



Freie Universität Berlin  
Fachbereich Mathematik und Informatik  
Takustraße 9, 14195 Berlin

# Master thesis

## User position prediction in 6-DoF mixed reality applications

Oleksandra Baga

Freie Universität Berlin  
Matrikelnummer 5480722  
Master Computer Science

E-Mail: [oleksandra.baga@gmail.com](mailto:oleksandra.baga@gmail.com)

Prof. Dr. Christian Forler  
Fachbereich VI - Informatik und Medien  
Beuth Hochschule für Technik Berlin

Prof. Dr. René Görlich  
Fachbereich VI - Informatik und Medien  
Beuth Hochschule für Technik Berlin

# Contents

|   |            |
|---|------------|
| <b>Abbildungsverzeichnis</b>                        | <b>I</b>   |
| <b>Listings</b>                                     | <b>II</b>  |
| <b>Abkürzungsverzeichnis</b>                        | <b>III</b> |
| <b>1 Introduction</b>                               | <b>1</b>   |
| 1.1 Problem statement . . . . .                     | 1          |
| 1.2 Structure of the Thesis . . . . .               | 1          |
| 1.3 Motivation for the research . . . . .           | 1          |
| <b>2 Theoretische Grundlagen</b>                    | <b>2</b>   |
| 2.1 Raspberry Pi Board und Betriebssystem . . . . . | 2          |
| 2.2 Kontaktlose Smartcards MIFARE . . . . .         | 4          |
| 2.3 Sender-Empfänger-System mit RFID . . . . .      | 6          |
| 2.4 Django Framework . . . . .                      | 7          |
| 2.5 Grundlagen der Webanwendungen . . . . .         | 9          |
| 2.5.1 Client-Server Kommunikation . . . . .         | 10         |
| 2.5.2 Grundlagen der REST API . . . . .             | 12         |
| 2.5.3 Clientseitiges JavaScript . . . . .           | 13         |
| <b>3 Network Architecture</b>                       | <b>14</b>  |
| 3.1 Layers . . . . .                                | 14         |
| <b>4 Implementierung</b>                            | <b>15</b>  |
| <b>5 Analyse</b>                                    | <b>16</b>  |
| <b>6 Zusammenfassung</b>                            | <b>18</b>  |
| <b>Glossar</b>                                      | <b>I</b>   |
| <b>Literaturverzeichnis</b>                         | <b>III</b> |

# Abbildungsverzeichnis

# Listings

|                              |    |
|------------------------------|----|
| 4.1 StudentFactory . . . . . | 15 |
|------------------------------|----|

# Abkürzungsverzeichnis

|              |   |
|--------------|---|
| <b>AJAX</b>  | Asynchrones Javascript und XML                    |
| <b>API</b>   | Application Programming Interface                 |
| <b>ASGI</b>  | Asynchronous Server Gateway Interface             |
| <b>ATR</b>   | Answer-To-Reset                                   |
| <b>BGA</b>   | Ball Grid Array                                   |
| <b>CPU</b>   | Central processing unit                           |
| <b>DOM</b>   | Document Object Model                             |
| <b>GPU</b>   | Graphic processing unit                           |
| <b>HTTP</b>  | HyperText Transfer Protocol                       |
| <b>JSON</b>  | JavaScript Object Notation                        |
| <b>IDE</b>   | Integrated Development Environment                |
| <b>IEEE</b>  | Institute of Electrical and Electronics Engineers |
| <b>ISO</b>   | International Organization for Standardization    |
| <b>MAC</b>   | Media Accesss Control Layer                       |
| <b>NFC</b>   | Near Field Communication                          |
| <b>SDRAM</b> | Synchronous dynamic random-access memory          |

|              |  |
|--------------|--|
| <b>SoC</b>   | System-on-Chip   |
| <b>RAM</b>   | Random-access memory                                     |
| <b>Raspi</b> | Raspberry Pi Board und Minicomputer im PSE-Labor der BHS |
| <b>RFID</b>  | Radio-frequency identification                           |
| <b>URI</b>   | Uniform Resource Identifier, Unified Resource Identifier |
| <b>URN</b>   | Uniform Ressource Name                                   |

# Introduction

The topic of this Master's thesis is the process of predicting human head motion with a 6-dimensional degree of freedom (DoF) for Extended Reality (XR) applications.

## 1.1 Problem statement

A non-full-text database that typically contains article metadata, abstracts, and subject classifications. Used by researchers to locate publications relevant to their research.

## 1.2 Structure of the Thesis

The organization of this thesis is as follows. The literature review chapter introduces the concepts of XR technologies and principles of motion prediction algorithms. It follows an overview of previous research.

**Chapter 1** - Introduction.

The following chapter introduces the preprocessing pipeline, architecture, and evaluation process used to develop ahead motion prediction algorithm.

## 1.3 Motivation for the research

The following chapter introduces the preprocessing pipeline, architecture, and evaluation process used to develop ahead motion prediction algorithm.

# Theoretische Grundlagen

Als schon im Abschnitt "Einleitung" erwähnt wurde, bei der Aufgabestellung es um eine Entwicklung einer Webanwendung geht. Die zu realisierende Software basiert sich, wie die meisten Webanwendungen, auf einer Client-Server-Architektur, wobei der Client Informationen eingibt, während der Server die eingegebene Daten empfängt, bearbeitet und speichert.

## 2.1 Raspberry Pi Board und Betriebssystem

Raspberry Pi ist ein Einplatinencomputer, wobei die verschiedene Teile des Computers, die sich normalerweise auf separaten Platinen befinden, werden hier nur auf einer dargestellt. Wie die meisten Einplatinencomputer ist der Raspberry Pi so klein wie eine Kreditkarte. Der Raspberry Pi ist eine kostengünstige Plattform - sein empfohlener Verkaufspreis beträgt weniger als 50€. Ein Mikrocomputer verfügt über alle Funktionen eines Personal Computer (PC): Prozessor, Speicher, Betriebssystem, Anschluss an einen Monitor (TV), die Vernetzung. Der Raspberry Pi verfügt im Gegensatz zu einem PC über zusätzliche Peripheriegeräte wie GPIO-Ports (General Purpose Input / Output). Über diese Pins kann der Mikrocomputer mit der elektronischen Welt der Sensoren, Bildschirmen und Aktoren interagieren. In Abbildung ??[Pri] sind die Anschlüsse schematisch dargestellt.

Für die Entwicklung der Abschlussarbeit und die Verwendung im Labor wird Raspberry Pi 3 Model B+ benutzt, an dem der RFID Leser angeschlossen wird. Das Modell "Raspberry Pi 3 B" ist eine kontinuierliche Weiterentwicklung zum Vorgängermodell "Raspberry Pi 2 B". Der Raspberry Pi 3 enthält einen Quad-Core-Prozessor mit 1,2 GHz von Broadcom und einen SDRAM-Arbeitsspeicher mit 1 GByte. Auf der Unterseite des Raspberry Pi befindet sich einen weiteren Chip, der wie ein kleines schwarzes Plastikquadrat aussieht. Dies ist der Direktzugriffsspeicher (RAM) des Mikrocomputers. Wenn am Pi gearbeitet wird, wird die laufende Arbeit auf dem RAM gespeichert. Nur wenn die Daten explizit auf die microSD-Karte geschrieben werden, dann werden sie zum nächsten Verwendung gespeichert und nach dem Ausschalten nicht gelöscht. Der Raspberry Pi erlaubt den Austausch des aktuellen RAM-Chips nicht. Der aktuelle RAM-Chip ist als BGA-Gehäuse direkt auf die Oberseite der CPU gelötet. Das Hartlöten macht das Entfernen und / oder Ersetzen des



RAMs sehr schwierig und deshalb es ist beim Kauf des bestimmtes Board auf die Größe des RAMs zu achten.

Die Besonderheiten des Raspberry Pi 3 ist, dass WLAN nach IEEE 802.11 b/g/n und Bluetooth Low Energy onboard sind und nicht durch externe USB-Adapter nachgerüstet werden müssen. Es ist für die vorgesehene Aufgabe wichtig ist, da die erforderlichen Komponenten nicht zusätzlich gekauft werden muss, was das Budget der PSE-Labors erspart und eine gewisse Zeit nicht geplant muss, dass die zusätzlich gekauften Komponenten irgendwie zum funktionieren bringen. Hier gab es in der Vergangenheit immer wieder Schwierigkeiten mit der Hardware-Erkennung oder Treiber-Probleme.

Die ARMv8-Architektur enthält Cortex-A53-Rechenkerne, die bei gleichem Takt schneller sind als die alten Cortex-A7-Kerne des Raspberry Pi 2 B. Von der 64-Bit-Fähigkeiten der ARMv8-Architektur profitiert allerdings nur neue Software, da 64 Bit auf der Hardware-Seite allerdings vom Betriebssystem und der Software auch unterstützt werden muss. Die folgende Liste gibt es eine kurzen Ansicht auf die Architektur des Mikrocomputers.

- System-on-Chip: BCM2837 64 Bit ARMv8 von Broadcom
- Prozessor: Quad-Core-Prozessor mit 1,2 GHz
- GPU: Dual-Core-GPU VideoCore IV mit OpenGL ES 2.0 und OpenVG mit Hardwarebeschleunigung und 1080p30 H.264 High-Profile-Decoding
- Arbeitsspeicher: 1 GByte LPDDR2-SDRAM
- WLAN: BCM43143 onboard für IEEE 802.11b, g und n im 2,4 GHz-Bereich
- Bluetooth: Bluetooth Classic und Low Energy (BLE) onboard (Bluetooth 4.1)

Eines der beliebtesten Betriebssysteme für den Raspberry Pi ist das Raspbian-Betriebssystem. Das Raspbian-Betriebssystem basiert auf der ARM-Version von Debian 8 Jessie, ist für die Raspberry Pi-Hardware optimiert und enthält die Standardprogramme wie die LibreOffice Office Suite, einen Webbrowser, Claws Mail, eine leichtgewichtige Desktop-Umgebung und einige Programmier-Lernwerkzeuge. Neuere Versionen des Betriebssystems Raspbian verfügen über einen völlig modernen Chromium-Browser, mit dem auch komplexe Webseiten korrekt angezeigt werden können. Um eine Information auf einem großen Bildschirm anzuzeigen, wird Chromium nur im Vollbildmodus gestartet, den Mauszeiger ausgeblendet und den Bildschirmschoner ausgeschaltet. Es gibt verschiedene Möglichkeiten, Raspbian auf

einem Raspberry Pi 3 zu installieren. Die erste besteht darin, das Dienstprogramm NOOBS zu verwenden, die zweite darin, den Inhalt des Bildes des Betriebssystems direkt auf die Karte zu schreiben. Während der Entwicklung des Register-Clients wurde das Betriebssystem auf MicroSD-Karte geschrieben. Das Raspbian-Betriebssystem bootet von einer Micro-SD-Karte und das gesamte Betriebssystem läuft von der Karte.

Nach der kurzen Beschreibung der Raspberry Pi Architektur und sein Betriebssystem Raspbian ist schlusszufolgern, dass die Verwendung eines Raspberry Pi Mikrocomputers für die Implementierung des Register-Klient mit dem angeschlossenen RFID Leser ist eine lohnenswerte Entscheidung für in dieser Abschlussarbeit geschriebenen Aufgabe.

## 2.2 Kontaktlose Smartcards MIFARE

Anfänglich war die Smartcard eine Plastikkarte im ID-1-Format mit einer Größe von  $85,60 \times 53,98$  mm und abgerundeten Ecken (eine Standardbank / Kreditkarte hat die gleiche Form und Abmessungen). Darin ist ein Mikrochip eingebettet, dessen Kommunikationskontakte zu einer Seite herausgeführt werden. Später erschienen Smartcards im ID-000-Format, sie wurden in Mobiltelefonen verwendet und sind uns als SIM-Karten bekannt. Dann erschienen Smartcards ohne externes Kontaktfeld, und sie begannen, einen Funkkanal für die Kommunikation und Energieübertragung zu verwenden. Smartcards haben keine eigene Stromquelle und sind auf eine externe angewiesen. Der Hauptvorteil einer Smartcard ist die physische Sicherheit der darauf gespeicherten Daten. Der Mikrochip ist sehr klein und alles notwendiges befindet sich darauf, ohne dass interne Kontakte hergestellt werden müssen, an die es sich zum Abfangen angeschlossen werden kann.

Der ungefähre Lebenszyklus einer Smartcard (z. B. einer Bankkarte mit Chip) besteht aus mehreren Phasen. Daran sind der Chiphersteller, der Smartcard-Hersteller, der Kunde und der Kunde beteiligt:

- **Herstellung von Chips/Prozessoren.** Zu diesem Zeitpunkt schreibt der direkte Hersteller des Chips nach der physischen Produktion die vom Hersteller der Smartcards bereitgestellte universelle und für alle Karten des gleichen Typs dieselbe Firmware auf.
- **Initialisierung der Smartcard.** Die Chips werden an den Kartenhersteller gesendet. Er ändert die Firmware nach Bedarf. Beispielsweise schreibt er

eine eindeutige Seriennummer in jede Karte. Anschließend deaktiviert er die Möglichkeit, diese mit einer speziellen Anforderung zu ändern.

- **Smartcard-Herstellung.** Der Hersteller legt den Chip in Karten des gewünschten Formats ein und sendet diese an den Kunden, beispielsweise einer Bank.
- **Personalisierung.** Der Kunde schreibt unter Verwendung der Methoden der Firmware auf der Karte seine Anwendungen darauf, z. B. Bankgeschäfte, sowie zusätzliche Daten, z. B. Kundenname, Kontonummer usw. Danach wird die Karte durch eine spezielle Anfrage finalisiert, wonach beispielsweise die Aufzeichnung neuer Anträge eingeschränkt wird.
- **Ausgabe.** Die Karte wird dem Kunden ausgegeben und vom Kunden verwendet.
- **Zerstörung.** Die Karte wird weggeworfen.

Einer der größten Hersteller von Smartcards ist die Firma NXP Semiconductors mit Hauptsitz in den Niederlanden. Mit einer großen Produktvielfalt schafft diese Lösungen für kontaktlose Zugangs- und Zeitkontrollen und sichere kontaktlose Automobilzugangskontrollen. Die welt-weit meistgenutzte kontaktlose Chipkartentechnik „MIFARE“ wurde von NXP Semiconductors entwickelt und die Produktfamilie umfasst mittlerweile vier Produktreihen (siehe Abbildung ??) [NXP]. Alle MIFARE-ICs sind konform zur Norm ISO/IEC 14443 und erfüllen somit die Standards für die kontaktlose Kommunikation zwischen Chipkarten. Sie arbeiten mit 13,56 MHz und ihre Leseentfernung ist bis 10 cm möglich. Kontaktlose MIFARE-Smartcards verfügen über einen 1-KByte-Speicher, der in 16 Sektoren mit jeweils 16 Byte unterteilt ist. Die Datenspeicherdauer beträgt bis zu 10 Jahre. Die Anzahl der Umschreibzyklen beträgt 100.000 Zyklen. Neben NXP Semiconductors werden Chips für Mifare-Smartcards von der deutschen Firma Infineon im Rahmen einer Lizenzvereinbarung hergestellt [Too]. Nur Smartcards mit Chips von NXP Semiconductors und Infineon dürfen die Marke Mifare in ihrem Namen tragen. Nur Karten mit diesen Chips können als "Original" bezeichnet werden.

Die Studentenkarte der Beuth Hochschule wurden mit Mifare DESfire EV1 Kartenchip hergestellt. MIFARE DESFire EV1 ist die nächste Generation von Mifare Desfire mit einigen verbesserten Funktionen der Sicherheit und Verschlüsselung [Chi14, p.83]. Unberechtigte können sie aufgrund der AES-Verschlüsselung nicht auslesen. Zudem enthalten die Karten elektronisch keine persönlichen Daten [TBb]. Alle auf der Karte gespeicherte Daten werden kodiert, z.B. mit der UID der einzelnen

Karte verschlüsselt. MIFARE DESFire wird in vielen NFC- und RFID-Anwendungen verwendet, da er als sicherer Transponder mit einem von drei verschiedenen Typen von Verschlüsselung gesichert ist: Single DES, Triple DES oder AES. Im Allgemeinen wird AES als die sicherste Verschlüsselungsstufe der oben aufgeführten Methoden angesehen. Der AES-Authentifizierungsprozess besteht aus mehreren Schritten, in denen der NFC / RFID-Reader und das MFDFEV1-Tag verschlüsselte Daten austauschen, um zu überprüfen, ob sie denselben Schlüssel verwenden. Während dieses Vorgangs wird ein Sitzungsschlüssel erstellt, der für bestimmte Befehle wie den ChangeKey-Befehl verwendet wird [JI].

## 2.3 Sender-Empfänger-System mit RFID

Im vorherigen Kapitel 2.2 wurden die Grundlagen und Vorteilen der MIFARE-Technologie für Smartcards erklärt. Im aktuellen Kapitel wird über die Sender-Empfänger-System mit RFID beschrieben. Für die Implementierung der Aufgabe wird sowohl MIFARE-Smartcards als auch RFID-Tag-Mikrochips verwendet. Der letzte wird zur Identifizierung der auszuleihenden Raspi-Boards benutzt und auf der Rückseite des jeden Boards unter dem Schutzschirm geklebt werden. Beide Typs können mit dem RFID-Leser ACR122U-A9 von Advanced Card Systems abgelesen werden. Ab hier wird weiter nur den Begriff "Tag" sowohl für MIFARE-Smartcards als auch für RFID-Tag-Mikrochips verwendet und den Unterschied in Namen nicht weiter angemerkt. Die gespeicherte Daten warten darauf, gelesen zu werden. Die Antenne des Tags erhält Energie von einer RFID-Leseantenne. Mit der Stromversorgung der internen Batterie oder des Leseegeräts sendet das Tag Funkwellen an das Leseegerät zurück. Der Leser nimmt die Funkwellen des Tags auf und interpretiert die Frequenzen als Daten. RFID-Tags, die über einen Teil des elektromagnetischen Spektrums gesendet werden, und die genaue Frequenz können ausgewählt werden, um Interferenzen mit anderen elektronischen Geräten zu vermeiden. Der Leser sendet ein Signal an das Tag und liest seine Antwort [Tec]. Der Leser verfügt über einen Funkempfänger, der als Transceiver bezeichnet wird und ein codiertes Funksignal an das Tag sendet. Das Signal aktiviert das Tag und der Transponder wandelt das Signal dann in eine nutzbare Leistung um und sendet auf den Leser. Das Tag empfängt die Nachricht und antwortet dann mit seiner Identifikation und anderen Daten. Dies kann eine eindeutige Seriennummer des Produkts oder produktbezogene Daten sein. In der Abschlussarbeit wird nach der UID des Tags gefragt.

Während sich jedes Sender-Empfänger-System mit RFID hinsichtlich Gerätetyp und Komplexität unterscheidet, enthält jedoch jedes RFID-System mindestens die folgenden vier Komponenten: Leser, Antennen, Tags und Kabel. Das einfachste System kann aus einem mobilen Hand-RFID-Lesegerät (mit integrierter Antenne)

und RFID-Tags bestehen, während komplexere Systeme mit Multi-Port-Lesegeräten, GPIO-Boxen, zusätzlichen Funktionsgeräten, mehreren Antennen, Kabeln und RFID-Tags ausgelegt sind und ein komplettes Software-Setup benötigen können. RFID-Tag-Typ Bestimmt das RFID-System. Derzeit gibt es drei verschiedene Arten von Tags: **aktiv, semi-passiv und passiv**. Ein aktives Tag sendet mit seiner Batterie Radiowellen an einen Leser, während eine semi-passive Tag-Batterie in Gegenwart eines Lesers aktiviert wird. Aktive und semi-passive Tags werden über größere Entfernungen gelesen. Sie senden hohe Frequenzen von 850 bis 950 MHz, die von 30 Meter oder mehr gelesen werden können. Zusätzliche Batterien können die Reichweite eines Tags auf über 90 Meter erhöhen. Passive RFID-Tags haben keine Batterie und verwenden die vom Lesegerät übertragene Funkenergie als Stromquelle. Diese Tags werden bis zu ein paar Meter entfernt gelesen und sind kostengünstiger [Tec]. Für die Markierung der Raspi-Boards im PSE-Labor werden die passive RFID-Tags verwendet, die für vorgesehenen Zwecken ideal dienen. Die runde benutzte im Abschlussprojekt RFID-Tags sind auf der Abbildung ?? im Vergleich in der Größe zu 1-Euro Münze zu sehen. RFID-Systeme können nach Tag- und Lesertyp klassifiziert werden: passiver Leser - aktiver Tag (PRAT), aktiver Leser - passiver Tag (ARPT) und aktiver Leser - aktiver Tag (ARAT). Das ARPT-System wird im PSE-Labor eingesetzt und verfügt über einen aktiven Leser und empfängt Authentifizierungssignalantworten von passiven Tags. Ich möchte an dieser Stelle auch noch anmerken, dass der Initialisierungsprozess für eine kontaktlose Karte viel komplizierter als für eine Kontaktkarte ist. Das meiste davon wird vom sogenannten Antikollisionszyklus besetzt. Eine Kollision tritt auf, wenn mehr als eine Karte gleichzeitig auf das elektromagnetische Feld der RFID-Leser trifft und der diese Karten voneinander unterscheiden muss. Der Algorithmus dieses Prozesses ist sehr komplex und umfasst mehrere zehn Seiten Beschreibung in den Normen ISO/IEC 14443-2 und ISO / IEC 14443-3. Daher werde ich ihn hier nicht angeben, da es nicht von Entwickler wirklich etwas zu machen benötigt wird - das Terminal und der RFID-Leser sind voll damit beschäftigt.

## 2.4 Django Framework

Django ist ein Open Source-Webentwicklungsframework, das in der Python-Sprache geschrieben ist. Es wurde entwickelt, um so viele Prozesse wie möglich zu automatisieren, sodass es sich auf die Softwareentwicklung konzentriert werden können, ohne Zeit mit dem Einrichten des Servers verschwenden zu müssen. Django wurde ursprünglich so konzipiert, dass es lose Kopplung zwischen verschiedenen Teilen der Infrastruktur hat, sodass es unabhängig voneinander gearbeitet werden kann. Diese Unabhängigkeit bedeutet, dass es in der Entwicklungsprozess nur die Teile von Django verwendet werden kann, die benötigt werden, ohne sich um Probleme mit

der Interdependenz von Komponenten kümmern zu müssen. Durch die Verwendung von Django wird die erforderliche Codemenge reduziert, wodurch das Schreiben von Webanwendungen schneller und die Wartung Ihrer Anwendung in Zukunft erheblich vereinfacht wird. Django folgt strikt dem DRY-Prinzip (Don't Repeat Yourself), dass jeder einzelne Code oder jede einzelne Daten an nur einem Ort gespeichert werden sollte. Dies vereinfacht und beschleunigt den Softwareänderungsprozess erheblich, da eine Anwendung, die geändert werden muss, an einem einzigen Ort ausgeführt werden muss.

Nach der Django Installation und dem Erstellen eines Projekts ist es nützlich, die erstellte Projektstruktur zu betrachten [Foub]:

- **manage.py** ist ein Befehlszeilenprogramm, mit dem auf verschiedene Weise mit Django-Projekt interagiert wird . Diese Datei muss nicht geändert werden.
- **acaLoanRaspiBoard** Projektarbeitsverzeichnis enthält:

*\_\_init\_\_.py*: Eine leere Datei, die Python mitteilt, dass dieses Verzeichnis für das Python-Paket bestimmt ist.

*settings.py*: Die Einstellungen und Konfigurationsdatei für das Django-Projekt.

*urls.py*: Die URL-Beschreibungsdatei für dieses Django-Projekt.

*asgi.py*: Damit kann die Anwendung mit einem Webserver unter Verwendung des ASGI-Protokolls arbeiten.

Die generierte Datei settings.py enthält eine Grundkonfiguration für die Verwendung einer SQLite-Datenbank und eine Liste von Django-Anwendungen, die standardmäßig zum Projekt hinzugefügt werden sollen. In der Datei settings.py ist es möglich, den Debug-Modus des Projekts zu aktivieren / deaktivieren. Wenn "Debug=true" festgelegt ist, zeigt der Server bei Auftreten einer nicht erfassten Ausnahme detaillierte Fehlermeldung an. Es muss beim Wechsel zur Produktionsversion auf "false" gesetzt, da mit aktiviertem Debugging alle Benutzer vertrauliche Projektdaten sehen können.

Die Architektur von Django basiert auf den Ideen des The Model-View-Controller (MVC) Design Pattern, womit Anwendungsschicht, Benutzeroberfläche (UI) und Datenzugriffsebenen getrennt werden, sodass jede dieser Ebenen unabhängig von

anderen geändert werden kann. Als Konzept ist das MVC-Entwurfsmuster wirklich einfach zu verstehen [Geo17, pp. 15-16]:

- **Das Modell (M)** ist ein Modell oder eine Darstellung Ihrer Daten. Es handelt sich nicht um die tatsächlichen Daten, sondern um eine Schnittstelle zu den Daten. Mit dem Modell können Sie Daten aus Ihrer Datenbank abrufen, ohne die Feinheiten der zugrunde liegenden Datenbank zu kennen. Das Modell stellt normalerweise auch eine Abstraktionsschicht mit Ihrer Datenbank bereit, sodass Sie dasselbe Modell mit mehreren Datenbanken verwenden können.
- **Die Ansicht (V)** ist das, was mit den Augen gesehen werden kann. Dies ist die Präsentationsebene für das Modell. Auf dem Computer wird die Ansicht im Browser für eine Web-App oder in der Benutzeroberfläche für eine Desktop-App angezeigt. Die Ansicht bietet auch eine Schnittstelle zum Sammeln von Benutzereingaben.
- **Der Controller (C)** steuert den Informationsfluss zwischen dem Modell und der Ansicht. Mithilfe der programmierten Logik wird entschieden, welche Informationen über das Modell aus der Datenbank abgerufen und welche Informationen an die Ansicht übergeben werden. Es erhält auch Informationen vom Benutzer über die Ansicht und implementiert die Aufgaben der Anwendung: entweder durch Ändern der Ansicht oder durch Ändern von Daten über das Modell oder durch Ändern die beiden Elementen.

## 2.5 Grundlagen der Webanwendungen

In diesem Kapitel werden die grundlegenden Begriffe zum Entwerfen und Erstellen einer Webanwendung erläutert. Hierbei konzentrieren wir uns auf die Probleme im Zusammenhang mit der Verarbeitung von HTTP-Anfragen und -Antworten (siehe Kapitel 2.5.2). Da in den letzten Jahren sich Webanwendungen rasant weiterentwickelt und die Desktop-Lösungen schrittweise ersetzt haben, es sollte auch nicht unerwähnt bleiben, welche Vorteile die Webanwendungen haben:

- **Zugriff von jedem Gerät** Die Webanwendung kann überall auf der Welt von einem Computer, Tablet oder Smartphone zugegriffen und verwendet werden. Notwendig ist, dass dem Gerät eine Internetverbindung zur Verfügung steht.
- **die Kostenersparnis** Webanwendungen können auf allen Plattformen ausgeführt werden und müssen nicht mehr separat für Android und iOS entwickelt werden.



- **Anpassungsfähigkeit** Wenn native Anwendungen bestimmte Betriebssysteme erfordern, jedoch können jedes Betriebssystem (Windows, MAC, Linux usw.) und jeder Browser (Internet Explorer, Opera, FireFox, Google Chrome usw.) für die Arbeit mit einer Webanwendung. ) verwendet werden.
- **Keine Software zum Herunterladen** Es ist günstig und einfach dem Endnutzer zu liefern, zu warten und zu aktualisieren. Das Aktualisieren auf die neueste Version erfolgt beim nächsten Laden der Webseite.
- **Netzwerksicherheit** Das Websystem verfügt über einen einzigen Einstiegspunkt, der zentral geschützt und konfiguriert werden kann.
- **Skalierbarkeit** Mit zunehmender Belastung des Systems ist es nicht erforderlich, die Leistung des Computer von Endbenutzer zu erhöhen. Mit einer Webanwendung kann in der Regel nur mithilfe von Hardwareressourcen eine größere Datenmenge verarbeitet werden, ohne den Quellcode neu zu schreiben und die Architektur ändern zu müssen.
- **Verhinderung von Datenverlust** Benutzerdaten werden in der "Cloud" gespeichert, für deren Integrität die Hosting-Anbieter verantwortlich sind, deswegen vom Verlust geschützt, falls die Festplatte des Computers beschädigt wird.

Ich möchte an dieser Stelle auch noch ein Konzept des Caches erklären, bei dem häufig verwendete Daten, anstatt jedes Mal aus der Datenbank abgerufen, berechnet oder auf andere Weise vorbereitet, an einem schnell zugänglichen Ort gespeichert werden. Zum Beispiel erhielt Django eine Anfrage, Daten für ein Diagramm in einem Bericht abzurufen. Daten aus der Datenbank werden abgeholt, vorbereitet und abgelegt in einer Datenbank mit schnellem Zugriff z.B. "memcached", die beispielsweise für eine Stunde zwischengespeichert wird. Bei der nächsten Anfrage werden die notwendigen Daten sofort von "memcached" erhalten und an das Frontend gesendet. Wenn es festgestellt wird, dass die Daten nicht mehr relevant sind, werden sie ungültig beziehungsweise gelöscht aus dem Cache.

### 2.5.1 Client-Server Kommunikation

Als nächstes geht die Frage nach, wie Client und Server miteinander kommunizieren können. Der Client spricht mit dem Server über das HTTP-Protokoll. Das HTTP-Protokoll setzt die Verwendung einer Client-Server-Datenübertragungsstruktur voraus. Es verwendet normalerweise Port 80 und als Transportschichtprotokoll - TCP. Das HTTP-Protokoll ist in RFC 1945 (HTTP 1.0 [TBL]), 2068 [RF] und 2616 (HTTP 1.1) definiert. Die Clientanwendung generiert eine Anforderung und sendet sie



an den Server. Anschließend verarbeitet die Serversoftware diese Anforderung, generiert eine Antwort und sendet sie an den Client zurück [RR03, pp.33-34]. Die Clientanwendung kann dann weiterhin andere Anforderungen senden, die auf ähnliche Weise behandelt werden. Wenn ein Webserver eine Anforderung zum Bereitstellen einer statischen Webseite erhält, sendet er die Seite direkt an den Browser.[Ado] Wenn jedoch eine dynamische Seite angefordert wird, sind die Handlungsweise des Webserver nicht so einfach. Der Server übergibt die Seite an ein spezielles Programm, das die letzte Seite bildet. Ein solches Programm wird als Anwendungsserver bezeichnet. Der Anwendungsserver liest den Code auf der Seite, rendert die letzte Seite gemäß dem gelesenen Code und entfernt sie dann von der Seite. Als Ergebnis all dieser Operationen wird eine statische Seite erhalten, die an den Webserver übertragen wird, der sie wiederum an den Client-Browser sendet. Alle Seiten, die der Browser empfängt, enthalten nur HTML-Code [Ado].

Dieses grundlegende Paar Anforderung-Antwort wird im nächsten Kapitel 2.5.2 untersucht. In der Praxis kommunizieren Server und Browser selten direkt - dazwischen befinden sich ein oder mehrere Proxys. Eine Verbindung ist als virtuelle Verbindung definiert, die aus HTTP-Agenten besteht, einschließlich des Browsers, des Servers und der am Austausch teilnehmenden Zwischenproxys (Abbildung ?? [RR03, pp.33-34]).

HTTP definiert eine Reihe von Anforderungsmethoden, um die gewünschte Aktion anzugeben, die für eine bestimmte Ressource ausgeführt werden soll. Sie werden als HTTP-Verben bezeichnet. Diese Anforderungsmethoden sind unten aufgeführt:

- **GET** ist die einfachste Anforderungsmethode. Wenn eine URL in Browser eingeben oder auf einen Hyperlink geklickt wird, um eine andere Seite zu besuchen, verwendet der Browser die GET-Methode, um die Anforderung an den Webserver zu senden.
- **POST** wird benutzt, um eine neue Ressource zu erstellen. POST-Anforderungen enthalten normalerweise Daten zum Erstellen einer neuen Ressource.
- **PUT** ist zum Aktualisieren einer vorhandenen Ressource. Der Inhalt kann aktualisierte Daten für die Ressource enthalten.
- **DELETE** ist zum Löschen einer vorhandenen Ressource.

## 2.5.2 Grundlagen der REST API

In der Client-Server-Kommunikation haben wir zunächst zwei Partner: den Client und den Server. Um die Kommunikation zwischen diesen beiden Partnern zu verstehen, müssen wir einige einfache Themen kennen:

- **Anforderungen** werden vom Client gesendet, um den Server nach Daten wie Dateien zu fragen oder den Server über Ereignisse zu informieren, z. B. dass sich ein Benutzer mit seinen Anmeldeinformationen anmelden möchte
- **Antwort** wird vom Server an den Client gesendet und ist die Reaktion des Servers auf eine Anforderung des Clients.

Die ankommenden Anforderungen muss vom Server interpretiert und weitergeleitet werden. Während dieses Schritts übernimmt der Webserver die Verantwortung für Bestimmung, welche Maßnahmen zur Verarbeitung der Anforderung erforderlich sind. Diese Aktion kann das einfache Abrufen einer statischen HTML-Datei oder der Zugriff zur Datenbank sein. Das kann komplexe Funktionen ausführen, die mit einer vollständigen Webanwendung verbunden sind [RR03, p.204]. Wenn ein Benutzer versucht, auf eine Webanwendung zuzugreifen, sendet der Browser zuerst eine HTTP-Anforderung an einen Webserver. Die mit der Anforderung verknüpften Informationen (häufig als Anforderungskontext bezeichnet) umfasst die weitere Informationen, die in der URL, dem Anforderungsheadern und dem Nachrichtentext enthalten sind. Es enthält auch Sitzungsinformationen, die den Anwendungsstatus über aufeinanderfolgende Anforderungen hinweg beibehalten [RR03, p.205].

Die Client-Server-Kommunikation ist zustandslos. Dies bedeutet, dass sich der Server keine Informationen über den Status des Clients merken muss, während der Client alle Informationen in seine Anforderung aufnehmen muss.

Das REST- oder RESTful-API-Design (Representational State Transfer) ist eine Client-Server-Architektur und nutzt normalerweise HTTP-Protokoll [ML]. Der Server speichert und bearbeitet Informationen und stellt sie dem Benutzer auf effiziente Weise zur Verfügung. Der Client nimmt diese Informationen und zeigt sie dem Benutzer an und verwendet sie, um nachfolgende Informationsanforderungen auszuführen. Diese Trennung von Funktionsweise ermöglicht es sowohl dem Client als auch dem Server, sich unabhängig voneinander zu existieren und entwickelt werden, da nur die Kommunikationsschnittstelle gleich bleiben muss [Dix]. In einem REST-API-Aufruf werden eine Teilmenge der HTTP-Methoden für die Aktionen verwendet, die wir ausführen müssen. Diese Methoden wurden oben im Kapitel 2.5.1 schon erklärt. REST-APIs geben normalerweise die von Benutzer angeforderten Daten im

JSON-Format zurück, dessen Payload eine String (Zeichenfolge) ist. Bevor die Daten tatsächlich verwendet werden können, müssen diese Zeichenfolge analysiert und in Objekte umgewandelt werden. Die Funktionsweise des REST-APIs von acaLoan-System sind in Kapitel ?? nachzulesen.

### 2.5.3 Clientseitiges JavaScript

Das Document Object Model (DOM) ist eine Programmierschnittstelle für HTML- und XML-Dokumente. Es stellt die Seite dar, sodass Programme die Dokumentstruktur, den Stil und den Inhalt ändern können [conb]. Clientseitiges JavaScript (CSJS) ermöglicht die Verbesserung und Bearbeitung von Webseiten über den sofortigen Zugang zu den Knoten des DOM: in einer Browserumgebung hat Ihr Code Zugriff auf Dinge, die nur vom Browser bereitgestellt werden, wie das Dokumentobjekt für die aktuelle Seite, das Fenster, Funktionen wie Warnungen, die eine Nachricht anzeigen usw. Die Hauptaufgaben von clientseitigem JavaScript sind die Validierung von Eingabe, Animation, Bearbeitung von UI-Elementen, Anwenden von Stilen und einige kleine Berechnungen können durchgeführt werden. Bei der Webentwicklung ist es der Browser auf dem Computer des Benutzers, der diesen Javascript Code ausgeführt [conc]. Bei der Implementierung der Aufgabestellung für acaLoan-System wird davon ausgegangen, dass der Chrome-Browser als Rendering-Frontend verwendet wird, dies ist jedoch nicht streng Anforderung und der Client sollte ohne Probleme mit anderen modernen Browsern arbeiten, da der Code des clientseitigen JavaScripts in einer Mehrheit von Browsern ausgeführt werden kann.

Ein wichtiges Merkmal von JavaScript ist die Möglichkeit, Ereignishandler (event handlers) zu definieren - beliebige Codeteile, die ausgeführt werden, wenn ein bestimmtes Ereignis auftritt. Normalerweise werden diese Ereignisse vom Benutzer ausgelöst, wenn er beispielsweise die Maus über einen Hypertext-Link bewegt, einen Wert in ein Formular eingibt oder in einem Formular auf die Taste Senden klickt. Diese Funktion zur Ereignisbehandlung hat eine wichtigste Bedeutung, da für die Programmierung mit grafischen Oberflächen wie HTML-Formularen ein ereignisgesteuertes Modell erforderlich ist. JavaScript kann jede Art von Aktion als Reaktion auf Benutzerereignisse auslösen. Typische Beispiele könnten sein, eine spezielle Nachricht in der Statuszeile anzuzeigen, wenn der Benutzer die Taste drückt. Sodass wird es im acaLoan-System nach dem Drücken der Taste "Loan / Return Board" und erfolgreichen Ablesen der Studentenkarte den Name der Studierende und die Nummer des schon zugewiesenen Boards angezeigt.

In diesem Kapitel wurden die Grundlagen der Webanwendungen erklärt, die als Überblick zum Verstehen der zukünftigen Verwendung der entwickelten Software im PSE-Labor dienen sollten.

# Network Architecture

The history of artificial neural networks is filled with colorful, creative individuals from a variety of fields, many of whom struggled for decades to develop concepts that we now take for granted. This history has been documented by various authors. One particularly interesting book is *Computing: Foundations of Research* by John Anderson and Edward Rosenfeld. They have collected and edited a set of some 43 papers of special historical interest. Each paper is preceded by an introduction that puts the paper in historical perspective.

## 3.1 Layers

The artificial neural networks discussed in this text are only remotely related to their biological counterparts. In this section we will briefly describe those characteristics of brain function that have inspired the development of artificial neural networks.

# Implementierung

Here some code for my super puper neural network:

```
class StudentFactory(DjangoModelFactory):
    class Meta:
        model = Student
    student_card = factory.SubFactory(StudentCardFactory)
    first_name = factory.Faker('first_name')
    second_name = factory.Faker('last_name')
    matricul_no = factory.Faker('random_int', min=850000, max=
        =950000)
    hrz_no = factory.Faker('random_int', min=65000, max=69000)
    group = StudentGroup.A_GROUP
    is_home_loan_enabled = True
```

**Listing 4.1:** StudentFactory

## Analyse

Bei der Entwicklung benutzerdefinierter Webanwendung "acaLoan" wurde Python einer anderen Programmiersprache bevorzugt, um ihre einfache und ausdrucksstarke Syntax zu nutzen. Da Django in der Programmiersprache Python geschrieben ist, wird es auch sinnvoller, die Abschlussarbeit in Python Code zu erstellen, indem die alle Komponenten Syntaxregeln von Python nutzen. Django stellt die Ressourcen bereit, die während der Implementierung der analysierte Software sofort benötigt wurden: allgemeine Vorgänge wie Datenbankmanipulation, HTML-Vorlagen, URL-Routing, Sitzungsverwaltung und Sicherheit. Mit der MVC-Prinzip (Model-View-Controller) sind die Benutzeroberfläche (UI) und die Geschäftslogikebenen des acaLoan-Systems getrennt. Django unterstützt verschiedene Betriebssysteme wie Windows, Linux und MacOS, damit es keine strenge Softwareabhängigkeiten bei acaLoan-System während sowohl der Bereitstellung des Djangos mit ASGI als auch in der Zukunft gegeben werden sollten.

Für sowohl eine Markierung der Raspi-Boards als auch für eine Authentifizierung der Studierenden werden RFID-Tags benutzt. Da RFID Funkwellen zur Kommunikation verwendet werden, müssen sich RFID-Tags nur innerhalb des Lesebereichs des Lesegeräts befinden. Das lässt die kleine NFC-Tags auf der Rückseite des jeden Boards unter dem Schutzschirm zu kleben, sodass der Board am RFID-Leser ohne präzise Positionierung auf dem Lesefeld angelesen werden kann. Da die Studentenkarte der Beuth Hochschule mit MIFARE DESFire kontaktlose Chipkartentechnik hergestellt sind, wird es möglich, in acaLoan-System nur mit einem RFID-Leser zu arbeiten und sowohl die Daten der Studierende als auch die Raspi-Boards abzulesen.

An dieser Stelle muss es gesagt werden, dass die RFID-Tags bestimmte Nachteile haben, die wurden zwischen den Mitarbeiter und Autorin der Abschlussarbeit diskutiert. Es wurde jedoch die Entscheidung getroffen, dass RFID-Tags für die vorgesehenen Zwecken des Verfolgen des Raspi-Boards (welchen Raspi-Board von welchem Student am welchen Tag ausgeliehen wurde und bis zum welchen Tag zurückgegeben muss) die Sicherheitserwartungen der PSE-Labor Mitarbeiter erfüllen. RFID-Tags sind aus mehreren Gründen im Vergleich nicht ideal. Da ein RFID-Tag nicht zwischen Lesegeräten unterscheiden kann, können die Informationen von fast jedem gelesen werden, sobald sie die ursprüngliche Lieferkette verlassen haben. Weil RFID-Lesegeräte so tragbar sind und die Reichweite einiger Tags so groß ist, können

Betrüger Informationen sammeln, auf die sie sonst keinen Zugriff hätten. Dies bedeutet, dass jeder ohne Wissen einer Person potenziell sensible Informationen sammeln kann. Diese Nachteile der RFID-Tags wurden vernachlässigt, da sowohl Studentenkarte als auch geklebte auf den Raspi-Boards RFID-Tags keine sensible Information behalten und eine kleine Reichweite von bis zu 10 cm haben. Es gäbe zwei Arten von Bedrohungen. Entweder könnte die Studentenkarte von einem Täter geklont werden, um sich für einen Student auszugeben und ein Raspi-Board stehlen zu könnten. Oder ein Raspi-Board Tag könnte geklont werden, um einen Datensatz in der Datenbank zu erzeugen, über das Zurückgeben des Boards, obwohl in der Realität der Raspi-Board nie ins Labor zurückgekommen war. Gegen das Klonen des Raspi-Tags wurde es besprochen, dass in der Zukunft im PSE-Labor im Schrank mit den Raspi-Boards jeden Platz für jeden entsprechenden Raspi-Board mit einem Gewichtssensor ausgerüstet werden wird und acaLoan-System auf der Erscheinung des bestimmten Gewicht erwarten werden wird. Dies ist aber nicht der Teil bestehenden Abschlussarbeit und von Mitarbeiter des PSE-Labor als eine spannende Aufgabe für die andere Abschlussarbeit vorgesehen ist. Gegen das Klonen der Studentenkarte wurde es zuerst entschieden, dass ein bestehender Zugang zu einem Schrank mit Raspi-Boards entlang der beiden Arbeitstische der Mitarbeiter des PSE-Labor eine bestimmte Sicherheit gewährleisten könnte. Die anderen Lösungen werden nach der Lieferung der Software diskutiert und liegen ebenfalls außerhalb des Rahmens der bestehenden Abschlussarbeit.

Obwohl oben die Nachteile der RFID-Tags erwähnt wurde, lässt sich zusammenfassend sagen, dass die neue Studentenkarte, die an der Beuth Hochschule ab Sommersemester 2018 verwendet wurde, eine zuverlässige und zeitgemäße Lösung ist. Die Karte beinhaltet keine elektronischen persönlichen Daten der Studierenden und die Campus-Automaten abrufen alle persönlichen Daten anhand eines Pseudonyms online abrufen müssen (d.h. liegen in den Automaten auch keine persönlichen Daten vor). Sodass im Fall des Verlusts die persönlichen Daten von den Unberechtigten nicht ausgelesen werden können [TBb]. Das stand im Fokus der Entscheidung, eine Studentenkarte als einziges elektronisches Identifizierungsmittel beim Ausleihe/Rückgabevorgängen im PSE-Labor zu benutzen.

In dieser Abschlussarbeit wurde nachgewiesen, dass das Verleihprozedere für die Raspi-Boards (Lab und Home) mit gewählten Mitteln für die Verwendung im PSE-Labor erfolgreich automatisiert wurde.

## Zusammenfassung

Das zu Beginn der Arbeit gesetzte Ziel der Entwicklung von Software für PSE-Labor wurde erfolgreich erreicht. Die Implementierung aller Bestandteile und ein Zusammenspiel unter der Verwendung des Entwicklungsservers wurde erfolgreich den Mitarbeitern des PSE-Labors präsentiert. Mit dem acaLoan-System werden die Studierende in die Lage versetzt, einen Raspi-Board selbständig im PSE-Labor für die Übungen auszuleihen und später zurückzugeben. Damit wird ein Arbeitszeitverbrauch für die Verwaltung der Bordstandortbestimmung reduziert und die Mitarbeitern können sich auf weitere neuen geistliche wissenschaftlichen Herausforderungen konzentrieren und damit in der Weiterentwicklung des acaLab-Projekts hineinbringen. Der Funktionsumfang des acaLoan-Systems erlaubt es den Studierenden, den Raspberry Board entweder für die Übung im Labor oder für die selbständige Arbeit zu Hause auszuleihen.

Darüber hinaus wird der Kommunikation der Studierende mit dem acaLoan-System auf notwendiges Minimum reduziert, sodass ein Ausleihe- / Rückgabevorgang nur mit drei Tastendruck erledigt wird. Zuerst wird die Taste "Loan / Return Board" gedrückt, um eine neue Benutzersitzung zu starten. Das Ablesen der Studentenkarte und Raspi-Boards geschieht ohne weitere Tastendrucke nur über das Register-Client, der die abgelesenen Daten dem Server in den Endpunkt der REST API schickt. Es wird abhängig von dem Typ des Boards bestimmt, ob abgelesenes Board zum Ausleihen oder Rückgabe ist. Mit dem zweiten Tastendruck bestätigt der Studierende seinen Wunsch, das Board auszuleihen oder zurückzugeben. Mit dem dritten Tastendruck nach dem erfolgreichen Beenden der Operation, beendet der Studierende seine Sitzung und gibt das acaLoan-System dem nächsten Studierenden wieder frei.

Für die Implementierung des acaLoan-Systems ist ein ausführliches Software Engineering in Form der objektorientierten Analyse und des Designs der endlichen Zustandsmaschine durchgeführt worden. Das acaLoan-System konnte ohne Server nicht realisiert werden, da Benutzerdaten länger als eine Sitzung gespeichert werden müssen: die Studentenkarten und die Datensätze aller Ausleihvorgänge müssen mindestens für ein laufendes Semester gespeichert werden, damit die Mitarbeiter des PSE-Labors immer einen Zugang zu allen gespeicherten vorherigen Leihvorgängen (von der Ausleihe bis zur Rückgabe eines Boards) bekommen können. Der Server ist erfolgreich mit Django Webframework realisiert. Es ist auch vorgesehen, dass am Ende des Semesters



nach dem letzte Rückgabe eines Boards die Datensätzen des zu Ende gegangen Semesters gelöscht werden können. Der Zugriff von Server auf Display-Client wird durch entsprechende Schnittstellen mithilfe der RESTful Kommunikation ermöglicht. In der realisierten Abschlussarbeit wurde auch ein notwendigen Teil namens Register-Client entwickelt, an dem ein RFID-Leser angeschlossen wurde. Der Register-Client verfügt selbst über keinen Datenbank und darf nur die abgelesene Daten dem Server schicken. Es geht um eine Simplex-Verbindung, da ein Nachrichtenverkehr asymmetrisch ist, weil der Register-Client keine Daten vom Server zurückbekommen darf und über den erfolgreiche oder gescheiterte Leihvorgang nicht wissen muss. Für die Implementierung des Register-Clients wurde uComputer Raspberry Pi benutzt, der schon zur Verfügung im PSE-Labor stand und ergänzend nicht geliefert werden muss.

# Glossar

## AJAX

AJAX (asynchrones Javascript und XML) ist der allgemeine Name für Technologien, mit denen asynchrone Anforderungen (ohne erneutes Laden von Seiten) an den Server gestellt und Daten ausgetauscht werden können. Da die Client- und Serverteile der Webanwendung in verschiedenen Programmiersprachen geschrieben sind, müssen zum Austausch von Informationen die Datenstrukturen (z. B. Listen und Wörterbücher), in denen sie gespeichert sind, in das JSON-Format konvertiert werden.

## ASGI

Asynchronous Server Gateway Interface ermöglicht nicht nur mehrere eingehende und ausgehende Ereignisse für jede Anwendung, sondern auch eine Hintergrundkoroutine, damit die Anwendung andere Aufgaben ausführen kann.

## BGA

BGA oder Ball Grid Array ist eine Art oberflächenmontiertes Gehäuse, das in elektronischen Produkten zur Montage integrierter Schaltkreise wie Mikroprozessoren, FPGAs, WiFi-Chips usw. verwendet wird. Die Anschlüsse liegen in Form von Lötugeln vor, die in einem Gitter angeordnet wie Muster auf der Unterseite des Gehäuses sind, um den für die Verbindungen verwendeten Bereich zu vergrößern.

## Daemon

Ein Unix-Daemon ist ein Programm, das "im Hintergrund" ausgeführt wird, ohne dass die Steuerung über ein Terminal erforderlich ist, und dem Benutzer die Möglichkeit bietet, andere Prozesse "im Vordergrund" auszuführen. Der Dämon kann entweder von einem anderen Prozess gestartet werden, z. B. von einem der Systemstartskripte, ohne auf ein Steuerterminal zuzugreifen, oder vom Benutzer von einem beliebigen Terminal aus. In diesem Fall "entführt" der Dämon das Terminal jedoch nicht, während es ausgeführt wird.

## DOM

DOM (Document Object Model) ist die Struktur einer HTML-Seite. Bei der Arbeit mit dem DOM werden HTML-Tags (Elemente auf einer Seite) gefunden, hinzugefügt, geändert, verschoben und entfernt.

## GPU

Die Grafikverarbeitungseinheit ist ein programmierbarer Prozessor, der auf das Rendern aller Bilder auf dem Computerbildschirm spezialisiert ist. Eine GPU bietet die schnellste Grafikverarbeitung, und für Gamer ist die GPU eine eigenständige Karte, die an den PCI Express (PCIe) -Bus angeschlossen ist. GPUs wurden ursprünglich entwickelt, um Bilder für Computergrafik zu erstellen, jedoch seit Anfang 2010 können GPUs auch verwendet werden, um Berechnungen mit großen Datenmengen zu beschleunigen.

## HOST

Host - Dies der Name der IP-Adresse für den Webserver, auf den zugegriffen wird. Dies ist normalerweise der Teil der URL, der unmittelbar auf den Doppelpunkt und zwei Schrägstriche folgt.[RR03, p.31]

## HTML-Vorlage

Eine HTML-Vorlage ist eine intelligente HTML-Seite, die Variablen anstelle bestimmter Werte verwendet und verschiedene Operatoren bereitstellt: if (if-then),

for-Schleife (Durchlaufen einer Liste) und andere. Das Abrufen einer HTML-Seite aus einer Vorlage durch Ersetzen von Variablen und Anwenden von Operatoren wird als Vorlagenrendering bezeichnet. Die resultierende Seite wird dem Benutzer angezeigt. Falls einen anderen Abschnitt zu öffnen ist, muss ein anderen Musters geladen werden. Wenn andere Daten in der Vorlage verwendet werden müssen, werden sie vom Server angefordert. Alle Formularübermittlungen mit Daten sind auch AJAX-Anforderungen an den Server.

## HTTP

HTTP steht für HyperText Transfer Protocol, Hypertext Transfer Protocol". HTTP ist ein weit verbreitetes Datenübertragungsprotokoll, das ursprünglich für die Übertragung von Hypertextdokumenten vorgesehen war (Dokumente, die möglicherweise Links enthalten, mit denen Sie den Übergang zu anderen Dokumenten organisieren können). Die Basis dieses Protokolls ist eine Anforderung von einem Client (Browser) an einen Server und eine Serverantwort an einen Client.

## JSON

JSON (JavaScript Object Notation) ist ein universelles Format für den Datenaustausch zwischen einem Client und einem Server. Es ist eine einfache Zeichenfolge, die in jeder Programmiersprache verwendet werden kann.

## PORT

Dies ist ein optionaler Teil der URL, der die Portnummer angibt, die der Zielwebserver abhört. Die Standardportnummer für HTTP-Server ist 80, einige Konfigurationen sind jedoch so eingerichtet, dass sie eine alternative Portnummer verwenden. In diesem Fall muss diese Nummer in der URL angegeben werden. Die Portnummer wird direkt mit einem Doppelpunkt, der unmittelbar auf den Servernamen oder die Adresse folgt, eingegeben.[RR03, p.31]

## RFID

RFID (englisch radio-frequency identification oder „Identifizierung mit Hilfe elektromagnetischer Wellen“) bezeichnet eine Technologie für Sender-Empfänger-Systeme

und wird bei der Abschlussarbeit verwendet, um die vorhandenen zur Ausleihe Raspberry Pi Boards zu markieren und identifizieren.

## URI

Uniform Resource Identifier ist ein Pfad zu einer bestimmten Ressource (z.B. einem Dokument), für die eine Operation ausgeführt werden muss (z. B. bei Verwendung der GET-Methode bedeutet dies das Abrufen einer Ressource). Einige Anforderungen beziehen sich möglicherweise nicht auf eine Ressource und in diesem Fall kann der Startzeile anstelle des URI ein Sternchen (Symbol "\*") hinzugefügt werden.

# Literaturverzeichnis

- [Ado] Adobe. *Understand web applications*. <https://helpx.adobe.com/dreamweaver/user-guide.html/dreamweaver/using/web-applications.ug.html>. abgerufen am 29. September 2020.
- [Chi14] Ugo Chirico. *Smart Card Programming. A comprehensive guide to smart card programming*. lulu.com London, 2014. ISBN: 978-1-291-61050-5.
- [cona] MDN contributors. *Django Tutorial Part 10: Testing a Django web application*. <https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Testing>. abgerufen am 12. November 2020.
- [conb] MDN contributors. *Introduction to the DOM*. [https://developer.mozilla.org/en-US/docs/Web/API/Document\\_Object\\_Model/Introduction](https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction). abgerufen am 7. November 2020.
- [conc] Net informations.com contributors. *JavaScript: Client side Vs. Server side*. <http://net-informations.com/js/iq/side.htm>. abgerufen am 8. November 2020.
- [CR] David Corcoran and Ludovic Rousseau. *pcscd(8) - Linux man page*. <https://linux.die.net/man/8/pcscd>. abgerufen am 05. Oktober 2020.
- [Dix] Shubhang Dixit. *Beginners Guide to Client Server Communication*. <https://medium.com/@subhangdxt/beginners-guide-to-client-server-communication-8099cf0ac3af>. abgerufen am 5. November 2020.
- [Edi] Perdita Stevens. School of Informatics. University of Edinburgh. *Class diagrams and architectural design*. <http://www.inf.ed.ac.uk/teaching/courses/inf2c/lectures/SE/architectureclassH0.pdf>. abgerufen am 17. Oktober 2020.
- [Foua] Django Software Foundation. *The Django admin site*. <https://docs.djangoproject.com/en/3.1/ref/contrib/admin/>. abgerufen am 2. November 2020.
- [Foub] Django Software Foundation. *Writing your first Django app, part 1*. <https://docs.djangoproject.com/en/3.1/intro/tutorial01>. abgerufen am 12. Oktober 2020.

- [Fouc] Django Software Foundation. *Writing your first Django app, part 2*. <https://docs.djangoproject.com/en/3.1/intro/tutorial02>. abgerufen am 13. Oktober 2020.
- [Foud] Django Software Foundation. *Writing your first Django app, part 3*. <https://docs.djangoproject.com/en/3.1/intro/tutorial03>. abgerufen am 29. Oktober 2020.
- [Geo17] Nigel George. *Mastering Django: Core. The Complete Guide to Django 1.8 LTS*. Leanpub book, 2017. ISBN: 978-0-9946168-0-7.
- [Hal19] Gareth Halfacree. *The Official Raspberry Pi. How to use your new computer. Beginner's Guide*. aspberry Pi Press; 3rd Revised edition, 2019. ISBN: 1912047586.
- [ION] IONOS. *Komponentendiagramm – Effiziente Systemmodellierung mit Softwaremodulen*. <https://www.ionos.de/digitalguide/websites/webentwicklung/komponentendiagramm>. abgerufen am 18. Oktober 2020.
- [JI] Ralph Jacobi and Josh Wyatt. Texas Instruments. *MIFAREDESFireEV1 AESAuthenticationWithTRF7970A*. <https://www.ti.com/lit/an/sloa213/sloa213.pdf>. abgerufen am 07. Oktober 2020.
- [lib] libusb. *libusb-1.0 API Reference. A cross-platform user library to access USB devices*. <http://libusb.sourceforge.net/api-1.0>. abgerufen am 05. Oktober 2020.
- [Ltd] Advanced Card Systems Ltd. *ACR122U USB NFC Reader*. <https://www.acs.com.hk/en/products/3/acr122u-usb-nfc-reader/>. abgerufen am 23. Oktober 2020.
- [MD] Space Telescope Science Institute. Michael Droettboom. *Understanding JSON Schema*. <https://json-schema.org/understanding-json-schema/reference/array.html>. abgerufen am 10. Oktober 2020.
- [ML] a Salesforce company MuleSoft LLC. *What is a REST API*. <https://www.mulesoft.com/resources/api/what-is-rest-api-design>. abgerufen am 5. November 2020.
- [NXP] NXP. *MIFARE® contactless tag IC family overview*. <https://www.nxp.com/docs/en/product-selector-guide/MIFARE-ICs-linecard.pdf>. abgerufen am 06. Oktober 2020.
- [OG] One Blog: it is all in the name. One Guy. *ACR122U NFC USB READER ON A RASPBERRY PI*. <https://oneguyoneblog.com/2015/09/16/acr122u-nfc-usb-reader-raspberry-pi/>. abgerufen am 05. Oktober 2020.
- [Pod] Mikhail Podgurskiy. *Django friendly finite state machine support*. <https://github.com/viewflow/django-fsm>. abgerufen am 28. Oktober 2020.

- [Pri] Author: SV 2d/3d Printing. *Raspberry Pi programming basics*. <https://api-2d3d-cad.com/microcomputer-programming-basics/>. abgerufen am 03. Oktober 2020.
- [RF] J. Gettys J. Mogul H. Frystyk T. Berners-Lee R. Fielding UC Irvine. *Hypertext Transfer Protocol – HTTP/1.1*. <https://tools.ietf.org/html/rfc1945>. abgerufen am 30. Oktober 2020.
- [RR03] Leon Shklar und Richard Rosen. *Web Application Architecture. Principles, protocols and practices*. John Wiley & Sons Ltd, 2003. ISBN: 0-471-48656-6.
- [SCT] Reiner SCT. *cyberJack® RFID basis. Chipkartenleser für den neuen Personalausweis*. <https://shop.reiner-sct.com/chipkartenleser-fuer-die-sicherheitsklasse-3/cyberjack-rfid-basis>. abgerufen am 04. Oktober 2020.
- [SSG] Sparx Systems Ltd und SparxSystems Software GmbH. *Anwendungsfalldiagramm (Use Case Diagram)*. <https://www.sparxsystems.de/ressourcen/literatur/leseprobe-zu-projektentwicklung-mit-uml-und-enterprise-architect/anwendungsfalldiagramm-use-case-diagram/>. abgerufen am 18. Oktober 2020.
- [TBa] Beuth Hochschule für Technik Berlin. *acaLab: Aktivitäten und Abschlussarbeiten*. <https://labor.beuth-hochschule.de/pse/acalab-aktivitaeten-und-abschlussarbeiten>. abgerufen am 21. August 2020.
- [TBb] Beuth Hochschule für Technik Berlin. *Campus-Card*. <https://www.beuth-hochschule.de/campus-card>. abgerufen am 08. Oktober 2020.
- [TBc] PSE-Labor der Beuth Hochschule für Technik Berlin. *acaLoan-Raspi*. <https://labor.beuth-hochschule.de/pse/acalab-aktivitaeten-und-abschlussarbeiten/acaloan-raspi/>. abgerufen am 15. Oktober 2020.
- [TBL] H. Frystyk T. Berners-Lee R. Fielding. *Hypertext Transfer Protocol – HTTP/1.0*. <https://tools.ietf.org/html/rfc1945>. abgerufen am 30. Oktober 2020.
- [Tec] Syрма Technology. *How Radio-Frequency Identification Systems Work*. <https://www.syrmatech.com/rfid-system/>. abgerufen am 09. Oktober 2020.
- [Too] NFC Tools. *ISO/IEC 14443 Type A*. <http://nfc-tools.org/index.php/ISO14443A>. abgerufen am 07. Oktober 2020.