über das Internet und geräteunabhängig – zeitnah und mit wenig Aufwand geteilte Computerressourcen als Dienstleistung, etwa in Form von Servern, Datenspeicher oder Applikationen, bereitstellt und nach Nutzung abrechnet.[1]

* + 1. Cloud Computing Modelle
    2. Servicemodelle
  1. Motivation
  2. Ausgangslage

Diese Bachelorarbeit ist in Kooperation mit der digatus software GmbH, in der ich als Werksstudent tätig bin entstanden. Über das Unternehmen wird im Kapitel 1.6 genauer eingegangen.

* 1. Zielsetzung

Die Ziele dieser Arbeit umfassen die Planung und Implementierung einer Applikation, die Daten aus dem Bereich Immobilienbewertung von verschiedenen Quellen zusammenführt und verwerten kann. Eine besondere Vorgabe an die Applikation ist die Fähigkeit im späteren Verlauf Daten zu importieren, für die es zum Zeitpunkt der Erstellung noch keine Verwendung gibt.

* 1. Unternehmen digatus it group AG

Digatus beschreibt sich auf seiner Website wie folgt:

digatus ist eine IT-Beratungs- und Dienstleistungsgruppe mit Sitz in München. Von verschiedenen Standorten aus arbeiten wir täglich für den Erfolg unserer Kunden – von vor der eigenen Haustür bis in ganz Europa verteilt. Als schnellwachsendes Unternehmen fördern wir Verantwortung und haben unsere Kompetenzfelder in die folgenden Einheiten aufgestellt. [2]

* + 1. Unternehmensdaten

Das Unternehmen wurde im Jahre 2015 Stephan Bals und Felix Kirschner in Augsburg gegründet. Inzwischen sind im Unternehmen über 100 Mitarbeiter angestellt. Außerdem wurden Firmenstandorte in München, Augsburg, Nürnberg, Frankfurt und Kapstadt eröffnet.

* + 1. It-consulting GmbH
    2. software GmbH
    3. it-services GmbH
    4. digital GmbH

## Grundlagen

* 1. Definitionen
     1. Authentisierung, Authentifizierung und Autorisation

Lorem ipsum

**Authentisierung und Authentifizierung**

Will ein Benutzer ein Anwendungssystem benutzen, dann muss er sich authentisieren. Dies geschieht in der Regel durch die Angabe eines Benutzernamens (user id) und eines Passwortes. Das Anwendungssystem führt nach der Authentisierung durch den Benutzer eine Authentifizierung durch, indem es prüft, ob der Benutzername und das Passwort mit entsprechenden gespeicherten Daten übereinstimmen. [3]

Bei der Authentifizierung weist ein Nutzer nach, dass er die Person ist, die er behauptet zu sein. Um die Identität nachzuweisen, gibt es mehrere Methoden, die verwendet werden können.

**Wissen**

Nutzer hat Kenntnis über Benutzerkennung und oder Passwort.

**Besitz**

Die Person besitzt einen Schlüssel, mit dem er sich Zugriff verschaffen kann. Dabei kann es sich um einen klassischen Schlüssel für ein Türschloss handeln, oder aber auch um einen Firmenausweiß oder ein One-Time-Passwort (OTP). Ein OPT wird zum Beispiel über einen Hardware-Token generiert. Dies können kleine Geräte sein, die nach Bedarf ein OPT generieren, oder eine Smartcard. Alternativ gibt es auch Softwarebasierte Token, die z.B. auf einem Smartphone installiert werden können das OTP algorithmisch erzeugen.

**Biometrische Verfahren**

Der Benutzer weist seine Identität durch seine eigene Anwesenheit nach. Hierfür können zum Beispiel Fingerabdrücke oder Gesichtserkennung dienen.

**Multi-Faktor-Authentifikation**

TODO: lorem ipsum

**Autorisation**

In der Regel erhalten Benutzer, die authentifiziert wurden, nicht alle die gleichen Zugriffsrechte innerhalb eines Anwendungssystems. Um nicht für jeden einzelnen Benutzer individuelle Zugriffsrechte in einem Anwendungssystem festlegen zu müssen, werden Benutzern Rollen zugeordnet. Eine Rolle fasst dabei eine Reihe von Zugriffsrechten zu einer Einheit zusammen. Benutzer, die einer bestimmten Rolle zugeordnet sind, sind dann autorisiert, entsprechende Zugriffe durchzuführen. [3]

* 1. Umsetzungskonzept
  2. Verwendete Methoden
     1. SCRUM

## Entwicklung

Der gesamte Entwicklungsprozess der verschiedenen Softwareteile wurde in agilen dreiwöchentlichen Sprints mithilfe von SCRUM realisiert. Das Projekt gliedert sich in drei Elemente. Ein Webservice der Daten aus einer relationalen Datenbank bereitstellt. Eine Dokumentbasierte Datenbank, um die Ursprungsdaten zu speichern. Und anschließend die Synchronisationsapplikation, welche die Ursprungsdaten selektiert, validiert und in die relationale Datenbank überträgt.

* 1. Tools

In diesem Abschnitt gehe ich kurz auf die verwendeten Werkzeuge ein, die mir die Entwicklung dieses Projektes ermöglicht haben.

* + 1. Visual Studio

Visual Studio ist eine integrierte Entwicklungsumgebung (IDE), die von Microsoft entwickelt wurde. Die IDE kann zur Entwicklung von Programmen mit unterschiedlichen Hochsprachen eingesetzt werden. Typisch in Verbindung mit dem .NET Framework eingesetzt, können aber auch native Programme mit Sprachen wie C, C++, JavaScript, Python und weitere entwickelt werden. Mit Visual Studio ist es möglich Desktop, Web, mobile Anwendungen und mehr zu erstellen.

* 1. DataGrip

Ebenfalls eine DIE, allerdings für die Entwicklung von Datenbanken vorgesehen. Dieses Programm ermöglicht die Anbindung verschiedener Datenbanken und ist dabei nicht auf Structured Query Language (SQL) beschränkt. Neben den klassischen SQL-Datenbanken wie Oracle, Microsoft SQL-Server und MySQL werden auch NoSQL Datenbanken wie z.B. MongoDB unterstützt. DataGrip beinhaltet neben der Abfrage Konsole viele Werkzeuge den Entwicklern die Arbeit mit Datenbanken erleichtern.

* + 1. Docker

Diese Software ist eine Containerverwaltung und dient zur Bereitstellung von Anwendungen. In Docker können Programme zusammen mit all den Bibliotheken und Abhängigkeiten, die es benötigt, in ein Image (Abbild) gepackt werden. Dieses Abbild oder auch Container genannt, ist anschließend auf Systemen auf dem die Docker Engine installiert ist, lauffähig.

Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly despite differences for instance between development and staging. [4]

* 1. Kubernetes

Kubernetes (K8s) ist ein Open-Source-System zur Automatisierung der Bereitstellung, Skalierung und Verwaltung von containerisierten Anwendungen. [5]

Mit dieser Orchestrierungssoftware können Docker Container automatisch je nach aktuellem Bedarf bereitgestellt werden. Hierfür werden Container in einen sogenannten Pod platziert. Für K8s wird ein Master und ein oder mehrere Node-Server benötigt. Die Nodes fungieren dabei als Worker-Server und führen die Pods aus die ihm vom Masterserver zugeteilt werden.

Um die Last einer Applikation zu verteilen kann ein Pod dynamisch und automatisch auf mehrere Nodes ausgeführt werden.

The HorizontalPodAutoscaler resource (commonly called HPA) can be used to scale Deployments automatically depending on the CPU usage of the current replicas. [6]

Hiermit ist es ebenfalls möglich Ausfallsicherheit zu gewährleisten. Falls ein Node ausfallen sollte, wird automatisch ein Pod auf einem anderen bereitgestellt und gestartet. Diese Art des Hostings ist ideal, um Microservices zu betreiben.

* 1. Azure Cosmos Emulator
     1. Postman
  2. Entwicklung des Webservices

Der Webservice stellt den Kern des Projektes dar, und soll die erfassten Daten effizient zur Verfügung stellen.

* + 1. Anforderungen

Lorem ipsum

1. Abfragen sollen mit möglichst geringer Latenz ausgeführt werden.
2. Das Ergebnis der Abfrage soll durch Filter begrenzt werden, um die anzufragenden Daten einzuschränken.
3. Der Service muss eine Authentifizierung für den jeweiligen Nutzer auf Basis eines Microsoft Kontos ermöglichen. Hiermit sollen die Daten vor unberechtigtem Zugriff geschützt werden.
4. Adressen, die sowohl aus Fremdquellen wie auch von Nutzern stammen können, müssen validiert werden.
   * 1. Implementierung

Lorem ipsum

**Authentifizierung**

Um Daten aus der Applikation abfragen zu können, müssen Nutzer sich zuerst authentifizieren. Microsoft stellt für ASP.NET Core über das Nuget Packet-Manager-System die Microsoft Identity Web-Authentifizierungsbibliothek zur Verfügung. Mithilfe dieser Bibliothek, die als Authentifizierungsmiddleware dient, muss sich ein anfragender Nutzer zuerst bei Azure Active Directory (Azure AD) anmelden. Nach der erfolgreichen Authentifizierung erhält der Webservice ein Sicherheitstoken, das die Identität sicherstellt und die autorisierten Benutzerrollen enthält.

Um diese Methode der Authentifizierung nutzen zu können, sind seitens Microsoft Azure einige Vorbedingungen erforderlich. Im Azure Portal muss die Applikation als App registriert sein. Somit wird der App automatisch ein Universally Unique Identifier (UUID) zugeordnet, um Applikationen eindeutig identifizieren zu können. Um nach der Authentifizierung wieder auf den Webservice zugreifen zu können muss ein Redirect- Uniform Resource Identifier (URI) angegeben werden.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Abbildung 1: Einstellungen der Redirect-URI in portal.azure.com

Außerdem können in der Token Konfiguration zusätzliche Informationen vom Nutzer gefordert werden. Dies kann zum Beispiel die E-Mail-Adresse des Nutzers sein. Der Webservice erhält dann alle geforderten Tokens, sofern der Nutzer der Aufforderung seine Daten weiterzuleiten zustimmt.

In der Konfigurationsdatei der Applikation werden einzelne Adressen für CORS definiert, welche als einziges für den Zugriff akzeptiert werden sollen. Das Programm kann mit folgenden Zeilen Code geschützt werden:



Abbildung 2: Authentication Service in Startup.cs

Um den Zugriff weiter einzuschränken, müssen Benutzerrollen autorisiert werden. Auf dem Azure Portal können im Bereich API permissions rollen eingetragen werden, die jeweils einzelnen Nutzern oder Gruppen zugeordnet werden können. Ein Controller kann nun auf diese zugreifen, und entscheiden, ob ein Benutzer zugriffsberechtigt ist.



Abbildung 3: Autorisation in einem Controller

Das Authorize Attribut prüft vor jedem Zugriff auf einen Controller, ob der Zugriff authentifiziert wurde und gibt ansonsten einen 401 Unauthorized HTTP Statuscode zurück. Mit der Methode HttpContext.VerifyUserHasAnyAcceptedScope können übergebene Rollen geprüft werden. Sollte ein Nutzer hierauf zugreifen, ohne die benötigte Rolle zu besitzen, antwortet der Service mit einem 403 Forbidden Statuscode und weist darauf hin, dass die nötige Rolle nicht zugeordnet ist. In diesem Falle bricht der Programmcode an dieser Stelle ab.

Mit diesen Techniken ist der Webservice dem letzten Stand der Technik nach vor unbefugtem Zugriff geschützt.

TODO: SSO

* 1. Datenbanken
  2. Synchronisation
     1. Anforderungen

1. Hausnummern müssen in Blöcke gruppiert werden, um die genaue Position eines Gebäudes zu verschleiern. Dies soll personenbezogene Daten schützen.

## Diskussion

## Zusammenfassung

## Anhang

## Abkürzungsverzeichnis

|  |  |
| --- | --- |
| **2FA** | Zwei-Faktor-Authentifizierung |
| **API** | Application Programming Interface |
| **APP** | Applikation |
| **Azure AD** | Azure Active Directory |
| **HTTP** | Hypertext Transfer Protocol |
| **IDE** | integrierte Entwicklungsumgebung |
| **MFA** | Multi-Faktor-Authentifizierung |
| **OTP** | One-Time-Passwort |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locator |
| **UUID** | Universally Unique Identifier |
| **K8s** | Kubernetes |
| **SQL** | Structured Query Language |

## Literaturverzeichnis

[1] „Cloud Computing“, *Wikipedia*. Juni 18, 2021. Zugegriffen: Mai 08, 2021. [Online]. Verfügbar unter: https://de.wikipedia.org/wiki/Cloud\_Computing

[2] „Über uns“, *digatus*. https://www.digatus.de/ueber-uns/ (zugegriffen Juni 16, 2021).

[3] H. Balzert und P. Liggesmeyer, *Lehrbuch der Softwaretechnik. 2: Entwurf, Implementierung, Installation und Betrieb / Helmut Balzert. Unter Mitw. von Peter Liggesmeyer*, 3. Auflage. Heidelberg: Spektrum, Akademischer Verlag, 2011.

[4] „What is a Container? | App Containerization | Docker“. https://www.docker.com/resources/what-container (zugegriffen Juni 11, 2021).

[5] „Produktionsreife Container-Orchestrierung“, *Kubernetes*. https://kubernetes.io/de/ (zugegriffen Juli 11, 2021).

[6] P. Martin, *Kubernetes: preparing for the CKA and CKAD certifications*. 2021. Zugegriffen: Juli 11, 2021. [Online]. Verfügbar unter: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=2694885

## Abbildungsverzeichnis

[Abbildung 1: Einstellungen der Redirect-URI in portal.azure.com 8](#_Toc76912243)

[Abbildung 2: Authentication Service in Startup.cs 9](#_Toc76912244)

[Abbildung 3: Autorisation in einem Controller 9](#_Toc76912245)