



Masterarbeit

**Ein System zur partiellen Synchronisation
von Wissensbasen für dezentrale soziale Netzwerke**

von Jens Grundmann
15. September 2015

Hochschule für Technik und Wirtschaft Berlin
Fachbereich Wirtschaftswissenschaften II
Studiengang Angewandte Informatik

Erstgutachter/in Prof. Vorname Name
Zweitgutachter/in Vorname Name

Insert later

- Themen Shark Framework.

Inhaltsverzeichnis

1	Einleitung	3
1.1	Zielsetzung	3
2	Grundlagen	3
2.1	Shark Framework	3
3	Konzeption	3
3.1	Funktionale und nicht funktionale Anforderungen	3

1 Einleitung

Ziel dieser Arbeit ist es eine Softwarekomponente zu entwickeln, die partiellen Synchronisation von Wissensbasen ermöglicht. Eine Wissensbasis ist dabei nichts anderes als eine Menge an Daten, die in einer bestimmten Stuktur vorliegen bzw. durch eine abstrakte Darstellung beschreibbar sind.

2 Grundlagen

Das folgende Kapitel widmet sich den Grundlagen, auf denen die Arbeit aufbaut. Es werden die Themen Shark Framework,[*insert more here*] besprochen.

2.1 Shark Framework

Das ist ein Zitat. [2]

3 Konzeption

3.1 Funktionale und nicht funktionale Anforderungen

Im Folgendem werden die funktionalen und nicht funktionale Anforderungen besprochen. Diese beschreiben welche Features die Softwarekomponente bereitstellen soll, sowie wichtige Aspekte der Qualitätssicherung.

Funktionale Anforderungen

- **Beschreibbarkeit:** Es ist möglich einen Raum von Daten zu beschreiben und diesen von einem anderen Raum von Daten abzugrenzen.
- **Abhängigkeiten:** Es ist möglich Abhängigkeiten zwischen Räumen zu definieren. So soll beispielsweise der Raum Java-Chat ein Kind des Programmiersprachen-Chat Raumes sein können.
- **Persistenz:** Es soll möglich sein die Beschreibung der Räume von Daten persistent zu speichern. Die gespeicherten Räume bleiben somit erhalten und und können zu späterem Zeitpunkt neu geladen werden.
- **Synchronisation:** Es ist möglich die Räume von Daten und ihre Abhängigkeiten mit Peers in einem Peer to Peer Netzwerk zu synchronisieren. Ziel ist es, dass die Räume nach der Synchronisation identisch von Aufbau und Inhalt sind.

Nicht funktionale Anforderungen

- **Build-Management:** Die Softwarekomponente ist mittels eines zu bestimmenden Build Tools so eingerichtet, dass das Aufsetzen der Entwicklungsumgebung für andere Entwickler schnell und einfach zu erledigen ist. Mögliche Synergien des gewählten Tools mit anderen Systemen zur Softwareentwicklung und Qualitätssicherung, zum Beispiel Jenkins [1], sind wünschenswert.
- **Testbarkeit:** Die zu Softwarekomponente ist modular so aufgebaut, dass sie durch Modultest testbar ist.
- **Modultest:** Es existieren bereit eine Reihe von Modultest, welche die grundlegende Funktionalität der Softwarekomponente sicherstellen.

Webseiten

- [1] *Jenkins*. URL: <http://jenkins-ci.org/> (besucht am 15.09.2015).
- [2] Prof. Dr. Thomas Schwotzer. *Shark framework*. URL: <http://www.sharksystem.net/> (besucht am 31.08.2015).

Abbildungsverzeichnis

Tabellenverzeichnis