Richtlinie zum „Software-Sprint“

robur – sichere digitale Infrastruktur zum selbst Betreiben

Schlussbericht

Zuwendungsempfänger:

Stefanie Schirmer , Hannes Mehnert

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS18S01 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Das Projekt ist motiviert durch den Gedanken, dass viele private Daten momentan zentral bei Google, Facebook, etc. gespeichert sind. Wir wollen es Nicht-ExpertInnen ermöglichen, ihre digitale Infrastruktur selbst zu betreiben. Als ersten Prototyp haben wir einen robusten Kalenderservice (CalDAV) implementiert, der mit bestehenden Clients interoperiert.

Wir hatten folgende Meilensteine geplant:

- Nach 2 Monaten wollen wir einen ersten Prototyp mit minimalen Featureset (WebDAV RFC4918 und CalDAV RFC4791) implementiert haben, der mit Android, Thunderbird und Apple iOS interoperiert.

- In Monaten 2-4 wollen wir gefundene Interoperability Bugs fixen, einen Proxy-Modus implementieren (um andere CalDAV server zu pollen und Einträge zu pushen).

- In den verbleibenden Monaten 4-6 werden wir weiter Bugs fixen, aber auch ein verschlüsseltes Backup via git implementieren und weitere scheduling Features (RFC 6638) implementieren.

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

Die Zielgruppe sind Kalendernutzer der beliebtesten Clients (Thunderbird, Calendar.app, DAVDroid) und des CalDAV-Protokolls. Nutzer aller Betriebssysteme haben die Moeglichkeit mit dem jeweiligen bestehenden Client unseren Server zu nutzen. Der Server kann als MirageOS Unikernel auf den

Plattformen Unix, Linux KVM, Xen, kommerzlelle Cloud-Anbieter, Raspberry Pi3 selbst betrieben werden. Der Kalenderserver leistet einen wichtigen Beitrag zu den Themenfeldern Sicherheit und Infrastruktur, da er unabhaengige und robuste Infrastruktur fuer alle verfuegbar macht.

Ausführliche Darstellung der Ergebnisse

Im Projektverlauf haben wir folgende Meilensteine erreicht:

- Nach 3 Monaten hatten wir einen ersten Prototyp mit minimalen Featureset (WebDAV RFC4918 und CalDAV RFC4791) implementiert, den wir anhand der von Apple bereitgestellten Testsuite (https://github.com/apple/ccs-caldavtester) validiert haben.

- In den Monaten 4-5 haben wir an der CalDAV-Funktionalitaet fuer wiederholende Datumsberechnungen und Reports gearbeitet, welche sich als komplex herausgestellt haben, und das implementieren einer Datumsbibliothek erforderten.

- In dem verbleibenden 6. Monat haben wir an der Interoperabilitaet mit verschiedenen CalDAV-Clients der grossen Betriebssysteme gearbeitet:

    - Thunderbird Lightning fuer Windows und Linux

    - Calendar.app fuer macOS und iOS

    - DAVDroid fuer Android

    - FirefoxOS Calendar

    - Blackberry Calendar

Ausserdem haben wir Access Control Listen implementiert, um verschiedenen Nutzern verschiedene Kalender zugaenglich zu machen.

Als letzten Arbeitsschritt haben wir unseren Kalenderserver mit einem konfigurierbaren Storage-Baclend ausgestattet. Er kann entweder ein Dateisystem oder ein git-Repository verwenden.

Wir haben unseren Prototyp auf einer Veranstaltung mit 30 TeilnehmerInnen in der Praxis erprobt und angepasst.

Ohne die Foerderung haetten wir das Projekt nicht finanziert bekommen. Die woechentlichen Standup-Meetings im Chat waren eine gute Moeglichkeit, um mit den anderen Gruppen in Kontakt zu bleiben. Wir waren auch mehrfach beim OKF-Mittagessen und haben uns ausgetauscht. Das Coaching war fuer unser Projekt nur bedingt relevant, da wir kein User-Interface haben – wir haben einen Server geschrieben und das CalDAV-Protokoll implementiert. Wir koennen uns aber gut vorstellen, dass fuer UX-Entwicklung das Coaching hilfreich ist.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Wir haben eine Liste von Interessenten, die unseren CalDAV-Server beretis testen und gespannt darauf warten, ihn in der Praxis zu betreiben. Diese Gruppe setzt sich aus AktivistInnen, Security-ExpertInnen, und Free-Software-Enthusiasten zusammen.

Es sind verschiedene Bibliotheken entstanden, die in MirageOS benutzt werden koennen. Die Arbeit hat sich auch darauf ausgewirkt, wie bestehende MirageOS-Bibliotheken funktionieren. Der Storage-Stack wurde komplett ueberholt, wir haben Bugs in Webmachine (HTTP-Server, https://github.com/inhabitedtype/ocaml-webmachine) gefunden , gemeldet und repariert. Auch in der CalDAV-Spezifikation (https://tools.ietf.org/html/rfc4791) haben wir einen Fehler gemeldet.

Wir haben bereits angefangen, auch CardDAV (zum Verwalten eines Adressbuchs) zu implementieren.

Wir arbeiten auch noch daran, die Installation noch einfacher fuer den Laien zu gestalten (Erklaerungsvideo, Ein-Klick-Installation).

Hannes hat primaer Hintergrund in der Protokoll-Entwicklung, und hat in dem Kalender-Server-Projekt viel ueber Anwendungsentwicklung gelernt. Stefanie hat einen Hintergrund im Betreiben grosser HTTP-Services, und im Projekt zum ersten Mal einen Unikernel programmiert.

Unsere Arbeitsweise, die Pair-Programmierung mit einer Person, die tippt, und einer, die entscheidet, hat dazu gefuehrt, dass Konzepte von beiden Programmierenden geteilt werden. Design-Entscheidung haben wir gemeinsam getroffen. Die entstandene Software ist robust, und Kompromisse zwischen Sicherheit und Praktikabilitaet wurden gut begruendet.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Wir haben uns einige Tage mit der Generierung von JavaScript aus dem Servercode beschaeftigt, um ein User-Interface zu entwickeln. Dies stellte sich als schwierig heraus, da die Bibliothek fuer einen Nutzerflow in JavaScript kaputt war. Wir haben den Fehler gemeldet, und derweil am Server weitergebaut. Mittlerweile ist der Fehler behoben, aber dier Foerderzeitraum vorbei.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Die Webseite zum Projekt ist http://robur.io/Projects/CalDAV – dort findet sich ein Installationsanleitung und ein Test-server.

Die entwickelten Software-Bibliotheken sind frei verfuegbar: https://github.com/roburio/icalendar, https://github.com/roburio/mirage-fs-mem, https://github.com/roburio/caldav

Unsere Bibliotheken koennen weiter verwendet werden, um CardDAV zu implementieren, wiederkeherende Daten zu berechnen, und einen WebDAV-Server zum Datei-Sharing zu verwirklichen.

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Es gab einen Mehraufwand: die Datumsberechnungen waren komplexer als erwartet, und mussten mehrfach durchdacht werden; bestehende Clients nehmen das CalDAV-Protokoll nicht so genau, wir mussten Workarounds in unseren Server implementieren, was viel Zeit erfordert hat. Wir haben die Foerdersumme ueberstiegen, da wir mehr Stunden gearbeitet haben als geplant.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Fuer das Storage-Backend hatten wir ein append-only Datenmodell auf Basis von git geplant, das von externen Entwicklern fuer den Sommer zugesagt wurde. Diese Entwicklung hat sich bis Oktober verzoegert. Wir konnten die git-basierte Loesung daher erst spaeter einbauen. Der positive Aspekt an dieser Verzoegerung ist, dass der Storage-Stack zu diesem Zeitpunkt komplett von allen Beteiligten ueberarbeitet wurde, da unsere Anforderungen bekannt waren, und in das Design mit einfliessen konnten.