



SOFTWARE SPRINT (PROTOTYPE FUND) AUSWAHLRUNDE SEPTEMBER 2018

Konsolidierter Schlussbericht

Förderkennzeichen:

01S18S43
01S18S44
01S18S45
01S18S46
01S18S47
01S18S48
01S18S49
01S18S50
01S18S51
01S18S52
01S18S53
01S18S54
01S18S55
01S18S56
01S18S57
01S18S58
01S18S59
01S18S60
01S18S61
01S18S62
01S18S63
01S18S64
01S18S65

Vorhabenbezeichnung: Software Sprint – 23 Einzelvorhaben

Laufzeit der Einzelvorhaben: 01.03.2019-31.08.2019

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 Autoren (Zuwendungsempfängern).

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel Software Sprint – 23 Einzelvorhaben	
4. Autor(en) [Vorname(n), Name(n)] Frank Rieger, Gordon Gremme, Matthias Menk Hannes Mehnert Thomas Benjamin Senior Harald Welte Sebastian Meier, Daniel Lommes Marcus Hoffmann Peggy Sylopp, Aislyn Rose Lars Marvin Wißfeld Christopher Adams Robert Buchholz Jonas Hörsch Stefanie Schirmer Thomas Werkmeister Nastasja Krohe Matthias Hannich Adam Harvey, Jules LaPlace Andreas Dewes, Katharine Jarmul Tilman Miraß Katharina Rasch Matthias Fratz Florian Dold Thomas Viehmann Johannes Filter	5. Abschlussdatum des Vorhabens 31.08.2019
	6. Veröffentlichungsdatum 11.12.2019
	7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) 01IS18S43 Frank Rieger, Gordon Gremme und Matthias Menk GbR 01IS18S44 Hannes Mehnert 01IS18S45 Dr. Thomas Benjamin Senior 01IS18S46 Harald Welte 01IS18S47 ULTRAPOP - Sebastian Meier und Daniel Lommes GbR 01IS18S48 Marcus Hoffmann 01IS18S49 Peggy Sylopp und Aislyn Rose GbR 01IS18S50 Lars Marvin Wißfeld 01IS18S51 Christopher Adams 01IS18S52 Robert Buchholz 01IS18S53 Jonas Hörsch 01IS18S54 Stefanie Schirmer 01IS18S55 Thomas Werkmeister 01IS18S56 Krohe, Dehm, Hohbach GbR 01IS18S57 Matthias Hannich 01IS18S58 Adam Harvey und Jules LaPlace GbR 01IS18S59 Andreas Dewes & Katharine Jarmul GbR 01IS18S60 Tilman Miraß 01IS18S61 Katharina Rasch 01IS18S62 Fratz & Held GbR 01IS18S63 Florian Dold 01IS18S64 Dr. Thomas Viehmann 01IS18S65 Johannes Filter	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen *) 01IS18S43-01IS18S65
	11. Seitenzahl 121
12. Fördernde Institution (Name, Adresse) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. Literaturangaben
	14. Tabellen

*) Auf das Förderkennzeichen des BMBF soll auch in der Veröffentlichung hingewiesen werden.

	15. Abbildungen
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum) Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR) DLR Projektträger Gesellschaft, Innovation, Technologie Softwaresysteme und Wissenstechnologien Berlin	
18. Kurzfassung Der Schlussbericht umfasst die Einzelschlussberichte der fünften Auswahlrunde der Fördermaßnahme Software Sprint (Einreichungsdatum Skizze: 30.09.2018). Die Auswahl der Vorhaben erfolgte nach externer Begutachtung unter Abstimmung mit dem BMBF.	
19. Schlagwörter Software Sprint, Prototype Fund	
20. Verlag	21. Preis

*) Auf das Förderkennzeichen des BMBF soll auch in der Veröffentlichung hingewiesen werden.

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report	
3. title Software Sprint – 23 Einzelvorhaben		
4. author(s) (family name, first name(s)) Frank Rieger, Gordon Gremme, Matthias Menk Hannes Mehnert Thomas Benjamin Senior Harald Welte Sebastian Meier, Daniel Lommes Marcus Hoffmann Peggy Sylopp, Aislyn Rose Lars Marvin Wißfeld Christopher Adams Robert Buchholz Jonas Hörsch Stefanie Schirmer Thomas Werkmeister Nastasja Krohe Matthias Hannich Adam Harvey, Jules LaPlace Andreas Dewes, Katharine Jarmul Tilman Miraß Katharina Rasch Matthias Fratz Florian Dold Thomas Viehmann Johannes Filter	5. end of project 31.08.2019	6. publication date 11.12.2019
	7. form of publication	
	8. performing organization(s) (name, address) 01IS18S43 Frank Rieger, Gordon Gremme und Matthias Menk GbR 01IS18S44 Hannes Mehnert 01IS18S45 Dr. Thomas Benjamin Senior 01IS18S46 Harald Welte 01IS18S47 ULTRAPOP - Sebastian Meier und Daniel Lommes GbR 01IS18S48 Marcus Hoffmann 01IS18S49 Peggy Sylopp und Aislyn Rose GbR 01IS18S50 Lars Marvin Wißfeld 01IS18S51 Christopher Adams 01IS18S52 Robert Buchholz 01IS18S53 Jonas Hörsch 01IS18S54 Stefanie Schirmer 01IS18S55 Thomas Werkmeister 01IS18S56 Krohe, Dehm, Hohbach GbR 01IS18S57 Matthias Hannich 01IS18S58 Adam Harvey und Jules LaPlace GbR 01IS18S59 Andreas Dewes & Katharine Jarmul GbR 01IS18S60 Tilman Miraß 01IS18S61 Katharina Rasch 01IS18S62 Fratz & Held GbR 01IS18S63 Florian Dold 01IS18S64 Dr. Thomas Viehmann 01IS18S65 Johannes Filter	
12. sponsoring agency (name, address) Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	9. originator's report no.	
	10. reference no. 01IS18S43-01IS18S65	
	11. no. of pages 121	
	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 Gesellschaft, Innovation, Technologie Softwaresysteme und Wissenstechnologien Berlin	
18. abstract The final report includes the individual final reports for the 5th call of participants within the public funding activity Software Sprint. The selection of the projects (call No. 5 deadline: 30.09.2018) took place after external evaluation under coordination with the BMBF.	
19. keywords Software Sprint, Prototype Fund	
20. publisher	21. price

Schlussbericht CypherLock—Nötigungsresistente Festplattenverschlüsselung

Frank Rieger, Gordon Gremme und Matthias Menk GbR

2019-09-30

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S43 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei den Autoren.

Kurze Darstellung der Aufgabenstellung und Motivation

Das Ziel des Projektes *Cypherlock* ist die Implementierung eines Softwaretools und der dazugehörigen Backend-Infrastruktur um Geheimnisse (im Konkreten: Passphrases zur Festplattenverschlüsselung) durch zeitbasierende Regeln vor Zugriff zu schützen und damit nötigungsresistente Festplattenverschlüsselung zu ermöglichen. Dieses Ziel ist durch die Tatsache motiviert, dass es weltweit einen Anstieg an forensischen Angriffen und Zwangsmitteln gegen Journalisten bei Grenzübertritt oder auch im Rahmen von Checkpoints in Konfliktregionen gibt. Diese Angriffe gefährden das Wohl der Journalisten als auch ihrer Quellen.

Das Projekt ist in vier Felder unterteilt worden:

1. Implementierung notwendiger Algorithmen, Datenformate und Protokolle für verteilte zeitbasierende Regeln.
2. Implementierung von Backend-Software zur Beantwortung von Entschlüsselungsanfragen und zur Verwaltung serverseitiger geheimer Schlüssel.
3. Implementierung von Client-Software zur Verwaltung von geheimen Schlüsseln und verschlüsselten Nachrichten.
4. Implementierung von Benutzeroberflächen.

Als Milestones wurden definiert:

1. Datenformate und Serialisierungsbibliothek für schützenswerten Daten in geschütztem Arbeitsspeicher ohne Allokation.

2. In-Memory Verwaltung von Geheimnissen, mit Schutz gegen Angriffe durch privilegierte User wie z.B. durch Memory-Scanning.
3. Implementierung der notwendigen kryptographischen Verfahren für den Einsatz in geschütztem Arbeitsspeicher ohne Allokation.
4. Implementierung des verteilten Wiederherstellungs-Protokolls für Geheimnisse.
5. Protokoll-Implementation für ein byzantines Synchronisierungs-Protokoll, sowie dafür notwendige Basisalgorithmen (speziell verifizierbare Merkle-Trees).
6. Implementierung eines nachträglich offenlegbaren Random-Number-Generators für Schlüssel- und Nonce-Erzeugung.
7. Implementation der Client-seitigen Speicherung persistenter Daten unter Berücksichtigung von Informations-Leaks durch Zugriffscharakteristika.
8. Anbindung von Bluetooth-HID Hardware für die Kommunikation zwischen Benutzersoftware (auf Smartphone) und zu schützenden Gerät (Laptop).
9. Implementierung von Benutzersoftware: CLI (primär Linux), iOS und Android.
10. Erstellung relevanter Dokumente zur technischen Beschreibung des Systems.
11. Erstellung von Schulungsmaterial für Benutzer und Dritte.

Die Aufgaben wurden über mehrere Projekt-Teilnehmer verteilt und iterativ bearbeitet und in wöchentlichen Koordinierungs-Treffen besprochen.

Neben dem Ziel ein Produkt zu schaffen, dass die beschriebenen Angriffe erfolgreich abwehren wurde zusätzlich ein Fokus auf die Sicherheit der Implementation gegen fortgeschrittene Angriffe, die sichere und einsatz-relevante Benutzung, sowie die langfristige Pflege der Software gelegt.

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

Die primäre Zielgruppe unserer Lösung sind Journalisten, die auf Rechercheisen ihre lokale Daten gegen den Zugriff Dritter schützen müssen.

Bei Benutzung unserer Software können sie die Freigabe von kryptographischen Schlüsseln erfolgreich verhindern in dem sie eine Herausgabe kurzfristig verzögern. Danach sind sie in der Lage zu beweisen, dass kein Zugriff auf die notwendigen Schlüssel mehr möglich ist.

Dazu steht ihnen ein Software-Tool sowie die dafür nötige Infrastruktur zur Verfügung die eine Benutzung auch in Stress-Situationen sicher ermöglicht.

Das Projekt wurde im Rahmen der 5. Runde des Prototype-Funds unter dem Teil-Schwerpunkt “Datensicherheit” gefördert. Cypherlock erhöht die Datensicherheit für besonders gefährdete Berufsgruppen wie Journalisten signifikant. Der gewählte Ansatz ist innovativ, bisher noch nie realisiert worden und hochgradig praxisrelevant.

Ausführliche Darstellung der Ergebnisse

Die Milestones 1-7 wurden vollständig erreicht, Milestone 9 nur teilweise, Milestones 8, 10 und 11 sind nicht erreicht:

Datenformate und Serialisierungsbibliothek für schützenswerten Daten in geschütztem Arbeitsspeicher ohne Allokation

Wir haben eine allgemein nutzbare Implementierung für das serialisieren von Daten geschaffen die ausschliesslich pre-allozierten Speicher benutzt und in Verbindung mit bestehenden Schutzmechanismen die einfache, effiziente und sichere Konvertierung von Strukturen aus Standard-Typen in Byte-Messages erlaubt. Damit wird die Umsetzung von sicherer Software in der gewählten Programmiersprache *Go* unterstützt und wesentlich vereinfacht.

In-Memory Verwaltung von Geheimnissen, mit Schutz gegen Angriffe durch privilegierte User wie z.B. durch Memory-Scanning.

Als fundamentales Modul unserer Lösung haben wir eine Bibliothek für die Erzeugung und Verwaltung von geheimen Schlüsseln (primär Curve25519 und symmetrische Schlüssel) in geschütztem Speicher geschaffen. Diese Bibliothek erlaubt das Erzeugen, Nutzen und persistente Speichern von Schlüsseln auf Systemen die nicht garantiert gegen den Zugriff durch privilegierte Nutzer gesichert sind.

Weiterhin beinhaltet diese Bibliothek die Erzeugung und Verwaltung von zeitgesteuerten Schlüsseln für die Umsetzung von Richtlinien für zeitlich begrenzten Zugriff auf Schlüsselmaterial. Damit ist eine Implementierung von Systemen mit Zeitrelevanz einfach möglich.

Implementierung der notwendigen kryptographischen Verfahren für den Einsatz in geschütztem Arbeitsspeicher ohne Allokation.

Ein wesentlicher Bestandteil unseres Systems ist eine Bibliothek für die Komposition von geteilten Geheimnissen (shared secrets) auf Basis von Curve25519. Wir haben ein allgemeines Framework für die Definition solcher Komposite geschaffen mit der z.B. auch multi-party Datei-Verschlüsselung oder sichere Messenger

einfacher umsetzbar sind. Im Kontext unseres Projektes erlaubt uns diese Bibliothek die Abbildung aller kryptographischer Envelopes mit öffentlichen Schlüsseln auf eine einzige Konstruktion.

Weiterhin beinhaltet diese Bibliothek eine Implementierung aller von uns genutzten kryptographischen Algorithmen für die Verwendung von geschütztem Speicher.

Implementierung des verteilten Wiederherstellungs-Protokolls für Geheimnisse.

Kernelement unseres Projektes ist das Protokoll für die Wiederherstellung von verteilten Geheimnissen. Hierzu wurden die notwendigen Protokoll-Nachrichten, die persistente Speicherung der States, sowie der Protokollfluss implementiert.

Protokoll-Implementation für ein byzantines Synchronisierungs-Protokoll, sowie dafür notwendige Basisalgorithmen (speziell verifizierbare Merkle-Trees).

Für die byzantine Version unseres Systems, das auch mit der Kontrolle eines Angreifers auf Teile der Backend-Infrastruktur sicher umgehen kann, wurden die Nachrichten-Typen, Speicherung der States sowie der Protokollfluss implementiert. Als weiteren wiederverwendbaren Programmcode wurde in diesem Kontext eine Bibliothek für das Erzeugen und Verifizieren von Merkle-Trees geschaffen die auf Basis von validierten und bewiesenen Konstruktoren basiert.

Implementierung eines nachträglich offenlegbaren Random-Number-Generators für Schlüssel- und Nonce-Erzeugung.

Um das Verhalten und die Funktion unserer Software durch Dritte beweisbar zu machen wurde ein Modul entwickelt, dass die Erzeugung von Zufallszahlen im Nachhinein verifizierbar macht. Damit kann sichergestellt werden, dass alle Daten, die von der Applikation gesendet oder gespeichert wurden, frei von Seitenkanälen sind und das Zufallszahlen ohne Manipulation erzeugt wurden. Der erzeugte Beweis ist für die Applikation nicht veränderbar, vorhersehbar oder unterdrückbar.

Implementation der Client-seitigen Speicherung persistenter Daten unter Berücksichtigung von Informations-Leaks durch Zugriffscharakteristika.

Die Speicherung von schützenswerten Daten für unsere Applikation wurde in einem Modul realisiert, dass den randomisierten Zugriff auf Dateien implementiert

die verschlüsselt gespeichert sind. Die Verschlüsselung benutzt unvorhersehbare „Decoy-Schlüssel“ um das Erpressen von Passwörtern unwirksam zu machen. Ein Angreifer hat keine Möglichkeit zwischen validen und gefälschten Daten zu unterscheiden. Diese Methode ermöglicht dem Benutzer die Wahrung von „Plausible Deniability“ und verhindert forensische Analysen.

Als Bestandteil dieses Modules wurde die sichere Löschung von Daten auf iOS eingebunden, die auf das in Apple-Geräten vorhandene Secure Element zur Schlüsselerzeugung und Verwaltung aufbaut.

Implementierung von Benutzersoftware: CLI (primär Linux), iOS, Android.

Aufgrund technischer (sichere Löschung) und zeitlicher Gründe wurde dieser Milestone nur teilweise erreicht.

Es besteht ein Prototyp der iOS App, die User Interface und Benutzerführung implementiert. Die Einbindung von Bluetooth und Netzwerk-Protokollen fehlt und wird im Nachlauf der Förderphase implementiert.

Neben den Ergebnissen der Implementierungsarbeiten ist insbesondere die Arbeit im Bereich UX/UI entscheidend gewesen. Hierbei hat insbesondere die Arbeit mit einer vom Prototyp Fund zur Verfügung gestellte externe Beraterin dazu beigetragen unsere Annahmen zum Benutzer-Verhalten und Ansprüchen zu überprüfen und entscheidend weiterzuentwickeln. Dabei wurde ein vermutlich kritischer Fehler in unseren Annahmen frühzeitig erkannt und konnte somit rechtzeitig in der Entwicklung korrigiert werden.

Weiterhin wurden im Rahmen des Projektes Fähigkeiten zur Frontend-Entwicklung (Flutter) und der Integration von Cross-Platform-Code auf mobilen Endgeräten aufgebaut (go-mobile Integration mit Flutter/Dart).

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Unsere Zielgruppe erhält die Möglichkeit schützenswerte Daten sicher zu verwalten und gegen den Zugriff Dritter in Zwangssituationen zu schützen sowie selber den Umfang einer solchen Situation zu beeinflussen. Damit sind unsere User befähigt Recherchen mit besser einschätzbaren Risiken durchzuführen.

Die erstellten Software-Bibliotheken ermöglichen weiterhin die Implementierung von Produkten mit Hinblick auf sichere Speicher-Nutzung und Schlüsselverwaltung, sowie die einfache Umsetzung von Protokollen die kompositive Verschlüsselung über ECDH beinhalten (Messaging, Forward-Security, Multi-Party Kommunikation). Zusätzlich sind die erarbeiteten Module zur Serialisierung in sicherem Speicher für den Einsatz in Produkten geeignet, die kryptographische Software auf Geräten mit geringer Sicherheit implementieren. Zudem ist unsere

Implementierung für eine nachträglich verifizierbare Quelle für Zufallszahlen ein allgemein nutzbares Modul für die Beweisbarkeit von kryptographischen Protokollen und kann z.B. dafür genutzt werden Seitenkanäle im Nachrichtenaustausch aufzudecken.

Die Software wird auch in Zukunft von uns weiterentwickelt und fehlende Elemente implementiert. Dieses ist bis Ende 2019 geplant und wird in Teilzeit umgesetzt. Es sollen keine Funktionen hinzugefügt werden, die nicht in der ursprünglichen Leistungsbeschreibung vorgesehen waren.

Weiterhin wird das Endprodukt als App im Apple AppStore angeboten werden. Die notwendige Infrastruktur für den Betrieb des Systems wird durch Erlöse aus dem App-Verkauf erzielt. Weiterhin werden wir Dritte beim Aufbau von dedizierten Backends unterstützen. Eine Weiterentwicklung der Android-Version ist nicht geplant, da sie innerhalb der definierten Sicherheitsparameter im Verlaufe des Projekts als nicht umsetzbar erwiesen hat.

Es gibt weiterhin Versuche die Client-Software auf mobile Peripherie-Geräte/single-board computer (wie z.B. USB-Armory Mk II) zu portieren um eine höhere Sicherheit in der Anwendung zu erreichen. Dieses hätte den weiteren Vorteil, dass Abhängigkeiten zu einer einzigen Plattform gemindert werden könnten.

Für unser Team ist das Projekt mit einer starken Lernkurve verbunden gewesen. Neben der Einarbeitung in Frontend-Technologien (Flutter, UX/UI Prozesse) ist insbesondere die Möglichkeit sich umgehend mit den forensisch relevanten Aspekten von mobiler Hardware auseinanderzusetzen eine Herausforderung sowie auch eine Bereicherung gewesen.

Wir haben es damit erreicht, zwei wesentliche Fähigkeiten für zukünftige Projekte aufzubauen:

1. Benutzerzentriertes Vorgehen in den Bereichen User Experience und insbesondere auch User Interfaces. Insbesondere die Erweiterung unseres Horizonts im Hinblick auf nicht-technische Benutzer ist bereichernd und wird uns für weitere Projekte befähigen.
2. Die Auseinandersetzung mit der Sicherheit mobiler Endgeräte und möglicher sicherer Peripherie hat einen grossen Einfluss auf weitere Pläne unsererseits. Als Konsequenz dieser Lektionen werden wir uns in Zukunft stark auf die Entwicklung von sicheren Basistechnologien konzentrieren.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Während der initialen UI/UX Phase ist unsere Annahme, dass Benutzer ständig Zugriff auf das Internet haben verworfen worden. Daraus hat sich ergeben, dass wir schutzwürdige Daten sicher von Endgeräten löschen können müssen.

Nach langwieriger Recherche und mehreren Implementationsversuchen mussten wir allerdings feststellen, dass eine forensisch sichere Löschung auf aktuellen Android-Geräten nicht sichergestellt werden kann. Wir haben dazu keine Lösung gefunden und haben daher die Entwicklung des Android Clients eingestellt.

Für iOS Geräte konnte eine akzeptable Lösung gefunden und umgesetzt werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Der aktuelle Stand der Entwicklung (Source Code, Projekt-Dokumente) sind über die Cypherlock Webseite¹ beziehbar. Die Website wird auch in Zukunft fortlaufend mit neuen Ergebnissen aktualisiert.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Zwei Ereignisse haben einen Mehraufwand von insgesamt ca. 160 Stunden erzeugt:

1. Während der initialen UI/UX Phase wurde unsere Annahme, dass Benutzer ständig Zugriff auf das Internet haben, verworfen. Dieses hatte zur Folge, dass wir unsere Protokoll-Spezifikation verändern mussten. Ohne Online-Fähigkeit ist es notwendig geheime Daten vom Endgerät so sicher löschen zu können, dass sie auch durch forensische Analyse nicht wieder hergestellt werden können. Weiterhin macht es erforderlich, dass Zeit-Regeln nicht nur einzeln sondern auch in grösseren Mengen durch einzelne Nachrichten devalidiert werden können.

Aus Konsequenz daraus mussten wir Teile des Protokolls grundlegend verändern sowie die Backend-Implementation für das Speichern von User-Daten neu programmieren. Weiterhin haben wir signifikanten Aufwand für die Recherche zum sicheren Löschen von Daten auf mobilen Endgeräten vorgenommen und mehrere Methoden dazu implementiert und validiert.

2. Zum letzten Drittel des Projektes wurde ein fundamentaler Angriff gegen die Verschlüsselung des Bluetooth-Protokolls bekannt (die sog. KNOB Attack²). Dieser zielt auf die Implementierung des Bluetooth-Protokolles und ist breit wirksam. Der Angriff erlaubt es einem Angreifer ohne physikalischen Zugriff auf die verwendeten Geräte die Verschlüsselung der Nachrichtenübertragung abzufangen und im Klartext zu lesen. Da unser Projekt eine sichere Uebertragung von Daten über Bluetooth zwingend voraussetzt mussten wir die beschriebene Schwachstelle in der von uns vorgesehenen

¹<https://cypherlock.org>

²<https://knobattack.com>

Hardware verifizieren. Daraus ergab sich eine Kooperation mit dem Hersteller um die Auswirkungen auf die Sicherheit unseres Produktes zu mindern. Dieser Prozess ist noch nicht abgeschlossen und ein erfolgreicher Ausgang ist momentan nicht sicher.

Daher haben wir weiterhin alternative Hardware-Plattformen recherchiert und validiert. Sollten eine Absicherung der bestehenden Hardware durch den Hersteller nicht erfolgen so besteht für uns die Möglichkeit Anpassungen an alternativen Hardware-Produkten vorzunehmen um die Sicherheit und Nutzbarkeit unserer Software zu gewährleisten.

Grundlegend haben wir in der initialen Projektplanung den Aufwand für die Umsetzung im Kontext von unsicherem Speicher unterschätzt. Zu Anfang sind wir davon ausgegangen, dass wir alle notwendigen kryptographischen Algorithmen als Implementierungen vorfinden würden die einfach für die Verwendung mit unsicherem Speicher nutzbar sind. Im Projektverlauf mussten wir allerdings erheblichen Aufwand in die Implementierung von Speicher-Schutz-Mechanismen stecken und waren weiterhin gezwungen zusätzlichen Programmcode für grundlegende Operationen wie Serialisierung und Speicherung zu schaffen. Als Resultat daraus haben wir ca. 160 Stunden zusätzlich in die Implementierung von Bibliotheken investiert, die den Parametern des Projektes gerecht werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Bis auf den bereits oben erwähnten fundamentalen Angriff auf das Bluetooth-Protokoll gab keine Entwicklungen oder Veröffentlichungen Dritter die einen Einfluss auf Zielsetzung oder Arbeitsprozesse hatten.

Richtlinie zum „Software-Sprint“

OpenVPN-Client – Robuster OpenVPN-Client mit geringem Ressourcenverbrauch

Schlussbericht

Zuwendungsempfänger:

Hannes Mehnert

DLR PT SW Berlin

Eing.am:

16. Sep. 2019

Eingangsnr.: 5216/18

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S44 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

OpenVPN-Client ist eine auf dem Unikernel MirageOS aufsetzende, zuverlässige und einfach zu bedienende OpenVPN-Implementierung, die z.B. Zusammenarbeit von unterschiedlichen Standorten aus mit Hilfe einer kryptographisch sicheren Verbindung.

Die Motivation war eine Alternative zu den existierenden OpenVPN-Implementierungen, die in der fehlerträchtigen Programmiersprache C geschrieben sind, zu entwickeln. Auch das Schreiben einer Spezifikation des OpenVPN-Protokolls war Motivation.

Die Meilensteine waren wie folgt:

- Nach 1,5 Monaten: eine Spezifikation des OpenVPN Protokolls, die wir während der Projektlaufzeit noch detaillierter ausführen werden.
- Nach 3 Monaten ist die Unterstützung von „tun“ Interfaces in MirageOS
- Nach 4 Monaten eine erste OpenVPN Bibliothek inklusive Tests und einen Konfigurations-Dateiformat Parser
- Nach 5 Monaten ist ein OpenVPN Unikernel fertiggestellt
- In dem verbleibenden Monat: Interoperabilitäts-Tests und weitere Dokumentation

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

Die Zielgruppe sind Aktivisten, Journalisten, und Open Source Nutzer. Wir wollen ein Drop-in replacement für OpenVPN-Client entwickeln - und das Konfigurationsformat übernehmen - um einen leichten Umstieg zu ermöglichen. Im Rahmen der Förderung wurde auch ein OpenVPN-Server entwickelt.

Der OpenVPN-Client bietet einen wichtigen Beitrag zu dem Themenfeld Sicherheit. Der Server ist zusätzlich ein wichtiger Beitrag in den Themenfeldern Sicherheit und Infrastruktur, da er unabhängige und robuste Infrastruktur für alle zum selbst Betreiben verfügbar macht

Ausführliche Darstellung der Ergebnisse

Im Projektverlauf wurden die Programmier-meilensteine größtenteils erreicht, mit einiger Verzögerung, die Dokumentation und Spezifikation ist leider nicht in gewünschtem Ausmaß fertig geworden.

- Nach 3,5 Monaten war ein erster OpenVPN-Client, der mit einem bestehenden OpenVPN-Server eine Verbindung aufbauen konnte, entwickelt worden. Dieses beinhaltet den Meilenstein „OpenVPN Bibliothek und Konfigurations-Dateiformat Parser“

- Ein OpenVPN-Client, der ein „tun“ Interface verwendet, war nach 4 Monaten fertiggestellt. Dies ist keine direkte Integration in MirageOS, hat aber für das OpenVPN-Client Projekt den gleichen Effekt.

- Nach 4,5 Monaten war ein OpenVPN Unikernel, der den Tunnel an andere virtuelle Maschinen zur Verfügung stellt, entwickelt

Ohne die Förderung hätten wir das Projekt nicht finanziert bekommen. Die wöchentlichen Standup-Meetings im Chat waren eine gute Möglichkeit, um mit den anderen Gruppen in Kontakt zu bleiben.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Der Nutzen der Zielgruppe ist ein OpenVPN-Client mit geringem Speicherverbrauch. Dies ist zB für Benutzer von QubesOS sehr interessant, wo bereits an einer Integration gearbeitet wird. Die Entwicklung der OpenVPN-Bibliothek (statt nur einer Applikation) führte schon zu der Wiederverwendung der Bausteine als Server-Applikation! Nur durch die Offenlegung des Source codes können andere unabhängig der Software vertrauen, dass sie genau das tut, was sie soll, und nichts anderes. Durch die Entwicklung habe ich mehr über das OpenVPN-Protokoll gelernt, ein weit verbreitetes VPN Protokoll, und wie sowas implementiert wird.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Das Hinzufügen von Timeouts zum initialen Prototypen, um Protokollkonform zu sein, führte zu der Erkenntnis, dass viel Code nochmals umgeschrieben werden musste. Bei der Entwicklung der Auswahl des Server-Endpunktes führte wieder zu der Erkenntnis, dass hier viel Code umgeschrieben werden musste, und die Interaktion zwischen Unikernel und Bibliothek nochmals überdacht werden musste.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Eine Projektwebseite befindet sich noch im Aufbau. Die OpenVPN-Bibliothek wird in das OCaml package Archiv released werden, sobald sie in einem guten Zustand ist.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Das Analysieren (Reverse-Engineering) des Protokolls war aufwendiger als erwartet. Das Formulieren der Spezifikation hat den Zeitrahmen gesprengt, und ist somit nicht fertig geworden – aber in der Freizeit nach der Förderung wird daran weiter gearbeitet.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Der OpenVPN-Client Unikernel verwendet zwei virtuelle Netzwerkkarten, die eine ist mit dem Internet verbunden, wo der Tunnel zum OpenVPN-Server aufgebaut wird. Die andere ist mit den virtuellen Maschinen verbunden, deren Datenverbindungen über den Tunnel rausgeschickt werden. Die Möglichkeit, mehrere Netzwerkkarten mit MirageOS (und moderne Hypervisor wie KVM, BHyve oder seccomp) zu verwenden, wurde erst während des OpenVPN-Projektes von einem externen Entwickler in der Slowakei fertiggestellt. Diese Projekte ergänzen sich, und wir haben uns viel über die Anforderung, das Design, und der konkreten Implementierung von mehreren Netzwerkkarten ausgetauscht. Diese Erweiterung von MirageOS wird in den nächsten Wochen released, sowie das OpenVPN-Projekt als ersten Nutzer.

EyeSkillsAtHome

Schlussbericht

Zuwendungsempfänger:

Dr. Thomas Benjamin Senior

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S45 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

In einer früheren Prototype Fund - Runde (Kohorte 3) habe ich eine Reihe von Open-Source-Virtual-Reality-Umgebungen erstellt, welche die neurologische Natur des Schielens (Unterdrückung des Seheindrucks und Innen-/Aussenstellung des Auges, was zu mangelndem 3D-Sehen führt) experimentell demonstrieren konnten. Diese Umgebungen waren in allen Fällen, in denen wir Tests durchführten, in der Lage, die Wahrnehmung in beiden Augen wieder zu aktivieren. In vielen Fällen konnte auch gezeigt werden, dass eine Person in der Lage war, das 3D-Sehen schnell wiederherzustellen und ihre Augen sogar vorübergehend parallel zu stellen. Dieses erste System erforderte die Verwendung durch einen Fachexperten, um den Teilnehmer durch die verschiedenen Umgebungen zu führen und deren Bedeutung zu interpretieren.

Die Motivation für dieses aktuelle Projekt bestand darin, ein Vehikel zu erstellen, mit dem der Erfolg des ersten Projekts einem breiteren Publikum präsentiert werden kann. In der Praxis bedeutete dies die Erstellung einer „Tutorial“-ähnlichen Struktur, in die wir die Erfahrungen aus der vorherigen Iteration einbetten konnten. Diese Struktur sollte es den Teilnehmern ermöglichen, Fortschritte bei der Erforschung der Fähigkeiten ihrer Augen zu erzielen, die wissenschaftlichen Hintergründe zu verstehen und dies ohne die Anwesenheit eines Praktikers zu tun. Auf diese Weise können wir das Wissen um und die Verwendung der Techniken skalieren und gleichzeitig weitere Belege für die Leistung und Anwendbarkeit des Ansatzes sammeln.

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

EyeSkills ist in erster Linie eine Form von Civic Tech, also ein Instrument, mit dem Bürgerinnen und Bürger ihre eigenen Anliegen besser verstehen können, und das ihnen zudem Tools zur Verfügung stellt, mit denen sie sich selbst helfen können. Es ist jedoch auch ein Versuch, eine tiefere, datengetriebene Grundlage für das Verständnis des vielfältigen Wahrnehmungsraums zu schaffen, in

dem die Teilnehmer sich befinden. Ein künftiges, eventuell KI gesteuertes System könnte Forschern ermöglichen, Muster und Korrelationen zu ermitteln, die ansonsten nicht offensichtlich wären.

Ausführliche Darstellung der Ergebnisse

Ein Arbeitsrahmen wurde fertiggestellt, der es ermöglicht, bestehende neuro-visuelle Erfahrungen in eine selbstgesteuerte Trainings-/Lehr-Applikation zu integrieren.

Im weiteren Verlauf des Projekts wurde deutlich, dass wir mehrere zentrale Probleme zu lösen hatten, um dies zu erreichen:

1. Traditionelle grafische Benutzeroberflächen würden nicht funktionieren, da wir nicht genau wissen können, was die Person sehen kann.

2. Die Menge an Fremdwörtern und die neurowissenschaftliche Komplexität des Schielens ist für neue Teilnehmer überwältigend.

3. Es ist immens wichtig, zu erklärende Inhalte auf das Wesentliche zu reduzieren und so schnell verständlich zu machen.

4. Um unsere experimentellen Ziele zu erreichen, wäre es unerlässlich, eine Struktur zu schaffen, die die Teilnehmer auf deterministischen (aber leicht veränderbaren) Wegen durch die Applikation führt, je nachdem, was sie wahrnehmen.

Um diese Probleme zu überwinden:

1. Wurde ein überwiegend audiogetriebenes Interface ausgewählt, das rein auf einem VR-Headset basiert, ohne zusätzlichen Controller. Interaktion in Form von Auswahlmöglichkeiten erfolgt über einfache Gesten.

2. Es wurde ein erzählerischer Ansatz entwickelt, beginnend mit einer visuellen Metapher von zwei zankenden Schwestern (den Augen), die zum Vehikel für die allmähliche Einführung eines neuen Wortschatzes und eines tieferen Verständnisses wird.

3. Die IDE (Integrated Development Environment) in Unity wurde erweitert, um Inhaltsmodule schnell darstellen zu können, welche sich aus Steuerklassen, Visualisierungsklassen und Audioquellen zusammensetzen. Diese werden in Echtzeit dynamisch geladen, wobei der IDE Filter hinzugefügt werden, damit der Entwickler die Entwicklung effizient auf bestimmte Aspekte des Inhalts zu konzentrieren kann.

4. Eine Kombination aus IDE und API ermöglicht es dem Entwickler, mögliche Wege eines Teilnehmers zu definieren und zu visualisieren. Da es sich um einen dynamischen Prozess handelt, der nicht zeitsicher kompiliert werden kann, wurde ein Mechanismus zur visuellen Überprüfung hinzugefügt, der zeigt, ob alle Pfade vollständig sind und ob alle Inhalte (Bilder/Audio- und dynamisch geladene Klassen) tatsächlich existieren und vor der Kompilierung verfügbar sind.

Ich habe die Funktion des Frameworks validiert, indem ich Inhalte für zwei Testtage erstellt habe. Diese erstellten Inhalte sind das Ergebnis Dutzender Iterationen, in deren Verlauf allmählich

problematische Aspekte aufdeckt wurden, welche wiederum den Prozess der Software-überarbeitung vorantrieben.

Ich habe gelernt, dass Content-Produktionsprozesse komplexer, spezialisierter und zeitaufwendiger sind, als ich es zunächst dachte!

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Im Laufe dieses Projekts habe ich die EyeSkills-Community auf über hundert registrierte Personen erweitert - einige von ihnen haben Ideen, Feedback und Lösungen eingebracht. Ich werde diese Gemeinschaft weiterhin ermutigen, die Verantwortung für die Software zu übernehmen und ihre Entwicklung fortzusetzen.

EyeSkills ist ausserdem dabei, ein großes US-amerikanisches Open-Source-Unternehmen zu gewinnen, das den Aspekt der Datenerfassung unseres Projekts als Vorbild für seine Datenverarbeitungssysteme betrachtet. Das Projekt hat auch die Pläne für die ersten Studien in Zusammenarbeit mit der Abteilung für Entwicklungspsychologie der Universität Gießen konkretisiert.

Der enorme Bedarf an spezialisiertem Wissen hat jedoch gezeigt, dass ein deutlich größerer Mitteleinsatz erforderlich ist, um von einem Proof of Concept zu einem wirklich fertigen und reifen Produkt zu gelangen und gleichzeitig die Marketinginfrastruktur aufzubauen, die erforderlich ist, um die Menschen zu erreichen. Diesem Prozess widme ich mich nun im Anschluss.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Viel Code wurde geschrieben, getestet und nicht in das endgültige Ergebnis aufgenommen - aber dieses Projekt adressiert ein weitgehend unbekanntes Terrain - daher ist es nicht verwunderlich, dass Lösungen durch Versuch und Irrtum gefunden werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Ich betreibe einen Chat-Server unter <https://chat.eyeskills.org>, einen YouTube-Kanal für kleine Demos und Nachrichten und führe eine Mailingliste. Einen Snapshot des aktuellen Framework-Codes habe ich auf bit bucket unter <https://bitbucket.org/eyeskills/eyeskillsathome/src/master/> veröffentlicht.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Im vorherigen Projekt entdeckte ich ernsthafte Probleme mit der Art und Weise, wie Unity (die Engine zur Erstellung der Virtual-Reality-Erfahrungen) mit Übergängen zwischen 3D und 2D umgeht. Diese Probleme entstehen durch die ungewöhnliche Art und Weise, in der ich virtuelle Kameras zur Unterstützung von Augenfehlstellungen einsetzen müssen. Ich war entschlossen, dieses Problem während dieses Projekts zu lösen, aber letztendlich fand ich das Problem im zugrunde liegenden Unity-Framework, auf das ich keinen Zugriff habe. Der wahren Natur des Problems auf die Spur zu kommen, es in reproduzierbare Tests zu zerlegen, Unity das Problem zu melden und einen brauchbaren Weg um das Problem herum zu finden, war ein unendlich frustrierender Prozess, der mich mehrere Wochen schlafloser Nächte kostete.

Ich habe das Niveau an Expertenwissens deutlich unterschätzt, welches für den Einsatz von Animations- und 3D-Modellierungspaketen wie Blender erforderlich ist. Die Lernkurve (Dunning-Krüger) war extrem steil, da ich sah, wie ich hinter meinem Zeitplan herhinkte, aber letztendlich war es mit Expertenratschlägen aus der Gemeinschaft, in der wir einige erfahrene 3D-Künstler haben, möglich, wieder aufzuholen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Es gab und gibt eine gewisse Unsicherheit darüber, ob EyeSkills ein Medizinprodukt ist oder nicht. Meine Kontakte mit den zuständigen Behörden haben die Situation nicht viel klarer gemacht. Dies hat dazu geführt, dass ich mich mit Öffentlichkeitsarbeit und der Veröffentlichung neuer Versionen zurückhalte, bis ich die wahre Rechtslage durchdrungen habe.

Leipzig, 16.09.2019, 

Schlussbericht

Zuwendungsempfänger:
Harald Welte

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S46 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Inhaltsverzeichnis

Aufgabenstellung und Motivation.....	1
Beitrag des Projektes zu den Zielen der Förderinitiative „Software-Sprint“	2
Ausführliche Darstellung der Ergebnisse.....	2
Erweiterung OsmoBTS.....	2
Erweiterung OsmoBSC.....	3
Erstellung eines Cell Broadcast Centers (OsmoCBC).....	3
wireshark.....	4
Zielgruppe, Nutzen und mögliche Weiterentwicklungen.....	4
Arbeiten, die zu keiner Lösung geführt haben.....	5
Präsentationsmöglichkeiten für mögliche Nutzer.....	5
Arbeits- und Kostenplanung.....	6
Ergebnisse bei anderen Stellen.....	6
Glossar.....	7

Aufgabenstellung und Motivation

Es gibt seit ca. einem Jahrzehnt diverse mobilfunkbasierte Notfallwarnsysteme. Beispiele hierfür sind das ETWS (Earthquake and Tsunami Warning System) in Japan, KPAS (Koreana Public Alerting System) in Korea, CMAS/EWA (Wireless Emergency Alert) in den USA, und EU-Alert in der EU.

Diese Systeme dienen der unmittelbaren Benachrichtigung aller Mobilfunknutzer in der geographischen Region einer Gefahrensituation. All diese Systeme verwenden SMSCB (Cell Broadcast SMS) als Transporttechnologie im Mobilfunksystem.

Mit den Osmocom-Projekten haben wir ebenfalls seit knapp 10 Jahren Open Source Implementationen von Mobilfunkinfrastruktur - leider bislang nur mit rudimentärem Support für SMSCB, und keinem Support der darauf aufsetzenden Notfallwarnsysteme.

Die geplante und größtenteils umgesetzte Vorgehensweise wird wie folgt dargestellt:

- Vervollständigung des CBCH (Cell Broadcast Channel) Supports in OsmoBTS + OsmoBSC, automatische Testsuite dazu

- CBSP (Cell Broadcast Service Protocol) Protokoll encoding/decoding funktionen implementieren
- OsmoBSC erweitern, Funktionen zur Verwaltung, Routing und Scheduling von SMSCB einbauen
- Libraryfunktionen für Encoding von PWS, CMAS/WEA und ETWS-Nachrichten
- Neues OsmoCBC Programm (Cell Broadcast Centre) als zentrale Instanz zur Verteilung/Verwaltung der SMSCB
 - CBSP Protokoll zu OsmoBSC
 - SABP (Service Area Broadcast Protocol) zum RNC bzw. OsmoHNBGW
 - REST interface zum User Interface
- WebUI fuer OsmoCBC zum erfassen/schedulen/löschen der Nachrichten
- OsmoHNBGW: Verteilung von SABP Nachrichten vom CBC zu den einzelnen hNodeB

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

Systeme, die im Notfall die Bevölkerung in Sekundenschnelle alarmieren können sind im Bereich „Civic Tech“ Teil der Förderinitiative „Software-Sprint“ angesiedelt.

Ausführliche Darstellung der Ergebnisse

Erweiterung OsmoBTS

Die bereits vorab existierende und vom Autor seit 2011 entwickelte Software OsmoBTS hatte bislang nur extrem rudimentäre und nicht kontinuierlich getestete und gepflegte Unterstützung für SMSCB, sowie gar keine Unterstützung für Notfallwarnsysteme (nachfolgend verallgemeinernd als ETWS bezeichnet). Im Rahmen dieses Projektes wurden folgende Ergebnisse erzielt:

- vollständige Unterstützung für das Aussenden von SMSCB über CBCH BASIC und CBCH EXTENDED, sowohl auf den Kanalkombinationen SDCCH/4 und SDCCH/8
- SMSCB Flow Control mittels CBCH LOAD INDICATIONS über RSL
- Erweiterung des RSL-Protokolls zur Entgegennahme von ETWS Primary Notification vom BSC
- Weiterleitung von ETWS Primary Notifications an die PCU zwecks Übermittlung an Telefone mit aktivem TBF.
- Implementation des Encodierens, Segmentierens und Aussenden von ETWS Primary Notification über P1 Rest Octets auf dem PCH

- In TTCN-3¹ entwickelte Testsuite zur kontinuierlichen Verifikation der SMSCB- und ETWS-Funktionen in OsmoBSC

Alle Ergebnisse sind in den offiziellen Hauptentwicklungszweig der OsmoBTS-Software aufgenommen worden.

Erweiterung OsmoBSC

Die in 2008 durch den Autor initiierte Software OsmoBSC implementiert die Funktionen eines Base Station Controllers im GSM Netzwerk. Vor Beginn des geförderten Projekts waren keine Standardkonformen Funktionen für SMSCB vorhanden, es gab lediglich die als proof-of-concept gedachte Möglichkeit, einen vom Anwender händisch encodierte SMSCB-Nachricht als Hexdump auf der Kommandozeile zu pasten, die dann an eine BTS geschickt wurde.

Im Rahmen des geförderten Projekts wurden folgende Erweiterungen zu OsmoBSC implementiert:

- Implementation eines Encoders und Decoders für das CBSP-Protokoll gemäß 3GPP TS 48.049 als Teil der Bibliothek libosmocore
- Implementation der Server- und Client-Rolle des CBSP-Protokolls
- Implementation der Funktionen zur Entgegennahme, Verteilung, Routing, Verwaltung und Übertragungswiederholung von SMSCB
- Implementation der BSC-Seite des SMSCB Flow Controls mittels RSL CBCH LOAD INDICATION
- Implementation eines Encoders und Decoders für das CBSP-Protokoll gemäß 3GPP TS 48.049 in der Programmiersprache TTCN-3
- Übertragung von ETWS Primary Notifications an alle Telefone mit aktivem dedizierten Kanal
- Übertragung von ETWS Primary Notifications an alle relevanten BTS über RSL
- In TTCN-3 entwickelte Testsuite zur kontinuierlichen Verifikation der SMSCB- und ETWS-Funktionen in OsmoBSC

Alle Ergebnisse sind in den offiziellen Hauptentwicklungszweig der OsmoBTS-Software aufgenommen worden.

Erstellung eines Cell Broadcast Centers (OsmoCBC)

Das CBC (Cell Broadcast Center) ist die Zentrale Instanz zur Verwaltung des Cell Broadcast Services und Warnungen (ETWS, PWS, CMAS, WEA) in einem Mobilfunknetz. Vor Beginn des geförderten Projekts gab es keine Open Source Implementation in diesem Bereich.

Im Rahmen des geförderten Projekts wurde ein CBC (genannt OsmoCBC) mit folgenden Funktionen von grund auf neu entwickelt:

¹ <https://www.ttcn-3.org/>

- Implementation eines REST/JSON basierten web-interfaces zur Einstellung, Verwaltung und Löschung von SMSCB und ETWS
- Implementation von Client und Server für CBSP zur Verbindung mit den BSCs.
- Verteilung der SMSCB und ETWS an die BSCs

Die folgenden, ursprünglich angedachten Ziele konnten nicht in dem geförderten Zeitraum umgesetzt werden:

- Implementation des SABP-Protokolls zur Weiterleitung von SMSCB und ETWS an RNC
- Implementation eines Web-Interfaces zur einfachen Einstellung von Nachrichten über das REST/JSON Interface

Der Autor hat vor, diese Funktionen im Nachgang auch ohne weitere Förderung zu implementieren

wireshark

In verteilten Systemen wie den Mobilfunknetzen ist es zur Entwicklung und zur Fehlersuche unabdinglich, die Kommunikation zwischen den Elementen beobachten zu können.

Netzwerkprotokolle werden für gewöhnlich mit dem weit verbreiteten Open Source Werkzeug wireshark² analysiert. Während der Arbeiten am geförderten Projekt wurden fehlende Funktionen bzw. Bugs in Bezug auf SMSCB/ETWS gefunden:

- Kein Dissector für das CBSP-Protokoll
- fehlerhafte Dekodierung von ETWS Primary Notification im GSM RR Dissector
- kein Dissector für ETWS PN über RSL

Diese fehlenden Funktionen wurden durch den Autor ergänzt bzw. Bugs behoben, und beim wireshark-Projekt eingereicht^{3,4,5} und von diesem in den offiziellen Hauptentwicklungszweig Aufgenommen. Dies ermöglicht jedermann, Protokollanalyse bzgl. der SMSCB/ETWS-Kommunikation zwischen den einzelnen Netzwerkelementen durchzuführen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Direkte Zielgruppe sind Betreiber von Osmocom-Basierten Mobilfunknetzen. Dies sind bislang vor allem kleine Netzbetreiber in ländlichen Regionen von Entwicklungs- bzw. Schwellenländern (siehe z.B. Rhizomatica⁶ bzw. TIC⁷).

Indirekt geht es darum, dass die Nutzer solcher Mobilfunknetze in Gefahrensituationen alarmiert werden können. Da diese Netze in diesen Regionen oft die einzige Kommunikationstechnologie ist,

² <https://wireshark.org/>

³ <https://code.wireshark.org/review/#/c/34465/>

⁴ <https://code.wireshark.org/review/#/c/34464/>

⁵ <https://code.wireshark.org/review/#/c/29745/>

⁶ Rhizomatica, <https://rhizomatica.org/>

⁷ TIC (Telecomunicaciones Indígenas Comunitarias), <https://www.tic-ac.org/>

ist eine alternative Alarmierung (abgesehen von Sirenen o.ä.) nicht möglich. Gefahren sind hier nicht nur klassische Naturkatastrophen, sondern z.B. kriminelle Übergriffe, Brand, ...

Da die Schnittstellen zwischen den Netzelementen im Mobilfunk durch die 3GPP standardisiert sind, wäre allerdings auch jeder andere Netzanbieter teil der möglichen erweiterten Zielgruppe. Er kann einzelne Komponenten, wie z.B. OsmoBTS, OsmoBSC oder OsmoCBC auch einzeln verwenden, und diese in seine Netzinfrastruktur integrieren.

Als mögliche und sehr wahrscheinliche Weiterentwicklungen sind folgende Bereiche erkennbar:

- Fertigstellung der SABP- und damit 3G-Unterstützung
- Erweiterung des OsmoCBC um eine SGs-Schnittstelle zwecks Unterstützung von 4G/LTE
- Erweiterung des OsmoCBC um das Common Alerting Protocol (CAP)⁸, einem international inzwischen auch von der ITU⁹ angenommenen Standard zur Übertragung von Notfallwarnungen z.B. von einer Behörde zu einem CBC wie OsmoCBC

Arbeiten, die zu keiner Lösung geführt haben

Das SABP-Protokoll wird zwischen dem CBC und den RNCs im 3G-Netzwerk verwendet, und sollte ursprünglich mit als Teil des Projekts implementiert werden. Leider wurde die Komplexität unterschätzt, u.a. deshalb, weil dieses Nachrichtenbasierte Protokoll nicht wie sonst üblich mittels SCTP übertragen wird, sondern im Stream-Orientierten TCP. TCP kennt keine Nachrichtengrenzen, so dass der Anfang bzw. die Länge einer SABP-Nachricht bei der Übertragung nicht übermittelt werden. Ohne ein explizites Längensfeld zu Beginn jeder Nachricht ist es relativ schwierig, herauszufinden wo in dem TCP-Datenstrom Nachrichten anfangen und enden.

Ein detaillierteres Studium der SABP-Spezifikation hätte in der Vorbereitung den erhöhten Zeitbedarf aufzeigen können.

Im geförderten Projekt wurde deshalb leider nur eine unvollständige SABP-Implementation erstellt, die nicht in OsmoCBC oder die entsprechende Testsuite integriert wurde. Der Autor beabsichtigt, die fehlenden Teile auch nach Ende der Förderung noch zu implementieren.

Für die primäre Nutzergruppe der community-basierten Mobilfunknetze in Entwicklungs-/Schwellenländern hat das Fehlen dieser Funktion keine unmittelbaren Nachteile. Alle dem Autor bekannten derzeitigen Netze dieser Art benutzen keinen 3G-Standard.

Präsentationsmöglichkeiten für mögliche Nutzer

Mögliche Nutzer können sich über die Projektergebnisse wie folgt informieren bzw. diese nutzen:

- Projekthomepage OsmoCBC: <https://osmocom.org/projects/osmo-cbc/wiki> sowie den verwandten Seiten:

⁸ <http://docs.oasis-open.org/emergency/cap/v1.2/CAP-v1.2-os.html>

⁹ International Telecommunications Union, <https://itu.int/>

- Osmocom-Wiki zu Cell Broadcast: https://osmocom.org/projects/cellular-infrastructure/wiki/Cell_Broadcast
- Osmocom-Wiki zu Emergency Warning: https://osmocom.org/projects/cellular-infrastructure/wiki/Emergency_Warning
- Videoaufzeichnung Vortrag „Production-Grade Cell Broadcast for Osmocom“: <https://media.ccc.de/v/osmodevcon2019-107-production-grade-cell-broadcast-for-osmocom>
- Quellcode zu OsmoBTS: <http://git.osmocom.org/osmo-bts/>
- Quellcode zu OsmoBSC: <http://git.osmocom.org/osmo-bsc/>
- Quellcode zu OsmoCBC: <http://git.osmocom.org/osmo-cbc/>
- TTCN-3 Testsuites dazu: http://git.osmocom.org/osmo-ttcn3-hacks/tree/bts/BTS_Tests_SMSCB.ttcn sowie http://git.osmocom.org/osmo-ttcn3-hacks/tree/bsc/BSC_Tests_CBSP.ttcn

Arbeits- und Kostenplanung

Der Projektbeginn hat sich aus privaten Gründen zeitlich deutlich verzögert, da sich der Autor unerwartet um die häusliche Pflege eines Familienmitglieds kümmern musste. Auch während der Arbeitsphase an dem Projekt war es aufgrund der vielseitigen Verpflichtungen (Gesellschafter-Geschäftsführer-Tätigkeit; Projektleitung des Osmocom-Projektes) eine erhebliche zusätzliche Belastung, regelmässig die geplante Zeit mit dem geförderten Projekt zu verbringen.

Ergebnisse bei anderen Stellen

Zu Projektbeginn oder zumindest zum Zeitpunkt der Antragstellung gab es keine anderen Open Source Projekte im Mobilfunkumfeld, die Funktionen im Bereich der Notfallwarnung implementiert hatten.

Inzwischen hat sich dies geändert: Das Projekt nextepc des Koreanischen Entwicklers Sukchan Lee, welches einen EPC (Evolved Packet Core, das LTE/4G Kernnetz) implementiert, hat die standardisierte SGs-Schnittstelle in seiner MME (Mobility Management) ergänzt.

Die SGs-Schnittstelle erlaubt es dem CBC, Notfallwarnungen über das LTE-Netz zu versenden. Insofern wäre die angestrebte Erweiterung von osmo-cbc um die SGs-Schnittstelle eine ideale Ergänzung.

Glossar

BSC	Base Station Controller; Komponente des 2G Radio Access Networks
BTS	Base Transceiver Station; Mobilfunk-Basisstation im 2G-Netz
CBC	Cell Broadcast Centre; zentrale Instanz für den Cell Broadcast Dienst
CBCH	Cell Broadcast Channel; Logischer Kanal im GSM; 3GPP TS
CBS	Cell Broadcast Service; 3GPP TS 23.041
CBSP	Cell Broadcast Service Protocol; zwischen CBC und BSC; 3GPP TS 48.049
EPC	Evolved Packet Core; Kernnetz des 4G/LTE Standards
ETWS	Earthquake and Tsunami Warning System
JSON	JavaScript Object Notation; https://www.json.org/
KPAS	Korean Public Alerting System; Koreanisches System zur Notrufwarnung
MME	Mobility Management Entity; Komponente des EPC
OsmoBSC	Osmocom BSC Software; https://osmocom.org/projects/osmobsc/wiki
OsmoBTS	Osmocom BTS Software; https://osmocom.org/projects/osmobts/wiki
Osmocom	Open Source Mobile Communications; https://osmocom.org/
OsmoPCU	Osmocom PCU Software; https://osmocom.org/projects/osmopcu/wiki
PCH	Paging Channel; Logischer Kanal im GSM
REST	Representational State Transfer; Programmierparadigma für Webservices
RNC	Radio Network Controller; Komponente des 3G Radio Access Networks
RR	Radio Resource; sub-layer des Layers 3 in GSM; 3GPP TS 48.018
RSL	Radio Signaling Link; zwischen BSC und BTS; 3GPP TS 48.058
SABP	Service Area Broadcast Protocol; zwischen CBC und RNC
SCTP	Stream Control Transfer Protocol; IETF RFC 2960 + RFC 4960
SDCCH	Slow Dedicated Control Channel; Logischer Kanal im GSM
TCP	Transmission Control Protocol;
WEA	Wireless Emergency Alert; US-Amerikanisches System zur Notrufwarnung
CAP	Common Alerting Protocol; OASIS und ITU-Standard für Notfallmeldungen
TTCN-3	Testing and Control Notation; domainspezifische Programmiersprache zum Testen von kommunikationsbasierten Systemen
SMSCB	Short Message Service Cell Broadcast; 3GPP TS 23.041

Richtlinie zum „Software-Sprint“

Crossfoam - Browser-Plugin zur Analyse und Visualisierung von Filterbubbles mit Hilfe von maschinellem Lernen

Schlussbericht

Zuwendungsempfänger:

ULTRAPOP - Sebastian Meier und Daniel Lommes Gbr

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S47 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Im Gegensatz zum Begriff der Öffentlichkeit, in dem ideal-typisch alle sprechen und von allen gehört werden können, werden in Filterblasen homogene Meinungen einem homogenen Publikum vorgetragen. Die fortschreitende Verschiebung des Nachrichtenkonsum in Soziale Medien, gepaart mit deren algorithmischer Kuratierung, führt zum weitgehenden Verblenden der ersten, während tausende Filterblasen blühen. Im Gegensatz zu Ansätzen der Zensur und/oder der Priorisierung „klassischer“ Medienunternehmen versuchen wir Menschen zu sensibilisieren, und ihnen ein Werkzeug an die Hand zu geben, um Filterblasen zu erkennen und Inhalte in diesen verorten zu können. Dadurch wird Data Literacy aktiv gefördert.

Aus Datenschutzgründen wollen wir versuchen die Anwendung soweit wie möglich ohne Cloud-Infrastrukturen aufzubauen. Ziel ist es eine Client-seitige Anwendung zu entwickeln, welche im Browser der User*in läuft. Die Wahl eines Browserplugins ist nicht nur technologisch motiviert, sondern zielt darauf ab die Nutzung für möglichst viele Menschen so niederschwellig wie möglich zu machen. Das Plugin selber nutzt JavaScript als primäre Programmiersprache. Das Projekt wird sich in der ersten Phase auf Twitter konzentrieren und darauf aufbauend für andere Netzwerke erweitert werden. Für Twitter nutzen wir die API um Nutzer*innen-Netzwerke und deren Inhalte zu erschließen. Ein datenbank-ähnliches System auf Seiten der Nutzer*innen dient als Cache der Informationen. Diese können dann über Visualisierungen und Erklärende Darstellungen erkundet werden.

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

Die Zielgruppe unserer Anwendung sind Nutzer*innen sozialer Medien. Speziell Nutzer*innen welche soziale Medien für den Konsum von Nachrichten nutzen. In Bezug auf diese Zielgruppen, versucht unser Projekt nicht nur ein Werkzeug zur Verfügung zu stellen, sondern durch die Sensibilisierung für das Thema der Filterblasen, die Nutzer*innen zur digitale Selbstermächtigung zu befähigen. Wir nehmen durch diesen Fokus direkt Bezug auf das Themenspektrum B der Data-Literacy. Das Tool selber nutzt verschiedene Methoden der Datenvisualisierung und des daten-gestützten Storytellings und nimmt somit auch Bezug auf das Themenspektrum A.

Ausführliche Darstellung der Ergebnisse

Vorgehen:

Im Rahmen der Förderung konnten wir erfolgreich die anvisierte Browser-Erweiterung umsetzen. Im ersten Projektabschnitt haben wir die Basisinfrastruktur der Erweiterung entwickelt. Hierbei lag die Herausforderung darauf die Erweiterung von Anfang an für mehrere Browser-Engines zu entwickeln. Hilfreich hierbei war der von Mozilla bereitgestellte Polyfill für Browser-Erweiterungen. Nichts desto trotz stellte uns die Cross-Browser-Compatibility vor große Probleme, z.B. in Hinblick auf das Styling oder die Datenspeicherung. Bei der Datenspeicherung mussten wir auch von der vorgesehenen SQLite Datenbank abweichen und statt dessen einen simplen Key-Value-Storage nutzen. Um diesen speicher- und performanzeffizient nutzen zu können, mussten wir unsere Informationsarchitektur entsprechend anpassen. In der zweiten Projektphase wurde das client-seitige Daten-Harvesting entwickelt. In dieser ersten Stufe der Erweiterung wurde dies für Twitter entwickelt.

Die dritte Phase konzentrierte sich auf die Visualisierung der in Phase zwei gesammelten Daten. Mit Hilfe der JavaScript Bibliothek D3 wurden eine Reihe an Visualisierungen entwickelt, welche den Nutzer*innen erlauben die gesammelten Daten visuell zu explorieren.

Die vierte Phase konzentrierte sich, anstatt des vorgesehenen Austausches über ein offenes Repository, auf die Optimierung der Visualisierungen. Im offenen Repository sahen wir zu viele potentielle Gefahren, gerade für nicht technisch versierte Nutzer*innen. Statt dessen konzentrierten wir uns auf die Optimierung der Visualisierungen. Speziell große Netzwerke führten die Performanz der Browser an seine Grenzen. Deshalb entwickelt wir in der vierten Phase ein auf Web-GL aufgebautes System, welches die D3-Visualisierungen optimierte.

Den letzten 5. Milestone konnten wir zeitlich nicht ganz erreichen, da das Veröffentlichen der Erweiterungen in den jeweiligen Browser-Shops doch länger dauerte als angenommen. Wir hoffen, dass dies in den nächsten Wochen abgeschlossen werden kann und das Projekt somit seinen vorerst letzten Milestone erreicht.

Ethisches Design & Entwickeln und Nutzertests:

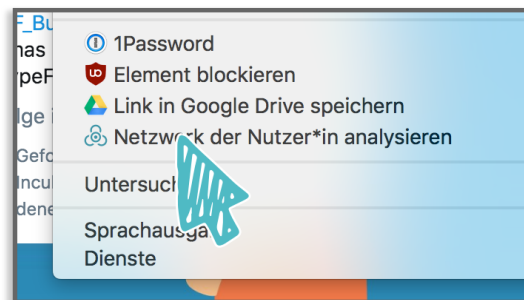
Über alle Phasen hinweg, haben wir auf mehreren Metaebenen das Projekt überwacht. Zum einen haben wir mehrere Evaluationen mit Nutzer*innen durchgeführt, um frühzeitig Probleme des Systems zu erkennen und eine agile und nachhaltige Umsetzung zu gewährleisten. Hierüber haben wir z.B. festgestellt, dass die lange initiale Wartezeit von Nutzer*innen nicht besonders gut aufgenommen wurden. Deshalb wird die Erweiterung nun mit einem voroptimierten Datensatz ausgeliefert, welcher von den Nutzer*innen sofort exploriert werden kann. Zusätzlich gibt es eine progressive Visualisierung, welche den Benutzer*innen einen visuellen Status gibt, während die Daten weiter gesammelt werden. Eine weitere Erkenntnis aus den Tests war auch, dass Nutzer*innen

das Werkzeug nicht nur zur Erforschung ihrer eigenen Netzwerke in Bezug auf Filterblasen nutzen, sondern, dass sich das Werkzeug auch eignet, um das Netzwerk zu optimieren und aufzuräumen. Auf einer weiteren Ebene haben wir den Entwicklungsprozess aus einer ethischen Perspektive überwacht. Mit der Entwicklung jedes neuen Moduls haben wir versucht, zu analysieren in wie weit durch dieses neue Module nachhaltige Effekte für potentielle Nutzer*innen entstehen können.

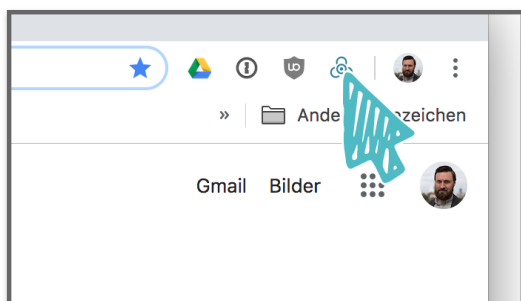
Wie funktioniert Crossfoam:



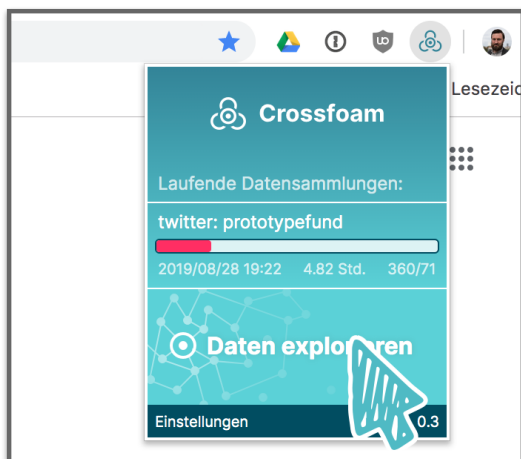
Nutzer*innen können eine beliebige Nutzer*in auf Twitter auswählen...



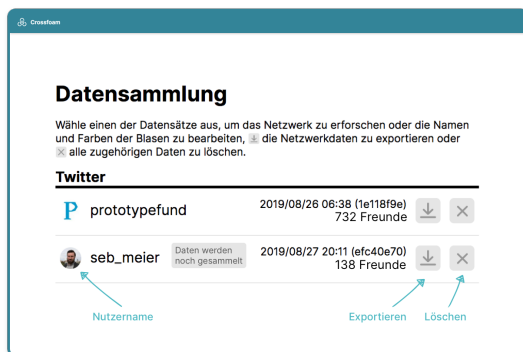
.. und direkt im Browser über die Erweiterung Crossfoam analysieren lassen.



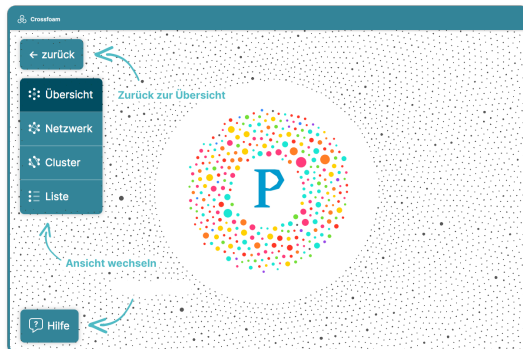
Damit Nutzer*innen sich über die Filterblasen informieren können. Müssen die zugehörigen Daten erst gesammelt werden.



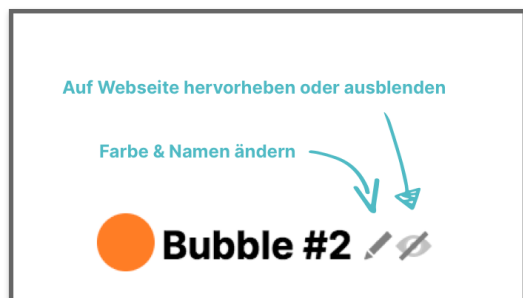
Den Status dieser Datensammlung kann von der Nutzer*in jeder Zeit bequem über das Erweiterungs Menü abgerufen werden.



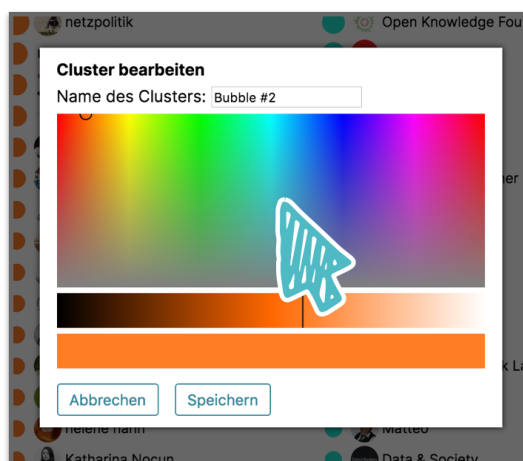
Auf der Übersichtsseite der Erweiterung werden alle Datensammlungen angezeigt, sowie deren Status. Von hier können die Daten exportiert werden oder direkt in der Erweiterung visualisiert werden.



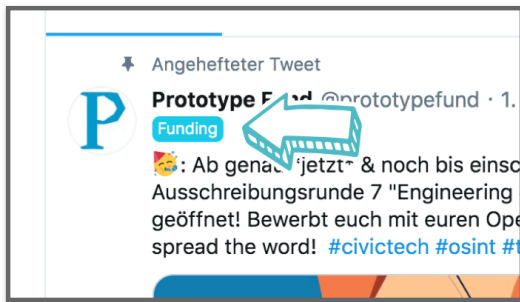
Alle Visualisierungen haben eine Hilfefunktion. Über diese werden Nutzer*innen durch die Funktionalität der Visualisierung geleitet.



Die vom System erkannten Blasen können von der Nutzer*in angepasst werden um den persönlichen Wünschen zu entsprechen.

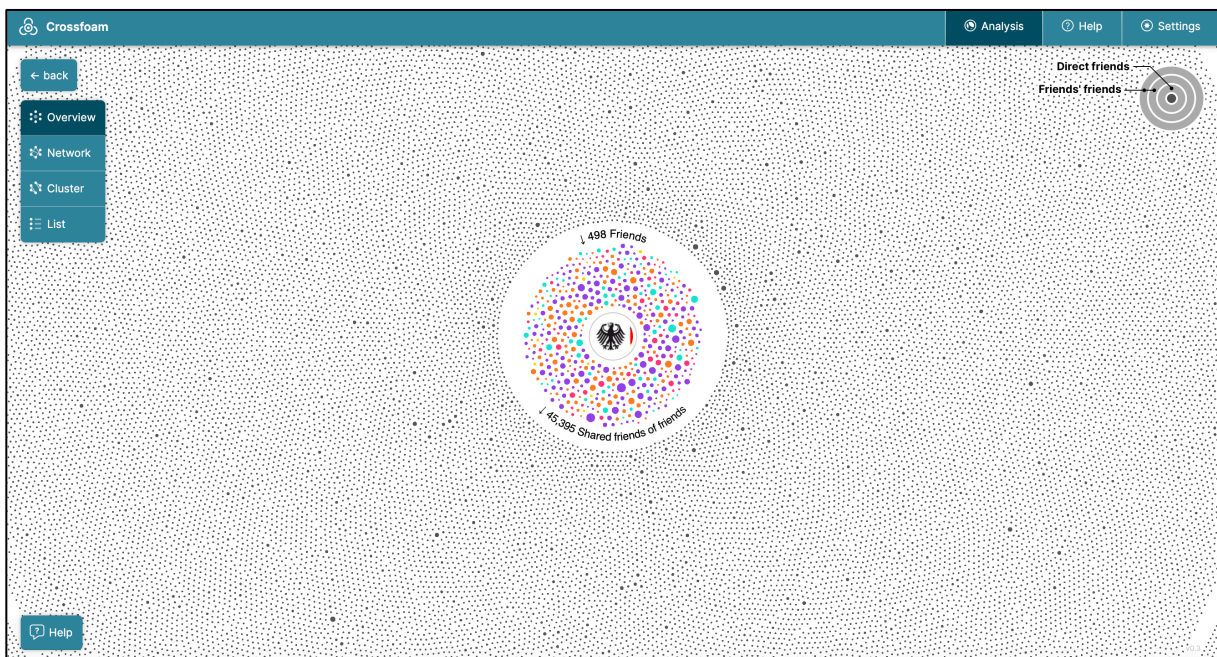


Nicht nur die Beschriftung auch die Farbe kann entsprechend angepasst werden.

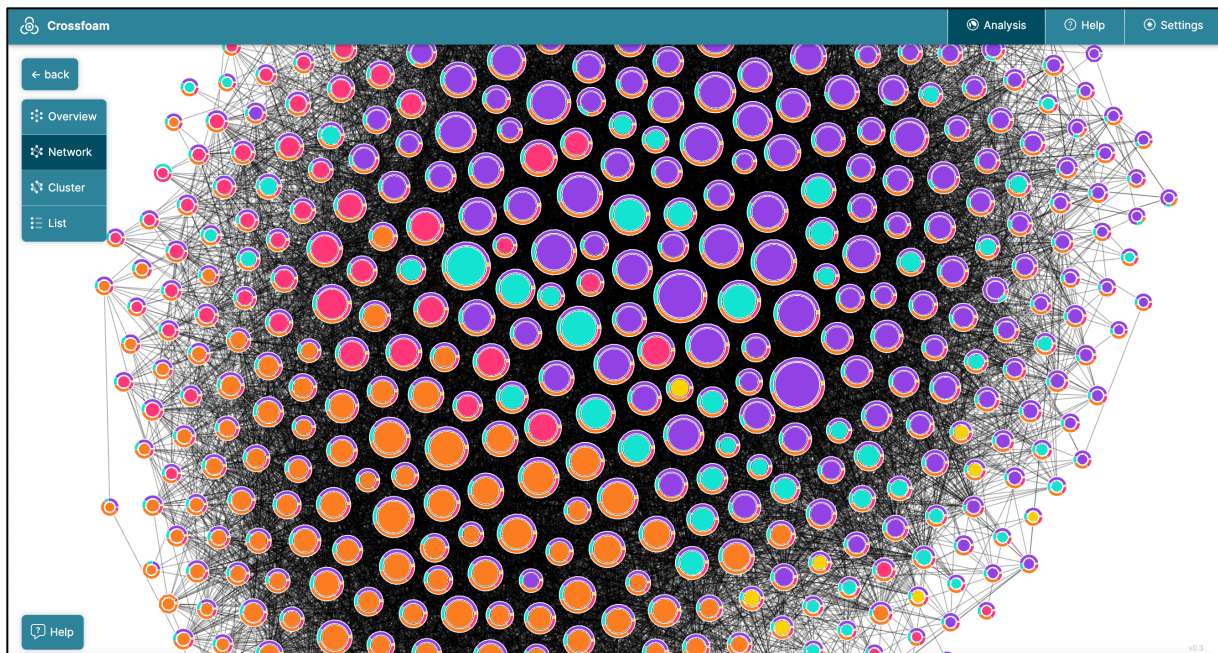


Wenn die Nutzer*in durch das soziale Netzwerk oder andere Seiten surft, werden erkannte Personen, mit entsprechenden Labels versehen. So lässt sich schnell erkennen zu welchen Blasen eine bestimmte Nutzer*in gehört.

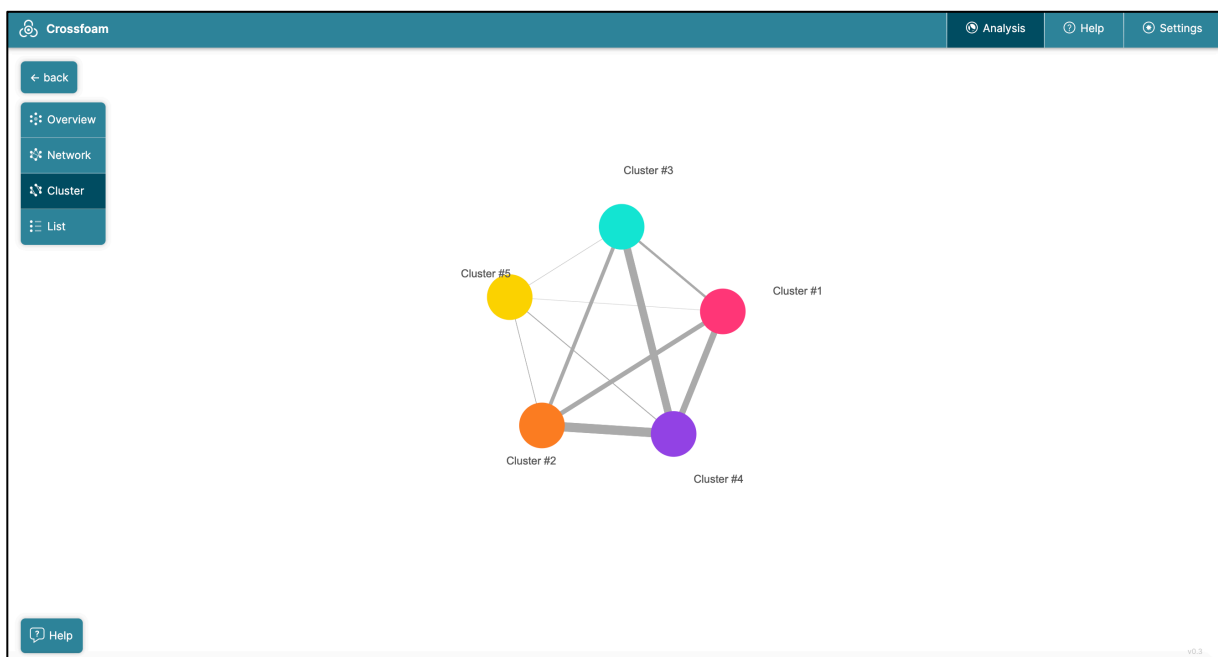
Visualisierungen:



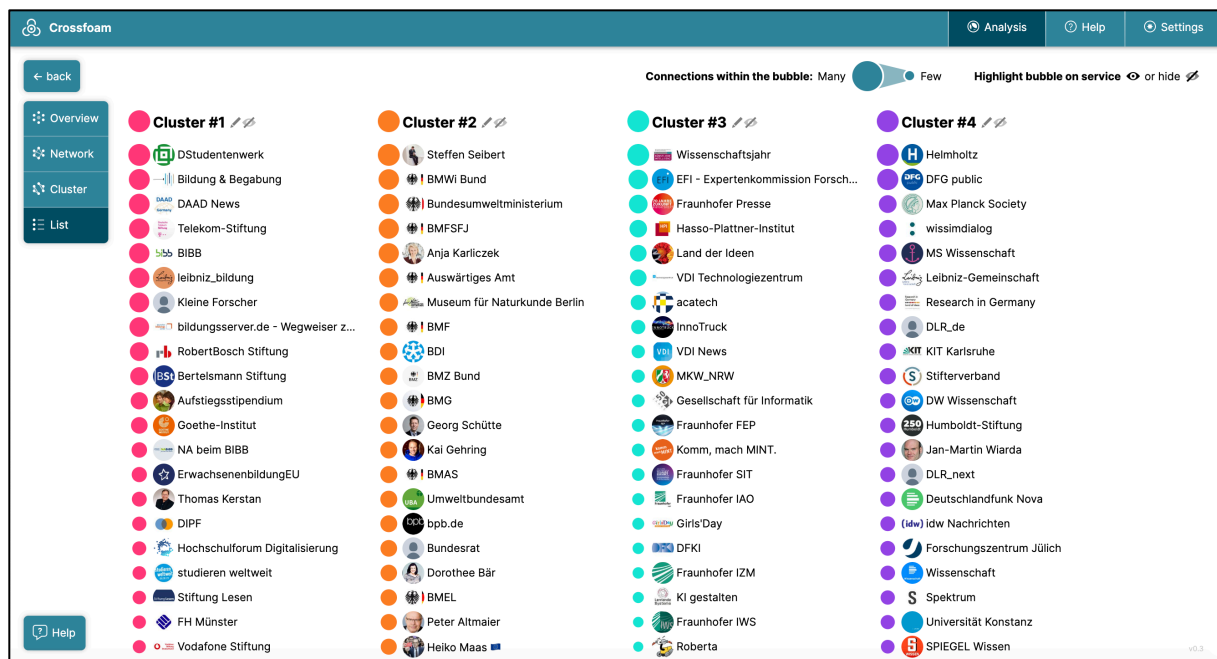
Die Übersichtsvisualisierung vermittelt den Nutzer*innen einen Eindruck über die Größe ihres primären und sekundären Netzwerks. Die Visualisierung verschafft einen ersten Eindruck, welche Personen einen besonders großen Einfluss auf die Informationen haben, welche der Nutzer*in in ihrem Netzwerk angezeigt werden.



Die Netzwerkansicht gibt einen Einblick in die Filterblasen. In dieser Ansicht werden nicht nur die zentralen Akteur*innen der Blasen sichtbar sondern auch, dass Nutzer*innen zwar einer Blase primär zugeordnet werden, aber durchaus Überlappungen zwischen einzelnen Blasen bestehen.



In der Filterblasenansicht liegt der Fokus auf den Blasen selber. Neben den Überlappungen, welche schon in der Netzwerkansicht sichtbar wurden, kann die Nutzer*in in die einzelnen Netze eintauchen und dort tieferegehende Analysen durchführen.



Die Listenansicht gibt der Nutzer*in einen schnellen Überblick über alle Personen in Ihrem Netzwerk und deren Blasenzugehörigkeit. Von hier aus können Nutzer*innen auch die Blasen anpassen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Mit Hilfe der Browser-Erweiterung können Nutzer*innen sich über ihre persönlichen Sozialen Netzwerke informieren. Den Nutzer*innen werden so Einblicke in die Hetero- bzw. Homogenität ihrer Informationssphären gegeben. Nutzer*innen sollen durch die Interaktion mit den eigenen Daten für die Thematik der Filterblasen sensibilisiert werden. Aktuell ist die Erweiterung nur auf Twitter anwendbar. Bei der Entwicklung wurde aber bereits darauf geachtet, dass alles modular und flexibel programmiert wurde, sodass weitere Services zukünftig ergänzt werden können. Im nächsten Schritt soll die offene Plattform Mastodon ergänzt werden.

Die unter offenen Lizenzen veröffentlichten Module sollen andere Entwickler*innen dazu befähigen, ebenfalls privatsphären-zentrierte Anwendungen zu entwickeln, welche auch komplexe und langwierige Prozesse auf den Systemen der Nutzer*innen durchführen. Zu diesem Zweck haben wir z.B. ein spezielles Queue-Management Modul entwickelt, welches auch über den Verlust der Internetverbindung oder den Neustart des Rechners hinweg einen Ablauf an Aufgaben abarbeiten kann.

Während wir in der Vergangenheit schon diverse web-basierte Anwendungen konzipiert und umgesetzt haben, war die Anwendung dieser Kenntnisse und Fähigkeiten auf die Entwicklung einer Browser-Erweiterung eine neue Herausforderung.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

In der initialen Projektbeschreibung war vorgesehen, dass Methoden des Natural-Language-Processing (NLP) in Kombination mit Ansätzen des maschinellen Lernen eingesetzt werden. Gleichzeitig haben wir versucht, uns an den Prinzipien des ethischen Designs zu orientieren und ethische Konsequenzen der entwickelten Technologien zu antizipieren. Während der Implementierung von algorithmischen Vergleichen von Textähnlichkeiten, innerhalb nutzer*innen-

spezifischer sozialer Netzwerke, wurde deutlich, dass dies negative Effekte haben könnte. Ein Problem sozialer Netzwerke ist die Verfolgung von einzelnen Individuen. Um dieser Verfolgung entkommen zu können, löschen Nutzer*innen ihre Accounts und erstellen unter einem neuen Pseudonym einen neuen Account. Über die automatisierte Analyse von Textähnlichkeiten über eine große Anzahl an Accounts hinweg, wäre es mit großer Wahrscheinlichkeit möglich die neuen Pseudonyme zu identifizieren. Als Alternative, um weitere Erkenntnisse über soziale Netzwerke zu ermöglichen, wurden weitere Netzwerkanalysefunktionen implementiert.

Außerdem haben wir schlussendlich die Verwaltung der Service-Anbindungen nicht über ein öffentliches Repository geregelt, sondern die Services direkt in das Plugin verbundet. Da die Module für die Service-Anbindung direkt mit Internet kommunizieren müssen, war uns die Gefahr zu groß, das Nutzer*innen sich Schadcode installieren. Deshalb werden etwaige Service-Erweiterungen von uns nun erst evaluiert und dann in das Bundle integriert.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

- Es gibt eine Projektwebsite: <http://www.ultrapop.de>
- Der Code befindet sich in einem GitHub-Repository: <https://github.com/ultrapop-de>
- Die Browser-Erweiterung wurde in den Erweiterungs-Store für Firefox & Chrome geladen und wird dort aktuell noch überprüft und wird bald verfügbar sein.
- Außerdem schreiben wir aktuell einen Artikel für den Launch

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die initial aufgestellte Kostenplanung konnte eingehalten werden. In der Arbeitsplanung wurden die Pakete des NLP in die Netzwerkanalyse überführt.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Seid der Beantragung des Projektes hat Microsoft angekündigt, dass der Edge Browser zukünftig die Chromium Engine nutzen wird. Dank dieser Entwicklung, konnten wir die Browser-Erweiterung auch für den Edge Browser verfügbar machen.

Richtlinie zum „Software-Sprint“

AndroidOpenPush – Implementierung eines Open-Source Push services für Android Apps

Schlussbericht

Zuwendungsempfänger:

Marcus Hoffmann

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS17S48 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)?

Ich bin Kern-Entwickler und Maintainer beim F-Droid Projekt, dessen Ziel es ist, einen vollständig freien App-Store für Android bereitzustellen. Viele Entwickler von Open-Source Anwendungen sind gerne bereit ihre App neben dem Google Play Store auch über F-Droid zur Verfügung zu stellen. Da F-Droid nur freie Software erlaubt, sind gelegentlich Änderungen am Quelltext der App notwendig um sie in F-Droid zu integrieren. Die einzige Komponente, die bei der F-Droid Version dabei meist keinen geeigneten Ersatz findet sind Push-Benachrichtigungen. Diese sind meist ausschließlich über Googles closed-source Softwarebibliotheken (Firebase Cloud Messaging) realisierbar.

Push-Benachrichtigungen funktionieren, indem der Client (das Smartphone) eine langlebige Verbindung zu einem Server aufbaut und diese durch das gelegentliche senden von 'Keepalive' Paketen aufrecht erhält. Idealerweise sind diese Keepalives nur im Bereich von mehreren 10 Minuten notwendig um den Akku des mobilen Geräts zu schonen.

Der Push-Server ordnet einzelne aktive Verbindungen den Clients zu und schickt bei Bedarf über die offene Verbindung eine Benachrichtigung an den Client. Das Gerät wird durch die eingehende Datenpakete aus dem Sleep-Mode geweckt.

Im Rahmen des Projekts wurde eine Android Client-App zusammen mit einer Serverkomponente entwickelt.

Die Android App kümmert sich um das Aufrechterhalten der Verbindung und Registrierung des Clients. Die Serverkomponente stellt eine Schnittstelle für Webservices bereit um registrierten mobilen Endgeräten Push Nachrichten senden zu können.

Als Protokoll zwischen Client und Push Server sollten COAP und MQTT evaluiert werden.

Die Entwicklung Schnittstellen (Client-Lib und Push-Server API) sollte in Anlehnung an Google's FCM Dienste erfolgen.

Als Meilensteine dienen die Evaluierung eines Protokolls, die Festlegung der Pushserver APIs, die Implementierung des Pushservers und die Implementierung der Client-App.

Als optionaler Meilenstein wurde die Integration in eine bestehende App definiert.

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 freie Softwareprojekte, die auf mobilen Plattformen Push-Benachrichtigungen benötigen. Beispiele solcher Projekte sind RocketChat, Matrix und Nextcloud.

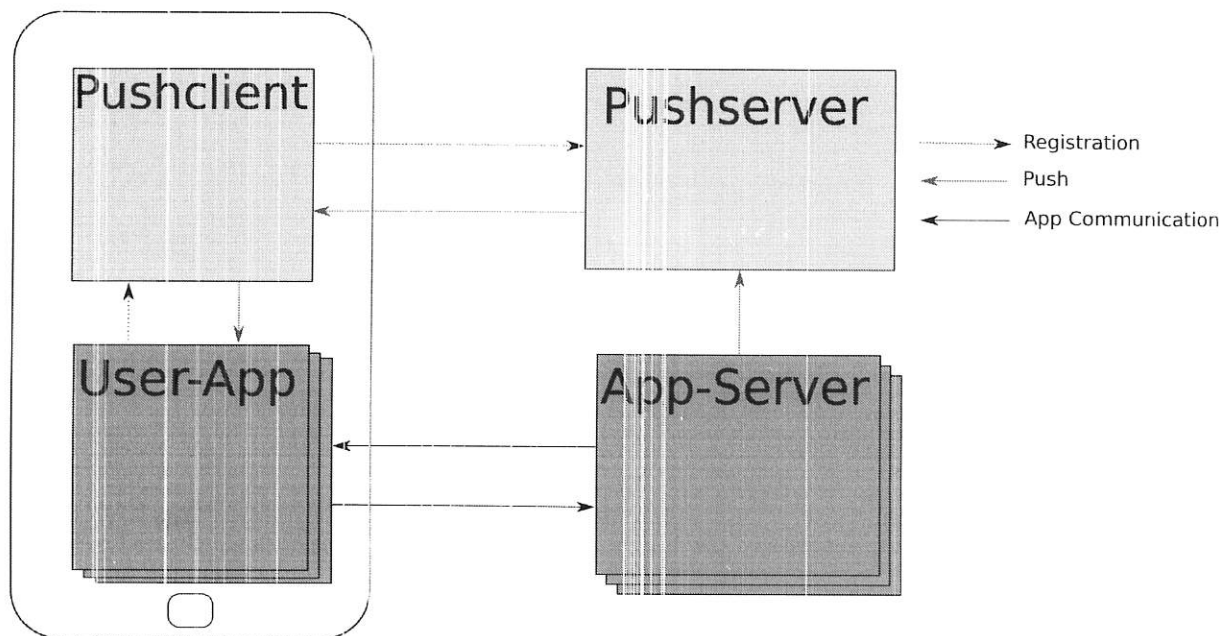
Bisher ist die einzige Möglichkeit auf einfache Weise Push-Benachrichtigungen für Mobile Apps zu senden die Nutzung von entweder Google's oder Apples Push Diensten. Diese erfordern einerseits die Nutzung von proprietären Systemdiensten und Bibliotheken und andererseits erlauben sie Google (oder Apple) das Sammeln von allen Metadaten für alle Push Nachrichten. Selbst wenn die App den Inhalt der gesendeten Nachricht verschlüsselt kann man über die Analyse der Metadaten Rückschlüsse auf das Kommunikationsverhalten von Nutzern ziehen. Außerdem macht es alle Apps abhängig von Googles Infrastruktur und Terms of Services.

Mein Projekt ist somit wohl dem Förderbereich Infrastruktur als auch Datensicherheit zuzuordnen, da es das Selbsthosten und somit das Vermeiden der (Meta-)Datenweitergabe an Dritte ermöglicht.

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?

Ein grober Überblick über das Push-System sieht wie folgt aus:



Die oberen beiden Komponenten (Pushclient und Pushserver) sind dabei im Rahmen dieses Projektes entwickelt wurden.

Es wurde eine Protokollspezifikation für die Pushserver/Pushclient Interaktion geschrieben und auf Server und Clientseite implementiert. Verschiedene Protokolle für die Pushverbindung wurden evaluiert und sich für die erste Implementierung für Server-Sent-Events (SSE) entschieden. SSE's sind ein sehr einfaches Protokoll, und haben sich in ersten Tests als ausreichend für diese Anwendung erwiesen. Gleichzeitig ist das Protokoll des Pushkanals bewusst unabhängig von der Architektur des Gesamtsystems gehalten. Dies lässt Potential für spätere Optimierungen, ohne, dass das Auswirkungen auf Benutzer des OpenPush Projektes hat. Außerdem bietet es die Möglichkeit Pushbenachrichtigungen in Zukunft über einen schon bestehenden Kanal, wie z.B. IMAP Idle oder eine XMPP Verbindung zu senden. Dafür muss dann serverseitig (im IMAP Server oder XMPP Server) die OpenPush /message API implementiert werden.

Als initiale Herangehensweise an das Projekt war die Anlehnung der API und Architektur an Googles FCM geplant. Im Laufe des Projektes stellte sich heraus, dass ein kompletter Nachbau der API nicht sinnvoll ist. Somit war es Möglich ein teilweise einfacheres Design zu wählen und manche Self-Hosting betreffende Use-cases besser zu unterstützen (siehe Zielgruppe/Nutzen Abschnitt).

Die Protokollspezifikation erfolgte als OpenAPI (früher Swagger) Spezifikation. Diese lässt sich als Maschinenlesbare und ausführbare Dokumentation auffassen.

Die Implementierung der Pushserver Komponente ist im Python Flask Webframework erfolgt. Dabei wird mittels des Connexion (<https://github.com/zalando/connexion>) Frameworks die OpenAPI Spezifikation direkt geladen und eingehende API-Anfragen anhand dieser verifiziert.

Die Client-Implementierung der API wird mittels des openapi-generator (<https://github.com/OpenAPITools/openapi-generator>) Projektes aus derselben Spezifikation als Java Code generiert.

Dieser Java API-Client Code wird in das Pushclient Android Projekt eingebunden.

Dies ermöglicht das Pflegen der API Dokumentation an einer einzigen Stelle. Auf Server-Seite werden Änderungen durch Connexion sehr leicht umsetzbar, auf Client Seite können neue oder geänderte API Funktionen sofort benutzt werden.

Der Client-App Meilenstein wurde nicht komplett erreicht. Fertiggestellt wurde eine Client-App für das Android Gerät, welche sich bei einem vom User gewählten Pushserver registrieren kann. Apps, die den Pushservice benutzen wollen registrieren sich bei dieser App und erhalten dabei die Pushserver URL und einen App-Token. Um die App-Registrierung zu erleichtern sollte die Client-Library erstellt werden, korrespondierend zur FCM Client-Library. Die Registrierung muss im Moment noch manuell erfolgen, bis diese Library fertiggestellt ist. Das Zustellen von Nachrichten von der Pushclient-App and die registrierte User-App wurde ist noch nicht implementiert.

Die optional definierten Meilensteine, die die Integration der OpenPush Lösung in bestehende System vorsah wurden nicht erreicht.

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?

Ein Nutzen für oben genannten Softwareprojekte ist, dass es eine Lösung für eine vollständig freie Android Version ihrer Client-Apps mit Push-Benachrichtigungen geben kann. Somit können diese Versionen auch über den F-Droid App-Store verbreitet werden, welcher nur vollständig freie Software akzeptiert.

Ein weiterer großer Nutzen für Softwareprojekte, die Selfhosting als erklärtes Ziel haben, ist es keine Abhängigkeit auf externe Webservices zu haben. Im Google/Firebase Ökosystem gibt es üblicherweise eine gestaffelte Abhängigkeit zum einen zu Googles Firebase Cloud Messaging Servern, zum anderen aber auch zu einem Push-Proxy oder Gateway Server (z.b. <https://github.com/matrix-org/signal>). Dieser Gateway Server muss vom Distributor der bei Google Play verbreiteten App gehostet werden, da nur der Ersteller der App den Token hat um der App Push-Nachrichten zu senden. Dies macht z.B. die Entwickler der Riot-Android (matrix client) App verantwortlich für die Zustellung sämtlicher Mobilen Pushbenachrichtigungen, auch wenn der Nutzer seine eigene Serverinfrastruktur hostet. Da dieser „Entwickler-Token“ keine Sicherheitsrelevante Rolle spielt, sondern auf Googles Seite nur Accounting von Push-Anfragen macht, ist im OpenPush System kein so ein Token notwendig und das Hosten von Push-Proxies oder Neukompilieren von eignen App-versionen für jeden undabhängigen Hoster entfällt.

Der Plan für das weitere Vorgehen ist vorerst ist die Fertigstellung der clientseitigen Implementierung um das Projekt in der Praxis demonstrieren zu können. Als nächster Schritt steht die Integration in bestehende Projekte an, hierbei hoffe ich auf finanzielle und technische Unterstützung von den

Entwicklern der jeweiligen Plattformen zu bekommen, da eine offene, selbsthostbare Pushlösung viele Vorteile für diese Plattformen mit sich bringt (s.O.).

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

Das Förderprojekt gab den Anlass als selbstständiger Softwareentwickler zu Arbeiten, was ich nach dem Projektende weiter fortsetzen werde.

Ich habe viel dadurch gelernt, eine eigene API Schnittstelle zu designen, spezifizieren und zu implementieren. Dabei war es sehr hilfreich die 6 Monate nutzen zu können, um die Spezifikation der Schnittstelle zu verbessern, ohne dass es externe Konsumenten von dieser gab. So konnten initiale Denkfehler effizient vor einer ersten Veröffentlichung behoben werden. Die Fokussierung auf eine OpenAPI Spezifikation als Input für Server und Clientimplementierung hat initial etwas mehr Zeit gekostet, hat sich aber sehr bewährt um schnelle Iterationen der API vornehmen zu können, ohne den gleichen Boilerplate Code ständig anpassen zu müssen.

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 wurde versucht auf die bestehenden freien Push-Notification Lösung „Gotify“ aufzubauen. Dieser Ansatz hat sich aber als nicht zielführend erwiesen. Gotify adressiert Benachrichtigungen an User, wohingegen Push-Benachrichtigungen (im Sinne von FCM/APN) an einzelne Apps und per Gerät adressiert sind. Ohne komplette Änderungen der bestehenden Gotify Architektur lässt sich das nach Angaben der Autoren auch nicht ändern.

Als Alternative wurde ein eigener Pushserver in Python und ein Android Pushclient entwickelt.

Eine Integration vom Gotify-Pushservice in die Nextcloud Server-Software wurde als funktionstüchtiger Prototyp entwickelt. Diese Arbeit wurde auch nicht weiterverfolgt, die daraus gewonnenen Erkenntnisse können aber für die folgende Integration des eigenen OpenPush-Services wiederverwendet werden.

Ein Einbau der Push-Client Funktionalität in microg wurde begonnen, aber aufgrund von Zeitmangel vorerst verschoben. Es war schneller und einfacher eine alleinstehende Pushclient App zu entwickeln. Eine Integration in microg ist nach wie vor zu einem späteren Zeitpunkt geplant.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Projektwebseite: <https://bubu1.eu/openpush>

Pushserver-API: <https://gitlab.com/Bubu/pushserver/blob/master/openapi.yml>

Serverimplementierung: <https://gitlab.com/Bubu/pushserver>

Clientimplementierung: <https://gitlab.com/Bubu/pushclient>

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?

Hauptsächlich hat die Implementierung der Android Clientseite deutlich mehr Zeit gekostet als geplant. Dies lag zum einen an der zuerst versuchten direkten Integration in das microg Projekt, was sich als zu inflexibel für die am Anfang benötigten schnelleren Iterationszyklen herausstellte. Zum anderen wurde die Komplexität der Android Entwicklung allgemein unterschätzt was zu dem nicht vollständig erreichten Android Meilenstein geführt hat.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

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

Soweit ich das mitbekommen habe gab es während des Förderzeitraums keine Weiterentwicklung oder Veröffentlichung einer vergleichbaren oder ähnlichen Lösung.

Richtlinie zum „Software-Sprint“

NoIze – Selektiver Geräuschfilter

Schlussbericht

Zuwendungsempfänger: Peggy Sylopp und Aislyn Rose GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S49 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Wir wollten den Prototypen eines selektiven Noisefilter erstellen, der von dem/der Benutzer*in gewählte Störgeräusche im Alltag unterdrücken kann.

Motivation für die Initialisierung des \\NoIze//-Projekts war die moderate Höreinschränkung einer der GbR Gesellschafterin, Peggy Sylopp. Mit ihrer Höreinschränkung verbunden ist die Erfahrung von Beeinträchtigung der Sprachverständlichkeit in einer geräuschhaften Umgebung.

Durch ihrer Forschungsarbeit in der Arbeitsgruppe „Personalisierte Hörsysteme“ am Fraunhofer IDMT realisierte sie, dass Störgeräusche für die meisten Menschen, auch wenn sie über sehr gutes Hörvermögen verfügen, in bestimmten Situationen ein Problem darstellen oder zumindest als störend empfunden werden.

So beeinflussen Störgeräusche generell maßgeblich die Verstehbarkeit von Sprache. Für Menschen mit guter Hörfähigkeit bedeuten das Gespräche in Geräuschkulisse eine erhöhte Höranstrengung. Menschen können schon bei geringen Hörverlust Gesprächspartner in Geräuschkulissen nicht mehr verstehen.

Störgeräusche senken auch die Aufmerksamkeitsschwelle auf gelesenen Text und können selbst bei geringer Lautstärke Stress verursachen. Starke Störgeräusche können Hörschäden zur Folge haben.

Mit der Entwicklung von \\NoIze// wollten wir die Grundlagen für eine App schaffen, die negative Auswirkungen von Störgeräuschen mildert.

Folgende Meilensteine waren geplant:

1. Meilenstein Ende März 2019:

Trainingsdaten bestimmen

Zum Trainieren unseres Algorithmus greifen wir auf bestehende Sound-Datenbanken zurück, wie freesound.org, Urban Sound Classification, Soundarchiv Deutschlandfunk.

2. Meilenstein Ende April 2019:

Festlegung Lernalgorithmus und Variablen, Programmiersprachen, Bibliotheken

Wir parametrisieren die Trainingsdatensätze, z.B. bzgl Frequenz, Samplelängen bzw. Zeit-Intervalle und Relationen

3. Meilenstein Ende Mai 2019: Implementierung des Prototypen auf Linux System

Programmierung des Lern-Algorithmus

Recherche: Wir entscheiden, welche Technologien wir benutzen, und ob wir eine Topologie (z.B. RNNoise) deployen

4. Meilenstein Ende Juni 2019:

mehrere Testläufe mit diversen Datensätzen durchgeführt

Training: Die ML Topologie wird auf einer lokalen Testmaschine (mit entsprechender GPU und Arbeitsspeicher) trainiert

5. Meilenstein Ende Juli 2019:

Iteration, Anpassungen, selektiver Noisefilter getestet, Dokumentation aufgesetzt, Test außerhalb der Maschine in definierter Testsituation

1-3 wird iteriert. Wir testen die Topologie mit den im Selbstversuch aufgenommen Geräuschdatensätzen und passen die Topologie ggf. entsprechend an.

6. Meilenstein Ende August 2019:

Stresstest, Finetuning, Nolze-Filter für ein autonomes System, Prototyp für personalisierten Noisefilter evaluiert, Dokumentation fertig

Wir testen und evaluieren die Noisefilter im realen akustischen Raum

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

Wie zu Anfang des Dokuments in der Motivation beschrieben betreffen die Anwendungsszenarien viele Menschen. Wir leben in einer lärmenden Umwelt, wobei selbst leise Umgebungsgeräusche Stress bewirken können. Viele Menschen kennen das Problem mit dem "Cocktail-Party-Effekt", sie verstehen ihr Gegenüber bei Gesprächen mit Nebengeräuschen schlecht bis gar nicht.

Mit \\Nolze// wollten wir die Grundlagen für einen selektiven Open Source Geräuschfilter schaffen, der in verschiedenen Hardware- und Softwareumgebungen implementiert werden kann. Ziel ist, dass \\Nolze// in mobilen Lösungen wie Hearables oder Smartphones das selektiven Filtern von user-bestimmten Alltags-Geräuschumgebungen ermöglicht.

Die Entwicklung von \\Nolze// sollte allen Menschen zugänglich, auch für darauf basierende Folgeprojekte, nachhaltig gestaltet sein. Daher war es uns wichtig, unter der GPL3.0 zu veröffentlichen.

Mit dem Einsatz von maschinellen Lernen sollte die Architektur für einen dynamischen Geräuschfilter geschaffen werden, der über die Lernfähigkeit sich immer weiter verbessert.

Durch gezieltes Herausfiltern von belastenden und die Verstehbarkeit einschränkenden Nebengeräuschen kann \\Nolze// einen Beitrag zur für Gesundheit, Prävention von Hörschäden und besseres Hören leisten.

Ausführliche Darstellung der Ergebnisse

Wir konnten alle Meilensteine erreichen und darüber hinaus noch weitere Entwicklungen veröffentlichen.

Unsere Ergebnisse anhand der Meilensteine:

1. Meilenstein: Wir definierten unsere Ziele und organisierten den Arbeitsablauf. Die Open Source Sound Datenbank freesound.org hat uns die Daten geliefert, um den Algorithmus entwickeln zu können. Wir recherchierten zu Ansätzen zur Echtzeit-Filterung im Zusammenspiel mit Deep Learning und geringer Rechenleistung. Wir experimentierten mit den ersten Bausteinen wie der Voice Activity Detection mit Hilfe von Convolutional Neural Networks (CNN)

2. Meilenstein: Überarbeitung des CNN zur Klassifizierung der Geräusche mit Data Science Techniken. Erste Überlegungen zum Wiener Filter wurden angestrengt.

3. Meilenstein: a) Die Recherche des Lern-Algorithmus¹ lieferte uns wissenschaftliche Publikationen¹², die uns als Orientierung für die Implementierung dienten.

b) Die Programmierung der ersten Implementierung des Algorithmus¹ fokusierte zunächst dessen Funktionalität, das Zusammenspiel von Geräusch-Klassifizierung und Wiener Filter wurde mit ersten praktischen Ergebnissen realisiert. Ein Spectral Subtraction Filter erweist sich als Sackgasse, da zu viele Bugs entstehen.

4. Meilenstein: Trainingsdaten wie auch die Implementierung wurden weiter verfeinert, bis zufrieden stellende Ergebnisse erzielt werden konnten. Ein Test wurde entwickelt (PyTest) und die erste Dokumentation aufgesetzt.

5. Meilenstein: Der Code wurde reorganisiert, so dass er als Python Tool einsetzbar wurde. Wir testeten den Algorithmus in verschiedenen Szenarios, u.a. mit von uns erstellten alltagsakustischen Aufnahmen.

6. Meilenstein: Wir setzen die Seite für die Projektdokumentation und das finale Repository auf. Der Code wurde final reorganisiert. Tabellen und Graphen zur Dokumentation erstellt. Über den vorher definierten Meilenstein hinaus veröffentlichen wir den Code auf Jupyter Notebook, so dass selbst für Laien die Funktionsweise des Filters online nachvollziehbar wird. Die Software Demo mit guter Dokumentation erlaubt einen Einstieg in die Programmierung von Soundklassifizierung mit Python.

Die Ausschreibung und das offene Konzept der Open Knowledge Foundation hat uns ermutigt, ein Projekt zu starten, an dass wir uns unter anderen Umständen nicht heran gewagt hätten. Insbesondere als Mütter mit sehr beschränkten Zeitrahmen für ehrenamtliche Tätigkeiten haben wir erst durch die finanzielle Unterstützung des Prototypefund die Möglichkeit erhalten, uns intensiv dem Open Source Projekt zu widmen.

Das Coaching machte uns klar, wie wir die Idee hinter unserem noch nicht alltagstauglichen Prototypen auf akademischer Basis auch für Laien nachvollziehbar machen können. So wurden wir dazu angeregt, ein GUI für eine App zu skizzieren, die die Steuerung von Nolze möglich machen könnte. Das hat das Potential für Vermittlung und Weiterentwicklung ermöglicht.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Jede Person, die sich an bestimmten Geräuschen stört, ist potentiell in der Zielgruppe von \\Nolze// enthalten.

Die Implementierung, Anpassung und Wartung der \\Nolze// Anwendung in Hearables wie Headsets, Earbuds oder Computer und Smartphones erlaubt eine kommerzielle Verwertbarkeit.

Der Ergebnisse des Nolze Prototypen wollen wir kontinuierlich erweitern in Bezug auf Zuverlässigkeit der Algorithmen und Anwendung in alltagsakustischen Situationen sowie Sicherheit im Umgang mit den Daten.

Durch die intensive Auseinandersetzung mit wissenschaftlichen Publikationen und deren Implementierung haben wir uns fachlich sehr viel weiter gebildet.

Der Schritt, die Verantwortung für eine solche Herausforderung zu übernehmen, hat uns persönlich gestärkt. Der lange Weg von der Idee zur Implementierung, von den anfänglich frustrierenden Erfahrungen bis hin zur Machbarkeit hat unser Selbstvertrauen gestärkt.

1 A Real-Time Personalized Noise Reduction Smartphone App for Hearing Enhancement, Nasim Alamdari, Shashank Yaraganalu, Nasser Kehtarnava, Department of Electrical & Computer Engineering, University of Texas at Dallas, Richardson, TX, USA, 2018

2 Efficient Musical Noise Suppression for Speech Enhancement Systems, Thomas Esch and Peter Vary, Insitute of Communication Systems and Data Processing, RWTH Aachen University, Germany, 2009

Eine mögliche Weiterentwicklung wäre ein Pilot für eine GUI für eine bestimmte Alltagssituation mit Störgeräusch, beispielsweise dem Arbeiten mit Laptop in einem Büro mit akustisch störender Klimaanlage. So könnte ein praktisch erfahrbarer Erfolg gezeigt werden, der die Bahn für weitere Anwendungen frei macht.

Weiterhin könnte die Optimierung der Signalverarbeitung im Sinne von noch geringerer Prozessorbelastung die Implementierung auf einem Smartphone oder Raspberry Pi ermöglichen. Dies könnte weitere unabhängige Weiterentwicklungen in vielfältigen Situationen außerhalb von geschlossenen Räumen nach sich ziehen, wie z.B. im öffentlichen Verkehr.

Die Entwicklung einer App zum Upload und Austausch von Open Source Audiodaten würde zur Generierung von Audio-Datenbanken führen. Diese werden benötigt, um den bestehenden Geräuschfilter trainieren und optimieren zu können. Bestehende freie Datenbanken verfügen nicht über ausreichende Alltagsgeräusche.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die im Antrag formulierten Ziele haben wir soweit erreicht. Darüber hinaus haben wir im Vorfeld und während der Projektzeit noch folgende Ideen diskutiert:

Wir hatten überlegt, den Algorithmus soweit zu optimieren, dass er Sound-Datenströme, (also live-Umgebungs-Akustik) ohne Delay verarbeiten kann, etwa wie ein Noisecancelling Kopfhörer oder ein Hörgerät. Dies stellte sich aber als eine Herausforderung heraus, die wir mit unseren zeitlichen und personellen Ressourcen nicht realisieren konnten.

Weiterhin hatten wir darüber nach gedacht, die Implementierung auf einen Raspberry Pi (siehe unter Meilenstein 6: Test auf autonomen System) zu portieren. Auch dieses war mit unseren Ressourcen nicht machbar.

Wir entwarfen im Laufe des Projekts die Idee, die Implementierung als Basis für Workshops zu nutzen. Aus dieser Idee heraus entwickelten wir in mehreren ein Demo und ein Tutorial auf der online Jupyter-Notebook Plattform, das es auch Newbies möglich macht, die Arbeitsweise unserer Implementierung nach zu vollziehen. Es ist uns aber zum jetzigen Zeitpunkt des Projektabschlusses noch nicht klar, ob wir diesen Ansatz wirklich verwirklichen wollen, da die Vermittlung der Implementierung sehr komplex ist.

Hier können sich Interessenten detailliert über unsere Projektergebnisse informieren:

Webseite: <https://pexlab.space/index.php/de/50-selektiver-geraeuschfilter-auf-basis-von-ai>

Github: <https://github.com/pgys/Nolze>

Tutorial: <https://notebooks.ai/a-n-rose/>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die anfängliche Recherche mit dem Durcharbeiten verschiedener Publikationen nahm sehr viel Zeit als geplant in Anspruch. Die Implementierung der Klassifizierung der Klänge stellte sich als kleineres Problem heraus. Wesentlich mehr Entwicklungszeit nahm die Implementierung des Wiener Filters in Anspruch, da wir keine Implementierung im Netz finden konnten, die wir modifizieren und weiter entwickeln konnten und auch nicht über keine Expertise in der Soundverarbeitung verfügen. Die Überarbeitung und Reorganisierung des Codes, so dass er auch für Außenstehende nachvollziehbar wird, kostete auch viel Zeit, ebenso wie die Erarbeitung eines Demos mit Tutorials. Da wir mit einem akademischen Ansatz arbeiteten, kostete es auch einiges an Überlegung, ein Konzept für die Übermittlung der Idee zu entwickeln.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Peggy Sylopp ist für das Citizen Science Projekt „Hear How You Like To Hear“ am Fraunhofer IDMT an der Anpassung des Open Source Hörgerätealgorithmus OpenMHA und der Entwicklung dessen Steuerung mit einer App beteiligt. Der Einblick in die Funktionsweise von Hörgerätealgorithmen und der Herausforderung deren Implementation hat sie vielfältig inspiriert. So gewann sie den Eindruck, dass das selektive Herausfiltern von Geräuschfiltern vielen Menschen zu Gute kommen könnte. Sie erfuhr, dass die Entwicklung eines selektiven Geräuschfilters mittels maschinellen Lernens noch eine Herausforderung in der Wissenschaft darstellte.

Die Mit-GbR-Inhaberin Aislyn Rose konnte auf ihre Erfahrung in der Sprach-Klassifizierung mit Python zurück greifen. Diese war Basis für die Implementation des Geräusch-Klassifizierers.

Folgende wissenschaftliche Publikationen haben uns inspiriert:

„A Real-Time Personalized Noise Reduction Smartphone App for Hearing Enhancement“, Nasim Alamdari, Shashank Yaraganalu, Nasser Kehtarnava, Department of Electrical & Computer Engineering, University of Texas at Dallas, Richardson, TX, USA, 2018

Efficient Musical Noise Suppression for Speech Enhancement Systems, Thomas Esch and Peter Vary, Institute of Communication Systems and Data Processing, RWTH Aachen University, Germany, 2009

Speech Enhancement, Theory and Practice, Second Edition, Philipos C. Loizou, 2013

Berlin, 19.09.2019,

Peggy Sylopp, contact@peggy-sylopp.net

Aislyn Rose, arose510@gmail.com

microG Project

Schlussbericht

Zuwendungsempfänger:

Marvin Wißfeld

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S50 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Jederzeit mit einem Smartphone ausgestattet zu sein ist in unserer Gesellschaft inzwischen der Standard. Wer nicht die gängigen Apps für jeden Zweck nutzt oder nutzen kann wird aus Teilen der Gesellschaft ausgeschlossen oder hat weniger Komfort. Diese Apps erfordern jedoch häufig die Nutzung eines der Betriebssysteme von Apple oder Google. Und obwohl Teile von Google's Android open-source sind, sind viele Apps damit nicht oder nicht vollständig kompatibel. Dabei ist gerade Google immer wieder in der Kritik, da sie die Privatsphäre ihrer Nutzer nicht achten.

microG möchte die Kompatibilität von allen Apps mit freier Software ermöglichen und dabei so datenschutz-freundlich wie möglich sein. Somit können auch Nutzer die auf Privatsphäre und ein freies Betriebssystem Wert legen, wieder alle Apps und eventuelle Zusatzgeräte (Android Wear, Chromecast) nutzen. Dabei kann der Nutzer genau auswählen, welche Dienste von Google er weiterhin nutzen möchte, private Daten werden dabei in jedem Fall soweit möglich verschleiert.

Im Rahmen dieses Projekts soll microG:

1. weiterhin gepflegt und auf dem aktuellen Stand gehalten werden.
2. eine neue Funktion zu Darstellung von Karten (dort wo Apps sonst Google Maps einbetten) entwickelt werden (die aktuelle ist fehlerhaft und muss komplett neu-entwickelt werden).
3. Unterstützung von Smart-Watches mit "Android Wear"-Betriebssystem hinzugefügt werden.

Darüber hinaus soll in Zusammenarbeit mit dem Projekt "Implementierung eines Open-Source Push services für Android Apps" microG Bibliotheken und Tools angepasst werden, um die Nutzung eines alternativen Push Dienstes (statt nur den von Google wie aktuell) zu ermöglichen.

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

microG richtet sich an alle, die nicht bereit sind ihre privaten Daten mit Google oder Apple zu teilen, aber dennoch alle Apps benutzen wollen, die sie für ihre gesellschaftliche Teilhabe benötigen.

Da die Installation von microG auf vielen Smartphones nicht ohne weiteres möglich ist, ist darüber hinaus eine Person mit technischem Know-How erforderlich. Nach erfolgreicher Einrichtung und Einführung kann aber jeder microG benutzen und auf dem aktuellen Stand halten.

Sobald microG eine ausreichend hohe Qualität erreicht hat, kann es auch direkt von Herstellern vorinstalliert werden, sodass auch unerfahrene Nutzer die Funktionalität von microG nutzen können.

Ausführliche Darstellung der Ergebnisse

Während der Förderzeit konnten folgende Konkrete Ergebnisse erzielt werden:

1. Das Projekt wurde weiter gepflegt. Dazu wurden die Software für bessere Kompatibilität mit der neuesten Version von Android angepasst, insbesondere in Bezug auf Stromsparfunktionen und der Account-Verwaltung. In der Förderzeit sind zwei Updates für Endnutzer entstanden, die neben diesen Änderungen und den weiteren „großen“ Förderzielen auch weitere Fehlerkorrekturen bereitgestellt haben.
2. Die Funktion der Darstellung von Karten wurde von Grund auf neu-entwickelt. Das Kartenmaterial von OpenStreetMap wird jetzt durch die freie Software Mapbox verarbeitet und in die Anwendungen integriert. Die neue Implementierung wurde insbesondere auf Kompatibilität mit populären Apps wie DB Navigator und Uber getestet. Zum Ende der Förderzeit war diese Implementierung voll einsatzbereit.
3. Die Unterstützung für Android Wear Smart-Watches ist technisch fortgeschritten, wurde aber bisher ausschließlich bei Nutzung der Google Android Wear App in Zusammenspiel mit einem Android Wear Emulator erfolgreich getestet. Im aktuellen Stand können Apps über Gerätegrenzen hinweg miteinander kommunizieren, Benachrichtigungen vom Smartphone werden auf die Smartwatch übertragen und können dort bestätigt oder beantwortet werden. Zudem kann über die Google Android Wear App die Smartwatch konfiguriert werden.

Zur Fertigstellung der Android Wear Smartwatch Unterstützung ist weitere Arbeit notwendig, insbesondere die Kompatibilität mit über Bluetooth angeschlossener Geräte und Integrations-Tests mit verschiedenen Smartwatch-Typen, sowie Integration mit der Software GadgetBridge als Ersatz zur Google Android Wear App.

Außerdem haben Gespräche mit Marcus vom Projekt "Implementierung eines Open-Source Push services für Android Apps" zu einem langfristigen Plan zur Integration der dabei entwickelten, offenen Push-Technologie in microG geführt und auch sein Projekt konnte von den Erfahrungen mit microG's Implementierung der Google Push-Technologie.

Insgesamt hat der Software Sprint das Projekt microG große Schritte weitergebracht. Dazu hat auch die Betreuung durch die Open Knowledge Foundation, insbesondere das Coaching für Benutzerschnittstellen beigetragen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die aktuellen Nutzer von microG konnten durch die weitere Pflege und die neuen Funktionen von einer besseren Unterstützung verschiedener Apps, insbesondere populäre Apps aus dem Bereich der Mobilität, profitieren. Außerdem hat das hinzufügen der neuen Funktionen einigen Nutzern das erstmalige nutzen von microG und einem Android-System ohne Google-Apps ermöglicht, da diese auf Apps angewiesen waren, die mit microG vorher nicht kompatibel waren.

Die Änderungen und neuen Funktionen wurden in ihrem Fertigungsprozess regelmäßig über das Open-Source-Repository von microG auf GitHub mit anderen Entwicklern geteilt. Dadurch konnten sie diese frühzeitig testen, kommentieren oder Fehler berichten. Langfristig ermöglicht der offene Quellcode auch anderen Entwicklern, Dienste zu entwickeln die mit Google kompatibel sind.

Das Projekt microG wird in Zukunft ohne Förderung offen weiterentwickelt. Insbesondere soll die Android Wear Unterstützung wie oben beschrieben abgeschlossen und erweitert werden. Außerdem soll auch weiterhin die Kompatibilität mit neuen Android- und App-Versionen garantiert werden. Außerdem soll microG insbesondere auch professionell besser einsetzbar werden, dazu soll enger mit professionellen Nutzer wie Herstellern zusammengearbeitet werden.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wie oben beschrieben konnte die Android Wear Kompatibilität nicht abgeschlossen werden und ist deshalb in der Praxis für Nutzer noch nicht verwendbar (da ausschließlich der Android Wear Emulator und keine physischen Geräte genutzt werden können). Dies ist jedoch nicht auf grundsätzliche Fehler in der Arbeit zurückzuführen, sondern auf unerwartete Komplexität der genutzten Bluetooth-Schnittstelle.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Ergebnisse wurden vollständig in den Code von microG integriert und sind somit über das zugehörige GitHub Code-Repository unter <https://github.com/microg> erreichbar. Die im Laufe des Förderzeitraums erstellten Endnutzer-Versionen sind über den Download-Bereich unter <https://microg.org/> verfügbar.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Durch unvorhersehbare, zusätzliche, private Verpflichtungen konnte ich leider nicht die gewünschte Arbeitszeit und insbesondere keine zusätzliche (über die Förderung hinausgehende) Arbeitsstunden aufbringen, wodurch nicht alle Projekt-Ziele (wie oben beschrieben) vollständig erreicht werden können. Die fehlenden Arbeiten sollen aber im Anschluss nach zeitlicher Verfügbarkeit und ohne die Notwendigkeit weiterer Fördermaßnahmen abgeschlossen werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Förderzeit gab es, abgesehen von der Veröffentlichung von Vorab-Versionen von Android 10 durch Google, keine signifikanten Ergebnisse die microG beeinflusst hätten. Die finale Veröffentlichung von Android 10 erfolgte erst nach Abschluss der Förderphase am 3. September und erfordert keine signifikanten Änderungen an microG.

Richtlinie zum „Software-Sprint“

Open Green Web

Schlussbericht

Zuwendungsempfänger:

Christopher Adams

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S51 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Im Jahr 2019 läuft das Internet noch immer überwiegend mit fossilen Brennstoffen. Das heißt, wenn wir das Internet nutzen, verursachen wir vermeidbare Schäden, weil fossile Brennstoffe verbrannt werden.

Das Internet muss jedoch nicht mit fossilen Brennstoffen betrieben werden. Wir können zu einem Internet übergehen, das mit erneuerbaren Energien betrieben wird, aber dazu müssen mehr Menschen wissen, wie man das verlangt, und wir brauchen mehr Transparenz.

Mein Projekt, OpenGreenWeb, sollte dies tun, indem es Daten und Quellcode aus einem bestehenden Projekt erzeugt, öffnet und eine Community um die Idee eines grünen Internets herum aufbaut.

Beitrag des Projektes zu den Zielen der Förderinitiative

„Software-Sprint“

Meine Hauptzielgruppe für OpenGreenWeb sind IT-Profis, die digitale Produkte und Websites erstellen.

Dies ist eine ausgebildete Gruppe, und wenn sie die Auswirkungen des Aufbaus digitaler Dienste mit fossilen Brennstoffen verstehen, wollen sie das normalerweise verändern.

Dieses Projekt hilft ihnen, indem es die Werkzeuge und Daten zur Verfügung stellt, die sie benötigen, um Entscheidungen zu treffen zu können, um zu verändern, wie sie digitale Produkte entwickeln und umweltfreundlichere Dienste zu nutzen.

Die wichtigste Verbindung zu den Zielen der Sprints ist die Erstellung der Daten, die für jede Art von maschinellem Lernen in diesem Bereich benötigt werden.

Ausführliche Darstellung der Ergebnisse

Ich war in der Lage, einen offenen Datensatz zu erstellen, in dem die Art der Stromversorgung - klimaneutral oder nicht - der beliebtesten Websites der Welt abrufbar ist und diesen zu veröffentlichen. Er ist verfügbar unter:

<https://www.thegreenwebfoundation.org/green-web-datasets/>

Ich habe auch den Quellcode für die Plattform veröffentlicht, um es einfacher zu machen, diesen Daten zu vertrauen. Es ist auf Github verfügbar unter:

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

Diese veröffentlichten Daten werden nun in anderen Produkten und Dienstleistungen verwendet. Wir stellen den Datensatz über eine API zur Verfügung, unter:

<https://api.thegreenwebfoundation.org/>

Seitdem ich angefangen habe, an dem Projekt zu arbeiten und den Code und die Daten zu veröffentlichen, hat sich die Nutzung dieser API um das 30-fache erhöht.

Ich weiß, dass die deutschen Grünen damit alle ihre eigenen Seiten überprüfen, da ich ihnen geholfen habe, ihre Tools zu integrieren, um sie nutzen zu können.

Ich habe auch ähnliche Projekte und Gruppen aus der Zusammenarbeit mit der Open Knowledge Foundation gefunden.

So arbeite ich beispielsweise jetzt mit der EU-Kommission zusammen und trage zu neuen Leitlinien für eine umweltfreundliche, öffentliche Auftragsvergabe in der EU bei, um Dienstleistungen zu fördern, die keine fossilen Brennstoffe verwenden.

Ich habe fast alle Meilensteine erreicht. Ein Meilenstein, den ich nicht erreichen konnte, war die Erstellung einer wissenschaftlichen Arbeit zu dem von uns erstellten Datensatz. Hierfür wurde noch kein Papier veröffentlicht, das dies tut.

Ein weiterer Meilenstein vor Beginn war, dass die Mitarbeiter den Datensatz über eine API aktualisieren können. Nachdem ich mit den Benutzern gesprochen hatte, stellte ich fest, dass sie die Daten nicht ausreichend geändert haben. Ich habe meine Bemühungen stattdessen auf andere Orte konzentriert.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Es gibt neue, externe Mitwirkende an dem von mir veröffentlichten Code. Es wurden neue Tools entwickelt, die die API und Daten nutzen - mittlerweile gibt es rund 30 Projekte auf der Github-Website der Green Web Foundation. Von diesen werden rund 5 aktiv bearbeitet.

Ein Beispiel ist Green Cost Explorer - ich habe mit Mitarbeitern eines Online-Kartierungsunternehmens zusammengearbeitet, um ein Tool zu entwickeln, das es Menschen ermöglicht, ihre eigenen Ausgaben für Cloud Computing zu analysieren und zu sehen, wie viel sie für Infrastrukturen mit fossilen Brennstoffen ausgeben. Es ist verfügbar unter:

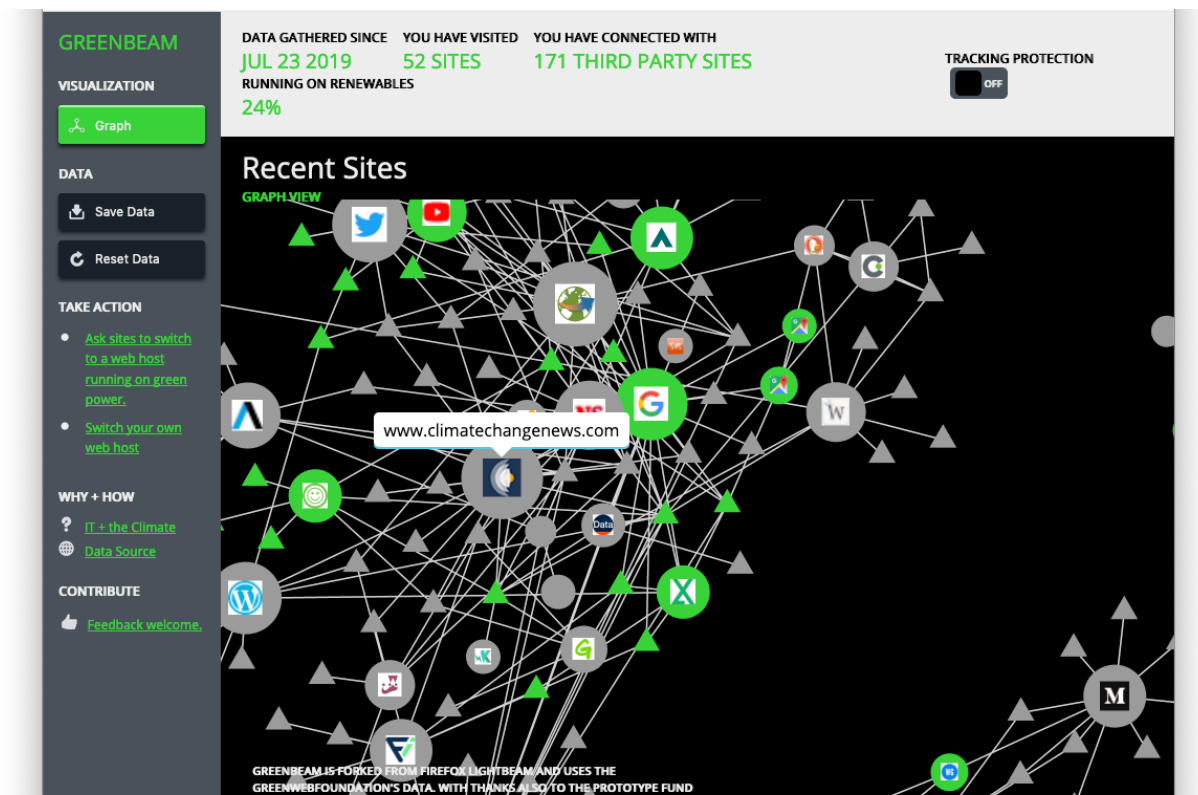
<https://github.com/thegreenwebfoundation/green-cost-explorer>

service	Green Cost by service	Grey Cost by service
Amazon Simple Storage Servi...	99.6% (\$120.09)	0.4% (\$0.51)
Amazon Lightsail	82.4% (\$18.49)	17.6% (\$3.94)
Amazon CloudFront	65.0% (\$8.07)	35.0% (\$4.34)
Amazon Elastic Compute Clou...	100.0% (\$0.01)	0.0% (\$0.00)
AWS Lambda	100.0% (\$0.00)	0.0% (\$0.00)
EC2 - Other	100.0% (\$0.00)	0.0% (\$0.00)
Tax	0.0% (\$0.00)	100.0% (\$48.16)
Amazon Route 53	0.0% (\$0.00)	100.0% (\$21.71)
Amazon Registrar	0.0% (\$0.00)	100.0% (\$24.00)
Amazon EC2 Container Service	0.0% (\$0.00)	100.0% (\$49.82)

Green Cost Explorer

Ein weiteres Projekt war Greenbeam, basierend auf Lightbeam, einem früheren Open-Source-Projekt aus einer früheren Prototyp-Runde. Ich habe mit einem Künstler zusammengearbeitet, um Lightbeam zu erweitern, um den Nutzern zu zeigen, wie viel von dem Web, das sie nutzen, auf fossilen Brennstoffen läuft. Der Code ist hier verfügbar und steht als Add-on für den Firefox-Browser zur Verfügung.

<https://github.com/eveahe/greenbeam/>



Greenbeam

Ein weiteres Projekt, an dem ich gearbeitet habe, heißt Greenhouse. Es verwendet die von mir veröffentlichten Daten, um Designern und Entwicklern zu helfen, zu sehen, wie viele Dienste von Drittanbietern, die sie in ihren Anwendungen verwenden, auf Strom aus fossilen Brennstoffen basieren.

Dies ermöglicht es ihnen, Änderungen vorzunehmen, bevor sie neuen Code für die Welt freigeben. Es ist online unter:

<https://github.com/thegreenwebfoundation/lighthouse-plugin-greenhouse>



Sustainable Web

These checks show changes to make to reduce the carbon emissions from what you build. Climate crisis, remember?

Page summary

- Page is built using resources from servers running on fossil fuels ^

Burning fossil fuels to power servers is avoidable, and contributes to climate breakdown. See the [W3C Ethical Web Principles](#), on Sustainable Web, and learn more at [The Green Web Foundation](#)

GreenHouse

Während meiner Arbeit am Open Green Web habe ich auch zu den Ethical Web Principles beigetragen, die von der Technical Architecture Group des W3C veröffentlicht wurden, um die CO2-Emissionen in ihrer eigenen Anleitung zu nennen. Sie können die Pull-Anfrage unten sehen:

<https://github.com/w3ctag/ethical-web-principles/issues/11>

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Als ich anfang, dachte ich, dass die API sehr wichtig sein würde. Es stellte sich heraus, dass sich die Infrastruktur bei vielen Anbietern nicht schnell genug ändert, um dies zu erreichen.

Ich dachte auch, dass die Datensätze von Wissenschaftlern oder Datenjournalisten verwendet werden könnten. Während ich am Ende mit Journalisten und Wissenschaftlern sprach, waren sie nicht so sehr an den Datensätzen interessiert. Allerdings gefiel ihnen, dass die Plattform Open Source war, so dass sie sehen konnten, wie die Daten generiert

wurden.

Als Antwort darauf konzentrierte ich mich auf die Unterstützung neuer Entwickler bei der Verwendung der API und der von mir veröffentlichten Daten sowie auf das Schreiben von Leitfäden, die es Politikern erleichtern würden, sich für eine Abkehr vom Ökostrom in der IT einzusetzen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Die Hauptseite ist auf:

<https://www.thegreenwebfoundation.org/>

Die Datensätze werden veröffentlicht unter:

<https://www.thegreenwebfoundation.org/green-web-datasets/>

Die Open-Source-Projekte sind alle auf Github aufgelistet:

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

Bei der Green Web Foundation bieten wir jetzt kommerzielle Beratung und Training für größere Unternehmen in Deutschland und Großbritannien an, die die CO2-Emissionen ihrer digitalen Produkte reduzieren wollen. Was wir jetzt anbieten, sehen Sie unten:

<https://www.thegreenwebfoundation.org/services/>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ich wollte mehr Zeit damit verbringen, eine API zu erstellen, mit der Benutzer die von uns veröffentlichten Datensätze aktualisieren können. Anstatt Zeit damit zu verbringen, nutze ich die Zeit, um anderen Entwicklern zu helfen, Open-Source-Tools zu entwickeln, die unsere API nutzen.

Das bedeutet, dass sich die Gesamtzeit nicht verlängert hat, aber ich habe neben dem ursprünglichen Plattform-Code auch zu Greenhouse, Greenbeam und Green Cost Explorer beigetragen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Bei der Arbeit am Open Green Web veröffentlichte Google eine neue Version seines Web Page Auditing Tools, Lighthouse, mit einer neuen Plugin-Architektur.

Da ich mich dafür entschieden hatte, weniger Zeit mit der API (einem der früheren Meilensteine) zu verbringen, konnte ich Zeit damit verbringen, Greenhouse zu bauen, ein Projekt, das die von mir veröffentlichten Daten verwendete. Dies erregte mehr Aufmerksamkeit und Beiträge anderer Entwickler.

Außerdem hatte ich ein Mentorenprogramm mit einer Online-Community, ClimateAction.tech, durchgeführt. Ich betreute eine Person aus diesem Mentorenprogramm, und am Ende bauten wir gemeinsam eine erste Version von Greenbeam, die auch die Green Web-Daten verwendete.

Richtlinie zum „Software-Sprint“

Photownica

Private Fotogalerien sicher verschlagworten

Schlussbericht

Zuwendungsempfänger:

Robert Buchholz

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S52 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Photownica ist eine Bildersuche ohne Cloud-Dienste. Private Fotos werden mittels Verfahren der Künstlichen Intelligenz verschlagwortet, ohne das Endgerät der Nutzer verlassen zu müssen. Das ermöglicht die Navigation der eigenen Fotosammlung anhand von Bildinhalten.

Nutzerinnen haben bisher die Wahl zwischen Preisgabe privater Daten oder Verzicht auf Funktionalität. Vergleichbare Funktionen (Foto-Browsen nach Person, Ort oder Bildinhalt) werden ermöglicht, wenn eine Nutzerin ihre Fotos z.B. zu Facebook oder Google hochlädt. Apple zwingt ihre Nutzerinnen hierfür nicht zum Upload, jedoch in ihr proprietäres Ökosystem. Nutzerinnen von Open-Source Betriebssystemen (z.B. LineageOS, Linux) und Datenschutz-Affine sind von dieser Entwicklung ausgeschlossen.

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

Das Projekt richtete sich an Datenschutz-affine Hobby-Fotografinnen, welche Fotos über nur über sichere und private Kanäle teilen möchten. Diesen bleibt bisher verwehrt, eine schlagwort-basierte Inhaltssuche zu nutzen, um zielgerichtet Fotos finden zu können.

Ausführliche Darstellung der Ergebnisse

Die in der Planung vorgesehen Meilensteine teilten sich in ein Applikation „A“ zur Verschlagwortung und eine (möglicherweise separate) Applikation „B“ zur Anzeige der Bilder nach Schlagworten mit Suche. Diese Meilensteine unterteilten sich funktional aufeinander aufbauend: A1. Objekterkennung,

A2. Speicherformat XMP (Exif) und Datei-Speicherung, A3. Personen-Erkennung, A4. Android Hintergrund-Dienst sowie B1. Schlagwort-Anzeige.

Leider konnte im Förderzeitraum keiner der Meilensteine abgeschlossen werden, sondern lediglich Ergebnisse der Vorbereitungs- und Designphase erarbeitet werden. Diese teilen sich in inhaltliche und technische Ergebnisse. Inhaltlich wurde ein UX-Konzept zur Anzeige und Navigation der Schlagworte (App „B“) entwickelt, welches an die Applikation Apple Fotos auf iOS angelehnt ist. Nach dem Studium des Quellcodes der Android (AOSP) Gallery wurde klar, dass die Entwicklung einer neuen Applikation zur Schlagwort-Navigation mit rudimentärer Foto-Anzeige weniger aufwändig wäre als die Ergänzung der Navigation in die Gallery. Entsprechend erschien es sinnvoll, die Funktionen A und B in der gleichen Applikation abzubilden.

Die technische Evaluation sollte vor allem die (frühen) Fragen der Integration/Frameworks klären: Nutzung von TensorFlow oder PyTorch zur Entwicklung und Anbindung des neuronalen Netzes? Für das User-Interface standen als aktuelle Alternativen React Native, Flutter und Native Android-Entwicklung zur Verfügung. Eine kurze Evaluation der Adapter von TensorFlow und PyTorch machte deutlich, dass der TensorFlow-Adapter stabiler und für Produktiv-Entwicklung geeignet erschien. Um die UI bewusst einfach zu halten, habe ich mich für native Entwicklung entschieden.

Leider war es mir nicht möglich, nach Abschluss der Evaluation die Entwicklung der Applikation weiterzuführen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Leider gibt es keinen praktischen Nutzen durch Fehlen einer Anwendung, jedoch habe ich im Rahmen der Veranstaltungen des Prototype Fund für die Idee Öffentlichkeitsarbeit durchgeführt. Ich bin zuversichtlich, dass dies zu einer erhöhten Aufmerksamkeit für die Problematik führt und sehe die grundlegenden technischen Ansätze (offline verfügbare KI-Anwendungen) vermehrt in anderen Produkten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Den Ansatz der Zweiteilung der Applikation in Indizierung und Suche habe ich verworfen und stattdessen eine integrierte Anwendung konzipiert.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Ergebnisse und Roadmap finden sich im GitHub Repository: <https://github.com/rbu/photownica/>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Leider konnte ich nur einen kleinen Teil der für das Projekt vorgesehenen Zeit abrufen, da ich für ein anderes Projekt entgegen meiner ursprünglichen Planung benötigt wurde. Dadurch war es mir schon nach wenigen Tagen nicht mehr möglich, Zeit zur Durchführung einzubringen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Im Rahmen des Nextcloud-Projektes gibt es einige Entwicklungen, welche serverseitig KI einsetzen, um Fotos zu verschlagworten. Das aktivste Projekt in diesem Bereich ist „Face Recognition“ (<https://github.com/matiasdelellis/facerecognition>). Dieses und andere Ansätze werden in Issue 281 der Nextcloud Gallery diskutiert: <https://github.com/nextcloud/gallery/issues/281>.

Richtlinie zum „Software-Sprint“

EnergyModels

Schlussbericht

Zuwendungsempfänger:

Jonas Hörsch

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS18S53** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Die Energiewende in Europa und auf der Welt ist in mancher Hinsicht noch ein technisches, aber zu großen Teilen inzwischen mehr ein gesellschaftliches Problem. Durch das komplexe Zusammenspiel zwischen variierender, erneuerbarer Erzeugung, Speichern und Übertragungsleitungen sind die vorgegebenen Referenzszenarien der EU und der Netzbetreiber weder von Bürgerinitiativen, noch von Umweltorganisationen prüfbar und dadurch auch nur schwer diskutierbar. Der undurchsichtige Prozess begünstigt Fehlinformationen und drückt die soziale Akzeptanz. Im Vorhaben EnergyModels soll ein modulares Framework geschaffen werden mit dem sowohl Investitionen als auch der Betrieb von Energiesystemen auf dem Weg zur Dekarbonisierung modelliert und analysiert werden können. Dabei sind Einschränkungen wie globale und sektorale Emissionsziele genauso wie gesellschaftlich gewünschte Ausbaubeschränkungen (bspw. der Übertragungsleitungen) flexibel wählbar.

EnergyModels ist ein Julia Paket, das den Datenfluß und die Struktur eines zu einem Graphen verbundenen Energiesystems mit variabler Zeitauflösung vorgibt. Es stellt die physikalischen Gleichungen für einen Satz von Standardkomponenten (Generatoren, Speichern, Übertragungsleitungen) bereit, erlaubt es aber neue Komponenten hinzuzufügen oder bereits bestehende zu ersetzen. Die Optimierungsvariablen und Zwangsbedingungen werden mit JuMP an quell-offene oder kommerzielle Optimierungssoftware kommuniziert und gelöst.

1. Entwurf und Dokumentation des zugrundeliegenden Komponentenmodells auf neuer Projektwebseite und Einrichtung und Verteilung der zugehörigen Mailingliste
2. Entwicklung der Kernfunktionalität des Datenmanagement und der Kommunikation mit der Optimierungssoftware

3. Erstellung eines Satzes nachvollziehbarer Tutorials in Form von Jupyter Notebooks zur Lösung einfacher Inselformen mit EnergyModels
4. Ausrichtung eines Do-a-thons auf dem Workshop der Open Energy Modelling Initiative
5. Integration der standardmäßigen Plotfunktionalität: Karten, Standardbalken und -kurvenplots
6. Vorbereitung und Lösung eines europäischen und eines deutschen Datensatzes auf Grundlage des PyPSA-Eur Modells
7. Entwicklung der Exportfunktionalität zu einer interaktiven, kartenbasierten Webseite
8. Ausrichtung von zwei Webinaren für zivilgesellschaftliche Organisationen (1) und für erfahrene Energiesystemmodellierer (2)

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

Energiesystemmodellierer*innen in der Wissenschaft und in zivilgesellschaftlichen Organisationen erhalten eine flexibel erweiterbare Arbeitsgrundlage um Dekarbonisierungsszenarien zu entwerfen und Referenzszenarien auf Plausibilität zu testen. Das Projekt wurde mit den Entwicklern der derzeitigen offenen Frameworks zur linearen Energiesystemoptimierung, PyPSA, Calliope, urbs und oemof konzipiert.

Ausführliche Darstellung der Ergebnisse

[EnergyModels.jl](#) wurde als Optimierungsframework auf Github unter einer offenen Lizenz (MIT) veröffentlicht. Die Bibliothek enthält Modelle für die gängigsten Technologien (Generatoren, Speichertechnologien, HVAC- und HVDC-Leitungen) um Elektrizitätsnetzwerke zu optimieren. Neue Technologien können vom Benutzer mit wenigen Funktionsdefinitionen ergänzt werden. Daten werden aus einer einzigen speziell angeordneten NetCDF Dateien on-demand geladen, um den Arbeitsspeicher-Verbrauch gering zu halten. Der Übergang von Julia 0.6 zu 1.0 und der damit verbundenen Umstellung des Optimierungsframeworks JuMP von 0.18 zu 0.19 führte zu erheblichem Programmiermehraufwand für die Fertigstellung des Komponentenmodells und machte es unter anderem notwendig Anpassungen in fremden Bibliotheken (der Datenbibliothek [AxisArrays#160](#) und vor allem des Optimierungsframeworks [JuMP#2003](#)) vorzunehmen und einzupflegen. (Meilensteine 1 + 2).

Auf dem Workshop der Open Energy Modelling Initiative in Aarhus wurde ein [Do-a-thon](#) angeboten, um EnergyModels.jl in die anderen Optimierungsmodelle und -bibliotheken in der Programmiersprache Julia einzuordnen (Meilenstein 4).

Basierend auf dem Workshop-Feedback, dass vorhandene Frameworks für linearisierte Energiesystemoptimierungen generell ausreichen, konkrete Modelle inklusive Daten allerdings schwierig zu finden und überprüfen wären, wurden einige Zielanpassungen vorgenommen:

1. Der Modellgenerator [PyPSA-Eur](#) wurde überarbeitet, um beliebige Länder innerhalb Europas auswählen zu können (z.B. Modell für DE oder BE-NL-FR-DE-AT-CH) und eine ausführliche [Dokumentation](#) wurde erstellt (Meilenstein 6)
2. Die Datenkompatibilität für EnergyModels.jl wurde erweitert um direkt mit PyPSA, wie auch PowerSystems/PowerSimulations (und damit auch MATPOWER, PSS/E) Datensätzen arbeiten zu können.
3. Da Energiesystemoptimierungen mit nicht-linearen Flussmodellen von den verglichenen Frameworks nur unzureichend unterstützt werden, wurde das Komponentenmodell um reaktive Leistungseinspeisungen und eine feingliedrigere Modellwahl erweitert und eine Anbindung an die Flussmodellbibliothek PowerModels.jl begonnen.

Die feingliedrige Modellwahl bedeutet, dass die möglichen Einsatzplanungsmodelle von den Flussmodellen und Kapazitätserweiterungsmodellen getrennt wurden, so dass eine nicht lineare oder auch block-/leitungsscharfe Kapazitätserweiterung mit linearem und blockscharfem Einsatz mit einem linearen Flussmodell kombiniert werden kann (und in naher Zukunft voraussichtlich auch einem nicht-lineare second-order cone basierten Flussmodell).

Die Umkonzeptionierung wurde gut begleitet durch das Betreuungsgespräch mit dem PTF Team und Projektmanagement Coaching durch Simon Höher.

Die Neuausrichtung zu mehr Interframeworkkompatibilität und dem Fokus auf spezialisierten Fähigkeiten, wie der Kombination von PowerModels Flussmodellen mit den Komponenten in EnergyModels, ersetzt die Plotfunktionalität und den Kartenexport die der Benutzer mit anderen Bibliotheken (PowerSimulations.jl, PyPSA oder einfach StatsPlots) realisieren sollte (Meilensteine 5+7).

Am 17.06. fand das OpenMod/NGO bridge meeting in Berlin mit Vertretern von 5 verschiedenen NGOs statt (mehr waren eingeladen) mit dem Ziel eine kollaborative Modellierung des Energiesystems anzustoßen. NGO-Vertreter vertraten klar die Position, dass Sie keine Tools benötigen, die Sie eigener Modellierung unterstützen würden, sondern vor allem wissenschaftliche Ansprechpartner, die zu konkreten Fragestellungen Stellung beziehen und Analysen erstellen würden (statt Webinar 1 in Meilenstein 8).

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Energiesystemmodellierer können nun auf ein klar strukturiertes Komponentenmodell und eine Verknüpfung mehrerer Optimierungsbibliotheken zu einem kohärenten Framework zurückgreifen, die im Programmiercode sehr übersichtlichen Einblick in die verwendeten Gleichungen gibt und auch ohne Veränderung des Hauptprogrammcodes von Benutzer um neue Komponenten erweitert werden kann, die ein oder mehrere Flussmodelle unterstützen.

Die Veröffentlichung des Codes unter der freien MIT-Lizenz ermöglicht die unbedenkliche Nutzung und Weiterentwicklung der Projektergebnisse oder auch Teile der Projektergebnisse in sowohl akademischen als auch kommerziellen Projekten.

Mögliche Weiterentwicklungen sind:

1. Die Fertigstellung der PowerModels-Anbindung ([EnergyModels.jl#21](#)) zur Verwendung nicht-linearer Flussmodelle (wahrscheinlich)
2. Die Bereitstellung eines tabellarischen Interfaces, um übersichtlich Zugriff auf statische und dynamische Systemparameter und die Lösungargumente zu bekommen (basierend auf Tables.jl)
3. Verbesserung der Plot- und Exportfunktionen
4. Erweiterung der eingebauten Komponenten und verfügbare Standardbibliothek an Parametern

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Anbindung an PowerModels.jl um nicht-lineare Flussmodelle zu verwenden, wurde konzipiert, aber noch nicht ausreichend getestet, um in eine neue Version des Optimierungsframeworks übernommen zu werden. Der jetzige Stand ist als sogenannter Pull Request ([EnergyModels.jl#21](#)) diskutierbar veröffentlicht. Fabian Neumann, Doktorand der Energiesystemmodellierungsgruppe am Institute for Automation and Applied Informatics des Karlsruhe Institut für Technologie möchte diese Arbeit weiterführen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Github-Seite von EnergyModels.jl, <https://github.com/PyPSA/EnergyModels.jl>, gibt direkt eine kurze Einführung in das Projekt und Framework. Dort sind auch die bereits gehaltenen Präsentationen inklusive der enthaltenen Tutorials aufgeführt.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Das Projekt schloss die in der „Ausführlichen Beschreibung der Ergebnisse“ dokumentierten Meilensteine in abgerechneten 535 Stunden von den beantragten 540 Stunden ab.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Die Bibliothek [PowerSimulations.jl](#) von NREL, Berkeley, entwickelt und gepflegt, stellt eine sich schnell weiterentwickelnde Alternative zu EnergyModels.jl dar, die bereits nicht-lineare Flussmodelle inkorporiert und eine bessere Anbindung an gängige Datenformate wie PTI (PSS/E) und M (MATPOWER case) bietet. Zusätzlich wurden einige Plot Rezepte in der Bibliothek verankert.

EnergyModels.jl kann daher nun Optimierungen auf Basis des PowerSimulations.jl Datentyp ausführen.

Für die Python-basierte Bibliothek PyPSA wurde eine minimale Bibliothek mit dem Namen [nomopyomo](#) geschrieben, mit der das Kapazitätsexpansions- und Einsatzplanungsproblem mit minimal-möglichem Speicheraufwand ausgeführt werden kann. Eine Erweiterung auf nicht-lineare Flussmodelle ist allerdings aufgrund des gewählten Ansatzes nicht möglich.

Richtlinie zum „Software-Sprint“

Leichtgewichtige und portable Firewall basierend auf MirageOS – Firewall_Qubes

Schlussbericht

Zuwendungsempfänger:

Stefanie Schirmer

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS18S54** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

QubesOS gilt als sicherstes Betriebssystem und wird von Snowden empfohlen. Wir haben eine sparsame Firewall für QubesOS entwickelt. Eine Firewall verbietet durch Paketfilter-Regeln unbefugten Zugriff auf Netzwerkkommunikation und ist wichtiger Teil des Sicherheitskonzepts. QubesOS führt jede Applikation isoliert in einer anderen virtuellen Maschine (VM) aus. Ein Angriff, z.B. des Browsers, bleibt begrenzt auf dessen VM. Die existierende Linux-Firewall für QubesOS belegt viel Arbeitsspeicher, der unter allen VMs geteilt werden muss und daher knapp ist. Unsere Firewall basierend auf dem Minimal-Betriebssystem MirageOS belegt weniger Arbeitsspeicher und bietet weniger Angriffsfläche.

Die ersten Meilensteine wurden erreicht, d.h. die Entwicklung der State-Machine zur Klassifizierung der Netzwerk-Pakete, inklusive guter Testabdeckung und Fuzz-Testing. Ausserdem wurden die genannten Bauteile zu einem Unikernel kombiniert der Veränderungen der Filterregeln von Qubes entgegennehmen kann. Darüberhinaus wurde eine DNS-Client-Bibliothek integriert, damit Filterregeln bequem als menschenlesbare Domainnamen geschrieben werden können und nicht wie zuvor als maschinenlesbare IP-Adressen.

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

Die Zielgruppe unseres Projektes sind Aktivisten, Journalisten, Security-Professionals und andere Menschen die das sichere Betriebssystem Qubes nutzen. Sie können durch die Leichtgewichtige Firewall besser arbeiten, da diese weniger Speicher verbraucht, und bequem Filterregeln anpassen, ohne dass die Firewall neu gebaut werden muss. Aufgrund

von Speichermangel waren in der Vergangenheit viele Nutzer gezwungen, die Firewall abzuschalten. Die neue, leichtgewichtige Firewall fuehrt zum sicheren Einsatz des Netzwerks in QubesOS und schuetzt vor unbefugtem Netzwerkverkehr und Hacker-Angriffen.

Es gibt keinen direkten Bezug zum speziellen Thema der Runde, das Projekt ist eher allgemein und im Bereich Netzwerk-Sicherheit und Software-Infrastruktur angesiedelt.

Ausführliche Darstellung der Ergebnisse

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

Die Firewall ist soweit komplett und funktioniert, die Testabdeckung ist auch sehr gut.

Automatisiertes Fuzz-Testing wurde angewandt, dieses testet vor allem die Network-Address-Translation-Bibliothek auf der die Firewall basiert. Diese bestehende Bibliothek wurde auch im Projekt „refactored“, also aufgeraeumt.

Die Meilensteine Paketklassifikation und Unikernel wurden erreicht. Es gab noch einen ueberraschenden neuen Punkt der die Arbeit etwas aufgehalten hat, und das war der Einbau der DNS-Bibliothek. Diese musste erst an unsere benoetigte Architektur angepasst werden, was eine Zusammenarbeit mit vier Entwicklern erforderte.

Der letzte Meilenstein, Veroeffentlichung als Template in QubesOS, ist noch nicht erledigt, da es auch hier einige Hindernisse gibt. Qubes benutzt ein anderes Virtualisierungsschema in seiner neuesten Version (pvh anstatt pv). MirageOS kann damit noch nicht umgehen, so dass dort ein anderer Weg gefunden werden muss.

Es ist bereits moeglich, die Firewall auf Github herunterzuladen und selbst in Qubes zu bauen. Der naechste Schritt waere allerdings bequemer, da man dann die Firewall direkt aus einem Menu als Qube auswaehlen koennte. Dieser Schritt ist in Arbeit.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Alle Qubes-Nutzer koennen die Leichtgewichtige Firewall nutzen und so Speicher einsparen. Unsere Firewall ist auch deshalb beliebt, da sie in OCaml, einer besonders sicheren Programmiersprache geschrieben ist. Es gibt bereits einige Nutzer.

In der Weiterentwicklung steht wie beschrieben die Eingliederung in QubesOS noch an. Insgesamt hat micht die Arbeit sehr weitergebracht, da ich zum ersten Mal im Feld Netzwerksicherheit arbeiten konnte, nachdem ich mich schon jahrzehnte lang damit in meiner Freizeit beschaefte. Der Umstieg in dieses recht anspruchsvolle Feld waere ohne dieses Projekt nicht so einfach moeglich gewesen.

Auch die Zusammenarbeit mit der Qubes-Community war fuer mich neu und spannend.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Es gab eher unvorhergesehene Seitenprobleme, wie z.B. das refactoring fuer die DNS-Bibliothek. Bei der Planung dachte ich, sie sei direkt ohne Umbau fuer uns nutzbar.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Es wurde komplett offen auf Github gearbeitet, das Haupt-Repository ist

<https://github.com/yomimono/qubes-mirage-firewall>

vor allem im Branch static-pf-rules.

Ein Vortrag ueber die Firewall wird Ende Oktober in New York stattfinden:

<https://radicalnetworks.org/archives/2019/participants/stefanie-schirmer/>

Es wird einen Videostream im Internet geben.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Einbau der DNS-Bibliothek hat den Rest verzögert, so dass das integrieren in QubesOS noch nicht geschafft wurde.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

QubesOS bringt einige Anforderungen mit sich, an die wir uns mit der Firewall anpassen muessen (QubesDB fuer Updates, Virtualisierungsumgebung, Build-Chain). Das war allerdings von Anfang bekannt und wir haben uns auch getroffen um uns abzustimmen, und werden an der Integration gemeinsam weiterarbeiten.

Richtlinie zum „Software-Sprint“

YV – Your Voice

Schlussbericht

Zuwendungsempfänger:

Thomas Werkmeister

DLR PT- Berlin
Eing.am:

01. Okt. 2019

Eingangsnr.: 5452 137

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S55 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Meine Motivation mit dem Projekt war auf neue Entwicklungen in der Sprachsynthese mittels neuronaler Netze hinzuweisen, die es ermöglichen natürliche und emotionale Sprache zu erzeugen und dabei auch die Stimmen von Personen nachempfinden können. Diese Technologie hat viele sinnvolle Anwendungen, aber bietet auch viel Raum für Missbrauch.

Die wichtigsten Meilensteine waren:

- Reproduktion der Ergebnisse aus den wissenschaftlichen Papieren von Google und der Tacotron Architektur für die Deutsche Sprache
- Verfeinerung der Ergebnisse bezüglich Natürlichkeit, Ausdrucksvermögen (Emotionen), und Anzahl der Sprecher
- Veröffentlichung einer Website, die die besagte Technik erläutert und ausprobierbar macht

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

Die Zielgruppe meines Projektes ist die an Technologie interessierte Allgemeinheit, Berufsgruppen die mit Missbrauch der Technologie in Kontakt kommen könnten, sowie Entscheider. Das Projekt YourVoice sollte sie auf die möglichen Konsequenzen der Technologie hinweisen und gleichzeitig technisches Wissen vermitteln, sodass sinnvolle Entscheidungen getroffen werden können.

Ausführliche Darstellung der Ergebnisse

Leider hat sich mein Projekt als weitaus forschungslastiger, komplexer und schwieriger herausgestellt als anfänglich gedacht. Ich habe es auch bis heute, Ende September, nicht geschafft den ersten Meilenstein zu erreichen und die Ergebnisse für Deutsch zu reproduzieren. Das in dem Kontext, dass ich weit über die veranschlagten Stunden gearbeitet habe und außerdem noch Rückenwind durch eine Kooperation mit Mozilla und der Telekom bekommen habe, die im Austausch von Fachwissen bestand. Das Projektziel ist nicht unmöglich, und ich habe auch große Fortschritte gemacht und diese auch zur offenen Implementierung von Mozilla beigetragen. Dennoch ist das Ziel sehr hoch gesteckt gewesen, da nicht alle Faktoren vollständig kontrollierbar waren. Vor allem hat sich die Datenqualität der bisher frei verfügbaren Datensätze für die Deutsche Sprache als ungenügend herausgestellt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Leider gibt es bis heute keine Ergebnisse im Sinne der ursprünglichen Zielstellung. Dennoch wurden Schritte in die richtige Richtung übernommen. So habe ich viele Fehler in der öffentlichen Implementierung der neuronalen Sprachsynthese von Mozilla behoben, diese für die Verwendung von Sprachen außer Englisch verbessert, die Verwendung von mehreren Sprechern für das Training implementiert, die Lerngeschwindigkeit bis zur Konvergenz stark gesteigert, und neue Lösungsansätze für besonders fragile Teile des neuronalen Netzwerks entwickelt. Darüber hinaus habe ich hunderte Experimente durchgeführt, um die mangelnde Datenqualität durch bessere Hyperparameter und Architektur des Netzwerks auszugleichen. Diese haben aber doch noch nicht den erhofften Erfolg gebracht.

Auch zu diesem Zeitpunkt, Ende September, bin ich noch mit dem Projekt beschäftigt und unter anderem daran weitere Datensätze für die deutsche Sprache öffentlich verfügbar zu machen oder möglicherweise, finanziert durch Dritte, erstellen zu lassen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Es hat sich herausgestellt, dass der Weg zur Lösung länger und beschwerlicher ist, als das ein sechsmonatiges Projekt zulässt. Was besonders viel Zeit gekostet hat, war die Verwendung von ungeeigneten Datensätzen, wie z.B. des Common Voice Datensatzes von Mozilla, der zwar für Spracherkennung gut geeignet ist, aber aufgrund der geringen Audioqualität und der geringen Menge an Material, wenig für die Sprachsynthese zu bieten hat. Aber auch andere offene Datensätze wie z.B. gelesene Audiobücher von librivox, die mehr Material von einem einzelnen Sprecher haben, haben noch nicht das gewünschte Ergebnis gebracht. Hier wirken sich Segmentierungsfehler, Atemgeräusche, und Versprecher negativ auf die Leistung des Systems aus. Für einen dieser Audiobuch Datensätze habe ich in knapp 4000 von 17000 Audiodateien Segmentierungsfehler mit einem halbautomatischen heuristischen Verfahren korrigiert, leider ohne direkt messbaren Einfluss auf die Resultate.

Natürlich zählen hierzu auch etliche Experimente, die bisher noch zu keiner Lösung geführt haben, mich aber mehr und mehr mit der Architektur vertraut gemacht haben. Die Architektur des neuronalen Netzes ist sehr komplex und besteht aus 5 Subnetzen, die sich je auf verschiedenste Weisen gestalten lassen. Insgesamt besteht das Netz aus mehr als 60 Schichten. Dazu kommen nochmal 40 bis 50 Hyperparameter für die Vorverarbeitung der Sprachsequenzen, Texte, des Lernverhaltens sowie der konkreten Ausprägung der verschiedenen Schichten. Darüberhinaus ist

durch die sequentielle Natur der Aufgabe, d.h. die schrittweise und nicht parallelisierte Erstellung der gesprochenen Sprache, ein einzelnes Training wenigstens 24 Stunden aber eher mehrere Tage dauert. Die Vorbereitung, Auswertung, Analyse von Fehlern, und Korrektur, Konzipierung neuer Experimente etc. hat äußerst viel Zeit beansprucht. Dennoch haben die Ergebnisse die gesteckten Ziele bisher nicht erreichen können.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Zu diesem Zeitpunkt gibt es hierfür das Repository von Mozilla und meinen persönlichen Fork auf dem ich sehr viele Experimente basiert habe.

<https://github.com/mozilla/TTS>

<https://github.com/twerkmeister/TTS>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Ich bin mit meiner Arbeitszeit deutlich über die veranschlagten Stunden gekommen. Mit fast 200 Stunden Mehraufwand als ursprünglich geplant. Diesen Mehraufwand war ich bereit zu leisten, da ich weiterhin vom Wert des Projektes überzeugt bin.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Hier ließe sich die Zusammenarbeit mit Mozilla und der Telekom anführen. Die vielen fachlichen Diskussionen haben unterstützend auf das Projekt gewirkt. Die Ziele meines Projektes standen dabei für mich an erster Stelle. Das habe ich offen kommuniziert und glücklicherweise haben sie sich auch mit denen der besagten Unternehmen gedeckt. Diese waren zwar nicht an der eigentlichen Aufbereitung und Erklärung der Ergebnisse interessiert, aber an der Verwendung an den technischen Zwischenergebnissen die bis dorthin erbracht werden müssen. Dazu zählen v.A. die bisher erarbeiteten technischen Verbesserungen der offenen Sprachsynthese Software. Durch die offene Entwicklung stehen diese Verbesserungen natürlich auch vielen anderen Entwicklern offen. Das Repository von Mozilla hat über eintausend Interessenten.

Richtlinie zum „Software-Sprint“

Datenklaus

Schlussbericht

Zuwendungsempfänger: Krohe, Dehm, Hohbach GbR

Nastasja Krohe

Julian Dehm

Annemarie Hohbach

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S56 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Obwohl nutzerorientiertes Anwendungs-Wissen über Daten und Datenschutz einen immer wichtigeren Stellenwert einnehmen sollte, existiert kaum schularten- und fächerübergreifende Infrastruktur zur Heranführung an das Thema. Der Datenklaus-Prototyp bedeutet einen Schritt in die richtige Richtung zur Erfüllung dieses Desiderats: Unabhängig vom Kenntnisstand der betreuenden Lehrkraft einsetzbar, lassen sich mithilfe der App und ihren thematischen Modulen niedrigschwellig . interessante Lehr-Szenarien organisieren. Theoretisch und praktisch erlernen Schüler_innen dabei, wie sie ihre Daten schützen können. Auch die hierfür notwendigen Internet-Basics werden vermittelt. Alle Gestaltungsebenen orientieren sich an den Bedürfnissen aller Beteiligten (Lehrende und Lernende).

- *interdisziplinäre Entwicklung von Lehr-Inhalten
- *Direkte und indirekte Kontaktaufnahme zu Testimonials und Multiplikator_innen in Schulen, Organisationen, etc.
- *Grundgerüst programmieren und testen
- *Fertigstellung eines ansprechenden User Interface: Design, Dialoge und Comics
- *Ausarbeitung eines Manuals
- *Beratung durch Demokratie- & Medienpädagogen aus unserem Netzwerk (zugesichert)
- *Pre-Tests zur Zusammenstellung von bisher unberücksichtigten Ansprüchen und Wünschen aus der Zielgruppe
- *Auswertung und Evaluation des Prototyps

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

Der Datenklaus Prototyp ist ein Beitrag zur Umsetzung von Data Literacy im Unterricht. Die App richtet sich an Lehrkräfte und Workshop-Leiter_innen in ihrer Rolle als proaktive Gatekeeper_innen zur Weitergabe von Wissen sowie indirekt – nichtsdestotrotz auch gleichermaßen – an Schüler_innen einer weiterführenden Schule, welche die 5.-6. Klasse besuchen (äquivalent zu den Gymnasial-Lehrplänen in Sachsen). Erfahrungen und Wünsche von Gatekeeper_innen wurden im Entwicklungsprozess berücksichtigt. Der Open Source Code von Datenklaus steht beliebigen Personen zur Verwendung und Erweiterung offen.

Ausführliche Darstellung der Ergebnisse

Der Datenklaus Prototyp beinhaltet erste Module für Lehr-Szenarien, welche auch mit dem Thema weniger vertrauten Lehrkräften und Workshop-Leiter_innen eine Behandlung von Internet-Basics und Datenschutz im Unterricht ermöglichen.

Die App erlaubt die Erstellung eines Raums für jede Lern-Gruppe und einen anonymisierten Log-In für Teilnehmende. Im Backend für Lehrende ist der Fortschritt der jeweiligen Gruppe nachvollziehbar, was zur Erleichterung der Moderation von Lehr-Szenarien dient. Außerdem ermöglicht das Backend Unterbrechungen zum Zweck von Zwischenbesprechungen off-screen.

Die App beinhaltet ein Lexikon, welches alle verwendeten Fachbegriffe und Fremdwörter in möglichst einfach gehaltener Sprache behandelt. Das Lexikon lässt sich im Ganzen aufrufen; zusätzlich sind die Lemmata in den Modulen direkt verlinkt und unmittelbar abrufbar. Außerdem wurde ein Diceware Game realisiert, anhand dessen Lernende spielerisch dazu befähigt werden, sich ein sicheres Passwort zu erstellen.

Das zielgruppenorientierte User Interface Design im Front-End unterstützt die abstrahierte Vermittlung komplexer Sachverhalte, beispielsweise in Form von Comics.

Die Begleitung durch die Open Knowledge Foundation eröffnete dem Team von Datenklaus neue Erkenntnisse zur Umsetzung eines Designs, in dessen Mittelpunkt die Nutzer_innen stehen.

Die pädagogische Evaluation des Prototyps konnte noch nicht während des Förderzeitraums realisiert werden, wird aber in geraumer Zeit aber von einer Lehrerin und Fortbilderin in Demokratiepädagogik für Lehrkräfte durchgeführt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Aufbauend auf dem Open Source Prototypen sollen der Code und die Inhalte der App communitybasiert weiterentwickelt werden. Interessierte können sich den Code aber auch vollständig aneignen und eine App mit komplett neuen Lehr-Inhalten erstellen.

Die Jugendpresse Deutschland möchte Datenklaus im Rahmen von Workshops der Mobilen Medienakademie ins Klassenzimmer bringen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Trotz der Mitgliedschaft im Netzwerk Digitaler Schulen in Sachsen (NetDiS) und einer persönlichen Vorstellung des Projekts in diesem Rahmen hat es sich als schwierig erwiesen, konkrete Zusagen und Hilfestellungen von Lehrkräften (vorwiegend in Tabletklassen) schon in der Entwicklungsphase zu bekommen. Dies mag einerseits dem Umstand geschuldet sein, dass während des Förderzeitraums noch kein fertiges Produkt zur Demonstration und Illustration des Vorhabens vorlag. Andererseits muss erwähnt werden, dass einige Lehrkräfte nicht dazu bereit sind, einem jungen, interdisziplinären, lediglich mit nicht-schulischen Lehr-Szenarien vertrauten Team von Entwickler_innen einen Vertrauensvorschuss zu gewähren, um in proaktiven Austausch zu treten.

Die erfolgreiche Kooperation mit der Jugendpresse Deutschland während der Förderphase kompensiert dieses Manko: Die Geschäftsführerin des Vereins sowie die Projektleiterin der Mobilen Medienakademie waren auch ohne Vorab-Demonstration von der Vision der Entwickler_innen überzeugt. So durfte das Datenklaus-Team einen zweitägigen Workshops für Peer-Scouts der Mobilen Medienakademie geben und das Vorhaben vorstellen. Der Kreis der Multiplikator_innen konnte durch den Workshop also um die Peer Scouts der Mobilen Medienakademie erweitert werden, welche die App für ihre eigenen Workshops nutzen wollen.

Durch Projekt-Praktika am Institut für Informatik der Universität Leipzig wurden außerdem Lehramts-Studierende auf das Projekt Datenklaus aufmerksam.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Unter datenklaus.org findet sich eine Vorstellung des Projektes sowie ein Link zu GitHub.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Der Zeitrahmen der Förderphase hat nicht ausgereicht, um Module Inhalte im ursprünglich vorgesehenen Umfang zu erstellen. Daher ist es angedacht, auf ehrenamtlicher Basis weitere Inhalte einzuspeisen sowie aktive, permanente Contributor_innen zu gewinnen.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Im Rahmen des Projekt-Praktikums bei Datenklaus legte ein Lehramtsstudent eine wissenschaftliche Arbeit über Anforderungen an eine App für Lehr-Szenarien vor. Die Ergebnisse flossen in die Entwicklung mit ein. Weiterhin wurde das Feedback von Lehrkräften umgesetzt.

Eine fachlich-pädagogische Evaluation durch eine Lehrkraft und Fortbilderin für andere Lehrkräfte steht noch aus.

Richtlinie zum „Software-Sprint“

TRANSKRIPT

Schlussbericht

Zuwendungsempfänger:

Matthias Hannich

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S57 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Qualitativ hochwertige Transkripte gesprochener Sprache sind im Moment nur mit hohem finanziellen und personellem Aufwand erzeugbar. Das Projekt TRANSKRIPT vereinheitlicht und vereinfacht den Erzeugungsprozess eines Transkripts, indem es eine Web-Plattform bereitstellt, auf der Menschen ohne Expertenwissen Transkripte erzeugen und korrigieren können. Dieser Prozess wird durch Bereitstellung automatischer Spracherkennung ergänzt.

Dadurch haben unkommerzielle Medienproduzent*innen, Podcaster*innen und andere Initiativen aus der Zivilgesellschaft die Möglichkeit, die Reichweite ihrer aufgezeichneten Sprachaufnahmen (z.B. für Menschen mit Hörbeeinträchtigungen) zu erhöhen und Transkripte verfügbar zu machen.

Das Projekt teilte sich in zwei wesentliche Arbeitspakete: Erstellung eines Editors zum Transkribieren und Korrigieren bzw. Anbindung automatischer Spracherkennungssoftware, die Hypothesen für das Transkript generieren kann. Letztere sollen im Editor korrigierbar sein.

Der Editor-Teil wurde unter Nutzung bzw. Erweiterung des kollaborativen Texteditors Etherpad-Lite realisiert. Für die Anbindung der Spracherkennungssoftware Kaldi wurde kaldigstreamer-server genutzt.

Zur Umsetzung des Vorhabens mussten die einzelnen Workflows (Segmentierung der Audiodatei zur Integration in den Editor, manuelles Transkribieren bzw. Import vorhandener Transkripte, Korrektur und Vorgaben für lesbare Verschriftlichungen, Integration automatisch generierter Hypothesen) klar voneinander getrennt werden. Aus den Anforderungen der einzelnen Workflows wurden die notwendigen, zu implementierenden Features ermittelt.

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

Zielgruppe sind alle, die Sprachaufnahmen selbst erzeugen oder aufzeichnen. Diese können durch Unterstützer*innen Transkripte erzeugen lassen oder selbst erzeugen und die Reichweite ihrer Themen erhöhen. Tondokumente können dadurch nicht nur angehört, sondern gelesen werden.

Darüber hinaus können Forscher*innen die zugänglich gemachten Audio-Aufnahmen zum Training eigener Modelle nutzen. Neue, in Modellen bisher unberücksichtigte Aufnahmen (z.B. freieradios.net mit unter by-nc-sa Creative Commons-Lizenz veröffentlichten Beiträgen) können dafür genutzt werden. Diese sind in großem Umfang Spontanaufnahmen – im Gegensatz zu den bereits verfügbaren aus Büchern oder Wikipedia vorgelesenen Datensätzen. Die daraus entstehenden Modelle sind demzufolge besser für die Erkennung spontaner Sprache geeignet.

Ausführliche Darstellung der Ergebnisse

Es wurde ein Plugin für den kollaborativen Editor Etherpad-Lite erstellt, mit dem einzelne Segmente einer Tondatei in einem ersten Schritt transkribiert und in einem zweiten Schritt durch andere Nutzer*innen korrigiert werden können. Das Plugin enthält grundlegende Funktionalität, um Audio-Dateien innerhalb des von Etherpad-Lite bereitgestellten Editors abspielbar zu machen und dabei segmentweise vorzugehen.

Die Software kaldi-gstreamer-server, mit der mittels Kaldi fertig trainierte Modelle genutzt werden können, wurde im Anschluss so eingebunden, dass diese für die einzelnen Segmente Hypothesen generiert. Diese Komponente erscheint in Etherpad-Lite als normaler Autor, weshalb aus Sicht der Nutzer*innen kein wesentlicher Unterschied zwischen den von Menschen erzeugten Transkripten und den durch die Spracherkennungssoftware erzeugten Hypothesen besteht.

Ergänzt wurde der an die Spracherkennungssoftware angebundene Webeditor durch einen prototypischen Crawler, mit dem mehrere Audioaufnahmen segmentiert und in den Editor eingepflegt werden können.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Unkommerzielle Medienproduzent*innen erhalten durch das Tool die Möglichkeit, ohne Kosten automatisch Transkripte ihrer Aufnahmen zu erstellen bzw. diese selbst oder durch Unterstützer*innen transkribieren und korrigieren zu lassen.

Im Rahmen des Projekts wurden vorhandene Aufnahmen indiziert, die als Creative Commons auf freie-radios.net lizenziert veröffentlicht wurden. Daraus ergeben sich neue Rohaufnahmen für die Integration in vorhandene Korpora. Sprachwissenschaftler können daraus neue Modelle trainieren.

Eine Weiterentwicklung der Plattform zur Umsetzung einer Volltextsuche für Audiodateien wäre denkbar. Der Schwerpunkt würde sich dann verlagern von einzelnen Dateien, die Nutzer*innen selbst hochladen bzw. verlinken können, hin zu einzelnen Segmenten, die zum Transkribieren bzw. Korrigieren vorgeschlagen werden.

Der Editor muss im Funktionsumfang erweitert werden, damit Funktionalitäten (Rechtschreibung, Grammatik-Korrektur, Abkürzungen, usw.) abgedeckt werden, die vorhandene (nicht browserbasierte) Anwendungen wie LibreOffice bereits enthalten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ursprünglich kannte ich nur oTranscribe als eine Möglichkeit browser-basiert Transkripte zu erstellen, weshalb es naheliegend schien, dieses Tool zu erweitern. Im Zuge anfänglicher Recherchen und durch Interviews mit potenziellen Nutzer*innen bzw. Medienproduzent*innen wurde schnell deutlich, dass es verschiedenste Tools bereits gibt bzw. dass es eine Vielzahl an Wünschen gibt, die durch eine solche Plattform abgebildet werden sollen. oTranscribe ist als privacy-preserving WebApp konzipiert – Dateien verlassen niemals den Browser. Diese Eigenschaft wird durch die kollaborative Plattform prinzipiell verletzt, weil Dateien entweder bereits online veröffentlicht sind oder direkt in die Plattform geladen werden bzw. andere Nutzer*innen die Transkripte einsehen können.

Das wavesurfer.js-Projekt kann hingegen alternativ als Bibliothek genutzt werden und bietet einen ähnlichen Funktionsumfang wie oTranscribe.

Die Planung des gesamten Projektzeitraums in einzelnen Arbeitsschritten war herausfordernd. Aufgabenverteilung im Team hätte den Entwicklungsprozess vereinfacht.

Vor allem die Anbindung externer Spracherkennungssoftware ist komplizierter, als in der Planungsphase erwartet. Durch Nutzung von [kaldi-gstreamer-server](https://github.com/kaldi-gstreamer-server) konnten verschiedene Modelle integriert und deren Hypothesen verwendet werden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Softwarekomponenten (Editor-Plugin, Crawler für Audio-Ressourcen, Schnittstelle zu kaldi-gstreamer-server) und die Serversoftware zum Aufsetzen der Plattform werden unter <https://github.com/transkription> veröffentlicht. Der Release der Plattform ist bis zum Jahresende 2019 geplant, da insbesondere Bugs im Zusammenhang mit der Korrektur-Funktionalität und die bisher nicht abgeschlossene Integration mehrerer Modelle eine sinnvolle Nutzung blockieren.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Aus ersten Gesprächen bzw. Interviews mit potenziellen Nutzer*innen ergab sich der Bedarf, bereits vorhandene Transkripte integrieren zu können, auch wenn diese fehlerhaft sind oder nur in Form von Sendungsskripten vorliegen. Die Möglichkeit bereits vorhandene, außerhalb der Plattform erzeugte Textdateien mit den Sprachaufnahmen zu verknüpfen ist sinnvoll, um auch Nutzer*innen erreichen zu können, die nicht zwangsweise mit der Plattform transkribieren möchten, diese aber zum Beispiel zur Korrektur von Transkripten nutzen wollen. Der Zeitaufwand für eine prototypische Umsetzung dieser Import-Funktionalität lag bei ca. drei Wochen, jedoch konnten daraus Erkenntnisse in den Meilenstein „Anbindung eines Spracherkenners“ fließen, wodurch dieser entsprechend verkürzt wurde.

Ursprünglich geplant war, auch den Trainingsprozess für neue Modelle abdecken zu können. Die Komplexität dieses Arbeitspakets wurde unterschätzt, so dass unter Berücksichtigung des Zeitrahmens nur die automatische Erweiterung um eigene Wörter bzw. Wortgruppen, die von der Spracherkennungssoftware im Anschluss erkannt werden können, implementiert wurde und nicht die Erweiterung der akustischen Modelle. (Meilenstein 5)

Desweiteren war geplant nicht nur mit einen, sondern mit mehreren Modellen verschiedene Hypothesen zu generieren, so dass Nutzer*innen zwischen den besten auswählen können. Das ist eine Herausforderung unter dem Gesichtspunkt der Übersichtlichkeit innerhalb des Editors, weshalb für den Prototyp vorübergehend nur ein Modell gleichzeitig angebunden, jedoch zwischen verschiedenen Modelle gewählt werden kann. (Meilenstein 4)

Die Arbeitsstunden teilten sich innerhalb des Projektzeitraums im Wesentlichen auf die zwei großen Arbeitspakete Entwicklung des Editors und Integration bzw. Anbindung der Spracherkennungssoftware gleichmäßig auf (Meilensteine 1-3 bzw. 4-6).

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Der Bereich Spracherkennung ist durch eine Vielzahl aktueller Entwicklungen gekennzeichnet, die Lösungsansätze für Teilprobleme liefern oder ganze Workflows im Umgang mit Transkripten abbilden sollen. Es existieren vorhandene Ansätze zur Korrektur automatisch generierter Transkripte bzw. zur Umsetzung von Voting-Verfahren, um das beste Transkript zu ermitteln (z.B. NYPL Transcript Editor oder BBC Transcript Editor).

Auch für das Einpflegen der Transkripte unter Zuhilfenahme unterschiedlicher Editoren gibt es bereits Weblösungen (z.B. OCTRA von der Uni München).

Der Vielzahl an Ansätzen wurde innerhalb des Projekts insofern Rechnung getragen, als dass die einzelnen Bestandteile optional in Form von Komponenten über eine möglichst generische API integrierbar sein sollen. Es soll in Zukunft möglich sein, einzelne Editoren auszutauschen, sowohl in Hinblick auf den Transkriptionsprozess, als auch auf den Korrekturprozess.

Da die vorhandenen Tools meist als Einzelwerkzeuge und nicht als modulare Komponenten entwickelt werden, ist deren (Code-)Komplexität relativ groß, weshalb sie nicht von vorn herein integriert werden konnten. Eine punktuelle Anpassung der vorhandenen Lösungen konnte nicht

erfolgen, weil die Einarbeitungszeit im zur Verfügung stehenden Projektzeitraum einen unverhältnismäßigen Schwerpunkt gesetzt hätte.

Die Zielgruppe vieler Tools ist noch zu sehr auf ein Fachpublikum aus dem linguistischen oder phonetischen Bereich zugeschnitten.

Richtlinie zum „Software-Sprint“

VCAT – Visual Collections and Training Data

Schlussbericht

Zuwendungsempfänger: Adam Harvey / Jules LaPlace

Name des Zuwendungsempfängers: Adam Harvey / Jules LaPlace

Das Projekt welches diesem Bericht zugrunde liegt wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS18S58** ermöglicht. 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 Grundmotivation die zur Entwicklung des VCAT-Projekt geführt hat, ist ein Mangel an Trainingsdaten die für Computer-Vision genutzt werden können. Trainingsdaten sind ein wesentlicher Bestandteil von Neuronalen Netzen, die im Bereich des Computer Vision eingesetzt werden. Die öffentlich zugänglichen Datensätze sind jedoch generisch und nicht auf die speziellen Forschungsziele von Menschenrechtsgruppen und investigativen Journalisten anwendbar. Das VCAT-Projekt wurde entwickelt, um eine Open-Source-Softwarelösung für dieses Problem bereitzustellen, indem 3D-modellierte synthetische Bildtrainingsdaten entwickelt wurden. Das Projekt hatte zum Ziel, zwei Anwendungen zu entwickeln: eine zur Generierung synthetischer Daten und die andere zum Aufbau einer visuellen Suchmaschine zur Bereitstellung der Bildverarbeitungsmodelle.

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 des Projekts sind Forschungsgruppen, die mit visuellen Medien aus Konfliktzonen arbeiten. Diese Gruppen können von VCAT profitieren, indem sie das synthetische Datensystem verwenden, um sehr spezifische neuronale Objekterkennungsnetze aufzubauen.

Des Weiteren profitieren die Zielgruppen, indem sie die browserbasierte visuelle Suchmaschine verwenden, um große Videoarchive zu verarbeiten.

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?

VCAT erreichte sein Hauptziel, die Entwicklung einer Open-Source-Prototyp-Softwareanwendung und -Methodik zur Generierung synthetischer Bildtrainingsdaten für Objekte in Konfliktzonen von Syrien und Jemen. Die beiden Hauptanwendungen sind: das synthetische Datengenerierungssystem und die visuelle Suchmaschine.

Das synthetische Datensystem umfasst eine Sammlung von Python-Skripten und -Anwendungen, die mit einer Blender 3D-Modellierungssoftware 3D-Szenen erzeugen und automatisch in annotierte Bilder umwandeln, die für die Erkennung von Trainingsobjekten und Algorithmen zur Bildklassifizierung verwendet werden. Das synthetische Datengenerierungssystem kann eine Vielzahl von Variablen **randomisieren**, einschließlich Hintergrund, Himmel, Boden, Objektposition, Beleuchtung, Drehung, Größe des Kamerasensors, Zoom der Kamera und Farbtemperatur.

Nach einer Auswertung anderer 3D-Modellierungssoftware wurde das Programm Blender wegen seiner hochwertigen Renderings, der Open-Source-Lizenz und der Möglichkeit, aus der Ferne zu rendern, ausgewählt. Dadurch kann das VCAT-System auf entfernten Cloud-Servern betrieben werden, um die Vorteile einer höheren Rechenleistung beim Erzeugen großer Datensätze zu nutzen.

Die finale Ausgabe des Systems, wie auf der GitHub-Seite mit synthetischen Daten (https://github.com/vframeio/vframe_synthetic) dokumentiert, umfasst die fotorealistischen Renderings, farbcodierte Annotationsbilder, CSV-formatierte Annotationsdateien und animierte Demos, die das Beheben von Fehlern und Anzeigen der Ergebnisse erleichtern. Das System ermöglicht die Generierung synthetischer Daten mit 2 verschiedenen Render-Engines: Cycles und Eevee. Ersteres wird für hochwertiges Rendering und letzteres für Prototyping verwendet. Die Renderzeiten pro Bild für hochwertige Daten reichen von 15-200 Sekunden, je nach Bildgröße und Hardware. Bei Daten geringerer Qualität betragen die Renderzeiten mit der Eevee Render-Engine typischerweise 1-10 Sekunden pro Bild. Während der Entwicklung verwendete VCAT zwei NVIDIA GTX1080-Ti GPUs mit einem Intel i7-7700 für das Rendering.

Die zweite Hauptkomponente des Projekts ist die visuelle Suchmaschine. Die Suchmaschine ist einzigartig, weil sie einen ModelZoo oder eine Sammlung von Modellen beinhaltet, die erweitert werden können. Der Administrator der visuellen Suchmaschine kann dann aus verschiedenen Modellen auswählen, um Merkmalsvektoren zu berechnen, die für die Suche nach einer Sammlung von Videos und Bildern verwendet werden. Für spezifische Untersuchungen können verschiedene Merkmalsvektoren verwendet werden. So könnte beispielsweise ein Tankdetektor mit 3D-Modellen von Tanks trainiert und dann verwendet werden, um einen Suchindex zu erstellen, der in der Lage ist, zwischen verschiedenen Tanktypen zu unterscheiden und dem Benutzer genauere interaktive Suchergebnisse zu liefern.

Durch die Entwicklung des Projekts haben wir Einblicke in Bereiche gewonnen, in denen die Computer Vision Industrie die Zugänglichkeit fehlt, nämlich durch die Bereitstellung von Open Source Datensätzen und einer Schnittstelle zur Nutzung von Open Source Model-

len mit handelsüblicher Desktop-Hardware. Es ist in diesem Bereich in dem wir der Meinung sind, dass VCAT einen wesentlichen Beitrag zur Unterstützung von Menschenrechtsforschern und investigativen Journalisten leisten kann.

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?

VCAT erreichte sein Hauptziel, in dem es eine Softwareanwendung, 3D-Modelle sowie eine Methodik zur Generierung synthetischer Bildtrainingsdaten für Objekte in Konfliktzonen von Syrien und Jemen entwickelte. Durch die Entwicklung haben wir Erkenntnisse darüber gewonnen, wo die Bildverarbeitungsindustrie nicht zugänglich ist, nämlich bei der Bereitstellung von Open-Source-Datensätzen. Wir glauben, dass VCAT hier einen wesentlichen Beitrag leisten 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?

Wir fanden heraus, dass die Entwicklung neuer manueller Annotationswerkzeuge nicht notwendig war. Es gibt bereits viele bestehende Tools in diesem Bereich. Die Annotierung durch externe Arbeitskräfte ist teuer. Und die Weitergabe von Daten an Fremdfirmen ist rechtlich nicht möglich. Stattdessen werden die synthetischen Datenobjektdetektoren verwendet, um neue Datenquellen semi-automatisch zu annotieren.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Unter <https://vframe.io/research/synthetic-datasets/> finden Sie Informationen über die Forschung und unter <https://github.com/vframeio/> finden Sie den Code.

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 haben den Zeitplan geändert, um mehr Zeit für die Entwicklung der 3D-Modelle zu haben. Damit wurde fast ein Drittel des ursprünglich für die Entwicklung des Codes vorgesehenen Budgets verbraucht und es war schwieriger als ursprünglich geplant. Des Weiteren haben wir von der Verwendung von Unity und Unreal Engine auf Blender für die 3D-Entwicklung umgestellt, weil es Open-Source-freundlicher ist. Dies dauerte länger, da sich die

3D-Software während unserer Entwicklung noch in der undokumentierten Beta-Phase befand.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

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

Während der Entwicklung beobachteten wir aufkommende Trends in den Bereichen KI/ML und Computer Vision und stellten fest, dass es ein stetiges Wachstum bei den Verbesserungen der Techniken zur Objekterkennung und Bildklassifizierung gab, aber wenig Wachstum bei den öffentlich zugänglichen Trainingsdatensätzen. Den größten Wachstums in Datensätze konnten im Bereich **verwandte** autonome Fahrzeuge (Mapillary-Datensatz), Roboternavigation und Antenne (xView-Datensatz) beobachtet werden. Wir haben auch eine zunehmende Anzahl von Start-ups für die Computer-Vision bemerkt, die Remote Annotation Services anbieten. Es gab auch eine gewisse Zunahme der Aktivitäten in der Unternehmensentwicklung von synthetischen Datensätzen, aber dies betraf auch hauptsächlich Industrieroboter oder autonome Fahrzeuge.

Algoneer

Schlussbericht

Zuwendungsempfänger: Andreas Dewes & Katharine Jarmul GbR

Dr. Andreas Dewes

Berlin, den 30.09.2019

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen **01IS18S59** gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Algorithmische Systeme werden in vielen Bereichen eingesetzt, um Entscheidungen und Abläufe durch Software zu automatisieren. Oft werden hierfür Verfahren des maschinellen Lernens (ML) bzw. der künstlichen Intelligenz (KI) eingesetzt. Im Gegensatz zu herkömmlichen Algorithmen werden diese Verfahren dabei nicht manuell programmiert, sondern anhand von Daten auf ein gegebenes Problem „trainiert“. Moderne ML-Verfahren sind hochkomplex und selbst für die Programmierer der Systeme nicht einfach nachzuvollziehen. Dies schafft eine Reihe von Problemen beim alltäglichen Einsatz dieser Systeme: Mangelnde Nachvollziehbarkeit untergräbt Vertrauen der Benutzer und kann zu ernststen Problemen führen, beispielsweise wenn algorithmische Systeme falsche Entscheidungen treffen die Menschen persönlich einschränken, sich in kritischen Situationen falsch entscheiden, Personen aufgrund befangener Trainingsdaten diskriminieren oder die Systeme von außen manipulierbar sind. Bisher gibt es nur wenige Ansätze um ML-Systeme systematisch auf Robustheit, Sicherheit, Transparenz und Nachvollziehbarkeit zu testen. Da solche Systeme in immer mehr Bereichen eingesetzt werden stellt dies ein ernstes Problem für die Gesellschaft dar. Mit Algoneer wollen wir eine Lösung schaffen, die Entwicklern aber auch Betroffenen und Auditoren von ML-Systemen ermöglicht, diese besser zu verstehen und gemeinwohlorientiert zu gestalten. Unser Ansatz ist, eine Open-Source Software zu schaffen, die einfach in bestehende ML-Systeme integriert werden kann und Entwicklern hilft, diese kontinuierlich zu analysieren und zu überwachen. Die Software soll ermöglichen, aktuelle Forschungsergebnisse und Methoden aus der akademischen Forschung für die Untersuchung von ML-Verfahren einfach in der Praxis zu nutzen. Gleichzeitig sollen die Ergebnisse der

Testverfahren in einfacher und ansprechender Weise aufbereitet werden, so dass sie auch von technisch wenig erfahrenen Nutzern interpretiert werden können. Die Lösung soll sprach- und systemunabhängig funktionieren und über eine Web-Anwendung gesteuert werden können. Wir haben uns zu Beginn des Projekts folgende Meilensteine gesetzt:

- **Umsetzung des grundlegenden Backends mit Python und Docker (Workflow-System und API)**
- **Umsetzung des grundlegenden Frontends mit React.js**
- **Umsetzung eines ersten Beispielpugins (z.B. LIME)**
- **Regelmäßige Diskussion mit Stakeholdern (Data Scientists + Stiftungen / Organisationen)**
- **Whitepaper zu Methodik und Umsetzung**
- **Automatisiertes Deployment für das Projekt (Docker-Installer / Ansible-Setup)**
- **Dokumentation der Systemkomponenten**
- **Webseite für das Projekt**
- **Vorstellung auf relevanten Konferenzen**

Während des Projekts haben wir einzelne Meilensteine basierend auf gewonnenen Erkenntnissen angepasst, so haben wir das Backend z.B. nicht als Docker-basiertes System sondern als reine Python-Anwendung umgesetzt, wobei wir eine API sowie eine Python-Bibliothek zur einfachen Anbindung an unterschiedliche ML-Systeme implementiert haben.

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

Wir haben allgemein drei Zielgruppen für unsere Lösung definiert:

- **Entwickler von ML-Systemen**
- **Menschen, die mit ML-Systemen arbeiten oder von diesen betroffen sind**
- **Auditoren die ML-Systeme begutachten und bewerten sollen**

Im Rahmen des Prototype Fund (PF) Projektes haben wir die Gruppe der Entwickler priorisiert, da wir denken, dass diese Zielgruppe am stärksten von unserer Lösung profitieren kann. Zusätzlich haben wir uns bei den Testmethoden für ML-Verfahren auf Methoden für die Nachvollziehbarkeit dieser Verfahren beschränkt, um mit dem limitierten zeitlichen Umfang des Projektes ein relevantes Ergebnis erzielen zu können. Unsere Lösung bezieht sich thematisch daher sehr genau auf das Themenfeld des Software Sprints „Maschinen Lernen Lassen“ und wir sehen die erzielten Ergebnisse als sehr guten Beitrag, da die erstellte Software aktuell eine einfache und leicht einzusetzende Möglichkeit bietet, ML-Verfahren systematisch und kontinuierlich zu testen und zu überwachen. Unsere Software macht damit aktuelle Forschungsergebnisse in diesem Bereich einer breiteren Nutzerschicht verfügbar. Insbesondere im Rahmen der KI-Initiative der Bundesregierung und der europäischen Union

sehen wir Algoneer als eine wichtige Komponente zur Umsetzung von Nachvollziehbarkeit und Transparenz bei dem Einsatz algorithmischer Systeme.

Ausführliche Darstellung der Ergebnisse

Wir haben die uns gesteckten Ziele vollständig erreicht und zum Projektende im August einen funktionsfähigen Prototyp präsentiert, der alle wesentlichen geplanten Funktionen enthält und voll lauffähig ist. Im Rahmen der geplanten Meilensteine haben wir im Detail die folgenden Ergebnisse erzielt.

Umsetzung des grundlegenden Backends mit Python und Docker

Wir haben zwei grundlegende Backend-Komponenten implementiert: Eine Python-Bibliothek (algoneer), welche einfach in bestehende Python-basierte ML-Systeme integriert werden kann und in der Lage ist, mit unterschiedlichen Datenformaten und ML-Bibliotheken zu arbeiten. Benutzer können über die Bibliothek automatisiert verschiedene Tests auf ML-Verfahren anwenden (u.a. SHAP, PDP, ALE) und die Ergebnisse systematisch speichern und zu Dokumentationszwecken mit spezifischen Modellen, Datensätzen und Algorithmen assoziieren. Die Testergebnisse können direkt über die Bibliothek begutachtet werden oder über eine REST-Schnittstelle an die Algonaut-API gesendet werden, was eine Analyse über die Web-Anwendung ermöglicht. Wir haben neben drei implementierten Tests beispielhaft Schnittstellen für populäre Python-Bibliotheken zur Datenverwaltung (pandas) sowie für das maschinelle Lernen (scikit-learn) erstellt, die erläutern wie Algoneer an unterschiedliche ML-Technologien angepasst werden kann. Alle Schnittstellen wurden dabei so generisch wie möglich definiert, um eine hohe Anpassbarkeit zu gewährleisten, wir hoffen damit eine Anbindung an weitere ML-Bibliotheken einfach zu machen. Die Python-Bibliothek wurde mit einer modernen Python-Version erstellt (3.7) und setzt konsequent auf eine hohe Code-Qualität. Zur Sicherung der Qualität wurden für alle wesentliche Systemkomponenten ausführliche Unit-Tests definiert, zudem wurde die komplette Code-Basis mit Typinformationen annotiert.

Zusätzlich zur Python-Bibliothek haben wir eine Python Web-API erstellt, Algonaut, die eine REST-Schnittstelle zur Speicherung der Testergebnisse der Algoneer-Bibliothek bietet. Die API kann unabhängig von Algoneer betrieben werden und bietet eine Möglichkeit, Testergebnisse von Algoneer aber auch aus anderen Quellen systematisch zu erfassen und zu speichern. Die REST-API wurde genau wie die Python-Bibliothek mit modernen Methoden der Software-Entwicklung erstellt, verfügt über umfangreiche Unit-Tests und ist zudem über eine OpenAPI-Spezifikation vollständig dokumentiert. Die Dokumentation per OpenAPI-Standard ermöglicht zudem, in einer Vielzahl an Programmiersprachen automatisiert Schnittstellen zu der API zu definieren und so einfach Daten an diese zu senden.

Umsetzung eines ersten Beispielplugins (z.B. LIME)

Wir haben im Rahmen der Evaluation von Testmethoden für ML-Verfahren alle gängigen Methoden und Bibliotheken evaluiert, u.a. SHAP¹, LIME² sowie eine Reihe weiterer Verfahren^{3,4}. Wir haben exemplarisch drei dieser Verfahren in unserer Algoneer-Bibliothek implementiert. Die Einbettung in Algoneer macht die Verfahren dabei einfach nutzbar da eine Konfiguration der einzelnen Tests durch den Nutzer entfällt. Vielmehr muss der Nutzer lediglich einmalig die Datenstruktur der Testdaten definieren und kann dann alle aufgeführten Tests automatisiert ausführen. Dies spart gerade bei der Entwicklung von ML-Verfahren wertvolle Entwicklerzeit und macht Testergebnisse robuster und systematisch nachvollziehbar.

Regelmäßige Diskussion mit Stakeholdern

Im Rahmen des Projektes haben wir mit 15 Organisationen detaillierte Interviews geführt, u.a. mit privaten Unternehmen, zivilgesellschaftlichen Organisationen, Stiftungen aber auch Privatpersonen und Gewerkschaften. Basierend auf diesen Interviews haben wir die Feature-Entwicklung priorisiert und angepasst und haben sehr viel wertvolles Feedback zur Gestaltung einer relevanten Lösung von unseren Zielgruppen erhalten.

Whitepaper zu Methodik und Umsetzung

Wir haben die Ergebnisse des Projekts im Rahmen eines ausführlichen Blog-Posts zusammengefasst und haben diesen auf unserer Webseite publiziert.

Automatisiertes Deployment für das Projekt

Wir haben basierend auf Nutzerfeedback ein komplexes Deployment unserer Lösung komplett vermieden. Die erstellte Software kann vielmehr in wenigen Schritten als Python-Bibliothek installiert und genutzt werden, ein komplexer Deployment-Prozess ist somit nicht erforderlich. Wir haben die Installation der einzelnen Systemkomponenten (Algoneer, Algonaut und Frontend) ausführlich dokumentiert.

Dokumentation der Systemkomponenten

Wir haben für die Algoneer Python-Bibliothek eine ausführliche Dokumentation mithilfe des „Sphinx“ Dokumentationssystems erstellt. Für die Algonaut API haben wir ebenfalls eine vollständige Dokumentation aller Endpunkte mithilfe der OpenAPI-Spezifikation erstellt. Das Frontend wurde im Code selbst dokumentiert, die Nutzung ist weitgehend selbsterklärend und erfordert keine ausführliche Dokumentation.

¹ <https://arxiv.org/abs/1705.07874>

² <https://arxiv.org/abs/1602.04938>

³ <https://christophm.github.io/interpretable-ml-book/ale.html>

⁴ <https://christophm.github.io/interpretable-ml-book/pdp.html>

Webseite für das Projekt

Wir haben eine statische Website mithilfe einer Page-Building Tools erstellt und auf <https://algoneer.org> veröffentlicht. Die Inhalte der Webseite wurden mehrfach angepasst und erweitert basierend auf Feedback, das wir während der Projektphase erhalten haben.

Vorstellung auf relevanten Konferenzen

Wir haben Algoneer u.a. auf lokalen Veranstaltungen in Berlin (z.B. bei Gewerkschaften) als auch in virtuellen Konferenzen (z.B. im Rahmen von Bitkom-Events) vorgestellt. Wir haben im Rahmen einer Bitkom-Publikation (Erscheinungsdatum Oktober 2019) den Ansatz von Algoneer vorgestellt und exemplarische Tests erläutert, die für die Nachvollziehbarmachung von ML-Verfahren genutzt werden können.

Wir haben im Rahmen der Arbeit an dem Projekt unser Verständnis der Notwendigkeit einer Lösung für das Testen von ML-Verfahren stark konkretisiert und erste klare Marktbedürfnisse identifiziert. Wir planen, den im Rahmen des PF erstellten Prototypen kontinuierlich weiterzuentwickeln und gegebenenfalls im Rahmen einer professionellen Lösung anzubieten. Die Förderung hat uns dabei geholfen, die nötige Marktrecherche und Interviews sowie die Entwicklung des Prototyps zu finanzieren. Die Unterstützung durch das Prototype Fund Team war insbesondere zu Anfang des Projektes hilfreich. Ohne die Förderung des PF und BMBF hätten wir das Projekt nicht realisieren können.

Umsetzung einer Web-Anwendung

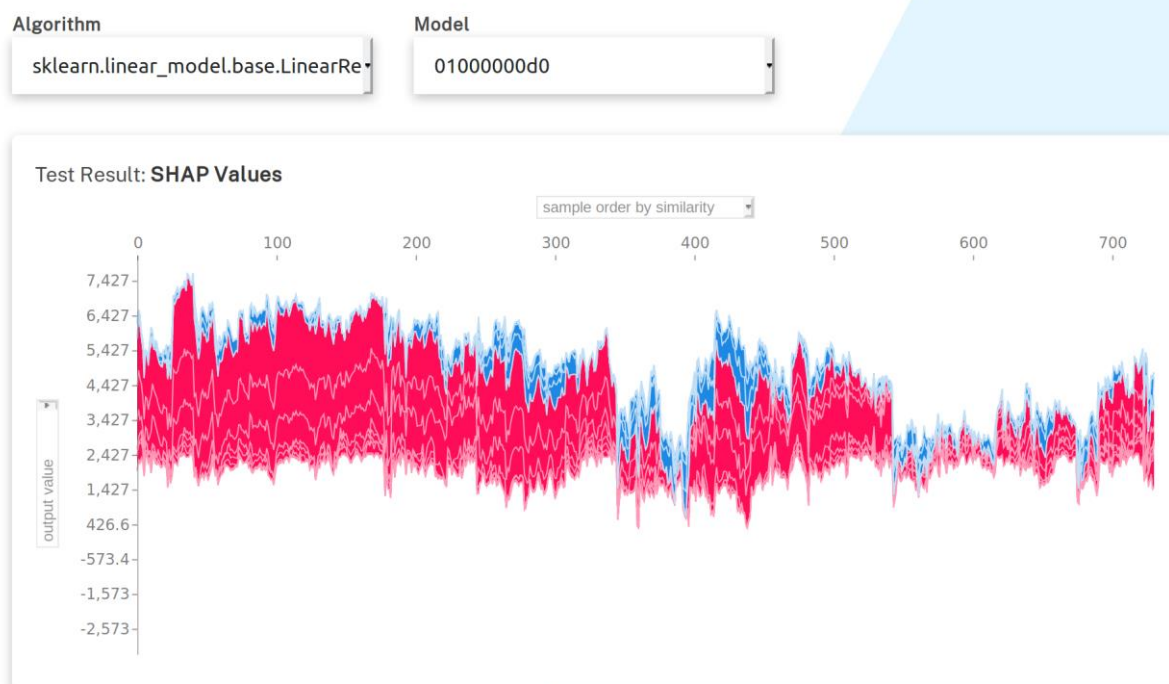


Abb. 1: Algoneer Web-Anwendung. Beispiel-Ergebnis eines SHAP-Tests für ein lineares ML-Modell.

Um die mit Algoneer erstellten Testergebnisse möglichst einfach nutzbar und teilbar zu machen, haben wir eine Web-Anwendung erstellt. Diese Anwendung nutzt die REST-API von Algonaut, um Testergebnisse von einem Server abzurufen und darzustellen. Wir haben im Rahmen des Projektes exemplarisch Testergebnisse eines einzelnen Verfahrens (SHAP) in der Web-Anwendung implementiert, was eine einfache Auswertung dieser Ergebnisse für den Nutzer ermöglicht. Die Web-Anwendung ist Multi-User fähig und kann mit verschiedenen Authentifizierungs-Lösungen betrieben werden, wir haben exemplarisch Logins via Github, Gitlab, Google sowie E-Mail und Passwort angebunden. Nutzer können in der Software Organisationen anlegen und andere Nutzer zu ihren Organisationen einladen, Testergebnisse können so leicht in Teams geteilt und nachvollzogen werden. Abb. 1 zeigt einen Screenshot der Anwendung, welcher das Ergebnis des SHAP-Tests für ein Beispiel ML-Verfahren zeigt.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Wir haben mit Algoneer gezeigt, dass automatisiertes Testen von ML-Verfahren möglich und sinnvoll ist, und den Aufwand hierfür für Anwender drastisch reduziert. Wir erlauben damit Entwicklern von ML-Verfahren, ihre Systeme kontinuierlich und systematisch zu testen und die Testergebnisse mit allen relevanten Stakeholdern zu teilen. Wir planen, die Lösung sowohl durch die Anbindung weiterer Testmethoden als auch durch die Unterstützung zusätzlicher ML-Bibliotheken zu erweitern. Zusätzlich planen wir den Betrieb einer SaaS-Lösung und möglicherweise die Schaffung einer professionellen Lösung basierend auf unserem Open-Source Tool. Wir planen aktuell ca. 2 Personentage pro Woche für die Weiterentwicklung von Algoneer zu nutzen und weiterhin kontinuierlich Feedback für die Lösung einzusammeln. Wir sind zudem in Gesprächen mit verschiedenen Organisationen und Unternehmen, die Interesse gezeigt haben, Algoneer in der Praxis zu nutzen, um ML-Verfahren zu testen und zu überwachen. Die Tatsache, dass unsere Software als Open-Source Lösung frei verfügbar ist und flexibel genutzt werden kann macht die konkrete Nutzung sehr einfach. Dies ermöglicht z.B. Unternehmen, Organisationen und Privatpersonen, die Software ohne hohe Risiken einzusetzen und einfach an ihre Bedürfnisse anzupassen.

Die Arbeit an dem Projekt hat uns persönlich in unserer fachlichen Entwicklung stark vorangebracht. Wir haben heute ein sehr viel besseres Verständnis über die Gestaltung und die Überwachung von ML-Verfahren und haben zudem unsere Kenntnisse in moderner Software-Entwicklung und nutzergetriebener Produktgestaltung ausbauen können. Wir sind auf gutem Weg, Experten im Bereich sicherer, nachvollziehbarer KI zu werden und sehen basierend hierauf viele Möglichkeiten, unsere im Rahmen des Projekts gewonnenen Kenntnisse in der Praxis weiter einzusetzen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben ursprünglich geplant, das Analyse-Backend mit „Docker“ zu realisieren, dies haben wir zu Gunsten einer einfacheren Lösung verworfen und haben stattdessen eine reine Python-Bibliothek implementiert. Dies hat sich rückblickend als der richtige Ansatz herausgestellt und hat uns zudem erlaubt, schnell erste Ergebnisse zu erzielen und diese für Feedback von Nutzern einzusetzen. Die Gestaltung des Frontends haben wir im Laufe des Projekts ebenfalls stark vereinfacht, was uns erlaubt hat schneller eine funktionsfähige Version zu veröffentlichen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wir haben verschiedene Materialien erstellt, die potenziellen Nutzern erlauben mehr über Algoneer zu erfahren:

- Unsere Website <https://algoneer.org> (Abb. 2) enthält wesentliche Informationen über unser Projekt und die Zielgruppen sowie Einsatzmöglichkeiten, sie dient damit als erste Anlaufstelle für potenzielle Nutzer.
- Unsere Github-Seite <https://github.com/algoneer> enthält alle Software-Komponenten, die während des Sprints erstellt wurden und erlaubt Nutzern, einen schnellen Überblick über die Installation und Nutzung dieser Komponenten zu erhalten. Zusätzlich enthalten einzelne Repositories Beispiele (z.B. <https://github.com/algoneer/algoneer/tree/master/examples>) zu Testmethoden, die von Algoneer unterstützt werden. Aktuell befinden sich auf Github die Projekte „algoneer“ (Python-Bibliothek), „algonaut“ (Python-basierte REST API), „algoneer-app“ (Algoneer Web-Anwendung / Frontend), „datasets“ (Beispieldatensätze die zum Testen von Algoneer eingesetzt werden), „website“ (Algoneer-Webseite) und „algonaut-plugins“ (Plugins für die Algonaut REST-API).
- Unsere Gitlab-Seite <https://gitlab.com/algoneer> enthält die gleichen Komponenten wie unsere Github-Seite, zusätzlich werden hier jedoch automatisierte Tests bei jeder Code-Änderung ausgeführt. Diese Seite erlaubt die Zusammenarbeit mit Entwicklern, die uns bei der Weiterentwicklung von Algoneer unterstützen wollen.
- Unsere Dokumentation <https://docs.algoneer.org> enthält detaillierte Angaben zu den einzelnen Systemkomponenten und zeigt exemplarisch deren Nutzung.
- Der Prototype Fund hat freundlicherweise zwei Videos produzieren lassen, die unser Algoneer Projekt vorstellen bzw. allgemein über die 5. Wettbewerbsrunde berichten: <https://www.youtube.com/watch?v=xLtrilzwUE> sowie <https://www.youtube.com/watch?v=MpFtAmz7UTQ>. Das erste Video ist ebenfalls auf unserer Webseite verfügbar.
- Im Rahmen einer Bitkom-Publikation zum Thema „nachvollziehbare KI-Methoden“ haben wir einen ausführlichen, mehrseitigen Abschnitt über Algoneer sowie

Testmethoden für nachvollziehbare ML-Verfahren im Allgemeinen beigetragen. Die Publikation wird voraussichtlich im Oktober 2019 erscheinen.

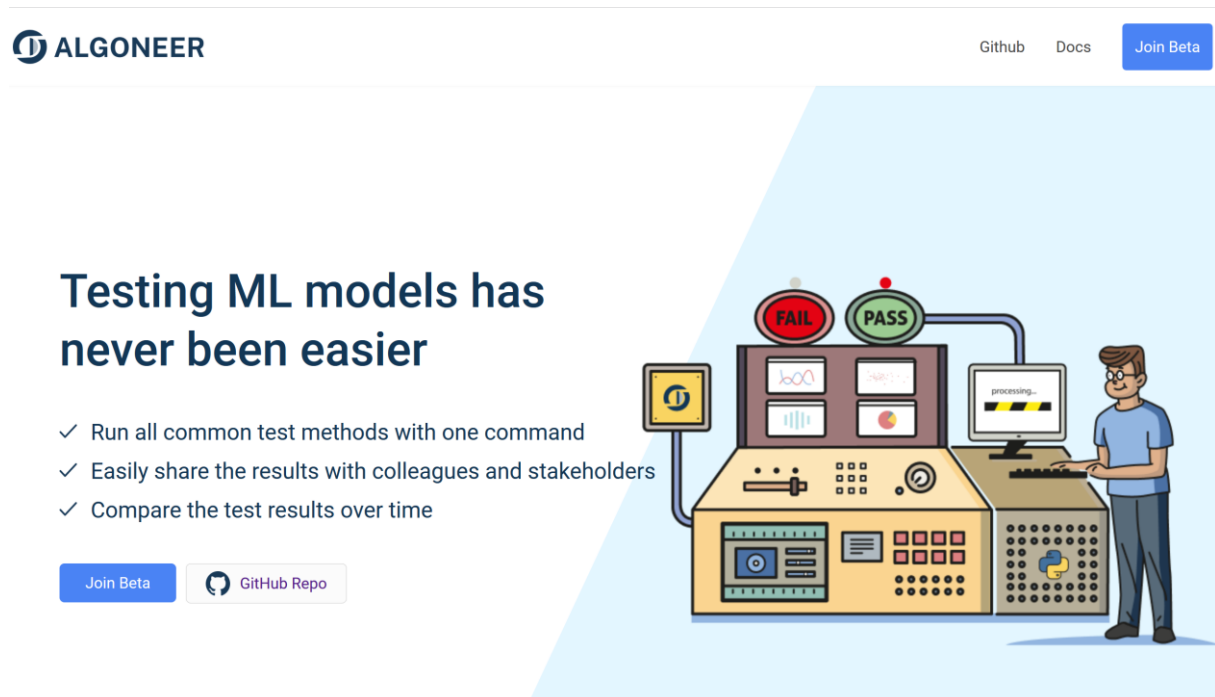


Abb. 2: Algoneer-Webseite (Stand September 2019)

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es hat sich herausgestellt, dass Projektmanagement, Interviews mit potenziellen Nutzern sowie Forschungs- und Recherchearbeiten einen höheren Aufwand verursacht haben als ursprünglich geplant. Zusätzlich mussten durch den Ausfall von Frau Jarmul, die sich leider nicht an dem Projekt aktiv beteiligen konnte, die Arbeiten alleinig von Andreas Dewes durchgeführt werden, was eine Umplanung erforderte, die jedoch machbar war.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Seit Einreichung der Bewerbung im Herbst 2018 gab es eine Reihe von Publikationen und Veröffentlichungen im Bereich nachvollziehbarer ML-Verfahren, u.a. haben IBM und andere große Stakeholder Bibliotheken veröffentlicht, um ML-Systeme beispielsweise auf Fairness zu untersuchen. Publikationen wie das „Interpretable ML Book“ von Christoph Molnar⁵ haben zudem die Recherche und Implementierung geeigneter Testverfahren für uns stark vereinfacht, was unser Projekt sogar beschleunigt hat. Insgesamt haben wir gesehen, dass das Interesse an Verfahren für das Testen von ML-Systemen im letzten Jahr sprunghaft angestiegen ist, was sicherlich mit der starken Thematisierung von künstlicher Intelligenz im Allgemeinen sowie den zahlreichen Bundesinitiativen in diesem Bereich zusammenhängt.

⁵ <https://christophm.github.io/interpretable-ml-book/pdp.html>

Insgesamt sehen wir diese Entwicklung als sehr positiv und denken, dass wir mit Algoneer genau zum richtigen Zeitpunkt ein Tool schaffen, das ein zunehmend relevantes Problem auf einfache Art löst. Bisher hat sich noch keine größere Entwickler- und Nutzer-Community um Algoneer gebildet, wir planen jedoch für Oktober den offiziellen Launch einer gehosteten Version der Software und damit einhergehend auch größere Aufmerksamkeit von seitens der Entwickler-Community sowie möglicher Nutzer. Die große Popularität des Themas KI/ML sollte uns hierbei eher helfen.

Wir sind sehr dankbar für die erhaltene Förderung und die Betreuung durch den Prototype Fund und hätten Algoneer in der aktuellen Form ohne diese Förderung niemals realisieren können.

Berlin, den 30.09.2019

Dr. Andreas Dewes

Richtlinie zum „Software-Sprint“

Mietenwatch

Schlussbericht

Zuwendungsempfänger:

Tilman Miraß

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S60 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 steigende Dramatik auf dem Berliner Wohnungsmarkt wird seit Jahren für immer mehr Menschen spürbar. Ob bei der Wohnungssuche, durch Mieterhöhungen oder die Verdrängung aus dem bisherigen Umfeld. Die emotionalen Debatten um diese Problematik werden zum Teil durch Zahlen gestützt. Da diese statistischen Auswertungen sich jedoch oftmals auf abgeschlossene Zeiträume und geographisch grob aggregierte Räume beziehen, ist es schwierig ein aktuelles Bild des Mietmarkts und seiner Dynamiken zu vermitteln.

Mit „Mietenwatch“ wollte ich diese Lücke schließen, denn Daten zur aktuellen Situation auf dem Mietmarkt sind in Form von Angeboten auf Online-Portalen verfügbar. Die einzelnen Informationen liefern jedoch noch kein Gesamtbild. Mit der Sammlung von Online-Inseraten und der Aufbereitung in Form einer thesengestützten Erzählung auf einer Webseite wollte ich die Situation greifbar und erlebbar machen.

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

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

Das Projekt richtet sich an eine breite Zielgruppe von Menschen, die sich für Wohnungspolitik interessieren, über diejenigen, denen die Lage auf dem Mietmarkt bislang selbst nicht erfahren haben bis hin zu Diskursteilnehmenden am aktuellen Mietendiskurs, für die die Analysen Argumentationsgegenstand sein können. Die Zugänglichmachung des Gesamtbilds aus einzelnen Inserationsinformationen ist eine civic tech-Lösung, die einen data literacy-Ansatz verfolgt: Daten werden für zivilgesellschaftliche Akteur*innen aufbereitet und in einen gesamtgesellschaftlichen Kontext eingebettet vermittelt. Dies fügt sich in die Ziele des Prototype Funds.

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?

Seit April 2018 wurden mittels eines Scrapers täglich neue Angebote der Plattform ImmobilienScout24¹ in Berlin erfasst. Dabei entstand ein Datensatz mit einem Umfang von über 80 000 Beobachtungen. Die Auswertung des Datensatzes erfolgte nach der Bereinigung um unplausible Werte mittels der Statistiksoftware R (<https://github.com/mietenwatch/mw-analytics>). Bei der ersten Veröffentlichung wurden Duplikate im Datensatz belassen, da es sich nicht abschließend und eindeutig bestimmen ließ, ob es sich bei den Duplikaten um Wiedervermietungen, Wohnungen mit denselben Parametern oder tatsächliche Duplikate ein und desselben Inserats handelt. In einem Update Mitte November 2019 wurde mittels eines komplexen Filters der Versuch unternommen, eben diese Unterscheidung vorzunehmen, um Duplikate aus der Betrachtung auszuschließen. Darüber hinaus wurden die Analysen um die Berücksichtigung der WBS-Pflicht für einzelne Angebote, die Filterung von Kurzzeitvermietungen erweitert und Korrekturen in der Auswertung der Daten vorgenommen.

Die Ergebnisse der Auswertungen sind vielfältig. In drei Kapiteln wird eine Erzählung zur Situation auf dem Berliner Mietmarkt entspannt. Zum einen können Nutzer*innen im Kapitel „Leistbarkeit“ erfahren, wie leistbarer Wohnraum in Berlin für unterschiedliche Einkommensklassen verteilt ist und welche Gebiete einem besonders hohen Verdrängungsdruck ausgesetzt sind. Im Kapitel „Wohnen als Ware“ werden die auf dem

¹ ImmobilienScout24 ist die marktbeherrschende Plattform für Online-Mietwohnungsinserate. Nach eigenen Angaben erreicht die Marktabdeckung 80 %. Bestimmte Marktsegmente werden über andere Wege vermietet (etwa das Luxus-Segment, Wohnungen von Genossenschaften oder preisgebundene Wohnungen aus dem Sozialwohnungsbau) daher beziehen sich die Angebotsanalysen nur auf das online frei einsehbare Segment..

Online-Wohnungsmarkt agierenden Anbieter betrachtet. Dabei geht es unter anderem um die Charakterisierung der Bestände der zehn größten Anbieter. Zusätzlich wird eine Berechnung einer „kostendeckenden Miete“² anhand der einzelnen Angebote vorgenommen. Das Kapitel betrachtet darüber hinaus, wie bisherige regulatorische Maßnahmen sich auf den Mietmarkt auswirken. Das dritte Kapitel „Antworten“ skizziert Lösungen der Dramatik auf dem Wohnungsmarkt. Dabei geht es vor allem darum, wie sich die Leistbarkeit der betrachteten Angebote im Falle eines umgesetzten Mietendeckels, wie er zum Zeitpunkt der Veröffentlichung des Projekts Anfang Oktober 2019 diskutiert wurde, verändern würde. Außerdem wird skizziert, wie sich die Leistbarkeit im Falle einer erfolgreichen Enteignung der großen Wohnungskonzerne, wie sie die Kampagne „DW und Co. enteignen“ fordert, auswirken würde.

Neben der Darstellung der Ergebnisse in einer Web-Erzählung ist auch eine API für die Ermittlung des Berliner Mietspiegels (<https://github.com/mietenwatch/mw-mietspiegel-api>) für jede Berliner Adresse entstanden. Die Ermittlung des Mietspiegels gründete bisher auf einem schwer zugänglichen Verfahren, das das Nachschlagen der Lage des jeweiligen Objekts in einem Straßenverzeichnis vonnöten machte. Dieses Hindernis zur Ermittlung der ortsüblichen Vergleichsmiete wird durch die Datenschnittstelle obsolet.

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?

Für die Zielgruppen können die Ergebnisse des Projekts argumentativ genutzt werden, um die Dramatik des Angebotsmarkts datengestützt zu belegen. Durch die Open Source-Stellung werden die Ergebnisse nachvollziehbar und gewinnen an Seriosität. Aufgrund der großen Arbeitsbelastung habe ich noch keine konkreten Pläne für eine Weiterentwicklung fassen können.

Die Arbeit an dem Projekt hat mir vielerlei Erfahrung im Selbstmanagement geliefert, die ich in weiteren Projekten gewinnbringend einsetzen kann.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

² Bei der „kostendeckenden Miete“ handelt es sich um ein theoretisches Konstrukt, das schätzt, wie hoch eine profitfreie Miete (über alle (Vor-)Eigentümer*innen hinweg) wäre, um sämtliche Bau- und Instandhaltungskosten zu decken.

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

Aufgrund der Zielsetzung, Techniken des Machine Learnings in die Analysen aufzunehmen, habe ich versucht, ML sinnvoll einzusetzen. Dabei stellte sich heraus, dass die Verwendung klassischer statistischer Methoden mehr von Nutzen sind als die Anwendung von ML. Dennoch habe ich mittels eines Random Forests geschätzt, wie groß der Einfluss einzelner Parameter auf die Nettokaltmiete ist.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Die Präsentation des Projekts erfolgt über die Projekte www.mietenwatch.de, die Anfang Oktober 2019 veröffentlicht wurde.

Zusätzlich ist der in der Projektzeit entwickelte Code auf <https://www.github.com/mietenwatch/mietenwatch> veröffentlicht. Damit werden die Berechnungen und Darstellungen nachvollziehbar. Aus Urheberrechts- und Datenschutzgründen enthalten die veröffentlichten Repositories nicht den zugrundeliegenden Datensatz.

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 gesteigerte Aufwand, der sich insbesondere durch den großen inhaltlichen Anteils des Projekts und durch die hohe Dynamik des Diskurses ergeben hat, hat dazu geführt, dass die im Projektplan veranschlagten Stunden überschritten wurden. Auch die Veröffentlichung des Projekts nach Abschluss des Projektzeitraums führt dazu, dass zusätzliche Arbeit notwendig wird.

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?

Die Dynamik auf dem Berliner Mietmarkt hat die gesamte Projektzeit geprägt. Mit der Diskussion um den Mietendeckel seit Juni 2019 mussten die inhaltlichen Linien des Projekts grundlegend angepasst werden. Da die Diskussion zum jetzigen Zeitpunkt nicht abgeschlossen ist, werden bis zur Veröffentlichung eventuell weitere Anpassungen notwendig werden.

Richtlinie zum „Software-Sprint“

Manipulation - Manipulation: Erfahrbarmachung der Manipulierbarkeit von Standardmethoden der KI

Schlussbericht

Zuwendungsempfänger:

Katharina Rasch

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S61 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Aktuell findet der Diskurs zu Manipulation und anderen Schwachstellen künstlicher Intelligenz hauptsächlich zwischen Akademikern statt und ist sehr technisch/mathematisch. Ich will mit meinem Projekt dazu beitragen, dass sich auch die Zivilgesellschaft gut informiert mit dem Thema künstliche Intelligenz beschäftigen kann.

Ziel meines Projektes war es also, künstliche Intelligenz ein wenig zu entzaubern. Dies wollte/will ich erreichen, indem ich der Zivilgesellschaft die Möglichkeit gebe, selbst auszuprobieren, wie leicht sich Standardmethoden der Bilderkennung (wie sie beispielsweise von facebook und Co eingesetzt werden) manipulieren lassen.

Die geplante Vorgehensweise war:

1. Recherche: welche Bilderkennungsverfahren werden aktuell eingesetzt
2. Aktualisieren meines Wissens zur state of the art in Bildmanipulation / adversarial images
3. Erstellung eines Konzeptes für ein Tutorial, welches Manipulation von Bilderkennungsalgorithmen für die Nutzer praktisch erfahrbar macht
4. Umsetzung des Tutorialkonzeptes

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

Meine Zielgruppe sind Menschen, die die Debatte um KI verfolgen und die zwar etwas Technikinteresse haben, sich aber selbst wenig mit Technik auskennen (Voraussetzung: wissen wie man mit einem Smartphone ein Foto macht).

Ziel meines Projektes ist es, diesen Menschen die Möglichkeit zu geben, interaktiv moderne Bilderkennungssysteme auszuprobieren und ihre Grenzen auszutesten. Zusammen mit allgemein verständlichen Erklärungen zu Bilderkennungssystemen (wie funktioniert das / was funktioniert / was funktioniert noch nicht?) soll der Zielgruppe so ermöglicht werden, selbst besser einschätzen zu können, welche Möglichkeiten und Gefahren der KI echt sind, und welche nur Hype.

Ein Zitat aus der Ausschreibung für Runde 5: “Emergente Technologien können perspektivisch dazu beitragen, gesellschaftliche Themen anders und besser als bisher zu erschließen und zu bearbeiten. Um das zu erreichen, müssen wir aber dafür sorgen, dass sie auch von möglichst vielen Menschen verstanden, interpretiert und mitgestaltet werden können.“ Und genau beim Punkt “möglichst viele Menschen sollten die Technologie verstehen” setzt mein Projekt an.

Ausführliche Darstellung der Ergebnisse

Während des Förderzeitraums sind zwei Prototypen entstanden:

1. Online-Spiel “Schlag die KI”

In diesem Spiel tritt ein Mensch gegen eine KI an. Aufgabe ist Bilderkennung: wer erkennt schneller, welches Objekt sich auf dem gezeigten Bild befindet. Das Spiel läuft als Web-Applikation im Browser; unterstützt werden sowohl Computer als auch Mobiltelefone. Mit Hilfe von tensorflow.js läuft direkt im Browser ein Bildklassifizierungsmodell, welches die gleichen Aufgaben bekommt wie die menschlichen Spieler*innen.

Den Kontrahent*innen werden nacheinander drei Bilder von zufällig gewählten Alltagsgegenständen gezeigt:

- Das erste Bild hat keinerlei Manipulationen. In den meisten Fällen wird die KI hier schneller antworten als der Mensch.
- Das zweite Bild ist zu Beginn stark verrauscht, dieses Rauschen wird dann in kleinen Schritten verringert und der Gegenstand langsam deutlicher zu sehen. Bei diesem Bild ist es sowohl für Mensch als auch KI durch das starke Rauschen zu Beginn unmöglich zu erkennen, was auf dem Bild dargestellt ist. Mit reduziertem Rauschen wird die Aufgabe für beide langsam einfacher. Es zeigt sich, dass in vielen Fällen der Mensch schneller ist -- das heisst mit stärkerem Rauschen umgehen kann als die KI.
- Das dritte Bild hat ist mit einem speziellen "adversarial" Muster manipuliert. Für den Menschen stellt sich dieses als zufällig aussehendes Muster von "kaputten" Pixeln (bspw falsche Farbe) dar; der gezeigte Gegenstand ist allerdings für Menschen trotzdem offensichtlich. Die KI hingegen schafft es hingegen nicht zu erkennen was zu sehen ist, selbst wenn die Stärke dieses Musters reduziert wird.

Ziel dieses Spieles ist es, zu vermitteln, dass automatische Bilderkennung zwar mit Menschen mithalten kann -- aber nur unter den richtigen Bedingungen. Während die KI nur statistische Muster aus den Trainingsdaten gelernt hat, arbeiten Menschen auf einem weitaus höheren Abstraktionsniveau. Das macht das menschliche Sehen und Erkennen viel robuster (aber natürlich nicht unfehlbar) gegen Manipulationen.

Der Prototyp dieses Spiels ist unter <https://was-kann-ki.gitlab.io/schlag-die-ki/> erreichbar. Bitte beachten: Die Implementation ist zwar funktionsfähig, aber für die Öffentlichkeit noch nicht wirklich zugänglich, insbesondere fehlen noch einige Texte und ein Fazit. Ich bitte darum, diesen Link noch nicht außerhalb dieses Berichtes weiterzugeben.

2. Livedemo/Installation: Gesichtserkennung austricksen

In dieser Installation sollen Fähigkeiten und Grenzen von Gesichtserkennung für Menschen erlebbar gemacht werden. Angedachter Einsatzbereich für die Installation sind Messen / Konferenzen.

- An einem Monitor ist eine Kamera und in kleiner Computer angebracht. Wenn eine Besucher*in das erste mal vor die Kamera tritt wird ihr Gesicht erfasst und gespeichert
- Ab jetzt kann die Besucher*in erkannt werden. Immer wenn sie vor die Kamera tritt, wird angezeigt dass sie erkannt wurde / bzw nicht erkannt wurde.
- Jetzt kann die Besucher*in versuchen, die Gesichtserkennung auszutricksen, beispielsweise durch: Drehen des Kopfes, Verdecken von Gesichtsteilen mit den Händen, mit Hilfe von bereitgestellten Gegenständen (bspw Clownsnase, Perücke, Mütze, schwarze Balken aus Plastik, evtl Laserpointer, etc).

- Ein Poster präsentiert den aktuellen Stand der Wissenschaft zum Thema Gesichtserkennung.
- Alle gespeicherten Gesichter werden direkt nach Ende der Ausstellung gelöscht (bei Bedarf auch automatisch jede Stunde oder so). Alles wird lokal verarbeitet, keinerlei Daten gelangen ins Internet.

Ich habe während des Projektes diese Installationsidee technisch umgesetzt und als Vorschlag für die <https://ki-convention.com/> eingereicht. Dort könnte die Idee erstmals im größeren Umfang getestet werden -- bei Erfolg werde ich die Installation dann (mit eventuell nötigen Anpassungen) auch bei anderen Konferenzen / Ausstellungen einreichen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Während des Projektes habe ich mit Mitgliedern der Zielgruppe zusammengearbeitet, um meine Ideen und Umsetzungen zu verifizieren (mit Hilfe von Papierprototypen und elektronischen Prototypen). Aufgrund positiven Feedbacks bin ich guter Hoffnung, dass meine Prototypen tatsächlich helfen können, Menschen die Grenzen von Bilderkennungssystem aufzuzeigen.

Insbesondere die Installation zur Gesichtserkennung ließ sich allerdings nur bedingt testen, da sich die realen Bedingungen (viele hunderte Menschen, ständig treten neue Menschen vor den Monitor, viel Bewegung im Hintergrund) nur schwer nachstellen lassen. Aus diesem Grund habe ich die Installation zunächst bei einer kleineren Konferenz angemeldet. Ich gehe davon aus, dass sich bei der Konferenz noch Anpassungsbedarf zeigen wird.

Für die Weiterentwicklung / Fertigstellung meiner Prototypen habe ich bereits Zeit eingeplant. Insbesondere für die Installation zur Gesichtserkennung hoffe ich, dass es durch die Open-Source-Stellung leichter sein wird, Kopien dieser Installation in verschiedenen Ausstellungen zu zeigen.

Persönlich habe ich während des Projektes eine Menge teilenswertes Wissen und Material zu den Themen "Grenzen der KI" und "Sicherheitsaspekte von KI-Methoden" aufgebaut. Bei der [Data Natives Konferenz](#) im November 2019 in Berlin werde ich einen Vortrag mit dem Titel "Attacks against Computer Vision applications" geben. Ein Blogeintrag zum Thema "Reverse engineering of black-box Computer Vision models" ist in Arbeit (siehe auch letzter Link unter "Kurze Angabe von Präsentationsmöglichkeiten für nötige Nutzer"). Weitere Vorträge und Blogeinträge sind geplant.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Im Projektantrag erwähnte ich unter anderem, dass es mit Hilfe von adversarial images möglich sein könnte, manipulative Bildfilter zu erstellen, um Gesichts/Bilderkennungssysteme

hereinzulegen. Im Rahmen von Milestone 2 "Aktualisieren meines Wissens zur state of the art in Bildmanipulation / adversarial images" habe ich eine Literaturrecherche zum Thema adversarial images vorgenommen. Dabei hat sich herausgestellt, dass die meisten aktuelle Methoden zur Generierung von adversarial images einen direkten Zugriff auf das Bilderkennungsmodell voraussetzen (sogenannte white-box attacks).

Üblicherweise sind tatsächlich im Einsatz befindliche Modelle allerdings nicht direkt zugreifbar, sondern beispielsweise durch eine API (z.B. Amazon Face Rekognition API) oder durch ein grafisches Benutzerinterface geschützt (z.B. Facebooks Gesichtserkennung). Sogenannte black-box Attacks, die auch in einem solchen Szenario funktionieren, sind leider noch weitaus weniger erforscht und aktuell noch äußerst ressourcenintensiv (>100.000 API calls für die Generierung eines einzigen manipulierten Bildes). Aufgrund dieser Erkenntnis erübrigte sich die Idee "manipulative Bildfilter" somit recht schnell.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Sämtlicher sourcecode ist auf gitlab verfügbar. Insbesondere:

Webapp: <https://gitlab.com/was-kann-ki/schlag-die-ki>

Live Gesichtserkennung auf NVIDIA Jetson: <https://gitlab.com/was-kann-ki/face-recognition-live>

Benchmark zur erreichbaren framerate von Modellen für Gesichtserkennung (Vorarbeit für Live Gesichtserkennung) <https://gitlab.com/was-kann-ki/face-detection-benchmark>

Experimente zum Thema "Was passiert eigentlich wenn ich generierte Gesichter (also nicht existierende Menschen) an eine Gesichtserkennungs-API schicke":
<https://observablehq.com/d/5a0732cf5267ad90>

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Planung wurde eingehalten.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Ich habe mich in meiner Arbeit auf diverse wissenschaftliche Artikel zum Thema Manipulierbarkeit von Gesichtserkennung gestützt. Von Projekten wie CV Dazzle (<https://ahprojects.com/cvdazzle/>) von Adam Harvey wurde ich inspiriert. Vielen Dank an Ellen König für die Hilfe bei der Ausarbeitung der Wirkungslogik. Vielen Dank an meine Freunde für viele interessante Diskussionen und Ideen. Die Live Gesichtserkennung ist eine Weiterentwicklung von Adam Geitgeys Demoprojekt (Verbesserung von Stabilität und Framerate, mehr Einstellungsmöglichkeiten). (<https://medium.com/@ageitgey/build-a-hardware-based-face-recognition-system-for-150-with-the-nvidia-jetson-nano-and-python-a25cb8c891fd>).

Richtlinie zum „Software-Sprint“

ml-ransomware – Ransomware war gestern: Undo von Ransomware mittels Machine Learning

Schlussbericht

Zuwendungsempfänger:

Fratz und Held GbR

Gesellschafter:

Matthias Held, Waldhornstraße 2, 77933 Lahr

Matthias Fratz, Haidelmoosweg 27, 78467 Konstanz

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S62 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Welches Problem wolltest Du mit Deinem Projekt lösen?

Der Verlust von privaten Bildern, Videos oder wichtigen Dokumenten ist für viele Menschen ein Schreckgespenst. Kriminelle haben es deshalb auf genau solche Dateien abgesehen, nehmen sie durch Verschlüsselung in virtuelle Geiselschaft und fordern Lösegeld, mit dem ungewissen Versprechen, die Daten wieder zu entschlüsseln. Was für Privatpersonen den Verlust wertvoller Erinnerungen bedeutet, kann für Unternehmen existenzbedrohend sein: Britische Krankenhäuser waren 2017 mehrere Tage kaum funktionsfähig, der internationale Güterverkehr war wochenlang behindert.

Was war Deine Motivation?

Im Rahmen der Masterarbeit von M. Held hatten wir grundlegende Eigenschaften und Erkennungsmerkmale von Kryptotrojanern (engl. Ransomware) analysiert. Die Haupte Erkenntnis war,

dass bei Nutzung eines modernen Cloudspeichers, wie der persönlichen Open-Source-Cloudlösung Nextcloud, die betroffenen Dateien leicht aus dem Cloudspeicher wiederhergestellt werden können.

Die Zuverlässigkeit der prototypischen Implementierung als App für Nextcloud war noch nicht optimal. Deshalb sollten im Rahmen des PrototypeFund-Projekts nun Machine-Learning-Techniken zur besseren Klassifikation zum Einsatz kommen. Dabei sollte das gesamte System so angelegt werden, dass auch weitere Cloudspeicher leicht von diesem Schutz profitieren können.

Wie war die geplante Vorgehensweise zur Problemlösung?

Moderne Cloudspeicher wie Nextcloud speichern ältere Versionen aller Dateien. Anhand des Schreibverhaltens und der geschriebenen Daten lässt sich ein Kryptotrojaner-Angriff erkennen, so dass für jede Datei die letzte unbeschädigte Version wiederhergestellt werden kann. Die im Rahmen der genannten Masterarbeit entwickelte Nextcloud-App „Ransomware Recovery“ verwendet handgeschriebene Regeln zur Erkennung dieser letzten unbeschädigten Version. Erste Tests hatten gezeigt, dass Machine-Learning-Techniken eine erhebliche Verbesserung der Erkennungsrate bringen sollten. Daher sollte ein Machine-Learning-Ansatz wissenschaftlich evaluiert und in die Ransomware Recovery-App integriert werden.

Dazu wollten wir das Zugriffsverhalten und die geschriebenen Daten echter Kryptotrojaner erfassen und frei zur Verfügung stellen. Anhand dieser Daten wollten wir ein geeignetes Klassifizierungsmodell bestimmen, welches zur Erkennung von Kryptotrojaner-Angriffen trainiert werden kann.

Um diese Technik sinnvoll einsetzbar zu machen, sollte direkt nutzbares vortrainiertes Erkennungsmodell zur Verfügung gestellt, sowie eine REST-basierte Komponente für die Einbindung in andere Cloudspeicher-Software entwickelt werden. Diese unabhängige Komponente kann dann aus verschiedenen schlanken Apps und Plugins für Cloudspeicher angesprochen werden und damit Schutz gegen Kryptotrojaner bieten.

Milestones:

- **M1 Datensatz:** Erweiterung und Systematisierung unseres Datensatzes der Zugriffsmuster und zerstörten Dateien von Kryptotrojanern, so dass dieser für Machine Learning geeignet ist.
- **M2 Modellauswahl:** Auswahl und Evaluierung eines geeigneten Modells für die Erkennung von Kryptotrojanern.
- **M3 fertiges Modell:** Training des gewählten Machine-Learning-Modells auf Basis dieses Datensatzes, um ein auslieferbares Modell zu erhalten.
- **M4 Schnittstellen:** Entwicklung von Schnittstellen für den Zugriff auf das Machine-Learning-Modell zur Integration in weitere Cloudspeichersysteme.
- **M5 Integration:** Integration in Cloudspeichersysteme, insbesondere Nextcloud.

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

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

Die Allgemeinheit profitiert von unserer Lösung durch Zugriff auf Cloudspeichersysteme, in die an eine Instanz unseres Klassifizierungs-Dienstes angebunden sind. Im Falle von Nextcloud sind dies grundsätzlich jegliche Nextcloud-Instanzen, unabhängig davon ob diese von Firmen, Organisationen oder Privatpersonen betrieben werden, da sich die App mit wenigen Klicks in der Administrationsoberfläche der Nextcloud-Instanz aktivieren lässt.

Durch das Angebot unseres Klassifizierungswerkzeugs als Open-Source-Software können technisch versierte Personen dies in andere Cloudspeichersysteme integrieren oder darauf aufbauend eigene Systeme entwickeln. Zielgruppe sind hier zunächst vor allem Open-Source-Projekte, aber auch Anbieter proprietärer Systeme können die entwickelte Software nachnutzen. Eine entsprechende Sichtbarkeit wird durch ein öffentliches Projekt auf GitHub unter einer geeigneten Open-Source-Lizenz erreicht.

Forschende profitieren zusätzlich von dem zusammengestellten Ransomware-Datensatz, welcher lauffähige Samples für die meisten bekannten Ransomware-Familien der letzten Jahren enthält und für die Analyse dieser Malwaregruppe eingesetzt werden kann. Ergänzt wird dieser durch die von jedem Ransomware-Sample verschlüsselten Dateien aus unserem Referenz-Datensatz. Forschende können anhand dieser Datensätze neuartige Erkennungsmethoden oder -systeme entwickeln, welche mittelfristig wiederum in den Klassifizierungsdienst einfließen können. Die wissenschaftliche Community soll durch eine Datenpublikation im Sinne des Open Science sowie einer zugehörigen wissenschaftlichen Veröffentlichung erreicht werden.

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

Sicherheit ist eines der Grundbedürfnisse des Menschen und muss sich im Zeitalter der Digitalisierung auch auf die „Digitale Welt“ erstrecken. Dazu gehört einerseits der Schutz vor Kriminellen, andererseits aber auch vor Überwachung durch Unternehmen oder staatlichen Stellen.

Die Wiederherstellung von durch Ransomware zerstörten Dateien schützt einerseits direkt vor kriminellen Erpressern, und kann damit zur IT-Sicherheit unserer Nutzergruppe beitragen. Die Nextcloud-App liefert zusätzlich ein weiteres kleines Alleinstellungsmerkmal von Nextcloud gegenüber öffentlichen Cloud-Anbietern. Dadurch unterstützt sie das Ziel, den Wechsel von kommerziellen Cloud-Anbietern mit teilweise fragwürdigen Geschäftspraktiken auf offene und vertrauenswürdige Infrastrukturen zu fördern.

Ein offener Datensatz von Ransomware-verschlüsselten Dateien unterstützt direkt den Open-Science-Gedanken. Daraus abgeleitete Erkenntnisse über die Verhaltensmuster von Ransomware sowie die Erkennung derselben mittels Machine Learning können dem allgemeinen wissenschaftlichen Fortschritt dienen.

Durch Offenlegung unserer Vorgehensweise möchten wir außerdem das Bewusstsein der interessierten Allgemeinheit für das Thema Ransomware schärfen. Wir hoffen dadurch sowohl die allgemeine Informationssicherheit zu verbessern, aber auch den Menschen durch Zugriff auf alle

Informationen die Möglichkeit bieten, sich an diesem Thema über das versehentliche Ausführen von Ransomware auf dem heimischen Rechner hinaus zu beteiligen.

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt?

- Die Nutzeroberfläche der Nextcloud-App wurde, dank der Unterstützung von SimplySecure im Rahmen des UI-Coachings, bereits erheblich nutzerfreundlicher gestaltet. Aus einer zweiten Runde von Feedback gegen Ende des Projekts ergaben sich weitere Verbesserungsmöglichkeiten, die nun ebenfalls eingepflegt werden sollen.
- Anhand von Statistiken aus einer großen (>1000 aktive Accounts) Nextcloud-Instanz wurde ein 1GB großer Datensatz mit „typischen“ Dateien geschaffen, der zur Analyse von Ransomware-Verhalten geeignet ist. Die Verteilung von Dateitypen und -größen in diesem Datensatz der Victim-Dateien entsprechen in etwa den Verteilungen auf der betrachteten Nextcloud-Instanz und sollten einigermaßen realistisch für verschiedene Dateisammlungen sein. Diese Dateien sind teilweise synthetisch aus Wikipedia-Artikeln erstellt, teilweise aus Wikimedia Commons entnommen, und mit den jeweiligen Provenienzmetadaten versehen, so dass diese Victim-Daten unproblematisch weiterverbreitet werden können.
- Eine Liste von 61 konkreten Ransomware-Samples, die in der gewählten Cuckoo-VM funktionieren und Dateien verschlüsseln. Viele Ransomware-Samples sind nicht mehr lauffähig, da sie entweder auf nicht mehr erreichbare Server im Internet angewiesen sind oder die unübliche Konfiguration der VM (Single-Core mit nur 2GB RAM / 32GB Festplatte) erkennen und die Funktion verweigern. Diese Liste kann besonders neuen Forschenden im Ransomware-Feld also erheblich Zeit sparen.
- Die Open-Source-Software Cuckoo Sandbox wurde so konfiguriert und angepasst, dass sie für die Analyse von Ransomware auf großen Datenmengen geeignet ist. Damit können die ausgewählten Ransomware-Samples zuverlässig auf den 1GB großen Referenzdatensatz angewendet werden.
- Aus der Ausführung dieser Samples ergibt sich ein weiterer, einzigartiger Datensatz von Dateien, die von Ransomware zerstört wurden. Zusätzlich zu den 45GB der eigentlichen Dateien sind in fast allen Fällen die exakten Timings der Schreib- und Löschvorgänge vorhanden, die Aufschluss über das Verhalten des jeweiligen Ransomware-Samples geben.
- Ein Satz von Machine-Learning-geeigneten Features auf diesen Dateien, die in einer Nextcloud-App leicht aus der Datei zu bestimmen sind. Da PHP dafür nur bedingt geeignet ist, sollten sich diese Features auch in jeder anderen Programmiersprache und -umgebung leicht bestimmen lassen. Für jedes Feature besteht zudem eine initiale Abschätzung, wie indikativ dieses Feature für Ransomware-zerstörte Dateien ist. Daraus lässt sich begründen, einige der initialen Features wegzulassen, insbesondere da sich die am schwersten zu berechnenden Features als nicht besonders nützlich erwiesen haben.
- Zu jeder Ransomware-zerstörten Datei sind die extrahierten Features sowie einige Metadaten als zusammengefasster Datensatz vorhanden. Dieser bildet die Grundlage für

sämtliche Machine-Learning-Anwendungen des Projekts und kann grundsätzlich auch für weitere Forschung in diesem Bereich genutzt werden.

- Es konnte nachgewiesen werden, dass bereits einfachste Modelle (in diesem Fall ein Linearer Classifier) recht gute Ergebnisse (ca. 96%) bringen. Damit sollte die Anwendung von Machine Learning hier eindeutig zielführend sein.
- Eine vorläufige REST-Schnittstelle des Klassifizierungs-Dienstes wurde definiert, um diesen aus beliebigen Programmiersprachen und damit Cloud-Diensten ansprechen zu können. Diese vorläufige Schnittstelle lässt sich leicht um weitere Features erweitern und kann damit bereits als Basis für Entwicklungen dienen.

Konnten alle Meilensteine erreicht werden?

- **M1 Datensatz:** abgeschlossen. Die erzeugten Datensätze von lauffähiger Ransomware, repräsentativen Dateien vor deren Zerstörung, sowie Ransomware-zerstörter Dateien sind bei den „konkreten Ergebnissen“ beschrieben.
- **M2 Modellauswahl:** begonnen. Die Anwendung einfacher Modelle zeigt, dass bereits diese gut funktionieren. Darauf ausgehend konnte die Menge der erforderlichen Features bereits reduziert werden, und die Nützlichkeit des Datensatzes mit Ransomware-zerstörten Dateien wurde nachgewiesen. Für den Abschluss dieses Milestones wird eine große Menge von Dateien benötigt, die für „ganz normale“ Dateien (d.h. vor der Zerstörung) repräsentativ sind. Die für die Victim-Dateien angewandte Methode stößt hier an ihre Grenzen.
- **M3 fertiges Modell:** begonnen. Ein initiales Modell ist vorhanden und sollte ersten Tests zufolge eine Genauigkeit von 96% erreichen. Dieses Modell wird entsprechend der Ergebnisse von M2 Modellauswahl fortlaufend aktualisiert. Hier ist abzusehen, dass bei Abschluss von M2 Modellauswahl dieses Modell nahezu ohne Mehraufwand zur Verfügung steht.
- **M4 Schnittstellen:** größtenteils abgeschlossen. Die vorläufige REST-Schnittstelle des Klassifizierungs-Dienstes muss noch um weitere Machine-Learning-Features erweitert werden, was aber zu keinen strukturellen Änderungen mehr führen wird. Das entsprechende Backend befindet sich in aktiver Entwicklung.
- **M5 Integration:** größtenteils abgeschlossen. Die Oberfläche der Nextcloud-App wurde benutzerfreundlicher gestaltet und auf ein strukturiertes UI-Framework umgestellt. Die Kommunikation mit unserem Klassifizierungsdienst ist implementiert und muss noch Integrationstests unterzogen werden.

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

Während der Projektphase hat sich die Koordination der Entwicklung an verschiedenen Standorten als schwieriger herausgestellt als ursprünglich erwartet. Durch Aufteilung der Arbeit in unterschiedliche, voneinander unabhängige Module konnten wir die Entwicklung aber auf relativ unabhängige Arbeit am jeweiligen Modul fokussieren. Für die verbleibende Koordination der Entwicklung, insbesondere der Schnittstellen und wichtiger technischer Entscheidungen, waren Internet-Telefonkonferenzen mit der Möglichkeit zum Screen-Sharing sehr hilfreich. Dies könnte als

Hilfestellung für andere Projektteams dienen, die ebenfalls nicht am selben Standort arbeiten können.

Das UI-Coaching durch SimplySecure brachte große Verbesserungen an der Oberfläche der Nextcloud-App. Wir würden dieses Coaching deshalb allen Projekten nahelegen, die unentschieden sind, welches der beiden angebotenen Coachings (UI oder Projektmanagement) für sie sinnvoller ist. Eine weitere Empfehlung daraus ist, nach Möglichkeit bereits mit einem Mockup oder einer bestehenden Oberfläche mindestens ins zweite Coaching zu gehen, weil dadurch die Iteration im Rahmen des Coachings erheblich produktiver ist.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

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

Wie bei den Zielgruppen beschrieben, ergibt sich für diese der folgende Nutzen:

- Die Nextcloud-nutzende Allgemeinheit kann die App nutzen, um nach einem Ransomware-Angriff zerstörte Dateien einfach wiederherzustellen.
- Technisch versierte Personen können unsere Komponenten als Basis für eigene Entwicklungen nutzen.
- Forschende können die erstellten Datensätze im wissenschaftlichen Bereich einsetzen.

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

Durch die Open-Source-Stellung unseres Vorgehens sowie unserer Ergebnisse bei der Ransomware-Analyse hat jeder Einblick in unsere Arbeit und kann bei Interesse nachvollziehen, wie wir Ransomware analysiert, die Ergebnisse interpretiert und aus diesen interpretierten Daten eine Software entwickelt haben, welche Nutzer vor Ransomwareangriffen schützt. Das dient einerseits dem Open-Science-Gedanken, Einblick in die eigene Forschungsarbeit zu gewähren, und verbessert andererseits die Vertrauenswürdigkeit der entwickelten Software, da die Herkunft der Modelle besser nachvollzogen werden kann. Das Klassifizierungsmodell kommt nicht aus einem undurchsichtigen Forschungsprojekt oder ist Firmengeheimnis einer Antivirus-Firma, sondern ist das Ergebnis eines offenen und dokumentierten Prozesses.

Die Open-Source-Stellung ist in unserem Fall auch ein wichtiges Mittel, um die Nachhaltigkeit der Projektergebnisse sicherzustellen. Die Methode an sich sowie die Nextcloud-App im speziellen lassen sich, auch wegen vieler Softwarepatente in diesem Bereich, nicht soweit kommerzialisieren, dass die Entwicklung und Pflege ohne Unterstützung durch die Open-Source-Community möglich ist.

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

- Wir möchten einen größeren (50-100GB) Referenzdatensatz von nicht zerstörten Dateien sammeln. Dieser Referenzdatensatz einerseits und unser Datensatz von Ransomware-zerstörten Dateien andererseits bildet dann die beiden Klassen des Supervised Learning, mit dem wir die Modell- und Feature-Auswahl weiterführen möchten.

- Anhand des daraus gewonnenen Feature-Sets soll eine erste stabile Version der REST-Schnittstelle des Klassifizierungs-Dienstes definiert werden. Parallel soll eine Referenzimplementation des Klassifizierungs-Dienstes erstellt werden, welche dann für die Integration in die Nextcloud-App genutzt werden kann.
- Die Nextcloud-App soll fortlaufend an neuere Nextcloud-Versionen angepasst werden. Soweit möglich, planen wir außerdem neue Ransomware-Familien weiterhin zu analysieren und die Modelle entsprechend anzupassen.

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

Die Arbeit am Projekt ermöglichte uns einerseits die fachliche Weiterbildung im Bereich der Analyse von Malware, der Methoden des Machine Learning sowie dem Entwurf und der Entwicklung grafischer Benutzeroberflächen. Andererseits konnten wir unser Wissen über Open-Source-Software und Projektentwicklung weiter vertiefen und hatten Einblick in die Möglichkeiten und Herausforderungen der selbstständigen Softwareentwicklung, die wir so vor dem Projekt noch nicht in Betracht gezogen hatten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

- Die ursprünglich vorgesehene Filterung nach dem Vorkommen des Wortes „ransomware“ in den Erkennungen von VirusTotal war zu selektiv, da nur wenige Hersteller erkannte Ransomware auch mit dem Wort „ransomware“ beschreiben. Üblicher sind Erkennungen der Art „Trojan.Ransom.*“ oder „Ransom.*“. Daher wurde stattdessen nur nach „ransom“ gesucht, sowie nach einer Liste von benannten Ransomware-Familien, die manuell aus verschiedenen Quellen zusammengetragen wurde.
- Die Gruppierung von Ransomware in Familien mittels des automatisierten Werkzeugs AVclass war für eine Auswahl der auszuführenden Samples nicht ausreichend. Es wurden mehrere tausend Familien erkannt, ohne dass diese anhand der Anzahl Samples weiter gefiltert werden konnten: Auch Familien mit nur einem erkannten Sample enthielten bekannte Ransomware. Initial wurde daher die (kuratierte) öffentliche „theZoo“ Sammlung genutzt. Danach wurden Samples gewählt, die auf die manuell zusammengetragenen Namen der Ransomware-Familien passten.
- Bei 1GB Daten funktionierte die in Cuckoo eingebaute Funktion zur Sammlung veränderter Dateien nicht mehr zuverlässig. Außerdem waren einige Ransomware-Samples in der Lage, die Analyse durch Cuckoo nahezu vollständig zu umgehen und dadurch unbeobachtet Veränderungen an Dateien innerhalb der VM vornehmen. Daher wurde nach jeder Analyse ein Snapshot der Festplatte der VM erstellt, aus dem die vollständige Liste der veränderten oder zerstörten Dateien in jedem Fall wiederhergestellt werden kann. Hierbei geht allerdings die Information zum Timing der Schreib- und Löschzugriffe größtenteils verloren.

- Cuckoo ist in der Lage, den Netzwerkzugriff für Malware einzuschränken, unterstützte in unserer Umgebung aber nur Vollzugriff oder keinen Netzwerkzugriff. Durch eine kleine Anpassung am Source Code konnten wir zusätzlich den Zugriff nur auf Web-Dienste ermöglichen, ohne vollständigen Internetzugriff zu ermöglichen. Diese Anpassung ist für unsere Anwendung sinnvoll, für Cuckoo sollte aber eventuell eine andere, generellere Methode entwickelt werden.
- Ursprünglich war geplant, mittels tensorflow.js auch einer Erkennung ohne laufenden Klassifizierungsdienst zu ermöglichen. Wegen der erforderlichen Datenweitergabe vom Backend bis ins Frontend im Browser erscheint diese Lösung aber nicht wirklich praktikabel. Wie die Architektur entsprechend angepasst werden muss ist derzeit noch nicht entschieden.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Die beschriebenen Zielgruppen sollen durch Platzierung der jeweiligen Ergebnisse in jeweils geeigneten Kanälen erreicht werden:

- Die aktualisierte Version der Nextcloud-App „Ransomware Recovery“ wird nach angemessenen Tests über die normalen Update-Kanäle verbreitet.
- Alle Daten und Informationen zu den durchgeführten Experimenten, sowie der Source Code der entwickelten Software, werden öffentlich und zentral gesammelt über eine GitHub-Organisation (<http://github.com/undo-ransomware>) zugänglich gemacht.
- Die Datensätze sollen auf Zenodo bereitgestellt werden. Zusätzlich ist geplant, die Datensätze und das Vorgehen durch eine wissenschaftliche Publikation der Öffentlichkeit zur Verfügung zu stellen.
- Die jeweiligen Ressourcen sollen auf der Informationsseite zum Projekt auf prototypefund.de (<https://prototypefund.de/project/undo-von-ransomware-mittels-machine-learning/>) verlinkt werden, so dass sie ausgehend vom Projekt leicht gefunden werden können.

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?

Beim Meilenstein „M1 Datensatz“ stellte sich die Systematisierung unseres Datensatzes als wesentlich umfangreicher als ursprünglich geplant heraus. Jedoch ist und war er zentraler Meilenstein unseres Projekts, da er einerseits die Grundlage für den Machine-Learning-Aufgabenteil darstellt, andererseits aber auch das größte Potential für die Nachnutzung durch die wissenschaftliche Community bietet. Um unbrauchbare Ergebnisse durch schlechte Eingabedaten zu vermeiden, wurde der Schwerpunkt auf diesen Milestone gelegt und die Modell- und Feature-Auswahl entsprechend etwas zurückgestellt. Dies führte dazu, dass der Milestone „M1 Datensatz“ parallel zu allen anderen Meilensteinen bis zum Projektende weitergeführt wurde, da wir immer

weiteren Einblick auf das Verhalten von Ransomware bekamen. So war es uns möglich einen einzigartigen Datensatz zu erstellen, der das schlussendlich zu entwickelnde Modell auf wissenschaftlich solide Beine 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? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Ein weitere Suche nach wissenschaftlichen Veröffentlichungen in diesem Bereich förderte etliche Publikationen während der Projektlaufzeit zu Tage. Diese unterteilen sich in die folgenden Familien:

- Anwendung von Machine-Learning-Methoden auf den Maschineninstruktionen oder dem (lokalen) Verhalten eines Ransomware-Programms zur Erkennung desselben. Diese Methoden können, wie klassische Antivirus-Programme, die Recovery auf Nextcloud komplementieren, sind aber für das Projekt an sich nicht relevant.
- Anwendung von Machine-Learning-Methoden auf den Netzwerkverkehr eines Rechners, um den Netzwerkverkehr von Ransomware zu erkennen. Dies kann im Rahmen des Netzwerkmanagements bei größeren Institutionen sehr hilfreich sein und die Ausbreitung von Malware effektiv einschränken. Unsere Erfahrung zeigt aber, dass viele Ransomware-Familien auch ohne Netzwerkzugriff sehr effektiv Dateien zerstören. Wir sehen diesen Forschungsbereich deshalb als orthogonal zu unserem Projekt an.
- Anwendung von Machine-Learning-Methoden auf die geschriebenen Daten.

Die Ergebnisse von 2 Veröffentlichungen dieser letzten Familie sind direkt relevant im Rahmen des Projekts, da eine sehr ähnliche Fragestellung erprobt wird. Aufgrund des Publikationsdatums gegen Ende des Projekts konnten diese Ergebnisse im Projekt leider nicht mehr genutzt werden, sondern sind in die Ideen zur Weiterentwicklung unserer Lösungen eingeflossen.

- K. Lee, S. Lee and K. Yim, „Machine Learning Based File Entropy Analysis for Ransomware Detection in Backup Systems,“ in *IEEE Access*, vol. 7, pp. 110205-110215, 2019.
<https://doi.org/10.1109/ACCESS.2019.2931136>
Lee et al. verwenden sehr ähnliche Methodiken in einem effektiv identischen Setup (Backups). Die dortigen Ergebnisse zeigen, dass die Erprobung weiterer Features und Machine-Learning-Algorithmen zielführend ist.
- M. J. May and E. Laron, „Combating Ransomware using Content Analysis and Complex File Events,“ 2019 10th IFIP International Conference on New Technologies, Mobility and Security (NTMS), CANARY ISLANDS, Spain, 2019, pp. 1-5.
<https://doi.org/10.1109/NTMS.2019.8763851>
May und Laron verwenden eine sehr viel umfangreichere Analyse der Dateiinhalte mittels Apache Tika. Eine direkte Anwendung muss erwogen werden, sofern in PHP und als Nextcloud-App umsetzbar. Die Evaluation anhand einer einzelnen Ransomware-Familie bestärken uns aber auch darin, unseren Datensatz sorgfältig zu beschreiben und zu veröffentlichen.

Im Bereich der Ransomware selbst trat mit Sodinokibi eine neue Familie (bzw. Variante von GandCrab) in Erscheinung. Da unser Basis-Datensatz für Ransomware vom Februar 2019 stammt, war Sodinokibi in diesem Datensatz nicht vorhanden. Daher wurden 7 Samples von Sodinokibi manuell von VirusShare heruntergeladen und in unserer Ransomware-VM analysiert. Davon erwiesen sich 6 als lauffähig und sind in unserem Datensatz zerstörter Dateien vertreten.

Richtlinie zum „Software-Sprint“

Digital_Bargeld – Infrastruktur für bargeldlose Zahlungen ohne Überwachung

Schlussbericht

Zuwendungsempfänger:

Florian Dold

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S63 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Durch technischen Fortschritt verdrängen bargeldlose Bezahlssysteme immer mehr die Benutzung von Bargeld im Alltag vieler Bürger*innen. Während die Bequemlichkeit dieser Zahlungsmöglichkeiten willkommen ist, werden die Nachteile oft übersehen: Die derzeitige Umsetzung ermöglicht die Totalüberwachung der ökonomischen Aktivität von Benutzer*innen.

Proprietäre Lösungen wie Apple Pay und Google Pay basieren auf geschlossenen Implementationen/Schnittstellen und werden von mächtigen US-Amerikanischen Oligopolen kontrolliert. Diese erlangen so Zugriff auf die Aktivitätsprofile von Benutzer*innen, welche dann für Werbezwecke ausgewertet und/oder verkauft werden.

GNU Taler ist ein freies Online-Bezahlssystem welches die Privatsphäre von Kund*innen schützt und gleichzeitig das Einkommen von Händler*innen gegenüber Buchprüfer*innen offenlegt. Damit werden illegale Aktivitäten wie Steuerhinterziehung oder der Handel mit illegalen Waren erschwert. Die bisherige Implementation von GNU Taler war jedoch stark auf das Bezahlen von Online-Inhalten im Browser fokussiert.

Ziel dieses Projekts war es, GNU Taler mit NFC-Zahlungen auf Hardware-Plattformen (primär Smartphones) zu erweitern, um somit auch bequeme bargeldlose Zahlungen in Präsenzsituationen zu ermöglichen. Dies umfasst Szenarien, in denen nur eine der beiden Parteien Internetzugang hat.

Es waren folgende Meilensteine geplant:

Meilenstein 1 (1. März – 31. März 2019): Design und Spezifikation des Taler NFC-Protokolls zwischen Käufer*innen und Bezahlterminal der Händler*innen.

Meilenstein 2 (1. April - 15. Mai 2019): Implementation und Testing einer Android-Bibliothek für das Taler NFC-Protokoll.

Meilenstein 3 (16. Mai - 30. Juni 2019): Implementation einer Bezahlterminal-App für die Android-Plattform, welche das NFC-Protokoll aus Meilenstein 1 und 2 unterstützt und mit dem existierenden GNU Taler Backend der Händler*innen kommuniziert.

Meilenstein 4 (1. Juli - 31. Juli 2019): Portierung des existierenden GNU Taler Wallets (= digitale Geldbörse) auf die Android-Plattform.

Meilenstein 5 (1. August - 31. August 2019): Integration der NFC-Zahlungen und der dazugehörigen Benutzeroberfläche in das GNU Taler Wallet auf Android, mit abschließendem Testing.

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

Das Projekt richtet sich an Institutionen und Organisationen die ihren Mitgliedern oder Kund*innen ein bargeldloses Zahlungsmittel anbieten wollen, ohne dabei die Privatsphäre der Käufer*innen zu verletzen. Im Kleinen könnte das Mensa/Cafeteria einer Schule, Universität oder Firma sein.

In einem Größeren Umfang können Finanzdienstleister (wie z.B. Banken) Händler*innen und Kund*innen untereinander Zahlungen abwickeln lassen und von Transaktionsgebühren profitieren.

Kunden und Händler profitieren dabei von der Austauschbarkeit des Zahlungsdienstleisters und Wettbewerb unter den Dienstleistern, da das Bezahlssystem auf einem offenen Protokoll mit offener Implementation basiert.

Ausführliche Darstellung der Ergebnisse

Folgende Ergebnisse wurden erzielt:

- Entwicklung und Dokumentation der Taler NFC Spezifikation (<https://docs.taler.net/taler-nfc-guide.html>) und der "taler://" URI-Schema-Spezifikation (<https://docs.taler.net/core/taler-uri.html>). Diese beiden Spezifikationen ermöglichen gemeinsam das Auslösen von Bezahlvorgängen (so wie Geldabhebevorgängen) in der Taler Wallet per NFC ("tap-to-pay") so wie durch das Scannen eines QR-Codes.
- Entwicklung von *node-android-v8*, einem Fork des Node.JS Projekts, der die nötigen Änderungen am Code und Buildsystem enthält, um den Node.JS Interpreter für Android-Geräte mit ARM Prozessor bauen und ausführen zu können.
- Entwicklung der *akono*-Bibliothek (Android Kotlin Bindings for Node.JS), welche es ermöglicht, beliebige Node.JS-Anwendungen von einer Kotlin-Anwendung auf Android zu starten und mit diesen Anwendungen per IPC (Interprozesskommunikation) zu interagieren.
- Erweiterung des existierenden Taler-Merchant-Backends. Das Taler-Merchant-Backend ist ein Dienstprozess, welcher eine einfache Schnittstelle für Händler*innen zur Zahlungsabwicklung zur Verfügung stellt. Im Rahmen dieses Vorhabens wurde das neu entwickelte "taler://" URI-Konzept in das Backend integriert.

- Entwicklung der *taler-merchant-terminal* Android-Anwendung. Diese Anwendung erlaubt es, eine Zahlungsaufforderung für eine Bestellung zu erzeugen und einen Zahlvorgang in der Wallet-Anwendung des Kunden anschließend per NFC oder QR-Code auszulösen. Bei erfolgreichem Abschluss der Zahlung wird dem Händler in der Anwendung eine Mitteilung angezeigt. Zudem kann die Historie der Zahlungsvorgänge betrachtet werden.
- Entwicklung der *android-wallet* Android-Anwendung. Diese Anwendung ermöglicht es Benutzer*innen, digitales Geld abzuheben und anonym auszugeben. Dabei wird die existierende Wallet mit Hilfe der *akono*-Bibliothek auf Android ausgeführt.
- Entwicklung von *idb-bridge*, einer Bibliothek welche die von der W3C standardisierte und in Browsern weit verbreitete IndexedDB Datenbankschnittstelle (<https://www.w3.org/TR/IndexedDB/>) für Anwendungen in der Node.JS-Umgebung zur Verfügung stellt. Die implementation basiert auf der existierenden *fakeIDB* Bibliothek, welche mit Persistenz und einem neuen Speicher-Backend basierend auf persistenten B+-Bäumen erweitert wurde.
- Integration des QR-Code Konzepts auf den Taler-Demo Websites. Es ist nun möglich, mit der Android-Wallet einen QR-Code auf einer Website im Browser eines andere Computers zu scannen, um z.B. für einen digitalen Zeitungsartikel anonym zu bezahlen.
- Überarbeitung der Projektseite (<https://taler.net/en/>) im Rahmen des UX-Coachings das durch die Open Knowledge Foundation organisiert wurde.

Im Rahmen der Arbeit an dem Projekt konnte die technische Herangehensweise der Portierung validiert und als erfolgreich eingestuft werden.

Der ursprünglich geplante Funktionsumfang konnte erreicht werden. Alle Meilensteine (mit den im Abschnitt "Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung" erläuterten Änderungen) wurden erreicht.

Im Rahmen des durch die Open Knowledge Foundation organisierte UX-Coaching wurde daran gearbeitet, die Kommunikation des Projekts mit der Öffentlichkeit zu verbessern. Es wurde gemeinsam die Informationsarchitektur der GNU Taler Website überarbeitet. Zudem wurden Werkzeuge erläutert, mit denen die Verständlichkeit der Anwendungen für Benutzer*innen verbessert werden kann. Ein hier besonders wertvolles Werkzeug war das Aufstellen von Wortlisten die in der öffentlichen Kommunikation benutzt (mit Begründung) bzw. nicht benutzt (mit Alternative) werden sollen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Die Umsetzung dieses Projekts stellt einen wichtigen Meilenstein in der Entwicklung einer freien Infrastruktur für digitale Bezahlsysteme dar, welche die Privatsphäre der Benutzer*innen nicht verletzen, jedoch trotzdem das Vorgehen gegen Steuerhinterziehung oder illegale Geschäfte ermöglichen.

Die "Nebenprodukte" dieses Vorhabens sind als Open-Source auch generell einsetzbar: Die *Akono*-Laufzeitumgebung kann als Startpunkt verwendet werden, um andere JavaScript/Node.JS Projekte nach Android zu portieren. Die "*idb-bridge*" Bibliothek kann zum Portieren und Testen von Programmen verwendet werden, die ursprünglich in einer Browser-Umgebung laufen.

Durch die Arbeit an dem Projekt war es mir möglich, mich tief in die Node.js+V8 Laufzeitumgebung einzuarbeiten und die Entwicklung von Android-Anwendungen mit Kotlin zu erlernen. Zudem lernte ich bei dem UX Coaching hilfreiche Werkzeuge, die ich auch in weiteren Projekten einsetzen kann.

Noch dieses Jahr wird voraussichtlich eine Gruppe von Studenten der Berner Fachhochschule an einem Projekt arbeiten, in dem das GNU Taler NFC-Protokoll in einen Warenverkaufsautomat integriert werden soll. Dieser Arbeit liegt die im Rahmen dieses Vorhabens entwickelte NFC-Spezifikation zu Grunde, so wie die Android Wallet-Application.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Der anfängliche Arbeitsplan enthielt einen Meilenstein zur Implementation einer Hilfsbibliothek für das NFC-Protokoll. Die Architektur des Bezahlterminals wurde in der Planungsphase jedoch sehr stark vereinfacht. Anstatt selbst Anfragen des Wallets zu bearbeiten, leitet das Bezahlterminal diese Anfragen an das existierende Merchant-Backend weiter. Dort werden die Anfragen des Wallets entweder direkt bearbeitet, oder an die Exchange (d.h. den Zahldienstleister) weitergeleitet. Die per NFC gesendeten Anfragen werden als in JSON (JavaScript Object Notation) gekapselte HTTP-Requests dargestellt. Diese vereinfachte Architektur und das vereinfachte Format von Anfragen macht die Implementation einer gesonderten NFC-Bibliothek für das Bezahlterminal unnötig. Stattdessen musste nur eine einfache Weiterreichung von Anfragen an das Merchant-Backend implementiert werden.

Zudem wurde ursprünglich geplant, die graphische Oberfläche der Android-Komponenten in einer plattformunabhängigen Programmierumgebung (z.B. Flutter oder React Native) zu implementieren. In der Designphase der grafischen Oberfläche stellte sich jedoch heraus, dass dieser Ansatz nicht zielführend war, da viele Interaktionsmuster der grafischen Oberfläche auf verschiedenen Plattformen zur Verbesserung der Benutzbarkeit anders gestaltet werden müssen.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Der Quellcode des GNU Taler Projekts kann auf der folgenden Website abgerufen werden und auch in das Versionskontrollprogramm Git importiert werden: <https://git.taler.net/>

Speziell im Rahmen des Vorhabens "Digital_Bargeld" wurde primär an den Repositories "akono.git", "wallet-core.git", "android-node-v8.git", "wallet-android.git", "merchant.git", "merchant-terminal-android.git", "docs.git" und "exchange.git" gearbeitet.

Generelle Informationen über das gesamte Projekt so wie Dokumentation können unter <https://taler.net/> abgerufen werden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Kostenplanung wurde weitgehend eingehalten. Die Komplexität der Portierung der Wallet (Meilenstein 4) erforderte einen Mehraufwand. Zudem wurde dieser Meilenstein in der Projektplanung an den Anfang verlegt.

Die Implementation der NFC-Bibliothek (Meilenstein 2) wurde aus der Projektplanung gestrichen, da in dem Design des NFC-Protokolls (Meilenstein 1) eine vereinfachte Architektur herausgearbeitet

wurde, die eine Taler-spezifische NFC-Bibliothek unnötig macht. Mit dieser Änderung konnte der anfänglich geplante Funktionsumfang gänzlich erreicht werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Während der Förderung bestand regelmäßiger Austausch mit anderen Mitgliedern des GNU Taler Projekts. Dies diente hauptsächlich der Absprache von Protokolländerungen, die im Rahmen des Vorhabens nötig waren.

Zudem bestand kurzer Kontakt mit dem V8-Projektteam von Google, um gemeinsam einen Fehler (siehe <https://bugs.chromium.org/p/v8/issues/detail?id=9171>) in dem V8 JavaScript-Interpreter zu beheben, da dieser Fehler ein Hindernis bei der Portierung des GNU Taler Wallets ein Hindernis war.

Richtlinie zum „Software-Sprint“

LOTRANSLATE – KI-Übersetzung für LibreOffice

Schlussbericht

Zuwendungsempfänger:

Dr. Thomas Viehmann

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S64 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 war es, eine leicht nutzbare KI-basierte Übersetzungsfunktion für die LibreOffice Writer-Textverarbeitung zu entwickeln. Diese sollte es Benutzern ermöglichen Texte offline, d.h. ohne dass die Daten den eigenen Computer verlassen, zu übersetzen. Die Möglichkeit der Offline-Nutzung war mir im Hinblick auf die Privatsphäre und breiter Anwendbarkeit wichtig.

Die wichtigsten geplanten Meilensteine waren:

1. Erstellen eines Minimal nutzbaren Produkts
2. UX-Konzeption
3. Hardkodierung durch Konfigurationsmöglichkeiten ersetzen
4. Erweiterung der Sprachmodelle und Benutzerfreundliche Trainingsmöglichkeit inkl. Fine-Tuning-/ Domain-Adaptation-Möglichkeit, d.h. die Möglichkeit den „Wortschatz“ auf eigenes (Fach- o.ä.) Vokabular anzupassen.
5. Konzeption für die Nutzbarkeit von nicht Satz-gematchten Corpora
6. Breite Vorstellung auf Vorträgen etc., Veröffentlichung als „fertige“ LibreOffice-Extension

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 Office-Benutzer, insbesondere solche, die aus Kostengründen oder Besorgnis über ihre Privatsphäre LibreOffice benutzen (Wikipedia gibt für den Vorgänger OpenOffice einen Marktanteil von ca. 20% in Deutschland an und 9-22% in Europa).

LibreOffice hat einen "AppStore" (<https://extensions.libreoffice.org/>). Dort wurde die Erweiterung zur Verfügung gestellt (und der Quelltext auf GitHub). Auf der internationalen LibreOffice-Konferenz habe ich das Projekt und die Ergebnisse vorgestellt. Das Interesse der LibreOffice-Entwickler an der Funktionalität ist groß, so dass die Chance besteht, durch stärkere Integration in den LibreOffice-Kern einen noch breiteren Kreis an Nutzern zu erschließen.

Das Projekt macht so eine aktuelle KI-Entwicklung breiten Nutzerschichten frei zugänglich, ohne dass sie die Hoheit über ihre Daten verlieren.

Ausführliche Darstellung der Ergebnisse

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

Das zentrale Ziel – die Erweiterung für LibreOffice, die es ermöglicht mit einfachster Schnittstelle KI-basiert Texte zu übersetzen zu erstellen, veröffentlichen und bekannt zu machen – wurde entlang der Meilensteine 1-3 und 6 sehr gut erreicht. Insbesondere gibt es eine herunterladbares LibreOffice-Extension – für Windows auch als „sorglos“-Paket, dass auch unerfahrene Benutzer installieren können.

Das über die OKF / den Prototype Fund vermittelte UX-Coaching durch 360° lieferte ein für die Konzeption der Bedienung ungemein wertvolle Impulse. Dies ermöglichte, die Ursprünglich (in Anlehnung an bestehende kommerzielle Lösungen) übermäßig komplexe Benutzerschnittstelle radikal zu vereinfachen.

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 Erweiterung ist auf sehr gute Resonanz gestoßen, denn sie füllt eine echte Lücke.

Da die Entwickelte Erweiterung *LibreOffice Translate* die – nach meinem Kenntnisstand – einzige Umsetzung einer KI-basierten Übersetzungsfunktionalität in LibreOffice ist, stößt sie hier für die sehr große LibreOffice-Nutzerschaft eine Tür auf. Entsprechend stieß die Extension nach meinem Vortrag auf der LibreOffice-Konferenz auf große Resonanz.

Ich habe die Hoffnung – und erste Anzeichen sprechen dafür – dass der Open-Source-Charakter nicht nur zur Verbreitung und ggf. Weiterentwicklung aus neuen Anwendungsfeldern beiträgt, sondern vor allem auch aus der Community weitere Übersetzungsmodelle zur Verfügung gestellt werden.

Insofern hat die Vervollständigung der Trainingsfunktionalität hohe Priorität als Weiterentwicklung über den im Software-Sprint erreichten Stand hinaus.

Für mich persönlich war einerseits das UI-Coaching sehr hilfreich und ich werde bei neuen Projekten die Techniken, die ich durch es für mich entdeckt habe, nutzen können. Andererseits war es – gerade in dem übermäßig kommerzialisierten Gebiet der KI – für mich persönlich sehr wichtig, hier in einem mir sonst nicht möglichen Umfang an einem gemeinwohlorientierten Projekt zu arbeiten zu können.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

Die Benutzerschnittstelle wurde radikal vereinfacht und auf alle nicht direkt nützliche Funktionalität verzichtet. Dies betrifft einerseits die Kernfunktion der Übersetzung – die mit einem Klick durchführbar ist – bei der die gleiche Funktionalität einfacher gestaltet wurde, als auch die erweiterten Analysefunktionen, die aufgrund der UX-Überlegungen als letztlich nicht ausreichend nützlich erschienen.

Für das Training der Modelle wurden Meilensteine 4 und 5 nicht komplett erreicht. Die „Standard“-Trainings-Verfahren sind sehr rechenaufwändig. Hierdurch wurde der Aufwand der Bearbeitung relativ – sowohl an Rechenkapazitäten als auch Entwicklungszeit – stark erhöht, so dass hier bei Meilenstein 4 sehr gute Ansätze entwickelt wurden, diese aber in den nächsten Monaten noch komplettiert und veröffentlicht werden müssen. Zu Meilenstein 5 – der bewusst sehr ambitioniert und deshalb spekulativ war – wurden nur Vorarbeiten durchgeführt. Erschwerend kam hinzu, dass eine angedachte Datenquelle – übersetzte Literatur von Projekt Gutenberg – aus aufgrund eines deutschen Gerichtsurteils und der Reaktion des Projekts – aus Deutschland nicht mehr abgerufen werden kann. Dies wäre technisch zu umgehen, für eine *Anleitung* zur eigenen Weiterentwicklung ist es aber hinderlich.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Das Programm ist als LibreOffice-Extension unter <https://github.com/lernapparat/lotranslate> veröffentlicht. Eine Verlinkung auf <https://extensions.libreoffice.org/> ist in der Prüfung durch die Administratoren der Seite.

Ich habe auf der LibreOffice Conference Vorgetragen – Vortragsfolien sind unter <https://libocon.org/2019/program/schedule/sept-13th-friday/> abrufbar, ein Video des Vortrags folgt.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

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

Die Programmierung im LibreOffice-Kontext stellte sich als viel herausfordernder dar als zunächst angenommen. Erwartungsgemäß stieß das Projekt in den verschiedenen Communities (LibreOffice, OpenNMT) auf großes Interesse. Bedingt durch die extrem hohe Komplexität der LibreOffice-Software und die überschaubar große Entwickler-Community und damit der verfügbaren Entwicklerdokumentation sowie Beispielen war das Umsetzen jeder Funktion mit relativ großem experimentellen / Debug-Aufwand verbunden. Kompensiert wurde dieser Mehraufwand vor allem durch die in Folge der UX-Analyse radikal einfach gehaltene Benutzerschnittstelle.

Bei den trainingsbezogenen Teilen ergab sich – wie in den anderen Abschnitten dargestellt – ein Mehraufwand, der sich darin niederschlägt, dass die Projektziele nicht vollständig im Rahmen des Projektes erreicht wurden. Die partiellen Ergebnisse zwar wenig sichtbar, jedoch fruchtbar, so dass die Umsetzung in Anwenderfreundliche Dokumentation und ein Trainings-Tool zeitnah als erfolgen kann und damit das Projekt vervollständigt.

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?

Die KI-Technologie zur maschinellen Übersetzung entwickelte sich schnell weiter – durch Entwicklungen im Bereich der Sprachverarbeitung wie BERT oder OpenAI GPT2, die zwar in anderem Kontext mediale Aufmerksamkeit erhielten, aber deren zugrundeliegende Techniken auch auf die Übersetzung Anwendung finden. Einerseits kann LibreOffice Translate durch die Verwendung von OpenNMT davon profitieren, andererseits sind gehen damit auch einige der erhöhten Aufwände einher, die wesentlich dazu beitrugen, dass für das Training der KI-Modelle die Projektziele nicht erreicht wurden. Dennoch scheint es als wäre durch diese Entwicklungen – wenn auch mit Verzögerung – das Projektergebnis besser im Sinne von besseren Übersetzungen und damit noch nützlicher.

Richtlinie zum „Software-Sprint“

HASS_FILTERN

Hasskommentare automatisiert filtern – Eine interaktive Erklärung

Schlussbericht

Zuwendungsempfänger:

Johannes Filter

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S65 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)?

In den Kommentarspalten des Internet tobt der Hass. Um diesen zu einzudämmen, müssen Kommentare manuell geprüft werden. Weil dies viele Redaktionen überfordert, schließen sie die Kommentarfunktion gleich ganz ab. In diesem Kontext gelobt Machine Learning Besserung, indem automatisiert Kommentare ausgewertet werden. Die aktuellen Verfahren sind jedoch unausgegoren. So können sie leicht „ausgetrickst“ werden oder sind rassistisch.

Aktuell ist jedoch noch nicht klar, welche Aufgaben wir den Maschinen überlassen wollen. Damit es darüber eine gesellschaftliche Debatte gibt, müssen zuerst breite Teile der Bevölkerung die dahinter liegenden technischen Mechanismen verstehen. Weil die zugrundeliegenden Mechanismen für Laien schwer zu durchdringen sind, bereiten wir diese pädagogisch auf, damit sich mehr Menschen in die Debatte um Machine Learning einbringen können.

In diesem Projekt werden Techniken des Natural Language Processing am Beispiel der automatisierten Identifikation von Hasskommentaren erklärt. Neben Grafiken und Text gibt es einen

interaktiven Teil, mit dem die Nutzer*innen live im Browser ein Machine-Learning-Verfahren nutzen können. Durch die spielerische Aufbereitung werden Möglichkeiten aber auch Risiken erläutert.

Zudem war es das Ziel, dass das Softwaresystem mit innovativen Methoden umgesetzt wird. Hier gab es einige Änderungen zu dem ursprünglichen Plan. Schlussendlich entstand ein Softwarepaket, dass sich an NLP-Expert:innen richtet. Es benutzt ein Verfahren, dass sich in wissenschaftlichen Veröffentlichungen bewährt hat. Bis jetzt gab dazu keine effiziente Implementierung.

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 Webseite richtet sich an Menschen, die den aktuellen Hype um Machine Learning nachvollziehen wollen. Diese müssen explizit nicht “in der Materie stecken”, daher wird kein Wissen vorausgesetzt um die Seite nutzen zu können. Die erste Zielgruppe profitiert von dem Bildungsangebot und somit wird deren Data Literacy gestärkt. Der Umgang mit modernen Technologien ist heute essentiell. Die Zweite Zielgruppe umfasst die Natural Language Community.

Das Projekt war an den Themenschwerpunkt “Maschinelles Lernen” orientiert. Zu einem wurde notwendige Bildungsarbeit geleistet aber zum anderen auch eine Arbeit die dem technischen und wissenschaftlichen Fortschritt dient.

Ausführliche Darstellung der Ergebnisse

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

Das Ziel des Projektes war eine Webseite auf der erklärt wird, wie Natural Language Processing am Beispiel von Kommentaren funktioniert. Die meisten Meilensteine wurde erreicht, jedoch ist die finale Veröffentlichung noch offen. Die Webseite besteht aus Text, Grafiken und einem interaktiven Teil. Die Idee der “Word Embedding” werden auf der Seite erklärt. Dabei handelt es sich um ein Verfahren des Natural Language Processing. In diesem wird Bedeutung in Wörtern ‘erzeugt’. Der Computer kann Text nicht den Sinn von Wörtern erfassen. Word Embedding generiert ein Model, bei dem ich z. B. zu einem gegebenen Wort ähnlich Wörter oder Synonyme erhalte. Die Ähnlichkeit wird darüber erzeugt, in welchem ähnlichem Kontexten sie vorkommen.

Die technische Umsetzung, sprich das Erstellen der Word Embeddings, war auch Teil des Projekts. Es gibt in diesem Bereich schon viel Forschung, aber wenig einfach zu benutzende Open-Source-Software. Und oft verlangen bestehende Softwarelösungen große Datenmengen. In vielen Fällen, wie in unserem, gibt es diese jedoch nicht. Daher ist unsere Lösung für Bereiche, in denen es wenig Daten gibt. Es wurde das Paper von Omer Levy et al. “Improving Distributional Similarity with Lessons Learned from Word Embeddings” implementiert. Die Autoren der Studie hatten mit ihrem Paper Code veröffentlicht, dieses war jedoch nicht für den normalen Gebrauch geeignet. In diesem

Forschungsumfeld spielen die Qualität keine Rolle und die meiste Software ist sehr prototypisch. Die Method ist jedoch weiterhin State-of-the-Art und so entschied ich mich die Software weiter zu entwickeln. Dazu habe ich Teile wiederverwendet, aber viele Teile neu geschrieben. Mit der Software habe ich mehrere Word Embeddings trainiert und auch zu weiteren Verwendung zum download angeboten.

Die Mitarbeiter und Mitarbeiterinnen der Open Knowledge Foundation Deutschland haben mir bei Fragen zur Förderung stets zurate. Ich konnte mit ihnen auch inhaltliche Fragen in Bezug zur Öffentlichkeitsarbeit klären. Zudem habe ich mich mit mehreren Akteuren ausgetauscht, die sich intensiv mit der Kommentarkultur im Internet beschäftigen.

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?

Durch die Open-Source-Lizenz kann eine Weiterarbeit problemlos erfolgen. Ich bin bereits in Gesprächen mit Forschern, die den Code für eine Veröffentlichung benutzen wollen.

In dem Projekt konnte ich vor allem meine technischen Fähigkeiten weiterentwickelt. Es war wichtig, dass der Code möglichst effizient läuft.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

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

Zunächst gab es die Planung direkt im Browser Machine Learning durchzuführen. Erste Tests mit Nutzer:innen ergaben jedoch, dass es für sei irrelevant ist, wie die Idee genau technisch umgesetzt werden sollte. So kam es durch eine Überarbeitung der Projektplanung und der finalen Ergebnisse. Am Anfang gab es den Plan richtiges Hass auf der Webseite darzustellen, jedoch wurde die Idee verworfen. Dieser sollte nicht noch mehr Aufmerksamkeit erhalten.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

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

Die Ergebnisse werde final auf der Webseite <https://kommentare.vis.one> präsentiert werden. Die Software ist unter <https://github.com/jfilter/ptf> zu finden. Da die Arbeit in mehrere kleinere Softwarepakete aufgeteilt it, gibt es Auflistung mit Links und Erläuterungen.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

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

Es gab einen Mehraufwand zu Ende des Projektes, sodass die finale Veröffentlichung verschoben werden musste. Wahrscheinlich wird sie im Oktober 2019 erfolgen wird.

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.