

SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE SEPTEMBER 2020

Konsolidierter Schlussbericht

Förderkennzeichen:

01IS21S01
01IS21S02
01IS21S03
01IS21S04
01IS21S05
01IS21S06
01IS21S07
01IS21S08
01IS21S09
01IS21S10
01IS21S11
01IS21S12
01IS21S13
01IS21S14
01IS21S15
01IS21S16
01IS21S17
01IS21S18
01IS21S19
01IS21S20
01IS21S21
01IS21S22
01IS21S23
01IS21S24
01IS21S25
01IS21S26
01IS21S27
01IS21S28

Vorhabenbezeichnung: Software Sprint – 28 Einzelvorhaben

Laufzeit der Einzelvorhaben: 01.03.2021-31.08.2021

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

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE SEPTEMBER 2020 Konsolidierter Schlussbericht	
4. Autor(en) [Vorname(n), Name(n)] Norbert Rost Thorsten Hinrichsmeyer Annika Hannig Mareike Ippen Jochen Stärk Monika Zielinski Jonathan Meyer Ulf Köther Michael Kamphausen Daniel Kehne Timo Kösters Gregory Jones Jan Hartmann Lisa Passing Mathias Steudtner Sid Moreira da Silva Bodo Tasche Linus Hagemann Michael Helmbrecht Melina Rüttimann Niklas Stein Simon Deeg Magdalena Noffke Niklas Jordan Jonas Meurer Basile Simon Jie Liang Lin Klaus Herberth	5. Abschlussdatum des Vorhabens 31.08.2021 <hr/> 6. Veröffentlichungsdatum 14.10.2021 <hr/> 7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) 01IS21S01 HackingPolitics - Norbert Rost, Jakob Schumann, Tina Hoffmann, Philipp Munzert, Pauline Grahlmann, Frieder Jacobi GbR 01IS21S02 Hinrichsmeyer Swillus GbR 01IS21S03 Matthias Hannig	9. Ber. Nr. Durchführende Institution <hr/> 10. Förderkennzeichen 01IS21S01-01IS21S28

01IS21S04	Zemke-Ippen GbR	11. Seitenzahl 136	
01IS21S05	Jochen Stärk		
01IS21S06	Zielinski & Kürten GbR		
01IS21S07	Jonathan Meyer		
01IS21S08	Köther-Schubert-Rentzsch-Molitor GbR		
01IS21S09	Kamphausen Schmidpeter Zwick GbR		
01IS21S10	Lunes - Sven Seeberg, Lukas Böhm, Daniel Kehne, Julia Götz GbR		
01IS21S11	Timo Kösters		
01IS21S12	Pegasus GbR		
01IS21S13	Hartmann, Eliasson GbR		
01IS21S14	Lisa Passing		
01IS21S15	Mathias Steudtner		
01IS21S16	Moreira Veit Heinzmann Schumann GbR		
01IS21S17	Bodo Tasche		
01IS21S18	Hagemann & Prochaska GbR		
01IS21S19	Michael Helmbrecht		
01IS21S20	Gatchbin GbR		
01IS21S21	Stein Timme Edelkötter Rohmund GbR		
01IS21S22	Deeg Picker GbR		
01IS21S23	Magdalena Noffke		
01IS21S24	Niklas Jordan		
01IS21S25	Meurer Wiehle GbR		
01IS21S26	Basile Simon		
01IS21S27	Jie Liang Lin		
01IS21S28	Klaus Herberth		
12. Fördernde Institution (Name, Adresse)			13. Literaturangaben
Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn			14. Tabellen
			15. Abbildungen
16. Zusätzliche Angaben			
17. Vorgelegt bei (Titel, Ort, Datum) Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) DLR Projektträger Berlin			
18. Kurzfassung Der Schlussbericht umfasst die Einzelschlussberichte der neunten Auswahlrunde der Fördermaßnahme Software Sprint (Einreichungsdatum Skizze: 30.09.2020). Die Auswahl der Vorhaben erfolgte nach externer Begutachtung unter Abstimmung mit dem BMBF.			
19. Schlagwörter Software Sprint, Prototype Fund			
20. Verlag		21. Preis	

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report	
3. title SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE SEPTEMBER 2020 Konsolidierter Schlussbericht		
4. author(s) (family name, first name(s)) Norbert Rost Thorsten Hinrichsmeyer Annika Hannig Mareike Ippen Jochen Stärk Monika Zielinski Jonathan Meyer Ulf Köther Michael Kamphausen Daniel Kehne Timo Kösters Gregory Jones Jan Hartmann Lisa Passing Mathias Steudtner Sid Moreira da Silva Bodo Tasche Linus Hagemann Michael Helmbrecht Melina Rüttimann Niklas Stein Simon Deeg Magdalena Noffke Niklas Jordan Jonas Meurer Basile Simon Jie Liang Lin Klaus Herberth	5. end of project 31.08.2021	6. publication date 14.10.2021
	7. form of publication	
	8. performing organization(s) (name, address) 01IS21S01 HackingPolitics - Norbert Rost, Jakob Schumann, Tina Hoffmann, Philipp Munzert, Pauline Grahlmann, Frieder Jacobi GbR 01IS21S02 Hinrichsmeyer Swillus GbR 01IS21S03 Matthias Hannig	
9. originator's report no.		10. reference no. 01IS21S01-01IS21S28

01IS21S04	Zemke-Ippen GbR	11. no. of pages 136	
01IS21S05	Jochen Stärk		
01IS21S06	Zielinski & Kürten GbR		
01IS21S07	Jonathan Meyer		
01IS21S08	Köther-Schubert-Rentzsch-Molitor GbR		
01IS21S09	Kamphausen Schmidtpeter Zwick GbR		
01IS21S10	Lunes - Sven Seeberg, Lukas Böhm, Daniel Kehne, Julia Götz GbR		
01IS21S11	Timo Kösters		
01IS21S12	Pegasus GbR		
01IS21S13	Hartmann, Eliasson GbR		
01IS21S14	Lisa Passing		
01IS21S15	Mathias Steudtner		
01IS21S16	Moreira Veit Heinzmann Schumann GbR		
01IS21S17	Bodo Tasche		
01IS21S18	Hagemann & Prochaska GbR		
01IS21S19	Michael Helmbrecht		
01IS21S20	Gatchbin GbR		
01IS21S21	Stein Timme Edelkötter Rohmund GbR		
01IS21S22	Deeg Picker GbR		
01IS21S23	Magdalena Noffke		
01IS21S24	Niklas Jordan		
01IS21S25	Meurer Wiehle GbR		
01IS21S26	Basile Simon		
01IS21S27	Jie Liang Lin		
01IS21S28	Klaus Herberth		
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn			13. no. of references
			14. no. of tables
			15. no. of figures
16. supplementary notes			
17. presented at (title, place, date) Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) DLR Projektträger Berlin			
18. abstract The final report includes the individual final reports for the 9th call of participants within the public funding activity Software Sprint. The selection of the projects (call No. 9 deadline: 30.09.2020) took place after external evaluation under coordination with the BMBF.			
19. keywords Software Sprint, Prototype Fund			
20. publisher		21. price	

Richtlinie zum „Software-Sprint“

Hacking Politics Online

Schlussbericht

Zuwendungsempfänger:

hacking politics GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S01 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Politikverdrossenheit speist sich aus Unverständnis über das politische System und zu diffuse Einflussmöglichkeiten. Wer mal einen Stadtratsantrag geschrieben und in den Rat gebracht hat, spürt Selbstwirksamkeit, versteht das System und kann Protest in Prozess verwandeln. Die Schulbildung fokussiert auf die Bundespolitik, im Lebensalltag ist aber die kommunale Ebene näher am Menschen.

Daher erhöht es das politische Verständnis und die Selbstwirksamkeit, wenn Menschen anhand konkreter Vor-Ort-Fragen sich mit Politik befassen. Diese Befassung kann in besonderer Art passieren, wenn sie einen Stadtratsantrag schreiben. Wie aber schreibt man Stadtratsanträge? Dazu gab es bereits ein Workshop-Format und ein Arbeitsblatt, das bereits mehrfach erfolgreich analog erprobt wurde. Im Rahmen des Vorhabens sollte eine digitale Plattform entstehen, mit der die Entwicklung von Stadtratsanträgen leicht verständlich digital möglich wird.

Das entwickelte Tool verbessert das Politikverständnis (SDG4) und bietet selbstbedienbare Zugänge ins politische System – beginnend auf der Stadtratsebene, aber offen für Landtage und Bundestag. Es unterstützt so insbesondere die Transformation der Städte (SDG 11).

Die entstandene Plattform „HackingPoliticsOnline“ bietet einen strukturierten Workflow zur kollaborativen Entwicklung von Stadtratsanträgen, mit dem politikunerfahrene Bürger*innen Stadtratsanträge Schritt für Schritt entwerfen lernen und Mandatsträger kollaborativ mit Betroffenen Anträge entwickeln: Problem & Handlungsauftrag, Argumente & Gegenargumente, Ratsmehrheiten & Fraktionsinteressen, Antragstext & Einbringer.

Der Workflow führt die User Schritt für Schritt von der Problemlage zum einreichbaren Stadtratsantrag, der als OpenDocument-Dokument „rausfällt“ und dann in den formalen

Geschäftsgang im Rat eingebracht wird. Niederschwellige Bedienbarkeit und moderne Usability senken die Nutzungshürden.

Das Projekt ist als Webapplikation umgesetzt, um auf vielen Geräten nutzbar und für alle zugänglich zu sein. Es wurde auf responsives Design und Accessibility geachtet. Die Applikation basiert auf PHP & Symfony, Daten werden über eine JSON-REST-API verfügbar gemacht, es wurde möglichst auf erprobte OpenSource-Bibliotheken und -Tools zurückgegriffen um zukünftige Weiterentwicklung zu erleichtern. Die Software wird damit auf vielen Servern und Container-Umgebungen aufwandsarm einsetzbar.

In der **Konzeptionsphase** wurden die Erfahrungen aus den Workshops strukturiert und in ein Lastenheft formuliert inkl. Use-Cases, Datenkatalog, Workflow-Design, Funktionsbeschreibungen.

In der **Usability-Prototyping-Phase** wurde vom Nutzer aus gedacht ein Prototyp des Clients in Form eines Click-Dummys entwickelt, von dem aus Anforderungen an die API abgeleitet wurden und im Lastenheft beschrieben werden.

In der **Programmierphase** wurden die Anforderungen des Lastenhefts in Software umgesetzt. Aufscheinende Programmier-Probleme wurden über eine Anpassung der Konzeption gelöst.

Ein **Erprobungs-Workshop** zusammen mit der Anstiftung führte zu wertvollen Erfahrungen und zu 5 ersten realen Antragsprojekten.

In der Phase der **Usability-Optimierung** werden grafische Elemente und HTML/CSS so angepasst, dass für die Nutzer eine moderne, selbsterklärende und leichtgängige Nutzungserfahrung entsteht.

Das Team aus 6 Leuten brachte dazu unterschiedliche Perspektiven und Erfahrungen ein: API-Entwicklung, Frontend-Entwicklung, Usability & Grafikdesign, Projektmanagement sowie politikwissenschaftliche Erfahrung.

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

Es ist ein OpenSource-Werkzeug entstanden, das (zivil)gesellschaftliche Aktivitäten (CivicTech) unterstützt, insbesondere im Bereich Transparenz und Bürgerbeteiligung. Über „die Politik“ wird viel gemeckert, Protestbewegungen formulieren Protest, aber gezielt-strukturierte Einflussmöglichkeiten sind rar. Ein Tool zur strukturierten, kollaborativen Stadtratsantrag-Entwicklung soll Mandatsträgern wie Politikinteressierten einen Zugang zum politischen System geben. Dies kann sowohl durch eine online betriebene Plattform für alle deutschen Gemeinden, wie auch in Form lokaler Instanzen erfolgen.

Zugleich ist ein unter MIT-Lizenz stehender Quellcode entstanden, auf dem Weiterentwicklungen passieren können sowie andere Projekte mit ähnlicher Quelltextbasis entstehen können. Dafür wurden bestehende technische Grundlagen (PHP-Symfony, REST-API, Nuxt.js/Vue.js, Hocuspocus-Server) zu einem neuen Gesamtsystem prototypisch kombiniert. Die Beteiligten haben vertiefte Kenntnisse in der Anwendung der Technologien für einen spezifischen Anwendungsbereich erworben.

Ausführliche Darstellung der Ergebnisse

(Kommunal-)Politik ist ein herausforderndes Geschäft, das in der politischen Bildung nur eine untergeordnete Rolle spielt. Selbst Politikwissenschaftler*innen sagen oft, mit Kommunalpolitik hätten sie sich nicht beschäftigt und in der schulischen Gemeinschaftskunde beginnt „Politik“ meist erst auf Bundesebene. Dabei lässt sich auf lokaler Ebene sehr viel gestalten – der individuelle Einfluss- und Identifikationsfaktor ist hoch. Die Plattform „hacking politics online“ basiert auf einem erprobten Workshop, bei dem Teilnehmende lernen, Stadtratsanträge für ihre eigenen Anliegen zu formulieren, ohne selbst zwingend Mitglied im Stadtrat/Gemeinderat/Senat usw. sein zu müssen. Die Plattform hilft Einzelnen, ein Verständnis für Politik zu entwickeln, indem sie eigene politische Vorstellungen auf die lokale Ebene herunterbrechen und in Form eines Stadtratsantrages strukturieren.

Dazu fragt die Plattform in der „**Denkarbeit**“ strukturiert solche Aspekte ab, die den Nutzenden helfen, sich dem Antragstext anzunähern: Was ist das Problem, das in der Gemeinde gelöst werden soll? Was soll die Verwaltung tun, um dieses Problem zu lösen? (Handlungsauftrag) Welche Argumente sprechen gegen diesen Handlungsauftrag? Und warum sollte der Gemeinderat den Antrag dennoch beschließen? Welche Fraktion im Gemeinderat hat welche Eigeninteressen an dem Antragsthema? Und welche Kombination der existierenden Fraktionen ergeben eine Mehrheit?

Diese Fragestellungen helfen den Nutzer*innen, ihr Anliegen von verschiedenen Perspektiven auszuleuchten. Sie sammeln in dieser Denkarbeit nicht nur Inhalte für den eigentlichen Antragstext, sondern werden sich auch klarer darüber, was ihr eigentliches Anliegen ist und wie sie es in einen Handlungsauftrag für die Verwaltung übersetzen müssen.

An die Denkarbeit schließt die „**Schreibarbeit**“ an. Die Plattform bietet dafür ein integriertes „Etherpad“, in dem der eigentliche Antragstext geschrieben wird. Allerdings werden den Nutzenden alle zuvor in der Denkarbeit gesammelten Informationen sichtbar gemacht, so dass die „Übersetzung“ von der gesammelten Denkarbeit in die Schreibarbeit leichter fällt. Das Produkt ist eine OpenDocument-Datei, die in gängigen Textverarbeitungsprogrammen weiterverarbeitet werden kann. Sie ist die Grundlage, um mit Stadträten und Fraktionen ins Gespräch zu kommen, dafür wie der Antrag „in den Geschäftsgang“ kommt, sprich: im Stadtrat behandelt wird.

Die Bearbeitung von Schreibarbeit und Denkarbeit kann **kollaborativ** erfolgen: Online wird jeder angestrebte Stadtratsantrag als „Antragsprojekt“ behandelt. An einem Antragsprojekt können mehrere Menschen mitarbeiten, so dass kollektives Wissen gesammelt und eingebracht werden kann. Die Plattform unterstützt das dadurch, dass Bereiche, die von einem User bearbeitet werden, für die anderen User gesperrt werden. Verlässt die*der Bearbeiter*in den Bereich, wird dieser für die anderen User freigegeben. So können mehrere Menschen konfliktarm an einem gemeinsamen Stadtratsantrag arbeiten und ihre Expertise zusammenfügen.

Die Plattform hat zwei zentrale Funktionen: Die Plattform führt auch politisch Unbedarfte Schritt für Schritt auf ein Antragsdokument hin, das im politischen Raum genutzt werden kann. Sie hilft damit, aus der gängigen „**Forderungskultur**“ eine „**Beteiligungskultur**“ zu machen. Aus „die Stadt müsste mal...“ wird „lass uns einen Stadtratsantrag schreiben!“ Mehr und qualitativ hochwertige Stadtratsanträge können das politische Leben in Kommunen fördern und mehr politischen Anliegen Gehör verschaffen. Dazu trägt bei, dass mehrere Menschen am gleichen Antragsprojekt arbeiten können und so Vernetzung gleichartiger Interessen möglich wird. Die Plattform befähigt politisch

Interessierte dazu, ihr politisches Interesse in eine wirksame Form zu bringen: Sie ist eine **Plattform für politische Bildung**. Durch die schrittweise Hinführung (auf den Umwegen der „Denkarbeit“) werden Interessierte mit den Eigenheiten lokaler Politik und dem Instrument des Stadtratsantrags vertraut gemacht. Die Kollaborationsmöglichkeit erleichtert, dass Menschen voneinander lernen.

Um die Hintergründe erfolgreicher Kommunalpolitik und wirksamer Stadtratsarbeit auszuleuchten, wurden wiederkehrende Fragen in einer FAQ zusammengetragen und als Wiki veröffentlicht. Informationshäppchen auf der Plattform geben einen Vorgeschmack und verlinken dann zu tiefergehenden Informationen im Wiki. So soll das learning-by-doing durch kontextgenaue Hintergrundinformationen unterstützt werden.

Die Plattform stellt damit einen öffentlich nutzbaren digitalen Raum dar, in welchem politische Bildung vermittelt wird, ohne zu belehren. Vielmehr erarbeiten die Nutzer*innen sich eigenständig anhand einer eigenen politischen Idee oder eines zu lösenden lokalen Problems, wie sich Ideen und Probleme in das politische Instrument des Stadtratsantrags übersetzen lassen. Das angestrebte Ergebnis – ein Antragstext – ist ein konkret nutzbares Artefakt: Mit diesem kann man auf jene Ansprechpartner*innen in der eigenen Kommune zugehen, die man im Rahmen der Denkarbeit recherchiert hat. Selbst wenn der Antragstext von den Mandatsträger*innen und Fraktionen noch einmal überarbeitet und per Änderungsanträgen im Stadtrat verändert wird, so öffnet sich dennoch ein konkreter Einflussraum. Nur Mandatsträger*innen können Stadtratsanträge in den Geschäftsgang bringen. Prinzipiell können aber alle Menschen Stadtratsanträge schreiben. Lernen mehr Menschen, wie man solche Dokumente gestaltet und wie man sie an die Mitglieder eines Stadt- oder Gemeinderates heranträgt, so belebt dies die politische Kultur. Aus Machtlosigkeit wird Einflussfähigkeit. Aus der weit verbreiteten **Forderungskultur** („Die Stadt müsste mal...“) erwächst eine stärkere **Beteiligungskultur**, wenn politische Werkzeuge wie „Stadtratsanträge“ für mehr Bürger*innen nutzbar werden. Das politische Instrument des Stadtratsantrags zu demokratisieren und mehr Menschen Zugang zu diesem zu geben, ist Ziel der Plattform.

Das Ergebnis liegt als OpenSource-Software unter MIT-Lizenz auf github:

<https://github.com/HackingPolitics/>

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Plattform kann künftig von Akteuren der politischen Bildung sowie von politischen Akteuren eingesetzt werden. Nutzer*innen lernen damit, Stadtratsanträge zu schreiben und sollten ein vertieftes Verständnis für politische Zusammenhänge und politisches Handeln erlangen. Der Sourcecode sollte wiederum nutzbar sein, um auch andere digitale Werkzeuge zu entwickeln. Insbesondere die Kollaborations-Funktionalität dürfte für andere Tools nützlich sein: dass User bei gleichzeitiger Nutzung nicht ihre Daten überschreiben, sondern parallel aber konfliktfrei am gleichen Antragsthema arbeiten.

Ein Geschäftsmodell für die Weiternutzung der Plattform gibt es aktuell noch nicht. Es soll versucht werden, mit Akteuren der politischen Bildung eine verlässliche Einsatzmöglichkeit zu erarbeiten und so auch eine Weiterentwicklung des Werkzeugs zu ermöglichen.

Alle Teammitglieder haben sowohl technisch, politisch, wie auch persönlich dazugelernt.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die ursprünglichen Ziele wurden mit den ursprünglich geplanten Arbeiten erreicht.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

- Öffentliche Erläuterung des Projekts: <https://demoweeek.prototypefund.de/projects/03-hackingpoliticsonline.html>
- Quelltext: <https://github.com/HackingPolitics/>
- Testplattform: <https://hacking-politics.de/>
- FAQ: <https://www.hacking-politics.de/faq>
- Hintergrundinformationen: <https://zukunftsstadt.de/hacking-politics>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Arbeits- und Kostenplan wurde eingehalten. Der krankheitsbedingte Ausfall eines Teammitglieds wurde durch Verlagerung von Aufgaben auf die anderen Teammitglieder ausgeglichen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Uns sind keine relevanten Ergebnisse bei anderen Stellen bekannt geworden.

Richtlinie zum „Software-Sprint“

Mehr kollaborative Apps und Reichweite für Openappstack

Schlussbericht

Zuwendungsempfänger:

Hinrichsmeyer Swillus GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S02 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Motivation & Vorgehensweise

OpenAppStack (OAS) ist ein freies/Open-Source-Softwareprojekt, welches Nutzer*innen ermöglicht, Online-Applikationen schnell und einfach in der eigenen Infrastruktur bereitzustellen und zu administrieren. Wir wollen nicht-kommerzielle Alternativen zu den großen Online-Diensten für möglichst viele Nutzer*innen zugänglich zu machen. In Zeiten von Covid19 ist der Bedarf nach kollaborativen Tools erheblich gestiegen. Wir wollen OAS deshalb um neue Applikationen erweitern, die Teams in der Zusammenarbeit unterstützen. OAS soll dazu zivilgesellschaftlichen Organisationen und Einzelpersonen alles bereitstellen, was zur Organisation und Zusammenarbeit benötigt wird. Und das ohne Kompromisse bei den Themen Datensicherheit und Privatsphäre einzugehen.

Um die Reichweite des Projekts zu erhöhen namen wir uns als ersten Meilenstein vor, die Plattform für Organisationen mit bereits vorhandener Hosting-Infrastruktur kompatibel zu machen. Dazu planten wir eine Überholung von Kernkomponenten die letztlich auch zur einfacheren Erweiterbarkeit der Plattform beitragen sollte. Währenddessen war es unser Ziel durch Kontakt zur OpenSource Community und potenziellen Nutzern herauszubekommen, welche Applikationen wir in unserem zweiten Meilenstein, der Integration neuer Apps integrieren sollen. Zudem namen wir uns vor die Authentifizierungs- und Administrationsoberfläche zu verbessern und technisch umzugestalten um die Sicherheit und Benutzerfreundlichkeit der Plattform zu verbessern.

Unser gesellschaftlicher Beitrag

Die Zentralisierung von Internet-Infrastruktur wirkt sich negativ auf die Entwicklung einer offenen Gesellschaft aus. Covid19 und die damit verbundene Homeoffice-Situation hat die Zentralisierung weiter vorangetrieben und zu einem rasanten Anstieg von Nutzer*innen kommerzieller Online Dienste wie Zoom oder Google Docs geführt. Dies verlangsamt die Entwicklung freier Alternativen und gefährdet die digitale Mündigkeit von Nutzer*innen, die Dritten oft unwissentlich sensible Bereiche ihrer Privatsphäre preisgeben.

Wir wollen freie Alternativen zu den großen Online-Diensten für mehr Menschen nutzbar und attraktiv machen, deren Daten-Souveränität und -Sicherheit für uns im Vordergrund steht. Wir entwickeln OAS vor allem für NGOs und Unternehmen, welche bei diesen Themen keinen Kompromiss machen wollen oder können. Ziel von OAS ist es aber auch eine freie und offene Plattform zu schaffen, der sich Entwickler*innen anschließen können um das Thema der Dezentralisierung gemeinsam, „bottom-up“ weiterzutragen.

Ergebnisse

Wir konnten während des Förderungszeitraumes 3 weitere Applikationen in OpenAppStack integrieren. In die Nextcloud installation OpenAppStacks ist nun ein kollaborativer Online- Kalender und ein Passwort-Manager integriert und Nutzern steht nun das Kanban-Tool Wekan zur Verfügung, um gemeinsam Projekte zu organisieren. Wir haben außerdem an Kernkomponenten von OpenAppStack gearbeitet, was bedeutet, dass sich die Plattform nun auf bereits bestehenden Kubernetes-Clustern installieren lässt. Damit erhöht sich nicht nur die Anzahl potentieller Nutzer, denn die Komplexität der Installationsroutinen und die damit verbundene Komplexität automatisierter Testroutinen haben sich durch diese Änderungen deutlich verringert. Um diesen Meilenstein zu erreichen haben wir Flux durch Flux2 ausgetauscht, die Installation und Initialisierung von Applikationen von der Installation des Kubernetes-Clusters entkoppelt und umfangreiche Änderungen in der Konfiguration der Continuous Integration Plattform (Gitlab) vorgenommen. Testsuits, welche die Integration unserer Applikationen automatisch testen haben wir von Python-Behave auf Taiko migriert um nötige Anpassungen der Tests in zukunfft zu vereinfachen. Die Single sign-on Implementierung von OAS haben wir darüber hinaus umgestaltet, so dass kritische Komponenten nun nicht mehr von uns, sondern von freien Open-Source Projekten upstream maintained werden. Dazu haben wir den von uns selbst entwickelten Identity Management Service gegen das Open-Source Projekt Kratos ausgetauscht.

Die Integration des neuen Identity Management Services und die Umgestaltung der Kernkomponenten erwies sich aufwendiger als geplant wodurch sich unser Fokus während der Förderungsrunde ein wenig verschoben hat. Wir haben einige Meileinsteine deshalb nicht erreichen können. Zur Umgestaltung der Administrationsoberfläche sind wir leider nicht mehr gekommen und wir hätten gerne wir auch noch mehr Apps integriert, wofür wir am Ende leider keine Zeit mehr hatten (wie zum Beispiel Jitsi als freie Video-Konferenzlösung).

Vor allem mit Blick auf die Zusammenarbeit mit upstram Projekten haben wir durch das Projekt neue Erkenntnisse gewonnen. Wir haben gelernt, das die Zusammenarbeit mit anderen Projekten zwar sehr fruchtbar ist, aber viel Arbeit erfordert und das Fortschreiten weniger planbar machen kann, da oft nicht unmittelbar klar ist wie sehr sich Fortschritte durch Entwicklungen beim upstream Projekten verzögern können.

Im Blick auf die Zusammenarbeit mit dem OKF haben wir ausschließlich positive Erfahrungen zu berichten. Die unkomplizierte Zusammenarbeit hat es und stets ermöglicht uns voll auf unsere Arbeit zu konzntrieren.

Künftige Weiterentwicklungen

Greenhost als Haupt-Maintainer des Projektes plant, OpenAppStack Anfang 2022 im Produktivbetrieb einzusetzen - nicht nur intern sondern auch als Service, der ihren Kund*innen angeboten wird. Damit ist der Grundstein für eine nachhaltige Weiterentwicklung gelegt und wir hoffen, dass sich OpenAppStack in naher Zukunft zu einem großen Teil selbst finanzieren wird. Wir sind froh, dass unser Beitrag zu dieser aussichtsreichen Zukunft beitragen konnte.

In ein paar Wochen wird ein Beta-Test stattfinden, an dem interessierte Organisationen teilnehmen können. Wir stellen interessierten dann einen OpenAppStack Cluster zur Verfügung um potenziellen Nutzern zu ermöglichen uns Feedback zu geben.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Keine

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wir haben zum Ende der Förderung ein neue Version veröffentlicht (siehe auch den englischen Openappstack 0.7.0 release blogpost [1]). Wie diese installiert werden kann, ist in der ebenfalls aktualisierten Dokumentation [2] beschrieben.

Die Openappstack Website [3] ist aktualisiert worden und listet die neu hinzugefügten Applikationen auf.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Arbeiten am Fundament von Openappstack haben mehr Zeit gebraucht als geplant, weshalb wir nach der Hälfte des Förderzeitraums umplanen mussten. Aus diesem Grund sind leider die oben Erwähnten Meilensteine (Umbau des Administrations-Frontends und Integration weiterer Apps) leider nicht erreicht worden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Openappstack wird hauptsächlich von der Firma Greenhost [4] entwickelt und war wie viele andere freie Software-Projekte auch vom politischen Kahlschlag beim Open Technology Fund unter der Trump-Administration [5] betroffen, was einen abrupten Auszahlungs-Stopp im letzten Jahr zur Folge hatte. Zeitgleich zur PTF-Förderung erhielt OAS eine Anschluss-Förderung vom OpenCollective [6], so dass 2 Teams nach langer Pause mit 2 unterschiedlichen Förderungen mit der Weiterentwicklung starteten. Dies war sehr hilfreich da es ansonsten schwierig gewesen wäre ein ordentliches Code-Review unserer Merge requests zu gewährleisten. Generell war die Zusammenarbeit sehr gut.

Allerdings hat die langsame Weiterentwicklung seitens Greenhost in den ersten Fördermonaten dazu geführt dass wir mehr Arbeit für die Grundfesten von OAS aufbringen mussten als am Anfang geplant, um eine gute, stabiles Grundgerüst zu bekommen mit dem wir unsere geplante Arbeiten durchführen konnten.

[1] <https://openappstack.net/2021/08/23/Openappstack-0.7.0-release.html>

[2] <https://docs.openappstack.net/en/v0.7>

[3] <https://openappstack.net>

[4] <https://greenhost.net/>

[5] <https://netzpolitik.org/2020/politischer-kahlschlag-trifft-open-technology-fund>

[6] <https://opencollective.com/openappstack>

Richtlinie zum „Software-Sprint“

B3scale – Projekttitel

Schlussbericht

Zuwendungsempfänger:

Annika Hannig

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS17S03** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin.

Kurze Darstellung der Aufgabenstellung und Motivation

Seit Beginn der Covid-19 Pandemie wurde der Bedarf an Lösungen für den digitalen Remote-Unterricht kontinuierlich größer. BigBlueButton hat sich dabei als gute, nicht proprietäre und datenschutzfreundliche Lösung im Einsatz an Bildungs- und Schuleinrichtungen etabliert.

Eine besondere Herausforderung ist dabei ein effizienter Betrieb von großen Server-Clustern um wirtschaftlich konkurrenzfähig und somit eine attraktive Alternative zu bestehenden nicht-freien Angeboten zu bieten.

Bestehende Lösungen hatten sich dabei in der Praxis als instabil gezeigt oder hatten einen unzureichenden Funktionsumfang.

Die bisher am weitesten verbreitete Skalierungslösung „Scalelite“ ermöglichte zwar einen Betrieb von BigBlueButton-Clustern zur Last-Verteilung, allerdings musste für jede Einrichtung ein separates Cluster zur Verfügung gestellt werden. Dies führte dazu, dass viele Instanzen weitestgehend ungenutzt verblieben.

Für eine effiziente Auslastung mussten alle Einrichtungen das selbe Cluster ohne gegenseitige Interferenz nutzen können. Grundsätzlich sollte dabei allerdings die Kompatibilität mit Scalelite erhalten bleiben.

Weiter sollten für unterschiedliche Integrationen individuell einstellbare Optionen wie die hinterlegte Standard-Präsentation ermöglicht werden.

Eine weitere Anforderung war die Verteilung von Meetings anhand von Kriterien die für jedes Frontend einstellbar sein mussten um beispielsweise dedizierte Bereiche zeitweise zu reservieren.

B3scale selbst musste ebenfalls einen Cluster-Betrieb ermöglichen um Ausfallsicherheit zu gewährleisten.

Die Konfiguration sollte mittels einer API erfolgen um eine Automation des Betriebs zu ermöglichen.

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

Kommunikation ist ein grundlegendes Bedürfnis des Menschen und ein fundamentaler Stützpfeiler unserer Gesellschaft.

Der Betrieb von Kommunikationsinfrastruktur ist etwas, was allen Menschen zu gute kommt. Dabei ist Offenheit der Schnittstellen und Offenheit der verwendeten Software wichtig um die Integrität und Sicherheit zu gewährleisten und Zugangshürden abzubauen.

B3scale ist dabei ein weiteres Werkzeug um die Verbreitung von offenen Lösungen wie BigBlueButton zu fördern. Dies wirkt zusätzlich einer Software-Monokultur, die anfällig für Schad- und Erpressungssoftware ist entgegen und fördert die europäische Datensouveränität.

Eine intelligentere Auslastung der Server ist nicht nur wirtschaftlich sondern auch mit Hinblick auf Umweltschutz und Green-IT wünschenswert.

Ausführliche Darstellung der Ergebnisse

B3scale wurde in der Architektur eines klassischen Proxy-LoadBalancers implementiert: HTTP-Anfragen werden entgegengenommen und dekodiert. Für das Routing wichtige Attribute wie die "MeetingID" oder die API-Ressource werden dabei strukturiert. Mit dieser Anfrage wird nun das Gateway aufgerufen und der Ablauf einer Middleware-Kette angestoßen.

In einzelnen Middlewares wird die Anfrage aufbereitet und nach Bedarf verändert. So kann beispielsweise die Standard-Präsentation bei Erzeugung eines Raumes angepasst werden. Am Ende der Kette sitzt eine Handler-Middleware, welche die BigBlueButton-API implementiert. In den meisten Fällen bedeutet das, ein Backend auszuwählen und die Anfrage dort hin weiterzuleiten. Anfragen, die aus dem Zustand der Applikation direkt erzeugbar sind werden ohne hinzuziehen der BigBlueButton-Backends beantwortet. Dazu gehört unter anderem die Auflistung der derzeit laufenden Meetings im Cluster.

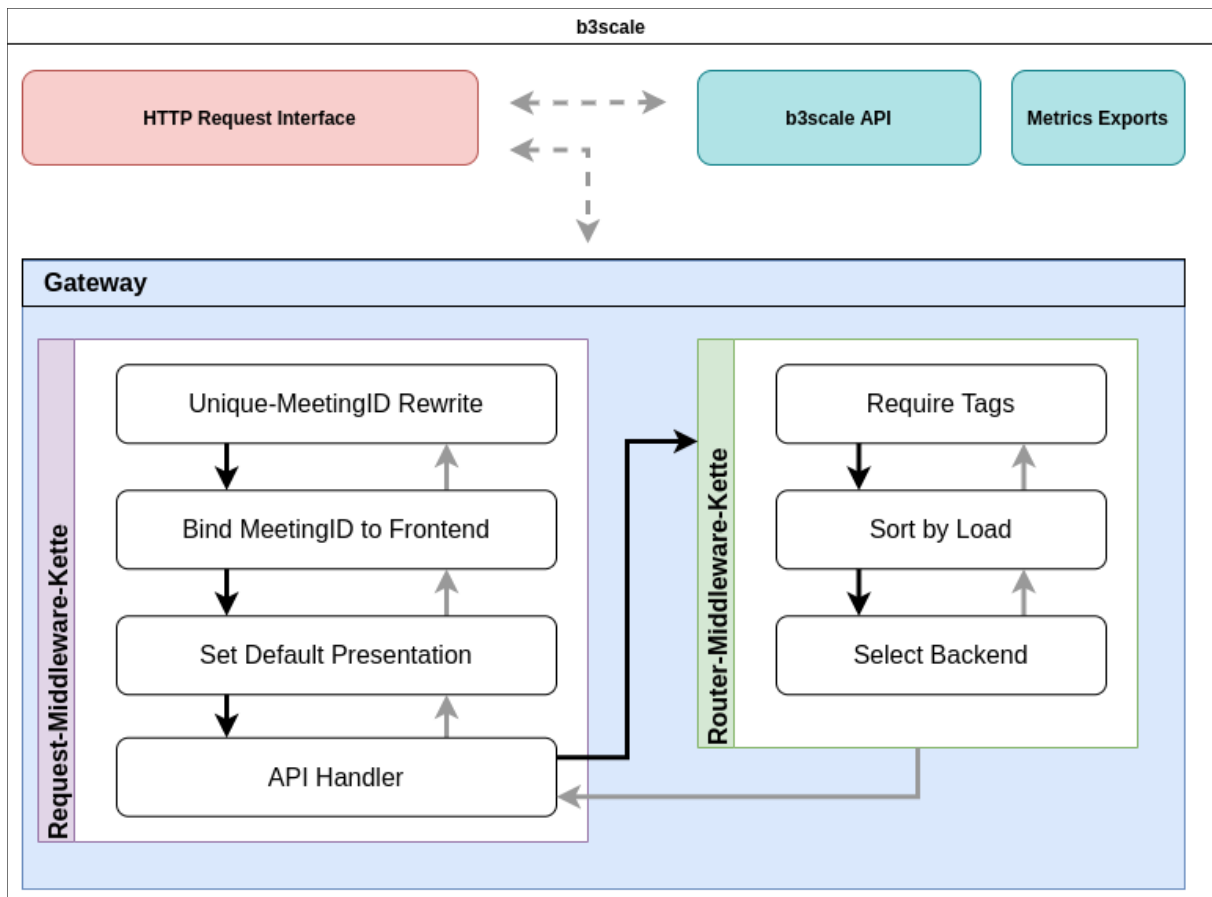


Abb. 1: Middleware-Ketten zur Verarbeitung einer BigBlueButton Anfrage.

Muss zur Beantwortung einer Anfrage auf ein BigBlueButton-Backend zugegriffen werden, so muss dies zunächst durch den Router ausgewählt werden. Die Anfrage wird dabei, analog zum Gateway, durch eine Middleware-Kette geleitet und BigBlueButton-Backends anhand von Kriterien selektiert oder verworfen.

Hier wird ebenfalls die Sortierung der Backends nach Last und die Partitionierung des Clusters mittels Tags durchgeführt.

Einzelne Middleware-Module, wie beispielsweise das zum Einstellen der Standard-Präsentation bei Raum-Erzeugung erfordern eine persistente Konfiguration zur Laufzeit des Clusters. B3scale stellt allen Middlewares eine Schnittstelle zur Speicherung und Abfrage von Einstellungen zur Verfügung. Diese Schnittstelle wird ebenfalls über die HTTP-API angeboten. Diese wurde nach dem Prinzip einer RESTful-API implementiert. Die Authentifikation erfolgt mittels JsonWebToken. Als Austauschformat wird JSON verwendet.

Genutzt wird diese API von der b3scalecli Terminal-Anwendung, um Backends und Frontends zu verwalten und Konfigurationsvariablen zu setzen.

Durch die API kann die Verwaltung des BigBlueButton-Clusters in Automatisations- und Verwaltungs-Lösungen integriert werden.

Periodische Aufgaben, wie das Abgleichen der Raum-Listen mit den Backends, oder deren Prüfung auf Erreichbarkeit werden nebenläufig, im Hintergrund, durchgeführt.

Durch die Anforderung des Cluster-Betriebs von b3scale, müssen diese Aufgaben auf die einzelnen Instanzen verteilt werden. Dazu wurde eine Job-Queue als Teil des gemeinsamen Zustandes implementiert. Alle Instanzen verarbeiten diese, allerdings kann jede Aufgabe nur durch jeweils eine Instanz gleichzeitig selektiert werden. Ausgenutzt wird dabei der Row-Level-Locking-Mechanismus der Postres-Datenbank.

Die Aufgaben werden dadurch auf die einzelnen b3scale-Instanzen aufgeteilt.

Um eine größtmögliche Stabilität bei der Verarbeitung der Aufgaben zu erreichen, wird die Job-Queue periodisch abgefragt.

Um den Zustand des BigBlueButton-Backends mit dem von b3scale synchron zu halten werden periodisch Meeting-Informationen von Backends abgefragt. Eine hohe Frequenz hätte dabei eine große Server-Last zur Folge. Mit einer niedrigen verliert im Laufe der Zeit die interne Zustandsrepräsentation und damit lastverteilungs-relevante Werte, wie die Anzahl Teilnehmender, ihre Genauigkeit.

Für das Erreichen einer möglichst exakten Abbildung des Backend-Zustandes im Cluster wird durch b3scale auf interne Ereignisse (Events) von BigBlueButton zurückgegriffen: Ein auf dem Backend-Server installierter Dienst (b3scalenoded) abonniert die von BigBlueButton produzierten Ereignisse auf dem lokal laufenden Redis-Cache und aktualisiert den geteilten Zustand des Clusters. Wird ein Raum betreten oder Verlassen, wird so diese Information unverzüglich dem Cluster zugänglich gemacht und kann so in Routing-Entscheidungen einfließen.

Dieser Dienst kann so gestartet werden, dass neue Backends automatisch am Cluster registriert werden. Dies erleichtert die Inbetriebnahme weiterer Instanzen.

Erfassung von Fehlern in Protokollen im Betrieb ist ebenfalls ein wichtiger Bestandteil von b3scale: Detailgrad und Ebene des Protokolls können eingestellt werden. Für die Ausgabe kann dabei entweder eine klassische, textorientierte oder eine strukturierte Variante gewählt werden: Bei der strukturierten Protokollierung erfolgt die Ausgabe der Nachrichten in einem JSON-Format und kann dadurch im operativen Betrieb besser durch eine zentrale Protokollerfassung verarbeitet werden.

Zentral für den Betrieb von Clustern ist die permanente Erfassung der Zustände der einzelnen Systeme. B3scale stellt dazu über HTTP Metriken zur Verfügung,

die durch Datenbanken wie Prometheus (oder kompatibel) erfasst und als Zeitserien aufgearbeitet werden können.

Zu den exportierten Daten gehören sowohl Kennzahlen über die Applikations-Instanz selbst, wie Speicherverbrauch oder Threads, als auch die Anzahl von Meetings und Teilnehmenden im Cluster.

Alle Meilensteine, die für einen Produktivbetrieb von B3scale notwendig waren konnten erreicht werden. Zu diesem Zeitpunkt fehlt noch eine Unterstützung der in BigBlueButton integrierten Aufnahme-Funktion. Diese befindet sich aktuell noch in Entwicklung, wurde aber aufgrund datenschutzrechtlicher Bedenken mit niedriger Priorität verfolgt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die primäre Zielgruppe sind Betreibende von BigBlueButton-Clustern. Diese können bereits jetzt vom Einsatz von b3scale profitieren, wenn sie auf Aufnahmen verzichten können.

Die transparente Entwicklung in einem öffentlichen Repository, führte dazu, dass bereits früh Systemadministrierende auf das Projekt aufmerksam wurden und Probleme meldeten, die schnell behoben werden konnten.

Mögliche Weiterentwicklungen sind die Implementierung einer „Reverse-Proxy-Funktionalität“ zur Maskierung der einzelnen Server. Hier sind ggf. Änderungen an BigBlueButton selbst nötig. Die Integration der Aufnahme-Funktionen von BigBlueButton soll ebenfalls fertiggestellt werden.

Ein besonderer Lern-Erfolg war die effiziente Handhabung von Datenbank-Verbindungen und Transaktionen auf einer niedrigen Ebene in einer verteilten, hoch verfügbaren, Anwendung.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Einige Konzepte, zur optimalen Nutzung von Datenbank-Verbindungen und Transaktion wurden getestet und auf ihre Stabilität unter Last untersucht. Verworfen wurde dabei die Nutzung von einer langlebigen Transaktion für jede HTTP Anfrage. Die Verbindungs-Allokation innerhalb der Anfragen-Verarbeitung wurde ebenfalls verworfen. Letztlich wurde eine Verbindungs-Allokation am Beginn der Anfrage und die Verwendung mehrerer möglichst kurzlebiger Transaktionen innerhalb der Anfragenverarbeitung gewählt.

Ein erster Prototyp einer Funktion zur Maskierung der einzelnen BigBlueButton-Instanzen („Reverse-Proxy“) des Clusters wurde implementiert, aber verworfen, weil sich die vollständige Integration als derzeit unpraktikabel ohne Änderungen am Code von BigBlueButton selbst herausgestellt hat.

Die erste Implementierung der Job-Queue nutzte einen Publish-Subscribe Mechanismus der die Einzelnen Instanzen über die Präsenz einer neuen Hintergrundaufgabe informierte. Dieser Mechanismus erwies sich allerdings unter Last als instabil und wurde durch eine periodische Abfrage ersetzt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

B3scale wurde von Anfang an in einem öffentlich einsehbaren Gitlab-Repository entwickelt: <https://gitlab.com/infra.run/public/b3scale>

Ein Umzug zu Github (<https://github.com/b3scale/b3scale>) zur Senkung der Partizipationshürde ist vorgesehen.

Weiter geplant sind veröffentlichte Container z.b. in der Github-Container-Registry.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Im wesentlichen verlief alles nach Plan. Das regelmäßige veröffentlichen des Arbeitsfortschrittes im Chat des Prototypefundes hat dabei geholfen den Überblick über noch offene Aufgaben zu behalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Die Entwicklung an Scalelite geht weiter. Die noch ausstehende Unterstützung für Recordings kann noch ein Hinderungsgrund für den Umstieg auf b3scale sein.

Richtlinie zum „Software-Sprint“

SARchive

Schlussbericht

Zuwendungsempfänger:

Zemke Ippen GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S04 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: Täglich steigen Menschen in seeuntüchtige Boote, um nach Europa zu gelangen. Zivile Seenotrettungsorganisationen werden zunehmend für ihre Arbeit kriminalisiert und wegen vorgeschobener Mängel oder Quarantäneauflagen davon abgehalten, Seenotrettung zu betreiben. Teilweise schaffen die Boote den Weg alleine nach Lampedusa. Häufiger allerdings werden die Menschen von der sogenannten libyschen Küstenwache aufgehalten und zurück nach Libyen gebracht, was nach internationalem Recht eine Menschenrechtsverletzung darstellt. Das Seerechtsübereinkommen der Vereinten Nationen (UNCLOS) besagt, dass aus Seenot gerettete Menschen nicht an einen Ort gebracht werden dürfen, an dem ihnen Verfolgung, Gewalt oder Folter drohen, was jedoch in Libyen der Fall ist.

Neben diesen sogenannten Pull-Backs nach Libyen gab es in den letzten beiden Jahren erneut mehrere Unglücke, bei denen Menschen ertranken, weil Staaten wie Malta oder Italien Meldungen zu Booten einfach ignorierten und die Menschen nicht aus Seenot retteten, obwohl sie rechtlich dafür zuständig waren.

Diese Schiffsunglücke und Menschenrechtsverletzungen, in denen Menschen zurück nach Libyen gebracht werden oder europäische Staaten sie wissentlich ertrinken lassen, werden nur teilweise dokumentiert. Unterschiedliche zivile Akteur:innen sammeln vereinzelt Daten. Insgesamt mangelt es aber an einer integrierten Übersicht der verfügbaren Informationen. Mithilfe des SARchives möchten wir die unterschiedlichen Datensammlungen zusammenführen, um eine möglichst lückenlose Dokumentation der Vorfälle aufzubauen. Dies trägt auch dazu bei, einen bestmöglichen Überblick zu bekommen und mögliche Unklarheiten zu Vorfällen klären zu können.

Vorgehensweise: Die kartenbasierte OneFleet App, die in der 7. Phase des PTF gefördert wurde und Bewegungen auf dem Mittelmeer und Informationen zu Booten in Seenot trackt, liefert die aktuellen Daten für das SARChive. Aber auch der Import anderer Datenquellen wie etwa Excel Tabellen ist möglich. Das SARChive nutzt die bestehende Open-Source Anwendung Apache Superset. Des Weiteren nutzen wir Delta Lake zum Speichern der Daten sowie Dagster mit Spark als Pipeline.

Wichtigste Milestones:

1. Gemeinsam mit User:innen definieren, welche Daten gesammelt werden sollen
2. Durch die Zusammenarbeit und das Einholen von Feedback herausarbeiten, wie Daten aufbereitet werden müssen und welche Funktionen und Tools (wie Such- und Filterfunktionen) das SARChive braucht, um für Recherche- und Studienzwecke genutzt werden zu können.
3. Diskussion und Neubau des SARChives
4. Anpassung des Datenbankschemas in Adaption zur OneFleet App
5. Implementierung eines Services für die automatisierte Migration der Daten
6. Zusammenführung der bisherigen unterschiedlichen historischen Datensammlungen ziviler Seenotrettungs-NGOs
7. Ausarbeitung des Erstellens von automatisierten Reports
8. Ausarbeitung eines gruppenbasierten Access- und Usermanagements
9. Erstellung eines Open Data Portals für ausgewählten Daten

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 des SARChives wollen wir ihnen ein Tool zur Verfügung stellen, das die Dokumentation verbessert und effizienter macht. Aber auch Journalist:innen und Forscher:innen gehören mit zur Zielgruppe, denn ihnen wollen wir die Ergebnisse und Teile der Datenbank für Forschungs- und Recherchezwecke zugänglich machen.

Die aktuelle Runde des Prototype Funds hatte keinen zusätzlichen Themenschwerpunkt. Unser Projekt liegt im Bereich Civic Tech. Das SARChive ist eine neue Anwendung, die es der Zivilgesellschaft erlaubt, Daten zu erzeugen, sinnvoll und zielgerichtet zu nutzen und zu teilen. Das SARChive ermöglicht es, Daten zu visualisieren und aufzuklären, um somit Transparenz im Diskurs zu Migration über das Mittelmeer herzustellen.

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?

Nach einer langen Phase des Feedback Einholens und Gesprächen mit User:innen haben wir gemerkt, dass viele Funktionen wie etwa das automatisierte Erstellen von Statistiken, Filter- und Suchfunktionen benötigt werden. Nach einigen Diskussionen und Abwägungen haben wir beschlossen, das SARchive, wie es ist, zu verwerfen und eine schon bestehende Open-Source Lösung zu nutzen, und an unsere Bedürfnisse anzupassen. Wir nutzen nun Apache Superset.

Da also viel Zeit in diese ungeplante Neuausrichtung des Projekts investiert werden musste, konnten wir leider nicht alle Milestones erreichen. Es konnte noch kein gruppenbasiertes Access- und Usermanagement ausgearbeitet werden (8.) und die Erstellung eines Open Data Portals (9.) wurde nicht begonnen.

Die gemeinsamen Gespräche mit anderen Projekten, die von der Open Knowledge Foundation gefördert wurden, fanden wir stets interessant und bereichernd. Wir haben viel von anderen lernen und mitnehmen können. Der gemeinsame Austausch beispielsweise über verschiedene Tools, die die Arbeit erleichtern, war sehr aufschlussreich für uns und einige nutzen wir nun für unserer Arbeit.

Des Weiteren hatten wir ein gemeinsames Coaching mit Simply Secure zum Thema UX. Dabei hat sich herausgestellt, dass uns unabhängig vom Thema UX generelle Strukturen und Arbeitsprozesse fehlen. Diese haben wir dann selbstständig aber mit vorheriger Hilfestellung des Coaches erarbeitet.

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 unterstützen. Indem wir ihre Dokumentationsarbeit erleichtern und vereinfachen, nehmen wir ihnen Arbeit ab und sie können ihre Ressourcen auf andere Dinge konzentrieren, wie der aktiven Seenotrettung. Durch ein übersichtliches und möglichst lückenloses Archiv stellen wir den Organisationen Daten zur Verfügung, die sie für weitere Arbeiten nutzen können, etwa im Media und Advocacy Bereich.

Unser Projekt ist ein Archiv zu Vorfällen im Mittelmeer. Generell kann es aber als Lösung für jegliche Art von Archiven genutzt werden. Konkrete Ideen für andere Archive gibt es noch nicht, aber wir freuen uns sehr, wenn die Open-Source Community und andere Akteur:innen auf uns zukommen, um das Projekt aufzugreifen und auf andere Kontexte auszuweiten.

Wir haben viel zu den Themen Projektmanagement und Projektkoordination gelernt und wie man das Online-Zusammenarbeiten gestalten und organisieren kann. Außerdem haben wir uns fachlich in Apache Superset 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?

Wie oben erwähnt, wurde das „alte“ SARchive verworfen. Dies wurde ursprünglich in einem Front- und einem Backend entwickelt. Wir haben dann aber gemerkt, dass es schon viele bestehende Open-Source Lösungen gibt, die wir auf unsere Bedürfnisse anpassen können und wir das Rad nicht neu erfinden müssen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Unser Projekt ist als Open-Source Projekt auf Gitlab unter <https://gitlab.com/coGIS> 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?

Natürlich hat der ganze Neubau des SARchive unsere Zeitplanung grundlegend verändert. Milestone 4, Anpassung des Datenbankschemas in Adaption zur OneFleet App, war aufwendiger als gedacht, da User:innen sehr häufig immer wieder mit neuen Vorschlägen und Wünschen kommen, welche Daten sie sammeln möchten.

Da die Projektleitung auch die Aufgabe hatte, User:innen Research zu betreiben und im ständigen Austausch mit den User:innen stand sowie für jegliche Kommunikation zuständig war, haben wir für diese Punkte mehr Stunden gebraucht, als erwartet. Dies spiegelt sich auch in der Endabrechnung wieder.

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, es gab keine Entwicklungen anderer Personen oder Institutionen, die Einfluss auf unsere Arbeiten gehabt hätten.

04. Okt. 2021

Eingangsnr.:

5799/11A

ERechungsviewer – Quba Viewer

Schlussbericht

Zuwendungsempfänger:

Jochen Stärk

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S05 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)?

Eine Stärkung von Datenschutz und Datensicherheit im Bereich elektronische Rechnungen durch eine Stärkung der Souveränität der Empfänger indem eine bessere semantische Prüfbarkeit von elektronischen Rechnungen gewährleistet wird. Dies unterstützt EntwicklerInnen, die Ihre Rechnungen probeweise visualisieren können sowie Personen, die keine Integration des entsprechenden Rechnungsformates in Ihr Rechnungseingangswesen haben oder diese (wie möglicherweise Wirtschafts- oder Steuerprüfer) nicht nutzen wollen.

Außerdem erlaubt es eine schrittweise Migration um zum kompletten Workflow strukturierter e-Rechnungen zu migrieren: Aufgrund Richtlinie 2014/55/EU sind der Behörden (sowie der größte Teil der Zulieferer) seit November 2020 verpflichtet, strukturierte e-Rechnungen zu unterstützen.

Meilensteine:

- Logo, Namensfindung, Projekthomepage und -infrastruktur (Git etc)
- Technische Vorbereitung öffentlich verfügbarer XML-Transformationen. Im Hinblick auf der vermutlich zu verwendenden Transformationsbibliothek Saxon.js bedeutet dies beispielsweise auch eine Umwandlung von XSLT in SEF.
- XRechnungen/CII visualisieren können: XRechnungen sind als reine XML-Dateien leichter zu unterstützen als das PDF-hybride Factur-X/ZUGFeRD-Format

- XRechnungen/UBL visualisieren können: Neben dem UN/CEFACT Cross Industry Invoice (CII)-Format erlaubt die Norm (EN16931) zur Richtlinie (2014/55/EU) auch das XML-Format Universal Business Language UBL
- Veröffentlichung Alpha
- Darstellung von PDFs mit der Open-Source-Bibliothek PDF.js
- XML aus Factur-X/ZUGFeRD extrahieren und bis einschließlich des Profils EN16931 visualisieren
- Erweiterung der XSLT-Style Sheets auf das Factur-X/ZUGFeRD Extended-Profil
- Veröffentlichung der Anwendung

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

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

Durch das Projekt werden elektronische Rechnungen, wie sie von Behörden bereits gefordert werden, erfahrbarer und einsehbarer und verlieren den „Schrecken“ auf überkomplizierte Weise möglicherweise Dinge „verbergen“ zu können. Außerdem eignet sich der Viewer für Entwickler, die selbst elektronische Rechnungen anbieten möchten, als semantisches Test-Tool um beispielsweise zu verifizieren, dass die benutzten Felder richtig ausgespielt werden. Logischerweise erlauben strukturierte elektronische Rechnungen Effizienzverbesserungen durch einen Medienbruchfreien Workflow.

Während des Projektzeitraums wurde zudem klar, dass Frankreich wie Italien bereits 2019, tatsächlich in absehbarer Zeit verpflichtend E-Rechnungen nicht nur im B2G sondern auch im B2B-Bereich einführen wird. Hintergrund waren unter anderem Mehrwertsteuer-Mehreinnahmen in Italien durch Rückgang des Umsatzsteuerbetruges. Visualisierung ist Teil der Anforderung dieser Einführung in Frankreich und in Deutschland gibt es Initiativen, diese Einführung ebenfalls anzustreben. So hat der Bundesrechnungshof das Thema erwähnt, der Wissenschaftliche Dienst des Bundestages hat es untersucht und eine Partei hat verpflichtende e-Rechnungen für den B2B-Verkehr in Deutschland gefordert.

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?

Version 1.0 wurde am 23.08.2021 veröffentlicht. Ein Beitrag durch die Firma Symtrax in Form der französischen Übersetzungen führte zum Release der 1.1 am 20.09.2021. Alle Meilensteine wurden

erreicht, letztlich die Anzahl der Factur-X-Extended-Attribute in den XSLTs sind aber noch sehr begrenzt.

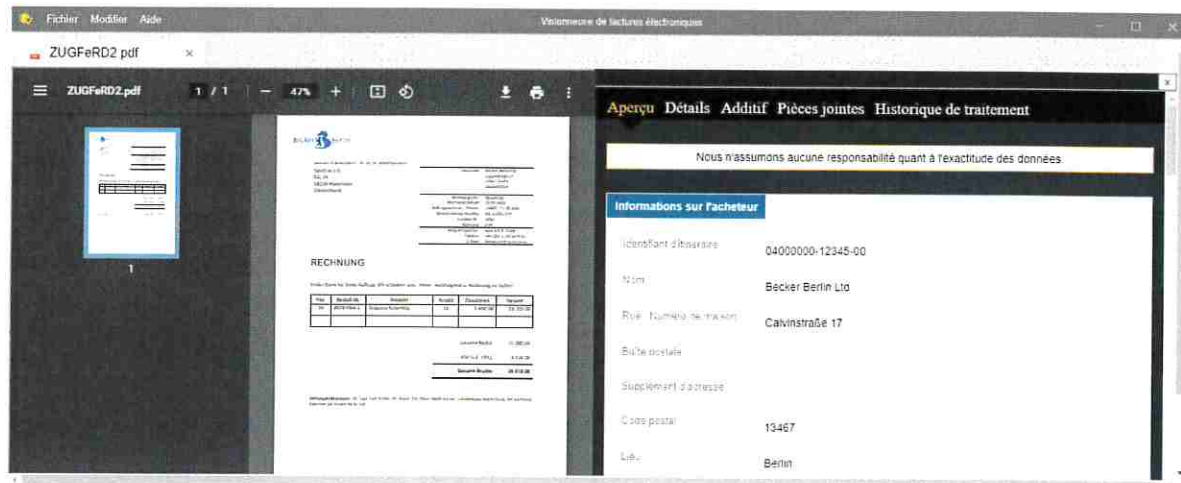


Schaubild 1: Screenshot der französischen Ansicht von Quba 1.1 mit einem geöffneten deutschen Hybrid-PDF: Thumbnail- und Seitenansicht (Links bzw. Mitte), Visualisierung des ins PDF eingebetten Rechnungs-XMLs (Rechts).

Die gewonnenen Erkenntnisse waren einerseits Erfahrungen technischer Natur, wie zur Interprozesskommunikation zwischen dem darstellenden Browser und dem Backendprozess im Electron-Framework. Andererseits gab es eine Anregung von anderen Teilnehmern der Runde 9 über die Accessibility des Projekts. Das hat zu einer intensiveren und nicht ganz uninteressanten Auseinandersetzung zum Thema formeller und reeller Blindenfreundlichkeit geführt.

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?

Die Zielgruppen können sich die Rechnungen jetzt ansehen und prüfen. Die Tatsache, dass zusätzlich auch Blinde (siehe oben) und fremdsprachige Nutzer Rechnungen so besser verstehen können hat sich erst in der Projektlaufzeit herauskristallisiert.

Die Veröffentlichung als Open-Source hat

1. während der Entwicklung unter anderem den Maintainerwechsel erleichtert, seit der 1.1 kümmert sich im Wesentlichen jemand anders um die Wartung und Weiterentwicklung,
2. den Beitrag durch Symtrax sowie

3. dessen Präsentation auf dem E-Rechnungsgipfel in Neuss 2021 erleichtert.

Hätte es sich um eine kommerzielle Anwendung gehandelt wäre damit zu rechnen gewesen, dass der Einsatz von Symtrax-Mitarbeitern für die Belange einer anderen Firma hinterfragt worden wäre und möglicherweise Rückfragen zu einem möglichen „Werbeblock“ auf dem E-Rechnungsgipfel hätten entstehen können.

Weiterhin ermöglicht die Veröffentlichung als Open-Source, insbesondere im programmiersprachenunabhängigen XSLT, anderen Visualisierern die Möglichkeit, Rechnungen in englisch oder französisch darzustellen und/oder auch Factur-X Extended zumindest teilweise zu nutzen: Für ZUGFeRD 1.0 steht die entsprechende XSLT-Datei noch als Open-Source zur Verfügung, ab ZUGFeRD 2.0 wurde sie leider unter proprietärer Lizenz weiterentwickelt.

Persönlich viel gelernt habe ich unmittelbar über die Übersetzung von XSLT-Dateien und mittelbar (durch die Teilnahme am E-Rechnungsgipfel zur Vorstellung der Version 1.1) beispielsweise über Länder mit CIUS, Marktanteile von ZUGFeRD und XRechnung und das EDI-Netzwerk Peppol. Ziemlich interessant war die Tatsache, dass Quba, mit seiner EN16931-UBL-Unterstützung bereits Peppol-Rechnungen visualisieren kann.

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?

Es gab nicht weiter verfolgte Lösungsansätze tatsächlich nur in Bereichen die ohnehin nicht durch die Projektanmeldung abgedeckt waren und damit nicht mitfinanziert wurden. Es gab zum Beispiel Proof of Concepts zum Thema

- Einbau von Validierern, also einer syntaktischen und rechnerischen Prüfung,
- eine informelle Anfrage ob man einzelne Attribute auch beschreiben könnte oder
- einen Beitrag zum Thema Mapping offizieller IDs auf die Rechnungsattribute.

Die Arbeiten haben gezeigt, dass und wie das jeweils möglich ist und ermöglichen eine Weiterführung der Arbeiten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Neben der offiziellen Website mit dem Deutsch/Englisch/Französischem Download für Windows, Linux und Mac steht der Quelltext auf GitHub öffentlich unter liberaler Open-Source-Lizenz (APL2) zur Verfügung.

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?

Unerwarteterweise funktionieren weder bestimmte Online-Buildservices noch war ein Cross-Compiling möglich, was bei einer Javascript-Hybridanwendung eigentlich zu erwarten gewesen wäre. Deshalb wurde privat ein Mac angeschafft.

Es gab keinen Mehraufwand der finanziert hätte werden müssen. Der letzte Entwicklung-Meilenstein (Factur-X/ZUGFeRD Extended) war absichtlich in einem einigermaßen breiten Korridor definiert. Mehr Ergebnisse in diesem Bereich wurde von vielen kleinen Problemen verhindert, in denen man eher einen schnellen Fortschritt erwartet hätten, wie Probleme mit Build, Deployment und den Automatischen Updates. Eigentlich war die Erwartung, dass das verwendete und mittlerweile einigermaßen ausgereifte Framework (Electron) diese Funktionen einfach und zuverlässig zur Verfügung stellt.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten?

Die Recherche vor der finalen Einreichung ergab zwei existierende offline e-Rechnungsviewer ([Ultramarinviewer](#) und [Open XRechnung Toolbox](#)). In der Folge wurde die Zielsetzung von Quba präzisiert (beispielsweise Darstellung von PDFs und Factur-X/ZUGFeRD-Unterstützung).

Auch im Zusammenhang mit dem UX Coaching ist aufgefallen, dass Attributcodes ausgeschrieben und nicht nur als Code aufgeführt werden sollten, ansonsten würde dargestellt, dass 15 „H87“ in Rechnung gestellt werden statt 15 Stück (H87=Piece). Interessanterweise hat genau die Open-Xrechnung-Toolbox diese Fähigkeit als [Open-Source-XSLT](#) zur Verfügung gestellt. Diese Funktionalität konnte dann in Quba auch sehr einfach eingebaut werden.

Richtlinie zum „Software-Sprint“

WaldbrandApp – Waldbrand-App-Baukasten

Schlussbericht

Zuwendungsempfänger:

Zielinski & Kürten GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S06 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 Wald als klassischer Offlineraum stellt Einsatzkräfte trotz 4G und digitalem Geodatenportal vor Herausforderungen. Anders als in der Stadt, fehlt es im naturnahen Raum an zuverlässigem Internet, sodass selbst die Bemühungen des Geoportal Forst mit seiner digitalen Waldbrandschutzkarte nur bedingt greifen. Hinzu kommt, dass das Kartenmaterial dort nur verzögert aktualisiert wird. Feuerwehrentnahmen brauchen jedoch zuverlässige Daten und die gibt es speziell im Fall der Wasserentnahmestellen nicht. Lückenhafte Daten finden sich zudem bezüglich Befahrbarkeit von (Wald-)Straßen. Bisher behilft sich die Feuerwehr mit einer Mischung aus analogen und digitalen Lösungen, die sehr mühevoll ist. Dieses Problem wollten wir mit einer digitalen Lösung angehen, die gleichzeitig auch eine Zusammenarbeit aller Feuerwehren in ganz Brandenburg ermöglicht. Folgende Meilensteine waren geplant:

- Alle nötigen Kartenkomponenten ausfindig machen
- App entwickeln und bestehende Daten und Karten integrieren
- Karteneditierfunktion bereitstellen durch Aufsetzen einer privaten OpenStreetMap-Instanz inklusive Standard-Editor (wahrscheinlich iD)
- Bereits vorhandene Basisdaten dort hinein importieren, Feuerwehrleute zur Eingabe schulen
- Baukasten entwickeln und vertesten
- Mit Baukasten gebaute Map im (Test)Einsatz vertesten
- Code laufend auf Github veröffentlichen

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?

Zielgruppe ist die brandenburgische Feuerwehr. Mit 38.000 Einsatzkräften hat die freiwillige Feuerwehr den größten Anteil. Zudem gibt es 9 Werk- und 5 Berufsfeuerwehren sowie die Landesfeuerweherschule in Eisenhüttenstadt. Das Tool soll die Zielgruppe durch unseren Kontakt zur Landesfeuerweherschule, persönliche Kontaktaufnahme und Pressearbeit erreichen. Die Zielgruppe profitiert direkt von den Ergebnissen, weil sie von Projektbeginn an mit einbezogen wurde und das praktische Tool mit getestet hat. Durch die Waldbrand App kann die brandenburgische Feuerwehr netzunabhängig auf den gängigen Androidgeräten auf einsatzrelevante Informationen zugreifen und diese auch noch selbstständig pflegen. Durch unsere Projektwebseite und die Veröffentlichung der App im Google Play Store, hat die Feuerwehr Zugriff auf unsere Projektergebnisse. Wir arbeiten mit offenen Daten und veröffentlichen den Code des Projekts unter einer offenen Lizenz, sodass auch andere mitarbeiten und das Projekt nach ihren Bedürfnissen weiterentwickeln 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?

Wir haben die wichtigsten Meilensteine erfolgreich erreicht. Dazu zählt die Veröffentlichung der Offline-App und die Entwicklung des Editors zur Kartierung von Wasserentnahmestellen. Diese Kernbestandteile des Projekts haben wir frühzeitig umgesetzt und dadurch bereits während der Laufzeit zahlreiche Nutzer gewonnen, was sich in den Downloadzahlen der App und den registrierten OpenStreetMap-Nutzern für den Editor widerspiegelt. Die geplanten Meilensteine haben wir somit erreicht, auch wenn es unser Ziel war, mehrere verschiedene Datenthemen in die App zu implementieren. So wollten wir beispielsweise auch Munitionsverdachtsflächen und für Feuerwehren befahrbare Wege integrieren. Letztlich umgesetzt haben wir zwei Datenthemen: Wasserentnahmestellen und Rettungspunkte. Neben der App haben wir auch noch einen Online-Editor bereitgestellt. Der Editor hat dabei die Funktion, bestehende Wasserentnahmestellen (Hydranten, Brunnen, Löschteiche, Saugstellen, Wasserbehälter) anzuzeigen, bearbeiten zu können und neue hinzuzufügen. Zu jeder Wasserentnahmestelle kann ein Kommentar und Bilder hinzugefügt werden. Zudem kann der Innendurchmesser der Zuleitung bei Hydranten eingetragen werden, woraus die Durchflussleistung errechnet wird. Bei Wasserbehältern kann das Volumen eingetragen werden. In der Offline-App lassen sich alle Wasserentnahmestellen mit Detailinformationen sowie die Rettungspunkte anzeigen. Zusätzlich gibt es hier auch Koordinaten für jeden ausgewählten Datenpunkt und ein Fadenkreuz mit GPS-Ortung des Standortes. Durch eine Suchfunktion lassen sich Straßen und POI auf der Karte leicht finden. Offline-App und Online-Editor sind durch eine Verlinkung mit der ausgewählten Kartenpositionierung verknüpft.

Uns war es wichtiger am Projektende eine lauffähige und stabile Lösung für die Nutzer zu haben, sodass wir uns auf die Optimierung konzentriert haben und die thematische Erweiterung

zurückstellten. Wir haben weiterhin gelernt, dass der Kontakt und Umgang mit Nutzern sehr wichtig ist, um das Projekt einerseits durch deren Feedback stetig zu validieren und um es andererseits bekannt zu machen. Gleichzeitig nimmt diese Aufgabe auch viel Zeit und Energie in Anspruch, dennoch haben wir früh mit der Vernetzung begonnen, welche uns auch sehr von der Open Knowledge Foundation empfohlen wurde.

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 unser Projekt war es äußerst sinnvoll, dass die Datenlage nicht nur lokal bei der jeweiligen Feuerwehr gepflegt wird, sondern dass auch ortsfremde Feuerwehren auf vollständige Daten zugreifen können. Da wir mit offenen Daten arbeiten und keinerlei Restriktionen gesetzt haben für die Sichtbarkeit oder Bearbeitung der relevanten Daten in Brandenburg, konnte und kann eine bundeslandweite Datenbank aufgebaut werden, die jedem Feuerwehrmann und jeder Feuerwehrfrau auch in Großschadenslagen helfen kann. Während es Projekts gab es Ideen und Wünsche für weitere Inhalte und Funktionen seitens unseres fachlichen Beraters aus der Feuerwehr und den Nutzern. Wir haben alle Ideen aufgenommen und die Prioritäten während der Projektlaufzeit angepasst. Entstanden ist zudem eine neue Meilensteinliste für Erweiterung der Waldbrand-App, die in einer zweiten Entwicklungsrunde umgesetzt werden könnte. Davon abgesehen gab es den Wunsch nach einer App-Version für iOS, welcher ebenfalls für die Weiterentwicklung aufgenommen wurde.

Die Projektarbeit hat unsere interne Projektorganisation verbessert und uns Tools an die Hand gegeben, die wir sicherlich beibehalten werden. Unsere Kompetenz im Umgang mit Stakeholder und Nutzern wurde ebenfalls geschärft.

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?

Der ursprüngliche Plan war es, die Daten des offenen Datenportals vom Landesbetrieb Forst für die Pflege der Wasserentnahmestellen und andere Daten zu nutzen. Wir waren uns jedoch nicht sicher, in welchem Umfang dies möglich sein würde. Es stellte sich heraus, dass die Daten zwar genutzt werden könnten, eine Aktualisierung der Daten (ein Rückfluss unsererseits an das Datenportal) jedoch nicht möglich ist. Deshalb haben wir bezüglich der Wasserentnahmestellen eine andere offene Datenbank gewählt (OpenStreetMap), mit welcher sich Daten bearbeiten und hinzufügen lassen. Dieser Punkt war essenziell für unser Projekt, da sich die Problemlage speziell auf die Unvollständigkeit der Wasserentnahmestellen bezog. Wir erfuhren weiterhin den Grund für die Unvollständigkeit beim Landesbetrieb Forst. In deren Datenbank werden nur die Forst-relevanten Wasserentnahmestellen gespeichert, also nur jede, die sich in hoheitlichen Waldgebieten befinden und diese sind wiederum in Zuständigkeiten unterteilt, welche die Datenpflege bezüglich der Wasserentnahmestellen in ihrem Gebiet übernehmen müssen. Für die Feuerwehr sind jedoch auch

Wasserentnahmestellen in Waldnähe relevant, daher erachten sie den Datensatz des Landesbetriebs nicht als umfassend genug.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Projektwebseite und Editor: <https://waldbrand-app.de/>

Android App: <https://play.google.com/store/apps/details?id=de.waldbrandapp.brandenburg>

Code: <https://github.com/waldbrand>

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?

Wir konnten innerhalb der Projektlaufzeit und zudem auch termingerecht zu unseren geplanten Meilensteinen die wichtigsten Projektbestandteile umsetzen. Von der Kosten- und Arbeitsplanung musste somit nicht abgewichen werden. Wir halten uns offen, nach der Projektlaufzeit weiterhin an kleinen Optimierungen zu arbeiten, die sich als wünschenswert von unseren Nutzern herausgestellt haben. Das betrifft beispielsweise die visuelle Darstellung von Elementen in der App.

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 hat sich bei unserer Arbeit herausgestellt, dass es diverse ähnliche Lösungen auf dem Markt gibt, die von Feuerwehren genutzt werden. Dabei handelt es sich meist um kommerzielle Lösungen. Diese scheinbaren Konkurrenten haben uns zwei Aspekte gezeigt. Einerseits stellen sie nicht wirklich Konkurrenzprodukte dar, weil sie sich in der Infrastruktur deutlich von uns unterscheiden, indem sie nicht offen und partizipatorisch sind und indem sie kostspielig sind. Andererseits erfüllen diese bereits existierenden Lösungen einen Zweck für die Feuerwehren und so haben wir teilweise das Feedback erhalten, dass die Feuerwehren nicht zwei Systeme pflegen wollen. Unsere Bestrebungen beziehen sich daher auf die Zukunft und umfassen einerseits den technischen Ausbau unserer Lösung, um sie inhaltlich konkurrenzfähig zu machen und andererseits den Mehrwert aufgrund der Partizipation durch den OpenSource-Ansatz als großen Vorteil herauszuarbeiten. Dadurch wollen wir die Waldbrand App von anderen Lösung abgrenzen.

Schlussbericht

Zuwendungsempfänger:

Jonathan Meyer

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S07 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Ausgangspunkt war kollaborative Archive und spielerische Geschichtsschreibung als Technologien für den Kulturbereich umzusetzen. Hier gibt es besondere Herausforderungen: die Überlieferung ist vielfältig (vom Flyer, über Fotos bis zu Überbleibseln von Performances oder Clubnächten, von Kostümen bis zu Zeichnungen) und das Publikum ist nicht an Archivstandards gewohnt, sondern kultiviert Überlieferung hauptsächlich auf kommerziellen Social Media Kanälen mit immensen Datenschutzproblemen. Hier wird jegliche Souveränität über Daten an kommerzielle Akteure abgegeben. Es gibt also einen Bedarf kollektive Autor*innenschaften auf Archive bezogen mit Technologie zu fördern.

Das Vorgehen war eine Mischung aus Recherchen im Bereich Digital Humanities und insbesondere Archivsoftware (Daten-Triples, Linked Open Data) und Interfaces (Activating Environments), sowie zur Zielgruppe (Clubkultur). Darauf folgte die technische Konzeption (Design Thinking) und die Umsetzung von Backend und Frontend. Test-Events der Community-Archivierung waren während der Umsetzung Teil des Vorgehens. Zur Öffentlichkeitsarbeit/Vermittlung wurde eine Dokumentation als Knowledge Base angelegt.

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

Geschichte wird nicht von allen geschrieben. Doch die Art und Weise, wie Dokumente und Objekte in großen Archiven und Museen gespeichert und beschrieben werden, betrifft alle, diese Kulturtechnologie ist mit Politik und Governance von oben nach unten verbunden. Während die Digitalisierung in diesen Silos oft als Prozess der Demokratisierung oder zumindest der Transparenz verstanden wird, ist auch das Gegenteil der Fall: Die zentrale und hierarchische Macht dieser Institutionen wird in den Spektakelwelten der digitalen Medien und der kybernetischen Governance vervielfacht. Dann gibt es eine Geschichte von Gegenarchiven und alternativen Archiven, die sich den

Motiven und Praktiken der großen Institutionen entgegenstellen. Mit cooArchi soll erprobt werden, wie in einem digitalen Kontext neue und dynamische Wege für antihierarchische und experimentelle Formen des kollektiven Schreibens und Dokumentierens möglich sind.

Kommerzielle soziale Medien haben ein neues Zeitalter der Medien- und Archivgemeinschaften geschaffen. cooArchi ist als basisdemokratische Alternative gedacht: viele cooArchi-Instanzen anstelle von Facebook-Gruppen oder Google-Drives. Kommerzielle Social-Media-Plattformen sind ein wichtiger Ort, an dem sich Medienarchive von Gemeinschaften bilden. Um die Macht dieser Datenextraktionsunternehmen zu brechen, stellt das cooArchi Interface nachhaltigere und ethischere Wege der Kollaboration und Interaktion vor.

Ausführliche Darstellung der Ergebnisse

Der cooArchi-Prototyp wurde von der alternativen Club- und Festivalszene bzw. von Raves im Allgemeinen inspiriert. Die Veranstaltungen und Räume in diesem Kontext sind oft temporär und performativ, was ein immaterielles und ephemeres Erbe impliziert.

In einem cooArchi können solche Beziehungen und Nicht-Beziehungen in einem Netzwerk von Sätzen, längeren Texten und Medien geschrieben werden. Dieses Experiment, eine Software zu schreiben, die von der Rave-Kultur inspiriert ist und von ethischen Designprinzipien geleitet wird, steht in krassem Gegensatz zum üblichen Umgang mit dem kulturellen Erbe. Wir haben aber auch an Museen gedacht und hoffen, dass zukünftig das pluralistische Beschreibungssystem und die semantikbasierte Koproduktion in cooArchi in etablierten Kontexten und Institutionen übernommen wird.

Die Software beinhaltet eine neuartige Eingabemaske im Frontend (drei Felder Maske zur Eingabe von Triples) und ein Backend für die Community-Arbeit (Invites, etc.). cooArchi verfolgt einen streng pluralistischen Ansatz, um ein möglichst breites Spektrum an Daten zu erfassen. Für den Prototyp bedeutet das, dass der Hauptunterschied der eingegebenen Daten Elemente und ihre Beziehungen sind. Diese werden dann zu Kanten und Knoten in der Netzwerklogik des Archivs. cooArchi unterscheidet nicht zwischen Daten und Metadaten, alle Daten in einem cooArchi stehen in einem netzwerkartigen Zusammenhang. Das Konzept ist eine Datensculptur auf dem Bildschirm, das heißt, die Daten haben eine haptische Qualität. Man kann die Daten nicht nur als Landschaft mit Scrollen und Zoomen durchqueren, sondern auch einzelne Knotenpunkte ziehen.

Durch die Fokussierung auf einen unbegrenzten Informationsraum und eine sehr reduzierte Formalisierung der Datenfelder versucht cooArchi, unvorhersehbare Arten von archivarischen Stimmen einzubeziehen, von der Erfindung der Terminologie bis zum kreativen Mediengebrauch und einer Vielzahl von Möglichkeiten, strukturierten Text zu schreiben, von Subjekt-Prädikat-Objekt bis zu weniger semantischen und mehr assoziativen Möglichkeiten, Elemente und ihre Beziehungen zu schreiben. Das Datenmodell von cooArchi eignet sich besser für Gemeinschaftsarchive als die Datenbanken kommerzieller soziale Medien, während es Aspekte der Netzwerkkultur wie Live-Interaktion und Inhaltsverkettungen aufgreift. Andere Aspekte wie die Unterscheidung von Beitrag und Kommentar in sozialen Medien und die damit verbundene hierarchische Baumstruktur werden in einem cooArchi aufgelöst: die Netzwerke von Daten, die in einer cooArchi gespeichert werden,

sind dezentral, z.B. können Medien andere Medien beschreiben, da keine Form von Text oder Medien zentral ist und keine Beschreibung wichtiger ist als eine andere Beschreibung.

Zusätzliche Erkenntnisse betreffen früher auf Folgeförderung zu schauen, früher zum Abschluss und Testing mit Bug-Fixing zu kommen, auch früher auf die Coaching Angebote zurückzugreifen, die hilfreich waren.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Mit der Eingabemaske von cooArchi wird ein neuer Weg vorgestellt gemeinsam an Daten zu arbeiten. Dies konnte bereits mit der Zielgruppe erprobt werden, zum Beispiel auf einem großen Musikfestival im August 2021. Die Kernidee von cooArchi ist sehr einfach: Wenn Menschen dreiteilige Informationen schreiben, können diese als Knoten und Kanten dargestellt werden. Das könnte sogar auf einem Whiteboard oder einer Wand geschehen. Die digitale Visualisierung dient nur dazu, diesen Daten eine skulpturale Form zu geben, sie in Clustern zu sortieren und sie wie eine Informationslandschaft durchquerbar zu machen. Da sich die Datenfelder von dem unterscheiden, was die Nutzer von anderen digitalen Archiven oder Diensten kennen, könnte die intuitive Nutzung eingeschränkt oder sogar frustrierend sein. Daher verfügt die Schnittstelle über eine Startseite und eine Anleitungsseite, auf der eine Einführung in die Funktionalität der Schnittstelle gegeben wird.

Die Effekte der Open-Source-Stellung sind noch nicht abzuschätzen, erste Ideen der Weiterentwicklung betreffen unrealisierte Ideen wie das dialogische Eingeben (das sogenannte Fragespiel), aber auch neu entstandene Bedarfe wie Clustern und Filtern des Graphen. Der Prototyp ist im Bereich digitales Kulturerbe und Museen auf Interesse gestoßen und für diesen Kontext sollen neue Features entwickelt werden, angestrebt mit einer neuen Prototypförderung 2022.

Die Arbeit am Projekt war für alle (auch die ehrenamtlich) Mitwirkenden eine intensive persönliche Phase und hat vielfältig zur fachlichen Weiterentwicklung beigetragen, durch die Recherchen, die Konzeption, die Testings und die Entwicklung selbst.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ein Moderationsinterface konnte nur begrenzt umgesetzt werden, es stellte sich heraus dass dies zu viel Aufwand für den Projektrahmen war. Da die Schnittstelle darauf abzielt, die Autorität von den Institutionen auf die Nutzer zu verlagern, ist die Moderation der Inhalte ein komplexes Problem, da sie mit einem Machtungleichgewicht einhergeht. Zwei Ansätze erschienen interessant. Erstens, keine Löschoption im Backend zu haben, sondern die Freiheit des Zuhörens: User*innen könnten entscheiden, was und wen sie nicht sehen wollen, indem sie andere oder einzelne Einträge blockieren. Zweitens, eine Löschoption im Backend zu haben, aber immer den Grund für die Löschung zu dokumentieren. Momentan gibt es nur die zweite Option, um denjenigen, die ein cooArchi installieren, die Möglichkeit zu geben, zum Beispiel schnell auf Hassreden oder Doxxing zu reagieren. Dieser Ansatz setzt das Vertrauen zwischen User*innen und Admins voraus. Im gemeinschaftsorientierten Ansatz von cooArchi ist diese Beziehung als sozial und nicht technologisch ausgehandelt zu verstehen. Die Admins sollten sich um User*innen kümmern und explizit dokumentieren, wo sie die Grenze für unerwünschte Inhalte ziehen, z.B. auf der Grundlage des Berlin Code of Conduct. Auch sollten die Admins die Gründe für die Löschung transparent machen.

Das dialogische Interface stellte sich auch als zu aufwändig für den Projektrahmen heraus, der Fokus musste erst auf der Kernfunktionalität liegen (der Eingabe von Datentriples). Die Hauptansicht der Dateneingabe besteht aus 3 leeren Feldern. Bestehende Einträge können dort eingefügt werden, um zusammenhängende Informationen wachsen zu lassen. Eine andere Möglichkeit der Schnittstelle die skizziert wurde wäre ein Fragespiel: Die Schnittstelle würde bestehende Datentripel nehmen und sie als Fragen umstrukturieren. Wenn es einen Eintrag A Relation B gibt, wäre eine Möglichkeit, nach Beziehungen zu A zu fragen, oder einfach generische Fragen wie "Was ist" A oder "Wo ist" B. Die Antworten würden dann als neue Tripel in der cooArchi gespeichert werden.

Der Plan einer eigenen flexiblen digitalen Ontologie wurde verworfen, zugunsten einer freien Eingabe von Triples. Die Datenontologie ist damit sehr reduziert. cooArchi unterscheidet nur zwischen Element und Beziehung (als Knoten und Kanten im Netz). Elemente können einzelne Wörter, längere Texte in einem Feld oder Mediendateien sein. Im ersten Prototyp kann ausgewählt werden, dass ein Eintrag ein Ort ist, oder dass er eine Triggerwarnung trägt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Webseite: <https://www.cooarchi.net/>

Dokumentation (technisch und politisch): <https://cooarchi.github.io/documentation/>

Github: <https://github.com/cooarchi/>

Newsletter: <https://lists.riseup.net/www/info/cooarchi>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Arbeits- und Kostenplanung wurde eingehalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Museen (u.a. Mitarbeiter*innen des Museums für Naturkunde) äußern Interesse an der Software um sie im Ausstellungskontext und zur Communityarbeit einzusetzen. Dafür scheint sie aber noch nicht geeignet, daher ist die Idee eines Folgeprojekts entstanden, eines zweiten Prototyps der mit den Erfahrungen aus dem Archivinterface für Clubkultur eine Version für die kollaborative und partizipative Datenarbeit in Museen entwickelt: cooArchi 2.0.

Richtlinie zum „Software-Sprint“

Klinik-DB: Die freie Krankenhaus Datenbank

Schlussbericht

Zuwendungsempfänger: **Köther-Schubert-Rentzsch-Molitor-GbR**

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS21S08** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren (Ulf Köther, Malte Schubert, Steven Rentzsch und Philipp Molitor).

Kurze Darstellung der Aufgabenstellung und Motivation

Im Jahresverlauf 2020 wollten wir uns aus Eigeninteresse einen umfassenden Einblick in die deutsche Krankenhauslandschaft der letzten Jahre verschaffen, insbesondere durch die Nutzung der sogenannten Qualitätsberichte der Krankenhäuser (QBK). In kurzen Worten zusammengefasst beinhalten diese Berichte detaillierte Angaben jeder deutschen Klinik, die in die Planung zur Versorgung gesetzlich versicherter Patienten einbezogen ist. Eine jede solche Klinik muss diese Berichte standardisiert und jährlich an den Gemeinsamen Bundesausschuss (GBA) übermitteln. Von diesem kann auf schriftlichen Antrag hin eine Kopie dieser Berichte bezogen werden. In der Gesundheitswirtschaft wurde zwar schon lange kolportiert, dass einige technische Hürden auf dem Weg zur sinnvollen Nutzung dieser Daten existieren, aber bis zu unserem eigenen Versuch in dieser Angelegenheit war uns nicht bewusst, wie aufwändig sich dieser Prozess eigentlich darstellt. Da die Berichte nur im XML-Format auf DVD an die Beantragenden geliefert werden, liegen sie nicht in einem für eine berichtsübergreifende Recherche oder weitere Analyse geeigneten Format vor. Andere Quellen für QBK-Daten existieren zwar, sind aber entweder nur in aggregierter Form (z.B. Destatis), oder nicht Klinikvergleichend bzw. nicht maschinenlesbar abrufbar (z.B. „Weisse Liste“). Daher haben wir uns mit dem Projekt Klinik-DB folgende Ziele gesetzt: Wir wollten möglichst viele Berichtsjahre vom GBA anfordern (die ursprüngliche Planung sah 10 Jahre vor), um sie automatisiert und validiert in eine SQL-Datenbankstruktur zu überführen, und diese SQL-Datenbank dann Dritten zur Verfügung stellen, einmal per Datenbank-Dump, also zur Reproduktion der gesamten Daten auf einem eigenen Datenbankserver, und einmal per REST-API, also als strukturierte Programmierschnittstelle, damit z.B. Softwareanbieter oder andere interessierte Dritte Daten aus den QBK für eigene Anwendungen nutzen können. Die Software für diesen Prozess der Datenextraktion aus dem originalen XML-Dateien wollten wir veröffentlichen, so dass in Zukunft auch Andere daran partizipieren bzw. davon profitieren können. Zusätzlich wollten wir über vorgefertigte Analyseumgebungen interessierten Nutzern Beispiele an die Hand geben, was dann mit den Rohdaten an Analysen möglich ist. Die ursprüngliche Planung sah vor, hierfür containerisierte Beispiele mit R (Shiny) und in Python (Bokeh oder Dash) zur Verfügung zu stellen.

Geplante Meilensteine im Förderzeitraum:

- Anfordern von 10 Jahren an Qualitätsberichten der Krankenhäuser vom GBA
- Erstellung eines Software-Parsers, der das automatisierte Einlesen und Überführen der Daten in eine SQL-Datenbank übernimmt
- Damit einhergehend Erarbeitung eines entsprechenden Datenbankmodells
- Identifikation und Bereitstellung von zusätzlichen Datensätzen, die im Zusammenspiel mit den QBK inhaltlich relevante Analysen ermöglichen (z.B. Bevölkerungszahlen)
- Bereitstellen der gesamten Daten als Download und über eine REST-API
- Bereitstellung von Beispielanalysen in R und Python als Docker-Container, so dass interessierte Nutzer die Analysen innerhalb einer lokalen Analyseumgebung nutzen können

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

Spätestens durch die Corona-Krise wurde nochmal der unmittelbare Zusammenhang zwischen politischen Entscheidungen über die deutsche Kliniklandschaft und den daraus resultierenden gesellschaftlichen Auswirkungen klar ersichtlich. Es wurde aber auch deutlich, dass es nur auf Basis von Daten und Fakten möglich ist, solche politische Entscheidungen fundiert von außen zu beurteilen, und die dafür notwendige Datengrundlage muss für die Allgemeinheit transparent und einfach verfügbar sein.

Beim Förderprogramm „Software Sprint“ liegt der Fokus¹ auf „kreativen Ideen mit Bezug zu gesellschaftlich relevanten Lösungen in einer datengetriebenen Welt, [...] bei der innovative Modelle und Werkzeuge zur Erzeugung, zum Teilen und zur Nutzung von Daten gefragt sind. Die Lösungen sollen es ermöglichen, dass sich die Öffentlichkeit stärker an gesellschaftlichen Entwicklungen beteiligen kann und beim souveränen Umgang mit Daten und Informationen unterstützt wird.“

Vor dem Hintergrund dieser Definition glauben wir klar sagen zu können, mit dem Projekt Klinik-DB den Anspruch an durch den „Software Sprint“ zu fördernde Projekte zu erfüllen: Wir haben gesellschaftlich relevante Daten einer breiteren Öffentlichkeit zugänglich gemacht, haben die Allgemeinheit in die Lage versetzt, mit diesen Daten souverän umzugehen, und haben Werkzeuge veröffentlicht, die diesen Prozess in Bezug auf die QBK-Daten auch in Zukunft vorantreiben können.

Ausführliche Darstellung der Ergebnisse

Der erste Schritt war das Schreiben eines R-Pakets (<https://klinik-db.gitlab.io/qbgbaReader>), womit die XML-Berichtsdateien automatisiert eingelesen und geparsed werden können. Parallel haben wir ein einheitliches SQL-Datenbankschema erstellt (ebenfalls im Sourcecode von qbgbaReader verfügbar) und die Daten in einer entsprechenden Datenbank hinterlegt. Dieser Vorgang wurde für die letzten fünf verfügbaren Berichtsjahre (2015 bis 2019) vollzogen. Viel Arbeit erforderte hierbei das Matchen eines Krankenhauses über die verschiedenen Jahre hinweg, da sich herausstellte, dass dies nicht automatisiert möglich ist, da die einzelnen Berichte keine reliable übergreifende Identifikationsmöglichkeit bieten. Hier war leider „Handarbeit“ notwendig, um die Informationen so miteinander zu verknüpfen, dass die Kliniken über die Jahre verfolg- und vergleichbar sind. Einen

¹ Adaptiert von <https://www.softwaresysteme.pt-dlr.de/de/software-sprint.php>, abgerufen am 13.09.2021.

Anfangspunkt lieferten uns zwar die sogenannten „Positivlisten“ des GBA, die zwar einen übergreifenden ID-Code liefern, diesen aber leider wiederum nicht auf die Daten in den eigentlichen Berichtsdateien matchen, so dass dies per Hand erfolgen musste.

Da uns zur Zeit der Förderbeantragung nur ein Berichtsjahr zur Verfügung stand, haben wir leider im Vorfeld den Umfang dieser Arbeit eklatant unterschätzt, vor allem auch deshalb, weil ältere Berichtsjahre sich in ihrer Datenqualität umso schlechter darstellen desto älter sie sind. Als Nebenbemerkung sei angeführt, dass wir im Nachhinein auch erfahren haben, dass der GBA seit dem Berichtsjahr 2015 mit einem neuen IT-Dienstleister zusammenarbeitet, was ab diesem Berichtsjahr auch die allgemeine Datenqualität spürbar verbessert hat. Jedoch mussten wir aufgrund dieses Umstands unsere ursprüngliche Zeitplanung hinsichtlich des leistbaren Umfangs der Projektarbeit anpassen, weswegen wir in Absprache mit unserer Projektbegleitung und dem Projektträger die einzulesenden Jahre von zehn auf fünf (2015 bis 2019) reduziert haben. Vor dem gleichen Hintergrund haben wir die ursprüngliche Planung, die geplanten Beispielanalysen in R und in Python zu erstellen, auf eine Programmiersprache reduziert, d.h. auf die Nutzung von R und Shiny. So konnten wir zwar nicht alle Meilensteine in ihrem vollem Umfang wie geplant erfüllen, haben aber doch jede der avisierten Grundfunktionalitäten bereitstellen können.

Um die Nutzbarkeit und den Informationsgehalt von Klinik-DB weiter zu steigern, wurden die GBA-Daten wie geplant mit weiteren Drittdaten „angereichert“, die sich für weitere Analysen einfach als weiteres R-Paket (<https://klinik-db.gitlab.io/qbgbaExtraData>) installieren lassen. Damit sich alle User einen Eindruck von den Möglichkeiten der Klinik-DB machen können, haben wir bereits verschiedene Beispielanalysen erstellt, die sich mit den Daten realisieren lassen. Diese haben wir über einen Docker-Container bereitgestellt, in dem eine R-Analyse-Umgebung und eine Shiny-basierte interaktive Weboberfläche enthalten ist (<https://gitlab.com/klinik-db/qbgbaAnalysis>). Damit besteht z. B. die einfache Möglichkeit, sich die Kliniken auf einer interaktiven Karte anzeigen und sich zu jeder Klinik zusätzliche Informationen über die verfügbaren Jahre ausgeben zu lassen. Neben diesem nutzer*innenfreundlichen Analysetool steht ein kompletter Daten-Download für die Jahre 2015 bis 2019 im SQL-Format zur Verfügung: Sämtliche eingelesenen Daten können als vollständiger SQL-Dump (<https://gitlab.com/klinik-db/qbgbaSQLData>) heruntergeladen werden, so dass sich die SQL-Datenbank mit allen Informationen problemlos reproduzieren lässt. Da die Gesamtdaten mit mehr als 2 GB jedoch zu groß für einen Docker-Container sind, haben wir einen kleineren Auszug aus den Daten für die oben genannten Analysen zusammengestellt und bieten diesen auch in unserem R-Datenpaket (<https://klinik-db.gitlab.io/qbgbaExtraData>) an. Darüber hinaus haben wir für die gesamte Datenbank eine maschinenlesbare REST-API erstellt, über die (aus Gründen der Serverkapazität mit gewissen Limits pro Request) jede Tabelle der Datenbank abgefragt werden kann. Für den API-Zugang genügt eine einfache Anmeldung (<https://klinik-db.de/site/klinikdb/api>). Schlussendlich stellen wir eine Websuche zur Verfügung, über die einzelne Krankenhäuser gesucht und verschiedene Datenpunkte aus den Berichten direkt eingesehen werden können (<https://klinik-db.de>). Die Dokumentation für alle diese Softwarebestandteile ist auf den jeweiligen Webseiten verfügbar.

Wir haben in der Förderzeit sehr viel Detailwissen über die Struktur (und die Sinnhaftigkeit?) der QBK gewinnen können, und auch unsere Fähigkeiten im Umgang mit XML-Strukturen wurde zwangsweise und über alle Maßen trainiert. Insgesamt müssen wir klar sagen, dass wir nicht nur über die Förderung durch das BMBF, sondern auch über die Unterstützung durch die Open Knowledge Foundation und die Begleitung durch den Projektträger DLR sehr dankbar waren und von der strukturierten Anleitung profitiert 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?

Da wir den Programmcode für unseren XML-Parser unter einer Open-Source-Lizenz verfügbar machen (R-Package „qbgbaReader“, <https://klinik-db.gitlab.io/qbgbaReader/>), kann das Einlesen der Daten aus den XML-Dateien in eine Datenbank von allen, die den Code nutzen, reproduziert werden, solange sie die GBA-Rohdaten zur Verfügung haben. Außerdem kann der Parser an zukünftige Änderungen an der Berichtsstruktur einfach angepasst und entsprechend auch in der Zukunft verwendet werden. Damit ist der zukünftige Arbeitsaufwand für uns und auch andere interessierte Nutzer, die Daten ab dem Berichtsjahr 2020 einlesen wollen, stark reduziert. Wir gedenken, diese Arbeit auch für zukünftige Berichtsjahre zu leisten (das Berichtsjahr 2020 wird in den kommenden Monaten vom GBA veröffentlicht), und auch die aktualisierte Fassung des Parsers frei verfügbar zu machen, so dass eine Weiterentwicklung des Projekts gegeben ist.

Die Arbeit im Projekt hat uns auch in fachlicher Hinsicht weitergebracht, denn wir haben aufgrund des Zeitdrucks doch relativ viele Bausteine in kurzer Zeit und enger Zusammenarbeit erstellen müssen. Das konsequente Nutzen der entsprechenden Tools und ein engmaschiges Projektmanagement waren dabei von großem Vorteil, und da wir dazu konsequent gezwungen wurden, haben wir uns auch hier inhaltlich und fachlich weiterentwickeln können.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

In dieser Hinsicht gibt es bei unserem Projekt nichts zu berichten, da die technischen Probleme alle gelöst werden konnten, und wir keine Lösungsansätze aufgrund mangelnder Eignung verwerfen mussten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wir haben unsere Ergebnisse in mehrfacher Form veröffentlicht: Als erstes ist die Website <https://klinik-db.de> zu nennen, wo Nutzer anhand einer Web-GUI nach Krankenhäusern suchen und sich aktuelle Daten aus den QBK anzeigen lassen können. Als zweites stehen unter <https://gitlab.com/klinik-db> alle erstellten Softwarepakete inkl. Dokumentation zur Verfügung.²

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wie schon oben beschrieben, haben wir aufgrund der Tatsache, dass wir erst nach Projektbeginn alle Rohdaten einsehen konnten, und somit weder von der fehlenden Identifizierungsmöglichkeit der Kliniken über die Berichtsjahre hinweg und von der schlechten Datenqualität älterer Berichtsjahre keine Kenntnis hatten, den eigentlichen Projektaufwand vorab deutlich unterschätzt. Daher war im Rahmen der Förderung eine vollständige Abbildung auch von Berichtsjahren vor 2015 und auch die Bereitstellung der Analysen in zwei verschiedenen Programmiersprachen nicht realistisch. Durch die dann erfolgte Reduzierung der zu verarbeitenden Berichtsjahre und der Fokussierung auf eine Programmiersprache für die Beispielanalysen konnten wir die Kostenplanung aber letztlich einhalten.

² Für weitere Details der einzelnen Pakete siehe oben.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Die einzige Veränderung in der Verfügbarkeit der QBK-Daten an anderer Stelle waren die vom GBA selbst bereitgestellten „Referenzberichte“, die erstmals mit den 2019er Daten im Förderzeitraum verfügbar gemacht wurden (<https://g-ba-qualitaetsberichte.de/#/search>). Da diese jedoch nur einzeln pro Klinik als PDF-Version abgerufen werden können und teilweise 800-1.000 Seiten pro Klinik umfassen, sind auch sie für übergreifende Analysen vollständig ungeeignet. Wir haben die Referenzberichte jedoch (wie auch vom GBA gedacht) als Hilfestellung für die Überprüfung unserer Parsing-Software nutzen können, um zu sehen, dass wir alle relevanten Rohdaten im Ergebnis richtig extrahieren konnten.

Richtlinie zum „Software-Sprint“

PAID – Pflege-Abrechnung in Deutschland

Schlussbericht

Zuwendungsempfänger:

Kamphausen Schmidtpeter Zwick GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S09 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Für die Abrechnung mit Pflegekassen müssen Pflege-Dienstleister auf proprietäre Software zurückgreifen und sind damit an die übrige Funktionalität dieser Programme gebunden. Abrechnungen in Papierform scheiden als ineffizientere und teurere Alternative aus. Die Spezifikationen der Datenformate und Schnittstellen für den elektronischen Datenaustausch nach § 105 SGB XI sind sehr umfangreich und für Software-Entwickler nur mit hohem Aufwand zu erschließen.

Deshalb entwickeln wir eine Open-Source-Library namens PAID als Softwaremodul für die Abrechnung von Leistungen der Pflege-Dienstleister mit den Pflegekassen in Deutschland. Unser Ziel ist, Pflegekräfte unabhängiger zu machen beim Auswählen geeigneter Software, die sie im Alltag zeitlich entlastet, und gleichzeitig Open-Source-Lösungen als vollwertige und erweiterbare Alternativen im Gesundheitswesen zu stärken.

Um die Anbindung an das Abrechnungssystem zu erleichtern, soll die PAID-Library die Komplexität der Konvertierung und Übermittlung der Abrechnungsdaten gemäß § 105 SGB XI übernehmen und kapseln. Sie soll als eigenständiges Modul in Pflege-Software eingebunden werden können und somit die Entwicklung innovativer Anwendungen vereinfachen. Die PAID-Library soll in TypeScript implementiert werden, damit sie sowohl in webbasierten oder hybriden Client-Apps, als auch serverseitig mit Node.js verwendet werden kann.

Der Datenaustausch zwischen der Library und dem Hauptprogramm soll über eine zu dokumentierende Schnittstelle mittels Methodenaufrufen und strukturierten Daten im JSON-Format erfolgen. Um die Schnittstellen praxisnah zu erproben, planen wir die PAID-Library in die Open-Source-Pflege-Software CoopCare als Referenzintegration einzubinden.

Die wichtigsten Meilensteine des Projekts sind:

- Kostenträgerdateien einlesen und automatisiert aktualisieren
- automatisierte Validierung der Abrechnungsdaten
- Abrechnungsdaten konvertieren in Datenformat gemäß Spezifikation der GKV
- Kryptographische Verschlüsselung der Abrechnungsdaten
- Anmeldung zum Testverfahren und Erprobungsverfahren bei Pflegekassen
- Datenübermittlung nach § 105 SGB XI implementieren
- Testverfahren mit Datenannahmestellen der Pflegekassen durchführen und Fehlerbehebung
- Dokumentation des Datenaustauschs
- Schnittstellenentwicklung der PAID-Library und Einbindung in die Open Source Pflege-Software CoopCare
- Export von Abrechnungen zum Ausdrucken ermöglichen
- technische Projekt-Dokumentation

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

Erstmals ist die aufwändige digitale Anbindung von Pflege-Dienstleistern an das Abrechnungssystem der gesetzlichen Kranken- und Pflegeversicherungen mittels Open Source Software möglich. Dieser Schritt erleichtert es Teams von Pflegekräften, sich selbst zu organisieren und selbst abzurechnen ohne Abhängigkeit von einem Abrechnungszentrum. Dadurch kann Open-Source-Software im Gesundheitswesen zu einer vollwertigen Alternative zu proprietärer Software weiterentwickelt werden, wie beispielsweise die Pflege-Software CoopCare, die die Verbesserung von Mitarbeiterzufriedenheit, Verweildauer im Beruf, Qualität der Pflege, sowie Autonomie und gesellschaftlicher Teilhabe von Pflegebedürftigen zum Ziel hat. Zudem sinkt der Aufwand für Pflege-Dienstleister weitere innovative, bedürfnisorientierte Software zu entwickeln oder anzupassen, da sie auf PAID aufbauen können. Wir erkennen darin eine Ermächtigung von Pflegekräften im Sinne von „Civic Tech“, die der Zielsetzung des „Software-Sprints“ entspricht. Im Rahmen der Förderung wurde mit dem Softwaremodul PAID der Grundstein gelegt, so dass auch weitere Gruppen von Leistungserbringern im Gesundheitswesen PAID mit vergleichsweise kleinen Erweiterungen ebenfalls zur digitalen Abrechnung in ihre Software integrieren können, zum Beispiel orthopädische Schuhmacher oder im Rehabilitationssport.

Ausführliche Darstellung der Ergebnisse

Am Anfang der Projektarbeit stand sowohl die Einarbeitung in die äußerst umfangreiche Dokumentation des GKV-Spitzenverbandes über die elektronische Datenübermittlung, als auch Gespräche mit Pflegediensten um die Bedürfnisse und Anforderungen aus Sicht von Pflegekräften einzubeziehen. Dabei haben wir gelernt, dass Pflegedienste neben der Abrechnung gemäß § 105 Abs. 2 SGB XI in der Regel auch Leistungen nach § 302 Abs. 2 SGB V abrechnen. Somit mussten wir den Umfang des Projektes früh anpassen um die Abrechnung von SGB V Leistungen zu ermöglichen. Zwar gleichen die Prinzipien der Datenübermittlung von SGB V Leistungen denen für SGB XI Leistungen, dennoch unterscheiden sich die Datenstrukturen für die Nutzdatendatei im Detail an vielen Stellen. Aufgrund dieses Mehraufwands konnten wir nicht alle Meilensteine im Förderzeitraum umsetzen.

Es ist uns gelungen, einen Konverter für Abrechnungsdaten im JSON-Format in das EDIFACT-basierte Nutzdaten-Dateiformat für SGB XI und SGB V zu entwickeln entsprechend der GKV-Dokumentation. Dazu gehört auch die Erzeugung von Auftragsdateien, die jeweils mit einer Nutzdatendatei paarweise an die Datenannahmestellen der Kostenträger übermittelt werden. Trotz der zweifellos sinnvollen Ausführlichkeit der Spezifikation des GKV-Spitzenverbandes kamen bei der Entwicklung unsererseits etliche Detailfragen auf, die wir nur durch die Kommunikation mit dem GKV-Spitzenverband klären konnten.

Um den jeweils zuständigen Kostenträger und die zuständige Datenannahmestelle für jede Kranken- und Pflegekasse je Bundesland und Leistungsart als Empfänger der Abrechnungen ermitteln zu können, haben wir einen Parser und für die sogenannten Kostenträgerdateien entwickelt, die diese Kontaktdaten enthalten. Diese Dateien werden auf der Webseite des GKV-Spitzenverbandes öffentlich zum Download bereitgestellt und regelmäßig aktualisiert. Auf der Webseite des ITSG Trust Centers werden die Zertifikate der Datenannahmestellen öffentlich bereitgestellt, deren öffentliche kryptografischen Schlüssel für die Verschlüsselung der Nutzdaten vor der Datenübermittlung erforderlich sind. Wir verknüpfen mit PAID die Daten der jeweils aktuellen Versionen dieser Dateien aus beiden Quellen, und integrieren diese Daten im JSON-Format in die PAID-Library. Zusätzlich haben wir eine Funktion für PAID entwickelt, mit der die Daten bei Bedarf automatisiert heruntergeladen und dynamisch aktualisiert werden können. Bei der Analyse der Kostenträgerdateien sind uns mehrere Inkonsistenzen aufgefallen, so dass wir in zahlreichen Mailwechseln mit den zuständigen Kostenträgern und Datenannahmestellen geklärt haben, inwieweit es sich hierbei um Missverständnisse unsererseits oder Fehler in den Kostenträgerdateien handelte.

Darüber hinaus haben wir einen Parser in PAID integriert um Positionsnummern und Produktdaten von Hilfsmitteln aus dem bundeseinheitlichen Hilfsmittel- und Pflegehilfsmittelverzeichnis zu Abrechnungszwecken in der Library im JSON-Format bereitstellen zu können.

Desweiteren haben wir einen Validator für die Abrechnungsdaten programmiert, die als Parameter von der PAID-Library entgegen genommen werden, um sicherzustellen, dass die Abrechnungsdaten vollständig und konsistent sind um daraus valide Nutzdaten- und Auftragsdateien entsprechend der GKV-Spezifikation erzeugen zu können. Bei Bedarf könnte dieser Validator in Zukunft auch weitergehende Plausibilitätsprüfungen der Abrechnungsdaten vornehmen um sie dem Nutzer als Warnungen vor dem Absenden anzuzeigen.

Nach Fertigstellung des Konverters in das Nutzdatendatei-Format haben wir uns darum bemüht, Kontakte zu Datenannahmestellen herzustellen um ein Testverfahren durchzuführen, wie es laut Dokumentation des GKV-Spitzenverbandes vorgesehen ist. Leider haben wir nur eine einzige Antwort erhalten und zudem nach Ablauf des Förderzeitraums, so dass wir das Testverfahren erst jetzt im Anschluss mit der Datenannahmestelle durchführen können und werden.

Wir haben mit der Integration von PAID in CoopCare begonnen, aber noch nicht abschließen können. Entstanden sind hierbei bis jetzt Benutzeroberflächen für die Erfassung von abrechnungsrelevanten Daten der pflegebedürftigen Klienten, als auch der abrechnungsrelevanten Einstellungen, Kennzeichen und Codes des Pflege-Dienstleisters, sowie zum Teil auch der Vergütungsvereinbarungen zwischen dem Leistungserbringer und den Kassenverbänden.

Letztlich gehört auch zu unseren Arbeitsergebnissen, dass wir den Programmcode der PAID-Library gut dokumentiert haben. Dabei haben wir besonders die Erweiterung von PAID und die Anpassung bei potenziellen zukünftigen Änderungen der Dateiformate und Verfahrensweisen im Blick, in dem wir durch zahlreiche Quellenverweise die Verbindung zur GKV-Spezifikation sichtbar machen.

Durch die Projektarbeit konnten wir unser Wissen und Verständnis über die Regelwerke zur Abrechnung von Pflege-Leistungen mit den Kostenträgern entscheidend vertiefen. Zu diesem Lernprozess gehörten neben den technischen Anforderungen insbesondere die rechtlichen Vereinbarungen zur Vergütung und die Prozessabläufe, etwa dass die Abrechnung von SGB V Leistungen auf der Grundlage ärztlicher Verordnungen erfolgt, während die Abrechnung von SGB XI Leistungen in der Regel auf Kostenvoranschlägen des Pflege-Dienstleisters basieren.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

PAID richtet sich an Entwickler von Software für Pflege-Dienstleister und die Open-Source-Community, um die Anbindung an das Abrechnungssystem der gesetzlichen Kranken- und Pflegeversicherungen mit überschaubarem Aufwand zu erleichtern und einen größeren Anteil von Entwicklungszeit für Innovationen zu ermöglichen. Da PAID freie Software ist, kann die Open-Source-Community die Library mit überschaubarem Aufwand so erweitern, dass auch andere Leistungserbringer im Gesundheitswesen wie beispielsweise Sozialtherapeuten oder Hebammen PAID zur Abrechnung einsetzen und mit ihrer branchenspezifischen Software verbinden können.

Da vom GKV-Spitzenverband immer wieder Anpassungen an der Spezifikation der Dateiformate und der Datenübertragung für die Abrechnung vorgenommen werden, wird es zudem erforderlich sein, diese Änderungen zu verfolgen und den Programmcode von PAID jeweils zeitnah anzupassen.

Gleichzeitig soll PAID einer weiteren Zielgruppe, den Pflegekräften, mehr Freiheit bei der Wahl ihrer Pflege-Software ermöglichen, indem wir Open-Source-Software zu einer vollwertigen Alternative zu proprietärer Software in der Pflege und im Gesundheitswesen weiterentwickeln. Zudem werden Pflege-Dienstleister unabhängiger von Abrechnungszentren, da es leichter wird direkt mit den Kassen abzurechnen. Durch die Integration von PAID in die Open-Source-Pflege-Software CoopCare können die darin erfassten Pflege-Dokumentationen und Durchführungsprotokolle auch für die Abrechnung verwendet und somit doppelte Erfassung vermieden werden. Damit wird der Wechsel zu dieser Open-Source-Alternative attraktiver, zumal CoopCare ohnehin auf die Selbstorganisation von Pflege-Teams, die Verbesserung von Mitarbeiterzufriedenheit und Qualität der Pflege, sowie die Selbständigkeit und soziale Vernetzung von Pflegebedürftigen abzielt. Wir kooperieren mit einem Netzwerk von selbstorganisierten Pflege-Dienstleistern, die diese Software einsetzen wollen und lernen aus dem Feedback dieser Community um CoopCare und PAID bestmöglich auf die alltäglichen Bedürfnisse und Anforderungen von selbstorganisierten Pflegekräften abzustimmen.

Nach einer geschlossenen Test-Phase von CoopCare mit dem integrierten PAID-Abrechnungsmodul ist geplant, die Software als App zu veröffentlichen und den Pflegeteams anzubieten in Verbindung mit der gehosteten Datensynchronisierung zwischen den App-Clients. Es wird angestrebt, für diesen Zweck eine Produktivgenossenschaft zu gründen, die den Pflegekräften einen zuverlässigen, wirtschaftlichen Betrieb, Support und Weiterentwicklung der Software garantiert. Um dies finanzieren zu können, soll für die App-Nutzung eine für Nutzer und Entwickler faire Vergütung vereinbart werden. Auf diese Weise können wir auch die notwendige Aktualisierung und

Weiterentwicklung von PAID sicherstellen. Individuelle Anpassungen an Kundenwünsche und Schulungen sind weitere Bausteine wirtschaftlicher Verwertung.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben damit begonnen, Konverter in das Nutzdatenformat für weitere Leistungserbringer von SGB V Leistungen zu programmieren, wie beispielsweise Leistungserbringer von Hilfsmitteln, Heilmitteln und Krankentransportleistungen. Unser Fokus lag jedoch auf Pflege-Dienstleistern, insbesondere ambulanten Pflegediensten, deshalb haben wir angesichts der begrenzten Zeitressourcen diese zusätzlichen Entwicklungen zeitnah abgebrochen. Wir haben den Programmcode dennoch beibehalten um bei Bedarf diese Entwicklung zu einem späteren Zeitpunkt abzuschließen und PAID damit zu erweitern.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Code-Repository für PAID: <https://github.com/coop-care/paid>

Blogpost zur Demo-Week: <https://demowebk.prototypefund.de/projects/11-paid.html>

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

CoopCare Webseite: <https://www.coopcare.de>

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

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Mit der gewonnenen Erkenntnis, dass nicht nur die Abrechnung gemäß § 105 Abs. 2 SGB XI, sondern auch nach § 302 Abs. 2 SGB V von Pflegediensten für die digitale Abrechnung benötigt wird, haben wir uns entschieden, den Funktionsumfang von PAID entsprechend zu erweitern. Aufgrund der notwendigen Doppelarbeiten und des erhöhten Zeitaufwandes dafür, aber auch allgemein einer tendenziellen Unterschätzung des Aufwandes für die Kostenträgerdateien, die Konvertierung in das Nutzdatenformat und die Integration in CoopCare, konnten wir nicht alle Meilensteine umsetzen. Nichtsdestotrotz wollen wir alle Meilensteine, die wir nicht während des Förderzeitraums abschließen konnten, im Anschluss fertigstellen um unsere Ziele zu erreichen.

Nach der Anmeldung zum Testverfahren bei einer Datenannahmestelle fehlt noch die Durchführung des Testverfahrens für die SGB V und SGB XI Nutzdatenformaten mit anschließender Fehlerbehebung. Außerdem werden wir noch die Signierung und Verschlüsselung der Nutzdatenformaten ergänzen, sowie die Transliteration unzulässiger Zeichen im Zeichensatz. Eine untergeordnete Priorität haben wir der Erfassung und Berechnung von Zuschlägen und Abzügen, sowie der Berechnung von Zuzahlungen durch den Patienten eingeräumt. Beide Funktionen werden wir im Anschluss in PAID ergänzen.

Ein wesentlicher unvollständiger Meilenstein ist die Integration von PAID in CoopCare. Dazu gehört das Programmieren von Benutzeroberflächen für die Erfassung von Verordnungen, den Arbeitszeiten der Einsätze und der Zuordnung von Leistungskomplexen zu Interventionen im Omaha-Klassifikationssystem, sowie die Erstellung von Kostenvoranschlägen und konventionellen Rechnungen in druckbarer Papierform für Leistungen, die privat vom Patienten oder anderen Kostenträgern als den gesetzlichen Kassen bezahlt werden. Letztlich werden wir auch eine Benutzeroberfläche bauen, damit Pflegekräfte die Abrechnungen vor der elektronischen

Datenübermittlung an die Datenannahmestellen prüfen können, gegebenenfalls korrigieren und dann via SMTP versenden. Dazu gehört auch die Anzeige der Empfangsbestätigungen beziehungsweise Fehlermeldungen, die zu Korrekturlieferungen führen können, sowie die digitale Archivierung jedes Datenaustausches zu Dokumentationszwecken.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

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

Richtlinie zum „Software-Sprint“

Lunes

Schlussbericht

Zuwendungsempfänger:

Lunes - Sven Seeberg, Lukas Böhm, Daniel Kehne, Julia Götz GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S10 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Ein wesentlicher Schlüssel für erfolgreiche Integration am Arbeitsplatz ist das Erlernen der Sprache. Für Auszubildende, insbesondere für zugewanderte Fachkräfte, stellt das Erlernen von Fachwörtern eine zusätzliche Herausforderung dar. Viele zu uns gekommene Menschen brechen ihre Ausbildung ab, da die Sprachbarrieren zu hoch sind oder die Verzahnung von Berufsschule und Ausbildung nicht ideal verläuft. Einer der möglichen Problemlöser wurde bereits identifiziert: Fachsprache und berufsbezogenen Wortschatz besser lehren. Lunes sollte eine mobile App werden, die es ermöglicht berufsspezifisches Vokabular zu erlernen. Ohne großen Aufwand sollte Lunes direkt nach dem Herunterladen produktiv mit vorhandenen branchen- und berufsbezogene Stammwortschätzen genutzt werden können. Als begleitende App, die das Lernen in individuellem Tempo ermöglicht, war es unsere Motivation Lunes als digitalen Baustein für berufliche Sprachförderung zu platzieren.

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

Lunes richtet sich nicht nur, aber vorwiegend, an Menschen mit Deutsch als Zweitsprache. Dabei unterstützt die App unter anderem im Übergang von der Schule in Ausbildung und Beruf. Lunes kann als quelloffene Lösung auch von oder gemeinsam mit Unternehmen, Schulen und Integrationsakteuren mit individuellen Vokabellisten und Wörtern aufgesetzt werden und sich somit an speziellere oder eigens definierte Zielgruppen richten.

Als Civic Tech Projekt, das ganzheitlich Open Source entwickelt wurde, hilft es somit auch Sprachkursträgern, Schulen und Integrationsträgern ihre Ziele im Bereich Bildung und Sprachlernförderung zu erreichen.

Ausführliche Darstellung der Ergebnisse

Innerhalb des Förderzeitraums konnten wir – bis auf die angekündigte Pilotphase in Schulen, die sich coronabedingt verzögert hat – alle uns gesetzten Meilensteine erreichen. Wir konnten nicht nur eine vollfunktionsfähige mobile App entwickeln (für Android + iOS), sondern auch eine einfach zu bedienende Vokabelverwaltung, die es ermöglicht Berufe, Lektionen und Vokabeln ohne IT-Kenntnisse einzufügen und zu aktualisieren. Zusätzlich konnten wir Kooperationen mit der Handwerkskammer Braunschweig und dem Bildungswerk der hessischen Wirtschaft schließen, um auch Inhalte (vor allem Stammwortschätze + Bildmaterial) für den Start der App, der zum 01. Juli 2021 erfolgte, zu gewinnen. Die Projektarbeit innerhalb der OKF hat uns geholfen auch im Austausch mit den anderen Projekten eine Lösung zu entwerfen, die nicht nur technisch, sondern auch organisatorisch und politisch gut platziert werden konnte. Die Pilotphase mit den Schulen startet Ende 2021 und wird bis Mitte 2022 andauern.

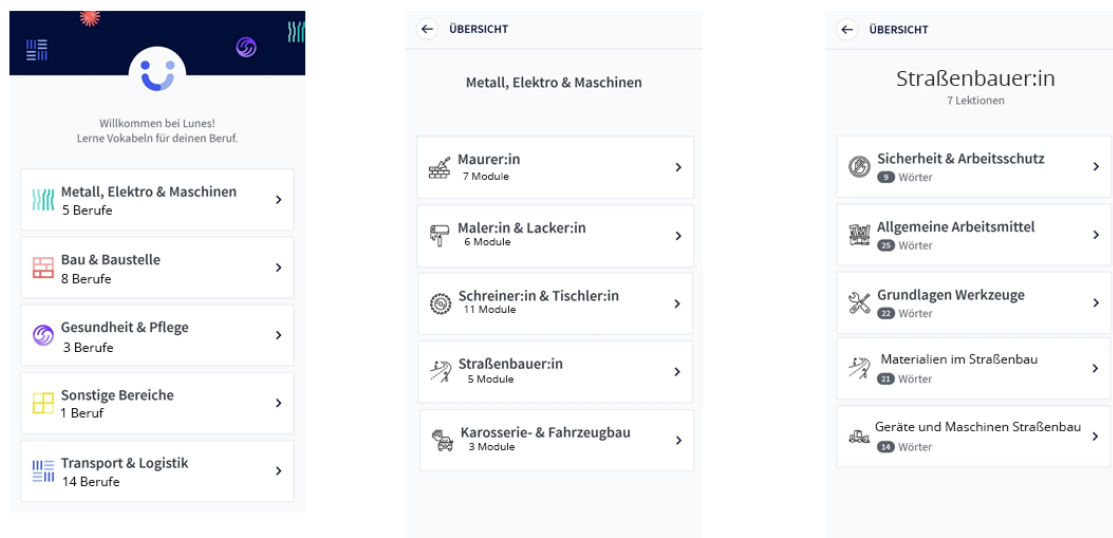


Abbildung 1: Lunes - Screenshots der App

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Profitieren können von Lunes vor allem Schülerinnen und Schüler in Berufsvorbereitungs- und Berufsintegrationsklassen, aber auch Auszubildende und Lehrlinge, die sich am Anfang oder in Mitten ihrer Ausbildung befinden. Durch die Quelloffenheit der Lösung ergeben sich für Schulen und Unternehmen unkomplizierte Individualisierungsmöglichkeiten, um ohne großen Aufwand eigene Vokabelbereiche und somit digitale Lernangebote zu schaffen. Da alle Inhalte (Vokabellisten, Bilder, Audiodateien) ebenfalls frei unter Creative Commons lizenziert sind, sorgt das Lunes-Projekt automatisch für eine freie Datenbank an Vokabelmaterial aus dem sich auch andere Projekte und Organisationen bedienen können. Mit dem Ende des Software Sprints überführen wir Lunes in eine wirtschaftlich agierende, aber gemeinnützige Organisation, um die Weiterentwicklung sicherzustellen und Lunes zu einem unterrichtsbegleitenden Medium weiterzuentwickeln. Aus Workshops und Feedbacks haben sich noch unzählige Anforderungen und Ideen ergeben, die es umzusetzen gilt. Der Kreis Olpe, den wir schon zu Beginn des Förderzeitraums als Partner gewinnen

konnten, startet Mitte Oktober 2021 eine große Werbekampagne für Lunes, ab 2022 werden die Malteser Werke gGmbH und ggf. die Landeshauptstadt München und die Stadt Augsburg sich ebenfalls als Kooperationspartner anschließen um Lunes vor allem auch inhaltlich weiter zu verbessern.

Während des Projektzeitraums konnten wir 5 neue Ehrenamtliche gewinnen, die wir im Projekt etablieren konnten. Eine der Personen wird ab 2022 auf eine Minijob-Stelle für das Partner- und Schulmanagement bei Lunes wechseln können.



Abbildung 2: Entwürfe für Marketing-Kampagne

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Aufgrund der Corona-Pandemie waren Schulbesuche und Tests im Präsenzunterricht schwierig umzusetzen. Entsprechend hat sich die Pilotphase zeitlich verzögert.

Ebenfalls verworfen wurde die Idee, dass Schulen und Unternehmen eine komplett eigene Lunes-App erhalten können. Zwar ist dies aufgrund der Open-Source-Architektur theoretisch natürlich möglich, mehr Sinn macht es aber, wenn die bestehende Lunes-App mit ihren Stammwortschätzen und Grundkategorien um individuelle Bereiche erweitert werden kann. Dies macht aus Aufwands- aber auch aus Nutzersicht mehr Sinn.

Über einen individuellen 10-stelligen-Code können Nutzerinnen und Nutzer der App Bereiche von Schulen oder Unternehmen mit eigens erstellten Vokabellisten zur Lunes-App hinzufügen.

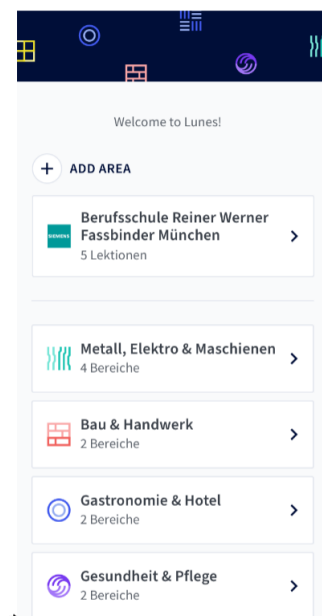


Abbildung 3: Externe Organisationen können eigene Lunes-Bereiche erhalten

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die primäre Webseite (informiert über aktuelle unterstützte Berufe, Neuigkeiten via Blog und das Projekt allgemein): www.lunes.app

Die App selbst kann im AppStore von Apple und im PlayStore von Google heruntergeladen werden:

Für Google Android: <https://play.google.com/store/apps/details?id=app.lunes>

Für Apple/iPhone: <https://apps.apple.com/de/app/lunes/id1562834995>

Für technisch-interessierte Entwickler:innen und Akteur:innen steht unser GitHub bereit:

Vokabelverwaltung (Backend): <https://github.com/digitalfabrik/lunes-cms>

Lunes-App (Frontend): <https://github.com/digitalfabrik/lunes-app>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es wurden mehr Stunden in das Projekt investiert als zunächst geplant. Da dies aber über ehrenamtliche Personalressourcen erfolgt ist, konnte die benatragte Kostenplanung eingehalten werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Corona war ein Einflussfaktor der unsere Zeitplanung in der Zusammenarbeit mit Schulen nach hinten geworfen hat.

Positiv hervorzuheben ist, dass wir deutlich einfacher Kooperationspartner finden konnten als wir es in der Kürze des Förderzeitraums für möglich gehalten haben. Die Überführung in die gemeinnützige Tür an Tür – Digitalfabrik gGmbH zum Ende des Förderzeitraums um eine Nachnutzung und vor allem professionelle Weiterentwicklung gewährleisten zu können, hat trotz Urlaubszeitraum Ende August/Anfang September ebenfalls gut geklappt.

Richtlinie zum „Software-Sprint“

MatrixConduit – Erweiterung und Stabilisierung des Conduit Matrix Servers mit einem Fokus auf Föderation

Schlussbericht

Zuwendungsempfänger:

Timo Kösters

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S11 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Heutzutage benutzen viele Menschen Chat-Programme wie WhatsApp, Discord oder Microsoft Teams: Geschlossene Plattformen, von denen man abhängig ist und dessen Sicherheit man nicht überprüfen kann. MatrixConduit hilft dem Matrix Projekt, welches die Kontrolle über Chat-Nachrichten zurück in die Hände der Nutzer bringen will. So kann man sich zum Beispiel sicher sein, dass kein Dritter unbefugten Zugriff auf die Daten hat und durch Ende-zu-Ende Verschlüsselung ist zudem sichergestellt, dass selbst Besitzer von Servern nicht die Nachrichten von allen Nutzern auf diesem Server lesen können.

Geplante Meilensteine:

1. Vollständige Client-Server API Unterstützung: Alle möglichen Funktionen, die das Programm eines Endnutzers nutzen könnte sind möglichst vollständig implementiert.
2. Föderation mit anderen Servern: Damit wird ermöglicht, dass ein MatrixConduit Server mit jedem anderen Matrix Server zuverlässig kommunizieren und Daten austauschen kann.
3. App-Service Kompatibilität: App Services sind Programme, die über einen besonderen Weg mit einem Matrix Server sprechen können um Features wie z.B. Brücken in Netzwerke außerhalb von Matrix zu ermöglichen.
4. Optimierung: MatrixConduit benötigt wenige Ressourcen und Daten werden effizient in der Datenbank abgelegt.
5. Administrations-Hilfen: Implementierung von Features, die das Administrieren, Moderieren und Steuern des Servers vereinfachen.
6. Migrations-Hilfen: Der Wechsel von dem offiziellen Matrix Server (Synapse) zu MatrixConduit soll leichter werden

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

MatrixConduit ist nun in der Beta-Phase und ermöglicht Einzelpersonen ohne große Kosten selber seine Chat-Daten privat zu speichern. Dabei steht vor allem der gesellschaftliche Nutzen im Vordergrund.

Nach Veröffentlichung der MatrixConduit Beta wurde die Webseite <https://conduit.rs/release-0-2-0/> von über 12.000 unterschiedlichen Personen aufgerufen. Ein paar davon haben noch am selben Tag MatrixConduit heruntergeladen und auf ihrem eigenen Server ausprobiert.

Man kann zum Beispiel an dem Romeo and Juliet benchmark <https://github.com/timokoesters/romeo-and-juliet-benchmark> erkennen, dass Conduit Vorteile über Synapse und Dendrite bietet.

Ausführliche Darstellung der Ergebnisse

Meilenstein 1: Vollständige Client-Server API Unterstützung

- Lesebestätigungen über /receipt
- Push Nachrichten. Clients können Nachrichten empfangen ohne dass sie aktiv danach suchen
- Korrekte Generation von Vorschaubildern (Thumbnails) für Bilder
- Deutlich schnelleres Erkunden von Räumen
- Ende-zu-Ende Verschlüsselung

Meilenstein 2: Föderation mit anderen Servern

- State Resolution. Dies war das wichtigste Feature, da es die Sicherheit von Matrix ermöglicht
- Lücken in dem Nachrichtenverlauf werden erkannt und automatisch geschlossen
- Einladungen in Räume über Föderation
- Lesebestätigungen über Föderation
- Fehlgeschlagene Verbindungsversuche werden wiederholt

Meilenstein 3: App-Service Kompatibilität

Dank diesem Meilenstein ist es nun möglich einen MatrixConduit Server als Brücke in viele andere Netzwerke zu verwenden. Getestet wurden zum Beispiel Discord, Telegram und Hangouts.

Meilenstein 4: Optimierung

- Support für Jaeger um Performance Probleme zu analysieren
- Deutlich bessere /sync Performance, indem effizienter berechnet und gespeichert wird, was der Client wissen muss
- Neu entwickelter Algorithmus der den Zustand eines Raumes in einer sehr effizienten Form mit Hilfe von Differenzen zu anderen Zuständen speichert
- Sehr viele Caches für alle möglichen Operationen

Meilenstein 5: Administrierungs-Hilfen

Es wurden mehrere Dokumente geschrieben, die das Aufsetzen und Upgraden der Conduit Installation, sowie das Verbinden mit App-Services erklären, jedoch konnten Moderations-Hilfen aus zeitlichen Gründen noch nicht programmiert werden.

Meilenstein 6: Migrations-Hilfen

Dieses Ziel konnte leider nicht umgesetzt werden da es einen grösseren Umfang als erwartet hat und nicht mehr viel Zeit blieb. Jedoch war es als letzter Meilenstein auch nur als Bonus gedacht und ist für die Nutzung von Conduit nicht notwendig sondern lediglich eine Erleichterung für Personen die schon einen anderen Server haben, aber auf Conduit umsteigen möchten.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

MatrixConduit kann auf günstigen Servern installiert werden und erlaubt es Einzelpersonen, unabhängig von großen Unternehmen, Nachrichten sicher an andere teilen zu können.

Ich konnte über 500 Personen überzeugen den offiziellen #conduit Chat-Raum auf Matrix beizutreten, wo Fragen gestellt werden und auch über die Entwicklung von Conduit diskutiert wird. Mir haben über 40 Personen bei der Entwicklung von Conduit geholfen indem sie Code oder Dokumentation beigetragen haben. Die Pläne für die Zukunft sind die fehlenden Meilensteine und weitere Verbesserungen die MatrixConduit zu einem besseren Projekt machen. Außerdem ist es möglich, dass Conduit für Experimente genutzt wird, die dem gesamte Matrix Projekt helfen. Ich konnte viel über die Arbeit an Open-Source Projekten lernen.

Die nächsten Ziele sind folgende:

- Datenbank Optimierungen, sodass weniger IO notwendig ist und weniger RAM gebraucht wird.
- Nutzer Verifikation der Schlüssel bei der Ende-zu-Ende Verschlüsselung ist momentan nur lokal möglich, für Föderation müssen jedoch noch Endpunkte implementiert werden.
- Das Laden von Chat Nachrichten von bevor man einen Raum betreten hat.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Aktuelle Informationen zum Projekt sind auf <https://conduit.rs>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Optimierung der RAM- und Festplattennutzung hat deutlich mehr Zeit in Anspruch genommen als im vorhinein absehbar war, deshalb konnten die letzteren Meilensteine nicht erreicht werden. Ein Grund dafür war, dass wir von sled auf sqlite umsteigen mussten, da sled zu viel RAM benötigte und nicht wieder freigegeben hat.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Meine Arbeit war auch mit vielen kleinen Verbesserungen an anderen Open-Source Projekten verbunden. Zum Beispiel bei diesen Projekten:

- <https://github.com/matrix-org/matrix-doc>: Matrix Spezifikation
- <https://github.com/matrix-org/matrix-ios-sdk>: Matrix iOS SDK
- <https://github.com/matrix-org/matrix-js-sdk>: Matrix Javascript SDK
- <https://github.com/matrix-org/matrix-react-sdk>: SDK für Browser Matrix Client
- <https://github.com/vector-im/element-android>: Matrix Client für Android
- <https://github.com/ruma/ruma>: Matrix Bibliothek für Rust

Richtlinie zum „Software-Sprint“

Cobox – Bausteine für eine kooperative Cloud

Schlussbericht

Zuwendungsempfänger:

Pegasus GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S12 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Wir entwickeln ein Datenspeicher- und Backupsystem für die Bedürfnisse kleiner Organisationen.

Unsere Bedarfsermittlung hat gezeigt, dass viele Gruppen mit den Funktionen existenter Clouddienste zufrieden sind. Jedoch herrschen häufig Bedenken hinsichtlich des Datenschutzes und der Einhaltung der DSGVO, bezüglich extraktivistischer Geschäftsmodelle und generell der Abhängigkeit von externen Anbieter*innen.

Unsere Projekt wird genutzt wie gängige Clouddienste, bietet jedoch eine Lösung für wachsende Sicherheitsbedenken, da kein zentraler Server verwendet wird.

Vor der beginn dieses Projekt, Cobox existiert schon als Alphaversion, mit begrenzten Funktionalität. Wir haben geplant, einen Match-Making-Dienst zu erarbeiten, um Backups zwischen verschiedenen Gruppen zu vereinbaren.

Die Aufgabe wäre, eines sicheren HTTP-API für den Match-Making-Dienst zu entwickeln, sowie Back-end und UI Komponenten der Cobox-Client-Software für Anfragen an die API.

Wir haben geplant eine Bedarfsanalyse zum Design der Registrierungsprofile, der Backupvereinbarungen sowie zum Prozess des Match-Making durchgeführt, damit dieses System an die Bedürfnisse unserer Benutzer anpasst.

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

Wir sind in der Kooperativenbewegung verwurzelt und haben bereits eine Reihe von Partner*innenorganisationen, die CoBox verwenden wollen. Diese sind sehr unterschiedlich in ihrer Größe, technischen Ausrichtung, ihren Adressat*innen und Partner*innen. Einige Organisationen möchten als Hosts oder „Superknoten“ dienen, andere benötigen ein Backup-System, das „einfach funktioniert“.

Was die Ziele des Software Sprint betrifft, so will CoBox eine alternative Software-Infrastruktur für die Speicherung und Sicherung von Dateien und Daten für Gruppen und Einzelpersonen anbieten, die die Bequemlichkeit der Unternehmens-Cloud zu schätzen wissen, aber das datenextraktivistische Geschäftsmodell, auf dem sie beruht, ablehnen. Stattdessen ermöglicht es CoBox den Nutzer*innen, sich gegenseitig zu helfen, indem sie die Daten der anderen mit ihrem eigenen freien Festplattenspeicherplatz sichern, und zwar auf eine Art und Weise, die keine zentralen Server benötigt und die Privatsphäre, die Sicherheit und das Eigentum an den Daten selbst gewährleistet.

CoBox steht daher in engem Zusammenhang mit den Themen "Security und Software-Infrastruktur" und "Datensicherheit" von Software Sprint.

Ausführliche Darstellung der Ergebnisse

Wir haben in diesem Förderzeitraum viel erreicht. Die Meilensteine für die ersten zwei Monate wurden genau so erreicht, wie wir es geplant hatten. Jedoch stellten die Ergebnisse der Nutzer*innenforschung zu Beginn des Projektes unerwartete Stolpersteine dar. Unsere Nutzer*innenforschung ergab, dass Benutzer*innen es viel wichtiger fanden, die Integrität ihrer Backups überprüfen zu können, als bei der Suche nach Partner*innen unterstützt zu werden, die Backups bereitstellen könnten.

Daraufhin haben wir unseren Arbeitsplan überprüft und ihn an diese Ergebnis angepasst. Die nachfolgenden Meilensteine wurden dann weitgehend innerhalb des vorgesehenen Zeitrahmens erreicht.

Mit der verbesserten Funktion können die Nutzer*innen nun auch erkennen, wann ihre Kolleg*innen online sind. Außerdem können sie sehen, wann jeder Peer seine Daten zuletzt gesichert hat und wann ihre eigenen Datenkopien zuletzt synchronisiert wurden. Dies geschieht über eine intuitive und leicht verständliche Benutzer*innenoberfläche.

Wir planen nun, die Frontend-Komponente fertig zu stellen und eine offizielle Freigabe zu erteilen. Eine Veröffentlichung war nicht Teil unserer Roadmap für den Förderzeitraum, aber wir glauben, dass unser Projekt als Ergebnis dieser Förderung reif für eine neue Version ist, und wir möchten dies zu den Ergebnissen dieser Arbeit hinzufügen.

Die wichtigsten Erkenntnisse, die wir gewonnen haben, beziehen sich auf die Erfahrung und die Auswirkungen des nutzer*innenzentrierten Designs. Wir haben gelernt, dass es die Bereitschaft der Entwickler*innen erfordert, ihre Annahmen in Frage zu stellen, ihre Prioritäten zu überdenken und sich an das anzupassen, was die Nutzer*innen mitteilen, damit die Nutzer*innengemeinschaft den Entwicklungsplan wirklich anführen kann.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Zielgruppe von CoBox sind kleine Organisationen mit kooperativen Werten. Da CoBox Open Source ist, kann diese Zielgruppe CoBox bereits in ihren Gruppen nutzen, zusammen mit den neuen Funktionen. Es gibt noch einiges zu tun, bevor CoBox "produktionsreif" wird, da es sich um ein für seine Ambitionen noch sehr junges Projekt handelt. Die Finanzierung des Prototyps hat das Projekt jedoch erheblich vorangebracht.

Da wir den Code als Open Source zur Verfügung stellen, erwarten wir mehr Beiträge von Entwickler*innen aus der Community, sobald wir eine zweite Version veröffentlichen (was in Kürze der Fall sein wird). So haben wir bereits nach der Veröffentlichung unserer Alpha-Version im letzten Jahr wir mehrere Beiträge von Community-Mitgliedern erhalten. Was die weitere Entwicklung betrifft, so werden wir uns um weitere Finanzmittel bemühen, um die Arbeit fortzusetzen.

Was die persönliche und berufliche Entwicklung betrifft, so hat diese Finanzierungsrunde unseren Teammitgliedern sehr geholfen, ihr Wissen, ihre Fähigkeiten und ihr Verständnis von Softwareentwicklung und Mentoring zu verbessern. Wir schätzen diesen Aspekt der Arbeit sehr.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Unser ursprüngliches Ziel, eine HTTP-API zum Matching von Benutzer*innengruppen zu erstellen, wurde nicht weiterverfolgt. Der Grund dafür ist, dass wir uns zu einem nutzer*innenzentrierten Designprozess verpflichtet hatten. In den ersten Workshops, die wir durchführten, teilten die Benutzer*innen mit, dass es für sie wichtiger sei, den Status ihrer Backups zu überwachen, bevor sie anderen Gruppen die Sicherung ihrer Daten anvertrauen würden. Daher haben wir unsere Roadmap so angepasst, dass wir diese Funktion zuerst einrichten.

Das Ergebnis ist eine "Peer-Präsenz"-Funktion, die es den Benutzer*innen ermöglicht, zu überwachen, wann ihre Peers online sind, wann jeder Peer seine Daten zuletzt gesichert hat und wann ihre eigenen Daten zuletzt synchronisiert wurden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

Während dieses Förderzeitraums haben wir zwei Berichte erstellt, die beide auf unsere Website hochgeladen wurden:

Bericht über die Peer-Discovery-Workshops: <https://cobox.cloud/web/peer-discovery-report.pdf>

Aktualisierter Bericht über die Benutzer*innengeschichten: <https://cobox.cloud/web/health-check-user-stories.pdf>

Wir haben einen Blogpost verfasst, in dem wir unsere Erfahrungen schildern und insbesondere beschreiben, wie wir uns an die Ergebnisse des nutzer*innenzentrierten Designs angepasst haben:

<https://demoweeek.prototypefund.de/projects/02-cobox.html>

Unser gesamter Code ist auf Gitlab verfügbar: <https://gitlab.com/coboxcoop>

Wir planen eine zweite Alpha-Version für Ende 2021, die diese Entwicklungen enthalten wird. Eine Ankündigung wird auf unserer Website, über soziale Medien und per E-Mail erfolgen.

Unsere Website: <https://cobox.cloud>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wir haben nutzer*innenzentrierte Design-Workshops mit frühen Anwender*innen durchgeführt, um ein tieferes Verständnis dafür zu erlangen, wie Organisationen zueinander finden, was das Vertrauen zwischen ihnen stärkt und welche Art von Informationen oder Unterstützung wünschenswert ist, damit Organisationen in der Lage und bereit sind, die Daten der anderen zu sichern. Das Hauptziel der Workshops bestand darin, den Bedarf und die Anforderungen an einen "Peer-Discovery"-Mechanismus in CoBox zu untersuchen, den wir ursprünglich aufbauen wollten.

Aus den Workshops ging hervor, dass die Organisationen, die zu den frühen Nutzer*innen von CoBox gehören, bereits stark vernetzt sind und es vorziehen, Vertrauen außerhalb von technischen Netzwerken aufzubauen. Um dieses "menschliche Vertrauen" in "technisches Vertrauen" umzuwandeln, mussten die Nutzer*innen jedoch zunächst in der Lage sein, zu überprüfen und zu visualisieren, dass ihre Backups regelmäßig aktualisiert werden und im gesamten Netzwerk verfügbar sind, um den technischen Aufwand und die damit verbundenen Ängste für nichttechnische Gruppen zu verringern.

Dies war sowohl für die Nutzer*innen wichtig, die sich bereit erklärten, die Daten anderer Nutzer*innen zu sichern, als auch für diejenigen, die sich damit einverstanden erklärten, dass ihre Daten von anderen Nutzer*innen (und nicht

von einem großen Unternehmen) gesichert wurden. Die Nützlichkeit eines Peer-Discovery-Mechanismus hing daher von der Lösung dieses Vertrauens- und Zugangsproblems ab.

Daher haben wir unseren Arbeitsplan angepasst und statt der ursprünglich geplanten Komponente ein System zur Überprüfung der Integrität der Backup entwickelt, die von verschiedenen Partner*innen bereitgestellt werden. Damit haben wir die wesentliche Grundlage dafür geschaffen, dass Gruppen später Verbindungen über das CoBox-Netzwerk herstellen.

Die COVID19-Pandemie hat bedeutet, dass unsere Workshop-Pläne angepasst werden mussten. Wir hatten gehofft, dass es eine Pause in der Abriegelung geben würde, so dass wir die Workshops in Präsenz durchführen könnten, aber letztendlich mussten sie hauptsächlich online stattfinden. Dazu haben wir Videokonferenzen genutzt und mehr Workshops mit kleineren Gruppen als ursprünglich geplant durchgeführt. Allerdings waren wir so auch zeitlich flexibler bei der Durchführung der Workshops, da wir keinen Ort organisieren mussten und nicht alle Teilnehmer*innen zur gleichen Zeit am gleichen Ort sein mussten.

Trotz dieser Änderungen und Schwierigkeiten haben wir alle Meilensteine in unserem aktualisierten Arbeitsplan erreicht und die Funktion zur Überprüfung der Backup-Integrität für Cobox erfolgreich implementiert.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Unsere Co-Design-Workshops wurden mit Unterstützung von Simply Secure durchgeführt, die einen Bericht über die Einbeziehung von Benutzer*innenerfahrungen bei der Gestaltung von Benutzer*innenoberfläche für dezentralisierte Apps veröffentlicht haben. In ihrem Bericht und bei unserem Treffen mit ihnen teilten sie einige "Design-Muster" und Prinzipien mit uns, die wir für die Gestaltung unserer Benutzer*innenoberfläche nutzten.

Während des Förderzeitraums hatten wir auch zwei Sicherheitsüberprüfungs-Workshops von Least Authority, bei denen wir die Implementierung des von CoBox verwendeten Krypto-Moduls diskutierten. Im Anschluss an diese Treffen haben wir eine Reihe von Aufgaben in unsere zukünftige Roadmap aufgenommen.

Die Anregungen, die wir von Simply Secure und Least Authority erhalten haben, werden in Zukunft unsere Roadmaps für Finanzierungsvorschläge und das Schnittstellendesign beeinflussen.

Richtlinie zum „Software-Sprint“

HubGrep – Meta-Suchmaschine für OpenSource Code Repositories

Schlussbericht

Zuwendungsempfänger:

Hartmann, Eliasson GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS17S13 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

GitHub, betrieben von einer ClosedSource Software, in den Händen Microsofts, ist für viele die einzige Anlaufstelle für OpenSource Code Hosting. Teilweise wird GitHub (eine Hosting Plattform) gar als Synonym zu git (einem Versionierungstool) benutzt. Dadurch sind viele Projekte, die nicht auf GitHub liegen, sondern beispielsweise auf selbst gehosteten GitLab oder Gitea Servern, praktisch unauffindbar.

Mit HubGrep erstellen wir einen Suchindex über viele Hosting Services, und bieten über eine Suchmaschine allen Repositories die Chance geben, gefunden zu werden, und die

Ergebnisse gleichwertig darstellen - egal von welcher Plattform sie kommen.

Die Vorgehensweise war dabei wie folgt:

1. Auf Grundlage eines Kommandozeilen Prototyps wurde eine Meta-Suchmaschine als Webservice gebaut. Interessierte können ihre eigenen Hosting Instanzen hinzufügen (per Web Interface oder Pull Request), sofern der genutzte Service unterstützt wird.

2. Es wurde Dokumentiert, wie Interessierte eine eigene Instanz aufsetzen können, deployment Tools geschrieben.

3. Indexer und Crawler: Um die Limitierungen der Meta-Suche zu umgehen, haben wir Crawler geschrieben, die die Eckdaten der Repositories an einen Indexer schicken, um einen lokalen Index für schnellere Suche anzulegen und eventuelles Rate Limiting der Dienste zu umgehen.

5. Die Meta-Suche (Meilenstein 1) wurde durch eine Suche auf Basis des Suchindexes ersetzt.

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

Durch eine Liste freier Hosting Services, die User selbst erweitern können, und dem entsprechenden Index aller Inhalte wurden Datensätze erzeugt, die es allen ermöglichen, vorher unzugängliche Projekte zu finden, oder die eigenen Projekte auffindbarer zu machen. Ausserdem können die veröffentlichten Rohdaten Grundlage für weitere Projekte sein.

Ausführliche Darstellung der Ergebnisse

Alle Meilensteine wurden erreicht.

Als Ergebnis des ersten Meilensteins entstand eine Metasuchmaschine (https://github.com/HubGrep/hubgrep_search/tree/legacy/metasearch), die benutzbar ist, und vorerst unter <https://meta.hubgrep.io/> weiter betrieben wird.

Angefangen mit dem zweiten Meilenstein wurde das System dokumentiert. Die entstandene Dokumentation wird weiter gepflegt und ist unter <https://docs.hubgrep.io/> erreichbar. Hier wird die Suchmaschine selbst, als auch das Ökosystem darum (Crawler und Indexer, Bedienung durch den Endnutzer) beschrieben.

Zum dritten Meilenstein wurden Crawler für drei Hosting Services geschrieben: Github, Gitea und Gitlab.

Ausserdem wurde ein "Indexer" als Backend geschrieben, in dem alle Daten gesammelt werden.

Als Ergebnis können hier die letzten Snapshots der Daten der Hoster abgerufen werden: <https://indexer.hubgrep.io/>

Der vierte Meilenstein befasste sich mit der Integration der gesammelten Daten in die Suchmaschine in Meilenstein 1. Der Plan war hier, die Suchergebnisse aus der Metasuche mit den Suchergebnissen aus dem eigenen Suchindex zu kombinieren. Da wir aber keine Möglichkeit haben, einen einheitlichen "Score" für externe Suchergebnisse und Suchergebnisse aus unserem eigenen Index zu bilden, wurde dies zugunsten der zuverlässigeren, schnelleren Suche im eigenen Index verworfen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Unsere Zielgruppe hat nun Zugriff auf eine Suchmaschine, die über viele Hoster hinweg OpenSource Repositories finden kann. Nutzer können ihre eigenen, oder ihnen bekannte Hosting Services hinzufügen, um ihre eigenen Projekte auffindbar zu machen.

Außerdem gibt es die Möglichkeit, auf Basis des gesammelten Datensatzes andere Projekte aufzubauen.

Durch die Open-Source-Stellung der Software können Nutzer selbst Crawler für neue Hosting Service Typen schreiben, Bugs fixen, neue Funktionen hinzufügen, oder den kompletten Stack selbst betreiben, um beispielsweise möglicher Zensur vorzubeugen.

Zur Weiterentwicklung gibt es eine Reihe von Möglichkeiten:

Da wir gerade erst angefangen haben, unser eigenes Such-Backend zu nutzen, liegt unsere Priorität auf dem Verbessern der Gewichtung und dem Sortieren der Ergebnisse. Eine andere naheliegende Erweiterung wäre, mehr Hosting-Services zu unterstützen und zu durchsuchen - beispielsweise Bitbucket.

Einige APIs stellen mehr Informationen bereit als andere, deshalb kennen wir beispielsweise die Lizenz oder Programmiersprache nicht von allen Projekten. Alle Projekte zu klonen, zu analysieren, und um die fehlenden Metadaten zu erweitern wäre eine ressourcen- und zeitintensive Aufgabe, aber wir haben eine Datenbasis, auf der wir aufbauen könnten. Mit den geklonten Projekten hätten wir dann, über die API Metadaten hinaus, auch alle Commit IDs und könnten über alle Hoster hinweg Forks und Mirrors von Projekten erkennen, und in den Ergebnissen gruppieren.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Registrierungspflicht

Die erste veröffentlichte Version von HubGrep hatte eine Registrierungspflicht: um Hoster hinzuzufügen und später zu bearbeiten, musste ein Benutzerkonto erstellt werden. Der Grund dafür war, dass wir geplant haben, dass die Administratoren ihre Hoster selbst hinzufügen sollten.

In der Praxis war eine der ersten Beschwerden, dass dieses Vorgehen "nicht besonders Datensparsam" sei. Dazu kommt, dass für die Crawler, ausgenommen die Github Crawler, keine API Keys oder ähnliches benötigt werden.

Diese Funktion wurde also wieder entfernt, und jeder kann beliebige Services hinzufügen, solange sie von HubGrep unterstützt werden.

Suchproxy

Die erste Version von HubGrep war ein Proxyservice zu den Suchfunktionen der einzelnen Hoster. Da wir zwei Meilensteine später einen eigenen Suchindex aufgebaut hatten, und die Integration der Proxysuche mit der Suche im eigenen Index komplexer war als gedacht, und der eigene Index, selbst mit dem großen Github Datensatz, gut funktioniert, haben wir beschlossen ausschließlich den eigenen Suchindex zu nutzen.

Github Crawler

Aufgrund des Aufbaus der Github APIs war es nötig, verschiedene Varianten des Crawlers zu bauen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Eine Instanz der Suchmaschine kann unter <https://hubgrep.io> aufgerufen und genutzt werden. Die Dokumentation zur Nutzung und zum Betrieb einer Instanz sind unter <https://docs.hubgrep.io> zu finden.

Der development-Blog befindet sich unter <https://blog.hubgrep.io>.

Der Code für die einzelnen Projekte und Dokumentation ist in der Github Organisation unter <https://github.com/hubgrep> zu finden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Jan konnte nicht so viele Stunden wie geplant für das Projekt leisten. Stattdessen hat Henrik mehr Stunden geleistet als ursprünglich geplant, um dies zu kompensieren. Dadurch konnte der Zeitplan eingehalten werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab keine Entwicklungen Dritter die Einfluss auf das Ergebnis hatten.

Richtlinie zum „Software-Sprint“

Feminizidmap – Datenbanksoftware zum Dokumentieren von Femi(ni)ziden in Deutschland

Schlussbericht

Zuwendungsempfänger:

Lisa Passing

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21514 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Femi(ni)zide werden in Deutschland nicht explizit in der Kriminalstatistik erfasst. Erst seit wenigen Jahren wertet die Statistik Morde in intimen Partnerschaften aus, nur eine von mehreren Unterkategorien von Femi(ni)ziden.

In den Sozialwissenschaften wird seit vielen Jahren zu diesem Phänomen geforscht und in der Zivilgesellschaft nehmen sich zunehmend mehr Projekte diesem Thema an. Ihnen fehlt eine solide Datengrundlage, da offizielle Zahlen fehlen. Die Daten müssen von Hand gesammelt werden, oft geschieht das pro Projekt in einem Spreadsheet.

Hier setzt Feminizidmap an: Eine Datenbank mit automatisierter Auswertung soll Projekte dabei unterstützen Fälle zu dokumentieren, Daten strukturiert zu sammeln und anderen zur Verfügung zu stellen.

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

Die entstehende, quelloffene Software unterstützt sie beim dokumentieren, verfolgen, verwalten und auswerten von Fällen, entsprechende Rohdaten und automatisierte Auswertungen bleiben online als Informationsquelle für alle bestehen.

Ausführliche Darstellung der Ergebnisse

Das Kerntool, an dem ich im Rahmen der Förderung gearbeitet wurde heißt Mapper. Mapper nutzt eine klassische Server-Client-Architektur. Der Server, der sowohl eine interne, authentifizierte API bereitstellt, als auch später die öffentliche API, ist mit dem Framework Ruby on Rails umgesetzt und nutzt eine Postgres Datenbank. Der Frontendclient, der die interne API konsumiert, basiert auf dem Framework VueJS. Für einen einfachen Einsatz auf Produktionsservern stehen Dockerfiles und ein Docker-Compose-Script bereit.

Mapper soll alle wesentlichen Aufgaben übernehmen, die im Austausch mit dem Team erarbeitet wurden. Davon sind folgende vollständig oder teilweise umgesetzt:

- NutzerInnenverwaltung mit verschiedenen Rollen und Rechten
- Interface um Datenstruktur der gesammelten Daten veränder- bzw erweiterbar zu machen
- Konzept und Beginn der Umsetzung des Dateneingabe-Interface, welches sowohl ein Schritt-für-Schritt-Workflow als auch ein punktuell Springen zwischen Datenebenen ermöglicht
- Alle Änderungen am Datenstanz werden automatisch gespeichert und Bearbeitungslogs geführt werden. Hieraus soll später "History" pro Eintrag entstehen, aus der zur Not vergangene Zustände wiederhergestellt werden können.

Weitere Aufgaben sind noch in frühen Konzeptionsphasen, sollen aber in vor der Übernahme durch das Team fertig gestellt werden, da sie weitere aufwändige, händische Arbeit ersetzen. Dazu gehören:

- Automatisiertes erstellen von PDFs der Originalquellen zur Dokumentation
- Such- und Filter-Interface um schnell bestimmte Fälle zu finden und weiter bearbeiten zu können
- Review- bzw Nachbereitungworkflow sowie ein Veröffentlichungsmechanismus von jährlichen Berichten und Statistiken

Neben der Mapper entstand auch die auf dem Ruby Framework Sinatra basierte App Briefkasten, die der Zivilgesellschaft das direkte, und auf Wunsch anonyme, Melden von Femi(ni)ziden an das Projekt ermöglicht.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Für die Zielgruppe es Dokumentationsteams ist ein Prototyp entstanden, der in den Grundzügen alle wichtigen Arbeitsschritte im Team abbildet. Der Prototyp ist noch nicht reif genug den bisherigen, Spreadsheet-basierten Workflow zu ersetzen. Das bisherige Feedback aus dem Team stimmt zuversichtlich, dass viel manuelle Arbeit und häufige Fehlerquellen zukünftig wegfallen werden.

Die Pläne für den Prototyp bis zur benutzbaren App sind bereits ausgearbeitet und warten auf ihre Umsetzung, die nach der Förderphase langsamer laufen wird.

Als aktivistisches Open Source Projekt hoffe ich sowohl Unterstützung in der Tech-Community zu finden, als auch das Projekt selbst zu übersetzen und für ähnliche Dokumentationsgruppen nutzbar zu machen.

Mich persönlich hat die Arbeit am Projekt vor allem in den Bereichen Projektplanung, Konzeptionierung von Datenbank bis User-Experience mehr gefordert als ich es gewohnt bin, weil ich normalerweise bestehende Konzepte umsetze. Hier habe ich definitiv viel dazu gelernt.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die erste Version des Datenbankschemas wurde fast eins zu eins von der Projekt bisher genutzten Tabellenstruktur übernommen. Nachdem ich darauf aufbauend bereits die ersten Endpunkte der Schnittstelle programmierte, stellte sich in mehreren Meetings mit den Freiwilligen aus dem Projekt heraus, dass das Schema an einigen Stellen nicht nur vereinfacht werden kann, sondern an anderen Stellen viel flexibler gedacht werden muss.

Daraufhin plante ich das Datenbankschema komplett neu, mit einem viel höherem Abstraktionslevel, und verwarf alle Teile, die mit dem alten Schema zusammenhingen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Das Projekt ist für EndnutzerInnen noch nicht live.

Der Code kann auf GitHub eingesehen werden

<https://github.com/feminizidmap/feminizid-mapper>. Technische Dokumentation und Entwicklungsblog befinden sich unter <https://tech.feminizidmap.org/>.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Arbeits- und Meilensteinplan waren von vorn herein sehr optimistisch angelegt. Bei der Planung und Umsetzung der Umstrukturierung des Datenbankschemas verlor ich viel Zeit, die ich nicht mehr aufholen konnte und deswegen früh alle „Nice-to-Haves“ strich.

Dazu kamen, dass ich die Ansprüche an die User-Experience des Dateneingabeinterfaces unterschätzte. Nach mehreren Gesprächen und „Shadowing“ von Freiwilligen bei ihrer Arbeit wurde klar, dass das Interface sowohl Flexibilität für verschiedene Workflows bieten muss, aber gleichzeitig potentielle Fehler bei der Eingabe verhindern und, wo es Sinn macht, Schritte

automatisieren soll. Die Konzeptionierung nahm viel nicht-eingeplante Zeit in Anspruch.

Insgesamt konnte ich große Teile des Arbeitsplans nicht oder nicht vollständig umsetzen, aber realistischer planen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Keine.

29. Sep. 2021

Eingangsnr.: *5733/6*

Richtlinie zum „Software-Sprint“

Mat-o-Wahl – ein einfacher Wahl-o-Mat für große und kleine Wahlen

Schlussbericht

Zuwendungsempfänger:

Mathias Steudtner

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S15 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Der Mat-O-Wahl ist ein Programm zum Vergleichen von Parteipositionen mit der eigenen Meinung. Er orientiert sich am Wahl-O-Mat der "Bundeszentrale für politische Bildung" (bpb). Dieser ist urheberrechtlich geschützt und technisch komplex, was den Einsatz im "kleinen Rahmen" erschwert.

Politisch interessierte Bürger können den Mat-o-Wahl mit einfachen Textdateien für Fragen, Parteien und Einstellungen administrieren sowie DSGVO-konform im Web bereitstellen. Das erleichtert den Einstieg, macht Politik transparenter und fördert somit die Bürgerbeteiligung. Der Mat-o-Wahl arbeitet mit Standard-Webtechniken (HTML, CSS und JavaScript) und passt sich automatisch an die Bildschirmgröße der Nutzer an („responsive design“).

Die Aufgaben im Prototype Fund (PTF) gliederten sich in vier Arbeitspakete, welche erfolgreich abgeschlossen wurden.

- 1) Modul: Antworten der Parteien anzeigen – nach Partei sortiert, nicht nach Thesen
- 2) Modul: „Mehrere gleichzeitige Wahlen“ (Textfilter) erweitern
- 3) Modul: Vorabfrage „Lieblingspartei“
- 4) Modul: optische Verbesserungen ermöglichen und implementieren

Eine detaillierte Übersicht findet sich im Kapitel „Ausführliche Darstellung der Ergebnisse“.

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

Auf „kleiner“ Ebene - in den Städten, Landkreisen oder Universitäten - wissen die Wähler oft nicht, wofür ihre Kandidaten stehen. Dies könnten z.B. Themen wie eine neue Umgehungsstraße oder den Umweltschutz betreffen, den sozialen Wohnungsbau, eine neue KiTa oder die Aufnahme neuer Schulden. Ähnlich verhält es sich bei Landtags- und Bundestagswahlen wenn man den Fokus auf ein spezielles Thema richten möchte. (z.B. Wo stehen die Parteien speziell im Bereich Verkehrswende, Inklusion oder Lärmschutz.)

Demgegenüber stehen politikinteressierte Menschen bzw. Vereine, welche diese Informationen transparent darstellen wollen. Aber ihnen fehlt ein Werkzeug – wie ein Wahl-o-Mat - um viele Wahlberechtigte schnell, einfach und spielerisch zu informieren.

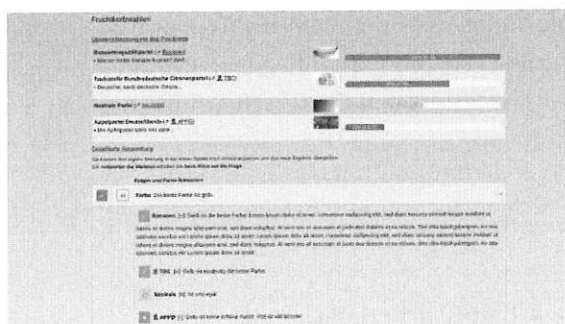
Der Mat-o-Wahl ist dieses Werkzeug und schließt damit die Lücke zwischen den Gruppen der Kandidaten und Wähler. Als typisches „Civic Tech“-Tool verbessert der Mat-o-Wahl den Zugang zu Informationen und regt zu einem intensiveren Austausch mit (und Interesse an) der Politik an.

Ausführliche Darstellung der Ergebnisse

Der Prototyp wurde erfolgreich entwickelt und alle Arbeitspakete (AP) wurden abgeschlossen.

AP 1) Modul: **Antworten der Parteien anzeigen** – nach Partei sortiert, nicht nach Thesen

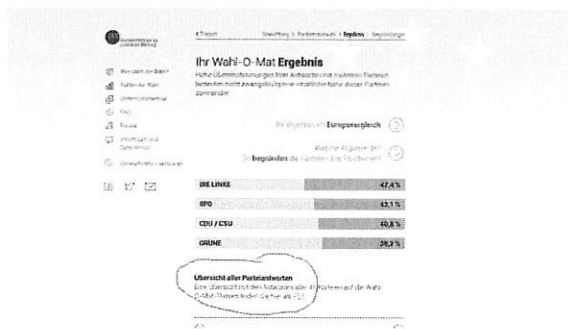
In den älteren Versionen des Mat-o-Wahl wurden am Ende alle Fragen wieder angezeigt. Darunter konnte man die jeweilige Antwort der Partei einblenden. Es fehlte aber eine Möglichkeit eine einzelne Partei auszuwählen und nur ihre dazugehörigen Antworten anzuzeigen. Der originale Wahl-o-Mat bietet solch eine Gesamtansicht aller Partei-Antworten (als PDF) an.



Mat-o-Wahl (alt)
Fragen und Antworten werden am Ende standardmäßig eingeblendet



Wahl-o-Mat (bpb)
Fragen und Antworten können am Ende eingeblendet werden.



Wahl-o-Mat (bpb)
Link zur „Übersicht der Parteipositionen“



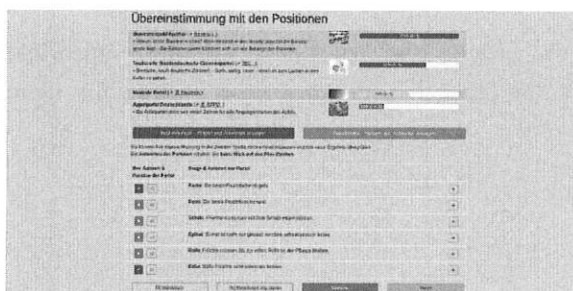
Wahl-o-Mat (bpb)
Übersicht der Parteipositionen

Ergebnis: Die Funktion wurde hinzugefügt und durch die Testpartner evaluiert.

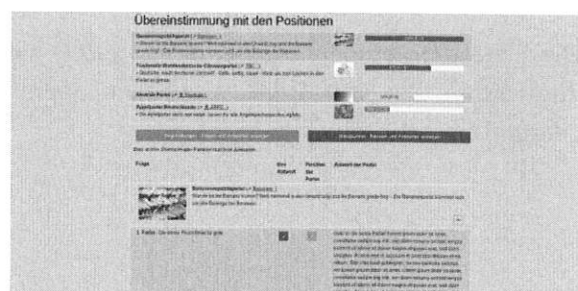
Technischer Hintergrund: Die Informationen lagen intern bereits vor aber sie mussten noch in das richtige Format gebracht werden.

Die Schwierigkeit lag zum einen darin, eine Bedien-Logik innerhalb der Übersicht zu finden, die auch mit der bisherigen Ansicht gut funktioniert. z.B. Soll das Icon vor den Text kommen oder danach? Wie sieht es auf dem Smartphone aus? Welche Informationen kommen in die Titelzeile? Wie kann die richtige Reihenfolge der Parteien sichergestellt werden?

Zum anderen existierten nun zwei Listen unter den Ergebnissen, wo es vorher nur eine (sofort sichtbare) Übersicht mit den Fragen gab. Dies brachte neue Herausforderungen mit sich, wie diese Übersichten nutzerfreundlich dargestellt werden können. Am Ende wurden die Listen hinter zwei Buttons „versteckt“ und werden bei Bedarf eingeblendet.



Mat-o-Wahl (neu)
Fragen und Antworten werden am Ende auf Knopfdruck angezeigt



Mat-o-Wahl (neu)
Antworten der Parteien werden am Ende auf Knopfdruck angezeigt

AP 2) Modul: „Mehrere gleichzeitige Wahlen“ (Textfilter) erweitern

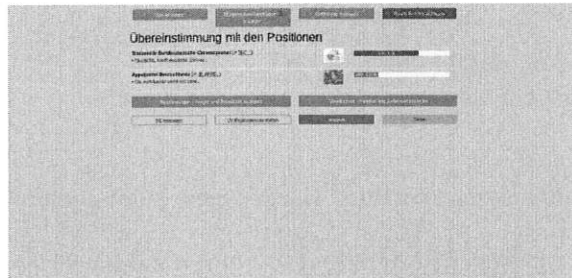
Es gibt Einsatzfälle, in denen man den Mat-o-Wahl mit den gleichen Fragen für verschiedene „Wahlen“ ausstatten möchte.

Das könnte z. B. die gleichzeitige Wahl zum Stadtrat und Bürgermeister sein, eine Wahl in mehreren Stadtteilen (Nord, Ost, Süd, West) oder andere Kriterien. In diesem Fall erhalten alle Kandidierenden die gleichen Fragen, aber die Nutzer können in der Ergebnisübersicht filtern, wen sie sehen möchten.



Mat-o-Wahl (alt)

Es konnten nur zwei Filter-Optionen definiert werden.



Mat-o-Wahl (neu)

Es können beliebig viele Filter-Optionen definiert werden (Bsp.: vier)

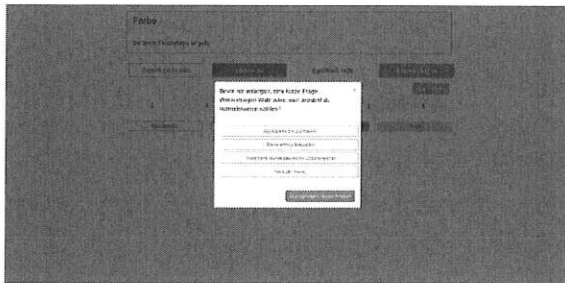
Ergebnis: Die Funktion wurde hinzugefügt und durch die Testpartner evaluiert.

Technischer Hintergrund: Im PTF wurde ein Erweiterungsmodul geschrieben, mit der man die Ergebnisse nach bestimmten Kriterien filtern kann. Die Kriterien werden vorher in der Mat-o-Wahl-eigenen Parteien-Datei sowie im eigentlichen Erweiterungsmodul definiert. Zum Schluss wird die Erweiterung in der globalen Einstellungs-Datei aktiviert und filtert nun die gewünschten Ergebnisse.

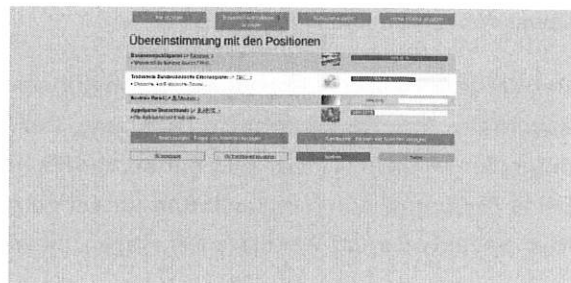
Die Herausforderung bestand dabei, nicht nur die „Hauptliste“ zu filtern, sondern auch die Zusatzlisten. Das betraf die vorhandene Übersicht der „Fragen und Antworten“ sowie die neue „Partei-Übersicht“ aus Arbeitspaket 1.

AP 3) Modul: Vorabfrage „Lieblingspartei“

Laut aktueller Wahl-o-Mat-Begleitforschung¹ nutzen über die Hälfte der Teilnehmer das Werkzeug um ihren eigenen Standpunkt zu überprüfen. In diesem Arbeitspaket sollte deshalb eine Vorabfrage erstellt werden, bei der die Nutzer ihre „Lieblingspartei“ auswählen. Diese Auswahl wird am Ende wieder farblich hervorgehoben. So sieht man, wie nah (oder auch weit) die eigene Präferenz von der Erwartung abweicht.



*Mat-o-Wahl (neu)
Vorabfrage zur Lieblingspartei*



*Mat-o-Wahl (neu)
Farbliche Hervorhebung der Auswahl*

Ergebnis: Die Funktion wurde hinzugefügt und durch die Testpartner evaluiert.

Technischer Hintergrund: Zuerst werden alle Parteien aus der entsprechenden Parteien-Datei gelesen. Diese werden anschließend in einem Popup-Fenster („Modal“) angezeigt. Die Nutzer haben die Möglichkeit die Abfrage zu überspringen und keine Antwort abzugeben.

Die Herausforderung bestand darin, das Abfrage-Fenster erst nach der Willkommensnachricht erscheinen zu lassen. Dies wurde mit einer dynamisch erzeugten Klick-Funktion auf dem Start-Button gelöst. Danach musste die Auswahl in einer Variable abgespeichert werden und schließlich in der Ergebnis-Übersicht farblich hervorgehoben werden. Hier halfen eindeutige ID-Kennungen (Nummern) in der Übersichtsliste um die richtige Partei zu finden.

¹ Marschall, Prof. Dr. Stefan: Ergebnisse der Online-Befragungen zum Wahl-O-Mat, Heinrich-Heine-Universität Düsseldorf, <https://www.sozwiss.hhu.de/institut/abteilungen/politikwissenschaft/politik-ii/prof-dr-stefan-marschall/forschungsprojekte/wahl-o-mat-forschung/online-befragungen> 14.09.2021.

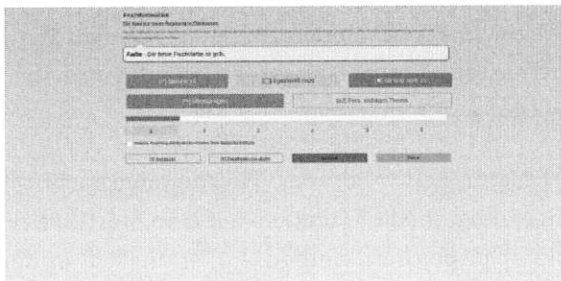
AP 4) Modul: **optische Verbesserungen** ermöglichen und implementieren (UI, UX)

Im Zuge der anderen Punkte wurde auch Wert auf Verbesserungen in der Optik (User Interface / UI) und Nutzerfreundlichkeit (User Experience / UX) gelegt. Dazu wurden mit Hilfe der Open Knowledge Foundation (OKFN) im Prototyp Fund einige Coachings mit UI/UX-Spezialist vermittelt.

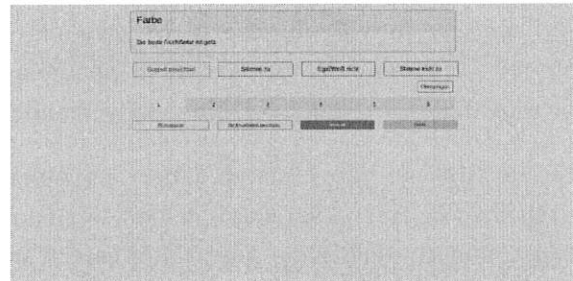
Dabei kam z. B. heraus, dass die bisherigen Auswahl-Buttons zu „unruhig“ waren und schon eine gewisse Vorauswahl provozieren. Dementsprechend wurden die Buttons im Arbeitspaket umsortiert sowie die Farben und Icons reduziert.

Des Weiteren wurden redundante Informationen entfernt. Das betraf den Fortschrittsbalken und die dauerhafte Anzeige der Wahlbeschreibung. Der Fortschritt wird bereits anhand der Nummerierung unter den Fragen sichtbar und die Beschreibung der Wahl erscheint nur noch einmal am Anfang. Beide Änderungen hatten wiederum Auswirkungen auf andere Funktionen, wie z.B. das Starten des Mat-o-Wahl oder das Wechseln der Fragen, die ebenfalls angepasst werden mussten.

Schlussendlich wurde den Nutzern mehr Freiheit bei der Gestaltung gelassen. Alle Formatvorlagen befinden sich nun nicht mehr in einer großen, unübersichtlichen CSS-Datei, sondern in klar definierten Stylesheet-Dateien für bestimmte Zwecke. (z.B. Buttons, Standard, ...) Diese können individuell geladen werden.



Mat-o-Wahl (alt)



Mat-o-Wahl (neu)

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Haupt-Zielgruppen des Mat-o-Wahl unterscheiden sich in „Kunden“ und „Anwender“.²

- „Kunden“ sind interessierte Bürger, Vereine, Bildungseinrichtungen (Schulen, Universitäten / Hochschulen) Forschungseinrichtungen (z.B. Statistik und Wahlforschung) aber auch Medien (z.B. Tageszeitungen, Blogs).
- „Anwender“ sind alle wahlberechtigten Personen der jeweiligen Wahl, welche den Mat-o-Wahl nutzen um sich ein Bild der Kandidaten zu machen.

Die „Kunden“ profitieren allgemein von vereinfachten Einstellungsmöglichkeiten und reduzieren ihren Aufwand im Speziellen bei der Darstellung von komplexen Sachverhalten. Am meisten profitieren jedoch die „Anwender“ von den Neuerungen. Sie erhalten mehr Informationen, welche jedoch übersichtlicher dargestellt werden.

Die Open Source-Stellung der Ergebnisse (und des gesamten Mat-o-Wahl-Systems) erlaubt es Kunden und Anwendern alle Ergebnisse selbst zu überprüfen und schafft Vertrauen. Dies ist besonders bei sensiblen Themen, wie Wahlvorlieben³, ein wichtiger Punkt.

Durch die Arbeit im Projekt konnte ich mich auch selbst fachlich verbessern und neue Kontakte sowie Unterstützer hinzugewinnen. So entstanden auch Ideen für Weiterentwicklungen. Dies umfasst z.B. eine Kategorisierung der Fragen oder feinere Abstufungsmöglichkeiten bei den Auswahlbuttons. (Nebenbemerkung: Letzteres ist auch ein stetiger Kritikpunkt des originalen Wahl-o-Mat, da die Aussagen nur zwischen „Zustimmung“ und „Ablehnung“ schwanken. Eine Partei könnte z.B. „für“ die Windkraft sein aber gleichzeitig hohe Abstandsregeln beim Ausbau fordern, welche den Ausbau behindern. z.B. 1.000 Meter in NRW⁴ oder die 10-fache Höhe in Bayern.⁵)

² Best Management Practice: ITIL Service Strategy, Kapitel 2.1.5, Stakeholders in Service Management, Cabinet Office, 2. Auflage, London, 2011.

³ § 9 Abs. 1 DSGVO (Verarbeitung besonderer Kategorien personenbezogener Daten)

⁴ Löttgen, Bodo (Mdl): „Akzeptanz durch Abstand – Sicherheit für den Ausbau“, 23.12.2020 URL: <https://www.cdu-nrw-fraktion.de/artikel/akzeptanz-durch-abstand-sicherheit-fuer-den-ausbau> [14.09.2021].

⁵ Storch, Lorenz: „Keine Priorität für CSU: Windkraft in Bayern weiter ohne Schwung“, in: Bayerischer Rundfunk AöR, 20.07.2021, URL: <https://www.br.de/nachrichten/bayern/keine-prioritaet-fuer-csu-windkraft-in-bayern-weiter-ohne-schwung,SddHF8o> [14.09.2021].

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Während der Arbeiten an den Arbeitspaketen AP 2 (Mehrere gleichzeitige Wahlen) und AP 4 (optische Verbesserungen) stand die Entscheidung, ob die Ergebnisübersichten als HTML-Tabelle oder HTML-/CSS-Grid dargestellt werden soll.

Eine Tabelle ist vorteilhafter für barrierefreie Seiten, da Hilfstechnologien, wie z.B. Screenreader, diese Syntax sofort interpretieren können. Deshalb wurde zuerst in diese Richtung entwickelt.

Nachteil ist jedoch, dass Tabellen sich an kleine Bildschirme, wie z.B. Smartphones schlecht anpassen. Die Nutzung mobiler Geräte zum Surfen erreicht europaweit inzwischen über 50%.⁶ In Deutschland beträgt der Anteil über 40%.⁷ Mittelfristig steigen die Nutzerzahlen weltweit an.

Am Ende wurden die Tabellen zugunsten des modernen Gridsystems und des „responsive design“ ersetzt. Der barrierefreie Zugang wurde im Gridsystem über sogenannte „Rollen“ wiederhergestellt.

⁶ Statista Research Department: „Anteil mobiler Endgeräte an allen Seitenaufrufen nach Regionen weltweit im Jahr 2020“, 09.02.2021, URL: <https://de.statista.com/statistik/daten/studie/217457/umfrage/anteil-mobiler-endgeraete-an-allen-seitenaufrufen-weltweit/> [14.09.2021].

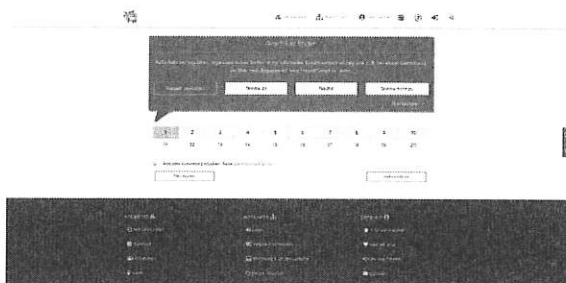
⁷ Saarbrücker Zeitung Verlag und Druckerei GmbH: „Internetnutzung 2019 – Mobile vs. Desktop“, URL: <https://www.sz-medienhaus.de/internetnutzung-2019-mobile-vs-desktop/> [14.09.2021].

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Der Mat-o-Wahl steht im Quelltext inklusive Changelog auf der Open-Source-Plattform Github allen Zielgruppen frei zur Verfügung. <https://github.com/msteudtn/Mat-O-Wahl/>. Außerdem gibt es seit vielen Jahren die dazugehörige Webseite <http://www.mat-o-wahl.de/> mit einer ausführlichen Anleitung, Demo-Version und FAQ. Beide Seiten werden auch nach dem Prototype Fund weiter gepflegt.

Auf Basis des Mat-o-Wahl entstand vor einiger Zeit der „Mitwirk-O-Mat“ <https://mitwirk-o-mat.de/>. Dabei beantworten die Nutzer Fragen zu ihren Interessen und bekommen anschließend diejenigen Gruppen, Vereine und Initiativen vorgeschlagen, mit denen sie die größte Übereinstimmung für ein mögliches Ehrenamt haben. So finden (junge) Menschen ein passendes Engagement und die bestehenden Gruppen erhalten neue Mitglieder.

Im Juni 2021 gewann der „Mitwirk-O-Mat“ den Preis in der Kategorie „Digitales Engagement“ der Initiative „Digital für alle“ aus einer Auswahl von rund 300 Bewerbern.⁸ Die Jury lobte u. a. den „niedrigschwelligen Einstieg ins Ehrenamt“ und den „intuitiven und unterhaltsamen Beitrag zur politischen Bildung“.



Mitwirk-o-Mat



Pressemitteilung des „Digitaltags“

Zu den Jury-Mitgliedern gehörten Vertreter von Medien, Start-Up-Firmen, des Bitkom e. V., Förderer von Barrierefreiheit, eine Staatssekretärin aus dem „Bundesministerium für Familie, Senioren, Frauen und Jugend“, Frau Julia Klöckner (Bundesministerin für Ernährung und Landwirtschaft) sowie weitere Vertreter.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Projektplan konnte eingehalten werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab während des Projekts keine Entwicklungen oder Ergebnisse bei anderen Stellen, welche die Zielsetzung beeinflusst haben.

⁸ Hosenfeld, Anna-Lena; Lamm, Luisa: Zwei Projekte mit 'Preis für digitales Miteinander' ausgezeichnet, 16.06.2021, URL: <https://digitaltag.eu/presse/zwei-projekte-mit-preis-fuer-digitales-miteinander-ausgezeichnet> [14.09.2021].

Richtlinie zum „Software-Sprint“

OAS - OpenAudioSearch

Schlussbericht

Zuwendungsempfänger:

Moreira Veit Heinzmann Schumann GBR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS21S16** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Im Internet gibt es große und schnell wachsende Bestände an interessanten Radio-Features und Podcasts, die unter freien Lizenzen veröffentlicht sind. Ein gutes Beispiel ist unser Partner in diesem Projekt, das [community broadcasting archive \(cba\)](#) oder auch [freie-radios.net](#). Auf diesen Plattformen ist eine Vielzahl an relevanten Audio-Inhalten archiviert, aber die einzelnen Beiträge und Features sind oft nur schwer wiederaufzufinden. Insbesondere für ehrenamtlich getragene Projekte bleibt oft wenig Zeit, um Beschreibungstexte, Metadaten oder gar Transkripte einzupflegen, nachdem die eigentliche Produktion abgeschlossen ist – das gilt umso mehr für Inhalte, die vor zwanzig Jahren produziert wurden und nun im Rahmen von Digitalisierungsprojekten online gestellt werden.

Hier wollen wir mit Open Audio Search Abhilfe schaffen, einer intelligenten Suchmaschine für Audiodateien. Über automatische Spracherkennung wird gesprochene Sprache in Text umgewandelt, der dann in einer Volltextsuchmaschine indiziert wird. Die Engine kann RSS-Feeds abonnieren, um Inhalte von verschiedenen Quellen zu importieren. Über eine webbasierte Oberfläche können Nutzer*innen die generierten Transkripte von Radiosendungen und Podcasts durchsuchen und direkt von der Zeitmarke der Suchergebnisse abspielen. Open Audio Search wird als kollaboratives Open-Source-Projekt entwickelt und kann leicht auf einem beliebigen Server installiert werden.

Mit Open Audio Search wollen wir dazu beitragen, eine zukunftsichere und gut wartbare Open-Source-Plattform für *community media discovery* herzustellen. Wir hoffen, dass wir damit Community-Media-Produzent*innen mit mehr Reichweite und einem breiteren Publikum versorgen können. Denn wenn Inhalte nicht so schnell nach ihrer Produktion wieder verschwinden, sondern stattdessen Teil eines lebendigen und leicht durchforstbaren Archivs werden, dann gehen diese Inhalte auch nicht so schnell wieder verloren, wie es derzeit oft der Fall ist.

Die Meilensteine für das Projekt waren:

- Implementierung des Grundsystems: Python-Webserver mit Job-Queue für ASR-Extraction, Anbindung an Elastic Search, grundlegendes Datenmodell, REST-API
- Design und Implementierung des Such-Indexes: Schema, Queries, Facets, Aggregations
- Domänenspezifische Analyzer für Elastic Search
- Design und Implementierung eines User Interfaces für Volltext-Suche und Filter (Facets)
- Import-Pipeline für mindestens zwei Anbindungen

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

Es gibt bereits kommerzielle Lösungen zur Spracherkennung, aber diese sind für freie Radios und andere nichtkommerzielle Medienprojekte nicht gut verfügbar: Ein minutenbasiertes Preismodell für die Sprachtranskription ist schlicht nicht finanzierbar oder zumindest sehr unattraktiv. Zwar sind Open-Source-Tools sowohl zur Spracherkennung als auch zur Volltextsuche vorhanden, aber diese sind bislang schwer aufzusetzen und nicht integriert - was einen großen händischen Aufwand bedeutet, um zu nutzbaren Ergebnissen zu kommen.

Open Audio Search kann als Tool von Community-Media-Produzent*innen für die Recherche genutzt werden, oder um relevante Beiträge zu verknüpfen. Das Tool erleichtert das Auffinden dieser Inhalte für Interessierte und erhöht so die Reichweite zivilgesellschaftlich erstellter Inhalte. Außerdem werden durch das Bereitstellen einer einfachen Nutzer*innen-Oberfläche mit intelligenten Suchfunktionen KI-Technologien für diesen Sektor zugänglich gemacht. Damit wird ein Beitrag geleistet zur Diskussion um ethische Nutzung von KI und zivilgesellschaftlicher Teilhabe am politischen Diskurs. Das Verwenden von Open-Source-Infrastruktur macht unabhängiger von großen Plattformen und unterstützt damit auch Content- und Datensouveränität.

Ausführliche Darstellung der Ergebnisse

Open Audio Search steht beim ersten Preview-Release, ein größeres Beta-Release folgt bald. Mit heutigem Stand sind die folgenden Features implementiert und nutzbar:

- RSS-Feeds können von beliebigen Quellen importiert und in eine lokale Datenbank gespeichert werden
- Alle in den RSS-Feeds verlinkten Audiodateien können automatisch über unsere Speech-To-Text-Pipeline transkribiert werden
- Die Zuordnung von Metadaten-Feldern in den RSS-Feeds kann angepasst werden
- Umfassende Volltext-Suche in sowohl den Metadaten als auch den automatisch erstellten Transkripten
- Ein responsives, webbasiertes User Interface, über das alle importierten Medien abgespielt und alle indizierten Daten durchsucht und gefiltert werden können
- Wenn die Suche Ergebnisse aus einem Transkript findet, dann kann der zugehörige Audiotrack direkt von der Zeitmarke des Suchergebnisses im Transkript abgespielt werden
- Eine (bislang minimalistische) Admin-UI, um die eingebundenen RSS-Feeds zu verwalten
- Eine HTTP-API mit Authentifizierung, über die Anbindungen an Drittsysteme entwickelt werden können

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Da wir selbst einen Hintergrund in Community-Media-Netzwerken haben und unsere Motivation aus diesem Bereich stammt, besteht unsere Zielgruppe im ersten Schritt aus gemeinnützigen Anbieter*innen deutschsprachiger Audio-/audiovisueller, selbstgehosteter Inhalte. Z.B. Freie Radios im DACH-Raum, individuelle Podcaster*innen, Open GLAM Initiativen, Museen oder Bildungsinitiativen wie Wikimedia.

Wir haben schon einige konkrete Ideen, woran wir in der näheren Zukunft arbeiten wollen:

- Mit unseren Partner*innen von [cba](#) und [freie-radios.net](#) wollen wir in den nächsten Monaten eine Open Audio Search Produktivinstanz für deutschsprachige Community Media in Betrieb nehmen.
- In einem Schwesterprojekt, gefördert durch [netidee](#), setzen wir unsere eigene Trainings- und Evaluationspipeline auf, um die Spracherkennung zu verbessern. Außerdem wollen wir durch Natural Language Processing und das Matchen mit einer Wissensdatenbank wie [Wikidata](#) bessere und bedeutsamere Suchergebnisse erzielen.
- Nächstes Jahr wollen wir ein Replikationssystem zwischen Open Audio Search Instanzen implementieren. Das ist der nächste Schritt in Richtung eines europäischen Inhalte-Netzwerks für Community Media. Mit Unterstützung der [European Cultural Foundation](#) und Partner*innen in Irland und Spanien wollen wir in absehbarer Zukunft eine erste Pilotversion dieses [european cultural backbone](#) starten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben zu Beginn unser Backend als Proof of Concept in Python entwickelt. Während der Förderzeit haben wir entschieden, das Backend stattdessen in Rust zu implementieren. Bis diese Entscheidung finalisiert war, sind ein paar Stunden in später dann obsolete Weiterentwicklung des Python-Proof-Of-Concepts geflossen. Der Rust-Port hat sich in der Zwischenzeit als die richtige Entscheidung erwiesen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

- Wir haben ein funktionsfähiges Setup, das in unserer [Demo](#) ausprobiert werden kann. Diese Demo-Seite indiziert derzeit einigen aktuellen Inhalt von [cba](#) und [freie-radios.net](#). Es handelt sich um ein Alpha-Release, deswegen ist mit Bugs und fehlenden Features zu rechnen. Die Demo wird in unregelmäßigen Abständen aktualisiert und/oder zurückgesetzt.
- Open Audio Search hat eine Website: [openaudiosearch.org](#)
- Der vollständige Quelltext zu Open Audio Search ist auf [GitHub](#) veröffentlicht.
- Die Anleitung zu Installation und Administration findet sich in der [README](#).
- Eine Dokumentation der technischen Architektur ist [hier in unserem Repository](#) zu finden.
- Alle weiteren Ressourcen (z.B. Dokumentation der HTTP/REST-API sowie der internen Rust-API) sind in der [README](#) verlinkt.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Arbeits- und Kostenplanung war überwiegend realistisch und musste nicht angepasst werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Der regelmäßige Austausch mit unseren Projektpartner*innen vom Community Broadcasting Archive (cba) und die Ausrichtung auf eine baldige Production-Instanz war sehr fruchtbar für die gute Zielorientierung der Entwicklung. Durch eine Förderung der [netidee](#), die vom cba akquiriert werden konnte, wird derzeit eine eigene Trainingspipeline zur Verbesserung der Spracherkennungs-Modelle entwickelt, sowie die Sprachanalyse (NLP) zur automatischen Verschlagwortung verbessert.

Richtlinie zum „Software-Sprint“

HierBautBerlin – Übersicht aller Bauprojekte der Stadt Berlin

Schlussbericht

Zuwendungsempfänger:

Bodo Tasche

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S17 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Wenn ein*e Bürger*in Berlins wissen will, warum plötzlich die eigene Straße umgebaut wird, reicht theoretisch ein Blick in das Amtsblatt von Berlin. Dort sind alle Bauausschreibungen vorhanden. Leider sind diese nicht einfach zu verstehen und zu durchsuchen. Mit diesem Projekt wird ein Tool geschaffen, mit dem jede*r genau die Informationen des Amtsblatts und weiterer Webseiten bekommen kann, die das direkte Lebensumfeld betreffen. Alle Bauvorhaben werden auf einer Karte dargestellt. Optional wird es eine E-Mail-Benachrichtigung für neue Projekte im Wohnumfeld geben.

Ich frage mich selber oft, was genau ein Bauvorhaben bezwecken soll. Sei es direkt vor der Haustür oder an anderen Orten in Berlin. Diese Frage zu beantworten ist teilweise nur mit hohem Aufwand möglich. Dies soll Hier Baut Berlin verbessern.

Hierzu gab es zwei große Meilensteine:

- Das Extrahieren der Informationen aus den verschiedenen Quellen, wie z.B. dem Amtsblatt oder dem Presseportal. Dabei wird jede Meldung ein genauer Ort auf einer Karte zugewiesen.
- Darstellen der Informationen auf einer Karte. Benutzer*innen können sich registrieren und ihren Wohnort einstellen, um bei jedem neuen Bauvorhaben eine E-Mail zugesendet zu bekommen.

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

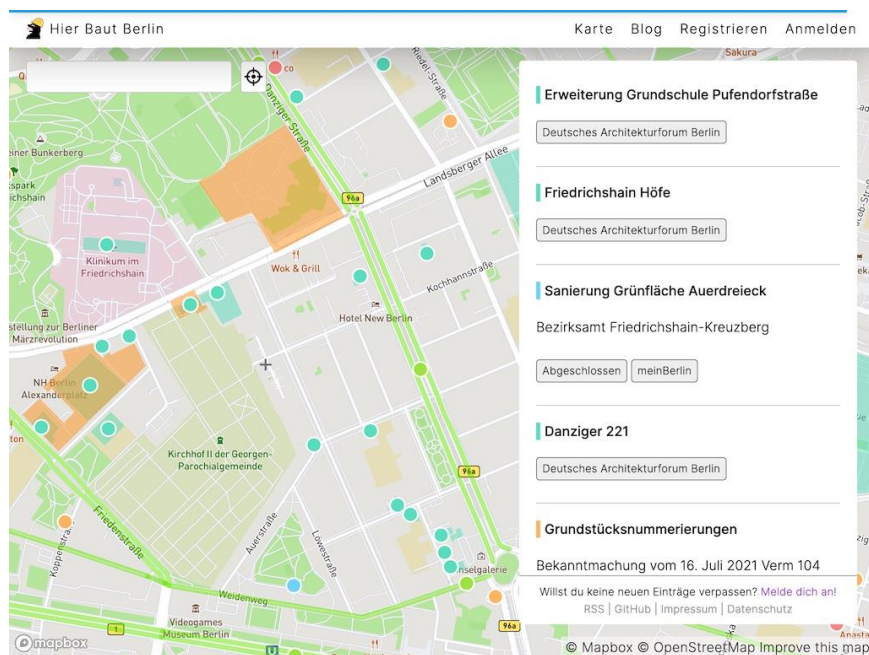
Mit Hier Baut Berlin wird es das erste Mal ein Tool geben, das vielen Informationen zum Thema Bauvorhaben und Veränderungen im eigenen Kiez visualisiert und erfassbar macht. Ohne dieses Tool war es fast unmöglich, aus der Vielzahl an Webseiten die notwendigen Informationen zu gewinnen. Dies ermöglicht es Bürger*innen, sich rechtzeitig zu informieren und bei Beteiligungsverfahren die eigenen Meinungen und Wünsche einzubringen.

Die komplette Anwendung wurde offen gestaltet, sodass es einfach möglich ist, weitere Informationsquellen und Städte hinzuzufügen.

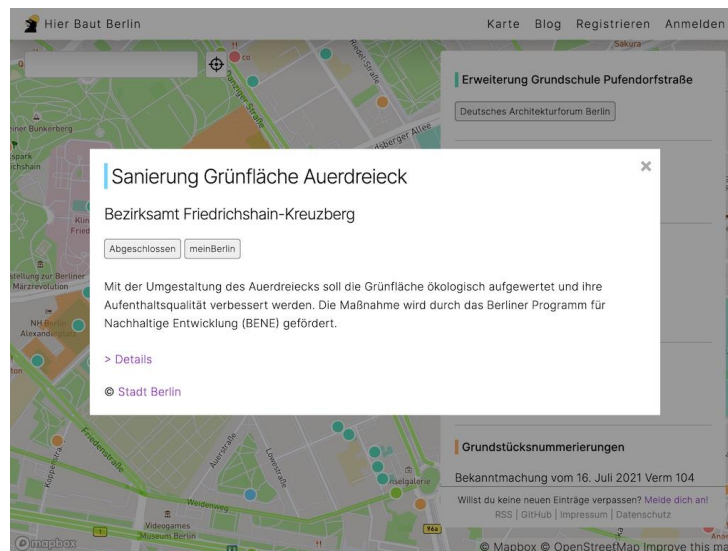
Durch eine API wird es in Zukunft auch möglich sein, die so gewonnen Daten in anderen Anwendungen zu nutzen.

Ausführliche Darstellung der Ergebnisse

Das Projekt wurde erfolgreich umgesetzt. Alle Meilensteine konnten erreicht werden. Unter <https://hierbautberlin.de> findet sich nun eine Karte mit allen Informationen.



Aktuell werden 10 Quellen regelmäßig auf neue Daten geprüft und diese auf der Karte hinzugefügt. Bei Klick auf eine der Überschriften oder der farbigen Punkte erscheint ein Dialog mit Detail-Informationen.



In dem Dialog gibt es zusätzlich die Möglichkeit, weitere Informationen auf der Ursprungsseite der Information zu bekommen. Zum Beispiel um sich auf mein.berlin.de bei einem Beteiligungsverfahren einzubringen.

Falls gewünscht, kann bei neuen Einträgen in einem einstellbaren Radius um einen Ort eine E-Mail versandt werden. Des Weiteren steht ein RSS-Feed bereit, um über neue Einträge informiert zu werden.

Die UX-Schulung, die im Rahmen der Förderung möglich war, hat maßgeblich zum jetzigen Design beigetragen.

Der Quelltext steht auf Github bereit.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Anwendung kann von Bürger*innen genutzt werden, um mit einem einfachen Blick auf die Karte eine Übersicht über alle Bauvorhaben im eigenen Umfeld zu gewinnen. Zusätzlich gibt es die Möglichkeit, sich über neue Einträge via E-Mail benachrichtigt zu werden.

Durch die Open-Source-Stellung können nun andere Entwickler diese Anwendung einfach um weitere Quellen für Berlin oder andere Städte und Gemeinden erweitern. Einige der Komponenten können auch losgelöst für andere Anwendungen genutzt werden. Zu nennen wäre da unter anderem die Erkennung von Adressen und Orten in langen Fließtexten und PDFs.

In den nächsten Monaten werden zu den vorhandenen 10 Quellen weitere 15 angebunden. Daneben wird es noch weitere Verbesserungen der Kartendarstellung und des Filterns der Einträge geben. Eine API zum Auslesen der Daten ist ebenfalls in Planung.

Ich persönlich habe sehr viel über die Analyse von großen Textmengen und deren Darstellung auf einer Karte gelernt. Das letztere Thema wurde von mir zu Beginn unterschätzt.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Es gab keine Arbeiten, die zu keiner Lösung geführt haben.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Anwendung befindet sich unter <https://hierbautberlin.de>. Der Quelltext kann unter <https://github.com/HierBautBerlin/website/> aufgerufen werden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Aufwand der Entwicklung der Kartendarstellung wurde etwas unterschätzt, aber der Mehraufwand konnte durch einen Minderaufwand in der Implementierung des PDF-Parsers für das Amtsblatt wieder aufgefangen werden. Somit hat das Projekt die Arbeits- und Kostenplanung eingehalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab keine Entwicklungen von anderen Personen, die Einfluss auf die Arbeiten an HierBautBerlin hatten.

Richtlinie zum „Software-Sprint“

EP-Votes – Abstimmungen im Europaparlament offen und einfach

Schlussbericht

Zuwendungsempfänger:

Hagemann & Prochaska GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S18 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Die Europäische Union ist einer der größten Demokratien der Welt. Das Europäische Parlament (EP) vertritt über 450 Millionen Menschen. Die Nachvollziehbarkeit von parlamentarischen Abstimmungsprozessen und -ergebnissen ist notwendige Voraussetzung für die politische Meinungsbildung. Zwar wird im EP in den meisten Fällen namentlich abgestimmt. Die Ergebnisse werden trotzdem nur schwer zugänglich und nicht barrierefrei zur Verfügung gestellt.

EP-Votes ändert dies, in dem wir die offiziellen Abstimmungsdaten wie sie vom Parlament bereitgestellt werden zusammen mit weiteren Daten über beispielsweise die Abgeordneten im Parlament vereinheitlichen und übersichtlich, leicht durchsuchbar und verständlich sowie barrierefrei aufbereiten.

Initial war die Arbeit in die folgenden Meilensteine gegliedert:

- Entwicklung eines einheitlichen Datenbackends zur Speicherung der Daten sowie „Scraper“ zur automatisierten Sammlung von Abstimmungsdaten, auch kontinuierlich für zukünftige Abstimmungen.
- Entwicklung eines Twitter-Bots, der automatisch und zeitnah zu Abstimmungen die Ergebnisse auf Twitter teilt.

- Entwicklung einer Webanwendung, die das Suchen, Filtern und Visualisieren von Abstimmungsergebnissen ermöglicht, inklusive Möglichkeiten, die von uns gesammelten und aufbereiteten Daten auch in anderen Anwendungen/für andere Zwecke nutzbar zu machen.

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

Unsere Zielgruppe sind Menschen, die sich für europäische Politik, insbesondere die Arbeit im Europaparlament interessieren. Ohne Vorkenntnisse vorauszusetzen, können diese sich mit Hilfe von EP-Votes über Abstimmungsergebnisse, sowie ,durch die Parlamentsverwaltung neutral zusammengefasst, die Inhalte von Abstimmungen informieren.

Die von uns gesammelten und aufbereiteten Daten stellen wir unter einer CC Lizenz zur Verfügung. So können andere Projekte oder beispielsweise Journalist*innen auf unserer Arbeit aufbauen.

Ausführliche Darstellung der Ergebnisse

Während unserer Arbeit an EP-Votes haben wir die wesentlichen Punkte, welche wir uns vorgenommen haben, erreicht. EP-Votes bietet mit Ende der Förderphase die Möglichkeit, einzelne Abstimmungen, das Abstimmungsverhalten aller Parlamentarier*innen (auch nach Gruppe und Land aggregiert) sowie eine inhaltliche Zusammenfassung der meisten Abstimmungen anzusehen.

Außerdem ist für jedes Mitglied des Parlaments auf einer eigenen Seite ersichtlich, wie dieses jeweils abgestimmt hat. Zusätzlich werden grundlegende Informationen zu den Abgeordneten angezeigt (z.B. Kontaktmöglichkeiten, Nationalität).

EP-Votes verfügt über ein Suchmaschine, die beim Eingeben der Suchanfrage in Echtzeit Ergebnisse anzeigt und es ermöglicht, Abstimmungstitel als auch die jeweiligen inhaltlichen Zusammenfassungen zu durchsuchen. So können konkrete Abstimmungen gefunden werden, ohne dass Nutzer*innen den exakten Titel des jeweiligen Anliegens im Parlament kennen müssen.

Der Datenbestand deckt momentan 2021 fast vollständig ab. Zukünftige Abstimmungen werden regelmäßig automatisch von der Parlamentsseite extrahiert.

Bei der Entwicklung haben wir ein großes Augenmerk auf eine gute Abdeckung durch automatisierte Tests und eine Wahl von möglichst einfachen Technologien gelegt. Außerdem haben wir Responsivität und Barrierefreiheit beachtet.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

EP-Votes ermöglicht es die Abstimmungsergebnisse im Parlament an einer zentralen Stelle und einfach verständlich abzurufen. Dies vereinfacht den Prozess im Vergleich zum Vorgehen auf der offiziell-

len Parlamentsseite. Dort sind die Ergebnisse an verschiedenen Stellen verteilt und schwieriger nachzuvollziehen.

Die Open-Source-Lizenz der Anwendung ermöglicht es, dass weitere interessierte Menschen in Zukunft an EP-Votes mitwirken können. Außerdem machen wir unsere Datenbank frei nutzbar, wodurch die Daten für andere Anwendungsfälle, zum Beispiel journalistischer Natur, nutzbar werden.

Wir werden EP-Votes ehrenamtlich um weitere Features erweitern und die korrekte Sammlung zukünftiger Abstimmungsdaten sicherstellen. Es besteht außerdem erster Kontakt zu Bildungsinitiativen, die zur Europäischen Politik arbeiten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die initiale Idee, einen Twitter-Bot zur zeitnahen Veröffentlichung der Abstimmungsergebnisse zu programmieren, haben wir während der Förderphase verworfen. Dies liegt daran, dass unsere Daten momentan auf zwei verschiedenen Datenquellen des Parlaments fußen, von denen eine erst mit deutlicher Verzögerung veröffentlicht wird. Dieser Umstand war uns initial nicht bewusst.

Um die Verbreitung der Anwendung anzuregen, haben wir auf mobilen Geräten eine Möglichkeit zum einfachen Teilen von Abstimmungen bereitgestellt und zeigen beispielsweise die Diagramme, die das Ergebnis einer Abstimmung darstellen, auch als Linkvorschau in sozialen Netzwerken und Messenger-Diensten an.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Unsere Webseite ist unter <https://howtheyvote.eu/> erreichbar. Den Quellcode haben wir auf [GitHub](#) veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Wir haben den initial veranschlagten Kosten- und Arbeitszeitplan einhalten können.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Nicht zutreffend.

Richtlinie zum „Software-Sprint“

Accessy – Barrierefreiheit für sehbeeinträchtigte Menschen in Städten

Schlussbericht

Zuwendungsempfänger:

Michael Helmbrecht

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S19 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)?

Blinde Menschen sind in fremden Umgebungen auf Hilfe angewiesen. Normalerweise wird ihnen ein Wege mehrmals von Freunden oder Assistenzdiensten gezeigt. Erst danach können sie diesen Weg eigenständig ablaufen. Der Nachteil ist, dass es sich nur um diesen einen Weg handelt. Man kann also nicht wirklich von einer Auflösung dieser Barriere sprechen.

Kartendienste von namenhaften Anbietern helfen an dieser Stelle auch nicht weiter. Sie beinhalten keine Informationen zu Bürgersteigen, Ampeln etc. und sind damit nur für sehende Menschen ausgelegt.

Accessy versucht dieses Problem auf mehrer Arten zu lösen. Zum einen über eine iOS App und zum anderen über eine Serveranwendung.

Freund:innen, Bekannte, Assistenzdienste, Städte oder grundsätzlich jeder andere kann über die App Routen anlegen. Zum Beispiel vom Bahnhof zur Universität oder von einem zu Hause zum anderen. Diese Routen können dann mit der anderen Person geteilt werden. Wenn man Routen anlegt, dann generiert man automatisch Infrastrukturen (Gehwege, Treppen, Ampeln, Zebrastreifen, Verkehrsinseln etc.). Diese Infrastruktur steht automatisch allen zur Verfügung. Sie können sie erweitern oder an einer anderen Stelle ganz neue hinzufügen. Dadurch entsteht nach und nach ein Netzwerk für die verschiedenen Städte. Je mehr Infrastruktur es gibt, desto mehr Informationen hat Accessy um eigenständig Routen berechnen zu können.

Das Ablaufen einer Route wird dann blindengerecht über eine Sprachausgabe ermöglicht. Anders als bei konventionellen Kartendiensten gibt Accessy weitere Informationen zu den einzelnen Wegabschnitten aus: Anzahl an Treppenstufen, ob es ein Geländer gibt und wenn, auf welcher Seite, Vibration oder Piepton an Ampeln, ob es sich um eine Fußgängerzone handelt und vieles mehr.

Meine persönliche Motivation für dieses Thema entwickelte sich im Sommer 2019 im Rahmen eines Service Design Prozesses für ein Verkehrsunternehmen. Dort interviewte ich eine blinde Studentin. Sie wohnte nicht in dieser Stadt und erzählte mir, dass sie nur den Weg vom Bahnhof zu ihrer Universität kennt. Dieser Weg wurde ihr einige Male von einem Assistenzdienst gezeigt. Ansonsten konnte sie sich nicht frei in der Stadt bewegen.

Ich war in diesem Moment fassungslos, dass es noch keine digitale Lösung gibt, die eine freie Mobilität für blinde Menschen ermöglicht. Auch 6 Monate später hat mich dieses Problem nicht losgelassen und ich habe mich an eine Lösung gesetzt.

Geplant war, das Projekt direkt mit einer Research Runde zu starten und dafür alle öffentlichkeitsarbeitsrelevanten Medien vorzubereiten (Website, Social Media etc.), dann mit den Ergebnissen in die Designkonzeption zu gehen und dann mit der Programmierung zu starten.

Die geplanten Meilensteine waren:

- User Research mit Service Design Methoden (zB. mit der Blindenschulen in Marburg)
- User Stories
- Wireframes und barrierefreie Bedienkonzepte erstellen und mit Usern testen
- grobes Prototyp-Design entwickeln und mit Usern testen
- Serverinfrastruktur aufsetzen
- wichtigsten Features entwickeln (App):
- Navigation in App
- verschiedene Marker eingeben können und damit Routen erstellen
- Route ablaufen können durch Sprachausgabe
- Online Marker und Routen teilen können
- Route aus anderen Routenteilen berechnen können
- barrierefreie POIs angezeigt bekommen

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?

Meine Zielgruppe sind als aller erstes blinde Menschen. Sie sind die primären Endnutzer:innen der App. Hinzu kommen alle Gruppen, die für das Eintragen der Infrastruktur verantwortlich sein können. Dazu zählen Familie, Freunde, Städte und Assistenzdienste.

Blinde Menschen profitieren von Accessy durch eine neue Freiheit in der Mobilität. Sie können unabhängig Wege ablaufen, die sie vorher nicht kannten. Dabei profitieren sie von den Details der Infrastruktur (Ampeln, Straßen, Treppen etc.) die andere für sie angelegt haben. Unsicherheiten, ob

die Treppe vor ihnen ein Gelände hat oder nicht und wieviele Stufen sie hat, werden aus dem Weg geräumt.

Städte profitieren durch die erweiterte Barrierefreiheit für ihre Mitbürger:innen. Das macht Städte, in denen Accessy vorhanden ist, attraktiv für Menschen mit Sehbeeinträchtigung. Auch können Städte Routen anlegen, die an „schönen“ Orten vorbeiführen wie z.B. an einem Bach oder eine Strecke durch den Stadtpark.

Für Familien und Freunde ist es eine Erleichterung, wenn die blinde Person selbstständig verschiedene Wege ablaufen kann. So bleibt mehr Zeit für einen schönen Spaziergang zusammen, der nicht unbedingt mit Erledigungen verbunden ist.

Da Accessy als Open Source Projekt angelegt ist und eine REST Api zur Verfügung stellt, können andere Dienste auf den Service zugreifen. Diese Dienste profitieren dabei von der vorhandenen Infrastruktur der Städte und dem Routing Service von Accessy. So sind auch Kombinationen aus verschiedenen Diensten vorstellbar. Ich habe bereits eine Anfrage der Stadt Kaiserslautern zur Integrationen in deren Mobilitätsapp für blinde Menschen, die auch eine Navigation in Innenräumen ermöglichen möchte. Accessy ist zudem Sprach- und Ortsunabhängig. Die App wird in Deutsch und Englisch erscheinen. So kann der Service auch in allen anderen Ländern genutzt werden. Eine Erweiterung der Sprachen ist auch vorstellbar und leicht zu realisieren.

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?

Von allen geplanten Meilensteinen konnte nur ungefähr die Hälfte realisiert werden. Das lag zum größten Teil an technischen Problemen, die nach und nach ersichtlich wurden. Zudem wäre es gut gewesen direkt mit der Entwicklung zu starten und erst später Dinge wie die Website zu machen. So wären die technischen Probleme viel schneller erkennbar gewesen und man hätte mehr Zeit gehabt diese zu lösen.

Konkret wurde das Anlegen der Infrastruktur in der App und auf dem Server umgesetzt. Hierbei gab es die meisten technischen Probleme, die mit dem Kartendienst von Apple zu tun hatten. Durch die ungenaue GPS Ortung von Handys musste auch der Flow zum Anlegen der Infrastruktur überdacht werden. Hinzu kam eine größere Recherche und ein Vergleich von verschiedenen Kartendienstleistern (Apple, Google, OpenStreetMaps). Hinzu kommen viele kleine Helfer, die grundsätzlich beim Umgang mit dem Apple Kartendienst unterstützen. Für SwiftUI ist eine Library entstanden, die die wichtigsten Funktionen vom UIKit zur Verfügung stellt. Außerdem noch eine Library, die Berechnungen mit GPS Koordinaten erleichtern und z.B. Winkel zwischen zwei Wegen berechnen kann, entstanden.

Für Drittnutzer wurde eine API Library für iOS veröffentlicht, die die Kommunikation mit dem Accessy Server ermöglicht.

Meine größte Erkenntnis aus dem Projekt ist sicher, dass es sinnvoll ist, wenn man alleine an dem Projekt arbeitet, direkt mit der Entwicklung zu starten und nicht den Weg über Research zu gehen. So erhält man selbst ein gutes Bild davon, was möglich ist und was nicht und kann dann im Research sehr viel besser darauf eingehen.

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?

Durch die technischen Probleme kam es zu einer Verschiebung des Arbeitsplans um ca. 1,5 Monate. Da die Kostenplanung durch meinen Stundensatz nach ca. 4 Monaten aufgebraucht war und eh geplant war, die weiteren 2 Monate trotzdem an dem Projekt zu arbeiten, kam es hier grundsätzlich zu keiner Verschiebung in den Kosten.

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?

Richtlinie zum „Software-Sprint“

DonationMap

Schlussbericht

Zuwendungsempfänger:

Gatchbin GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S20 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation?

Unsere Motivation war es, durch die Optimierung des Spendenprozesses und die Förderung des Recyclings durch Wiederverwendung Abfall zu reduzieren.

Welches Problem wolltest Du mit Deinem Projekt lösen?

Wir lösen das Problem von verstreute, nicht leicht verfügbare (erfordert viele Klicks, um die nützlichen Informationen zu finden) und/oder veraltete Informationen über Spendezentren, danke eine zuverlässige zentrale Datenbank von Spendezentren für Kleider oder Lebensmittel.

Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

1. Backend mit Datenmodellen implementieren.
2. Schreiben einer umfangreichen Integrationstest-Suite zur Sicherstellung der funktionalen Domänenlogik
3. Front-End implementieren (einschließlich Karten in OpenLayers mit Datenpunkten, Zoom-Funktion, Straßenansicht usw.)
4. Hinzufügen der Möglichkeit für einen Benutzer, Spendenzentren hinzuzufügen/zu bearbeiten.
5. Hinzufügen der Möglichkeit für einen Benutzer, Bewertungen hinzuzufügen/zu bearbeiten.
6. Bereitstellung und QA-Tests
7. Sammeln Sie den ersten Datensatz von NRO's und anderen Organisationen, um die Datenbank mit einem speziellen Ruby-Skript zu füttern.
8. Werbung für die Webapplikation.

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

Wer ist die Zielgruppe für Deine Lösung?

Die App richtet sich an zwei Zielgruppen: an Wunschspender, die nicht wissen, wo und wie sie Spendenzentren in ihrer Nähe finden, welche Richtlinien sie verfolgen und wie ihre Spende verteilt wird; und an Spendenzentren, die mehr Spenden als bisher erhalten werden.

Wie profitiert sie von den Ergebnissen Deines Projekts?

Davon profitieren das Spendenzentrum und die Endempfänger der Spenden. Auch die Menschen, die spenden wollen, profitieren davon, da sie unerwünschte Gegenstände loswerden können.

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

Civic Tech (Werkzeug für das Engagement der Bürger) und Softwareinfrastruktur (Infrastruktur für die Wiederverwendung)

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt?

Wir haben ein MVP mit folgenden Funktionen erstellt:

- Auf der Homepage wird eine Vollbildkarte angezeigt
- Geolokalisieren Sie den Standort der Benutzer und zeigen Sie Spendenzentren in ihrer Nähe an
- Ein Benutzer kann auf das Spendenzentrum klicken, um detailliertere Informationen zum Spendenzentrum zu erhalten
- Folgende Informationen werden über ein Spendenzentrum angezeigt
 - Name
 - Spendenkategorien
 - Kontaktdetails
 - die Anschrift
 - Öffnungszeiten
 - So werden Spenden verwendet
- Benutzer können auch ein Spendenzentrum auf der Karte hinzufügen

Folgende Softwarebibliotheken wurden verwendet, um MVP zu erstellen:

Python, Django, OpenLayers, Javascript, HTML und CSS

Konnten alle Meilensteine erreicht werden? Ja

Ja

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

Wir haben gelernt, dass es besser ist, mit einem kleineren Markt zu arbeiten, um eine Anpassung des Produkts an den Markt zu erreichen, und ein Playbook zu erstellen, um den Expansionsprozess zu dokumentieren und zu wiederholen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

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

Die Vorteile sind mehr Spenden und weniger Abfall.

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

Wir sind noch dabei, das herauszufinden.

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

Ja, wir sind dabei, das Feedback, das wir zu unserem MVP erhalten haben, in die Neugestaltung der Website einfließen zu lassen und Ideen zu erforschen, um eine ausreichende Beteiligung der Nutzer zu erreichen.

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

Ja, es hat uns wertvolle Arbeitserfahrungen gebracht, von der Gründung einer GbR bis hin zum Engagement der Nutzer.

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?

Am Anfang hatten wir die Idee, die Website für ganz Deutschland zu implementieren, aber wir haben bald gemerkt, dass es besser ist, sich zu konzentrieren. Es war schwierig, die Daten für das ganze Land zu trennen. Es war schwierig, Strategien für die Nutzeransprache in Deutschland zu entwickeln. Indem wir uns auf Berlin konzentrierten, konnten wir die Nutzergruppen leichter erreichen und den Markt besser verstehen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Unsere Website: <https://donationcentermap.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?

Zero360 und PrototypeFund gaben nützliches Feedback zum Projektmanagement.

Richtlinie zum „Software-Sprint“

LEF – Local Emission Framework

Schlussbericht

Zuwendungsempfänger:

Stein Timme Edelkötter Rohmund GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S21 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Der Klimawandel ist die Herausforderung unserer Generation und erfordert Verhaltensänderungen von jedem Mitglied der Gesellschaft. Aktuell entwickeln viele Städte unabhängig voneinander eigene Strategien zur Umsetzung der vom Bund vereinbarten Klimaziele. Das Local Emission Framework (LEF) bietet ein Open-Source-Portal zur Visualisierung der kommunalen Klimastrategien in ganz Deutschland.

Strategien, Ziele und Maßnahmen und die zugehörigen Informationen über den Klimawandel direkt vor der eigenen Haustür können so für Bürger*innen auf einen Blick dargestellt werden.

Dies ermöglicht eine einfachere Kommunikation zwischen Städten und Städten, Bürger*innen und der kommunalen Verwaltung sowie zwischen Bürger*innen und Bürger*innen und erlaubt somit mehr Partizipation und eine effizientere Umsetzung von lokalen Klimaschutzmaßnahmen.

Ursprünglich geplante Meilensteine des Projekts waren:

1. Datenstruktur: Konzipierung einer flexiblen Datenstruktur für städtische und nutzerbasierte Klimadaten
2. Datenbank: Implementierung eines Datenbanksystems mit lokal als auch öffentlich zugänglichen Bereichen, in der die zuvor erarbeitete Struktur eingerichtet wird
3. Web-API: Entwicklung einer Schnittstelle, die Klimadaten, ebenso wie dynamische Nutzerdaten, entgegennimmt und über welche gespeicherte Daten aggregiert abgefragt werden können.

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

Die Lösung richtet sich an Kreise, Städte und Kommunen, die zugleich Sichtbarkeit und Effizienz der kommunalen Klimastrategie optimieren wollen und dabei eine nachhaltige und sichere Open-Source-Lösung einsetzen möchten.

Diese können die entwickelte Open Source Plattform über entsprechend eingerichtete Verwaltungs-Accounts nutzen, um ihre Klimaziele übersichtlich und online transparent für Interessierte zur Verfügung zu stellen. Davon profitieren dann auch Bürger*innen und die Zivilgesellschaft. Zusätzlich lassen sich unterschiedliche Teile des LEFs als interaktive Inhalte auf Webseiten von Kreisen und Kommunen einbetten.

Insgesamt gesehen bietet unser Projekt die erste Open Source Plattform für eine Übersicht über den kommunalen Klimaschutz in Deutschland. Dies könnte langfristig eine einfachere Evaluation desselben ermöglichen und damit einen wichtigen Beitrag zur Strategieentwicklung bzgl. dieses Themas leisten.

Das Projekt hat somit viele Anknüpfungspunkte zu den Themen Umweltschutz, Open-Data und Transparency.

Ausführliche Darstellung der Ergebnisse

Innerhalb des Projektes wurde eine flexible Datenstruktur für kommunale Klimadaten vieler Art entwickelt. Gleichzeitig wurde ein entsprechendes Backend zur Verwaltung entsprechender Daten aufgesetzt. Darüber hinaus wurde eine Homepage als Webschnittstelle zum Abruf aller in der Datenbank verfügbaren Daten in ansprechender und aufbereiteter Form entwickelt.

Das LEF gibt es als Portal und als eingebettete Version. Wer sich einfach einen Überblick über mehrere Städte verschaffen möchte, kann dies am Handy oder im Browser auf portal.emission-framework.org tun. Außerdem haben Städte die Möglichkeit, die gleichen Informationen auf ihren Webseiten einzubetten. Das Portal ist nach Leitfragen organisiert. Alle Abschnitte (sogenannte Widgets) bieten Informationen zu einer Frage:

Im Widget “Ziele & Maßnahmen” können Städte ihre Ziele zum Thema Klimaschutz auf einem Zeitstrahl darstellen. Für Städte gibt es die Möglichkeit, zu jedem Ziel bereits durchgeführte oder geplante Maßnahmen hinzuzufügen.

Auf Basis der Daten des Deutschen Wetterdienstes haben wir zudem für die allermeisten Städte in Deutschland Wetterdaten der letzten Jahrzehnte aufbereitet. Diese können als Warming Stripes oder Klimadiagramme dargestellt werden. Die Nutzer*innen haben zusätzlich die Möglichkeit, Parameter der Darstellung interaktiv zu verändern und können so die historischen Wetterdaten selbst erkunden.

Um es Verantwortlichen in Kommunen und Kreisen möglichst einfach zu machen, ihre Informationen einzutragen und aktuell zu halten, haben wir die Verwaltung des Portals nach dem “What you see is what you get”-Prinzip gestaltet. So lassen sich neue Ziele mit wenigen Klicks hinzufügen und veröffentlichen. Jede Stadt kann in ihrem von uns freigeschalteten Account ihre Region verwalten

und entscheiden, welche Informationen dargestellt werden sollen. Es gibt außerdem die Möglichkeit in benutzerdefinierten Widgets eigene Daten hochzuladen und darzustellen.

Damit wurden alle Meilensteine zumindest in angepasster Form erreicht. Eine wichtige Schlussfolgerung aus der Arbeit im Projekt war für uns die Erkenntnis, dass es bei der Kommunikation von kommunalen Projekten nicht nur um die Erhebung und Transparenz von Daten geht, sondern insbesondere auch um ihre ansprechende Aufbereitung. Weiterhin wurde uns klar, dass es bisher kaum eine Stelle gibt, an der die kommunalen Strategien zusammenfließen und in Austausch gebracht werden können. Der Bedarf hierfür besteht aber dringend.

Zusätzlich haben wir in unserem Projekt gelernt, dass es immer hilfreich ist, Zwischenergebnisse mit Außenstehenden zu evaluieren. Dadurch wurde uns u.a. klar, dass wir die Nutzung unseres Projektes barrierefrei und auf unterschiedlichen Webseiten gestalten wollen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Das Ziel einer online Open Source Lösung für kommunale Klimaschutzmaßnahmen wurde erreicht. Das Portal steht für alle Interessierten unter <https://portal.emission-framework.org/> zur Verfügung und kann genutzt werden. Unsere Zielgruppe hat dadurch eine funktionierende Open Source Plattform zur Darstellung von Klimaschutzplänen unterschiedlicher Kommunen in Deutschland.

Durch die Open Source Stellung unseres Projekts, kann dieses von allen Interessierten genutzt und weiterentwickelt werden. Zudem können Datenstruktur und Backend auch als Grundlage für ähnliche Projekte zur Darstellung von Projekten in der Kommunalpolitik genutzt werden.

Falls wir es schaffen, die Datenbank weiter auszubauen und mit vielen Kommunen in Deutschland ins Gespräch zu kommen, könnte sich das LEF in Zukunft zudem zu einer sehr wertvollen und offenen Datenquelle für kommunale Informationen bezüglich des Klimaschutzes entwickeln. Für Interessierte und Wissenschaft wollen wir in Zukunft gerne eine entsprechende Schnittstelle (API) entwickeln. Dafür bedarf es allerdings noch vielen weiteren Datenpunkten, die auf unserer Plattform eingepflegt werden müssen.

Uns persönlich hat die Arbeit in unserer persönlichen und fachlichen Weiterentwicklung vorwärtsgebracht. Während der Entwicklung der Plattform haben wir viele Dinge wie bspw. Rechteverwaltung unterschiedlicher Nutzer, automatische Generierung von Service-Mails, Aufbereitung und Sammlung von Wetterdaten etc. zum ersten Mal ausprobiert und konnten viele Erkenntnisse direkt sinnvoll im LEF umsetzen. Weiterhin war es äußerst bereichernd, das komplette Projekt in selbständiger Arbeit durchzuführen und somit für alle Ziele, Anpassungen und auch das Projekt- und Gruppenmanagement selbst verantwortlich zu sein.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ursprünglich hatten wir überlegt, ob es möglich ist, dynamische Nutzerdaten von Bewohner*innen der unterschiedlichen Städte zu sammeln und darzustellen. Leider existieren entsprechende Daten in dieser Form kaum. Daher haben wir den Fokus unseres Projekts auf eine attraktive und vielseitig anpassbare Visualisierung der Klimaschutzpläne der Kommunen und entsprechender kommunal verfügbarer Daten gesetzt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

<https://emission-framework.org/>

<https://portal.emission-framework.org/>

<https://github.com/Local-Emission-Framework/lef-portal>

<https://github.com/Local-Emission-Framework/lef-backend>

<https://demoweeek.prototypefund.de/projects/01-local-emission-framework.html>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Im gesamten Projektverlauf mussten wir im Vergleich zu unseren Erwartungen etwas mehr Data Science und etwas weniger Zeit in die Backendentwicklung investieren. Zudem stellen wir rückblickend fest, dass die Arbeitslast zum Ende des Projekts leicht zugenommen hat.

Dennoch konnten wir die geplante Arbeits- und Kostenplanung gut einhalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Zu Beginn der Projektlaufzeit und Entwicklung haben wir das bereits bestehende – aber nur teilweise aktuelle Projekt <https://www.klimaschutz-planer.de/> entdeckt, welches mit dem LEF vergleichbare Ziele verfolgt.

In Absprache mit der Projektbetreuung haben wir mehrere Schlussfolgerungen gezogen:

- 1) Die Bedienung unseres Portals muss für Bürger*innen, aber auch Vertreter der kommunalen Verwaltung so einfach wie möglich sein.
- 2) Auch wenn genaue CO²-Bilanzierung natürlich wichtig für den Klimaschutz und kommunale Klimaschutzpläne ist, ist sie aus unserer Sicht (in Bezug auf das Gelingen der Klimastrategie), nicht die wichtigste zu transportierende Information, denn Bilanzierung ist aufwändig, multifaktoriell beeinflussbar und schwer zu verstehen.
- 3) Ob und inwieweit die Ziele ausreichend sind, ist Gegenstand von wissenschaftlichen und politischen Debatten. Für eine mündige Bürgerentscheidung zentral sind daher folgende Informationen in übersichtlicher und kompakter Form:
 - a. Festgelegte Klimaschutzziele
 - b. Geplante und durchgeführte Klimaschutzmaßnahmen
 - c. Zugrundeliegende Daten als Zwischenstand der aktuellen Ziele und Maßnahmen

Auf Basis dieser Schlussfolgerungen haben wir anschließend das LEF entwickelt und hoffen, dass es eine moderne, offene, übersichtliche (und damit einfach zu verstehende) und aktiver genutzte Übersicht über den kommunalen Klimaschutz in Deutschland bietet.

Richtlinie zum „Software-Sprint“

OPEN JOYN

Schlussbericht

Zuwendungsempfänger:

Simon Deeg, Andreas Picker

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S22 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Aktuell fehlen einfach zu nutzende Softwarewerkzeuge für Bürger*innen, um Holzkonstruktionen anwenderfreundlich parametrisch zu individualisieren. Das heißt auf die Bedarfe diverser Nutzer*innen einzugehen und unterschiedlichen Alltagsrealitäten gerecht zu werden. Weiter ein gut gestalteter Prozess, um Anwender*innen zu befähigen Konstruktionen zu individualisieren, Bauteile einzukaufen und diese aufzubauen. Das Projekt OPENJOYN soll ein relevanter Beitrag sein, um dieses beschriebene Problem zu lösen.

Unsere Motivation für das OPENJOYN Vorhaben war somit die Entwicklung eines benutzerfreundlichen Softwarewerkzeuges, welches möglichst viele Anwender*innen befähigt Holzkonstruktionen, wie Möbel, für den Alltagsgebrauch individuell anzupassen, die benötigten Bauteile selbstständig einzukaufen und die Konstruktionen aufzubauen. Auf einer Onlineplattform können Konstruktionen aus einem visuell ansprechenden Katalog ausgewählt und parametrisch auf den Individuellen Bedarf angepasst werden (bspw. Stehtisch für Homeoffice-Arbeitsplatz).

Anschließend können Einkaufs- und Aufbaupläne automatisch generiert werden, welche Anwender*innen durch den regionalen Einkauf der Teile und die Montage der individualisierten Konstruktion leiten. Da Laien angesprochen werden, müssen Lösungen entwickelt werden, die diese Zielgruppe explizit abholt und durch den skizzierten OPENJOYN-Prozess leitet.

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

Der Theorie des Civic Tech folgend, soll sich OPENJOYN zu einem freien Konstruktionskatalog der global vernetzten Gesellschaft entfalten und kann über die potentiell große Anwenderzahl einen großen Impact erreichen. OPENJOYN legt dabei einen starken Fokus auf anwenderzentriertes Prozessdesign, um einen möglichst großen Personenkreis ohne handwerkliche Vorkenntnisse zu

ermächtigen. Mit OPENJOYN können neben zeitlosen Produkten auch aktuelle, gesellschaftlich relevante Konstruktionen integriert werden. (Covid19- Hygienetrenner). Bürger*innen sollen durch OPENJOYN ganz praktisch befähigt werden, mit eigenen Händen Konstruktionen und Möbel herzustellen, welche zuvor individualisiert wurden. Weiter können Anwender*innen einen neuen Zugang zu der gegenwärtigen Produkt- und Konsumbeziehung erhalten, diese reflektieren und durch die Nutzung des Tools erweitern. So bringt die selbstständige Auswahl und der Einkauf der Baumaterialien die Möglichkeit mit sich, über die Herkunft der Materialien zu reflektieren und diese selbst zu beeinflussen. Beispielsweise wird es möglich, rein regionale Produkte zu wählen.

Ausführliche Darstellung der Ergebnisse

Alle geplanten Meilensteine wurden erfolgreich bearbeitet. Die relevantesten Ergebnisse der Entwicklung werden im folgenden dargelegt.

1.

User-Journey für OPENJOYN und Zielgruppenbefragung

Die User-Journey des OPENJOYN-Gesamtprozesses wurde entwickelt und die wichtigsten, daraus resultierenden Fragen und Hypothesen in einer online Google-Forms Umfrage eingearbeitet. Diese Umfrage wurde in mehreren Netzwerken wie z.B. Hochschulen, Vereinen und anderen Organisationen verteilt. Mit dem Feedback der Teilnehmer*innen konnten die wichtigsten Hypothesen des OPENJOYN-Prozesses bestätigt werden:

So wurde deutlich, dass das geplante Softwaretool durch ein Tutorial in Text und Bild flankiert werden muss, da es viele interessierte Anwender gibt, die nicht viel, bis gar keine Erfahrung mit handwerklicher Holzverarbeitung haben. Ein praktisches Beispiel hierfür ist, dass zwar fast alle Anwender*innen einen Akkuschrauber besitzen, ein Großteil jedoch nicht in der Lage ist damit präzise Löcher in eine Holzlatte zu bohren. Diese besonderen Anforderungen konnten wir durch unserer Befragung identifizieren und beeinflussten die Entwicklung von OPENJOYN maßgeblich.

2.

Aktualisierung der Recherche: Stand der Technik und technische Einbindungsmöglichkeiten

In diesem Schritt wurde die Recherche über den Stand der Technik spezifiziert. Hierbei wurden bestehende Open Source Tools und technische Möglichkeiten recherchiert, welche die Entwicklung des OPENJOYN-Tools unterstützen könnten. Tools wie z.B. OpenSCAD, oder Shapediver wurde genauer untersucht, und entschieden, in wieweit eine Integration oder Kooperation zielführend sein könnte. Ein wichtiges Ergebnis war hierbei, dass viele Tools und Anwendungen kostenpflichtig sind oder der Funktionsumfang nicht zu unserer Featureliste passt. Schlussendlich verzichteten wir darauf, andere OpenSource-Softwaremodule zu integrieren, da unser spezieller Prozess nicht unterstützt werden konnte.

3.

Generische Parametrisierungsstrategie für Holz-Konstruktionen

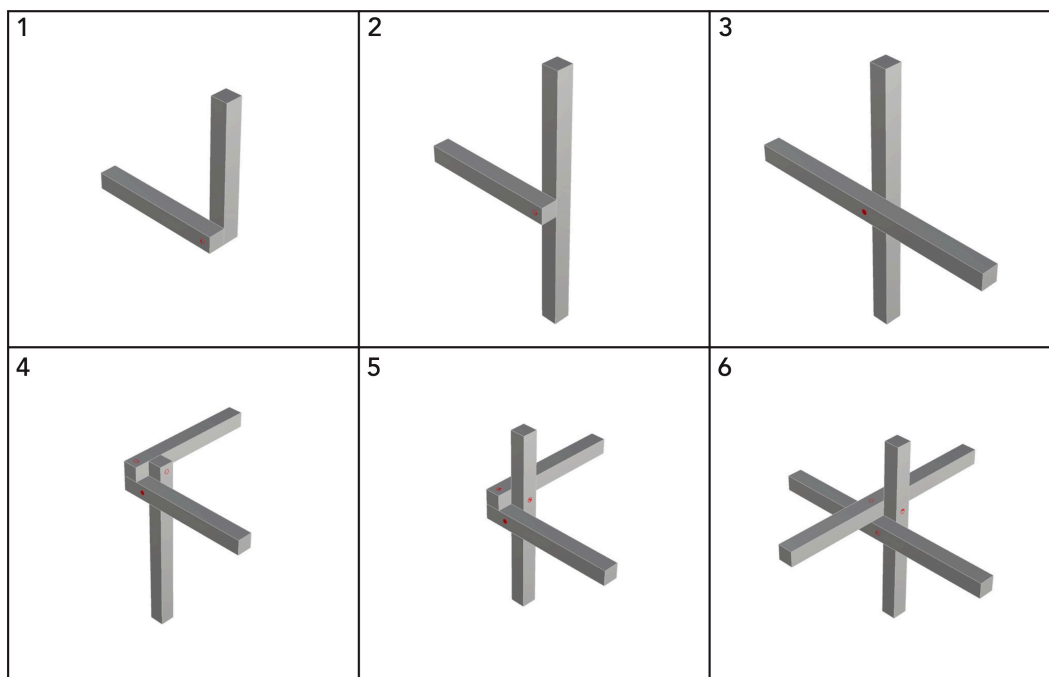
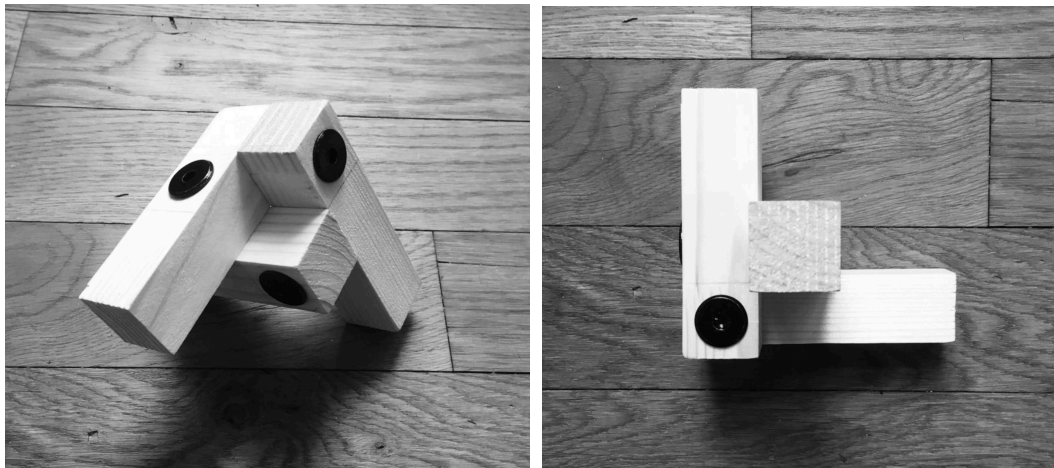
Der entscheidende Punkt in diesem Entwicklungsschritt war es eine Parametrisierungsstrategie für Holz-Konstruktionen zu entwickeln, welche auf viele unterschiedliche Konstruktionen anwendbar ist. Gleichzeitig war zu berücksichtigen, dass diese Strategie niederschweligen Konstruktionsparametern folgt, da die Bearbeitung und der Aufbau händisch erfolgen soll. Es ging hierbei darum, die technischen Möglichkeiten der Softwareentwicklung streng mit den Fähigkeiten der Anwender

abzustimmen. So wurde ein Gestaltungsrahmen für die Parametrisierung der Konstruktionen entwickelt, der als Grundlage für die Entwicklung des Verbindungskatalogs genutzt werden konnte. In diesem Schritt wurden mit dem Plugin *Grasshopper* der CAD-Software *Rhinoceros* Prototypen für parametrische Konstruktionen erstellt.

4.

Verbindungskatalog

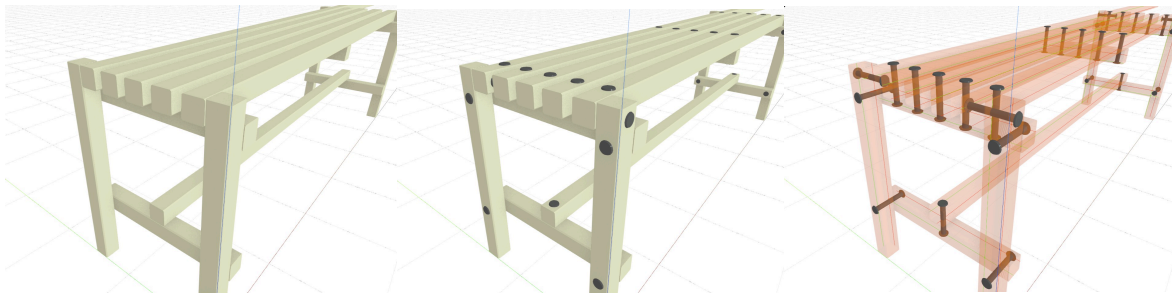
Für die Entwicklung des Verbindungskatalogs wurden die Fähigkeiten der Anwender*innen, die Verfügbarkeit von Baumaterialien (wie Holzlatten und Verbindungselemente), sowie die für den Aufbau benötigten Werkzeuge, die wichtigsten Fixpunkte. Um diese Abhängigkeiten darzustellen und die gesuchten Überschneidungen abzubilden wurde eine Feasibility Map erstellt. Durch Testreihen und deren Validierung wurde am Ende ein einfacher, aber sehr stabiler Holzknotten als Grundlage des Verbindungskatalog gewählt, welcher mit ästhetisch anspruchsvollen Flachkopfmuttern und Gewindebolzen hochfest und einfach montiert werden kann. Dieser Holzknotten ist auch als kartesischer Knoten bekannt. Siehe folgende Abbildung.



5.

Browser-Based-Softwareprototyp

Einer der größten Arbeitspakete stellte die iterative Entwicklung des auf JavaScript basierten Webtools dar. Diese Entwicklung erstreckte sich über die komplette Projektzeit. Dabei wurden alle anderen Entwicklungsergebnisse in der Software kondensiert und fortlaufend integriert. Die relevantesten Einzelergebnisse sind hierbei, die Entwicklung einer OPENJOYN-Skriptsprache mit Funktionen und Klassen für die zielführende Programmierung von parametrischen Konstruktionen. Die Entwicklung einer adäquaten, interaktiven 3D-Darstellung der Konstruktionen, die live gerendert wird und Verbindungen und Verbindungselemente darstellen kann.



Die Implementierung von UI-Funktionen, wie Slider, Buttons, Textfields und Dropdowns für die Generierung der UI-Elemente. Die Integration einer Vielzahl von Features, die Anwender*innen in der Nutzung unterstützten. Beispiele hierfür sind: Die Details und Beschreibungen der UI Elemente und Konstruktionen, die Unterteilung in Baugruppen und die Beschreibung des individuellen Aufbaus.

Latten Maße	34 mm	Breite	1200 mm
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Höhe	400 mm	Tiefe	300 mm
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Höhe der Verstrebung	120 mm	Lücke der Sitzlatten	11 mm
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Hier kann der Entwurf individualisiert werden! Nutze dazu die Slider und Buttons des User-Interface. Achte darauf, zu Beginn das Lattenmaß genau einzustellen. Gib anschließend deiner Konstruktion einen eindeutigen Namen und wähle die Länge deiner Latten aus. Um die Einkaufs- und Aufbaupläne deiner Konstruktion zu generieren klicke auf **Pläne erstellen** und scrolle runter. Ganz am Ende gibt es auch die Möglichkeit die Pläne auf A4 Querformat auszudrucken.

Name meiner Konstruktion	Länge der Latten	3000 mm
<input type="text"/>	<input type="text"/>	<input type="text"/>
Pläne erstellen		

6.

Validierung: Useability, Parametrisierungsstrategie des Verbindungskatalogs

Die entwickelten JavaScript Funktionen ermöglichten eine sehr individuelle Parametrisierung von Konstruktionen. Der Verbindungskatalog, auf Basis des kartesischen Knotens, erwies sich als ideale Balance zwischen der Minimalisierung des Systems und der sich daraus ergebenden Möglichkeiten für die Konstruktionen.

7.

Anwendung des Verbindungskatalogs auf weitere Konstruktionen

In diesem Arbeitspaket wurde der Verbindungskatalog auf einen einfachen Couch-Tisch, eine Bank und eine Hocker angewandt und das entwickelte System erfolgreich validiert.



Bauen!



Bauen!

8.

Test mit Anwendern

Wie auch in den oben aufgelisteten Schritten, waren kurze Tests durch Testnutzer*innen oft Teil der Methode, um unserer Hypothesen und die darauf folgende Entwicklung zu überprüfen. Zum Ende der Entwicklung gab es dadurch nicht viele neue Erkenntnisse, als der gesamte OPENJOYN-Prozess von Anfang bis Ende mit Testnutzer*innen durchlaufen wurde. Die unterschiedlichen Fähigkeiten und Erfahrungslevel der Nutzer*innen nahm wie erwartet den größten Einfluss auf die Ergebnisse. Auch die Hilfsbereitschaft der Baumarktmitarbeiter*innen am Zuschnitt variierte und beeinflusste die User Experience.

9.

Release von OpenJoyn

Das Software-Tool wurde auf einer Website gehostet, damit es von Bürger*innen genutzt werden kann: <https://openjoyn.milz.studio>

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Zielgruppe von OPENJOYN sind alle Bürger*innen und Gruppen wie gemeinnützige Vereine, welche ein Bedürfnis nach individualisierten und oder kostengünstigen Konstruktionslösungen haben. Da die Lösungen in Form von Möbeln und anderen Holzkonstruktionen parametrisch individualisierbar sind und der Gesamtprozess der Nutzung stark intuitiv ist, werden auch

unerfahrene Laien befähigt diese Konstruktionen selbständig herzustellen. Dies gilt gerade für Personen die bisher zur Maker-Community und dem Holzhandwerk wenig Kontakt hatten.

Wir planen das Projekt weiterzuentwickeln und auszubauen. Durch den OpenSource-Charakter gab es schon in der Projektphase vermehrt produktiven Kontakt zu Mitwirkenden und zu Interessierten, welche die Projektidee aktiv unterstützen wollen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Bürger*innen können sich unter folgenden Links detailliert über die Projektergebnisse informieren und das OPENJOYN Tool nutzen:

<https://openjoyn.milz.studio>

<https://github.com/zlimcode>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

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

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

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

Richtlinie zum „Software-Sprint“

OLT – Open Legal Tech

Schlussbericht

Zuwendungsempfänger:

Magdalena Noffke

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S23 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)?

Kommunikation von Behörden mit Bürger*innen orientiert sich regelmäßig an Erfordernissen der Verwaltungen, nicht aber an den Bürger*innen selbst. Dies führt dazu, dass beispielsweise Bescheide von Arbeitsämtern oder Ordnungsämtern für die Betroffenen häufig unverständlich und ohne Jura-Studium kaum zu beantworten sind.

Motivation hinter dem Projekt ist es daher, einfache Rechtsberatungsprozesse automatisierbar zu machen und online anzubieten. Im kommerziellen Bereich gibt es bereits verschiedene Online-Anwendungen, die automatisiert rechtliche Dokumente für Betroffene erstellen. Beispiele sind automatisiert erstellte Einsprüche gegen Strafzettel für zu schnelles Fahren oder Entschädigungsforderungen gegenüber Fluggesellschaften, wenn der Flug ausgefallen ist. Die Rechtsberatung dieser einfachen Fälle erfolgt hier mithilfe eines Online-Fragebogens.

Im Rahmen des Projekts sollte nun eine entsprechende Open-Source-Software entwickelt werden, die vor allem gemeinnützigen Organisationen zugute kommt.

Geplant war die Softwareentwicklung in drei Meilensteinen. Die ersten beiden Monate der Projektlaufzeit sollten dafür genutzt werden, gemeinsam mit den fachlichen Partnerorganisationen, den ersten Anwendungsfall zu spezifizieren. Auch sollte Aufbau und Struktur der Software geplant und entschieden werden wie die verschiedenen User Interfaces gestaltet werden können.

Die nächsten drei Monate waren für Entwicklung des Prototyps eingeplant. Ziel war, bis dahin, den im Meilenstein zuvor definierten Anwendungsfall umzusetzen. Die Entwicklung sollte in einem iterativen Prozess passieren. Der verbleibende Monat war für Dokumentation eingeplant.

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?

Zielgruppe sind Initiativen, die für ihre gemeinnützigen Angebote technische Unterstützung suchen. Dazu gehören beispielsweise ehrenamtliche Beratungen von Geflüchteten, Sozialberatungen und Verbraucherzentralen. Open Legal Tech kann ihnen vor allem dann weiterhelfen, wenn Betroffene in den jeweiligen Bereichen oft gleichlautende standardisierte Briefe erhalten, auf die regelmäßig ähnliche Antworten erfolgen sollen.

Für die Initiativen ergibt sich der Vorteil, dass sie wiederkehrende gleichlautende Anfragen nicht mehr selber sondern z. T. automatisiert über das System beantworten können. Sie werden dadurch entlastet und können sich auf die komplizierten Fragen konzentrieren. Darüber hinaus können sie die Fragebögen auch in verschiedenen Sprachen anlegen und so Sprachbarrieren leichter überwinden.

Für Endnutzer*innen des Systems – also Menschen, die Rechtsberatung in Anspruch nehmen wollen – kann das System einen leichteren, anonymen Einstieg in rechtlich oft komplizierte Themen bieten. Im besten Fall kann es dazu beitragen, Hemmschwellen in der Auseinandersetzung mit Behörden abzubauen. Das System kann auch so konfiguriert werden, dass der Fragebogen als Einstieg für eine darauffolgende persönliche Rechtsberatung genutzt werden kann. Aber auch das automatische Generieren fertiger rechtlicher Dokumente auf Basis der Eingaben von Nutzer*innen ist möglich.

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?

Alle der geplanten Meilensteine konnten umgesetzt werden. Den geplanten iterativen Prozess mit Wireframes und Usertests konnte ich als Einzelperson allerdings aufgrund von Zeit- und sonstigen Ressourcengründen nicht in dem Umfang wie geplant umsetzen. In zukünftigen Projekten werde ich mich daher darum bemühen, entsprechende Expertise mehr einzubinden. Das von der Open Knowledge Foundation organisierte Coaching war sehr gut, kam allerdings etwas zu früh für mich, ich hätte eher ein Coaching zu bereits implementierten Interfaces gebraucht.

Der Ansatz, vor Beginn der Implementierung, Interviews mit potenziellen Nutzer*innen zu führen, war sehr hilfreich. Dadurch habe ich ein gutes Verständnis für potentielle Use Cases gewonnen. Auch einen der Use Cases als ersten praktischen Anwendungsfall auszuwählen, den die Software abbilden soll, hat mir sehr geholfen, den Entwicklungsprozess zu strukturieren und zu organisieren.

Am Ende der Projektlaufzeit ist nun eine erste lauffähige Software entstanden, mit der automatisierte Online-Fragebögen zur Rechtsberatung erstellt werden können. Zudem ist es möglich auch Dokumentvorlagen mit Platzhaltern für bestimmte Antworten der User*innen anzulegen. Auch können Textblöcke in den Dokumentvorlagen bei bestimmten Antworten ausgeblendet werden.

Die Software ist als App einbindbar in ein bestehendes Django-Softwareprojekt kann aber auch unabhängig von einem bestehendem Projekt aufgesetzt werden.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung? Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Gemeinnützige Organisationen, die Rechtsberatung anbieten, können die Software kostenlos in ihre bestehende IT-Infrastruktur einbinden und eigenständig Beratungsprozesse gestalten. Da es sich um Open-Source-Software handelt, können Organisationen die Software auch weiterentwickeln oder auf eigene Bedürfnisse anpassen. In der Zukunft wäre es auch denkbar, die Software als SaaS-Plattform aufzusetzen. Dann könnten die Organisationen ihre Prozesse, Entscheidungsbäume und Dokumentvorlagen auf der Plattform anlegen, ohne das System selber aufsetzen und warten zu müssen. Die eigentlichen Fragebögen könnten sie dann z.B. per Iframe auf Ihrer eigenen Webseite einbinden.

Zudem sind auch andere Anwendungsgebiete vorstellbar. Bisher wurde das System nur für den Bereich Rechtsberatung konzipiert, es lässt sich aber in allen Bereichen einsetzen, in denen Entscheidungsbäume durchlaufen werden müssen. Vorstellbar wäre z.B. der Einsatz anstelle von komplizierten FAQs oder als Ausfüllhilfe für Formulare.

Mögliche Weiterentwicklungen wären die Implementierung von Multi-language-support für das Anlegen der Fragebögen und Dokumentvorlagen zu ermöglichen sowie das Auswerten hochgeladener Dokumente. Hier kann auf bereits bestehende Open-Source-Projekt zurückgegriffen 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?

Zu Beginn war geplant, hochgeladene Behördenbriefe auszuwerten. Da das für den ersten Anwendungsfall nicht nötig war, aber stattdessen das Anlegen der Dokumentvorlagen sich als deutlich komplexer erwiesen hat als zu Beginn gedacht, habe ich mich darauf konzentriert und das Auswerten von hochgeladenen Behördenbriefen bisher nicht umgesetzt.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Website: <https://openlegaltech.github.io/>

Github: <https://github.com/OpenLegalTech/django-legal-advice-builder>

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?

Der Arbeits- und Kostenplan wurde eingehalten.

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 keine nennenswerten Entwicklungen anderer Personen oder Institutionen, die Einfluss auf meine Arbeit und die Zielsetzung hatten.

Richtlinie zum „Software-Sprint“

OSD – OpenSpaceData demokratisiert Satellitendaten

Schlussbericht

Zuwendungsempfänger:

Niklas Jordan

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S24 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Pro Tag liefern allein die 12 Copernicus-Satelliten der ESA eine Datenmenge von 250 TB. Das Problem: Oft braucht es technisches und fachliches Wissen, um Daten abzurufen und zu nutzen. So benötigt es beispielsweise die Benutzung von Application Programming Interfaces (APIs), für die der Nutzer eine relevante Programmiersprache, wie Python, erlernen muss oder er muss wissen was es mit fachlichen Termini, wie Satellitenbänder oder spezifische Indizes, auf sich hat. So bleibt die Nutzung der Daten vor allem Fachleuten, Wissenschaftlern und Software-Entwicklern vorbehalten. Dabei bieten die Daten gerade auch für fachfremde Menschen ein riesiges Potenzial: So können NGOs oder Journalisten drohende Umweltkatastrophen schnell erkennen. Gleiches gilt für humanitäre Organisationen, die nach Katastrophen wie der Explosion in Beirut, handeln wollen.

Deshalb haben wir uns dazu entschieden eine Applikation zu entwickeln die Erstnutzern von Satellitendaten eine einfache Benutzeroberfläche zu Verfügung stellt und keinerlei fachliche oder technische Voraussetzung mit sich bringt. OpenSpaceData soll einen einfachen Zugang zu diesen gesellschaftsrelevanten Informationen ermöglichen.

Zielstellung ist dabei, dass man nach der Nutzung unserer Applikation das notwendige Fachwissen bzgl. Termini und Methodik erlangt hat um mit professioneller Software weiterzuarbeiten. Deshalb legen wir neben dem einfachen Interface auch großen Wert auf die Anleitung zur Verarbeitung der Daten.

In der anfänglichen Planung haben wir die im Folgenden aufgeführten Meilensteine definiert.

Meilenstein 1: Architektur

- Planung einer geeigneten Informationsarchitektur (mit nötigen Relationen, Abhängigkeiten etc.)

- Aufsetzen der Datenbankstruktur mit einem geeignetem Backend
- Anbindung der Schnittstelle der ESA

Meilenstein 2: Backend

- Such-Algorithmus entwickeln, der den Nutzenden auf Basis seiner Anforderungen die passenden Daten bei der ESA sucht und ausgibt

Meilenstein 3: Interface und Content

- Gestaltung und Entwicklung der Benutzeroberfläche
- Entwicklung der Anleitungen (Guides) und Inhalte, die es den Nutzenden erleichtern die Daten zu verarbeiten und grundlegende Methodiken der Satellitendatennutzung erläutern

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

In Deutschland besteht ein Recht auf Informationen. Öffentliche Institutionen sind, mit Einschränkungen, dazu verpflichtet Informationen, bspw. in Form von offenen Daten, herauszugeben. Offene Daten bieten Transparenz und ermöglichen die Chance zur Teilhabe an gesellschaftlichen Prozessen und Innovationen. Deshalb gilt: Daten müssen überhaupt erstmal verfügbar sein und vor allem müssen sie nutzbar sein – für alle.

Das World Economic Forum hat in seinem Global Risk Report 2021 die wachsende digitale Ungleichheit auf Platz 7 von insgesamt 35 Risiken gesetzt. Laut dem Report fördert das nicht nur die Schere zwischen Arm und Reich, sondern befeuert auch Missinformation. Heutzutage ist der Zugang zu Daten so groß wie noch nie in der Geschichte. Aber oftmals können diesen vor allem Privilegierte nutzen, also Menschen, die wissen, wie sie mit modernen Technologien arbeiten, wie sie Daten zu interpretieren haben und welchen Nutzen – aber auch welche Risiken – diese mit sich bringen.

Satellitendaten bieten unglaubliche Möglichkeiten und einen unerschöpflichen Wissensschatz über unsere Umwelt und deren Bewohner*innen. So kann man beispielsweise mit den Sentinel-Satelliten der Europäischen Weltraumagenturen Vegetationsveränderungen von Wäldern oder auch Agrarflächen beobachten. Was erstmal banal klingt, eröffnet uns unglaubliche Möglichkeiten. Diese Informationen bieten ein riesiges Archiv über das Ernteverhalten von Bauern und Bäuerinnen: Wie stark hat sich aufgrund von Witterung und klimatischen Veränderungen zum Beispiel der Erntezeitpunkt verschoben? Wir können aber auch urbane Entwicklungen beobachten oder sehen, an welchen Grenzen Militärbasen expandieren oder gar neue entstehen. All diese Informationen stehen nicht nur dem Militär oder einigen Wissenschaftler:innen zur Verfügung sondern allen.

Mit OpenSpaceData wollen wir dieses Problem lösen und die Welt der offenen Satellitendaten demokratisieren, indem wir einen Zugang bereitstellen, die es jede:m ermöglicht auf diese Daten zuzugreifen und sie zu verarbeiten – unabhängig vom fachlichen, technischen oder finanziellen Background.

Ausführliche Darstellung der Ergebnisse

Zentrales Ergebnis ist die Bereitstellung der Software unter MIT-Lizenz bei GitHub:

- <https://github.com/OpenSpaceData>

Die geplanten Meilensteine wurden erreicht und die Entwicklungsphasen abgeschlossen. Alle angestrebten Funktionen konnten umgesetzt werden.

Wie funktioniert OpenSpaceData im Detail?

OSD setzt keinerlei Programmierfähigkeiten oder Vorkenntnisse zu Erdbeobachtungsdaten voraus. Sprich, das Interface überfällt weder mit Fachbegriffen, noch benötigt es komplexe Eingaben. Lediglich drei Angaben sind notwendig:

1. Was möchte ich analysieren? Hier kann der Nutzende aus verschiedenen, vordefinierten Use Cases wählen. Beispielsweise ob die Gesundheit der Vegetation ermittelt werden soll oder auch die Auswirkungen der Dürre auf einen bestimmten See.
2. Für welchen Ort bzw. welches Gebiet interessiere ich mich? Der Nutzende hat die Möglichkeit, ein bestimmtes Gebiet über eine Sucheingabe, ähnlich wie bei bekannten Mapping-Diensten wie bspw. Google Maps, oder ein Gebiet auf der Karte auszuwählen.
3. Welcher Zeitraum soll analysiert werden? Die Erde verändert sich ständig. Deshalb müssen Veränderungen auf der Erdoberfläche immer in einem zeitlichen Kontext gesehen werden. Nutzer*innen können deshalb angeben, ob sie die aktuellsten Daten möchten oder die Daten aus einem bestimmten Zeitraum, beispielsweise aus einem bestimmten Monat eines bestimmten Jahres.

Unser Algorithmus sorgt nun dafür, dass Nutzer:innen das bestmögliche Satellitenbild bekommen und stellt ihnen eine Anleitung dafür bereit, wie sie die Daten verwenden, analysieren und anschließend die richtigen Erkenntnisse daraus ableiten können.

Die Begleitung der Open Knowledge Foundation und vor allem das Coaching durch Simply Secure hat sehr geholfen das Interface maximal Nutzerfreundlich zu entwickeln und die Bedürfnisse der Endnutzer durch Befragungen zu ermitteln.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Für die Entwicklung haben wir uns im wesentlichen auf folgende Zielgruppen konzentriert:

- **Journalist:innen** haben mit OSD ein neues Tool an der Hand, welches bei der Recherche zu bspw. Umweltthemen unterstützt.
- **Lehrer:innen** können mit Schülern und/oder Studenten spielend leicht mit Erdbeobachtungsdaten arbeiten und so ihren Unterricht interaktiver gestalten.
- **NGOs und Aktivist:innen** können nun auf Daten zugreifen, die ihnen aufgrund fachlicher oder technischer Expertise nicht zur Verfügung standen.

Es gibt bereits viele Ideen sowohl das Interface noch leichter bedienbar zu machen, als auch weitere Daten zugänglich zu machen. Aktuell basiert das Interface auf der Suche von Sentinel-2 Daten. Wir

streben an auch Daten der Sentinel-1 und Sentinel-3 Satelliten über OpenSpaceData zugänglich zu machen. Ausserdem könnten weitere Satelliten anderer Weltraumorganisationen, wie der NASA oder auch EUMETSAT, eingebunden werden.

Ein weiteres Hauptaugenmerk liegt in der Zukunft auf die Weiterentwicklung der Guides und neuer Use Cases. So sind wir aktuell mit verschiedenen humanitären Organisationen im Gespräch spezielle Anwendungsfälle für deren Bereich abzudecken.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die ursprüngliche Idee war die ESA-eigene OpenSearch API zu nutzen um die Satellitendaten zu filtern und die Produkte zu downloaden. Allerdings ist die Nutzung der API auf 2 parallele Downloads limitiert. Das heißt, würde unsere Software von 3 Menschen gleichzeitig genutzt werden, würde die Schnittstelle blockieren. Deshalb haben wir nach Alternativen gesucht und uns letztlich für die Sentinel-2 Open Data Registry auf AWS entschieden. Diese wird von Sinergise betrieben, einem ESA-nahen Startup. Ein weiterer Vorteil dieser Lösung war, dass wir die Downloads für den Nutzer als GeoTiff bereitstellen konnten. Dies ist ein Dateiformat, welches sich mit den meisten Bildprogrammen öffnen lässt. Dies reduziert nochmals die Hürde bei der Nutzung der Daten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

- Website: <https://www.openspacedata.org/>
- Fertige Applikation (Beta): <https://openspacedata.netlify.app/>
- GitHub: <https://github.com/OpenSpaceData>
- Twitter: <https://twitter.com/openspacedata1>
- Veröffentlichungen:
 - Interview: [OpenSpaceData will Satellitendaten demokratisieren](#)
 - Fachartikel: [Warum Daten nicht nur offen sondern auch nutzbar sein müssen](#)
 - Vortrag auf re:publica: [Satelliten im Umwelt- und Klimaschutz: Sternstunden und Schattenseiten](#)
 - Vortrag auf der Machine Learning Week Europe: [Climate Expert Round: Space Data for Earth Observation](#)
 - Podcast: [Satelliten im Umwelt- und Klimaschutz](#)
 - Fachartikel: [Astronautenblick für alle: Wie offen sind die Erdbeobachtungsdaten der ESA?](#)

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ein Faktor den wir bei der ursprünglichen Planung nicht berücksichtigt hatten, ist die Kommunikation und Öffentlichkeitsarbeit, die das Projekt begleitet haben. Steigendes Interesse für das Projekt und

das generelle Thema der Erdbeobachtung haben dafür gesorgt, dass wir einige male gefragt wurden Gastartikel zu schreiben, Interviews zu geben oder Vorträge zu dem Projekt halten. Diese Aktivitäten waren anfangs nicht eingeplant und führten dazu, dass wir Prioritäten im Projekt neu setzen mussten.

Ein weiterer Aspekt der Kommunikation waren die Gespräche mit Interessenten zur Beteiligung an dem Projekt. Wir haben viele Gespräche mit Personen geführt, die sich an dem Projekt nachhaltig beteiligen wollten, mussten die Personen "onboarden" und anleiten. Dies hat ebenfalls zu einer weiteren zeitlichen Verknappung geführt, die in der anfänglichen Planung nicht vorgesehen waren.

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“

NCPlenum – Hilfe bei digitalen Meetings

Schlussbericht

Zuwendungsempfänger:

Meurer Wiehle GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S25 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Für gemeinsames Organisieren sind Treffen essentiell. In tech-affinen Gruppen werden diese schon länger digital unterstützt. Digitale Tools gewannen, besonders während der Corona-Pandemie, auch für Community-Projekte und soziale Bewegungen an Bedeutung. Treffen gut zu strukturieren, ist schon in persona eine Herausforderung, wie zahlreiche Leitfäden belegen.

Wir wollen es Gruppen ermöglichen, ihre digitalen Treffen besser zu organisieren. Unsere Erweiterung der Nextcloud Collectives App ermöglicht selbstorganisierten Gruppen, ihre Treffen in Nextcloud besser zu organisieren und zu dokumentieren.

Zu Beginn hatten wir uns folgende Meilensteine gesetzt:

1. Protokoll-Vorlagen in der Collectives App
2. Geschwindigkeits-Verbesserungen bei kollaborativer Synchronisation in der Text App
3. Best-Practice Dokumentation für Gruppen-Treffen in Nextcloud
4. Unterstützung von Technik-Kollektiven bei der Bereitstellung der durch uns entwickelten Lösungen
5. Unterordner in der Collectives App
6. Kollaborative Dokumenten-Übersicht in der Collectives App
7. Autoren-Cursor in der Text App
8. "Table of Contents" in der Text App

9. Teilen von Protokoll-Vorlagen unter verschiedenen Gruppen in Nextcloud

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

Mit einem einfachen, verständlichen Tool wollen wir vor allem Gruppen unterstützen, deren Hauptbetätigungsfeld nicht im digitalen Bereich liegt. Da Nextcloud mittlerweile zur Organisierung unter politischen Gruppen und Community-Projekten weit verbreitet ist, profitieren diese direkt davon, die Nextcloud Collectives App entsprechend der beschriebenen Ziele weiterzuentwickeln.

Verbesserungen an internen Apps (Text, Talk, Circles) werden durch die Installation einer neuen Version von Nextcloud automatisch verfügbar.

Wir haben uns auf den Förderungs-Schwerpunkt *Civil Tech* beworben, da unsere Software-Entwicklungen zivilgesellschaftliche Initiativen unterstützen wollen.

Ausführliche Darstellung der Ergebnisse

Wir konnten die Nextcloud Collectives App im Rahmen der Förderung deutlich weiter entwickeln und wichtige Funktionen für Online-Treffen einbauen. Auch zu anderen Nextcloud-Komponenten konnten wir Verbesserungen beitragen.

Die Collectives App haben wir um Unterseiten erweitert - etwa zur Abbildung von AGs - oder um alle Protokolle an einem Ort zu sammeln. Wir haben die App wirklich kollaborativ gemacht, indem neue und umbenannte Seiten automatisch sofort allen Teilnehmenden angezeigt werden. Auch die Live-Verfolgung von Änderungen im Dokument wurde verbessert. Gemeinsames Arbeiten in einem Kollektiv soll Spaß machen und flüssig von der Hand gehen.

Unterseiten ermöglichen ein besseres Strukturieren des gemeinsamen Wissens. Änderungen anderer werden gleich mit angezeigt.

Das Verlinken von Seiten per Drag&Drop ermöglicht schnelle und unkomplizierte Querverweise. So wollen wir es erleichtern, gemeinsam gewonnenes Wissen besser zu strukturieren und Verbindungslinien zu ziehen.

Bereits bei analogen Treffen haben sich Protokollvorlagen als hilfreich erwiesen, eine gemeinsam erarbeitete Struktur beizubehalten. In Nextcloud Collectives ermöglichen Protokollvorlagen, digitale Treffen besser zu strukturieren, diese Struktur zu reflektieren und weiterzuentwickeln.

Mit Backlinks - also eine Liste der Seiten, die auf die Themenseite verweisen - kann nachvollzogen werden, bei welchen Treffen über ein bestimmtes Thema gesprochen wurde.

Insgesamt schaffen wir damit eine Umgebung, welche im gesamten Prozess (Vorbereitung, Durchführung und Nachbereitung/Dokumentation) eines Treffens genutzt werden kann.

Zusätzliche wird das Bearbeiten von Seiten und deren Inhalten durch zahlreiche von uns eingereichte kleine und große Verbesserungen in der Nextcloud Text App deutlich angenehmer. Die Dialoge zum Verlinken anderer Seiten wurden verbessert und Emojis können nun über einen Picker ausgewählt und in den Text eingefügt werden.

Von den eingangs beschriebenen Meilensteinen konnten wir bis einschließlich Meilenstein 7 alle erfüllen. Lediglich zur Implementierung einer „Table of Contents“ Funktion (Meilenstein 8) sowie der Möglichkeit, Vorlagen unter verschiedenen Gruppen zu teilen (Meilenstein 9) kamen wir nicht mehr.

Eine wichtige Erkenntnis aus dieser Förderrunde ist, dass wir nur so weit kommen konnten, da wir mit der Entwicklung nicht bei null anfangen mussten. Aufbauend auf Nextcloud allgemein und der Collectives App konkret konnten wir uns auf die Feinheiten der beschriebenen Funktionen und Verbesserungen fokussieren. Von daher fragen wir uns, ob der Ausschluss von Folgeförderungen im Rahmen des Prototypenfunds nicht ein Hindernis bei der Unterstützung zivilgesellschaftlicher Software-Projekte darstellt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Vorteile für unsere Zielgruppe haben wir bereits oben erläutert. Dass sowohl Nextcloud als auch die Collectives App als Open Source Software entwickelt und veröffentlicht wird, erlaubte uns erst, beides als Grundlage für unsere Weiterentwicklungen zu nehmen. Dadurch, dass wir die Ergebnisse ebenfalls als Open Source veröffentlichen, können alle Benutzer:innen von Nextcloud von ihnen profitieren. Wir halten es zudem für essentiell, dass Software für die Zivilgesellschaft von jeder und jedem genutzt, verändert, nachvollzogen und studiert werden kann.

Wir werden die Nextcloud Collectives App auch in Zukunft weiter pflegen und verbessern.

Persönlich konnten wir unsere Fähigkeiten bei der Entwicklung von Webapplikationen weiterentwickeln und den Blick auf User-zentrierte Entwicklung schärfen – sprich Software so zu designen, dass sie für die Zielgruppe möglichst zugänglich ist.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Bei einigen technischen Design-Entscheidungen sehen wir bereits Verbesserungspotential und werden evaluieren, ob wir sie überdenken müssen. Ein Beispiel:

Bei der Implementierung der Möglichkeit, Unterseiten in der Collectives-App anlegen zu können, folgten wir dem Nextcloud-Konzept, dass eine Datei mit dem Namen `Readme.md` immer als Übersicht des Ordner-Inhalts dient. Diese Datei wird in der Nextcloud Files App als Richtext-Workspace oberhalb der Ordner-Inhalte angezeigt.

Die Entscheidung, `Readme.md` als Index-Seite für einen Ordner zu verwenden, erwies sich im Rahmen der Collectives App allerdings als verwirrend für die User, wie User-Feedback zeigte. Wir überlegen nun, zu einer Struktur mit `Subpage.md` im Überordner als Index für die Inhalte im Unterordner `Subpage` zu wechseln.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Unsere Ergebnisse sind auf der Projektseite <https://gitlab.com/collectivecloud/collectives> einzusehen.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die ursprüngliche Kalkulation der vorgesehenen Zeiträume für einzelne Meilensteine passte im groben überraschend gut. Die ursprünglich veranschlagten Stunden konnten wir einhalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Der regelmäßige Austausch mit den Entwickler:innen bei Nextcloud war essentiell für viele Prozesse in unserem Projekt. In einigen Fällen haben wir von Verbesserungen bei Nextcloud selbst profitiert (etwa die Entwicklung von Autor:innen-Farben in der Text App, Meilenstein 7). Wir freuen uns, im Rahmen des Projekts noch weiter in die Nextcloud-Community herein gewachsen zu sein und werden auch in Zukunft eng mit ihr zusammenarbeiten.

Richtlinie zum „Software-Sprint“

Akronym – Projekttitel

Schlussbericht

Zuwendungsempfänger:

Basile Simon

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS17S26** 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)?

Ziel des Toolkits war es, eine einfache und kostengünstige Archivierungslösung für Einzelne oder kleine Forscherteams anzubieten. Es zielte darauf ab, sich auf die folgenden zentralen Probleme zu konzentrieren:

- Digitales Material zu archivieren ist schwierig und erfordert langfristiges Denken,
- Material (z.B. Photos, Videos, Text) verschwindet aus dem Internet und muss sachgerecht aufbewahrt werden,
- Dies mit dem Ziel zu tun, rechtlichen Standards zu genügen, ist sehr schwer.

Hierzu habe ich an drei Bereichen gearbeitet:

1. Ich habe Vertreter von NGOs und Anwaltskanzleien konsultiert und interviewt. Hierbei habe ich herausgefunden, dass das zu entwickelnde Werkzeug auf dem Webbrowser implementiert werden sollte, da er am besten zu ihren Arbeitsabläufen passt.
2. Ich habe die Proof-of-Concept-Software (einen funktionierenden Prototyp) entwickelt, die einfache Verschlüsselung, Cloud-Speicherlösungen und Authentizitätsdemonstration kombiniert. Dieser Teil kann als "Backend" bezeichnet werden und stellt den Kern meiner Arbeit dar.
3. Um das Produkt für die Zielgruppe nutzbar zu machen, habe ich eine Benutzeroberfläche implementiert, die so wie sie ist von einem kleinen Team sofort verwendet kann und damit die volle

Funktion des Toolkit demonstriert. Das Backend ist jedoch so konstruiert, dass eine beliebige andere Benutzeroberfläche darauf aufbauen kann.

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

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

Das Projekt trägt in beiderlei Hinsicht zu den Zielen der Förderinitiative bei.

Die Zielgruppe des Toolkits sind kleine Forscherteams, die ein Phänomen (zum Beispiel Menschenrechtsverletzungen) dokumentieren. Diese Forschung zielt auf die Interaktion mit der Rechtswelt, den Gerichten – kurz der Justiz. Das Digital Evidence Toolkit ist ein Werkzeug Individuen, die sich durch Sammlung von rechtlich anerkanntem Beweismaterial in die Zivilgesellschaft einbringen möchten.

Zum Thema “Data Literacy”: Die gesetzlichen Anforderungen an die Führung einer Beweiskette sind hoch und der damit konforme Umgang mit digitalen Beweismitteln schwierig. Das Toolkit schlägt eine Lösung dafür vor.

Neben der Software-Seite des Projekts habe ich im Anschluss an Interviews mit Praktikern eine Newsletter-Kampagne durchgeführt, die darauf abzielte Personen aus der Zielgruppe des Toolkits mit dem Problem zu konfrontieren und “data literate” 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?

In nur sechs Monaten wurde das Toolkit zu einer funktionalen Software mit den folgenden Funktionen:

- Das Frontend kann Informationen direkt vom Browser (der Benutzeroberfläche) an das Backend senden, welche dort verarbeitet werden,
- Das Backend kann diese Information gemäß den geltenden gesetzlichen Richtlinien speichern,
- Das Backend kann diese Information gemäß den führenden internationalen Spezifikationen mit einem Zeitstempel versehen,
- Es kann dann den Inhalt über eine Webschnittstelle anzeigen, sodass es bearbeitet und mit Anmerkungen versehen werden kann.

Neben den technischen Features hat das Projekt durch Öffentlichkeitsarbeit (Newsletter und Benutzerinterviews) zahlreiche Fachleute, Praktiker und Experten erreicht.

Das Feedback bestätigt, dass das Toolkit eine neue Methode der digitalen Beweiserhaltung aus dem Web implementiert.

Das Tool zog in weniger als einem halben Jahr Beiträge aus der Community an, welche zu neuen Partnerschaften führten. Diese Partnerschaften stehen in direktem Wettbewerb mit den anderen etablierten Spielern der Branche (Syrian Archive, Carnegie Mellon University, LexisNexis), wodurch es diesem Projekt gelungen ist, sich von anderen Entwicklungen zu differenzieren.

Neben der finanziellen Förderung, half die OKFn, durch wichtige persönliche Coaching-Sitzungen zusätzliche Märkte zu identifizieren. Neben diesem Marktansatz habe ich ein Produktdenken entwickelt und nährte den Wunsch, aus diesem Projekt ein nachhaltige Geschäftsidee zu entwickeln.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts? Welche weitergehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse? Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung? Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Der Hauptvorteil für meine Zielgruppe ist die Verfügbarkeit eines kostenlosen, qualitativ hochwertigen Tools zur digitalen Beweiserhaltung. Das Wachstum meines Newsletters zeigt auch die Sensibilisierung für das Problem bei kleinen NGOs.

Da das Toolkit eine Open-Source-Software ist, profitiert die Zielgruppe direkt von:

- der kostengünstigen Betreibung der Tools,
- der Transparenz und Vertrauenswürdigkeit,
- der einfachen Integration und Ausbaufähigkeit

All diese Kriterien sind zentral für die Wettbewerbsfähigkeit des Toolkits auf dem Markt, wie Interviews und potenzielle Partnerschaften gezeigt haben.

Es gibt weitere Entwicklungspläne (die von der Finanzierung/dem Geldeingang abhängig sind).

Das Tool wurde flexibel und modular aufgebaut, um sich an unterschiedliche Anwendungsfälle anpassen zu können, und ich möchte dies demonstrieren, indem ich die Interoperabilität weiterentwickle. Um die zu demonstrieren plane ich die Integration mit anderer Dokumentenverwaltungssoftware (Open Source oder kommerziell).

Sechs Monate allein zu arbeiten war ein echtes Privileg und eine echte Herausforderung. Der Rahmen der regelmäßigen Berichterstattung an den Prototype Fund und des professionellen Coachings hat bei der Durchsetzung des Projekts jedoch sehr geholfen.

Ich habe gelernt mein eigenes Projekt von der Idee bis zur Umsetzung und Kommunikation zu managen und an potenzielle Marktchancen anzupassen.

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?

Mehrere technische Lösungen wurden nicht weiter verfolgt:

- Ursprünglich wurde das Projekt aufgrund der unveränderlichen Datenstrukturen um die Programmiersprache Clojure herum konzipiert. Schließlich habe ich mich für eine andere Programmiersprache entschieden (Node, Typescript) weil es dazu eine grössere Community gäbe und es die Integration mit der Cloud-Infrastruktur erleichtert hat.
- Aus den selben Gründen wurde die Verwendung von Clojure-zentrierten DataScript-Produkten aufgegeben, und durch eine Datenbanklösung von Amazon (QLDB) ersetzt.

Von Seiten des Projektmanagements war ursprünglich die Veröffentlichung eines Berichts über den Stand der Beweissicherung geplant, aber angesichts des zeitlichen Umfangs der Softwareentwicklung war dieser Teil nicht mehr möglich. Anstelle eines großen Berichts habe ich jedoch im Laufe der Förderperiode einen Newsletter aufgebaut, welcher eine Art Dokumentation darstellt und sich als großartige Quelle für nächste Ansprechpartner erwiesen hat.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

- Projekt Website: <https://digitalevidencetoolkit.org/>
- Code repositories: <https://github.com/digitalevidencetoolkit/>
- Dokumentation: <https://digitalevidencetoolkit.notion.site/Getting-started-15521f4125534f4aa758a2575c27ad5c>
- Newsletter: <https://digitalevidencetoolkit.substack.com/>

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?

Von März bis Mai, war das Volumen der Arbeit durch andere Auftraggeber zu groß und das bedeutete, dass ich in dieser Zeit weniger an dem Toolkit arbeiten konnte als geplant bzw. In dieser Periode viele Überstunden gearbeitet habe um meinen Aufgaben gerecht zu werden.

Ab Mai habe ich keine anderen Projekte als Freiberufler angenommen und konnte mich damit voll auf den Toolkit konzentrieren.

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?

Zu Beginn des Projekts wurde deutlich, dass ein ähnliches Projekt des Syrian Archive vor zwei Runden von Prototype finanziert wurde. Es stellte sich die Frage: Was kann ich anders machen? Ist meine Zielgruppe dieselbe?

Weiterhin bin ich über eine passende Amazon-Lösung gestolpert, die anfangs nicht zu meinen technischen Lösungsansätzen zählte aber viele Vorteile brachte. Ich führte Benutzerinterviews, um zu überprüfen, ob dies (wie in: ein Cloud-System) akzeptabel wäre. Dadurch konnte ich mich am Ende auf die Implementierung und den Workflow konzentrieren.

Ich habe auch mehrere Anwaltskanzleien (ECCHR, David Akerson, UN) involviert, um die Idee zu validieren, auf Unternehmenssoftware umzuschwenken und mit diesen zu integrieren.

Richtlinie zum „Software-Sprint“

MIGR-AI-TION

Schlussbericht

Zuwendungsempfänger:

Jie Liang Lin

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S27 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)?

Die Probleme im Technologiesektor lassen sich grob als Datenkolonialismus zusammenfassen und umfassen Themen wie KI-Voreingenommenheit, unsichtbare Arbeit und extraktiven Kapitalismus. Ziel war es, eine Plattform zu schaffen, auf der die Datenerhebung als Mittel zur Verbesserung des Dialogs in der Gemeinschaft, der Datenkompetenz und der Vertretung von Migranten genutzt werden kann. Die Motivation war, KI-Prozesse als Methoden und Werkzeuge für die öffentliche Beteiligung neu zu gestalten, um Menschen am Rande der Gesellschaft eine Stimme in der zukünftigen Medienlandschaft zu geben. Auf der Grundlage von Hintergrundrecherchen zu laufenden ausbeuterischen KI-Praktiken wie biometrischen Experimenten und dem Training von Computer Vision in Flüchtlingsarbeitslagern, unterbezahlter Plattformarbeit wie Mechanical Turk sowie der zunehmenden Überwachung von Migranten und Randgruppen wurde beschlossen, dass das Ergebnis dieses Projekts ein einfacher Bilddatensatz sein sollte, der von Migranten erstellt wurde, um die Öffentlichkeit über ihre Erfahrungen zu informieren und KI-Forscher über Voreingenommenheit aufzuklären - und der in einem späteren Entwicklungsstadium zum Training von Bilderkennungsmodellen verwendet werden könnte.

Die geplanten Meilensteine waren die folgenden:

Phase 0: Sammeln eines Beispieldatensatzes von 500 Bildern

Phase 1. Entwurfsarchitektur (Datenspeicherung, Bildannotation, interaktive Visualisierung)

Phase 2. Aufbau einer Back-End-Datenspeicherung mit Python

Phase 3. Aufbau einer React-Bildergalerie

Phase 4. Software für die Integration/Migration von Bildanmerkungen

Phase 5. Entwurf und Implementierung der Datenvisualisierung in D3

Phase 6. Erstellen/Integrieren interaktiver Komponenten in React/D3

Beitrag des Projektes zu den Zielen der Förderinitiative

„Software-Sprint“

Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts?

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

Die Zielgruppe sind Menschen mit Migrationshintergrund, nicht nur solche mit Flüchtlingsstatus. Ziel war es, die "Flüchtlingsgeschichte", die von den Mainstream-Medien überstrapaziert wird, zu erweitern. Die Entwicklung dieses Projekts würde bedeuten, dass die Standarddefinition der Datenerfassung auf den Kopf gestellt wird, so dass die Teilnehmer und Mitwirkenden tatsächlich davon profitieren können, ihre Erfahrungen zu teilen. In Zusammenarbeit mit zwei Expressive-Arts-Therapeuten wurden die Bildkommentar-Tools neu gestaltet und als therapeutische "Intermodalitäten" integriert (Möglichkeiten, mit denen die Teilnehmer vergangene Erfahrungen reflektieren und umgestalten können, um im Körper gespeicherte Traumata zu lösen). So wurde die Datenerfassung zu einem Weg der Heilung und des Dialogs zwischen verschiedenen Mitgliedern der Migrantengemeinschaft.

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?

Mit React.js und P5.js wurde eine neue und benutzerfreundliche Benutzeroberfläche implementiert, die drei Werkzeuge zur Bildkommentierung enthält, die in Zusammenarbeit mit zwei Expressive Arts Therapeuten entwickelt wurden. Außerdem verfügt die Plattform über eine Suchfunktion nach Bild-Tags, einen Netzwerkgraphen und eine Bildergalerie. Die Datenbank ist fast fertig, aber es gibt noch einiges zu tun, bevor das Projekt online gehen kann. --Darüber hinaus hat das Projekt aufgrund seines interdisziplinären Charakters das Interesse von Menschen mit unterschiedlichem beruflichem Hintergrund und aus verschiedenen Kulturen geweckt. Die Open Knowledge Foundation hat die interdisziplinären und experimentellen Ansätze dieses Projekts unterstützt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

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

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

Die Vorteile des Austauschs von Bildern und des gemeinsamen Sprechens über Migrationserfahrungen als Teil eines intermodalen Ansatzes in der Kunsttherapie und der Gemeinschaftsbildung wurden während eines organisierten Workshops sofort deutlich. Das nächste Annotationswerkzeug wäre für Videos und für die Arbeit mit mehr verkörperten Ausdrucksformen und Tanztherapie.

Die Projektarbeit war insofern sehr vorteilhaft, als sie die Möglichkeit bot, verschiedene Interessenbereiche in einem Projekt zu vereinen und somit ein Publikum aus vielen Berufsgruppen (Künstler, Sozialwissenschaftler, Designer, Journalisten, Juristen, Techniker, Gesundheitsexperten) und mit unterschiedlichem kulturellen Hintergrund zu erreichen, die sich im gemeinsamen Interesse an der Schaffung einer besseren Datenzukunft zusammenschließen können.

Die Open-Source-Effekte der Ergebnisse müssen noch abgewartet werden, aber es wird mehr Möglichkeiten für Entwickler geben, sich zu beteiligen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

Ursprünglich waren drei einzigartige Bildbeschriftungstools für die Migrationsthemen "Heimat", "Solidarität" und "Wohlbefinden" unter Verwendung der P5.js-Bibliothek geplant. Aus Zeitgründen wurde das Tool "Wohlbefinden" zugunsten einer Standard-Bildkommentierung ohne Zeichenwerkzeug (nur Wort-Tags) fallen gelassen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

<https://github.com/jielianglin/m-v2-prod>

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?

Das Projekt stieß bei der Fertigstellung der Datenbank an zeitliche Grenzen. Die Datenbank befindet sich derzeit noch in der Phase der Fehlersuche. Außerdem wurde beschlossen, das Projekt nicht auf Amazon, Azure oder einem der großen Unternehmensserver zu hosten. Glücklicherweise hat sich eine Lösung für dieses Problem gefunden. Bitte lesen Sie den letzten Absatz unten.

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?

Eine Zusammenarbeit mit zwei Expressive Arts Therapeutinnen während des Förderzeitraums hat gezeigt, wie wichtig es ist, Werkzeuge für die Bildkommentierung zu entwickeln, die sich für den emotionalen Ausdruck eignen. Eine Demonstration der Werkzeuge während des Workshops bestätigte dies durch die Teilnehmer. Der experimentelle Charakter des Projekts führte zu vielen positiven Ergebnissen. Die Gruppentherapie unter Verwendung persönlicher Bildarchive und digitaler Werkzeuge für die Bildkommentierung wurde bisher im Bereich der Expressive Arts Therapy noch nicht erprobt, obwohl sie ein zentraler Ansatz für die Gemeinschaftsentwicklung in der angewandten visuellen Anthropologie ist. Es besteht also ein immenses Potenzial, eine interdisziplinäre Methodik für kollektives Wohlbefinden zu definieren und sie über eine virtuelle Plattform zu verbreiten.

Feminist Server, ein loses Kollektiv von Feministinnen, die in ganz Europa unabhängige Server betreiben, ist an das Projekt herantreten, um es im Frühjahr 2022 im Rahmen einer "digitalen Residenz" auf seiner Infrastruktur zu hosten. Damit wird das Hauptproblem der Nutzung von Amazon Web Services für das Hosting des Projekts gelöst, was gegen die ethischen Grundsätze des Projekts verstoßen würde. Hoffentlich kann danach eine längerfristige Zusammenarbeit mit Feminist Server aufgebaut werden.

09. Sep. 2021

5388/Bo

XMPP-Media-Conf – Föderierte Audio- und Video-Konferenzen via XMPP

Schlussbericht

Zuwendungsempfänger:

Klaus Herberth

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS21S28 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Der Mensch hat ein natürliches Bedürfnis nach Kommunikation und vor allem Kommunikation in Gruppen ist ein fundamentales Verlangen. Gerade in Zeiten in denen man einander nicht sehen kann, bieten Audio- und Video-Konferenzen eine einfache Möglichkeit sich über verschiedene Themen auszutauschen. Dieses Bedürfnis befriedigen die meisten Personen über Plattformen die häufig ihren Sitz im Ausland haben und somit unklar ist wer alles Zugriff auf diese hoch sensiblen Daten hat. Gerade im Geschäftsbetrieb oder bei Behörden ist dieses Problem von höchster Bedeutung. Unser System soll genau so offen und föderal sein wie zum Beispiel die Telefonie oder E-Mail.

Um dieses Problem zu lösen wollten wir uns an bestehenden Vorschlägen in der XMPP Community orientieren und diese für die Nutzung in Gruppen entsprechend anpassen. Für die Implementierung wurde der bestehende JavaScript XMPP Client (JSXC) entsprechend erweitert. Hierzu haben wir folgende Meilensteine definiert und erfolgreich abgeschlossen.

1. Implementierung einer P2P Videokonferenz für kleine Gruppen auf Basis der bestehenden Protokollvorschläge (XEP-0272).
2. Feedback und aktuellen Stand der Implementierung des entsprechenden XEP mit der Entwickler Community teilen.
3. Evaluierung der bestehenden Videobridge mit dem Ziel eine möglichst gute A/V Qualität zu erreichen.
4. Implementierung einer Videokonferenz mit Videobridge die die Last der einzelnen Clients reduziert.
5. Ebenfalls Feedback zur weiteren Standardisierung des entsprechenden XEP-0340 an die Entwickler Community.

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

Die Zielgruppe sind alle Personen die in Gruppen via Audio/Video kommunizieren möchten oder müssen. Durch die schon bestehende breite Nutzerbasis von JSXC und die Integration in Nextcloud, Sogo, Wordpress, usw., werden die neuen Funktionen ebenfalls vielen Menschen schnell zugänglich sein.

Unser Projekt erlaubt es somit Administratoren eine offene und sichere Kommunikationslösung für ihre Nutzer*innen bereitzustellen mit einem sehr geringen Konfigurationsaufwand. Die Gewissheit, dass alle ausgetauschten Informationen innerhalb der eigenen Organisation bleiben fördert die Redefreiheit und somit eine offene Gesellschaft.

Ausführliche Darstellung der Ergebnisse

Architektur

Das Ziel einer Online-Konferenz ist es unterschiedliche Medien, wie Audio, Video, Folien, Bilder, oder Videos, mit mehreren Teilnehmenden zeitgleich zu teilen. Der einfachste Ansatz hierfür ist es, direkte Verbindungen zwischen allen Teilnehmer*innen aufzubauen (peer-to-peer, oder auch P2P abgekürzt). Dies bedeutet jede*r Teilnehmer*in sendet und empfängt einen Datenstrom an jede*n andere*n Teilnehmer*in (ein Mesh-Netzwerk). Der Vorteil dieser Methode ist, dass sie sehr einfach zu implementieren ist, wenn bereits Anrufe zwischen zwei Teilnehmenden möglich sind und keine weiteren Dienste involviert werden müssen. Ein Verbindungsaufbau ist also nur abhängig von den verwendeten Clients. Bei einer größeren Anzahl von Teilnehmer*innen stößt diese Methode aber schnell an ihre Grenzen, da ggf. der Datenstrom pro Teilnehmer*in unterschiedlich verarbeitet werden muss, was besonders bei mobilen Endgeräten oder Computern mit wenig Rechenleistung zu Problemen führen kann. Zusätzlich bieten die meisten privaten Internetanschlüsse eine asynchrone Übertragungsgeschwindigkeit, welche den Upload von Datenverbindungen, also das Senden von Medienpaketen, begrenzt. Gerade bei hochauflösenden Bildschirmübertragungen an viele Teilnehmende, kann dies zu einem limitierenden Faktor werden.

Aus diesem Grund empfiehlt es sich nach Möglichkeit, einen zentralen Vermittlungsdienst zu nutzen der die ein- und ausgehenden Medienverbindungen organisiert. Dies bedeutet, dass Clients Verbindungen zu diesem zentralen Service aufbauen und von dort die Daten von allen anderen Teilnehmer*innen erhalten. Je nach Implementierung spricht man von MCU (Multipoint Control Unit) oder SFU (Selective Forwarding Unit). MCU kombiniert beispielsweise alle Videosignale in ein einziges, indem es die Videos in einem vorgegebenen Raster anordnet und benötigt somit die geringste Bandbreite aller Möglichkeiten. Der Nachteil ist, dass das Layout der Videos vom Dienst vorgegeben wird und somit schlecht auf die jeweiligen Anforderungen des Endgeräts angepasst werden können. Zudem benötigt es für die Zusammenführung auf dem Server viel Rechenleistung. Aus diesen Gründen haben wir uns, neben der Implementierung der P2P Lösung, für die Nutzung eines SFU entschieden. Dieser Dienst verteilt alle eingehenden Signale an alle Teilnehmenden. Die*Der einzelne Teilnehmer*in sendet also nur einen Datenstrom, empfängt aber pro Teilnehmer*in einen. Dies sorgt für eine optimale Ausnutzung der asynchronen Bandbreite der meisten Internetanschlüsse. Zudem kann je nach Rechenleistung oder Verbindungsgeschwindigkeit

nur ein Teil der z. B. Videosignale angefordert werden. Dies erlaubt es auch Personen mit schlechter Internetanbindung an einer Videokonferenz teilzunehmen.

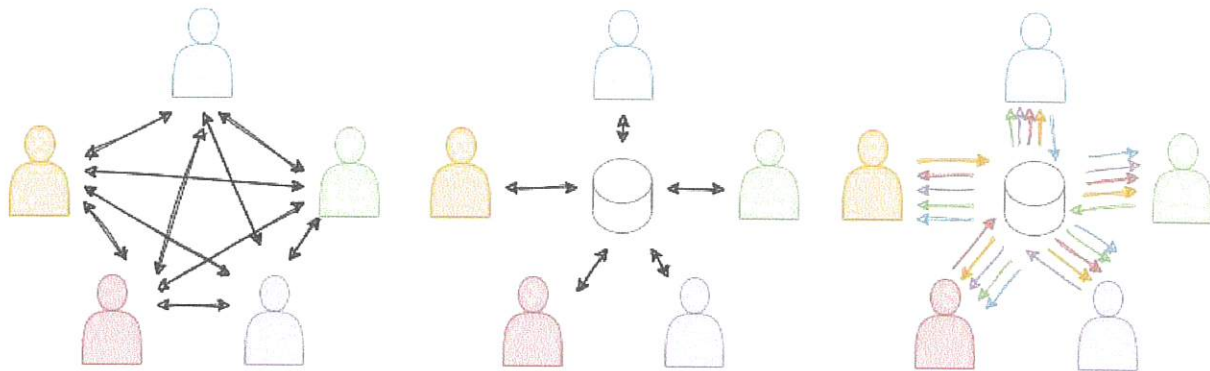


Figure 1: Links die Darstellung eines Mesh-Netzwerks aus P2P-Verbindungen zwischen den Teilnehmenden. In der Mitte eine MCU, bei der alle Teilnehmenden exakt eine ausgehende und eingehende Verbindung haben. Bei einer SFU (rechts) muss jede*r Teilnehmende nur eine ausgehende Verbindung aufbauen, aber für jede*n Teilnehmende*n eine eingehende.

Implementierung

Das Ziel für die Implementierung war es, den Aufwand möglichst gering zu halten und auf bestehende Techniken wie Jingle (XEP-0166) zurück zu greifen. Jingle ist bereits weit verbreitet und wird genutzt um P2P-Media-Verbindungen auszuhandeln. Um diese Technik nun für mehrere Verbindungen gleichzeitig einsetzen zu können, bestand die Herausforderung darin, allen beteiligten Konferenzteilnehmer*innen mitzuteilen, wann die Sitzung startet und die Liste aller Personen zwischen allen Mitgliedern zu synchronisieren.

Hierfür nutzen wir als Basis den Multi-User Chat (XEP-0045) mit einer nicht anonymisierten Konfiguration, da hier das Benutzermanagement schon vorhanden ist und alle Personen die gegenseitige Benutzer ID (Jabber ID) bereits kennen. Wenn in einem dieser Räume nun eine Online-Konferenz gestartet werden soll, schickt ein Mitglied eine entsprechende maschinenlesbare Nachricht (PROPOSE) in den Raum und damit an alle Mitglieder. Der Ablauf und Aufbau dieser Nachrichten ist angelehnt an Jingle Message Initiation (XEP-0353). Diejenigen, die teilnehmen möchten, senden nun eine passende "ACCEPT" Antwort. Da alle Nachrichten in derselben Reihenfolge bei allen Gruppenmitgliedern eingehen, wird nun diese genutzt um die Initiatoren der einzelnen Verbindungen festzulegen. Jedes Mitglied muss die Verbindung zu allen Personen initiieren, die ihre Antwort nach der eigenen geschickt haben. Zu beachten gilt, dass Personen nur Verbindungen von Benutzer*innen zulassen, die tatsächlich in diesem Raum sind und auch eine entsprechende Zustimmung zur Konferenz erteilt haben. Des Weiteren muss jede Nachricht eine Session ID enthalten, um mögliche Kollisionen zu unterbinden und jedes Mitglied muss seine vollständige User ID bei jeder dieser Nachrichten einfügen, da in MUC Räumen Nutzer*innen mit mehreren eigenen Geräten (multi-client support) angemeldet sein können. Zum Schließen einer Sitzung sendet jede Person eine dementsprechende Nachricht in den Raum um nachfolgenden Mitgliedern einen Überblick zu gewähren, wer gerade im Raum ist und wer nicht.

Die Implementierung für die Variante mit zentralem Dienst um Bandbreite zu sparen, funktioniert ähnlich wie der P2P-Ansatz. Es werden ähnliche Nachrichten in den Raum geschickt, aber die Nutzer*innen bauen nun nicht mit den anderen Mitgliedern eine Verbindung auf, sondern mit dem zentralen Server. Wir haben uns hier für den leicht zu erweiternden WebRTC Server Kurento entschieden der über eine Programmierschnittstelle konfiguriert werden kann. Sollte ein*e Nutzer*in sich nun dazu entscheiden, eine Konferenz zu starten, öffnet sie*er einen Kanal auf dem entsprechenden Server und startet eine Medienverbindung mit diesem. Im Anschluss veröffentlicht die Person die Kennung des Kanals im Raum. Alle Personen, die nun an der Konferenz teilnehmen möchten, öffnen selbst eine Medienverbindung mit dem Dienst und fragen alle Verbindungen von Personen, die ebenfalls teilnehmen möchten, am Server an. Im Anschluss wird in dem Raum wieder eine "ACCEPT" Nachricht gesendet, damit alle anderen Mitglieder wissen, dass sie nun die neue Verbindung anfordern können. Der Verbindungsabbau funktioniert analog zur P2P Variante.

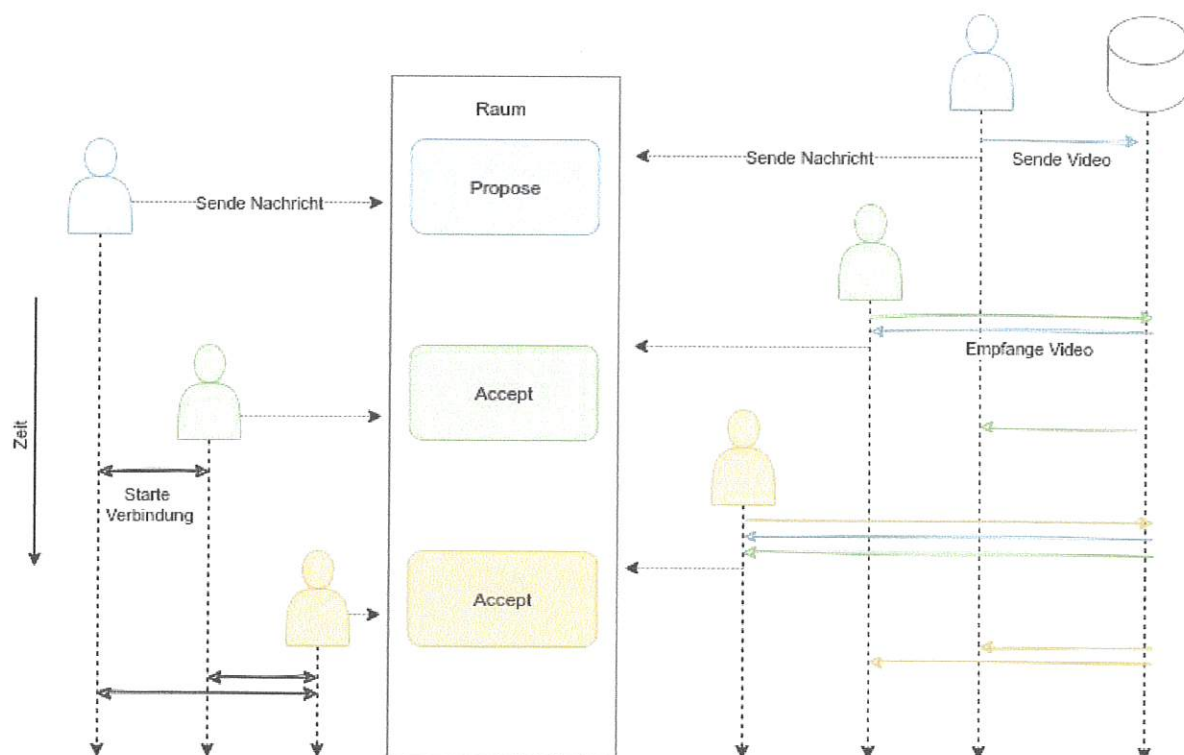


Figure 2: Die linke Seite beschreibt den Ablauf in der P2P-Variante, bei der jedes Mitglied eine Verbindung zu allen Personen aufbaut, die nach ihr der Konferenz beitreten. Die rechte Seite zeigt den Ablauf einer SFU. Wie ersichtlich ist, werden im Raum die gleichen (propose, accept) Nachrichten gesendet. Nur der Verbindungsaufbau unterscheidet sich jeweils.

Fazit

In der Projektlaufzeit konnten alle Meilensteine erreicht werden und somit können Benutzer*innen nun Audio- und Video-Konferenzen mit einem open Protokoll nutzen. Die Begleitung durch die Open Knowledge Foundation hat gezeigt, dass eine regelmäßige Evaluierung der eigenen Arbeit und des Umsetzungsstandes elementar für eine erfolgreiche Projektumsetzung ist. Die Miteinbeziehung von externen Experten half die Probleme neu zu verstehen und andere Lösungsansätze zu entdecken.

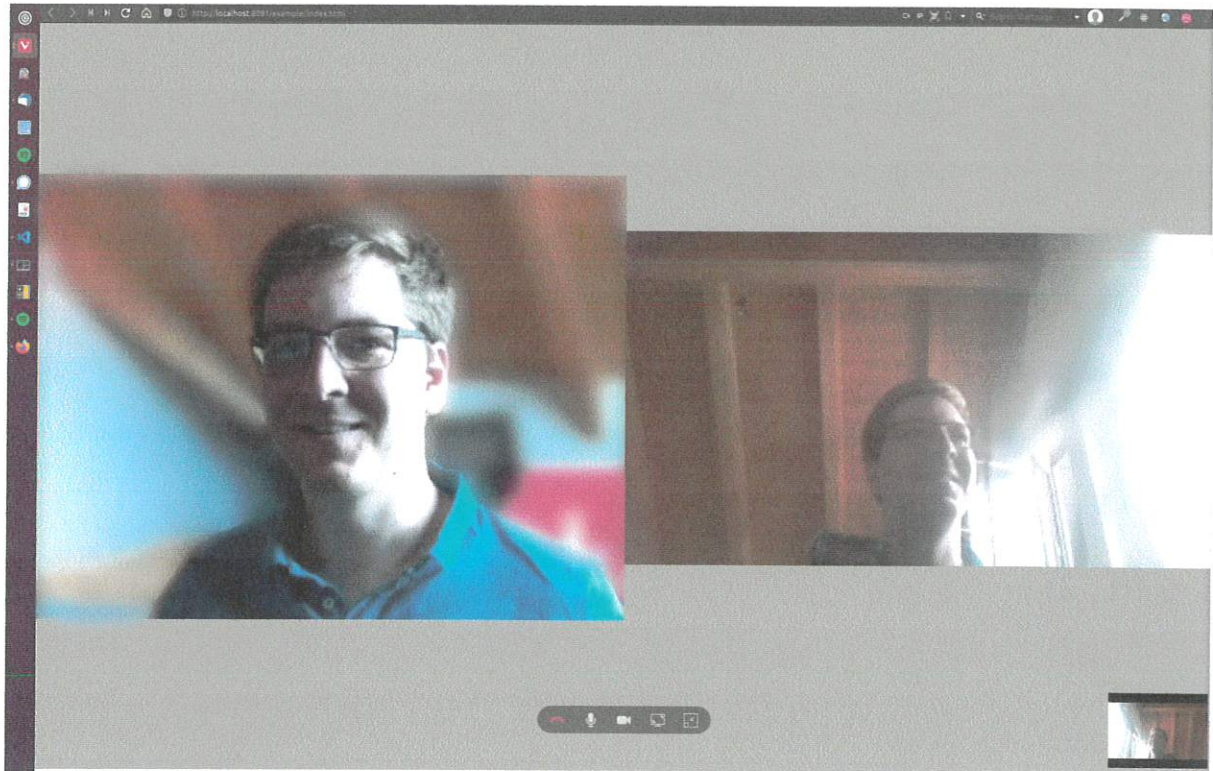


Figure 3: Darstellung einer Videokonferenz in JSXC mit drei Teilnehmer*innen

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Möglichkeit seinen eigenen Client zu nutzen um organisationsübergreifend Video-Konferenzen durchzuführen machen die tägliche Arbeit nicht nur sicherer, sondern auch einfacher und komfortabler. Um den Nutzer*innen tatsächlich eine Auswahlmöglichkeit von Clients zu bieten, sind wir zur Zeit in Absprache mit weiteren Entwickler*innen von populären XMPP Clients für sowohl Desktop, als auch Mobilgeräte. Das langfristige Ziel ist es die nun entwickelte Lösung in einen für alle offenen XMPP Standard zu überführen. Hierfür muss die aktuelle Schnittstelle zum SFU Server noch in eine XMPP Server Komponente überführt werden um einen Technologiewechsel für Client Entwickler*innen zu vermeiden.

Das Projekt hat uns ermöglicht noch tiefer in die Thematik WebRTC und Echtzeitkommunikation im Browser einzusteigen. Durch das erlangte Wissen erhoffen wir uns eine schnelle Umsetzung des Konzeptes in einen Standard.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Durch eine gute Vorbereitung konnten wir unsere Arbeitszeit zielgerichtet einsetzen und mussten keine größeren Ansätze verwerfen. Einzig die Anbindung des SFU Servers hatten wir uns ohne Zwischenschicht vorgestellt, aber in der Umsetzungsphase mussten wir feststellen, dass dies doch nötig ist und auch für eine spätere Standardisierung wichtig ist.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Interessierte können weitere Informationen im Bericht zur Demoweeek des Prototype Funds¹, in unserem Blog Eintrag², oder im Repository³ auf Github erhalten. Entsprechende Demos sind ebenfalls in den Artikeln verlinkt.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Auf Grund der begrenzten Projektlaufzeit mussten wir einige Komfortfunktionen wie das Hervorheben des Sprechenden, oder eine automatische Umschaltung zwischen P2P und SFU abhängig von der Gruppengröße zurückstellen. Hierdurch konnten wir im geplanten Zeit- und Kostenrahmen bleiben.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Bis auf die Diskussionen mit anderen XMPP Entwicklern, gab es keine Einflüsse von Dritten die den Verlauf des Projektes maßgeblich beeinflusst haben.

1 <https://demoweeek.prototypefund.de/projects/26-javascript-xmpp-client.html>

2 <https://www.jsxc.org/blog/2021/08/31/A-group-call-proposal.html>

3 <https://github.com/jsxc/jsxc/tree/feat-muc-media>