

Aperta - Das smarte Garagentor

DIPLOMARBEIT

verfasst im Rahmen der

Reife- und Diplomprüfung

an der

**Höheren Abteilung für Informationstechnologie mit
Ausbildungsschwerpunkt Medientechnik**

Eingereicht von:

Benjamin Golic

David Hauser

Simon Koll

Betreuer:

Prof. Christian Aberger

Leonding, April 2022

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Die Arbeit wurde bisher in gleicher oder ähnlicher Weise keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Die vorliegende Diplomarbeit ist mit dem elektronisch übermittelten Textdokument identisch.

Leonding, April 2022

Benjamin Golic & David Hauser & Simon Koll

Abstract

APERTA is a garage door system that was developed by Benjamin Golic, David Hauser and Simon Koll as part of their diploma thesis. The system can be retrofitted to electric garage doors regardless of the manufacturer and enables entry into the world of the Internet of Things, the IOT. Depending on the configuration, the system consists of up to 3 components, each of which represents an access option to the garage. These are a classic numpad, an RFID reader, and a camera, which is used for license plate recognition. In the future, these can be expanded to include other or further authentication options. A web dashboard is used to manage the combinations of the numpad, NFC card details or license plates. These can be provided with a time validity range. There is also an online store where the product can be configured and spare parts or expansion modules can be purchased. The dashboard, which was implemented using Angular, communicates with the Node-JS server as the backend. It acts as the core and passes the information to the MongoDB database and is available for requests from the Raspberry Pi, which is responsible for the hardware.



Abbildung 1: Rendered Prototyp

Zusammenfassung

APERTA ist ein Garagentorsystem, welches von Benjamin Golic, David Hauser und Simon Koll im Rahmen der Diplomarbeit entwickelt wurde. Das System ist herstellernabhängig bei elektrischen Garagentoren nachrüstbar und ermöglicht den Einstieg in die Welt des Internet der Dinge, dem IOT. Das System besteht, je nach Konfiguration, aus bis zu 3 Komponenten, welche je eine Zutrittsmöglichkeit zur Garage darstellen. Es handelt sich hierbei um ein klassisches Nummernfeld, einem RFID-Lesegerät, und einer Kamera, welche für eine Kennzeichenerkennung genutzt wird. Zukünftig können diese noch um andere oder weitere Möglichkeiten der Authentifizierung erweitert werden. Über ein Web-Dashboard werden die Nummernfeldkombinationen, NFC-Kartendetails oder Kennzeichen verwaltet. Diese können mit einem zeitlichen Gültigkeitsbereich versehen werden. Darüber hinaus gibt es einen Onlineshop, in welchem das Produkt konfiguriert werden kann sowie Ersatzteile oder Erweiterungsmodule erworben werden können. Das Dashboard, welches mithilfe von Angular umgesetzt wurde, kommuniziert mit dem Node-JS Server als Backend.

Dieser fungiert als Herzstück und reicht die Informationen an die MongoDB-Datenbank weiter und steht für Anfragen des Raspberry Pi, welcher für die Hardware zuständig ist, zur Verfügung.



Abbildung 2: Gerenderter Prototyp

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemsituation [SK]	1
1.2	Ziele[SK]	1
2	Glossar	3
3	Umfeldanalyse	4
4	Technologien	5
4.1	Auswahl der Technologie - Client	5
4.2	Auswahl der Technologie - Datenbank	5
4.3	Auswahl der Technologie - Kennzeichenerkennung[SK]	5
4.4	Auswahl der Technologie - Backend[SK]	6
4.5	Auswahl der Technologie - Hardware[SK]	8
5	Lösungsansätze	10
5.1	Profil Management	10
5.2	Webshop	10
5.3	Automatic Number Plate Recognition (ANPR)	10
5.4	Backend	10
6	Systemarchitektur	11
6.1	Frontend (Angular-Applikation)	11
6.2	Frontend (React-Applikation)	11
6.3	Backend (NodeJS-Server)	11
6.4	Kennzeichenerkennung	11
7	Umsetzung	12
7.1	Deployment	12

8 Persönliche Ziele	14
8.1 Projektverlauf	14
8.2 Erkenntnisse von Benjamin Golic	14
8.3 Erkenntnisse von David Hauser	14
8.4 Erkenntnisse von Simon Koll	14
9 Zusammenfassung	15
Literaturverzeichnis	VI
Abbildungsverzeichnis	VII
Tabellenverzeichnis	VIII
Quellcodeverzeichnis	IX
Anhang	X

1 Einleitung

1.1 Problemsituation [SK]

Autos sind aus dieser Welt nicht mehr wegzudenken. Alleine in Österreich sind mehr als 5 Millionen Personenkraftwagen zugelassen. Dieser Bestand wuchs nach Angaben der Statistik Austria seit mehr als 15 Jahren kontinuierlich an. Um die Sicherheit der Insassen gewährleisten zu können, wird empfohlen, das Fahrzeug in einer Garage oder einem überdachten Gebiet abzustellen. Dort sind Umwelteinflüsse wie Hagel keine große Gefahr mehr. Mit der immer weiter voranschreitenden Vernetzung der nahen Umwelt, wie Haustüren mit Fingerabdruck-Zugangskontrolle oder mit dem Smartphone bedienbaren Jalousien, kommt mit APERTA das IOT in die Garage. APERTA ist ein Komplettsystem, welches bei Garagen mit elektrischem Tor nachgerüstet werden kann. Es besteht die Möglichkeit, mithilfe eines Nummernfeldes oder einer NFC-Karte das Garagentor wie gewohnt zu öffnen. Was APERTA aber auszeichnet ist eine integrierte Kennzeichenerkennung, bei der man keinen Handsender oder ähnliches mehr benötigt. Das Kennzeichen wird über das Web-Dashboard eingegeben, und das Tor kann dann bei Annäherung an das Tor dieses erkennen und steuert den Toröffnungsmechanismus an.

1.2 Ziele[SK]

- *Herstellerunabhängigkeit:* APERTA soll bei jedem Garagentor mit bereits verbautem Motor nachrüstbar sein. So können mehr potentielle Käufer angesprochen werden.
- *Modularität:* Aufgrund der vielen Möglichkeiten kann APERTA für manche Käufer in der Vollausrüstung nicht geeignet sein. Das Produkt soll daher im Onlineshop konfiguriert werden können, sodass bestimmte Komponenten entfernt werden können. Diese sollen nachträglich eingebaut werden können.
- *Übersichtliche Verwaltung:* Um den Nutzer zu unterstützen, soll die Verwaltung von Nummernfeldkombinationen, NFC-Details und Kennzeichen einfach und

schnell funktionieren. Der Käufer soll mithilfe von Texteingaben neue Einträge hinzufügen können und diese mit einem Knopfdruck wieder entfernen können.

- *Intuitiver Bestellvorgang:* Um die Erfahrung für den Käufer von Anfang an gut zu gestalten, soll ein optisch ansprechender Onlineshop der erste Kontakt mit dem Produkt sein.

2 Glossar

3 Umfeldanalyse

Citing [1] properly.

Was ist eine Globally Unique Identifier (GUID)? Eine GUID kollidiert nicht gerne.

Kabellose Technologien sind in abgelegenen Gebieten wichtig [2].

4 Technologien

4.1 Auswahl der Technologie - Client

4.1.1 Unterschied Framework und Library

4.1.2 Technologie zur Entwicklung des Frontends

4.1.3 Angular

4.1.4 React

4.2 Auswahl der Technologie - Datenbank

4.2.1 MongoDB

4.2.2 Unterschied Relationale und nicht Relationale Datenbanken

4.3 Auswahl der Technologie -

Kennzeichenerkennung[SK]

Das Herz von APERTA ist die Kennzeichenerkennung. Dieses Alleinstellungsmerkmal separiert das Projekt von möglicher Konkurrenz. Um eine schnelle, und problemfreie Lösung zu liefern, versuchte das Team, eine für österreichische Kennzeichen optimierte Lösung zu implementieren. Da dies aufgrund fehlender Datensets für genannte Kennzeichen schwierig bis nicht realisierbar war, griff man auf eine API zurück, welche das aufgenommene Bild erhält und das Kennzeichen zurück liefert. Diese Abfrage wird mittels eines HTTP-Requests ausgeführt.

4.4 Auswahl der Technologie - Backend[SK]

4.4.1 Anforderungen an das Backend

Für das Backend kamen mehrere Technologien in Frage, wie unter anderem Java, JavaScript, Python, PHP, C#, und viele mehr. Um das Backend zu realisieren, muss die Technologie einige bestimmte Eigenschaften besitzen.[3]

- *Java:*

Die Vorteile von Java liegen in der Fehlerbehandlung, sowie in Bereichen wie Multithreading und Performanz. Die strikte Fehlerbehandlung führt dabei aber zum Verlust von Flexibilität und Kompaktheit des Codes.

- *JavaScript:*

Die Syntax von JavaScript ähnelt der von Java. Entwickelt als Scripting-Sprache für HTML, ist JavaScript einfach zu lernen und zu benutzen. Bei der Entwicklung von Websites kann JavaScript direkt in den Quellcode der HTML-Seite eingearbeitet werden. Aber auch im Backend-Bereich kann mit NodeJS in JavaScript entwickelt werden.

- *Python:*

Python ist eine der mit Abstand am leichtesten zu lesenden Programmiersprachen. Die flache Hierarchie ermöglicht ein einfaches Verständnis von Programmen und Codestücken. Weiters macht Python den Entwickler auf jeden Fehler aufmerksam, wenn dieser nicht ausdrücklich ignoriert werden soll. Jedoch ist Python manchmal langsamer in der Ausführung als die konkurrierenden Sprachen. Zusätzlich ist durch die Verwendung von Leerzeichen zur Einrückung ein häufiger Fehlergrund hinzugekommen.

- *PHP:*

Die PHP Syntax erinnert an eine Mischung aus C, Java und Perl. Das Ziel von PHP ist es, Entwickler schnell und einfach dynamisch generierte Webpages zu bauen. Die vermischte Syntax ist jedoch etwas chaotisch, darum ist es leicht sich in falschen Angewohnheiten zu verirren und Sicherheitslücken offen zu lassen.

Aufgrund vorhandener Vorkenntnisse standen für das Team 3 der oben genannten Technologien zur Auswahl:

- Java,
- JavaScript und
- Python

4.4.2 Verwendung von NodeJS

Von diesen konnte sich JavaScript durchsetzen. Die Gründe dafür waren:[4]

- NPM: Der NPM oder *Node Package Manager*, ist ein Paketmanager für JavaScript, welcher bei NodeJS standardmäßig mitgeliefert wird. Bei NPM werden wiederverwendbare Programmteilen veröffentlicht. Diese können mittels des NPM eingenen Command Line Interfaces installiert werden. Weiters bietet der NPM eine integrierte Versionsverwaltung der Pakete sowie eine Verwaltung der Abhängigkeiten. In diesem Projekt wurden beispielsweise die Module *express* und *mongodb* verwendet. *express* ist ein Framework, welches vor allem in NodeJS Projekten verwendet wird. Die Vorteile von Express sind unter anderem die dem Team bereits bekannte Programmiersprache JavaScript, die Unterstützung der Google V8 engine für bessere Performance, die Robustheit bei einer Vielzahl an HTTP-Anfragen, sowie die einfache Einbindung weiterer Module und Drittanbieterapplikationen. [5]

mongodb stellt dem Entwickler eine API zur Verfügung, welche die Nutzung einer MongoDB-Datenbank stark vereinfacht.

- Verwendung einer NoSQL Datenbank: Aufgrund des Formates, mit dem die Daten aus dem Frontend kommen, bat sich eine nicht relationale Datenbank für das Team an. Die dokumentenorientierte Datenbank MongoDB ist bekannt für ihre hohe Verfügbarkeit, sowie die gute Skalierbarkeit.[6]
- Behandlung von JSON: NodeJS zeichnet sich durch seine einfach Verwendung von JSON-Daten aus. Diese können ohne Parsing oder andere Konvertierungen verarbeitet und darauf zugegriffen werden. Dank NodeJS können JSON Objekte mittels REST-API Anfragen direkt für den Client bereitgestellt werden. Danke NodeJS kann eine Einfache Verbindung zwischen Frontend-Clients und dem Backend-Server geschaffen werden.

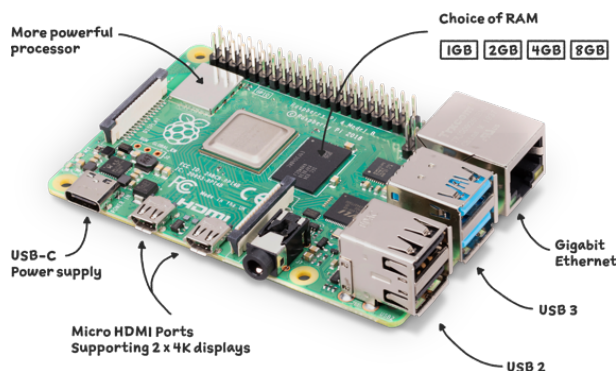
4.5 Auswahl der Technologie - Hardware[SK]

4.5.1 Anforderungen an die Hardware

Das Projekt sollte so vielseitig wie möglich, jedoch auch so kompakt wie möglich sein. Dazu musste auf kleine Komponenten gesetzt werden. Diese soll dennoch leistungsfähig genug sein, um jede der drei Zugangsmöglichkeiten parallel zu verwalten.

Raspberry Pi

Die Wahl des Herzstückes fiel auf einen Raspberry Pi. Der Raspberry Pi ist ein vollwertiger Computer, welcher etwas größer als eine Kreditkarte ist. Er besitzt alle bekannten Anschlüsse eines normalgroßen PCs, wie HDMI-Ausgänge für Monitore, USB-Ports für Peripherie wie Maus, Tastatur oder Webcams, sowie einen LAN-Port für eine kabelgebundene Netzwerkverbindung. Als Betriebssystem des Raspberry Pi wurde das vom Hersteller empfohlene Raspbian OS verwendet. Dieses bietet eine grafische Benutzeroberfläche, sowie die Möglichkeit den Raspberry auch ohne angeschlossenen Monitor betreiben zu können. [7] [8]



Auszeichnungsmerkmale des Raspberry Pi sind unter anderem die geringen Anschaffungskosten von ab 35 US-\$, sowie seine leistungstarken Komponenten.

Der Raspberry Pi ist einer der am weitesten verbreiteten Ein-Platinen-Computer der Welt. Trotz der im Verhältnis zu größeren Systemen schwache Leistung im Jahr 2020 mehr als 7 Millionen mal verkauft worden. Daraus ergibt sich ein Marktanteil von allen PCs von 2.69%. [10] Für ein ausgewogenes Verhältnis zwischen Kompaktheit und Leistung griff man auf einen Raspberry Pi 4 Model B in der Ausführung mit 4GB Arbeitsspeicher zurück. Weiters waren die Anschaffungskosten von etwa 100\$ ein weiterer Grund für die Auswahl. **Bestandteile des Raspberry Pi**

Tabelle 1: Übersicht der Komponenten des Raspberry Pi [9]

Komponente	Spezifikation	Besonderheiten
Prozessor	Broadcom BCM2711 - Quad Core Prozessor @ 1.5GHz	ARM Architektur 64-Bit SoC
RAM	1GB, 2GB, 4GB oder 8GB LPDDR4 SDRAM	Taktfrequenz von 3200MHz
USB	2 USB 3.0 Ports 2 USB 2.0 Ports	
GPIO	40 Pin Header	Abwärtskompatibel mit Vorgängermodellen
Display	2 micro-HDMI Ports	jeweils bis zu 4k60 möglich
Speicher	Micro-SD Kartenslot	Speicherplatz für Betriebssystem und Daten
Strom	5V Eingang über USB-C Port 5V Ausgang über GPIO-Header	Anforderung an Stromquelle: mindestens 3A

NFC-Leser**Numpad****Kamera****Display**

5 Lösungsansätze

5.1 Profil Management

5.2 Webshop

5.3 Automatic Number Plate Recognition (ANPR)

5.4 Backend

6 Systemarchitektur

6.1 Frontend (Angular-Applikation)

6.2 Frontend (React-Applikation)

6.3 Backend (NodeJS-Server)

6.4 Kennzeichenerkennung

7 Umsetzung

7.1 Deployment

Siehe tolle Daten in Tab. 2.

Siehe und staune in Abb. 3. Dann betrachte den Code in Listing 1.

Listing 1: Some code

```
1  # Program to find the sum of all numbers stored in a list (the not-Pythonic-way)
2
3  # List of numbers
4  numbers = [6, 5, 3, 8, 4, 2, 5, 4, 11]
5
6  # variable to store the sum
7  sum = 0
8
9  # iterate over the list
10 for val in numbers:
11     sum = sum+val
12
13 print("The sum is", sum)
```

	Regular Customers	Random Customers
Age	20-40	>60
Education	university	high school

Tabelle 2: Ein paar tabellarische Daten

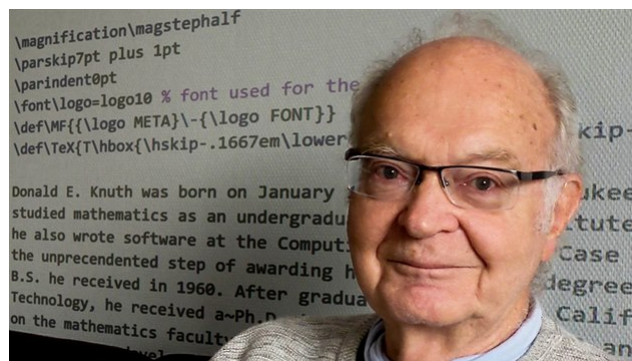


Abbildung 3: Don Knuth – CS Allfather

8 Persönliche Ziele

8.1 Projektverlauf

8.2 Erkenntnisse von Benjamin Golic

8.3 Erkenntnisse von David Hauser

8.4 Erkenntnisse von Simon Koll

Das Projekt nahm immer mehr Substanz an, je weiter die Entwicklung fortschritt. Es kamen neue Ideen hinzu, die Anfangs noch gar nicht im Raum standen. Die Erkenntnisse und Inhalte des ITP-, INSY- sowie SEW-Unterrichts haben mich in der

9 Zusammenfassung

Aufzählungen:

- Itemize Level 1
 - Itemize Level 2
 - Itemize Level 3 (vermeiden)
- 1. Enumerate Level 1
 - a. Enumerate Level 2
 - i. Enumerate Level 3 (vermeiden)

Desc Level 1

Desc Level 2 (vermeiden)

Desc Level 3 (vermeiden)

Literaturverzeichnis

- [1] P. Rechenberg, G. Pomberger *et al.*, *Informatik Handbuch*, 4. Aufl. München – Wien: Hanser Verlag, 2006.
- [2] Association for Progressive Communications, „Wireless technology is irreplaceable for providing access in remote and scarcely populated regions,” 2006, letzter Zugriff am 23.05.2021. Online verfügbar: <http://www.apc.org/en/news/strategic/world/wireless-technology-irreplaceable-providing-access>
- [3] Learneroo, Zuletzt besucht am 25.03.2022. Online verfügbar: <https://www.learneroo.com/modules/9/nodes/620>
- [4] J. D. Tomislav Capan, Zuletzt besucht am 25.03.2022. Online verfügbar: <https://www.toptal.com/nodejs/why-the-hell-would-i-use-node-js>
- [5] S. CM, Zuletzt besucht am 25.03.2022. Online verfügbar: <https://www.techomoro.com/what-are-the-benefits-of-using-express-js-for-backend-development/>
- [6] MongoDB, Zuletzt besucht am 25.03.2022. Online verfügbar: <https://www.mongodb.com/why-use-mongodb>
- [7] R. P. Foundation, Zuletzt besucht am 25.03.2022. Online verfügbar: <https://www.raspberrypi.org/help/what-%20is-a-raspberry-pi/>
- [8] —, Zuletzt besucht am 26.03.2022. Online verfügbar: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- [9] —, Zuletzt besucht am 26.03.2022. Online verfügbar: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/>
- [10] I. ARC, Zuletzt besucht am 26.03.2022. Online verfügbar: <https://www.industryarc.com/Report/19454/industrial-raspberry-pi-market.html>

Abbildungsverzeichnis

1	Rendered Prototyp	I
2	Gerenderter Prototyp	II
3	Don Knuth – CS Allfather	13

Tabellenverzeichnis

1	Übersicht der Komponenten des Raspberry Pi [9]	9
2	Ein paar tabellarische Daten	12

Quellcodeverzeichnis

1	Some code	12
---	---------------------	----

Anhang