

**Konzeption und Entwicklung einer Schnittstelle für
ein modernes Messaging-System**
in Hinblick auf professionelle Wartung und Performance

Abschlussarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

im Fachbereich Wirtschaftswissenschaften II (Fachbereich 4)
Studiengang Angewandte Informatik

an der
Hochschule für Technik und Wirtschaft Berlin

Verfasser: Maximilian Mackeprang
Betrieb: MOTOR-TALK-GMBH
Anschrift:

1.Prüfer: Prof. Dr.-Ing. Elke Naumann
2.Prüfer: Dipl. Inf. Michael Schwalbe
Abgabedatum: 04.09.2012

Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VI
1 Einleitung	1
1.1 Betriebliches Umfeld	1
1.2 Projektumfeld	1
1.3 Motivation	1
1.4 Zielstellung	1
2 Grundlagen	2
2.1 Kommunikationstheorie	2
2.2 Netzwerkprotokolle zur Kommunikation	3
2.3 Sicherheitsaspekte	5
2.3.1 OAuth 2.0	5
2.4 Messaging-Protokolle	6
2.5 Performance	8
2.5.1 Parallelisierung	8
2.5.2 Denormalisierung	9
2.5.3 Fragmentierung	10
2.6 Wartbarkeit	11
2.6.1 Lesbarkeit des Quellcodes	11
2.6.2 Wartbarkeit auf Funktionsebene	11
2.6.3 Wartbarkeit auf Klassenebene	12
2.6.4 Wartbarkeit auf Modulebene	13
2.7 Wartbarkeit im Java-Umfeld	14
2.7.1 Prinzipien	14
2.7.2 Dependency Injection	15
2.7.3 Architektur	16
2.8 Spring	17
2.8.1 DI Container	17
2.8.2 Unterstützung aspektorientierter Programmierung	18
2.8.3 Transaktionen	18
2.8.4 Spring Web MVC	18
2.9 Stripes	18
2.9.1 Darstellung	18
2.9.2 Logik	19
3 Analyse und Anforderungsdefinition	20
3.1 Marktanalyse	20
3.1.1 Allgemeine Anforderungen	20
3.1.2 Systemspezifische Analyse	20
3.2 Ist-Analyse	21
3.3 Probleme der derzeitigen Lösung	21
3.4 Anforderungsdefinition	22
3.4.1 Zielbestimmung	22

3.4.2	Produkteinsatz	24
3.4.3	Produktfunktionen	24
3.4.4	Qualitätsanforderungen	27
4	Entwurf	28
4.1	Vergleich der Lösungsansätze	28
4.2	Datenentwurf	28
4.2.1	Konzept Notification	28
4.2.2	Datenbank-Modell	29
4.3	Komponenten	30
4.3.1	Aufteilung	30
4.3.2	Server	30
4.3.3	Client	33
4.4	Schnittstellen	33
4.5	Authentifizierung	34
4.5.1	Client	34
4.5.2	Server	34
5	Implementierung	36
5.1	Umgesetzte Anwendung	36
5.2	Datenmodell	36
5.3	Schnittstelle	36
5.3.1	Das Motor-Talk RPC Framework	36
5.3.2	ServletBasedExecutionContext	38
5.3.3	Schnittstellen-Klassen	38
5.4	Konfiguration	38
5.4.1	Konfiguration Server	39
5.4.2	Konfiguration Testclient	39
5.5	Autorisierung / Authentifizierung	40
5.6	Persistenz	40
5.6.1	Konzept Query	40
5.6.2	Datenbank	40
5.7	Server-Logik	41
5.8	Testclient	42
5.9	Codequalität	43
5.9.1	Metriken	43
5.9.2	Untersuchung der Klasse MessageManagement	45
5.9.3	Coding Conventions	46
6	Tests	47
6.1	Funktionale Tests	47
6.1.1	Unit Tests	47
6.2	Integrationstests	47
6.3	Lasttests	48
6.3.1	Zielstellung	48
6.3.2	Vorgehen	48
6.3.3	Testumgebung	49

6.3.4 Testergebnisse	50
7 Zusammenfassung	51
7.1 Ergebnisse	51
7.2 Bewertung	51
7.3 Ausblick	52
A Mockups des Client	53
B Bilder des Prototypen	54
C Produktfunktionen	56
C.1 Nutzerfunktionen	56
C.2 Moderatorfunktionen	65
C.3 Administratorfunktionen	68
D Lasttest Ergebnisse	70
E Vergleich der Aufwände	72
Literaturverzeichnis	74
Abkürzungsverzeichnis	78
Ehrenwörtliche Erklärung	79