Richtlinie zum „Software-Sprint“

TimeTiles – Wiederverwendbare Komponenten und Widgets für zeitabhängige Geodaten

Schlussbericht

Zuwendungsempfänger*in:

Johannes Filter

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Forschung, Technologie und Raumfahrt unter dem Förderkennzeichen 16IS25S03 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei dem*der Autor*in.

Kurze Darstellung der Aufgabenstellung und Motivation in der regulären Förderphase (Entwicklung eines Prototypen)

Chroniken – etwa zu Konfliktdaten, historischen Erinnerungsorten, Demonstrationen oder gesellschaftlichen Ereignissen – sind ein zentrales Werkzeug der politischen Bildung. Sie verwandeln abstrakte Statistiken in nachvollziehbare Einzelfälle, schaffen Kontext und stärken Data Literacy. In den letzten Jahren habe ich an fünf solcher Chroniken mitgearbeitet und dabei immer wieder festgestellt: Jede dieser Webseiten erfindet das Rad neu – Karten, Filter, Zeitleisten, Kategorienauswahl, Datenpflege. Vorhandene Lösungen wie Forensic Architectures timemap, umap oder kepler.gl decken jeweils nur Teilbereiche ab; wer mehr will, muss forken oder selbst bauen.

Mein Ziel war eine Open-Source-Grundlage, die Journalist*innen, Forscher*innen, Aktivist*innen und NGOs erlaubt, eigene Chroniken aus orts- und zeitbezogenen Daten zu erstellen, ohne grundlegende Funktionen erneut implementieren zu müssen. Die geplante Vorgehensweise war eine konfigurierbare TypeScript-Software-Bibliothek für React und Next.js mit Widgets für Karte, Filter, Suche, Zeitleiste und Datenanbindung.

Die im Antrag benannten Meilensteine der regulären Förderphase:

1. Design-System in Figma gestalten
2. Datenverarbeitung (z. B. Geocoding)
3. Layout-Komponente (z. B. Listenansicht)
4. Filter- und Suchfunktionen

5. Visualisierungskomponenten

6. Starterkit-Funktion als Kommandozeileninterface umsetzen

7. Finale Dokumentation

8. Veröffentlichung von Version 1.0

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

TimeTiles adressiert beide Themenfelder des Software-Sprints: Es ist eine wiederverwendbare digitale Infrastruktur für die Aufbereitung und Veröffentlichung orts- und zeitbezogener Daten und steht unter AGPLv3 frei zur Verfügung. Statt für jede Chronik eine eigene Codebasis aufzubauen, können zivilgesellschaftliche Akteur*innen auf einer gemeinsamen Plattform aufsetzen, die Datenimport, Geocoding, Karten, Filter, Suche und Zeitleisten standardisiert.

Der Sicherheits- und Demokratiebezug ergibt sich aus den typischen Anwendungsfällen: Konfliktdaten (etwa aus dem Uppsala Conflict Data Program, UCDP, oder dem Humanitarian Data Exchange der UN, HDX), historische Erinnerungsarbeit (Stolpersteine, Mauertote), Demonstrations- und Versammlungsdokumentation, Recherchen im Bereich Menschenrechte und politische Gewalt sowie die Aufarbeitung gesellschaftlicher Konflikte. Das Projekt senkt die Hürde, solche Recherchen für eine breite Öffentlichkeit verständlich aufzubereiten, und stärkt damit Transparenz, journalistische und zivilgesellschaftliche Recherche sowie kritische Datenkompetenz.

Durch konsequente Open-Data-/Open-Source-Stellung der Komponenten, der Beispieldatenpakete und der Dokumentation auf docs.timetiles.io trägt das Projekt zudem zur Verbreitung etablierter offener Standards im zivilgesellschaftlichen Tech-Stack bei – insbesondere PostGIS (Geodaten in PostgreSQL), MapLibre (Kartendarstellung), GeoJSON (Datenformat für räumliche Daten) und H3 (hexagonale räumliche Indexierung von Uber).

Ausführliche Darstellung der Ergebnisse in der regulären Förderphase (Entwicklung eines Prototypen)

Sieben der acht Antrags-Meilensteine wurden erreicht, einer (Starterkit-Funktion als Kommandozeileninterface) wurde nach einem Pivot bewusst aufgegeben (Details siehe Abschnitt zu nicht weiterverfolgten Arbeiten). Der wesentliche Kurswechsel der ersten Phase: Statt einer reinen Komponentenbibliothek ist TimeTiles zu einer integrierten Open-Source-Plattform geworden, weil sich früh zeigte, dass eine Library allein das eigentliche Problem (Daten kuratieren, importieren, hosten) für die Zielgruppe nicht löst.

Bilanz pro Antrags-Meilenstein:

1. Design-System – inhaltlich umgesetzt, mit Schwerpunktverschiebung vom reinen Figma-Artefakt zum lauffähigen Code-Designsystem auf Basis von Tailwind, shadcn/ui und Radix. Diese Verschiebung machte das Designsystem direkt produktiv nutzbar, statt einen separaten Design-zu-Code-Übersetzungsschritt zu erfordern.

2. Datenverarbeitung (z. B. Geocoding) – erreicht. Multi-Provider-Geocoding (Nominatim, Google) mit Provider-Konfiguration über die Datenbank und persistentem Location-Cache, Schema-Erkennung für CSV, Excel und ODS, konfigurierbare Field-Mappings.

3. Layout-Komponente (z. B. Listenansicht) – erreicht. Listen-/Grid-Ansicht mit URL-Synchronisation, Kartenseite mit Clustering und Zeitleiste, Detail-Modal mit URL-State.

4. Filter- und Suchfunktionen – erreicht. Dynamische Filter nach Kategorie, Datum und Ort, Volltextsuche, Histogramm-basierte Zeit-Filter, geteilte URL-Filter-States.

5. Visualisierungskomponenten – erreicht. Interaktive MapLibre-Karte mit Clustering, automatische Histogramme/Zeitleiste, Bar-Charts, Detailansichten.

6. Starterkit-Funktion als Kommandozeileninterface (Python/Textual/cookiecutter) – nicht weiterverfolgt. Stattdessen wurde ein integrierter Web-basierter Import-Wizard mit Schema-Detection und Mapping-Vorschlägen umgesetzt, der dieselbe Aufgabe ergonomischer und für Nicht-Entwickler*innen zugänglich löst.

7. Finale Dokumentation – erreicht. Eigene Dokumentationsseite unter docs.timetiles.io mit Anwender*innen-Guide, Entwicklungs-Doku und API-Referenz.

8. Öffentliche Veröffentlichung und Versionierung – erreicht, mit angepasster Release-Strategie. Das Repository github.com/jfilter/timetiles ist unter AGPLv3 öffentlich verfügbar, dokumentiert und testbar; der Stand zum Ende der regulären Förderphase ist als Tag v0.1.0-alpha.1 markiert. Der ursprünglich im Antrag als „Version 1.0" bezeichnete Meilenstein wurde damit inhaltlich erfüllt: Der Code ist offen, dokumentiert und nachvollziehbar versioniert. Auf eine stabile 1.0-Bezeichnung wurde jedoch bewusst verzichtet, weil der Praxiseinsatz zeigte, dass insbesondere Datenpipeline und Multi-User-Zugriffskontrolle weitere Härtung benötigen – wie bei vergleichbaren jungen Open-Source-Projekten üblich.

Über die Antrags-Meilensteine hinaus entstanden vier zentrale Bausteine, die die Grundlage für die produktive Nutzung legen: erstens eine skalierbare Monorepo-Architektur (pnpm + Turbo) mit Headless-CMS-Integration über Payload CMS 3, die Content-Management und Page-Builder-Blöcke direkt in die Plattform einbindet; zweitens eine PostGIS-basierte Geodaten-Pipeline für die Verarbeitung orts- und zeitbezogener Daten; drittens Mehrnutzer*innen-Funktionen mit einem Trust-Level-System (Stufen 0–5), Quotas, Rate Limiting, eigentums-basierter Zugriffskontrolle, umfassendem Audit-Logging und Webhook-Triggern für geplante Imports; viertens eine durchgängige Qualitätssicherung über CI/CD mit GitHub Actions, SonarCloud-Code-Analyse, Trivy-Security-Scans und Pre-Commit-Hooks.

Die Begleitung durch die Open Knowledge Foundation – Zwischengespräche, Sprechstunden, der Austausch mit der Kohorte und das Demo Day-Format – hat geholfen, Prioritäten realistisch einzuschätzen und den Pivot zur Plattform an den Bedürfnissen der Zielgruppe auszurichten.

Kurze Darstellung der Aufgabenstellung und Motivation in der Second-Stage-Förderung (Aufbau einer nachhaltigen Projektstruktur)

Mit dem Ende der ersten Phase lag ein funktionsfähiger Prototyp der Plattform vor. Für eine nachhaltige Projektstruktur fehlte vor allem die Verbreitung im Zielgruppenkreis und ein Plan, wie sich das Projekt nach Förderende selbst trägt. Ziel der Second-Stage-Förderung war daher der Aufbau einer nachhaltigen Projektstruktur mit Schwerpunkt Verwertung: TimeTiles bekannter machen, anhand realer Anwendungsfälle stresstesten und genug Aufmerksamkeit aufbauen, damit die Weiterentwicklung im Anschluss über Kundenaufträge und/oder

Folgeförderungen finanziert werden kann. Kern der Vorgehensweise war eine Kooperation mit einer Partnerorganisation samt begleitender Content-Strategie.

Die im Antrag benannten Meilensteine der Second-Stage-Verlängerung:

1. Kooperation mit einer Organisation vereinbaren

2. Erstellung eines ersten Prototyps ohne Anpassungen von TimeTiles

3. Anpassung von TimeTiles und Veröffentlichung der Version 2.0

4. Zweiter Prototyp im Rahmen der Kooperation

5. Weitere Anpassungen von TimeTiles und Veröffentlichung der Version 3.0

6. Finaler Prototyp für die Kooperation

Ausführliche Darstellung der Ergebnisse in der Second-Stage-Förderung (Aufbau einer nachhaltigen Projektstruktur)

Der Schwerpunkt der Second Stage lag dem Antragsplan entsprechend nicht auf reiner Software-Entwicklung, sondern auf Verwertung – also auf Netzwerken, Coaching, der konkreten Zusammenarbeit mit einer Partnerorganisation und dem Aufbau eines Finanzierungs- und Geschäftsmodells für die Zeit nach der Förderung. Software-Entwicklung war in dieser Phase nur ein kleinerer Anteil der Gesamtarbeit. Die Plattform wurde dabei gezielt anhand realer Anforderungen weiterentwickelt, statt isoliert weitere Features zu produzieren.

Bilanz entlang der Antrags-Meilensteine:

1. Kooperation mit einer Organisation vereinbaren – erreicht. Es kam eine Kooperation mit einer Berliner zivilgesellschaftlichen Organisation (NGO/Think-Tank) zustande, in deren Rahmen TimeTiles für die Aufbereitung interner Datenbestände eingesetzt wurde. Die Akquise und Abstimmung lief über bestehende Kontakte.

2./4./6. Erstellung mehrerer Prototypen im Rahmen der Kooperation – inhaltlich umgesetzt. Die Zusammenarbeit verlief iterativ entlang realer Datenbestände und Rückmeldungen der Partnerorganisation. Technisch validiert wurden insbesondere Import-Workflows aus tabellarischen Daten, wiederkehrende Datenaktualisierungen, Field-Mapping, Filterlogik, Zugriffsbeschränkungen sowie die Verständlichkeit der Oberfläche für nicht-technische Nutzer*innen. Die Vertraulichkeit bezog sich auf die konkreten Dateninhalte, nicht auf die technischen Erkenntnisse – die identifizierten Bedarfe sind direkt in die öffentliche TimeTiles-Codebasis zurückgeflossen und stehen damit auch anderen Anwender*innen zur Verfügung.

3./5. Anpassung von TimeTiles (im Antrag als Version 2.0 / 3.0 benannt) – als aufeinanderfolgende Alpha-Releases umgesetzt. Diese Entscheidung war bewusst: Die Kooperation zeigte, dass TimeTiles bereits produktiv erprobt werden kann, aber noch nicht die Stabilität einer öffentlichen Major-Version versprechen sollte. Die technische Weiterentwicklung ist nachvollziehbar dokumentiert über die Tags v0.1.0-alpha.2 (Refactoring und Code-Qualität, 12.03.2026), v0.1.0-alpha.3 (Scheduled-Imports-UI und Edit-Wizard, 22.03.2026) sowie v0.1.0-alpha.4 (Dataset-zentrierte Filter, 31.03.2026 – Stand zum Förderende).

Netzwerkarbeit war ein Hauptfokus der Second Stage und hat einen substanziellen Teil der Arbeitszeit beansprucht: regelmäßige Zwischengespräche mit dem PTF-Team, Sprechstunden (u.a. Superbloom), aktive Teilnahme am Demo-Day-Format und am PTF-Community-Abend, kontinuierlicher Austausch mit der PTF-Kohorte, Pitch zur Second-Stage-Verlängerung samt Pitch-Deck-Erstellung, gezielte Sondierungsgespräche mit potenziellen Partnerorganisationen aus dem zivilgesellschaftlichen Recherche- und NGO-Umfeld sowie der Aufbau eines Verteilers für die spätere Anschlussverwertung. Begleitend wurden die im Programm angebotenen Coachings (u.a. mit Iris Wellmann und mit VDI/VDE-IT) gezielt entlang der Verwertungsfragen genutzt: Geschäftsmodell und Rechtsform für ein Open-Source-basiertes Angebot, kostenpflichtige Zusatzleistungen rund um die offene Kernsoftware, Organisationsentwicklung und Vision für eine perspektivische Skalierung sowie strategisches Netzwerken im zivilgesellschaftlichen Tech-Umfeld.

Im Rahmen des PTF-Abschlussevents wurde zudem ein eigener User-Testing-Workshop mit eingeladenen Gästen aus Zielgruppen-Organisationen (potenzielle Anwender*innen aus Recherche-, Datenjournalismus- und NGO-Kontexten) durchgeführt, um die Plattform an realen Anforderungen zu validieren und Feedback für die Weiterentwicklung zu sammeln. Aus diesen und weiteren Sondierungsgesprächen sind derzeit konkrete Folgegespräche aktiv – mit der bestehenden Partnerorganisation zur Verstetigung sowie mit weiteren potenziellen Anwender*innen aus dem Recherche- und Datenjournalismus-Umfeld – die die Grundlage für eine Anschlussverwertung nach Förderende bilden sollen.

Als zentrales Kommunikationsmittel für die zweite Phase wurde bewusst ein Newsletter gewählt: eine direkte, dauerhafte Verbindung zu interessierten Nutzer*innen, Partnerorganisationen und potenziellen Anwender*innen, unabhängig von Plattform-Algorithmen. Die Newsletter-Anbindung ist entsprechend auch direkt in die Plattform integriert.

Begleitend wurden Ergebnisse der ersten Phase konsolidiert (Plattform-Stabilität, Datenpipeline, Visualisierung, Dokumentation), um die produktive Nutzung in der Kooperation und durch perspektivische weitere Anwender*innen zu ermöglichen.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zielgruppe sind Journalist*innen, Forscher*innen, Aktivist*innen und NGOs, die orts- und zeitbezogene Daten verständlich publizieren möchten, sowie Entwickler*innen, die solche Webseiten für diese Akteur*innen bauen. Angesprochen wird die Zielgruppe über mehrere Kanäle: die direkte Zusammenarbeit mit einer Partnerorganisation als reale Anwendungsreferenz, die Dokumentationsseite docs.timetiles.io, das öffentliche GitHub-Repository (Issues und Discussions), sowie das im PTF-Programm aufgebaute Netzwerk und Folgegespräche mit weiteren potenziellen Anwender*innen. Für die Anschlussphase sind bereits konkrete Projektvorstellungen vor einer klar umrissenen Berliner Zielgruppe aus Recherche-, Datenjournalismus- und NGO-Kontexten in Planung.

Konkreter Nutzen: Eine Tabelle mit Titel, Datum und Ort wird mit wenigen Klicks zu einer interaktiven, durchsuchbaren Darstellung mit Karte, Zeitleiste, Filtern und Detailansicht. Recherchen und Datenprojekte, die bisher Wochen Implementierungszeit gekostet haben, lassen sich deutlich schneller publikationsfähig aufbereiten. Auch Organisationen mit internen, nicht

öffentlichen Datenbeständen können die Plattform nutzen, ohne ihre Daten an Dritte abgeben zu müssen.

Stand der Nutzbarkeit zum Förderende: TimeTiles ist lokal installierbar, dokumentiert und anhand mitgelieferter Beispieldatenpakete testbar. Parallel hat die schrittweise öffentliche Bereitstellung unter timetiles.io begonnen, sodass Interessierte das Projekt zunehmend ohne eigenen Server-Betrieb kennenlernen können. Für den produktiven Eigenbetrieb wird in dieser Übergangsphase weiterhin technische Begleitung empfohlen.

Offene Punkte: Härtung für produktive Multi-Tenant-Nutzung (Voraussetzung für den weiteren timetiles.io-Rollout), weitere Tests mit heterogenen Datensätzen und eine Vereinfachung des Deployments für weniger technische Organisationen stehen in der Anschlussphase im Fokus. Diese Schritte sind im Repository als Issues und ADRs dokumentiert.

Modulare Nachnutzung: Trotz des Pivots zur integrierten Plattform bleibt die Codebasis als Monorepo organisiert, sodass einzelne Bausteine auch außerhalb der Gesamtplattform nachgenutzt werden können – insbesondere die wiederverwendbare React-/UI-Komponentenbibliothek (`packages/ui`, basierend auf shadcn/ui und Radix). Damit ist das ursprüngliche Antragsziel – wiederverwendbare Komponenten und Widgets – auf modularer Ebene weiterhin erfüllt.

Transfer in die Community: Durch die öffentliche Dokumentation, das aktive GitHub-Repository (Issues, Discussions) und die nachvollziehbar getaggten Alpha-Releases können andere Projekte bereits jetzt auf den Ergebnissen aufbauen – sei es durch Einbinden einzelner Pakete oder durch Forken/Selbsthosten der gesamten Plattform – auch wenn die stabile 1.0 noch aussteht.

Open-Source-Effekte: Die AGPLv3-Stellung erschwert proprietäre Abschottung und stellt sicher, dass veröffentlichte oder als Dienst bereitgestellte Weiterentwicklungen unter den Bedingungen der AGPL wieder dem Ökosystem zugutekommen. Die Plattform basiert vollständig auf offenen Standards (PostGIS, GeoJSON, MapLibre, OpenStreetMap).

Weiterentwicklung: Aus der laufenden Kooperation und den parallelen Sondierungsgesprächen ergeben sich konkrete Anschlussmöglichkeiten. Die unter „Stand der Nutzbarkeit" beschriebene öffentliche Bereitstellung unter timetiles.io ist dabei bewusst als Einstieg in ein gehostetes Angebot auf Basis der offenen Kernsoftware konzipiert, das insbesondere nicht-technische Organisationen ohne eigenen Server-Betrieb adressiert. Einnahmen sollen mittelfristig über Hosting, Wartung, Beratung, Datenmigration und projektspezifische Anpassungen entstehen; der offene Kern von TimeTiles bleibt dabei weiterhin öffentlich verfügbar. Vergleichbare Modelle gibt es etwa bei Plausible, Cal.com, Sentry und PostHog. Mögliche Folgeförderungen sollen die strukturelle Weiterentwicklung des offenen Kerns absichern.

Persönliche Weiterentwicklung: Die zehn Monate haben mich nicht nur technisch weitergebracht, sondern – insbesondere in der Second Stage – vor allem in den Bereichen Akquise, Kund*innen- und Partner*innen-Kommunikation, Anforderungsklärung sowie Aufbau eines tragfähigen Verwertungsmodells. Die Begleitung durch die Open Knowledge Foundation, die im Programm angebotenen Coachings und der Austausch mit der Kohorte haben geholfen, die technische Arbeit konsequent auf reale Verwertung und Zielgruppenbedarfe auszurichten.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Ein größerer Lösungsansatz aus dem Antrag wurde im Projektverlauf bewusst aufgegeben:

• Starterkit-Funktion als Kommandozeileninterface (Python-CLI mit Textual und cookiecutter, Meilenstein 6 der ersten Phase): Gespräche mit potenziellen Nutzer*innen aus dem Chronik- und Recherche-Umfeld in der Konzeptionsphase zeigten, dass die größte Hürde nicht das initiale Aufsetzen einer Webseite ist, sondern der laufende Datenimport, die Datenpflege und das Hosting. Ein CLI-Generator hätte dieses Kernproblem nicht gelöst. Stattdessen wurde TimeTiles zu einer integrierten Plattform mit Web-basiertem Import-Wizard weiterentwickelt – das deckt denselben Anwendungsfall ergonomischer ab und macht das Tool auch für Nicht-Entwickler*innen zugänglich.

Darüber hinaus wurden im Verlauf einzelne kleinere technische Optionen verworfen, wenn sich im Austausch mit der Partnerorganisation oder der Kohorte zeigte, dass eine andere Umsetzung den realen Anforderungen besser gerecht wird. Diese Entscheidungen sind jeweils im öffentlichen Repository nachvollziehbar.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

Code-Repository: https://github.com/jfilter/timetiles (Monorepo, AGPLv3-Lizenz).

Dokumentation: https://docs.timetiles.io

Issue-Tracker und Diskussionen: https://github.com/jfilter/timetiles/issues und https://github.com/jfilter/timetiles/discussions.

Förder-Hinweise und Logos sind in der README.md des Repositories sowie im Footer der Projektseite (timetiles.io) und der Dokumentationsseite (docs.timetiles.io) eingebunden.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Generell erfolgte die Arbeit entsprechend dem Plan. Im Verlauf gab es zwei inhaltliche Anpassungen, die sich im Aufwand weitgehend ausgeglichen haben: zum einen ein Mehraufwand durch den Pivot von einer reinen Komponentenbibliothek zu einer integrierten Plattform – als Voraussetzung dafür, dass das Tool in einer realen Kooperation einsetzbar ist. Zum anderen ein Minderaufwand durch den Wegfall des ursprünglich geplanten Python-CLI-Starterkits.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Nein, externe Entwicklungen hatten keinen wesentlichen Einfluss auf Zielsetzung oder Vorgehen des Projekts.