

Symcloud: Filesync and Collaboration platform

Wachter Johannes

Contents

| | | |
|----------|--|----------|
| 1 | Einleitung | 1 |
| 1.1 | Projektbeschreibung | 2 |
| 1.2 | Überblick | 2 |
| 2 | Mögliche Themen (nur Notizen) | 2 |
| 2.1 | P2P File-Sharing und Groupware | 4 |
| 2.2 | Datenhaltung | 4 |

1 Einleitung

In Zeiten der NSA und anderen Geheimdiensten ist es immer mehr Menschen wichtig die Kontrolle über die eigenen Daten zu behalten. Aufgrund dessen wecken Projekte wie Diaspora¹, ownCloud² und ähnliche Software Lösungen immer mehr Aufmerksamkeit. Die beiden genannten Software Lösungen decken zwei sehr wichtige Bereiche der Datenkontrolle ab.

Das eine “Diaspora” ist ein dezentrales soziales Netzwerk ohne der Nutzer von einem Betreiber abhängig ist. Es bietet also die Möglichkeit seinen Freunden bzw. Familie eine private Plattform anzubieten. Das interessante daran ist, das sich sogenannte Pods (dezentrale Knoten) beliebig verknüpfen lassen und damit ein P2P Netzwerk einrichten lassen. Der Betreiber jedes Pods bestimmt damit wer in sein Netzwerk eintreten und damit wer seine Daten sehen kann. Die verbundenen Pods tauschen ohne einen Zentralen Knoten Daten aus und sind dadurch unabhängig. Dies garantiert die volle Kontrolle über seine Daten im Netzwerk.

¹<https://diasporafoundation.org/>

²<https://owncloud.org/>

Das Projekt “ownCloud” ist eine Software, die es ermöglicht Daten in einer Privaten Cloud zu verwalten. Mittels Endgeräte-Clients können die Daten synchronisiert und über die Plattform auch geteilt werden. Insgesamt bietet die Software einen ähnlichen Funktionsumfang gängiger Kommerzieller Lösungen an. Zusätzlich bietet es allerdings wie z.B. Google Drive einen online Editor, über den mehrere Nutzer gleichzeitig ein Dokument bearbeiten können. Diese Technologie basiert auf der JavaScript Library WebODF³.

1.1 Projektbeschreibung

Symcloud ist eine private Cloud-Software, die es ermöglicht über dezentrale Knoten (ähnlich wie Diaspora) Daten über die Grenzen der eigenen Cloud hinweg zu teilen. Verbundene Knoten tauschen über sichere Kanäle Daten aus, die dann über einen GIT-Client mit dem Endgerät synchronisiert werden können.

Die Software baut auf modernen Technologien auf und verwendet als Basis das Framework Symfony2⁴. Dieses Framework ist eines der beliebtesten PHP-Frameworks und bietet neben der Abstraktion von HTTP-Anfragen auch einen DI-Container, Event-Dispatcher, Routing uvm. Dies erleichtert die Entwicklung von großen PHP-Projekten.

Als Basis für die Plattform verwendet Symcloud das Content-Management-Framework der Vorarlberger Firma MASSIVE ART WebServices aus Dornbirn. Es bietet ein erweiterbares Admin-UI, eine Benutzerverwaltung und ein Rechtssystem. Diese Features ermöglichen Symcloud eine schnelle Entwicklung der Oberfläche und deren zugrundeliegenden Services.

1.2 Überblick

Die Abbildung 1 beschreibt die Abhängigkeiten auf denen Symcloud aufsetzt.

TODO diagram erweitern

2 Mögliche Themen (nur Notizen)

Es bieten sich an dieser Stelle 2 große Themen (mit meiner Meinung nach Thesis Relevanz). Je nach Schwerpunkt der Schriftlichen Arbeit würde ich den jeweils anderen Teil hinten anstellen.

In beiden Varianten würde ein File-Abstraction-Layer entstehen, der den Datei Zugriff abstrahieren würde und damit den Storage der Daten verbirgt. Ebenfalls in beiden Fällen würde ich standardmässig auf GIT setzen, da ich es spannend

³<http://webodf.org/>

⁴<http://symfony.com/>

| Symcloud Web Platform | |
|----------------------------------|-----------------------------------|
| Sulu CMF | File Abstraction Layer |
| Symfony2 | GIT |

Figure 1: Überblick über die Komponenten

finde diese Technologie mal anders zu verwenden. Durch diese Abstraktion sollte es kein Problem sein später einen anderen Storage zu verwenden (bsp.: Dateisystem und Webdav), falls sich herausstellt, dass git ungeeignet für das Projekt ist.

Der große Vorteil, den ich an GIT sehe und warum ich daran festhalten will ist:

- Funktionierende komprimierung
- Automatische Versionierung
- Volle Kontrolle der Daten, da unabhängig von der Software
- HTTP-Schnittstelle
- Erweiterbar und daher gut geeignet um eigene Scripts einzubinden

Nachteile, die allerdings entstehen könnten:

- Große History durch unendliche Versionen
- Schwierigeres Teilen eines Teils der Daten
- Verschlüsselung Client seitig, daher keinen Zugriff auf die Daten am Server (wenn aktiviert)

2.1 P2P File-Sharing und Groupware

Ein spannendes Thema wäre eine durchleuchtung von Diaspora und die ummünzung auf Symcloud. Wie in der Einleitung schon erläutert, ermöglicht Diaspora die vernetzung von Knoten untereinander. Dies wäre auch bei einer File-Sync Plattform sinnvoll.

Denkt man zum Beispiel an eine Firma mit mehreren Standorten, einige der Daten werden von beiden Standorten genutzt, einige jeweils nur von einem. Teile dieser Daten werden aber unter den Nutzern beider Standorte geteilt. Da wäre es doch sinnvoll, wenn die Nutzerdaten der Server synchronisiert und damit es ermöglichen jeweils den nächsten Server zu verwenden um seine Daten zu verwalten. Ausserdem könnten gewisse Daten, die in beiden Standorten verwendet werden synchronisiert und damit ermöglichen immer schnell an seine Daten zu gelangen, auch wenn man zu Gast beim anderen Standort ist. Dazu könnte dann das LAN verwendet werden.

GIT würde auch hier mit seinem dezentralen System helfen diese Möglichkeiten voll auszuschöpfen. Für die verteilung der Nutzerdaten könnte ein sicherer Tunnel aufgebaut werden und die Daten dadurch sicher von A nach B Transportiert werden.

Interessante Themen wäre hier:

- Der sichere Austausch von Daten
- Die Anmeldung an einem fremden (verbundenen) Server der mein Passwort nicht wissen sollte (oAuth könnte eine Lösung sein)
- Damit verbunden könnte der Fokus mehr auf die Plattform gelegt werden und dort eine Art Sozial Media, Colaboration aufgebaut werden.

Es wäre sicher eine Spannende Sache und damit verbunden eine Herausforderung in der Implementierung.

2.2 Datenhaltung

GIT als Backup-System. Dazu würde es eine GIT-Implementierung in reinem PHP benötigen, bei dem es möglich den Zugriff Storage der Daten in eine Datenbank umzuleiten, die History und damit die Commits zu manipulieren und der Web-Oberfläche den Datei-Tree und die Dateien zur verfügung zu stellen. Dazu habe ich mich intensiv mit dem inneren von GIT beschäftigt und mich mit den Entwicklern verständigt. Diese Implementierung, wäre sicher auch für die PHP-Community interessant.

Die Arbeit würde sich dann um die GIT-Internals und die Verwendung von GIT bei Dateien die nicht nur Code enthalten. Ausserdem könnte evaluiert werden, ob GIT für diesen Zweck überhaupt geeignet ist.

Zusätzliche Themen wären auch die Sicherheit der Daten (Verschlüsselung, Backup, ...) und die Teilbarkeit von Repositories also die Dynamische erzeugung von Repositories aus verschiedenen Teilen anderer.

Zum letzten Thema gibt es kaum ansätze, wäre also für eine Masterarbeit geeignet.

Interessant Links:

- <https://tent.io/>: protocol for personal data and communications ... Tent is an open protocol for personal evented data vaults and decentralized real-time communication. Tent has two APIs: one lets Tent apps talk to Tent servers, the other lets Tent servers talk to each other. A Tent server can have one or many users, and anyone can run their own Tent server. (https://wiki.diasporafoundation.org/Diaspora_powered_by_Tent)
- <http://xanadu.com/>: ???
- <https://github.com/depot/depot>: PHP-Library for tent
- <https://github.com/Cacauu/librejo>: -||-
- <http://florianjacob.de/tentio-new-hope.html>: Blog Post about tent.io
- <https://github.com/tent/tent.io/wiki/Related-projects>
- https://wiki.diasporafoundation.org/Diaspora_powered_by_Tent#Missing_Tent_Features: Diaspora powered by Tent