Richtlinie zum „Software-Sprint“

OpenPIMS 2.0 - Zentrale Einwilligungsverwaltung, Cookie-Banner abschaffen

Schlussbericht

Zuwendungsempfänger*in:

Stefan Böck

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Forschung, Technologie und Raumfahrt unter dem Förderkennzeichen 16IS25S07 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei dem*der Autor*in.

Kurze Darstellung der Aufgabenstellung und Motivation in der Förderphase

Motivation: Cookie-Banner sind ein Symbol für den unzureichenden Umgang mit Datenschutz im Internet. Sie überfordern Nutzer*innen mit komplexen Einwilligungsfragen und führen oft zu einer Blindes-Klicken-Kultur, bei der der Schutz personenbezogener Daten verloren geht. Menschen mit körperlichen Einschränkungen stoßen dabei auf zusätzliche Barrieren, und Unternehmen kämpfen mit der komplexen technischen Umsetzung datenschutzrechtlicher Anforderungen.

Problemstellung: Aktuell fehlt eine datenschutzfreundliche, nutzerzentrierte Infrastruktur, die Cookie-Banner obsolet macht und gleichzeitig die Einhaltung gesetzlicher Vorgaben sicherstellt. Mit der Verabschiedung des Telekommunikation-Digitale-Dienste-Datenschutz-Gesetzes (TDDDG), Paragraph 26, im Oktober 2024 stehen die finalen Rahmenbedingungen für die Abschaffung von Cookie-Bannern fest.

Geplante Vorgehensweise und Meilensteine:

1. TDDDG-Anpassungen: Anpassung von OpenPIMS an die finalen TDDDG-Anforderungen (Überarbeitung der Datenmodelle, Backend-Logik, Webinterface, API)

2. Browsererweiterungen: Entwicklung der Browsererweiterung für Firefox und Safari mit Tests zur plattformübergreifenden Kompatibilität

3. Cookie-Privacy-Proxy: Implementierung des universellen Cookie-Privacy-Proxys in Node.js, inklusive grundlegender Filterlogik und API-Anbindung an OpenPIMS

4. Sicherheitskonzept: Erstellung eines Sicherheitskonzepts als Vorbereitung für die Anerkennung durch den Bundesbeauftragten für Datenschutz und Informationsfreiheit

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

Bezug zum Themenfeld Infrastruktur: OpenPIMS 2.0 schafft eine zentrale Open-Source-Infrastruktur für die Verwaltung von Nutzereinwilligungen. Die Lösung kann sowohl lokal als auch als Online-Service betrieben werden und bietet plattformübergreifende Browser-Unterstützung sowie Integration für mobile Geräte über einen universellen Cookie-Privacy-Proxy.

Bezug zum Themenfeld Sicherheit: Das Projekt adressiert direkt die digitale Selbstbestimmung und den Datenschutz der Nutzer*innen. Durch die zentrale Einwilligungsverwaltung wird die Kontrolle über personenbezogene Daten an die Nutzer*innen zurückgegeben. Das Webinterface ist mit 2FA gesichert und das Sicherheitskonzept ist Voraussetzung für die offizielle Anerkennung durch den Bundesbeauftragten für Datenschutz.

Gesellschaftliche Relevanz: OpenPIMS 2.0 gibt Nutzer*innen die Kontrolle über ihre Daten zurück, reduziert Barrieren (insbesondere für Menschen mit körperlichen Einschränkungen) und ermöglicht eine transparente, datenschutzkonforme Interaktion mit digitalen Diensten. Durch die Open-Source-Entwicklung wird eine nachhaltige und überprüfbare Grundlage geschaffen, die für alle zugänglich ist.

Ausführliche Darstellung der Ergebnisse in der Förderphase

Erzielte Ergebnisse:

- TDDDG-Anpassungen: Die Referenzimplementierung wurde vollständig an die TDDDG-Anforderungen angepasst. Kernstück ist ein 3-stufiges Einwilligungssystem: Nutzer können Cookies wahlweise nach Kategorien (z.B. "alle Analytics-Cookies"), nach Anbietern (z.B. "Matomo ja, Google Analytics nein") oder einzeln verwalten. Technisch notwendige Cookies werden gemäß § 25 Abs. 2 TDDDG automatisch erkannt und erfordern keine Einwilligung.

- Browser-Erweiterungen: Alle geplanten Erweiterungen wurden erfolgreich umgesetzt. Die Extensions für Chrome, Firefox, Safari, Edge, Brave und Opera basieren auf einer gemeinsamen Codebasis und kommunizieren DSGVO-konform über den User-Agent mit teilnehmenden Websites.

- Cookie-Privacy-Proxy: Statt der ursprünglich geplanten Node.js-Lösung wurde ein Python-basierter mitmproxy-Addon entwickelt, der funktional gleichwertig ist und sich

besser in bestehende Infrastrukturen integriert. Zusätzlich entstanden Integrationen für Cloudflare Worker und WordPress.

- Sicherheitskonzept: Das Sicherheitskonzept basiert auf einem deterministischen Token-System mit täglicher Rotation. Tokens werden nicht gespeichert, sondern bei Bedarf berechnet. Die passwortlose Anmeldung per Magic Link eliminiert Passwort-bezogene Risiken. Die DSGVO-Compliance umfasst Auskunfts-, Lösch- und Exportfunktionen.

Meilenstein-Erreichung:

Erreicht: TDDDG-konforme Referenzimplementierung, alle Browser-Erweiterungen, Sicherheitskonzept, DSGVO-Compliance, Cookie-Definition-Standard

Angepasst: Proxy-Implementierung in Python statt Node.js (funktional gleichwertig)

Offen: E-Mail-Änderungsfunktion, vollständige Consent-Historie

Erkenntnisse aus der Projektarbeit:

Die größte Herausforderung für PIMS-Systeme ist das Henne-Ei-Problem: Ohne teilnehmende Websites fehlt der Nutzen für Anwender, ohne Nutzer fehlt der Anreiz für Websites. Der Cookie-Consent-Markt ist durch etablierte Anbieter dominiert, was die Adoption erschwert.

Die OKF-Begleitung half bei der Priorisierung zwischen technischer Perfektion und praktischer Nutzbarkeit. Der Austausch mit anderen Prototype Fund Projekten bot wertvolle Perspektiven für die Weiterentwicklung.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Primäre Zielgruppe: Unternehmen und Institutionen (Vereine und nicht kommerzielle Organisationen), die ein PIMS-System zur Einwilligungsverwaltung betreiben möchten, wie im TDDDG gefordert.

Sekundäre Zielgruppe: Entwickler*innen und Betreiber*innen von Websites sowie mobilen Diensten, die datenschutzkonforme Lösungen umsetzen wollen.

Nutzen für die Zielgruppe:

- Zentrale Verwaltung von Nutzereinwilligungen ohne störende Cookie-Banner

- Plattformübergreifende Unterstützung (Desktop und Mobile)

- Rechtskonforme Umsetzung der TDDDG-Anforderungen

- Barrierefreie Alternative für Menschen mit Einschränkungen

Effekte der Open-Source-Stellung: Durch die Open-Source-Entwicklung wird Transparenz und langfristige Nachhaltigkeit geschaffen. Die Lösung ist für alle zugänglich und kann von der Community weiterentwickelt werden. Dies fehlt proprietären Alternativen wie OneTrust oder Cookiebot.

Pläne zur Weiterentwicklung:

- Gründung einer Betreiberorganisation für OpenPIMS als Referenzplattform

- Anerkennungsverfahren beim Bundesbeauftragten für Datenschutz und Informationsfreiheit

- Integration in nicht-kommerzielle Webseiten (Bildungs- und gemeinnützige Organisationen)

- Community-Ausbau durch Dokumentation, Tutorials und Konferenzteilnahme (z.B. FOSDEM)

Persönliche/fachliche Weiterentwicklung:

Die Projektarbeit hat meine Expertise im Spannungsfeld zwischen Datenschutzrecht und technischer Umsetzung erheblich vertieft. Die intensive Auseinandersetzung mit TDDDG, DSGVO und ePrivacy-Verordnung schärfte mein Verständnis für die praktischen Herausforderungen bei der Umsetzung regulatorischer Anforderungen in Software.

Technisch erweiterte das Projekt meine Kenntnisse in der plattformübergreifenden Browser-Extension-Entwicklung, insbesondere bezüglich der Unterschiede zwischen Manifest V2 und V3 sowie den jeweiligen Browser-APIs. Die Entwicklung eines deterministischen Token-Systems ohne serverseitige Speicherung vertiefte mein Wissen über Privacy-by-Design-Prinzipien.

Der Austausch mit der Open-Source-Community und anderen Prototype Fund Projekten förderte meine Fähigkeiten in der Dokumentation und Kommunikation technischer Konzepte für unterschiedliche Zielgruppen. Die Begleitung durch die OKF schulte zudem meinen Blick für die Balance zwischen technischer Vollständigkeit und praktischem Nutzen – eine wertvolle Erkenntnis für zukünftige Projekte.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

X-OpenPIMS Header statt User-Agent-Signalisierung:
Ursprünglich kommunizierten die Browser-Erweiterungen über einen eigenen HTTP-Header (X-OpenPIMS) mit Websites. Dieser Ansatz wurde verworfen, da Custom Headers auf Drittanbieter-Websites datenschutzrechtlich problematisch sind und von manchen Browsern zunehmend eingeschränkt werden. Die Lösung war der Wechsel zur User-Agent-Signalisierung, die DSGVO-konform ist und keine zusätzlichen Tracking-Vektoren schafft.

Separate Browser-Extension-Repositories:
Anfangs wurden für Chrome, Firefox und Safari separate Codebases gepflegt. Dies führte zu Inkonsistenzen und hohem Wartungsaufwand. Nach mehreren Synchronisationsproblemen wurde auf eine unified Codebase mit dem WXT-Framework migriert, die alle Browser aus einer Quelle bedient.

Node.js Cookie-Privacy-Proxy:
Die geplante Node.js-Implementierung des Proxys erwies sich als weniger praktikabel

für den Einsatz in bestehenden Infrastrukturen. Stattdessen wurde ein mitmproxy-Addon in Python entwickelt, das sich nahtlos in verbreitete Proxy-Setups integriert.

Passwort-basierte Authentifizierung:
Das ursprüngliche Login-System mit Passwörtern wurde zugunsten von Magic Links aufgegeben. Passwörter erhöhten die Komplexität (Zurücksetzen, sichere Speicherung) ohne Mehrwert für ein System, das ohnehin eine verifizierte E-Mail-Adresse voraussetzt.

Serverseitige Token-Speicherung:
Ein früher Entwurf sah die Speicherung von API-Tokens in der Datenbank vor. Dies wurde durch ein deterministisches System ersetzt, bei dem Tokens bei Bedarf berechnet werden – weniger Angriffsfläche, keine Token-Datenbank, automatische tägliche Rotation.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer*innen

GitHub Repository: https://github.com/openpims

Entwickler-Profil: https://github.com/stefanboeck

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Die geplanten 640 Stunden wurden im Wesentlichen eingehalten, wobei es zu Verschiebungen zwischen einzelnen Arbeitspaketen kam.

Mehraufwand:

- Die Migration auf Manifest V3 für Chromium-Browser erforderte mehr Zeit als geplant, da die declarativeNetRequest API grundlegend anders funktioniert als die bisherige webRequest API

- Die Entwicklung der unified Extension-Codebasis war initial aufwändiger, reduzierte aber den laufenden Wartungsaufwand erheblich

- Die DSGVO-Compliance-Implementierung (Auskunfts-, Lösch- und Exportfunktionen) wurde umfangreicher als ursprünglich vorgesehen

Minderaufwand:

- Der Wechsel von Node.js zu mitmproxy für den Proxy verkürzte die Entwicklungszeit, da auf eine etablierte Basis aufgebaut werden konnte

- Die Magic Link Authentifizierung war schneller umgesetzt als ein vollständiges Passwort-System mit Zurücksetzen-Funktion

Planungsanpassungen:

- Die Safari-Extension für iOS wurde später im Projekt priorisiert als geplant, da Apple die Unterstützung für Web Extensions auf iOS während des Förderzeitraums verbesserte

- Zeit für die E-Mail-Änderungsfunktion und vollständige Consent-Historie wurde zugunsten der Kernfunktionalität zurückgestellt

Insgesamt blieb das Projekt im geplanten Rahmen, wobei Flexibilität bei der Priorisierung einzelner Features notwendig war.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Relevante externe Entwicklungen:

Bestehende Consent-Management-Plattformen (CMPs) wie OneTrust oder Cookiebot bieten Cookie-Banner und zentralisierte Einwilligungsverwaltung, sind jedoch proprietärer, teuer und auf Webbrowser beschränkt. Open-Source-Alternativen wie Klaro! fokussieren ebenfalls auf Banner-Lösungen, bieten jedoch keine nahtlose Integration für mobile Geräte.

Differenzierung von OpenPIMS 2.0: OpenPIMS 2.0 geht über diese Ansätze hinaus, indem es Cookie-Banner vollständig ersetzt und durch eine zentrale Infrastruktur fuer Einwilligungsverwaltung ergänzt. Der Cookie-Privacy-Proxy ermöglicht die Nutzung auf mobilen Endgeräten und stellt eine plattformunabhängige Lösung dar.

Weitere externe Einflüsse:

- Google Manifest V3: Die erzwungene Migration für Chrome-Erweiterungen erforderte grundlegende Architekturanpassungen und beschleunigte den Wechsel zur User-Agent-Signalisierung.

- Apple iOS Web Extensions: Verbesserte Safari-Unterstützung auf iOS während des Förderzeitraums ermöglichte die Entwicklung einer mobilen Extension.