

# Projektbreicht Smart Music Player

Anton Bracke  
Jan Eberlein  
Tom Calvin Haak  
Julian Hahn  
Nick Loewecke

4. Dezember 2020

## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>2</b>
1.1	Unternehmen . . . . .	2
1.2	Projekt . . . . .	2
1.2.1	Minimal Requirements . . . . .	2
1.2.2	Stretch Goals . . . . .	3
<b>2</b>	<b>Durchführung</b>	<b>3</b>
2.0.1	Technologien und Hilfsmittel . . . . .	3
2.0.2	Projekt Management . . . . .	3
2.0.3	Deployment Cycle . . . . .	3
2.0.4	Probleme während der Durchführung . . . . .	3
<b>3</b>	<b>Design Mockups</b>	<b>3</b>
<b>4</b>	<b>Machbarkeitsstudie</b>	<b>3</b>
4.1	NFC Tag . . . . .	3
4.1.1	lesen . . . . .	3
4.1.2	schreiben . . . . .	3
4.2	Raspberri Pi . . . . .	4
4.2.1	Docker Integration . . . . .	4
4.2.2	Öffentlich zugängliches Web Interface . . . . .	4
4.2.3	URL für UI festlegen . . . . .	4
4.3	User Interface . . . . .	4
4.3.1	Zugriff auf NFC Reader im Cloud Szenario . . . . .	4
4.3.2	Login via Spotify, Youtube, etc. . . . .	4
4.3.3	Gleichen Nutzer bei verschiedenen Loginvarianten wieder- erkennen . . . . .	4
4.3.4	Musik Artwork laden . . . . .	5
4.3.5	Eigene Bilder hochladen . . . . .	5
4.3.6	Spotify connect Lautsprecher auswählen . . . . .	5
4.3.7	Spotify connect Lautsprecher speichern . . . . .	5
4.3.8	In Cloud UI die jeweilige Box auswählen . . . . .	5

4.3.9	Boxdaten über Cloud UI ändern . . . . .	5
4.3.10	Unterstützung von Youtube . . . . .	5
4.3.11	Unterstützung von Apple Music . . . . .	5
4.3.12	Unterstützung von Deezer . . . . .	5
4.3.13	Unterstützung von Tidal . . . . .	5
4.3.14	Unterstützung von eigener Musik (USB Stick, MicroSD Karte, Cloud) . . . . .	5
4.4	Sonstiges . . . . .	5
4.4.1	3D Print version . . . . .	5
4.4.2	Sound Wiedergabe auf der Box selbst . . . . .	5
4.4.3	Box unter 30€ Kosten . . . . .	5
<b>5</b>	<b>Code Walkthrough</b>	<b>5</b>
<b>6</b>	<b>Testing</b>	<b>5</b>
<b>7</b>	<b>Technische Diagramme</b>	<b>5</b>

# 1 Einleitung

## 1.1 Unternehmen

was macht macio aus

## 1.2 Projekt

Im Rahmen des Bachelor-Moduls Projekt Informatik möchte Macio deren Portfolio im IoT-Bereich erweitern sowie den Empfangsraum in Kiel verschönern. Smarte Spielzeuge mit NFC-Funktion gibt es im Kommerzbereich viele, daher soll dieses Projekt eine Open-Source-Alternative dafür schaffen. Genauer handelt es sich um einen Smarten Music Player, der NFC sowie Spotify Connect unterstützt. Dieser Music Player hat einen Microcontroller mit NFC-Reader. Auf die Box können durch kleine Figuren in Form von z.B. Figuren mit dem integrierten NFC-Tag Events getriggert werden. Bei einem Event wird vom Microcontroller über jeweilige Musikdienst-API auf einem verbundenen Gerät Musik abgespielt. Die NFC-Tags und zugehörigen Events sollen über ein Web-Frontend konfiguriert werden können. Dieses Web-Frontend wird von der Box ausgeliefert und soll für primär eine Smartphone bedienung gestaltet werden. Da es sich um ein Open Source Projekt handelt mit entsprechender Lizenz, muss auch eine aussagekräftige, öffentliche Dokumentation verfasst werden. Macio stellt die benötigte Hardware und ist für Technische Fragen unterstützend verfügbar.

### 1.2.1 Minimal Requirements

1. NFC-Tags lesen, schreiben und entschlüsseln
2. Mit Spotify Connect verbinden und arbeiten
3. Responsive UI konzeptionieren und umsetzen
4. Aussagekräftige Dokumentation mit Benutzerhandbuch

### 1.2.2 Stretch Goals

1. Sound Wiedergabe auf der Box selbst
2. Unterstützung anderer Musikdienste / Plugin-Subsystem
3. 3D-Modellierung und Print einer passenden Box
4. Cloud-Anbindung der Box, Auslieferung des UI aus der Cloud

## 2 Durchführung

### 2.0.1 Technologien und Hilfsmittel

Vue, vscode, devops krams, etc Entwickelt wird mit Visual Studio Code, da dort einfach ein Subsystem mit Linux genutzt werden kann <sup>1</sup>.

### 2.0.2 Projekt Management

ticket pool in github, alles zentral

### 2.0.3 Deployment Cycle

ziehe ticket > assigne dir selbst > draft PR > wenn fertig, setze ündraft» assigne 2 reviewer > merge mas

### 2.0.4 Probleme während der Durchführung

zb

## 3 Design Mockups

setze pdfs ein

## 4 Machbarkeitsstudie

### 4.1 NFC Tag

#### 4.1.1 lesen

Um mit NFC Tags arbeiten zu können, müssen diese auch auf entschlüsselt bzw. gelesen werden können. Hierfür ist ein Hardware NFC-Reader notwendig. Dieser kann über USB mit dem Chip der Box verbunden werden, um von dieser ausgelesen zu werden. welche library wird genutzt?

#### 4.1.2 schreiben

Einen NFC-Tag zu beschreiben ist davon abhängig, ob der NFC-Tag über einen eigenen Speicher verfügt. Ist dieser vorhanden, können dort z.b. Kontaktdaten hinterlegt werden. Werden diese dann von einem Smartphone gelesen, öffnet sich die Kontakte-App und der auf dem NFC-Tag gespeicherte Kontakt kann abgespeichert werden. Möglich, kann die gleiche library wie zum lesen genutzt werden?

---

<sup>1</sup><https://code.visualstudio.com/docs/remote/wsl>

## 4.2 Raspberri Pi

### 4.2.1 Docker Integration

möglich <sup>2</sup>

### 4.2.2 Öffentlich zugängliches Web Interface

Um den Web Server vom Pi von außerhalb des Netzwerkes erreichen zu können, muss innerhalb des Routers der Port 80 **port forwarded** werden. Dann kann der Pi unter dessen IP xxx.xxx.xxx.xxx:80 erreicht werden. Wenn die IP Adresse sich nicht ändern soll, kann auch eine statische IP festgelegt werden <sup>3</sup>. Dazu könnte dann eine Domain gekauft werden, die dann z.b. mit Cloudflare auf diese statische IP weiterleitet. Alternativ ginge es auch ohne Port-Forwarding mit nginx und ngrok <sup>4</sup>.

### 4.2.3 URL für UI festlegen

todo

## 4.3 User Interface

### 4.3.1 Zugriff auf NFC Reader im Cloud Szenario

Tendenziell sollte es nicht notwendig sein, dass der Raspberri Pi, der das Web-UI hosted, auf den jeweiligen NFC-Reader der Box zugreifen muss. Sollte das doch der Fall sein, könnte der Chip der Box ein API Call an den Raspberri Pi mit den NFC-Tag Daten machen, um das jeweilige hinterlegte Event zu triggern.

### 4.3.2 Login via Spotify, Youtube, etc.

todo: ask julian

### 4.3.3 Gleichen Nutzer bei verschiedenen Loginvarianten wiedererkennen

Um gleiche Nutzer zu erkennen, müssten Merkmale angelegt werden, über diese gleiche Nutzer wiedererkennbar wären. Die E-Mail wäre hierbei das einzigartigste Merkmal. Über Scopes in der OAuth Anfrage kann diese auch mitgeliefert werden. Das setzt aber voraus, dass Nutzer immer die gleiche E-Mail nutzen, was nicht immer der Fall ist. Man könnte dem (bereits eingeloggten) User die Möglichkeit geben, weitere Accounts zu diesem hinzuzufügen, sodass diese Accounts den gleichen Nutzer einloggen. Bei diesem Ansatz sollten dann die jeweiligen verbundenen Accounts zu einer unique User-ID verbunden werden.

---

<sup>2</sup><https://phoenixnap.com/kb/docker-on-raspberry-pi>

<sup>3</sup><https://howchoo.com/pi/configure-static-ip-address-raspberry-pi>

<sup>4</sup><https://vatsalyagoel.com/setting-up-a-public-web-server-using-a-raspberry-pi-3/>

- 4.3.4 Musik Artwork laden
- 4.3.5 Eigene Bilder hochladen
- 4.3.6 Spotify connect Lautsprecher auswählen
- 4.3.7 Spotify connect Lautsprecher speichern
- 4.3.8 In Cloud UI die jeweilige Box auswählen
- 4.3.9 Boxdaten über Cloud UI ändern
- 4.3.10 Unterstützung von Youtube
- 4.3.11 Unterstützung von Apple Music
- 4.3.12 Unterstützung von Deezer
- 4.3.13 Unterstützung von Tidal
- 4.3.14 Unterstützung von eigener Musik (USB Stick, MicroSD Karte, Cloud)
- 4.4 Sonstiges
  - 4.4.1 3D Print version
  - 4.4.2 Sound Wiedergabe auf der Box selbst
  - 4.4.3 Box unter 30€ Kosten

## 5 Code Walkthrough

vielleicht interessante Code Snippets?

## 6 Testing

wie haben wir getestet, haben wir getestet?

## 7 Technische Diagramme

ER Diagramme, UML, solcher krams