Symcloud: Filesync and Collaboration platform

Wachter Johannes

Inhaltsverzeichnis

1 Einleitung

Seit den Abhörskandalen durch die NSA und anderen Geheimdiensten ist es immer mehr Menschen wichtig, die Kontrolle über die eigenen Daten zu behalten. Aufgrund dessen erregen Projekte wie Diaspora[1], ownCloud[2] und ähnliche Software Lösungen immer mehr Aufmerksamkeit. Die beiden genannten Software Lösungen decken zwei sehr wichtige Bereiche der persönlichen Datenkontrolle ab.

Diaspora ist ein dezentrales soziales Netzwerk. Die Benutzer von diesem Netzwerk sind durch die verteilte Infrastruktur nicht von einem Betreiber abhängig. Es bietet die Mög-lichkeit, seinen Freunden bzw. der Familie, eine private Plattform anzubieten. Das Interes-sante daran ist, dass sich sogenannten Pods (dezentrale Knoten), beliebig untereinander vernetzen lassen und damit ein P2P Netzwerk aufbauen lässt. Pods können von jedem in-stalliert und betrieben werden; dabei kann der Betreiber bestimmen, wer in sein Netzwerk eintreten darf und welche Server mit seinem verbunden sind. Die verbundenen Pods tau-schen ohne einen zentralen Knoten Daten aus und sind dadurch unabhängig. Dies garan-tiert die volle Kontrolle über seine Daten im Netzwerk (siehe „Was ist Dezentralisierung¿‘ 2015).

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 Funktion-sumfang gängiger kommerzieller Lösungen an (siehe „Owncloud Features“ 2015). Zusätzlich bietet es eine Kollaborationsplattform, mit der zum Beispiel Dokumente über einen online Editor, von mehreren Benutzern gleichzeitig, bearbeitet werden können. Diese Technologie basiert auf der JavaScript Library WebODF[3].

1.1 Projektbeschreibung

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

TODO genauere Beschreibung

Die Software baut auf modernen Web-Technologien auf und verwendet als Basis das PHP-Framework Symfony2[4]. Dieses Framework ist eines der beliebtesten in der Open-Source Community. Es bietet neben der Abstraktion von HTTP-Anfragen auch einen Dependency-Injection-Container und viele weitere

[1] https://diasporafoundation.org/
[2] https://owncloud.org/
[3] http://webodf.org/
[4] http://symfony.com/

Komponenten wie zum Beispiel Routing und Event Dispatcher. Zusätzlich erleichtert es die Entwicklung von großen PHP-Projekten, durch die Möglichkeit den Code in Komponenten, sogenannten Bundles, zu gliedern. Diese können dann mit der Community geteilt werden.

Als Basis für die Plattform verwendet Symcloud das Content-Management-Framework SULU[5] der Vorarlberger Firma MASSIVE ART WebServices[6] aus Dornbirn. Es bietet ein erweiterbares Admin-UI, eine Benutzerverwaltung und ein Rechtesystem. Diese Features ermöglichen Symcloud eine schnelle Entwicklung der Oberfläche und deren zugrundeliegenden Services.

1.2 Inspiration

TODO Noch einmal ownCloud - Diaspora und Ted Nelson mit dem Xanadu Projekt

1.3 Technologie

Dieses Kapitel beschreibt die verwendeten Technologie etwas genauer. In Abbildung 1 zeigt die Abhängigkeiten als Schichten Diagramm. Ganz oben ist zum einen die Oberfläche von Symcloud, die in die Sulu Umgebung eingebettet ist. Sulu selbst ist eine "One-Page application", die verschiedene Javascript Komponenten zur Verfügung stellt, um die Anwendung erweitern zu können. Die andere Schnittstelle ganz oben ist der Synchronisierung Client, der es ermöglich Daten über ein Kommandozeilen Programm zu synchronisieren. Beide "Oberflächen" sprechen das Backend über eine gesicherte REST-API an. Diese API wird verwendet um Daten zu empfangen aber auch daten an den Server zu senden. Zum Beispiel sendet der Synchronisierungs Client einen POST-Request an den Server um eine Datei hochzuladen.

Auf der Server-Seite gibt es zum einen die standard API-Schnittstellen von Sulu und zum anderen die Erweiterung durch Symcloud mit der File-API. Als Persistence-Schicht für die Metadaten, der Dateien in der Symcloud, wird die Abstraktionsschicht PHPCR verwendet. Diese Schnittstelle bietet einen Zugriff auf verschiedenste Content-Repositories an.

[5] http://www.sulu.io
[6] http://www.massiveart.com/de

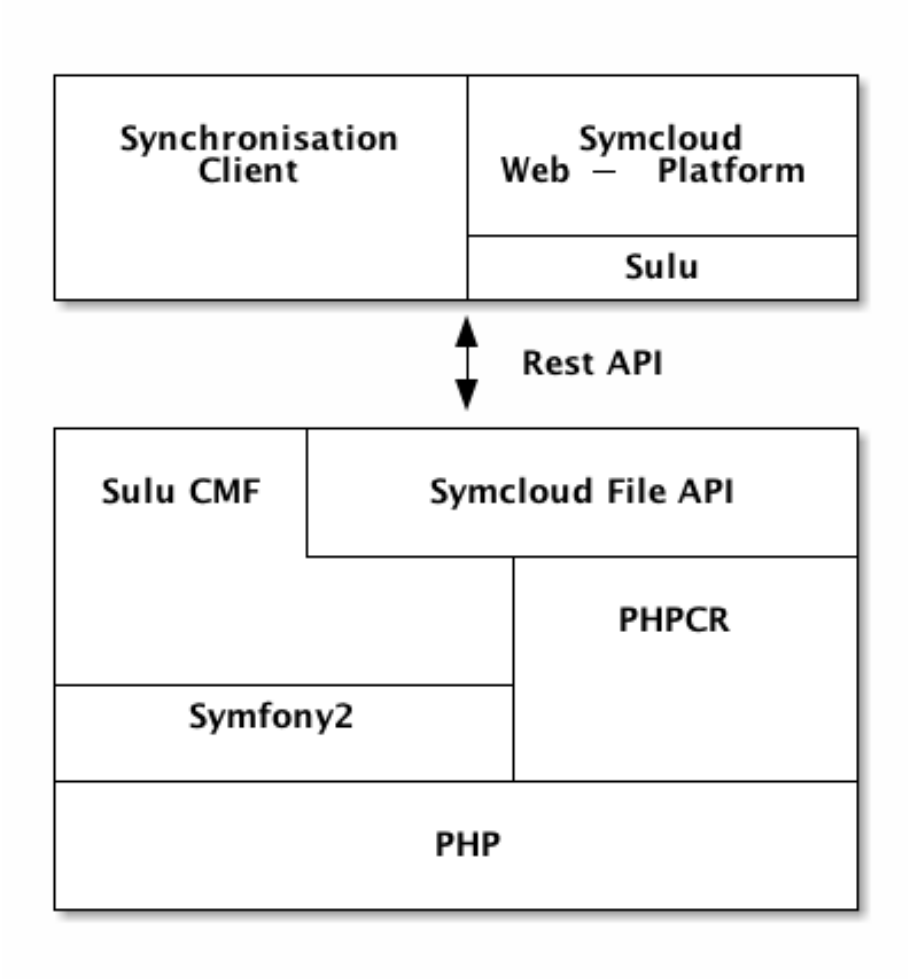

Abbildung 1: Überblick über die Komponenten

1.3.1 Sulu CMF

1.3.2 Symfony2

1.3.3 PHPCR

1.3.4 PHP

1.4 Anforderungen

2 Stand der Technik

In diesem Kapitel werden moderne Anwendungen anhand ihrer Architektur durchläuchtet, um die jeweilige Vorteile für Symcloud zusammenzufassen.

2.1 Verteilte Systeme

Ein verteiltes System ist laut Andrew Tanenbaum (siehe Tanenbaum ; Steen 2008, pp. ???):

```
"Ein verteiltes System ist eine Menge voneinander unabhängiger
Computer, die dem Benutzer wie ein einzelnes kohärentes
System erscheinen"
```

Ein nicht mehr ganz neues aber immer noch sehr aktuelles Verteiltes System ist das Netzwerk-Protokoll NFS (Network File Service).

TODO überprüfen ob dieses zitat echt ist TODO beschreibung eines netzwerk protokolles TODO trennung metadaten und inhalt

2.2 Dropbox

Dropbox-Nutzer können jederzeit von ihrem Desktop aus über das Internet, über mobile Geräte oder über mit Dropbox verbundene Anwendungen auf Dateien und Ordner zugreifen. Alle diese Clients stellen Verbindungen mit sicheren Servern her, über die Sie Zugriff auf Dateien haben, Dateien für andere Nutzer freigeben können, und verknüpfte Geräte aktualisieren können, wenn Dateien hinzugefügt, verändert oder gelöscht werden. Der Dropbox-Service betreibt verschiedene Dienste, die sowohl für die Handhabung und Verarbeitung von Metadaten als auch von unformatiertem Blockspeicher verantwortlich sind. (siehe „Wie funktioniert der Dropbox-Service¿‘ 2015)

In der Abbildung 2 werden die einzelnen Komponenten in einem Blockdiagram dargestellt. Es gliedert sich in drei größere Blöcke:

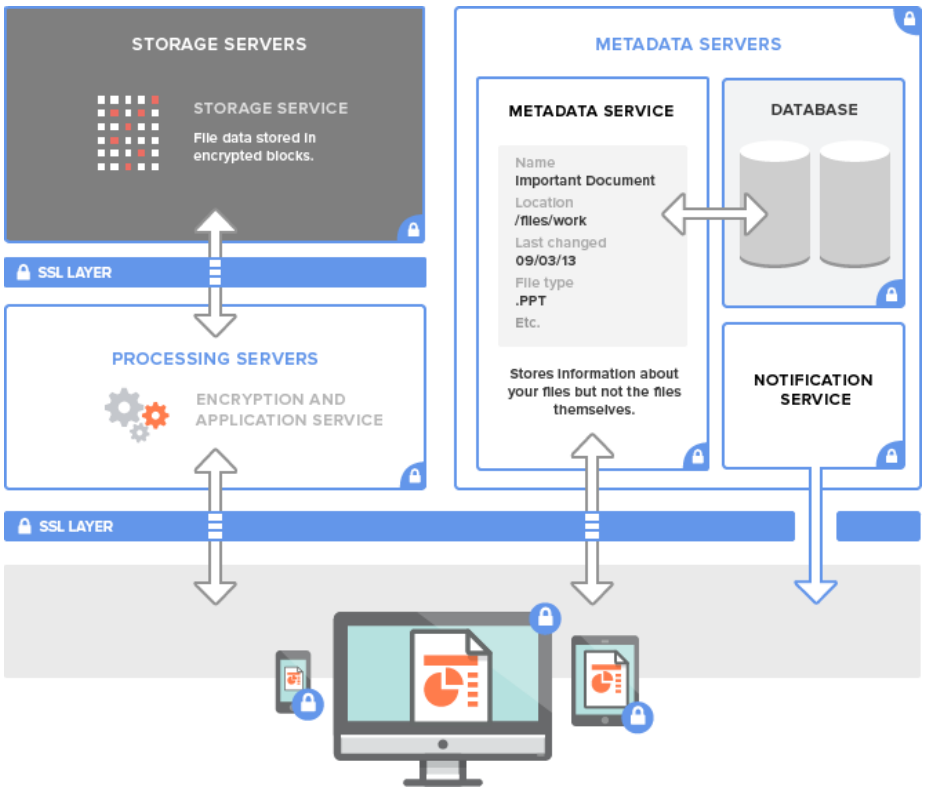

Abbildung 2: Blockdiagram der Dropbox Services (Quelle https://www.dropbox.com/help/1968)

- Metadata Servers
- Storage Servers
- Processing Servers

Wie im Kapitel 2.1 beschrieben trennt Dropbox intern die Dateien von ihren Metadaten. Der Metadata Service speichert die Metadaten und informationen zu ihrem Speicherort in einer Datenbank aber der Inhalt der Daten liegt in einem seperaten Storage Service. Dieser Service verteilt die Daten wie ein "Load Balancer" über viele Server.

Der Storage Service ist wiederum von aussen durch einen Application Service abgesichert. Die Authentifizierung erfolgt über das OAuth2 Protokoll (siehe „Core API Dokumentation" 2015). Diese Authentifizierung wird für alle Services verwendet, also auch für den Metadata Service und der Notification Service.

2.3 ownCloud

2.4 Diaspora

2.5 Zusammenfassung

3 Implementierung

4 Diskussion

Anhang

„Core API Dokumentation" (2015): Core API Dokumentation. . Online im Internet: https://www.dropbox.com/developers/core/docs (Zugriff am: 26.03.2015).

„Owncloud Features" (2015): Owncloud Features. . Online im Internet: https://owncloud.org/features (Zugriff am: 05.03.2015).

Tanenbaum, A.S. ; Steen, M. van (2008): Verteilte Systeme: Prinzipien und Paradigmen. Pearson Studium. (= Pearson Studium - IT). Online im Internet: http://books.google.de/books?id=V6I6PQAACAAJ

„Was ist Dezentralisierung¿' (2015): Was ist Dezentralisierung? . Online im Internet: https://diasporafoundation.org/about (Zugriff am: 05.03.2015).

„Wie funktioniert der Dropbox-Service¿' (2015): Wie funktioniert der Dropbox-Service? . Online im Internet: https://www.dropbox.com/help/1968 (Zugriff am: 26.03.2015).