

Origo - social client

Dokumentation

Mario Volke

Origo - social client: Dokumentation

Mario Volke

Copyright © 2008 Mario Volke, Alle Rechte vorbehalten.

Inhaltsverzeichnis

Vorwort	v
1. Motivation & Vision	1
Motivation	1
Vision	1
2. Konzept	3

Vorwort



Diese Dokumentation entstand parallel zur Konzeptions- und Entwicklungsphase. Ziel dieses Dokuments ist es, dem Leser zum einen aufzuzeigen, welche Problemstellungen mit *Origo* gelöst werden können und zum anderen werden die Konzepte und Module von *Origo* vorgestellt, um die angesprochenen Probleme zu lösen.

Kapitel 1. Motivation & Vision

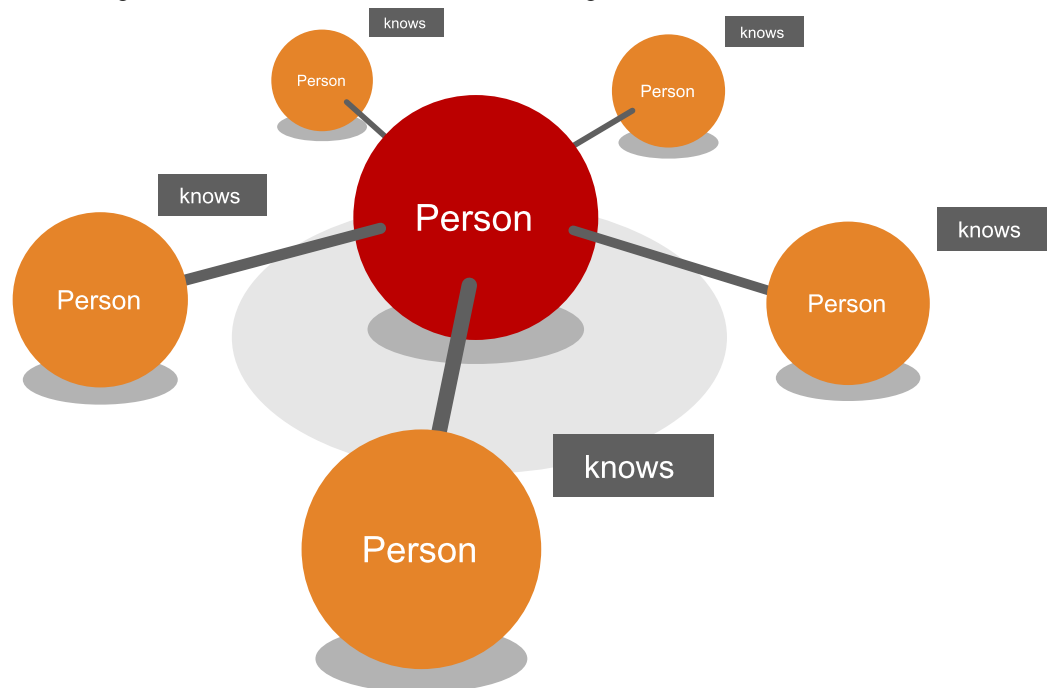
Motivation

Seit den Anfängen des World Wide Web, über den Boom von Web 2.0, bis heute sind unzählige soziale Netzwerke mit unterschiedlich starken Nutzerzahlen entstanden und auch wieder verschwunden. Diese Netzwerke sind bis heute kaum untereinander verbunden. Ein User in Netzwerk A kann in der Regel keinen Kontakt zu einem User in Netzwerk B aufnehmen ohne sich mit einem neuen Account dort zu registrieren. Dadurch besteht die eigene Identität im Web meistens aus einer ganzen Fülle an unterschiedlichen Netzwerken, Benutzernamen, Passwörtern, etc.

Vision

Das sog. *Semantic Web*, welches durch mehrere Sprachen und Standards vom W3C [<http://www.w3c.org>] definiert wird, stellt den nötigen Rahmen bereit, um offene Schnittstellen über Ontologien zu erstellen und damit die Datensilos heutiger sozialer Netzwerke aufzubrechen. Durch die *Friend of a Friend* (FOAF) Ontologie ist eine solche Schnittstelle um Personen im Semantic Web zu beschreiben bereits vorhanden. Mit dieser Ontologie ist es nicht nur möglich sich selbst über verschiedenste Metadaten zu beschreiben, zusätzlich lassen sich Verbindungen und Beziehungen zwischen Personen darstellen.

Die Daten werden in der Regel in RDF/XML serialisiert. Solche RDF-Dokumente, die FOAF-Daten enthalten, können von den verschiedenen sozialen Netzwerken veröffentlicht werden. Personen werden über URIs identifizierbar. Damit wird es möglich Personen über die bisherigen Grenzen sozialer Netzwerke hinweg miteinander zu verbinden. Es entsteht ein großes **verteiltes soziales Netzwerk**.



Ein verteiltes soziales Netzwerk

Damit ein User die vollständige Kontrolle über seinen persönlichen URI und seine veröffentlichten Daten behält, sollte er die RDF-Daten auf seinen eigenen Webspace oder Webserver laden. Die Verwaltung dieser Daten wird aber schnell unübersichtlich und die XML-Dokumente selbst zu schreiben kostet Zeit.

Der *Orgio* Client schafft eine einfache Schnittstelle zum User. Dieser kann seine Daten dadurch auch ohne direkte Kenntnisse über die Semantic Web Technologien vollständig selbst verwalten. Er macht sich nicht mehr wie bisher von einem Dienst abhängig. Auf bereits bestehende Profile in anderen sozialen Netzwerken kann mit Hilfe von Origo verlinkt werden, sofern die Nutzerdaten auch semantisch verfügbar sind.

Kapitel 2. Konzept

Origo soll dem Nutzer auf komfortable und flexible Weise erlauben seine Identität und sein soziales Netzwerk im semantischen Web zu überschauen und zu verwalten. Das Wort *Origo* kommt aus dem Lateinischen und bedeutet „*Ursprung*“. Der URI des Nutzers wird damit quasi zum Ursprung seines persönlichen Graphen im *Semantic Web*.

Das *Origo* Paket besteht aus mehreren Teilen, die teilweise optional zu verwenden sind:

- **Content Negotiator**

Der Content Negotiator leitet eine HTTP-Anfrage abhängig vom HTTP-Header entweder auf das RDF-Dokument oder auf ein festgelegtes HTML-Dokument weiter.

Pfad: *origo/*

- **RDF Distributor**

Der RDF Distributor generiert die RDF-Daten aus dem sog. Triple Store und liefert diese über einen Caching-Mechanismus aus.

Pfad: *origo/rdf*

- **Origo Client**

Das Hauptmodul von *Origo* ist der Client. Er besteht wiederum aus zwei Modulen, die allerdings nur logisch voneinander getrennt sind:

- **Origo Editor**

Mit dem *Origo Editor* kann der Nutzer seine persönlichen Daten verwalten und seinen persönliche URI mit weiteren Profilen verknüpfen.

- **Origo Browser**

Der *Origo Browser* erlaubt es dem Nutzer sein soziales Netzwerk zu durchstöbern und neue Freunde hinzuzufügen.

Der Client bietet verschiedene Authentifizierungsmöglichkeiten und nutzt einen SPARQL-Endpoint, um auf den Triple Store zuzugreifen.

Pfad: *origo/client/*

- **Konfiguration**

Die Konfiguration von *Origo* wird über eine zentrale INI-Datei vorgenommen. Der Negotiator wird separat konfiguriert, da sich dieser auch auf einem anderen Server befinden kann.

Pfad: *origo/config/*

- **Installer (optional)**

Die Konfiguration von *Origo* kann bequem über den Installer vorgenommen werden. Dieser schreibt die benötigten Konfigurationsdateien und unterstützt den User darin, die Zugriffsberechtigungen auf die verschiedenen Ordner richtig zu setzen.

Pfad: *origo/install/*

- **Identifizier** (optional)

Der Identifizier ist eigentlich kein richtiges Modul. Es ist vielmehr eine Vorgehensweise, um über eine 303 Weiterleitung einen beliebigen URI, der sich auch auf einer anderen Domain befinden kann, quasi vorzuschalten. Ein Vorteil dieser Methode ist, dass der persönliche URI beliebig gewählt werden kann und dieser dadurch kein Hash-URI mehr sein muss. Diese Möglichkeit ist nur für fortgeschrittene Nutzer geeignet.

Pfad: *origo/optional/identifizier/*