Richtlinie zum „Software-Sprint“

arso – Archipel

Schlussbericht

Zuwendungsempfänger:

Nieß, Heinzmann, Brock-Wenzek u. Sievers GbR

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S41 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Das Internet bietet viele Wege, Material zu veröffentlichen, doch ist dessen Erhalt und dauerhafte Zugänglichkeit keineswegs sicher. Oft landen Medien auf proprietären Plattformen, wo alles an deren Fortbestehen, Geschäftsinteresse und Zugangsoffenheit hängt. Ein mögliche Alternative ist die eigene Website, die aber Kosten hat, nur mit Aufwand gut kopierbar und bei temporären Unterfangen wie Konferenzen oder sozialen Bewegungen oft mit den Inhalten irgendwann wieder offline ist. Auch große Mediensammlungen sind davon z.B. bei auslaufender Finanzierung betroffen. Digitale Selbstbestimmung, die über den Moment hinaus reicht, braucht also Werkzeuge, um Dinge auf Dauer teilen und zugänglich machen zu können.

Grundsätzlich ist das System für unterschiedlichste Arten und Formen von Archiven einsetzbar. Unser Fokus wird auf bereits bestehenden Sammlungen und Archiven gemeinnütziger TrägerInnen liegen. Zu Beginn wird es dabei um bereits vorhandenes, öffentlich zugängliches Material gehen. Wird der NutzerInnen-Kreis über diese erste Schiene hinaus erweitert, werden wir auch auf datenschutzrechtliche Einschränkungen sowie auf das "Recht auf Vergessen"

aufmerksam machen. Technische Wege zur Depublizierung werden ebenfalls implementiert.

Wie:

Im Zentrum steht die archipel-app: Eine Desktop-App, um Archive zu erstellen, anzusehen und zu teilen. Archipel basiert auf Dat, einem progressiven Open-Source-Projekt. Dieses versionierte, dezentrale und sichere P2P-Dateiystem mit Datenbank ist unser Backend. Die Archipel-App wird in Javascript/Node "universal" geschrieben und läuft damit sowohl als offline-first P2P-App, als installierte Desktop-App (via Electron) sowie als Webapp an einem simplen Backend. Die App wird über verschiedene Plugins Inhalt hinzufügen (Medien importieren, ganze Websites archivieren, Twitter-Feeds inkl. verlinkter Medien auslesen) sowie grundlegende Metadaten erfassen können. Sie kann erstellte Archive über das Dat-Netzwerk veröffentlichen, außerdem andere Archive lokal spiegeln, teilen und betrachten. Vorhandene Open-Source-Tools (Hashbase, Datbase) können die Archive auf Servern spiegeln und für alle Browser zugänglich machen, diese werden wir geringfügig für unseren Usecase erweitern. Durch eine Vielzahl vorhandener Libraries aus dem Dat-Umfeld und diverse Vorarbeiten können wir die App als MVP im Prototype-Fund zuverlässig umsetzen. Wir legen dabei mehr Wert auf saubere Implementierung, Tests, Dokumentation und Reusability als auf möglichstviele Features gleich zu Beginn. Bei der Entwicklung werden wir auch die Skalierbarkeit hin zu vielen und/oder großen Archiven untersuchen. Das Dat-Projekt, auf dem wir aufbauen, wurde schon für sehr große Datensätze (mehrere hundert GB) erfolgreich eingesetzt. Das Dat-Protokoll ist für eine weitgehende Skalierbarkeit ausgelegt. Im produktiven Einsatz wurden hier schon mehrere Zehntausende Archive über einzelne Rechner erfolgreich geteilt. Wir werden entsprechende Weiterentwicklungen verfolgen und auch selbst im Rahmen des Förderprogramms Tests mit großen, vielen und weit verteilten Archiven vornehmen.

Meilensteinen

1. Backend für die Verwaltung mehrerer Dat-Strukturen (hyperdb, hyperdrive, hypergraph).
2. Im Backend UI-kompatible Wrapper für die grundlegenden Funktionen der Dat-Strukturen erstellen. Also öffentliches announcing der Strukturen, Anlegen von Dateneinträgen, etc. Hierbei zunächst Fokus auf Dateien, Metadatenmanagement in einer späteren Iteration.
3. Entwicklung eines modularen Setups für das User Interface mit React.js und verschiedenen Bundling-Konzepten.
4. Entwicklung des konkreten User Interfaces: Archivverwaltung, Datei-Management.
5. Entwicklung eines Metadaten-Store zum Anlegen und Verwalten von Meta-Informationen über die Dateien (zunächst Titel, Beschreibung, Tags). Dies beinhaltet entsprechende Backend- und Frontend-Komponenten. Das Backend ist ein Triple-Store aufbauend auf den Dat-Datenstrukturen.
6. Exemplarische Importer: Ein algorithisches Hinzufügen großer Datenbestände, z.B. ganzer Websites. Importer sind immer stark vom jeweilig importierten Inhalt abhängig und müssen dementsprechend immer wieder neu angepasst werden.
7. Metadaten-Hub. Die Darstellung/das Browsing des Archivs anhand der bestehdenen Metadaten zu dessen Inhalten.

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

Mit arso/Archipel tragen wir folgendermaßen zu den Zielen bei: Archipel soll den Erhalt von kleinen Blogs und Websites sowie der Datenbestände gemeinnütziger Initiativen, Vereine und Grassroots-Bewegungen vereinfachen. Gerade diese oft mit begrenzten finanziellen Mitteln ausgestatteten gesellschaftlichen Akteure sind von Data-Deterioration, also dem Verfall der von ihnen erstellten digitalen Inhalte, betroffen. Archipel leistet einen Beitrag dies aufzuhalten und die von den

genannten Akteuren erstellten Inhalte zu archivieren. Dies ist insbesondere wichtig, da diese gemeinnützigen Initiativen oft Fördergelempfänger*innen sind und der Verfall ihrer Arbeit auch den Verfall von Steuergeldern bedeutet.

Ausführliche Darstellung der Ergebnisse

Archipel ist eine Node.js-Anwendung die sowohl lokal als auch remote ausgeführt und in beiden fällen mittels Webbrowsers, im ersten auch als Desktop-App (via Electron) bedient werden kann. Im Backend ist es möglich unterschiedliche Datenstrukturen des Dat-Stacks (hyperdb, hyperdrive, hypergraph) zu verwalten und die in Dat bereits vorhandene peer-to-peer-Technologie zu nutzen. Die Dat-Strukturen werden zu Archiven zusammengefasst und können als gesamtes Archiv oder einzeln im peer-to-peer-Netz geteilt weren. Mittels des User Interfaces ist es möglich mehrerer dieser Archive zu verwalten. Dies umfasst das Erstellen eines neuen Archivs, das Hinzufügen des Archivs von anderen Peers, sowie die Änderungen an einer remote-Kopie eines Archivs zurück-zu-sychronisieren. Innerhalb eines Archivs ist es möglich Dateien und Ordner hochzuladen und zu erstellen. Gängige Dateitypen (Rohtext, PDF, JPEG, PNG, MP3, OGG, ...) können dargestellt bzw. wiedergegeben werden. Die Möglichkeit diese Dateien mit Metadaten zu versehen und im Archiv anhand dieser Metadaten zu browsen ist in Grundzügen vorhanden. Ebenso ist die Möglichkeit bestehende Webseiten und Blogs in ein Archiv zu importieren in Grundzügen vorhanden.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Zur Zielgruppe zählen alle, die an dem langfristigen Erhalt ihrer digitalen Inhalten interessiert sind. Insbesondere möchten wir Vereine, Initiativen und Grassrootsprojekte ansprechen. Diese leisten oft einen immensen Beitrag zur politischen und kulturellen Bildung und erstellen zu diesem Zweck liebevoll Webseiten und Blogs mit äußerst interessanten Beiträgen. Leider haben eben diese Akteure oft eine begrenzte Lebensdauer und damit auch die von ihnen

veröffenlichten Inhalte. Zudem sind die Inhalte oft nur dann zu finden, wenn man genau weiß wonach man sucht.
Wir konnten Grundlagen schaffen für die Weiterarbeit an diesem Usecase. Wir haben sehr viel gelernt in den letzten 6 Monaten, sowohl im Feld der Softwareentwicklung als auch beim Projektmanagment und dem Aufbau eines zukunftsfähigen Projekts.
Wir haben nicht nur offenen Quelltext produziert, sondern sind nun auch Teil der aktiven Entwickler-Community rund um das Dat-Projekt, dessen Fortentwicklung unserem Projekt unmittelbar zunutzen kommt. Die enge Verzahnung dieser Projekte ist nur dank Open-Source möglich.
Konkreter Nutzen für die genannten Zielgruppen ergibt sich, wenn unser Prototyp die notwendige Weiterentwicklung erfährt, um vom Alpha-Stadium zum getesteten und zuverlässigen Release weiterentwicklet werden kann. Wir sind optimistisch, dass wir mit arso diese nächsten Schritte gehen können - dank der Grundlagenarbeit, die im Rahmen des Software-Sprints geleistet werden konnte.
Konkrete, anstehende Weiterentwicklungen sind:

- Tests, Error-Reporting, Stabilität
- Ausbau der Metadaten-Verwaltung
- Entwicklung von Werkzeugen um Websites und Blogs direkt aus einem Archipel-Archiv zu betreiben um so die die Trennung von Veröffentlichung, Sicherung und Archivierung aufzuheben

Für die Umsetzung dieser und weiterer Schritte sehen wir nicht nur technisch, sondern auch organisatorisch mit den im Prototype Fund gewonnen Erfahrungen optimistisch entgegen.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Die Kopplungspunkte zwischen Front- und Backend in unserem isomorphen Stack mussten wir mehrfach überdenken und teilweise neu ansetzen, da die erste Implementierung die Umsetzung wichtiger Features erschwert hat. Im Rahmen eines iterativen Programmierkonzepts ist dies jedoch ein normaler Vorgang.

Aktuelle Entwicklungen von JavaScript haben zu teils unerwarteten Komplexitäten geführt aufgrund von Inkompatibilitäten zwischen Browsern (Frontend) und NodeJS (Backend). Diese konnten wir jedoch soweit möglich in den Griff bekommen.

Die Architektur der Metadatenverwaltung erforderte und erfordert ausgiebige Abwägung zwischen der Komplexität von RDF und semantischer Korrektheit einerseits und dem Anspruch, unnötige Komplexität zu vermeiden andererseits.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

https://arso.xyz/archipel

https://github.com/arso-project/archipel

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die Entwicklung mancher von uns verwendeten Open-Source-Modulen aus dem Dat-Stack verlief langsamer, als von uns (und dem Dat-Projekt selbst) zu Beginn der Förderperiode erwartet. Dadurch musste von unserer Seite teils mehr Aufwand für Grundlagenarbeit investiert werden. Aufgrund dessen mussten wir den Metadatenaspekt in Teilen weiter nach hinten und damit aus der Förderperiode hinaus verschieben.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Unser Produkt basiert in weiten Teilen auf dem Dat-Projekt. Dort steht derzeit eine größere Erneuerung an ("multidrive hyperdrive"), die in der Fertigstellung mehr Zeit in Anspruch nimmt als anfangs erwartet. Das hat uns teilweise vor Probleme gestellt. Zugleich haben wir enorm profitiert von der Entwickler-Community rund um das Dat-Projekt.