

Statistik Erfahrbar

Schlussbericht

Zuwendungsempfänger:

Name des Zuwendungsempfängers

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS19S17** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin.

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)?

Die siebte Runde des Prototype Fund brachte uns, das Team hinter dem Projekt [Datenguide](#) (gegründet in der zweiten Runde des Prototype Fund 2017/2018), zurück zu unserer ursprünglichen Mission: Statistiken für alle! Wir wollen die amtliche Statistik erklären und nutzbar machen – auch für Menschen ohne besondere Vorkenntnisse. Unsere großen Vorbilder waren dabei [„Explorable Explanations“](#): interaktive, ins Auge fallende Visualisierungen komplexer Zusammenhänge.

Ziel des Projekts Statistik erklärt war es, herauszufinden, ob und wie wir solche interaktive Darstellungs- und Vermittlungsformen einsetzen können, um Hintergründe amtlicher Statistik so erklären, dass mehr Menschen sich trauen, damit zu arbeiten. Denn die Lernkurve zur ersten Datenvisualisierung oder zum ersten Blogpost, basierend auf amtlichen Daten, ist relativ steil. Um mit amtlichen Statistiken zu arbeiten, braucht es Wissen über Datenformate, über regionale Einteilungen und natürlich über Statistik. Das Problem: NUTS-Level, amtliche Gemeindeschlüssel oder Tabellenformate wirken wahlweise Furcht einflößend oder einschläfernd. In der Bevölkerung gibt es ein fehlendes Verständnis für Grundlagen der Statistik. Die New York Times hat z.B. einen Data Literacy Kurs für ihren Newsroom entwickelt. Viele Menschen begegnen Zahlen und Statistiken mit Unbehagen und Misstrauen. Dies betrifft zudem Frauen und andere unterrepräsentierte Gruppen, die seltener Berufe ergreifen, in denen statistisches Wissen vermittelt wird.

Um Daten spannend zu machen, braucht es Darstellungsformen, die Spaß machen. Unsere Explorable Explanations machen Statistiken haptisch, visuell und spielerisch erfahrbar. Diese neuen Darstellungsformen stehen dann zur Übernahme für andere Themen zur Verfügung.

Die geplanten wichtigsten Ziele waren eine Reihe von Artikeln im Web, die in einer Kombination mit interaktiven Datenvisualisierungen und spielerischen Bestandteilen Grundbegriffe der Statistik erklären und erfahrbar machen. Die Artikel und interaktiven Grafiken sollten für Beispiele und Visualisierungen offene Daten verwenden, insbesondere Daten der Statistischen Ämter. Dafür verwenden wir die Datenschnittstelle von Datenguide (<https://datengui.de> – 2017 gefördert durch den Prototype Fund unter dem Arbeitstitel “Datenland”).

Zweiter Meilenstein war darüber hinaus die Bereitstellung der visuellen und/oder interaktiven Komponenten zur Übernahme für andere Projekte. Ein Werkzeugkasten aus open-source-Komponenten für die Entwicklung von interaktiven Erklärstücken wie im ersten Meilenstein beschrieben. Damit soll die weitere Entwicklung solcher Anwendungen vereinfacht werden. Der Werkzeugkasten baut auf bestehenden open-source-Lösungen auf, insbesondere auf React, D3 und MDX (Markdown).

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?

Das Projekt “Statistik erklärt” (gefördert unter dem Arbeitstitel “Statistik erfahrbar”) trägt zu den Zielen zweier Schwerpunkte der Förderinitiative bei: “Data Literacy” sowie “Civic Tech”. Im Sinne der “Data Literacy” wird die Einstiegshürde für Nutzer:innen ohne Vorkenntnisse deutlich verringert. Dies kann Berührungsängste vor amtlichen Statistiken nehmen und die Hemmschwelle, diese zu benutzen, senken. Unterstützt wird das davon, dass der Arbeitsaufwand für eigene Datenrecherchen und -aufbereitung für die Nutzer:innen erheblich gesenkt wird. Über ‘Statistik erklärt’ finden Nutzer:innen, die nach Fachtermini wie ‘NUTS-Level’ oder ‘Amtlicher Gemeindeschlüssel’ gesucht haben, zum Projekt Datenguide und werden direkt zum ebenfalls in diesem ‘Software Sprint’ veröffentlichten Prototypen des Datenguide Datenportals geleitet. Beide Projekt ermöglichen für den Bereich “Civic Tech” Bürger:innen einen deutlich vereinfachten Zugang zu Daten, die zwar grundsätzlich der Öffentlichkeit zur Verfügung stehen aber aufgrund der Art ihrer Veröffentlichung kaum von Bürger:innen verwendet werden können.

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?

Für Datenguide sind die Ergebnisse dieser Prototype-Fund-Runde Meilensteine:

- Es gibt nun ein einfaches Tool, das zeigt, welche Daten für eine bestimmte Region verfügbar sind und mit dem auch Nutzer:innen ohne Vorkenntnisse auf die Daten der amtlichen Statistik zugreifen können.
- Das Datenguide-Datenportal steht nun als Prototyp zur Verfügung. Darüber lassen sich die Daten aus regionalstatistik.de abrufen und in einem Format herunterladen, das sich einfach in anderen Visualisierungstools wie zum Beispiel Datawrapper weiterverarbeiten lässt.
- Die Schritte von der Datensuche über Tabellenformate bis zur ersten fertigen Visualisierung werden Nutzer*innen in einem Blogpost erklärt.

The screenshot shows the Datenguide web application. On the left, there is a sidebar with a search bar and a list of German states (Bundesländer) from Saarland to Bayern. The main content area displays search results for 'Berlin'. It shows the selected region 'Berlin' and the specific statistic 'BEV001 - Lebend Geborene'. Below this, there is a table with the following data:

ID_Region	Region	Jahr	Merkmal	Wert	Statistik
11	Berlin	1995	Lebend Geborene	28648	Statistik der Geburten
11	Berlin	1996	Lebend Geborene	29905	Statistik der Geburten
11	Berlin	1997	Lebend Geborene	30369	Statistik der Geburten
11	Berlin	1998	Lebend Geborene	29612	Statistik der Geburten

Datenguide ermöglicht es Nutzer:innen ohne Vorkenntnisse, amtliche Statistiken herunterzuladen.

- In einem weiteren Tool können Nutzer:innen den Namen einer Region eingeben, um den entsprechenden amtlichen Gemeindeschlüssel samt Erklärungen zurück zu bekommen.

Gemeinde auswählen oder suchen
08425125 - Unterwachingen

08 4 25 125

Unterwachingen · Gemeinde 08425125

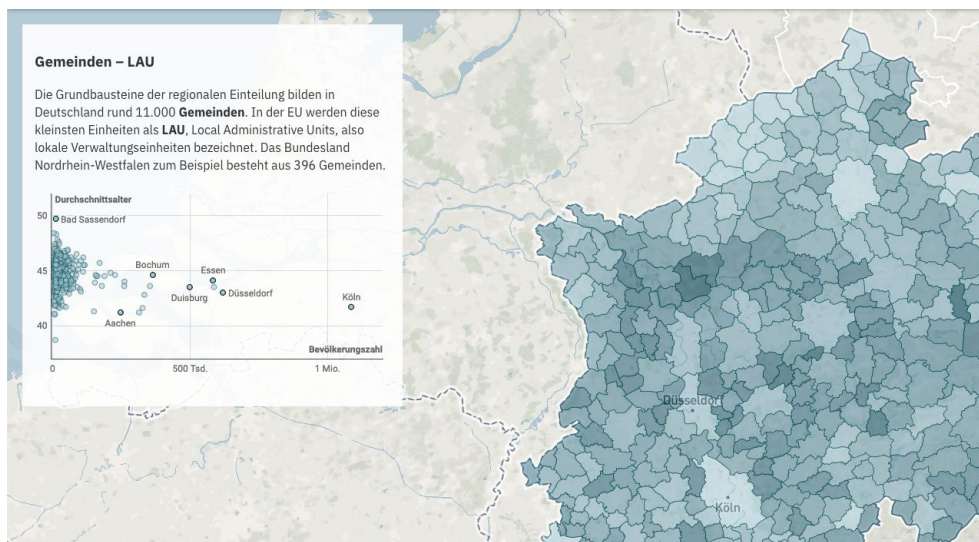
Alb-Donau-Kreis, Landkreis · Kreis 08425 NUTS 3

Tübingen, Regierungsbezirk · Regierungsbezirk 084 NUTS 2

Baden-Württemberg, Land · Bundesland 08 NUTS 1

Interaktive Erklärungen zum Amtlichen Gemeindeschlüssel (AGS)

- Ein Blogpost, erklärt interaktiv anhand von Karten, wie die europäische NUTS-Klassifikation für Deutschland funktioniert.



Interaktive Erklärungen zum NUTS-System

Alle diese Bausteine bieten Einstiegsmöglichkeiten in das Arbeiten mit amtlichen Statistiken und wir erhoffen uns, dass dadurch mehr Nutzer:innen zu Datenguide finden – auch über Suchmaschinen.

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?

Zur Zielgruppe: Bei Datenguide arbeiten wir an einer digitalen Plattform, die frei nutzbare Daten der statistischen Ämter durchsuchbar macht, klar und prägnant darstellt und es euch ermöglicht, diese Statistiken für eure eigene Analysen und Projekte weiterzuverwenden. Im Lauf der Zeit hat sich dabei der Fokus verschoben, hin zu einer Plattform für Expert:innen (wie Data Scientists und Datenjournalist:innen), die über Datenschnittstellen (APIs) auf Statistiken zugreifen. Mit dem neuen Projekt „Statistik erklärt“ laden wir auch Menschen ohne Vorkenntnisse einladen, amtliche Statistiken zu nutzen. Wir stellen dafür digitale Tools bereit, die helfen, regionale Statistiken zu verstehen und sinnvoll einzusetzen.

Ein weiteres Ergebnis des Projekts ist eine Sammlung von technischen Komponenten, mit denen interaktive Erklärstücke entwickelt werden können. Für die Umsetzung verwendeten wir JavaScript und React, sowie verschiedene spezialisierte Bibliotheken wie MapboxGL für Kartendarstellungen, MDX für das Einbetten von interaktiven Komponenten in Markdown-Dokumente und Scrollama für Scrollytelling-Interaktion. Sie können aber auch von allen weiterverwendet werden, der ähnliche Erklärstücke umsetzen möchten. Wir hoffen damit der deutschen Datenjournalisten-Community Impulse zu geben, sodass in Zukunft mehr ‘Explorable Explanations’ umgesetzt werden – gerne auch mit Daten aus der Regionalstatistik.

Im Rahmen der Projektarbeiten wurden wichtige fachliche Kenntnisse erlernt. So besteht nun bei den Projektteilnehmern ein umfassendes Wissen über amtliche Statistiken in Deutschland und deren Grundlagen, wie dem Amtlichen Gemeindeschlüssel oder der NUTS-Klassifikation. Zudem wurde musterhaft ein Prozess beschrieben, wie Nutzer:innen Daten per Datenguide abrufen und schnellstmöglich visualisieren können. Im Zuge der Programmierung der Explorable Explanations wurden neue Technologien erlernt und angewendet: Zum Beispiel Mapbox GL für die interaktive Karten in Kombination mit Scrollama für die Interaktivität.

Ebenfalls neue Erkenntnisse erlangten die Teilnehmer durch die wahrgenommenen Coachings im Bereich User Experience Design, die über die Open Knowledge Foundation angeboten wurden. Diese fachliche Weiterbildung kann nun auch im Arbeitsalltag angewendet werden.

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?

Zunächst war das Datenguide-Team sehr beeindruckt und inspiriert von den spielerischen Umsetzungen und technischen Möglichkeiten der „Explorable Explanations“. Das Start-Coaching durch den Prototype Fund unterstützte uns bei der Fokussierung auf die Nutzer:innen. Es ging darum herauszufinden, welches Wissen die Nutzer:innen brauchen, um amtliche Statistiken über Datenguide sinnvoll abrufen und nutzen zu können. Wir legten uns deshalb auf folgende Themen fest:

1. Wie funktioniert NUTS, die europäische Klassifizierung statistischer Regionen?
2. Wie ist der amtliche Gemeindeschlüssel aufgebaut und was sagt er uns über die Hierarchie der Regionen?
3. Welche Datensätze gibt es in der Regionaldatenbank und für welche Regionen sind sie verfügbar?
4. Wie bereitet man Daten aus der Regionalstatistik so auf, dass man sie als Grafiken darstellen kann?

So sinnvoll und nötig für die Arbeit mit Regionalstatistik diese Themen sind, ein wenig ernüchtert hat uns die inhaltliche Recherche auch. Denn zum Beispiel die Aufteilungen in Länder und Gemeinden wird zwar auf europäischer Ebene maximal verklausuliert NUTS genannt und ist ohne Frage nötiges Grundwissen, um sortiert und schnell auf die Daten zugreifen zu können – bei genauerem Hinblick ist sie aber nicht übermäßig komplex. Dazu kam die Erkenntnis, dass es wirklich schwer ist, komplexe Interaktionen, die auf verschiedenen Ebenen (Text, Karte, Highlighting) miteinander verzahnt sein sollen, im Kopf oder auf Papier vor auszuplanen.

Wichtig war uns immer wieder die Rückbesinnung auf die Nutzer:innen. Ursprünglich geplant waren Long-Read-Artikel mit interaktiven Erklärungen. Wir merkten aber schnell, dass es zum Beispiel beim Amtlichen Gemeindeschlüssel (AGS) gar nicht so viel im Text zu sagen gab, sondern dass der Fokus eher auf einem personalisierten Tool liegen muss, in dem Nutzer:innen für ihre Gemeinde den AGS herausfinden können und die Erklärung dadurch mitgeliefert bekommen.

Für Oktober war die Verwendung der Komponenten durch das Datenteam des Westdeutschen Rundfunks und der Redaktion von 'Die Story' geplant. Dies wurde jedoch von Seiten der Redaktion nicht umgesetzt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Webseite: <https://datengui.de/>

Statistik erklärt: <https://datengui.de/statistik-erklart>

Datenportal: <https://datengui.de/statistiken>

Github:

- Datenguide-Website, inklusive der interaktiven Erklärstücke:
<https://github.com/datenguide/datenguide>
- Unsere Sammlung von Komponenten für interaktive Erklärstücke:
<https://github.com/datenguide/explorables>

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?

Ursprünglich geplant war die Umsetzung in gemeinsamen Sprints vor Ort in Berlin und Köln. Bedingt durch die Corona-Pandemie konnten diese Sprints nur via Videokonferenz stattfinden. Dies machte teilweise das gemeinsame Arbeiten schwerer und führte dazu, dass der Kostenplan zwar insgesamt eingehalten werden konnte, sich aber die Verteilung des Arbeitsaufwandes innerhalb des Teams deutlich verschob. Ursprünglich war die Drittelung des Aufwandes geplant, zuletzt verschob sich die Umsetzung deutlich in Richtung von Simon Jockers, unter anderem weil sich die technische Umsetzung der Scrollytelling-Karte mit MapboxGL für Kartendarstellungen und Scrollama für Scrollytelling-Interaktion deutlich mehr Entwicklungszeit benötigte, als wir geplant hatten.

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?

Während der Projektlaufzeit gab es ein Update der Genesis-Online-Datenbank des Statistischen Bundesamtes. Dort lassen sich nun auch so genannte 'Flat'-Tabellen herunterladen. Dies entspricht einer Empfehlung, die das Projekt 'Datenguide' bei Gesprächen im statistischen Bundesamt geäußert hatte. Auch das Portal regionalstatistik.de soll zeitnah um diese Funktion ergänzt werden.

Allerdings bieten auch die 'Flat'-Tabellen noch ausreichend Hürden, die das nahtlose Weiterverarbeiten der Daten erschweren, so dass das Datenguide Datenportal für die Nutzer trotzdem einen Mehrwert und verringerten Zeitaufwand bietet. Dies konnten wir auch in einem Blog-Beitrag aufgreifen.

Wir verfolgen die Weiterentwicklung der offiziellen APIs sehr interessiert und greifen diese auch gerne auf. Datenguide versteht sich nicht als Konkurrenzprodukt, sondern als Brücke und Initiative, um die Themen API, Datenformate, Zugänglichkeit voranzutreiben.

Schlussbericht

**Richtlinie zum „Software-Sprint“
DDD: Deutsche Dokumente Digitalisieren**

Johannes Filter

30. September 2020

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S18 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

PDF zu Fließtext mit maschinellem Lernen

PDFs sind für Menschen gemacht und nicht für Maschinen (Computer). Das führt dazu, dass wir sie lesen können, aber Maschinen Probleme damit haben, Text zu extrahieren. Dies ist jedoch notwendig, um zum Beispiel große Mengen von PDFs im Rahmen journalistischer Recherchen auszuwerten. Auch Personen mit Seheinschränkungen sind darauf angewiesen, dass Computer ihnen Texte vorlesen. Im Rahmen der bereits erfolgten oder geplanten Digitalisierung deutscher Behörden müssen zudem große Aktenbestände digitalisiert werden.

Text extrahieren mit *pd3f*

Im Rahmen der Prototype-Fund-Förderung von „DDD: Deutsche Dokumente Digitalisieren“ ist die Software-Lösung *pd3f* entstanden, um „guten“ Text aus PDF zu rekonstruieren - „gut“ bedeutet hier, dass der ursprüngliche Text ohne unnötige Zeilenumbrüche wiederhergestellt werden kann. Aus dem zerstückelten Text im PDF wird somit wieder ein digitaler Fließtext. Im Deutschen gibt es viele lange Wörter und deswegen die Besonderheit, dass Wörter am Zeilenende getrennt werden. Bei einer üblichen Textextraktion werden getrennte Wörter jedoch nicht wieder zusammengefügt. Damit kann das ursprüngliche Wort zum Beispiel nicht mehr per Suche gefunden werden. Auch weiterführende Anwendungen, wie zum Beispiel die automatisierte Erkennung von Eigennamen (Named-Entity Recognition), werden erschwert.

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

Sämtliche erstellte Software steht unter einer Open-Source-Lizenz. Zudem wurden Kernfunktionalitäten ausgelagert, sodass neue Projekte einfach auf diese Software aufbauen können. Damit wird sichergestellt, dass diese Arbeit der Open-Source-Community zugutekommt. Inhaltlich ist durch diese Förderung die Anwendung von maschinellen Lernen auf deutschen Dokumenten bzw. Texten gefördert worden. Da viel Software nur auf die englische Sprache abzielt, wurde mit der Förderung insbesondere die deutsche Open-Source-Community gestärkt.

Ausführliche Darstellung der Ergebnisse

Automatisierte Texterkennung

Texterkennung auf gescannten Dokumenten (Optical Character Recognition/OCR) erfolgt schon heute zufriedenstellend mit Open-Source-Lösungen. Es ist aber aufgrund des veralteten Portable Document Format (PDF) weiterhin schwierig, die Wörter aus einem PDF zu einem Text zusammenfassen. Das Format folgt der Idee des Druckens, weshalb Fließtext darin nicht als Text dargestellt wird. Stattdessen wird teilweise jeder einzelne Buchstabe als Zeichen kodiert und ihm eine Position (x- und y-Wert für Höhe und Breite) auf dem (virtuellen) Blatt Papier zugewiesen.



Teilweise werden in PDF nur einzelne Buchstaben zusammen gespeichert. Wörter müssen erst aus einzelnen Buchstaben zusammengesetzt werden.

Um aus diesem Buchstabensalat nutzen bestehen Tools Heuristiken, um Buchstaben zu Wörter zusammen zu setzen. Aus Wörtern müssen wieder ganze Zeilen und diese Zeilen anschließend zu Paragraphen zusammengefasst werden. Das ist ohnehin eine schwierige Aufgabe, da es für den Satz eines Textes nahezu endlose Möglichkeiten gibt. Das Open-Source-Tool *Parsr*¹ des französischen Versicherungskonzerns Axa sorgt bereits für Besserung, denn es zerlegt relativ erfolgreich ein PDF in Zeilen und Absätze. Das Tool ist wenige Monate vor Start der Projektförderung erschienen und erwies sich in der Projektphase als nützlich. Unsere Software *pd3f* füttert zunächst das PDF in Parsr und nutzt die Ausgabe von Parsr, um darauf aufbauend guten Text wiederherzustellen.

Zeilenumbrüche entfernen

Worttrennung an Zeilenumbrüchen zu entfernen, ist eigentlich eine einfache Aufgabe: Alle Wörter mit einem „-“ am Zeilenende werden mit dem Wort auf der darauffolgenden Zeile zusammengefügt wie in diesem Beispiel.

... die Bedeutung der finan-
ziellen Interessen der Union ...

¹<https://github.com/axa-group/Parsr>

Das Wort „finanziellen“ entspricht dem ursprünglichen Text. Es gibt aber auch „-“ am Zeilenenden, das nicht entfernt werden darf, weil es Bestandteil des Wortes ist:

Auch andere EU-
Staaten, wie bspw. Polen, ...

Um an dieser Stelle weiterzukommen, braucht es mehr Verständnis über die deutsche Sprache. Hier kommt maschinelles Lernen in Form von Sprachmodellen zum Einsatz.

Was sind Sprachmodelle? (Language Models)

Bei Sprachmodellen geht es darum, dass ein Computerprogramm neue Wörter auf der Basis bereits genutzter Wörter lernt. Zum Einsatz kommen Sprachmodelle zum Beispiel auf Smartphones bei der Autovervollständigung. Sprachmodelle verinnerlichen die Charakteristiken der deutschen Sprache und können vorhersagen, welche Wörter oder Buchstaben wahrscheinlich als nächstes kommen. So kommt nach „Sehr geehrte“ wahrscheinlich „Frau“ als nächstes.

Solche Sprachmodelle operieren auf ganzen Wörtern oder auch nur auf Buchstaben (um nachfolgende Buchstaben zu erraten).

Texte reparieren mit *dehyphen*

Eine Unterkomponente von *pd3f* ist das Softwarepaket *dehyphen*, welches ebenfalls im Rahmen der Förderung entstand. Es benutzt Sprachmodelle, um zu entscheiden, ob ein „-“ am Zeilenende entfernt werden sollte oder nicht. Die Grundidee ist dabei eine Berechnung darüber, welche die wahrscheinliche Möglichkeit ist, zwei Zeilen zu verbinden.

Auch andere EU-
Staaten, wie bspw. Polen, ...

Bei diesem Beispiel kommt *dehyphen* zum richtigen Ergebnis: „EU-Staaten“ ist korrekt, nicht „EUStaaten“. *dehyphen* kann als Modul von anderen Software-Entwickler*innen einfach wiederverwendet werden.

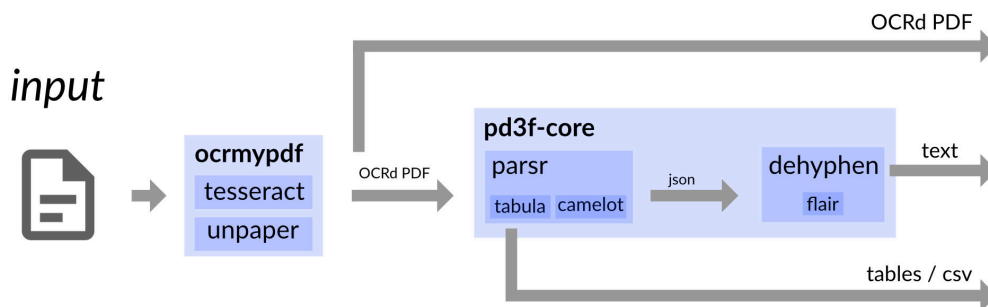
Datenverarbeitungs-Pipeline *pd3f*

Das Hauptergebnis der Förderung ist *pd3f*: Eine komplette Anwendung und eine Datenverarbeitungs-Pipeline für PDFs. Mit ihrer Hilfe können (deutsche) Dokumente digitalisiert werden. Auf einem gescannten Dokument wird der Text automatisiert gescannt, dann mithilfe von Parsern der Text in Wörter, Zeilen und Absätze unterteilt. Anschließend wird mithilfe von *dehyphen* der gute Text extrahiert. Anbei eine schematische Auflistung der benutzten Komponenten.

Die Demo von *pd3f*: <https://demo.pd3f.com>.

Der Fokus liegt auf der deutschen Sprache, deren besonderes Charakteristikum lange Wörter sind, *pd3f* kann aber auch für andere Sprachen angewandt werden. Es ist aktuell auch für Englisch, Spanisch und Französisch verfügbar.

pd3f – PDF Text Extractor output



Dieses Projekt baut auf einer großen Zahl von anderen Open-Source-Projekten auf.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Zielgruppe sind Civic-Tech-Entwickler*innen in Deutschland und andere Personen, die aus Dokumente Texte extrahieren möchte, zum Beispiel im Rahmen journalistischer Recherchen.

Da Dokumente in so vielen unterschiedlichen Formen vorkommen, funktioniert pd3f noch nicht für alle PDFs. Gerade bei schlecht gescannten PDFs ist der extrahierte Text zudem noch stark verbesserungsbedürftig. Es ist unwahrscheinlich, dass pd3f jemals für alle PDFs funktionieren wird, doch es wird weiterhin gearbeitet, die Resultate, zum Beispiel für mehrspaltige Dokumente, zu verbessern.

Was noch fehlt, ist eine systematische Evaluation der Textextrakte von *pd3f*. Diese wird voraussichtlich im April 2021 erfolgen.

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

Ja, denn diese Art von experimenteller Produktentwicklung mittels maschinellem Lernen war auch für mich neu. Ich konnte in diesem Projekt eine Vielzahl meiner Fähigkeiten einbringen. So war es zum Beispiel zum Ende des Projekts wichtig die Ergebnisse auf einer eigens gebauten Webseite zu bewerben. Meine anderen Fähigkeiten rund um das maschinelle Lernen und der Python-Programmierung konnte ich ebenso einbringen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Zunächst wurde mehr auf die Software *Parsr* aufgebaut. Es stellte sich jedoch heraus, dass viele Features von *Parsr* nicht zufriedenstellend. So gibt *Parsr* an, dass die Software auch Überschriften in einem Dokument erkennen kann. Die Genauigkeit dieser Identifikation zeigte sich jedoch als zu gering, um sie für dieses Projekt einsetzen zu können.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzerinnen

Auf der Webseite <https://pd3f.com> gibt es sowohl in Deutsch als auch Englisch eine Einführung in das Projekt. Auf der Webseite ist ebenso ein Video verlinkt, in dem innerhalb eines 20-minütigen Vortrags die Software präsentiert wird. Zudem gibt es auf <https://demo.pd3f.com> eine Demo der Software.

Der Code ist online auf GitHub² und wird dort weiter gepflegt. Die Hauptfunktionalitäten von *pd3f* sind zusätzlich in einem eigenen Python-Paket³ gebündelt, sodass auch hier eine einfach weitere Verwendung als Bestandteil in einem anderen bzw. neuen Python-Projekt möglich ist.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Generell erfolgte die Arbeit dem Plan. Zum Ende des Projekts gab es noch eine Vielzahl von weiteren Möglichkeiten für zukünftige Arbeit. Durch eine Priorisierung von Arbeitspaketen blieben mehrere Pakete übrig (wie zum Beispiel eine systematische Evaluation der Ergebnisse).

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Einige Monate vor Förderbeginn hat der französische Versicherungskonzern Axa eine Open-Source-Bibliothek names *Parsr*⁴ zur Dokumentverarbeitung veröffentlicht. Diese Veröffentlichung änderte den Verlauf der Entwicklung, denn die Software erwies sich als nützlich und so baut *pd3f* in Teilen auf *Parsr* auf.

²<https://github.com/pd3f/pd3f>

³<https://github.com/pd3f/pd3f-core>

⁴<https://github.com/axa-group/Parsr>

Richtlinie zum „Software-Sprint“

Boromeo – Erkenntnisse aus Dokumentensammlungen sichtbar machen

Schlussbericht

Zuwendungsempfänger:

Simon Wörpel

*Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS19S19** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.*

Stand: 29. September 2020

Kurze Darstellung der Aufgabenstellung und Motivation

Die Diskussion um “Vertrauen in die Medien” oder “alternative Fakten”, um nur zwei Schlagworte zu nennen, ist gerade in den vergangenen Monaten der weltweiten Corona-Pandemie wieder erneut entflammt. Doch schon länger sind Medien, und damit ihre Rechercheur*innen, einem wachsenden Vertrauensanspruch seitens der Konsument*innen ausgesetzt. Erkenntnisse sollen und müssen transparent und nachvollziehbar sein. Nur so entsteht Glaubwürdigkeit. Eine Möglichkeit, um diesem geforderten Anspruch gerecht zu werden, kann darin bestehen, originale Quellen, auf die sich Fakten & Aussagen stützen, ebenfalls zu veröffentlichen. Sofern es der Quellenschutz zulässt.

Damit das überhaupt möglich ist, müssen Recherche-Teams, egal ob in Medien oder anderen Organisationen, auch intern einige Herausforderungen meistern: Wie generieren wir Erkenntnisse aus großen Dokument-Sammlungen? Wie überführen wir diese in strukturierte Fakten? Wie stellen wir sicher, dass wir unsere Erkenntnisse bzw. ihre Quellen später wiederfinden – zum Beispiel für eine neue Recherche? Wie können wir kollaborativ aber möglichst sicher zusammenarbeiten? Wie können wir unsere eigenen, auf Dokumenten basierenden Recherchen, effizient fakt-checken?

Das Vorhaben „Boromeo“ sollte diesen Überlegungen die nötigen technischen Lösungen bieten. Geplant war eine Plattform, die es Teams ermöglicht, an Dokumenten zu arbeiten und die daraus gewonnenen Erkenntnisse kollaborativ zu teilen sowie zu veröffentlichen. “Boromeo” sollte sich dabei auf einen möglichst einfachen Recherche-Workflow, die Verschneidung verschiedener Datenquellen und ein modulares und offenes System konzentrieren, um mit anderen, bestehenden Lösungen Daten & Erkenntnisse auszutauschen und zusammen zu arbeiten.

Im Verlauf des Vorhabens wurde deutlich, dass diese Ziele zu allgemein und zu umfangreich gesteckt waren, und es wurde sich auf einen konkreten Anwendungsfall konzentriert: Dem Annotieren von PDF-Dokumenten. Diese Prototyp-Lösung bekam den Namen „Annotate & Chill“ und wird im weiteren Verlauf dieses Berichts so benannt. Dennoch wurden weitere Aspekte des Vorhabens bearbeitet, worauf später noch eingegangen wird.

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

Die Zielgruppen des Vorhabens sind vielfältig. Primäre Zielgruppe sind Journalist*innen, die mit großen Dokumenten-Sammlungen, sogenannten Leaks, arbeiten. Das können interne Unterlagen, Ermittlungsakten aus einem Verfahren oder eine Sammlung von E-Mails sein. Diese Zielgruppe ergab sich aus dem bisherigen Betätigungsfeld des Projektteilnehmers, dem investigativen Journalismus. Jedoch sind alle Personengruppen, die in ihrer professionellen Arbeit mit solchen Dokumenten zu tun

haben, Teil der größeren Zielgruppe, da die technische Lösung für verschiedenste Arten von Dokumenten funktioniert.

Die nun in einem Prototypen veröffentlichte Lösung ermöglicht in erster Linie das Annotieren von PDF-Dokumenten sowie das (vom Dokument unabhängige, s.u.) Teilen dieser Daten mit anderen Mitarbeiter*innen. Dadurch können Recherche-Gruppen von der oben skizzierten Arbeitsweise profitieren, was ihrer Arbeit zu Transparenz und Vertrauen verhilft. Gleichzeitig ist das System auf die hohen Sicherheitsanforderungen solcher Teams ausgerichtet. Genau diese Ansprüche finden sich auch im Themenschwerpunkt „Engineering trust“ des Vorhabens wider.

Ausführliche Darstellung der Ergebnisse

Wie bereits berichtet wurde das Vorhaben auf einen konkreten Anwendungsfall, das Annotieren von Quell-Dokumenten, konkretisiert. Diese Überlegung wurde erst durch das Coaching, das im Rahmen der Begleitung durch die Open Knowledge Foundation ermöglicht wurde, deutlich und der Prozess hiervon begleitet. Es wurde zunächst analysiert, welche Lösungen es bereits gibt und daraus eine für das Vorhaben entwickelt, dass drei Voraussetzungen des oben skizzierten Nutzungsszenarios erfüllt, die bisher von bestehenden Lösungen noch nicht angeboten wurden: Erstens, Annotierung kann lokal und offline auf dem Arbeitsgerät des Mitarbeitenden stattfinden ohne in einer Cloud gespeichert zu werden. Zweitens, Annotierungen sind getrennt von den Quell-Dokumenten und können unabhängig davon verbreitet und geteilt werden. Und drittens gibt es kein festgelegtes Format für Annotationen, es kann je nach Anwendungsfall und verwendeter Dritt-Software flexibel gestaltet werden.

Dies wird durch „Annotate & Chill“ dadurch gelöst, dass pro annotiertem Dokument Dateien im CSV-Format generiert werden, die unabhängig vom Quell-Dokument über sichere Kommunikationswege (beispielsweise verschlüsselte E-Mails) geteilt werden kann. Besitzen Mitarbeitende eines Recherche-Teams das selbe Dokument, müssen sie nur die Datei mit den Annotationen untereinander teilen und können diese auf ihrem Gerät mit dem Quell-Dokument verbinden. Das erfüllt die gestellten Sicherheitsanforderungen solcher Teams.

Die Anwendung funktioniert offline, also lokal auf dem Rechner. Es werden PDF-Dokumente importiert, die dann mit Annotationen versehen werden. Diese können über das Interface exportiert, auf einem anderen Gerät importiert und weiter bearbeitet werden. Die eindeutige Identifikation von Dokumenten funktioniert über deren Prüfsumme. Aus jeder Datei lässt sich diese Zeichenkette erstellen, die als eindeutige ID funktioniert. So können Annotationen einer Datei immer zugeordnet werden, egal ob das Dokument sich in einen Fall auf dem Laptop einer Person oder in einem anderen Fall in einer Online-Datenbank befindet.

Die Definition, auf welche Stelle sich die Annotation innerhalb eines Dokuments bezieht, wird als ein Rechteck auf der betreffenden Seite angegeben, das die Rechercheur*innen mittels des Interfaces mit der Maus zeichnen. So ist sichergestellt, dass sowohl PDF-Dokumente, die ein sog. „Text-Layer“ (also

extrahierten, maschinenlesbaren Text) enthalten, annotiert werden können, als auch Dokumente ohne Text (eingescannte PDFs) oder Bilder. Auf die Möglichkeit, Text-Passagen zu markieren und darauf basierend Annotationen zu erstellen, wurde aus Gründen der Einheitlichkeit verzichtet, allerdings besteht im Server-Backend (s.u.) diese technische Möglichkeit.

Das Interface, sozusagen der „Client“ von „Annotate & Chill“, läuft derzeit im lokalen Browser des Nutzenden. Es ist jedoch geplant, diese Lösung als eigenständige Applikation für Windows, Mac und Linux anzubieten, damit sie einfacher zu nutzen ist. Auch wenn vordergründig damit der dezentrale, lokale (und somit sehr sichere) Anwendungsfall abgedeckt ist, wurde darüber hinaus auch eine Server-Anwendung entwickelt, die kollaboratives Arbeiten in der Cloud ermöglicht. Diese kann von Teams selbst betrieben werden, um an die jeweiligen Sicherheitsansprüche eines Projekts angepasst zu werden (z.B. Erreichbarkeit nur über einen VPN-Zugang). Mittels dieser Plattform werden Annotationen mit eindeutigen URLs versehen und können der Allgemeinheit zugänglich gemacht werden. Eine Datenbank hält vor, an welchen öffentlichen Orten im Netz die Quell-Dokumente verfügbar sind und verlinkt dorthin. Dieser Server-Teil von „Annotate & Chill“ soll demnächst als ergänzendes Angebot zum Prototypen veröffentlicht werden.

Somit konnte das hauptsächliche Anliegen, eine Lösung zu entwickeln um Erkenntnisse aus großen Datensammlungen strukturiert & kollaborativ zu bearbeiten, umgesetzt werden, wenn auch in etwas abgewandelter Form als zunächst in den Meilensteinen des Vorhabens skizziert. So wurde wie bereits erwähnt durch das angebotene Coaching das ursprüngliche Produkt, eine größere, allgemeine Rechercheplattform, in kleinere Teil-Lösungen unterteilt. Dies war zum Einen durch den Umfang dieses Vorhabens, der den Projekt-Rahmen deutlich gesprengt hätte, bedingt, und zum Anderen vor allem durch sinnhafte Überlegungen geprägt:

So gibt es bereits sehr gut funktionierende Komplett-Lösungen für Dokumenten-Recherchen, wie zum Beispiel „Datashare“ vom International Consortium of Investigative Journalists (ICIJ) und insbesondere „Aleph“ vom Organized Crime and Corruption Project (OCCRP). Beides sind Open-Source-Lösungen, die auch von anderen Newsrooms und Organisationen eingesetzt und mitentwickelt werden. Mit dem „follow the money“-Werkzeugkasten als Teil des Aleph-Systems ist es möglich, die unstrukturierte, reale Welt in maschineller Form so zu modellieren, sodass sie auch ein Computer verstehen kann.

Das hier beschriebene Vorhaben wurde dementsprechend konkretisiert, um den bestehenden Lösungen bisher nicht berücksichtigte Features (Annotation, getrennt vom Dokument) hinzuzufügen. „Annotate & Chill“ könnte in die bestehenden Plattformen integriert werden (s.u.) und ist glücklicherweise keine weitere „All-in-One“-Lösung eines in sich geschlossenen Systems geworden.

Als weiterer Meilenstein wurde eine Materialsammlung von Schulungsmaterial skizziert. Auch wenn durch die Grundsatz-Recherche während des Vorhabens hierzu bereits sehr viel Dokumentation gesammelt und aufgebaut werden konnte, reichte die Zeit nicht aus, dieses auch in gewinnbringender Form für andere Rechercheur*innen zu veröffentlichen. Dennoch wurde deutlich, dass die

Informations- bzw. Schulungslandschaft in diesem Bereich immer noch sehr wenig anbietet, vor allem für Einsteiger, sodass es ein großes (auch persönliches) Anliegen des Projektteilnehmers ist, diese Inhalte noch im Nachgang des Förderzeitraums auf einer Plattform zu veröffentlichen.

Insgesamt hat insbesondere die Coaching-Begleitung der Open Knowledge Foundation dazu beigetragen, das „übergroße“ Plattform-Vorhaben in kleinere, machbare Teil-Lösungen herunterzubrechen, von der die wichtigste, „Annotate & Chill“, bereits als Prototyp fertiggestellt und veröffentlicht wurde. Weitere Teilbausteine sind bereits in Arbeit oder sollen zeitnah angegangen und veröffentlicht werden, worauf weiter unten eingegangen wird.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die oben skizzierten Zielgruppen können mit „Annotate & Chill“ nun auf einfache und unkomplizierte Weise Dokumente annotieren. Dies kann, wie beschrieben, kollaborativ und sicher geschehen und benötigt keine Server-Installation der Software. Die einfache Trennung von Quell-Dokument und Annotationen war ohne Cloud-Lösung auf diese Weise vorher nicht möglich.

Auch wenn „Annotate & Chill“ eine eigene Lösung ist, kann der technische Unterbau dank der Open-Source-Veröffentlichung in andere Plattformen integriert werden. Hierbei ist jede Plattform denkbar, die einen „PDF-Reader“ integriert hat, wie zum Beispiel „Aleph“ (<http://alephdata.org/>) oder Projekte der Open Knowledge Foundation wie „FragDenStaat“ (<https://fragdenstaat.de/>) oder „SehrGutachten“ (<https://sehrgutachten.de/>). Dazu müssten nur die technischen Komponenten, die die Dokumente auf der Website anzeigen, durch die zugrunde liegende Lösung von „Annotate & Chill“ ausgetauscht werden, die dann das Annotieren dieser Dokumente ermöglichen würde.

Außerdem soll, wie bereits erwähnt, die Client-Software als „Applikation“ ausgeliefert werden, die einfach auf den Arbeitsgeräten der Rechercheur*innen installiert werden kann. Hierzu soll in Kürze die Client-Anwendung in das „Electron“-Framework (<https://www.electronjs.org/>) integriert werden, was eine einfache technische Lösung für die Bereitstellungen von browserbasierten Webanwendungen als installierbare „System-Programme“ für Windows, Mac und Linux bietet.

Im Rahmen der Arbeit ist ein weiteres mögliches Projekt entstanden, das auch in Teilen dem ursprünglichen Vorhaben-Gedanken von „Boromeo“ entspricht: Eine Online-Plattform mit aufbereiteten und kuratieren Datensätzen, die einfach auffindbar sind und von Rechercheur*innen heruntergeladen und in ihre Recherche-Systeme integriert werden können. An dieser Lösung wurde als Teil des Projekt-Vorhabens exemplarisch bereits mit dem Daten-Team des „Organized Crime and Corruption Projects“ (OCCRP) gearbeitet. Diese Plattform soll noch in diesem Jahr 2020 veröffentlicht werden.

Darüber hinaus wurde im Projektverlauf durch die intensive Arbeit an grundsätzlichen Themen zu Dokumenten-Verarbeitung und Verwaltung und das Studium der verschiedenen bestehenden

technischen Plattform-Lösungen ein Prototyp entwickelt, der das stillgelegte Projekt „Kleine Anfragen“ (<https://kleineanfragen.de/>) durch eine neue Lösung weiterführen könnte. Hierzu werden bereits Gespräche mit den entsprechenden Stellen geführt, um aus den gewonnenen Erkenntnissen dieser Projektlaufzeit die KleineAnfragen-Plattform neu zu entwickeln und weiterleben zu lassen.

Insgesamt hat nicht nur die technische Entwicklungsarbeit am Prototypen „Annotate & Chill“, an der neue Technologien ausprobiert werden konnten, den Projektteilnehmer fachlich weitergebildet, sondern vor allem die damit einhergehende Grundsatz-Recherche zu bestehenden (und immer noch fehlenden) Lösungen im weiten Feld der Dokumenten-Recherche. Der Projektteilnehmer konnte sich dadurch im Vorhabenverlauf intensiv persönlich und fachlich in den vielfältigen Themenbereichen fortbilden und hat nun ein deutlich größeren Wissens- und Erfahrungsschatz, der in der weiteren beruflichen Laufbahn sehr förderlich werden kann.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

In der ursprünglichen Vorhabens-Planung war angedacht, die Annotationen auf der bestehenden Lösung “hypothes.is” (<https://web.hypothes.is/>) aufbauend zu realisieren. Dieses Projekt stellt den Source-Code ebenfalls Open-Source zur Verfügung. Allerdings stellte sich bei genauerer Recherche heraus, dass diese technische Lösung nicht zu den identifizierten Nutzungs-Szenarien passt. So werden die Annotationen auf einem Server gespeichert und es wäre technisch sehr kompliziert gewesen, die Möglichkeit der offline-Bearbeitung in der Art, wie sie mit “Annotate & Chill” realisiert werden konnte, mit Hilfe des “hypothes.is”-Systems zu realisieren. Aus einer Kosten-Nutzen-Abwägung wurde im weiteren Verlauf verzichtet, diesen Weg weiter zu verfolgen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Eine einfache Demo ist unter <https://demo.annotate.cool> zu erreichen. Hier können PDF-Dokumente annotiert werden und diese Annotationen im CSV-Format exportiert (und wieder importiert) werden. Auch wenn die Anwendung auf einem Server läuft, funktioniert sie technisch gesehen nur im Browser (Client) der Nutzer*innen, die PDF-Dokumente sowie die Annotationen werden niemals irgendwohin hochgeladen sondern nur lokal gespeichert.

Der Open-Source-Code dieser Anwendung ist unter <https://github.com/simonwoerpel/annotate-and-chill/> erreichbar. Außerdem wurde im Rahmen der „Demo Week“ dieser Förder-Runde eine Website veröffentlicht, auf der das Vorhaben in einem Blog-Post konkret beschrieben und erklärt wurde: <https://demoweek.prototypefund.de/projects/05-annotate-and-chill.html>

Darüber hinaus soll in der nächsten Zeit unter dem ursprünglichen Projektnamen „Boromeo“ auf <https://boromeo.org> eine Präsentations-Seite der verschiedenen bearbeiteten Lösungen entstehen, die als Einstiegs-Portal dienen soll. Hier können sich User*innen über die technischen Angebote für das Recherchieren in Dokumenten informieren und die für sie passende Lösung, also unter anderem „Annotate & Chill“ samt Nutzerdokumentation, finden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Zu Beginn des Projektes konnten nicht die geplanten Mittel abgerufen werden, da aufgrund der Corona-Pandemie und damit verbunder Mehraufwand in anderer freiberuflicher Arbeit des Projektteilnehmers weniger Arbeitszeit für die Fördermaßnahme aufgewendet werden konnte als zunächst geplant. Das wurde im weiteren Verlauf des Förderzeitraums wieder ausgeglichen. Deshalb wurde die zu Beginn veranschlagte Kostenplanung weitestgehend eingehalten.

Außerdem wurde der Arbeitsschwerpunkt auf die Entwicklung der technischen Lösung „Annotate & Chill“ gelegt und weniger auf die Plattform für Schulungsmaterial als ursprünglich geplant. Es wurde aber im Förderzeitraum viel dies betreffendes Material gesammelt und generiert, das noch im weiteren Verlauf der Arbeit an diesem Projekt veröffentlicht werden soll (s.o.)

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Eine verwandte Software-Lösung ist „Aleph“ vom Organized Crime and Corruption Project (OCCRP). Der Projektteilnehmer war bereits vor dem Förderzeitraum mit diesem Angebot vertraut. Ursprünglich arbeitete er an einer alternativen Lösung zu Aleph, die ebenfalls alle Anwendungsfälle vom Einlesen verschiedener Dateiformate, Analysieren der vorhandenen und generierten Daten und das kollaborative Recherchieren darin abbilden sollte. Es gab konkrete Feature-Ideen, die sich von Aleph und dem dort vorhandenen Angebot unterschieden hätten. Jedoch wurde die Software Aleph insbesondere in der ersten Jahreshälfte 2020 von dem dortigen betreuenden Team deutlich weiterentwickelt und um wichtige Features erweitert, sodass der Projektteilnehmer sich entschloss, die Arbeit an einer alternativen, „All-in-One“-Lösung einzustellen und sich stattdessen auf eine zugespitzte, modulare Lösung zu konzentrieren, aus der „Annotate & Chill“ entstanden ist.

Berlin, 30. September 2020



Simon Wörpel

Richtlinie zum „Software-Sprint“

NCDASH – Nextcloud Community-Dashboard

Schlussbericht

Zuwendungsempfänger:

Meurer Wiehle GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS189S20 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Soziale Bewegungen und politische Aktivist:innen organisieren ihre Arbeit zunehmend online. Dazu benötigen sie Kommunikations- und Kollaborationstools. Existierende Tools sind meist kommerziell, werten die sensiblen Daten der Nutzer:innen aus oder verkaufen sie gar weiter. Vorhandene, frei verfügbare und datenschutzfreundliche Tools präsentieren sich entweder als Datenspeicher (wie ein Aktenregal) oder als flüchtiger Echtzeit-Chat (wie ein Cafégespräch). Wenn wir die beiden Aspekte kombinieren, wird der gemeinschaftliche Charakter erfahrbar und ein gemeinsam erarbeitetes Resultat entsteht. So kann im Prozess Vertrauen wachsen. Mit einer neuen Community Dashboard App für Nextcloud wollten wir einen modernen Ansatz schaffen.

Um das Problem anzugehen, waren folgende Meilensteine vorgesehen:

1. Simple Dashboard mit Übersicht über Dokumente, Texte, Kalender und Umfragen in Form dynamischer Kacheln
2. Grundlegende Wiki-Funktion (Markdown-Syntax) für Start- und Unterseiten der App
3. Gruppenübergreifende Teilbarkeit der Dashboard- und Wiki-Inhalte
4. Markdown-Erweiterungen für Integration und Referenzierung von Inhalten anderer Apps
5. Ausführliche Nutzer*innen- und Entwickler*innen-Dokumentation
6. (optional) Unterstützung für Föderation: Teilen von Seiten/Themenbereichen über Nextcloud-Instanzen hinweg

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

Mit dem Projekt wurden verschiedene Ziele des „Software-Sprint“ verbessert. Mit der App wurde ein Ort geschaffen, um kollektives Wissen dauerhaft in Nextcloud zu speichern und zu organisieren. Wir hoffen, die primäre Zielgruppe, soziale Bewegungen und kollaborativ arbeitende Teams damit in ihren kollektiven Arbeitsprozessen zu unterstützen. Dies halten wir für einen wichtigen Beitrag zum Schwerpunkt „Civic Tech“. Auch die Upstream-Verbesserungen an Nextcloud – insbesondere an der Text und Circles App – kommen vielen zivilgesellschaftlichen Initiativen, NGOs und Projekten zugute, die Nextcloud für die Organisation ihrer Arbeit verwenden.

Ausführliche Darstellung der Ergebnisse

Wir haben erfolgreich den Prototypen einer App entwickelt und damit das Potential unserer Idee untermauert. Konkret haben wir in den letzten sechs Monaten eine Nextcloud App mit grundlegenden Wiki-Funktionalitäten geschaffen, welche die Nextcloud Text App als Editor verwendet. Da der Fokus der App auf dem Arbeiten in Gruppen liegt, wurden Gruppenverwaltungs-Funktionen fest in die Grundfunktionen der App eingewoben – auf Basis von Nextcloud Kreisen.

Da den Nextcloud Text und Circles Apps einige benötigte Funktionen fehlten, haben wir diese in enger Zusammenarbeit mit den Upstream-Entwickler:innen entwickelt und als Beiträge eingereicht.

Nicht alle ursprünglich anvisierten Meilensteine konnten erreicht werden. Schon zu Beginn des Projektes war klar, dass wir uns zunächst auf einen der zwei Schwerpunkte *Wiki-Funktionen* oder *Dashboard* fokussieren mussten. Da die Upstream-Nextcloud-Entwickler:innen bereits Pläne für ein persönliches Dashboard hatten und an diesen aktiv arbeiteten, lag es nahe, uns auf die *Wiki-Funktion* zu konzentrieren und das Thema Dashboard zu einem späteren Zeitpunkt anzugehen. Die Meilensteine 1 und 4 sind daher im Rahmen der Förderung nicht angegangen worden.

Ein zentrales Erkenntnis für die Förderung ist daher auch, dass der durch den Prototypenfund gegebene Handlungsspielraum, seine Ziele im Rahmen der Förderung anpassen und aktualisieren zu können, sehr wichtig ist. Wären wir an die zu Beginn genannten Meilensteine gebunden gewesen, so hätten wir sicher aus der Not heraus an einem Projekt gearbeitet, das am Ende weniger erfolgreich und zielführend gewesen wäre.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zunächst stellt das Open Source Konzept die Grundlage für unser Projekt. Wäre das gesamte Nextcloud Ökosystem nicht Open Source, so könnten wir auf dessen Grundlage nicht unsere App entwickeln. Auch die Möglichkeit, Verbesserungen an den Upstream Nextcloud Komponenten einzureichen existiert nur auf Grundlage des frei verfügbaren Quellcodes.

Unsere Zielgruppe – vor allem soziale Bewegungen und Community Projekte – dürften mit der App einen Ort zur Organisation ihres Wissens erhalten haben. Wir planen, die App in den kommenden Wochen im Nextcloud App Store zur Verfügung zu stellen und somit sämtlichen Nextcloud-Administrator:innen zugänglich zu machen. Außerdem planen wir koordinierte Nutzer:innen-Tests mit einzelnen Projektgruppen im Laufe der kommenden Monate. Auch haben wir vor, die App als Open Source Projekt weiter zu entwickeln.

Wir selbst haben im Rahmen des Projektes viel über die Entwicklung von Prototypen in komplexeren Ökosystemen gelernt, aber auch schlicht unsere Fähigkeiten beim Entwickeln moderner Web-Applikationen weiterentwickeln können.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Im Großen und Ganzen sind wir sehr positiv angetan davon, wie zielstrebig unsere Arbeit war und wie wenige Irrwege wir bei der Entwicklung eingeschlagen haben. Das hatten wir definitiv anders erwartet.

Bei der Entwicklung einer Lösung für das konkrete technische Problem von „kollektivem Eigentum“ in Nextcloud hatten wir zunächst ein vorübergehendes Konzept mit Hilfe eines so genannten „Hilfs-Accounts“ entwickelt – konkret sollten die geteilten Inhalte eines Wikis diesem (versteckten) Hilfs-Accounts gehören um nicht an die Accounts von konkreten Mitgliedern der Arbeitsgruppe geknüpft und damit von deren fortwährender Anwesenheit abhängig zu sein. Dieses Hilfs-Konzept haben wir nach einigen Tests wieder verworfen und stattdessen damit begonnen, grundlegendere und nachhaltigere Lösungen für das Problem zu entwickeln.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Unser Projekt inklusive Sourcecode, Problemtracking und Anleitungen findet sich unter <https://gitlab.com/collectivecloud/collectives/>. Online-Dokumentation findet sich zusätzlich unter <https://collectivecloud.gitlab.io/collectives/>.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wir haben uns stark auf die Implementierung von Wiki-Funktionen fokussiert, die Dashboard-Funktionalität konnten wir im Rahmen der Förderung nicht umsetzen. Letztlich lässt sich sagen, dass die Arbeit an einem Teilbereich eines größeren Ökosystems (Nextcloud) immer mehr Koordinierungsaufwand erfordert als erwartet. Auch die notwendigen Vorarbeiten an den Upstream-Projekten (Text und Circles App) haben mehr Zeit in Anspruch genommen, als wir zu Beginn veranschlagt hatten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab keine nennenswerten Einflüsse durch andere Institutionen, die den Verlauf des Projekts maßgeblich beeinflusst haben.

Richtlinie zum „Software-Sprint“

Ncube – Suche und Analyse dokumentierter Menschenrechtsverletzungen

Schlussbericht

Zuwendungsempfänger:

Abdul Hadi Habal

Christo Buschek

Giovanni Civardi

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01|S19S21 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)?

Daten sind mittlerweile ein integraler Bestandteil moderner Gesellschaften geworden. Datenbasierte Untersuchungen haben an Bedeutung gewonnen – auch in der täglichen Arbeit von Menschenrechtsaktivist*innen und Forscher*innen.

Forscher*innen sind mit einer Unzahl von Werkzeugen konfrontiert und müssen manchmal tiefgreifende technische Fähigkeiten entwickeln, um diese Werkzeuge zu benutzen. Wir haben uns zum Ziel gesetzt, ein Werkzeug zur Datenuntersuchung zu entwickeln, das es Forscher*innen ermöglicht, Daten zu sammeln, zu recherchieren und zu verifizieren.

Wir hatten bereits vorherige Erfahrungen mit dem Entwickeln von Dateninstrumenten für die Menschenrechtsforschung und sind aktiv an der Durchführung solcher Forschungen beteiligt. Daher hatten wir ein ziemlich klares Verständnis der Merkmale, die wir brauchten und auf wen wir abzielten. Wir definierten die Meilensteine so, dass wir schon sehr früh eine funktionierende Version

hatten, die wir ausprobieren konnten. Im Laufe des Projekts hatten wir mehrere Versionen (<https://github.com/critocrito/ncube/releases>), die es uns erlaubten, das Produkt regelmäßig zu testen.

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?

Wir richten uns an Menschenrechtsforscher*innen und investigative Journalist*innen, die inspirierende Narrative veröffentlichen wollen, die Machtmissbrauch und Menschenrechtsverletzungen aufdecken. Wir legen großen Wert auf die Verifizierung von Daten, um vertrauenswürdige und sachliche Gegenpole in einer Post-Truth Welt zu schaffen. Forscher*innen müssen ihren Daten vertrauen können. Ncube wendet daher ein hohes Maß an forensischer Validität für die Datensammlung an und ermöglicht die strukturelle und systematische Anwendung von Verifikationsmethoden um den Wahrheitsgehalt der Daten zu bestimmen.

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?

Das Ziel von Ncube ist es Menschenrechtsaktivist*innen und anderen Forscher*innen zu ermöglichen, Daten zu sammeln, zu recherchieren und zu verifizieren. Wir haben eine Desktopanwendung entwickelt, die derzeit unter Linux und macOS läuft. Wir hatten auch eine Windowsversion geplant, schafften es aber nicht innerhalb des Zeitrahmens diese fertigzustellen. Wir hoffen dies in den kommenden Wochen nachzuholen. Wir haben eine eigenständige Desktopanwendung entwickelt, die derzeit unter Linux und MacOS läuft. Wir planen auch eine Windows-Version, schafften es aber nicht innerhalb des Zeitrahmens. Die Desktopanwendung richtet sich an Forscher*innen, die Daten in ihren täglichen Arbeitsabläufen verwenden wollen. Bei der Anwendung handelt es sich um eine local-first Software (<https://www.inkandswitch.com/local-first.html>). Dieser Ansatz verbessert die Sicherheit, den Datenschutz, die langfristige Bewahrung und die Benutzerkontrolle von Daten. Ncube hat zwei Teile, eine UI und ein Backend. Die UI ist in TypeScript geschrieben, und das Backend verwendet Rust als Implementierungssprache. Für die Desktopanwendung werden diese beiden Teile in eine einzelne Binärdatei gepackt und kann so einfach installiert werden.

Für die Server-Installation gibt es ein separates Build, den Ncube Daemon. Dieser Build enthält nur das Backend von Ncube und wird mit einem Verwaltungsprogramm für die Terminal-Nutzung geliefert. Forscher*innen können den Ncube Daemon auf ihrem Server einrichten, um entweder die Datensammlung über die Limitierungen eines Desktop-PCs hinaus zu skalieren oder die Kollaboration mit anderen Forscher*innen zu ermöglichen. Die Verbindung zu dem Server wird innerhalb der Desktopversion von Ncube hergestellt. Auf diese Weise ist der Wechsel zwischen lokalen und externen Arbeitsbereichen für die Forscher*innen fließend.

Die Betreiber*innen eines Servers kann einzelnen Forscher*innen Zugriff gewähren. Dieses Dokument (<https://github.com/critocrito/ncube/blob/master/doc/auth-workflow.pdf>) beschreibt die details des Authentifizierungs- und Autorisierungsmechanismus.

Pakete für die verschiedenen Plattformen sind hier verfügbar:
<https://github.com/critocrito/ncube/releases/tag/0.9.3>.

Um den gesamten Arbeitsablauf von Forscher*innen bei der Recherche zu verbessern, haben wir auch eine Erweiterung für den Firefox-Browser entwickelt. Dies vereinfacht browserbasierte Forschungs- und Discoveryworkflows. Diese Erweiterung ist eine Ergänzung zu Ncube. Sie ermöglicht es Forscher*innen, Links an Ncube zur Weiterverarbeitung zu senden. Wir haben die Erweiterung auf [addons.mozilla.org](https://addons.mozilla.org/en-US/firefox/addon/ncube-discovery) veröffentlicht: <https://addons.mozilla.org/en-US/firefox/addon/ncube-discovery>.

Es ist uns zwar nicht gelungen, alle Funktionen zu entwickeln, die wir uns vorgenommen hatten (z.B. Windows support), aber wir haben alle geplanten Meilensteine erreicht. Einige dieser Funktionen wurden absichtlich fallen gelassen, da wir das Gefühl hatten, dass wir nicht genug Feedback hatten, um sie sinnvoll zu planen. Wenn wir Werkzeuge entwickeln wollen, die von Forscher*innen benutzt werden, müssen wir als Teil der Projektplanung genügend Zeit einplanen, damit sie diese Werkzeuge ausprobieren können. Durch den Umfang unseres Projektes und den äusseren Umständen ist uns das leider nicht immer gelungen.

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?

Wir planen, die Arbeit an diesem Projekt fortzusetzen. Vorläufige Rückmeldungen zeigen, dass es ein enormes Interesse und Bedarf an solchen Werkzeugen gibt. Wir sind mit Individuen und Organisationen im Bereich der Menschenrechtsforschung und -dokumentation und des investigativen Journalismus im Gespräch. Von Anfang an waren sie unser Zielpublikum, und wir werden uns weiterhin auf diese Zielgruppe konzentrieren. Persönlich war es großartig, Mittel für diese Technologieentwicklung zu erhalten. Es ermöglichte uns, neue Technologien zu lernen und anzuwenden. Wir sind überzeugt, dass diese neuen Fähigkeiten in Zukunft für unsere Arbeit auf dem Gebiet der datengestützten Untersuchungen anwendbar sein werden.

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?

Von Anfang an hatten wir eine ziemlich klare Vorstellung davon, was wir erreichen wollten. Bevor wir überhaupt angefangen haben Code zu schreiben, haben wir uns viel Zeit genommen, um die Ziele und Funktionen von Ncube zu planen. Dadurch hatten wir einen guten und detaillierten Plan, den wir umsetzen konnten. Dadurch konnten wir vermeiden uns in eine Sackgasse zu verlaufen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Der Code für dieses Projekt ist auf Github unter <https://github.com/critocrito> verfügbar. Wir haben eine Website eingerichtet, die als Dach für verschiedene datenbezogene Werkzeuge fungiert (<https://sugarcubetools.net/>).

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?

Es gab kein einzelnes Ereignis, das einen spezifischen Einfluss hatte. Wir bauen ein Werkzeug für eine Gemeinschaft von der wir auch ein Teil sind. Deshalb führen wir kontinuierliche Gespräche mit anderen Mitglieder*innen in unserem Bereich. Viele dieser Gespräche bildeten die Grundlage für unseren Wunsch, Ncube zu entwickeln.

Richtlinie zum „Software-Sprint“

CoopCare – Omaha-System basierte, kooperative Software für Pflegekräfte in selbstorganisierten Teams

Schlussbericht

Zuwendungsempfänger:

Michael Kamphausen

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S22 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Pflegekräfte erleben extremen Leistungsdruck, die der Gesundheit von Gepflegten und Pflegenden schadet, sowie zu kurzen Verweildauern im Beruf und zu Personalmangel führt.

Mit CoopCare möchte ich Pflegekräfte bei der Pflegedokumentation und -planung entlasten, indem Pflegesituationen kurz und präzise erfasst werden, Fortschritte und Rückschritte auf einen Blick sichtbar werden und die Übergaben und der Austausch mit Kolleg*innen vereinfacht werden. Darüber hinaus möchte ich Pflgeteams systematisch darin unterstützen, sich in kooperativer Weise selbst zu organisieren und die Selbständigkeit und Gesundheit ihrer Klienten bestmöglich zu fördern.

Um diese Ziele zu erreichen, habe ich die plattformübergreifende, webbasierte Open-Source-App CoopCare für digitale Pflegedokumentation, Planung und Evaluation entwickelt. Es ist die erste deutschsprachige und quelloffene Implementierung des Omaha-Systems. Das Omaha-System ist ein gemeinfreies Klassifikationssystem für Pflegedokumentation, Planung und Evaluation. Mit der Verwendung des Omaha-Systems greife ich den Wunsch von Pflegekräften auf, weil dieses Klassifikationssystem mit Auswahllisten und Textbausteinen hilft, Pflegesituationen kurz, präzise und gleichzeitig individuell zu beschreiben. Bei der Evaluation der Pflege legt das Omaha-System einen starken Fokus auf die Selbständigkeit der Patient*innen entsprechend ihrer Möglichkeiten, und quantifiziert die Bewertung des Status quo und der Pflege-Ziele auf einer Skala von 1 bis 5, was die Kommunikation über den Krankheitsverlauf und die Überprüfung der Wirksamkeit der Pflege sehr erleichtert.

Im Rahmen der Förderung plante ich die Benutzeroberfläche und die Funktionalität zu bauen und zu verbessern, so dass alle drei Komponenten des Omaha-Systems Problemklassifikation, Pflegeplanung, Ergebnis-Evaluation von Pflegekräften erfasst und ausgewertet werden können. Das Feedback von Pflegekräften sollte entscheidenden Einfluss auf die Ausgestaltung bekommen. Da das Omaha-System in Deutschland wenig bekannt ist, hatte ich auch eine didaktische Einführung in das Omaha-System vorgesehen. Weiterhin plante ich die Erfassung von Patient*innen-Stammdaten, Berichtsfunktionen wie Pflegebericht und Durchführungsprotokoll, den Datenaustausch zwischen zuständigen Pflegekräften unter Berücksichtigung rechtlicher Vorgaben, eine Schicht-übergreifende Erinnerungsfunktion an geplante Aufgaben, sowie den sicheren Betrieb des Servers zu ermöglichen.

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

Die Zielgruppe von CoopCare sind Pflegefachpersonen, die durch den Einsatz dieser App bei der Pflegedokumentation, Planung und Evaluation entlastet werden. Sie profitieren von reduziertem Dokumentationsaufwand bei gleichzeitig besserer Auswertbarkeit, so dass mehr Zeit für die Beziehungen zu den Patient*innen bleibt. Durch die Unterstützung der Pflegeteams in Selbstorganisation, Kommunikation und ganzheitlicher Arbeitsweise, sollen gleichermaßen Mitarbeiterzufriedenheit, Verweildauer im Beruf und Qualität der Pflege wachsen. Von der Förderung der Selbständigkeit der Pflegebedürftigen profitieren diese selbst sowohl gesundheitlich, als auch in ihrer Autonomie und verbesserter gesellschaftlicher Teilhabe.

Dies entspricht der Zielsetzung des „Software-Sprints“ sowohl in Bezug auf die Ermächtigung von Bürger*innen, in diesem Fall Pflegenden und Gepflegten, im Sinne von „Civic Tech“, als auch zur Stärkung des Vertrauens zwischen den Akteuren Pflegefachkräften, Pflegebedürftigen und Pflegekassen. Gleichzeitig geht es bei dieser Open-Source-Lösung darum, den hohen Anforderungen an Datensicherheit und Datenschutz für Gesundheitsdaten gerecht zu werden.

Ausführliche Darstellung der Ergebnisse

Ich habe die Benutzeroberfläche und die Funktionalität gebaut und kontinuierlich verbessert, um die Erfassung und Auswertung der drei Komponenten des Omaha-Systems Problemklassifikation, Pflegeplanung und Ergebnis-Evaluation durch Pflegefachpersonen zu ermöglichen. Eine Zusammenfassung der erfassten Daten und des Krankheitsverlaufs werden im Pflegebericht übersichtlich dargestellt. Darüber hinaus werden alle Änderungen und Löschungen dokumentiert und in der Historie des Änderungsprotokolls sichtbar gemacht. Es wurde ein Farbschema und ein Farbleitsystem entworfen und umgesetzt. Zur Verbesserung der Einfachheit und Verständlichkeit der Benutzeroberfläche wurde diese mehrmals iterativ überarbeitet.

CoopCare wurde mit einem responsiven Design für unterschiedliche Bildschirmformate vom Smartphone bis hin zum Desktop-PC entwickelt und hat eine mehrsprachige Benutzeroberfläche in deutsch und englisch. Darüber hinaus wurde die Software darauf ausgelegt, den Nutzer*innen zur Unterstützung bei der Erfassung der Pflegedokumentation und -planung Vorschläge für häufige Probleme und häufige Pflegemaßnahmen passend zum jeweiligen Krankheitsbild oder Problem anbieten zu können.

Tests und Feedback zur Software durch Pflegefachkräfte waren angesichts der Pandemie nur sehr eingeschränkt möglich, da diese Zeit für meine Ansprechpersonen in der Pflege turbulent war. Ich habe wertvolle Rückmeldungen bekommen, allerdings seltener als ich gebraucht hätte, so dass ich Annahmen treffen musste, die ich erst nach dem Förderzeitraum mit weiterem Feedback überprüfen werde.

Weiterhin ist eine Aufgaben-Verwaltung entstanden, die die Daten der Pflegeplanung in einer Kalender-ähnlichen ToDo-Liste mit Erinnerungsfunktion darstellt. Sowohl einmalige, als auch regelmäßig wiederkehrende Aufgaben können darin angelegt, bearbeitet, erledigt, verschoben oder beendet werden. Diese Daten sind wiederum die Grundlage des Durchführungsprotokolls, das ebenfalls als Ergebnis der Förderung entstanden ist.

Aus zeitlichen Gründen konnten nicht alle geplanten Aufgaben umgesetzt werden, deshalb fehlen eine umfassende didaktische Einführung in das Omaha-System, ebenso wie die Funktion zur Erfassung von Patient*innen-Stammdaten. Leider konnte das Projekt aus Zeitnot auch nicht zur Teamfähigkeit ausgebaut werden, daher fehlen die vorgesehenen Funktionen zum gemeinsamen Bearbeiten und Austausch über die Daten zwischen Pflegefachpersonen eines Teams mit rollenbasierten Zugriffsrechten. Für ein Projekt, das kollaborative Arbeitsweisen im Fokus hat, ist das ein Problem, deshalb möchte ich im Anschluss an den Förderzeitraum dieses Manko als erstes beheben. Allerdings konnte ich während der Förderung bereits das technische Konzept hierfür ausarbeiten, das zur Nutzung im Team auch Offline-Fähigkeit, Synchronisation, das automatische Lösen von Merge-Konflikten, Undo-Funktion und Ende-zu-Ende-Verschlüsselung berücksichtigt.

Anstelle der nicht umgesetzten Aufgaben ist eine Projektwebseite und ein knapp fünfminütiges Video zur Erläuterung, Veranschaulichung und Präsentation des Projektes für die Zielgruppe der Pflegefachpersonen entstanden. Das Video vermittelt dabei auch eine erste Einführung in das Omaha-System. Weiterhin ist eine Online-Demo der Software entstanden, die es Interessierten ermöglicht, die Benutzeroberfläche und Funktionalität der Software niedrigschwellig und nahezu vollständig zu testen, denn sie erfordert keinen Login und enthält Beispiel-Patient*innendaten, aber persistiert und sendet keinerlei Daten, denn alle Bearbeitungen erfolgen nur lokal und flüchtig im Browser des Testenden.

Außerdem habe ich eine englischsprachige Contributing-Richtlinie und ein Contributor License Agreement für potenzielle Beitragende zu diesem Open-Source-Projekt formuliert.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die primäre Zielgruppe von CoopCare sind Pflegefachpersonen. Die Anwendungsmöglichkeiten des Omaha-Systems sind allerdings vielseitig genug, dass es sich für viele Bereiche von Care-Arbeit eignet, wie zum Beispiel Betreuung, Geburtshilfe oder Sozialarbeit.

Alle Nutzer*innen von CoopCare und ihre Klient*innen profitieren vom reduzierten Aufwand für Dokumentation, Pflegeplanung und Evaluation. Weil Pflegesituationen mit CoopCare kurz, präzise und gleichzeitig individuell beschrieben werden, bleibt mehr Zeit für die Beziehung zu Klient*innen und ihren Bedürfnissen. Die Evaluation ermöglicht es, Fortschritte und Rückschläge im Krankheitsverlauf auf einen Blick zu erkennen. Dabei wird der Fokus auf die Förderung der Selbständigkeit und Gesundheit der Klient*innen ebenso gestärkt, wie die Überprüfung der Wirksamkeit von Pflegemaßnahmen um sie den Bedürfnissen der Klient*innen anzupassen. Sobald

die Teamfähigkeit implementiert sein wird, werden auch die Übergaben und der Austausch mit Kolleg*innen über Pflegesituationen einfacher. Da die meisten Daten aus dem Klassifikationsschema des Omaha-Systems ausgewählt werden, sind sie semantisch gut strukturiert. Dies ermöglicht Auswertungen, wie beispielsweise welche Pflegemaßnahmen häufig mit der Verbesserung eines Problems einhergingen. So kann ein Pflegeteam voneinander aus der dokumentierten Erfahrung der anderen Teammitglieder lernen. Damit wird das Ziel erreicht, Pflegedokumentation mehr zu dem zu machen, was sie immer sein sollte: eine nutzenstiftende Aufgabe, die die Effektivität des Teams erhöht, anstatt lediglich rechtliche Anforderungen zu befriedigen.

Ich werde die Arbeit an CoopCare fortsetzen. Dabei stehen zunächst vier Bereiche im Fokus:

- Die Implementierung der Mehrbenutzerfähigkeit an sich und die Erweiterung um Offline-Fähigkeit und Ende-zu-Ende-Verschlüsselung. Conflict-free Replicated Data Types (CRDT) sollen dabei eine konfliktfreie, parallele Bearbeitung der Daten auch offline und eine sichere, vertrauenswürdige Versions-Historie ermöglichen.
- Die Akzeptanz von CoopCare bei der Zielgruppe zu erhöhen durch eine didaktische Einführung in das Omaha-System. Mit kurzen Tutorial-Videos, Übungen und interaktiven Fallbeispielen als Bestandteil der App soll sie an Bekanntes anknüpfen und dabei helfen, den Einstieg zu erleichtern und Sicherheit im Umgang mit dem Omaha-System und CoopCare zu erreichen.
- Die Benutzeroberfläche mit Hilfe des Feedbacks von Pflegekräften weiter zu vereinfachen und zu verbessern und die im Förderzeitraum mangels Feedback getroffenen Annahmen zu überprüfen.
- Den Funktionsumfang zu erweitern: dazu gehören die Erfassung der Patient*innen-Stammdaten, eine Dienst- und Tourenplanung und die Integration der Abrechnung. Vorgesehen sind auch Auswertungen in Form von Visualisierungen der Dokumentationsdaten für das jeweilige Pflegeteam, um häufige Bedürfnisse und besonders wirksame Pflegemaßnahmen auf einen Blick sichtbar zu machen mit dem Ziel der evidenzbasierten Verbesserung der Pflege. Weitere Potenziale sehe ich in der Förderung der Quartierspflege, also der Vereinfachung der Kommunikation mit dem informellen Netzwerk aus Angehörigen und Freunden des Gepflegten, als auch mit dem formellen Netzwerk wie Ärzten, Therapeuten und Apotheken.

Nach einer geschlossenen Test-Phase von CoopCare ist geplant, die Software Pflegeteams als gehosteten Dienst mit Support anzubieten. Es wird angestrebt, für diesen Zweck eine Produktivgenossenschaft zu gründen, die den Pflegekräften einen zuverlässigen, wirtschaftlichen Betrieb, Support und Weiterentwicklung der Software garantiert. Individuelle Anpassungen an Kundenwünsche und Schulungen sind weitere Bausteine wirtschaftlicher Verwertung. Gleichzeitig steht der Programmcode als Open-Source-Software der Community zur Weiterentwicklung zur Verfügung und bietet das Potenzial, durch die Entwicklung von Schnittstellen weitere Pflege-relevante Software anzubinden.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Es gab keine Arbeiten, die aus fachlichen Gründen nicht weiter verfolgt werden konnten. Wie in den anderen Abschnitten dargestellt, wurden lediglich einzelne Aufgaben aus zeitlichen Gründen umpriorisiert und verschoben.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

Webseite & Video: <https://www.coopcare.de>

Interaktive Online-Demo: <https://demo.coopcare.de>

Video & Blogpost zur Demo-Week: <https://demowebk.prototypefund.de/projects/04-coopcare.html>

Code: <https://github.com/coop-care/web-app>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Entwicklung der Aufgaben-Verwaltung stellte sich als zeitaufwändiger heraus als geplant, insbesondere die Erinnerungsfunktion mit ihrer Unterstützung für vielfältige Wiederholungsmuster. Erfreulicherweise konnte hierbei der Aufwand dennoch in einem vertretbaren Rahmen gehalten werden durch den Einsatz der Open-Source-Bibliothek in JavaScript „rrule“ für die Berechnung der Wiederholungen entsprechend des Musters. Allerdings musste wegen des Mehraufwands die Entwicklung der didaktischen Einführung in das Omaha-System und die Erfassung von Patient*innen-Stammdaten auf einen späteren Zeitpunkt verschoben werden.

Es war ursprünglich nicht vorgesehen, ein Projekt-Video und eine Online-Demo von CoopCare zu produzieren, was eine weitere Anpassung der Planung erforderlich machte. Dies entstand aus der Pandemie-bedingten Notwendigkeit, das Projekt online zu präsentieren anstatt bei einem Demo-Day in Berlin. Um das Video und die Online-Demo rechtzeitig fertigstellen zu können, musste die Umsetzung der Mehrbenutzerfähigkeit verschoben werden. Nichtsdestotrotz handelt es sich bei beidem um nützliche Formate das Projekt anschaulich zu erklären, die zu einem späteren Zeitpunkt ohnehin entstanden wären. Außerdem eignet sich das Video als einfache Einführung in das Omaha-System, auch wenn es kein Ersatz für das ursprünglich geplante, umfassendere Konzept sein kann.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Mir sind keine Ergebnisse bei anderen Stellen bekannt geworden, die im Förderzeitraum Einfluss auf dieses Projekt und seine Zielsetzung hatten.

Richtlinie zum „Software-Sprint“

Dystonse - Wahrscheinlichkeits-basierte ÖPNV-Routensuche

Schlussbericht

Zuwendungsempfänger:

Dystonse - Lena Schimmel und Kirstin Rohwer GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S23 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Bisher bestehende Werkzeuge zur Routen-Planung im ÖPNV und Fernverkehr arbeiten primär auf Fahrplandaten und beziehen mögliche und tatsächliche Verspätungen und Ausfälle allenfalls nachrangig mit ein. Dadurch entsteht bei den Reisenden ein Gefühl der Unsicherheit - Wird der Anschlusszug erreicht? Was, wenn nicht? - und die Abwägung, ob eine vorgeschlagene Route ausreichend sicher ist, muss rein subjektiv erfolgen.

Im Projekt „Dystonse“ sollte eine Routen-Suche entwickelt werden, die durchgängig mit Wahrscheinlichkeitsverteilungen arbeitet und somit nicht nur Ergebnisse für den Best-Case liefert, sondern eine realistische Angabe für die Ankunfts-Zeiten und -Wahrscheinlichkeiten. Die inhärente Unvorhersehbarkeit des öffentlichen Nah- und Fernverkehrs sollte damit einschätz- und handhabbar werden.

Die geplante Software sollte aus der Such-Komponente (Backend) sowie einer Website und/oder App bestehen, um die Suche zu bedienen (Frontend).

Im Backend sollten Echtzeitdaten von Verkehrsmitteln gesammelt und ausgewertet werden. In die Suche sollten sowohl aktuelle Daten eingehen, als auch statistische Modelle aus vergangenen Daten.

Das Frontend sollte den Reisenden ähnliche Grundfunktionen bieten wie bestehende Nah- und Fernverkehrs- Apps, jedoch mit dem grundlegenden Unterschied, dass nicht nur feste Zeitangaben auf einer definitiven Route angezeigt werden, sondern die relevanten Alternativen und Eventualitäten.

Im Ergebnis sollte das Vertrauen und Sicherheitsgefühl der Reisenden vor und während der Reise erhöht werden. Bestehende Nutzer_innen des öffentlichen Verkehrs sollten in ihrer Nutzung bestärkt werden, und Menschen, die wegen der Unsicherheiten bisher nicht den öffentlichen Verkehr nutzen, sollten dazu animiert werden.

Um dies umzusetzen, wurden folgende Meilensteine geplant:

1. Kontinuierliche Verspätungs-Datensammlung planen, aufbauen und betreiben
2. Statistische Analyse der gesammelten Echtzeit-Verspätungs-Daten
3. Prognose-Berechnung entwickeln
4. Routensuche neu implementieren
5. Web-Frontend anpassen
6. Userbefragung, UX- und UI-Design
7. Weitere Datenquellen/Formate und Testgebiete hinzufügen, inkl. neuer Analyseschritte
8. eigene API anbieten
9. eigene App entwickeln oder Dystonse in andere Apps integrieren
10. (ggf. weitere Milestones, die sich aus User-Befragung etc. ergeben)

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

Reisende im öffentlichen Personenverkehr (ÖPV) sollen letztlich die Software für die Planung ihrer Reisen nutzen und profitieren von zuverlässigeren Verbindungen. Sie sparen damit Zeit und/oder Nerven, da weniger Unsicherheit bzgl. des Reiseverlaufs besteht.

Der ÖPV soll dadurch attraktiver werden und idealerweise seinen Anteil im Modal Split der Verkehrsträger erhöhen. Davon profitieren die Verkehrsbetriebe, und durch die verbesserte Klimabilanz praktisch alle.

Dadurch, dass die Software unter einer freien Lizenz (MIT-Lizenz) veröffentlicht wird, profitieren letztlich auch diejenigen, die ähnliche Software entwickeln und dabei auf unseren Beiträgen aufbauen können. Dies betrifft ebenso private (Hobby-)Entwicklerinnen wie auch Unternehmen.

Historisch bedingt wurden Nahverkehrsdaten oft nur angeboten, um diese unverändert bereitzustellen. Das Projekt Dystonse soll praktisch aufzeigen, wie freie Daten unter offenen Lizenzen darüber hinausgehende Anwendungen ermöglichen, und damit hoffentlich dazu beitragen, dass weitere Beteiligte ihre Daten ebenfalls freigeben.

Ausführliche Darstellung der Ergebnisse

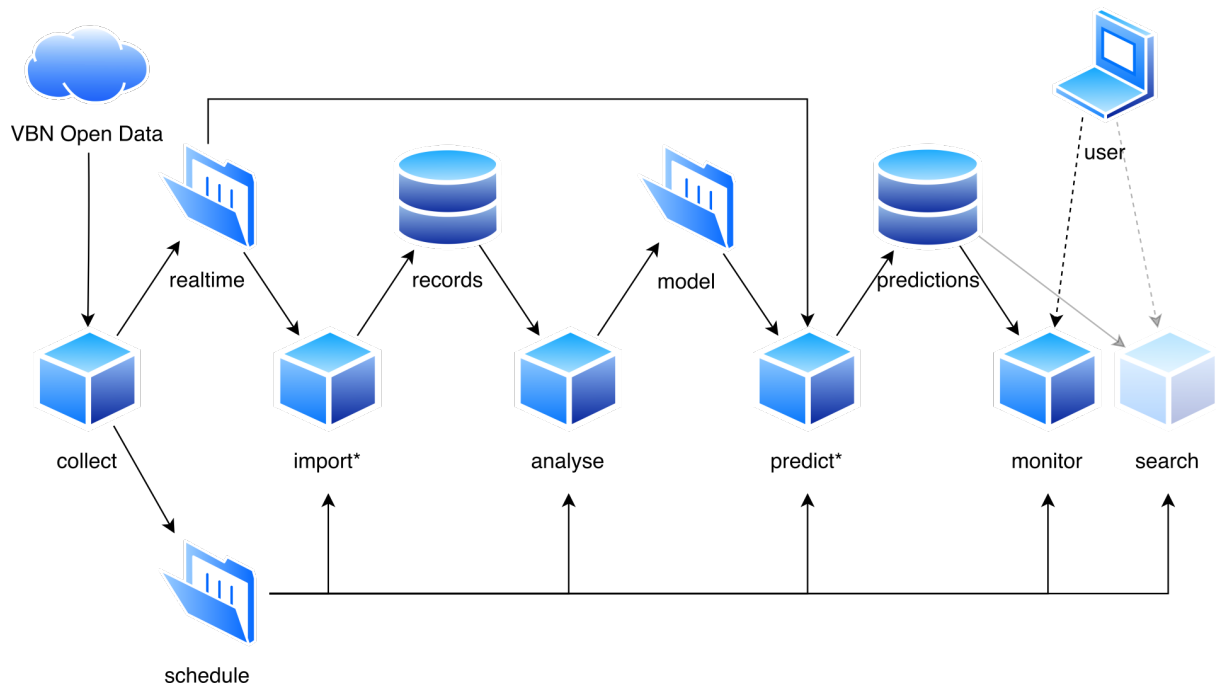
Angestrebt wurde die Entwicklung eines Backends und eines Frontends zur Routensuche. Zusammenfassend kann gesagt werden, dass das Backend erfolgreich umgesetzt wurde und seit Monaten im Dauerbetrieb funktioniert. Das Frontend wurde nicht in der ursprünglich geplanten Weise umgesetzt, sondern dessen Zweck und Funktionsweise angepasst. Im Rahmen dieser Planänderung ist auch das Frontend als erfolgreich umgesetzt anzusehen.

Die Gründe für die Neudefinition des Frontends sind im Abschnitt „Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung“ erläutert, während die Funktion des Back- und Frontends weiter unten in diesem Abschnitt beschrieben sind.

Konkret konnten wir den folgenden Anteil der Meilensteine erreichen:

1. **erreicht:** Kontinuierliche Verspätungs-Datensammlung planen, aufbauen und betreiben
2. **erreicht:** Statistische Analyse der gesammelten Echtzeit-Verspätungs-Daten
3. **erreicht:** Prognose-Berechnung entwickeln
4. **nicht erreicht:** Routensuche neu implementieren
5. **teilweise erreicht:** Web-Frontend anpassen
6. **teilweise erreicht:** Userbefragung, UX- und UI-Design
7. **nicht erreicht:** Weitere Datenquellen/Formate und Testgebiete hinzufügen, inkl. neuer Analyseschritte
8. **teilweise erreicht:** eigene API anbieten
9. **nicht erreicht:** eigene App entwickeln oder dystonse in andere Apps integrieren
10. **nicht zutreffend:** (ggf. weitere Milestones, die sich aus User-Befragung etc. ergeben)

Über eine Vielzahl von Verarbeitungsschritten generiert das Backend fortlaufend Prognosen für Fahrzeuge des ÖPV, wobei für jedes künftige Abfahrts- oder Ankunftsereignis nicht nur eine konkrete Zeit berechnet wird, sondern die Wahrscheinlichkeitsverteilung über einen ausgedehnten Zeitraum. Dabei werden diese Verteilungen nicht auf statistische Grundparameter herunter gebrochen, sondern detailliert berechnet, gespeichert und übertragen, um damit im Nachgang komplexe Berechnungen und Visualisierungen zu ermöglichen.



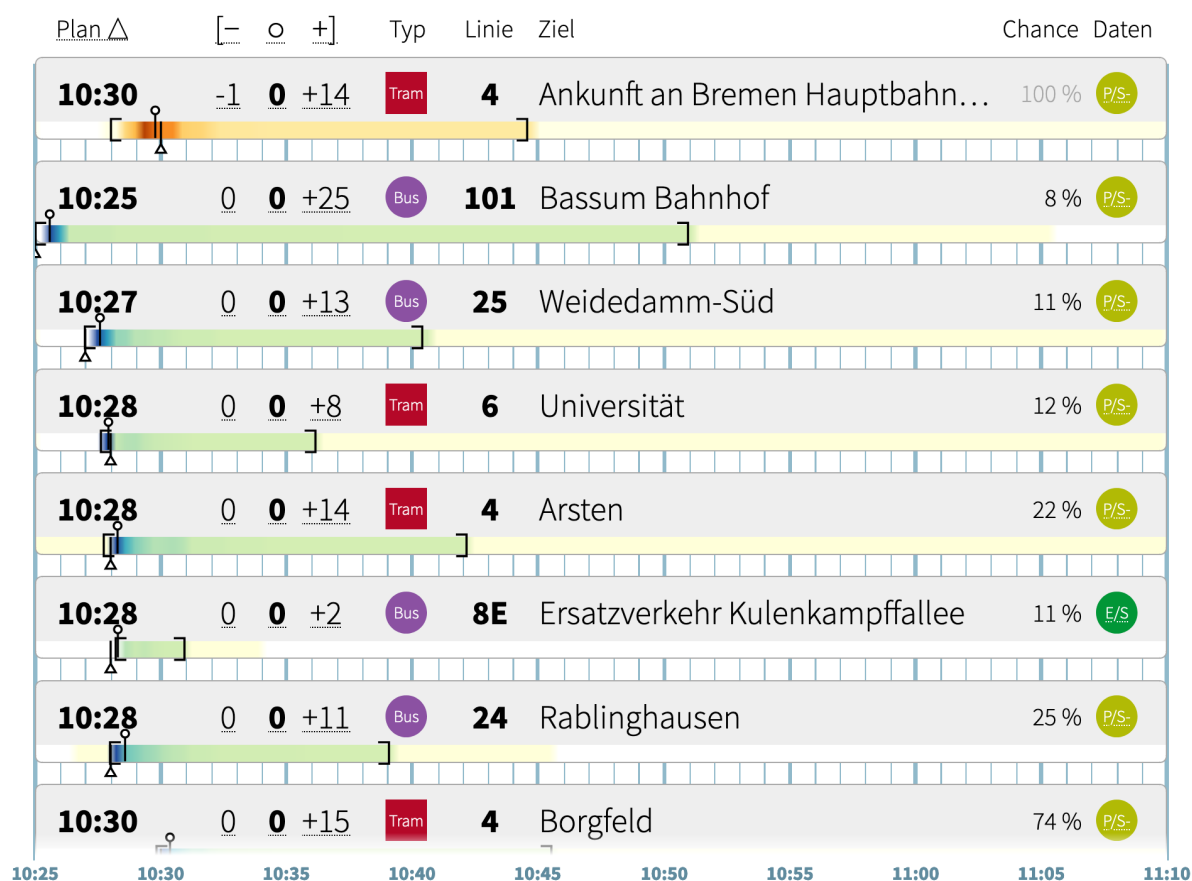
Das Backend und die davon erzeugten Daten eignen sich dabei gleichermaßen für die (noch nicht umgesetzte) Routensuche, als auch für den (ersatzweise umgesetzten) erweiterten Abfahrtsmonitor.

Statt einer vollständigen Routensuche wurde ein erweiterter Abfahrtsmonitor entwickelt, der quasi die Vorstufe einer Routensuche darstellt. Für einen beliebigen, von den Nutzer_innen gewählten Haltepunkt werden die Abfahrten der nächsten halben Stunde angezeigt. Dabei wird, entsprechend der vom Backend berechneten Wahrscheinlichkeitsverteilungen, nicht nur ein einzelner Zeitpunkt angezeigt, sondern die gesamte Verteilung. Die Darstellung umfasst absolute Zeiten laut Fahrplan, dazu relative Angaben für die früheste, wahrscheinlichste und späteste zu erwartende Abfahrtszeit. Diese vier Zeitpunkte werden durch Symbole auf einer Zeitskala dargestellt, die durch Helligkeit und Farbton außerdem die detaillierte Wahrscheinlichkeitsverteilung über den gesamten Zeitraum anzeigt.

🔍 → Bremen Sielhof → Tram 4 → Bremen Hauptbahnhof

Hilfe

Abfahrten für Bremen Hauptbahnhof (und 1 weitere), Montag, 24. August von 10:25 bis 11:10



Zusätzlich zum bloßen Betrachten der Abfahrtszeiten können Reisende mit wenigen Klicks manuell eine Reisekette zusammen stellen. Für jede Abfahrt kann der weitere Fahrtverlauf gezeigt werden, wobei alle folgende Halte ebenfalls eine Verteilung möglichen der Ankunftszeiten zeigen. Mit einem weiteren Klick kann ein Ausstieg geplant werden, wobei alle dort verfügbaren Umstiege gelistet werden. Aus den Ankunfts- und Abfahrtswahrscheinlichkeiten wird automatisch die Chance errechnet, dass der jeweilige Anschluss erreicht wird. Bei längeren Reiseketten werden diese Chancen multipliziert, um die Gesamtchance zu zeigen, mit der die Reise planmäßig vollendet werden kann.

Für mögliche Umstiege werden auch Abfahrten von nahegelegenen Haltepunkten mit einbezogen. Die nötigen Fußwege werden in die Zeitplanung mit einberechnet. Auch hier wird statt einer Konstanten eine Wahrscheinlichkeitsverteilung von Reisezeiten errechnet, bei der z.B. verschiedene Geh-Geschwindigkeiten und Faktoren für nötige Umwege eingerechnet werden.

Das Monitor-Frontend basiert auf dynamisch generierten HTML-Seiten. Diese wurden im Hinblick auf Effizienz, Kompatibilität, Barrierefreiheit und Datenschutz komplett ohne Javascript, Sessions, Cookies, etc. umgesetzt.

Die gesamte Software wurde in der Programmiersprache Rust umgesetzt. Das Bauen und Betreiben der Software wurde mittels Docker und Docker-Compose automatisiert, so dass interessierte Personen mit möglichst geringem Aufwand eine eigene Instanz aufsetzen können.

Am Rande der Entwicklung sind kleinere Nebenergebnisse entstanden, wie z.B. eine [Übersicht](#)¹ über die Versorgungslage mit offenen ÖPV-Echtzeitdaten und ein [Werkzeug](#)² zur Sammlung der Angaben über Verkehrsbetriebe in Deutschland.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Durch die Anpassung des Entwicklungsziels - Abfahrtsmonitor statt Routensuche - wurde sichergestellt, dass am Ende der Förderzeit bereits eine Software besteht, die für interessierte Endanwender_innen, hier Reisende im ÖPV, nutzbar und nützlich ist. Sie können den Monitor bei der Planung und Ausführung von Reisen im ÖPV nutzen, und darüber hinaus ihr "individuelles, intuitives Verspätungsmodell" mit den Statistiken abgleichen. Der Monitor kann auf Desktop- und Mobilgeräten genutzt werden.

Für andere Software-Entwickler_innen bietet unsere Software mehrere Anknüpfungspunkte. Andere Projekte, die ebenfalls aufgezeichnete Verspätungsdaten, daraus abgeleitete Verspätungsmodelle und/oder Prognosen benötigen, können eine eigene Instanz von Dystonse betreiben und z.B. auf Datei- oder Datenbank-Ebene auf die Ergebnisse zugreifen. Die ständig aktualisierten Prognosen sollen zudem in Kürze als GTFS-Realtime-Feed bereitgestellt werden. Damit können diese über das Internet abgerufen werden, ohne eine eigene Instanz von Dystonse zu betreiben.

Wir sind nach wie vor daran interessiert, die ursprünglich geplante Routensuche umzusetzen. Wir versuchen derzeit, einen organisatorischen und finanziellen Rahmen dafür zu finden, idealerweise durch eine Beauftragung durch oder Kooperation mit einem Verkehrsbetrieb.

Für das Backend und den Abfahrtsmonitor gibt es eine Reihe von Ideen zur Weiterentwicklung:

- Bewertungsfunktion für Routen, die mit anderer Software ermittelt wurden, und Integration dieser Bewertungen in vorhandene Apps und Websites
- Werkzeug zur Fahrplanoptimierung, welches Hotspots für häufige Betriebsverzögerungen findet und die realen Auswirkungen auf Umstiege bemisst
- Anbindung weiterer Formate und APIs, Interoperabilität zwischen Datensätzen (ggf. als eigenständiges Paket/Projekt)

¹ <https://github.com/dystonse/dystonse/blob/master/project-status/Datenquellen.md>

² <https://github.com/dystonse/dystonse-tools/tree/master/agencies>

- Einbindung von OpenStreetMap und [Pfadle](#)³, um die konkreten Fahrtverläufe darstellen zu können
- Fußweg-Routing und exaktere Umsteige-Zeiten (basierend auf OSM und/oder dem [Graphenmodell aus GTFS](#)⁴)

Für uns persönlich und als Gruppe hat das Projekt ebenfalls positive Auswirkungen:

Zu Umsetzung von Dystonse haben wir die Programmiersprache Rust neu erlernt, die sich als exzellente Wahl für dieses Projekt heraus gestellt hat und sicher auch bei künftigen Projekten zum Einsatz kommen wird.

Wir konnten unsere Verbindungen in die Open-Source-, Open-Data- und ÖPNV-Communities ausbauen. Durch unsere eigene Recherche und den Austausch mit eben diesen Communities haben wir nun einen besseren Überblick darüber, welche Probleme und Lösungen es in diesen Gebieten gibt und was aktuelle Entwicklungen sind.

Auf der persönlichen Ebene konnten wir außergewöhnlich positive Erfahrung mit der Teamarbeit im gemeinsamen Home Office machen und unsere Produktivität und Organisations-Skills deutlich verbessern, trotz der bedrückenden globalen Lage in Zeiten der Pandemie.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Anfangs haben wir versucht, die Fahrplandaten, die uns im GTFS-Format geliefert werden, in eine relationale Datenbank zu importieren. Vorhandene Lösungen waren zu unperformant, und anstatt eine eigene Implementierung zu entwickeln, haben wir stattdessen eine Library ausgewählt, die Fahrpläne aus GTFS-Dateien in den RAM lädt und in Rust-Programmen effizienter und direkter nutzbar macht.

Fahrplan-Daten werden von den Verkehrsbetrieben bereitgestellt und häufig verändert. Unsere Aufzeichnungen ergeben, dass mindestens täglich Änderungen an den Daten vorgenommen werden. Für unsere Arbeit erschien es wichtig, eine Vorstellung davon zu bekommen, welcher Art und Umfang diese täglichen Änderungen sind. Wir haben Versuche unternommen, um GTFS-Daten in git-Repositories zu versionieren. Neben der Analysierbarkeit mit gängigen git-Werkzeugen erhofften wir uns außerdem Einsparungen bei der Archivierung der Daten. Es stellte sich heraus, dass Fahrplandaten schon für einen einzigen Verkehrsverbund zu umfangreich sind, d.h. dass git nicht in der Lage ist, csv-Dateien mit der dort zu erwartenden Menge an Zeilen absturzfrei zu verarbeiten.

Ein weiterer fruchtloser Versuch war der Betrieb der Dystonse Software auf CPUs mit ARM-Architektur. Zwar gelten alle Komponenten von Dystonse als portabel und auf ARM einsetzbar, doch die Integration der Komponenten mittels Docker scheitert an Programmfehlern. Die Fehler liegen im Bereich von Docker und/oder dem Linux-Kernel und sind dort bekannt und dokumentiert.

Und schließlich wurde die Routensuche vorerst verworfen, um schneller zu einer benutzbaren Software für Reisende zu kommen.

³ <https://github.com/ad-freiburg/pfaedle>

⁴ <https://developers.google.com/transit/gtfs/reference#pathwaystxt>

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Der erweiterte Abfahrtsmonitor ist unter <https://monitor.dystonse.org> erreichbar und funktioniert sowohl auf Desktop- als auch auf Mobilbrowsern. Eine ausführliche Dokumentation für Nutzer_innen ist über den Link [Hilfe](#)⁵ ebenso erreichbar.

Das Projekt wurde, wie alle Projekte der 7. Prototype-Fund-Förderrunde, durch einen [Artikel](#)⁶ auf der Website des Prototype Fund vorgestellt.

Technische Hintergründe und Berichte aus der Entwicklungszeit haben wir außerdem in [unserem eigenen Blog](#)⁷ beschrieben. Die wöchentlichen Statusberichte, die im Rahmen der Förderung entstanden sind, sind ebenfalls [online verfügbar](#)⁸ und werden vorerst über den Förderzeitraum hinaus weiter gepflegt.

Informationen zum Bauen und Betreiben einer eigenen Software-Instanz finden sich in den Readme-Dateien unserer GitHub-Repositories, insbesondere in [dystonse-docker](#)⁹.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die ersten Schritte der Entwicklung, vor allem die Suche nach geeigneten Datenquellen und die Implementierung der Datensammlung in Verbindung mit einer Datenbank, stellten sich als zeitaufwändiger heraus, als vorher geplant.

Zusätzlich hat sich die angespannte Lage durch die Coronavirus-Pandemie im April und Mai deutlich auf unsere generelle Produktivität ausgewirkt, so dass wir in den ersten Monaten deutlich langsamer voran kamen, als ursprünglich geplant.

Als absehbar war, dass die Abschlussveranstaltung (Demo Day) nicht vor Ort stattfinden kann, wurde stattdessen das Konzept “Demo Week” als Online-Format geplant – das brachte unter anderem mit sich, dass viele Ergebnisse schon deutlich früher als üblich fertig sein sollten, um in der Präsentation an das neue online-Format angepasst zu werden. Daher haben wir uns dann entschieden, zunächst eine andere Frontend-Variante vorzuziehen, da diese besser planbar war als die eigentliche Routensuche. So entstand der interaktive, erweiterte Abfahrtsmonitor, der alle Elemente der Wahrscheinlichkeitsbasierten Routensuche visualisiert, aber die Suche nicht automatisch durchführt, sondern den Nutzenden die Möglichkeit gibt, sich selbst eine Reisekette zusammenzustellen. Die Entwicklung der voll-automatischen Routensuche haben wir stattdessen auf einen späteren Zeitpunkt verschoben.

Die Entwicklung des erweiterten Abfahrtsmonitors ging erfreulicherweise schneller als zunächst erwartet, so dass wir innerhalb eines Monats sogar viele optionale Features einbauen konnten.

⁵ <https://monitor.dystonse.org/help>

⁶ <https://demoweek.prototypefund.de/projects/02-dystonse.html>

⁷ <https://blog.dystonse.org/>

⁸ <https://github.com/dystonse/dystonse/tree/master/project-status/Montagsupdates.md>

⁹ <https://github.com/dystonse/dystonse-docker>

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gegen Ende der Förderzeit wurden wir auf die [Dissertation](#)¹⁰ von Dr. Mohammad Hossein Keyhani mit dem Titel “Computing Highly Reliable Train Journeys” aufmerksam gemacht. Darin wird ein Ansatz beschrieben, der dem von Dystonse sehr ähnlich ist. Der Ansatz wurde als Erweiterung von [MOTIS](#)¹¹ entwickelt. Zwar ist MOTIS seit April 2020 unter der MIT-Lizenz verfügbar, das betrifft jedoch nicht die Teile aus der Arbeit von Keyhani.

Die wahrscheinlichkeitsbasierte Routensuche in Dystonse umzusetzen bleibt damit weiterhin relevant. Bevor dies geschieht, werden wir die genannte Quelle genau prüfen, um nützliche Ansätze zu übernehmen.

¹⁰ <https://tuprints.ulb.tu-darmstadt.de/6227/>

¹¹ <https://github.com/motis-project/motis>

30. Sep. 2020

Eingangsnr.: **LfBoBo**

BRIGHTSKY

SCHLUSSBERICHT

Zuwendungsempfänger:

JAKOB DE MAEYER

28. September 2020

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01S19S24** gefördert.

Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Zusammenfassung

Der Deutsche Wetterdienst erzeugt Wetteraufzeichnungen und -vorhersagen für das gesamte Bundesgebiet. Die so erhaltenen öffentlich finanzierten Daten werden der Öffentlichkeit zwar kostenfrei zur Verfügung gestellt, zur Nutzung derselben ist aber ein erheblicher Programmieraufwand – zur Zusammenführung der vielen verschiedenen Wetterparameter und zum Parsing der Vielzahl unterschiedlicher Datenformate auf dem DWD-Server – notwendig. Dem gegenüber stehen kommerzielle Lösungen, welche fertig aufbereitete Wetterdaten in sehr leicht zu bedienenden Schnittstellen anbieten. Bright Sky überbrückt diesen Spalt und stellt auf Basis der öffentlichen Wetteraufzeichnungen eine kostenfreie Schnittstelle bereit, welche in der Einfachheit ihrer Nutzung den kommerziellen Anbietern ebenbürtig ist.

INHALTSVERZEICHNIS

1	Motivation	2
2	Meilensteine	2
3	Beitrag zu den Zielen der Förderinitiative „Software Sprint“	3
4	Ergebnisse	4
3	Präsentationsmöglichkeiten	5
6	Ausblick	6

1 MOTIVATION

Der Deutsche Wetterdienst (DWD) bietet eine Vielzahl wetter- und klimabezogener Dienste an. Neben der Bereitstellung von Wetterwarnungen und der Erstellung von Wettervorhersagen gehört dazu auch der Betrieb von knapp 200 hauptamtlichen Wetterstationen in Deutschland sowie das Sammeln von weiteren Aufzeichnungen von etwa 1800 ehrenamtlich betreuten Wetterstationen. Die 2017 in Kraft getretene umfangreiche Reform des Gesetzes über den Deutschen Wetterdienst berechtigte den DWD diese öffentlich finanzierten Aufzeichnungen und andere Leistungen entgeltfrei der Öffentlichkeit bereitzustellen. Seitdem wurde das Angebot an Wetterdaten auf dem DWD Open Data Server¹ kontinuierlich verbessert und erweitert.

Doch das Format, in dem Wetteraufzeichnungen gespeichert werden, ist naturgemäß bestimmt durch die Art, wie diese erhoben werden: als Zeitreihen einzelner Messinstrumente. So findet sich auf dem Open-Data-Server beispielsweise eine einzelne Datei mit allen Niederschlagsaufzeichnungen der Station 1766 (Münster/Osnabrück) von 1995 bis 2019. Dem gegenüber sind typische Nutzer*innen von Wetterdaten – z. B. der Betreiber einer Wetterapp, oder die Wissenschaftlerin, welche Ernteaufträge untersucht – weder an den Details des DWD-Stationsnetzes (welche Station liegt wo?, welche Station misst welche Parameter?) noch an lediglich einzelnen Messinstrumenten interessiert; ihre Frage hat die Form „wie waren (oder werden) Temperatur, Niederschlag, Wind und Wolkendichte an den Koordinaten XY zum Zeitpunkt Z?“

Zur Nutzung der DWD-Aufzeichnungen ist daher ein erheblicher Programmieraufwand – zur Zusammenführung der vielen verschiedenen Wetterparameter und zum Parsing der Vielzahl unterschiedlicher Datenformate auf dem DWD-Server – notwendig. Die Konsequenz ist häufig die Aufgabe des Vorhabens oder das Ausweichen auf kommerzielle Anbieter, welche Wetterdaten im gewünschten Format bereitstellen. Ziel von Bright Sky ist es, diesen Spalt zwischen Format der DWD-Daten und Nutzer*innenansprüchen zu überbrücken und auf Basis der öffentlichen Wetteraufzeichnungen eine kostenfreie Schnittstelle bereitzustellen, welche in der Einfachheit ihrer Nutzung den kommerziellen Anbietern ebenbürtig ist.

2 MEILENSTEINE

Bright Sky's technische Umsetzung lässt sich grob in drei Aufgabengebiete teilen:

1. Laden und Parsing der Rohdaten vom DWD-Datenserver,
2. Entwicklung einer Datenbankstruktur zum Hinterlegen der aufbereiteten Wetter- und Stationsdaten, sowie
3. Bereitstellung der aufbereiteten Daten durch eine öffentlich verfügbare, leicht bedienbare, maschinelle Schnittstelle (JSON API).

Aus diesen lassen sich die folgenden Meilensteine ableiten, welche als Grundlage zur Strukturierung der sechsmonatigen Entwicklungszeit dienen:

1. Erste stabile Version eines Parsers für Wettervorhersagen (MOSMIX)
2. Erste stabile Version eines Parsers für historische Wetterdaten

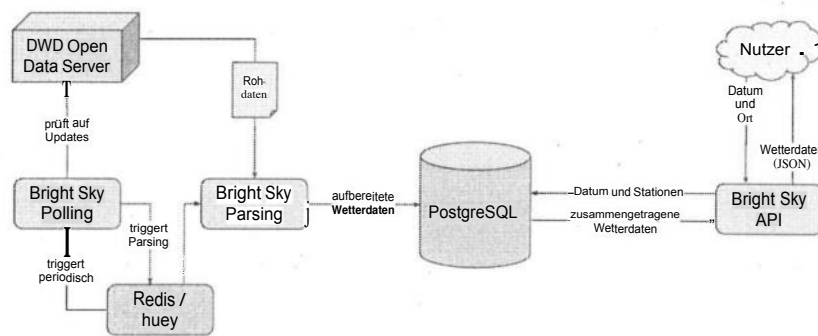


Abbildung 1: Bright Sky's Architektur und Datenfluss.

3. Erste stabile Version eines Parsers für augenblickliche Wetterdaten (SYNOP)
4. Fertigstellung der Datenbankstruktur zum Speichern von Stations- und Wetterdaten
3. Automatische Erkennung neuer Updates auf dem DWD-Server inkl. Verarbeitung und Update der Datenbank
6. Erste Version der JSON API
7. „Hardening“ der API unter hoher Auslastung
8. Deployment der Infrastruktur in Production-Umgebung
9. Fertigstellung einer Landing Page und Dokumentation der API

3 BEITRAG ZU DEN ZIELEN DER FÖRDERINITATIVE „SOFTWARE SPRINT“

Zielgruppe des *Software Sprint* sind Softwareprojekte, die zivilgesellschaftliche Interessen realisieren. Dass ein solches Interesse besteht, zeigt allein der Posten von fast 400 Millionen Euro im jährlichen Bundeshaushalt², mit dem die Bundesrepublik den Deutschen Wetterdienst finanziert.

Durch die Nutzarmachung der DWD-Wetteraufzeichnungen tritt Bright Sky vor allem als *Wegbereiter* für Vorhaben in den Bereichen Civic Tech und Civic Science auf. So wurde die Projektidee beispielsweise geboren, als Code for Münster, ein lockerer Zusammenschluss an Open-Data-interessierten münsteraner Menschen, nach langem Hinwirken bei der Polizei an einen Datensatz zu Verkehrsunfällen im Raum Münster gelangen konnte.³ Korreliert mit den DWD-Wetterdaten ließen sich hieraus beispielsweise Hinweise ableiten, welche Kreuzungen oder Straßenabschnitte bei Nässe besonders gefährlich sind. Doch der hohe Aufwand, der mit dem Auslesen der DWD-Rohdaten verbunden war, legten diese Auswertung damals (vor der Veröffentlichung von Bright Sky) auf Eis. Schon kurz nach der ersten Veröffentlichung einer frühen Version von Bright Sky erreichten mich Emails von Anwenderinnen, die die Wetterdaten für Forschung an Erntevorhersagen, in Open-Source-Projekten zur Heimautomatisierung, in Software zur Unterstützung von Segelvereinen und sogar als Warnsystem vor Starkregen in Mittelamerika nutzen.

4 ERGEBNISSE

Bright Sky's Entwicklung profitierte vom glücklichen und seltenen Umstand, dass das Endergebnis stets sehr klar definiert war: Eine JSON-API für historische Wetterdaten ähnlich den kommerziellen Anbietern, nur auf Basis des DWD-Datensatzes. Als Folge entsprach jeder der in Abschnitt 2 vorgestellten Meilensteine einem klaren und umsetzbaren Entwicklungsschritt. So konnte nach drei Monaten eine erste Version der Wetter-API öffentlich verfügbar gemacht werden, und der Förderzeitraum kulminierte mit der Veröffentlichung einer stabilen Version 1.0.

```
$ curl https://api.brightsky.dev/weather?lat=52&lon=7.6&date=2020-04-21
```

```
{
  "weather": [
    {
      "timestamp": "2020-04-21T00:00:00+00:00",
      "source_id": 6007,
      "cloud_cover": 0,
      "dew_point": -2.5,
      "precipitation": 0,
      "pressure_msl": 1015.1,
      "relative_humidity": 40,
      "sunshine": 0,
      "temperature": 10.6,
      "visibility": 50000,
      "wind_direction": 70,
      "wind_speed": 12.6,
      "wind_gust_speed": 33.5,
      "condition": "dry",
      "icon": "clear-night"
    },
    ...
  ],
  "sources": [
    {
      "id": 6007,
      "dwd_station_id": "01766",
      "wmo_station_id": "10315",
      "station_name": "Muenster/Osnabrueck",
      "observation_type": "recent",
      "first_record": "2020-01-01T00:00:00+00:00",
      "last_record": "2020-08-13T23:00:00+00:00",
      "lat": 52.1344,
      "lon": 7.6969,
      "height": 47.8,
      "distance": 16365
    }
  ]
}
```

Auflistung 1: Beispielanfrage und Auszug aus Bright Sky's Antwort für das Wetter vom 21. April 2020 am Breitengrad 52, Längengrad 7.6.

Bright Sky bietet drei Schnittstellen zum maschinellen Auslesen von Wetterdaten:

- Stündliche Wetterdaten, sowohl historisch (rückgehend bis 2010) als auch prognostiziert (bis zu 10 Tage)
- Aktuelle SYNOP-Meldungen
- Aus den aktuellen SYNOP-Meldungen zusammengefasstes „derzeitiges Wetter“

Alle Endpunkte lassen sich sowohl mit Breiten- und Längengrad als auch mit DWD-Stationsnummer ansprechen. Der Datensatz enthält die meisten der gängigen Wetterparameter, u.a.:

- Temperatur
- Taupunkt
- Bewölkung
- Luftdruck
- Luftfeuchtigkeit
- Sichtweite
- Windstärke und -richtung

Eine öffentlich verfügbare Instanz der Schnittstelle ist unter

<https://api.brightsky.dev/>

verfügbar. Diese bedient derzeit etwa 10.000 Anfragen pro Tag. Updates von Datensätzen auf dem Open-Data-Server des DWD sind typischerweise nicht später als fünf Minuten nach ihrer Veröffentlichung auch in Bright Sky eingepflegt.

Bright Sky folgt vielen „Good Practices“ in der Softwareentwicklung. So ist die Schnittstelle in einer OpenAPI-Spezifikation abgebildet, folgt die gesamte Codebasis dem in Python gängigen Stylestandard PEP8 und sind alle kritischen Teile des Projektes mit Unit Tests abgedeckt. Die für eine Instanz erforderliche Infrastruktur lässt sich durch ebenfalls veröffentlichte modulare Docker Compose-Konfigurationen leicht reproduzieren.

5 PRÄSENTATIONSMÖGLICHKEITEN

Die Ergebnisse dieses Fördervorhabens sind für die Öffentlichkeit unter

<https://brightsky.dev/>

mit einer Landing Page inklusive Online-Demo und ausgiebiger technischer Dokumentation der Schnittstelle aufbereitet. Der Quellcode und die Projektverwaltung sind über GitHub veröffentlicht und über die Adresse

<https://github.com/jdemaeyer/brightsky>

abrufbar. Parallel dazu finden sich Konfigurationsdateien und Informationen zur Infrastruktur unter

<https://github.com/jdemaeyer/brightsky-infrastructure>.

6 AUSBLICK

Das primäre Ziel dieses Fördervorhabens war eine öffentliche, für alle Zwecke kostenfrei verwendbare und nutzerfreundliche Schnittstelle für Wetterdaten auf Basis der im Rahmen des Open-Data-Programms des Deutschen Wetterdienstes veröffentlichten Wetteraufzeichnungen und -prognosen. Eine solche API steht der Öffentlichkeit und insbesondere den Civic-Science- und Open-Source-Communities nun und für die absehbare Zukunft unter

<https://brightsky.dev/>

zur Verfügung.

Trotz seines jungen Alters wird Bright Sky schon jetzt bspw. in Home Assistant, einem beliebten Open-Source-Projekt zur Heimautomatisierung⁴, eingesetzt. Für ihre Lösung zur Flugplanung und -auswertung im Segelfliegen haben Entwickler von WeGlide⁵ zudem Unterstützung bei der Erweiterung um eine Schnittstelle für flächendeckende Wetterkarten angekündigt.

Die monatlichen Kosten zum Betrieb der öffentlichen Instanz liegen unter zehn Euro, so dass ich diese und auch das Management für die weitere Entwicklung von Bright Sky mit Freude weiterhin ehrenamtlich betreibe.

ANMERKUNGEN

¹<https://www.dwd.de/DE/leistungen/opendata/opendata.html>

²https://www.dwd.de/SharedDocs/downloads/DE/allgemein/zahlen_und_takten.pdf

³<https://crashes.codeformuenster.org/>

⁴<http://www.home-assistant.io/>

⁵<https://weglide.org/>

Richtlinie zum „Software-Sprint“

OpenRedact – Deutsche Texte von personenbezogenen Daten bereinigen

Schlussbericht

Zuwendungsempfänger:

Berrang, Langhabel, Ostendorff und Ostendorff GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S25 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 Wunsch von Unternehmen, Kund*innendaten vollumfänglich für ihre geschäftlichen Interessen zu nutzen, widerspricht dem Bedürfnis der Nutzer*innen nach Datenschutz. Das Bewusstsein für Datenschutz ist seit der Einführung der europäischen Datenschutzgrundverordnung gestiegen und stellt eine Herausforderung für bestehende Datensätze sowie das Erschließen neuer Geschäftsfelder dar. Staatliche Stellen müssen bislang enormen Aufwand betreiben, um ihre Daten für die Veröffentlichung zu anonymisieren, denn die händische Bereinigung dieser Daten ist aufwändig und durch den erheblichen Umfang kaum realisierbar.

Das Ziel von OpenRedact (ursprünglicher Name: Schwärzer) ist die Stärkung der Transparenz der Verwaltung durch den Einsatz von technischen semi-automatisierten Anonymisierungs-Applikationen, um die Verwaltung bei Informationsfreiheitsgesetzanfragen und bei der Veröffentlichung von Gerichtsentscheidungen zu entlasten. Auch Firmen und Kund*innen können von einer Anonymisierungslösung profitieren, in dem Daten in neuen Geschäftsfeldern genutzt oder Kundendaten besser geschützt werden können.

Die geplante Vorgehensweise war es, sowohl eine prototypische Anonymisierungs-Applikation zu implementieren, als auch einen juristischen Leitfaden zur Anonymisierung zu schreiben. Um eine größtmögliche Nachnutzbarkeit zu ermöglichen haben wir geplant die Logik der Applikation in

mehrere funktionale Komponenten aufzuteilen, die als eigenständige Open Source Projekte bestehen können. Diese sind ein Modul zum Extrahieren und Modifizieren des Textinhalts von Dokumenten, ein Modul zum Erkennen von personenbezogenen Daten, und ein Modul zur Anonymisierung von Daten.

Die wichtigsten Meilensteine waren:

- MS0: Entwicklung einer juristischen Richtlinie für Anonymisierung.

Auswertung von Anonymisierungsmöglichkeiten, Empfehlungsleitfaden für die Anonymisierung und Pseudonymisierung bei Urteilen, evt. Aufsatz zum Thema der Anonymisierung und Pseudonymisierung von Urteilen und der Frage, müssen diese überhaupt anonymisiert werden.

- MS1: Erstellen eines Anonymisierungs-Framework das gegebene Textpassagen anonymisiert, pseudonymisiert beziehungsweise mit Methodiken der Differential Privacy verschleiert.

Textpassagen können je nach Kategorie mit Hilfe von unterschiedlichen Methoden anonymisiert werden. Folgende Anonymisierungstechniken sollen unterstützt werden (leicht nach schwer): Schwärzen, Pseudonymisieren, Generalisieren, Differential Privacy.

- MS2: Regelbasierte und statistische Identifikation von personenbezogenen Daten mit verfügbaren Sprachmodellen.

Mindestens 10 verschiedene Arten von personenbezogenen Daten können erkannt werden. Mindestens 5 davon sollen mit hohem F2-Maß erkannt werden.

- MS3: Implementation der Web App.

Die App kann sowohl über eine grafische Nutzeroberfläche als auch eine API angesprochen werden. Der Fokus der grafischen Nutzeroberfläche liegt dabei auf einfacher Nutzbarkeit. Das Tool kann deutsche Texte in verschiedenen Formaten (mindestens jedoch einfacher Text) analysieren. Der Anonymisierungsvorschlag bzw. die identifizierten personenbezogenen Daten können manuell durchgegangen und angepasst werden. Der Benutzer erhält mit der Web App auch die Möglichkeit die Performance auf von ihm gegebenen Testdaten zu messen. Die Performance wird insbesondere über Indikatoren wie "false positive" und "false negative" Raten angegeben.

In der Planungsphase zu Beginn der Förderung haben wir folgenden ursprünglich angegebenen Meilensteinen, welche der Optimierung der Erkennung von personenbezogenen Daten dienen, niedrige Priorität gegeben.

- MS4: Erstellen eines Benchmark Corpus für die Identifikation der personenbezogenen Daten und messen der Performance unter Angabe der "false positive" und "false negative" Raten.
- MS5: Verbessern der Performance des statistischen Sprachmodells durch Transfer Learning und durch generieren von Trainingsdaten.

Am Ende steht ein Prototyp, der beispielhaft zeigt, wie Anonymisierung und Pseudonymisierung funktionieren könnte.

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 Haupt-Zielgruppe für die semi-automatisierte Lösung ist die öffentliche Verwaltung, da durch die semi-automatisierte Erkennung von personenbezogenen Daten eine zeitsparende Anonymisierung ermöglicht werden kann. Dadurch wird der öffentlichen Verwaltung die Realisierung von eGovernance und Open Data sowie der zeitgleiche Schutz von personenbezogenen Daten in Einklang gebracht.

Weitere Zielgruppen sind Unternehmen und Journalisten. Diese profitieren vom Projekt indem sie ihre Daten auf vielfältigere und umfassender Weise nutzen können unter gleichzeitiger Wahrung ihrer Verpflichtungen die Daten zu schützen. Auf der anderen Seite profitieren die Kunden beziehungsweise Bürger, indem ihre Daten besser anonymisiert und damit geschützt werden können.

Auf technischer Ebene haben wir OpenRedact in unabhängige Komponenten gegliedert. Jede einzelne Komponente erfüllt eine eigenständige Aufgabe und kann dadurch unabhängig von OpenRedact einen Mehrwert liefern. Zielgruppe für diese Komponenten sind Software Entwickler*innen die in den Bereichen Anonymisierung, Bearbeiten von Dokumenten und Erkennung von Daten in Dokumenten arbeiten. Auf diesem Wege fördern wir die langfristige Nachnutzung der im Projekt entstandenen Software.

In einem interdisziplinären Team haben wir mit dem Projekt an den Zielen Transparenz und Datenschutz gearbeitet, welche die ganze Gesellschaft betreffen. Transparenz und Datenschutz stehen mit den Zielen des Software Sprints im Einklang, und wir leisten mit OpenRedact einen Beitrag diese Ziele mit freier Open Source Software erreichbar zu machen.

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 Teilnahme am Prototype Fund hat es OpenRedact ermöglicht, die Idee von einer offenen teil-automatisierten Anonymisierungs-Applikation umzusetzen. Im Laufe der letzten sechs Monate hat OpenRedact technische und juristische Lösungen entwickelt, die helfen können den Aufwand und die Kosten für die Veröffentlichung von Dokumenten als Open Data zu senken.

Mit Hilfe unserer Web-App können PDFs und andere Dokumente von personenbezogenen Daten bereinigt werden. Bei der Entwicklung des Prototyps haben wir besonderen Wert auf die Nachnutzbarkeit gelegt. Alle Komponente sind eigenständig nutzbar. Wir zeigen mit unserem Prototypen die technischen Möglichkeiten von teil-automatisierter Anonymisierung und bieten eine Grundlage für die zukünftige Zusammenarbeit und Nutzung des öffentlichen Sektors.

Der Anonymisierungsleitfaden soll Behörden und der Justiz das Vorhaben von Open Data unter Wahrung des Datenschutzes ermöglichen. Grundsätzlich sind personenbezogene Daten zu anonymisieren. Insbesondere der Anonymisierungsleitfaden soll auch über die Projektgrenzen hinweg eine juristische Diskussion zu Open Data und Datenschutz anstoßen.

Darüber hinaus haben unsere Arbeiten gezeigt, dass beim Einsatz von OpenRedact branchenspezifische Anforderungen beachtet werden müssen. Aus diesem Grund haben wir Anpassungsmöglichkeiten besonders berücksichtigt. Beispielsweise können die Erkennungsmethoden der Komponente zur Erkennung von personenbezogenen Daten an den genauen Anwendungsfall angepasst werden.

Die Meilensteine 0 bis 3 konnten wie geplant erreicht werden. Die Meilensteine 4 und 5, denen in der Planungsphase zu Beginn der Förderung eine niedrige Priorität gegeben wurde, wurden nicht angegangen. Der Grund für deren niedrige Priorität und das Nicht-bearbeiten ist, dass es sich bei diesen Meilensteinen um eine für den Prototypen nicht notwendige Optimierung der Erkennung von personenbezogenen Daten handelt. Stattdessen haben wir eine Möglichkeit geschaffen, schnell und einfach regelbasierte Methoden zur Erkennung von personenbezogenen Daten selbst zu schreiben, was wir vor dem Einsatz auf einer neuen Art von Dokumenten als notwendig erachten. Die von der Jury vorgegebene Angabe von "false positive" und "false negative" Raten ist dennoch möglich, und zwar auf Basis des jeweils bearbeiteten Dokuments. Die Performance der automatischen Erkennung von personenbezogenen Daten wird auf Basis der manuellen Korrekturen bestimmt und angezeigt.

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?

OpenRedact zeigt beispielhaft wie eine teil-automatisierte Anonymisierung in der öffentlichen Verwaltung eingesetzt werden kann, beispielsweise bei Informationsfreiheitsgesetzanfragen oder bei der Veröffentlichung von Urteilen, indem personenbezogene Daten geschwärzt werden. Mit dem Einsatz einer Anonymisierungs-Applikation kann die öffentliche Verwaltung entlastet werden, weil keine händische Anonymisierung erfolgen muss. Sicherlich bleibt eine abschließende Überprüfung notwendig, jedoch kann die teil-automatisierte Anwendung Informationsfreiheitsgesetzanfragen erleichtern.

Für die teil-automatisierte Anonymisierung von Dokumenten in der öffentlichen Verwaltung stellt OpenRedact derzeit einen Prototypen dar. Für den finalen Einsatz einer semi-automatisierten Anwendung bedarf es einer Untersuchung zum Einsatz von OpenRedact in der öffentlichen Verwaltung. Die Arbeit zeigt in technischer Hinsicht was mit Automatisierung und Methoden der künstlichen Intelligenz ermöglicht werden kann.

Durch die Open Source Veröffentlichung der eigenständigen Komponenten haben wir Lücken gefüllt, für die es noch keine Software-Bibliotheken gab. Wir hoffen dadurch langfristigen Nutzen für mögliche Anwender*innen geschaffen zu haben. Ohne die Open Source Veröffentlichung wäre dies nicht möglich. Mit dem Anonymisierungsleitfaden haben wir eine Grundlage für die weitere Diskussion von Anonymisierung amtlicher Werke geschaffen.

Fachlich war das Projekt sehr lehrreich. Im interdisziplinären Team haben wir unsere Kenntnisse für den gesamten Software-Entwicklungs-Zyklus zusammen getragen und uns dadurch nicht nur neues Wissen erarbeitet, sondern auch Bekanntes weitergegeben. Insbesondere auch die UX-Coachings haben unsere Weiterentwicklung unterstützt.

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 von der Bereitstellung einer Live-Demo des Prototypen aufgrund der Haftung bei Fehlern des Programmes abgesehen.

Des Weiteren haben wir unsere Arbeiten an einer Kooperation mit einer Plattform für Informationsfreiheitsgesetzanfragen eingestellt. Ziel einer solchen Kooperation war es Trainingsdaten und echten Anwendungsdokumente zu erhalten. Der Grund dafür war, dass die verfügbaren Daten für uns nicht als Trainingsdaten geeignet waren, und wir dem Trainieren von Modellen des Maschinellen Lernens ohnehin eine niedrige Priorität gegeben hatten.

Anstatt einer Komponente die nur personenbezogene Daten in deutschen Dokumenten identifiziert, haben wir eine Komponente für allgemeine Named-entity recognition (NER) geschrieben. Wir haben gelernt, dass Erkennungsmethoden – um beste Ergebnisse erzielen zu können – spezifisch für einen Sprachraum oder eine Art von Dokumenten sein müssen. Statt den Fokus auf eine Art von Dokumenten zu legen, haben wir eine universell nutzbare Suite für NER gebaut, die das Ziel hat, dass Nutzer*innen schnell eigene Erkennungsmethoden für die spezifischen vorliegenden Daten schreiben können.

Die Suche nach Kooperationspartnern von denen wir Testdokumente erhalten und die Tester darstellen können, hat zu keinen Ergebnissen geführt. Kontaktierte offizielle Stellen, welche unsere Hauptzielgruppe darstellen, waren uns gegenüber nicht offen für die Durchführung einer Testphase mit einem Prototypen – im Gegensatz zu einem fertigen Produkt. Auch die Corona-Pandemie hat die Kontaktaufnahme erschwert. Alternativ haben wir die Anforderungen der Software auf den Erkenntnissen basiert, die wir im Team erlangt und im UX-Coaching optimiert haben.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Auf unserer Homepage (<https://openredact.org/>) kann man sich über unser Projekt informieren. Medienwirksam sind unser Blogpost der Demo-Woche des Prototype Funds, unsere detaillierten technischen Blogposts auf Medium, und unsere Aktivität auf Twitter. Der Quellcode ist auf GitHub (<https://github.com/openredact>) veröffentlicht.

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?

Nach der initialen Planphase nach Förderbeginn wurden keine große Änderung des Plans mehr vorgenommen. Die Bearbeitungszeit der Teilaufgaben war stets im Rahmen des Plans.

Durch die frühen Abgabefristen für die Demo-Woche mussten einige Arbeiten vom August in den Juli vorgezogen werden.

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 mit Behörden über sog. Anonymisierungsleitfaden ausgetauscht. Die dadurch gewonnen Erkenntnisse wurden in der Erstellung unseres Anonymisierungsleitfadens berücksichtigt.

Richtlinie zum „Software-Sprint“

CO2-Datenbank

Schlussbericht

Zuwendungsempfänger:

Lars Wissler Elias Philipp GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S26 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Der Klimawandel ist die größte Gefahr für die Menschheit und die Erde – wir alle müssen unseren Teil dazu beitragen, die globale Erwärmung zu reduzieren. Viele bemühen sich bereits um ein nachhaltiges Leben, doch wer nicht gut informiert ist, kann keine wissensbasierten Entscheidungen treffen. Die EU hat dieses Problem bereits erkannt und unter anderem die Richtlinie 2006/12/EG ins Leben gerufen, welche große Unternehmen dazu verpflichtet, regelmäßig über ihre Nachhaltigkeitsbestrebungen und die Aufschlüsselung ihrer CO₂-Produktion zu informieren. Leider sind die generierten Informationen ohne zentrale Sammel- und Anlaufstelle nicht hilfreich. Erst mit einer solchen können die Daten aufbereitet sowie verglichen werden und das ökologische Verhalten großer Unternehmen transparent gemacht werden. Hierfür sollte eine Webplattform inklusive Web-Crawler und Parser entwickelt werden, die Unternehmensberichte nach Veröffentlichungen von CO₂-Emissionsdaten durchsucht, strukturiert und frei zugänglich anbietet.

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

Kernzielgruppe sind Nutzer, die in ihrem professionellen Umfeld umfassende, gut belegte und vergleichbare Daten zu CO₂-Emissionen von Unternehmen benötigen. Insbesondere zielt das Projekt auf Journalisten, denen die Themenfindung und Berichterstattung zum Einfluss großer Unternehmen auf das Klima vereinfacht wird. Aus diesem Grund bietet das Portal direkte Integration in die populärste Datenvisualisierungsumgebung der Branche, Datawrapper.de, an. Jeder Datenpunkt der ethidat-Datenbank ist mit der entsprechenden Quelle hinterlegt und kann auch als CSV-Datei abgefragt werden. So wird transparenter, welche Unternehmen wirklich klimaneutraler agieren als zuvor und die Berichterstattung kann von harten Fakten untermauert werden.

Ausführliche Darstellung der Ergebnisse

Um unser Ziel zu erreichen, haben wir einen Web-Crawler entwickelt, der ausgehend von einer Liste von 25.000 Unternehmenswebseiten das Internet nach Nachhaltigkeitsberichten und Jahresabschlüssen durchsucht. Als Grundlage für den Crawler haben wir Nutch, die Open Source Crawling Library von Apache verwendet, die weitreichende und skalierbare Funktionalitäten bereitstellt für Crawling bereitstellt. Ausgehend von einer Seed-Liste an URLs werden alle URLs, die auf einer Seite vorkommen extrahiert und aufgerufen. Um mit den gecrawlten Seiten später arbeiten zu können, müssen die Ergebnisse gespeichert werden. Das heißt, Inhalt, URL und Metadaten wie Zeitpunkt des Aufrufs müssen in einer Weise in eine Datenbank aufgenommen werden, sodass performant nach Schlüsselwörtern auf den Homepages gesucht werden kann, ohne zu viel Festplattenspeicher zu verwenden. Dieser Prozess nennt sich Indexierung. Apache hat für diesen Zweck die Bibliothek Solr entwickelt, die optimal mit Nutch zusammenarbeitet, die Indexierungsaufgaben übernimmt und per API aus verschiedenen Sprachen, in unserem Fall Java mit SolrJ, abgefragt werden kann. Der Aufbau der Crawler-Architektur war der erste Meilenstein des Projektes, der wiederholte Verbesserungen erforderte, um die Indexierung angemessen schnell und zielgerichtet durchführen zu können. Der Crawler lief seit Mai konstant und konnte in diesem Zeitraum mehr als 50 Millionen Webseiten in einer Solr-Datenbank von 3 TB indexieren.

Die resultierenden Dokumente, meist PDFs, wurden durch eigens entwickelte Parser geschickt und die CO2-Daten nach Scopes und Unternehmen strukturiert ausgelesen und abgespeichert. Zu diesem Zweck haben wir eine Java-Applikation entwickelt. Mit der Bibliothek SolrJ wurde Java mit der Solr-Datenbank verbunden und nach Schlüsselwörtern wie „sustainability report“ oder „climate change report“ mögliche Seiten mit Emissionsdaten abgefragt. Diese Seiten wurden mittels der URL nun neu aufgerufen, denn der Solr-Index speichert nur den rohen Text ohne Metainformationen wie HTML oder PDF, die für den Parser nötig sind. War das Ziel der URL ein PDF, wurde der Text zeilenweise mit PDFBox, der Apache Library zum Lesen von PDF Dokumenten, ausgelesen. Für HTML-Dokumente wurde JSoup verwendet. Die gefundenen Dokumente waren „Echttext“-Dokumente, sodass zum Glück keine Texterkennung mittels OCR notwendig war. Die meisten relevanten Daten waren in Tabellen zu finden, sodass ein Großteil der Arbeit an dem Parser in die Entwicklung eines robusten Tabellenerkennungsfunktion geflossen ist, was insbesondere bei PDF Dokumenten eine anspruchsvolle Aufgabe ist. PDF Tabellen werden gezeichnet und sind nicht als Tabellen markiert. Das heißt eine Tabelle besteht letztlich nur aus Wörtern mit vielen Leerzeichen zwischen ihnen und einigen horizontalen und vertikalen Linien um die Wörter herum. Dabei sieht natürlich die Tabelle jedes Unternehmens verschieden aus, was die Erkennung sehr schwierig macht. Glücklicherweise haben viele der Tabellen eine ähnliche Struktur, insofern sie eine Entwicklung über mehrere Jahre in der Horizontalen und Dimensionen der Daten in der Vertikalen darstellen. Ohne diese grundlegende Struktur wäre ein robuster Parser kaum realisierbar gewesen. Ein zweiter wichtiger Punkt für den Parser war die Standardisierung des Emissions-Reportings in die Klassen direkte, indirekte und Dritt-Emissionen (Scope 1, Scope 2, Scope 3).

Dank dieser beiden Punkte war es möglich, Tabellen auch in PDF-Nachhaltigkeitsberichten mit einer Trefferquote von über 70% auszulesen und zu strukturieren. Ungefähr 20 % der Dokumente konnten von vornherein nicht abgefragt werden, da die Homepage des Unternehmens die automatische Anfrage blockierte. In etwa 50 % aller Nachhaltigkeitsberichte deutscher Unternehmen beinhalteten trotz der eingangs genannten EU-Richtlinien keine verwertbaren Informationen zu ihren Treibhaus-Emissionen. In diesen Fällen konnte auch ein Mensch die entsprechenden Daten nicht auslesen.

Im Laufe der Projektlaufzeit konnten wir die Daten von 300 deutschen und internationalen Unternehmen auslesen, die wir auf www.ethidat.de zur Ansicht und zu Download bereitstellen. Im Durchschnitt konnten wir pro Unternehmen eine Historie von 3 Jahren Emissionen auslesen, genug um erste Entwicklungen darzustellen. Subjektiv können wir feststellen, dass die Emissionen über diesen Zeitraum tendenziell konstant geblieben sind, was bei einem anhaltenden Wachstum letztlich auf eine leichte Verringerung der Emissionen pro Produktionseinheit schließen lässt. Auf Ethidat können diese Ergebnisse selbst eingesehen und entsprechende Schlüsse gezogen werden. Für die Seite haben wir den Mean-Stack (MongoDB, Express, Angular, NodeJS) verwendet. Ein Backend stellt API-Funktionalität für die Kommunikation mit der MongoDB bereit. Über diese API speicherte der Java Parser die in JSON strukturierten Ergebnisse ab und das Frontend greift so diese Daten ab. Wir haben eine Änderungsfunktion in die Webseite integriert, um schnell und unkompliziert in einem Community-Effort zu ermöglichen, fehlerhafte Datenpunkte zu korrigieren oder weitere hinzuzufügen und so die Möglichkeit geschaffen, die automatische Datenextraktion bei Bedarf manuell zu korrigieren. Für jeden Datenpunkt kann mit zwei Klicks eine Änderung angeregt werden mit dem korrigierten Wert und unter Angabe einer Quelle. Über einen solchen Antrag werden wir automatisch per Email benachrichtigt, können die Änderung prüfen und freigeben oder ablehnen.

Mit diesen Ergebnissen wurden alle geplanten Ziele erreicht. Im Rahmen der Projektarbeit konnten wir zusätzlich das Nutzerprofil deutlich schärfen bzw. verschieben. Im Laufe der Zeit und im Rahmen des projektinternen Austauschs und den Coachings zeichnete sich deutlich eine Kernzielgruppe von professionellen Nutzern wie Journalisten, Politikern oder NGOs, die beruflich Daten benötigen, anstatt, wie initial angedacht, den „normalen“ Bürger direkt ansprechen und informieren zu wollen. Deshalb wurde Kontakt mit Freien Journalisten u.A. aus dem Projektumfeld aufgenommen und deren Bedürfnisse direkt in das Portal eingearbeitet. Eine wichtige Erkenntnis war, dass eine fremde graphische Aufarbeitung für Journalisten nur begrenzt hilfreich ist, da jeder seine eigenen gewohnten Tools und persönliches Layout nutzt. Eine weit verbreitete Variante der graphischen Präsentation in der Branche ist www.datawrapper.de, bei der mittels API und CSV-Import Graphiken und Charts erstellt werden können und die unter anderem von der New York Times, dem Spiegel und der Zeit verwendet werden. Dementsprechend haben wir eine Anbindung dorthin bereitgestellt und uns nicht wie zuvor geplant auf die eigene exportierbare Aufbereitung konzentriert.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Daten zur Entwicklung von Emissionsdaten von Unternehmen im Rahmen von Scope 1, 2 & 3 Emissionen zu bekommen wird deutlich vereinfacht. Anstatt die Daten manuell auf den Homepages oder in den Berichten der Unternehmen suchen zu müssen, gibt es nun eine zentrale Anlaufstelle, auf der die Daten graphisch aufbereitet gesammelt präsentiert werden und sie direkt als CSV oder in Datawrapper exportiert werden können. Die Ergebnisse und Methoden können von der Community weiterentwickelt werden. Insbesondere wurde das Portal so konzipiert, dass Nutzer neue Daten hinzufügen oder korrigieren können, belegt durch Quellen und nach einem Review. So kann sich im Wikipedia-Stil durch Crowdsourcing eine umfassende Datenbasis geschaffen werden. Perspektivisch soll die Datenbank nicht nur CO2-Daten beinhalten, sondern auch Daten zu Abfall und Recycling, Wasserverbrauch, Frauenquote oder Zertifizierungen und so zu einer allgemeinen Ethik-Datenbank anwachsen. Persönlich konnten wir unsere Programmierfähigkeiten erweitern. Highlights dabei waren die Extraktion von Tabellen aus PDF-Dateien und die Umsetzung einer durchklickbaren Tour auf der Homepage.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ursprünglich sollte deutlich breiter nach Unternehmensberichten gesucht werden. Nachdem die Größe des Internets klar wurde, mussten die Ausgangspunkte auf bekannte Unternehmenshomepages beschränkt werden, damit Größe und Dauer des Indexierungsprozesses in einem akzeptablen Rahmen blieb.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Auf www.ethidat.de oder unter www.demoweeek.prototypefund.de finden sich alle Ergebnisse und Dokumente.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die initial gemeinsam mit Prototype Fund erstellte „Road-Map“ der Software konnte in allen Meilensteinen erfüllt werden. Daher konnte der geplante Arbeitsaufwand und die geplanten Kosten eingehalten werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Uns sind bisher keine Entwicklungen in der Art von ethidat / Co2-Datenbank von anderen Personen bekannt. Auch von institutionelle Entwicklungen wurden wir nicht beeinflusst. Falls es die hier entwickelte Software institutionell gibt, ist diese bisher nicht einer breiten Öffentlichkeit zugänglich gemacht worden.

ReactiveRuntime

Schlussbericht

Zuwendungsempfänger:

Simon Danisch

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS19S27** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Veröffentlichungen und Reproduzierbarkeit ist ein Grundstein der wissenschaftlichen Methodik. Zur Bereitstellung von Daten, Analysen und Code erfreuen sich [Jupyter Notebooks](#) steigender Beliebtheit. Notebooks sind interaktive Dokumente, in denen man Code entwickeln kann und die Resultate als multi-media Output angezeigt bekommt. Die Veröffentlichung dieser Notebooks und die Interaktivität der generierten Grafiken gestaltet sich jedoch als Anspruchsvoll.

Bei diesem Projekt, wird die Kalkulation ähnlich wie bei Spreadsheets (z. B. Excel) durch einen Graphen dargestellt, der die Abhängigkeiten zwischen den benutzten Variablen beschreibt.

Dies ermöglicht die effiziente, automatische Ausführung des gesamten Notebooks, z.B. wenn sich Inputs verändern. Zusammen mit interaktiven Widgets wie Slidern, kann man so spielerisch komplexe Daten und wissenschaftliche Modelle erforschen.

Eine Hauptinnovation dieses Projekts ist es, dass man durch das Erfassen des gesamten Ausführungsgraph alle Berechnungen speichern kann - das heißt, egal ob die Berechnungen auf einem Supercomputer gelaufen sind, oder auf einem Laptop mit einem Komplizierten Setup, ist es möglich die Resultate interaktiv im Internet einem breiten Publikum vorzustellen.

Meilensteine:

1. Verbesserung der Stabilität, Performance und Dokumentation des aktuellen Prototypen
2. Verbesserung der Integration von Javascript und Makie¹ für bessere Visualisierungen

¹ <https://github.com/JuliaPlots/Makie.jl>

3. Entwicklung der Zwischenspeicherung und Vorberechnung von Ergebnissen mittels des Ausführungsgraphes. Dadurch wird Interaktivität im Browser ohne laufenden Julia Prozess ermöglicht
4. Wrappen der Bibliothek für die Verwendung von Python, Release von Version 1 als Python Bibliothek
5. Erste Veröffentlichungen von interaktiven Modellen mit den wissenschaftlichen Partnern

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

Die klassischen Zielgruppen von diesem Projekt sind Organisationen, die wissenschaftliche Artikel und Modelle veröffentlichen. Diese sind zum Beispiel das Umweltbundesamt, NASA, National Center for Environmental Information und Universitäten.

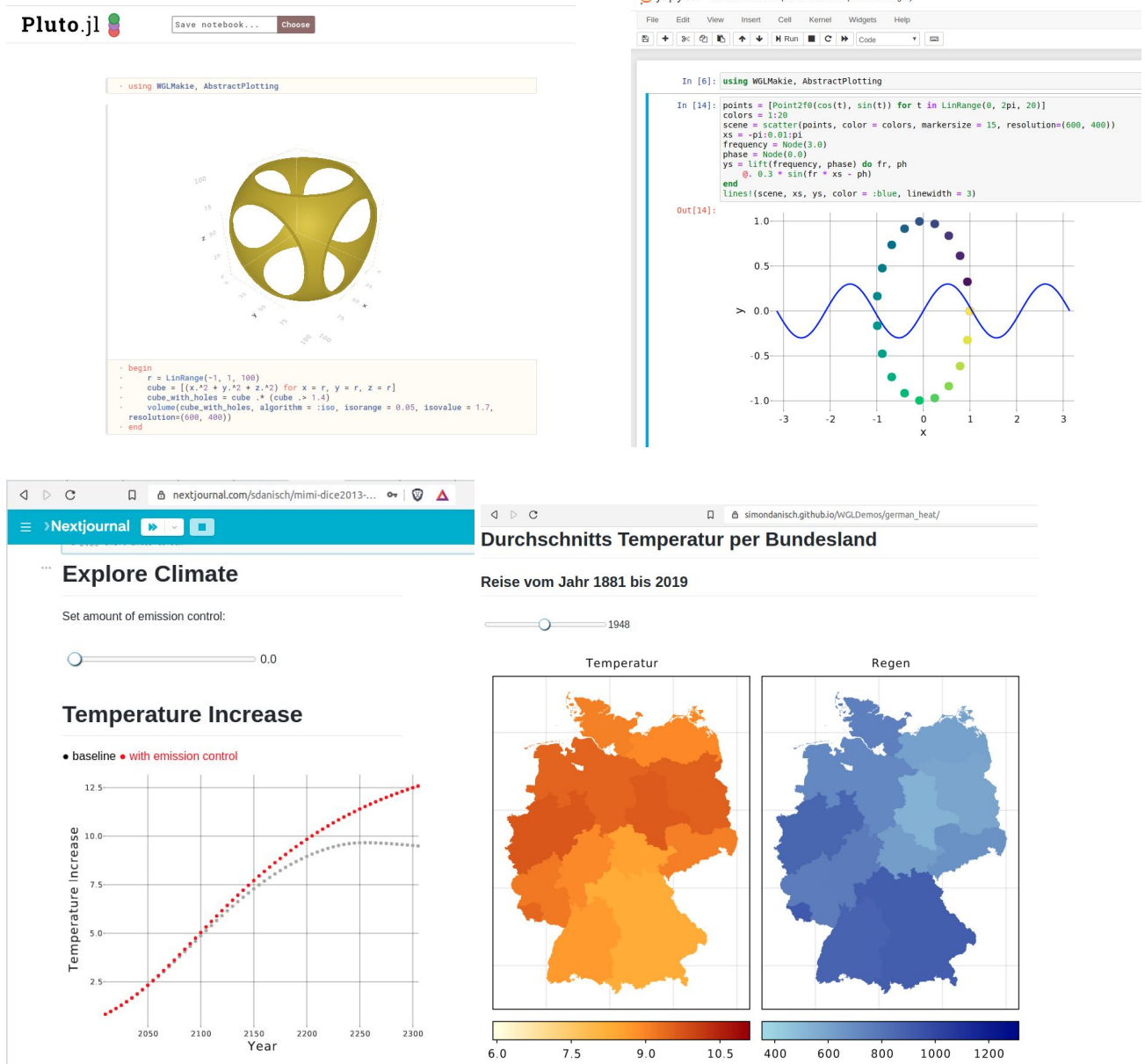
Eine einfache Veröffentlichung interaktiver, reproduzierbarer Grafiken hilft bei der Kommunikation der Wissenschaft. Wenn jeder Leser direkt mit den Modellen und Daten interagieren kann, hilft dies Vertrauen zu schaffen, da die Leser sicher sein können, dass Ihnen nicht nur handverlesene Parameter des Modells präsentiert werden, um eine bestimmte narrative zu gestalten.

Wenn interaktive Veröffentlichung so zugänglich gemacht werden, dass selbst einzelne Forscher mit wenig Software Entwicklungserfahrung ihre Daten und Modelle veröffentlichen können, hilft dies auf breiter Basis das Internet verständlicher zu gestalten und lässt die Menschen mehr am wissenschaftlichen Prozess teilhaben - was wiederum hilft besser demokratisch Entscheidungen zu fällen.

Diesen Beitrag leistet ReactiveRuntime mit Makie, da selbst Studenten ohne ein großes Team mit extra Web-Entwicklern ihre Forschung interaktiv im Internet präsentieren können.

Ausführliche Darstellung der Ergebnisse

Makie funktioniert dank des PrototypeFunds nun 100 % interaktiv in Jupyter², Pluto.jl³, Nextjournal⁴ und im Browser⁵:



Dies sind die meistgenutzten Plattformen für Code Explorationen. Mit ReactiveRuntime können diese Plattformen nun voll genutzt werden, und es wurde sehr einfach gemacht, von dort aus die interaktiven Visualisierungen zu publizieren.

² <https://jupyter.org/>

³ <https://github.com/fonsp/Pluto.jl>

⁴ <https://nextjournal.com/sdanisch/mimi-dice2013-climate-impact/>

⁵ https://simondanisch.github.io/WGLDemos/german_heat/

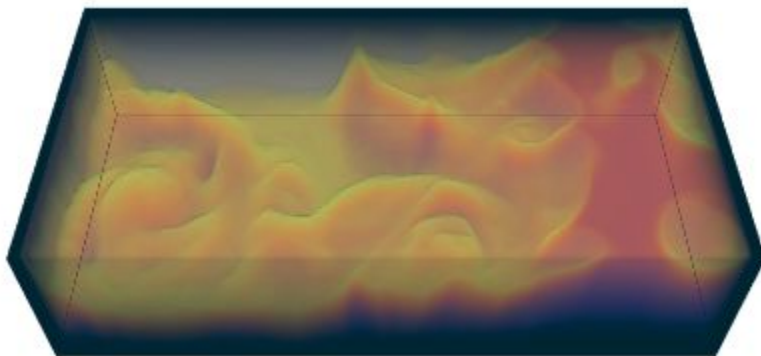
Pluto.jl ist eine sehr neue Julia Notebook Plattform, die es am Anfang meines Projektes noch nicht gab. Sie implementiert perfekt die Spreadsheet artige Notebook Ausführungsweise, die ich mir für ReactiveRuntime vorgestellt hatte. Dies hat Makie.jl mehr in den Mittelpunkt des Projektes gerückt und ich habe mehr Zeit investiert, dass Makie.jl besser im Web funktioniert und gut getestet und dokumentiert ist.

Der Rest ist in die Implementation des “offline-exports” geflossen. Dieser ermöglicht es ohne Laufenden Julia Prozess, auf jedem Webserver interaktive Datenanalysen/Grafiken einzubinden. Dies wird möglich gemacht, indem mithilfe des Ausführungsgraph alle möglichen Berechnungen gespeichert werden und so im Web verfügbar gemacht werden können.

Das Hauptergebnis dieses Projekts sind jedoch mehrere Veröffentlichungen mit den wissenschaftlichen Partnern!

CliMA⁶

CliMA (Clima Modeling Alliance) ist ein großes Projekt von dem CALTECH (California Institute of Technology) um das Klima zu modellieren und so gut wie möglich Strategien für die Bekämpfung der Klimakrise zu entwickeln. Im Moment arbeite ich mit dem Unterprojekt Oceananigans⁷ zusammen, um Meeres Simulationen interaktiv zu visualisieren⁸. Da alle Projekte der CliMA Organisation Julia benutzen, ist es zu erwarten, dass die Anwendung von Makie noch um einiges steigt.



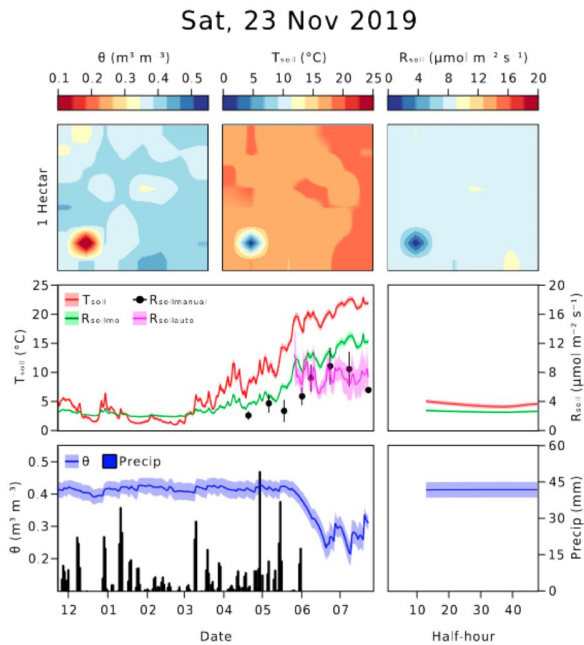
⁶ <https://clima.caltech.edu/>

⁷ <https://github.com/CliMA/Oceananigans.jl/>

⁸ <https://simondanisch.github.io/WGLDemos/oceananigans/>

Argonne National Laboratory⁹

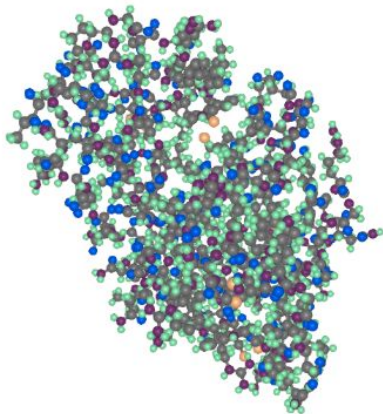
Benutzt Makie für eine interaktive Datenexploration. Dargestellt ist ein Datensatz bei dem die Feuchtigkeit, Temperatur und Niederschläge in heterogenem Grasland aufgezeichnet wurden. Es wird außerdem geplant, eine Webseite zu erstellen, um noch mehr Datensätze zu präsentieren und zu erklären.



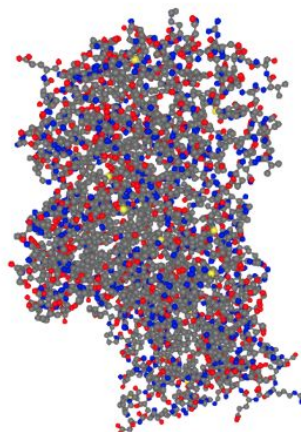
Iowa State University¹⁰

Benutzt Makie für interaktive Molekül Simulationen und Explorationen:

2VB1

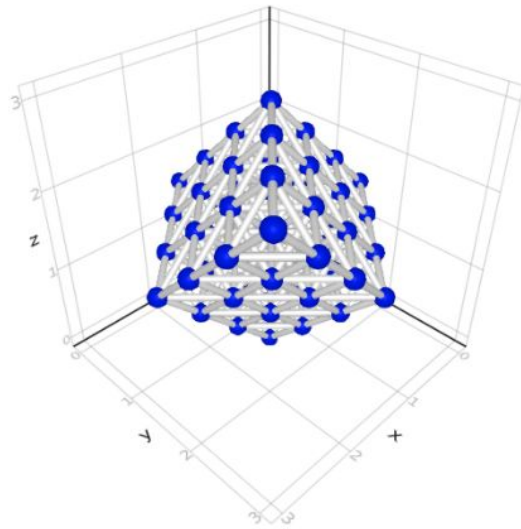
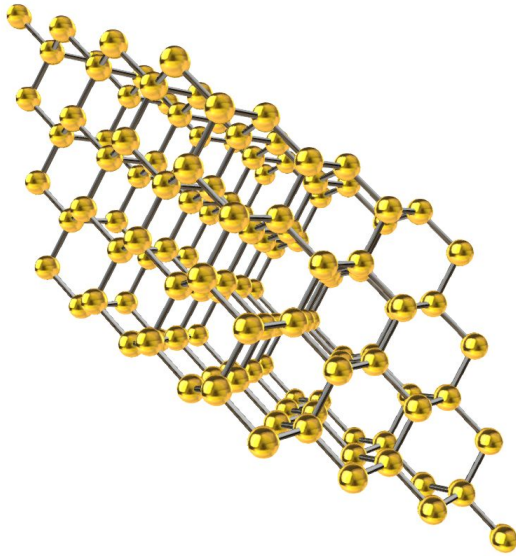


1LW3



⁹ <https://simondanisch.github.io/WGLDemos/soil/>

¹⁰ <https://github.com/kool7d/BioMakie.jl>



Von den Meilensteinen wurde hauptsächlich der Python-Wrapper ausgelassen, da die Zeit knapp geworden ist. Aber im Prinzip ist die Julia Python Interaktion ausgereift genug, dass jeder ohne großen Aufwand die entstandene Software von Python aus benutzen kann.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Wie schon in den Ergebnissen erwähnt, benutzen viele Gruppen schon Makie und ReactiveRuntime für ihre Veröffentlichungen. Es gibt auch mehrere interessante Projekte, die gerade noch in der Umsetzungsphase sind und erst bald veröffentlicht werden.

Man kann also sagen, dass die Makie Community sehr aktiv ist. Es gibt mehrere Open Source Mitarbeiter die wöchentlich Verbesserungen beitragen und ich habe auch viele Pläne für die Weiterentwicklung.

Eines der großen Ziele, dem Makie schon sehr nah ist, ist es reibungslos zwischen Browser, Desktop und PDF Publikationen zu wechseln und in diesen sehr unterschiedlichen Medien, die gleichen APIs und Benutzer Erlebnisse zu Verfügung zu stellen. Es soll also möglich sein, interaktive Visualisierungen genauso gut als Desktop Applikationen zu veröffentlichen, wie auf einer Webseite, oder eben auch direkt in einer klassischen wissenschaftlichen Veröffentlichung in einem Journal (PDF). Es gibt noch immer kleine visuelle Unterschiede in Makie zwischen Desktop

¹¹ <https://github.com/ffreyer/LatPhysPlottingMakie.jl>

¹² <https://github.com/janattig/LatticePhysics.jl>

(OpenGL^{13 14}), Web (JS + WebGL^{15 16}), PDF/SVG (Cairo^{17 18}), die ich in Zukunft eliminieren möchte.

Die nächste Baustelle ist das effiziente speichern und übertragen von den Berechnungen. Die kombinatorische Explosion ist groß, und somit fallen schnell große Datenmengen an. Es gibt zwei sehr vielversprechende Ansätze, die ich ausprobieren möchte:

- Kompilation von Julia nach Javascript, so, dass viele Berechnungen auch im Browser ausgeführt werden können und nicht gespeichert werden müssen.
- Streamen und komprimieren der Zustände. Im Moment werden alle benötigten Daten auf einmal zum Browser geschickt, was die Wartezeit in die Höhe treibt. Das gute ist jedoch, dass Streaming perfekt funktioniert, da die Zustände schon unabhängig gespeichert werden und somit leicht in einzelnen Paketen an den Browser geschickt werden können.

Außerdem möchte ich noch weiter an Pluto.jl arbeiten und einer performanten Makie Integration. Die aktuelle Integration basiert auf der Ursprünglichen Jupyter Integration, welche noch nicht ganz ausgereift ist, im Vergleich zu dem was möglich wäre.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Am Anfang der Planung bin ich noch davon ausgegangen spezifische Technologien wie CBOR¹⁹ oder ObservablesHQ²⁰ für die implementation der Kommunikation zwischen Julia und Javascript und für das Reaktive verhalten zu benutzen. Dies sind jedoch Implementations Details, die nichts wirklich in der groben Planung zu suchen hatten. Es kam dann relativ schnell raus, das es simplere alternativen gibt um Julia und Javascript zu kommunizieren zu lassen. Pluto.jl hat außerdem alle wünsche erfüllt, die ich mir von ObservablesQH versprochen hatte.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

- [Twitter](#)
- <https://github.com/JuliaPlots/Makie.jl>
- <https://github.com/SimonDanisch/JSServe.jl>
- [Julia Slack #makie channel](#)

¹³ <https://www.opengl.org/>

¹⁴ <https://github.com/JuliaPlots/GLMakie.jl>

¹⁵ <https://github.com/JuliaPlots/WGLMakie.jl>

¹⁶ <https://de.wikipedia.org/wiki/WebGL>

¹⁷ <https://github.com/JuliaPlots/CairoMakie.jl>

¹⁸ https://de.wikipedia.org/wiki/Cairo_%28Grafikbibliothek%29

¹⁹ <https://cbor.io/>

²⁰ <https://observablehq.com/>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ich habe ein Projekt mit einer Firma die Makie & JSServe²¹ benutzt angefangen, welches mir gegen Ende des PrototypeFunds ein wenig die Zeit geraubt hat die letzten Details zu implementieren, wie zum Beispiel die Python Integration. Deswegen habe ich auch nicht den letzten Zeitraum voll abgerechnet. Die guten Neuigkeiten sind, dass ich weiterhin bei der Firma bleibe und Makie & ReactiveRuntime weiter Open Source entwickeln werde!

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Pluto.jl kam aus dem nichts und hat eine unerwartet gute reaktive Alternative zu Jupyter geboten. Dies hat den Fokus von reaktiven Notebooks auf Geschwindigkeitsverbesserung, Stabilität, und Publikationen im Web gelenkt.

²¹ <https://github.com/SimonDanisch/JSServe.jl>

Richtlinie zum „Software-Sprint“

Real_MHD_App – Vertrauen in Lebensmittel über das MHD hinaus

Schlussbericht

Zuwendungsempfänger und Autor:

Matthias Ansorg

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S28 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Diese Veröffentlichung wird vom Autor unter einer Creative Commons Attribution 4.0 International Lizenz zur Verfügung gestellt.

Kurze Darstellung der Aufgabenstellung und Motivation

Hintergrund des Vorhabens ist die Lebensmittel-Verschwendung in Haushalten, hier insbesondere in industriell geprägten Gesellschaften. In der Bundesrepublik Deutschland etwa werden jährlich 6 Mio. t Lebensmittel in Haushalten weggeworfen, was mehr als der Hälfte aller Lebensmittelabfälle entspricht.¹ Lebensmittel-Verschwendung schadet der Umwelt, dem Klima und der Ernährungssicherheit und ist daher ein gesamtgesellschaftliches Problem. Lebensmittelverschwendung im weitesten Sinn trägt weltweit mindestens 20% zu den jährlichen Treibhausgasemissionen bei.²

Ein wesentlicher Aspekt der Lebensmittel-Verschwendung in Haushalten ist das Mindesthaltbarkeitsdatum (MHD). Wird es überschritten, werfen viele aus Unsicherheit auch unverdorbene Lebensmittel weg. Die Idee dieses Vorhabens ist, Nutzenden so komfortabel wie möglich detaillierte Informationen anzubieten, mit denen sie die Essbarkeit eines Lebensmittels selbständig bewerten können. Dazu sollte eine Mobilanwendung entwickelt werden, die nach dem Einlesen von Strichcodes auf Produktverpackungen die relevanten Informationen anzeigt.

Die ursprünglich geplante Vorgehensweise war wie folgt. Zu Beginn sollten die Anforderungen an die zu entwickelnde Mobilanwendung präzisiert und definiert werden (Meilenstein M1) und eine technische Schnittstelle zur Datenquelle „Open Food Facts“ vereinbart werden (M2). Die Anwendung sollte dann basierend auf der offiziellen Mobilanwendung von Open Food Facts umgesetzt werden (M3) während parallel Inhalte zur Lebensmittel-Haltbarkeit entwickelt werden (M4). In der letzten Phase

1 Zahlen von 2015 gemäß einer Studie der Universität Stuttgart. Siehe u.a.:

<https://www.zeit.de/wissen/2019-05/lebensmittelverschwendung-haushalte-essen-muell-deutschland>

2 Zahlen entspr. eigener Berechnungen im Rahmen der Prototype Fund Demo Week zur Vorstellung des Projektes. Details unter <https://demoweek.prototypefund.de/projects/10-food-rescue-app.html>

sollte dann die entwickelte Funktionalität in die offizielle Android-Anwendung von Open Food Facts integriert werden (M5) und entsprechend auch die Inhalt in deren Datenbank (M6). Optional war zum Projektabschluss noch eine Umsetzung für iOS geplant (M7).

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

Dieses Vorhaben realisiert Ziele der Förderinitiative „Software-Sprint“ insbesondere in zwei Bereichen. Einerseits im Schwerpunkt „Data Literacy“, denn die Mobilanwendung hilft Nutzenden bei der Interpretation der Angaben auf Lebensmitteln mit Bezug zur Haltbarkeit (Mindesthaltbarkeitsdatum, Verbrauchsdatum, Konservierungsstoffe, Art der Verarbeitung usw.). Andererseits geht es bei diesem Vorhaben um das Schwerpunktthema von Runde 7 des Software-Sprints: Vertrauen bauen. Denn das Ergebnis ist eine Software-Anwendung, die Vertrauen in Lebensmittel und in den eigenen Umgang mit Lebensmitteln stärkt.

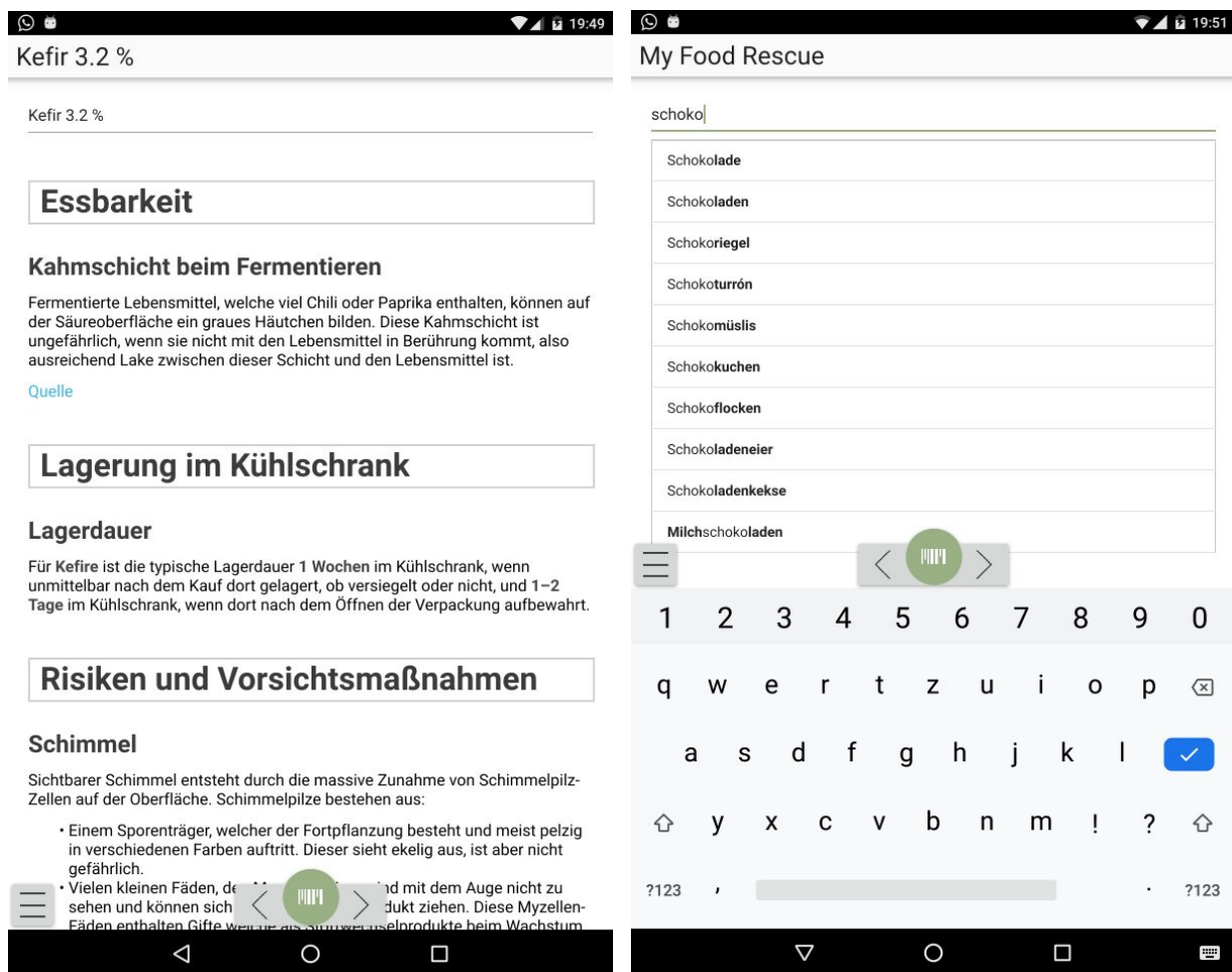
Ausführliche Darstellung der Ergebnisse

Hauptergebnis des Vorhabens ist eine Mobilanwendung für Android, die kostenfrei im Google Play Store unter dem Namen „My Food Rescue“ zur Verfügung steht³ und so weltweit einsetzbar ist:



3 URL des Eintrags im Google Play Store: <https://play.google.com/store/apps/details?id=org.fairdirect.foodrescue> . Dies ist die schnellste und komfortabelste Möglichkeit für Lesende, die entwickelte Lösung selbst zu testen.

Die Hauptfunktion und Hauptinnovation dieser Anwendung ist die Kombination eines Strichcode-Lesers mit Informationen zur Haltbarkeit und Essbarkeit von Lebensmitteln. Wenn Nutzende mit Hilfe der Kamera ihres mobilen Endgerätes den Strichcode auf einer Lebensmittel-Verpackung einlesen, zeigt die Anwendung alles an, was in der Datenbank zur Haltbarkeit und Bewertung dieses Lebensmittels hinterlegt ist. Diese Funktion basiert auf Zuordnungen von Produkten zu Kategorien und von Kategorien zu den hinterlegten Inhalten. Die relevanten Informationen zur „Rettung“ eines Lebensmittels sind so üblicherweise innerhalb von 2 Sekunden verfügbar, gemessen ab Klick auf den Knopf „Strichcode einlesen“. Bei ungünstigen Lichtverhältnissen kann das Einlesen eines Strichcodes auch einige Sekunden länger benötigen. Der praktische Nutzwert dieser Anwendung ist in Deutschland leider durch die Unvollständigkeit der genutzten Datenbasis eingeschränkt; dies wird im Abschnitt „Zielgruppe, Nutzen und mögliche Weiterentwicklungen“ näher erläutert.



Für Fälle, in denen es keine Treffer für einen Strichcode in der Datenbasis gibt, können Nutzende die relevanten Informationen auch durch Eingabe einer Lebensmittelkategorie nachschlagen. Dazu stehen bereits 3900 deutsche und 8100 englische Kategorienamen zur Verfügung, und Nutzende werden bei der Eingabe durch automatische Vervollständigung unterstützt.

Im Vergleich mit allen bisherigen Alternativen beschleunigt und vereinfacht diese Anwendung den Zugriff auf die angebotenen Informationen erheblich. Erklärtes Ziel davon ist, die Bereitschaft von Nutzenden zu erhöhen, etwas zur Rettung von Lebensmitteln zu unternehmen; denn der erste Schritt ist nun nahezu mühelos.

Im Vorhaben wurde besonderes Augenmerk auf eine Software-Architektur gelegt, die eine breite und globale Nutzung der Anwendung möglich macht. Konkret ermöglicht die gewählte Architektur die folgenden Funktionen und Anwendungsfälle:

- **Offline-Nutzung.** Die Anwendung enthält alle Daten in einer SQLite3-Datenbank, die auf dem Endgerät installiert wird. Sie ist daher an jedem Ort und unabhängig vom Internet nutzbar, sei es in einem Supermarkt ohne nutzbares Mobilfunksignal oder in einem dünn besiedelten Gebiet irgendwo auf der Welt. Aus der Offline-Nutzung ergibt sich auch ein hervorragender Datenschutz – die aktuelle Version 0.3 der Anwendung kommuniziert im installierten Zustand absolut nichts über das Internet.

Zur Zeit enthält die mitgelieferte Datenbank 448 224 Lebensmittelprodukte, 9818 Lebensmittelkategorien, 674 021 Zuordnungen von Kategorien zu Produkten und 882 Artikel zur Lebensmittel-Rettung. Das beinhaltet alle relevanten Inhalte aus dem Open-Data-Projekt Open Food Facts, das hier als Datenquelle dient. Die Downloadgröße der Datenbank beträgt trotzdem nur 10,6 MiB – etwa so viel wie 3-4 Fotos.

- **Mehrsprachiger Gebrauch.** Benutzeroberfläche und alle Inhalte der Anwendung sind auf Deutsch und Englisch verfügbar. Weitere Übersetzungen können hinzugefügt werden, ohne dass Anpassungen am Quellcode notwendig sind.
- **Mobile und Desktop-Nutzung.** Die Anwendung ist gleichzeitig sowohl echte Desktop-Anwendung als auch echte Mobilanwendung. Ermöglicht wird dies durch das quelloffene Framework KDE Kirigami,⁴ das wiederum auf dem plattformübergreifenden Framework Qt basiert.
- **Plattformübergreifender Einsatz.** Bisher wurde die Anwendung sowohl unter Android als auch unter Linux eingesetzt. Es werden aber prinzipiell alle Plattformen unterstützt, auf denen Qt verfügbar ist. Dazu gehören die offiziell unterstützten Plattformen Linux, macOS, Windows, Android, iOS und UWP („Windows Mobile“). Darüber hinaus ist es ohne größeren Aufwand möglich, auch die kleineren mobilen Plattformen Ubuntu Touch, Sailfish OS und LineageOS zu unterstützen.
- **Einsatz auf alter oder langsamer Hardware.** Die Anwendung ist so konzipiert, dass sie auch auf ressourcenschwacher Hardware verwendet werden kann. Dies können ältere Geräte sein oder auch besonders preisgünstige Android-Geräte, wie sie etwa in Indien verbreitet sind. Ermöglicht wird dies durch kompilierten C++-Code. Die resultierende Ausführungsgeschwindigkeit ist insbesondere für den integrierten Strichcode-Leser relevant.

Bei der gewählten Architektur mussten zwei Kompromisse eingegangen werden. So enthält die Anwendung aufgrund von Qt QML auch etwas JavaScript-Bytecode und eine zugehörige virtuelle Maschine mit entsprechendem Speicherverbrauch und geringerer Performance. Dieser Kompromiss besteht bis zum Erscheinen von Qt 6; ab dann wird es möglich sein, QML zu echtem C++-Code umzuwandeln.⁵ Ein weiterer Kompromiss musste bei der Installationsgröße auf Plattformen ohne Paketverwaltung gemacht werden. Hier müssen die Qt-Bibliotheken bei der Verteilung der Anwendung stets mitgeliefert werden, was die installierte Größe des Binärcodes von ca. 400 KiB (unter Linux) auf ca. 37 MiB (unter Android) erhöht. Zukünftig erscheint eine Größe von ca. 10 MiB realistisch, unter anderem durch Einsatz des quelloffenen Binärcode-Kompressors UPX.⁶

4 Projekt-Website: <https://kde.org/products/kirigami/>

5 Quellennachweis: <https://www.qt.io/blog/2019/08/07/technical-vision-qt-6>

6 Projekt-Website: <https://upx.github.io/>

Auf umfangreiche Dokumentation der Anwendung und des Quellcodes für zukünftige Entwickler wurde Wert gelegt. Alle nicht offensichtlichen Eigenschaften des Quellcodes wurden dokumentiert, so dass die Anwendung als Basis für weitere Entwicklungen gut geeignet sein sollte.

Zusammenfassend: Ergebnis des Vorhabens ist eine solide technische Basis für Anwendungen, die auf Basis von Strichcodes Auskunft über Lebensmittel oder andere Produkte geben. Die realisierte Anwendung besitzt zwei Alleinstellungsmerkmale: (1) einen Strichcode-Scanners, anders als alle anderen Anwendungen zur Lebensmittel-Rettung; und (2) Offline-Funktionalität, anders als alle anderen Anwendungen für Produktinformationen auf Basis von Strichcodes. Vor einem breiten Einsatz in Deutschland sind weitere Arbeiten zur Vervollständigung der Datenbasis notwendig.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Zielgruppe der entwickelten Anwendung umfasst alle, die ein mobiles Endgerät auch bei der Zubereitung von Lebensmitteln einsetzen würden. Aktuell müssen Nutzende außerdem Deutsch oder Englisch zumindest als Zweitsprache beherrschen – immerhin etwa 1,5 Milliarden Menschen weltweit. Weitere Übersetzungen sollen folgen.

In Tests ergab sich ein moderater Nutzen der Anwendung im Alltagseinsatz, der leider hinter den Erwartungen der Nutzenden zurückblieb. Bemängelt wurde, dass nach Einlesen des Strichcodes bisher nur für etwa jedes dritte bis vierte typische Supermarkt-Produkt in Deutschland relevante Informationen angezeigt werden. Als weitere Nutzungsmöglichkeit steht für diese Fälle die Suche nach Lebensmittel-Kategorien zur Verfügung. Die angezeigten Inhalte erfüllten jedoch zu oft noch nicht die Erwartung, möglichst spezifisch zum jeweiligen Lebensmittel zu informieren.

Bisher demonstriert die entwickelte Anwendung also erfolgreich aber prototypisch eine technische Lösung, um Informationen zur Vermeidung von Lebensmittel-Verschwendung komfortabel und schnell via Strichcodes verfügbar zu machen. Für einen breiten Einsatz in Deutschland sind jedoch noch umfangreiche Arbeiten zur Vervollständigung der Datenbasis notwendig. Dies betrifft sowohl die Texte zur Haltbarkeit und Rettung von Lebensmitteln als auch die Datensätze im Projekt Open Food Facts. Die Open Food Facts Datenbasis kennt bisher 63 400 in Deutschland verkaufte Lebensmittel, davon besitzen 41 700 die hier benötigten Kategoriedaten.⁷ In Frankreich ist die Datenbasis bereits umfangreich genug für einen breiten Einsatz: Open Food Facts kennt 719 000 in Frankreich verkaufte Lebensmittel, darunter 312 000 mit Kategoriedaten.⁸ Es zeigt sich deutlich, dass Open Food Facts ein Projekt mit Ursprung in Frankreich ist.

Zur weiteren Entwicklung ist daher zunächst geplant, die Anwendung auch auf Französisch verfügbar zu machen um so Erfahrungen im breiten Einsatz sammeln zu können. Es ist zu erwarten, dass sich die Produktdaten-Situation für Deutschland durch die Arbeit der Freiwilligen im Projekt Open Food Facts fortlaufend verbessert. Im Rahmen der Fortentwicklung dieser Lösung soll aber auch bezahlte Arbeit zur Unterstützung von Open Food Facts eingesetzt werden, insbesondere um die 21 700 deutschen Produkte ohne Kategoriedaten zu vervollständigen; dies entspricht ca. 4,5 Monaten Vollzeit-Arbeit.

7 Quelle sind folgende Abfragen an die Open Food Facts Webanwendung:

<https://world.openfoodfacts.org/country/germany>

<https://world.openfoodfacts.org/state/categories-to-be-completed/country/germany>

8 Quelle sind folgende Abfragen an die Open Food Facts Webanwendung:

<https://world.openfoodfacts.org/country/france>

<https://world.openfoodfacts.org/state/categories-to-be-completed/country/france>

Der Plan zur weiteren Entwicklung sieht vor, diese Arbeiten zusammen mit Erweiterungen der Funktionalität und anschließender Öffentlichkeitsarbeit auf Basis einer Anschlussförderung umzusetzen. Aktuell wird dazu u.a. eine Bewerbung für den Innovation Accelerator des Welternährungsprogramms UN WFP vorbereitet. Sobald die Anwendung für den breiten Einsatz geeignet ist, kommen nachhaltigere Finanzierungsmodelle in Betracht.

Aus der Veröffentlichung des Quellcodes der Anwendung ergab sich als weiterer Nutzen eine deutlich verbesserte Dokumentation⁹ für Kirigami, was hoffentlich zur weiteren Verbreitung dieses vielversprechenden Frameworks mit europäischem Ursprung beiträgt. Kirigami ist ein junges, quelloffenes Framework des KDE-Projektes; bisher erst acht in Google Play veröffentlichte Anwendungen verwenden Kirigami. Die Einarbeitung war durch fehlende Anleitungen problematisch, jedoch erwies sich das Framework dann als hervorragend geeignet für plattformübergreifende Mobilanwendungen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Nutzung der Mindesthaltbarkeitsdaten in Open Food Facts. Zu Beginn des Vorhabens wurden die in der Open Food Facts Datenbasis enthaltenen Mindesthaltbarkeitsdaten analysiert, um ggf. auch auf dieser Grundlage abgelaufene Lebensmittel bewerten zu können. Diese Daten waren bei ca. 113 000 Produkten (7,6% aller Produkte) vorhanden, jedoch mit gravierenden Datenqualitäts-Problemen. Von einer Nutzung wurde schließlich aufgrund einer Kosten-Nutzen-Abwägung abgesehen.

Nutzung der Open Food Facts Android-Anwendung als Basis. Die offizielle Android-Mobilanwendung von Open Food Facts¹⁰ sollte ursprünglich als Basis-Software in diesem Vorhaben dienen. Einige explorative Arbeiten auf Basis dieser Software wurden unternommen. Im April 2020 begann jedoch bei Open Food Facts die Entwicklung an der nächsten Generation der offiziellen Open Food Facts Mobilanwendung, basierend auf Google Flutter.¹¹ Im Sinne einer langen Nutzbarkeit der geförderten Lösung erschien es nicht sinnvoll, die Lösung als Erweiterung einer Mobilanwendung umzusetzen, die in absehbarer Zukunft ersetzt und nicht mehr weiterentwickelt wird. Es wurde daher eine unabhängige Anwendung geschaffen, die lediglich Daten von Open Food Facts verwendet. Weiterer ausschlaggebender Grund für diese Entscheidung war, dass eine eigenständige Anwendung unabhängig von einer Internetverbindung funktionieren kann, etwa in einem Supermarkt ohne Mobilfunknetzabdeckung.

Spendenmöglichkeit für überschüssige Lebensmittel. Während der Umsetzung des Vorhabens kam die Idee auf, innerhalb der entwickelten Mobilanwendung auch eine Spendenmöglichkeit für überschüssige Lebensmittel anzubieten und damit eine weitere Möglichkeit zur Vermeidung von Lebensmittelabfällen in Haushalten zu schaffen. Es wurde dazu versucht, eine existierende quelloffene Software (Open Food Network) als Online-Plattform zum Anbieten dieser Spenden einzusetzen. Diese Software erwies sich jedoch als ungeeignet, so dass eine Spendenmöglichkeit im Rahmen des Vorhabens nicht realisiert wurde. Ein separates Team entwickelte die Idee unter dem Namen „OpenFood-Bank“ jedoch im Rahmen des #WirVsVirus-Hackathons der Bundesregierung¹² weiter und gelangte als eines von 147 Projekten ins Finale.¹³ Eine Spendenmöglichkeit kann nun in der Zukunft durch API-Anbindung an diese Plattform realisiert werden.

9 Siehe README des Projektes, Abschnitt „Development Guide“: <https://github.com/fairdirect/foodrescue-app#readme>

10 Bisher die offizielle Android-Anwendung von Open Food Facts, mit Quellcode unter <https://github.com/openfoodfacts/openfoodfacts-androidapp>

11 Link zum Projekt und Quellcode: <https://github.com/openfoodfacts/smooth-app>

12 Vgl. den Artikel „Finale des #WirVsVirus Hackathons – Digitaler Lichtblick in der Krise“ unter <https://www.bundesregierung.de/-1792462>

13 Vgl. die Liste der Finalisten unter <https://wirvsvirus.org/finale/>

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

- **Google Play Store:** Um die entwickelte Mobilanwendung zu installieren und zu testen.
<https://play.google.com/store/apps/details?id=org.fairdirect.foodrescue>
- **Projekt-Website:**
<https://fairdirect.org/realmhd-app>
- **Artikel mit Präsentation der Ergebnisse und weiteren Pläne:**
<https://demoweek.prototypefund.de/projects/10-food-rescue-app.html>
- **Quellcode und Dokumentation für Entwickelnde:**
<https://github.com/fairdirect/foodrescue-app> und
<https://github.com/fairdirect/foodrescue-content>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Alle geplanten Meilensteine mit Ausnahme des in der Antragstellung als optional markierten Meilenstein 7 „Umsetzung für iOS“ konnten im Rahmen des geförderten Vorhabens erreicht werden. Meilenstein 7 wurde zu etwa 80% erreicht, da die Anwendung durch die Wahl von KDE Kirigami als Framework bereits plattformübergreifend funktioniert. Zur Fertigstellung der iOS-Version sind jedoch noch ca. 5 Tage Entwicklungsarbeit für Optimierungen, Tests und Veröffentlichung erforderlich.

Mehraufwand ergab sich durch Arbeiten, die zu keiner Lösung geführt haben (siehe oben) und durch folgende weitere Punkte:

- Fehlende Dokumentation zum eingesetzten Framework KDE Kirigami. Für Details vergleiche den Abschnitt „Zielgruppe, Nutzen und mögliche Weiterentwicklungen“.
- Ambitionierte Ausgestaltung von Meilenstein 3 „Umsetzung der Software“, wo zusätzlich zu grundlegenden Funktionen auch bereits die Internationalisierung der Anwendung umgesetzt werden konnte. Alle Funktionen und Inhalte stehen damit neben Deutsch auch auf Englisch zur Verfügung, was eine 10-15 mal größere potentielle Nutzerbasis bedeutet.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während des Vorhabens sind keine Ergebnisse oder Planungen anderer Stellen bekannt geworden, die sich mit der Kerninnovation des Vorhabens überschneiden; das heißt mit der Idee, Informationen zur Vermeidung von Lebensmittelabfällen auf Basis von Strichcodes bereitzustellen.

Allerdings ergaben sich im Projekt Open Food Facts während des Vorhabens Änderungen, die eine Anpassung der Projektplanung erforderlich machten. Details wurden im Abschnitt zu „Arbeiten, die zu keiner Lösung geführt haben“ dargestellt.

30. Sep. 2020

Eingangsnr.: **4972/183**

Sina Beckstein & Julian Rösner GbR
Weichselstraße 56
12045 Berlin

Richtlinie zum „Software-Sprint“

01IS19S29 – Identity-Stick

Schlussbericht

Zuwendungsempfänger: Sina Beckstein & Julian Rösner GbR

Name des Zuwendungsempfängers: Sina Beckstein, Julian Rösner

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS19S29** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Motivation

Wir haben von August bis Oktober 2019 an Tech4Germany teilgenommen, einem Programm zur Verwaltungsdigitalisierung unter der Schirmherrschaft des Bundeskanzleramts. Gemeinsam mit den Unternehmen Governikus und Yes wurde ein Prototyp zur Online-Authentifizierung mittels eines Sparkassen-Logins in ein Nutzerkonto implementiert. Dieser Prototyp zeigte zum ersten Mal die Möglichkeit, sich nicht nur mit der kaum genutzten eID für öffentliche Leistungen identifizieren zu können, sondern mit nutzerfreundlicheren Möglichkeiten. Während dieses Projekts entstand die Idee, eine bessere Authentifizierungs- und Identifizierungslösung zu entwickeln, die insbesondere den Login für private und öffentliche Dienste vereinen und erleichtern soll. Eine solche ganzheitliche Lösung existiert bislang nicht.

Problemstellung

Eine Identität kann im Analogen nur ein Staat vergeben. Im Digitalen sind private Unternehmen zentral und nehmen diese Aufgabe durch ihre Login-Services wahr. Die Privatisierung der Identität birgt aber gesellschaftliche Risiken: Normen wie Privacy und Sicherheit treten bei profitorientierten Modellen häufig in den Hintergrund. Gerade wegen der bevorstehenden Digitalisierung von 575 öffentlichen Leistungen bis 2022 (OZG) stellt

sich damit die Frage, wie staatliche Lösungen die Identität der Bürgerinnen wahren können – um Abhängigkeiten von Privaten zu vermeiden und Bürgerinnen eine nachhaltige, nutzerfreundliche und faire öffentliche Alternative zu bieten.

Vorgehensweise

Die Vorgehensweise beinhaltet zwei wesentliche Phasen: Zum Einen wurde in einem ersten großen Arbeitspaket das Identity-Stick Konzept entwickelt und getestet. Dies beinhaltet unter anderem die Analyse des FIDO2-Standards, Stakeholder- und Expertinneninterviews und Partnerabsprachen, als auch das Testen von frühen Versionen mit potentiellen Endnutzerinnen. In der zweiten Phase lag der Fokus auf der technischen Entwicklung und visuellen Umsetzung des Prototyps.

Meilensteine

Die Entwicklung des prototypischen Identity-Sticks gliedert sich in die konzeptuelle Vorarbeit und prototypische Fertigstellung des Identity-Sticks:

AP 1: Entwicklung und Test des Identity-Stick Konzepts

AP 1.1.: Analyse des FIDO2-Standards: Überprüfen bisheriger Implementationen des Standards auf mögliche Anknüpfungspunkte für Identitätsattribute

AP 1.2.: Partnerabsprachen: Festlegen eines gemeinsamen Projektrahmens bzw. Austausch zu Herausforderungen und Verbesserungsmöglichkeiten mit Governikus und/oder der Bundesdruckerei [Für die Erweiterung des FIDO2-Protokolls wird mit der Governikus KG und ggf. mit der Bundesdruckerei kooperiert]

AP 1.3.: Nutzertests: Iterative Entwicklung eines ersten Prototyps für Nutzertests zur Usability eines solchen Identity-Sticks.

AP 1.4: Open Source Veröffentlichung des Konzepts und Nutzertestergebnisse

AP 2: Erstellung des Identity Sticks Prototyps

AP 2.1: Erweiterung des FIDO2-Standards um verifizierte Identitätsattribute: Anbindung der Identitätsfeststellung: Der Prototyp soll mittels der von Governikus bereitgestellten Architektur eine Identitätsfeststellung durch den neuen Personalausweis ermöglichen.

AP 2.2.: Umsetzung des WebAuthn- und CTAP-Standards

AP 2.3.: Open Source Veröffentlichung des Codes des Identity-Sticks

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

Zielgruppe

Der Identity-Stick kann für verschiedene Zielgruppen interessant sein: Für Unternehmen, für den öffentlichen Bereich (öffentliche Verwaltungen, Universitäten, Schulen), für Bürgerinnen im digitalen Raum (ca. 84% der Bevölkerung) oder Personen mit erhöhtem Sicherheitsbedarf (z.B. Journalistinnen, Patientinnen). Da unser Produkt sowohl den anonymen und passwortlosen Login für private Services und die sichere Identifikation für öffentliche Leistungen während der Antragstellung ermöglichen soll wird eine sehr breite Zielgruppe angesprochen. All diese unterschiedlichen Zielgruppen müssen sich entweder selbst oder ihre Kundinnen digital identifizieren und profitieren von sicheren und einfachen Lösungen dafür. Eine einheitliche Lösung könnte für sie alle von Vorteil sein (vgl. auch Abschnitt "Zielgruppe, Nutzen und mögliche Weiterentwicklung).

Bezüge zu den Themenfeldern und Zielen des Software Sprints

"Vertrauen" im Internet ist ganz wesentlich an die digitale Identität geknüpft. Ob jemandem oder etwas vertraut werden kann oder nicht hängt auch in erster Linie davon ab, ob sich diese Person oder Maschine glaubwürdig identifizieren kann. So können zum Beispiel Personen, die sich mit einem Schlüssel den nur sie besitzen vertraut werden und damit legitim Zugriff auf ein Konto gegeben oder ein Antrag gestellt werden. Im Vergleich zur analogen Welt ist es im Internet schwierig, Vertrauen herzustellen und einfacher, größere Angriffe auf vertrauenswürdige Instanzen durchzuführen. Im Internet gibt es auch die Besonderheit, dass Andere als "Identity Provider" an die Stelle von vertrauenswürdigen Instanzen treten, wie man es vom "Sign in mit Google" oder Facebook kennt: Sie bürgen als Identitätsprovider dafür, dass dem Konto vertraut werden kann. Die Privatisierung der Identität birgt aber auch gesellschaftliche Risiken: Normen wie Privacy und Sicherheit treten bei profitorientierten Modellen häufig in den Hintergrund. Gerade wegen der bevorstehenden Digitalisierung von 575 öffentlichen Leistungen bis 2022 (OZG) stellt sich damit die Frage, wie staatliche Lösungen die Identität der Bürgerinnen wahren können – um Abhängigkeiten von Privaten zu vermeiden und Bürgerinnen eine nachhaltige, nutzerfreundliche und vertrauenswürdige öffentliche Alternative zu bieten.

Ausführliche Darstellung der Ergebnisse

Erzielte Ergebnisse

- Identity Stick: Code für einen Solo Key mit der Möglichkeit des Ausweisens mittels der Identity Stick Extension
- Identity Stick Extension: Beschreibung des Protokolls für den Austausch von Identitätsattributen mittels CTAP (FIDO2)
- Interface Demos: User Interface Demos, die ein mögliches Interface für passwortlosen Login bzw. Ausweisen simulieren
- Identity Stick Server: Skript zum Testen der Funktionalität eines Identity Sticks
- FIDO Debugger: Webseite zum schnellen Testen der Daten, die von einem FIDO Token gesendet werden

- Zwei Tutorials für die Programmierung des Solo Hacker Sticks
- Signed Identity: Tool zum Erzeugen von signierten Identitätsattributen
- Identity-Stick.github.io: Code für unsere Webseite & eigene Projektwebseite unter der Domain
- Demovideo und eigene Demoanwendung auf identity-stick.github.io
- Blogpost für Prototypfund¹ und Veröffentlichung der Inhalte auf Webseite
- Expertinnengespräche und Konzepttests; Dokumentation der Vorgehensweise und Ideen

Erreichte Meilensteine

Die Meilensteine konnten alle erreicht werden. Durch die Verwendung des “Solo Sticks for Hackers” konnte auf eine Umsetzung des Webauthn/CTAP aufgebaut werden. Dadurch konnte die Erstellung und Prüfung von Konzepten stärker fokussiert werden. Lediglich das Arbeitspaket der Nutzertests konnte durch die Covid-19 Pandemie nicht im geplanten Umfang durchgeführt werden. Da es sich bei dem Identity-Stick um ein Hardware Produkt handelt, war es schwierig, ein Testsetting zu entwerfen, dass in einem Fernzugriff Kontext funktioniert.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Für diese Zielgruppen könnte die Lösung interessant sein:

- **Für Unternehmen**
Unternehmen haben häufig besonders sensible Daten. Dafür braucht es gesteigerte Sicherheitsvorkehrungen. Ein USB-Schlüssel kann Mitarbeitenden Zugang zu ihren Accounts geben und dabei Datensicherheit an erste Stelle setzen.
- **Für den öffentlichen Bereich**
Schulen, das digitale Rathaus und vieles mehr: Wir brauchen Ausweise, um unsere Identität zu bestätigen. Mit einem digitalen Schlüssel können diese Prozesse vereinfacht werden.
- **Für Bürgerinnen**
Sicherheit soll nicht nur für Kryptoexpert:innen gelten. Bürgerinnen sollen ihre digitale Identität einfach schützen und online Ausweisprozesse integrieren können. In einem Produkt.
- **Für Personen mit erhöhtem Sicherheitsbedarf**
Manche Gruppen brauchen besonderen Schutz ihrer Daten und Accounts. Journalistinnen müssen ihre Quellen sichern oder Ärztinnen den Zugang zu Daten

¹ <https://demoweeek.prototypfund.de/projects/Q8-identity-stick.html>

von PatientInnen. Manche Personen legen besonderen Wert darauf sich online abzusichern.

Weitergehende Effekte der Open-Source-Stellung der Ergebnisse

Der Identity-Stick kann weiterentwickelt werden und für verschiedene Sicherheitsbedürfnisse angepasst werden, sodass bestimmte gewünschte Standards erfüllt werden. Außerdem ermöglicht dies auch den Einsatz für kleine Kreise, die ihn zum Ausweisen benutzen wollen. Dies könnten beispielsweise Vereine, andere Organisationen oder Einrichtungen wie Schulen sein.

Ideen für die Weiterentwicklung der Lösung und Pläne zu deren Umsetzung

Für ein vollumfängliches Produkt könnten weitere Entwicklungsschritte unternommen werden:

- **Personalisierung des Sticks und Signatur von Daten**

Die Personalisierung des Sticks bzw. Signatur der Daten müsste in einem Produktionsumfeld von einer vertrauenswürdigen Instanz gesichert werden. Dafür käme eine staatliche Stelle wie bspw. die Bundesdruckerei in Frage, die auch die technische Infrastruktur der eID betreut. Für eine einsatzfähige und sichere Version des Sticks müsste die Hardware von dieser Instanz hergestellt, die Maintenance gesichert und die entsprechenden individuellen Daten bei Erstausstellung für jede Bürgerin signiert werden. Es wäre nützlich, Bürgerinnen mit entsprechenden Informationen und Materialien über die Funktionsweise aufzuklären.

- **Weiterentwicklung des Protokolls**

In der momentanen Version des Identity Stick Prototyps werden die Daten signiert auf dem Stick abgelegt und dann an Diensteanbieterinnen weiter gesendet. Dies verhindert zum Einen keine Replay-Attacken. Zum Anderen könnten die bestätigten Daten von Diensteanbieterinnen an Dritte weitergegeben werden. Daher müssten weitere Verbesserungen des Protokolls erfolgen. So sollte ein authentifizierter sichererer Kanal zwischen Stick und Diensteanbieterin aufgebaut werden über den Daten ausgetauscht werden könnten.

- **Erweiterter Hardwarespeicher und Features**

Der Speicherplatz des Solo-Sticks ist limitiert, weshalb nur eine begrenzte Menge an Informationen darauf abgelegt werden kann. Mit der Produktion eigener Hardware könnte dieser Speicherplatz und damit die Funktionalität des Sticks erweitert werden. So könnte der Stick neben der sicheren (multi- und passwortlosen) Authentifizierung und Identifizierung auch einen sicheren Datenspeicher beinhalten, auf dem wichtige Dateien und Zertifikate abgelegt werden können, einen Passwortmanager beinhalten und vieles mehr.

- **Bekanntheitsgrad und Verbreitung von FID02**

Der noch recht junge Standard wurde bisher von relativ wenigen Webseiten komplett implementiert. In Zukunft ist damit zu rechnen, dass mehr Diensteanbieterinnen die

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden

Wir starteten mit der Frage "Wie lässt sich der Login mittels FIDO und das online Ausweisen verknüpfen?" Dazu hatten wir drei Konzeptrichtungen erstellt.

- **FIDO-Provider**

Identitätsprovider stellen eine Schnittstelle zwischen Nutzerinnen und Diensten dar. Wenn der Dienst den Identitätsprovidern traut, können diese etwas über die Identität von Nutzerinnen aussagen (z. B. deren Alter). Ganz genau wie bspw. eine Freundin und jemand Fremden vorstellt. Dazu müssen sich Nutzerinnen aber bei Identitätsprovidern einloggen. Dies geschieht bisher über Passwörter und teilweise zusätzliche zweite Faktoren wie mobile TANs. Dies könnte aber durch den passwortlosen Login mittels FIDO ersetzt werden. Nutzerinnen würden sich somit über Identitätsprovider ausweisen, nutzen dafür aber genau das gleiche Mittel wie bei einem Login, nämlich ihren FIDO-Sicherheitsschlüssel. Der Vorschlag wird auch vom Bundesamt für Sicherheit in der Informationstechnik erwähnt.³

- **FIDO mit Ausweis**

Mit dem deutschen Personalausweis können sich Nutzerinnen auch elektronisch ausweisen. Die Idee hinter diesem Konzept war es den Ausweis als Sicherheitsschlüssel für einen passwortlosen Login mittels FIDO2 zu nutzen. Den Ausweis mittels dem FIDO1 Standard (U2F-Protokoll) als 2. Faktor bei Webseiten zu hinterlegen ermöglicht bereits Fidelio.⁴ Mehr dazu wird in einem Video⁵ genauer erläutert. Die Überlegung wäre gewesen darauf aufbauend eine Art 2.0 Variante zu entwickeln, bei der der Ausweis direkt über die AusweisApp2 genutzt und der neue FIDO2 Standard eingesetzt werden könnte.

- **Fido-Vault**

Der Identity Stick könnte weiter entwickelt werden, um Privatpersonen eine stärkere Kontrolle über ihre Daten zu geben. So könnten neue Konzepte, wie zum Beispiel eine Art 'FIDOVault' Daten- und Identitätsschutz noch transparenter machen. In einem Vault könnte ein Sicherheitsschlüssel als Speicher für wichtige Daten benutzt werden, die mit der Identität verknüpft sind. Das könnte einen Passwortmanager, eine Auflistung der Dienste, die mit dem passwortlosen Login oder 2-Faktor Authentifikation genutzt wurden, oder auch Zertifikate enthalten. Online Dienste könnten selbst über entsprechende Schnittstellen Informationen im Vault ablegen (z. B. Röntgenbild) und Nutzende selbst Daten (z.B. das 'Elster Zertifikat') eintragen.

³ https://www.bsi.bund.de/DE/Publikationen/TechnischeRichtlinien/tr03t59/tr03159_node.html

⁴ FIDELIO verfügbar hier:

<https://play.google.com/store/apps/details?id=de.persoapp.android.FIDELIO&hl=de>

⁵ Video verfügbar unter: <https://youtu.be/Oc32AwPL2Nw>

Nutzende würden den Diensten dann ausgewählte Daten zur Verfügung stellen, wenn diese benötigt werden.

Hintergründe und alternative Vorgehensweise

Während der Analyse gäbe es folgende Aspekte, die gegen die obigen Konzepte sprachen:

- 'FIDO mit Ausweis' wurde unabhängig von uns während der Projektlaufzeit teilweise umgesetzt (es existieren Browser-Erweiterungen).⁶ Es gibt hier ein paar nicht ohne Weiteres zu lösende Usability Herausforderungen (unklar, wann Nutzende Daten pseudonym übertragen; unklar, dass der 'Ausweis' als "Sicherheitsschlüssel" verwendet werden kann; Download von Plugin und AusweisApps notwendig). Außerdem haben uns die Nutzertests gezeigt, dass Useninnen Bedenken hatten, den Ausweis als Login Mittel einzusetzen.
- Da bei 'FIDOProvider' nur die Art des Logins beim Identitätsprovider angepasst wird, ist das Potenzial einer solchen Lösung geringer. Weiterhin muss dem Identitätsprovider vertraut werden. Wenn dies eine private Organisation ist, so wird die Verantwortung für digitale Identitäten weiter auf den privaten Markt verschoben. Außerdem bietet eine einzelne Stelle wie sie der Identitätsprovider darstellt, Gefahren für den Datenschutz.
- 'FIDO-Vault' kam erst im späteren Projektverlauf auf. Da es als Konzept noch sehr schwammig war und von den benötigten Ressourcen ebenfalls unklar war, hat es leider nicht mehr in den Rahmen dieses Projekts gepasst

Stattdessen verfolgten wir vor allem das Konzept FIDO-Ident:

FIDO-Ident Konzept

Entgegengesetzt zur Nutzung eines Personalausweises zum Login ist die Idee von FIDO-Ident, einen FIDO2-Sicherheitsschlüssel ebenfalls zum Ausweisen einzusetzen. Dazu soll FIDO2 als Protokoll zur Übermittlung von Identitätsattributen genutzt werden, die sich auf dem Sicherheitsschlüssel befinden.

Das Konzept 'FIDOIdent' bietet auf Anbieterseite großes Potenzial, da der Standard schon von Unternehmen umgesetzt wird. Darauf aufzubauen bietet auch den Vorteil, dass sich Nutzerinnen viel häufiger bei Services einloggen, als sich online auszuweisen. Für beides dieselbe Methode zu nutzen, ist damit für Nutzerinnen praktischer.

Weiterleitung der FIDO Erweiterung durch den Browser

Der FIDO-Standard besteht aus zwei Teilen (Webauthn und CTAP). Webauthn bezieht sich auf die Kommunikation zwischen Webseite und Browser und CTAP auf die Kommunikation zwischen Browser und Sicherheitsschlüssel. Erweiterungen des FIDO-Protokolls müssen von der Webseite an den Browser gesendet, aber von diesem auch weitergeleitet werden. Dies wird vom Standard nicht strikt gefordert. Daher wurde in der Mitte des Projekts geprüft,

⁶ Browser Erweiterungen verfügbar unter:
<https://addons.mozilla.org/de/firefox/addon/webauthn-eid-for-firefox/>

inwieweit Browser eine solche Weiterleitung unterstützen. Google Chrome bzw. Chromium unterstützen die Weiterleitung nur für einige spezielle Erweiterungen. Der Code von Chromium ist Open Source. Daher wurde geprüft, inwieweit die Möglichkeit besteht, eine abgeänderte Variante von Chromium zu verwenden, die die Identity Stick Extension weiterleitet. Dies wurde schlussendlich nicht weiter verfolgt, da der Aufwand zu groß gewesen wäre und die Arbeit am Protokoll und dem Prototypen selbst dadurch stärker eingeschränkt worden wäre.

Im Zuge dieser Analyse konnte aber festgestellt werden, dass eine entsprechende Weiterleitung analog zu der Implementierung anderer Erweiterungen erfolgen könnte.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Neben dem Blogartikel und Video, die für die Demoweeek des Prototype Funds erstellt wurden⁷, gibt es eine Webseite (<https://identity-stick.github.io>) mit weiteren Erläuterungen und Beschreibungen. Von uns erstellter Code sowie die Beschreibung der Erweiterung finden sich auf Github (<https://github.com/Identity-Stick>). Darauf findet sich der Code für:

- Identity Stick: Code für einen Solo Key mit der Möglichkeit des Ausweisens mittels der Identity Stick Extension
- Identity Stick Extension: Beschreibung des Protokolls für den Austausch von Identitätsattributen mittels CTAP (FIDO2)
- Interface Demos: Demos, die ein mögliches Interface für passwortlosen Login bzw. Ausweisen simulieren
- Identity Stick Server: Skript zum Testen der Funktionalität eines Identity Sticks
- FIDO Debugger: Webseite zum schnellen Testen der Daten, die von einem FIDO Token gesendet werden
- Signed Identity: Tool zum Erzeugen von signierten Identitätsattributen
- Identity-Stick.github.io: Code für unsere Webseite

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ereignisse im Projektverlauf, die eine Anpassung der Planung erforderlich machten

Die Covid-19 Maßnahmen haben dazu geführt, dass Nutzertests und Interviews nicht in dem

⁷ Blogartikel, Video und Demoweeek Inhalte:

<https://demoweeek.prototypefund.de/projects/08-identity-stick.html>

Umfang durchgeführt werden konnten wie anfangs geplant. Dafür wurde mehr Arbeit in die Überprüfung der technischen Machbarkeit der verschiedenen Konzepte gesteckt.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Entwicklungen anderer Personen oder Institutionen, die Einfluss auf die Arbeit und Zielsetzung hatten

Während der Arbeit an den Konzepten hatte Christoph Kahlo FIDELIO⁸ weiterentwickelt. Dies hat eine mögliche Entwicklung von uns in diese Richtung voraus gegriffen. Die Umsetzung hat uns aber auch gezeigt, welche offenen Fragen bei einer solchen Lösung immer noch bestehen (siehe oben Konzept "FIDO mit Ausweis"). Da uns damit deutlich wurde, dass wir nicht viel dazu beitragen könnten, haben wir diese Richtung nicht weiter verfolgt.

Da wir auf den FIDO2 Standard aufbauen, hatte dieser Standard große Auswirkungen auf unsere Arbeit. Während der Projektlaufzeit hat sich zum Beispiel Apple der FIDO Allianz angeschlossen und unterstützt jetzt FIDO2 mit dem Safari-Browser.⁹ Der noch recht junge Standard FIDO2 befindet sich noch im Implementationsprozess und wird sich in Zukunft stärker verbreiten.

Wir haben den SoloKey Flacker verwendet. Dies ist ein Open Source (Hardware und Software) FIDO2 Token, dessen Code angepasst werden kann. Da die FIDO2-Funktionalität besteht, hat es sich angeboten diesen als Basis zu verwenden, um darauf aufzubauen und diesen um eine Ausweisfunktion zu erweitern. Allerdings hat dies auch bedeutet, dass Restriktionen durch die Hardware vorgegeben sind. So ist der Speicher begrenzt und ermöglicht nur eine begrenzte Länge beim Versenden der Nachrichten. Daher wurde das von uns entwickelte Protokoll so angepasst, dass es mit diesen Einschränkungen umgehen kann und zu lange Nachrichten aufgeteilt und die Teile einzeln versendet werden können.

⁸ <https://play.google.com/store/apps/details?id=de.persoapp.android.FIDELIO&hl=de>

⁹ Heise Online (2020): Passwort-Nachfolger: Apple bei FIDO-Allianz mit an Bord. Online: <https://www.heise.de/mac-and-i/melduna/Passwort-Nachfolger-ApDie-bei-FIDO-Allianz-mit-an-Bord-4657979.html>

Richtlinie zum „Software-Sprint“

Alarmdisplay - Einsatzdaten zur Unterstützung von Hilfsorganisationen

Schlussbericht

Zuwendungsempfänger:

Andreas Brain

DLR Projektträger Berlin

Eingegangen am:

01. Okt. 2020

Eingangsnr.:

4985/B3

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S30 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Das Projekt Alarmdisplay soll Hilfsorganisationen wie z. B. der Feuerwehr als sogenanntes Informationssystem dienen. Dabei werden eingehende Alarmmeldungen verarbeitet und in einer gut zu erfassenden Art und Weise dargestellt (z. B. auf einem Bildschirm in der Fahrzeughalle). Dieser Bedarf entstand aus der persönlichen Erfahrung des Autors bei der Freiwilligen Feuerwehr. Zwar gibt es bereits solche Systeme auf dem Markt, jedoch sind diese in der Regel mit laufenden Lizenzkosten verbunden. Ebenso kann die Beschaffung eines PCs mit Microsoft Windows erforderlich werden, der durchgehend eingeschaltet bleiben muss. Diese Höhe von initialen und laufenden Kosten steht für kleine Feuerwehren mit unregelmäßigem Einsatzaufkommen oft nicht im Verhältnis zum erwarteten Nutzen. Das Open-Source-Projekt AlarmWorkflow¹ setzt ebenfalls auf Windows auf und eine Weiterentwicklung war in den letzten Jahren kaum vorhanden.

Das Ziel des Projektes war es, ein frei verfügbares und betriebssystemunabhängiges Informationssystem zu schaffen, das auf Kleinstrechnern wie dem Raspberry Pi lauffähig ist. Damit sollten sowohl die Beschaffungskosten als auch die laufenden Kosten gesenkt werden. Zusätzlich sollte das System über Schnittstellen verfügen, über die es mit bestehender oder noch zu entwickelnder Software verknüpft werden kann.

Das Projekt wurde von Anfang an in zwei Komponenten unterteilt. Eine Komponente zur Erfassung, Verarbeitung und Weiterleitung von Alarmmeldungen (Alarmzentrale), sowie eine Komponente zur Darstellung der Daten zu einem Einsatz (Alarmanzeige). Die beiden Komponenten sollten auf einander abgestimmt sein, sich aber nicht gegenseitig bedingen. Somit sollte es auch möglich sein, nur eine der beiden Komponenten zu verwenden und mit eigener Software zu kombinieren.

Folgende Meilensteine wurden definiert:

- Aufsetzen der Projektstruktur und Build-Automatisierung
- Verwaltungsoberfläche für Displays und Setup-Prozess
- Ruhemodus mit Datum/Uhrzeit, Terminen und Wetterwarnungen
- Texterkennung (OCR) und Validierung beim Faxempfang
- Extraktion einsatzrelevanter Daten aus Texten nach bekanntem Muster (Fax der ILS Augsburg)
- Alarmanzeige mit den wichtigsten Daten
- Validierung von Adressen und Koordinaten
- Karten für die Alarmanzeige mit Zusatzdaten von OpenStreetMap (z. B. Hydranten)

¹ <https://github.com/OpenFireSource/AlarmWorkflow>

- Extraktion einsatzrelevanter Daten aus Texten nach einem frei definierbaren Muster
- APIs zur Datenübermittlung inkl. Authentifizierung
- Verwaltungsoberfläche zur Verknüpfung von Dateneingang, Verarbeitungsschritten und Datenausgabe
- Ausdruck von Einsatzdaten und Karten

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

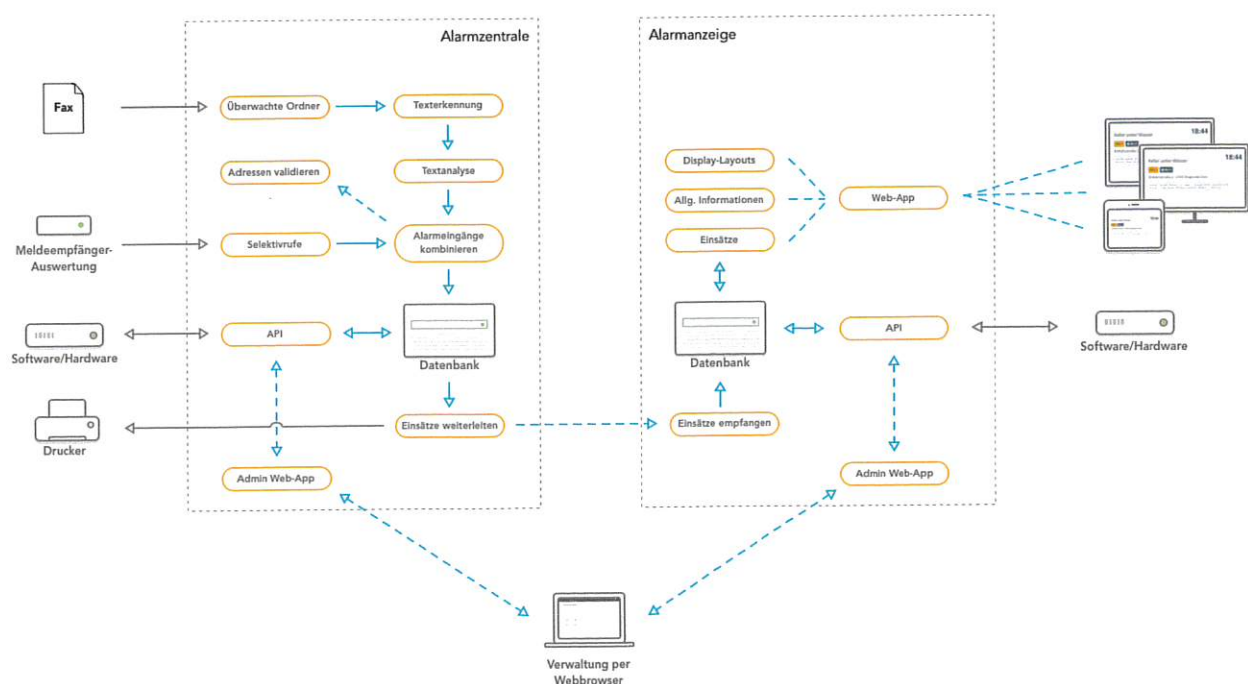
Neben den genannten kleinen Freiwilligen Feuerwehren sind grundsätzlich alle Hilfsorganisationen, die spontan Einsatzkräfte alarmieren müssen, Teil der Zielgruppe. Selbst große Organisationen, bei denen die Kosten für ein proprietäres System keine Rolle spielen würden, profitieren von mehr Transparenz und Interoperabilität. Anpassungen oder Integrationen mit bestehenden Systemen sind keine Frage von Lizenzen oder Verträgen, sondern die Nutzenden werden ermächtigt, diese selbst vorzunehmen oder zu beauftragen. Dies stärkt die Flexibilität und Schlagkraft der Hilfsorganisationen, was der Gesellschaft als Ganzes zugute kommt.

Ausführliche Darstellung der Ergebnisse

Es wurden wie geplant die zwei Komponenten Alarmzentrale und Alarmanzeige geschaffen. Beides sind eigenständige Projekte, die miteinander kompatibel sind und folgen intern dem gleichen Aufbau. Den Kern bildet jeweils ein Node.js-Server, der an eine relationale Datenbank angeschlossen ist. Sämtliche Entitäten der Software sind über eine REST-API per HTTP oder über eine WebSocket-Verbindung verwaltbar.

Ebenso liefert der Server eine grafische Oberfläche zur Verwaltung („Console“) als Webapp aus. Im Fall der Alarmanzeige wird noch eine zusätzliche Webapp für die Anzeige auf den Bildschirmen ausgeliefert. Somit ist jegliche grafische Interaktion über ein beliebiges Gerät mit einem Webbrowser durchführbar. Die Webapps wurden auf Basis von Vue.js geschrieben und sind per WebSocket mit dem Backend verbunden. Das ermöglicht nicht nur einen sehr schnellen Datenaustausch, sondern auch eine laufende Kontrolle ob angeschlossene Systeme nach wie vor verbunden und betriebsbereit sind.

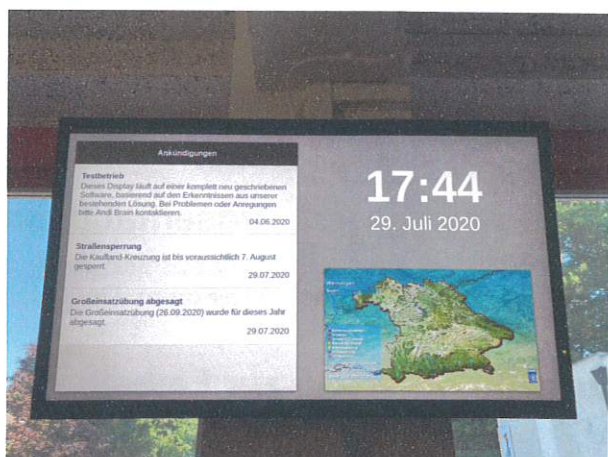
Der Zugang zur API wird über JSON Web Token (JWT) abgesichert, die an einen Benutzeraccount geknüpft sind und eine begrenzte Gültigkeit haben. Um auch Lösungen zu ermöglichen, in denen beispielsweise Mikrocontroller eingesetzt werden, können ebenso dauerhaft gültige API-Tokens generiert werden, die unabhängig von einzelnen Benutzern sind. Derzeit gibt es weder bei den Benutzerkonten noch bei den API-Tokens ein Konzept zur Beschränkung von Berechtigungen, soll aber noch eingearbeitet werden.



Die Alarmzentrale kann Ordner auf neu eintreffende PDF-Dateien (Alarmfaxe) überwachen und diese auslesen. Dazu kommt die OCR-Software tesseract² zum Einsatz, deren Ergebnisse dann nach relevanten Daten zu Einsätzen durchsucht werden. Nachdem jede Leitstelle ein anderes Layout für das Alarmfax verwendet, kann bisher nur das Fax der ILS Augsburg zuverlässig ausgewertet werden. Die Layoutdefinition zur Textanalyse wurde bereits in ein neutrales Format mit regulären Ausdrücken gegossen, sodass die Erweiterung um weitere Layouts keinen Eingriff in den Code erforderlich macht.

Auch von außen können Alarme per API zugeliefert werden, beispielsweise für per Funk empfangene Alarmierungen durch Fünftönefolgen. Mehrere eingehende Alarme werden zu Einsätzen kombiniert, konfigurierte Einsatzmittel werden erkannt und den Einsätzen zugeordnet. Enthaltene Adressen können optional mit Daten von OpenStreetMap abgeglichen werden. Die Einsätze werden an die Alarmanzeige weitergeleitet, weitere Schnittstellen (z. B. für die Zusatzalarmierung) sind geplant.

Die Alarmanzeige stellt den eingegangenen Einsatz möglichst knapp und übersichtlich dar: Einsatzgrund, Stichwort, Zeit seit dem Alarmeingang, Zieladresse und einen Freitext der Leitstelle. Wenn gerade kein Einsatz vorliegt, dient die Anzeige als Informationsübersicht. Bisher sind Datum und Uhrzeit, Unwetterkarten des Deutschen Wetterdienstes und Ankündigungen umgesetzt. Die Anordnung der einzelnen Informationen kann mit einem Drag-and-Drop-Editor in der Console verändert werden. Jedes verbundene Display ist individuell konfigurierbar, auch mehrere wechselnde Ansichten pro Display sind möglich.



Die Darstellung von Terminen wurde zugunsten der universeller einsetzbaren Ankündigungen in die Zukunft verschoben. Im Zeitrahmen der Förderung konnte die Kartendarstellung und der Ausdruck von Einsatzdaten nicht mehr umgesetzt werden, ist aber weiterhin vorgesehen. Eine grafische Oberfläche zur Verknüpfung von Dateneingang, Verarbeitungsschritten und Datenausgabe war vorerst nicht nötig, da die Optionen noch nicht so vielfältig sind.

Neben der ständigen Betreuung durch die Open Knowledge Foundation war auch das UX-Coaching durch *Simply Secure* eine große Hilfe, den Fokus auf die wichtigen Dinge zu richten.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Neben den unter „Beitrag des Projektes zu den Zielen der Förderinitiative“ genannten Vorteilen für die Zielgruppe und die Gesellschaft, ergeben sich alleine aus der Veröffentlichung des Quellcodes langfristig positive Effekte. So besteht eine Basisvariante eines Informationssystems, das Interessierten einen Einstieg bietet und für einige auch dauerhaft ausreichend sein wird. Komplexere Anforderungen können über die Zeit in Kooperation mit den betroffenen Organisationen umgesetzt werden. An FOSS-Lösungen interessierte Städte und Verbände könnten das Projekt finanziell

² <https://tesseract-ocr.github.io/>

oder mit Beiträgen zu Code und Dokumentation stärken und so den Einsatz von quelloffener Software in der Daseinsvorsorge vorantreiben.

Die Arbeit an diesem Projekt war eine bereichernde Erfahrung, da alle Aspekte von der Planung und dem UX-Design, über Code und Tests bis hin zur Dokumentation abgedeckt werden mussten. Dennoch ist die Verteilung der Aufgaben auf mehrere Schultern für die Zukunft wünschenswert.

Zunächst sollen weitere Alarmquellen angebunden und ebenso die Weiterleitung von Einsätzen an andere Lösungen ermöglicht oder vereinfacht werden. Ebenso sollen aktuelle Wetterdaten des DWD über das ebenfalls in der 7. Runde geförderte Projekt *Bright Sky*³ eingebunden werden. Zusammen mit den Nutzenden aus den verschiedenen Leitstellenbereichen soll die Anzahl der bekannten Alarmfaxlayouts erhöht werden. Die direkte Anbindung von Leitstellensystemen ist vermutlich nicht realisierbar, da bestimmte Hersteller mit Geheimhaltungsverträgen arbeiten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Arbeiten begannen mit der Alarmanzeige. Da der Server auf Basis von Node.js programmiert wurde, war das Ziel, die Zahl der Abhängigkeiten so gering wie möglich zu halten. In der Vergangenheit kam es immer wieder vor, dass einzelne npm-Module nicht mehr weiterentwickelt oder sogar gelöscht wurden. Daher wurde anfangs kein vollumfängliches Backend Framework verwendet. Mit der Zeit traten aber immer wieder Problemstellungen auf, die selbst bei weniger komplexen Anforderungen nicht innerhalb des Förderzeitraums in Eigenregie zu lösen waren. Dazu zählen insbesondere die Datenbankanbindung und die Benutzerauthentifizierung. Die Alarmzentrale wurde dann von Anfang an auf dem Framework FeathersJS aufgebaut, das die besagten Probleme löste. Zusätzlich vereinfachte es die Synchronisation von Front- und Backend, was es noch attraktiver machte. Daraufhin wurde das Backend für die Alarmanzeige auf Basis von FeathersJS neu geschrieben, ebenso musste die Anbindung des Frontends angepasst werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Das Projekt wird auf der Projektwebseite⁴ vorgestellt. Dort finden sich außerdem Links zur Dokumentation und zum Community-Forum. Der Quellcode wurde auf GitHub⁵ veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Projektbeginn war geprägt von der COVID-19-Pandemie und den Gegenmaßnahmen, insbesondere dem sogenannten Lockdown. Dadurch standen keine dedizierten Arbeitsumgebungen wie Co-Working-Spaces zur Verfügung. Diese Situation beeinflusste die Produktivität negativ, so dass die in das Projekt investierten Stunden hinter den Planungen zurück blieben. Im Laufe der Förderung besserte sich die Situation und die Stunden konnten teilweise wieder aufgeholt werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Im Laufe der Förderung wurde sowohl im Landkreis als auch in der Heimatgemeinde des Autors die Einführung eines proprietären Informationssystems vorangetrieben. Dies führte zum Austausch mit verantwortlichen Personen, um ihnen die Entstehung des Projekts Alarmdisplay bekannt zu machen. Dies brachte vor allem neue Erkenntnisse zu potentiellen Nutzenden für einen Pilotbetrieb und weitere Zielsetzungen für neue Funktionen nach der Förderphase.

³ <https://brightsky.dev/>

⁴ <https://alarmdisplay.org/>

⁵ <https://github.com/alarmdisplay>

Augsburg, 30.09.2020

AB

Richtlinie zum „Software-Sprint“

EyeCaptain – Eyetracking-Tools für Menschen mit Behinderung

Schlussbericht

Zuwendungsempfänger:
Besold Hörmann Ricksgers Tempes GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S31 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

EyeCaptain will einen einfachen Zugang zu technischen Hilfsmitteln ermöglichen und damit, Teilhabe und Selbstbestimmung durch technischen Fortschritt stärken. Hierzu erfolgt die Entwicklung von kostenfreien und quelloffenen technischen Assistenzsystemen für Menschen mit Behinderungen, die auf Blicksteuerung basieren. Ausgangspunkt ist der Prototyp einer auf Eyetracking basierten Rollstuhlsteuerung: Die Software ermöglicht die Steuerung des Rollstuhls über Blickbewegungen des Fahrenden, wenn der Rollstuhl nicht anders bedient werden kann.

Unsere Arbeitsschritte haben sich an den folgenden Meilensteinen orientiert:

Meilenstein 1: Prüfung, welche Rollstuhl-Steuerprotokolle am Markt bestehen, wie häufig diese jeweils genutzt werden und für welche Protokolle Schnittstellen für das Projekt entwickelt werden können.

Meilenstein 2: Risikoanalyse und Ausarbeitung eines Sicherheitskonzepts. Damit einhergehend eine rechtliche Beratung zur Haftung bei der Veröffentlichung von Open-Source Produkten sowie die Planung von Sensortechnik und redundanten Abschaltmechanismen.

Meilenstein 3: Implementierung der ausgewählten Steuerprotokolle, um die Übermittlung der Augenposition an den Rollstuhl zu ermöglichen. Zudem die Überarbeitung eines bestehenden mechanischen Adapters, um auch Unterstützung für seltene Rollstuhlmodelle zu ermöglichen.

Meilenstein 4: Weiterentwicklung des Proof-of-Concepts zur nutzbaren Tablet Software sowie der Rollstuhl-Steuersoftware.

Meilenstein 5: Gestaltung der Nutzeroberfläche aus UX-Design-Perspektive. Erarbeitung und Prüfung der Anleitungen sowie die Erstellung von Installationsroutinen für eine einfache Nutzbarkeit.

Meilenstein 6: Tests mit Nutzer:innen.

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

EyeCaptain richtet sich an Menschen mit schweren körperlichen Beeinträchtigungen wie amyotrophe Lateralsklerose, Multiple Sklerose oder spinale Muskelatrophie. Mit der Entwicklung und Bereitstellung von technischen Hilfsmitteln schafft EyeCaptain einen einfachen Zugang zu bestehenden und neuen Lösungen und stärkt damit die Teilhabe und Selbstbestimmung von Menschen mit Behinderung.

EyeCaptain bezieht sich auf die Themen und Ziele des Software-Sprints. Ein besonders starker Zusammenhang besteht zu den Bereichen Inklusion und Gesundheit, da EyeCaptain einen konkreten Beitrag zur gesellschaftlichen Teilhabe von Menschen mit körperlicher Behinderung leistet.

Ausführliche Darstellung der Ergebnisse

Technisch betrachtet besteht unsere Software aus zwei Teilen. Einerseits die Benutzeroberfläche, die auf einem Windows Tablet läuft. Hier werden die Blickbewegungen aufgezeichnet und interpretiert. Der zweite Teil läuft auf einem Mikrocontroller. Hier werden die Befehle in die jeweiligen Protokolle des Rollstuhls bzw. der mechanischen Hardware übersetzt. Zwischen den beiden Teilen werden permanent Daten in beide Richtungen ausgetauscht (über WLAN oder über USB). So werden die eingehenden Befehle des Users kontinuierlich mit der tatsächlichen Richtung und Geschwindigkeit des Rollstuhls abgeglichen.

Unsere Nutzeroberfläche basiert auf dem Node.js Framework Electron und einem darunter liegenden .NET-Layer, das Blickpositionen aus dem Eyetracker über einen Websocket zur Verfügung stellt. Als Eyetracker haben wir eine günstige Variante aus dem Gaming Bereich gewählt, die uns ausreichende Genauigkeit liefert. In der Oberfläche können Knöpfe mit Aktionen definiert werden: blickt ein:e Nutzer:in für einige Sekunden auf den Knopf, wird die zugeordnete Aktion ausgelöst. Wenn der Fahrmodus aktiviert ist werden die Koordinaten der Blickbewegungen ebenfalls über eine Websocket-Verbindung konstant an den Mikrocontroller gestreamt.

Als Microcontroller verwenden wir einen NodeMCU mit integriertem WLAN-Modul, jedoch kann auch ein Arduino Uno oder ein ähnliches Board verwendet werden. Der Microcontroller benötigt eine eigene Stromversorgung und stellt einen passwortgeschützten Access Point zur Verbindung mit dem Tablet bereit. Auf diese Weise können Koordinaten der jeweiligen Blickrichtung auf dem Tablet übertragen werden. Man könnte auch sagen, es werden virtuelle Joystick-Positionen übermittelt. Im Mikrocontroller werden diese Daten verarbeitet und für die jeweilige Ausgabe übersetzt. Dabei wurden drei Möglichkeiten umgesetzt: Wir haben die weit verbreiteten Protokolle R-Net und DX-Bus integriert, sowie für alle anderen elektrischen Rollstühle einen 3D gedruckten Adapter entworfen, der direkt auf den vorhandenen Joystick aufgesetzt werden kann und diesen mit Hilfe von zwei Servomotoren bewegt. Für die Steuerung mittels R-Net und DX-Bus muss ein zusätzliches CAN-Bus shield verwendet werden, das direkt mit der internen Steuereinheit des Rollstuhls verbunden wird. Beide Protokolle sind nicht offen verfügbar, weshalb für die Entwicklung das reverse engineering der Befehle nötig war. Damit können wir die Steuerungsbefehle in das Protokoll des jeweiligen Rollstuhls übersetzen.

Zu Beginn unserer Förderung konnten wir mit einzelnen Personen über den Prototyp sprechen, sowie erste Testungen durchführen. Im weiteren Verlauf mussten wir unseren Kontakt zur Zielgruppe auf

digitale Zugangswege umstellen, da durch das Infektionsgeschehen Covid-19 ein direkter Kontakt nicht mehr möglich war. Unsere Zielgruppe zählt zur Hochrisikogruppe, weshalb eine zusätzliche Gefährdung ausgeschlossen werden musste. Für die weiteren Schritte unseres Prototyps planen wir zusätzliche Testungen, auch mit der Möglichkeit für Nutzer:innen unseren Prototyp längere Zeit auszuprobieren.

Von der Unterstützung durch die Open-Knowledge Foundation konnten wir insbesondere wegen des hervorragenden Netzwerkes profitieren, in dem sich für viele Fragestellungen wie technische Hürden, Organisationsformen und rechtliche Themen kompetente Experten fanden. Einige sehr hilfreiche Kontakte werden hier auch über den Zeitraum der Förderung hinaus bestehen bleiben. Als wichtige Erkenntnis aus der Projektarbeit können wir insbesondere im Hinblick auf die Entwicklungen im Rahmen von COVID-19 mitnehmen, dass eine intensive Entwicklungsarbeit sehr wohl auch über Distanz zu verwirklichen ist. Einschränkungen mussten wir aber bzgl. des Kontakts zur Zielgruppe machen, sodass nur wenige der eigentlich wichtigen Tests mit Nutzenden durchgeführt werden konnten.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Durch unser Projekt bringen wir die stetig und schnell stattfindende Digitalisierung zu Menschen, die davon in hohem Maße profitieren könnten, jedoch häufig außer Acht gelassen werden. Wir ermöglichen durch unsere Entwicklungen und den Zugang zu anderen technischen Assistenzsystemen mehr Autonomie und Selbstbestimmung und dadurch mehr gesellschaftliche Teilhabe.

Indem wir unsere Entwicklungen Open-Source zur Verfügung stellen, ist eine individuelle Anpassung oder auch Weiterentwicklung unserer Ideen möglich. Wir freuen uns über alle Entwickler:innen, die in diesem Bereich aktiv sind oder aktiv werden möchten.

Die Arbeit im Projekt hat uns sowohl persönlich als auch fachlich neue Perspektiven aufgezeigt. Wir konnten uns sowohl in einige neue Technologien einarbeiten, haben aber auch viel über die Organisation eines Open-Source Projektes und Teams gelernt (hier seien die Stichworte Continuous Integration, konsistente Dokumentation, Außendarstellung und Remote-Work genannt).

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben einen funktionierenden Prototyp. Bis unsere Zielgruppe die Rollstuhlsteuerung im Alltag nutzen kann, gibt es noch einige Hürden. Im Moment prüfen wir, inwieweit unser Produkt außerhalb einer Klassifizierung als Medizinprodukt veröffentlicht werden kann. Hierzu haben wir uns von Rechtswissenschaftler:innen der KULeuven beraten lassen, die auf Open-Source-Software als Medizinprodukt spezialisiert sind. Wir haben diskutiert, inwiefern unsere Software unter das Medizinproduktegesetz fällt und welche Möglichkeiten es für die Veröffentlichung unter Berücksichtigung des Open Source Charakters gibt.

Dabei sind wir zu folgenden Ergebnissen gekommen:

1. Die Software ist als Medizinprodukt (Klasse 1 bzw. 2a) einzustufen, da sie zur Kompensation von körperlichen Beeinträchtigungen beiträgt und ähnliche Produkte (Sondersteuerungen für Rollstühle)

auch als Medizinprodukte eingestuft sind. Damit ist sie zulassungspflichtig nach dem Medizinproduktegesetz (MPG bzw. MDD/MDR).

2. Eine detaillierte Anleitung zum Nachbau/zur Anwendung des Systems darf aus diesem Grund nicht ohne Weiteres veröffentlicht werden. Es ist aber weniger problematisch Berichte zu veröffentlichen, in denen die Möglichkeit einer solchen Anwendung beschrieben wird.

3. Die Klassifizierung bezieht sich immer auf das gesamte "Produkt", Teile davon können anders klassifiziert werden. Das bedeutet, wir können Teile davon veröffentlichen, wenn wir sie nicht in ihrer medizintechnischen Anwendung beschreiben.

Leider haben sich damit unsere rechtlichen Befürchtungen bestätigt. Nichtsdestotrotz wollen wir aber so viel wie möglich veröffentlichen und vor allem auch die Möglichkeit bieten, unsere Ergebnisse weiterzuverwenden und langfristig für die Zielgruppe nutzbar zu machen.

Wir werden deshalb die Teile der Software, die sich auf die Ansteuerung des Eyetrackers, die Verarbeitung der Signale und die Nutzarmachung dieser Signale in einer eigenen Benutzeroberfläche beziehen, veröffentlichen. Ebenso werden die Kommunikationsschnittstelle zu einem externen Microcontroller veröffentlicht.

Die direkte Anbindung an den Rollstuhl über die proprietären Rollstuhl-Steuerprotokolle R-Net und DX-Bus sowie über einen mechanischen Adapter sehen wir als entscheidend für die Klassifizierung als Medizinprodukt an. Diese werden demnach nicht veröffentlicht.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

EyeCaptain soll natürlich möglichst vielen Leuten erreichen. Geplant ist, dass wir eine Plattform etablieren, die zunächst unsere, später ggf. auch auch andere freie technische Hilfsmittel dokumentiert, erklärt und nutzbar macht. Außerdem arbeiten wir an Konzepten für Schulungen von Multiplikator:innen, Alltagsbegleitende, Mitarbeitende von Wohlfahrtsverbänden, Ehrenamtliche und natürlich die Nutzer:innen selbst.

Informationen zu unserem Projekt EyeCaptain gibt es bislang durch unsere Social Media Kanäle und die Website. Unsere entwickelte Software stellen wir auf GitHub zur Verfügung (siehe hierzu <https://github.com/EyeCaptainProject>).

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die rechtlichen Hürden haben uns vor unerwartete Herausforderungen gestellt. Die intensive Auseinandersetzung mit dem Medizinproduktegesetz verbunden mit der Recherche nach passenden Ansprechpersonen im Bereich Open Source und Medizinprodukte war in dieser Form von uns vorab nicht kalkuliert. Da jedoch die Testung des Prototypen mit Nutzer:innen aufgrund des Infektionsgeschehens COVID-19 nicht in der ursprünglich geplanten Form möglich war, konnten wir diese gewonnene Zeit gut in die Auseinandersetzung mit den Rechtsfragen investieren.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab keine konkreten Entwicklungen von Dritten, die unsere Arbeit oder Zielsetzung beeinflusst haben.

Hundred Eyes – Werkzeuge für vertrauensbildende Maßnahmen

“Software Sprint” Schlussbericht

Zuwendungsempfänger: Hundert Augen Csuraji Vicari GbR

Astrid Csuraji und Dr. Jakob Vicari

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem **Förderkennzeichen 01IS19S32** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Kurze Darstellung der Aufgabenstellung und Motivation

Die Lage im Lokaljournalismus in Deutschland ist dramatisch: Abonnenten sterben, neue Leser*innen kommen kaum hinzu, viele Verlage stecken fest in ihren Digitalisierungsprozessen. Die Reichweiten sinken, damit sinken auch die Anzeigenerlöse. Weniger Budget bedeutet Personalkürzungen, wachsenden Zeitdruck in der Redaktion und damit verbunden sinkende Qualität in der Berichterstattung. In den USA haben nach Zahlen des Poynter-Instituts über 50 lokale Newsrooms in der Corona-Pandemie aufgegeben. Für Deutschland gibt es noch keine Zahlen. Fest steht: Es ist eine elende Abwärtsspirale.

Dabei zeigt gerade die Pandemie, dass wir Lokaljournalismus dringend brauchen: Als Bürger*innen wollen wir wissen, was in unserer Stadt und Region passiert, wer welche Entscheidungen warum trifft und Politik und Verwaltung zur Rechenschaft ziehen können. Wir wollen informiert über Vereine und Veranstaltungen, über Kultur, Sport und Freizeitaktivitäten.

Ohne Lokaljournalismus fehlt uns die Grundlage zur politischen Mitbestimmung und zur Teilhabe an unserer Gesellschaft.

Ohne Umdenken in Redaktionen und Verlagen wird der Lokaljournalismus trotzdem keinen Bestand haben, so sehr er auch gebraucht wird. Davon sind wir überzeugt. Redaktionen sollten nicht nur Sender sein, sie müssen auch Empfänger werden. Dazu braucht es eine stabile Verbindung zwischen Newsroom und Publikum. ‘Hundred Eyes’ soll genau das sein: ein tech-basiertes Hundert-Augen-Prinzip, das Journalist*innen und Communities einer Stadt wieder neu miteinander verbindet. Damit Informationen künftig besser in beide Richtungen fließen und partizipativer Journalismus ermöglicht wird.

Wie wir gearbeitet haben

Zum Auftakt von ‘Hundred Eyes’ haben wir einen Design Sprint veranstaltet. Die Methode wurde von Google zur Produktentwicklung etabliert und von uns mehrfach bei Projekten erfolgreich eingesetzt. Bei unserem Sprint haben wir zusammen mit anderen Journalist*innen und Entwickler*innen einen Papierprototypen gebaut und mit verschiedenen potenziellen Nutzer*innen getestet. Außerdem haben wir uns Expert*innen eingeladen und interviewt. Darunter einen 75-jährigen treuen Zeitungsleser und eine 16-jährige Fridays-for-Future-Aktivistin, die niemals die Zeitung lesen würde, um zu erfahren, wie mögliche Community-Mitglieder mit Redaktionen interagieren wollen.

Anschließend haben wir uns einen Entwicklungsfahrplan gegeben (Meilensteine) und unsere Anforderungen in Form von User Stories auf Github formuliert. Einzelne Programmieraufgaben haben wir in Issues aufgeschrieben und auch befreundete Entwickler*innen kommentieren und reviewen lassen. Open Source rules!

Die Entwicklung von ‘Hundred Eyes’ in 4 Meilensteinen

MILESTONE 1: Ideenentwicklung, Papier-Prototyp, Test (bis Ende 2. Monat)

MILESTONE 2: Funktionierender Prototyp, Software steht (bis Ende 3. Monat)

MILESTONE 3: ~~Einwöchiger Test in einer Lokalredaktion abgeschlossen~~ Test in einem eigenen Redaktionsbetrieb auf Riffreporter.de (s.u.) (bis Ende 5. Monat)

MILESTONE 4: Verbesserungen nach Test-Feedback (bis Ende 6. Monat)

Beitrag von ‘Hundred Eyes’ zu den Zielen der Förderinitiative „Software-Sprint“

Wir haben ‘Hundred Eyes’ als Open Source Software entwickelt, an der sich weitere Entwickler*innen auf Github beteiligen können. Mit jeder Crowdrecherche, für die die Software eingesetzt wird, soll auch das Tool immer besser werden. ‘Hundred Eyes’ sollte ein Werkzeugkasten für vertrauensbildende Maßnahmen werden – und genau das ist es: ein unkonventionelles Tool, das für mehr Vertrauen in den Journalismus sorgen und Menschen zur Mitarbeit und Kollaboration ermutigen kann. Ein Tool für die Allgemeinheit.

Die “Förderinitiative Software-Sprint” will freie Programmier*innen fördern. Für freie Kreative wie uns sind Initiativen wie diese kleine Inseln der Glückseligkeit.

Ausführliche Darstellung der Ergebnisse

Wir haben eine webbasierte Dialogsoftware gebaut. Sie ermöglicht den 1:1 Dialog mit Vielen. In der Software können Profile für Menschen aus verschiedenen Communitys angelegt werden. An diese ‘Hundred Eyes Community’ können Redakteur*innen Fragen stellen. Dafür steht ein einfach zu bedienendes UX zur Verfügung.

Die Community-Mitglieder erhalten diese Fragen entweder per eMail oder Telegram Messenger zugestellt – jeder hat die Wahl. Die Antworten aus Messenger und eMails werden zugeordnet und in einfach zu lesende Nachrichtenverläufe strukturiert. Die Redakteur*innen können Nachfragen an einzelne Community-Mitglieder stellen, gute Antworten markieren und in ihr Redaktionssystem kopieren. Die Software ist durch ein Passwort vor unautorisiertem Zugriff geschützt.

Hundred Eyes ist eine Ruby-on-Rails-Anwendung und besteht aus dem Anwendungsserver sowie einem Background-Worker zum Empfangen von eingehenden Telegram-Nachrichten. Wir haben Ruby 2.7 und die Datenbank PostgreSQL in der Version 12+ eingesetzt. Zum Nachrichtenversand nutzen wir die Telegram-API und Sendgrid. Wir bieten auf Github einige Skripte an, die das Aufsetzen von Hundred Eyes auf einem Webserver von Uberspace erleichtern.

Im Laufe unserer Förderung konnten wir, trotz der Corona-Einschränkungen, alle vier Meilensteine wie geplant erreichen, auch wenn der ganze Prozess länger gedauert und deutlich mehr Kraft gekostet hat (s. wöchentliche Status-Updates an den Prototype Fund):

Meilenstein 1, die Ideenentwicklung, ein Papier-Prototyp und Test, waren nach zwei Monaten fertiggestellt. Darauf basierend haben wir den **Meilenstein 2** umgesetzt, das heißt einen funktionierenden Prototypen gebaut. Da sich aufgrund der Corona-Krise kein Test mit einer Lokalredaktion umsetzen ließ, haben wir **Meilenstein 3** durch ein eigenes Projekt erreicht. Die Riff Freie Medien gGmbH hat uns ermöglicht, das Projekt **#50survivors** durchzuführen, eine Dialog-Recherche mit Corona-Überlebenden, bei der wir das ‘Hundred Eyes’ etwa sechs Wochen lang selber testen konnten. Das war aufwändiger als ein einwöchiger Test mit einer Lokalredaktion, aber auch sehr viel aufschlussreicher, sodass sich der Mehraufwand in jedem Fall gelohnt hat. Das Ergebnis haben wir als Koralle bei der journalistischen Genossenschaft der Riffreporter veröffentlicht:

www.riffreporter.de/50-survivors.

Während des mehrwöchigen Tests sind vor allem Schwierigkeiten mit dem eMail-Versand aufgefallen, die wir im **Meilenstein 4** nach dem Test-Feedback beheben konnten.

Zusätzliche Erkenntnisse aus der Projektarbeit und Begleitung durch die Open Knowledge Foundation

Die Begleitung durch die Open Knowledge Foundation war sehr engmaschig. Wir haben gelernt, ein Projekt in sinnvolle Teilschritte aufzudröseln. Das Projekt hat uns die Wichtigkeit von regelmäßigen Usertests wieder vor Augen geführt. Die engmaschige Begleitung hat viele Vorteile, vor allem weil man sich selber (jede Woche) hinterfragen muss, ehe man ein Status-Update verschickt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Vielfältiger Journalismus dient dem Gemeinwohl. Unsere primäre Zielgruppe sind Lokalredaktionen, die neuen Draht zu den Menschen in ihrer Stadt oder Region suchen. Unser Tool richtet sich aber auch an begeisterte und enttäuschte Leser*innen von lokalen Medien aus allen Schichten, die ein hohes Informationsbedürfnis haben. Oder Menschen, die zu einem bestimmten Recherchethema qualifiziert Auskunft geben können.

Wir sind überzeugt, dass lokale Communities ihre unterschiedlichen Perspektiven durch technologisch gestützten Dialog leichter in Redaktionen einbringen können. Das wird Diversität in der Berichterstattung verbessern. Außerdem können Redakteur*innen mit 'Hundred Eyes' ihre Arbeit stärker an den Nutzer*innen orientieren und nutzerzentrierte Angebote entwickeln, deren Akzeptanz den Lokaljournalismus wieder stärken werden.

Weitergehenden Effekte aus der Open-Source-Stellung

Die ökonomische Ausstattung vieler Lokalredaktionen ist ungenügend. Eine Open-Source-Lösung kann kostengünstig eingesetzt werden. Jede*r Teilnehmende kann sich von der Sicherheit der Software selbst überzeugen. Und mit jeder Crowdrecherche wird 'Hundred Eyes' besser werden. Viele Köpfe sorgen nicht nur für besseren Journalismus, sie ermöglichen auch bessere Softwarelösungen für unsere dringenden gesellschaftlichen Probleme.

Zu Entwicklungszwecken können Redaktionen Hundred Eyes mithilfe von Docker leicht lokal aufsetzen. Eine Anleitung dazu haben wir auf Github veröffentlicht.

Hundred Eyes ist ein erster Schritt der Übertragung des Open-Source-Gedankens auf die journalistische Arbeit.

Ideen und Pläne für die Weiterentwicklung

Aufgrund des sehr guten Feedbacks wollen wir 'Hundred Eyes' weiterentwickeln und weitere journalistische Projekte durchführen und anderen zeigen, wie sie die Software benutzen können.

Die Entwicklung von 'Hundred Eyes' hat uns gelehrt, wie wir ein komplexes Projekt aufsetzen, bei dem wir zum Kern eines Problems vordringen und genau dort mit der Entwicklung beginnen. Wir können nun als Berater*in den Einsatz der Dialogsoftware anbieten und Redaktionen besser in ihren Change-Prozessen hin zu nutzerzentrierter Produktentwicklung unterstützen.

Arbeiten, die zu keiner Lösung geführt haben

Wir haben alles versucht, aber es war nix zu machen: Der meistgenutzte Messenger "Whatsapp" ließ sich nicht an 'Hundred Eyes' anschließen. Auch die Idee, ihn mittels Broadcast-Listen einzubetten, musste verworfen werden. Der Grund ist die strikte Policy von Whatsapp gegenüber dem Massenversand von Nachrichten. Wir haben uns daher dazu entschieden, die Messenger-Alternative Telegram anzuschließen.

Präsentationsmöglichkeiten für mögliche Nutzer

Unsere Projektergebnisse kann man hier einsehen:

Website: <https://tactile.news/journalismus-im-dialog/100-eyes/>

Github: <https://github.com/tactilenews/100eyes/>

Beispiel-Use-Case: <https://riffreporter.de/50-survivors>

Einhaltung der Arbeits- und Kostenplanung

Da sich aufgrund der Corona-Krise kein Software-Test in einer Lokalredaktion ergeben hat, haben wir **Meilenstein 3** durch die Aufsetzung eines eigenen Projekts erreicht. Dies bedeutete erheblichen Mehraufwand. Durch eine inhaltliche Förderung der Riff Freie Medien gGmbH konnten wir das Projekt **#50survivors** durchführen, eine Dialog-Recherche mit Corona-Überlebenden, bei der wir das Tool etwa sechs Wochen lang im Team mit vier Journalist*innen getestet haben.

Etwaige Ergebnisse bei anderen Stellen

Andere Personen oder Institutionen hatten keinen Einfluss auf unsere Arbeit und die Zielsetzung.

**Wir danken der Open Knowledge Foundation Germany,
insbesondere den Mitarbeiter*innen des Prototype Fund und
des Projektträgers DLR, für die professionelle Begleitung und
freundliche Beratung während des Projektverlaufs.**



Richtlinie zum „Software-Sprint“

decidim – Decidim Berlin - Infrastruktur zur Zivilgesellschaftlichen Interessenvertretung

Schlussbericht

Zuwendungsempfänger: decidim GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S33 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Mit dem Projekt Decidim Berlin wollten wir verlorenem Vertrauen in partizipative Online-Verfahren und der Fragmentierung gemeinwohlorientierter Initiativen entgegenwirken. Das bei Initiativen entstandene Misstrauen ist meist in der fehlenden politischen Verbindlichkeit bei Partizipationsprozessen begründet. Zudem kommen technisch unausgereifte Systeme zum Einsatz, die einen Beteiligungsprozess nur unzureichend abbilden können und der Willensbildung nicht dienlich sind.

Auf der anderen Seite bestand die Arbeitshypothese des Projekts Decidim Berlin in der Feststellung, dass politische Initiativen in Berlin einen hohen Grad an Fragmentierung in ihrer politischen Arbeit aufweisen und so nur punktuell politische Kollektivität umsetzen. Die Vorgehensweise zur Problemlösung des Projektes Decidim Berlin bestand daher in der Überführung der best-practice Plattform Decidim in die stadtpolitische Initiativenlandschaft.

Meilensteine:

- Analyse zu Bedarfen stadtpolitischer Initiativen
- Community Building, Netzwerk-Arbeit und Einbeziehen relevanter Akteur_innen
- Testen der Plattform und Umsetzung
- Launch / Presse

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

Die Zielgruppe sind Initiativen, die einen gemeinwohlorientierten und commons-verbundenen Anspruch verfolgen. Stadtpolitischen Initiativen fehlt zur Aushandlung ihrer Positionen und der Konsolidierung ihrer Interessen eine effiziente Koordinationsstruktur, diese wollte wir mit Decidim

schaffen. Ziel war auch, zu verstehen, was den Initiativen weiterhin an digitalen Tools fehlt und ob sie verlorenes Vertrauen in partizipative Verfahren durch eigene partizipative Tools, also Beteiligung von unten, zurückgewinnen können. Dies wollten wir durch die Begleitung und Kuration eines Stadtpolitischen Festivals “von unten” umsetzen.

Ausführliche Darstellung der Ergebnisse

Im Rahmen des Projektes wurde zunächst eine Bedarfsanalyse für die Übertragung von Decidim nach Berlin konzipiert, die sich auf die folgenden Fragestellungen konzentrierte:

Wie kommunizieren stadtpolitische Initiativen intern? Welche Instrumente zur internen Organisation und Kommunikation werden benutzt? Welche Bedarfe nach Vernetzung mit anderen Initiativen bestehen? Wann gibt es Momente politischer Kollektivität? Welche Instrumente der politischen Einflussnahme und Beteiligung werden von den Initiativen eingesetzt? Dabei stand die Frage nach der Leistbarkeit von Decidim im Fokus. Zu diesem Zweck wurden qualitative Interviews mit sieben Schlüsselakteur*innen von bekannten stadtpolitischen Initiativen und Organisationen sowie fünf Expert*innen aus Politik und Verwaltung geführt. Die Interviews dauerten durchschnittlich ca. eine Stunde und wurden aufgezeichnet und transkribiert. Anschließend folgte eine Systematisierung und Kategorisierung der Antworten, um eine Inhaltsanalyse zu vereinfachen. Eine ausführliche Darstellung der Ergebnisse haben wir im Rahmen unserer Bedarfsanalyse verfügbar gemacht:

<https://demoweeek.prototypefund.de/projects/01-decidim.berlin.html>

Parallel dazu wurde Decidim in einigen Gruppen prototypisch ausgetestet:

Eine offene Gruppe politisch engagierter Menschen hat sich im Frühjahr - vor Corona - mit der Idee getroffen, ein neues “stadtpolitisches Festival” nach dem Vorbild des Berliner urbanize! Festival 2018 auf den Weg zu bringen. Decidim soll es Initiativen und interessierten Gruppen ermöglichen für das Festival ihre eigenen Veranstaltungen und Aktionen über die Plattform einzubringen und so das Festivalprogramm partizipativ zu gestalten.

Im Rahmen des inhaltlichen Ablegers **“United We Talk”** des Formats “United We Stream” haben wir gemeinsam mit vielen anderen Aktiven eine Talksendung zum Thema “Stadt für Alle?” auf den Weg gebracht.

Die Initiative für ein Volksbegehren **„Deutsche Wohnen & Co. enteignen“** mobilisiert für den Vorschlag, die Wohnungsbestände großer Wohnungsunternehmen (mit mehr als 3.000 Wohnungen) zum Zwecke der Sozialisierung zu enteignen und in die Verwaltung einer Anstalt öffentlichen Rechts zu überführen. Die Arbeitsgruppe nutzt Decidim mit Unterstützung des Decidim-Team zur Koordination des internen Arbeitsprozesses.

Die Plattform wurde aus dem katalanischen bzw englischen ins deutsche übersetzt. Der Source Code wurde in einem github Repository offen zur Verfügung gestellt. Es wurde außerdem eine Installationsanleitung und eine Anleitung zur weiteren Übersetzung der Plattform geschrieben. Alles ist unter https://github.com/decidim-de/decidim_berlin abrufbar. Die Plattform ist unter <https://decidim.de> und verschiedenen Subdomains wie <https://urban.decidim.de> erreichbar. Jede Initiative bzw. jedes Projekt hat eine eigene Instanz samt eigener Subdomain zum Testen bekommen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Ein für das Jahr 2020 geplantes stadtweites politisches Festival konnte wegen der Corona-Pandemie nicht stattfinden. Eine der möglichen Perspektiven ist das Festival in 2021 in Teilen digital stattfinden zu lassen und dafür die Plattform decidim zu nutzen, in Vorbereitung eines dann in 2022 größer stattfindendem Hybrid-Format.

Im Rahmen der Umsetzung des Projektes verstetigte sich außerdem die Arbeit mit einzelnen Akteuren, die für sich decidim als passendes Tool identifiziert haben und nutzen wollen. Hierzu zählt u.a. das Initiativenforum in Berlin und eine Initiative zur partizipativen Entwicklung der Tempelhofer Flughafengebäude.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Zu Beginn der Laufzeit begann die Corona-Pandemie sich auch in Deutschland auszubreiten. Durch die umfangreichen Kontaktbeschränkungen, war es uns nicht möglich, analoge Treffen mit Initiativen zu veranstalten. Ein Wechsel zwischen Online- und Offline-Veranstaltungen zur gemeinsamen Entwicklung der Plattform war erklärtes Ziel. Dies musste komplett online passieren und hat die gemeinsame Entwicklung gehemmt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Plattform ist unter <https://decidim.de> verfügbar. Der Software-Code ist unter offener Lizenz unter <https://github.com/decidim-de> abrufbar. Eine selbst erstellte Bedarfsanalyse unter Berliner Initiativen haben wir unter <https://demoweb.prototypefund.de/projects/01-decidim.berlin.html> veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Kostenplanung konnte insgesamt sehr gut eingehalten werden, trotz veränderter externer Faktoren wie der Corona-Pandemie.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Die Aufmerksamkeit für decidim ist insgesamt sehr hoch und wir bekommen laufend Anfragen. Die Konferenz "Offene Stadt Hamburg" hat uns für November 2020 eingeladen einen Workshop zu decidim zu halten. Wir werden im November außerdem einen Beitrag zu einer Publikation zu Partizipation der Stiftung Mitarbeit beitragen. Im Oktober 2020 sind außerdem auf der "Living the City"-Konferenz in Berlin mit einem Workshop eingeladen. Im Dezember werden wir für die HafenCity Hamburg ein Seminar mitgestalten.

*Recovery Cat – Blick in die Black Box von Patient*innen mit psychischen Störungen*

Schlussbericht

Zuwendungsempfänger:

Alissa Rohrbach, Alexey Rodriguez Yakushev, Andreas Pittrich

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S34 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Wir haben vor der Bewerbung beim Prototype Fund viele Gespräche mit Psychiater*innen und Patient*innen geführt um herauszufinden, wo die größten Schmerzpunkte in der Behandlung von psychischen Störungen liegen. Dabei wurde schnell deutlich, dass wir uns auf die Zeit der ca. 4-6 Wochen zwischen Therapiebehandlungen fokussieren sollten. In dieser "Blackbox" gehen viele Daten verloren, die für den Behandlungserfolg von großer Relevanz sind. Patient*innen haben Nebenwirkungen, Symptome schwanken gerade zu Behandlungsbeginn stark, es gibt einige "Non Responder" (die gar nicht auf die Medikation reagieren) und sogar suizidale Gedanken. Patient*innen fühlen sich allein gelassen und für Psychiater*innen ist sehr schwierig, in den kurzen Behandlungsterminen (ca. 15 Minuten) die letzten Wochen so zu rekonstruieren, dass die Therapie optimal angepasst werden kann.

Durch ein individualisierbares Self Reporting der Zeit zwischen den Behandlungen kann diese Blackbox aufgehoben werden und ein Fortschritt in der patientenzentrierten und digitalen Psychiatrie geschaffen werden. Dabei war es für uns wichtig, dieses Self Reporting nicht selbst vorzugeben, sondern durch eine einfache App behandelnde Ärzt*innen und Patient*innen zu befähigen, eigene Follow-Ups innerhalb weniger Klicks zu erstellen.

Die wichtigsten Schritte zur Umsetzung von Recovery Cat während des Prototype Funds umfassten:

- Entwicklung der Fragebogen Elemente zur Erhebung von Symptomen, Nebenwirkungen etc.
- Kooperation mit Forschungs- und Behandlungseinrichtungen (Charité und Theodor-Wenzel-Werk) zur klinischen Begleitung

- Nutzerinterviews - und Tests um den Wert der App sicherzustellen
- Erstes, einfaches Frontend für Patient*innen mit den wichtigsten Features (Dateneingabe, Darstellung des Symptomverlaufs im einfachen Histogramm)
- Erstes, einfaches Frontend für Ärzt*innen mit den wichtigsten Features (Zusammenstellung der Tracking Fragen, Übersicht über Patient*innen mit aktuellen Krankheitsverläufen)

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

Unsere Zielgruppen sind Psychiater*innen, Psychotherapeut*innen, Patient*innen in psychologischer/psychiatrischer Behandlung sowie Wissenschaftler*innen. Durch Recovery Cat profitieren diese Zielgruppen von einer komplett individualisierbaren und datengeschützten Möglichkeit, die Lücke zwischen Behandlungsterminen zu schließen. Durch geringfügigen Zusatzaufwand (wenige Minuten um einen Fragebogen zu erstellen, wenige Sekunden am Tag um die Fragen zu beantworten auf der Smartphone App) können für die Behandlung äußerst relevante Zusatzinformationen generiert werden. Das Ziel ist, für Patient*innen schneller die richtige Behandlung zu finden, und bei Komplikationen früher einzugreifen. Dies sind zwei große Schwierigkeiten innerhalb der Behandlung von psychischen Erkrankungen. Mit Recovery Cat wollen wir auch das Vertrauen in digitale Tools im gesellschaftlichen bzw. speziell medizinischen Kontext stärken. Unser Fokus auf Open Source und Data Privacy soll ein sicheres und transparentes Self-Reporting von sensiblen Daten ermöglichen und eine Weiterentwicklung in anderen Bereichen über die Psychiatrie hinaus durch weitere Entwicklerteams. Da ein weiterer Anwendungsfall die Bereitstellung von klinischen Alltagsdaten ist (hierzu werden User speziell gefragt), sollen Silos entgegengewirkt werden.

Ausführliche Darstellung der Ergebnisse

Grundsätzlich haben wir unsere Ziel erreicht, da wir einen funktionierenden technischen Prototyp entwickeln konnten, welcher auf Zusammenarbeit mit Nutzer*innen beruht. Allerdings ist die einfache Nutzung noch nicht ganz gewährleistet und muss noch verbessert werden (UX/UI bzw. allgemein Design). Bisher ist die Nutzung zwar theoretisch möglich, aber nicht sehr nutzerfreundlich.

Zudem haben wir uns ein starkes Netzwerk von Expert*innen und kooperierenden Einrichtungen aufgebaut (Charité, Theodor-Wenzel-Werk), unser Team vervollständigt und eine größere Förderung eingeholt.

Meilensteine mit einer kurzen Erläuterung der Ergebnisse:

- *Entwicklung der Fragebogen Elemente zur Erhebung von Symptomen, Nebenwirkungen etc.:*
 - die grobe Struktur für das Anlegen von Fragen sowie versch. Antwortformate (z.B. Skala, kategorial etc.) wurde entwickelt
- *Kooperation mit Forschungs- und Behandlungseinrichtungen (Charité und Theodor-Wenzel-Werk) zur klinischen Begleitung:*

- wir sind im Charité Accelerator (BIH), mit dem Theodor-Wenzel-Werk haben wir eine informelle Kooperation, die durch einen Pitch vor der Klinikleitung freigegeben wurde. Wir dürfen mit beiden Einrichtungen testen und bekommen zusätzlich inhaltlichen Support
- *Nutzerinterviews - und Tests um den Wert der App sicherzustellen:*
 - *wir haben durchgängig mit Nutzer*innen getestet (durch Interviews und Click-Dummies) bzw. inzwischen Nutzer*innen in unserem Team*
- *Erstes, einfaches Frontend für Patient*innen mit den wichtigsten Features (Dateneingabe, Darstellung des Symptomverlaufs im einfachen Histogramm)*
 - Patient*innen bekommen einen Reminder und beantworten Fragen. Daten werden in einer einfachen grafischen Übersicht gezeigt
- *Erstes, einfaches Frontend für Ärzt*innen mit den wichtigsten Features (Zusammenstellung der Fragebögen, Übersicht der Rückmeldungen mit aktuellen Krankheitsverläufen)*
 - Auch diese Feature wurden programmiert, wobei die UI/UX bisher noch recht kompliziert ist und wir den User Flow noch stark verbessern müssen, damit es für jeden anwendbar ist

Zusätzliche Erkenntnisse waren, dass es wichtig ist, sich von Anfang an zu überlegen, wie die Lösung den Nutzer erreicht. Auch wenn das Produkt im Kern erstmal Open Source und nicht-kommerziell ist, sollten Wege geschaffen werden, das Potential für Nutzer*innen wirklich auszuschöpfen. Dazu gehört eine langfristige Weiterentwicklung, wofür finanzielle Mittel da sein müssen. Deshalb sind wir froh, im nachfolgenden Accelerator der Charité verschiedene Geschäftsmodelle zu validieren.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Der Nutzen für die Zielgruppen wurde oben bereits mehrfach beschrieben. Wenn wir Recovery Cat bekannt und nutzbar machen können, können wir die Behandlung von psychisch erkrankten Menschen verbessern, ohne dabei die Privatsphäre zu gefährden.

Dank der freien Lizenz können Dritte unser Produkt erweitern und nutzbar für Bereiche machen, die wir durch die Fokussierung auf die Psychiatrie nicht mehr in den Blick genommen haben. Ebenso können Dritte neben uns die Integration in IT-Systeme von Ärzten und Krankenhäusern leisten und so zur Verbreitung und Interoperabilität von Recovery Cat beitragen. Und schließlich trägt die Offenheit des Sourcecodes dazu bei, dass Vertrauen erzeugt und erhalten bleibt, da der Umgang der App mit den Daten nach außen transparent und grundsätzlich für alle nachvollziehbar ist.

Die Förderung durch den Prototype Fund hat unsere berufliche Laufbahn geprägt. Aus der zunächst losen Idee wurde dadurch ein anfassbares Produkt und ein Netzwerk aus Unterstützer*innen, so dass wir nun auf dem Weg zur Unternehmensgründung sind. Wir haben eine größere Anschlussfinanzierung durch die Charité bekommen (BIH Accelerator) und können dadurch weitere Programmierer und Designer finanzieren. Neben der finanziellen Förderung durch den Prototype Fund, der die Freistellung von unseren bisherigen Tätigkeiten für uns möglich machte, waren wir sehr stark durch die Erwartungshaltung geprägt. Durch die Deadlines haben wir einen Push bekommen,

Recovery Cat wirklich zu entwickeln, was wir als Nebenprojekt evt. sonst nicht verwirklicht hätten. Auch die Coachings (vor allem zu Design) haben uns weitergeholfen. Die nächsten Monate und evt. Jahre werden wir nun an Recovery Cat arbeiten, und der Startschuss dafür war ganz klar der Prototype Fund. Dafür sind wir sehr dankbar.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Während wir am Anfang noch einen relativ breiten Anwendungsfall geplant hatten (über die Psychiatrie hinaus, z.B. auch für Hausärzte) haben wir schnell gemerkt, dass wir hier einen ersten Fokus brauchen. Auch der Fokus auf die Bereitstellung von Daten für die Forschung wurde erstmal verschoben. Uns war es wichtig, ein erstes funktionierendes, sinnvolles Produkt zu kreieren. In den nächsten Monaten und Jahren werden wir die Funktionalitäten und Zielgruppen weiter ausweiten. Die Coachings des Prototype Funds haben uns dabei geholfen, spezifischer zu werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

<http://www.recoverycat.de>

<https://gitlab.com/recoverycat>

E-Mail: recoverycat@posteo.de

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wir haben insgesamt etwas mehr gearbeitet, da der Prototyp doch aufwändiger war und wir viel Zeit mit der Arbeit mit Nutzer*innen verbracht haben, um wirklich die Schmerzpunkte zu verstehen. Außerdem haben wir im Projekt deutlich weniger auf Daten fokussiert als gedacht, weshalb die Verteilung der Stunden innerhalb des Teams abgewichen ist bzw. nicht gleich verteilt war (unser Data Scientist hatte weniger zu tun), was aber für alle in Ordnung war.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Wir hatten das große Glück, innerhalb der Laufzeit des Prototype Funds eine größere Anschlussförderung zu bekommen. Durch erfolgreiches Pitchen sind wir ab Oktober 2020 im Accelerator der Charité (BIH), können unser Team deutlich erweitern (wir sind inzwischen sieben Teammitglieder), weitere Entwickler und Designer bezahlen und Mentoring bekommen. Da wir hierfür bereits im Juni die Zusage bekamen, hatten wir zusätzliche Erwartungen. Es war umso wichtiger, bis September einen funktionierenden Prototyp zu entwickeln. Diese Ziele stimmten aber mit denen des Prototyp Funds überein. Wir freuen uns sehr, weiterhin und verstärkt an Recovery Cat zu arbeiten und erwarten, eine App zu gestalten, die wirklichen Nutzen stiften, viele User hat und auch langfristig weiterentwickelt wird.

Schlussbericht:

RIOT: Rapid Indexing of (Network) Traffic

Suchindex für Netzwerkverkehre

Alexander Wamack

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S35 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Nach wie vor sind Computer-Netzwerke das Ziel von Cyber-Angriffen. Wie kann man sich oder andere davor schützen? In einer idealen Welt wäre das Mittel der Wahl natürlich, alle Sicherheitskonzepte und Richtlinien konsequent umzusetzen. Die Realität aber ist eine andere: große, historisch gewachsene und unter Umständen schlecht dokumentierte Netzwerk-Topologien können nicht ohne Einschnitte des Betriebs auf einen entsprechenden Sicherheitsstandard gebracht werden. Schlimmer noch: Wie kann man sich gegen noch nicht *öffentlich bekannte*

Sicherheitslücken – wie beispielsweise CVE-2020-1350 – schützen? Die Antwort ist erschreckend: Erst einmal überhaupt nicht. Man kann jedoch versuchen, durch gezieltes Monitoring des Netzwerk-Perimeters und schnelles Handeln im Ernstfall nach einem erfolgreichen Angriff dessen Auswirkung einschätzen und Gegenmaßnahmen einleiten. Hier fügt sich *NYGMA* mit dem im Rahmen des Prototypenfund implementierten Erweiterung *RIOT* in die Liste der Hilfswerkzeuge ein.

Gängige Lösugen aus dem Network Security Monitoring (NSM) und Intrusion Detection (IDS) Bereich erkennen über Muster und Anomalien Angriffe in Netzwerken. Um auf diese angemessen reagieren zu können müssen so viel Informationen wie möglich zur Verfügung stehen. Die Observation des Angriffs bedeutet hier meist eine erfolgreiche Infiltration der Infrastruktur des Opfers die zeitlich u.U. bereits in der Vergangenheit liegt und anderen noch nicht bekannten Mustern gefolgt ist. *RIOT* stellt eine zuverlässige Datenbasis für forensische Post-Incident Analysen wie Retro-Hunting und Ereignis-Assessment zur Verfügung.

Die am weitesten verbreiteten Open-Source-Software (OSS)-Lösungen *Suricata* für regelbasierte Warnmeldungen und *Zeek* für Metadatenerfassung stellen zur Zeit die Grundlage für ein OSS-basiertes Netzwerk-Sicherheits-Monitoring dar.

Beide Systeme bieten allerdings nicht – oder nur sehr eingeschränkt – die Möglichkeit, auf Rohmaterial zurückzugreifen. Auch eine Funktion für das wiederholte und schnelle Beantworten von Suchanfragen bietet keine der beiden Lösungen. Besonders im Falle von reinen Warnmeldungen fehlen dem*der Analyst*in wichtige Kontextinformationen, um die Relevanz (false-positive?) und Kritikalität (true-positive) eines Ereignisses schnell und korrekt beurteilen zu können. Hier setzt *RIOT* an: *RIOT* ergänzt *Suricata* und *Zeek* um die

Möglichkeit, aus dem bestehenden Datenbestand Informationen im Rohformat gezielt abfragen zu können und weitere Erkenntnisse zu gewinnen oder tiefer gehende Analysen zu ermöglichen.

Das Projekt wurde in der Programmiersprache C++20 implementiert. Die Projektplanung erfolgte in Absprache mit dem Prototype Fund jeweils wöchentlich eine Woche im Voraus sowie mit einem kurzen Rückblick der vergangenen Woche.

Die Implementierung selbst erfolgte angelehnt an ein Test-Driven-Development: Für jede wichtige Funktionalität wurden einzeln ausführbare Unit-Tests implementiert. Die Tests wurden zusammen mit der tatsächlichen Implementierung entworfen.

Zusätzlich wurden auch Test-Treiber für ein Fuzzing der Komponenten bereitgestellt. Die Treiber bzw. Fuzzer basieren auf `libFuzzer` und `honggfuzz`. Durch die frühe Integration der Fuzzing-Komponenten in Kombination mit den Unit-Tests konnte parallel zur Entwicklung eine QA (Quality Assurance) des C++-Codes erfolgen.

Die wichtigsten bearbeiteten Meilensteine waren hierbei:

- Evaluierung der Primär Index Datenstruktur (B+Trees und RedBlack Trees)
- Implementierung / Integration Unit-Test Framework
- Entwurf Dateiformat für Indexstruktur
 - Serialisierung der Indexstruktur
 - Komprimierung der Indexstruktur (Integer Compression)
 - Integration von SIMD Integer Compression Bibliotheken
 - Implementierung Fuzzing-Tests
- Erweiterung des PCAP-Parsers
 - Fehlerbehebung
 - Unit-Tests
 - Erweiterung TCPO/UDP IPV4/IPV6 anhand der Test Daten
 - Implementierung Fuzzing-Tests
 - Erweiterung DNS Parser und Fehlerbehebung
- PCAP slicing und reassembly
 - Unit-Tests
- Design Abfragesprache
- Design der Query (Abfrage) - Engine (API)
 - Implementierung Fuzzing-Tests
- Integration Intel-Hyperscan für Regexp-Index
 - Support für die Abfragesprache
 - Sprache zum Einlesen von Mustern bzw. Indicators of Compromise (IOC)
 - Anwenden von Mustern auf DNS-Namen
- Beispiele und Dokumentation
 - Befehlszeilen Tool zum indizieren nach IP-Adressen und Ports
 - Befehlszeilen Tool zum indizieren nach Mustern
 - Grafische Benutzeroberfläche für die Demoweeek
- Evaluierung Fuzzing und Fehlerbehebung
- Benchmarking der Index-Algorithmen
 - hinsichtlich der Komprimierungsraten

- hinsichtlich der Geschwindigkeit
- Demo-App und Show-Case
 - Design UI
 - Aufbereitung von Testdaten
 - “Regexp-Index” Workflow
 - Videos für die DemoweeK
- Während der Planung ausgeschlossene Milestones
 - Spezialisiertes Format zum Speichern von NetzwerKdaten (CCAP)
 - Live Capture

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

Die Zielgruppe von NYGMA/RIOT sind Netzwerkadministrator*innen und Cyber-Security-Analyst*innen, die bereits mit **Suricata** oder **Zeek** arbeiten und eine Lösung für die Aufbereitung von Computer-NetzwerKdaten benötigen. Konkrete Anwendungsfälle reichen von der Fehlersuche in Netzwerken, leistungsorientierten Optimierungen oder forensischen Untersuchungen bis hin zu Analysen nach oder während eines erfolgreichen Angriffs (Incident-Response). Auch in Schulungen zu Netzwerken allgemein kann NYGMA/RIOT eingesetzt werden.

Die Stärke von NYGMA/RIOT liegt darin, dass es kein einzelnes Programm sondern eine Bibliothek ist die auf den konkreten Einsatzzweck – also das eigene Netzwerk – umfassend angepasst werden kann. Auch wird keinerlei Annahme über die Daten selbst getroffen. Zum Export der Daten werden Standardformate verwendet. Damit lässt sich die Software komfortabel integrieren.

Das Projekt verbindet den Schutz öffentlicher Infrastruktur durch eine Open-Source Lösung, Verbesserung der Tooling-Fähigkeiten bei Cyber-Defense Fragestellungen und den Schutz der Privatsphäre durch ein Basisframework für weitere Analysemöglichkeiten mit den Themen der Open-Knowledge-Foundation insb. des Prototypenfunds.

Ausführliche Darstellung der Ergebnisse

Im Rahmen der Prototypentwicklung lag der Focus ausschließlich auf der Implementierung der Index-Algorithmen um bereits bestehende Datenstände im Standard-PCAP Format durchsuchbar zu machen. Zusätzlich wurde für die Abschluss-Demonstrationen eine grafische Oberfläche implementiert, die den Einsatz aus Sicht einer forensischen Untersuchung ermöglicht.

Die zunächst geplante Entwicklung eines angepassten Speicherformats als auch die dazu notwendige Implementierung der Live-Erfassung von NetzwerKkarten wurde aufgrund des Umfangs des Projekts nach Rücksprache zurückgestellt.

Alle während der Projektzeit besprochenen Meilensteine wurden erreicht und erfolgreich abgeschlossen. Das kommandozeilen basierte Tool **ny** ermöglicht die Erstellung eines Suchindexes für vorhandene PCAP-Daten beliebiger Größe. Die Testdatensätze umfassen Einzeldateien mit einer Größe zwischen 50GB und 100GB je Datei. Das ebenfalls kommandozeilen basierte Tool **t3** erlaubt die Indizierung der Daten mittels selbst definierter Such-Muster unter Verwendung der Open-Source Software **Intel-Hyperscan**. Das Tool **the-owls** dient zur

grafischen Veranschaulichung der Index-Daten, zur Abfrage des Indexes und zum wiederholten Re-Analysieren des Datenbestandes mit dem Muster-Index durch `t3`.

Das Projekt wurde als Software-Bibliothek entwickelt, die beschriebenen Werkzeuge demonstrieren jeweils nur einen Teilaspekt des eigentlich möglichen Funktionsumfangs. Idealerweise wird die Verwendung auf die jeweilige Netzwerkstruktur angepasst. Hierin liegt auch die Stärke der implementierten Software: hohe Anpassbarkeit, keine komplizierten Software abhängigkeiten.

Für die Abfrage des Such-Indexes wurde eine kleine domänenspezifische Sprache implementiert die die normalen Menge-Operationen aus der Mathematik implementiert: Schnittmengenbildung, Vereinigungsmenge und Differenz. Die Bibliothek ermöglicht auch ein Abfrage des Invertierten-Indexes als auch eine spezielle Rückwärts-Suche, die es ermöglicht zusammengehörige oder ähnliche Daten auf einmal abzufragen. Diese Funktionalität ist nur in der Bibliothek enthalten und noch nicht über die Sprache selbst benutzbar.

Die Rohdaten wurden hierbei von folgenden öffentlichen Quellen benutzt:

- <https://www.netresec.com>
- <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>

Da RIOT für das Bearbeiten von großen Datenbeständen entwickelt wurde lag besonderer Schwerpunkt in der Komprimierung des Such-Indexes. Hierbei wird die zu analysierende Datei in virtuelle Segmente von je maximal 4GB unterteilt. Der Index selbst besteht aus einer Abbildung von Merkmalen wie IP-Adressen, TCP/UDP-Port Nummern oder beliebigen Benutzer definierten Identifikationsnummer zu Offset-Listen in die ursprüngliche Datei. Diese Offset-Listen eignen sich aufgrund ihres Aufbaus besonders gut für die Komprimierung mittels in der Literatur bezeichneten "Integer-Compression-Algorithms". Bei der Analyse der Testdaten konnte eine Größe per 4GB Segment Index Daten zwischen 25MB im komprimierten Fall und 60MB im unkomprimierten ermittelt werden.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Vision ist, durch die Kombination von *Suricata*, *Zeek* und *NYGMA/RIOT* die Cyber-Abwehrmöglichkeiten *für alle* zu verbessern. Da Monitoring-Lösungen im Herzen des eigenen Netzwerks platziert werden, ist es eigentlich nahezu unumgänglich, eine Open-Source-Lösung einzusetzen. Diese kann von unabhängigen Spezialist*innen eingesehen werden um sicherzustellen, dass keine Daten "abfließen" und die Software selbst gängigen Sicherheitsvorschriften genügt. Bei kommerziellen Black-Boxen ist ein (mehr oder weniger eingeschränkter) Fernzugriff oder Exfiltration nie auszuschließen, teilweise sogar verpflichtender Gegenstand eines Wartungs- oder Analysevertrags. Datenschutz- und Sicherheitsinteressen widersprechen dem selbstredend.

Beim Verarbeiten der Daten werden spezielle Befehle (SIMD: SSE und AVX) der Intel-X86 Prozessoren verwendet. Eine Portierung der Software für die 64Bit-ARM Architektur (aarch64/NEON) ist geplant.

Bei der Implementierung der Komprimierungsalgorithmen entstand die Idee für eine angepasste Methode mit höherem Komprimierungspotential. Die Umsetzung dieser Idee ist ein weiterer Meilenstein.

Im Rahmen der *Prototyp-Entwicklung* lag der Fokus auf forensischen Untersuchungen eines bestehenden Datensatzes. Dies ist besonders hilfreich, um bei der Entwicklung stets mit deterministischen Datenströmen arbeiten zu können. Als wichtige Erweiterung von NYGMA/RIOT steht die *Live-Datenverarbeitung* an oberster Stelle.

Mit Hilfe des *Live-Capturing* wird eine Lösung zum Domain-Name-System (DNS) Monitoring entwickelt, die basierend auf den Eigenschaften von NYGMA/RIOT als leichtgewichtiges Intrusion-Detection-System (IDS) eingesetzt werden kann.

Die Einarbeitung in die notwendigen Technologien während der Projektphase war mit erheblichen Lernkurven verbunden. Der Einsatz von Prozessor spezifischen SIMD Befehlen erfordert besondere Sorgfalt bei der Speicherverwaltung bzw des Speicherlayouts für die Komprimierungsverfahren. Leider waren viele Open-Source Projekte schlecht dokumentiert und die Implementierungen nicht ausreichend um Sie ohne “Reibungsverluste” zu verwenden. Viel Arbeit war notwendig um diese Bibliotheken einsetzen zu können. Auch die Wahl der Datenstruktur zum Erstellen des Indexes gestaltete sich aufwendig. Es wurden unterschiedliche **B+Tree** Implementierungen aus der Datenbanktechnik evaluiert und die Performanz der unter der für dieses Projekt zu erwartenden Work-Load abzuschätzen. Denn die Implementierungen aus der **C++** Standard-Template-Library (STL) waren immer langsamer und benötigten mehr Speicher als die Alternativen. Dies ist insbesondere für die nächste Ausbaustufe der Live-Analyse wichtig. Der Benutzer kann die Implementierung durch Templates in jedem Fall frei wählen und so dem Anwendungsfall besser anpassen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wie bereits anfänglich erwähnt wurde darauf verzichtet ein spezielles Format für die Indizierung der Roh-Daten einzuführen. Der Hintergrund dieser Entscheidung war, dass durch die Verbreitung von selbstkomprimierenden Dateisystemen wie **ZFS** es eigentlich nicht mehr notwendig ist ein neues Format zu entwickeln mit dem Hauptziel die Daten zu komprimieren. Bei bereits existierenden Datenbeständen kann man unter Umständen die Daten nicht einfach ein 2tes mal abspeichern bzw konvertieren da die Datenmenge an sich zu groß hierfür wäre denn zumindest kurzfristig würde ja die doppelte Speicherkapazität benötigt.

Eine Implementierung der Live-Erfassung (Capture) wurde ebenfalls aussen vor gelassen da hierfür nicht nur die Erfassungskomponente implementiert werden müsste sondern auch Betriebssystem und hardwareabhängige Faktoren beachtet werden müssten. Im Rahmen der Projektdauer wäre das schlicht weg nicht möglich gewesen.

Anstatt halbfertige Speicherformate und Erfassungs-Module zu entwickeln wurde der Fokus auf Test-Driven-Development gesetzt. So werden für jede Softwarekomponente Unit-Tests zur Verfügung gestellt. Weiterhin wurden Fuzzing-Module implementiert, die die einzelnen Softwarekomponenten einem automatisierten

und randomisierten Stresstest unterziehen können. Durch deren Einsatz schon frühzeitig während der Entwicklung konnten abweichende Testergebnisse ausgewertet und Fehler beseitigt werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Das Projekt kann über GitHub bezogen werden. Pull-Requests und der Issue-Tracker sind aktiviert. Das Projekt beinhaltet eine kurze Dokumentation sowie Demonstration wie es benutzt werden kann. Demonstationsvideos und ein Beispiel Workflow kann über die Prototypefund-DemoweeK Website erreicht werden. Der Link zur Prototypefund Seite wird noch in die Dokumentation mit aufgenommen.

- Projekt-Seite: <https://github.com/stackless-goto/nygma>
- DemoweeK-Präsentation: <https://demoweeK.prototypefund.de/projects/17-nygma.html>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die initial aufgestellte Kostenplanung konnte eingehalten werden. Bei der Projektplanung und Gestaltung der Meilensteine wurde diese durch wöchentliche Sprints bzw Sprint-Planung kontinuierlich überprüft. Kleinere Anpassungen bei der Zeitplanung konnten so frühzeitig erkannt und integriert werden. Größere Umplanungen oder Anpassungen waren nicht notwendig.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Entwicklung wurden Schwachstellen wie CVE-2020-1350 (Windows DNS Server) und CVE-2020-1472 veröffentlicht. Hierbei kann durch manipulierte Netzwerkpackete die Kontrolle von betroffenen Systemen übernommen werden. Schwachstellen wie diese sind eine der Anwendungsgebiete bei denen NYGMA/RIOT seine Stärken als flexibles Werkzeug zur forensischen Analyse im Incident-Response Fall zeigen kann.

Dem Autor ist keine Open-Source Software bekannt die dem Use-Case und der Mächtigkeit von NYGMA/RIOT entsprechen.

Richtlinie zum „Software-Sprint“

Civil Maritime Rescue Coordination Center (civilMRCC)

Schlussbericht

Zuwendungsempfänger:

Zemker, Wiegert, Reisner GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S36 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)?

Problem: Seit über einem Jahrzehnt gibt es von Seiten der europäischen Staaten keine angemessene Lösung für Menschen in Seenot. Seit einigen Jahren füllen private NGOs diese Lücke. Allerdings sind sie nicht ausreichend untereinander vernetzt. Dies beinhaltet die Kommunikation zwischen den einzelnen Schiffen und mit Flugzeugen, sowie mit den Teams an Land. Ein Problem dabei ist, dass es keinen frei zugänglichen Zugriff auf Daten von Flugzeugen und Verkehrsschiffen in dem Raum gibt.

Zusätzlich gibt es derzeit keine zentrale unabhängige Stelle für die Dokumentation der Geschehnisse auf dem Mittelmeer, wie etwa die Zahl von Booten, Art von Booten, Menschen an Bord, wie sie gerettet wurden und ähnliches.

Alle diese Probleme versuchen wir mit einer Karten-basierten App und einer Datenbank zu lösen.

Vorgehensweise: Ein Grundgerüst für die App stand schon zu Beginn der Förderphase. Für uns war es nun wichtig, zusammen mit der Zielgruppe zu erarbeiten, welche Funktionen noch fehlen und wie die App an die Bedürfnisse der User*innen angepasst werden kann.

Daneben war es uns wichtig, das Projekt in der Open-Source Community zu verbreiten, um dadurch noch Unterstützung in der technischen Entwicklung zu finden.

Schließlich waren die Gespräche mit unterschiedlichen AIS-Providern wichtig, um überhaupt an die benötigten Daten für das Tracken von Schiffen zu kommen, um diese dann in die App einbinden zu können.

Wichtigste Meilensteine:

1. Eine Testphase durchführen, um zu evaluieren, wo es noch Probleme bei der App gibt und wo Nutzer*innen eventuell ein anderes Design und weitere Funktionen brauchen.
2. Um den Einstieg ins Projekt zu erleichtern und im Voraus besser planen zu können, wollen wir die Anforderungen mit zivilen Organisationen in interdisziplinären Workshops definieren und diese im Product-Backlog mit klaren User-Stories abbilden
3. Ausarbeitung des Templatesystems, um eine einfache Anpassung der App zu ermöglichen
4. Deploymentstrategie entwickeln und ausrollen
5. Update, Build & Testsystem: Um hohe Codequalität zu gewährleisten und Updates schnell an Endnutzer*innen ausliefern zu können, wollen wir eine Buildpipeline implementieren, welche die Testing- und Buildprozess automatisiert und die Software an die gängigen Packetmanager ausliefert

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?

Unser Projekt richtet sich zuallererst an NGOs und Akteur*innen in der zivilen Seenotrettung auf dem Mittelmeer. Mithilfe der App, die ein klares Lagebild über die Situation auf dem Mittelmeer zeichnet, wollen wir die Arbeit der zivilen Seenotrettungsakteur*innen verbessern und effizienter machen. Aber auch Journalist*innen und Forscher*innen, sowie die gesamten Zivilgesellschaft gehören mit zur Zielgruppe, denn ihnen wollen wir die Ergebnisse aus unserer Datenbank für Forschungs- und Recherchezwecke zur Verfügung stellen. Dadurch wollen wir mehr Transparenz schaffen und die Zivilgesellschaft aufklären und so dem öffentlichen Diskurs um Migration, der von gezielter Desinformation und Hetze geprägt ist, etwas entgegensetzen.

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 konnten die App schon einem Großteil der NGOs vorstellen und sie in die Nutzung einführen. Dadurch haben wir viel Feedback erhalten, welche Funktionen und Features noch gebraucht werden und was nicht verständlich bzw. User-intuitiv ist. Wir konnten die App sehr gut an diese User*innenbedürfnisse anpassen. Leider befindet sich die Nutzung noch am Anfang, weshalb wir bislang wenig Daten für die Datenbank gesammelt haben.

Die Gespräche mit einem AIS Provider sind positiv verlaufen und wir konnten Daten erhalten.

Wir konnten leider nicht alle Meilensteine erreichen und das Projekt ist noch keinesfalls abgeschlossen. Wir arbeiten immer noch an der Deploymentstrategie und am Update, Build und Testsystem (4.+5.).

Da unser Software-Sprint natürlich sehr durch die Covid-19 Pandemie beeinflusst wurde, haben wir daraus einiges lernen können. So hatten wir zum einen alle etwas mehr Zeit, da viele Freizeitbeschäftigungen wegfielen. Zum anderen haben wir aber auch gemerkt, dass nur online zusammenzuarbeiten, ohne sich auch ab und zu wirklich persönlich zu sehen und zusammenzuarbeiten, wirklich anstrengend ist und nicht so gut funktioniert. Die Motivation wird sehr dadurch gedrückt, nur alleine vor seinem Computer zu sitzen.

Das gleiche gilt auch für die Zusammenarbeit und den Austausch mit anderen Projekten. Dieser und auch die Coachings und Zwischengespräche fanden nur online statt. Gerade was die anderen Projekte betrifft, hätte ein persönliches Treffen den Austausch und das Netzwerken viel einfacher gemacht. Trotzdem war der regelmäßige Austausch und die wöchentlichen Updates mit Marie und Thomas hilfreich, um den wöchentlichen Prozess und Fortschritt reflektieren zu können.

Die Coachings zu User-Stories und Security haben uns sehr geholfen und wir haben uns sehr gefreut, diese externe Unterstützung von Expert*innen erhalten zu haben.

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?

Für uns steht es im Vordergrund, die Arbeit der Akteur*innen in der zivilen Seenotrettung zu erleichtern und zu vereinfachen und ihnen somit Arbeit abzunehmen, damit sie ihre Ressourcen anders einsetzen können.

In der Zukunft würden wir die Karten-basierte App gerne weiterentwickeln und für andere Anwendungsfelder, abseits der Seenotrettung, bereitstellen. Genaue Pläne dazu gibt es noch nicht, aber wir freuen uns sehr, wenn die Open-Source Community das Projekt aufgreift und für andere Kontexte erweitern würde.

Wir haben viel zum Projektmanagement und, wie man das Online-Zusammenarbeiten gestalten und organisieren kann, gelernt. Außerdem haben wir uns fachlich in Docker, Node.js und und vue.js weiterentwickelt.

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?

Nein.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Unser Projekt ist als Open-Source Projekt auf Gitlab unter <https://gitlab.com/civilmrcc> zu finden. Außerdem gibt es ein Wiki unter civilmrcc.eu. Ansonsten können sich Interessierte bei mareike@seawatch.org melden.

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?

Insgesamt haben wir wohl nicht mit so vielen Wünschen und Forderungen von den User*innen gerechnet, weshalb wir mehr Zeit als gedacht dafür gebraucht haben, die App daran anzupassen. Dadurch sind wir mit dem Ausbau des Deploy- und Testsystems nicht so weit vorangekommen, wie 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?

Nein.

Richtlinie zum „Software-Sprint“

TrustPing – sichere Verbindungen für Menschen mit Krebs

Schlussbericht

Zuwendungsempfänger:

Nowottne, Otte und Schöneberg GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS19S37 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation?

Eine Krebsdiagnose ist ein Schock, verbunden mit einer großen Verunsicherung. Unsere Vision mit Trustping ist es, dass Menschen auf der dringenden Suche nach Austausch und Unterstützung, diese einfach finden können, ohne dabei ihre persönlichen Daten zu veröffentlichen und so ihre eigene Privatsphäre verletzen zu müssen. Im Moment findet dieser Austausch hauptsächlich in öffentlichen Internetforen oder Facebookgruppen statt. Aber Austausch und Unterstützung sind wichtig. In Studien konnte belegt werden, dass Selbsthilfe Gesundheitskosten senkt und, dass Menschen, die sich aktiv und informiert ihrer Krankheit stellen, bessere Prognosen haben.

Welches Problem wolltest Du mit Deinem Projekt lösen?

Trustping soll eine sichere Alternative bieten. Trustping ist eine App, die nach wählbaren Kriterien (Diagnose, Therapie, Lebenssituation) Gesprächspartner findet und ihnen sicheren Austausch via verschlüsseltem Chat ermöglicht. Dabei achten wir darauf, dass die Daten der Nutzer nicht frei zugänglich sind, die Kommunikation privat bleibt und wir keine personenbezogenen Daten speichern.

Außerdem leisten wir mit der Bereitstellung einen Beitrag zur Schaffung transparenter, verfügbarer und selbstbestimmter Infrastruktur. (Free and open source software)

Wie war die geplante Vorgehensweise zur Problemlösung?

Geplant war, eine **Android App** zu entwickeln, die drei Haupt-Features enthält:

- initiale **Dateneingabe** (Diagnose, Therapie, Lebenssituation),
- Matching potentieller Gesprächspartner basierend auf gewählten Kriterien, und
- **Kommunikation** per e2e-encrypted Chat.

Die Android App greift auf ein **Backend** zu. Das Backend wird mit Python (django/flask) und SQL entwickelt. Die SQL-Datenbank speichert nur Daten, die zum initialen Kontaktaufbau essentiell sind. Insbesondere werden keine personenbezogenen Daten (Name, Email, Wohnort, etc.) gespeichert. Die Hauptaufgabe des Backends ist das Matching von Gesprächspartnern.

Gesprächspartner tauschen sich per e2e-encrypted Chat aus. Der **Chat** wird mittels des Open Source Matrix Protocols (<https://matrix.org/>) realisiert. Da das Matrix Protocol e2e-encrypted ist, können wir als Betreiber der App die Kommunikation der Anwender nie einsehen.

Wir werden einen dedizierten Matrix Backend Server hosten (Synapse <https://github.com/matrix-org/synapse>), um die Kommunikation zu realisieren.

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?

Zentrale Zielgruppe sind junge Erwachsene mit Krebs und Angehörige. Die unvorhergesehene Konfrontation mit einer lebensbedrohlichen Erkrankung führt oft bei jüngeren Erkrankten zu einer sozialen Isolation. Auch fühlt sich diese Gruppe durch ihren Altersunterschied schlechter in klassischen Selbsthilfegruppen vertreten.

Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?

Das Thema des Softwaresprints dieser Runde »Vertrauen bauen« kann als Herz unserer Arbeit an Trustping beschrieben werden. Im Sinne des mündigen Patienten leisten wir einen Beitrag zum Austausch, indem wir vertrauenswürdige Infrastruktur schaffen. Vertrauen verstehen wir dabei nicht als »blindes« Vertrauen, sondern als begründet aufgrund von Transparenz, Nachvollziehbarkeit und Plausibilität.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt?

User Tests & User Personas

Wir haben den Fokus auf soziales Vertrauen per Design gerichtet und deshalb in unseren User Stories auch kontroverse und betrügerische Nutzer*innen (z. B. Menschen mit Münchhausen Syndrom) berücksichtigt. Auch den Entschluss, die Profildatenbank nicht durchsuchbar zu machen, haben wir an dieser Stelle gefasst. Die Auseinandersetzung hat uns ehrlicher und bescheidener bezüglich möglicher Sicherheitsversprechen gemacht. Es geht nicht nur um technische Sicherheit, sondern ebenso um soziale Sicherheit. Es entstanden dabei auch weiterführende Ideen, wie wir Trustping

noch vertrauenswürdiger gestalten könnten, die den Rahmen der Förderung des Prototype Funds aber gesprengt hätten (siehe mögliche Weiterentwicklungen).

Datenmodell

Mit Unterstützung und inhaltlichem Input der Deutschen Krebsgesellschaft haben wir ein konkretes Daten-Modell (am Beispiel Brustkrebs) entwickelt, was als Kern des Matching-Algorithmus dient.

App

Zunächste haben wir unsere Designentscheidungen nochmal überprüft und Tools und Libraries evaluiert. Wir haben uns entschieden, die App mit Dart und Flutter umzusetzen. Entscheidend waren v.a. folgende Kriterien:

- plattformübergreifend (potentiell iOS und Web in der Zukunft)
- schnelle Entwicklung
- stabile Community
- zahlreiche Libraries vorhanden (inkl. Matrix, aber nicht e2e Encryption)

Wir haben kein dediziertes Django-backend implementiert, sondern nutzen im Moment Google Firebase als Backend-as-a-Service. Dies bietet User Management (inkl. anonymen User), Database, etc. Die eigentliche Logik ist komplett in der Android App in Dart implementiert.

Ein Matching-Algorithmus basierend auf Übereinstimmung von User-Profilen wurde implementiert.

Testphase Klinik

Den fettesten Strich durch die Rechnung hat uns Corona bezüglich unserer Testphase in der Rehaklinik im Universitätsklinikum Freiburg gemacht. Durch die Herausforderungen und entsprechenden Einschränkungen war es unmöglich für uns, dies im geplanten Zeitrahmen umzusetzen. Aber wir haben die Zusage aus Freiburg und stehen im Austausch, sodass wir hoffen, die Testphase möglichst bald nachholen zu können.

Die Einbindung potenzieller Nutzer*innen war uns von Anfang an sehr wichtig. Also haben wir ein angepasstes Format entwickelt und in mehreren wöchentlichen Calls Walkthroughs mit Nutzer*innen gemacht, in denen jeweils eine Person vor dem Hintergrund ihrer persönlichen Erfahrungen durch die App navigieren durfte und ihre Gedanken und Eindrücke mit unserem Team geteilt hat. Das war extrem wertvoll für unseren Entwicklungsprozess.

Konnten alle Meilensteine erreicht werden?

Bezüglich der möglichen und wünschenswerten Security Features konnten wir noch nicht vollumfänglich unseren Ansprüchen genügen. Es bleiben folgende Ausbaustufen für die nahe Zukunft auf unsere Liste:

- Matrix-Protokoll mit Verschlüsselung implementieren (inkl. Synapse Server aufsetzen)
- zusätzliche Sicherheitsfeatures (Hashing, Salt & Pepper) für die Ablage von Daten in der Datenbank
- bereits geplante User Tests in Klinik durchführen (Kontakt und Absprache ist da, Corona-bedingt nicht möglich gewesen und im Moment noch nicht möglich)

Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Bestärkt durch den Input der OKF, als auch der Coachings, haben wir einen starken Fokus auf eine frühzeitige Einbindung potentieller User (auch User Tests und Entwicklung von User Personas) in die Entwicklung der App gelegt. Dies kostet viel Zeit, war aber für viele Entscheidungen besonders wertvoll. Viele neue Erfahrungen konnten wir durch diesen Ansatz der nutzerzentrierten Entwicklung machen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts?

- Möglichkeit zum Austausch
- Vermittlung von Gesprächspartnern anhand von spezifischen Themen (Diagnose, Nebenwirkungen, Lebenssituation, Interessen, etc.)
- **Wahrung der Privatsphäre** (verschlüsselte Kommunikation, Security by Design)
- Integrität der Daten (keine Weiterverwendung irgendwelcher Daten)

Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse?

Für uns ist die Open Source Stellung v.a. eine politische Entscheidung, die zu weitestgehender Transparenz und damit Vertrauen in die App führen soll. Sie ist damit nicht zusätzliches Feature, sondern Kern des Produkts.

Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Das Prinzip von Trustping lässt sich sowohl auf weitere Erkrankungen, als auch auf andere sensitive Informationsfelder (politische Aktivitäten, Menschenrechte etc.) übertragen und eröffnet somit über diesen ersten Test hinaus wichtige Perspektiven.

Unsere Vision ist und bleibt, mit Trustping eine vertrauenswürdige App auf den Weg zu bringen, die Menschen in äußerst schwierigen Situationen einen sicheren Einstieg in einen geschützten Austausch bereitet. Das Fundament dafür ist gebaut. Aber es sind viele Ideen und Gedanken entstanden, die noch umgesetzt werden wollen. Einige Ausblicke wollen wir noch geben.

Matching

Emotionales Herzstück unserer App sind die bunten Flächen, die jeweils für eine Menge von Nutzer*innen stehen, mit denen ich in einem Bereich (Diagnose, Therapie, Lebenssituation) Übereinstimmungen habe. Sie sind Filtermechanismus für den Fokus meiner Ping-Anfrage, aber auch spielerisches Moment und Logo von Trustping. Soweit die Idee und das Design an dieser Stelle bereits ausgearbeitet sind, fehlt noch die technische Umsetzung der animierten Nutzer*innenführung in der App.

Trust

Nicht nur im Namen tragen wir das Wort Vertrauen (Englisch: trust), auch viele Überlegungen ranken sich um diese Thema. Vertrauen bauen, das Thema der Runde 7 im Prototype Fund. Unser Thema. Wie entsteht Vertrauen im virtuellen Raum? Das einfachste ist: Vertrauen ist bereits vorher da. Das

heißt, wenn ich eine Person im realen Leben kenne, weil ich sie z. B. in der Chemotherapie kennengelernt habe. Trustping bietet die Möglichkeit, mit einem QR Code Menschen direkt zu verlinken, einen Chat zu starten und die Schritte des Ping Pong zu überspringen.

Verifizierte Profile

Ein sehr interessanter Ansatz, den wir weiter prüfen wollen ist, dass Trustping-Profile durch die behandelnden Ärzt*innen ausgestellt werden. In einer simplen browserbasierte Web-Applikation erstellen die behandelnden Onkolog*innen oder Rehaärzt*innen ein neues Profil mit zugehöriger Diagnose per ICDS Code. Die Patient*innen erhalten einen Code, mit dem sie das Trustping-Profil zuhause initialisieren und anpassen können. Diese Profile erhalten ein Signet als »verifizierte Profile«, da hier reale Person und korrekte Diagnose sichergestellt werden. Für die Ärzt*innen könnte die Möglichkeit bestehen, beim Anlegen des Profils zusätzlich relevantes digitales Info-Material aus vertrauenswürdiger Quelle (z. B. Krebsinformationsdienst) per Selektion zu hinterlegen.

*Nutzer*innen blockieren*

So sorgfältig wir alle Entscheidungen hinsichtlich Vertrauen treffen, ist klar, dass komplette Sicherheit nicht möglich ist. Sollte es also dennoch vorkommen, dass jemand sich von anderen Nutzer*innen belästigt fühlt oder aus anderen Gründen einen bestehenden Kontakt auflösen möchte, wird die Möglichkeit bestehen, Kontakte zu entfernen, also Nutzer*innen zu blockieren.

Persönlicher Tracker

Viel Zuspruch bei unseren Interview erhielt die Idee, einen Kalender mit kleinen Statistiken anzubieten, wo ich einfach mein Befinden, meine Medikamente, eventuell Ernährung oder sportliche Aktivitäten tracken und dokumentieren kann. Der Markt der Tracking-Apps ist umfangreich, doch meist gebe ich dafür meine Daten aus der Hand. Wir würden solche Daten gern auf dem Telefon belassen und nur die Funktionalität den Nutzer*innen zur Verfügung stellen, damit sie selbst die Vorteile solcher Auswertungen nutzen können. Sei es im Gespräch mit der Ärztin oder zum eigenen Erkennen von Zusammenhängen: Welche Medikamente führen zu welchen Nebenwirkungen und was passiert, wenn ich dies oder jenes ändere?

Gruppen

Im Moment ist der Eins-zu-eins-Chat implementiert. In Zukunft soll es die Möglichkeit zum Erstellen von offenen und privaten Gruppen und perspektivisch auch das Einrichten von Channels als Informationsmedium zu bestimmten Themen geben.

Forschung

Wenn es um Gesundheitsdaten geht, spielt das Potenzial für Forschung immer auch eine Rolle. Bei den meisten Menschen mit Krebs gibt es eine hohe Bereitschaft, Krebsforschung zu unterstützen. Auch wir würden diese Möglichkeit gern eröffnen - aber vollständig freiwillig: Mit einer Option, in eine Kontaktmöglichkeit für Forschende aktiv einzuwilligen. Für Forschende könnte die Möglichkeit, Menschen mit Krebs nach gewählten Kriterien selektiv zu kontaktieren ein wertvolles Instrument zur Kontaktaufnahme sein. Dies gilt eventuell nicht immer für die Rekrutierung von Studienteilnehmer*innen, die oft randomisiert erfolgen muss, wohl aber beispielsweise für qualitative Forschung, Hypothesenbildung und auch Grundlagenforschung, die auf biologische Tumoreigenschaften abzielt.

Nachhaltigkeit und Finanzierung

Wir machen Trustping aus Leidenschaft, aber wir sind nicht naiv. Um eine nachhaltige Ausarbeitung,

Weiterentwicklung und einen dauerhaften Betrieb einer App zu sichern, braucht es Ressourcen. Wir werden uns um eine Anschlussförderung bemühen und haben einige Überlegungen zur Produktisierung von Trustping angestellt. Wichtigstes Kriterium hierbei war für uns, keine kommerzielle Verwendung irgendwelcher Daten zu erwägen. Das Thema kam auch auf, als wir in Verhandlung mit dem Universitätsklinikum Freiburg über die Teilnahme an der Pilotphase standen. Es ist im Moment nichts entschieden, aber einige Ideen werden wir abwägen und evaluieren.

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

Unbedingt! Persönlich hat uns die unternehmerische Herangehensweise an eine Prototypentwicklung sehr gefordert, aber auch beflügelt. Viel Freude und Unterstützung in der Zusammenarbeit im Team. Und hier noch ein paar fachliche Aspekte, an denen wir wachsen durften (wahrscheinlich unvollständig)

- Android/ Mobile Entwicklung (wir kommen aus dem Data Science)
- Konzeptentwicklung
- User Testing und UI/ UX Entwicklung
- Security by Design

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?

Weiterführende Recherchen, Evaluation von Ideen und externes Coaching haben teilweise zu Anpassungen in den Designentscheidungen geführt. Wir hatten aber keine echten Sackgassen, im Sinne von getaner Arbeit, die zu keiner Verwendung geführt hat. Wir haben uns auch bemüht, mit dem Ansatz des schnellen Prototyping und iterativen Verbesserungen zu arbeiten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

<https://trustping.github.io>

<https://github.com/trustping>

<https://prototypefund.de/en/project/trust-ping-secure-connections-for-people-with-cancer/>

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?

Es gab v.a. ein großes Ereignis, was aus unsere Prototypefund-Phase (wie auch sicherlich der gesamten Runde) eine komplett andere gemacht hat, das war Corona! Schulen und Kitas bleiben

geschlossen und machen für uns Eltern den Alltag zur Kraftprobe. Innerhalb von Tagen verlagern sich bundesweit die Büroplätze in die heimischen Wohnungen, wo schon die Kinder und Partner*innen warten und quasi das gesamte soziale Leben in den virtuellen Raum. Wir haben also, wie alle anderen auch, viel virtueller zusammen gearbeitet, als wir uns das vorgestellt hatten. Und es fehlte sicher auch die ein oder andere ruhige konzentrierte Stunde. Aber im Großen und Ganzen hat die Arbeit an Trustping viel Spass gemacht und sind wir mit den Ergebnissen sehr zufrieden.

Es gab teilweise eine Verschiebung der Prioritäten und notwendige Anpassungen. Wir haben einen eindeutigen Mehraufwand in Sachen User Testing und Konzeptentwicklung gehabt, was dazu geführt hat, dass wir eine sehr stabile Basis für viele Designentscheidungen haben, aber teils offene Entwicklungsaufgaben aus unsere Ursprungsplanung.

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?

Bei weiterführenden Recherchen zum Matrix Protokoll stellten wir fest, dass zwar Matrix, aber nicht das E2E-Verschlüsselungsprotokoll in der Dart-Library implementiert ist. So entstand bereits zu Beginn des Prototype-Funds ein Austausch mit dem Entwickler der Library. Wir erfuhren, dass er bereits in der Umsetzung ist und der Zeitplan schien gut mit unserer Planung zu passen. Durch Verzögerungen, konnte die Implementierung nur vorbereitet, aber noch nicht abgeschlossen werden. Da wir zu keiner sinnvollen Alternative kamen, haben wir stattdessen einen stärkeren Fokus auf die UI und UX Entwicklung gelegt.