



# SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE SEPTEMBER 2021

## Konsolidierter Schlussbericht

---

Förderkennzeichen:	01IS22S01	Pierre Depaz und Tobias Schmidt GbR
	01IS22S02	Deeg Picker GbR
	01IS22S03	Lars Böttcher
	01IS22S04	Sergej Levich
	01IS22S05	Jonas Dreßler, Tobias Bernard GbR
	01IS22S06	Marta Maria Marczykowska-Górecka
	01IS22S07	Nicolas Kuhaupt
	01IS22S08	Ahrend Wagner GbR
	01IS22S09	Travis Brown
	01IS22S10	Peggy Sylopp
	01IS22S11	Heinemann & Woste GbR
	01IS22S12	Fereydun Khanide & Marc Weis GbR
	01IS22S13	Moritz Stückler & Fabian Schmidt GbR
	01IS22S14	Neugebauer-Ippen GbR
	01IS22S15	Katharina Brunner
	01IS22S16	Daniel Spaude und Ewa Flis GbR
	01IS22S17	Janis Estelmann
	01IS22S18	Blum, Ehrlich, Mielke, Stoffels & Witt GbR
	01IS22S19	Philipp Krause, Felix Salfelder, Benedikt Freisen GbR
	01IS22S20	Lorenz Sieben und Benjamin Altpeter GbR
	01IS22S21	Katharina Rasch
	01IS22S22	Sasse & Werner GbR
	01IS22S23	Lia Dürr
	01IS22S24	Klaus Scheuermann

---

Vorhabenbezeichnung: Software Sprint – 24 Einzelvorhaben

---

Laufzeit der Einzelvorhaben: 01.03.2022-31.08.2022

---

Die diesem Bericht zugrunde liegenden Vorhaben wurden mit Mitteln des Bundesministeriums für Bildung und Forschung unter den o.g. Förderkennzeichen gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den jeweils genannten Autorinnen und Autoren.



## Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE September 2021 Konsolidierter Schlussbericht	
4. Autor(en) [Vorname(n), Name(n)] Pierre Depaz Simon Deeg Lars Böttcher Sergej Levich Jonas Dreßler Marta Marczykowska-Górecka Nicolas Kuhaupt Eileen Wagner Travis Brown Peggy Sylopp Daniel Woste Fereydun Khanide Moritz Stückler Mareike Ippen Katharina Brunner Daniel Spaude Janis Estelmann Sarah Stoffels Philipp Krause Lorenz Ferdinand Sieben Katharina Rasch Jannik Werner Lia Dürr Klaus Scheuermann	5. Abschlussdatum des Vorhabens 31.08.2022  6. Veröffentlichungsdatum 23.11.2022  7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) 01IS22S01      Pierre Depaz und Tobias Schmidt GbR 01IS22S02      Deeg Picker GbR 01IS22S03      Lars Böttcher 01IS22S04      Sergej Levich 01IS22S05      Jonas Dreßler, Tobias Bernard GbR 01IS22S06      Marta Maria Marczykowska-Górecka 01IS22S07      Nicolas Kuhaupt 01IS22S08      Ahrend Wagner GbR 01IS22S09      Travis Brown 01IS22S10      Peggy Sylopp 01IS22S11      Heinemann & Woste GbR 01IS22S12      Fereydun Khanide & Marc Weis GbR 01IS22S13      Moritz Stückler & Fabian Schmidt GbR 01IS22S14      Neugebauer-Ippen GbR 01IS22S15      Katharina Brunner 01IS22S16      Daniel Spaude und Ewa Flis GbR 01IS22S17      Janis Estelmann 01IS22S18      Blum, Ehrlich, Mielke, Stoffels & Witt GbR 01IS22S19      Philipp Krause, Felix Salfelder, Benedikt Freisen GbR 01IS22S20      Lorenz Sieben und Benjamin Altpeter GbR 01IS22S21      Katharina Rasch 01IS22S22      Sasse & Werner GbR 01IS22S23      Lia Dürr 01IS22S24      Klaus Scheuermann	9. Ber. Nr. Durchführende Institution  10. Förderkennzeichen 01IS22S01-01IS22S24  11. Seitenzahl 127
12. Fördernde Institution (Name, Adresse)	13. Literaturangaben

Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	14. Tabellen
	15. Abbildungen
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum) Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) DLR Projektträger Berlin	
18. Kurzfassung Der Schlussbericht umfasst die Einzelschlussberichte der elften Auswahlrunde der Fördermaßnahme Software Sprint (Einreichungsdatum Skizze: 30.09.2021). Die Auswahl der Vorhaben erfolgte nach externer Begutachtung unter Abstimmung mit dem BMBF.	
19. Schlagwörter Software Sprint, Prototype Fund	
20. Verlag	21. Preis

## Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report
3. title SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE September 2021 Konsolidierter Schlussbericht	
4. author(s) (family name, first name(s)) Pierre Depaz Simon Deeg Lars Böttcher Sergej Levich Jonas Dreßler Marta Marczykowska-Górecka Nicolas Kuhaupt Eileen Wagner Travis Brown Peggy Sylopp Daniel Woste Fereydun Khanide Moritz Stückler Mareike Ippen Katharina Brunner Daniel Spaude Janis Estelmann Sarah Stoffels Philipp Krause Lorenz Ferdinand Sieben Katharina Rasch Jannik Werner Lia Dürr Klaus Scheuermann	5. end of project 31.08.2022
	6. publication date 23.11.2022
	7. form of publication
8. performing organization(s) (name, address) 01IS22S01 Pierre Depaz und Tobias Schmidt GbR 01IS22S02 Deeg Picker GbR 01IS22S03 Lars Böttcher 01IS22S04 Sergej Levich 01IS22S05 Jonas Dreßler, Tobias Bernard GbR 01IS22S06 Marta Maria Marczykowska-Górecka 01IS22S07 Nicolas Kuhaupt 01IS22S08 Ahrend Wagner GbR 01IS22S09 Travis Brown 01IS22S10 Peggy Sylopp 01IS22S11 Heinemann & Woste GbR 01IS22S12 Fereydun Khanide & Marc Weis GbR 01IS22S13 Moritz Stückler & Fabian Schmidt GbR 01IS22S14 Neugebauer-Ippen GbR 01IS22S15 Katharina Brunner 01IS22S16 Daniel Spaude und Ewa Flis GbR 01IS22S17 Janis Estelmann 01IS22S18 Blum, Ehrlich, Mielke, Stoffels & Witt GbR 01IS22S19 Philipp Krause, Felix Salfelder, Benedikt Freisen GbR 01IS22S20 Lorenz Sieben und Benjamin Altpeter GbR 01IS22S21 Katharina Rasch 01IS22S22 Sasse & Werner GbR 01IS22S23 Lia Dürr 01IS22S24 Klaus Scheuermann	9. originator's report no.
	10. reference no. 01IS22S01-01IS22S24
	11. no. of pages 127
12. sponsoring agency (name, address)	13. no. of references

Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	14. no. of tables
	15. no. of figures
16. supplementary notes	
17. presented at (title, place, date) Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) DLR Projektträger Berlin	
18. abstract The final report includes the individual final reports for the 11th call of participants within the public funding activity Software Sprint. The selection of the projects (call No. 11 deadline: 30.09.2021) took place after external evaluation under coordination with the BMBF.	
19. keywords Software Sprint, Prototype Fund	
20. publisher	21. price

# Richtlinie zum „Software-Sprint“

## CommonSyllabi - Eine Plattform zum Austausch von Lehrplänen

---

### *Schlussbericht*

Zuwendungsempfänger:

Pierre Depaz und Tobias Schmidt GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS22S01** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

*Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?*

Wir sind davon überzeugt, dass offene Weitergabe und offener Zugang den Fortschritt bringen. Derzeit sind die von Lehrkräften entwickelten Lehrpläne zu oft hinter den Mauern proprietärer Software und institutioneller Konten eingeschlossen. Obwohl die Lehrkräfte in der Regel die Rechte am geistigen Eigentum des von ihnen entwickelten Kursmaterials besitzen, gibt es nur wenige Tools, die speziell den offenen Austausch von Lehrplänen zwischen Lehrkräften unterstützen.

Zunächst möchten wir das Problem ansprechen, dass die derzeitigen LMS-Ökosysteme die seit langem bestehende Kultur des offenen Austauschs im akademischen Bereich behindern, insbesondere wenn es um Lehrpläne geht. Im Sinne der Open-Science-Bewegung kommt die freie Verbreitung von akademischen Materialien allen zugute, das gilt für Forschungsmaterialien ebenso wie für pädagogische Materialien.

Zweitens verhindern isolierte Ressourcen, dass Pädagogen einen Überblick über die Arbeit ihrer Kollegen in verschiedenen Bereichen erhalten. Da viele Pädagogen darauf zurückgreifen, Kursmaterialien auf persönlichen Websites zu veröffentlichen, ist das Auffinden und Konsultieren von Lehrplänen, die von Gleichgesinnten entwickelt wurden, eine mühsame Aufgabe, bei der man sich durch verstreute, nicht indizierte Ressourcen bewegen muss.

Schließlich gefährdet die Verzögerung der Einführung offener Formate auch die Langlebigkeit des Wissens, da sie die Schaffung offen zugänglicher Repositorien und/oder Archive behindert.

Hierfür gab es drei wesentliche Meilensteine:

- Die Entwicklung einer Parsing-Bibliothek für Common-Cartridge-Dateien
- Die Entwicklung einer Webschnittstelle für die Parsing-Bibliothek für CC-Dateien
- Die Entwicklung eines Archivs von Lehrplänen mit CC-Dateien.

## **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

*Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?*

Bildung ist ein entscheidender Bestandteil der Zivilgesellschaft und geht als solcher jeden auf einer bestimmten Ebene an. Die Verringerung des Aufwands für das Auffinden und die Verbreitung von Hochschulmaterialien hat zahlreiche Vorteile für angehende Lehrkräfte, für Forscher, die den Zustand und die Entwicklung der Hochschulbildung untersuchen wollen, und schließlich für die Studenten, die ein besseres Verständnis für die an verschiedenen Universitäten angebotenen Kurse bekommen können.

In diesem Sinne hoffen wir, mehr Wissen über die Struktur und die Möglichkeiten eines Lehrplans zu vermitteln und einen horizontalen Ansatz für die Zusammenstellung von Lehrplänen an verschiedenen Universitäten statt eines vertikalen Ansatzes über eine einzelne Universität vorzuschlagen.

## **Ausführliche Darstellung der Ergebnisse**

*Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?*



Wir haben das Ökosystem von Lernmanagementsystemen, wie Moodle oder Isis, weiter erforscht. Wir konzentrierten uns auf ein bestimmtes interoperables Format, IMS Common Cartridge. Da es keine nachhaltigen und effizienten Open-Source-Softwarelösungen zum Parsen dieses CC-Formats gab, haben wir eine Go-Bibliothek und ein Web-Frontend für erfahrene und nicht erfahrene Benutzer entwickelt.

Unsere weiteren Nachforschungen über das Ökosystem der LMS und wie sie sich aus der Sicht ihrer Nutzer unterscheiden, waren sehr hilfreich. Die Überwachung durch die Open Knowledge Foundation hatte, abgesehen von einigen nützlichen Ressourcen zu unserem Thema, keinen wesentlichen Einfluss auf die Ausrichtung des Projekts.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

*Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?*

*Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?*

Unsere Hauptzielgruppe sind Pädagogen, die Lehrplanmaterialien offen austauschen/veröffentlichen/archivieren und von anderen entwickelte Lehrpläne über das CC-Format durchsuchen möchten. Wir planen, eine internationale Nutzerbasis anzusprechen.

In der Anfangsphase werden wir uns auf Mundpropaganda in unseren persönlichen Netzwerken (z. B. Filmuniversität Babelsberg-Konrad Wolf, TU Berlin, Sciences Po und Sorbonne Universität) sowie auf die organische Suche verlassen, um die Nutzerbasis zu vergrößern. In der Zukunft, wenn ein verlässlich stabiles und sicheres Produkt entwickelt wurde, werden wir möglicherweise gezieltere Outreach-Projekte in Betracht ziehen.

Während wir nicht erwarten, dass wir mit dem Open-Sourcing der Go-Bibliothek und des IMSCC-Parsers eine große Wirkung erzielen, gehen wir davon aus, dass das Open-Sourcing der Kursmaterialien und des Lehrplans selbst eine sehr interessante und bedeutende Wirkung haben kann. In diesem Zusammenhang wird unsere weitere Entwicklung darin bestehen, Veranstaltungen an Universitäten mit Lehrern zu organisieren, um eine Gemeinschaftsdynamik aufzubauen, einschließlich einiger der oben erwähnten Universitäten.

Als akademische Forscherin und Lehrerin hat mir die Arbeit definitiv bei meiner beruflichen Entwicklung geholfen. Auch als Programmierer konnte ich meine Fähigkeiten verbessern.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

*Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?*

Die wichtigste Erkenntnis, die wir gewonnen haben, war, dass es zwar ein Format gibt, das Interoperabilität ermöglichen soll, dass es aber praktisch nicht angenommen wird, und unsere Nutzerforschung hat gezeigt, dass wenig Interesse an der Verwendung dieses Formats besteht, da die Lehrer bereits ihre eigene Datei- und Ordnerorganisation haben.

Wir schlossen also diesen Teil des Projekts ab und nutzten dann unsere Erkenntnisse, um eine erweiterte Plattform zu entwickeln, die das ursprüngliche IMSCC-Projekt als Komponente enthält.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

*Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?*

<https://common-syllabi.org>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

*Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?*

Die Nachforschungen über die genaue Spezifikation des IMSCC-Formats haben mehr Zeit in Anspruch genommen als erwartet, was auf die wenig hilfreiche Öffentlichkeitsarbeit des IMS-Konsortiums zurückzuführen ist, aber wir haben uns darauf eingestellt, indem wir einen empirischen Ansatz gewählt haben, der die verschiedenen Varianten der Dateien, die wir finden und erstellen konnten, abdeckt, anstatt bei einer abstrakten Spezifikation zu bleiben.

Infolgedessen haben wir etwa 15 % mehr Stunden gearbeitet als ursprünglich geplant.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

*Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?*

Wir haben uns an einen UX/UI-Designer gewandt, als klar wurde, dass die visuelle Darstellung des IMSCC-Formats nicht einfach war. Der Designer war sehr hilfreich bei der Erstellung von Wireframes und Schnittstellendesigns, was die Entwicklungszeit erheblich verkürzte.

# Richtlinie zum „Software-Sprint“

## OPENJOYN-CIRCULAR - ein Softwaretool zur Wiederverwendung vorhandener Rohstoffe und Materialien beim Aufbau von parametrischen Holz Konstruktionen

---

### *Schlussbericht*

Zuwendungsempfänger:

Deeg Picker GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S02 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Der Großteil unserer Produkte besteht nicht aus wiederverwerteten Materialien und landet am Ende auf dem Müll, da keine Zweitnutzung oder modulare Weiternutzung möglich ist. So werden z.B. Möbel nicht so entworfen, dass sie nach ihrem Lebenszyklus in einen Produktkreislauf rückgeführt werden können.

Der daraus resultierenden Herausforderung wird sich unsere Gesellschaft in Zukunft vermehrt stellen. Im Sinne der Kreislaufwirtschaft (CIRCULAR Economy), wird deswegen das Thema der mehrfachen und modularen Weiternutzung kreislauffähiger Produkte und Ansätze zunehmend an Relevanz gewinnen. Es geht nicht mehr nur, um die Funktion und das Aussehen von Produkten, sondern um die Entwicklung ganzer Systeme aus Nutzung und Weiternutzung.

Um mit an Lösungen für diese Herausforderung zu arbeiten, planen wir mit OPENJOYN - CIRCULAR Software-Funktionen zu entwickeln, die es Bürger:innen ermöglichen vorhandene Rohstoffe in neue Konstruktionen modular zu integrieren, um Ressourcen wiederzuverwerten. Durch einen online Katalog mit Holzkonstruktionen wird dieser Prozess gezielt innoviert und befähigt Bürger:innen den Bastelstatus zu verlassen und sowohl technisch als auch ästhetisch hohen Ansprüchen gerecht zu werden. In diese Softwareentwicklung eingebettet, sind Entwicklungs-Workshops mit Bürger:innen geplant, in denen ein bedarfsorientierter Katalog mit 15 relevanten Entwürfen entwickelt wird. Das Software Tool und der damit entwickelte Katalog wird zum Ende der Förderzeit in die OPENJOYN Web-Anwendung integriert und so für alle privaten Anwender:innen kostenlos zugänglich und nutzbar gemacht.

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Der Theorie des Civic Tech folgend, sollen Bürger:innen befähigt werden, privat vorhandene Rohstoffe und Objekte, wie Holzlatten und Stein- oder Holzplatten in parametrische OPENJOYN-Konstruktionen zu integrieren und diese somit wiederzuverwerten. Auf diesem Weg wird der bisherige Ansatz ausgebaut und Anwender:innen Alternativen, zu dem gegenwärtig prägnanten Konsumverhalten und dessen lineare Rohstoffnutzung, geboten.

## Ausführliche Darstellung der Ergebnisse

Alle geplanten Arbeitspakete wurden erfolgreich bearbeitet und die Meilensteine erreicht. Die relevantesten Ergebnisse der Entwicklung werden im Folgenden kurz dargelegt.

### 1. Bedarfserhebung zum Thema zirkuläre Rohstoffnutzung im Privatbereich

Zu Beginn des Projektes wurden Bürger:innen aus Berlin befragt und Hypothesen überprüft.

#### Relevante Ergebnisse:

Es konnte eine große Bereitschaft identifiziert werden, vorhandene Rohstoffe in DIY-Projekte zu integrieren. Des Weiteren wurde offensichtlich, dass alte Materialien in privaten Haushalten vorhanden sind, die hierfür genutzt werden könnten. Die Form dieser Rohstoffe und Baumaterialien konnte überwiegend in die folgenden drei geometrischen Hauptgruppen eingeordnet werden:

- a. Plattenwerkstoffe (Holz-, Glas- und Steinplatten)
- b. Latten und Vierkantprofile (Holzlatten und Stahlrohre, jeglicher Abmessung)
- c. Kisten und Boxen

### 2. Entwicklung der Parametrisierungsstrategie

#### 2.1 Priorisierung des Inputs

Auf diesen Erkenntnissen aufbauend, wurde die Parametrisierung entwickelt. Ein niederkomplexer, aber wichtiger Teil hierbei war, die Inputpriorisierung für die Generierung der Entwürfe. Der wichtigste Input der Anwender:innen und somit der Ausgangspunkt der Entwurfsgenerierung, ist stets das vorhandene Baumaterial. Diese Strategie führt beispielsweise dazu, dass zu Beginn die Maße einer schon vorhandenen Platte abgefragt werden und ausgehend von diesem Input der Rest des Entwurfes generiert wird. So wird alleine durch den Ablauf der Interaktion provoziert vorhandene Rohstoffe und Baumaterialien mitzudenken und so in Entwürfe zu integrieren.

#### 2.2 Individuelle Lattenmaße

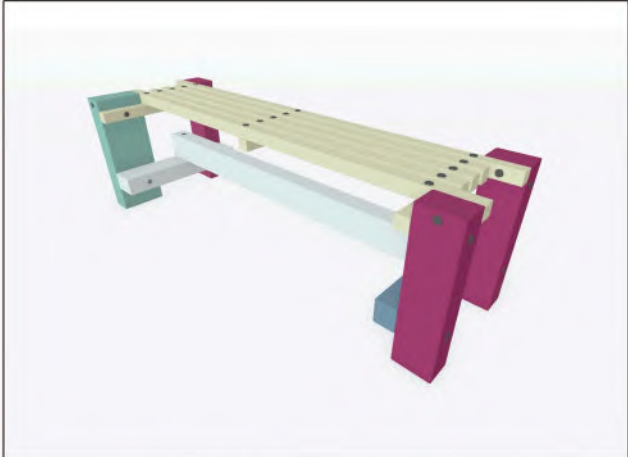
Des Weiteren wurden Funktionen entwickelt, Latten mit unterschiedlichen Seitenmaßen und Längen individuell (bzw. 31x45x2568mm) in Entwürfe zu integrieren, diese in Echtzeit digital „zuzuschneiden“ und in dem jeweiligen Entwurf zu platzieren. Bei diesem Funktionspaket wurde die entscheidende Aufgabe identifiziert, einen passenden Rahmen des Automatisierungsgrades zu formulieren. So wurde über Nutzertests ein Weg entwickelt, der es ermöglicht vorhandene Latten zu

integrieren, jedoch die Mitbestimmung der Nutzer:innen so einzuschränken, dass eine Überfunktionalität und die damit einhergehende Überforderung der Anwender:innen vermieden wird.

#### 4. Browser-Based-Softwareprototyp

Diese unter Punkt 1. und 2. beschriebenen Entwicklungen wurden in einem Prototyp zusammengebracht und in das vorhandene Webtool integriert, um es mit Anwender:innen testen zu können.

Ein wichtiges Ergebnis dieser Tests war z.B., dass die individuell platzierten Latten farbig markiert werden mussten, da kleine Unterschiede von individuellen Lattenmaßen der 3D-Darstellung im Browser nicht mehr zu erkennen ist. (siehe folgende Abb.)



<b>Latten Maße</b>	34 mm	<b>Breite</b>	1200 mm	<b>Höhe</b>	400 mm
<input type="text" value="34"/>		<input type="text" value="1200"/>		<input type="text" value="400"/>	
<b>Tiefe</b>	300 mm	<b>Höhe Verstrebung</b>	130 mm	<b>Lücke der Sitzlatten</b>	11 mm
<input type="text" value="300"/>		<input type="text" value="130"/>		<input type="text" value="11"/>	

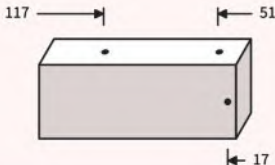
**1. individuelle Latte**

<b>Länge 1.Latte:</b> 500 mm	<b>SeiteA 1.Latte:</b> 70 mm	<b>SeiteB 1.Latte:</b> 150 mm
<input type="text" value="500"/>	<input type="text" value="70"/>	<input type="text" value="150"/>
<b>Farbe 1.Latte:</b>	<b>diese Latte nutzen?</b> <input checked="" type="checkbox"/>	
<input type="color" value="#4CAF50"/>		

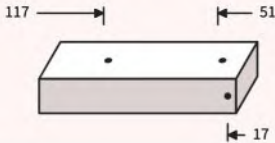
**2. individuelle Latte**

<b>Länge 2.Latte:</b> 1400 mm	<b>SeiteA 2.Latte:</b> 100 mm	<b>SeiteB 2.Latte:</b> 70 mm
<input type="text" value="1400"/>	<input type="text" value="100"/>	<input type="text" value="70"/>
<b>Farbe 2.Latte:</b>	<b>diese Latte nutzen?</b> <input checked="" type="checkbox"/>	
<input type="color" value="#C0392B"/>		

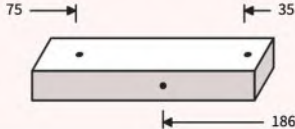
**1 × 1 × A\_1**  
70 × 150 mm 400 mm



**1 × 1 × A\_2**  
100 × 70 mm 400 mm



**1 × 1 × A\_3**  
89 × 60 mm 452 mm



#### 5. Bedarfserhebung für konkrete Konstruktionen des Web-Katalogs

In diesem Arbeitspaket wurden in zwei Workshops mit Bürger:innen Bedarfe erhoben und eine Liste mit möglichen Alltagskonstruktionen wie Möbeln erstellt. In einem zweiten Schritt wurden Entwürfe, Skizzen und Modelle erarbeitet, um das explorative Brainstorming mit konkreten Vorstellungen anzureichern. Die Ideen und Entwürfe aus den Workshops dienten als Grundlage für die Entwicklung der Konstruktionen für den Onlinekatalog. Mit einer online Google-Forms Umfrage wurde von Interessierten und zukünftigen Nutzern über eine finale Liste des Katalogs abgestimmt.

## 6. Entwicklung der einzelnen Konstruktionen des Katalogs

### 6.1 Entwicklung und Finalisierung der Entwürfe

Unter folgenden Prämissen wurden in diesem Schritt die Konstruktionen iterativ entwickelt und schlussendlich finalisiert.

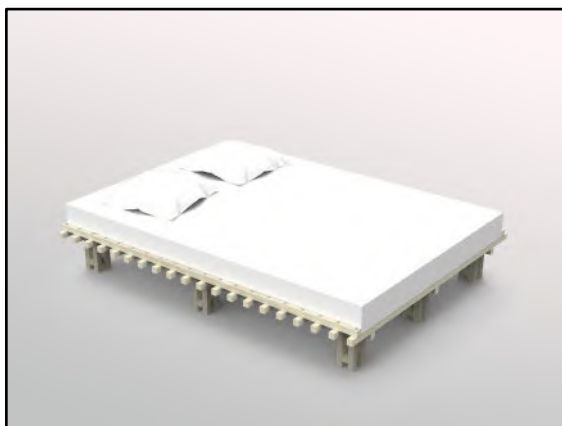
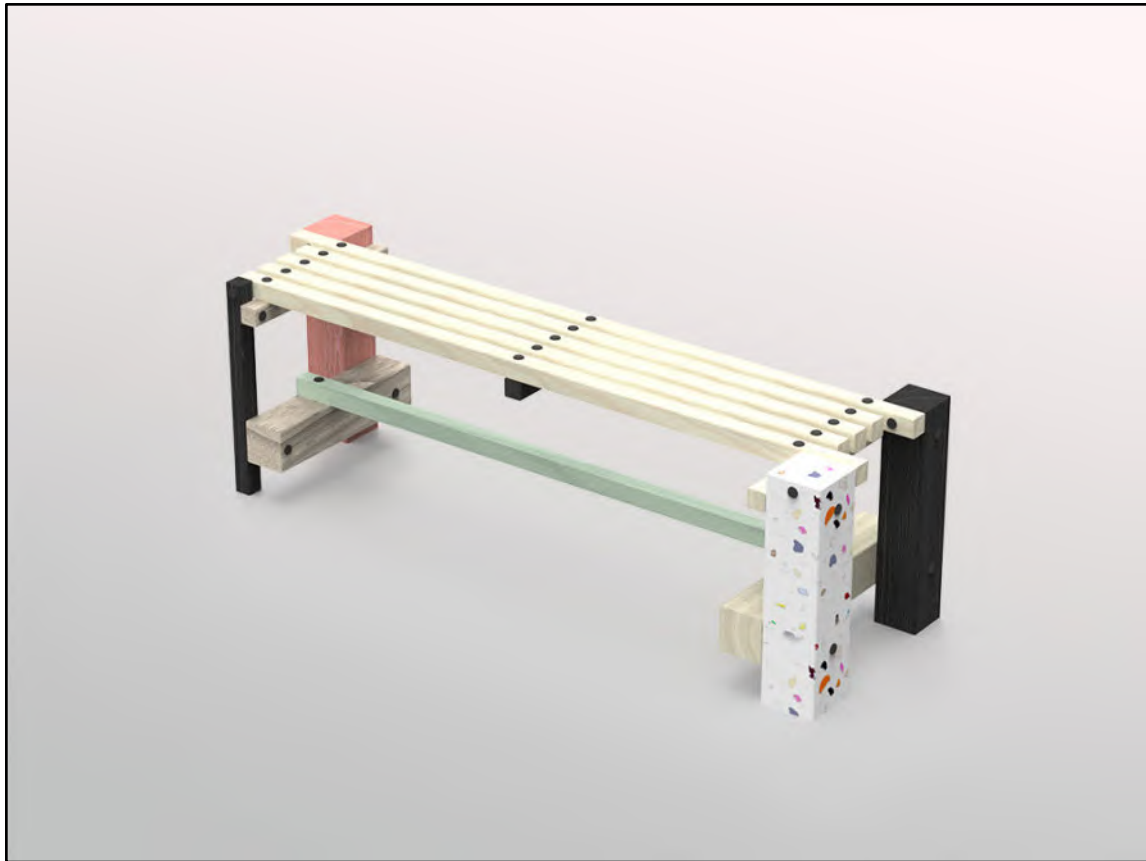
- form follows function
- klare Formensprache
- materialsparend
- Entwürfe sollten sich und ihren Gebrauch selbst erklären
- einfacher Aufbau
- hoher ästhetischer Anspruch

### 6.2 Die finalen Entwürfe des Online-Katalogs

- 1 Bank – circular
- 2 Esstisch – Stockhausen
- 3 Beistelltisch – Hohes-C
- 4 Garderobe – Change
- 5 Bett – Layover
- 6 Couchtisch – Little Japan
- 7 Bank – Seat Sheet
- 8 Esstisch – ChopChop
- 9 Garderobe – Float
- 10 Regal – Self Shelf
- 11 Schlafbank – Nap
- 12 Couchtisch – Superfici
- 13 Schreibtisch – One
- 14 Bank – Kartesi
- 15 Sideboard – Europe

Auswahl von Vorschau Bildern der Konstruktionen:





### 6.3 Entwicklung der Parameterräume

Da die Entwürfe individualisierbar sein sollten, wurden in einem folgenden, aufwändigen Schritt die Parameterräume entwickelt.

Dabei werden die entwurfsdefinierenden Parameter bestimmt, beispielsweise: Höhe, Breite, Tiefe einer Konstruktion. Oder auch individuelle Parameter, wie Abstand der Sitzlatten einer Bank.

Die Entwicklungsarbeit in diesem Arbeitspaket schließt die Validierung der Kombinationsmöglichkeiten der unterschiedlichen Parameter ein. Kurz gesagt, muss gewährleistet werden, dass sich bei allen möglichen Kombinationen, die Konstruktionen aufbaubar bleiben, den ästhetischen Ansprüchen genügen und für der Nutzung stabil genug sind. So werden manche Parameterskalen beispielsweise während der Nutzung des Tools, nach der Einstellung eines kleinen



Lattenquerschnittes durch Anwender:innen, verkleinert. (Eine Bank kann nicht 200cm breit sein, wenn sie aus 20mm Latten gebaut werden soll, im Vergleich zu 35mm Latten, wo dies funktioniert) Aus diesem Grund werden manche Konstruktionen bei bestimmten Parametereinstellungen automatisch erweitert. Beispielsweise erhält der Tisch- ChopChop ab einer bestimmten Breite ein drittes Bein im Tischgestell (siehe folgende Abb.)



Darüber hinaus besitzen manche Entwürfe konstruktive Optionen: So kann beispielsweise der Tisch „Stockhausen“ mit drei unterschiedlichen Verstrebungsvarianten angepasst werden (siehe folgende Abb.: 1. Sechs Beine, 2. Zwei Querstreben und 3. Diagonale)



## 7. Release des OPENJOYN CIRCULAR Tools

Das Software-Tool und der entwickelte Katalog wurde in die bestehende OPENJOYN Webanwendung integriert, damit es von Bürger:innen genutzt werden kann: <https://openjoyn.milz.studio>

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Zielgruppe des OPENJOYN - CIRCULAR Tools sind alle Bürger:innen und Gruppen wie Vereine etc., welche ein Bedürfnis nach individualisierten Konstruktionslösungen haben, die explizit vorhandene Objekte, Rohstoffe und Baumaterialien integrieren. Da die Lösungen in Form von Möbeln und anderen Holzkonstruktionen parametrisch individualisierbar sind und der Gesamtprozess der Nutzung stark intuitiv ist, werden auch unerfahrene Laien befähigt, die zur Maker-Community und dem Holzhandwerk bisher wenig Kontakt hatten.

Wir planen das Projekt weiterzuentwickeln und auszubauen. Für den Oktober 2022 ist bereits eine Workshop Reihe für und mit Bürger:innen im Aufbau Haus am Moritzplatz, Berlin in Planung.

### **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Die Arbeitspakete wurden mit einer iterativen Methode bearbeitet. So wurden für Probleme oder Funktionen stets mehrere unterschiedliche Hypothesen und passende Umsetzungsstrategien skizziert und durch Prototypen getestet. Nach einer Validierung im Team und mit User-Tests wurden nur die erfolgreichen Ansätze weiterverfolgt.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Bürger:innen können sich unter folgenden Links detailliert über die Projektergebnisse informieren und die Open Source Software nutzen. Durch die Bereitstellung einer laufenden Web-Anwendung wird die spezielle Zielgruppe (Laien) ab sofort befähigt das Tool, auch ohne technische Kenntnisse im Bereich Softwareentwicklung, zu nutzen.

<https://openjoyn.milz.studio>

<https://github.com/zlimcode>

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Projektplanung musste im Projektverlauf nicht angepasst werden. Die einzelnen Arbeitspakete wurden in der Planung mit einer iterativen Struktur angelegt, was eine agile und somit stets produktive Bearbeitung ermöglichte. Die Arbeits- und Kostenplanung wurde im Projekt eingehalten.

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Es gab keine Entwicklungen anderer Personen oder Institutionen, die Einfluss auf unsere Arbeiten und Zielsetzung hatten.

# Richtlinie zum „Software-Sprint“

## CIN – Eines KassenSichV 2020 kompatibles Open Source Kassensystem.

---

### *Schlussbericht*

Zuwendungsempfänger:

Lars Böttcher

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S03 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Die grundlegende Idee des CIN-Projektes war die Schaffung eines modernen und professionellen Kassen- und Warenwirtschaftsökosystems in den Händen einer engagierten Community. Die Notwendigkeit dazu ergab sich mit dem Inkrafttreten der Kassensicherungsverordnung (KassenSichV), welche momentan von keinem Open Source Kassensystem erfüllt wird. Die Kassensoftware sollte plattformunabhängig und auf dem Desktop und mobilen Geräten lauffähig sein.

Für das Kassensystem sollte eine Client-Server-Architektur eingesetzt werden. Der Server selbst liefert und empfängt die vom Kassensystem benötigten bzw. erzeugten Daten und speichert diese in einer zentralen Datenbank. Die Kommunikation zwischen den Clients und dem Server erfolgt über eine RESTful-JSON-API.

Die Verkaufstransaktionen sollten in einer Technischen Sicherheitseinheit TSE protokolliert werden und für die Finanzbehörde exportiert werden können.

Im Rahmen der Förderung wurde ein Prototyp des Frontends der Kassen inklusive TSE-Anbindung, die RESTful-API und das webbasierte Backend des Kassenservers implementiert.

Die drei Hauptaufgaben für das Kassensystem waren die Implementierung eines modernen User Interfaces, die Anbindung der Technischen Sicherheitseinrichtung TSE und die Umsetzung der Plattformunabhängigkeit.

Die serverseitigen Aufgabenblöcke bestanden aus der Implementierung der JSON-API, der Anbindung der Datenbank, sowie der Entwurfs des Datenbanken-Schemas und die Entwicklung des webbasierten Backends.

## **Beitrag des Projektes zu den Zielen der Förderinitiative**

### **„Software-Sprint“**

Die Kassensicherungsverordnung (KassenSichV) schreibt vor, dass alle in Deutschland eingesetzten Kassensysteme seit dem 30.08.2020 eine aktivierte Technische Sicherheitseinrichtung (TSE) besitzen müssen. Die TSE sorgt für eine manipulationssichere Aufzeichnung aller Kassenvorgänge und verhindert eine nachträgliche Bearbeitung der Kassendaten. Bis zum jetzigen Zeitpunkt existiert kein quelloffenes Kassensystem, welches eine Technische Sicherheitseinrichtung integriert hat. Mit Inkrafttreten der Kassensicherungsverordnung standen alle Unternehmen, die auf freie Software und freie Betriebssysteme setzten, vor großen Problemen. Es mussten proprietäre Kassensysteme angeschafft und eingeführt werden. Für Linux existieren aber keine bezahlbaren Kassensysteme, die den in den meisten Fällen benötigten Funktionsumfang bieten. Viele kleine Geschäfte und Restaurants sind zu Cloud-Lösungen gewechselt und geben seitdem ihre sensiblen Geschäftsdaten an die Cloud-Betreiber\*innen weiter. Wer auf freie Betriebssysteme setzte und seine Daten nicht in fremde Hände geben wollte, musste auf proprietäre Betriebssysteme wechseln. Damit ging oftmals ein größerer Umbau der IT-Struktur einher, aber noch viel entscheidender war das Abwenden von den Prinzipien freier Software und den eigenen Überzeugungen.

## **Ausführliche Darstellung der Ergebnisse**

Im Projektzeitraum wurden alle wesentlichen Meilensteine adressiert und ein funktionsfähiger Prototyp fertiggestellt. Dieser wurde auf der Abschlussveranstaltung, dem DemoDay, präsentiert. Im Rahmen der Entwicklung des webbasierten Backends ist das MXXN Framework entstanden. Dabei handelt es sich um ein Framework zur einfachen und schnellen Entwicklung von modernen Webanwendungen und/oder RESTful-JSON-API. Das Framework vereint eine gut überlegte Auswahl an Software-Komponenten, welche alle unter einer freien Lizenz stehen, weit verbreitet sind und sehr gut gepflegt und weiterentwickelt werden. Als Datenbank-ORM wurde auf SQLAlchemy gesetzt. Zur Versionierung des Datenbankschemas wurde Alembic verwendet. Das zugrunde liegende Webframework ist die asynchrone API vom Falcon. Mit all diesen Komponenten gab es keine größeren Probleme während der

Entwicklungsphase. Im Bereich des JavaScript-Frameworks zur Entwicklung des Web-Frontend wurde anfänglich auf RiotJS gesetzt. Leider hatte dieses Frontend nicht den benötigten Funktionsumfang und nach kurzer Zeit wurde die Entscheidung zum Wechsel auf Lit und TypeScript getroffen. Mit dieser Kombination konnten alle Ideen umgesetzt werden. Das Framework enthält ein Kommandozeilenprogramm das bei der Entwicklung von Anwendungen und des Datenbankschemas hilft. Nahezu alle Elemente des Frameworks können überladen und an die eigenen Bedürfnisse angepasst werden (Resources, URLs, Strings, Themes etc.).

Beim zweiten großen Meilenstein handelt es sich um die Anbindung der Technischen Sicherheitseinrichtung TSE. Hier wurde im Antrage die Verwendung der TSE der Bundesdruckerei genannt. Leider war es mir nicht möglich eine Entwicklerversion dieser zu erwerben/bekommen. Aus diesem Grunde wurde im Rahmen des Projektes auf die TSE der Firma Epson gesetzt, was sich als eine gute Wahl herausstellte. Es wurden das Python-Module tse.py entwickelt, welches sich im Python Package Index PyPI befindet und von anderen Projekten verwendet werden kann. Das Paket ist so konzipiert, dass es zukünftig auch TSE-Module anderer Hersteller als Backend nutzen kann. Es wurde der gesamte für die KassenSichV 2020 benötigten Funktionsumfang implementiert und mit den benötigten Test versehen. Des Weiteren wurden Langzeittests für den Mehrkassenbetrieb geschrieben.

Für das Kassen-Frontend, als dritter Meilenstein, wurden die grundlegenden Funktionen einer Kasse implementiert. Hierzu wurde die TSE , welche die Verkaufstransaktionen aufzeichnet, angebunden. Die Transaktionen können für die Steuerbehörde exportiert und archiviert werden. Die Kasse selbst ist unter den Betriebssystemen Linux, Windows und Android lauffähig. Sie bezieht die Produktdaten vom Backendserver und cached diese indiziert in einer lokalen Datenbank, was zu einer schnellen und effizienten Produktsuche führt. Das Kassen-Interface wurde für die Benutzung von Touchscreens optimiert. Es wurde auch die Möglichkeit der Verwendung von Barcodelesern implementiert. Wären der Entwicklung zeigte sich, dass das verwendete GUI-Toolkit auf mobilen Geräten etwas zu träge ist. Momentan arbeite ich an der Portierung auf ein leichtgewichtigeres und somit schnelleres Toolkit.

Die Unterstützung bei der Projektdurchführung durch die Open Knowledge Foundation empfand ich als nahezu perfekt. Sie waren immer ansprechbar und konnten mit ihrer langjährigen Erfahrung immer mit Rat zur Seite stehen. Auch die Unterstützung durch den Projektträger war optimal.

Eine der wichtigsten Erkenntnisse aus der Projektarbeit ist, dass es immer Fehlentscheidungen gibt und man nicht allzu lange an Workarounds arbeiten sollte, sondern sich einfach eingestehen kann, dass die erste Ideen vielleicht einfach nicht die richtige war.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Mein Prototyp zeigt die Funktionsweise und die Machbarkeit meiner Idee. In einem kritischen Bereich wie der Warenwirtschaft führt jeder Fehler und/oder der Ausfall einer Kasse zu großen Problemen und natürlich kann dann nichts verkauft werden. Hier muss die Software einen hohen Qualitätsstandard erfüllen und maximal getestet sein. Auch bedarf es eines hohen Funktionsumfangs, den meine Implementierung momentan noch nicht enthält. Ich werde weiter am Projekt arbeiten und eine kleine Gruppe von Menschen ist schon dazugestoßen. Es wurden zwei neue Projekte auf der Basis des MXXN-Frameworks gestartet und ich bin zuversichtlich, dass die Kassen die Produktreife erreicht. Das TSE-Paket wurde in letzten Monat ca. 100 mal heruntergeladen und wird somit wahrscheinlich auch schon von weiteren Entwicklern eingesetzt.

Im Rahmen des Projektes habe ich mir einige neue Technologien erschlossen und mich fachlich stetig weiterentwickelt und danke den Fördertopf für diese Möglichkeit.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Wie oben bereits erwähnt war es mir nicht möglich eine Technische Sicherheitseinrichtung der Bundesdruckerei zu bekommen und somit wechselte ich zu einer Lösung der Firma Epson.

Außerdem hatte das ausgewählte JavaScript-Framework leider nicht den benötigten Funktionsumfang und ich tauschte es gegen ein leistungsstärkstes aus

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Die jeweiligen Teile des Projektes können unter folgenden Links einsehen werden:

MXXN-Framework:

Gitlab: <https://gitlab.com/ccodein/mxxn/mxxn>

Website: <https://ccodein.gitlab.io/mxxn/mxxn/>

TSE-Paket:

Gitlab: <https://gitlab.com/ccodein/tse.py>

Website: <https://ccodein.gitlab.io/tse.py>

PyPI: <https://pypi.org/project/tse.py/>

Kassen-Frontend:

Gitlab: <https://gitlab.com/ccodein/cin/cin>

Gitlab: <https://gitlab.com/ccodein/cin/cin-mobile>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Bei der Entwicklung des TSE-Pakets kam es zu einem erheblichen Mehraufwand. Die Dokumentation der Hersteller war mehr als nur unzureichend und somit musste ich mir die Funktionsweise der TSE und vor allem die Fehlerbehandlung größtenteils durch ausprobieren erschließen. Da die TSE nach den meisten Fehler zurückgesetzt bzw. neu gestartet werden musste, bedeutete dies einen hohen Mehraufwand an Zeit, mit dem ich nicht rechnete. Diese Zeit fehlte dann in anderen Bereichen des Projektes. Auch der Austausch des JavaScript-Frameworks bedeutete ein deutlichen Mehraufwand.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Im Rahmen des Projektes habe ich das MXXN-Framework beworben und an zwei Wochenenden Workshops gegeben. Bei diesen Workshops haben wir uns ausgetauscht und über künftige Features und die Zusammenarbeit gesprochen.

# Richtlinie zum „Software-Sprint“

doc2data

---

## *Schlussbericht*

Zuwendungsempfänger:

Sergej Levich

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S04 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.



## Kurze Darstellung der Aufgabenstellung und Motivation

Die Verarbeitung von Dokumenten ist eine allgegenwärtige Aufgabe. Insbesondere in Verwaltungsprozessen wird diese hauptsächlich manuell ausgeführt, was zu hohen Kosten, Verzögerungen und teilweise Intransparenz führt. Covid hat die Problematik der Überlastung der Behörden aufgrund operativer Ineffizienz deutlich gemacht.

Mit doc2data verfolgte der Autor das Ziel, eine Software zur Beschleunigung und Vereinfachung der Dokumentenverarbeitung zu entwickeln. Konkret resultierte das Vorhaben in einer Python-Bibliothek, welche das Anlernen von Machine Learning (ML) Modellen für eine Reihe von Aufgaben ermöglicht. Damit können Dokumente in Teilen automatisiert verarbeitet werden.

Das ursprüngliche Vorgehen orientierte sich an einer Konzeptphase mit initialem Nutzerfeedback und einer anschließenden Entwicklungsphase. Vor Projektbeginn konnten jedoch zwei Kooperationspartner akquiriert werden, welche Dokumentendaten aus realen Prozessen zur Verfügung gestellt haben. Damit konnte die Entwicklung agil entlang echter und gängiger Problemstellungen durchgeführt werden. Folglich wurde mit doc2data eine eigenständige Bibliothek entwickelt und zeitgleich in der Praxis pilotiert.

Wichtige Meilensteine waren hierbei:

- Recherche (Parsing von PDFs, Deep Learning Architekturen)
- Entwicklung eines Moduls für die Erstellung und Verwaltung von PDF-Sammlungen
- Durchführung des Modelltrainings für vier Aufgaben der Dokumentenverarbeitung
- Entwicklung von Modulen für weitere Teile der Trainingspipeline
- Erstellung der Dokumentation

Durch die Beteiligung von Kooperationspartnern sind zusätzliche administrative Aufgaben bearbeitet worden. Dazu gehörten Abstimmungen, die Zusammenarbeit beim Aufsetzen der IT-Infrastruktur sowie die Durchführung von (Abschluss-)präsentationen.

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Das Projekt doc2data kann als Softwareinfrastrukturprojekt verstanden werden, da es nicht auf Endnutzer, sondern auf Entwickler\*innen als Zielgruppe ausgerichtet ist.

Die Entwicklung eigener ML-Modelle ist für Entwickler\*innen ohne entsprechende Erfahrung nur schwer möglich. doc2data verbindet eine Vielzahl bestehender Python Bibliotheken um eine klar strukturierte und einfach zu nutzende Pipeline für das Modelltraining bereitzustellen. Damit können Entwickler\*innen in wenigen Schritten ML-Modelle für die Dokumentenverarbeitung auf eigenen Daten anlernen.

Auf Basis dieser Modelle kann in der Folge eine Vielzahl von Prozessen effizienter gestaltet werden. Im Bereich der öffentlichen Verwaltung betrifft dies hauptsächlich administrative Prozesse, die Anträge auf Basis eingereichter Dokumente abwickeln. Beispiele sind die Verschlagwortung (Klassifikation) von Dokumenten im Posteingang, die Zuordnung von Dokumenten zu Akten oder das automatisierte Auslesen von Feldern aus nicht standardisierten Formularen.

Im Projektverlauf wurden neben den Verwaltungen auch Civic-Tech Entwickler\*innen als Zielgruppe identifiziert (z.B. FragDenStaat, investigative Journalist\*innen). Hier kann doc2data eingesetzt werden um große, bestehende Dokumentensammlungen zu strukturieren und zu analysieren. So kann eine kleine Anzahl von Dokumenten nach einem bestimmten Kriterium manuell annotiert werden, um anschließend den Rest der Sammlung durch das ML-Modell zu bewerten.

Die ML-basierte Dokumentenverarbeitung (Document AI) wird aktuell durch den kommerziellen Markt dominiert. Open-Source-Entwicklungen in diesem Bereich bieten kleineren bzw. finanzschwachen Organisationen einen Weg, von den neusten technologischen Entwicklungen zu profitieren.

## Ausführliche Darstellung der Ergebnisse

Aufgrund der agilen Entwicklung waren der Ressourceneinsatz fix und die Meilensteine variabel. Diese wurden im Verlauf des Projekts in insgesamt zehn Reviews angepasst. Dabei flossen die Beobachtungen und Herausforderungen aus der Arbeit mit realen Dokumentendaten in die Definition und Priorisierung der Meilensteine ein. Entsprechend wurde beispielsweise die Entwicklung eines UI zur Visualisierung der Ergebnisse zugunsten eines Annotationstools zurückgestellt. Ziel bei solchen Entscheidungen war stets, die Features so zu definieren, dass die spezifischen Probleme gelöst werden konnten, ohne dass die Bibliothek an Generalisierbarkeit einbüßt.

Insgesamt können die Ergebnisse aus vier Perspektiven betrachtet werden.

### 1 Funktionalität

Mit doc2data können Entwickler\*innen auf Basis von Dokumenten im PDF-Format und entsprechenden Annotationen ML-Modelle für die folgenden vier Aufgaben anlernen:

1. Rotation von einzelnen Dokumentenseiten
2. Zuschneiden von einzelnen Dokumentenseiten
3. Klassifikation von Textelementen innerhalb einer Dokumentenseite
4. Klassifikation von mehrseitigen Dokumenten als Ganzes

Mit dem Annotationstool doc2data UI können Entwickler\*innen Textelemente einzelner Dokumentenseiten annotieren um im nächsten Schritt mit doc2data Modelle für die Klassifikation von Textelementen (3) anzulernen.

## 2 Module der Bibliothek

doc2data besteht aus einer Anzahl von Python-Modulen, die Funktionalität für einzelne Modellierungsschritte bündeln. Im Wesentlichen können diese entlang einer sequenziellen Anwendung wie folgt dargestellt werden:

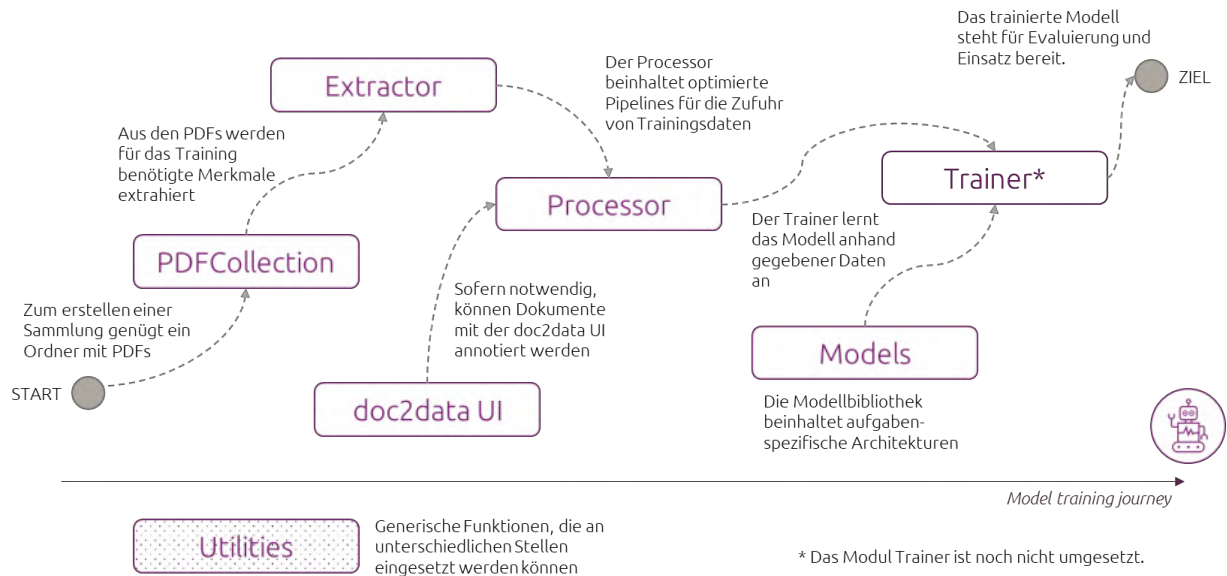


Abbildung 1. Übersicht der Module: Vom PDF zum KI-Modell mit doc2data.

Die vier verschiedenen Modellierungsaufgaben (siehe Funktionalität) werden in den Modulen *Processor* und *Models* eigens abgebildet. Alle anderen Module funktionieren unabhängig von der gewählten Aufgabe. Das Annotationstool doc2data UI ist als separate, Python-basierte Applikation konzipiert, die jedoch auf doc2data aufbaut.

## Technologie

doc2data basiert auf gängigen Open-Source-Bibliotheken, die hauptsächlich über Python-Schnittstellen eingebunden werden. Wesentliche Komponenten sind nachfolgend dargestellt:

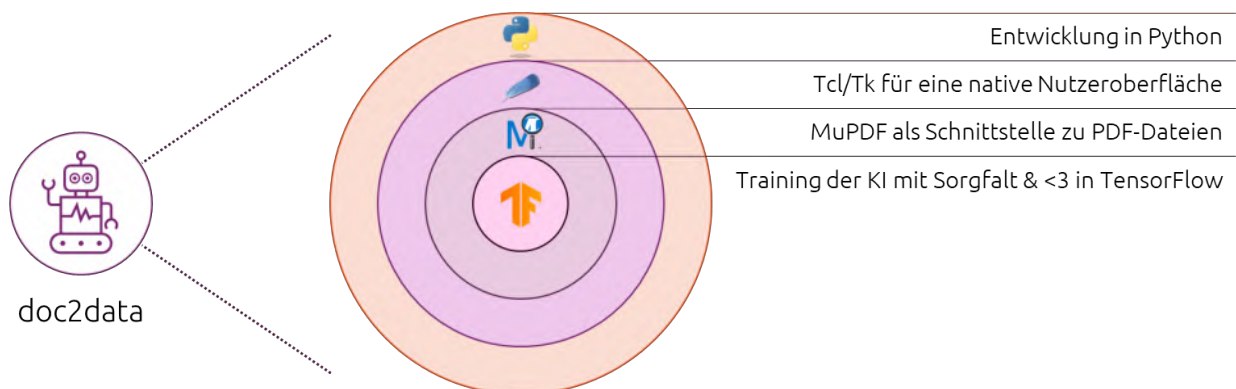


Abbildung 2. Technologische Komponenten von doc2data

Die Dokumentation basiert auf den Bibliotheken *MkDocs* und *mkdocstrings* und ist auf *Read the Docs* gehostet.

## Anwendungsfälle

Im Rahmen der Zusammenarbeit mit den erwähnten Kooperationspartnern wurde doc2data in zwei Projekten pilotiert:

1. Klassifikation von Dokumenten (Anwendungsfall bei einer Berufsgenossenschaft)
2. Verarbeitung von Kfz-Scheinen (Anwendungsfall bei einem Automobilclub)

Der erste Anwendungsfall ist im Posteingang angesiedelt und ist typisch für viele Behörden. Ziel ist hierbei die Zuordnung einer vorab definierten Kategorie für eingegangene Dokumente. Gängige Kategorien sind z.B. Rechnung, Arztbericht, Laborergebnis oder Anschreiben. Im Hinblick auf die Funktionalität von doc2data kam hier das Modell für die Klassifikation von mehrseitigen Dokumenten (4) zum Einsatz.

Der zweite Anwendungsfall beschäftigt sich mit der Extraktion spezifischer, vorher definierter Felder aus Dokumenten – einer weit verbreiteten Aufgabenstellung. Konkret ging es um die Ermittlung des Kennzeichens, des Fahrzeughalters sowie der Fahrzeugspezifika aus Kfz-Scheinen. Da Kfz-Scheine als Fotos/Scans vorlagen, musste eine Vorverarbeitung stattfinden. Damit kamen im Hinblick auf die Funktionalität von doc2data die Modelle für die Rotation (1), das Zuschneiden (2) sowie die Klassifikation von Textelementen (3) einer Dokumentenseite zum Einsatz.

Im Hinblick auf die Erkenntnisse aus der Projektarbeit wurde der Autor in seiner Einschätzung der Notwendigkeit und des Bedarfs nach Open-Source-Lösungen für die Dokumentenverarbeitung bestärkt. Auch sind die Herausforderungen und Chancen in der Zusammenarbeit mit staatlichen Akteuren deutlich geworden. Die Begleitung durch die Open Knowledge Foundation und den Prototypenfund war insbesondere durch die Vernetzung mit der Civic Tech Community und potenziellen Nutzer\*innen höchst wertvoll.

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Der zentrale Nutzen von doc2data besteht im einfachen Zugang zu ML-Modellen für die Dokumentenverarbeitung. In der Folge lassen sich Problemstellungen in Prozessen und Projekten effizienter lösen, als es durch eine manuelle Bearbeitung möglich wäre. Während die primäre Zielgruppe Behörden und freie Entwickler\*innen umfasst, ist das Thema Dokumentenverarbeitung so allgegenwärtig, dass ein deutlich größerer Anwenderkreis denkbar ist. Von Forschungseinrichtungen, die auf Datenbeschaffung aus großen Dokumentensammlungen angewiesen sind, zu KMUs, die operative Prozesse abwickeln - die Breite an kommerziellen und nicht-kommerziellen Anwendungsfällen ist signifikant.

Gegeben dessen, dass die Verarbeitung natürlicher Sprache<sup>1</sup> aktuell eine der bedeutendsten Stoßrichtungen im Bereich Künstlicher Intelligenz darstellt, ist hier in der Zukunft mit deutlichen technologischen Sprüngen zu rechnen. Projekte wie doc2data können dazu beitragen, die Lücke zwischen Forschung und Anwendung an großen kommerziellen Akteuren vorbei in einer transparenten Art und Weise zu schließen.

---

<sup>1</sup> Natural Language Processing (NLP)

Vor diesem Hintergrund soll doc2data weiterhin an der Schnittstelle von Wissenschaft und Anwendung entwickelt werden. Es werden Kooperationsmöglichkeiten mit Forschungsinstitutionen<sup>2</sup> ausgelotet. Ferner werden Möglichkeiten zur Monetarisierung von doc2data durch bspw. komplementäre Dienstleistungen für größere Organisationen untersucht.

Die Arbeit an doc2data hat den Autor auch im Hinblick auf die persönliche und fachliche Weiterentwicklung weitergebracht. Besonders hervorzuheben ist das gewonnene Wissen rund um die professionelle Entwicklung von Open-Source-Software in Python. Der Prototypenfund bietet aus der Sicht des Autors eine einmalige Gelegenheit, ein relevantes Problem mit einem ausschließlichen Fokus auf den Mehrwert der Lösung zu adressieren.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

In der Modellierung mit realen Daten wurden unterschiedliche Modellarchitekturen sowie Parameterkombinationen evaluiert. Entsprechend wurden nur die besten Experimente beibehalten und der Rest verworfen. Für die Aufgaben (1), (2) und (3) mussten Dokumente manuell annotiert werden, sodass der Datensatz nur wenige hundert Beispiele umfasst. Damit konnte keine hinreichend hohe Modellgenauigkeit erreicht werden. Daraufhin mussten Methoden der „data augmentation“, d.h. der künstlichen Anreicherung der Daten, eingesetzt werden, um die Qualität der Modelle zu erhöhen.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Der Quellcode des Projekts ist auf zwei Repositorien aufgeteilt (doc2data & doc2data UI).

doc2data (Bibliothek): <https://github.com/serge724/doc2data>

doc2data UI (Annotations-Tool): <https://github.com/serge724/doc2data-ui>

Dokumentation für doc2data: <https://doc2data.readthedocs.io/en/latest/>

Webseite des Anwendungsfalls (1): <https://sigai.de/featured/theses/thesis2/>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Das Projekt wurde agil durchgeführt, sodass die Planung in regelmäßigen Abständen angepasst wurde. Aufgaben, die größere Aufwände als geplant verursachten, betrafen das Einhalten von Standards in der Entwicklung, die Kommunikation mit den Kooperationspartnern sowie die Entwicklung des Annotations-Tools (doc2data UI). Teilaufgaben mit Minderaufwänden gab es dagegen bei der Implementierung der PDF-Schnittstelle sowie des vortrainierten Modells *LayoutXLM*.

Ein neuer Release des Open-Source-Tools *PaddleOCR* war inkompatibel mit den Abhängigkeiten von doc2data. Es konnte auf die Alternative *docTR* ausgewichen werden, was sogar zu einer höheren Qualität der Zeichenerkennung führte.

---

<sup>2</sup> ZHAW Zürcher Hochschule für Angewandte Wissenschaften, Institut für Medizinische Informatik der Charité

## Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während die meisten Modellarchitekturen vom Autor selbst aufgesetzt wurden, erfolgte die Integration des vortrainierten Modells *LayoutXLM* durch die Open-Source-Bibliothek *huggingface* (<https://github.com/huggingface>), sodass keine eigene Implementierung notwendig war. Zusätzlich ist im Entwicklungszeitraum die OCR Bibliothek *docTR* (<https://github.com/mindee/doctr>) verfügbar geworden.

# Richtlinie zum „Software-Sprint“

## GSM – GNOME Shell Mobile

---

### *Schlussbericht*

Zuwendungsempfänger:

Jonas Dreßler, Tobias Bernard GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S05 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

*Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?*

Der Handymarkt ist momentan fast komplett von Apple und Google kontrolliert. Dies ist ein Problem aus vielerlei Hinsicht, sei es Datenschutz, Sicherheit, Digitale Souveränität oder Innovation. In den letzten paar Jahren hat sich mit Projekten wie Librem 5 oder Pinephone hier aber einiges bewegt, sodass es jetzt erstmals mobile Hardware gibt, auf der freie Software langfristig unterstützt werden kann. Dies hat dazu geführt, dass sich ein Ökosystem von freien Betriebssystemen für Mobilgeräte entwickelt hat.

Seit der Entwicklung von GNOME 3 vor 10 Jahren ist die Unterstützung von verschiedenen Gerätetypen (neben Desktops und Laptops auch Tablets, und in den letzten paar Jahren Smartphones) ein grundlegender Anspruch von GNOME. Inzwischen ist ein großer Teil der GNOME Apps bereits responsive, und dadurch auch auf Smartphones benutzbar.

Dank unserer großen Community existiert bereits ein solides und schnell wachsendes App-Ökosystem, sowie ein moderner Technologie-Stack und gute Entwicklertools. All dies macht die GNOME Plattform zu einem vielversprechenden Ansatz für ein offenes, freies und einfach zu bedienendes mobiles Betriebssystem. Vor unserem Förderprojekt war die Shell die wichtigste fehlende Komponente für eine wirklich konkurrenzfähige User Experience auf mobilen Linux Geräten.

Die wichtigsten Meilensteile dabei waren:

1. Panel und Menüs: Anpassung der Systemmenüs in der oberen Leiste für Touchscreens, inklusive Schnelleinstellungen und Benachrichtigungen
2. Navigation: Hinzufügen einer unteren Leiste für systemweite Gestensteuerung, Implementierung von horizontalem Multitasking
3. App Raster: Kleinere Anpassungen am App-Raster um es für kleine Displays zu optimieren
4. Tastatur: Anpassung der Bildschirmtastatur für kleine Bildschirme

Diese haben wir alle wie geplant geschafft, und wir planen diese Arbeit über die nächsten Monate in das GNOME Shell Mutterprojekt einzupflegen.

## **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

*Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?*

Die Zielgruppe sind Benutzer:innen von Smartphones, also prinzipiell jede:r.

Es gibt bereits einige Linux Smartphones, die freie Betriebssysteme vorinstalliert haben. Wir gehen davon aus dass sich unsere Shell dank diesem Projekt als neuer Standard für diese Geräte durchsetzen wird. Zum Beispiel verwendet das Librem 5 heute die Phosh Shell, was technisch keine gute langfristige Basis darstellt. Mit den Verbesserungen unseres Projekts ist GNOME Shell inzwischen eine deutlich bessere User Experience.

Als Teil unseres Projekts konnten wir auch generelle Verbesserungen an der Oberfläche in GNOME Shell vornehmen (z.B. ein komplett überarbeitetes Gestensystem). Diese werden natürlich allen GNOME-Nutzern zu Gute kommen, z.B. auch auf Fedora oder Ubuntu.

## **Ausführliche Darstellung der Ergebnisse**

*Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?*

Die Ziele, die wir uns gesetzt hatten haben wir wie geplant erreicht: Wir haben jetzt Entwicklungszweige für GNOME Shell (die grafische Oberfläche) und Mutter (der Fenstermanager), die eine relativ komplette und stabile mobile Benutzererfahrung mit GNOME Shell ermöglichen. Dies beinhaltet moderne Gesten für Navigation, ein reorganisierbares Applikations Raster mit Ordern, ein neues Layout für die Suche, eine angepasste Tastatur mit neu gestalteter Emoji Eingabe, sowie Schnelleinstellungen und Benachrichtigungen, welche via Geste von oben geöffnet werden können. Diese haben wir beim Prototypenfund Demo Day im August auf einem Pinephone Pro vorgeführt, was beim Publikum gut ankam.



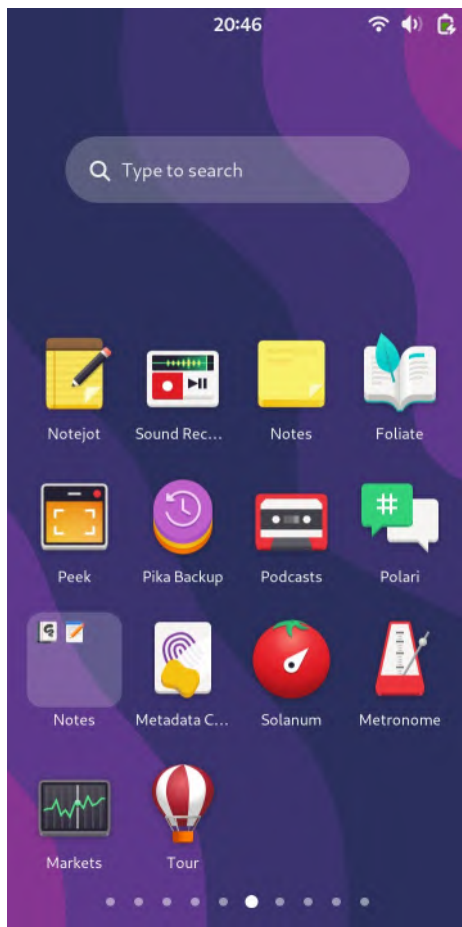


Figure 1: Das Applikations Raster

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

*Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?*

*Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?*

Der Nutzen für unsere Zielgruppe sind kurzfristig in erster Linie die Verbesserungen an Kernaspekten der GNOME Shell, vor allem das neue Gestensystem. Dieses erlaubt es erstmals, komplexe Gesten einfach in GNOME Shell zu implementieren. Dies wird auch in Zukunft weitere Gesten ermöglichen und gibt uns mehr Flexibilität in der Gestaltung. Ähnliches gilt für die Verbesserungen an der Bildschirmtastatur und dem Applikations Raster.

Längerfristig ermöglicht unsere Arbeit, die gleiche Shell Codebasis sowohl auf Desktops als auch auf Smartphones zu verwenden. Wie wir bereits an diesem Projekt gesehen haben, hat dies große Vorteile für beide, weil der Großteil der Arbeit nicht doppelt gemacht werden muss. Dies macht höhere Softwarequalität und schnellere Entwicklung mit den selben Ressourcen möglich.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

Ursprünglich war geplant, eine Übersichtsansicht mit nur den offenen Applikationen zu haben, aber im Laufe der Entwicklung und nach einigen informalen Benutzerinterviews haben wir letztlich entschieden, diese Idee nicht mehr zu verfolgen. Es war zu einfach möglich, aus Versehen Applikationen zu schließen, und diese Ansicht hatte keine Alleinstellungsmerkmale da ihre Funktionalitäten (Applikationen schließen, zwischen offenen Applikationen wechseln, etc.) auch im App Raster verfügbar sind.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

Wir haben auch einen Blog Post auf dem GNOME Shell Blog mit dem aktuellen Entwicklungsstand und Plänen für die Zukunft veröffentlicht: <https://blogs.gnome.org/shell-dev/2022/09/09/gnome-shell-on-mobile-an-update>

Die beiden Entwicklungszweige mit dem Code:

- <https://gitlab.gnome.org/verdre/gnome-shell/-/tree/mobile-shell>
- <https://gitlab.gnome.org/verdre/mutter/-/tree/mobile-shell>

Dieser Code wird über den nächsten Entwicklungszyklus in die Hauptentwicklungszweige von GNOME Shell und Mutter eingepflegt, aber wie im Blog Post erwähnt arbeiten einige Menschen bereits daran, den momentanen Stand auf Distributionen wie z.B. Postmarket OS zu packagen.

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Nachdem der genaue Projektverlauf unter anderem von externen Faktoren (z.B. der Arbeit anderer Entwickler oder Reviews durch Maintainer) abhängig war, mussten wir unseren Zeitplan einige Male angepasst werden, oder Aufgaben vorgezogen und aufgeschoben werden. Ein Beispiel dafür ist, dass eine der Teilaufgaben (das neue Design der Schnelleinstellungen) von einem anderen Entwickler unabhängig von unserer Arbeit erledigt wurde.

Generell sind wir aber zufrieden mit der Planung, da wir alles in etwa wie vorhergesehen geschafft haben und das Ergebnis sowhol in der Entwicklergemeinschaft als auch einem breiteren Publikum gut ankommt.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Ja, wie bereits oben erwähnt waren einige Teile unserer Arbeit abhängig von den Reviews durch Maintainer des Projekts. Zum Beispiel hatten wir gehofft, das neue Gestensystem in GNOME 43 diesen Herbst zu kriegen, aber es sieht jetzt so aus als ob das doch nicht möglich sein wird, weil die Maintainer nicht genügend Kapazitäten für Reviews hatten.

Für den Prototype Fund war unser Ziel in erster Linie, einen funktionierenden Prototypen am Ende der Förderzeit zu haben, wofür Reviews nicht so relevant sind, weil die Arbeit ja auch erst später eingepflegt werden kann. Es macht allerdings insofern einen Unterschied, dass diese Verbesserungen dadurch erst später bei den meisten GNOME-Nutzer:innen ankommen.

# Richtlinie zum „Software-Sprint“

## QubesOSConfig – Leicht bedienbares Konfigurationstool für Qubes OS

---

### *Schlussbericht*

Zuwendungsempfänger:

Marta Marczykowska-Górecka

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S06 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

## Kurze Darstellung der Aufgabenstellung und Motivation

Kommerzielle Betriebssysteme und Programme bieten ein optimiertes, angenehmes Nutzererlebnis. In Gegenteil, Open-Source-Software gilt weiterhin als schwierig, kompliziert und unfreundlich. Qubes OS ist ein gutes Beispiel – es ist ein auf Sicherheit fokussiertes Betriebssystem, welches dies durch Isolation gewährleistet. Qubes OS bietet Sicherheit, Datenschutz und Privatsphäre in verschiedenen Bereichen und ermöglicht eine höhere Stufe von Schutz gegen Malware, Spyware und andere böswillige Akteure im Internet. Die Nutzung von Qubes OS ist – vielleicht – nicht zu kompliziert, aber die Konfiguration ist sehr schwierig.

Zum Beispiel, ein von den wichtigsten Teilen des Systems – die Zugriffssteuerungskonfiguration von Isolierten *qubes* - hatte sich bisher auf Textkonfigurationsdateien verlassen. Es gibt keine einheitliche Lösung – eine Konfiguration für alle ist fast nutzlos, da verschiedene Nutzer\*innen sehr unterschiedliche Bedürfnisse in Bezug auf Isolation, Privatsphäre und Sicherheit haben.

Wie oft in Open-Source-Bereich, die Entwickler von Qubes OS sind erfahrene Experten und haben oft Schwierigkeiten, die Bedürfnisse der Nutzer\*innen zu verstehen. User Research ist nur selten bei Open Source Projekte benutzt – das wollte ich ändern. Ich habe geplant, ein Tool für Qubes OS Konfiguration zu entwickeln, das auf User Research basiert ist, und das für alle verfügbar und nutzbar ist, nicht für Power-User.

Um das Problem anzugehen, waren folgende Meilensteine vorgesehen:

- Erste Befragung von Nutzer\*innen
- Erstellen/Anpassen des Design-Entwurfes basierend auf dem Ergebnis der Befragung
- Interviews mit Nutzer\*innen zu dem erstellten Design-Entwurf
- Implementierung
- Tests
- Veröffentlichung

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Ein sicheres und datenschutzbewusstes Betriebssystem ist die Grundlage jeder Computernutzung. Wenn die wenigen verfügbaren Optionen benutzerunfreundlich oder unsicher/gefährlich für die Privatsphäre sind, dann kann kein Civic Tech gut verwendet werden. Das UX-Problem bei Open Source Software ist ein großes, bekanntes Problem; Open Source kann viele Probleme lösen, aber nur für erfahrene Nutzer\*innen und Expert\*innen. Mein Projekt hat versucht, einen kleinen Teil dieses Problem zu lösen: einen Baustein von sicheres Computing (das Betriebssystem) benutzerfreundlicher und zugänglicher zu machen.

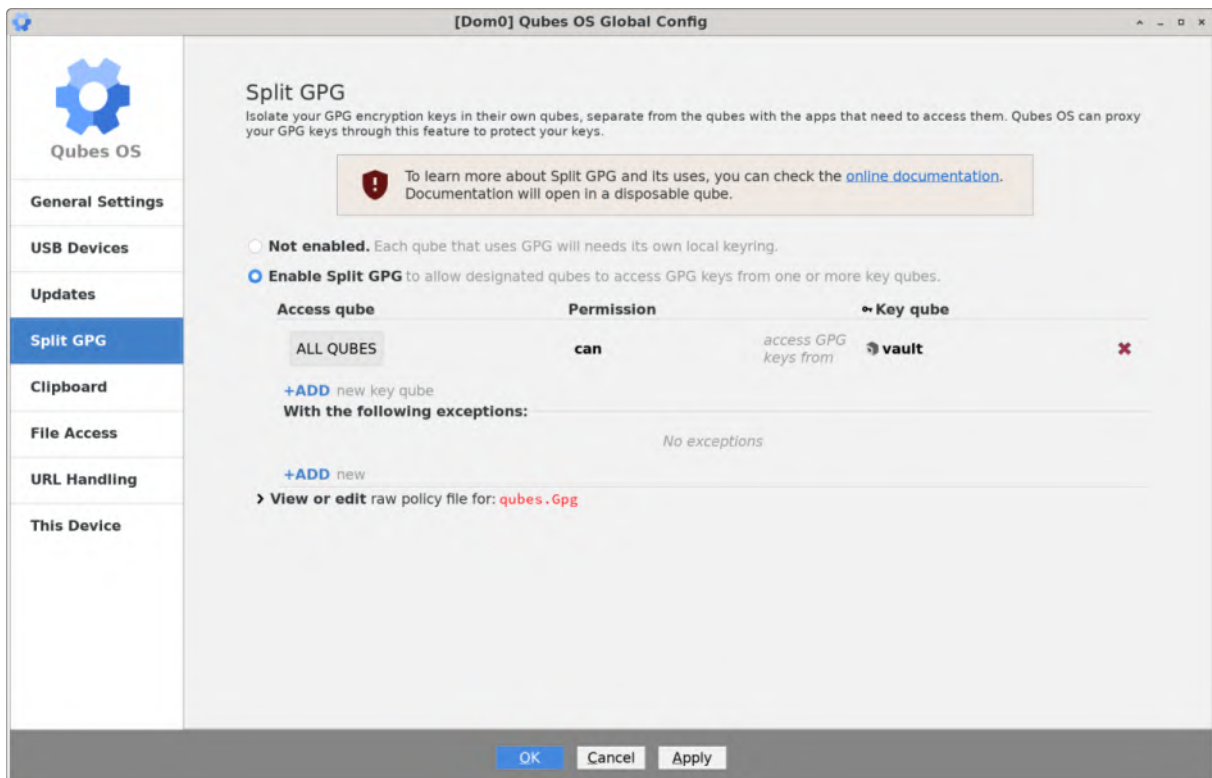
Es gibt ein großes Problem bei Nachhaltigkeit von Open-Source-Projekten: das UX ist oft ein nur ein nachträglicher Gedanke, und Benutzeroberflächen wachsen in einer zufälligen, benutzerunfreundlichen Weise. Ich möchte dieses wichtige Teil von Softwareinfrastruktur, das Betriebssystem, in diesem Bereich verbessern.

## Ausführliche Darstellung der Ergebnisse

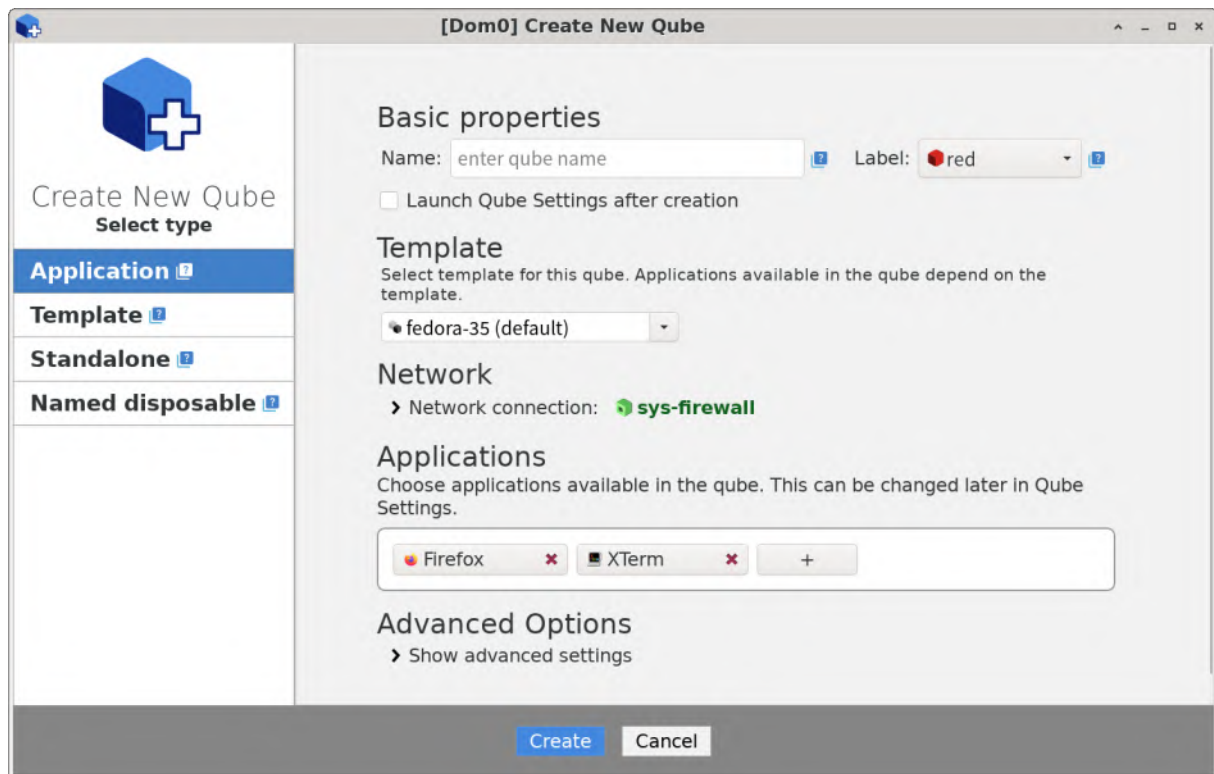
Ich habe alle geplante Meilensteile erreicht. Am Anfang habe ich eine große Umfrage und 16 ausführliche Interviews mit Nutzer\*innen geführt; die Ergebnisse davon waren etwas überraschend. Die Hauptprobleme betrafen die Zugriffssteuerungskonfiguration und Isolationskonfiguration, nicht die Programminstallation. Ich habe meinen Schwerpunkt auf diese Bereiche verlagert, und ein Konfigurationstool für Qubes OS implementiert, mit einem anderen kleinen Tool für Softwareinstallation.

Laut den Interviews, verstehen die Nutzer\*innen Zugriffssteuerungskonfiguration (die sogenannte .policy Dateien) fast nie; mit Qubes OS kann man, zum Beispiel, komplett die Zwischenablage von verschiedenen virtuellen Maschinen zu isolieren – aber wenige Nutzer\*innen können das schaffen. Verschiedene Teile von Qubes OS, die mehr Schutz und Sicherheit bieten, als z.B. *Split GPG*, ein Tool für Schutz von GPG-Schlüssel, und ein ähnliches Tool für Schutz for Zwei-Faktor-Authentisierungs-Tokens, *Qubes U2F*, werden von den Nutzern als nahezu unmöglich zu bedienen empfunden. Die Probleme mit Softwareinstallation, die ich zuvor als entscheidend hielt, sind für die Nutzer\*innen nicht so schwierig.

Auf der Grundlage dieser Umfrage und Interviews habe ich zwei Tools entwickelt: ein, das Konfiguration von Qubes OS erleichtert, und ein für die Erstellung und Konfiguration neuer virtueller Maschinen. Nach 5 Nachbefragungen mit Nutzer\*innen zu dem erstellten Design-Entwurf habe ich die Entwicklung begonnen und erfolgreich beendet.



Das Konfigurationstool (dieser Teil der Konfiguration war bisher nur durch manuelle Bearbeitung sehr komplexer Textdateien möglich)



### *New Qube Tool, mit Softwareinstallationsmöglichkeiten*

Die Projektarbeit hat mir gezeigt, wie wichtig User Research ist – meine ersten Pläne waren ganz anders als das, was die Nutzer\*innen wirklich brauchen. Auch die Begleitung durch die Open Knowledge Foundation war sehr hilfreich, insbesondere bei der Organisation der Arbeit. Die Vernetzung-Möglichkeiten waren auch hilfreich – ich habe viel Hilfe von anderen geförderten Projekte bekommen.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Qubes OS Konfigurationstool macht die Nutzung von Qubes OS möglich nicht nur für Experten, aber auch für Anfänger\*innen, Journalisten, Aktivisten usw. Qubes OS ist jetzt wichtig für viele Leuten die für offenes, sicheres digitales Gesellschaft arbeiten – jetzt ist ihre Arbeit ein bisschen leichter.

Ich denke, dass nur Open-Source Software wirklich sicher sein kann. Sicherheitslücken können ohne den Überblick der Community nicht konsequent entdeckt werden. Es gibt auch viele Möglichkeiten für weitere User Research und Entwicklung von weiteren nutzerfreundlichen Lösungen. Die Open-Source-Community hält UX/UI oft für unwichtig, aber ich glaube, dass es in diesem Bereich eine große Chance für Veränderungen und Verbesserungen gibt.

Für mich persönlich war die Möglichkeit, tiefgreifende Benutzerforschung zu betreiben und eine auf die Bedürfnisse der Benutzer ausgerichtete Entwicklung durchzuführen, sehr wichtig. Ich habe viel von den Nutzer\*innen gelernt und meine Verständnis von UX Design erweitert. Solche Gelegenheiten sind in der Open-Source-Welt selten verfügbar.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Im Großen und Ganzen bin ich sehr positiv angetan davon, wie zielstrebig mein Arbeit war und

wie wenige Irrwege ich bei der Entwicklung eingeschlagen habe. Natürlich gibt es immer mehr Bereiche, in denen Qubes OS einfacher und freundlicher sein konnte. Kleine Probleme bei Implementation, die etwas mit Gtk-Fähigkeiten zu tun haben, könnten mit Hilfe der Open-Source-Gemeinschaft überwunden werden.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Das Projekt ist verfügbar unter <https://github.com/marmarta/qubes-config-manager> ; es wird auch in der nächsten Version von Qubes OS enthalten sein.

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Ergebnisse von Nutzer\*innen Umfrage und Interviews veranlassten mich zu einer großen Änderung gegenüber den ursprünglichen Designs; daher musste ich die geplante Aufteilung von Projektförderzeit etwas ändern, um mehr Zeit für Design und UX zu haben.

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Ich habe viel Unterstützung von der Qubes Community und den anderen Entwicklern bekommen, die an Qubes OS arbeiten.



# Richtlinie zum „Software-Sprint“

## KIZervus - KI zur Erkennung vulgärer Sprache

---

### *Schlussbericht*

Zuwendungsempfänger:

Nicolas Kuhaupt

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S07 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Vulgäre Sprache im Sinne von unangemessen wird hier als Obermenge von sexistischer, rassistischer oder beleidigender Sprache verstanden. Diese tritt nicht nur auf Social Media Plattformen auf, sondern auch in Foren (z.B. Nachrichtenportale) oder in Profilen (z.B. Datingportale). Um vulgärer Sprache entgegenzuwirken, werden bisher Kommentarfunktionen für sensible Nachrichtenrubriken geschlossen oder Arbeitnehmer:innen überprüfen Texte manuell.

Durch eine auf KI basierende Erkennung kann ein Arbeitsschritt teilautomatisiert werden. Dadurch können Diskussionen geführt werden ohne vulgäre Sprache befürchten zu müssen.

Dieses Projekt wurde in drei Schritten umgesetzt:

#### 1. Datensammlung

Für das Training von KI-Modellen werden große Mengen an Daten benötigt. Diese wurden zusammengetragen und in ein einheitliches Format überführt werden.

#### 2. Training des KI-Modells

Das Training beziehungsweise Finetuning eines KI-Modells funktioniert nicht stringent nach einem Plan, sondern ist mit Experimenten verbunden. Für diesen Meilenstein wurden Experimente durchgeführt und ein Modell gefunden, das eine gute Performance bei der Klassifizierung vulgärer Sprache liefert.

#### 3. Implementierung als Service

Im finalen Schritt wurde das Modell in einen Docker Container eingebunden, der eine REST Schnittstelle bietet. Außerdem wurde eine technische Anleitung geschrieben, damit das KI-Modell mit den eigenen Daten angepasst und in die eigene Plattform integriert werden kann.

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Themenfeld des Softwaresprints war unter anderem Hate Speech. Während Hate Speech per Definition gegen Menschen gerichtet ist, umfasst die vulgäre Sprache auch Texte, die unangemessen formuliert sind und sich nicht zwangsweise gegen Menschen wendet. Damit ist vulgäre Sprache das Ausschlussziel jedes Plattformbetreibers. Adressiert werden hier jegliche online Angebote, die Veröffentlichung von User geschriebenen Text ermöglichen (Kommentarfunktionen, Chats, Profilseiten etc.) Betreiber von diesen Plattformen müssen Inhalte filtern und bei entsprechendem Inhalt löschen. Dies ist häufig noch manuelle Arbeit und kann mit einer KI unterstützt bzw. zum aktuellen Zeitpunkt teilautomatisiert werden. Wenige (große) Plattformen haben eigene Lösungen, bieten aber weder das KI-Modell, noch die zugrundeliegenden Daten öffentlich an. Daher ist hier auch der Open Source Ansatz wichtig. Auch in der Klassifizierung von Texten sind Biase und Diskriminierung präsent. Hier wird das KI-Modell inklusive Datenbasis offen zur Verfügung gestellt. Dadurch wird die viel kritisierte Intransparenz deutlich gesenkt. Gleichzeitig kann das bestehende Modell um eigene Daten ergänzt und auf eigene Bedürfnisse angepasst werden.

## Ausführliche Darstellung der Ergebnisse

In diesem Projekt wurde der größte bekannte Open Source Datensatz für vulgäre Sprache und Hate Speech zusammengetragen. Jeder andere Datensatz, der bereits Open Source war, wurde entsprechend in das Projekt integriert. Der Datensatz stammt aus diversen Quellen (Social Media, Nachrichtenkommentare und Chatnachrichten) und ist thematisch breit aufgestellt. Hierdurch sind insgesamt etwa 97000 klassifizierte Texte zusammengetragen worden.

Das trainierte Modell zeigt eine Performance von >80% Richtigkeit, d.h. 80% aller Texte werden korrekt als vulgär bzw. nicht vulgär klassifiziert. Der zugrundeliegende Datensatz besteht (durch Data Augmentation) zur Hälfte aus vulgären und nicht-vulgären Texten. Wird der Datensatz eingegrenzt und homogener gestaltet (beispielsweise aus einer der oben genannten Quellen), wird eine Performance von ~90% erreicht. Daher kann realistisch bei Finetuning mit eigenen Daten mit einer Performance von 90% gerechnet werden.

Damit das Finetuning bzw. die Adaption auf den eigenen Use Case und die Erweiterung auf die eigenen Daten stattfinden kann, ist der Trainingsprozess gut dokumentiert und einfach gestaltet. Die Daten müssen in das einheitliche Format überführt werden und das Training kann mit wenigen Schritten beginnen. Das Training findet dabei mit Parametern statt, die sich als robust erwiesen haben.

Im nächsten Schritt kann das Modell einfach deployed werden. Dafür existiert die Vorlage und eine Integration als REST Schnittstelle ist in wenigen Schritten möglich.

Ursprünglich wurde eine bessere Performance erwartet. Die Ergebnisse lassen keine vollständige Automatisierung der Spracherkennung zu. Nichtsdestotrotz kann das KIZervus Modell ein Baustein sein, der Inhalte vorfiltert. Anschließend müssen nur noch Texte mit hoher Unsicherheit der Klassifizierung (das Modell kann zusätzlich zur vorhergesagten Klasse einen Wahrscheinlichkeitsscore ausgeben) manuell überprüft werden.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Zielgruppe sind auf der einen Seite Betreiber von online-Plattformen, die Texte von Usern öffentlich machen, z.B. Nachrichtenportale, Social Media Seiten, oder Webseiten mit Chatfunktion. Das KIZervus-Repository kann als Grundlage für das eigene Modell genutzt werden. Hier sind zwei Dinge wertvoll. Erstens der Datensatz, der als Grundlage oder Ergänzung dienen kann. Es zeigt sich, dass KI Modelle besser funktionieren, je mehr Daten sie zum Training zur Verfügung haben. Zweitens ist ein Trainingsprozess implementiert, der schnell adaptiert werden kann und robust gute Ergebnisse liefert. Weiterhin kann das Modell anschließend mit wenigen Schritten produktiv gestellt werden.

Auf der anderen Seite kann das Repository für weitere Experimente genutzt werden. Wie bereits aufgezeigt, bietet das Modell bisher eine Performance von etwa 80% Richtigkeit. Aktuell gibt es schnelle Fortschritte im Bereich der Sprachverarbeitung. Mit dem Repository können schnell neue Experimente durchgeführt werden. Aber auch die Erweiterung durch neue Daten ist problemlos und schnell möglich.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Wie bereits erwähnt, wurde ursprünglich eine bessere Performance des Modells erwartet. Gleichzeitig deckt sich die Performance mit anderen Arbeiten, die eine Hate Speech Klassifizierung durchgeführt haben. Bei der Evaluierung des KIZervus Modells wurde insbesondere auf diejenigen Texte geschaut, die eigentlich vulgär sind, von dem Modell aber nicht als solche klassifiziert wurden. Hier wurden oft Metaphern, Vergleiche und Ironie angetroffen. Nach Einschätzung des aktuellen State of the Art ist es nicht möglich, diese zielsicher automatisiert zu erkennen. Es bleibt abzuwarten, ob dort bessere Sprachmodelle oder gar neue Techniken kommen werden.

Weiterhin wurde anfangs geplant, die Datensammlung durch eigens gescrapte Daten zu bereichern. Damit sind Daten gemeint, die auf Plattformen anfallen und automatisiert heruntergeladen und gespeichert werden. Diese müssten dann noch eigens als vulgär oder nicht vulgär gelabelt werden. Hier haben sich zwei Hindernisse bzw. Gründe dagegen gezeigt. Zum einen ist das Labeln der Daten nicht einfach. Dazu bestehen auch Best Practices, die unter anderem mindestens zwei Personen zum Labeln vorsehen. Damit soll die Qualität der Labels verbessert werden. Zum anderen war es deutlich schneller, bereits gelabelte Daten zu finden und zusammengetragen. Diese Daten bildeten auch eine ausreichende Grundlage für das Training des Modells.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Eine Demonstration der Klassifikation ist unter folgendem Link erreichbar:

<https://huggingface.co/KIZervus/KIZervus>

Das Github Repository ist hier erreichbar: <https://github.com/NKDataConv/KIZervus>

Bei Interesse kann eine Mail an [nk\[at\]data-convolution.de](mailto:nk[at]data-convolution.de) geschrieben werden.

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Ein Großteil des Projektes entfiel auf das Training des KI Modells. Wie bereits mehrfach erwähnt, wurde hier eine bessere Performance erwartet, sodass darauf viel Zeit verwendet wurde. Gleichzeitig kann dabei beliebig viel und beliebig lange experimentiert werden und stets eine bessere

Performance erreicht werden. Dadurch konnte der Arbeitsplan und die Zeit gut gemanagt werden. Während der Projektlaufzeit wurde der Meilenstein 3 (Deployment des Modells) vorgezogen, um sicherzustellen, dass das Projekt rund ist. Dann wurde zum Training zurückgekehrt und die Performance weiter verbessert.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Sprachverarbeitung ist im Moment ein wichtiges und verbreitetes Thema. Im speziellen handelt es sich um Large Language Modells (LLM). Das sind KI Modelle, die Sprache verarbeiten und groß im Sinne von vielen Neuronen sind. In der Forschung gibt es aktuell ständig neue Durchbrüche. Auch Startups mit LLMs sprießen aus dem Boden. Interessant und im Sinne dieses Projektes ironisch ist beispielsweise OpenAIs Copilot. Dies ist eine Autovervollständigung beim Schreiben von Code. Copilot geht über die klassische Autovervollständigung weit hinaus und kann ganze Code Passagen schreiben. Copilot basiert ebenfalls auf den Technologien, die hier im Projekt genutzt wurden (also LLMs). Ein auf LLM basierendes Modell hat also dabei unterstützt, ein anderes LLM zu programmieren.

## Richtlinie zum „Software-Sprint“

Fullscreen: Ein kollaboratives Whiteboard,  
das auch offline funktioniert

# Schlussbericht

Zuwendungsempfänger:  
Ahrend Wagner GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S08 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### Kurze Darstellung der Aufgabenstellung und Motivation

Für die virtuelle Zusammenarbeit benötigen wir immer mehr Anwendungen, die Teams und Organisationen in der Kommunikation und Dokumentation unterstützen. Dazu zählen auch Whiteboards, die virtuelle Meetings durch Visualisierungen vereinfachen.

Die Cloud-Infrastruktur hat gängige Standards für virtuelle Zusammenarbeit neu definiert: Nahtlose Synchronisierung ermöglicht Menschen im digitalen Raum die unmittelbare Interaktion. Dieser neue Standard kommt jedoch zum Preis von Leistung und Datenschutz, da Cloud-Infrastruktur heute meist nur als temporäres, kostenpflichtiges Abonnement der AWS-Infrastruktur erhältlich ist, und durch Ladezeiten an Performance verliert. Zudem kann man die gängigen Whiteboards wie Miro oder Mural überhaupt nicht offline bearbeiten.

Fullscreen ist eine “Local First” Whiteboard-Anwendung für Teams und Organisationen, die hier eine effiziente und sichere Open-Source-Alternative zu den Datensilos von Cloud-Infrastrukturen liefert. Die Anwendung synchronisiert Visualisierungen wie z.B. Haftnotizen und Diagramme für die virtuelle Zusammenarbeit, funktioniert aber gleichzeitig auch offline.

Der Kern der technischen Umsetzung von Fullscreen ist die Local-First-Architektur, ein neues Paradigma der Anwendungsentwicklung. Hierbei werden die Daten auf dem eigenen Gerät gespeichert und bearbeitet und zugleich über das Internet synchronisiert, um Zusammenarbeit zu ermöglichen. Dieses Modell vereint die Vorteile der Beständigkeit und direkten Verfügbarkeit klassischer Desktop-Anwendungen mit den flexiblen Kollaborationsmöglichkeiten der Cloud.

Da beim gleichzeitigen Bearbeiten von Daten auf mehreren Geräten immer Konflikte auftreten können, setzt Fullscreen CRDTs (Conflict Free Replicated Datatypes) ein. CRDTs sind eine neue Art von Datenstruktur, die das dezentrale Zusammenführen von

Änderungen ermöglicht, ohne dabei einen zentralen Server zur Konfliktauflösung zu benötigen.

Durch diese Technologien wandelt sich die Bedeutung der Cloud: Sie steht nicht mehr im Mittelpunkt der Datenverarbeitung, sondern wird austauschbar und dient nur noch der Weiterleitung von Informationen.

Geplant waren folgende Meilensteine:

### **M1: Architektur-Update und UX-Entwicklung**

Die Architektur ist auf Automerge und tldraw umgestellt und Ende-zu-Ende verschlüsselt. UX für Konfliktlösung und Synchronisierung wird neu entwickelt.

### **M2: Facilitation-Funktionen**

Haftnotizen, Cursor, Moderation, Timer und Agenda-Funktionen sind implementiert.

### **M3: Teams und Zusammenarbeit**

Teams, Zugang und Freigabe sind implementiert.

### **M4: Performance-Optimierung**

Gute Testergebnisse für langsame Verbindungen und Offline-Bearbeitung.

## **Beitrag des Projektes zu den Zielen der Förderinitiative**

### **„Software-Sprint“**

Unsere Hauptzielgruppe besteht aus Teams und Organisationen, die aufgrund von Sicherheits- oder Datenrichtlinien keine Cloud-basierten Lösungen am Arbeitsplatz nutzen können oder wollen und Whiteboarding-Funktionen benötigen (Haftnotizen, Diagramme, usw.). Sie profitiert von einem Produkt, das unabhängig von der Cloud-Infrastruktur funktioniert und gleichzeitig eine intuitive Bedienung anbietet.

Einerseits ist damit eine neue und innovative Software-Architektur implementiert wurden—CRDTs gibt es zwar schon seit ein bis zwei Jahren, aber bisher wurden sie kaum in Anwendungen erprobt—und andererseits gibt es nun eine Open-Source-Lösung für kollaborative Whiteboards, was zum Themenschwerpunkt der 11. Runde, „Arbeit“, einen Beitrag leistet.

## **Ausführliche Darstellung der Ergebnisse**

*Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?*

- Fullscreen gibt es sowohl als Web App (alle gängigen Browser) als auch als Native App (macOS) in der Alpha-Version.

- Die Local-First-Architektur konnte mit der CRDT *Y.js* umgesetzt werden und die wichtigsten Whiteboard-Funktionen deckte wie geplant die Open-Source-Library *tldraw* ab.
- Es wurden UX-Prozesse entwickelt und mit Probanden getestet, vor allem für Download und Synchronisation.
- Es wurde ein eigenständiges Dateiformat *.fullscreen* entwickelt.
- Whiteboard-Funktionen: Haftnotizen, Cursor, Board über URL teilen, Board als Datei speichern
- Ein eigenes Branding und eine einfache Website tragen zur Wiedererkennung des Produkts bei.

Damit haben wir M1 (bis auf E2E) vollkommen und M2 und M3 teilweise erreicht. M4 sind wir nicht angekommen, da wir mit der Weiterentwicklung warten müssen.

Eines unser ursprünglichen Ziele war, Behörden in ihrer Arbeit zu unterstützen. In unserer Nutzerbefragung haben wir erfahren, dass lokale Software-Installationen typischerweise einen langwierigen Prüfprozess durchlaufen müssen und daher dieser Einsatz sich zunächst schwierig gestalten lässt. Wir sind der Open Knowledge Foundation sehr dankbar, einen Kontakt mit dem Innovationslabor Baden-Württembergs hergestellt zu haben.

Die Vernetzung mit anderen Projekten sowie die Vorstellung beim Demo Day hat uns die akademische Zielgruppe nochmal näher gebracht. Besonders in den kreativen Disziplinen wird oft nach Whiteboard-Lösungen gesucht und es scheint eine Marktlücke für bezahlbare und offene Software zu geben.

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zunächst ist Fullscreen als Prototyp nutzbar. Wir stellen Software und Infrastruktur bereit, die für kleine Teams, die schnell eine Whiteboard aufsetzen möchten, die nötigsten Funktionen hat. Selbstverständlich können wir in der Alpha-Version keine Stabilität der Software garantieren und werden Nutzerfeedback weiter beobachten.

Als Prototyp hat Fullscreen aber vor allem eine Vorführfunktion. Während der Förderzeit haben wir Fullscreen in der Local-First-Community rund um *Ink & Switch* vorgestellt. Die Entwickler:innen von *Y.js* und *tldraw* kennen auch Fullscreen. Damit ist ein wesentlicher Beitrag für Anwendungen mit CRDTs als sog. „Proof of Concept“ für die erweiterte Community geschaffen. Die nächste Generation von Anwendungen mit CRDTs wird mit Fullscreen ein erstes, offenes Umsetzungsbeispiel für die Entwicklung und die User Experience haben. Das ist enorm.

Sofern der Prototyp weiterentwickelt wird (s. unten) wird auch ein entsprechendes Business Model entwickelt, um dieses Projekt weiter zu finanzieren. Dafür gibt es bereits erste Konzepte, die vielversprechend sind.

Für das Projektteam war die Auseinandersetzung mit CRDTs eine produktive Zeit. Die UX-Erkenntnisse werden in die offene Bibliothek *Decent Patterns* mit eingebunden, und gleichzeitig arbeitet ein Teammitglied an *p2panda*, einem neuartigen CRDT.

### **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Ursprünglich war unsere Projektidee, *tldraw* als Ausgangspunkt zu nutzen und unsere eigene Version auszubauen (neue UI und zusätzliche Funktionen). Doch mitten in der Förderzeit hat *tldraw* VC-Funding bekommen und sich für die Entwicklung einer Core-Library entschlossen. Das ist an sich eine sehr gute Nachricht für Fullscreen—denn wir können auf die Funktionen von *tldraw* direkt aufbauen—doch (1) ist die Core-Bibliothek noch nicht fertig bzw. ausgereift, und (2) geschieht die Entwicklung hinter verschlossenen Türen. Wir haben mit der zwischenzeitlichen Umstellung auf *tldraw/core* wertvolle Zeit verloren und mussten schlussendlich doch die veraltete Version von *tldraw* nutzen. Wir würden abwarten, bis *tldraw/core* in einer stabilen Version veröffentlicht wird. Entsprechend würden wir Fullscreen weiterentwickeln.

Des Weiteren haben wir unterschiedliche UX getestet. Es wurde schnell klar, dass Dateien im Browser-Kontext nicht geeignet sind, obwohl Browser wie Chrome diese UX unterstützen würde. Das haben die ersten User Tests ergeben. Wir haben uns also bewusst für die Native App entschieden, damit das Nutzungsverhalten klar ist.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Website: <https://fullscreen.space/>

Github: <https://github.com/interalia-studio/fullscreen/>

UX-Bibliothek: <https://decentpatterns.com/>

p2panda: <https://p2panda.org/>

Ink & Switch: <https://www.inkandswitch.com/>

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Wie bereits erwähnt hat uns die *tldraw/core* Umstellung wertvolle Zeit gekostet. Wir mussten wichtige Funktionen aus M2 und M3 streichen und haben uns stattdessen auf das Backend fokussiert.

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Das US-amerikanische Industrial Research Lab *Ink & Switch* hat uns strategisch und technisch inspiriert. Wir sind im regen Austausch mit der Local-First-Community und wollen auch zukünftig mitwirken und Local-First-Anwendungen—aufbauend auf CRDTs—entwickeln.



# Richtlinie zum „Software-Sprint“

## Hassreden-Tracker

### *Schlussbericht*

Zuwendungsempfänger:

Travis Robert Brown

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S09 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Wie im Vorschlag angegeben, besteht das Ziel des Projekts darin, Benutzer in die Lage zu versetzen, Muster von Belästigung und Hassreden zu erkennen, um ihre Communities zu verteidigen und die Täter zu entlarven. Das Projekt konzentrierte sich speziell auf die Erweiterung eines bestehenden Satzes von Softwaretools, die vom Autor entwickelt wurden, und darauf, diese Tools skalierbarer, robuster und über sichere Webschnittstellen für eine breitere Gruppe potenzieller Benutzer verfügbar zu machen.

#### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Welche Bezüge gibt es zu den Themenfeldern „Civic Tech“ und „Data Literacy“ des Software Sprints oder zu weiteren gesellschaftlich relevanten Zielen bzw. Lösungsansätzen?

Viele aktuelle extremistische Bewegungen haben sich im Kontext der sozialen Medien entwickelt. Die Teilnehmer löschen oft Inhalte oder ändern Benutzernamen, was es den Communities erschweren kann, sich gegen deren Aktionen zu wehren. Beispielsweise haben prominente rechtsextreme Accounts auf Twitter und Facebook ein gut dokumentiertes Muster entwickelt, um kontroverse und extremistische Inhalte an ihre Follower zu verteilen und diese dann zu löschen, bevor Moderatoren die Möglichkeit haben, darauf zu reagieren. So erreichen diese Accounts ihr Ziel, ihr Publikum zu radikalisieren und gleichzeitig die Gefahr zu minimieren, dass sie ihre Plattformen, ihre Finanzierung oder gemäßigte Unterstützer verlieren.

Die Anonymität und Kurzlebigkeit von Social-Media-Plattformen kann auch dazu genutzt werden, um Desinformationen zu verbreiten, schutzbedürftige Personen und Gruppen gezielt zu belästigen und sogar Gewalttaten zu provozieren.

Archivierungsprojekte von Organisationen wie das Internet Archive spielen eine zentrale Rolle bei der Bekämpfung dieses Verhaltens, aber diese Archive sind oft nicht so indiziert, dass Zielgruppen oder ihre Verbündeten die benötigten Informationen finden können. Das Hassreden-Tracker-Projekt soll diese Lücke schließen, indem es Indizes und Suchwerkzeuge für Datensätze bereitstellt, die sonst schwer zu navigieren wären.

#### **Ausführliche Darstellung der Ergebnisse**

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Die Projektentwicklung hatte mehrere große Themen, die vom ursprünglichen Vorschlag geleitet wurden, aber auch von den Bedürfnissen der Nutzer, die in jeder Phase Feedback gaben. Das prominenteste Produkt des Projekts ist ein Webdienst und eine Webanwendung, die einen Index historischer Twitter-Profilinformationen bereitstellen. Im Laufe des Projekts wurde dieser Webdienst von Journalisten genutzt, um wichtige Geschichten zu veröffentlichen, von anderen Plattformen (wie CounterSocial) integriert und von prominenten OSINT-Experten (Open Source Intelligence) wie Michael Bazzell vorgestellt.

Nach Rückmeldungen von Community-Mitgliedern im Juli 2022 wurde der Zugriff auf den Webdienst und die Webanwendung auf vertrauenswürdige Nutzer beschränkt, und die Authentifizierung wird über GitHub, Google und Twitter unterstützt. Als Beispiel zeigt der folgende Screenshot, wie der vertrauenswürdige Zugriff auf diese Webanwendung verwendet werden kann, um den Profilverlauf von **@OSINT\_Ukraine** zu identifizieren, einem Spam-Konto, das im Februar 2022 eine große Fangemeinde auf Twitter aufgebaut hat, indem es behauptete, über die russische Invasion der Ukraine zu berichten.

[Home](#)
[About](#)

[travisbrown](#)
[Sign in with Google](#)
[Sign in with Twitter](#)
[Sign out](#)

Welcome to memory.lol

This site is an instance of software from the Hassreden-Tracker project. Search results are limited for unauthenticated users (see this document for details).

Please contact us by email to discuss trusted access.

Twitter history search

4725638310					
The_HelpfulHand	2016-01-09	2020-09-27	Current Twitter ID	Current screen name	Wayback Machine
nitpromos	2022-02-10	2022-02-23	Current Twitter ID	Current screen name	Wayback Machine
OSINT_Ukraine	2022-02-24	2022-02-25	Current Twitter ID	Current screen name	Wayback Machine

Das Projekt entwickelte auch eine Browsererweiterung, die es Nutzern ermöglicht, Profilverläufe direkt in Twitter-Profilen anzuzeigen. Hier als Beispiel das Profil für **@libsoftiktok**, ein bekanntes Hasskonto, mit aktivierter Erweiterung, das die früheren Identitäten des Kontos zeigt.

←

**Libs of TikTok**  
 8,586 Tweets

...

✉

Follow

**Libs of TikTok**  
 @libsoftiktok

Previously: @shaya69830552 | @shaya\_ray | @chayamichik | @chayathepatriot | @economyustgo | @houseplantpotus

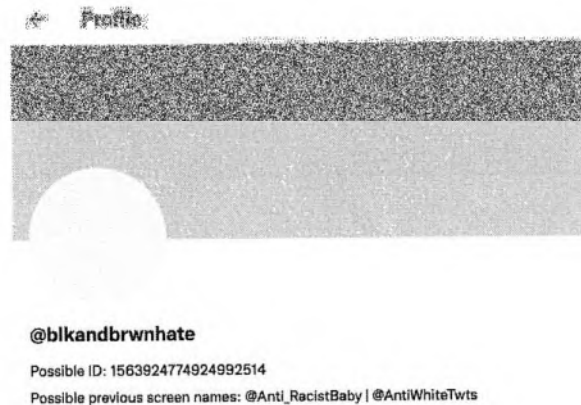
All videos belong to their respective owners. ✉ submissions@libsoftiktok.com. DM submissions.

📍 Depths of Hell 🔗 almylinks.com/libsoftiktok 📅 Joined November 2020

708 Following 1.4M Followers

Not followed by anyone you're following

Die Browser-Erweiterung liefert zusätzlich Informationen über gesperrte und deaktivierte Konten. In diesem Beispiel ein wegen rassistischer Belästigung gesperrtes Konto:



Diese Erweiterung funktioniert derzeit in den Browsern Firefox und Chrome und unterstützt nur Twitter, könnte aber für andere Plattformen erweitert werden.

Das Projekt hat auch eine Reihe von Berichten, Datensätzen und Schnittstellen entwickelt, die auf derselben zugrunde liegenden Indizierungs- und Datenerfassungs-Engine wie der Webdienst basieren. Ein Beispiel ist eine Reihe von Berichten, die Kontosperrungen nachverfolgen und seit ihrer Einführung durchschnittlich über tausend gesperrte Twitter-Konten pro Tag identifiziert haben. Diese Berichte bieten einen interessanten Einblick in die Entwicklung der Moderationsrichtlinien von Twitter, aber was noch wichtiger ist: sie unterstützen die Arbeit von Aktivisten, die gegen koordinierte Belästigung auf Twitter kämpfen, indem sie Accounts zur Umgehung von Verboten identifizieren und melden. Neben der Identifizierung von Sperren sammelt das System, das die Berichte erstellt, auch Informationen über neu erstellte Konten mit ähnlichem Verhalten und ermöglicht es in vielen Fällen, neue Konten, die von missbräuchlichen Personen und Gruppen erstellt wurden, mit gesperrten Konten zu verknüpfen.

Ein wichtiges Ziel, das in den Projektmeilensteinen skizziert wurde, war, dass die Software sowohl skalierbar als auch effizient ist. Es sollte möglich sein, große Datenmengen zu indizieren (in der Größenordnung von beispielsweise Milliarden von Profil-Schnappschüssen), aber die Kosten für die Erstellung und Bereitstellung dieser Indizes sollten minimiert werden, da das Ziel darin besteht, dass die Software von Aktivistengruppen oder anderen mit begrenztem Budget betrieben wird. Aus diesem Grund haben wir uns entschieden, den größten Teil der Datenplattform in Rust zu implementieren, einer Programmiersprache, die Wert auf Leistung legt. Insofern war das Projekt erfolgreich. Unsere Beispielinstantz baut auf einem Index von rund 34 Milliarden Profil-Snapshots auf, der auf einem einzigen Desktop-Computer erstellt wurde und seit seiner Einführung auf einer einzigen Amazon Web Services-Instanz des Typs *t4g.small* läuft, die pro Stück weniger als zwanzig Euro pro Monat kostet, um betrieben zu werden.

### Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Bisher waren die meisten Nutzer der Software Aktivisten, Journalisten und Forscher, die die Dienste und Berichte des Projekts genutzt haben, um Themen wie Verbindungen zwischen extremistischen Gruppen oder Social-Media-Aktivitäten rechtsextremer politischer Kandidaten zu untersuchen (siehe Links unten für einige Beispiele).

Diese drei Kategorien von Nutzern waren ein bedeutender Teil der im Vorschlag definierten Zielgruppen, aber das Projekt sollte auch Community-Betreuern und Einzelpersonen dienen, die von Belästigung betroffen sind, und zu diesem Zeitpunkt wurde es von diesen Gruppen weniger genutzt. Eine Ausnahme ist der „Follower Red Flag“-Bericht, der von Personen verwendet wurde, die Ziel von Belästigungen oder Massenmeldungen auf Twitter waren, um potenziell missbräuchliche Follower zu identifizieren (dies ist besonders wertvoll für Nutzer, die ihre Konten gesperrt haben und kontrollieren müssen, wer Zugang zu ihren Informationen hat).

Wir haben die meisten Elemente des Projekts nach dem Förderzeitraum weiter betrieben, einschließlich der öffentlichen Webanwendung und verschiedener Arten von Berichten, und beabsichtigen angesichts der geringen finanziellen Kosten und des minimalen Wartungsaufwands, diese Software auf unbestimmte Zeit weiter zu betreiben, unabhängig von der zukünftigen Finanzierung. Es gibt mehrere Richtungen, in die die zukünftige Entwicklung gehen könnte, einschließlich insbesondere einer zusätzlichen Indizierung anderer Social-Media-Plattformen (in der aktuellen Phase wurde hauptsächlich auf Twitter fokussiert). Angesichts der aktuellen Annahme des Projekts und der Sichtbarkeit einiger seiner Anwendungen denken wir, dass es möglich wäre, in Zukunft weitere Mittel zu akquirieren.

### **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

In den frühen Phasen des Projekts begannen wir mit der Arbeit an einer asynchronen Berichterstellungs-Engine, die auf dem Issue-Tracker von GitHub aufbaut. Der Nutzer würde ein Problem öffnen, in dem er anfordert, dass ein Bericht mit einem eingeschränkten Vokabular erstellt wird, das Problem würde von einem „Bot“ verarbeitet. Der Bot würde den Bericht generieren und das Problem schließen, indem er einen Link zur Ausgabe bereitstellt.

Dieser Ansatz hat mehrere Vorteile: Er bietet Nutzern Zugriff auf Datenanalysen, die für eine synchrone Schnittstelle (z. B. eine Webanwendung) zu rechenintensiv sind, ohne dass das Projekt eine asynchrone Berichtsplattform erstellen oder betreiben muss.

Wir haben jedoch festgestellt, dass die Anforderung, dass der Nutzer ein GitHub-Konto haben muss, zusammen mit der Schwierigkeit, mit dem Anforderungsvokabular zu arbeiten, zu einer geringeren Verbreitung des Ansatzes führ, als wir erwartet hatten. Wir haben auch festgestellt, dass viele nützliche Möglichkeiten für den Zugriff auf die Daten synchron bereitgestellt werden könnten (in einem Webdienst oder einer Anwendung). Wir haben daher die Arbeit an diesem Berichtsmodul depriorisiert, und obwohl die grundlegende Grundlage abgeschlossen ist, erfordern viele Arten von Berichten, die vom Projekt bereitgestellt werden, immer noch, dass der Nutzer Befehle über die Befehlszeile ausführt.

Wir begannen auch mit der Arbeit an einer Engine zum Sammeln von Benutzerprofilbildern von Plattformen wie Twitter, Gab und Telegram, wobei wir uns hauptsächlich auf die Wayback-Maschine als Datenquelle stützten. Diese Art der Datenerhebung hat mehrere Anwendungen im Bereich der Verfolgung von Hassreden:

- Hass-Symbole (wie das Sonnenrad oder Pepe der Frosch) können entweder manuell oder automatisiert identifiziert werden.
- Accounts zur Umgehung von Sperren können durch Bildabgleich identifiziert werden.
- Plattformübergreifende Links können ähnlich identifiziert werden.

Während wir in einem Pilotversuch mehrere Millionen relevanter Profilbilder gesammelt haben, haben wir diese Arbeit nicht mit anderen Projektelementen oder Schnittstellen integriert. Der Code ist öffentlich verfügbar und wir hoffen auf zukünftige Arbeiten in dieser Richtung.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

- Code-Repository der obersten Ebene: <https://github.com/travisbrown/hassreden-tracker>
- Hassreden-Tracker-Instanz verfolgt rechtsextreme Konten auf Twitter: <https://memory.lol/app/>
- Tägliche Twitter-Suspendierungsberichte: <https://github.com/travisbrown/twitter-watch/>
- Bericht der Washington Post, der Informationen zitiert, die durch das Projekt enthüllt wurden: <https://www.washingtonpost.com/technology/2022/04/19/lib-s-of-tiktok-right-wing-media/>
- Bericht von The News Daily über den Einsatz des Projekts bei den australischen Parlamentswahlen 2022: <https://thenewdaily.com.au/news/politics/australian-politics/federal-election-2022/2022/04/29/devotes-anti-trans-tweets-liberal/>
- Beispiel eines CNN-Artikels, der auf einen vom Projekt erstellten Bericht verweist: <https://edition.cnn.com/2022/06/23/politics/mayra-flores-capitol-riot-qanon-kfile/index.html>

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Die Arbeit sollte ursprünglich von einem Entwickler über vier Monate ausgeführt werden, aber angesichts unerwarteter Verfügbarkeit wurde die Entwicklung weitere fünf Wochen über das geplante Ende des Arbeitszeitraums hinaus fortgesetzt (ohne Änderungen am Budget).

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Nach der Steigerung der Sichtbarkeit durch die Berichterstattung in der Washington Post und die Integration des Webdienstes in ein CounterSocial-Produkt (SockWatch) äußerten einige Personen Bedenken, dass Elemente des Projekts dazu verwendet werden könnten, die Privatsphäre zu untergraben oder sogar Belästigung zu fördern. Wir hatten keine Hinweise auf einen solchen Missbrauch (und haben es bis heute nicht), aber wir haben den Zugriff auf den vollständigen Index auf vertrauenswürdige Nutzerbeschränkt, die sich über GitHub, Google oder Twitter authentifiziert haben. Nicht vertrauenswürdige oder nicht authentifizierte Benutzer haben nur Zugriff auf die letzten 60 Tage des Profilverlaufs, was ausreicht, um einige wichtige Anwendungsfälle zu unterstützen (z. B. die Identifizierung von Desinformationen), aber das Missbrauchspotenzial verringert.



# Richtlinie zum „Software-Sprint“

## TOHP – The Open Hearing Project

---

### *Schlussbericht*

Zuwendungsempfänger: Peggy Sylopp

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S10 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Etwa jede 6. Person verfügt über Hörverlust. In Deutschland sind das über 6 Mio, jedoch verwenden nur etwa 75 Prozent der Betroffenen eine Hörunterstützung.

Gründe dafür sind die immer noch anhaltende Stigmatisierung von Hörverlust, der hohe Anschaffungspreis für gute Hörgeräte, und der aufwändige Anpassungsvorgang mit oft nicht zufriedenstellenden Ergebnis.

Die wachsende Anzahl von Berufstätigen über 60 Jahre mit einem unversorgten, altersbedingten Hörverlust nimmt Folgen wie Arbeitsausfälle, Depressionen und Demenz in Kauf, mit hohen volkswirtschaftlichen Schaden. Gängige Hearables (am Ohr tragbare Geräte zur Hörverstärkung wie Kopfhörer, Headsets oder Hörgeräte) verwenden meist Audiogramme (Hörschwellen) als Grundlage zur Klangverstärkung, die jedoch nur unzureichend Informationen zu Klangpräferenzen geben, wie inzwischen wissenschaftlich nachgewiesen. Die Erstanpassung von Hörgeräten erfolgt von Hörgeräteakustiker\*innen in einer iterativen Feinanpassung im Hörlabor (Feinanpassung->Alltagstest->Feinanpassung-> ...) Oft werden keine zufriedenstellende Ergebnisse erreicht, was zur Frustration der Betroffenen führt. Klang-Selbstanpassung in individuellen Alltags-Klangumgebungen wird den Anforderungen an individualisierter Klangverstärkung gerecht und ist auf dem Hearable-Markt ein aufkommender Trend, jedoch stehen noch keine Open Source Lösungen zur Verfügung.

Die Antragstellerin verfügt selbst über einen milden bis moderaten Hörverlust und möchte mit der TOHP-App-Entwicklung diese Lücke füllen.

Die TOHP-App soll einen Beitrag dazu leisten, den Zugang zu Hörhilfen niedrigschwellig erhältlich und preiswert zu gestalten, das Verfahren der Klanganpassung zu verkürzen und individuelle Klang-Anpassung zu ermöglichen.

Aufgrund der Cross-Plattform Lösung kann der Code des TOHP-App-Prototypen sowohl für Android als auch für iOS kompiliert werden. Die App steht also sowohl für Android Smartphones als auch als iPhone zur Verfügung.

Der erste Meilenstein im Projektmonat 1 war mit der Entwicklung der App-Architektur sowie der Erstellung der Basis-Kommunikation zwischen der Audioverarbeitung und GUI veranschlagt. Weitere geplante Meilensteine schlossen die Fertigstellung der GUI mit Audio-Steuerungselementen und die Integration von Audio-Algorithmen ein, außerdem Debugging und Veröffentlichung des Codes.

### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Die Zielgruppe sind Enduser mit Hörproblemen sowie Menschen mit besonderen Ansprüchen an das Hören. Aber auch die Open Source Community soll angesprochen werden. Hierfür ist die Software-Architektur der App so konzipiert, dass sich Menschen mit Kenntnis im Audio- und Musikprogrammierung sich relativ niedrigschwellig einarbeiten und zum TOHP-Projekt beitragen können.

Enduser können den TOHP-App Prototyp zum besseren Sprachverständnis in schwierigen Hörsituationen einsetzen. Die Lautstärke und die Klangfarbe ist frei anpassbar. Für Menschen mit Audio-Expertise bietet sie ein Art Basis-Framework für eigene Modifikationen und Entwicklungen wie beispielsweise Geräusch-Unterdrückung. Aber auch Erweiterungen beispielsweise mit Audiofiltern aus dem Musikbereich sind denkbar.

Wie in der Projektbeschreibung erwähnt, ist Hörverlust eine Volkskrankheit. Da 75% der Menschen mit Hörverlust auf eine Hörhilfe verzichten, wird neben persönlich erlebter fehlender sozialer Teilhabe ein erheblicher volkswirtschaftlicher Schaden verursacht. Daher sind unbehandelter Hörverlust ein Problem öffentlichen Interesses.

### **Ausführliche Darstellung der Ergebnisse**

Die Android-App zur Anpassung des Umgebungsklangs ist umgesetzt. Die Meilensteine mussten etwas angepasst werden, da sich der Projektplan technisch so nicht umgesetzt werden konnte. Die Software-Architektur wurde neu aufgestellt, da der ursprüngliche Plan noch zu viel wissenschaftliche Abstraktion und spezialisierte Expertise im Hörforschungsbereich erfordert. Darüber hinaus wurde deutlich, dass ein Transfer des Codes der liketohear-Entwicklung (entstanden im von der Antragstellerin geleiteten wissenschaftlichen Forschungsprojekt Hear How You Like To Hear am Fraunhofer IDMT Oldenburg) in eine App nur mit erheblichen Aufwand umsetzbar ist. Sowohl die kritische Prüfung der geplanten Softwarearchitektur sowie deren Neubearbeitung war nicht als Meilenstein angelegt, was sich auf die Umsetzung der Meilensteine ausgewirkt hat. So konnten nicht alle geplanten Audio-Algorithmen und die Chat-Erweiterung implementiert werden.

Die neue Software-Architektur basiert auf JUCE, einem plattformübergreifendes Open-Source-C++-Anwendungsframework, das für die Entwicklung von Desktop- und mobilen Anwendungen verwendet wird. Das heißt der Code ist, anders als ursprünglich geplant, sowohl als Desktop-Anwendung als auch als Android oder Smartphone-App kompilierbar. Darüber hinaus ist die

Software-Architektur mit einer Plugin-Struktur ausgestattet, so dass sie selbst im laufenden Betrieb modular, sprich mit weiteren Funktionen, erweiterbar ist.

JUCE ist vorwiegend für die Entwicklung von Audio- und Musikanwendungen konzipiert. Diese Entwickler\*innenszene ist wesentlich größer als die Hörforschung und Open Source Konzepten eher zugewandt. Somit ist für interessierte Es entsteht eine steilere Lernkurve für Interessierte, die den Code verstehen und modifizieren möchten und somit einen Beitrag zur TOHP-App leisten.

Der Mehraufwand durch die Änderung der Software-Architektur ermöglicht somit ein breiteres Anwendungsspektrum, wodurch mehr Nutzende erreicht werden können. Die steile Lernkurve bei der Einarbeitung in die Entwicklung eröffnet bessere Chancen, eine aktive Open Source Community aufzubauen, die das TOHP Open Source Projekt nachhaltig lebendig hält.

Diese Erfahrung machte deutlich, dass nicht nur die Enduser in die Planung der Softwareentwicklung einbezogen werden sollten, sondern auch Aspekte, die das Open Source Projekt nachhaltig machen, wie beispielsweise die Einstiegshürde für potentiell Mitwirkende am Open Source Projekt. Darüber hinaus wurde klar, dass Inspirationen aus anderen Bereichen (in diesem Fall Audiotbearbeitung) neue Möglichkeiten eröffnen.

### **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Der unmittelbare Nutzen für die Enduser ist ein niedrigschwellig zugängliche, preiswerte Technologie, die als anpassbare Hörhilfe nach individuellen Bedürfnissen in Alltagsklumgebungen eingesetzt werden kann.

Darüber hinaus macht die durch die Software-Architektur angelegte niedrigschwellige Erweiterbarkeit die technische Anpassung und Erweiterung der App für verschiedene Anwendungen jenseits des Ausgleichs eines Hörverlusts möglich. Beispielsweise kann auch mit Audio-Effekten erweitert werden, die eine Hörerweiterung für musikalisch interessierte Menschen darstellen.

Auch nach Ende der Förderphase wird weiter an der TOHP-App gearbeitet. So werden weitere Audioalgorithmen für den Einsatz als Hörhilfe im Alltag implementiert.

Darüber hinaus ist der Einsatz der App in einem Forschungs- und Entwicklungsprojekt eines wissenschaftlichen Spin-Offs der Antragstellerin geplant. Ziel des Projekts ist die Entwicklung einer KI-Anwendung auf Basis von individuellen Klang-Anpassungen. Hierfür soll die TOHP-App mit wissenschaftlichen Erweiterungen ausgestattet werden. Damit aufgezeichnete Klangsszenen-Analyse-Daten sollen zusammen mit den Klang-Anpassungsdaten zur Entwicklung einer KI-Anwendung zur individualisierten Klang-Anpassung dienen. Die TOHP-App dient hierbei als Möglichkeit zum Sammeln von Daten. Insbesondere da sie als Open Source Projekt über eine niedrigschwellige Zugänglichkeit verfügt, besteht großes Potential für das Sammeln von größeren Datenmengen, die den potentiellen Erfolg des FuE Projekts fördern.

Durch die Arbeit an TOHP konnte die Antragstellerin ihre technische Expertise aus der Hörforschung vertiefen und um Expertise im Audibereich erweitern. Die vorgegebene Struktur des Prototypenfunds mit regelmäßigen Updates und Zwischengespräch haben geholfen, das Projekt zu strukturieren. Mit Hilfe des Coachings konnte ein Geschäftskonzept mit neuen Anwendungsfeldern und ein Call-to-Action für die TOHP-App entwickelt werden.



### **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Eine Prüfung der geplanten technischen Umsetzung der liketohear-Entwicklung erwies sich als zu spezialisiert und nicht auf die TOHP-Entwicklung übertragbar.

Die Software von liketohear ist zwar Open Source, die Forschung und Entwicklung für diese Software basiert jedoch auf der Hörforschung und ist für Studierende aus dem Audiologie-Bereich konzipiert. Ohne eine Basis-Ausbildung in der Hörforschung ist der Software-Code für Entwickler\*innen nur mit sehr flachen Lernkurve, das heißt langer Einarbeitungszeit, verstehbar.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Informationen zum TOHP befinden sich auf folgenden Plattformen:

<https://github.com/pgys/TheOpenHearingProject>

<https://theopenhearingproject.pexlab.space> (Domain wird voraussichtlich im Laufe des Oktober 2022 freigeschaltet)

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Entwicklung der Software-Architektur hat durch die zusätzliche Iteration erheblich mehr Zeit in Anspruch genommen. Auch das Debuggen war mit Mehraufwand verbunden. Um dies auszugleichen, wurde weniger Module implementiert und die grafische Oberfläche einfacher und damit weniger Zeitaufwändig umgesetzt, als zu Beginn geplant.

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Die grundsätzliche Idee zur App, eine anpassbare Hörhilfe, basiert auf der liketohear-Entwicklung. Der auf Hörforschung basierende wissenschaftliche Aufbau dieser Klangverstärkung ließ sich nicht so einfach als App umsetzen. Daher dient liketohear lediglich als Inspiration für die Funktionsweise der Klang-Anpassung des aktuellen TOHP-App Prototypen.

Aus der Audio-Entwicklungs-Community kam jedoch der Vorschlag des Einsatzes des JUCE Frameworks, dass sich als nachhaltiger erwiesen hat, da es für mehr Plattformen anwendbar ist, modular erweiterbar und einen niedrigrschwelligen Einstieg für Entwickler\*innen ermöglicht.

# Richtlinie zum „Software-Sprint“

## Open-Needs DB

---

### *Schlussbericht*

Zuwendungsempfänger:

Heinemann & Woste GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S11 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Aufgabe des „Open-Needs DB“ Projektes ist es eine einfache REST-API basierte Datenbank für Anforderungen und Spezifikation zur Verfügung zu stellen.

Diese kann benutzt werden, um für kritische SW-Projekte notwendige Dokumentationstiefe und -genauigkeit zu erzielen, indem Objekte wie Anforderungen oder Spezifikation aus anderen Projekten übernommen werden können.

Eine einfache Web-Applikation dient zum einfachen Suchen solcher Objekte.

#### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Gerade im kritischen Bereichen wie Energie oder Gesundheit erwartet der Gesetzgeber oft die Anwendung anerkannter Standards, die eine sichere SW-Entwicklung und deren Betrieb ermöglichen. Dieser Nachweis erfolgt oft über eine ausgiebige Dokumentation, dessen unterschiedliche Kapitel oft nur von Experten korrekt befühlt werden können.

Dies zu leisten sind Open-Source Projekte oft nicht im Stande, da Expertise aber oft einfach die Zeit fehlt, um diesen Aufwand treiben zu können.

Die Open-Needs DB soll er erlauben bestimmte Dokumentationselemente zu sammeln und auszutauschen.

## **Ausführliche Darstellung der Ergebnisse**

Technisch stellt die „Open-Needs DB“ heute eine SQL-basierte Datenbank zur Verfügung, die über eine REST-API benutzt werden kann.

Dabei wurde auch schon ein Anwender- und Rollenmanagement für die API implementiert, um diverse Sicherheitsaspekte zu gewährleisten.

Die SW wurde so aufgebaut, dass diese über eine Plugin-System verfügt, um eine einfach Weiterentwicklung auch durch neue Entwickler zu gewährleisten. Diese Plugin-System wird auch intern für nahezu alle Funktionen benutzt.

Zusätzlich wurde eine React-basierte Web-Applikation entwickelt, die die Inhalte der DB anzeigen kann. Auch eine Anbindung an das Schwesterprojekt „Sphinx-Needs“ wurde realisiert.

Es fanden prototypische Umsetzungen für Funktionen zur Text-Klassifizierung und Modellbildung und -überprüfung statt. Ein Release dieser Funktionen steht aber noch auch.

Mit <http://open-needs.org> wurde eine Webseite für die Organisation „Open-Needs“ erstellt, die neben Datenbank und WebApplikation und weitere kleine Tools verwaltet. Dort findet man auch sämtliche tool-spezifische Dokumentationen.

Zusätzlich wurden auch an einer Microsoft VsCode IDE Extension gearbeitet. Sie kann schon die Daten von dem Schwesterprojekt „Sphinx-Needs“ verarbeiten und darstellen. Ein Anbindung an die Open-Needs API ist in Planung.

Neben der technischen Umsetzung, wurde auch mit mehreren Benutzergruppen Featurewünsche und Umsetzungsdetails diskutiert, so das bestimmte interne Details nicht selten 2-3 mal umgesetzt wurden, um User-Wünschen zu entsprechen.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Es wurden Diskussion mit Zielgruppen geführt, die die Lösung bald innerhalb ihrer Organisation einsetzen möchten und bereit sind die Weiterentwicklung auch finanziell im Jahre 2023 zu fördern.

Diese Zielgruppen setzen schon das Schwer-Projekt „Sphinx-Needs“ parallel in mehreren Teams ein und erhoffen sich durch den Einsatz einer zentralen Open-Needs DB Wiederverwendungsbenefits für ihre Organisation.

Diese Zielgruppen beteiligen sich auch oft durch kleiner Code-Commits an der Software selber, so dass Open-Source auch ein wichtiges „Kaufkriterium“ für diese Zielgruppen ist und dem Projekt auch selbst stark hilft.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Es wurde viel diskutiert, wie man die Daten in ein Datenmodel überführt und wie stark dieses Model durch den Anwender überprüfbar sein soll. Hintergrund sind oft Prozesse in Organisationen, die eingehalten werden müssen.

Hier wurden verschiedene Lösung angeschaut und getestet: Freie Modelle, Modelldefinition per json-Schema, Modelldefinition per RDF.

Die Entscheidung liegt nun bei RDF, dessen Unterstützung in der DB aber noch nicht final umgesetzt wurde.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

<https://open-needs.org>

<https://github.com/open-needs>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Planung konnte größtenteils wie geplant umgesetzt werden.

Nur Funktionen zur automatischen Analyse der Datenbankinhalten konnten aufgrund ihrer Komplexität doch nicht wie gewohnt umgesetzt werden.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Ergebnisse anderer Stellen hatten keine Einfluss auf die Open-Needs DB Entwicklung.

---

# *Schlussbericht*

Richtlinie zum „Software-Sprint“

## Shar-E-Box Die Internet-Plattform für eMobilität

Fereydun Khanide und Marc Weis GbR

30. September 2022

---

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S12 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

## Kurze Darstellung der Aufgabenstellung und Motivation

Die Abkehr von Verbrenner-Fahrzeugen und Hinwendung zur Elektromobilität ist beschlossene Sache: das Verbrennen fossiler Stoffe trägt maßgeblich zur Klimakrise bei. Aktuell größter Hemmschuh für eine schnelle Verbreitung von Elektroautos ist die unzureichende Lade-Infrastruktur. Gerade im ländlichen Raum kommt die Erschließung für E-Mobilität kaum voran. Die Unsicherheiten bei der Reiseplanung, unnötige Wartezeiten und Umwege führen zu einer geringeren Attraktivität von E-Autos. Gleichzeitig werden private E-Ladesäulen nicht ausgelastet. Shar-E-Box erhöht für Privatpersonen, Wohnungseigentümergeinschaften, Wohnungsunternehmen, Wohnungsgenossenschaften und Bauträgern die Attraktivität der Investition in eine Anschaffung von Wallboxen: durch bessere Auslastung der Geräte sinken die umgelegten Betriebskosten; gleichzeitig werden zusätzliche Einnahmen generiert. Für Besitzer:innen von Elektrofahrzeugen erhöht die Verfügbarkeit eines wesentlich dichteren Netzes an Ladepunkten mit der Möglichkeit, Ladezeiten und -mengen verbindlich zu buchen, die Bereitschaft zur Anschaffung eines Elektrofahrzeugs.

Mit unserem Projekt wollen wir dazu beitragen, den Umstieg auf Elektromobilität zu fördern. Hierzu haben wir eine App erstellt, mit der Standorte und aktuelle Verfügbarkeiten von Wallboxen - privaten Ladestationen für Elektrofahrzeuge - auf einer zoombaren Umgebungskarte angezeigt werden können mit der Möglichkeit, einen freien Ladepunkt sofort zu buchen und vorab zu bezahlen. Ortspezifische Angebote zur individuellen Entschleunigung während des Ladevorgangs werden einbezogen. Den ökologischen Gedanken möchten wir mit Zusatzangeboten während der Ladezeit ergänzen durch Angebote zur Entschleunigung. In Kooperation mit den privaten Anbieter:innen der Ladestationen werden Informationen über örtliche Wanderwege, kulturelle Angebote, chillige Cafés etc. mit auf die App gestellt.

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Das Konzept von Shar-E-Box richtet sich an Personen, bei denen der monetäre Gedanke nicht im Vordergrund steht. Diese Auffassung wurde uns auch in Gesprächen mit potenziellen Nutzer:innen bestätigt: dass man mit dem Verleih bzw. der Miete von Wallboxen Geld verdienen bzw. einsparen kann, war weniger wichtig als die Haltung, eigene Ressourcen zu teilen und so bestmöglich zu nutzen. Und auch das Konzept des Aufbaus einer Community, die über das reine Laden von E-Autos hinaus kulturelle und Freizeit-Aktivitäten miteinander teilt, fand wie erwartet großen Anklang.



## Ausführliche Darstellung der Ergebnisse

Im Projektzeitraum konnten wir eine Web-App mit inkl. Datenbank entwickeln, die auf einem Dutzend Bildschirmmasken die grundlegenden Funktionalitäten als prototypische Anwendung bereitstellt. Im folgenden wird ein kurzer „Walk Through“ durch die Anwendung gegeben:

1. Bei Programmstart wird ein Splash Screen mit dem Shar-E-Box – Logo angezeigt und anschließend automatisch zur Anmelde-Seite gewechselt, wo bestehende Nutzer:innen sich mit Ihrem Namen und Kennwort am System anmelden bzw. Personen als neue Nutzer:innen registrieren können. Bei Letzterem wurde bewusst auf die Notwendigkeit zur Eingabe persönlicher Daten im ersten Schritt und auf einen „Opt-In“-Zwang mit Bestätigungs-eMail verzichtet, damit die Hürden für eine initiale Nutzung des Systems so gering wie möglich sind. Nach der Anmeldung / Registrierung gelangt man zum Hauptmenue



Da die beiden Nutzer:innengruppen „Besitzer:innen von Wallboxen (Anbieter:innen)“ und „Fahrer:innen von E-Autos (Konsument:innen)“ oftmals – wenn auch i.d.R. zu verschiedenen Zeiten – in ein und derselben Person zusammenfallen, bietet die App allen Nutzer:innen alle Funktionen.

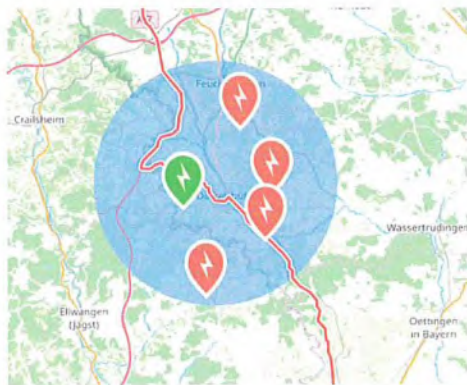
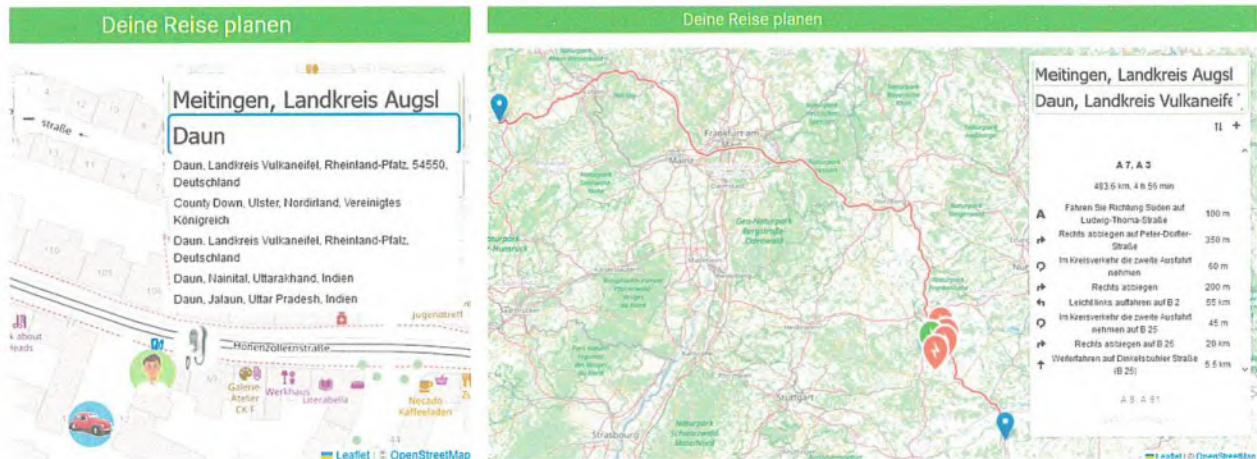
Als Fahrer:in möchte man eine Reise planen und sich anzeigen lassen, wo auf der Route in einem bestimmten Bereich (nämlich dann, wenn die Akku-Kapazität des eigenen Fahrzeugs voraussichtlich erschöpft sein wird und man den Akku wieder aufladen muss) Wallboxen befinden, die man buchen kann.

Bei Aufruf der Reiseplanung wird einem zunächst der eigene Standort, der Standort des eigenen Fahrzeugs sowie – falls man auch Anbieter einer Wallbox ist – auch der Standort der eigenen Wallbox angezeigt. Im Prototypen ist nur die Verwaltung jeweils einer eigenen Wallbox sowie eines eigenen Fahrzeugs möglich; dies soll aber erweitert werden auf jeweils mehrere Geräte.



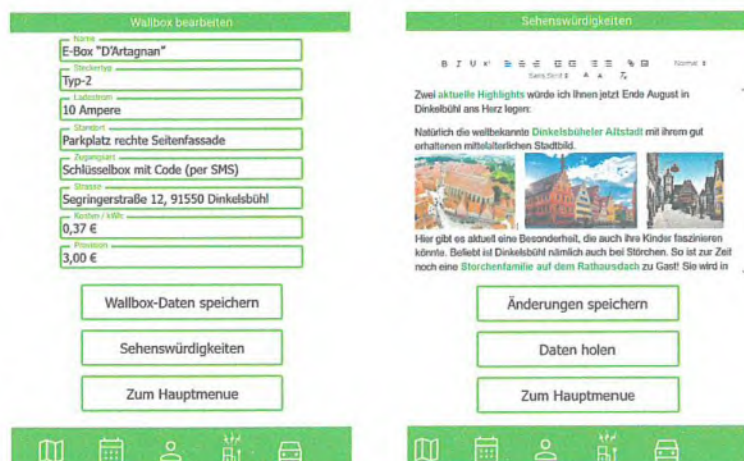
Um eine Reise zu planen, gibt man zunächst den Start- und den Zielpunkt ein. Dies kann als Texteingabe mit Ort, Postleitzahl, Straßennamen etc. erfolgen oder durch Anklicken Ortes auf der zoom- und verschiebbaren Karte. Als Routenplaner wird OSM (Open Street Map) verwendet und zur Kartendarstellung Leaflet.

Sobald Start- und Endpunkt der Reise feststehen, wird die Route graphisch und als Liste mit Fahrtanweisungen angezeigt und man kann sich für in einem bestimmten Bereich alle Wallboxen anzeigen lassen, die bei Shar-E-Box registriert sind.



Eine Vergrößerung der Karte zeigt einen Radius um den den gewählten Punkt an und alle Wallboxen, die sich in diesem Bereich befinden. Anhand der Farbcodierung sieht der/die Nutzer:in sofort, welche Wallbox zu dem voraussichtlichen Ankunftszeitpunkt frei ist und gebucht werden kann und welche Wallboxen belegt sind.

Durch Anklicken eines Symbols auf der Karte wird die entsprechende Wallbox ausgewählt und deren Daten werden eingeblendet. Neben ihrem Namen und Informationen technischer Art (Steckertyp, maximaler Ladestrom, ...) dem Standort und der Zugangsart sowie Preisinformationen kann man sich über Attraktionen und Sehenswürdigkeiten der Umgebung informieren. Dies ist sinnvoll, falls man sich für eine Buchung entscheidet und die Zeit des Aufladens sinnvoll nutzen möchte; insbesondere, wenn im Urlaub mit der Familie.





Sagen einem die Rahmenbedingungen zu und entscheidet man sich dafür, einen Zeitslot der Wallbox zu buchen, kann man sich deren Belegungsplan ansehen. Freie Termine werden durch Anklicken zunächst einmal unverbindlich reserviert (und erscheinen im entsprechenden Status auf dem Kalender des Anbieters), bis man – analog zum Verfahren eines „Warenkorbs“ beim Online-Shopping – die Wallbox endgültig und verbindlich bucht.

Buchungskalender ansehen

Alle Termine der kommenden Woche

	31.8.	1.9.	2.9.	3.9.	4.9.	5.9.	6.9.
00-01							
01-02							
02-03							
03-04						Schmidt	
04-05	Meier	Meier			Schmidt	Khanida	
05-06	Meier	Meier		Wartung		Khanida	
06-07	Meier	Meier		Wartung		Khanida	
07-08	Meier	Meier		Wartung		Khanida	
08-09				Wartung		Khanida	
09-10				Wartung			
10-11				Ableitung	Wartung		
11-12				Ableitung	Wartung		
12-13			Müller	Ableitung	Wartung	Weiss	
13-14			Müller	Wartung	Witz	Weiss	
14-15			Müller	Wartung	Witz	Weiss	
15-16				Wartung			
16-17	Khanida			Wartung			
17-18	Khanida			Wartung			
18-19			Weiss	Wartung			
19-20			Weiss				
20-21			Weiss				
21-22			Weiss				
22-23							
23-24							

Icons: Home, Kalender, Person, Liste, Fahrzeug

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Mit Shar-E-Box soll ein Marktplatz geschaffen werden, der Angebot und Nachfrage zweier Personengruppen (Besitzer:innen von Wallboxen und Fahrer:innen von Elektrofahrzeugen) zusammenführt, wobei zu erwarten ist, dass viele Nutzer:innen von Shar-E-Box beide Rollen einnehmen werden: zu gewissen Zeitpunkten sind sie Anbieter, zu anderen Zeiten (z.B. auf Reisen) treten sie als Konsumenten auf.

Nach dem erfolgreichen Start und Implementierung einer Grund-Funktionalität werden wir das System mit eigenen Mitteln über das jetzige Prototypen-Stadium hinaus entwickeln und insbesondere für eine Nutzung auf Mobilgeräten hin optimieren. Entsprechend dem Open Source – Gedanken wäre es wünschenswert, wenn das System von dritter Seite erweitert wird, insbesondere um Funktionen zur Anbindung externer Systeme (z.B. Auslesen der Zählerzustände von Wallboxen diverser Hersteller).

## Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Funktion einer konkreten Buchung von Wallboxen warf eine Reihe von Fragen technischer, rechtlicher und anwendungsorientierter Art auf wie zum Beispiel: Ist das entgeltliche Überlassen einer Wallbox ein Gewerbe? Was geschieht, wenn eine Buchung (mit/ohne Eigenverschulden) nicht in Anspruch genommen wird? Wie wird die Haftungsfrage geregelt (Schäden am Fahrzeug, an der Box, an der Elektro-Infrastruktur, an der Umgebung)? Welches Bezahlssystem soll verwendet werden, von wem wird es abgewickelt?

## Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Als Einstieg zum Gewinnen eines Überblicks über das System ist ein von uns erstellter Film unter <https://shar-e-box.com/shar-e-box.mp4> zu sehen, die Software selber findet man entsprechend unter <https://pwa.shar-e-box.com>. Die Adresse des GhitHub-Repository lautet <https://github.com/shar-e-box/shar-e-box>.

## Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Aufwand für „externe“ Verwaltungsaufgaben (Finanzamt, Gewerbe-Anmeldung, Bankverbindung, Mittelabruf) war höher als erwartet, konnte aber durch Mehrarbeit ausgeglichen werden, so dass die inhaltliche Arbeit wie geplant durchgeführt wurde.

## Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während des Projektzeitraums haben wir uns kontinuierlich über die Marktentwicklung – insbesondere im Hinblick auf technischen Neuerungen und Innovationen beim Einsatz von Wallboxen – auf dem Laufenden gehalten. Externe Personen oder Institutionen hatten aber keinen direkten Einfluss auf unsere Arbeiten und die Zielsetzung.

# Richtlinie zum „Software-Sprint“

## Threematrix - Eine Brücke zwischen Threema und dem Matrix-Protokoll

---

### *Schlussbericht*

Zuwendungsempfänger:

Moritz Stückler & Fabian Schmidt GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S13 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Matrix ist ein junges, offenes Kommunikationsprotokoll, welches vor allem als Messenger-Alternative zu Slack, Discord, WhatsApp und Co. Schlagzeilen macht. Es kann mittels sogenannter Bridges andere Netzwerke und Technologien integrieren. Matrix-Nutzer können damit alle besagten Netzwerke mit einer einzigen Anwendung bedienen. Es gibt bereits ein großes Ökosystem an Bridges, zum Beispiel zu IRC, E-Mail, Telegram, Signal, WhatsApp, SMS oder iMessage. Was noch fehlt: Eine Brücke zum – gerade im deutschsprachigen Raum – sehr beliebten Messenger Threema.

Das Ziel dieser Fördermaßnahme war die Erstellung einer solchen Messenger-Bridge zwischen Threema und Matrix. Die geplanten Meilensteine lauteten:

- Evaluierung und Test der existierenden Threema-Programmierschnittstellen und Bibliotheken
- Ggf. Aktualisierung/Bugfixing oder Neu-Implementierung der existierenden Threema-Bibliotheken
- Integration der besagten Threema-Bibliothek mit dem Matrix-Netzwerk

### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Fast jede Firma, jeder Sportverein, jede Musikschule und jede Behörde verwendet heute mobile Messenger für die schnelle Kommunikation. Durch Geheimdienstsandale und eine kritischere Zivilgesellschaft wird inzwischen häufig versucht, von den US-amerikanischen Anbietern wie WhatsApp und Slack loszukommen. Leider ergibt sich durch die vielen Alternativen wie Signal, Telegram oder Threema ein neues Problem: Fragmentierung der Kontakte. Die Aggregation und Interoperabilität dieser Messenger-Inseln ist deswegen eine große gesellschaftliche Herausforderung.

Das Matrix-Protokoll bietet sich als genau dafür hervorragend an. Organisationen wie Vereine, Behörden, Universitäten und Schulen profitieren ganz besonders von Matrix als Messenger-Basis, weil sich damit organisatorische Strukturen mit thematischen Räumen und Gruppen abbilden lassen und gleichzeitig eine globale Vernetzung mit anderen Nutzern des Matrix-Netzwerks möglich ist.

Die Stärkung von Matrix durch eine Threema-Bridge ist eine Stärkung der Wahlfreiheit in der Messenger-Welt und eine Stärkung von offenen Standards. Die Bridge würde es Nutzern ermöglichen, ihre Chats mit anderen Protokollen zu verknüpfen und die Leistungsfähigkeit ihrer Messenger zu erhöhen und aus einer geschlossenen Messenger-Welt leichter ausbrechen zu können.

## Ausführliche Darstellung der Ergebnisse

Alle geplanten Meilensteine konnten erfolgreich umgesetzt werden:

### Evaluierung und Test der existierenden Threema-Programmierschnittstellen und Bibliotheken

Threema bietet leider keine offene und kostenlose Programmier-Schnittstelle an (wie Signal, Telegram oder Matrix). Deswegen mussten sowohl die "offiziellen", als auch die "inoffiziellen" Alternativen für die Nutzung in einer Bridge evaluiert werden. Als Ergebnis wurde zunächst das kostenpflichtige, aber offizielle Threema-Produkt "Gateway API" für die Implementierung ausgewählt. Die Schnittstelle ist zwar für kleine, private Nutzer unattraktiv, weil relativ hohe Gebühren (Grundgebühren und Kosten pro Nachricht) anfallen, jedoch ist sie gut dokumentiert, und es gibt bereits einige Bibliotheken zur Implementierung dieser Schnittstelle. Die Bibliothek für die verwendete Programmiersprache Rust ist allerdings nicht vollständig in ihrem Funktionsumfang (zum Beispiel fehlt die Gruppennachrichten-Funktionalität oder die Möglichkeit, Dateianhänge herunterzuladen).

Die Alternative zur Nutzung der offiziellen Gateway-API wäre eine Implementierung einer Messenger-Bridge auf Basis der Client-Apps für Android oder iOS, deren Quellcode unter einer Open-Source-Lizenz öffentlich einsehbar ist. Eine solche Implementierung ist zeitaufwändiger, da kaum auf bestehende Bibliotheken zurückgegriffen werden kann, jedoch wäre die Attraktivität für potenzielle Bridge-Nutzer deutlich größer, da lediglich einmalig der Preis für die Threema-App (aktuell ca. 5 Euro) bezahlt werden muss.

Diese "Client-basierte"-Lösung könnte aber als "aggressiver" von Threema wahrgenommen werden, da damit das Preismodell der Gateway-API umgangen werden könnte. Über eine Matrix-Bridge sind potenziell auch umfangreiche Automatisierungs-Szenarien realisierbar, die eine deutlich größere Last auf der Threema-Infrastruktur verursachen würden als die regulären Smartphone-Apps. Die Kompatibilität von dieser Lösung mit den Threema-Nutzungsbedingungen ist ebenfalls unklar.

In einem freundlichen Gespräch mit der Threema-Geschäftsführung haben wir das Unternehmen auf diesen Zwiespalt aufmerksam gemacht. Dort zeigte man Verständnis für unser Problem und die unattraktive Preisgestaltung der Gateway-API. Eine konkrete Lösung konnte man aber auch nicht in Aussicht stellen. Denkbar wäre eine Anpassung des Gateway-Preismodells speziell für private Bridge-Nutzer, zum Beispiel durch ein separates Anmeldeformular oder eine spezielle E-Mail-Adresse. Ob und wann eine solche Änderung erfolgen könnte, ist weiterhin unklar.

## **Ggf. Aktualisierung/Bugfixing oder Neu-Implementierung der existierenden Threema-Bibliotheken**

Durch die Wahl der Gateway-API als Grundlage für die Messenger-Bridge war keine vollständige Neu-Implementierung irgendwelcher Komponenten notwendig. Für die Gateway-API existiert bereits eine Bibliothek für die Programmiersprache Rust unter dem Namen `threema-gateway-rs`, die von einem Threema-Mitarbeiter erstellt wurde.

Große Teile unserer Bridge-Lösung konnten mit dem existierenden SDK umgesetzt werden. Einige Teile mussten wir jedoch ergänzen: So sind bisher Gruppennachrichten nicht Teil des SDK. Wir haben entsprechende Erweiterungen an dem SDK vorgenommen und damit begonnen, diese im Open-Source-Sinn auch wieder in das bestehende SDK-Projekt zurückzuführen (sog. upstreaming)<sup>1</sup>.

## **Integration der besagten Threema-Bibliothek mit dem Matrix-Netzwerk**

Die Integration des Threema-SDK mit Matrix funktionierte größtenteils problemlos. Die größte Herausforderung bestand im Erlernen von Rust als Programmiersprache, mit der beide Teammitglieder bisher nicht vertraut waren. Außerdem war die Dokumentation des Matrix-Rust-SDK an einigen Stellen unzureichend.

Nach Umsetzung der Meilensteine und zum Ende der Projektlaufzeit verfügt die Threema-Matrix-Brücke nun über folgenden Funktionsumfang:

- Installierbar via Docker oder durch Compilierung des Rust-Codes
- Bidirektionales Bridging von beliebig vielen Gruppen-Räumen zwischen Threema und Matrix
- Nachrichten werden auf beiden Seiten durch einen "Bridge-User" bzw. "Bridge-Bot" gesendet und empfangen
- Unterstützung von Text- und Bildnachrichten

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Die Zielgruppe sind potenziell alle Threema- und Matrix-Nutzer, die Kontakt mit Nutzern des jeweiligen anderen Protokolls herstellen möchten. Mit einer solchen Bridge können sie Threema ohne komplizierte Web-App verwenden, Automatisierungen und Bots bauen oder Räume mit anderen Plattformen verknüpfen.

Der Nutzwert für die Zielgruppe ist aktuell durch die kostenpflichtige Threema-Schnittstelle noch eingeschränkt und unattraktiv. Die Anmeldung bei der Gateway-API kostet Nutzer eine einmalige Gebühr von 65 Euro und danach einen Betrag von 2 bis 5 Cent pro Nachricht. Eine Gruppennachricht wird dabei abgerechnet wie eine Einzelnachricht an alle Teilnehmer einer Gruppe. Eine größere Gruppe mit etwa 50 Mitgliedern zu bridgen kann damit also bis zu 2,50 Euro pro Nachricht kosten. Die Höhe und Unplanbarkeit der Kosten ist gerade für Privatanwender ein KO-Kriterium.

---

<sup>1</sup> <https://github.com/dbrgn/threema-gateway-rs/pull/62>



Eine mögliche Weiterentwicklung der Bridge wäre die Implementierung auf "Client-Basis", um damit die Umgehung der kostspieligen Gateway-API zu ermöglichen und die Attraktivität für private und kleine Anwender deutlich zu steigern.

### **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Auf Seiten des Matrix-Protokolls können Bridges als sogenannter "App Service" implementiert werden. Dafür bietet Matrix eine separate Programmierschnittstelle. Ein App Service bietet mehr Möglichkeiten, als ein einfacher "Bot-User". Mit einem App Service können zum Beispiel virtuelle User programmatisch erstellt und gesteuert werden (sog. Puppets), die dafür sorgen, dass alle Threema-Nutzer auch in Matrix als eigene Nutzer (mit eigenem Namen und eigenem Avatar) dargestellt werden können (im Gegensatz zur Zusammenfassung unter einem einzigen Bot-Nutzer).

Das zur Verfügung stehende Rust-SDK für die App-Service-API stellte sich jedoch als fehleranfällig und noch nicht sehr ausgereift heraus, so dass die Nutzung dieser Schnittstelle zum aktuellen Zeitpunkt nicht integriert werden konnte. Wir hoffen dieses Feature in Zukunft abschließen zu können, wenn die Stabilität des SDK verbessert wurde.

### **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Die zentrale Anlaufstelle für alle Informationen zum Threematrix-Projekt ist das GitHub-Repository des Projekts unter <https://github.com/bitbetterde/Threematrix>

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die veranschlagten Kosten, die sich ausschließlich aus Personalkosten zusammensetzen konnten eingehalten werden. Der geschätzte Zeitaufwand für die Implementierung erwies sich als realistisch und passend.

# Richtlinie zum „Software-Sprint“

## Claimasylum.eu – Nimm deine Rechte wahr, stell einen Antrag

---

### *Schlussbericht*

Zuwendungsempfänger:

Neugebauer-Ippen GbR

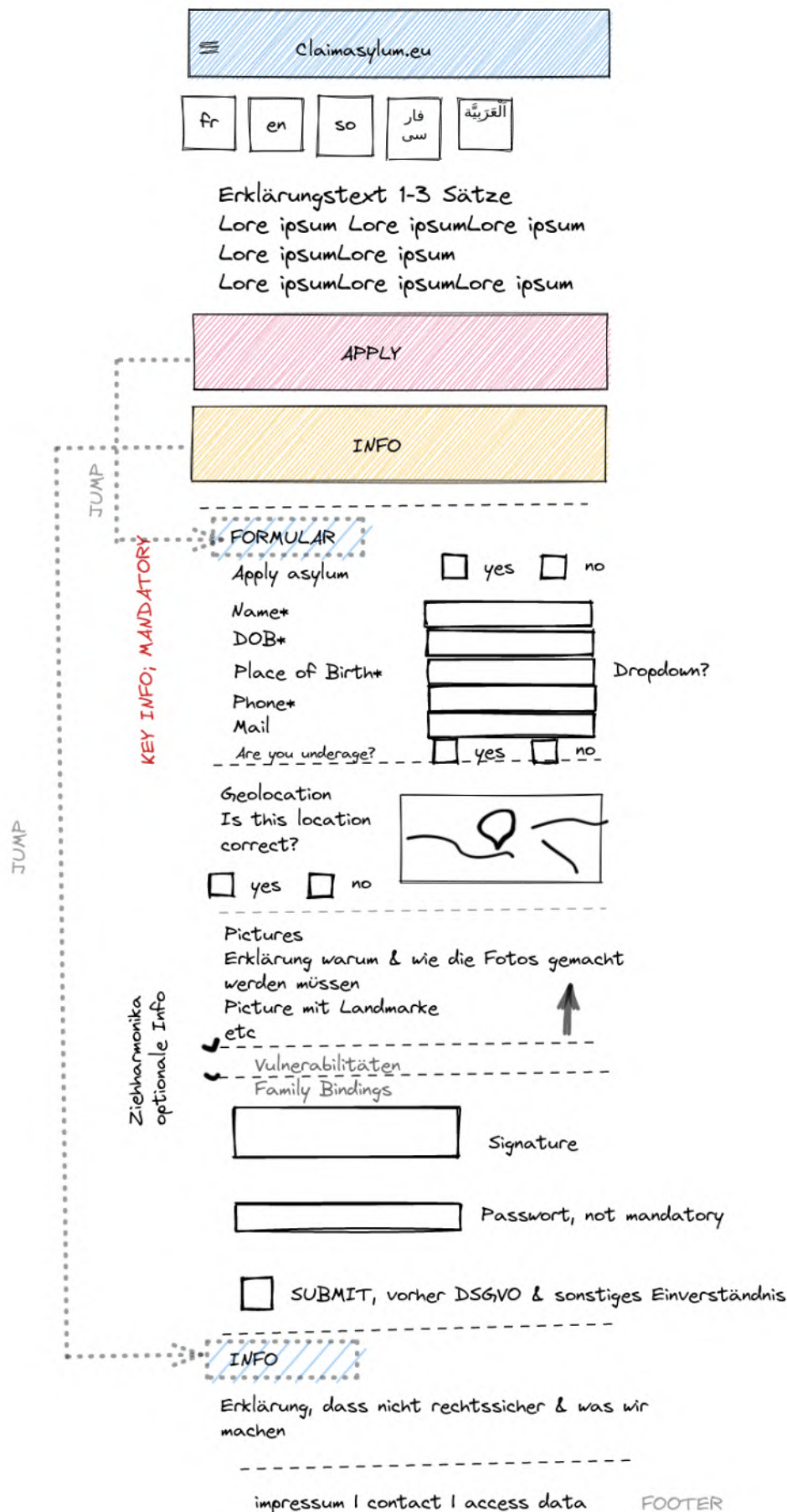
Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S14 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

#### **Motivation:**

Nach EU-Recht dürfen Menschen auf der Flucht bei ihrer Erstankunft in Europa einen Asylantrag stellen. Erst wenn dieser negativ ausfällt, dürfen diese aus Europa ausgewiesen werden. Die erschreckende Realität ist aber, dass Flüchtende immer wieder illegal und brutal aus europäischem Grenzgebiet vertrieben werden, etwa aus Polen nach Belarus oder aus Griechenland in die Türkei, ohne zuvor einen Asylantrag stellen zu können (Stichwort: Illegal Push-Backs). Mithilfe unserer WebApp können Flüchtende Menschen noch auf der Flucht, direkt nach Betreten europäischen Bodens, einen Asylantrag stellen, und somit ihr Recht auf Asyl wahrnehmen, bevor sie ausgewiesen werden. Dazu werden über ein Formular die nötigen Antragstexte generiert, an die zuständigen Behörden versandt und zusätzlich über das Speichern der Geolocation der eventuell benötigte Nachweis erbracht, dass die Person auch tatsächlich in Europa war.



## Vorgehensweise:

Wir haben deshalb eine WebApp bauen, mit derer Flüchtende einen Asylantrag einreichen können, sobald sie europäischen Boden betreten. Dazu werden über ein Formular die nötigen persönlichen Daten der Flüchtenden gesammelt und damit ein Antragstexte generiert, der dann an die zuständigen Behörden versandt. Zusätzlich soll über das Speichern der Geolocation sowie der Möglichkeit, ein Selfie vor einer eindeutigen Landmarke hochzuladen, der eventuell benötigte Nachweis erbracht werden können, dass die Person auch tatsächlich in Europa war.

Die WebApp wurde mit React im Front-End und Python und Django im Back-End entwickelt. Die ausgefüllten Formulare werden verschlüsselt an einen Mailserver geschickt. Ein weiterer Server, der Worker, greift auf den Mailserver zu und speichert die Formulare auf dem Server.



### **Meilensteine:**

- Erfassung der nötigen Formulare und Emailadressen der Behörden, Erstellung der jeweiligen Datenbanken
- Antragsformular bauen, Implementierung der digitalen Signatur und Erfassung der Geo-Location
- Erstellung von Form Types (geolocation, document-pdf, signed-selfie) und Empfänger:innen Types(Fax,E-Mail,Post), sowie von Pipelines zum Prozessmanagement
- Aufbau eines lightweight Webbackends, welches Anforderungen für MVP entspricht
- Evaluierung: Faxen nötig, falls ja, Implementierung
- Erstellung eines Systems, um Template Versionen verwalten und migrieren bzw. aktualisieren zu können und Einbindung dieser in die Web Applikation
- Übersetzung der Website in verschiedene Sprachen

### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Welche Bezüge gibt es zu den Themenfeldern „Civic Tech“ und „Data Literacy“ des Software Sprints oder zu weiteren gesellschaftlich relevanten Zielen bzw. Lösungsansätzen?

Claimasylum.eu richtet sich an unterschiedliche zivilgesellschaftliche Akteur:innen. Die Hauptzielgruppe sind Menschen auf der Flucht, um diese dabei zu unterstützen, ihr grundlegendes Recht auf Asyl wahrnehmen zu können. So soll rechtswidriges Verhalten von staatlicher Seite verhindert werden. Des Weiteren gibt es zivilgesellschaftlichen Akteur:innen, die Menschen auf der Flucht helfen, ein weiteres Tool zur Unterstützung. Schließlich trägt Claimasylum.eu auch zur Dokumentation der illegalen Push-Backs bei und macht auf diese Problematik aufmerksam. Interessierte Akteur:innen, wie Journalist:innen, Anwälte:innen oder Forscher:innen, können diese Daten weiter verwenden, um auf unterschiedlichen Wegen gegen diese Rechtswidrigkeiten anzugehen.

Damit fällt unser Projekt in die Themenfelder „Solidarität“ und „Civic Tech“ und leistet einen wichtigen Beitrag zu beiden Feldern.

### **Ausführliche Darstellung der Ergebnisse**

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Wir haben zum Ende der Förderzeit die Webapp sowie das Back-End fertig gebaut. Wir sind am Ende auf ein Problem gestoßen, das wir nicht mehr während der Förderzeit lösen konnten und an dem wir aktuell weiterarbeiten. Wir müssen die ausgefüllten Formulare an die jeweiligen zuständigen Behörden weiterleiten. In Griechenland sind jeweils die „Landkreise“ für Asylanträge zuständig. Das

bedeutet, dass abhängig von der Geo-Location der Antragsstellenden Person der Landkreis und damit die lokale Polizeistelle zugeordnet werden müssen. Wir hatten einige Probleme, an die Geodaten für die „Landkreise“ zu kommen. Es gibt zwar eine Website der griechischen Regierung, diese ist allerdings veraltet und teilweise nicht funktionsfähig. Wir probieren im Moment einen Ansatz aus, diese Daten mithilfe von Open-Street Map zu ermitteln.

Die Website ist ansonsten voll funktionstüchtig und in verschiedenen Sprachen verfügbar (Englisch, Französisch, Farsi, Arabisch, Somali). Aktuell arbeiten wir an dem Wunsch von Anwält:innen, die Website auch auf Türkisch zur Verfügung zu stellen.

Die gemeinsamen Gespräche mit anderen Projekten, die von der Open Knowledge Foundation gefördert wurden, fanden wir stets interessant und bereichernd. Wir haben viel von anderen lernen und mitnehmen können. Die Vorstellung anderer Projekte und ihre verwendeten Technologien haben dazu angeregt, eigene technologische Überlegungen zu hinterfragen und die technische Umsetzung des eigenen Projektes neu zu überlegen.

Vor allem der Demo Day war ein fruchtbarer Austausch, um auch mit anderen interessierten Akteur:innen in Kontakt treten zu können. So hat uns eine Person beispielsweise den Ansatz empfohlen, die Geodaten der „Landkreise“ mithilfe von Open Street Map zu ermitteln.

Des Weiteren hatten wir ein Coaching mit zero360 zum Thema „Projektmanagement“. Während des Coachings haben wir eine Stakeholder:innenanalyse gemacht, die bei uns jedoch eher kurz ausfiel. Danach haben wir die wichtigsten Meilensteine erarbeitet und mit diesen einen Zeitplan erstellt. Außerdem haben wir die Aufgaben den Personen mit Projekt zugeordnet. Wir haben unser Coaching sehr früh zu Beginn der Förderzeit gemacht und es hat uns sehr durch den Softwaresprint geholfen. Wir fühlten uns von Anfang an sehr gut vorbereitet und konnten immer die aktuell wichtigsten und anstehenden Aufgaben im Blick behalten.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Wie beschrieben, möchten wir mit Claimasylum.eu Menschen auf der Flucht ermöglichen, ihr grundlegendes Recht auf Asyl wahrnehmen zu können und somit rechtswidrige Push-Backs eindämmen. Der konkrete Usecase unserer Projekt ist zunächst auf die griechisch-türkische Grenze ausgerichtet. Langfristig möchten wir die Webapp auch für andere Grenzgebiete, wie etwa Polen-Belarus, erweitern.

Durch die Open-Source Stellung schafft unser Projekt über diesen konkreten Usecase hinaus ein Framework zum einfachen Abstrahieren formal-juristischer Vorgänge. Es ermöglicht, behördliche Anträge jeglicher Art einfach zu generieren. Das Stellen von Anträgen bei Behörden ist oftmals kompliziert. Häufig stellt schon die Sprache eine Barriere dar, selbst für Personen mit Deutsch als Erstsprache. Dies führt auch dazu, dass Menschen gar nicht verstehen und wissen, was ihre Rechte

sind und wie sie diese wahrnehmen können. Dies wird durch die WebApp mit Ausfüllhilfe und generierten Antragstexten verbessert.

In unserem Projekt haben wir mit verschiedenen Akteur:innen gearbeitet. Zum einen hatten wir einen stetigen Austausch mit Anwäl:tinnen in Griechenland, die Flüchtende juristisch beraten und unterstützen. Zum anderen hatten wir auch Kontakt zu unterschiedlichen zivilgesellschaftlichen Akteur:innen, die im Kontext Flucht und Migration nach Griechenland aktiv sind. Durch diesen Austausch haben wir viel neues über die Situation von Ort gelernt und erfahren und Hinweise und Vorgaben für die Webapp und den Usecase sowie die besonderen Bedingungen der User:inengruppe bekommen, die wir selber niemals alleine hätten beachten können. Dieser Austausch hat uns auf Problematiken aufmerksam gemacht, die wir selber nicht bedacht hatten.

Zum anderen hatten wir technische Unterstützung von erfahrenen Softwaredeveloper:innen und haben dadurch neue Technologien und technische Möglichkeiten zur Umsetzung des Projekt gelernt.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

Wir haben evaluiert, ob es eine Funktion zum Faxen braucht und wenn ja, wie diese implementiert werden kann. Die Anwäl:tinnen in Griechenland haben uns versichert, dass ein Asylantrag nicht per Fax geschickt werden muss, sondern per Email ausreicht. Daher haben wir Faxfunktion nicht weiter verfolgt. Es kann jedoch sein, dass wir bei einer Erweiterung der Webapp auf andere Usecases oder Grenzgebiete diese Funktion implementieren müssen.

Des Weiteren haben wir lange überlegt, ob wir die Funktion eines Benutzer:innenkontos einbauen. Letztlich haben wir uns aus verschiedenen Gründen dagegen entschieden. Zum einen sammeln wir sehr sensible Daten von einer vulnerablen Gruppe, die nicht in die Hände dritter gelangen sollten. Ein Benutzer:innenkonto bietet jedoch einen zusätzlichen Angriffspunkt, da Accounts gehackt und Daten abgegriffen werden können. Es gäbe Möglichkeiten wie sichere Passwörter und zusätzliche Maßnahmen wie 2FA. Diese sind für unsere User:innengruppe jedoch schwer umzusetzen: Menschen sind auf der Flucht und haben nur ein Smartphone. Passwörter sicher zu speichern gestaltet sich schwierig und Behörden nehmen teilweise die Endgeräte der flüchtenden Menschen ab und können somit in den Besitz dieser Passwörter kommen. Personen auf der Flucht verlieren ihre Endgeräte oder sie werden ihnen abgenommen und ihre Handynummern wechseln. Nicht alle Menschen besitzen eine Emailadresse. Zusätzliche Sicherheitsmaßnahmen zum Schutz des Benutzer:innenkontos gestalten sich also schwierig. Somit haben wir uns erst einmal dagegen entschieden, ein Benutzer:innenkonto einzurichten. Wenn Personen einen Antrag gestellt haben, können sie uns aber eine Mail schreiben und kontaktieren, um an ihre Daten zu kommen oder Infos zum Status des Asylantrags zu erhalten.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

Unser Website findet sich unter [claim-asylum.eu](https://claim-asylum.eu). Dort finden sich auch kurze Infotexte zu dem Projekt und weiterführende Links.

Unser Repository findet sich auf Gitlab unter claimasylum: <https://gitlab.com/Mareikei/claimasylum/>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Die Entwicklung des Front-Ends lief schneller als gedacht. Somit konnten wir eher mit der Entwicklung des Back-Ends beginnen, was gut war, da dieses länger als geplant gedauert hat. Wie oben beschrieben war es für uns schwierig, an die benötigten Geo-Daten der griechischen „Landkreise“ zu kommen, was somit eine Verzögerung nach sich gezogen hat.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Wir sind zunächst mit Equalrights beyond borders, einer Organisation die in Griechenland Rechtsberatung für Flüchtende und strategic litigation betreibt in die Thematik gestartet und werden die Website dort auch gemeinsam betreiben. Während der Arbeit an dem Projekt haben wir jedoch weitere Kooperationspartner gefunden die sehr interessiert an der Arbeit sind, insbesondere zu nennen ist hier das Alarmphone, ein Netzwerk das eine 24/7 besetzte Notrufnummer für Flüchtende betreibt. In der Vergangenheit hat sich das Alarmphone eigentlich vor allem mit Seenotfällen auf dem Mittelmeer und vor den Kanaren beschäftigt, aufgrund der zunehmenden Push-Backs an den Landesgrenzen zur Europäischen Union melden sich dort jedoch immer öfter Menschen, die an Land sind und Angst vor einem Push Back haben. Hier besteht großes Interesse seitens Alarmphone an der App. Ähnliches gilt für eine Gruppe, die in Polen ankommende Flüchtende aus Belarus unterstützt. Durch diese zusätzlichen Akteur:innen haben wir Feedback und Hinweise erhalten, was wir bei unserer Website und dem Projekt generell beachten müssen. Diese Akteur:innen haben uns auf mögliche Probleme aufmerksam gemacht. Außerdem helfen sie uns beim Verbreiten der Website und geben Flüchtenden Unterstützung in der Nutzung.

# Richtlinie zum „Software-Sprint“

---

## Remove NA – Knowledge Graph der LGBTIQ-Geschichte

---

### *Schlussbericht*

Zuwendungsempfängerin:

Katharina Brunner

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S15 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Remove NA erhöht die Sichtbarkeit von Daten zu queerer Geschichte, indem es analoge, queere Geschichte in verlinkte offene Daten verwandelt und sie in eine offene, frei zugängliche, gemeinwohlorientierte Dateninfrastruktur eingliedert.

Queere Geschichte wurde lange Zeit nicht strukturiert erfasst. Das Forum Queeres Archiv füllt seit 20 Jahren diese Lücken im analogen Raum. Die englische Abkürzung NA steht für "not available" und wird verwendet, wenn Informationen fehlen. Daten zur Geschichte von nicht-heterosexuellen Menschen sind häufig NA und das, was die nigerianische Künstlerin Mimi Onuoha als "missing data" bezeichnet: "Missing impliziert sowohl einen Mangel als auch ein Sollen: etwas fehlt, sollte aber existieren."

Um mit maschineller Hilfe fehlende Daten aufzuspüren, braucht es ein Positiv (den Knowledge Graph zur LGBTIQ\*-Geschichte), um durch einen Abgleich mit Linked-Open-Data wie GND oder WikiData das Negativ (die Leerstellen) aufleuchten zu lassen. Sie können dann in das System offener digitaler Informationen eingewebt werden, um die Vergangenheit für die Zukunft zu erhalten.

#### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?

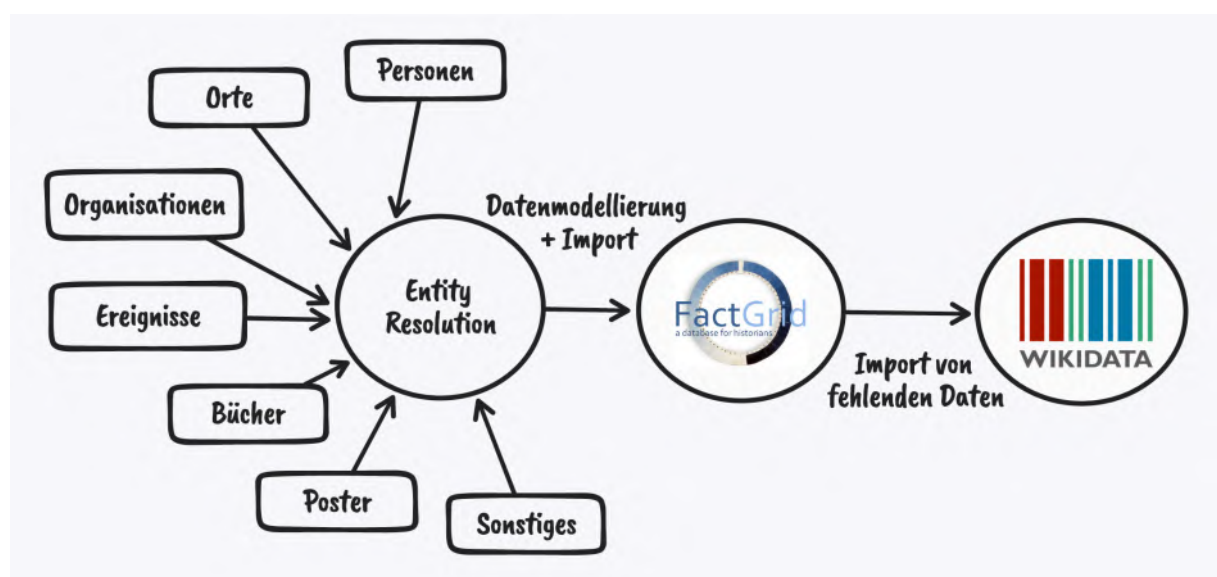
- Nutzer:innen des Archivs: Forschende und interessierte Laien, Angehörige der queeren Communities, die durch den Knowledge Graph Zusammenhänge des Bestands leichter entdecken können und möglicherweise auf hidden treasures stoßen
- Andere Community-Archive, die die technische Lösung, Dokumentation und Learnings für ihre eigenen Projekte nutzen können. Zum Beispiel ist das Queere Archiv Teil des Arbeitskreises Queer Search, in dem sich ein Dutzend ähnlicher Archive zur Digitalisierung austauscht und vernetzt
- Gemeinwohlorientierte öffentliche Dateninfrastruktur wie WikiData oder das an ein Fachpublikum gerichtete FactGrid, wo semi-automatisiert identifizierte Leerstellen ergänzt werden können, damit LGBTIQ\*-Geschichte so Einzug in verschiedenste Anwendungen und Modelle finden kann

## Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Der beste Weg um Daten und ihre Zusammenhänge und Bedeutungen für Menschen und Maschinen gleichermaßen interpretierbar zu veröffentlichen, sind verlinkte, offene Daten (Linked Open Data, kurz: LOD). Tim Berners-Lee, der Erfinder des World Wide Web, hat diese LOD als oberste Stufe eines [Fünf-Sterne-Modells](#) für offene Daten definiert.

Um mit maschineller Hilfe fehlende Daten aufzuspüren, braucht es ein Positiv (den Knowledge Graph zur LGBTIQ\*-Geschichte), um durch einen Abgleich mit Linked Open Data das Negativ (die Leerstellen) aufleuchten zu lassen. Aus dem Wissen des Forums habe ich Informationen zu verschiedenen Entitäten wie Personen, Orte, Organisationen, Ereignissen, Büchern, Postern gesammelt und diese zueinander in Beziehung gesetzt. In Zahlen: Insgesamt konnte ich 7415 Entitäten aus den Datenquellen des Forums Queeres Archiv München extrahieren, die mehr als 38 000 Verbindungen haben.



Diesen internen Knowledge Graph konstruierte ich in [FactGrid](#), das sich an ein Fachpublikum, mit akademisch-geschichtswissenschaftlichem Hintergrund richtet. Doch um sicherzustellen, dass die Daten möglichst großen, breitenwirksamen Einfluss haben, kann es nur ein Ziel in der Welt der Linked Open Data geben: Wikidata. Denn Wikidata ist die größte offene Wissensdatenbank [mit aktuell 99 Millionen Objekten](#) und das Scharnier schlechthin für externe Datenquellen. Zu jedem Objekt sind meistens nicht nur die Informationen hinterlegt, sondern auch Identifikatoren in andere Datenbanken. Jeder einzelne dieser Identifier ist ein Schlüssel in einen anderen Datensatz. Durch ein extra entwickeltes Vergleichstool, das automatisiert übereinstimmende und unterschiedliche Daten aufzeigt, konnte ich mehr als 11 200 Lücken in Wikidata schließen.

Alle Personen, die sich in diesem Datensatz zur queeren Geschichte befinden, existierten nicht für sich allein. Sie waren und sind umgeben von Menschen, die sie lieben, mit denen sie zusammenarbeiten, von denen sie verfolgt wurden, mit Menschen, von denen sie beeinflusst und solchen, die von ihnen inspiriert wurden. Kurzum: ihre Weggefährt\*innen.

Nimmt man nur die Daten aus *Remove NA*, wird man diesem Beziehungsgeflecht nicht gerecht. Erst im Zusammenspiel mit anderen Datenquellen kann man sich der Realität halbwegs annähern. Sinnvoll ist es, Wikidata und Wikipedia zu verbinden. [DBpedia](#) ist ein Knowledge Graph, der von Wikipedia abgeleitet und mehrmals im Jahr aktualisiert wird.

Ein Beispiel: Bei Magnus Hirschfeld weiß man von zwei Partnern, Karl Giese und Li Shiu Tong. Beide werden im Fließtext in Wikipedia erwähnt, doch in den strukturierten Daten ist das anders. Der eine, Karl Giese, war [nur in Wikipedia/DBpedia](#) hinterlegt, der andere, Li Shiu Tong, bis zum Juli 2022 [nur in Wikidata](#). Bis die Lücke geschlossen wurde, musste man beide Quellen abfragen.

Um die Abfrage zu persönlichen Beziehungen zwischen Menschen zu erleichtern, [habe ich eine Shiny-Webanwendung gebaut, die über Federated Queries mehrere Datenbanken abfragt](#).

Weitere Perspektiven und Abfragen auf die Daten:

- [Chronik queerer Geschichte](#)
- [München als Zentrum bayerischer LGBTIQ\\*-Geschichte](#)
- [Nationalsozialismus](#)
- [Beziehungsgeflechte](#)

Die dabei gegangenen Schritten entsprechen konzeptionell den im Antrag avisierten Meilensteinen, wenn auch zum Teil mit angepassten technischen Umsetzungen und Methoden:

- Erstellen der ersten Iteration des LGBTIQ\*-Knowledge-Graphs (LGBTIQ\*-KG) auf Basis möglichst vieler, gesäuberter Bestandsdaten des Archivs. Kernentitäten: Person, Organisation, Ort, Zeit. Record Linkage insbesondere zu Wikidata
- Speicherung und Veröffentlichung bei FactGrid
- Lücken finden zwischen FactGrid und Wikidata, also größten freizugänglichen Knowledge Graph der Welt durch einen strukturierten Abgleich und Massenimporten, beim Auffinden von Lücken



- Visualisierungen und Beispielabfragen bauen, um Perspektiven auf den Graph zu bekommen, siehe [queerdata.forummuenchen.org](http://queerdata.forummuenchen.org)

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Durch das Einweben in Wikidata stehen die Ergebnisse erstmal weiterhin für verschiedenste Anwendungen und Modelle zur Verfügung. Durch die Eingliederung in FactGrid ist ggf. auch ein Andocken an akademische Projekte leichter möglich.

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Ja, ich konnte mich durch Remove NA in das Themengebiet semantische Daten/Knowledge Graphs einarbeiten.

## Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hintergründe, und wie bist Du alternativ vorgegangen?

Ursprünglich hatte ich angedacht, mit Machine-Learning-Methoden die Struktur des Netzwerks genauer zu untersuchen. Dafür bietet sich etwa das Berechnen von Graph Embeddings an, um etwa Cluster zu finden. Machine-Learning-Methoden kann sich als recht aufwendig erweisen ohne dass die Ergebnisse wirklich weiterführen oder den Zusatzaufwand wert sind. Nach einer ersten Vorstudie habe ich ein ungünstiges Aufwand-Ertrag-Verhältnis erwartet und stattdessen auf klassische Netzwerkanalysen gesetzt.

## Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

Remove NA

Über das Projekt

Kontext ist Queen

Datenbilanz

Links und Referenzen

Perspektiven

Chronik queerer Geschichte

München als Zentrum bayerischer LGBT-Geschichte

Nationalsozialismus

Beziehungsgeflechte

Weggefährt\*innen

Remove NA

Strukturierte Daten zur queeren Geschichte

AUTORIN  
Katharina Brunner

VERÖFFENTLICHT AM  
31. August 2022


Kontext ist Queen

Das Projekt Remove NA verknüpft Data Science und Domänenwissen mit dem Ziel, queere Daten in das Netz der offenen, verlinkten Daten einzuweben.

[→ Read in English](#)

Queere Geschichte fehlt in Wissensdatenbanken

Der Mann auf dem Bild ist Max Spohr. Geboren 1850 in Braunschweig lebte er später vor allem in Leipzig, dem Zentrum der deutschen Buchwelt, Ende des 19. und im angehenden 20. Jahrhundert. Spohr war Verleger und Buchhändler - und jemand, den man heute einen LGBT-Aktivist nennen würde.



Auf dieser Seite

Kontext ist Queen

Queere Geschichte fehlt in Wissensdatenbanken

Verlinkte Daten in das Netz offenen Wissens einweben

Zahlen, bittet!

Die offene und die geschlossene Welt

Abwägung zwischen Modell und Wirklichkeit

Data Science und Domänenwissen: Es braucht beides

Daten im Kontext

Und was ist mit Max Spohr?

Erfunde die Daten

Zugang zu den Daten

Dank



- [queerdata.forummuemchen.org](http://queerdata.forummuemchen.org): Webseite mit Essay, verschiedenen Perspektiven auf die Daten und weiterführenden Informationen
- [Github-Repository](#)

### **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Nein

### **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Ja, [FactGrid](#), eine Datenbank für semantische Daten historischer Projekte und in Person Olaf Simon, war elementarer Teil für den Erfolg des Projektes. FactGrid ist an ein Fachpublikum mit akademisch-geschichtswissenschaftlichem Hintergrund gerichtet. Der interne Knowledge Graph von Remove NA ist dort Teil der gemeinwohlorientierten und öffentlichen Dateninfrastruktur.

# Richtlinie zum „Software-Sprint“

## BT-transientrefugees - Erweiterung von Boxtribute zur Unterstützung von mobilen Einsatzszenarios

---

### *Schlussbericht*

Zuwendungsempfänger:

Daniel Spaude und Ewa Flis GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S16 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

#### **Hintergrund: Was ist Boxtribute?**

Zahlreiche, oft kleine und mittelgroße Organisationen widmen sich der Bewältigung der humanitären Herausforderungen welche in Verbindung mit Flucht und Vertreibung auftreten: geflüchteten Menschen mangelt es oft an grundlegend nötigen Dingen wie z.B. angemessener Kleidung, Hygieneartikeln und teils auch an ausreichender Nahrung.

Die Open Source Software Boxtribute, deren erste Version im Rahmen der Geflüchteten-Krise 2015 von freiwilligen Helfern entwickelt wurde, adressiert logistische und operative Herausforderungen sowie die Anforderungen an adäquate datenbasierte und dennoch datenschutzkonforme Einsichten speziell kleinerer und mittlerer Organisationen, welche das Beschaffen, das Inventarisieren und die auf möglichst großer Wahlfreiheit der Geflüchteten basierende Ausgabe entsprechender Hilfsgüter betreiben.

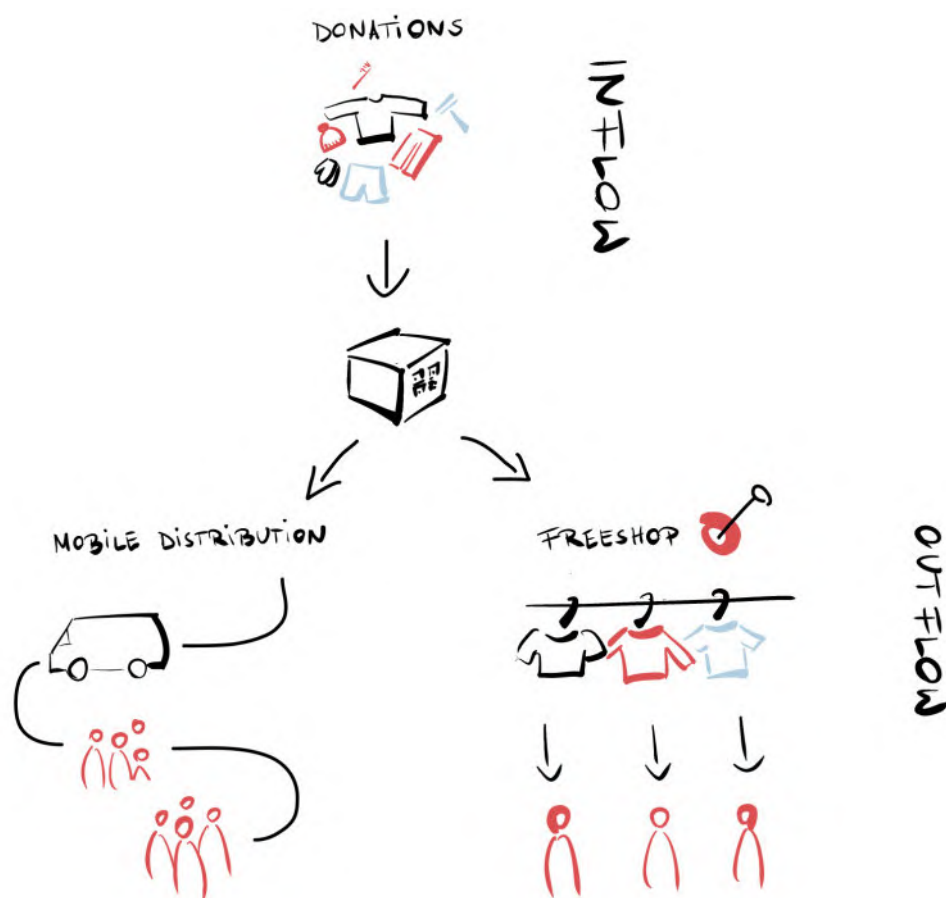
Dabei stehen ein QR-Code-basiertes Inventar-Management - wobei nicht auf individueller Artikel-Ebene, sondern auf Ebene von Boxen, welche mehrere Artikel der gleichen Art enthalten (z.B. "100 T-Shirts für Männer, XL") gearbeitet wird - und Features für das Betreiben sogenannter Freeshops (Ausgabe an die Geflüchteten/Bedürftige) im Fokus der Software.

Die Software kann prinzipiell von jedem selbst betrieben werden, jedoch haben sich fast alle Partnerorganisationen bisher dafür entschieden, das angebotene Hosting mittels Boxtribute zu nutzen.

#### **Was war das Ziel im Rahmen der Förderung?**

Die bisherige Version der Software konzentriert sich hinsichtlich der Ausgabe der Hilfsgüter auf die erwähnten stationären Freeshops: dort können die Geflüchteten sich einen Termin reservieren und dann mittels einer Art virtueller Währung jene Produkte aussuchen, welche für sie am relevantesten sind.

Im Rahmen der Förderung wollten wir einerseits Einsichten gewinnen, wie Boxtribute erweitert werden könnte um auch nicht-stationäre Versorgung besser zu unterstützen: hinsichtlich der Anforderungen von Organisationen, welche keine festen Ausgabeorte (Freeshops) betreiben, sondern welche im Kontext von “transient refugees” Hilfe leisten, welche also Geflüchtete adressieren, die sich auf ihrer Reise vom Herkunftsort zu ihrem Zielort befinden (z.B. auf der Balkanroute). Diese Menschen sind teils sogar noch vulnerabler und manchmal noch größeren menschenrechtlich relevanten Gefährdungen (z.B. gewalttätigen polizeilichen Übergriffen, kein Zugang zu grundlegender medizinischer Versorgung und Nahrung etc) ausgesetzt, als Geflüchtete, welche zumindest über einen gewissen Zeitraum über einen einigermaßen stabilen und teils staatlich anerkannten Aufenthaltsort verfügen.



Diese Organisationen organisieren dabei regelmässige Ausgabe-Ereignisse (oft mehrere pro Tag), womit das Anfahren der Organisationen bekannter, aber oft abgelegener Orte, wo sich Geflüchtete auf ihrer Durchreise aufhalten, mittels Transportern und die dortige Vergabe von Hilfsgütern gemeint ist. Dabei werden nicht nur Kleidung, sondern auch Nahrung sowie Hygiene-Artikel ausgegeben und einige Organisationen bieten auch das Duschen innerhalb der Transporter eingebauter Duschkabinen sowie das Aufladen elektronischer Geräte an.

Andererseits wollten wir die gewonnenen Erkenntnisse für solche Erweiterungspotentiale prototypisch umsetzen und mit mindestens einer Partnerorganisation, welche auf der Balkanroute täglich zahlreiche Geflüchtete unterstützt, testen.

Einhergehend damit sollte auch das Fundament für die neue Version von Boxtribute, welche auch die bereits existierenden Features des alten PHP-Stacks beinhalten soll, hinsichtlich des Frontends ausgebaut werden.

### **Ursprünglich geplante Meilensteine**

Die vor der erwähnten User Research-Phase zu Beginn der Förderung skizzierten Meilensteine sahen folgendes Vorgehen vor:

- M1: Design und fokussiertes User Research vor Ort
- M2: POC (Proof Of Concept) "Mobile Distribution Station" (Inventar-Tracking vor Ort bei der Ausgabe, Wahlfreiheit für Geflüchtete ermöglichen ohne logistische Probleme zu verursachen)
- M3: POC "Virtuelle Warteschlange und Erfassung der zu erwarteten Bedarfs der transient refugee Population"
- M4: POC "QR Code Registrierung" (Stichwort: Datenschutz, Gerechtigkeit, Rechenschaft und Vertrauen - z.B. "Wie wird sichergestellt, dass Güter gerecht verteilt werden?")
- M5: Integration; ausgiebiges Testen/QA des fertigen Prototypens; Bugfixing/Anpassungen; Planung nächster Schritte nach der Förderphase

### **Angepasste Meilensteine**

Ausgehend von den Einsichten der vierwöchigen User Research-Phase vor Ort in Serbien bei einer unserer Partnerorganisationen (davon haben wir zwei Wochen als vollwertige Volontäre die Organisation bei der Arbeit im Lagerhaus sowie bei den eigentlichen Fahrten zu den Geflüchteten und der Ausgabe der Hilfsgüter an diese unterstützt) haben wir die ursprünglich geplanten Meilensteine angepasst: statt direkt einen Fokus auf digitale Unterstützung während der eigentlichen Ausgabe-Situation zu legen (eine oft sehr dynamische Situation deren konkreter Ablauf auch von Organisation zu Organisation sehr unterschiedlich sein kann), haben wir uns entschlossen, die Prototyping- und Entwicklungs-Bemühungen auf die Vorbereitungs- und Nachbereitungsphasen der eigentlichen Auslieferungsfahrten zu konzentrieren.

Dem lag die durch die Koordinatoren der Organisation bestätigte Einsicht zugrunde, dass Features für diese Phasen

1. sowohl die operativen Prozesse vereinfachen (auch hinsichtlich einer schnelleren Einbindung der oft wechselnden Freiwilligen)
2. als auch die Datenqualität hinsichtlich der tatsächlich ausgelieferten Produkte (Produktart, Grösse, Zielgruppe und Anzahl) verbessern würden. Dies ist sowohl bei der Beschaffung und strategisch vorausschauenden Planung, einem besseren Verständnis zentraler demografischer Daten (wie Populationsgröße und Anteil von Männern, Frauen und Kindern), als auch hinsichtlich der Kommunikation mit Spendern und Unterstützern der Organisation relevant.

Dabei war ein Ziel, das Prototyping bereits von Anfang an direkt innerhalb der neuen Version von Boxtribute umzusetzen, was zwar teilweise mehr Koordination und Abstimmung und ein langsames Entwicklungstempo bedeutete, bei uns jedoch gleichzeitig ein stärkeres Vertrauen darin schuf, dass die entwickelten Features auch innerhalb der bestehenden Datenbank (Struktur und Daten - die Datenbank ist sowohl für die bestehende, seit mehreren Jahren im Produktivbetrieb befindliche als auch die neue Anwendung die gleiche) und API, welche bereits in der vorherigen Förderungsrunde

erheblich erweitert wurde, funktionieren würden. Zusätzlich erhofften wir uns davon ein leichteres und realistischeres Testen mit einer unserer Partnerorganisationen in Serbien, da diese bereits mittels der bestehenden Boxtribute-Version ihr Inventar zumindest während des Wareneinganges und der Lagerung verwaltete.

Die neuen Meilensteine, welche zum größten Teil keine strikte Sequenzierung mehr vorsahen sondern teils parallel bearbeitet wurden, sahen dann folgendermaßen aus:

- Konzeption zur Erweiterung der Datenbankstruktur und der API zur Unterstützung von Ausgabeorten und -Ereignissen
- Konzeption der Features und der User Experience sowie deren Umsetzung, sowohl im Frontend als auch im Backend: Unterstützung der Phasen Planung, Packen im Lagerhaus, Rückkehr von dem Ausgabeereignis zum Lagerhaus in der Software
- Umsetzen von in der bestehenden Boxtribute-Version vorhandenen Kern-Features wie z.B. die Box-Übersicht und Detail-Seite für Boxen in der neuen Version, ein für mobile Geräte optimiertes UI-Grundgerüst und -Menüstruktur, sowie die Integration eines QR-Scanners direkt in die Anwendung, statt der Verwendung eines externen QR-Scanners
- Regelmäßiges Nutzer-Feedback mit mindestens einer Partnerorganisation (Serbien)
- Iterationszyklen (Erweiterungen der Implementierung und Nutzer-Feedback sowie aktualisierte Anforderungserhebung)
- Nach finaler Entscheidung/Anpassung: Rollout in die Produktivumgebung und weiteres Live-Testing

## **Beitrag des Projektes zu den Zielen der Förderinitiative**

### **„Software-Sprint“**

Das primäre Ziel von Boxtribute und auch des konkreten geförderten Projektes besteht in der Unterstützung von vulnerablen Menschen in Notsituationen, primär von Geflüchteten. Der Schwerpunkt der Förderung bezog sich auf die oben bereits erwähnte spezifische Teilgruppe von geflüchteten Menschen, nämlich jene, welche sich nicht in "regulären" Unterkünften aufhalten und sich daher oft in sogar noch instabileren Verhältnissen befinden.

Dabei adressiert Boxtribute die Geflüchteten nicht unmittelbar, sondern mittelbar, indem wir kleinere und mittelgrosse Organisationen, welche die eigentliche Unterstützungsarbeit (Ausgabe von Hilfsgütern an Geflüchtete) leisten, durch die Bereitstellung unserer Open-Source Lösung in ihrer Arbeit unter anderem hinsichtlich operativer Effizienz und mit mehr Datenverständnis unterstützen.

## **Ausführliche Darstellung der Ergebnisse**

Die Ergebnisse lassen sich im wesentlichen in folgende Gruppen zusammenfassen:

1. Prototypische Umsetzung neuer Features, welche spezifisch den Anwendungsfall mobiler Ausgaben adressieren (siehe "Motivation"), für einen Test mit einer Partnerorganisation
2. Implementieren des Fundaments für das neue Frontend und die prototypische Übertragung erster zentraler Features aus der alten PHP-Anwendung (z.B. Boxes List View, Box Detail View und Box Edit View) sowie neuer Features wie die Möglichkeit, Boxen mittels Tags zu gruppieren
3. Prototyp eines direkt in die Web-App integrierten QR Scanners

4. Die Einsichten aus der User Research-Phase am Anfang der Förderung stellen ein weiteres wertvolles Ergebnis dar, welches Boxtribute bei der weiteren Priorisierung hinsichtlich der Entwicklung weiterer Features und der Produktausrichtung hilft




Die von der Open Knowledge Foundation vermittelten Workshops zum Thema User Research, Design Thinking und Testing, sowie der regelmäßige Austausch mit dem Projektträger DLR und der Open Knowledge Foundation haben uns überaus wertvolles, Feedback sowie Orientierungs- und Vernetzungswissen vermittelt.

### **Ergebnisse aus Nutzersicht**

Aus Nutzersicht lassen sich die neuen, auf mobile Ausgaben ausgerichteten Features wie folgt beschreiben:


- Anlegen und Verwalten von Ausgabe-Orten
- Anlegen von Ausgabe-Ereignissen
- Anlegen einer Packliste (zu packendes Produkt in der gewünschten Größe, der Zielgruppe - also z.B. Männer, Frauen, Unisex, Kids, sowie die zu packenden Anzahl) für ein Ausgabe-Ereignis
- die Möglichkeit, während der Vorbereitung eines Ausgabe-Ereignisses / des Packens, ganze Boxen (welche bereits im Boxtribute-Inventarsystem erfasst sind) oder einzelne Artikel aus Boxen im System zu einem Ausgabe-Ereignis zuzuordnen; dabei zeigt die Anwendung (welche wegen der Verwendung im Lagerhaus auf Mobilgeräten gut funktionieren muss) jene Einträge auf der Packliste als grün an, welche bereits den Zielwert hinsichtlich der Menge erfüllt haben, die noch nicht erfüllten Einträge werden rot markiert und stehen am Anfang der Packliste
- die Möglichkeit, nach dem Ausgabeereignis (bei Rückkehr zum Lagerhaus) Produkt-Rückläufer wieder zu erfassen, damit daraus Statistiken über die tatsächlichen Warenausgänge. Dabei muss es auch die Möglichkeit geben, die Rückläufer mehrerer Ausgabe-Ereignisse zusammen erfassen zu können, da aufgrund der oft dynamischen Abläufe während der Ausgabe-Fahrten nicht garantiert werden kann, dass sich Produktkontingente für verschiedene Ausgabeereignisse am gleichen Tag räumlich vermischen.
- die Möglichkeit, die erfassten Daten über Produkt-Ausgänge und -Rückläufer als CSV-Dateien zu exportieren

Nachfolgend finden sich exemplarisch einige Screenshots der erwähnten neuen Features (es handelt sich dabei um den Prototyp unter Verwendung von Testdaten).


BOXTRIBUTE

New Distro Event


**Distribution Spot**

Subotica 

**Date of the event:**

05.10.2022 


**Time of the event:**

14:00 

**Expected duration (in hours):**

2

**Name of the event:**

Subotica 

**Select Products for Packing List**

☐ Only show products in stock

**Men** Women Boy Girl

UnisexKid UnisexAdult

UnisexBaby No Gender

**Underwear / Nightwear**

☒ Socks




☒ Underwear

☐ Nightwear / Pyjamas

**Bottoms**

☐ Jeans

☐ Trousers


BOXTRIBUTE



Subotica  
Wed Oct 05 2022 (14:00 - 16:00)

1. Planning 2. Packing 3. On Distribution 4. Returned

Copy Packing List from Previous Event




Select Products for Packing List

**Packing List**

Long Sleeves (Men) X

SIZE	NO. OF ITEMS
S	3

✓ Successfully updated packing list entry X


BOXTRIBUTE



Subotica  
Wed Oct 05 2022 (14:00 - 16:00)


1. Planning 2. Packing 3. On Distribution 4. Returned

Scan Boxes for this Distro Event


**Packing List**

Long Sleeves (Men)	^
5 x M	+
3 x S	+
Shorts (Men)	v
Underwear (Men)	v

Add additional items to this Distro Event


BOXTRIBUTE




Scan the box X




Find Box by Label

Show fitting boxes




BOXTRIBUTE

Boxes
Mobile Distributions
Admin
SA


Box 614569

State: Donated

Long Sleeves

# 62 M

Women

Scrap

Lost

+

-

Move this box from Shop to:

LOST

SCRAP

Stockroom

WH

WH Women

WH Men

WH Children

WH Babies

WH Shoes

WH New arrivals

WH Hygiene

WH Seasonal

Assign this Box to Distribution Event:

Distro Spot 1

Fri Sep 30 2022 (10:30 - 12:30)

Planning

Assign

Distro Spot 1

Fri Sep 30 2022 (10:30 - 12:30)

Packing

Assign

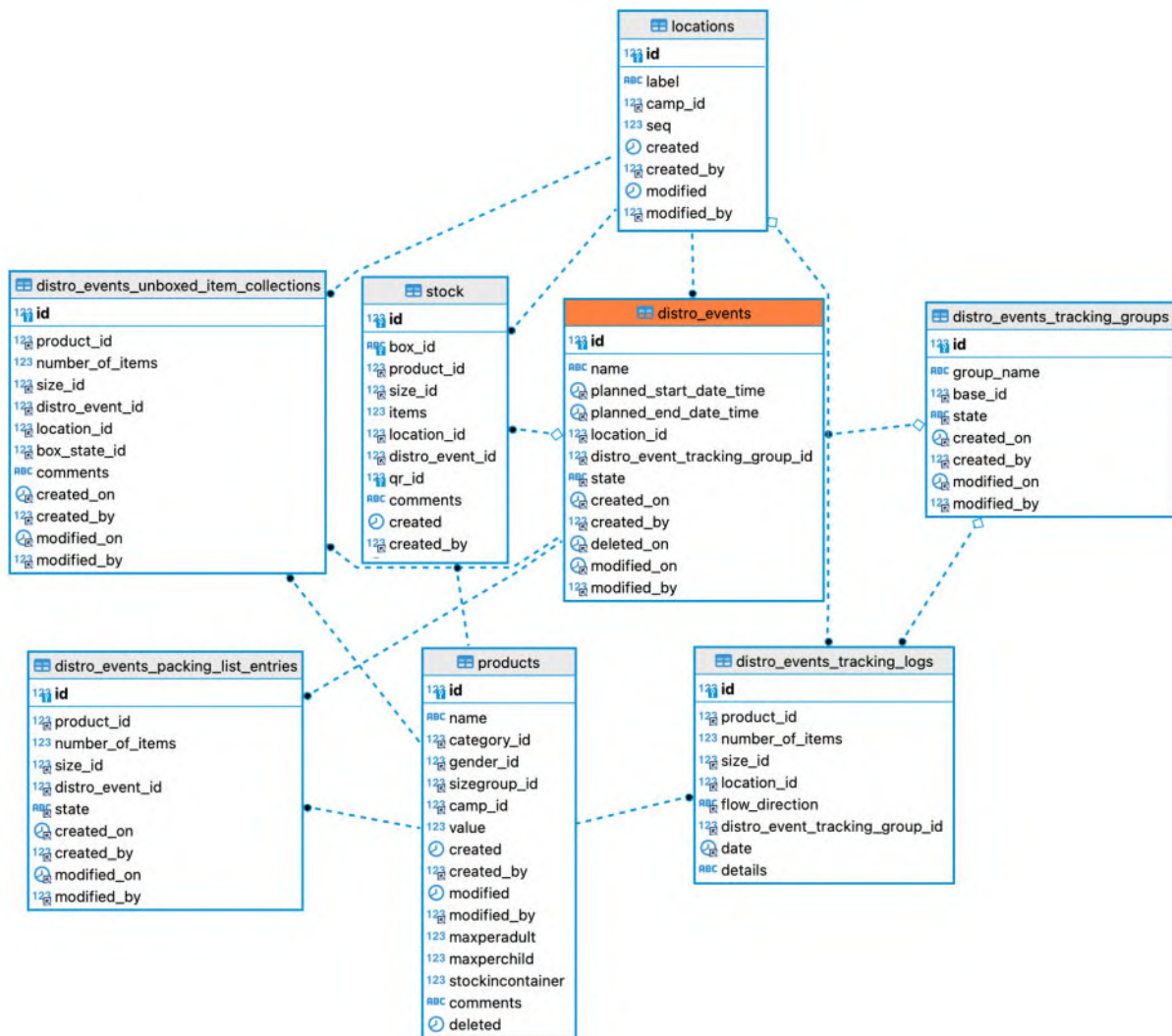
productName	sizeLabel	genderLabel	categoryLabel	outflow	inflow	earliestPossibleDistr	latestPossibleDistroD	potentiallyInvolvedD	involvedDistributionE
TShirt	M	Female	Clothing	40	20	2022-10-10 8:30:00	2022-11-15 11:30:00	Spot 1	Spot 2
TShirt	L	Female	Clothing	300	9	2022-10-10 8:30:00	2022-11-15 11:30:00	Spot 1	Spot 2
Jeans	M	Male	Clothing	50	5	2022-10-10 8:30:00	2022-11-15 11:30:00	Spot 1	Spot 2
Jeans	M	Male	Clothing	10	5	2022-09-10 16:00:00	2022-09-10 18:00:00	Spot 1	7
Jeans	M	Male	Clothing	10	5	2022-09-10 5:21:00	2022-09-10 17:19:00	Spot 1	8

Auch die erwähnte Umsetzung eines direkt in die Anwendung integrierten QR-Scanners, welcher nicht nur für die Features im Rahmen mobile Ausgaben, sondern für Arbeitsprozesse im Lagerhaus allgemein von Bedeutung ist, konnte als Prototyp erfolgreich umgesetzt werden.

## Ergebnisse aus softwaretechnischer Sicht

Neben den oben bereits aufgezählten Ergebnisgruppen, lassen sich folgende konkrete softwaretechnischen Artefakte als Ergebnisse benennen:

- die Erweiterung der neuen Boxtribute-GraphQL-API um diverse Methoden zur Abfrage, Erzeugung und Änderung von neuen Entitäten wie Ausgabeorten, Ausgabeereignissen, unverpackte Produktsammlungen (welche sich nicht in Boxen befinden, sondern teils lose im Transporter der Organisationen transportiert werden) etc welche zentral für die beschriebenen neuen Features sind
- Erweiterung der vorhandenen Datenbankstruktur (siehe nachfolgenden kleinen Teilausschnitt aus dem Datenbankdiagramm von Boxtribute - die Tabelle *distro\_events* ist eine der neuen Tabellen und stellt das Zentrum der Beziehungen zu anderen sowohl neuen als auch bereits vorhandenen Tabellen dar)



Zusätzlich haben wir im Frontend unter anderem folgende Erweiterungen vorgenommen:

- die Einführung einer verbesserten, auf UI-Komponenten konzentrierte Entwicklung mittels des Open Source-Tools Storybook.js
- die Evaluation und Einführung der Open Source Komponenten-Bibliothek Chakra UI
- Verbesserungen an der Integration der GraphQL-Bibliothek apollo.js (unter anderem das automatische Generieren von TypeScript-Typdefinitionen anhand des GraphQL-Schemas)
- die Evaluation und Einführung der Datenvalidierungs-Bibliothek zod.js

## User Research mittels NoCode-Tools und deren Potential

Zusätzlich haben wir im Rahmen unseres User Research mit anderen Organisationen, welche mit anderen Ausgabeformaten arbeiten, einen Prototypen entwickelt, welcher es den Geflüchteten ermöglicht, direkt über ein erweitertes Formular im Voraus die gewünschten Produkte zu bestellen (Pre-Ordering). Mit dem Ziel, schnelles Testen von frühen Feature-Ideen und Annahmen zu ermöglichen, haben wir diesen Prototypen auf Grundlage eines NoCode-Tools, welches auf die Erstellung und Verarbeitung komplexe und dennoch intuitiver Formulare spezialisiert ist, umgesetzt. Selbst diese frühe Version hat sich jedoch bereits als große Verbesserung hinsichtlich Arbeitsaufwand und Vermeidung von Missverständnissen bei der Bedarfserfassung erwiesen, dass dieser Prototyp nach wie vor produktiv bei einer Organisation im Einsatz ist. Ideen für zukünftige

Entwicklungsschritte, welche auch von dieser Organisation als sehr wünschenswert deklariert wurden, sind in diesem Kontext unter anderem eine direkte Anbindung an das Boxtribute-Inventarsystem, um den Geflüchteten nur jene Produkte anzuzeigen, welche momentan tatsächlich verfügbar sind, sowie eine nahtlose Integration hinsichtlich Vorbereitung der Auslieferung und der Erfassung von statisch relevanten Daten (Bedarf, Warenausgang, Warenrückläufer).

Das gleiche NoCode-Tool haben wir auch für einen Feldtest vor Ort in Serbien mit unserer primären Testorganisation genutzt: das Ziel hierbei war es, eine leichtere und geordnete Abwicklung der Ausgaben vor Ort zu gewährleisten. Im Gegensatz zum regulären Ausgabe-Modus der Organisation, bei welchem den Geflüchteten die verfügbaren Produkte direkt in Boxen vor Ort gezeigt bekommen (Mass Distribution) - was insbesondere bei größeren Gruppen zu teilweise unübersichtlichen, manchmal sogar spannungsgeladen Situationen kommen kann (nicht ohne Grund wird auch bei dem bereits existierenden stationären Freeshop-Modul von Boxtribute die Vergabe von Terminen an die Geflüchteten unterstützt), wurden hierbei ebenfalls mittels eines erweiterten, mehrsprachigen digitalen Formulars die Wünsche der Geflüchteten digital erfasst und diese Bestellungen dann ca. hundert bis zweihundert Meter von der Ausgabestelle entfernt bearbeitet und schließlich an die Geflüchteten übergeben.

Eine zentrale Erkenntnis war hier für uns, dass im Bereich von No-Code-Tools für humanitäre Organisationen wohl noch großes Potential vorhanden ist, insbesondere im Bereich von Open-Source No-Code-Plattformen (die von uns für das erwähnte schnelle Testen genutzte Plattform war kein OpenSource-Produkt) - konkret, die Möglichkeit für Organisationen individuelle Anforderungen hinsichtlich der eigenen Arbeitsprozesse auch ohne EntwicklerInnen-Ressourcen schnell zumindest prototypisch, oft aber auch produktiv, digital unterstützen zu können. Open Source No-Code-Plattformen haben im Vergleich zu den kommerziellen Äquivalenten dabei Vorteile wie Datenhoheit, Erweiterbarkeit und eine bessere Kostenkontrolle. Diese Einsicht führte auch zur Beantragung einer weiteren Förderung mit diesem Arbeitsschwerpunkt.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Die finale Evaluation der beschriebenen neuen Features innerhalb des Produktiveinsatzes über einen mehrwöchigen Zeitraum bei unserer Partnerorganisation in Serbien steht noch aus, doch lässt das bisherige Nutzerfeedback den Schluss zu, dass durch die neuen Funktionen einerseits eine verbesserte Datenqualität bezüglich der Warenausgänge und Rückläufer (und entsprechend auch der tatsächlich ausgegebenen Mengen je nach Warentyp - grundsätzlich ein wichtige Kenngröße für viele unserer Partner-Organisationen) und andererseits eine vereinfachte Planung, Vorbereitung und Nachbereitung in den täglichen operativen Prozessen erreicht werden wird. Letzteres spielt vor allem vor dem Hintergrund einer oft hohen Fluktuation der freiwilligen HelferInnen vor Ort bei den Organisationen eine Rolle: das Onboarding neuer freiwilliger HelferInnen kostet Zeit und das Potential für Missverständnisse zum Beispiel bei der Vorbereitung von Tagestouren und deren Ausgabe-Ereignissen kann durch leicht benutzbare und klare digitale Lösungen reduziert werden.

Die unmittelbaren Nutznießer sind dabei die Partnerorganisationen und deren freiwilligen Helfer vor Ort, die mittelbar, aber primäre Zielgruppe sind die geflüchteten Menschen, welche durch die erwähnten Optimierungen bei den Organisationen eine im Durchschnitt verbesserte Versorgung erwarten können.

Auch planen wir die Features, welche momentan nur für die erwähnte Partnerorganisation in Serbien freigeschaltet sind, bei erfolgreichem Produktivtest über einen längeren Zeitraum für weitere interessierte Organisationen mit ähnlicher Arbeitsweise und Herausforderungen bereitzustellen.

Auch andere Verbesserungen, wie z.B. die direkte Integration des QR-Scanners in die mobil-optimierte Web-Anwendung, wurden von Nutzern als sehr positiv bewertet.

Persönlich haben wir beide enorm viel gelernt. Wir hatten die Möglichkeit, die Domäne von Boxtribute und der humanitären Organisationen allgemein noch besser kennenzulernen (wir selbst sind bei dem Projekt, im Gegensatz zu anderen Mitglieder von Boxtribute, erst seit ca. einem Jahr vor Förderungsbeginn involviert gewesen), eine Problemstellung von Anfang, also dem User Research mit den Nutzern vor Ort, bis zum Implementieren und Testen eines Prototypens umzusetzen, Kenntnisse über neue Technologien zu erlernen bzw. zu vertiefen, uns in der Open Source Community zu vernetzen und größtenteils selbstbestimmt zu arbeiten - natürlich unter Abstimmung mit den schon länger involvierten Boxtribute-Teammitgliedern, insbesondere der Produktmanagerin und einem der Gründer - und dabei die für uns passende Form des Projektmanagements zu nutzen. Unser beider Erfahrungsstand bei der Softwareentwicklung ist noch relativ unterschiedlich und wir haben mehrmals deutlich realisiert, was für eine hervorragende Möglichkeit das Förderungsprogramm ist, trotz oder gerade wegen dieser Unterschiede im Kenntnisstand, zusammenzuarbeiten und voneinander zu lernen.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Wie eingangs erwähnt, war zu Beginn der Förderung geplant, primär die eigentlichen Ausgabe-Ereignisse vor Ort (z.B. anonymisierte Registrierung der Geflüchteten vor Ort in einer Art digitaler Warteschlange, Live-Erfassung der ausgegebenen Produkte etc) in dem Prototypen umzusetzen. Wegen der ebenfalls oben beschriebenen Erwägungen haben wir diese Features jedoch auf spätere Iterationen verschoben.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Interessierte können einerseits die Software aufsetzen und die Prototypen der neuen Features direkt ausprobieren. Der Code findet sich unter: [github.com/boxwise/boxtribute](https://github.com/boxwise/boxtribute)

Alternativ kann auf einer von Boxtribute betriebenen Demo- und Test-Umgebung ein Eindruck der Features vermittelt werden. Kontaktmöglichkeiten finden sich unter [www.boxtribute.org](https://www.boxtribute.org).

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Wie bereits im ersten Abschnitt beschrieben, wurde nach der anfänglichen User Research-Phase eine Umgestaltung der Planung und Anpassung der konkreten Zielsetzung in Absprache mit dem Projektträger vorgenommen, wobei das grundsätzliche Ziel (Besserer Unterstützung von Boxtribute für Organisationen, welche Geflüchtete ohne halbwegs festen Aufenthaltsort versorgen) das gleiche blieb.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Im Rahmen unserer User Research-Bemühungen kamen wir auch in Kontakt mit anderen Organisationen und Problemstellungen im humanitären Bereich, insbesondere auch im Kontext der gestarteten Unterstützungs-Initiativen in Verbindung mit dem Angriffskrieg auf die Ukraine. Diese Begegnungen haben unsere Einsichten in die Domäne insgesamt erweitert und uns auch den Bedarf nach flexiblen, leicht nutzbaren, kosteneffizienten und auf Datenhoheit und -schutz ausgelegten digitalen Werkzeugen verstärkt vor Augen geführt. Diese Einsichten haben zwar nicht direkt die Umsetzung des Prototypens beeinflusst, doch unser Verständnis für die Wichtigkeit von Lösungen wie Boxtribute sowie für den Bedarf nach weiteren potentiellen Features für Boxtribute und darüber hinaus geschärft.

# Giteaverse - Gitea im Fediverse

---

## *Schlussbericht*

Zuwendungsempfänger:

Janis Estelmann

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S17 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Gitea ist eine Anwendung zur Softwareentwicklung und stellt eine leichtgewichtige Alternative zu Quellcodehostern wie GitHub und GitLab dar, da sie sich sehr einfach selbst hosten lässt. Sie eignet sich für Teams unterschiedlichster Größe zur Quellcodeverwaltung. Hierbei wird den Benutzern ermöglicht eigene Git-Repositories anzulegen und über ein eingebautes Ticket-System zu verwalten.

Bestehende proprietäre Anbieter wie GitHub und Bitbucket hosten ihre Angebote üblicherweise in Rechenzentren und bieten dem einzelnen Nutzer kaum Möglichkeiten zu überprüfen, was mit seinen Daten dort geschieht. Einen Ausweg hiervon bietet die Migration der eigenen Daten zu einer selbstgehosteten Gitea Instanz. Hierbei verliert der Nutzer aber den „Community-Faktor“ der Plattform, über die er auch Projekte anderer Nutzer betrachten und an diesen beispielsweise durch Codeänderungen teilhaben kann. Durch die Zentralisierung müsste sich ein Benutzer im Extremfall bei jedem Projekt neu registrieren, um mit diesem zu interagieren.

Um diesen Nachteil auszugleichen, sollen Gitea Instanzen die Möglichkeit erhalten untereinander vernetzt zu werden. Dazu soll Gitea ein Teil des Fediverse werden und hierbei über standardisierte Protokolle Daten mit anderen Quellcodehostern austauschen. Dies würde es einem Benutzer ermöglichen mit seiner lokalen Gitea Instanz Projekte anderer Nutzer zu finden und diese zu nutzen. Fediverse ist als dezentrales Netzwerk konzipiert und führt damit nicht zu dem anfangs beschriebenen Vendor-Lock-in.

Zur technischen Umsetzung der Vernetzung soll ein Daten-Protokoll auf Basis von ActivityPub implementiert werden. Hierzu soll mit dem ForgeFed-Projekt zusammengearbeitet werden, das einen ersten Entwurf einer solchen Erweiterung für ActivityPub erstellt hat. Der Entwurf definiert unterschiedliche Entitäten und Strukturen für die Vernetzung von Quellcodehostern. Aufgrund der begrenzten Umsetzungszeit während des Prototype Fund ist das Ziel, dass in der Implementierung nur die Möglichkeit zum Datenaustausch bei Tickets umgesetzt wird. Gitea mit allen vorhanden Funktionen gefördert zu gestalten, ist in dem kurzen Zeitraum nicht möglich.

Als Meilensteine wurden die folgenden Punkte vor dem Start des Projekts definiert:

- Evaluation der Protokolle und Planung der anzupassenden Codestellen
- Schnittstellen-Implementierung für die Akteure Benutzer, Repository und Ticket
- Mit Vervis existiert eine Referenzimplementierung zum ForgeFed Protokoll. Mit dieser Software soll die Implementierung in Gitea getestet werden.

## **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Thematisch ist das Projekt in den Bereichen „Softwareinfrastruktur“ und „Datensicherheit“ angesiedelt. Durch die Implementierung des standardisierten Protokolls ActivityPub wird es ermöglicht, dass sich Gitea von einer Einzel-Instanz-Software zu einer Multi-Instanz-Software wandelt und hierdurch beliebig viele Anwender miteinander interagieren können. Die Möglichkeit die eigenen Daten nicht bei einem proprietären Anbieter hosten zu müssen, fördert die informationelle Selbstbestimmung eines jeden Nutzers.

## **Ausführliche Darstellung der Ergebnisse**

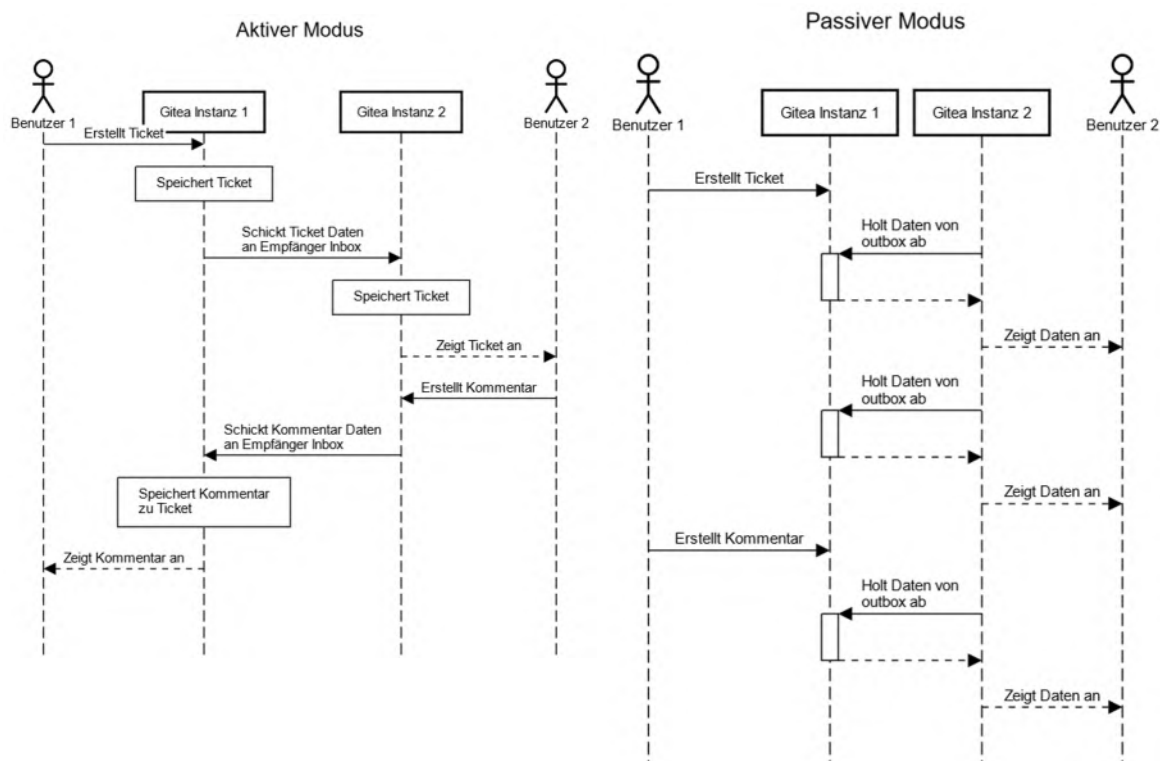
Um das ActivityPub Protokoll nutzen zu können, mussten die internen Gitea Strukturen an die im Protokoll verwendeten Entitäten angepasst werden bzw. ein Mapping zwischen diesen erstellt werden. Da die bestehenden Strukturen von Gitea bereits seit Jahren im Einsatz und nicht im Hinblick auf etwaige Vernetzungsmöglichkeiten designet wurden, waren hier Anpassungen kaum möglich. Daher wurde in Zusammenarbeit mit ForgeFed Projekt die bestehende ActivityPub Erweiterung angepasst, um ein Mapping zwischen dessen Strukturen und denen von Gitea zu ermöglichen.

Für alle Akteure (Benutzer, Repositories, Tickets) wurden die ein- und ausgehende Schnittstellen (inbox, outbox) erstellt, mit denen sich fremde Instanzen verbinden und darüber Daten austauschen können. Da die ausgetauschten Daten kryptografisch signiert werden müssen, wurde Gitea dahingehend erweitert, dass zu jedem registrierten Benutzer automatisch passende Schlüsselpaare erzeugt werden.

Für den Prototypen wurde das passive Abfragen von Daten über den Outbox Mechanismus implementiert. Hierbei kann der anfragende Dienst periodisch Änderungen bei der anderen Instanz abfragen und diese verarbeiten. Das aktive Senden von Änderungen an die Inbox einer anderen Instanz wurde nur für einige Benutzerinteraktionen wie der „Folgen“ Funktion umgesetzt.

Der geplante Meilenstein zum Test der Implementierung mit der bereits bestehenden Software Vervis konnte nicht umgesetzt werden. Ein Grund hierfür ist, dass die Vervis hostende Plattform während des Umsetzungszeitraums nicht mehr erreichbar war. Auf Nachfrage beim Entwickler wurde in Aussicht gestellt, dass Vervis bei einem anderen Hoster wieder zugänglich gemacht wird. Letztlich dauerte dies aber zu lange, sodass Vervis nicht mehr berücksichtigt werden konnte. Da wie geplant nicht alle Aspekte von ForgeFed berücksichtigt wurden, hätte ein Test der Teilumsetzungen möglicherweise auch nicht funktioniert, da eine komplette Implementierung zur Kommunikation nötig gewesen wäre. Tests erfolgten somit nur zwischen Gitea Instanzen.





## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Jeder, der bereits jetzt Gitea einsetzt, kann von der Erweiterung zum Vernetzen mit anderen Instanzen profitieren. Sobald eine neue Instanz dem Fediverse beiträgt, kann sie auf die Inhalte anderer Nutzer zugreifen und umgekehrt.

Auch private Netzwerke innerhalb einer Organisation lassen sich auf diese Weise bilden und ermöglichen somit den Zusammenschluss verschiedener Standorte oder Bereiche.

Die Veröffentlichung des Quellcodes unter der offenen MIT Lizenz ermöglicht es jeder Person Einsicht in den Quellcode zu nehmen, Dinge zu ändern oder allgemein Aufgaben bzw. neue Funktionen zu implementieren und damit zum Projekt beizutragen. Durch das kooperative Zusammenarbeiten fällt die entstehende Software diverser aus, da Ideen von unterschiedlichen Seiten einfließen können.

Bis in das Jahr 2023 ist außerdem eine Agenda mit zu erledigenden Aufgaben und Meilensteinen aufgestellt worden. Diese wurden mit der Gitea Community erarbeitet.

## Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Nach den ersten zwei Monaten hat sich herausgestellt, dass die bisher eingesetzte Bibliothek zur Bereitstellung der ActivityPub Funktionalität nicht flexibel genug für weitere Anforderungen war. Deshalb musste eine Alternative verwendet und der bereits vorhandene Quellcode auf diese umgeschrieben werden. Dies brachte allerdings wenig Probleme mit sich, da die bisherige Rahmen-Implementierung größtenteils beibehalten werden konnte.



## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Der Quellcode von Gitea ist unter der offenen MIT Lizenz auf GitHub veröffentlicht (<https://github.com/go-gitea/gitea>). Für das Jahr 2023 ist eine Migration von GitHub zu <https://gitea.com/gitea> geplant. Dort wird der Quellcode zukünftig zu finden sein. Teile der Arbeiten zum Fediverse sind auch dort erreichbar bis sie zur allgemeinen Verfügbarkeit in den Hauptzweig der Anwendung übertragen werden.

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Gitea in allen Bereichen gefördert zu gestalten wird noch viele weitere Monate dauern. Die für dieses Projekt herausgetrennten Teilbereiche konnten in der geplanten Zeit umgesetzt werden.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Das oben erwähnte Projekt ForgeFed hat bei Besprechungen teilgenommen und dabei auch schnell auf etwaige Probleme bzw. benötigte Funktionen reagiert, sodass diese im Einklang mit dem Gitea Projekt sind. Als Nebenprodukt ist hierbei ein standardisiertes Modell entstanden, das beschreibt in welchem Format Daten aus einem System zur Verwaltung von Software exportiert und in einem anderen System anschließend importiert werden können. Dieses Modell wurde von dem ForgeFriends Projekt gestaltet und in Gitea umgesetzt. Die verwendeten Strukturen ähneln hierbei denen von ActivityPub.

# Sponti – Spontanhelfer- Teamorganisation

---

## *Schlussbericht*

Zuwendungsempfänger:

Alexander Blum, Angelika Ehrlich, Thomas Mielke, Sarah Stoffels und Nils Witt

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS22S18** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Im Vorhaben Sponti sollte ein Portal realisiert werden zur Registrierung und Verwaltung von potenziellen Helfenden in Krisensituationen für Organisationen, Vereine oder andere Gruppen. Dabei sollten bestimmte Fähigkeiten und Materialien, die sie für verschiedenartige Einsätze zur Verfügung stellen können, mit aufgenommen werden. In konkreten Einsatzfällen sollen dann die Details kurzfristig kommuniziert und die Personen eingeladen werden können.

#### **Meilensteine**

- Implementierung der Verwaltung von Gruppen und Material im Portal
- Implementierung einer Terminverwaltung mit Rückmeldungs-Kanälen und -Auswertungen
- Evaluierung von datenschutzkonformen Messenger-Lösungen (z.B. Matrix-Server)
- Integration des Messengers in das bestehende Portal
- Bereitstellung der Serverinhalte als API
- Bereitstellung der User-Interfaces als App mit Push-Funktion ("progressive web app" oder native Apps)
- nach Möglichkeit auch Anbindung eines freien Pushservices (unifiedpush.org - im Gegensatz zu den kommerziellen proprietären Push-Services von Apple und Google)
- Ggf. Anbindung weiterer Kommunikationswege (SMS o.ä.)
- Ggf. Anbindung von Social-Media-Kanälen

## **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Es hat sich beim Einsatz der Vorläuferversion dieser Fördersoftware bei der Nutzung zur Einsatzkoordination gezeigt, dass es keine fertige Open Source-Lösung gibt, um sowohl Organisatoren, als auch Teilnehmern sehr kurzfristig verschiedene Kanäle zum Austausch und zur Organisation zu ermöglichen. Daher ist es ein Kernziel des Vorhabens, eine solche Lösung aus Open Source-Komponenten datenschutzfreundlich zu bauen, die (selbst-)organisierte Gruppen selbst betreiben oder durch einen unabhängigen Hostler betreiben lassen können, sowie bei Bedarf anpassen und erweitern können.

Zudem kann diese Software nicht nur von Organisationen sondern auch von „losen Gruppen“ ohne feste Struktur zu deren Koordination von z.B. Hilfsaktionen verwendet werden und unterstützt somit „Civic Tech“.

## **Ausführliche Darstellung der Ergebnisse**

### **Technologien und Aufbau**

Für das Backend wurde wie bereits in der Vorgängersoftware das Python-Framework Django verwendet, die Datenstruktur allerdings sehr ausgeweitet.

Zur Bereitstellung von API-Funktionen wurde GraphQL angebunden und für die API-Authentifizierung JSON Web Token (JWT) implementiert.

Für die Datenhaltung kommen Postgresql und Redis zum Einsatz. Mailhog wird für das E-Mail-Handling zu Testzwecken verwendet.

Frontends wurden mit React und MAUI umgesetzt und sind unabhängig voneinander verwendbar. Sie kommunizieren über die API-Funktionen mit dem Backend.

Zum Deployment der Anwendungen auf Servern wurden Automatisierungsskripte für die Container-Technologie Docker erstellt und jeder Serverdienst als separater Container angelegt.

## Beschreibung der typischen Use Cases und Abläufe

Zunächst muss in der Software von administrativer Seite eine Organisation angelegt werden. In diesem Bereich kann dann die zuständige Organisation Projekte (Abbildung rechts) und Einsätze anlegen.

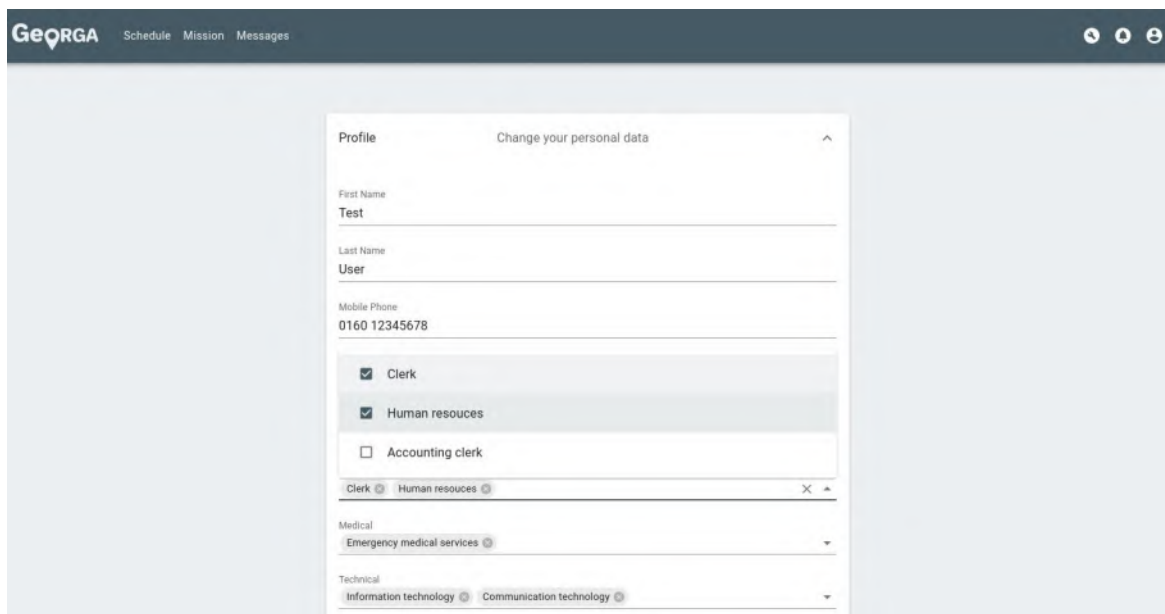
Nutzer können sich registrieren und ihre persönlichen Details eingeben sowie später editieren (Abbildung unten).

Personen können sich auch für mehrere Organisationen in der gleichen App anmelden, sofern in der Softwareinstanz mehrere Organisationen serverseitig angelegt wurden.

Einsätze enthalten weitere Details, u.a. gewünschte Tätigkeitsrollen, deren konkrete Aufgaben, Einsatzschichten und einen oder mehrere hinterlegte Einsatzorte. Die registrierten Personen, die sich für die Teilnahme an (Hilfs-)einsätzen interessieren, können dann konfigurieren, welche Projekte und Einsätzen sie in ihrer Weboberfläche oder App sehen wollen. Ebenso lässt sich einstellen, dass sie regelmäßige Anfragen dafür erhalten und über welche Kommunikationskanäle diese Anfragen erfolgen sollen.



*Native App - Projektliste*



*React Frontend - Ansicht zur Bearbeitung des Nutzerprofils*

## Erreichung der Meilensteine

Wir konnten somit folgende Meilensteine erarbeiten:

- Verwaltung von Gruppen und Material im Portal: Materialverwaltung möglich
- Terminverwaltung mit Rückmeldungs-Kanälen und -Auswertungen: **ausführlich**, einige Aspekte im Frontend noch zu implementieren
- Evaluierung von datenschutzkonformen Messenger-Lösungen (z.B. Matrix-Server): **Konzept** mit Fokus auf Matrix zur Erstellung von kurzzeitigen Chaträumen passend zu Einsätzen und deren Teilnehmern
- Bereitstellung der Serverinhalte als API: **ausführlich** mit GraphQL
- Bereitstellung der User-Interfaces als App mit Push-Funktion ("progressive web app" oder native Apps): React Frontend erstellt (auch als PWA nutzbar), native App wurde mit MAUI erstellt
- Anbindung eines freien Pushservices: **anzubinden** über Channel-Datenstruktur unserer Software
- Anbindung weiterer Kommunikationswege (SMS o.ä.): **anzubinden** über Channel-Datenstruktur unserer Software

Die Integration des Messengers in das bestehende Portal sowie Anbindung an Social Media hätten leider den möglichen Zeitrahmen und Aufwand gesprengt.

## Erkenntnisse aus der Projektarbeit

Als Backend für die Software verwenden wir Django. Dies ist ein etabliertes und gut gepflegtes Framework, jedoch sind wir bei unserer Implementierung an einigen Stellen an Grenzen gestoßen, an denen benötigte Funktionen manuelle Anpassungen erforderten. Dies betrifft eine Änderung der Subscription-Library nach Upgrade auf Django 4.1 und eigene Implementierung eines Rechtesystems, das auch auf Datenobjektebene (statt nur Klassenebene) Zuordnungen ermöglicht. Zudem haben wir die Erweiterung „graphene-django“ für die Anbindung von GraphQL angepasst, u.a.: zur Verwendung unterschiedlicher IDs für Dateneinträge per API und Datenbank-intern (aus Security-Gründen), Logging, „paging/slicing“ von Listen mit Einhaltung der Spezifikation von „Global Relay API“.

Die Begleitung durch die Open Knowledge Foundation hat uns u.a. Coachings zum Thema UI/UX ermöglicht und somit noch mehr Aspekte aufgezeigt, die es den Benutzern einfacher machen können, die Thematik und Möglichkeiten der Plattform und App unkompliziert zu erfassen. Ein Aspekt wäre, bei möglicher Zusammenarbeit mit Organisationen speziell auf diese zugeschnittene In-App-Tutorials zu integrieren, die den Benutzern ihre Möglichkeiten erläutern. Ebenso wären Integrationen in Social-Media-Systeme gut zur Steigerung des Bekanntheitsgrads und für die Benutzer ein weiterer Motivationsaspekt (z.B. wenn Teilnahme an Einsätzen im eigenen Social-Media-Account über die App kundgetan werden kann).

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Der Fokus der im Vorhaben entwickelten Software lag im Gegensatz zur Vorgängerversion nicht auf einem speziellen Einsatzszenario des derzeitigen Nutzers sondern wurde generalisierbar für verschiedene Arten von Organisationen und möglichen Einsätzen durchdacht. Somit können auch Gruppen mit weniger Strukturen als z.B. eine Hilfsorganisation diese Software zur Koordination verwenden.

Durch den modularen Aufbau und generalisierte Schnittstellen können verschiedene Kommunikationssysteme, je nach Bedarf der nutzenden Gruppe, angebunden werden. Die Benachrichtigungen („Message“) auf verschiedenen Ebenen sind in der Datenstruktur generalisiert, können aber an verschiedene „Channels“ wie In-App-Notifications, Push-Nachrichten, E-Mail, Chat und SMS verteilt werden.

Das Projekt kann von den nutzenden Gruppen selbst gehostet und erweitert werden, wir können diesen bei Bedarf aber auch die technische Bereitstellung anbieten.

Mögliche Weiterentwicklungen betreffen genauere Anpassung von Features auf spezielle Organisationen aber auch generalisierbare Schnittstellen z.B. zu verschiedenen bekannten Kommunikationssystemen. Aktuell finden unsererseits Gespräche mit Organisationen zur Nutzung der Software statt, die recht flache Hierarchien haben und unserer Hoffnung nach gut zur Open-Source-Community passen.

Für unsere persönliche fachliche Weiterentwicklung war das Projekt sehr geeignet, da es sich über verschiedene Technologien und Ebenen (Frontend, Backend, native App) erstreckt und an bestehende Open-Source-Projekte anzubinden ist.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

### **Solito als übergreifendes Frontend-Framework**

Der Ansatz, mit Solito Aufwand und Code durch ein übergreifendes Frontend-Framework zu sparen, war leider nicht vielversprechend, da geplante Features wie z.B. Responsive UI für Web viel umständlicher umzusetzen waren, als wenn man sie ohne Solito in React implementiert. Auch ist das Solito-Projekt bisher im Wesentlichen von einem einzigen Entwickler abhängig im Gegensatz zu React.

## **Rechtliche Diskussionen und Vermittlungsprobleme bzgl. Open-Source-Lizenzvorteile**

Mit der Organisation (im Folgenden „HiOrg“ genannt), die die Vorversion der hier geförderten Software genutzt hat, war zunächst geplant, mögliche fachliche Anforderungen als Musterbeispiel für Hilfsorganisationen für das Förderprojekt zu erarbeiten.

Inzwischen hatte jedoch das Thema Spontanhelferkoordination allgemein an „Prominenz“ gewonnen und es gibt nun im Umfeld der typischen Hilfsorganisationen einige, die eine dafür nutzbare Software als „Software as a Service“ mit proprietär entwickelter Software vertreiben. Dementsprechend gab es nun auch bei der HiOrg einen solchen Plan und daher wurde innerhalb der HiOrg die Zusammenarbeit mit uns zu fachlichen Definitionen nur unter der Bedingung erlaubt, dass die daraus entstehende Software exklusiv und mit alleinigem Wiederverkaufsrecht für die HiOrg entwickelt worden wäre, was der ursprünglichen Entwicklung als Open-Source-Software allerdings entgegensteht und mit dem Förderprojekt keinesfalls vereinbar ist.

Insgesamt gab es bei der HiOrg durchaus viele Befürworter der Zusammenarbeit unter den gegebenen Förderbedingungen, jedoch entschied sich der Vorstand dagegen.

Es gab auch an überregionaler Stelle dieser HiOrg Interesse, sich die ursprüngliche Version und später die geförderte Software anzuschauen, da man dort vor einem Kaufentscheidungsprozess für eine solche Software stand, allerdings wurde uns keine detailliertere Kommunikation mehr mit dieser Stelle erlaubt, da die HiOrg selbst ihrer überregionalen Stelle diese Software unter ihrem eigenem Namen verkaufen wollte und die Personen unserer GbR als Subunternehmer oder Angestellte unter Vertrag nehmen wollte.

Dies wurde durch eine Person der HiOrg sogar in einem persönlichen Gespräch mit einer Person unserer GbR durch Druck und Androhung von rechtlichen Schritten versucht, zu bewirken, sowie dass wir das Förderprojekt abbrechen sollten.

Allerdings gab es zwischen HiOrg und den Personen unserer GbR nie einen Arbeits- oder Auftragsvertrag oder eine Vergütung und solche Ansprüche sind nicht von einer ehrenamtlichen Vereinsmitgliedschaft gedeckt. Da wir uns vor allem wegen der offenen Lizenz und mit den Plänen einer Nutzbarkeit der Software für die Allgemeinheit für den Prototype Fund beworben hatten, war hier nun eine Zusammenarbeit mit der HiOrg unsererseits ausgeschlossen.

Insgesamt zeichnen unsere bisherigen Erkenntnisse nicht das Bild, dass von den klassischen Hilfsorganisationen in Deutschland eine übergreifende Zusammenarbeit an Software durch die Mitwirkung an und Investition in Open-Source-Projekte gewünscht ist, allerdings sind die möglichen Vorteile nach unserem Eindruck dort auch noch zu wenig bekannt.



Das gleiche Problem sehen wir bezüglich agiler Softwareentwicklung und Beratungsverträgen, die für eine Zusammenarbeit zwischen Open-Source-Projekt und Kunde an anderen Stellen gängige Praxis ist. In den klassischen Strukturen herrscht allerdings meist die Ausschreibungspraxis vor, in der eine Software zum Vergleich eine unveränderliche Feature-Liste und einen einmaligen Fixpreis haben muss und anderenfalls nicht in der Auswahl berücksichtigt wird.

Daher braucht es unserer Meinung nach noch etwas Arbeit, Open-Source-Projekte und Vertragstypen für agile Softwareentwicklung überhaupt in diesem Umfeld bekanntzumachen. Es wäre eine gute Vorbildfunktion, wenn bald allgemein bei Behörden, behördennahen Institutionen und steuerlich oder durch Spenden finanzierte Organisationen mehr Priorität auf Investitionen in Open-Source-Projekte gelegt würde, damit Budgets nicht mehr mehrfach für identische Programmieraufwände und Software ausgegeben werden.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Projekthomepage:

<https://www.georga.app>

GitHub:

<https://github.com/georga-app/georga-server-django>

<https://github.com/georga-app/georga-client-react>

<https://github.com/georga-app/GeorgaMobileClient>

<https://github.com/georga-app/georga-docker>

## Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Arbeitspaket	Geplante Stunden	Geleistete Stunden
Softwareentwicklung und Test (Backend, DB, API, Schnittstellen)	400	393,75
Softwareentwicklung und Test (Frontend, Webseite)	120	105,75
Softwareentwicklung und Test (Frontend - speziell App)	180	250,5
Systemadministration	40	29,75
Geschäftsführung & Organisatorisches	50	60
Projektleitung	40	65,25
Anforderungsanalyse	80	150,80
Öffentlichkeitsarbeit	42	11,80
Summe	<b>952</b>	<b>1067,6</b>

Die verbrauchten Stunden stimmen grob mit der Planung überein, außer dass wir bei der App-Entwicklung und Anforderungsanalyse noch mehr Zeit verbraucht haben.

## Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab keine Entwicklungen anderer Personen oder Institutionen, die Einfluss auf unsere Arbeiten oder Zielsetzungen hatten. Allerdings konnten wir beobachten, dass das Thema Spontanhelfer-Koordinationssoftware bei Hilfsorganisationen an Bedeutung gewinnt und sogar Softwareprodukte dafür entwickelt werden, jedoch alle ohne offenen Code und oft in Kombination mit weiteren Notfallrettungs-Funktionen (z.B. <https://www.retter4u.app/idee-konzept/>). Nach unserem Wissen ist bisher keine alternative Open-Source-Lösung erschienen.

# **Small Device C Compiler: Standardkompatibilität und Usability**

## **SDCC-STD-UX**

Philipp Krause, Felix Salfelder, Benedikt Freisen

27. September 2022

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S19 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

## **1 Aufgabenstellung und Motivation**

Die Informations- und Kommunikationstechnik trägt bei Herstellung und im Betrieb wesentlich zum globalen Ressourcen- und Energieverbrauch bei. Es sollten Einstiegshürden beim Entwickeln sparsamer Systeme verringert werden.

Mikrocontroller ( $\mu\text{C}$ ) sind in nahezu allen elektronischen Geräten enthalten, und werden dort eingesetzt, wo es starke Einschränkungen bei Kosten, Größe und Energieverbrauch gibt. Sie sind maßgeblich für das Internet of Things (IoT) und Sensornetzwerke. Obwohl der einzelne  $\mu\text{C}$  in Herstellung und Betrieb wenig Ressourcen erfordert, ist der Gesamtaufwand aufgrund der Stückzahlen enorm. Beispielsweise wurden 2020 weltweit 8-Bit- $\mu\text{C}$  für circa sieben Milliarden Euro verkauft, bei einem Stückpreis von oft unter einem Cent. Die Wahl eines  $\mu\text{C}$  kann den Ressourcen- und Energieverbrauch entscheidend beeinflussen. Wenn statt eines einfacheren 8-Bit  $\mu\text{C}$  ein 32-Bit  $\mu\text{C}$  eingesetzt werden muss, oder aus anderen Gründen ein vorhandener  $\mu\text{C}$  nicht mehr den Anforderungen genügt, und ersetzt werden muss, entstehen Sprungskosten im Verbrauch.

Aktuelle Software, insbesondere auch FOSS, verwendet oft Features aus aktuellen Sprachstandards<sup>1</sup>, die von GCC und LLVM unterstützt werden. Um solche Software auch auf einfacheren 8-Bit  $\mu\text{C}$  einsetzen zu können, ist ein benutzerfreundlicher Compiler für 8-Bit  $\mu\text{C}$  nötig, der aktuelle Standards unterstützt.

Der Small Device C Compiler (SDCC) ist ein ein Freier / Open Source (FOSS) Compiler, der in der Sprache C geschriebene Programme für 8-Bit Mikrocontroller  $\mu\text{C}$  übersetzt. Er schließt somit eine Lücke unterhalb der bekannten FOSS Compiler GCC und

---

<sup>1</sup>„Standard“ wird hier, wie in der Informationstechnik üblich, im Sinne des englischen „standard“ verwendet, was in der Bedeutung dem deutschen „Norm“ entspricht

LLVM, die vorwiegend Systeme ab 32 Bit unterstützen, wie man sie in PCs, Servern und Smartphones findet.

Das Projekt beabsichtigte Unterstützung für C-Sprachstandards in SDCC auszubauen um Interoperabilität und Benutzbarkeit zu verbessern. Es sollte Unterstützung für ISO C90, C99 vervollständigt, für C2X hinzugefügt werden. Es sollte standardkonforme Ausgabe von Debuginformationen implementiert werden, die Integration in Entwicklungsumgebungen verbessert werden.

## **2 Beitrag zu den Zielen der Förderinitiative „Software-Sprint“**

SDCC ist der bedeutendste freie Compiler für 8-Bit-µC, und somit ein essentieller Bestandteil der Softwareinfrastruktur für diese, eingebettete Systeme, das Internet of Things (IoT) und Sensornetzwerke. Die erreichten Verbesserungen in der Unterstützung für die C-Sprachstandards in SDCC stärken daher die Softwareinfrastruktur in diesem Bereich deutlich; SDCC übertrifft bezüglich der C-Standardunterstützung nun auch alle in diesem Bereich verbreiteten nicht-freien Compiler deutlich.

## **3 Einflüsse und Ergebnisse Anderer**

Ein Ziel des Projekts war, die Unterstützung für den zukünftigen C-Standard C2X zu verbessern. Bei Beantragung des Projekts war noch kaum abzusehen, welche Änderungen dieser Standard gegenüber dem aktuellen C17 aufweisen würde. Dies wurde durch das Normierungsgremium ISO SC22/WG14 entschieden, insbesondere auch auf zwei einwöchigen Sitzungen im Mai und Juli, die in die Projektlaufzeit fielen. ISO SC22/WG14 beeinflusste also direkt die Ziele des Projekts. Philipp Klaus Krause nahm als von der DIN entsandter Experte an den Sitzungen von ISO SC22/WG14 im Projektzeitraum teil.

## **4 Einhaltung der Planung**

Da über viele Aspekte des zukünftigen C-Standards C2X erst relativ spät während der Projektlaufzeit vom Normierungsgremium ISO SC22/WG14 entschieden wurde, stand für diese relativ wenig Zeit zur Implementierung zur Verfügung. Auch war der Implementierungsaufwand für manche Aspekte von C2X etwas größer als erwartet. Im Bereich der Usability (insbesondere bei der Integration von SDCC in die integrierte Entwicklungsumgebung Code::Blocks) gab es Beiträge aus der Community). Bezüglich der Ziele der Förderinitiative erschien uns die Unterstützung der C-Standards als wichtigster Aspekt. Daher haben wir vorrangig an der Verbesserung der Unterstützung der C-Standards gearbeitet, unter Vernachlässigung von Verbesserungen der Usability und Debuginformationen.

Zu Projektanfang wurde eine priorisierte (1 (am wichtigsten) bis 5) Liste <https://sourceforge.net/p/sdcc/wiki/SDCC-STD-UX/> von Teilzielen erstellt, die jeweils nach den Sitzungen von ISO SC22/WG14 ergänzt wurde, was schließlich 59 Teilziele ergab.

Zum Projektende waren von den Teilzielen jeweils im SDCC Subversion-Repository <https://sourceforge.net/p/sdcc/code/HEAD/tree/trunk/> und damit bereits den Nutzern von SDCC verfügbar: 100% der Teilziele der Priorität 1, 60% der Teilziele der Priorität 2, 61% der Teilziele der Priorität 3, 52% der Teilziele der Priorität 4, 50% der Teilziele der Priorität 5. Einige weitere Teilziele wurden im Projekt bereits soweit entwickelt, dass sie voraussichtlich innerhalb weniger Wochen nach Projektende erreicht werden.

## 5 Ergebnisse des Projekts

Es wurden entscheidende Verbesserungen in der Standardunterstützung erzielt; zu Projektende dürfte SDCC der Compiler mit der besten Unterstützung für C2X sein, wobei davon auszugehen ist, dass insbesondere GCC und LLVM hier bald auf- und überholen werden. Gegenüber nichtfreien Compilern dürfte der erreichte Vorsprung dagegen noch auf viele Jahre hinaus weiterbestehen.

Die wichtigsten im Projekt erreichten Teilziele waren (dabei bezieht sich C90 auf den ISO C-Standard von 1990, C99 auf den von 1999, C11 auf den von 2011, C2X auf den nächsten):

- Unterstützung für **struct** und **union** als Rückgabetyt von Funktionen (relevant für C90 bis C2X).
- Unterstützung für **struct** und **union** als Parametertyp von Funktionen (relevant für C90 bis C2X).
- Zahlreiche Bugfixes bei **\_Generic** (relevant für C11 bis C2X).
- Zahlreiche Bugfixes im Typsystem (relevant für C90 bis C2X).
- Zahlreiche Bugfixes bei Funktionsdeklarationen (relevant für C90 bis C2X).
- Unterstützung für Variably-modified Types (relevant für C99 und C2X).
- Präprozessormodernisierung (relevant für C11 bis C2X).
- Neues **assert** in der Standardbibliothek (relevant für C2X und Usability).
- **typeof** und **typeof\_unqual** (relevant für C2X).
- Refactoring des Parsers (relevant für C2X).
- Attribute, außer bei Sprungmarken (relevant für C2X).
- Bit-precise integer types (relevant für C2X).
- Übernahme der Tests von GCC 11 (relevant für C90 bis C2X und Usability).
- Übernahme der Tests von QuickC (relevant für C90 bis C2X und Usability).

- Übernahme der Tests von `tinycc` (relevant für C90 bis C2X und Usability).
- GCC-kompatible Syntax für Kommandozeilenoptionen (relevant für Usability).
- `auto` (relevant für C2X).
- Bit utilities in der Standardbibliothek (relevant für C2X).
- Checked integer arithmetic in der Standardbibliothek für alle Typen außer (unsigned) long long (relevant für C2X).
- Warnungen bei ungültigen Unicode-Sequenzen (relevant für C2X und Usability).
- Initialisierung mittels `{}` (relevant für C2X).
- Neue Schlüsselwörter (relevant für C2X).
- `unreachable` (relevant für C2X).
- UTF8-Literals vom Typ `char8_t` (relevant für C2X).
- `memalignment` der Standardbibliothek (relevant für C2X).
- Neue regeln für `enum`-Typen (relevant für C90 bis C2X).
- `nullptr` und `nullptr_t` (relevant für C2X).
- Neue Tag-compatibility-Regeln (relevant für C2X).
- Sprungmarken am Ende von Compound Statements (relevant für C2X).
- `call_once` der Standardbibliothek (relevant für C2X).

Daneben gab es zahlreiche diverse Bugfixes. Es wurde ein SDCC developer meeting organisiert und durchgeführt, es wurden dort, sowie auf der FSiC und der FrOSCon Vorträge zu SDCC gehalten.

Leider ist es bei einigen Teilzielen nicht gelungen, diese im Projektzeitraum zu erreichen, insbesondere:

- Compound Literals (relevant für C99 bis C2X).
- Bugfixes bei Debuginformationen (relevant für Debugging).
- ELF/DWARF-Unterstützung im mcs51-Backend (relevant für Debugging).
- Enhanced Enumerations (relevant für C2X).
- Checked integer arithmetic in der Standardbibliothek für (unsigned) long long (relevant für C2X).
- Qualifier-preserving functions in der Standardbibliothek (relevant für C2X).
- `constexpr` (relevant für C2X).
- ELF/DWARF-Unterstützung im sm83-Backend (relevant für Debugging).

## 6 Zielgruppe und Nutzen für diese

Die Zielgruppe sind Programmierer von  $\mu$ C, etwa Entwickler eingebetteter Systeme, und Nutzer von Geräten, die Ihr Recht auf Reparatur wahrnehmen. Dieses Projekt adressiert sowohl solche, die bereits SDCC verwenden, als auch solche, die bisher nur andere Compiler (nicht-freie Compiler für 8-Bit- $\mu$ C oder beliebige Compiler für größere 32-Bit- $\mu$ C) verwenden. Durch die verbesserte Standardunterstützung und Usability wird diese um einige Programmierer, die bisher nur für größere Systeme programmierten, erweitert. Über die neuen Features wird durch die Ankündigungen der jährlichen SDCC-Releases (auf den SDCC Mailinglisten, in Foren zu  $\mu$ C-Programmierung) informiert. Es gab Vorträge auf einschlägigen Konferenzen (FSiC, FrOSCon) dazu. Beim SDCC developer meeting (im Hackerspace haxko e.V., Andernach) befasste sich einer der drei öffentlichen Vorträge mit dem Projekt, und dessen Fortschritt zum damaligen Zeitpunkt.

## 7 Angestrebte Weiterentwicklung

Die erreichten umfangreichen Verbesserungen in der Unterstützung der C-Standards sollen in den nächsten Monaten noch geringfügig ergänzt werden. Es ist zu erwarten, dass mit steigender Verwendung von C2X noch Bugs durch Nutzer gefunden werden, die zu beheben sind.

Um Nutzer nichtfreier Compiler zum Umstieg auf SDCC zu bewegen erscheint es mittelfristig zielführend, hauptsächlich an Effizienz (im Bezug auf Größe und Geschwindigkeit des erzeugten Codes), Bugfixes und Verbesserungen der Debugunterstützung zu arbeiten. Hierzu wurden Daten von Benutzern und potentiellen Nutzern eingeholt und ausgewertet. Zur Beschleunigung der nötigen Arbeiten wird eine Förderung durch NGI Zero oder den Sovereign Tech Fund angestrebt.



## Schlussbericht

# dade-app – Eine App für Datenanfragen.de

Zuwendungsempfängerin: Lorenz Sieben und Benjamin Altpeter GbR

Datum: 26. September 2022

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S20 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin.

## Kurze Darstellung der Aufgabenstellung und Motivation

Datenanfragen.de ist ein Webportal, das Bürger\_innen den barrierearmen Zugang zu ihren Rechten aus der Datenschutz-Grundverordnung (DSGVO) ermöglicht. Die Webseite generiert clientseitig Anfragen zu den Betroffenenrechten aus Kapitel III DSGVO und bietet die Möglichkeit, dafür Adressaten aus einer offenen Datenbank mit Datenschutzkontakten von Unternehmen und anderen Organisationen auszuwählen. Im technischen Kontext der Webseite lässt sich allerdings das eigentliche Senden der Anfragen nicht ohne vermeidbare Übertragung an Dritte implementieren. Dadurch entsteht ein Medienbruch vom Browser in das E-Mail-Programm.

Im Rahmen des Projektes sollte eine App für Desktop und Handy entwickelt werden, die den gesamten Prozess der Anfrage, aber auch das Schreiben von Antworten und Mahnungen an die Verantwortlichen sowie Beschwerden an die Datenschutzaufsichtsbehörden, in einer einzigen kohärenten User-Experience zusammenführt. Nutzer\_innen sollten die App mit ihren E-Mail-Servern verbinden können, um die Anfragenkorrespondenz darin zu senden und empfangen. Durch zusätzliche Funktionen sollte das Stellen von Anfragen noch weiter erleichtert werden: Mit Benachrichtigungen soll an ausstehende Mahnungen erinnert werden, ein Beschwerdegenerator soll das Durchsetzen der eigenen Rechte über die Aufsichtsbehörden erleichtern und mithilfe eines einfachen Editors können Ergänzungen zur Kontaktdatenbank beigetragen werden.

Das Projekt wurde in die folgenden chronologischen Meilensteine unterteilt:

1. Refactoring der bestehenden Codebasis zu TypeScript und aktuellen Best Practices von Preact
2. Umsetzung der aktualisierten Codebasis als Electron- und Capacitor-App
3. Implementierung von lokalen Benachrichtigungen
4. Verbindung mit E-Mail-Servern zum Versenden von Anfragen und Empfangen von Anfragenkorrespondenz
5. Darstellung der gesendeten Anfragen und zugehörigen Nachrichten (E-Mails etc.) in einer Anfragen- und Korrespondenzübersicht
6. Mahnungs- und Beschwerdegenerator zum Generieren von Reaktionen auf falsch beantwortete Anfragen
7. Ergänzungsmaske für die Kontaktdatenbank im Anfragenfluss

## **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Zielgruppe des Projekts sind als Subjekte der Rechte nach der DSGVO alle Verbraucher\_innen der EU, die über ein kompatibles Endgerät verfügen und deren Daten von Unternehmen und Behörden in irgendeiner Weise verarbeitet werden. Durch das Anbieten einer mobilen App wird die bestehende Zielgruppe konkret auch um Handynutzer\_innen erweitert, bzw. Teilen der Zielgruppe der Zugang erleichtert.

Das Projekt bietet den Verbraucher\_innen deutliche Erleichterungen beim Nutzen und Geltendmachen der eigenen Datenschutzrechte. Durch Datenanfragen.de ist es nicht mehr nötig, sich selbst tief in die Rechtsmaterie einzulesen, eigene Schreiben aufzusetzen und die Kontaktdaten der Ansprechpersonen in Unternehmen und anderen Organisationen herauszusuchen. Durch die neue App wird der Prozess noch reibungsärmer. Auf diese Weise stärkt unser Projekt einzelne Verbraucher\_innen in ihrem Grundrecht auf informationelle Selbstbestimmung und wirkt auch auf gesellschaftlicher Ebene auf Verantwortliche für Datenverarbeitung ein, ihre Prozesse und Geschäftsmodelle datenschutzfreundlich zu gestalten. Datenanfragen.de ist also Civic Tech, eine Software, die durch den verbesserten und erleichterten Zugang zu Bürgerrechten gesellschaftliche Teilhabe und Bürgerbeteiligung stärkt und nachhaltigen Verbraucher\_innenschutz zum Ziel hat.

## **Ausführliche Darstellung der Ergebnisse**

Im Rahmen des Projektes wurden mobile Apps für Android, iOS und eine Desktop-App für Windows, Linux und macOS entwickelt. Dabei konnten wir alle vorgenommenen Meilensteine erreichen: Unsere Codebasis wurde auf die neuste Preact-Version aktualisiert, das State-Management konnte komplett in das State-Management-Framework Zustand umgezogen werden, und die Programmiersprache von JavaScript zu Typescript verändert werden. Die so veränderte Codebasis hat uns bereits im Rahmen der weiteren Entwicklung die Erweiterung des Funktionsumfangs erheblich erleichtert, viele alte Bugs entdeckt und neue verhindert und den Programmcode insgesamt lesbarer und zugänglicher gemacht. Damit konnte Datenanfragen.de sehr nachhaltig verbessert werden.

Auch die neu entwickelte Benutzer\_innenoberfläche (bzw. User-Experience) konnte erhebliche Verbesserungen erzielen: Unsere Tests mit Nutzer\_innen zeigten, dass es deutlich leichter und intuitiver war Anfragen zu stellen und auch Nutzer\_innen, die bisher wenig Berührung zum Datenschutz hatten, gelang es ihre Daten anzufragen und damit ihre Verbraucher\_innenrechte durchzusetzen.

Nicht zufrieden sind wir mit den Benachrichtigungen. Diese haben wir zwar erfolgreich implementiert, aber sie funktionieren nur, solange die App im Hintergrund läuft, da wir u.a. aus Datenschutzgründen darauf verzichten, Google- und Apple-APIs für Push-Notifications zu verwenden. Wir haben uns diesbezüglich mit UnifiedPush beschäftigt und wünschen uns, darüber in Zukunft eine bessere Funktionalität implementieren zu können.

Das Versenden von E-Mails via SMTP aus der App heraus funktioniert sehr gut und senkt wie erwartet die Reibung bei Verschicken von größeren Mengen von Anfragen. Auch der Import von E-Mails zu bestehenden Anfragen ließ sich erfolgreich implementieren, sodass sich Antworten von Unternehmen, sofern sie zugeordnet werden können, mit der App verwalten lassen. Dies kann, wie geplant in der Anfragen- und Korrespondenzübersicht erledigt werden, wo Nutzer\_innen eine Schnellübersicht über ihre vergangenen und laufenden Anfragen haben und die gesamte Korrespondenz diesbezüglich verwalten können.

Dort hinein integrieren konnten wir auch den Mahnungs- und Beschwerdegenerator. Dieser erlaubt es, auf Antworten (oder auch das Ausbleiben von Antworten) direkt zu reagieren, und die Rechtmäßigkeit der Antworten selbst einzuschätzen. Durch das Antworten auf Einschätzungsfragen generieren wir eine Mahnung an die Verantwortlichen oder Beschwerde an die zuständige Aufsichtsbehörde und helfen den Nutzer\_innen so effektiver bei ihrer Rechtsdurchsetzung. Dadurch, dass wir Nutzer\_innen nun ermutigen, auf fehlerhaftes Verhalten zu reagieren, erreichen wir, dass weniger Anfragen unbeantwortet bleiben und fehlerhafte Antworten geahndet werden.

Schließlich konnten wir die Ergänzungsmaske für die Kontaktdatenbank direkt in den Auswahlprozess für Unternehmen integrieren, in der Hoffnung, dass es zukünftig so für Nutzer\_innen einfacher wird, fehlende Unternehmen vorzuschlagen und an der Datenbank mitzuwirken.

Als Ergebnis des Coachings wollen wir uns in unserer Zielsetzung mehr gesellschaftlich ausrichten und die individuelle Rechtsdurchsetzung gesellschaftlich kontextualisieren. Wir wollen intensiver darauf hinwirken, tatsächlich nachhaltige Veränderungen in der Datenschutzpraxis in Europa zu erreichen. Die Möglichkeit dazu haben wir vor allem durch unsere mittlerweile größere Reichweite, die auch während der Förderung gewachsen ist und hoffentlich durch deren Ergebnisse weiter wachsen wird. Insofern war die Förderung nicht nur eine Möglichkeit, unser Angebot technisch zu verbessern, sondern auch mehr Sichtbarkeit zu erlangen und uns mit anderen Personen und Organisationen zu vernetzen, die sich mit ähnlichen Problemen und Fragestellungen beschäftigen.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Wie zuvor bereits dargelegt, erleichtert unser Projekt unseren Nutzer\_innen den Zugang zu ihren grundlegenden Rechten in Bezug auf den Datenschutz. Wir stärken Verbraucher\_innen beim Stellen von Anfragen im Rahmen der Datenschutz-Grundverordnung und stehen ihnen mit unserem Tool im gesamten Anfrage- und Beschwerdeprozess zur Seite. So versuchen wir, den – durchaus vorsätzlich – komplexen Prozess des Stellens von Anfragen so gut es geht zu automatisieren bzw. eine vereinfachte Schnittstelle zur Verfügung zu stellen, um die Schwelle, die eigenen Rechte tatsächlich nutzen zu können, zu senken.

Unser Tool, den Anfragengenerator, haben wir so modular entwickelt, dass er sich auch in andere Software bzw. Webseiten einbinden lässt. Mit der @datenanfragen/components-Bibliothek haben wir wiederverwendbare und neukombinierbare Teile unseres Projekts zur Verfügung gestellt, die sich zum Beispiel in einen Artikel oder ähnliches auf anderen Webseiten einbinden

lassen. So könnte eine Zeitung, die über dubiose Datenpraktiken eines Unternehmens berichtet, ein interaktives Anfragetool in diesen Artikel einbinden und ihre Leser\_innen zu einer Handlung ermutigen. Auch andere Teile unseres Codes haben wir möglichst so aufgeteilt, dass sie für andere Projekte nutzbar werden. Unsere Bibliothek zum Generieren von E-Mails und PDF-Briefen und -Faxen `letter-generator` wird beispielsweise auch von dem Projekt `tracktor.it` verwendet. Außerdem erhalten wir dadurch, dass unser Projekt mit offenem Quellcode veröffentlicht ist, viel Unterstützung von versierten Nutzer\_innen, die uns helfen, unsere Software in andere Sprachen zu übersetzen, Fehler melden oder anders aktiv an der Entwicklung mitwirken.

Weil eine technische Lösung des Problems allein nicht ausreichend ist, wollen wir unser Angebot gerne intensiver datenjournalistisch begleiten und auch Datenschutzforschung betreiben, um Menschen auf die Thematik aufmerksam zu machen. Wir wünschen uns Artikel, die bestimmte Anfragen genauer beleuchten, die bestimmte Rechtsbegriffe oder juristische Feinheiten erklären und die problematische Datenschutzpraktiken aufdecken und Handlungswege anbieten. Für die Umsetzung eines solchen Angebotes fehlt es uns zurzeit aber an den notwendigen finanziellen Mitteln.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Im Rahmen des Projekts traten einige Probleme auf. Zuerst sei zu nennen, dass beim Erstellen eines Kontos für die Google Play Developer Console erhebliche Datenschutzprobleme auftraten, die unsere persönlichen Rechte verletzt hätten. Im Anmeldeprozess wurde unter anderem rechtswidrig eine ungeschwärzte Kopie des Personalausweises verlangt. Entsprechend brachen wir den Prozess mit einer Beschwerde ab und hörten seitdem nichts weiteres von Google. Aus diesem Grund können wir die App leider nicht im Google Play Store veröffentlichen. Ähnliches gilt auch für den Apple App Store: Zur Teilnahme am Apple Developer Program, welches für die Veröffentlichung von Apps im App Store Voraussetzung ist, muss eine umfangreiche Vereinbarung mit Apple geschlossen werden, deren Konditionen wir nicht zustimmen konnten. Unter anderem verhängt Apple ein Verbot, Apps aus dem App Store zu analysieren, an welches wir uns aufgrund unserer Arbeit als Verbraucher\_innenschutzorganisation prozipiell nicht halten können. Leider können wir deshalb die Apps nicht in den beiden größten App Stores anbieten.

Außerdem hatten wir in der Projektbewerbung eingeschätzt, unser Code ließe sich problemlos auf React Native übertragen und für die Apps verwenden, da der vorhandene Code auf dem React-kompatiblen `Preact` basierte. In der genaueren Betrachtung mussten wir allerdings feststellen, dass React Native für unsere Umsetzung ungeeignet ist und wir einen Großteil des Projekts neu hätten entwickeln müssen. Deshalb entschieden wir uns nach einiger Suche für den Cordova-Nachfolger `Capacitor`, der – ähnlich wie `Electron` – Apps in einer `WebView` lädt, sodass sich der Code für die Webseite gut auf die Apps übertragen ließ.

Auch die Suche von Unternehmen, die wir in den Apps lokal implementieren wollten, um die App auch (bis auf das Versenden der Anfragen) komplett offline nutzbar zu machen, erwies sich als problematisch. Unsere bisherige Suchlösung `Typesense` ließ sich nicht in die Apps bündeln. Nach einem erfolglosen Versuch, den Suchserver in `WebAssembly` zu kompilieren, schwenkten wir auf



eine alternative Bibliothek mit dem Fokus auf lokale Suche um, die allerdings leider nicht alle Funktionen der Online-Lösung unterstützt.

Schließlich mussten wir, um den Mailimport zu ermöglichen, davon abrücken, alle E-Mails im Postfach auf ihre Relevanz für die Anfragen hin zu prüfen und entsprechend zu importieren. Stattdessen entschieden wir uns für die Lösung, Nutzer\_innen vorzuschlagen, die E-Mails im Vorhinein in einen separaten Ordner vorzusortieren, sodass wir dann nur noch relevante E-Mails ihren jeweiligen Anfragen zuordnen mussten.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Der Quelltest und die veröffentlichten Apps sind auf GitHub verfügbar:

<https://github.com/datenanfragen>

Die Webseite des Projekts, wo auch die App online genutzt werden kann, befindet sich hier (auf Deutsch, weitere Sprachen über das Menü aufrufbar):

<https://www.datenanfragen.de>

Eine Dokumentation mancher Probleme der Entwicklung und andere Veröffentlichungen auf Social Media finden sich im Fediverse:

[https://chaos.social/@dev\\_at\\_datarequestsORG](https://chaos.social/@dev_at_datarequestsORG)

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Insgesamt stimmte die grobe Einschätzung in der ursprünglichen Projektplanung. Auch wenn das Refactoring deutlich mehr Zeit in Anspruch nahm, als ursprünglich eingeschätzt, konnten die dadurch erzielten Vereinfachungen die Entwicklungszeit für spätere Teilaufgaben verringern.

Den Administrationsaufwand und den Arbeitsaufwand an den projektbezogenen Veranstaltungen teilzunehmen unterschätzten wir allerdings. Schlussendlich konnten nicht alle geplanten Arbeitsschritte innerhalb der geförderten Arbeitszeit erledigt werden, sodass einige Arbeiten ehrenamtlich und unbezahlt erfolgten.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Es gab keine Entwicklungen bei anderen Stellen, die die Arbeit am Projekt beeinflussten.

# Richtlinie zum „Software-Sprint“

## OpenMLKit – Open ML Kit für Android

---

### *Schlussbericht*

Zuwendungsempfänger: Katharina Rasch

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S21 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### Kurze Darstellung der Aufgabenstellung und Motivation

#### *Motivation*

Auf maschinellern Lernen bzw KI basierende Funktionen sind mittlerweile vom Smartphone nicht mehr wegzudenken. Google Lens erkennt und übersetzt automatisch Schrift in Bildern und hilft uns so, Straßenschilder in fremden Sprachen zu lesen. Beim Fotografieren mit dem Smartphone hilft die voll integrierte Gesichtserkennung der Kamera beim Fokussieren. Die in iOS integrierte Geräuscherkennung macht Menschen mit Hörschäden auf die Türklingel aufmerksam und hilft so, die Welt barrierefreier zu machen.

Allen drei Beispielen ist gemein, dass sie proprietärer Teil des Betriebssystems bzw. der vorinstallierten Apps sind. Wir wissen nicht, welche ML Modelle verwendet werden und welche Schwachstellen/Biases diese haben. Und wir können uns nicht sicher sein, ob nicht doch Bilder, Sound, Videos zur Verarbeitung in die Cloud geschickt werden.

Offene Alternativen sind gefragt -- aber leider gar nicht so einfach zu entwickeln. Insbesondere die Verwendung von ML/KI-Modellen direkt auf dem Endgerät (ohne Cloud) ist technisch immer noch sehr anspruchsvoll und erfordert tieferes Hintergrundwissen. Genau hier setzt das Projekt an.

#### *Herangehensweise*

Ziel ist, dass Android Entwickler\*innen nicht zu Closed Source Bibliotheken wie Google ML Kit greifen. Dafür muss es Open Source Lösungen geben, die genau so einfach zu verwenden sind. Die entwickelte Android-Bibliothek "Open ML Kit" versucht, genau diese Lücke schließen und ermöglicht es Android-Entwickler\*innen, KI-Funktionalität wie Texterkennung mit wenigen Zeilen Code in ihre Apps einzubinden.

Zum Glück musste im Projekt dafür nicht bei Null angefangen werden. Mit Tensorflow Lite und PyTorch Mobile gibt es Bibliotheken, um Machine Learning Modelle aufs Telefon zu bringen. Deren Zielgruppe sind allerdings ML Entwickler\*innen und sie verlangen einiges an Fachwissen. Projektaufgabe war es, all diese Komplexität hinter sehr einfachen Schnittstellen zu verstecken ("Bild rein -> Text raus").

Was noch wichtig ist: Alles läuft direkt auf dem Smartphone, nichts wird in die Cloud gesendet. Ziel sind kleine Modelle, so dass auch ältere Geräte unterstützt werden.

### *Geplante Milestones*

1. Genaueres Studieren / Ausprobieren des ML Kit von google
2. Evaluation von Tensorflow Lite und PyTorch Mobile anhand einfachen Use Cases (vermutlich Bildklassifizierung); Entscheidung für eines als Basis für die Bibliothek
3. Implementierung komplexerer Use Cases um das richtige Abstraktionsniveau für die Bibliothek zu finden (sowohl Bilderkennung als auch Textverarbeitung da unterschiedliche Ansprüche an Programmierschnittstelle)
  - Bild: vmtl Erkennung von Text in Bildern
  - Text: Erkennen der Textsprache, Entity Extraction?
4. Evaluierung durch Android Entwickler\*in
5. Konzept um ML community bei Umsetzung weiterer Use cases einzubinden -- Stichwort "Model zoo"
6. Falls Zeit: Integrierung weiterer Use cases

### *Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“*

Open ML Kit für Android ist ein Beitrag zur Civic Tech Infrastruktur. Es erlaubt Android-Entwickler\*innen, Apps für Endnutzer\*innen zu entwickeln, die wichtige KI-Funktionalitäten bieten, ohne dass Online-Dienste (die damit Zugriff auf persönliche Daten und Bilder bekommen) oder proprietäre Bibliotheken (bei denen man nicht genau weiß, wie die KI-Methoden eigentlich implementiert sind) genutzt werden müssen.

### *Ausführliche Darstellung der Ergebnisse*

Mit Ende des Förderzeitraums wurden folgende Ergebnisse veröffentlicht:

#### *Ergebnis 1: Android-Bibliothek 'openread'*

Im Unterschied zu klassischer OCR (Optical Character Recognition; für die Digitalisierung von Büchern und Dokumenten, Text ist üblicherweise schwarz auf weiß), konzentriert sich 'openread' auf die sogenannte Scene Text Recognition (komplizierte Hintergründe, Mischung aus verschiedensten Fonts und Farben; nützlich zum Beispiel wenn man Telefonnummern aus Schildern kopieren oder sich Speisekarten live übersetzen lassen will, siehe Google Lens / Google Translate Apps).



Die Texterkennung in 'openread' läuft folgendermaßen ab: Zunächst wird das eingeschickte Bild (Bitmap) auf 720 Pixel Breite und 960 Pixel Höhe vergrößert bzw. verkleinert. Ein erstes Deep Learning Modell, das Textfindemodell, produziert anhand dieses Bildes eine Segmentierung / Heatmap, welche für jeden Pixel des Bildes besagt, ob dieser Pixel wahrscheinlich Teil eines Buchstaben ist oder nicht. Aus dieser Heatmap werden in einem nächsten Schritt Rahmen (sogenannte Bounding Boxes) um jedes der Wörter auf dem Bild berechnet. Als nächstes wird für jedes gefundene Wort ein entsprechender Bildausschnitt mit Fokus auf dieses Wort angefertigt und in einem letzten Schritt an ein zweites Deep Learning Modell, das Textlesemodell, geschickt. Dieses gibt das gefundene Wort heraus.

Alle Schritte laufen direkt auf der Telefonhardware, keine Internetverbindung ist nötig. Für die Ausführung der Modelle auf dem Telefon wird [tensorflow-lite](#) verwendet. Der erste Schritt, das Textfindemodell, wurde bereits für die Telefonhardware optimiert und braucht auf einem Mittelklasse-Smartphone von 2019 circa eine Sekunde. Das Textlesemodell hingegen wurde noch nicht optimiert und braucht nochmal circa eine Sekunde pro gefundenes Wort.

Die Bibliothek kann mit zwei Zeilen Code als Dependency in beliebige Android Apps eingebunden werden. All die oben beschriebene Komplexität wird durch die Bibliothek verborgen. Entwickler\*innen brauchen nur eine Funktion anzurufen und dieser ein Bild als Bitmap zu übergeben. Die Funktion liefert eine Liste von gefundenen Wörtern (Bounding Box und das Wort an sich als String) zurück.

### *Ergebnis 2: Demo-App 'openlens'*

Eine Demo-App namens 'openlens' zur Demonstrations- und Dokumentationszwecken. In der App können Nutzer\*innen ein Bild aus ihrer Galerie auswählen. Dann startet automatisch die Texterkennung. Gefundene Wörter werden mit einem Rahmen markiert und der gefundene Text auf das Bild geschrieben.

Eine Demo-App ist im Android-Bereich eine übliche Art, eine Bibliothek zu dokumentieren. Entwickler\*innen können im Sourcecode der App direkt sehen, wie man die Bibliothek einbindet und benutzt.

### *Ergebnis 3: Für Telefonhardware optimiertes Textfindemodell*

Grund für diese Eigenentwicklung war, dass das ursprünglich eingesetzte [CRAFT](#)-Textfindemodell nur sehr eingeschränkt auf Telefonhardware funktioniert hat. Insbesondere konnte es nur auf sehr kleinen Bildern (320x240 Pixeln) ausgeführt werden und war trotzdem sehr langsam mit circa 20 Sekunden pro Bild. Schwerwiegender als die Langsamkeit war allerdings, dass bei so kleinen Bildern nur sehr großer Text überhaupt erkennbar bleibt.

Das veröffentlichte optimierte Modell ist eine Neuimplementierung von CRAFT mit einer weitaus leichtgewichtigeren Modellarchitektur. Da leider der Code zur Generierung der nötigen

Trainingsdaten und für das eigentliche Training von den CRAFT-Autor\*innen nicht veröffentlicht wurde, musste dieser neu implementiert werden (und wurde natürlich auch von mir veröffentlicht).

## Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Bibliothek und Demo-App sind bereits benutzbar. Die Texterkennung ist noch etwas langsam und auch die Erkennungsqualität noch verbesserungswürdig. Meines Wissens nach ist openread/openlens allerdings die bereits am besten funktionierende Open Source Alternative für Scene Text Recognition auf Android Smartphones.

Pläne gibt es sowohl für die Weiterentwicklung der Bibliothek als auch der App:

- Das Textlesemodell ist noch nicht für Smartphonehardware optimiert und sehr langsam. Plan ist es, hier wiederum auf eine leichtgewichtige Modellarchitektur umzusteigen. Dafür wird es allerdings nötig sein, auch dieses Modell neu zu trainieren. Glücklicherweise kann hier, im Gegensatz zum Textlesemodell, schon auf einigen Code aus der Community zurückgegriffen werden.
- Das Textfindemodell und das Textlesemodell sind nicht gut aufeinander kalibriert, was die Erkennungsrate schmälert. Um die Erkennung zu verbessern, wird es nötig sein, das Textlesemodell direkt auf Ausschnitten zu trainieren, die vom Textfindemodell berechnet wurden.
- Sobald die Erkennung verbessert / beschleunigt wurde, wird die Demo-App offiziell veröffentlicht (zur Zeit muss man die App selber kompilieren und manuell installieren). Veröffentlichung ist geplant auf jeden Fall für den F-Droid Store, potentiell auch für den Play Store, falls von Google erlaubt.

Ich werde diese Weiterentwicklungen zunächst erst einmal selbstfinanziert vorantreiben, kann mir aber mittelfristig auch vorstellen, noch andere Fördermittel für die Weiterführung der Entwicklung zu beantragen.

## Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Alle durchgeführten Arbeiten waren für die veröffentlichten Ergebnisse relevant. Es gab keine Arbeiten, die zu keiner Lösung geführt haben bzw. nicht weiterverfolgt wurden.

## Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Sowohl die Bibliothek als auch die Demo-App finden sich unter <https://gitlab.com/krasch/openread>. Der Trainingscode für das Textlesemodell ist in der Readme verlinkt.

## Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es gab eine Abweichung bei der Arbeitsplanung: Statt mindestens zwei Use Cases zu implementieren, wurde nur der Use Case "Texterkennung" implementiert. Dies hatte mehrere Gründe:

- Nach Review des Google ML Kit wurde klar, dass Texterkennung eine der wichtigsten Anwendung für ML/KI auf dem Smartphone ist. Text, der gelesen, digitalisiert, übersetzt werden muss, findet sich einfach überall.
- Gleichzeitig ist Texterkennung ein relativ komplizierter Use Case und involviert in der von mir gewählten Lösung das Zusammenspiel zweier Modelle. Priorisierung von Texterkennung hieß also für mich auch: leider keine Zeit für die Implementierung anderer Use Cases.
- Auf der anderen Seite hat Texterkennung aber den riesigen Vorteil, dass Trainingsdaten frei verfügbar sind (im Gegenteil zu vielen anderen ML Anwendungen, wo man außerhalb großer Firmen kaum Chancen hat an passende Trainingsdaten zu kommen). Insbesondere gilt für die Modelle, die ich in meiner Bibliothek einsetze, dass sie komplett auf synthetischen Daten trainiert werden können. Konkret bedeutet das: man nimmt ein Hintergrundbild ohne Text, schreibt Text drauf und trainiert das Modell dann darauf, diesen Text wiederzufinden. So lassen sich im Prinzip unendliche Mengen an Trainingsdaten generieren.

Da es zur Zeit keine konkreten Pläne zur Einbindung weiterer Use Cases gibt, wurde noch nicht mit dem Schritt "Aufbau einer Community von Mitentwickler\*innen" begonnen. Aktuell ist der Plan, die selbsttrainierten Modelle bei <https://huggingface.co/> zur Verfügung zu stellen, da dort bereits eine große Gemeinschaft von KI-Entwickler\*innen miteinander Modelle teilt.

Abgesehen von diesen Abweichungen wurde der Arbeitsplanung und der Kostenplanung gefolgt.

## Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Meine Arbeit baut auf Forschungsarbeiten des ClovaAI Lab zur Texterkennung auf. Insbesondere:

- Textfindemodell CRAFT <https://github.com/clovaai/CRAFT-pytorch>
- Textlesemodell <https://github.com/clovaai/deep-text-recognition-benchmark>

# QuellenReiter:in – Die “Fake News” Quiz-App

## Schlussbericht

Zuwendungsempfänger:

Sasse & Werner GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S22 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

## Kurze Darstellung der Aufgabenstellung und Motivation

Medienkompetenz ist eine Grundfähigkeit, um sich in der komplexen Medienlandschaft der heutigen Informationsgesellschaft zurechtzufinden. Der Anteil der Menschen, die sich in ihrem Alltag über das Internet informieren, steigt stetig an. Die Vertrauenswürdigkeit einzelner Quellen nachvollziehbar einstufen zu können, ist eine große gesellschaftliche Herausforderung.

Mit der App “QuellenReiter:in” sollen Nutzer:innen reale, belegte Nachrichten und getarnte Desinformationen vorerst bewerten und anschließend anhand von Quellen und Faktenchecks vergleichen. So lernen Spieler:innen “Fake News” zu entlarven und erhalten einen Überblick über die deutsche Medienlandschaft.

“QuellenReiter:in” verfolgt dabei mehrere Ansätze, um die Medienkompetenz der Menschen zu fördern:

- **Hinterfragen:** Um zu gewinnen, müssen Nachricht und Quelle hinterfragt werden. Die Entscheidung, ob man einer Nachricht glaubt oder nicht, hat also, wie in der Realität, Konsequenzen.
- **Quellen kennenlernen:** Beispielhaft dazu werden zu jeder Aussage (egal ob richtig oder falsch) mindestens zwei Quellen verlinkt, bei denen die Nutzer:innen weitere Informationen und Hintergründe finden können.
- **Medienlandschaft erkunden:** Durch eine Vielfalt an verschiedenen Aussagen und Quellen, wird ein Überblick über vertrauenswürdige Medien gegeben. Das hilft, die Medienlandschaft in Deutschland besser kennenzulernen.
- **Fakes wiedererkennen:** Viele Desinformationen kursieren immer wieder und in abgewandelter Form im Netz. Spieler:innen entwickeln ein Gespür für “Fake News” und erkennen diese im Internet wieder.



- **Funfaktor:** Es macht Spaß, ist kompetitiv und spricht die User:innen da an, wo sie sind: im Internet.

Auch wenn das Spiel durch die Quiz-Mechanik durchaus eine informierende und bildende Funktion mit sich bringt, dient es nicht der Nachrichtenbeschaffung. Da wir für die Auswahl der Meldungen und den dazugehörigen Quellen jedoch redaktionelle, journalistische Arbeit leisten, wollen wir hier unsere Arbeitsmethoden offenlegen.

### **Die Auswahl der Schlagzeilen:**

- Die Faktenlage zur Schlagzeile muss nachprüfbar sein.
- Alle richtigen und falschen Informationen wurden im Internet (Blogs, Social Media, News-Webseiten, Online Magazine etc.) veröffentlicht, "Fake News" wurden nicht von uns erfunden.
- Trennung von Unterhaltung und Satire: Satire gilt nicht als Desinformation und fällt somit aus unserer Auswahl heraus. Wenn allerdings Satire nicht als solche gekennzeichnet wird und von der Mehrheit der Rezipient:innen auch nicht als solche erkannt wird, kann sie als Desinformation verstanden werden.
- Die Aussagen müssen sich in ihrem Inhalt und in ihrer Form für ein Quiz eignen

### **Die Auswahl der Quellen:**

Die Quellen müssen vertrauenswürdig sein. Dies gilt, wenn das Medium:

- etabliert ist und
- nach journalistischen Standards arbeitet (siehe als Grundlage dazu: <https://www.presserat.de/pressekodex.html> oder bei Faktencheckredaktionen: <https://ifcncodeofprinciples.poynter.org/> ).
- in der Vergangenheit nicht durch eine fehlerhafte Recherche aufgefallen ist.

### **Die Darstellung der Informationen im Spiel:**

- Jede Aussage sollte mit mindestens zwei Quellen von verschiedenen vertrauenswürdigen Medien belegt werden.
- Es gibt Angaben zu Medienart, Datum, Sprache und Autor:in
- Bei Quellen wird der Inhalt doppelt verlinkt. Ein Link führt auf die aktuelle Version der Quelle, ein Link führt auf eine archivierte Version.
- Bei Desinformationen werden ausschließlich archivierte Versionen der Aussage verlinkt und mit einem Disclaimer versehen.

### **Open-Data-Strategie**

Der App liegt eine standardisierte Datenbank zugrunde, welche alle News und "Fake News" beinhaltet. Aufgelistet werden: Schlagzeile, Autor:in, Plattform, Datum, Sprache und Link. Außerdem gibt es die passenden Fakten Checks, welche den "Fake News" beigefügt werden. Diese Datenbank wird für alle Internetnutzer:innen zugänglich sein. Mit der Suchfunktion können "Fake News" gefunden werden – mit den dazugehörigen Faktenchecks.

## Ursprüngliche Meilensteine

In dem Projektzeitraum von sechs Monaten sollen die folgenden Punkte erreicht werden:

- Recherche von 500 spielbaren Aussagen, davon:
  - 50 % Desinformationen mit dazugehörigen Faktenchecks
  - 50 % korrekte Aussagen mit dazugehörigen Faktenchecks
- Austausch mit Schulen und Expert:innen im Bereich der Pädagogik und Medienkompetenzvermittlung
- Aufbau einer Datenbank zur Verwaltung der Desinformation und Faktenchecks
- Entwicklung eines webbasierten Frontends zur Einsicht der Datenbankinhalte
- Entwicklung einer plattformübergreifenden mobilen Quiz-App
- Testen und Evaluation der App an Schulen und anderen Bildungseinrichtungen, ggf. im Rahmen von Workshops oder Medienkompetenz-Unterrichtseinheiten

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Die Quiz-App „QuellenReiter:in“ soll gezielt und frei zugänglich Nachrichtenkompetenzen in der Zivilgesellschaft fördern. Die App ist ein niederschwelliges, werbefreies und ausschließlich digitales Lehrmittel. Wir wollen mit der App eine Plattform schaffen, auf der sich User:innen auch außerhalb des schulischen oder akademischen Kontexts mit dem Thema Desinformation auseinandersetzen.

## Ausführliche Darstellung der Ergebnisse

**Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?**

Um die Recherchearbeit zu erleichtern, wurde zu Beginn des Förderzeitraums direkt die „Fake News“ Datenbank aufgesetzt. Dafür wurde mit Flutter eine Webapp implementiert, die es authentifizierten Benutzer:innen ermöglicht, Schlagzeilen in die Datenbank einzupflegen. Außerdem ist die Datenbank eine Suchmaschine für Internetnutzer:innen. Diese Webapp ist online unter <https://fakten.quellenreiter.app> erreichbar.

Die Fake News Quiz App „QuellenReiter:in“ ist ebenfalls fertig und wurde am 31.08.22 für Android und iOS veröffentlicht. In den ersten Wochen gab es bereits mehrere hundert aktive Nutzer:innen.



Einzelnen Meilensteine wurden zum Teil aufgrund der zeitlichen Umsetzbarkeit angepasst. 500 spielbare Aussagen einzupflegen, war wegen des Rechercheaufwandes nicht möglich. Stand September 2022 gibt es knapp 200 spielbare Aussagen. Die Datenbank wird jedoch weiterhin erweitert. Das Testen der App an Schulen hat leider nicht geklappt, weil die Fertigstellung der App im Juli und August genau in die Sommerferien der Länder fiel. Wir sind jedoch im Kontakt mit dem Johanneum Lüneburg, um die App im Unterricht vorzustellen und zu testen. Außerdem wurden mehrere User:innen-Tests zu unterschiedlichen Zeitpunkten an kleinen Gruppen durchgeführt. Über die Meilensteine hinaus haben wir uns intensiv mit dem Thema Datenschutz beschäftigt, da dies für Schulen besonders wichtig ist. Beispielsweise haben wir uns nach Gesprächen mit Lehrer:innen gegen eine Abfrage von E-Mail-Adressen oder Telefonnummern beim Login entschieden. Unsere App ist DSGVO-konform und sammelt beinahe keine sensiblen Daten. Ausschließlich Nutzer:innenname, Passwort und Geräte ID.

## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Die App "QuellenReiter:in" soll vor allem junge Menschen ansprechen und ihnen die Verantwortung, die sie im Netz tragen, vor Augen führen. Sie ist jedoch auch für Erwachsene interessant, die generell Apps und Smartphone-Games gegenüber aufgeschlossen sind. Die App eignet sich für Spieler:innen ab 13 Jahren.

Für Schüler:innen und junge Erwachsene ist die App auf der einen Seite Unterhaltung. Auf der anderen Seite fördert sie die Informationskompetenz. Die App kann auch im schulischen Kontext benutzt werden, zum Beispiel als ergänzendes Lernmaterial zur Vermittlung von Nachrichten- und Informationskompetenz.

Die Desinformationsdatenbank ist als Recherchetool auch für Journalist:innen interessant, um kursierende "Fake News" zu durchsuchen und sich einen Überblick über die Faktenchecks zu verschaffen. Internetnutzer:innen können den Fact Browser als eine Art "Fake News"-Google verwenden.

### **Weiterentwicklung der Idee**

Es gibt bereits Vorschläge für Erweiterungen der App, wie beispielsweise einen Singleplayer-Modus oder eine verbesserte Einladefunktion über Deeplinks. All dies soll entweder von uns oder von der Open Source Community umgesetzt werden. Dafür wird noch ein einfaches Setup für die Entwicklung benötigt, da die Funktionen der App nur mit einer Anbindung an einen Entwicklungsserver getestet werden können.

Mit einer Art "Faktencheck-Hub" könnten wir unsere "Fake News"-Datenbank erweitern. Während die Datenbank aktuell nur die Möglichkeit bietet, "Fake News" mit dazugehörigen Faktenchecks zu durchsuchen, soll das Faktencheck-Hub die Nutzer:innen und Faktencheck Redaktionen in den Prozess der Erweiterung der Datenbank miteinbeziehen. Das Hub macht es verifizierten Redaktionen möglich, ihre Faktenchecks in die Datenbank selbst einzutragen, oder bereits bestehende Faktenchecks mit eigenen Quellen zu ergänzen. Außerdem bietet das Hub eine Art "Ticket System" an, welches einen Austausch zwischen Nutzer:innen und Redaktionen ermöglicht.



## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

User:innen-Tests an Schulen waren im Förderzeitraum wegen der Sommerferien nicht möglich, weshalb wir uns für User:innen-Tests in kleinen Gruppen entschieden haben. Die Tests an Schulen sollen deshalb nachgeholt werden. Die Erkenntnisse daraus können dann mit Software-Updates nachträglich eingearbeitet werden.

Um Nutzer:innen darüber zu informieren, dass sie bei einem Spiel an der Reihe sind oder eine neue Freund:innen-Anfrage haben, wollten wir ursprünglich auf Pushmitteilungen verzichten, da diese die Geräte ID benötigen und wir so wenig Daten wie möglich speichern wollten. Da es auf den Zielplattformen Android und iOS nicht möglich war, die App regelmäßig im Hintergrund zu starten und nach Updates suchen zu lassen, haben wir uns am Ende doch für die Pushmitteilungen entschieden. Diese können aber natürlich abbestellt werden.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Webseite von QuellenReiter:in: <https://quellenreiter.app/>

Fact Browser: <https://fakten.quellenreiter.app/>

GitHub: <https://github.com/QuellenReiter>

Podcastfolge über unser "Fake News" Quiz:

<https://shows.acast.com/podfaktisch/episodes/wie-du-fake-news-erkennst-quellenreiterin-das-fake-news-quiz>

Android App:

[https://play.google.com/store/apps/details?id=com.quellenreiter.quellenreiter\\_app](https://play.google.com/store/apps/details?id=com.quellenreiter.quellenreiter_app)

iOS App:

<https://apps.apple.com/de/app/quellenreiter-in/id1641339420?uo=4>

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Programmierung der App und die Recherche der Inhalte dauerten länger als erwartet, sodass wir im Förderzeitraum deutlich mehr Zeit in die Umsetzung investiert haben, als vorher angenommen. Das hat die Kostenplanung jedoch nicht beeinträchtigt.

# Richtlinie zum „Software-Sprint“

Pamaxie

---

## *Schlussbericht*

Zuwendungsempfänger:

Lia Dürr

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S23 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

### **Kurze Darstellung der Aufgabenstellung und Motivation**

Unsere Motivation entstand durch das Verlangen eine Open-Source alternative zu Google und Amazons Content Erkennungs Systeme für anstößige Inhalte zu finden.

Wir wollten hiermit zeigen, dass es für jeden möglich ist so ein System auf eigenen Servern laufen zu lassen, ohne hohe Kosten tragen zu müssen. Zudem wollten wir veranschaulichen das die Sicherheit durch geschlossene Systeme (auch in der Welt von Neuronalen Netzen) nicht immer besser ist als eine open-source alternative.

Um dies zu erreichen mussten wir eine Micro-Service Architektur aufbauen, die zuverlässig Bilder verarbeiten, scannen und das jeweilige Hash dieser Bilder zum Caching in eine Datenbank ablegen konnte.

Hier wurde eine Lösung entwickelt die als Microservice via Docker auf Infrastruktur ausgerollt werden kann. Dies ermöglicht es Projekten Problematische Inhalte (Inhalte mit Pornographie oder Blut) zu erkennen. Via eines API reuest kann hier ein Bild (im Binären Format) oder als Link eingereicht werden und es wird ein Resultat via JSON zurückgegeben.

Das Resultat (s. Bild) zeigt an ob ein Bild Anstößige Inhalte oder Zeichen wie Hakenkreuze beinhaltet.

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
Pretty Raw Preview Visualize JSON
{
  "Key": "11529218840461521683",
  "DataType": "Image",
  "DataExtension": "png",
  "ScanMachineGuid": "ebbcdee6-2677-4286-a252-551b6a99db33",
  "IsUserScan": true,
  "ScanResult": {
    "Key": "11529218840461521683/-/imgScanResult",
    "DetectedShieldLabel": "Pornographic",
    "DetectedShieldLabelLikelihood": 99.91901,
    "DetectedShieldLabelAnnotation": null,
    "DetectedShieldLabelAnnotationLikelihood": 0.0,
    "DetectedSymbolLabel": "None",
    "DetectedSymbolLikelihood": 99.99339,
    "AllShieldData": {
      "GoreLikelihood": 1.4815545E-05,
      "PornographicLikelihood": 99.91901,
      "RacyLikelihood": 0.06647685,
      "NoneLikelihood": 0.01450514,
      "DetectedLabel": "Pornographic",
      "NetworkVersion": 1,
      "VersionCommonName": "Anthelina",
      "ScanPPI": 0,
      "WasCompressed": false,
      "ScanDate": "2022-04-12T21:20:34.785113+00:00"
    },
    "AllShieldAnnotationData": null,
    "AllSymbolData": {
      "HitlerLikelihood": 3.4158876E-07,
      "SwastikaLikelihood": 0.006605523,
      "NoneLikelihood": 99.99339,
      "DetectedLabel": "None",
      "NetworkVersion": 1,
      "VersionCommonName": "Anthelina",
      "ScanPPI": 0,
      "WasCompressed": false,
      "ScanDate": "2022-04-12T21:20:34.838061+00:00"
    }
  }
}
```

## Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“

Da unsere Lösung offen und im Verbund mit unserem Ziel-Plattformen und unserer Community entwickelt wird, konnten wir neue Erkenntnisse in Problemen der Content-Erkennung machen. Was dazu führt dass wir hier eine gesellschaftliche Weiterentwicklung in Sachen Wahrnehmung für Content-erkennung mit kultureller Verständnis erreichen konnten.

## Ausführliche Darstellung der Ergebnisse

### Erkennung von Inhalten / Trainieren des Neuronalen Netzes

Die Erkennung von Inhalten funktioniert momentan durch eine relativ simple Aktivierungs-methode. Hier muss noch nachgebessert werden (wir wollen hier auf ResNet 16 umsteigen). Die primäre Problematik, die wir in der Erkennung unserer Systeme momentan feststellen, ist das hier ein Bias zu echten Inhalt (bedeutet Bilder mit echten Menschen) vorliegt. Dadurch ist die Erkennung von gezeichnetem Material wesentlich schlechter, und benötigt Nachbesserung.

Die Erkennung von Pornographie, Gore (Blut und ähnliches) und Anstößigen Inhalten ist soweit ziemlich gut, jedoch ist hier auch zuzugestehen, dass die Leistung noch durch ResNet 16 verbessert werden kann. Wir hatten hier vor dem Umstieg auf Python noch bessere Ergebnisse (um die 90% Genauigkeit). Diese ist hier eher um die 86%, dies ist auch eventuell darauf zurückzuführen, dass viel von unserem Material durch eine Daten-Panne leider korruptiert wurde.

Die Erkennung von „hass-Symbolen“ erstellt sich als problematisch, durch die kulturellen Einflüsse. Ein Hakenkreuz ist sehr einfach als „schlecht“ einzustufen, aber viele Symbole /Personen haben in der momentanen Kultur eine komplexere Meinung.

Diese muss noch genauer erforscht werden und hier ist eventuell ein Rebranding erforderlich auf „unerwünschte Symbolik“ um keine Meinung zu einem Symbol anzudeuten.

An sich lässt sich sagen, dass die Erkennung von Bildern funktioniert, jedoch unser Ziel nicht ganz erreicht werden konnte. Wir haben keine Erkennung zum Kulturellen Hintergrund in der Zeit der Förderphase entwickeln können, und sind hier noch in der Forschung inwiefern dies mit den Mitteln die wir haben möglich ist.

## Datenbank API

Die Entwicklung unserer Datenbank-API war eine der ersten Sachen die wir fertig stellen konnten. Hier wurde unser ursprünglicher Gedanke zur Speicherung mit einem flexiblen Daten-Backend erfolgreich umgesetzt. Jedoch haben wir festgestellt, dass dieses „flexible“ Daten backend sehr viele Probleme mit sich führt. Zudem hätten wir unsere API nicht auf REST, sondern auf GraphQL Basis entwickeln sollen, da dies für ein Daten-Backend wesentlich besser geeignet ist.

Die API skaliert sehr gut, und erlaubt es Benutzern Instanzen anzusteuern. Die Konfiguration der Datenbank muss noch vereinfacht werden. Viele Probleme entstehen auch durch das mögliche verwenden von anderen Datenspeichern, dadurch wird der Entwicklungsaufwand sehr viel komplexer. Wir glauben, dass wir hier eine Vereinfachung unserer Datenbank API in nächster Zeit vor dem „release“ unserer Software entwickeln müssen. Dies liegt daran das die hohe Komplexität den Zeitaufwand zu sehr in die Höhe treibt.

Hier wollen wir auf ein Festes Daten backend und GraphQL setzen. Zudem werden wir unsere Plattform in der Zukunft nur noch über Linux und Docker anbieten. Wir wollen zudem die Datenbank Komplexität reduzieren, was hierdurch (ohne wechselbare DB-Treiber / Datenspeicher) einfacher wird.

Ein weiterer Problempunkt ist das momentane Erstellen von Hashes auf Datenbank ebene, das sollte in Zukunft der Scan-Client übernehmen. Die Hashing Funktion die wir momentan für ein „similarity Hash“ verwenden, ist leider nicht akkurat genug und hier wollen wir auf pHash umsteigen. Damit lässt sich die Zeit-Verzögerung je Anfrage zudem auch noch reduzieren.

## Scan API

Unsere Scan API wurde soweit auch implementiert wie wir uns dies vorgestellt haben, jedoch gibt es hier Probleme mit der Kommunikation mit den individuellen Scan-Clients. Wir haben leider bisher noch keine gute Lösung für die echtzeit-kommunikation gefunden, die auch zuverlässig prüft ob Arbeit wirklich abgearbeitet wurde. Hier ist es eventuell von Nöten eine eigene Lösung zu implementieren, parallel sind wir uns auch momentan RabbitMQ am Anschauen. Bei RabbitMQ haben wir jedoch von branchen-kollegen von Problemen der Zuverlässigkeit gehört. Momentan setzen wir auf AmazonSQS, was an sich gut funktioniert, aber leider nicht unseren Wunsch nach einer offenen, selbst hostbaren Lösung Ermöglicht.

Die Verarbeitung der Bilder dauert momentan noch relativ lange, das liegt daran das wir auf viele verschiedene Endpunkte angewiesen sind. Hier ist weiteres Testen von Nöten um eventuelle „Zeit Fresser“ zu erkennen und zu beseitigen.

Die momentane zeit für den Scan eines Bildes dauert 4 Sekunden. Jedoch wird die meiste Zeit damit „verschwendet“ mit anderen Services zu reden. Hier kann durch eine gute Netzwerk Kommunikation eventuell zeit gespart werden. Wir glauben auch das die momentane Message-queue nicht wirklich schnell genug ist um kurze Scan Zeiten zu gewährleisten.

Wir glauben jedoch, dass unserer Scan-Client gut skaliert und haben damit bereits Tausende von Bildern gescannt ohne größere Fehler zu haben. Jedoch muss gesagt werden das wir momentan nicht

prüfen ob ein Bild was uns geschickt wird auch wirklich am Ende gescannt wurde. Hier ist auf jeden Fall noch einmal nachzubessern und zu überlegen wie wir dies bewerkstelligen können, ohne auf unsere Micro-Service Architektur zu verzichten.

## Scan Client

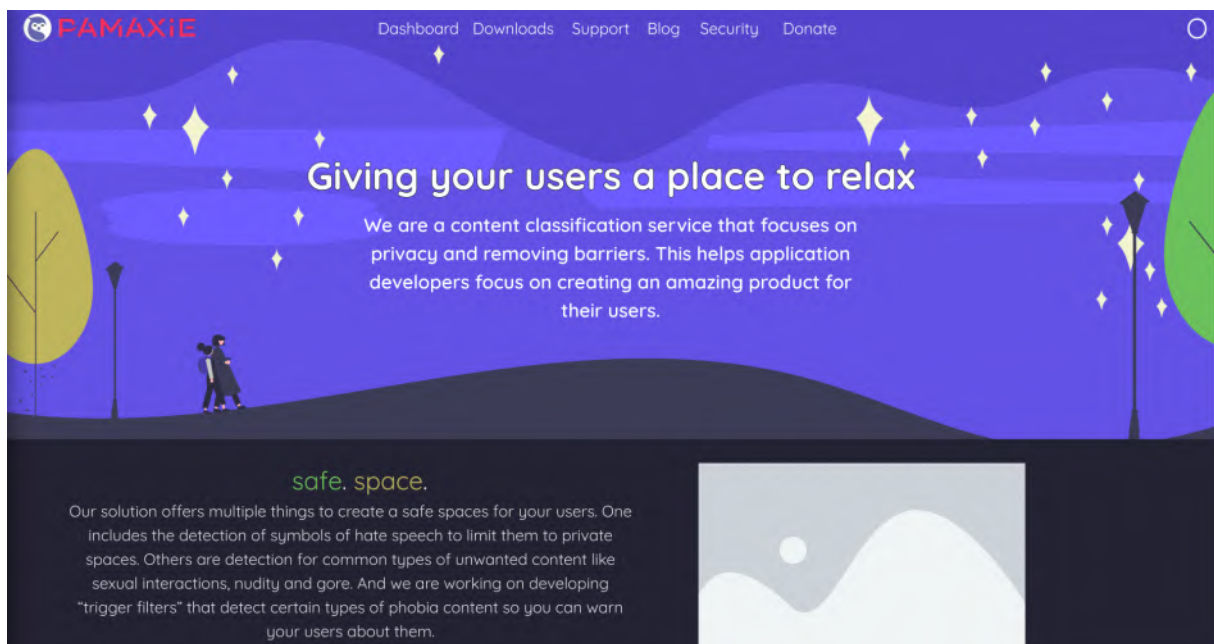
Unser Scan Client setzt momentan auf ein Python Skript auf was relativ unkompliziert auf anderen Systemen eingesetzt werden kann (Docker oder nicht macht hier wenig unterschied). Wir glauben jedoch, dass der Umstieg auf eine vorkompilierte Anwendung hier eine bessere Idee ist und werden vermutlich auf C# mit der SciSharp Toolchain setzen um den Scan Client in der Zukunft weiter zu entwickeln. Dies würde uns auch erlauben den Scan Client von Kunden einsetzen zu lassen, damit diese dann ihren eigenen Content scannen können. Sollten sie dies tun würden wir diese Erkennung jedoch nicht Cachen.

Damit kann die Last in Zukunft besser verteilt werden, für die momentane Version musste dieses Feature leider verschoben werden, da dies andere Probleme mit sich bringt. Eines dieser Probleme ist wieder wie man die Queue auf bestimmte Einträge filtert. Wir würden gerne auf eine Datenbank umsteigen um dies zu ermöglichen, jedoch wurde uns von einem Enterprise-Architect davon abgeraten da dies bei hohem Volumen schnell die Datenbank überlasten könnte.

## Website

Wir konnten die Entwicklung der Website leider nicht fertigstellen, da uns hierfür die Zeit nicht ausreichte. Diese befindet sich leider immer noch in der Initialen design Phase und ist momentan unser haupt-fokus zur Vervollständigung.

Leider waren die anderen Themen zu komplex um die Entwicklung der Website zu priorisieren. Im Design der Website werden wir auf eine „dunkle“ Seite mit leichten Einflüssen arbeiten um ein angenehmes Gefühl zu erreichen. Das Design ist noch in der Arbeit und hier wird mit Künstlern zusammengearbeitet um dieses zu vervollständigen. (Siehe Bild für ein Beispiel)





## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Wir glauben, dass unser Projekt durch ein paar Weiterentwicklungen Nutzer auf Plattformen effektiv schützen kann, was dazu führen würde das wir deren Privatsphäre besser achten können.

Wir benötigen mehr Entwickler und müssen unsere Komplexitäten der Microservices reduzieren, um eine Open-Source Stellung effektiv halten zu können. Dies liegt daran das das momentane einarbeiten in das Projekt sehr komplex ist. Zudem würden wir gerne in der Zukunft auch mehr als nur Bilder erkennen, jedoch sollte davor sichergestellt werden das unser Projekt einfacher erweiterbar ist und die bild-erkennung gut genug funktioniert.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Unsere Arbeiten haben alle zu Lösungen geführt, oder haben dazu geführt dass wir noch nach Lösungsansetzen suchen. Wir konnten feststellen, dass unser Projekt so auf jeden Fall realisierbar ist.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

Nutzer können uns über unseren GitHub ([github.com/pamaxie](https://github.com/pamaxie)), die verbundenen Emails auf diesem GitHub oder unsere Website [pamaxie.com](https://pamaxie.com) erreichen.

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Durch lange Arbeitszeiten, da viel Tooling nicht existiert, wurden einige Arbeiten verschoben oder noch nicht durchgeführt. Die jeweilige Arbeitszeit war leider zu kurz um alle Projektziele 100% zu erfüllen, jedoch konnten wir ein Prototyp entwickeln, der sicherstellt das es möglich ist hier eine gute Lösung aufzubauen. Wir werden uns nach weiteren Fördermaßnahmen umschaun um die Weiterentwicklung unseres Projektes sicherzustellen, fertig glauben wir ist unser Projekt nie. Die Personalkosten konnten im Projekt gut gedeckt werden und stellten kein Hindernis für die Entwicklung unseres Projektes da.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Die Entwicklung von Transfer-Learning ist für unser Projekt interessant und wir würden hier gerne den nutzen evaluieren. Jedoch fehlt uns momentan die Rechen Leistung durch die gegebenen kosten dieser.

Es gab viel kulturelle Forschung bezüglich internet-kultur jedoch wurde das Thema Moderation / Content Scannung nicht von vielen aufgegriffen. Wir haben hier von keinen Weiterentwicklungen gehört die uns bei unserem Projekt helfen / unser Projekt Thema betreffen.

# Richtlinie zum „Software-Sprint“

## master\_me (vormals soundsgood)

---

### *Schlussbericht*

Zuwendungsempfänger:

Klaus Scheuermann

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S24 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Vorbemerkung: In Absprache mit dem Prototype Fund, wurde das Projekt umbenannt in 'master\_me'. Der Name ist passender und findet mehr Anklang.

#### **Kurze Darstellung der Aufgabenstellung und Motivation**

Mit dem ersten Lockdown in der Covid19 Pandemie 2020 stolperten viele unabhängige Konferenzen und Veranstaltungen aus dem zivilgesellschaftlichem Feld ins Digitale. Klanglich ließ das oft zu wünschen übrig. Sie konnten mit kommerziell produzierten Inhalten nicht mithalten und man konnte sich schlecht auf die Inhalte konzentrieren.

Aufgabenstellung war die Entwicklung eines Open Source Plugins zur Verbesserung der Audioqualität bei Live-Streaming-Events. Es sollte über gängige Schnittstellen in Open-Source Streaming Software einzubinden sein.

Die Problemlösung bestand darin, einen DSP (digital signal processing) Prozess zu schaffen, der Lautstärke, Frequenzgang, Dynamik und Fehler kontrolliert. Diesen galt es in ein Plugin zu gießen, welches über gängige Schnittstellen (VST, LV2, JACK) in bestehende Open-Source Streaming Software integriert werden kann. Im Vordergrund stand die Audioqualität und die einfache Usability der Software.

#### **Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“**

Zur Zielgruppe zählen alle, die live streamen und die Audioqualität des Streams verbessern möchten. Beispielsweise Live-Streams von Konferenzen, Konzerten und sonstigen Events. Zudem gibt es offenbar ein reges Interesse an master\_me aus der Podcasting-Szene. Auch eine Anwendung bei Internet-Radiosendern ist denkbar.

Durch master\_me werden Open-Source Streaming Strukturen konkurrenzfähig zu kommerzieller Software.

Selbst Menschen, die wenig Expertise im Bereich Audio haben, können durch einfaches Einbinden des Plugins eine erhebliche Verbesserung des Klangs erreichen, was auf der Empfängerseite zu besserer Hörbarkeit und letztlich zu höherer Konzentration führt.

## Ausführliche Darstellung der Ergebnisse

In der ersten Stufe wurden Anforderungen gesammelt, bestehende Module getestet, neue Module definiert und erste Tests gemacht.

Im zweiten Schritt wurde mit potenziellen Anwender\*innen Tests vorgenommen. Darunter war ein Test mit einem Mitglied des VOC (Video Operation Center) des Chaos Computer Clubs im Hackspace X-Hain um die Nutzbarkeit in der Streaming Architektur des VOC zu evaluieren. Auf der Konferenz re:publica 2022 wurde master\_me drei Tage lang an Bühne 6 im Dauerbetrieb getestet und Parameter optimiert. Das Streaming Team der re:publica (wecap) war begeistert und wird master\_me voraussichtlich bei zukünftigen Live-Stream-Events einsetzen.

Nach Fertigstellung des Plugins wurde in Stufe 3 ein Video für den Demo-Day des Prototype Fund mit Inhalten von der re:publica 2022 erstellt. Der Originalton von Bühne 6 wird mit dem, in Echtzeit bearbeiteten Sound von master\_me, im A-B Vergleich gegenübergestellt wird.  
(<https://cloud.4ohm.de/s/WowqnDkqRMHW9tf>)

Mit dem Erreichen von Stufe 4, wurde master\_me zum Demo-Day des Prototype Fund am 31.8. in verschiedenen Plugin-Formaten auf Github veröffentlicht. Dort findet sich zudem eine ausführliche Dokumentation und Anleitung. [https://github.com/trummerschlunk/master\\_me](https://github.com/trummerschlunk/master_me)

master\_me kann nun als VST-Plugin in Hostsoftware wie OBS und Ardour integriert werden.

master\_me existiert jetzt als fertige Stable-Release. Es war mir wichtig, nicht nur einen Prototypen als proof-of-concept zu schaffen, sondern eine belastbare Software, die für alle Anwender nutzbar ist.

Erkenntnisse:

Die DSP Programmierung klappte wie geplant.

Die Umsetzung in ein plattformübergreifend lauffähiges Plugin, in allen gängigen Formaten, erforderte erheblich mehr Aufwand als erwartet, da einige bestehende Architekturen der Programmiersprache Faust nicht so fehlerfrei liefen, wie gedacht.

Mitte August lief master\_me schon sehr gut, sah aber noch nicht all zu ansprechend aus. Um mehr Akzeptanz und möglichen Einsatz in der Öffentlichkeit zu erreichen, wurde ein ansprechendes GUI geschaffen, welches sich schon jetzt großer Beliebtheit erfreut.

Die Open Knowledge Foundation stand mir mit dem Team des Prototyp Fund tatkräftig zur Seite. Ein Coaching wurde absolviert, welches zentrale Fragen zu GUI und Usability herausarbeitete.



## **Zielgruppe, Nutzen und mögliche Weiterentwicklungen**

Mit master\_me klingen Inhalte besser, die mit Open-Source Software gestreamt werden. Sie werden konkurrenzfähig zu kommerziell produzierten Inhalten und sind für den Empfänger angenehmer zu hören und damit leichter zugänglich. Ein Fachwissen zum Thema Audio ist nicht nötig, wodurch keine Ressourcen unnötig auf die Optimierung des Klangs aufgewendet werden müssen.

Die Veröffentlichung als Open-Source Projekt ermöglicht die kollektiven Weiterentwicklung, welche schon jetzt Fahrt aufgenommen hat. Mehrere Entwickler tragen als Kollaborateuren zum Projekt bei. Jede\*r kann aus dem Source-Code lernen.

Eine Idee, die immer wieder für Begeisterung sorgt, ist die mögliche Implementation von master\_me in Hörgeräte. Dies sollte man auf jeden Fall weiter verfolgen.

Ich habe durch die Entwicklung von master\_me sehr viel gelernt und bin dankbar, dass ich das Projekt mit der Hilfe des BMBF, der OKF und des Prototype Funds durchführen konnte.

## **Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben**

Ursprünglich gab es die Idee, master\_me auch in Video-Konferenz Software wie Jitsi oder BigBlueButton einzubinden. Dies ist jedoch schwierig, da es keine offenen, standardisierten Schnittstellen gibt. Nun, da das Projekt veröffentlicht ist und ausprobiert werden kann, werde ich den beiden Entwickler-Communities das Projekt vorstellen und bei der Anpassung des Codes helfen, sofern Interesse besteht.

## **Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer**

[https://github.com/trummerschlunk/master\\_me](https://github.com/trummerschlunk/master_me)

Ein Video des bekannten Open-Source-Audio Youtubers 'unfa' wird sich demnächst mit master\_me beschäftigen.

## **Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung**

Die Arbeits- und Kostenplanung wurde eingehalten.

## **Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen**

Wie oben schon kurz erwähnt, gab es fehlerhafte Architectures in der Programmiersprache Faust, auf die ich mich ursprünglich verlassen hatte. Das machte viel zusätzliche Arbeit, die aber letztlich erfolgreich war. Einige dieser Architectures wurden verbessert, sodass andere Programmierer\*innen davon profitieren.