

Universität Leipzig – Softwaretechnik Praktikum 2014 - xodx

# Entwurfsbeschreibung

Entwurfsbeschreibung für das Gesamtprojekt „Xodx“

Henrik Hillebrand

3.4.2014

## Inhaltsverzeichnis

Allgemein.....	2
Produktübersicht.....	2
Grundsätzliche Struktur- und Entwurfsprinzipien.....	2
Struktur- und Entwurfsprinzipien einzelner Pakete.....	4
Model .....	4
View .....	5
Controller.....	5
GroupController .....	5
MemberController.....	5
Datenmodell.....	6
Testkonzept .....	6
Glossar .....	6
Triple Store .....	6
Linked Data.....	7
ActivityStream.....	7
Semantic Pingback.....	7
DSSN .....	7
Quellen .....	7
Literatur .....	7
Abbildungsverzeichnis.....	7

## Allgemein

Hinsichtlich des immer stärker werdenden Trends hin zum Semantic Web und der immer dichter werdenden sozialen Vernetzung durch Social Networks wie Google+ und Facebook hat sich das AKSW der Aufgabe angenommen und ein DSSN geschaffen, zu welchem der Zugangsknoten Xodx entstand.

Unser Projekt ist es, diesen Knoten um eine weitere wichtige Funktion zu erweitern, welche es Benutzern ermöglicht sich in Interessensgruppierungen zusammenzufinden, Freundeskreise zu bilden oder Themengruppen zu schaffen. Zusätzlich soll dem Gründer die Möglichkeit gegeben werden seine Gruppen zu bearbeiten, zu löschen und ihre Nutzer zu verwalten.

Allgemein kann gesagt werden, dass diese Funktionserweiterung dem Nutzer die Möglichkeit einräumt eine gruppeninterne Kommunikationsplattform zu schaffen, welche in vielerlei Hinsicht eine effiziente Anwendung finden kann. Dazu sei als Beispiel genannt, dass nach Implementierung und Fertigstellung des Projekts, nachfolgende SWP-Gruppen unsere Arbeit als Mittel der Kommunikation nutzen könnten, indem sie einen Xodx-Knoten auf dem ihnen zugeteilten Praktikumsserver installieren.

## Produktübersicht

Nach erfolgreicher Installation einer Xodx-Instanz auf einem lauffähigem Server - oder wahlweise einer lokal laufenden virtuellen Serverumgebung - kann diese über jeden gängige Webbrowser aufgerufen und verwendet werden. Hierzu muss der Nutzer sich in seinen Xodx-Account einloggen oder einen neuen erstellen. Bei der Erstellung eines neuen Accounts besteht auch die Möglichkeit, diesen mit einem bereits vorhandenen URI zu verknüpfen.

Nach dem beschriebenen Login kann der Nutzer nun auf seinem eigenen Profil die von ihm betretenen und erstellten Gruppen einsehen und per Klick auf deren URI mit seinen Freunden, Kollegen oder Gleichgesinnten kommunizieren oder Links und Bilder mit ihnen teilen. Eine ähnliche Ansicht ist für den Nutzer auch auf dem Profil anderer Freunde o.Ä. zu sehen, wo dieser erfahren kann, welche Gruppen von seinen Kontakten betreten wurden.

Des Weiteren ist in der Gruppe eine Liste mit zugehörigen Benutzern und – dies jedoch nur für den Ersteller dieser Gruppe – ein Menü zu sehen, in welchem er die Gruppe verwalten kann. Dieses Menü enthält beispielsweise die Möglichkeit die Attribute, wie Name, Beschreibung usw. der Gruppenressource zu editieren oder diese zu schließen/löschen.

Das Teilen von Feeds innerhalb der Gruppe ermöglicht eine Kommunikation mit denen in dieser Gruppe befindlichen Mitgliedern. Diese Activity-Feeds können Links, normale Textnachrichten oder Bilder sein.

## Grundsätzliche Struktur- und Entwurfsprinzipien

Da unser gewähltes Projekt auf dem bereits erwähnten Xodx-Knoten aufbaut und für diesen als Erweiterung dient, kann als grundsätzliche Struktur der von unserer Gruppe entwickelten Softwarestudie das Strukturprinzip des Xodx als gegeben empfunden werden. Dies gilt auch, gesondert zu erwähnen, aus dem Grund, da in Xodx bereits die Funktion zum Hinzufügen von Freunden implementiert ist und die Lösch-Funktion selbiger eine Abwandlung dieser darstellt.

Grundsätzlich baut Xodx wiederum auf dem DSSN auf, welches intern in 3 verschiedene Ebenen (layer) unterteilt ist: Die Daten-, Protokoll- und Anwendungs- und Service-Ebene. Dies entspricht

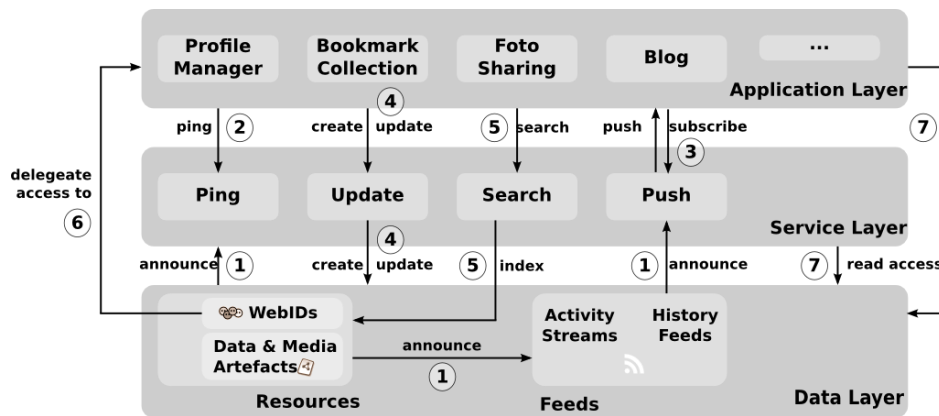


Abbildung 1: Ebenen-Modell des DSSN

Die Daten-Ebene beinhaltet die RDF-Ressourcen, Media-Daten wie Bilder und letztlich die Activity-Feeds. Diese Daten werden über einen URI eindeutig benannt und können per Linked Data abgerufen und benutzt werden. Zudem besteht noch die Möglichkeit jede Ressource zu kommentieren oder andere Relationen zwischen ihnen zu erzeugen (Freundschaften zwischen Personen-Ressourcen o.ä), welche auch in der Daten-Ebene gespeichert werden.

Die Protokoll-Ebene ermöglicht die Kommunikation der einzelnen Dienste des DSSN untereinander. Da es sich bei Xodx um eine Webapplikation handelt, ist dieser Ebene zum einen das HTTP zuzuordnen, allerdings auch nötige Protokolle wie das zuvor angesprochene Linked Data, das Semantic Pingback und PubSubHubbub.

Die Service- und Anwendungsebene stellt die Dienste oder Anwendungen bereit - mit anderen Worten die Oberfläche -, die gebraucht werden um innerhalb des Knotens auf Konten registrierter Xodx-Nutzer zuzugreifen, aber auch für beliebige andere Nutzer des DSSN. Die Ebene verfügt daher bspw. über den Profilmanager und das Teilen von Media-Daten, Benachrichtigungen und anderem.

Zusätzlich zu erwähnen sei, dass Xodx selbst, und damit auch unsere Erweiterung, auf bereits bestehende Bibliotheken und Frameworks zurückgreift, die die Arbeit mit den angesprochenen Protokollen und RDF-Ressourcen erheblich vereinfacht. Dieses vermeidet Dopplungen im Code und erhöht die Qualität des Quellcodes wenn um Struktur und Übersichtlichkeit geht. So nutzt Xodx das Erfurt-Framework - welches dem Programmierer den Zugriff auf Triple Stores und deren Modelle vereinfacht und mittel SPARQL die Nutzung dieser effizient ermöglicht -, das Saft-Framework - welches als Grundgerüst für Xodx dient und das MVC-Entwurfsmuster nutzt - und letztlich das lib-dssn-php - welches die Verarbeitung von Aktivitätsstreams erleichtert.

Da durch das SAFT-Framework das MVC-Entwurfsmuster verwendet wird, liegen als zusätzliche Strukturelle Ebene die Model-, View-, und Controllerebene über den angesprochenen DSSN-Ebenen, welche durch ein Layout-System auf der View-Ebene und der Controller "ResourceController" und zusätzliche actions in der Controller-Ebene die den Zugriff für den Programmierer noch weiter vereinfachen.

Erweitert wird dieses bisherige Modell um einige neue Klassen, die, wie eingangs beschrieben, das Xodx-Projekt um die Gruppenfunktion ergänzt. Die genaue Funktion der einzelnen Klassen wird im nachfolgenden Punkt genauer erläutert, die Klassen selbst im nachstehenden Klassendiagramm aufgeführt.

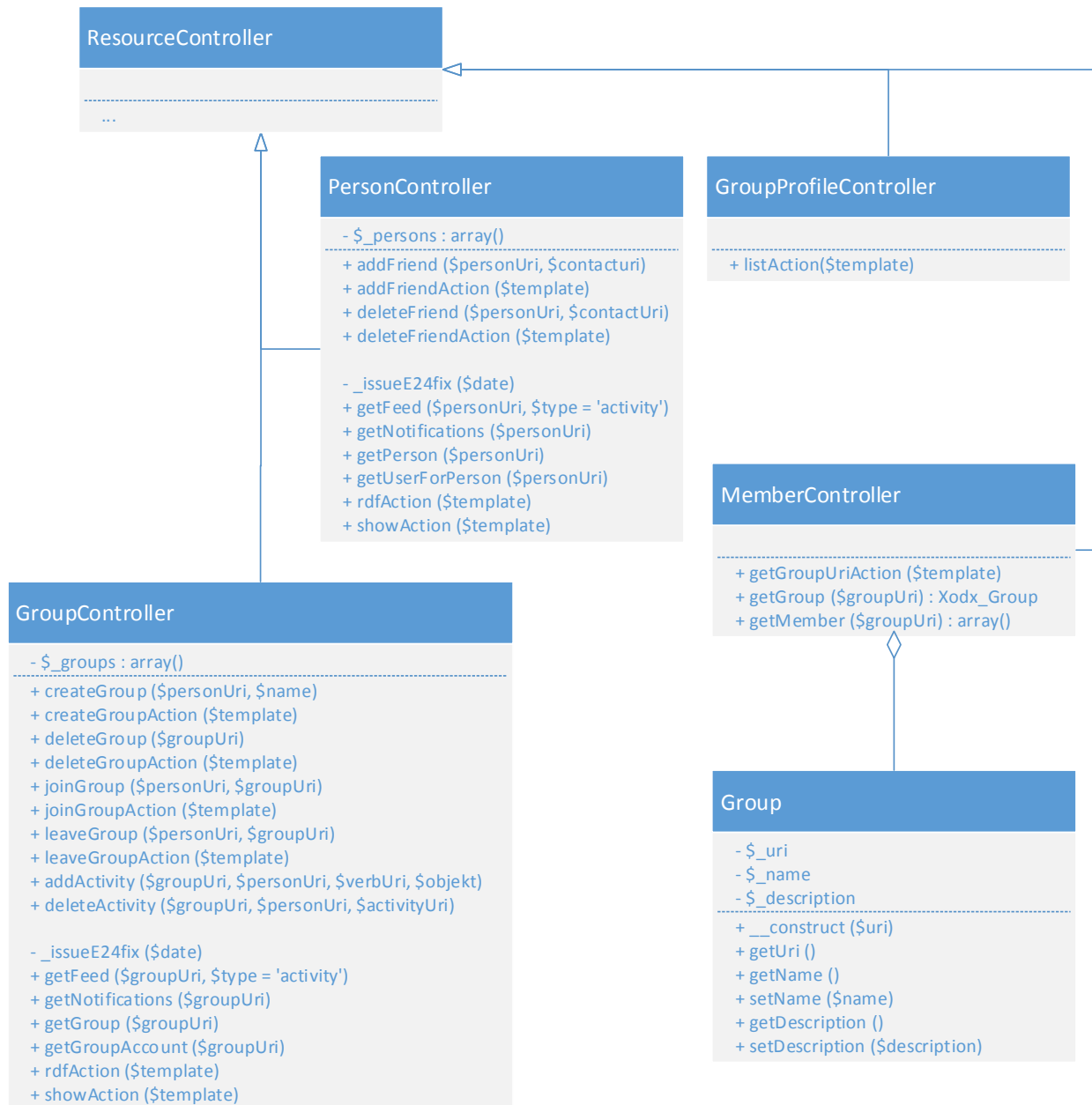


Abbildung 2: UML-Modellierung Gruppenkonzept

## Struktur- und Entwurfsprinzipien einzelner Pakete

Wie zuvor erwähnt dient das Soft-Framework dem Zweck, dem Entwurfsmuster MVC konkret in Xodx folgen zu können. Daher ist im nachfolgenden die Struktur der einzelnen Pakete in die Ebenen eben jenes Entwurfsprinzip unterteilt und in diesen näher erläutert.

### Model

Unsere Xodx-Erweiterung greift im Bereich der Model-Ebene, welche für die Speicherung der Daten und Anbindung an die Datenbank zuständig ist, auf die in dem Triple Store gespeicherten Einträge zu. Hier werden neue Relationen eingetragen, welche die Gruppenfeeds in Relation mit der Gruppe und dem Schreiber setzen und zusätzlich die Verbindungen zwischen den Mitgliedern und der Gruppe selbst über das Prädikat „foaf:member“.

Im Datenmodell sind weiterhin ersichtlich, welche Verbindungen in der Datenbank von uns gespeichert, verändert oder bei Auflösen der Gruppe gelöscht werden. Der Zugriff auf diese Relationen erfolgt wie auch in Xodx selbst und im Vorprojekt über den Triple Store Virtuoso.

## View

Auf der View-Ebene wird das Template des eigenen Profils und das anderer Profile so weit verändert, dass eine dynamisch erstellte Gruppenliste anhand des nebenstehenden Schemas (Abb. 2) zu sehen sein wird, welche auf die zugehörige Gruppenressource verweist. Zusätzlich wird es möglich sein, diese Gruppen per Klick auf den Button „Leave Group“ die Gruppe zu verlassen, und durch Eingabe einer gültigen Gruppen-URI, im zugehörigen Feld, einer neuen Gruppe beizutreten.

Zusätzlich wird ein neues Template erstellt werden, welches die Gruppenressource selbst widerspiegelt. Auf ihm wird ein Activity-Feed, einer Mitgliederliste und eventuelles Administrationsmenü sichtbar sein.

Der Activity-Feed ermöglicht es, wie eingangs erwähnt, mit den Mitgliedern zu kommunizieren, Bilder und Links zu teilen oder einen Verlauf über hinzugekommene und ausgetretene Mitglieder zu erhalten.

Das Template dieser Ressource ist bereits als vorläufiger Entwurf in Abbildung 3 zu sehen.

### Groups

Abbildung 3: Gruppenliste

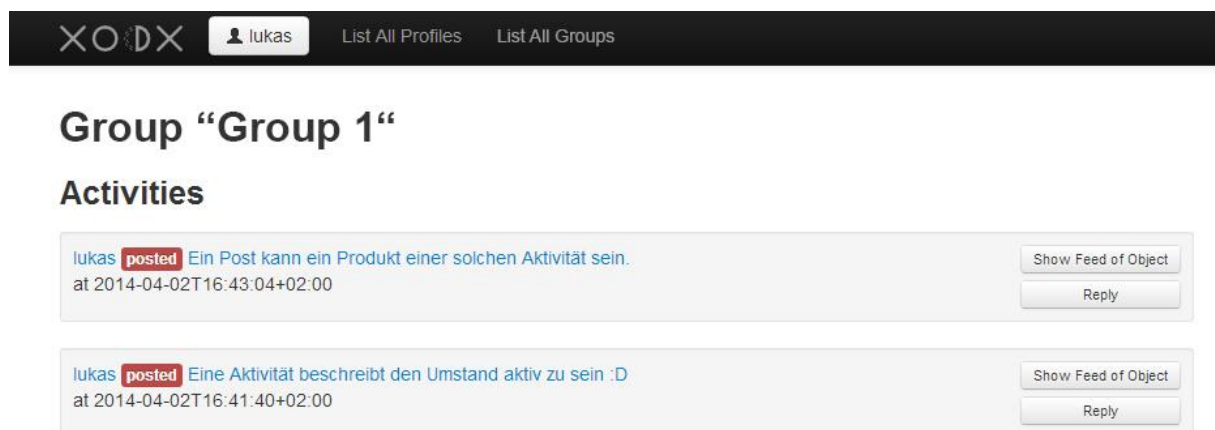


Abbildung 4: Darstellung der Gruppe (nicht beigetreten) in Xodx

## Controller

Innerhalb der Controller-Ebene werden ebenfalls einige Erweiterungen vorgenommen werden. So wird diese um die folgenden 2 Klassen erweitert werden:

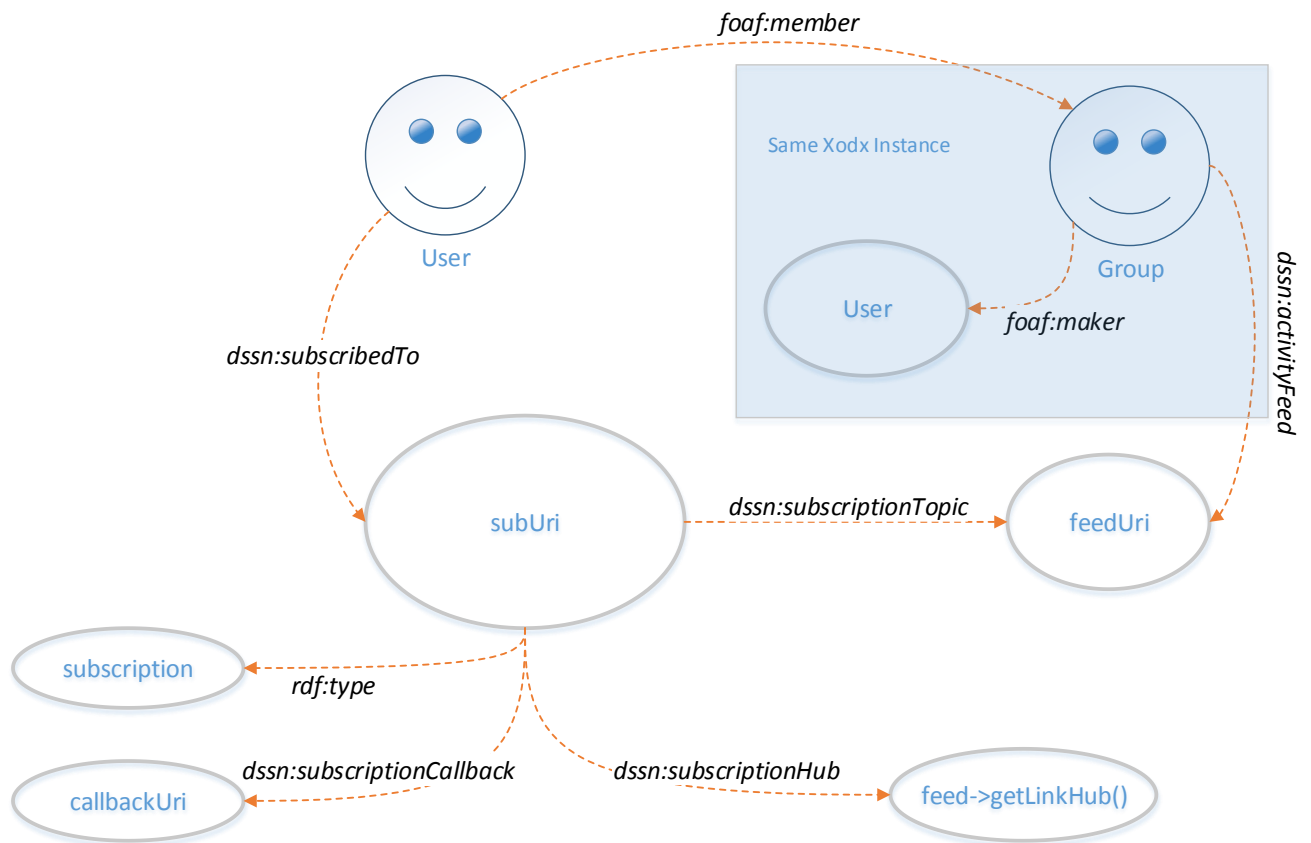
### GroupController

Der GroupController enthält alle nötigen Funktionalitäten um Gruppen zu erstellen, zu löschen, ihnen beizutreten und letztlich eben diese wieder zu verlassen. Außerdem dient er der Verwaltung der gruppeninternen Activities. Die genaue Funktionsübersicht ist dem Klassendiagramm zu entnehmen.

### MemberController

Der MemberController dient der Mitgliederübersicht und der Gruppenübersicht auf den Profilen. Wie auch im GroupController erwähnt, können für den MemberController ebenfalls alle Funktionen im Klassendiagramm ermittelt werden.

## Datenmodell



## Testkonzept

Innerhalb des Projekts greifen wir hauptsächlich auf zwei gängige Testkonzepte zurück. Zum einen ist dies PHPUnit, welches zum automatisierten Testen der erweiterten und neu implementierten Klassen dient. Hierzu entwickelt jedes Mitglied, welches an der Implementierung mitwirkt, eine Testklasse, welches das Durchführen automatisierter Test ermöglicht.

Zum anderen nutzen wir Selenium, in welchem wir sich oft wiederholende Testabläufe speichern können um diese im Anschluss daran automatisiert durchführen lassen können. Diese sogenannten Stresstests dienen der Simulation möglicher realer Extremsituationen, in denen eine hohe Auslastung auftritt. Da es sich bei unserem Projekt um eine soziale Plattform im Internet handelt, auf welche von überall zugegriffen werden kann, ist ein solcher Test der Performance äußerst wichtig.

Abseits dieser Testverfahren wird zusätzlich mit Hilfe von Mantis Bug Tracking ein effizientes Fehlermanagement organisiert und jedes Mitglied verfügt unabhängig der laufenden Xodx-Instanz des Praktikums-Servers eine eigene Instanz, lokal laufend auf einer VM.

## Glossar

### Triple Store

Ein Tripel ist die Einheit bestehend aus drei Elementen Subjekt, Prädikat und Objekt. Eine Ressource als Subjekt wird mit einer weiteren Ressource als Prädikat zusammen mit einer dritten Ressource als Objekt näher beschrieben. Das Tripel besteht aus den beiden Enden des Links und der Art des Links und ist somit eine logische Aussage.

Die Link-Struktur des Tripels bildet einen gerichteten benannten Graph, welcher die logische Aussage darstellt. Die Graphdatenbank benutzt diese um vernetzte Informationen darzustellen und zu speichern.

### Linked Data

Bezeichnet im WWW Daten, welche frei verfügbar sind und über einen zugehörigen URI direkt via HTTP abgerufen werden können. Zusätzlich besteht die Möglichkeit, auf andere Daten zu verweisen, welche auch über einen URI identifiziert werden.

### ActivityStream

Ein Activity Stream ist eine Liste von Aktivitäten (Activities), die ein Individuum (z.B. Person oder Gruppe), getätigt hat.

### Semantic Pingback

Semantic Pingback ist eine, auf die Veröffentlichung von RDF-Ressourcen mit Linked Data angepasste Methode, die es Web-Autoren erlaubt Benachrichtigungen anzufordern, falls jemand eine Ressource verlinkt.

### DSSN

Das DSSN ist ein dezentralisiertes semantisches soziales Netzwerk der AKSW

## Quellen

### Literatur

- "An Architecture of a Distributed Semantic Social Network" von Sebastian Tramp (28.03.14)
- Natanael Arndt (2013): Xodx, Konzeption und Implementierung eines Distributed Semantic Social Network Knotens (Masterarbeit)

## Abbildungsverzeichnis

Abbildung 1: Ebenen-Modell des DSSN.....	3
Abbildung 2: Gruppenliste des Xodx .....	5
Abbildung 3: Darstellung der Gruppe in Xodx.....	5