



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

**Philipp Prögel**

**Dienstkombination für kooperatives Arbeiten in der Lehre**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Philipp Prögel

**Dienstkomposition für kooperatives Arbeiten in der Lehre**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Martin Becke  
Zweitgutachter: Prof. Dr. Thomas Lehmann

Eingereicht am: 15. Juni 2016

## **Philipp Prögel**

### **Thema der Arbeit**

Dienstkomposition für kooperatives Arbeiten in der Lehre

### **Stichworte**

Web-Dienst, Dienstkomposition

### **Kurzzusammenfassung**

Die heutige Lehr- und Arbeitswelt entwickelt sich immer stärker in eine Richtung der agilen Gruppen- und Projektarbeit. Beide Welten haben das gemeinsame Ziel, die Herausforderungen in der Kooperation möglichst effizient zu lösen. Einen besonderen Einfluss auf den Gesamterfolg hat hier die Organisation [1] innerhalb des Teams.

Um die Zusammenarbeit zu erleichtern, sind verschiedene Web-Dienste für die Optimierung der Organisation[1] entstanden. Insbesondere verschiedene Ansätze aus den Service-orientierten Architekturen (SOA) werden in diesem Kontext immer erfolgreicher. So können zum Beispiel Dateien von überall aus der Welt gemeinsam aktiv bearbeitet werden. Auch Anwendungen die auf Echtzeitkommunikation basieren, wie textbasierte Chats oder interaktive Videokonferenzen, sind ein wichtiger Teil des Angebots. Gemeinsam ist all diesen Diensten, dass die Kommunikation und die Organisation innerhalb einer Gruppe deutlich vereinfacht wird. Sie lösen nicht das Problem, helfen aber bei der Problemlösung.

Doch diese Vielfalt an Diensten stellt auch eine Herausforderung für die einzelnen Gruppenmitglieder dar. Üblich ist, dass jeder dieser Dienste unabhängig für sich alleine angeboten wird, aber nicht unabhängig in der Nutzung zu sehen ist. So hat der Anwender eine noch wenig diskutierte aber wichtige Herausforderung in der Komposition dieser Dienste zu erfüllen. Beispielfhaft muss jeder Nutzer für sich organisieren, ob und wann er pro Dienst eine Applikation startet oder eine Webseite besucht. Dies ist eine nicht zu unterschätzende Aufgabe, die nicht nur mit der Zeit für die Synchronisation der Dienste zu beschreiben ist. Automatisierte Mechanismen zur Unterstützung sind bisher noch nicht etabliert. Diese Arbeit soll einen ersten Beitrag für zukünftige Lösungen anbieten mit der Umsetzung einer unterstützenden Middleware.

Zusammengefasst wird im Rahmen dieser Ausarbeitung eine erste Middleware für die Dienstkomposition zum kooperativem Arbeiten für Studenten entwickelt, die geeignete Dienste für

kooperatives Arbeiten in der Lehre zusammenfasst. Hierfür werden insbesondere externe Web-Dienste auf ihre Fähigkeit für die Dienstkomposition innerhalb eines Frameworks untersucht. Die Betrachtung der Schnittstellen, also die Application-Programming-Interfaces (API), spielen eine besondere Rolle und bedürfen daher einer besonderen Betrachtung. Auch weitere technische, dienstabhängige Kriterien der Dienstkomposition sind zu identifizieren und als Kriterienkatalog für die zu erstellende Middleware innerhalb dieser Arbeit bereitzustellen.

Eine weitere Designanforderung in der Entwicklung ist der Aufbau einer möglichst generischen Lösung. Primär bedeutet dies Gemeinsamkeiten der Dienste zu identifizieren und architektonisch zu abstrahieren. Hier spielt auch der Einsatz moderner Webtechnologien in einem systemadaptiven Anwendungsszenario eine besondere Rolle, um eine ressourcensparende und skalierende Lösung zu entwickeln.

Basis für die Auswahl der Dienste wird die Auswertung einer Studentenbefragung bieten, die auch im Rahmen dieser Arbeit erstellt werden wird.

**Philipp Prögel**

#### **Title of the paper**

Servicecomposition for cooperative work

#### **Keywords**

service, servicecomposition

#### **Abstract**

English abstract goes here TODO...

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problemstellung . . . . .	2
1.3	Zielsetzung . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Dienst . . . . .	4
2.2	Web-Dienst . . . . .	5
2.3	Dienstkomposition . . . . .	5
<b>3</b>	<b>Middleware</b>	<b>6</b>
3.1	Requirements Engineering . . . . .	6
3.1.1	Umfrage . . . . .	6
3.1.1.1	Aufbau . . . . .	7
3.1.1.2	Auswertung . . . . .	7
3.1.2	Funktionale Anforderungen . . . . .	10
3.1.3	Nichtfunktionale Anforderungen . . . . .	11
3.2	Technologie . . . . .	11
3.2.1	Übersicht . . . . .	11
3.2.2	Auswahl . . . . .	11
3.3	Analyse und Design . . . . .	11
3.4	Architektur . . . . .	11
<b>4</b>	<b>Implementation</b>	<b>12</b>
4.1	Besondere Herausforderungen . . . . .	12
<b>5</b>	<b>Anwendung der Middleware</b>	<b>13</b>
5.1	Validierung der Anwendungsfälle . . . . .	13
5.2	Stärken und Schwächen der Implementierung . . . . .	13
<b>6</b>	<b>Fazit</b>	<b>14</b>
6.1	Entwicklungsstand . . . . .	14
6.2	Ausblick auf zukünftige Entwicklungen . . . . .	14
	<b>Abkürzungsverzeichnis</b>	<b>15</b>

# Listings

# 1 Einleitung

## 1.1 Motivation

Aufgaben und Arbeiten während des Studiums werden immer öfter in Gruppen bearbeitet. Dabei hat die Kooperation innerhalb der Gruppe einen großen Einfluss auf das Ergebnis. Fehlende Kommunikation, planloses Vorgehen und keine Arbeitseinteilung sind Hinweise auf eine ungenügende Organisation innerhalb der Gruppe, und insbesondere bei Studenten.

Um dem entgegen zu wirken wurden verschiedene Web-Dienste für das kooperative Arbeiten entwickelt. Einige, für diese Ausarbeitung wichtige Web-Dienste, können in folgende Kategorien eingeteilt werden:

- Die Cloud-Storage-Dienste [1, 5] bieten elementare Funktionen für die Zusammenarbeit von Gruppen. Dateien können online gespeichert und über das Internet bereit gestellt werden. Dadurch können Dateien einfach mit Gruppenmitgliedern geteilt werden.
- Eine weitere Kategorie an Web-Diensten für kooperatives Arbeiten bilden die Instant-Messaging-Dienste [2, 80]. Mithilfe dieser können sich Gruppen in Echtzeit austauschen und aktuelle Ereignisse besprechen.
- Vor allem in der Informatik spielen die Web-Dienste für Versionsverwaltung eine elementare Rolle [3]. Sie ermöglichen die Koordinierung von mehreren Entwicklern an einer Datei und die Protokollierung von Änderungen, wodurch nachverfolgt werden kann, welches Gruppenmitglied etwas geändert hat.
- Angelehnt an die bereits angesprochenen Cloud-Storage-Dienste existieren auch die Document Collaboration-Dienste [4, 12]. Diese erlauben eine gleichzeitige und gemeinsame Bearbeitung von Dokumenten.

### 1.2 Problemstellung

Die kooperative Zusammenarbeit in der Gruppe bedarf also einer Vielzahl an Diensten. Dabei steht der Anwender vor mehreren Problemen. Der Nutzer verliert an Ordnung und Übersicht, weil er für jeden Dienst eine Webseite besuchen oder die Applikation auf seiner Zielplattform starten muss um dessen Funktionen zu nutzen. Außerdem bestehen keine sinnvollen Automatismen für die Verknüpfung von Diensten, weshalb der Anwender die Daten zwischen den Diensten eigenständig synchronisieren muss.

Es stellt sich die Frage, wie diese Probleme gelöst werden können. Eine mögliche Antwort darauf ist die Dienstkomposition. Dabei werden verschiedene Funktionen von mehreren Diensten miteinander verbunden. Im Zuge dessen können Verknüpfungen zwischen den Diensten geschaffen werden, welche in der autonomen Ausführung der einzelnen Dienste nicht existent sind.

Es könnte beispielsweise bei jeder Änderung an einer Datei eine Nachricht per Messenger-Dienst verschickt werden, um das Team über aktuelle Entwicklungen zu informieren. Dateien aus einem Cloud-Storage-Dienst könnten an einen Document Collaboration-Dienst weitergeleitet werden, wodurch die gemeinsame Bearbeitung an der Datei ermöglicht wird. Es sind also Verknüpfungen zwischen unterschiedlichen Diensten vorstellbar, die allesamt die Kooperation und Organisation innerhalb der Gruppe fördern und somit bei der Problemlösung helfen.

### 1.3 Zielsetzung

Ziel der vorliegenden Ausarbeitung ist die Erstellung einer Dienstkomposition für Web-Dienste in der Lehre. Dabei soll der Anwendungsbereich der Dienste die Zusammenarbeit und Organisation innerhalb einer Gruppe enthalten.

Um die relevanten Web-Dienste zu identifizieren, wird im Rahmen dieser Arbeit eine Umfrage erstellt, bei der die Studenten der HAW Hamburg im Fachbereich Informatik befragt werden. Des Weiteren werden Anforderungen an die Dienstkomposition aus den Umfrageergebnissen abgeleitet. Ein Kriterienkatalog für Zusammensetzbarkeit von Web-Diensten ist ebenfalls Teil dieser Ausarbeitung. Dadurch wird festgestellt, ob sich ein Dienst für die Komposition eignet oder nicht.

Für die Dienstkomposition wird zunächst eine *Middleware* konzipiert und implementiert. Die *Middleware* hat die Aufgabe einzelne Funktionen der Web-Dienste bereitzustellen. Die



benötigten Daten werden durch verschiedene Web-Schnittstellen von den Web-Diensten zur Verfügung gestellt. Eine besondere Designvorgabe an die *Middleware* ist eine wiederverwendbare und plattformunabhängige Lösung. Im Zuge dessen können zukünftige Arbeiten mit dem Thema Dienstkomposition auf die *Middleware* zurückgreifen und ihre Arbeit darauf aufbauen.

Abschließend wird mithilfe der implementierten *Middleware* eine Webseite entwickelt. Auf dieser sollen die Funktionen der Dienste, die Teil der *Middleware* sind, bereitgestellt werden.

## 2 Grundlagen

In diesem Kapitel werden wichtige Begrifflichkeiten im Zusammenhang mit dieser Ausarbeitung beschrieben. Dabei sollen Abgrenzungen zu bestehenden Definitionen getroffen und die Bedeutung der Begriffe beschrieben werden.

### 2.1 Dienst

Um im nachfolgenden Abschnitt den Begriff Web-Dienst definieren zu können muss zunächst die übergeordnete Kategorie Dienst für sich betrachtet werden. Ein Dienst beschreibt in der Informatik zumeist ein in sich geschlossenes System, welches zusammenhängende Funktionen über ein Themenfeld bündelt und mithilfe einer Schnittstelle zur Verfügung stellt.[5, 3]

Dienste können in unterschiedliche Arten beziehungsweise Aufgabenfelder eingeteilt werden. Zum Beispiel Systemdienste, Web-Dienste oder auch Netzwerkdienste. Dabei bestimmt die Art des Dienstes auf welche Weise die Schnittstelle erreichbar ist. Ein Systemdienst bietet seine Schnittstelle nur für das Betriebssystem an. Ein Netzwerkdienst für das gesamte Netzwerk. Und ein Web-Dienst ist von überall durch das Internet zu erreichen.

Ein Dienst kann in verschiedenen Bereichen spezifiziert werden. Wie zum Beispiel Sicherheit, Zuverlässigkeit oder *Performance*. [6, 89] Im Rahmen dieser Ausarbeitung ist die komplette Spezifikation für einen Dienst nicht notwendig. Aus diesem Grund wurden folgende Eigenschaften für einen Dienst aufgestellt:

- Der Dienst benötigt einen Namen oder *Identifier* um erreichbar zu sein. Falls der Dienst mit mehreren anderen Diensten in einem System gekoppelt wird, muss der Name oder *Identifier* eindeutig sein.
- Es wird eine Schnittstelle benötigt. Über diese Schnittstelle werden die Funktionen des Dienstes bereitgestellt. Innerhalb der Schnittstelle müssen die Eingabe-, Ausgabe- und Fehlerparameter als auch die Nachrichtentypen für die jeweiligen Funktionen beschrieben werden.

- Jeder Dienst besitzt einen Anwendungsbereich oder *Scope*. Dieser kann von der Applikation über das System bis hin zum Internet reichen.

### 2.2 Web-Dienst

Es stellt sich nun die Frage was einen Dienst zu einem Web-Dienst macht. Web-Dienste oder auch *Webservices* werden auf unterschiedliche Weisen definiert:

- "It exposes its features programmatically over the Internet using standard Internet languages and protocols"[7, 2]
- "It can be implemented via a self-describing interface based on open Internet standards (e.g., XML interfaces which are published in a network-based repositories)"[7, 2]

Die angesprochenen Standards für die Entwicklung der Web-Dienste sind Simple Object Access Protocol (**SOAP**) [8] und Representational State Transfer (**REST**) [9]. Durch diese Definitionen sind Dienste die ihre Schnittstellen mit anderen Protokollen anbieten ausgeschlossen. Beispiele für andere Protokolle sind zum Beispiel Advanced Message Queuing Protocol (**AMQP**) [10] oder Google Remote Procedure Call (**GRPC**) [11]. Im Gegensatz zu den oben beschriebenen einschränkenden Definitionen für Web-Dienste identifizieren in der vorliegenden Ausarbeitung folgende Eigenschaften einen Web-Dienst:

- Der Dienst erfüllt die oben eingeführten Eigenschaften eines Dienstes.
- Der *Scope* des Dienstes ist im *Web* anzusiedeln.
- Die Schnittstelle des Dienstes ist öffentlich erreichbar und für die Bereitstellung der Dienstfunktionen vorgesehen.

### 2.3 Dienstkomposition

Nachdem Dienste und Web-Dienste beschrieben sind, muss noch die Dienstkomposition definiert werden. Eine Dienstkomposition beschreibt den Zusammenschluss mehrerer Dienste zu einer neuen Anwendung oder auch einem neuem Geschäftsprozess. Dabei kann die entstandene Komposition wieder als neuer Dienst verstanden werden [12, 4]. Eine Dienstkomposition kann sowohl Proaktiv als auch Reaktiv aufgebaut sein. Proaktiv heißt, dass die zugehörigen Dienste der Komposition vor der Laufzeit bekannt sind. Wohingegen eine reaktive Komposition die Dienste dynamisch zusammenstellt [12, 5].

## 3 Middleware

Dieses Kapitel befasst sich mit der Umsetzung der Middleware der Dienstkomposition. Es werden Anforderungen aufgestellt, Übersicht und Auswahl zur Technologie besprochen und die Architektur definiert.

### 3.1 Requirements Engineering

#### 3.1.1 Umfrage

Im Rahmen der Ausarbeitung wurde eine Umfrage durchgeführt, mit dem Ziel, Web-Dienste zu finden, aus denen sich die Dienstkomposition zusammenschließen soll. Außerdem sollen anhand der Auswertung Anforderungen an Middleware und Frontend abgeleitet werden. Befragt wurden Studenten der Fachrichtung Informatik an der HAW Hamburg. Durch die Einschränkung auf eine Fachrichtung soll der Umfang an in Frage kommenden Web-Diensten verringert werden. Wie in Abbildung 3.1) zu sehen ist, haben insgesamt 100 Studenten teilgenommen, wobei mit 60 Studenten die Fachrichtung Technische Informatik am stärksten vertreten ist. Angewandte und-Wirtschaftsinformatik sind mit 20 beziehungsweise 17 Studenten vertreten.

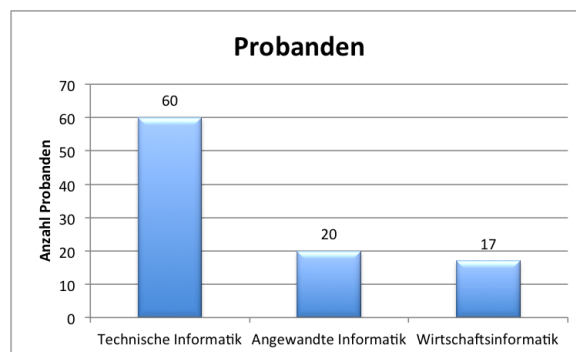


Abbildung 3.1: Verteilung der Probanden

Im Sommersemester 2016 befanden sich insgesamt 968 Studenten in den erwähnten Studiengängen [13]. Also wurden ungefähr 9,68% der in Frage kommenden Probanden abgedeckt. Die Durchführung der Umfrage fand im Juni 2016 an der HAW Hamburg statt. Die Befragten hatten eine Bearbeitungszeit von 10 Minuten.

#### 3.1.1.1 Aufbau

Am Anfang des Fragebogens wurden Studienfach und Fachsemester des Probanden abgefragt. In die erste Spalte des Fragebogens konnten die Probanden einen Web-Dienst eintragen, den sie im Rahmen ihres Studiums für kooperatives Arbeiten nutzen. Dies war nötig, weil die Menge an vorhandenen Web-Diensten zu groß für eine Vorauswahl ist. Einzig die Web-Dienste Owncloud und EMIL wurden als Vorauswahl angegeben, weil diese im Rahmen eines Informatik Studiums an der HAW Hamburg in der Fachrichtung Informatik genutzt werden. Zu jedem Web-Dienst wurden die gleichen Sieben Fragen gestellt, mit Einfach oder-Mehrfachauswahl. Dabei sollten folgende Eigenschaften im Nutzungsverhalten mit dem jeweiligem Web-Dienst geklärt werden:

- Häufigkeit
- Zufriedenheit
- Antrieb
- Umfeld

Des weiteren sollte das *Device* für die Nutzung des Web-Dienstes angegeben werden und ob über den *Browser* oder eine Applikation auf den Web-Dienst zugegriffen wird. Zum Schluss sollte angegeben werden, ob der Befragte Zugriffsrechte an Drittanbieter für den jeweiligen Web-Dienst freigeben würde. Diese Frage durfte auch mit "Weiß nicht" beantwortet werden, weil die Folgen auf die Antwort für jeden Web-Dienst andere sind, je nach dem wie sensibel die Daten des Web-Dienstes für den Befragten sind.

#### 3.1.1.2 Auswertung

Im folgenden Abschnitt werden zunächst allgemeine Ergebnisse der Auswertung vorgestellt. Diese sind unbedeutend für das Ziel der Umfrage (siehe 3.1.1), liefern aber interessante Informationen für zukünftige Arbeiten. In Abbildung 3.2 wird das Fachsemester des Studierenden mit der Anzahl der von ihm genutzten Web-Dienste gegenübergestellt. Ein Student im ersten

Semester nutzt im Schnitt Vier Web-Dienste für kooperatives Arbeiten. Bis zum Fünften Fachsemester steigt diese Anzahl auf ungefähr Sieben an. In den Semestern Sechs und Sieben sind es um die Sechs Web-Dienste. Die aufgeführten Beobachtungen und das Ergebnis der Korrelationsformel zeigen eine Korrelation zwischen Fachsemester und Anzahl der Web-Dienste. Je höher das Fachsemester desto höher die Anzahl der genutzten Web-Dienste. Ein Grund für diese Kausalität könnten die sich in jedem Semester ändernden Mitglieder für Gruppenarbeiten sein. Web-Dienste werden untereinander besprochen und schlussendlich genutzt. Der Abfall von genutzten Web-Diensten im Sechsten und Siebten Semester kann mit dem reduzierten Aufkommen von Gruppenaufgaben erklärt werden. Aufgrund von Bachelorarbeiten belegen die Studenten weniger Kurse und haben im Folge dessen eine geringere Anzahl an kooperativen Arbeiten, bei denen auf Web-Dienste zurückgegriffen werden müssten.

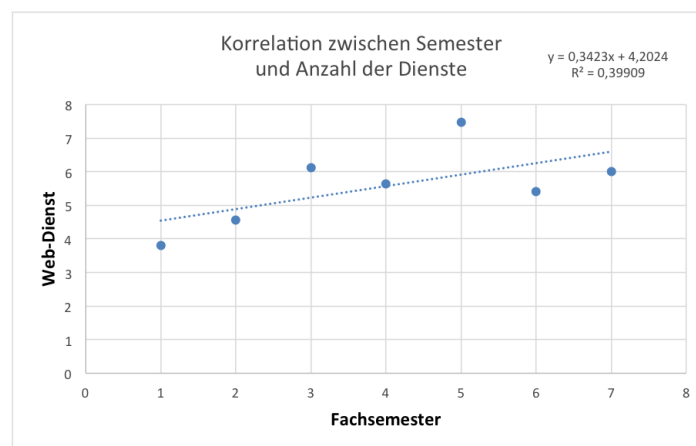


Abbildung 3.2: Korrelation zwischen Fachsemester und Anzahl der genutzten Web-Dienste

Wie in 3.1.1.1 erklärt, sollte jeder Befragte die von ihm genutzten Web-Dienste für kooperatives Arbeiten auflisten. In 3.4 sind alle angegebenen Web-Dienst und die zugehörige Anzahl der Nutzer zu sehen. Insgesamt wurden 23 verschiedene Web-Dienste für kooperatives Arbeiten angegeben. Die beiden von der HAW Hamburg vorgegeben Web-Dienste, Owncloud und EMIL, werden von 97% beziehungsweise 92% der Informatik Studenten genutzt. TODO weiter ausführen viellt

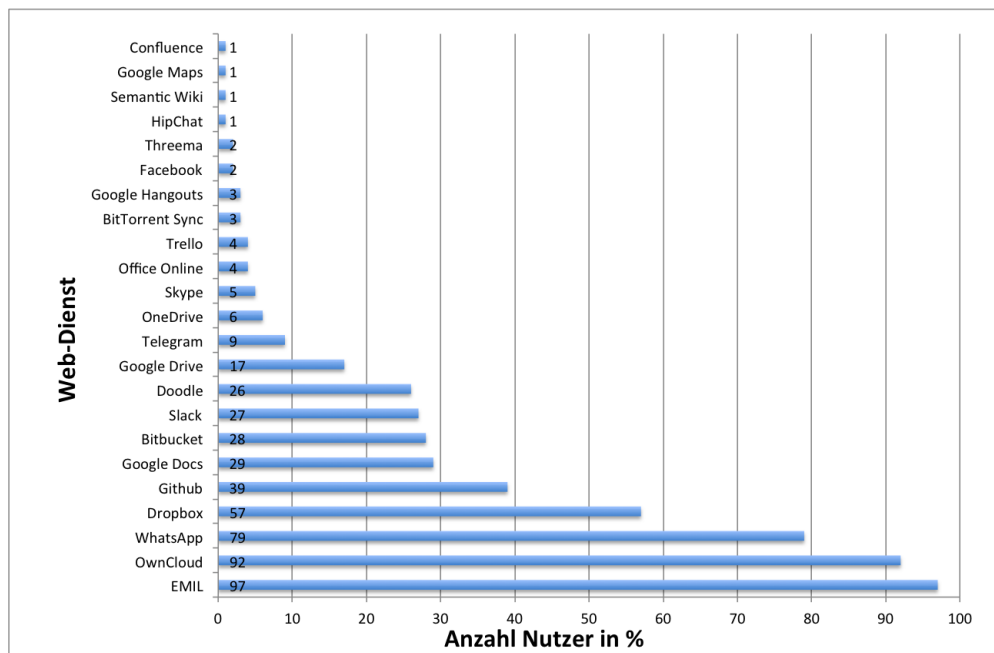


Abbildung 3.3: Liste der Web-Dienste und die Anzahl der Nutzer in Prozent

Um eine bessere Übersicht zu schaffen können die Web-Dienste in Kategorien eingeteilt werden.

- *Team Collaboration*: Dienste für die Zusammenarbeit und Organisation innerhalb eines Teams.
- *Cloud Storage*: Speichern und teilen von Dateien über das Internet.
- *Instant-Messaging*: Nachrichtenaustausch in Echtzeit.
- *Versionsverwaltung*: Verwaltung von Änderungen an Dateien.
- *Document Collaboration*: Gleichzeitige Bearbeitung von Dokumenten in Gruppen.

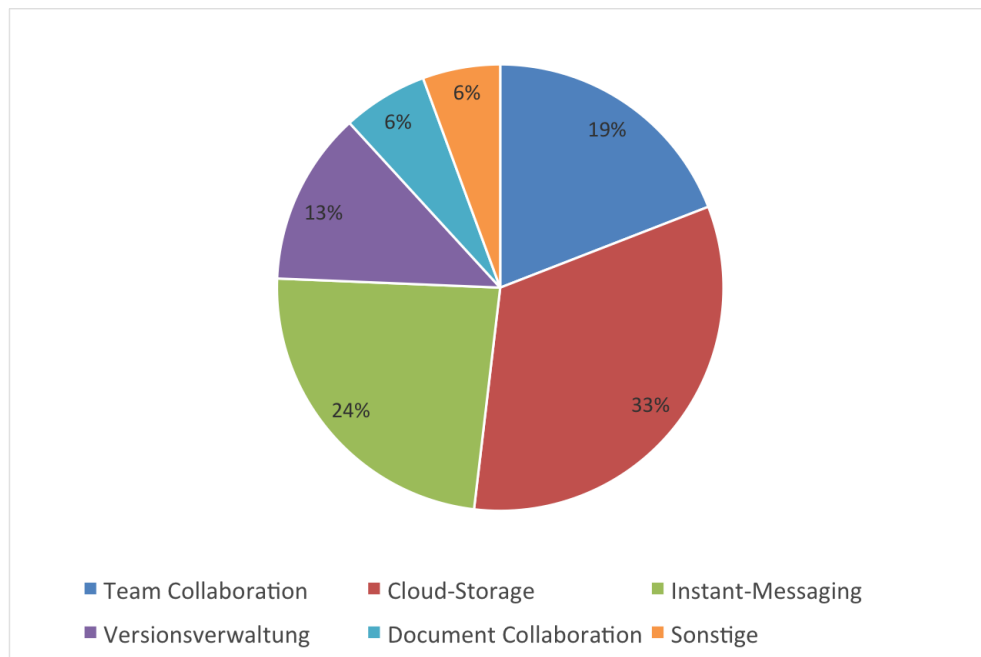


Abbildung 3.4: Verteilung der Dienste auf die Dienstkategorien

Grafik Nutzer : Dienst Grafik Dienst : Dienstgruppen Grafik Zufriedenheit : Anzahl Nutzer  
Korrelation

- Ziel der Umfrage - Dienstmenge für Middleware definieren
- Wo fand die Umfrage statt
- Wer wurde befragt und warum
- Wie wurde befragt - im Kurs persönlich 10 min zeit
- Aufbau des Umfragebogens
- Interessante Ergebnisse die nur ansatzweise mit der BA zu tun haben
- Korrelationen zwischen Daten - Korrelationen mit Quelle kurz erklären
- Dienstkategorisierung und Ergebnisse der einzelnen Kategorien
- ABC Einteilung der Dienste
- Welche Dienste sind ausgewählt worden, und warum

#### 3.1.2 Funktionale Anforderungen

- Anforderungen an die einzelnen Dienste, nur Grundfunktionen zeitlich möglich
- Anforderungen an ein kooperatives Element(User,Team,Notifications)
- Beschreibung das die Anforderungen am Ende zur Validierung des Ergebnisses herangezogen werden



### 3.1.3 Nichtfunktionale Anforderungen

- Plattformunabhängige Lösung - dadurch flexible Reaktion auf neue Technologie Entwicklungen
- TODO

## 3.2 Technologie

### 3.2.1 Übersicht

- Entwicklungsumgebung-Sprache-Middleware: Node und Java -Kommunikation: grpc,rabbitmq,soap,rest
- Datenformat: JSON,XML,PROTOBUF

### 3.2.2 Auswahl

Node grpc,rest JSON,PROTOBUF

## 3.3 Analyse und Design

UML Gateway Pattern.

Sicherheit wird aus Zeitgründen erstmal zurückgestellt Plattformunabhängigkeit sehr groß geschrieben

## 3.4 Architektur

Monolithisch oder Microservice. Vor und nachteile aufzeigen Das Zusammenspiel von Komposition und Microservice Architektur aufzeigen Reaktive oder Proaktive Komposition Koordination der einzelnen Microservices(Koordinator)

Komponentendiagramm. Hier oder Analyse Design?

## **4 Implementation**

### **4.1 Besondere Herausforderungen**

## **5 Anwendung der Middleware**

### **5.1 Validierung der Anwendungsfälle**

### **5.2 Stärken und Schwächen der Implementierung**

## **6 Fazit**

### **6.1 Entwicklungsstand**

### **6.2 Ausblick auf zukünftige Entwicklungen**

# Abkürzungsverzeichnis

**SOAP** Simple Object Access Protocol

**REST** Representational State Transfer

**AMQP** Advanced Message Queuing Protocol

**GRPC** Google Remote Procedure Call

# Literaturverzeichnis

- [1] L. Wang, J. Tao, M. Kunze, A. C. Castellanos, D. Kramer, and W. Karl, "Scientific cloud computing: Early definition and experience." in *HPCC*, vol. 8, 2008, pp. 825–830.
- [2] B. A. Nardi, S. Whittaker, and E. Bradner, "Interaction and outeraction: instant messaging in action," in *Proceedings of the 2000 ACM conference on Computer supported cooperative work*. ACM, 2000, pp. 79–88.
- [3] B. Collins-Sussman, B. Fitzpatrick, and M. Pilato, *Version control with subversion*. O'Reilly Media, Inc., 2004.
- [4] M.-S. E. Scale, "Cloud computing and collaboration," *Library Hi Tech News*, vol. 26, no. 9, pp. 10–13, 2009.
- [5] P. Bianco, R. Kotermanski, and P. F. Merson, "Evaluating a service-oriented architecture," 2007.
- [6] S. Jones, "Toward an acceptable definition of service [service-oriented architecture]," *IEEE software*, vol. 22, no. 3, pp. 87–93, 2005.
- [7] M. P. Papazoglou, "Service-oriented computing: Concepts, characteristics and directions," in *Web Information Systems Engineering, 2003. WISE 2003. Proceedings of the Fourth International Conference on*. IEEE, 2003, pp. 3–12.
- [8] W3C. (2016) Soap version 1.2 part 1: Messaging framework (second edition). [Online]. Available: <https://www.w3.org/TR/soap12-part1>
- [9] C. Pautasso, O. Zimmermann, and F. Leymann, "Restful web services vs. big'web services: making the right architectural decision," in *Proceedings of the 17th international conference on World Wide Web*. ACM, 2008, pp. 805–814.
- [10] S. Vinoski, "Advanced message queuing protocol," *IEEE Internet Computing*, vol. 10, no. 6, p. 87, 2006.

- [11] G. Inc. (2016) A high performance, open-source universal rpc framework. [Online]. Available: <http://www.grpc.io>
- [12] M. Reichert and D. Stoll, "Komposition, choreographie und orchestrierung von web services: Ein überblick," in *EMISA Forum*, vol. 24, no. 2, 2004, pp. 21–32.
- [13] H. Hamburg. (2016) Studierende der haw hamburg sommersemester 2016. [Online]. Available: [https://www.haw-hamburg.de/fileadmin/user\\_upload/Presse\\_und\\_Kommunikation/Downloads/C\\_1\\_Web\\_SoSe\\_16\\_Studierende\\_gesamt\\_20160617.pdf](https://www.haw-hamburg.de/fileadmin/user_upload/Presse_und_Kommunikation/Downloads/C_1_Web_SoSe_16_Studierende_gesamt_20160617.pdf)

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 15. Juni 2016   Philipp Prögel