

Institut für Architektur von Anwendungssystemen

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3448

**Ein Entscheidungsunterstüt-
zungssystem für die Migration
bestehender Anwendungen in die
Cloud**

Andreas Gross

Studiengang:	Informatik
Prüfer:	Prof. Dr. Frank Leymann
Betreuer:	Dipl.-Inf. Florian Haupt
Beginn am:	24. Januar 2013
Beendet am:	26. Juli 2013
CR-Klassifikation:	H.3.5, H.4.2, H.5.2

Kurzfassung

Cloud Computing hat ein großes Potential für viele Unternehmen. Um die Chancen wahrzunehmen müssen sowohl die Potentiale als auch Risiken betrachtet und abgewogen werden. Vor allem wenn bestehende Anwendungen *in die Cloud* migriert werden sollen kann dieser Entscheidungsprozess einen nicht unerheblichen Aufwand bedeuten. Neben den unterschiedlichen Methoden zur Entscheidungsunterstützung wird in dieser Arbeit ein bestehendes Vorgehensmodell analysiert und eine Implementierung dazu entworfen. Das als Webanwendung realisierte Entscheidungsunterstützungstool erlaubt die Erfassung, Organisation, Bewertung und Darstellung von Komponenten in einem Migrationsszenario. Dem Anwender des Werkzeugs wird damit eine Hilfestellung zur Evaluation der Entscheidung bereitgestellt um die Fragestellung zur Migration von bestehenden Anwendungen nachvollziehbar beantworten zu können.

Das MIGRATE! Vorgehensmodell bildet die Grundlage des Entscheidungsunterstützungsprozesses. Um dieses Modell umzusetzen werden zunächst die funktionalen und nicht-funktionalen Anforderungen erhoben. Dabei werden alle relevanten Punkte, wie beispielsweise die Modellierung, Projektverwaltung, Gewichtung, Darstellung, Erweiterbarkeit und Navigation berücksichtigt. Mit Hilfe von Mockups wird ein grobes Layout des User Interface entworfen. Das Werkzeug soll als moderne Webanwendung realisiert werden. Aus den einzelnen Objekten im Vorgehensmodell bilden sich die Klassen für das Datenmodell. Die Abhängigkeiten werden modelliert und der Prozess im Vorgehensmodell stellt den Ablauf der Webanwendung dar. Zur Erleichterung der Informationserfassung kann ein bestehendes Modell der Infrastruktur importiert werden. Zudem steht dem Anwender ein kleines Testszenario zur Verfügung.

Abkürzungsverzeichnis

IaaS	Infrastructure as a Service
PaaS	Platform as a Service
SaaS	Software as a Service
XaaS	Everything as a Service
CSP	Cloud Service Provider
NIST	National Institute of Standards and Technology
EUS	Entscheidungsunterstützungssystem
IT	Informationstechnologie
BITKOM	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.
STS	Spring Tool Suite
UML	Unified Modeling Language
JPA	Java Persistence API
RMS	Risikomanagement-Standard
SMI	Service Measurement Index
KPI	Key Performance Indicator
ISO	International Organization for Standardization
QoS	Quality of Service
AHP	Analytic Hierarchy Process
ITIL	IT Infrastructure Library
TCO	Total Cost of Ownership
EC2	Elastic Compute Cloud
RDS	Relational Database Service
S3	Simple Storage Service
AWS	Amazon Web Services
HPC	High Performance Computing
IKT	Informations- und Kommunikationstechnik
RPC	Remote-Procedure-Call
SOA	Service Oriented Architecture

CRM	Customer-Relationship-Management
SPI	Software-Platform-Infrastructure
RePro	Reengineering Prozesse
GUI	Graphical User Interface
NFA	Nicht-funktionale Anforderungen
FA	Funktionale Anforderungen
(MC ²) ²	Multi-Criteria Comparison Method for Cloud Computing
PDF	Portable Document Format
MDS	Migration Decision Support
JRE	Java Runtime Environment
JDK	Java Development Kit
IDE	Integrated Development Environment
MVC	Model View Controller
CSS	Cascading Style Sheets
JS	JavaScript
JSP	Java Server Pages
XML	Extensible Markup Language
POJO	Plain Old Java Object
EE	Enterprise Edition
JDBC	Java Database Connectivity
CRUD	Create, Read, Update, Delete
DAO	Data Access Object
EMF	Eclipse Modeling Framework
EAM	Enterprise Architecture Management

Inhaltsverzeichnis

Abkürzungsverzeichnis	5
1 Einleitung	11
1.1 Motivation	11
1.2 Ziel der Arbeit	12
1.3 Abgrenzung	13
1.4 Das MIGRATE! Projekt	14
1.5 Gliederung	15
2 Grundlagen	17
2.1 Grundbegriffe	17
2.2 Cloud Computing	18
2.2.1 Wichtige Charakteristiken	19
2.2.2 Diensttypen	20
2.2.3 Bereitstellungstypen	20
2.2.4 Das Software-Platform-Infrastructure (SPI) Modell	21
2.3 Migration von Anwendungen	21
2.4 Entscheidungsunterstützungssysteme	23
3 Stand der Technik	27
3.1 Migration in die Cloud	27
3.2 Entscheidungsunterstützung	27
3.2.1 Frameworks	28
3.2.2 Service Measurement Index (SMI) Framework	28
3.2.3 IT Infrastructure Library (ITIL) Framework	29
3.2.4 Risikoabschätzung	31
3.3 Migrations-Entscheidungsunterstützung	32
3.3.1 Multi-Criteria Comparison Method for Cloud Computing ((MC ²) ²)	32
3.3.2 Cloud Adoption Toolkit	33
3.3.3 Federal Cloud Computing Strategy	34
3.4 Angebote der Cloud Service Provider	35
3.4.1 Amazon	36
3.4.2 Microsoft Azure	36
3.4.3 Hewlett-Packard	37

4	Anforderungsanalyse und Entwurf	39
4.1	Funktionale Anforderungen (FA)	39
4.2	Nicht-funktionale Anforderungen (NFA)	45
4.3	Konzeptioneller Entwurf	46
4.3.1	Mockups	47
4.4	Datenmodell (Domain Model)	57
4.4.1	Der Schritt	58
4.4.2	Das Kriterium	59
4.4.3	Die Ausschließung	60
4.4.4	Die Beschreibung	61
4.5	Einbindung des bestehenden Modells	63
5	Implementierung des Tools	65
5.1	Prototypische Implementierung	65
5.1.1	Entwicklungsumgebung	69
5.1.2	Das Spring Framework	69
5.1.3	Aufbau der Webanwendung	71
5.2	Ein Testszenario	73
6	Zusammenfassung und Ausblick	75
6.1	Zusammenfassung	75
6.2	Ausblick	75
	Literaturverzeichnis	77

Abbildungsverzeichnis

1.1	Statistik zum Suchinteresse an „Cloud Migration“ (April 2013)	12
1.2	Möglichkeiten der Migration von verschiedenen Schichten und Cloud Bereitstellungstypen	13
1.3	Beschreibung und Partner des MIGRATE! Projekts	14
2.1	Aufteilung einer Anwendung in drei Schichten	17
2.2	Das Service Oriented Architecture (SOA) Dreieck	18
2.3	Die Dimensionen des SPI Modells	22
2.4	Entscheidungsprozess mit Hilfe eines EUS	25
3.1	Eine Anwendung schichtweise in die Cloud migrieren	28
3.2	ITIL-Entscheidungen, nach Situationsmatrix von Robrecht	30
3.3	Prozess des Risikomanagement nach ISO 31000	31
3.4	Ablaufprozess im (MC ²) ² Framework	33
3.5	Vorgehen des Cloud Adoption Toolkit	34
3.6	Federal Cloud Computing Stragegy Framework für Entscheidungen zur Cloud Migration	35
3.7	Aufbau des Migrationsleitfadens von Microsoft	37
4.1	Mockup: Grundgerüst	48
4.2	Mockup: Hauptseite	50
4.3	Mockup: Arbeitsbereich	51
4.4	Mockup: Projektübersicht	53
4.5	Mockup: Projektauftrag	54
4.6	Mockup: Administrationsbereich	55
4.7	Mockup: Benutzerkonto	56
4.8	Domain Model der Webanwendung	57
4.9	Schritte im Migrations-Prozess (Step)	58
4.10	Modell der Kriterien (Criteria)	59
4.11	Modell der Ausschließung (Exclusion)	60
4.12	Model der Beschreibung (Description)	61
4.13	Beschreibung der Objektmodelle als Ecore Diagramm	63
4.14	Beispieltopologie aus dem Enterprise Architecture Management (EAM)-Modell	64
5.1	Modell der Komponente (Component)	67
5.2	Übersicht des Spring Framework, verfügbare Module	70
5.3	Ablauf einer Anfrage im Spring Web MVC Modul	72

5.4	Aufbau des Testszenarios	73
-----	------------------------------------	----

Tabellenverzeichnis

4.1	Übersicht der Anforderungen	40
-----	---------------------------------------	----

Verzeichnis der Listings

5.1	Aufbau der Komponenten-Klasse	68
5.2	Datenbankinterface der Komponenten-Klasse	68
5.3	Abhängigkeit in einer pom.xml Datei	71
5.4	DispatcherServlet in der web.xml Datei	72

Einleitung

Das Einführungskapitel erklärt den Inhalt der Arbeit und zeigt die aktuelle Bedeutung des Themas. Mit der Abgrenzung wird der Rahmen der Arbeit beschrieben. Als Hintergrund wird das MIGRATE! Projekt kurz erläutert und ein Zusammenhang hergestellt. Die Gliederung schließt die Einleitung ab.

1.1 Motivation

In vielen Unternehmen, die eine ausgebaute Infrastruktur ihrer Informationstechnologie (IT) besitzen, stellt sich nach und nach die Frage der Umsetzung von Green IT [Muro8] [MCP09]. Sowohl im privaten Bereich als auch im Geschäftsumfeld ist die Bedeutsamkeit von umweltbewusstem Umgang ein anerkanntes Thema. Unabhängig von der Firmengröße können verschiedene Facetten ein Beweggrund zur Unterstützung der Nachhaltigkeit im IT-Bereich sein [KW10]:

- **Effizienter Umgang mit Abfall**, gefährlichen Substanzen und Energie im Produktlebenszyklus
- **Umdenken bei Energieverbrauch**, nachhaltigen Energiequellen und höheren Wirkungsgrad
- **Effizienzsteigerung** in anderen Bereichen mit Einsatz von Green IT

Um diese Aufgabenstellungen zu lösen kann der Einsatz der Cloud-Architektur ein wichtiger Faktor bei der Umsetzung sein. Anstatt eine eigene ausgebaute Infrastruktur zu verwalten bietet es sich an, die entsprechenden Dienste, Anwendungen und Prozesse *in der Cloud* bereit zu stellen. Die Aufteilung der Ressourcen, ein dynamischer Leistungsabruf und damit optimierte Kosten sprechen für eine Überlegung zur Migration von bestehenden Anwendungen in die Cloud. Die ökologischen Vorteile bei Nutzung der Cloud-Architektur sind anerkannt [KW10] [Hö11] und helfen den Unternehmen dabei ihre Green IT Vorhaben umzusetzen.

1 Einleitung

Mit diesem Vorhaben ergeben sich aber auch Risiken und Aufwand für die Umstellung. Eine wichtige Fragestellung ist deshalb, ob die Migration von Anwendungen in die Cloud lohnenswert ist. Da dies ohne Hilfestellung sehr aufwändig ist, wird, ähnlich wie in anderen Unternehmensbereichen, die Benutzung eines Entscheidungsunterstützungssystems vorgezogen. Damit soll die Auswahl zwischen den Möglichkeiten erleichtert werden und eine nachvollziehbare Begründung für die Entscheidung zur Verfügung stehen.

Ein weiteres Indiz für die Wichtigkeit des Themas zur Migration von Anwendungen in die Cloud zeigt die folgende Statistik (Abbildung 1.1) aus Google Insights:

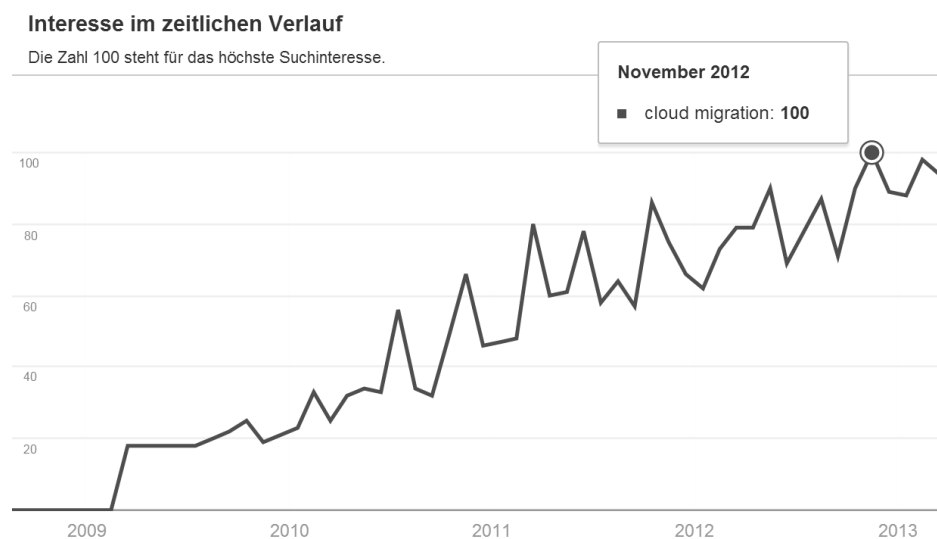


Abbildung 1.1: Statistik zum Suchinteresse an „Cloud Migration“ (April 2013)

Das Interesse an dem Vorhaben hat sich demnach seit Anfang des Jahres 2009 entwickelt. Ein Wachstumspotential [PM10] ist aber weiterhin gegeben, die Potentiale und Möglichkeiten des Cloud Computing werden erst in Zukunft voll ausgeschöpft werden können.

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es ein Werkzeug zu erstellen, welches Entscheidungen im Bereich der Cloud Migration unterstützen kann. Dieses Werkzeug ermöglicht die Identifikation und Evaluation von Migrationsvorhaben in eine Cloud Infrastruktur. Detaillierter ergeben sich folgende Ziele, die im Laufe der Arbeit erreicht werden:

- **Konzeptioneller Entwurf** des Entscheidungsunterstützungssystems (EUS)
- **Analyse und Auswertung** der Anforderungen
- **Entwurf eines Datenmodells** für das MIGRATE! Projekt

Das mit dieser Arbeit entstehende Entscheidungsunterstützungstool ermöglicht mit Hilfe von Komponenten, Szenarien und Gewichtungen eine Auswahl zu treffen. Die Trennung der Anwendung und wie später die Migration ausgeführt wird ist aber nicht Teil des Umfangs.

1.4 Das MIGRATE! Projekt

Das Resultat dieser Arbeit ist ein Teil der Ziele des MIGRATE! Projekts¹. Das Projekts befasst sich mit dem Thema Cloud Computing und der Energieeffizienz im Bereich der Informations- und Kommunikationstechnik (IKT). Durch das Modell des Cloud Computing und den vorteilhaften Eigenschaften (siehe Kapitel 2.2) ergeben sich hohe Potentiale, unter anderem für Green IT, in Unternehmen. Die Überführung von bestehenden IKT-Anwendungen in die Cloud ist aber nicht trivial und erfordert neue Verfahren und Werkzeuge für die Cloud-Migration. Das Projekt wird von dem Bundesministerium für Wirtschaft und Technologie gefördert und wird von mehreren Partnern unterstützt:



Abbildung 1.3: Beschreibung und Partner des MIGRATE! Projekts

Die Ziele des MIGRATE! Projekts wurden wie folgt festgelegt:

1. **Energieeffizienz durch Cloud-Migration:** Erreichen einer Steigerung der Energieeffizienz durch Migration von Anwendungen in die Cloud. Dabei werden die Geschäftsprozesszusammenhänge und demografische Fragestellungen adressiert.
2. **Entscheidungsunterstützung für Migrationen:** Durch Modelle und Methoden soll eine Bewertung potentieller Energieeinsparungen bei Migration zu Cloud-basierten Infrastrukturen ermöglicht werden. Dazu werden unter anderem Energieverbrauch, Geschäftsprozesse und Benutzerverhalten erfasst, Bewertungsmodelle entwickelt und parametrisiert. Die Energieeffizienz- und Wirtschaftlichkeitsberechnungen werden auf Basis von Simulationen durchgeführt.
3. **Cloud Migrationsprozesse:** Anstatt nur die technischen Aspekte zu betrachten werden die Wechselwirkungen zwischen Mensch und Technik durch die Veränderung der *Cloudifizierung* betrachtet. Zu den technischen Lösungen kommen sozio-technische und ökonomische Methoden und Werkzeuge zum Einsatz.

¹<http://www.migrate-it2green.de/>

4. **Energieoptimiertes Cloud-Management:** Für die Energieverbrauchsoptimierung werden organisationsübergreifende Lösungsansätze benötigt. Die Informationen über das Nutzerverhalten, die durch Migrationsprojekte gewonnen werden, werden durch Modelle auch in der Betriebsphase wiederverwendet.

Aus der Aufgabe zur *Entscheidungsunterstützung für Migrationen* resultiert ein Vorgehensmodell für Migrationsprojekte. Das im Rahmen dieser Arbeit entwickelte Entscheidungsunterstützungstool verwendet dieses Vorgehensmodell als Grundlage und ermöglicht eine Bewertung der Migrationsszenarien durch den Benutzer.

Das im Rahmen dieser Arbeit genannte Vorgehensmodell liegt vor, ist jedoch zum Zeitpunkt der Entwicklung noch nicht veröffentlicht worden. Die Annahmen hierzu basieren auf der Dokumentation in folgender Referenz:

Verbundprojekt MIGRATE!, „Modelle, Verfahren und Werkzeuge für die Migration in Cloud-basierte energieoptimierte Anwenderinfrastrukturen und deren Management“, Arbeitspaket 3 „Migrationsverfahren und -werkzeuge“,

Meilensteinbericht 3.1 vom 30.06.2013

<http://www.migrate-it2green.de/>

1.5 Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: Beschreibung der grundlegenden Begriffe dieser Arbeit. Es werden die drei Hauptthemen, Cloud Computing, Migration von Anwendungen und Entscheidungsunterstützungssysteme, kurz vorgestellt und die notwendigen Begriffe definiert.

Kapitel 3 – Stand der Technik: Ein Überblick zu der aktuellen Vorgehensweise bei Migration von Anwendungen in die Cloud. Bereits vorhandene Entscheidungsunterstützungsverfahren werden betrachtet und verglichen.

Kapitel 4 – Anforderungsanalyse und Entwurf: Analyse der Anforderungen und darauf aufbauend ein Entwurf des Tools und dem darunterliegenden Datenmodell.

Kapitel 5 – Implementierung des Tools: Die prototypische Implementierung wird in diesem Kapitel begleitet, das Vorgehen erläutert und die wichtigsten Elemente demonstriert.

Kapitel 6 – Zusammenfassung und Ausblick: Die Zusammenfassung der Arbeitsergebnisse und Vorstellung einer möglichen zukünftigen Entwicklung.

Grundlagen

Das zweite Kapitel beschreibt die Grundlagen die notwendig sind, um die Zusammenhänge in dieser Arbeit zu verstehen. Es wird zunächst auf die Grundelemente einer Anwendung eingegangen. Daraus entwickelte sich im Laufe der Zeit eine Architektur, die Cloud Computing ermöglicht. Durch diesen Fortschritt kommt es zu dem Gedanken, wie bestehende Anwendungen in die Cloud migriert werden können. Um diese anspruchsvolle Frage zu klären ist der Einsatz eines Entscheidungsunterstützungssystems hilfreich.

2.1 Grundbegriffe

Die meisten Anwendungen sind heutzutage nach dem OSI-Referenzmodell [Zim80] aufgebaut. Dieses Modell teilt die Anwendung in verschiedene Schichten auf, über die miteinander kommuniziert werden kann. Die einzelnen Schichten haben klar definierte Aufgaben [HL95]. Damit erreicht man eine bessere Modularisierung der Anwendung und somit bessere Wartbarkeit. Die Kommunikationsprotokolle zwischen den Schichten sind austauschbar.

Nach dem OSI-Referenzmodell sind sieben Schichten definiert. Diese kann man weiter zusammenfassen, so dass man folgende Aufteilung erhält:

- **Darstellungsschicht** Schnittstelle zum Anwender (User Interface)
- **Anwendungsschicht** Enthält die Logik, die ablaufenden Geschäftsprozesse, Workflow, Entities
- **Datenschicht** Ermöglicht Zugriff auf die Daten und Services

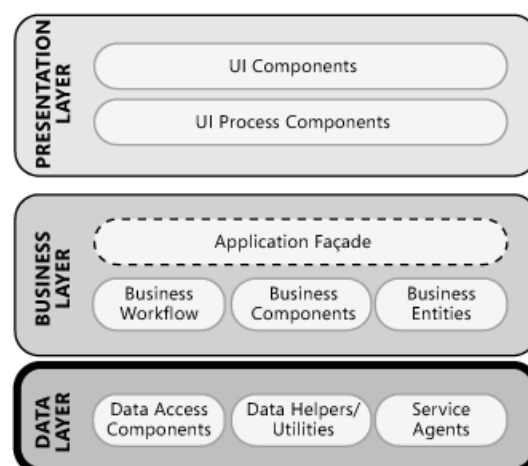


Abbildung 2.1: Aufteilung einer Anwendung in drei Schichten

Dieses Modell wird unter anderem von Microsoft [Mico9] (siehe Abbildung 2.1) und anderen großen IT-Firmen verwendet, um die Anwendungsentwicklung zu vereinfachen. Durch diese Abstraktion ist die Anwendungsentwicklung oder Anpassung bestehender Anwendungen deutlich einfacher.

Ein wichtiger Schritt hin zum Cloud Computing war die Einführung der Service Oriented Architecture (SOA) [Mel10]. Ziel war es, ein abstraktes Konzept für Anwendungen bereit zu stellen und die Dienste über ein Netzwerk anzusprechen. Einzelne Funktionen werden nicht direkt, etwa über einen Remote-Procedure-Call (RPC) aufgerufen, sondern es wird ein Dienst über eine bestimmte Schnittstelle angefragt. Um dies zu realisieren werden die folgenden Rollen definiert:

- **Verzeichnis** Stellt eine Sammlung an Diensten zur Verfügung, erlaubt das Eintragen und Finden von Diensten
- **Kunde** Sucht nach einem Dienst im Verzeichnis und verwendet ihn dann
- **Anbieter** Stellt einen Dienst bereit, der zuvor im Verzeichnis registriert wurde

Die folgende Abbildung 2.2 demonstriert das Zusammenspiel der drei Rollen im sogenannten SOA Dreieck. Das Verzeichnis (Service Broker) dient als Vermittler zwischen dem Kunden (Requester) und Anbieter (Provider).

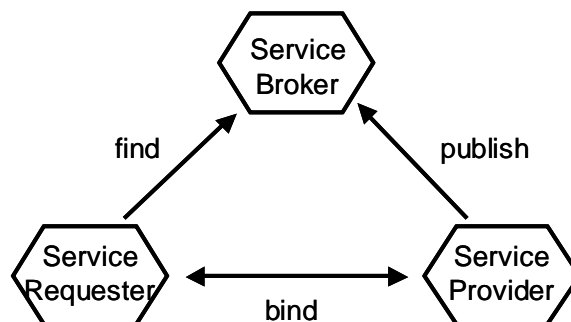


Abbildung 2.2: Das SOA Dreieck

2.2 Cloud Computing

Cloud Computing hat in letzter Zeit sehr viel Aufmerksamkeit erzeugt [SSW10]. Mehrere Analysten, Dienstleister und Experten haben einen deutlichen positiven Ausblick für diesen Architekturstil vorhergesagt. Die Ursprungsidee wurde aber bereits vor über 50 Jahren von J.C.R. Licklider [Lic63] vorgestellt, einem *Intergalactic Computer Network*. Die Technik hinter diesem Architekturstil gibt es also schon seit längerem. Es war aber die Evolution von bereits vorhandenen Technologien [Hö11] wie *Network-Computing*, *Server-Cluster* oder *Grid Computing*. Zusammen mit dem Trend zu *Utility-Computing*, also dass Rechenleistung nach

Nutzung abgerufen und bezahlt werden kann, den *Service Oriented Architectures (SOA)* und *Ubiquitous Computing* hat sich mit der Zeit die Cloud-Architektur herauskristallisiert. Erst die Nutzung dieser verschiedenen Technologien und der Akzeptanz der Anwender führte zu dem Durchbruch und aktuell steigendem Interesse an diesem Themengebiet.

Es gibt eine Vielzahl an Definitionen zu dem Cloud-Begriff [VRMCLo8]. Viele Unternehmen bedienen sich eigener Definitionen und Merkmalen, um ihre eigenen Produkte rund um die Cloud besser vermarkten zu können. Eine häufig angewandte Referenz ist die Definition des *National Institute of Standards and Technology (NIST)*. In dem offiziellen Dokument werden die Charakteristiken, Dienst- und Bereitstellungstypen beschrieben [MG11]. Laut der Beschreibung soll es dem Kunden ermöglicht werden, Zugriff auf eine Auswahl an konfigurierbaren Rechnerressourcen (wie Netzwerk, Server, Speicher, Anwendungen und Dienste) vorzunehmen. Der Zugriff wird dabei mit den Eigenschaften *ubiquitous, convenient, on-demand* beschrieben. Die Konfiguration der Ressourcen soll flexibel und mit minimalem Aufwand bereitgestellt werden können, die Abrechnung erfolgt auf Basis der tatsächlichen Nutzung. Die folgenden Auflistungen beschreiben die Definition des Cloud Computing. Die Begriffe werden dabei mit dem Englischen Bezeichner angegeben, um die Bedeutung nicht zu verfälschen:

2.2.1 Wichtige Charakteristiken

Grundlegende Eigenschaften, die eine Cloud-Infrastruktur erfüllen soll, ergeben die Fünf Charakteristiken des Cloud Computing.

On-demand self-service Die Dienste und Ressourcen werden automatisiert zur Verfügung gestellt. Der Kunde kann selbstständig und direkt über die bereitgestellte Leistung (wie Rechengeschwindigkeit, Speicherplatz, ...) verfügen, ohne den Zwischenschritt der Kommunikation mit einem Mitarbeiter des Anbieters.

Broad network access Der Zugriff auf die Leistung ist mit verschiedenen Plattformen über ein beliebiges Netzwerk möglich.

Resource pooling Die physischen und virtuellen Rechenressourcen werden gemeinsam verwendet und dynamisch bei Bedarf zugewiesen. Durch die Lokationsunabhängigkeit wird eine weitere Abstraktionsebene hinzugefügt.

Rapid elasticity Skalierbarkeit wird durch schnelle Ressourcenanpassung ermöglicht, die der Provider zur Verfügung stellt

Measured service Ressourcen können von dem Anbieter und Benutzer überwacht, gesteuert und analysiert werden. Je nach Dienstart ist eine nutzungsabhängige Abrechnung und damit eine bessere Überprüfung möglich.

2.2.2 Diensttypen

Die Diensttypen beschreiben aus welchem Umfang die angebotenen Ressourcen bestehen und wie weit darauf zugegriffen werden kann. Alle Modelle haben gemeinsam, dass die darunterliegende Cloud-Infrastruktur von dem Anbieter verwaltet und kontrolliert wird. Der Kunde hat je nach Modell aber unterschiedliche Möglichkeiten diese Infrastruktur zu nutzen.

Software as a Service (SaaS) Die zur Verfügung gestellten Ressourcen sind die Anwendungen, die bei dem Anbieter ausgeführt werden. Die Anwendungen können von verschiedenen Endgeräten oder anderen Programmen aufgerufen werden. Dabei wird auf Anbieterseite eine Infrastruktur verwendet, welche die zuletzt genannten Charakteristiken ermöglicht. Der Kunde steuert aber nicht die Infrastruktur (wie Netzwerk, Server, Speicher, ...) selbst, es wird lediglich die Anwendung an sich zur Verfügung gestellt.

Platform as a Service (PaaS) Die zur Verfügung gestellten Ressourcen entsprechen einer Plattform, welche der Kunde nutzen kann um eigene Anwendungen in der Cloud Infrastruktur auszuführen. Dabei werden vom Anbieter unterstützte Werkzeuge verwendet und der Kunde hat damit zusätzliche Einstellungsmöglichkeiten für die Umgebung der Anwendung.

Infrastructure as a Service (IaaS) Die zur Verfügung gestellte Ressource ist der Zugriff auf die Rechenleistung, Speicherplatz, Netzwerk und andere grundlegende Rechnerressourcen. Der Kunde hat Zugriff auf diese Komponenten und das Betriebssystem, so dass eigene Anwendungen eingerichtet und ausgeführt werden können.

Eine Fortsetzung dieser Modelle ist die Idee, die Möglichkeiten des Cloud Computing für mehrschichtige Dienste zu verwenden. Unter dem Begriff **Everything as a Service (XaaS)** wird eine Zusammenstellung mehrerer Dienste ermöglicht, so dass zum Beispiel virtualisierte physische Ressourcen, Infrastruktur, Middleware Plattformen und Geschäftsanwendungen als Dienst in der Cloud angeboten werden [LKN⁺09].

2.2.3 Bereitstellungstypen

Die Bereitstellungstypen (Deployment Models) beschreiben die Zielgruppe der zur Nutzung angebotenen Dienste und welche Einrichtung die Cloud Infrastruktur bereitstellt und verwaltet.

Private Cloud Die Cloud Infrastruktur wird für eine einzelne Organisation (mehrere Benutzer) eingerichtet. Die Verwaltung übernimmt ebenfalls diese Organisation, ein anderer Anbieter oder eine Kombination hiervon.

Community Cloud Die Zielgruppe ist eine Gemeinschaft von Benutzern mit gemeinsamen Interessen.

Public Cloud Die Infrastruktur ist zur öffentlichen Verwendung eingerichtet. Der Anbieter ist eine Firma, wissenschaftliche Einrichtung, eine Regierungsbehörde oder eine Kombination hiervon.

Hybrid Cloud Eine Kombination aus den drei soeben genannten Modellen bezeichnet man als Hybrid Cloud. Die einzelnen Anbieter stehen für sich, jedoch sind die Dienste durch eine standardisierte oder proprietäre Technologie miteinander verbunden, so dass Daten und Anwendungen portiert werden können.

Wird eine Infrastruktur der *Private Cloud* oder *Community Cloud* verwendet, so können die Bereitstellungen der Dienste auf zwei unterschiedlichen Modellen beruhen:

On-Premise Das traditionelle Modell beschreibt die Bereitstellung und Einsatz vor Ort. Der Begriff stammt aus der Softwarebereitstellung und beschreibt die Nutzung einer Software und deren Lizenzierung. Im Zusammenhang mit Cloud Computing ist die Infrastruktur örtlich bei dem Anbieter der Cloud Dienste untergebracht.

Off-Premise Als Gegensatz zu der *Vor-Ort-Infrastruktur* ist die Unterbringung außerhalb des Unternehmens angebracht und kann mit einem Hosting-Betrieb verglichen werden. Dies ist das typische Verfahren bei Nutzung von Cloud Diensten, die Infrastruktur wird von einem externen Anbieter angeboten und verwaltet.

Diese zwei Modelle schließen sich nicht gegenseitig aus.

2.2.4 Das Software-Platform-Infrastructure (SPI) Modell

Die verschiedenen Cloud Service Provider (CSP) stellen unterschiedliche Dienst- und Bereitstellungstypen zur Verfügung [KV10]. Ein einfacher aber verbreiteter Dienst ist beispielsweise das Web-Hosting in der Cloud. Es wird eine Webserver-Software zur Verfügung gestellt und der Anwender kann seine Daten dort bereitstellen. Für Unternehmen wäre der Einsatz einer Customer-Relationship-Management (CRM) Software in der Cloud interessant. Nimmt man dagegen eine IaaS in Anspruch, können ganze Speicherlösungen, Rechenleistung oder Netzwerkdienste in der Cloud betrieben werden. Diese unterschiedlichen Dienstypen, Bereitstellungstypen und Anwendungsmodelle ergeben mehrdimensionale Kombinationsmöglichkeiten. Die Anwendergruppen reichen dabei von Endanwendern (SaaS) über Anwendungsentwickler (PaaS) bis hin zu Netzwerkarchitekten (IaaS). Die Abbildung 2.3 veranschaulicht diese Kombinationsmöglichkeiten in einem 3-Dimensionalen Würfel [MKL09].

2.3 Migration von Anwendungen

Eine Eigenschaft der Informationstechnologie ist der rasche Fortschritt, sowohl bei der Hardware als auch bei Software-Produkten. Bereits zu Beginn des IT-Zeitalters hat Moore eine Voraussage getroffen [M⁺65], dass die Komplexität und damit die Leistung integrierter Schaltungen exponentiell steigen wird. Um dieses zusätzliche Potential auszuschöpfen

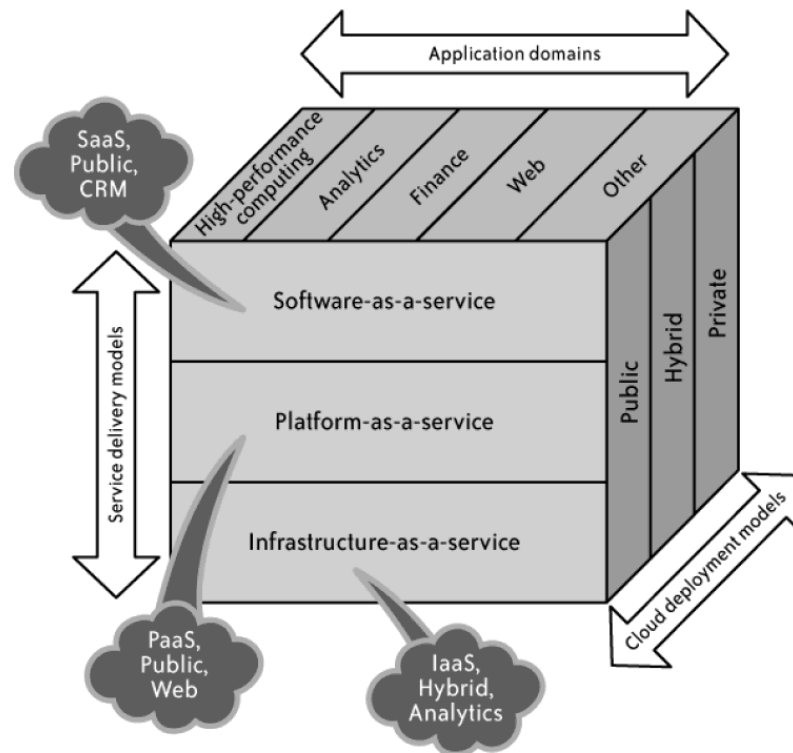


Abbildung 2.3: Die Dimensionen des SPI Modells

wird eine angepasste Software benötigt. Beispielsweise hat die Einführung der Multi-Core Architektur einen Änderungsbedarf bei der Programmierung hervorgerufen [CSM⁺05], und damit wurde eine bemerkenswerte Performance-Steigerung erzielt. Der gestiegene Energieverbrauch und die Hitzeentwicklung führten ebenfalls zu einem Umdenken bei dem Entwurf von Software [Gar10]. Damit trat der Nebeneffekt auf, die Energieeffizienz bei IT zu erhöhen.

Aufgrund der Änderungen im technischen Bereich kann aber auch eine neue Architektur, wie zuvor beschrieben das Cloud Computing, eine Funktionsänderung der Software erfordern. Gibt es neue Möglichkeiten eine bestimmte Aufgabenstellung zu lösen, wird eine Umsetzung hierzu auch neue Software verlangen. Durch den Wettbewerb bei den Software-Anwendungen, erwünschten Innovationen und neuen Anforderungen wird der Wunsch zur Erneuerung von bereits vorhandener Software verstärkt [SWH10]. Dadurch ergeben sich Fragestellungen der Art:

- Kann ich nahtlos auf die neue Hardware/Software umsteigen?
- Werden vorhandene Daten (fehlerfrei, komplett) übernommen?
- Passt die IT-Infrastruktur zu der Software?

Um die Hardware, Software oder Prozesse an diesem Fortschritt einzubeziehen ist eine Übertragung der Komponenten in ein anderes System notwendig. Diese Arbeit befasst sich mit der Migration von Anwendungen und es wird folgende Definition verwendet:

Unter einer Migration versteht man:

Die Überführung von Softwaresystemen in eine andere Zielumgebung oder eine neue Form, ohne hierbei deren Funktionalität zu ändern. [AGW05]

In unserem Fall wird die Zielumgebung nicht etwa ein virtualisiertes System oder ein Cluster-Verbund sein, die Anwendung wird in eine neue Form überführt, die das Modell des Cloud-Computing beschreibt.

Die seit 2004 jährlich stattfindende Workshop-Reihe *Reengineering Prozesse (RePro)* hat im Jahr 2005 den Schwerpunkt auf Migrationsprojekte gesetzt [RG05]. Dabei wurde die Bedeutsamkeit von Software-Migration untersucht und verschiedene Vorgehensweisen vorgestellt. Als wichtiger Auslöser von Migrationsvorhaben wurde die Änderung von Nicht-funktionalen Anforderungen, wie Weiternutzbarkeit der Software, Kosten und bessere Wartbarkeit aufgeführt. Alle Migrationsprojekte haben gemein, dass die Veränderungen nur auf dieser Ebene erfolgen und die eigentliche Logik oder Prozesse der Anwendung in der Regel nicht verändert werden.

Es gibt eine Reihe an anerkannten Migrationsstrategien [SWH10]. Diese haben eine unterschiedliche Herangehensweise in welchen Schritten die Migration abläuft, welche Teile wann migriert werden können und wie der Zugriff während des Prozesses ermöglicht wird.

- Chicken-Little
- Butterfly
- Renaissance
- COREM

Diese Strategien werden jedoch nicht weiter behandelt. Das allgemeine Vorgehen bei einer Migration sollte jedoch dem Anwender des Entscheidungsunterstützungstools bekannt sein um den Prozessablauf durchführen zu können.

2.4 Entscheidungsunterstützungssysteme

Wir Menschen treffen täglich tausende Entscheidungen. Diese werden häufig in sehr kurzer Zeit, spontan und unbewusst getroffen. Ziel einer Entscheidung ist es, aus mehreren Möglichkeiten oder Varianten eine Auswahl zu treffen, die der Entscheider als präferiert bestimmt [JPF05]. Die damit vorgezogene Option unterscheidet sich womöglich von den anderen Auswahlmöglichkeiten in der Hinsicht, dass der Wert, die Risiken und Konsequenzen hiervon optimal abgewogen sind und eine bestmögliche Entscheidung darstellen. Der

Prozess der Entscheidungsfindung kann durch vielfältige Faktoren beeinflusst werden, wie beispielsweise:

- Ziele und Auswirkungen
- Gegenstand und Inhalt
- Abschätzung von Risiko und Chance
- Zeitlicher Rahmen (spontanes oder rationelles Vorgehen)
- Tragweite (Unternehmen, Abteilung, Gruppe, Persönlich)
- Rahmenbedingungen
- Methoden und Verfahren der Anwendung

Bringt man die Entscheidungsfindung auf eine betriebliche Ebene, welche für den Erfolg eines Vorhabens von großer Tragweite ist, ist die Entscheidungsfindung ein komplexer und differenzierter Vorgang [Jac13]. Die Grundlage der Entscheidungen ist im Regelfall ein Informationsproblem. Das Unterbewusstsein, Wissen und das Gefühl für eine *gute Entscheidung* wirken in den Entscheidungsprozess mit ein um die bestmögliche Wahl zu treffen.

Nach Jacob [Jac13] gibt es verschiedene Ausprägungen der Entscheidungsunterstützung, von der

bloßen Bereitstellung von Informationen bis hin zu konkreten Vorschlägen und Bewertungen von Handlungsalternativen.

Die von Jacob veröffentlichte Darstellung (siehe Abbildung 2.4) veranschaulicht wie ein Entscheidungsunterstützungssystem (EUS) aus verschiedenen Informationsquellen, durch geeignete Modelle und Methoden verschiedene Hilfestellungen zur Verfügung stellt. Der Entscheider verwendet im Laufe des Prozesses diese Informationen und kann damit effektiver zu einer Entscheidung gelangen.

Mit dieser Arbeit soll ein EUS entwickelt werden, das dem Anwender bei der Entscheidung der Migration einer bestehenden Anwendung in die Cloud unterstützen soll.

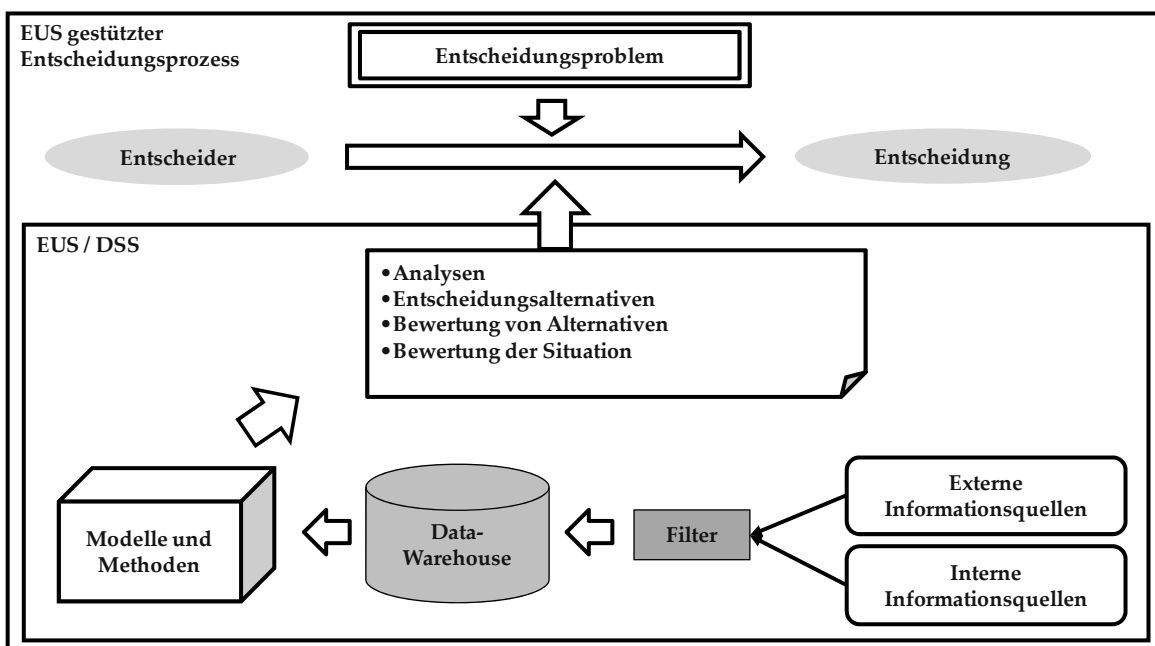


Abbildung 2.4: Entscheidungsprozess mit Hilfe eines EUS

Stand der Technik

Dieses Kapitel beschreibt den *Stand der Technik* innerhalb des Forschungsgebietes. Dabei werden auf verwandte Arbeiten eingegangen und bereits existierende Entscheidungsunterstützungstools betrachtet. Mit dem erworbenen Wissen wird die Entwicklung eines eigenen Werkzeugs zur Entscheidungsunterstützung vorbereitet.

3.1 Migration in die Cloud

Wie bereits im letzten Kapitel erläutert, kann man eine Anwendung in drei Schichten aufteilen. Da die Kommunikation zwischen den Schichten austauschbar ist, kann man diese voneinander trennen und die Schichten (teilweise oder ganz) in die Cloud auslagern. So können die einzelnen Schichten auf unterschiedliche Diensttypen oder Anbieter ausgelagert werden. Eine beispielhafte Darstellung dieser modularen Aufteilung wird in der Veröffentlichung von Leymann [Ley09] (siehe Abbildung 3.1) gezeigt.

In diesem Zusammenhang sei auch der Artikel *Moving Applications to the Cloud* [LFM⁺11] erwähnt, welcher eine Methode und Vorgehensweise zur Migration von Anwendungen in die Cloud beschreibt. Der Kernpunkt ist ebenfalls die Aufteilung einer Anwendung in die verschiedenen Schichten. Die Architektur eines unterstützenden Werkzeugs für dieses Vorgehen wird in der Veröffentlichung beschrieben.

3.2 Entscheidungsunterstützung

Wie die Migration von Anwendungen in die Cloud durchgeführt werden kann, wurde im vorangegangenen Abschnitt erläutert. Die Frage, ob die Migration auch sinnvoll ist muss jedoch ebenfalls geklärt werden. Dazu müssen die Erwartungen, Ziele und Vorteile mit den Konsequenzen der Migration abgewogen werden.

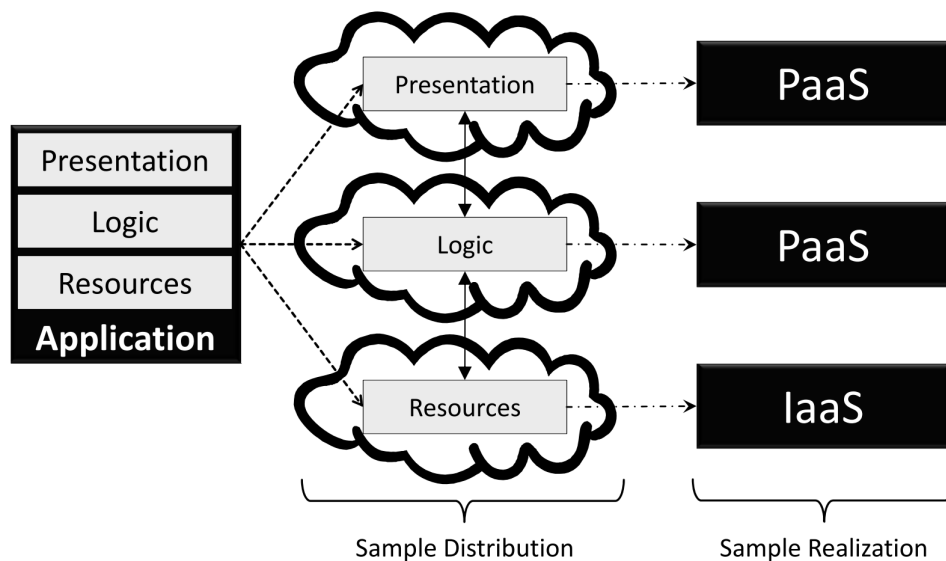


Abbildung 3.1: Eine Anwendung schichtweise in die Cloud migrieren

3.2.1 Frameworks

Eine Untersuchung von Gao und Schneider [GS12] untersucht 48 Frameworks aus der Sicht von Wirtschaftsinformatikern. Diese Frameworks sollen den Anwender bei Überlegungen um das Cloud Thema unterstützen. Ziel der Arbeit war eine Bewertung der Frameworks aufgrund des Reifegrad und Wirkungspotential. Abgedeckt werden die Themengebiete **Sicherheit, Zuverlässigkeit und Flexibilität, Management** sowie übergreifende Frameworks.

Exemplarisch werden, aus der großen Auswahl von Frameworks, in den nächsten Absätzen einige interessante Ansätze vorgestellt.

3.2.2 Service Measurement Index (SMI) Framework

Um die verschiedenen Faktoren bei der Migration gewichten zu können, kann man sich bereits bekannten Methoden aus der Betriebswirtschaftslehre bedienen. Der Ansatz des SMI Frameworks [GVB11] bietet eine Reihe an Key Performance Indicator (KPI) an, um Cloud-basierte Dienste zu messen und zu bewerten. Die standardisierten KPI werden in folgende Quality of Service (QoS) Gruppen zusammengefasst:

- Verantwortlichkeit
- Flexibilität
- Dienstsicherheit

- Kosten
- Leistung
- Sicherheit und Datenschutz
- Benutzbarkeit

Ziel der Methode ist es, eine Bewertung für die angebotenen Dienste zu erstellen und damit einen vergleichbaren Maßstab zu erhalten. Die Bewertungen für quantitative KPI werden nach mathematischen Methoden bestimmt, die qualitativen KPI werden dagegen von der Anwendererfahrung beeinflusst.

Als Bewertungsmethode wird der Analytic Hierarchy Process (AHP) [Saa94] als Grundlage verwendet. Diese Methode aus der Entscheidungstheorie soll unter anderem dabei helfen, eine Entscheidung zu finden und das Ergebnis nachweisbar zu begründen. Die hierarchische Struktur die beim Prozessablauf erstellt wird besteht aus Kriterien welche zum Beispiel Attribute, Merkmale oder Alternativen sein können. Der prozedurale Ablauf ist grob gegliedert in die Schritte

1. Sammeln der Daten
2. Daten vergleichen und gewichten
3. Daten verarbeiten

Die Grundprinzipien der Methode werden auch in dem MIGRATE! Vorgehensmodell (siehe Kapitel 1.4) bzw. in dem zu entwickelnden Entscheidungsunterstützungstool angewendet.

3.2.3 IT Infrastructure Library (ITIL) Framework

In den Veröffentlichungen von *HMD - Praxis der Wirtschaftsinformatik* (Heft 288) [RDMB12], wird auf das Thema Entscheidungsunterstützung für Cloud Computing näher eingegangen. Der Fokus liegt hierbei auf dem ITIL Framework, welches eine Sammlung von Best-Practices in der Informationstechnologie darstellt. Durch die Phasen des ITIL-Prozesses können beispielsweise die Fragestellungen, welche in Abbildung 3.2 als Mindmap dargestellt sind, geklärt werden. Diese Beschreibungen sind aus der Situationsmatrix von Robrecht [RDMB12] entnommen. Die **vorgelagerten Entscheidungen** dienen der Informationsbeschaffung und der grundlegenden Überlegung, ob eine Migration in die Cloud sinnvoll ist. In diesem Schritt wird auch das Portfolio erstellt, welches bei den **Beschaffungsentscheidungen** der Auswahl und Vertragsgestaltung dient. Ist ein Anbieter ausgewählt worden, muss die **Leistungserbringung** begleitet und überprüft werden. Sollte sich eine **Nicht-/Schlechtleistung** des Anbieters herausstellen, muss auf diesen Fall eine Reaktion erfolgen und ein Ablaufplan zur Behebung der Probleme vorliegen. Unter Umständen ist auch ein Anwenderwechsel oder ein Rückzug aus der Cloud denkbar.

Der Prozess der Entscheidungen beginnt, ähnlich wie das MIGRATE! Vorgehensmodell, bei den grundlegenden Fragen zu Cloud Computing:

- Genereller Einsatz von Cloud Computing sinnvoll?
- wahrgenommener Nutzen (zum Beispiel Kosten, Flexibilität, Qualität)
- wahrgenommene Risiken (zum Beispiel Sicherheit, Performance)

Weiter wird der Bereitstellungstyp (hier Betriebsmodell genannt) ausgewählt. Zu dieser Entscheidung wird der BITKOM Cloud Computing Leitfaden [Web10] empfohlen. Dort werden Aspekte der Private und Public Cloud beschrieben, sowie Mischformen erläutert. Das damit entstehende Cloud Portfolio beinhaltet strategische, technische und wirtschaftliche Entscheidungskriterien.

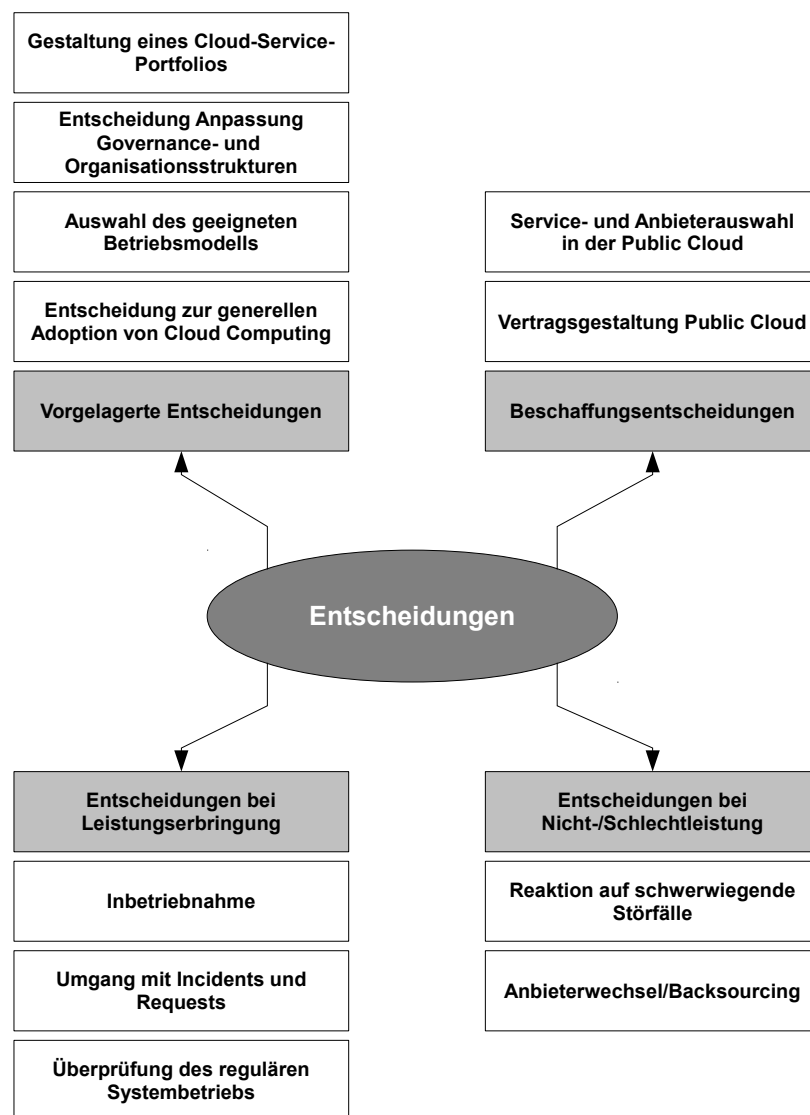


Abbildung 3.2: ITIL-Entscheidungen, nach Situationsmatrix von Robrecht

3.2.4 Risikoabschätzung

Aus dem Bereich der Unternehmensorganisation sind Methoden zum Risikomanagement bekannt, die eine ähnliche unterstützende Funktion bieten. Die *International Organization for Standardization (ISO)* veröffentlichte mit dem ISO 31000 einen Leitfaden [Pur10], welcher als Risikomanagement-Standard (RMS) etabliert ist. Ein von Microsoft veröffentlichtes Handbuch zur Risikoabschätzung [SN12] greift auf diese Methode zurück und gibt Hilfestellungen bei der Abschätzung über die Entscheidung, ob eine Migration in die Cloud durchführbar ist. Der Prozess ist in der Abbildung 3.3 veranschaulicht, als Referenz dient die Abbildung von Stone [SN12]. Nach der **Identifikation der Möglichkeiten** wird die **Risikoabschätzung** nach dem Standard durchgeführt. Die daraus entstehenden Ergebnisse werden dem Anwender präsentiert und darauf kann die Entscheidung aufgebaut werden.

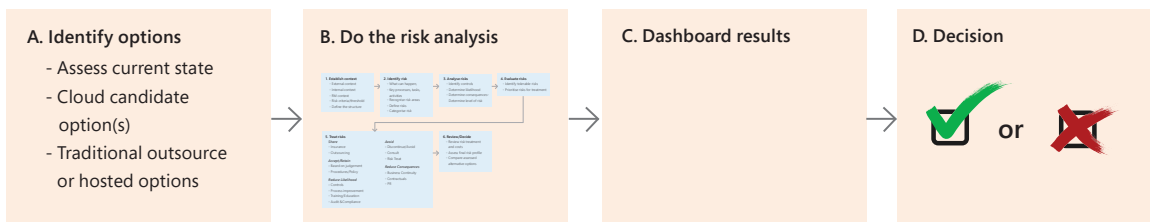


Abbildung 3.3: Prozess des Risikomanagement nach ISO 31000

Hauptbestandteil der Entscheidung bildet die Risikoanalyse nach dem genannten Standard. Die resultierende Auswirkung hängt von der Wahrscheinlichkeit ab, mit der das Ereignis eintritt und der möglichen Auswirkung. Somit lässt sich die Auswirkung eines Risikos wie folgt bestimmen:

$$\text{Gefährdungspotential} = \text{Wahrscheinlichkeit} \times \text{Auswirkung}$$

Für verschiedene Kategorien wird eine Risikoabschätzung nach oben genannter Formel evaluiert. Daraus kann man eine Übersichtstabelle erstellen, welche zeigt wo eine Migration, aus der Sicht des Risikomanagement, praktikabel erscheint. Die Kategorien für die Risikoabschätzung ermöglichen eine Aufteilung in Bereiche, um die Auswirkungen und Häufigkeit besser abschätzen zu können. Als Beispielaufteilung, die verschiedene Unternehmensbereiche abdecken soll, werden folgende vier Kategorien vorgeschlagen:

- Compliance Risiken
- Strategische Risiken
- Betriebliche Risiken
- Markt und finanzielle Risiken

3.3 Migrations-Entscheidungsunterstützung

Für den Umstieg zu Cloud Computing müssen die betrachteten Infrastrukturen sorgfältig verglichen und die einzelnen Aspekte davon betrachtet werden. Gewöhnliche Entscheidungsunterstützungstools bauen auf Evaluations-Methoden auf, die beispielsweise die Kosten der aktuellen und Cloud-Infrastruktur vergleichen können. Um mehrere Kriterien, speziell für den Cloud-Bereich, in diese Evaluierung mit einzubeziehen ist ein erweitertes Framework nötig.

3.3.1 Multi-Criteria Comparison Method for Cloud Computing ((MC²)²)

Menzel hat in seiner Arbeit das (MC²)² Framework entwickelt. Hiermit kann die Auswahl der bestmöglichen Infrastruktur durch Bewertung und Auswahl mehrerer Punkte unterstützt werden. Es erlaubt die Infrastrukturen auf verschiedenen Ebenen zu unterscheiden. Die Kosten können ebenso ein Faktor sein wie die Vorteile, Chancen und Risiken einer Cloud Migration. Durch die Allgemeinheit des Frameworks kann es jedoch auch für andere Entscheidungen bezüglich IT-Infrastrukturen angewendet werden.

Das Framework ist eine Weiterentwicklung des *CloudTCO* Ansatz von Klems [KNT09], welches die Total Cost of Ownership (TCO) als Maßstab verwendet. Die Erweiterungen gegenüber dem CloudTCO Ansatz sind:

- Mehrere, unterschiedliche Kriterien
- Auswerten von Alternativen nach einem Maßstab
- Erweiterbare Auswertungsmethoden

Der Ablauf des (MC²)² Frameworks ist in Abbildung 3.4 dargestellt. Ein wichtiger Punkt während den ersten Phasen ist eine interne/externe Informationsquelle wie beispielsweise:

- Wissensdatenbank im Unternehmen
- Business Intelligence Systeme (Data Mining, Text Mining)
- Softwaresysteme für Umfragen, Deployment und Analyse

Mit dem vorliegenden Wissen kann das Grundszenario festgelegt und mögliche Alternativen bestimmt werden. Die Kriterien und Anforderungen werden identifiziert und dienen als Datenbestand für eine Multi-Kriterien Entscheidungsmethode. Es wird hier nicht explizit ein Verfahren festgelegt, sondern je nach Anforderungen kann ein passendes ausgewählt werden. Eine Auswahl an geeigneten Methoden hierfür findet man in der Literatur [Kee93] [ZC82] [YH95]. Vorgeschlagen von den Autoren ist der *Analytic Network Process* (ANP) [Saa94] aufgrund der Eigenschaft mit komplexen Kriterien-Verflechtungen umgehen zu können. Ein komplexes Netzwerk aus Kriterien kann eine Entscheidungssituation besser modellieren und erlauben daher eine bessere Abschätzung.

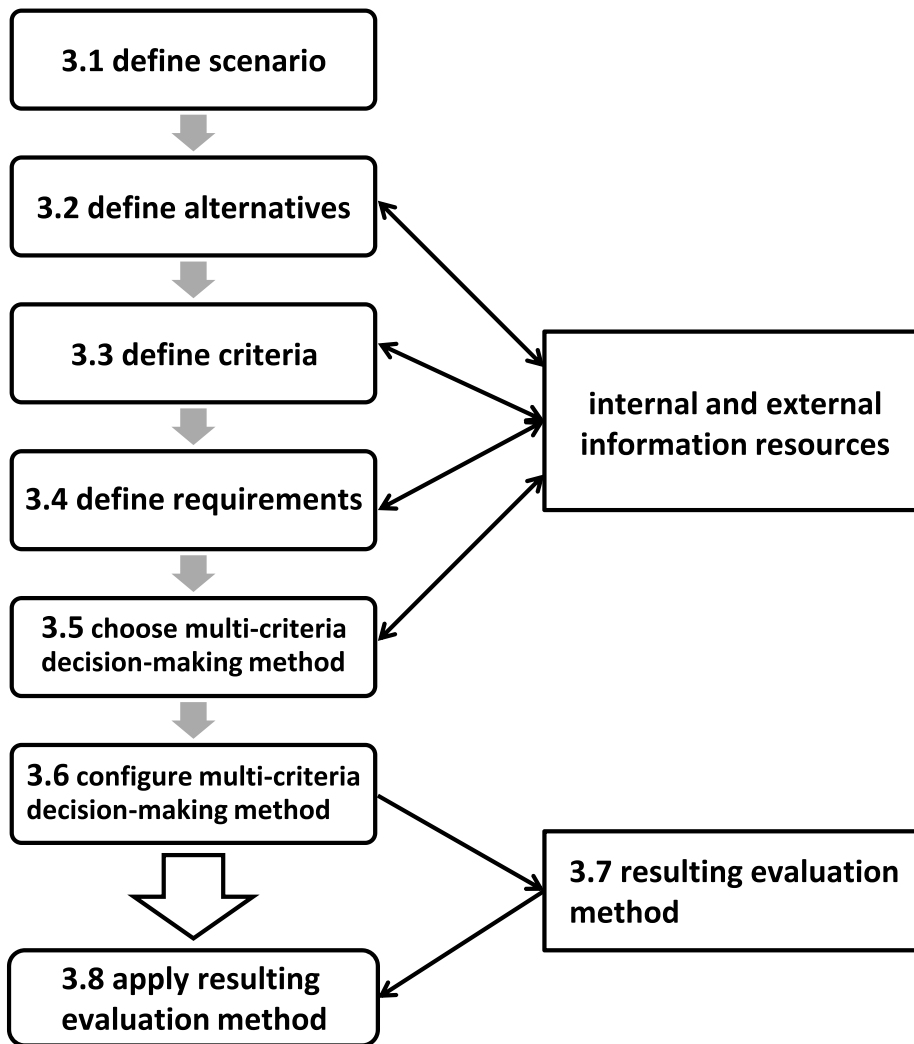


Abbildung 3.4: Ablaufprozess im $(MC^2)^2$ Framework

3.3.2 Cloud Adoption Toolkit

Eine spezielle Fallstudie zum Thema Cloud Migration wurde von Khajeh-Hosseini und Sommerville [KHGS10] veröffentlicht. Darin werden die Chancen und Risiken der Migration eines IT-Systems analysiert. Das Fallbeispiel befasst sich mit der Umstellung eines In-House Datenzentrums zu dem IaaS Angebot der Amazon Elastic Compute Cloud (EC2). Es werden die Interessen der beteiligten Parteien beachtet und eine technische und finanzielle Analyse durchgeführt. Durch die Kostenreduzierung und weniger Service-Aufwand könnte man zunächst von einer klaren Entscheidung zugunsten der Cloud Architektur ausgehen. Jedoch müssen auch die bedeutenden Risiken des Vorhabens abgewogen werden, welche unter Umständen eine Unternehmensweite Auswirkung haben kann. In diesem Rahmen wurde

auch ein Vorgehensmodell entwickelt, welches in Abbildung 3.5 dargestellt wird. Der Ablauf ist aus der Darstellung von Khajeh-Hosseini [KHGS10] entnommen. Im Zusammenhang mit Cloud Computing wird eine Einbeziehung der Nachhaltigkeit und des Energieverbrauchs beachtet, ebenso wie die Auswirkungen auf die beteiligten Parteien.

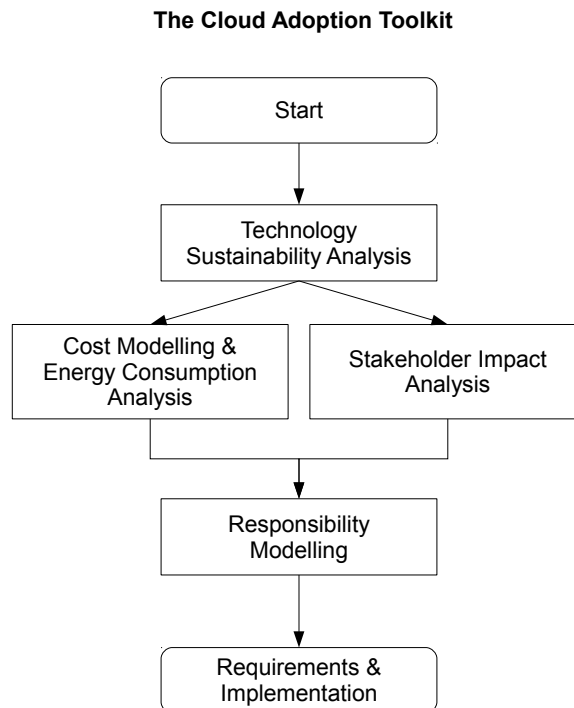


Abbildung 3.5: Vorgehen des Cloud Adoption Toolkit

3.3.3 Federal Cloud Computing Strategy

Eine Untersuchung im Auftrag von The White House Washington, veröffentlicht unter dem Titel *Federal Cloud Computing Strategy* [Kun11], analysiert unter anderem die Entscheidungen rund um das Cloud Computing. Das in Abbildung 3.6 dargestellte Vorgehen in drei Schritten besteht aus der **Auswahl**, **Bereitstellung** und **Verwaltung** der Cloud Anwendung.

1. Auswahl

Die Vorteile einer möglichen Migration werden identifiziert und die Cloudfähigkeit der Anwendungen bestimmt. Dies gleicht einer Risikoabschätzung und Aspekte wie Sicherheit, Compliance, Integrität, Datenkontrolle und Charakteristiken der Cloud Architektur sollten beachtet werden.

2. Bereitstellung

Die Dienste werden mit Beachtung der Effizienz bereitgestellt, dies erfordert die Berücksichtigung einiger Faktoren bei der Anbieterauswahl und Vertragsgestaltung.

3. Verwaltung

Anstatt das eigene Kapital (Hardware, Software, Prozesse) zu verwalten, muss ein Umdenken zur Verwaltung von Services stattfinden. Wenn mehrere Unternehmen an der Bereitstellung und Beanspruchung der Cloud Dienste beteiligt sind, erfordert dies ein angepasstes Management. Monitoring und Neu-Bewertung der Dienstleistung hat höhere Priorität als zuvor.

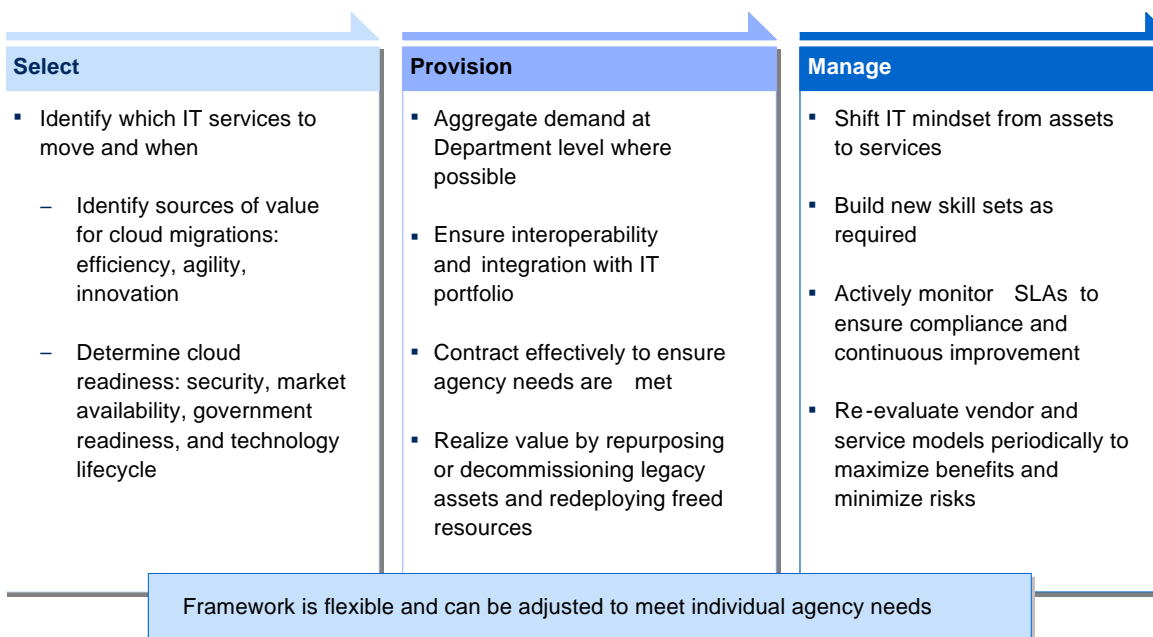


Abbildung 3.6: Federal Cloud Computing Strategy Framework für Entscheidungen zur Cloud Migration

3.4 Angebote der Cloud Service Provider

Dieser Abschnitt untersucht verschiedene CSP und deren Unterstützung für eine Migration von Anwendungen auf die angebotenen Cloud-Dienste. Die dabei erfassten Daten dienen zudem dem Datensatz der Stammdaten welche im Kapitel 4.1: Anforderungsanalyse und Entwurf unter dem Punkt FA-12: Stammdaten beschrieben ist. Die ausgewählten Anbieter stehen dabei dem Anwender als Vorauswahl zur Verfügung.

Die Informationsquellen hierzu finden sich meistens direkt auf den Webseiten der Unternehmen. Es gibt aber auch Artikel wie *Up in the air: Moving your applications to the cloud* [Lou10]

welcher die unterschiedlichen Anbieter in die Diensttypen kategorisiert und einen Überblick der Angebote gibt.

Die Reihenfolge dieser Auflistung ist alphabetisch und hat keine bevorzugende Bedeutung.

3.4.1 Amazon

Unter dem Namen Amazon Web Services (AWS) wird ein Komplettpaket für die Cloud Architektur angeboten. Das Leistungsspektrum umfasst, ähnlich wie bei anderen Anbietern, folgende Gruppen:

- Web-Anwendungen
- Große Datenmengen und High Performance Computing (HPC)
- Unternehmensanwendungen
- Disaster Recovery und Archivierung
- Öffentlicher Sektor

Die technische Umsetzung für die angebotenen Dienste teilen sich auf mehrere Produkte auf. Eine kleine Auswahl hiervon ist beispielsweise:

Elastic Compute Cloud (EC2) skalierbare Rechenkapazität und leistungsabhängige Abrechnung. Skalierung und Lastenverteilung können gewählt werden.

CloudFront Bereitstellung von Inhalten aus einem weltweiten Netzwerk

Relational Database Service (RDS) Relationale Datenbank, skalierbar, zuverlässig und sicher

Simple Storage Service (S3) Redundante Datenspeicherung-Infrastruktur

Amazon bietet selbst auf der Unternehmenswebseite eine Reihe an Whitepapers an, die sich mit den Themen zur Architektur, Sicherheit und Wirtschaftlichkeit der AWS befassen. Um die Entscheidung zur Migration in die Cloud zu unterstützen werden unter anderem **Best Practices**, **Leitfaden**, **Migrationshilfen**, **Preiskalkulationen** und **Checklisten** angeboten.

3.4.2 Microsoft Azure

Microsoft bietet unter dem Namen *Azure* eine Cloud Computing Plattform und Infrastruktur an. Ein Fokus liegt dabei auf die Entwicklung von Anwendungen, so werden Plattformen für Web-Seiten oder Cloud Dienste bereitgestellt. Die Entwicklungssprache kann dabei frei gewählt und eigene Tools und Frameworks verwendet werden. Dazu gehören auch Datenzentren und typische Cloud Vorteile wie Lastenausgleich oder Fehlertoleranz. Das Betriebssystem der Cloud ist konsequenterweise ebenfalls von Microsoft.

Auf der Unternehmenswebseite wurden mehrere Artikel und Blogbeiträge veröffentlicht um die Migration von Anwendungen zu erleichtern. Abbildung 3.7 zeigt den Aufbau des Buches *Moving Applications to the Cloud* [BHJ⁺12] von Microsoft. Anhand eines Fallbeispiels wird ein Szenario beschrieben, dass ein Unternehmen eine Anwendung in die Cloud migrieren möchte. Neben den Grundlagen werden die Möglichen Dienstypen erläutert (IaaS, PaaS) und genauer betrachtet. Die Auswahl und Begründung der Entscheidungen werden nachvollziehbar geschildert. Ebenso werden die Kosten für die Cloud-Lösung belegt.

Auch wenn das erwähnte Handbuch spezifisch für Microsoft Azure ausgelegt ist, kann man einige Vorgänge allgemein nachvollziehen oder auf andere Anbieter ausweiten.

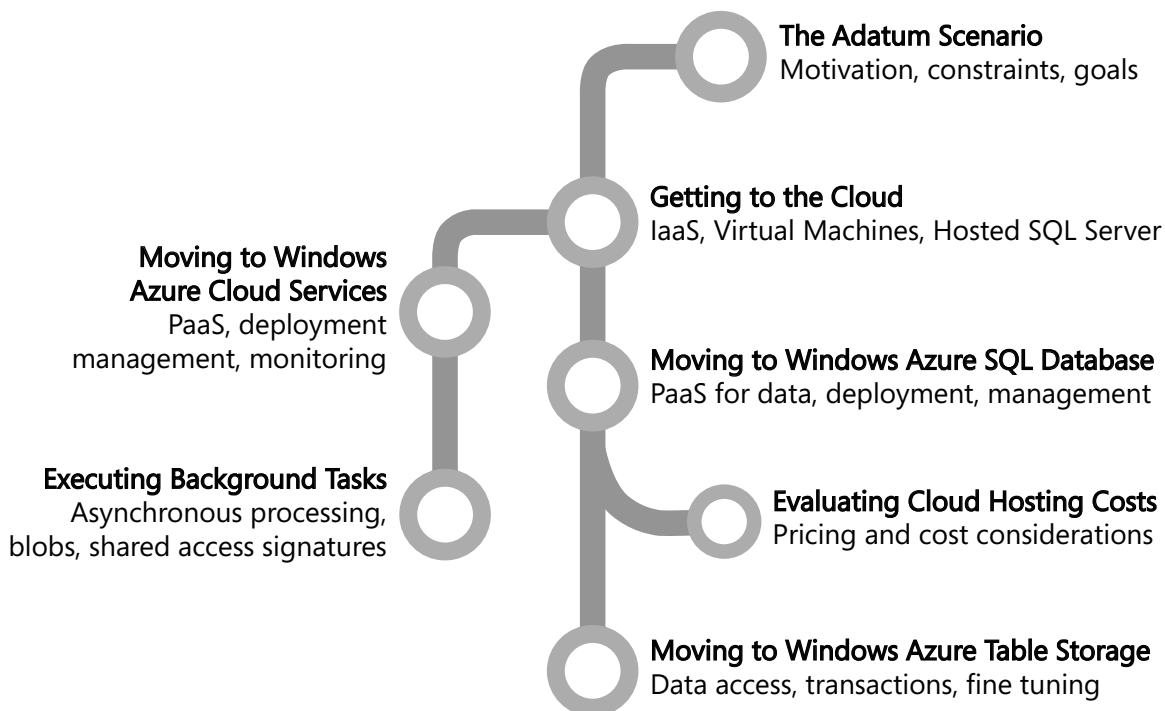


Abbildung 3.7: Aufbau des Migrationsleitfadens von Microsoft

3.4.3 Hewlett-Packard

Als einer der großen Anbieter hat Hewlett-Packard (HP) mit der *HP Cloud* Lösung einen Leitfaden zur Anwendungsmigration in die Cloud [HP13] entwickelt. Als Technologiepartner werden GigaSpaces, CliQr und AppZero aufgeführt. Die Herausforderungen die laut HP gemeistert werden müssen sind:

- Anpassungen von Anwendungen: Für das Re-Deployment in der Cloud

- Bereitstellung der Infrastruktur: Deployment von Virtuellen Maschinen, Netzwerk und Speicherlösungen
- Cloudfähige Anwendungen: Anwendungen müssen Monitoring, Skalierung und Speichermanagement unterstützen
- Hybrid Cloud Deployments: Bestimmte Anwendungen haben Anforderungen, die sich im Bereitstellungstyp unterscheiden. Hier muss eine Zusammenarbeit sichergestellt werden
- Betrieb und Verwaltung: Ressourcen werden benötigt um die Anwendungen zu betreiben, Skalierung zu unterstützen, Ausfälle und das Budget zu verwalten
- Sicherheit: Datenverschlüsselung bei Übertragung und Speicherung

HP bietet eine Dienstleistung an, bestehende Multi-Tier Architekturen über eine Schnittstelle in die HP Cloud zu migrieren. Durch das umfassende Angebot soll die Migration einfach und kostengünstig sein. Der Anwender kann damit die Vorteile des Cloud Computing direkt nutzen.

Anforderungsanalyse und Entwurf

Nach dem Überblick über den Stand der Technik wird in diesem Kapitel ein eigenständiges Werkzeug zur Entscheidungsunterstützung entworfen. Hierfür ist es nötig die Anforderungen zu analysieren und darauf aufbauend ein Entwurf des Tools und dem darunterliegenden Datenmodell zu erstellen.

Einen Überblick zu allen FA und NFA zeigt die Tabelle 4.1 (Übersicht der Anforderungen). Die Details hierzu sind in den einzelnen Unterkapiteln erläutert.

4.1 Funktionale Anforderungen (FA)

Die folgende Auflistung zeigt die festgesetzten funktionalen Anforderungen. Diese wurden zusammen mit dem Betreuer dieser Arbeit in mehreren Iterationen ausgearbeitet.

FA-01: Entscheidungsunterstützung

Dem Benutzer wird ein Werkzeug zur Entscheidungsfindung bereitgestellt. Dies umfasst vor allem eine komfortable Eingabe, Verwaltung und Darstellung der eingegebenen Daten bzw. Parameter. Zusätzlich wird durch Hilfestellungen eine Entscheidungsfindung erleichtert, indem beispielsweise Gegenüberstellungen den Sachverhalt übersichtlich darstellen.

Der Prozess der Entscheidungsunterstützung orientiert sich an dem Vorgehensmodell.

FA-02: Vorgehensmodell

Das im Rahmen des MIGRATE! Projekts entwickelte Vorgehensmodell ist durch das zu entwickelnde Tool umgesetzt, die einzelnen Schritte werden durch das Tool abgebildet. Das Tool zeigt eine Übersicht zu dem Vorgehensmodell als Einführung bei der Benutzung an.

Die einzelnen Schritte des Modells sind im Entscheidungsunterstützungssystem wiedererkennbar und leiten so den Benutzer entlang des Prozesses zum Ziel: der Identifikation und Evaluation von Migrationsvorhaben.

Nummer	Kurzbeschreibung	Kategorie
FA-01	Entscheidungsunterstützung	Allgemein
FA-02	Vorgehensmodell	
FA-03	Datenmodell	
FA-04	Persistente Datenspeicherung	
FA-05	Hilfesystem	Hilfe
FA-06	Projektverwaltung	Projekte
FA-07	Projektaufteilung	
FA-08	Gewichtung	Auswertung
FA-09	Graphische Repräsentation	
FA-10	Dateiverwaltung	Dokumentverwaltung
FA-11	Erweiterbarkeit	Anpassungen
FA-12	Stammdaten	Datensatz
FA-13	Beispielprojekt	
FA-14	Navigation	Layout
FA-15	Aktueller Schritt	
FA-16	Projektauftrag	
FA-17	Sprache der Oberfläche	
FA-18	Benutzerkonto	Benutzer
FA-19	Mehrbenutzerfähigkeit	
NFA-01	Funktionalität	Nicht-funktionale Anforderungen
NFA-02	Zuverlässigkeit	
NFA-03	Benutzbarkeit	
NFA-04	Effizienz	
NFA-05	Wartbarkeit	
NFA-06	Übertragbarkeit	
NFA-07	Skalierbarkeit	
NFA-08	Sicherheit	

Tabelle 4.1: Übersicht der Anforderungen

FA-03: Datenmodell

Ein bereits umgesetztes Datenmodell kann importiert werden. Im Schritt der Datenerfassung (Aktivitäten) hat der Benutzer die Möglichkeit bereits modellierte Komponenten zu laden. Diese verhalten sich nach dem Import wie manuell eingetragene Komponenten.

FA-04: Persistente Datenspeicherung

Die vom Benutzer eingegeben Daten werden konsistent, unabhängig und persistent gespeichert. Dies gilt auch für die damit abgebildeten Zusammenhänge. Es muss gewährleistet sein, dass bei dem Mehrbenutzerbetrieb keine Konflikte auftreten oder Informationen verloren gehen.

FA-05: Hilfesystem

Eine Kontextsensitive Hilfe steht dem Benutzer auf den Seiten zur Verfügung. Dies erleichtert die Eingabe der Daten oder zeigt bei Problemen Hinweise an. Ein Symbol hierzu darf nicht ablenkend wirken und ein Aufruf der Hilfe bringt den Benutzer nicht aus dem Prozessfluss.

FA-06: Projektverwaltung

Es können mehrere Projekte angelegt werden. Jeder Benutzer kann beim Anlegen eines Projekts folgende Eigenschaft festlegen:

Privat Das Projekt steht nur dem Ersteller zur Verfügung.

Öffentlich Das Projekt steht allen registrierten Benutzern zur Auswahl. Jeder Benutzer kann das Projekt auswählen und betrachten. Ein Projekt kann zu jedem Zeitpunkt von maximal einem Benutzer geöffnet sein.

Diese Eigenschaft kann später durch den Ersteller geändert werden. Wird die Änderung von *Öffentlich* zu *Privat* durchgeführt kann nur noch der Ersteller das Projekt bearbeiten und den anderen Benutzern steht der Eintrag nicht mehr zur Verfügung.

FA-07: Projektaufteilung

Bei ausreichendem Fortschritt der Datenerfassung und an sinnvollen Punkten ist es möglich, das Projekt aufzuteilen. Ab diesem Zeitpunkt gibt es konzeptionell zwei verwandte Teilprojekte. Dabei verhalten sich die Teile wie folgt:

- Der Teil bis zur Aufteilung ist identisch, nach dem Split des Projektes unterscheiden sich die zwei Projekte.

- Der Teil nach der Aufteilung nennt sich *Variante*. Der Benutzer kann nachvollziehen wann und wo das Projekt aufgeteilt wurde und dass er aktuell einer Variante bearbeitet.
- Der Abschnitt vor der Aufteilung wird weiterhin als eine gemeinsame Grundlage verwaltet, er wird (konzeptionell) nicht dupliziert.

Die Varianten können in einer Ansicht verglichen werden. Eine Variante kann verworfen werden, wobei die verwandte Variante dann wieder als einzelnes Projekt dargestellt wird. Ein Projekt kann jeweils nur einmal aufgeteilt werden, so dass zwei Varianten zur Verfügung stehen.

FA-08: Gewichtung

Die in einem Projekt vorhandenen Kriterien können mit einem Wert gewichtet werden. Diese Einstellung kann bei der Anzeige reflektiert werden, zum Beispiel können die Einträge automatisch sortiert werden. Durch die Gewichtung besonders (ir-) relevante Einträge können graphisch hervorgehoben werden um somit eine bessere Übersicht zu erhalten.

FA-09: Graphische Repräsentation

Die im Tool erfassten Daten können, wie im Vorgehensmodell bereits erwähnt, zur besseren Entscheidungsfindung graphisch aufbereitet werden. Die Darstellung als Portfolio wird unterstützt, sofern mindestens zwei Kriterien pro Portfolio verfügbar sind. Für das Portfolio gibt es typische Vorgaben von Kriterien-Paaren, welche eine relevante Aussagekraft haben. Optional: Portfolios individuell konfigurierbar

FA-10: Dateiverwaltung

Das Tool ermöglicht dem Benutzer Dokumente zu einem aktuell ausgewählten Eintrag hinzuzufügen. Dies wird bei den Eingabemasken angeboten, soweit die Einbindung eines Dokuments sinnvoll erscheint. Als Dateitypen sind gängige Formate erlaubt. Vor allem der Projektauftrag kann mit Dokumenten angereichert werden um entsprechende Quellen und Nachweise verfügbar zu halten.

FA-11: Erweiterbarkeit

Der Benutzer kann in einem dafür vorgesehenen Administrationsbereich Anpassungen vornehmen. Dies ermöglicht bei den Eingabemasken eine flexible Auswahl der Informationseingabe. Das Vorgehensmodell beschreibt eine Reihe von Elementen, für welche eine Erweiterbarkeit vorgesehen ist:

Kriterienkatalog

Der Kriterienkatalog besteht aus Gruppierungen von einzelnen, aber thematisch zusammenhängenden Kriterien. Diese werden bei der Analysephase verwendet um das Modell mit relevanten Informationen anzureichern. In der Grundversion stehen die folgenden Gruppen zur Verfügung:

- Technische Kriterien
- Rechtliche Kriterien
- Finanzielle Kriterien
- Energie Kriterien
- Dienstgüte Kriterien

Die einzelnen Kriterien dieser Gruppen sind nicht fest hinterlegt, es können Einträge hinzugefügt oder entfernt werden. Eine Auswahl an sinnvollen Kriterien, wie im Vorgehensmodell beschrieben, ist in der Grundversion des Tools bereits hinterlegt.

Zielinfrastrukturen

Die im Vorgehensmodell genannten Zielinfrastrukturen sind ebenfalls erweiterbar. Bei der Erfassung der Migrationsszenarien wird die Infrastruktur anhand von vorgegebenen Typen definiert.

- Anbieter
Einträge können hinzugefügt oder entfernt werden.
- Diensttypen
Es steht IaaS, PaaS, SaaS und XaaS zur Auswahl.
- Bereitstellungstypen
Die Standardwerte sind Private, Public und Hybrid.

Bei der Bestimmung der Zielinfrastrukturen werden die, gegebenenfalls modifizierten, Kriterien zur Auswahl angezeigt.

FA-12: Stammdaten

Das Tool enthält im Auslieferungszustand einen Grundsatz an Stammdaten. Die bereits erwähnten Kriterien (-gruppen) und Zielinfrastrukturen sind mit jeweils mindestens drei Einträgen vordefiniert.

FA-13: Beispielprojekt

Nach der Installation ist es möglich einen Datensatz einzuspielen, der ein Beispielprojekt repräsentiert. Hierbei sollen möglichst alle Aspekte des Tools demonstriert werden können. Dieser Datensatz kann auch zur Einarbeitung in das Werkzeug verwendet werden.

FA-14: Navigation

Es ist jederzeit erkennbar auf welcher Seite man sich befindet. Die Struktur der Navigation ist wie folgt aufgebaut:

- Die Einstiegsseite ist jederzeit aufrufbar
- Die Navigationspunkte orientieren sich an dem Prozess des Vorgehensmodells und sind klar verständlich
- Die Navigation zurück zu bereits absolvierten Schritten ist möglich
- Ein Überspringen noch nicht absolvierter Schritte ist nur möglich, wenn es sinnvoll erscheint
- Das Ändern von Daten in bereits absolvierten Schritten ist nur möglich, wenn es sinnvoll erscheint

FA-15: Aktueller Schritt

Der Fortschritt im Vorgehensmodell wird auf den entsprechenden Seiten kenntlich gemacht. Damit sieht der Benutzer welches der aktuelle Schritt ist und kann sich besser im Prozess orientieren.

FA-16: Projektauftrag

Der Projektauftrag ist ein zentrales Element des Prozesses und jederzeit einsehbar. Der Benutzer kann den Projektauftrag betrachten, ohne dadurch im Fortschritt des Entscheidungsprozesses unterbrochen zu werden.

FA-17: Sprache der Oberfläche

Die Anzeigesprache ist durch den Benutzer festlegbar. In der Grundversion stehen die Sprachen Englisch (Grundeinstellung) und Deutsch zur Auswahl.

FA-18: Benutzerkonto

Die Anwendung bietet dem Benutzer an einen neues Konto zu erstellen oder sich für ein bestehendes Konto anzumelden. Der Benutzer kann Einstellungen für sein Konto vornehmen, wie zum Beispiel den Namen oder das Passwort ändern.

FA-19: Mehrbenutzerfähigkeit

Die Software kann von mehreren Benutzern unabhängig benutzt werden. Es darf keine Beeinträchtigung durch andere Benutzer entstehen (siehe Projektverwaltung). Diese Anforderung betrifft vor allem die Datenspeicherung und Projektverwaltung. Hier müssen Konflikte und Inkonsistenzen vermieden werden, so dass ein reibungsloser Mehrbenutzerbetrieb ermöglicht wird.

4.2 Nicht-funktionale Anforderungen (NFA)

Bei den NFA wurde als Orientierung die Norm ISO 9126 [ISO01] bzw. die DIN 66272 [DIN94] (welche zurückgezogen wurde) verwendet. Diese Normen beschreiben die Qualitätsmerkmale einer Software als Produkt, welche in der folgenden Auflistung dargestellt sind.

NFA-01: Funktionalität

Die spezifizierten Ausprägungen der funktionalen Anforderungen müssen vorhanden sein und die festgelegten Eigenschaften aufweisen.

NFA-02: Zuverlässigkeit

Die Software zeigt reproduzierbar ein korrektes Verhalten auf. Auch bei fehlerhaften Eingaben durch den Benutzer muss die Software ordnungsgemäß reagieren.

NFA-03: Benutzbarkeit

Die Anwendung ist verständlich aufgebaut und intuitiv bedienbar. Ein neuer Benutzer soll in der Lage sein, die Anwendung ohne größeren Schulungsaufwand zu bedienen. Zur Unterstützung wird dem Benutzer eine Hilfe angeboten.

NFA-04: Effizienz

Die Antwortzeit der Anwendung ist gering gehalten. Der Ressourcenbedarf liegt im Rahmen der üblichen Anforderungen für die gewählte Anwendungsumgebung.

NFA-05: Wartbarkeit

Es besteht die Möglichkeit für Änderungen an der Software, sei es Anpassungen, Verbesserungen oder Fehlerbehebungen.

NFA-06: Übertragbarkeit

Die Anwendung kann, ohne größere Anpassungen oder Aufwand, in einer neuen Umgebung funktionsfähig sein. Voraussetzung für den Betrieb ist, dass das andere Zielsystem die entsprechende Laufzeitumgebung eingerichtet hat.

NFA-07: Skalierbarkeit

Die Erweiterungen des Datensatzes und deren Verarbeitung haben keine negativen Auswirkungen auf die anderen Anforderungen des Systems.

NFA-08: Sicherheit

Die Anwendungsdaten werden mit Rücksicht auf Informationssicherheit, Datenintegrität und Verfügbarkeit abgespeichert. Sensible Daten wie der Benutzer-Login werden verschlüsselt hinterlegt.

4.3 Konzeptioneller Entwurf

Die Hauptaufgabe des Entscheidungsunterstützungstools ist es, eine Interaktion des Anwenders mit dem Werkzeug zu ermöglichen. Der Aufbau der Benutzeroberfläche und Interaktionsmöglichkeiten bestimmen wie effektiv das Werkzeug verwendet werden kann. Die Grundlagen hierzu sind in der Forschung entstanden und bauen auf Erfahrungen der Anwendungsentwickler und Anwender auf. Die Arbeiten von Shneiderman [SB03] und Nielsen [NM90] zeigen *goldene Regeln* und Heuristiken, die man bei der Mensch-Computer Interaktion beachten sollte. Folgende Auflistung zeigt einige dieser Regeln, wobei die Reihenfolge eine mögliche Priorisierung darstellt:

1. **Konsistenz:** Schriftbild, Farben, Beschriftungen, Möglichkeiten, Verhalten

2. **Darstellung:** Ansprechendes und einfaches Design, keine Informationsüberladung
3. **Rückmeldung:** (visuelles) Feedback des Systemstatus
4. **Abgeschlossenheit:** Gruppierungen, selbsterklärende und abgeschlossene Abläufe
5. **Fehlerbehandlung:** Fehler vermeiden, Fehlermeldungen einfach gestalten, Lösung vorschlagen
6. **Reversibilität:** Freiheit des Anwenders, Aktionen rückgängig zu machen
7. **System und reale Welt:** Systemsprache, Begriffe und Konzepte sollten auf natürliche Weise präsentiert werden, angepasst an den Benutzer
8. **Flexibilität:** Erfahrene Nutzer können effizient die Anwendung bedienen, neue Nutzer erhalten Hilfestellungen
9. **Unterstützung:** (optionale) Hilfe und Dokumentation bereitstellen, passend zum aktuellen Themenbereich

Diese Punkte bilden die Grundlage für das Design und Layout des Entscheidungsunterstützungstools. Die Entwürfe des folgenden Unterkapitels setzen die geforderten Regeln um, damit eine möglichst gute Benutzererfahrung erreichbar ist.

4.3.1 Mockups

Um das Layout der Oberfläche Graphical User Interface (GUI) festzulegen werden die verschiedenen Ansichten grob mit einem Entwurfswerkzeug erstellt. Es gibt eine Reihe unterschiedlicher Tools um die Modelle, auch Wireframe genannt, zu erzeugen. Neben verschiedenen Plattformen, Funktionsumfang und Nutzungsgebühr wurde zu Gunsten des Web-Tools von EverCoder Software SRL entschieden, genannt *Moqups*¹. Mit dieser nativen HTML5 Anwendung können im Webbrowser direkt Benutzeroberflächen entworfen werden. Die Bedienung ist intuitiv und die Auswahl an den Vorlagen passt optimal zu den Anforderungen des EUS-Tools:

- Text und Grafiken
- Menüleiste
- Fortschrittsleiste
- Schaltflächen
- Tabellen
- Formulare

¹<http://moqups.com/>

4 Anforderungsanalyse und Entwurf

Ein Export des Layouts ist direkt als Vektorgrafik im Portable Document Format (PDF) Format möglich. Die entstandenen GUI-Modelle werden in den folgenden Unterkapiteln erläutert.

Grundgerüst

Einen groben Überblick zu dem Grundgerüst der Webanwendung zeigt die Abbildung 4.1. Es wurde ein klassisches 2-Spalten Layout gewählt mit einer Kopf- und Fußzeile.

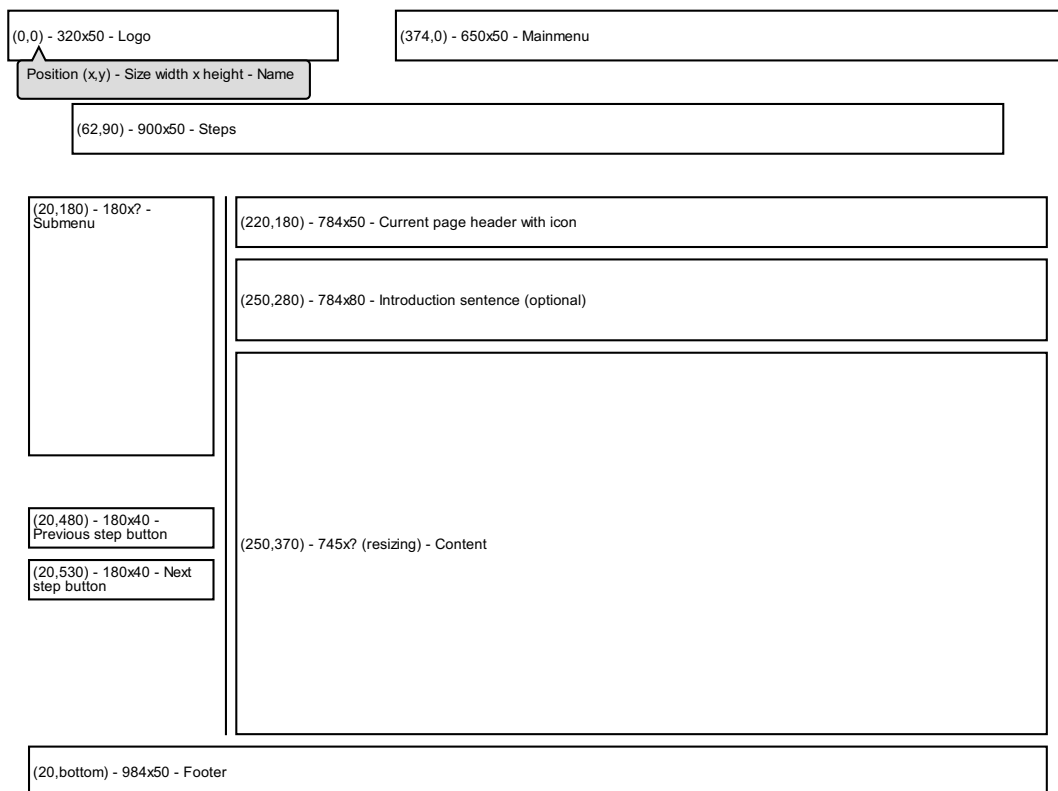


Abbildung 4.1: Mockup: Grundgerüst

Als Grundlayout wurden folgende Parameter verwendet:

- Seitenbreite: 1024 Pixel
- Seitenhöhe: 768 Pixel

Der Aufbau der Seite ist wie folgt:

Kopfbereich Beinhaltet das Logo des EUS-Tools (Name) und die Hauptnavigationsleiste mit den Unterpunkten: **Dashboard, Projects, Administration, My Account**

Aktueller Schritt im Prozess Stellt anschaulich dar, in welchem Schritt der Benutzer sich befindet. Dies orientiert sich am MIGRATE! Vorgehensmodell.

Menü der aktuellen Seite Stellt eine Navigation für die Unterschritte dar.

Vor und Zurück Schaltfläche Unterstützt den Benutzer beim Navigieren durch den Prozess. Es wird demnach der letzte beziehungsweise der nächste Schritt verlinkt.

Überschrift Ein klarer Begriff macht dem Anwender klar, auf welcher Seite er sich aktuell befindet.

Inhaltsbereich Der Hauptbereich der aktuellen Seite. Stellt die Formulare, Textinhalte oder Tabellen des aktuellen Schritts im Prozess dar.

Fußbereich Enthält ein Link zur Hilfe, Rückmeldung an den Entwickler und zum Impressum.

Hauptseite

Die Abbildung 4.2 zeigt den Aufbau der Hauptseite. Diese soll dem Anwender einen kurzen Überblick über den Prozess verschaffen und den Workflow des Tools darstellen. Es wird bewusst auf zu viel Text verzichtet. Der Anwender soll die Elemente der **Navigation** und den **Ablauf** erkennen, denn diese sind auf jeder Unterseite vorhanden. Die Grafik des Workflows ist nur auf der Hauptseite vorhanden. Diese zeigt die einzelnen Schritte im Prozess und die Ein-/Ausgabe Elemente. Damit kann der Anwender sich ein Bild davon machen, welche Informationen er bei den Schritten benötigt und was nach Erfüllung des Schritts als Ergebnis erzeugt wird.

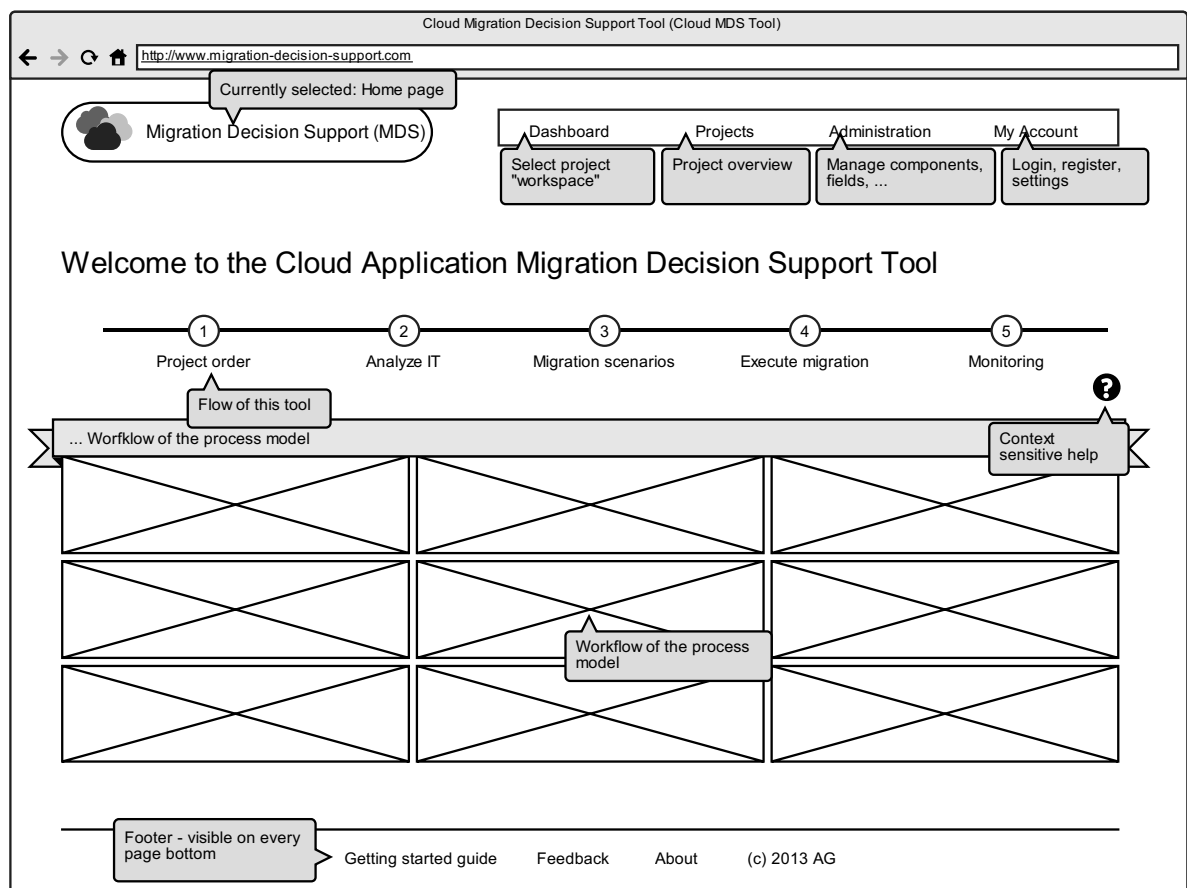


Abbildung 4.2: Mockup: Hauptseite

Dashboard

Das Dashboard ist der Arbeitsbereich des EUS-Tools. Hier navigiert der Anwender durch den Prozess, wobei der aktuelle Schritt in der Prozessleiste angezeigt ist (siehe FA-15: Aktueller Schritt). Das Mockup ist in der Abbildung 4.3 dargestellt. Der Aufbau des linken Menüs setzt sich wie folgt zusammen:

- Die Unterpunkte des aktuellen Schritts werden hierarchisch angezeigt.
- Der aktuelle Unterpunkt ist hervorgehoben.
- Eine Navigation ist nur zum aktuellen und vorherigen Schritten möglich. Um zum nächsten Schritt zu gelangen müssen die erforderlichen Informationen eingegeben werden. Über das Navigationselement **Vorwärts** unter dem Menü kann dann der folgende Unterpunkt aufgerufen werden.

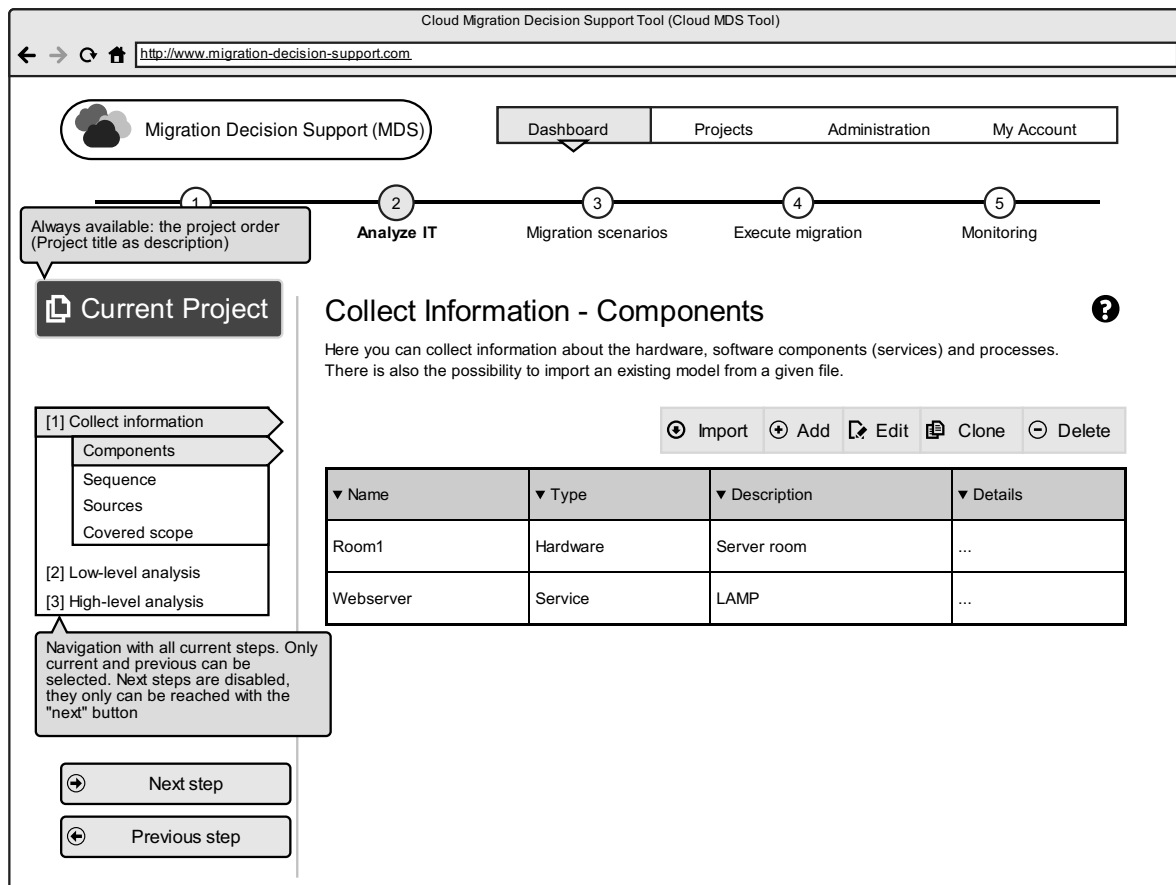


Abbildung 4.3: Mockup: Arbeitsbereich

Direkt unter dem Menü befindet sich die Vor-/Zurück-Navigation. Diese beziehen sich auf die Schritte im Prozess. Der Anwender sollte diese Schaltflächen verwenden und nicht die Browser-internen Vor-/Zurück Elemente der Navigation.

Über dem Menü befindet sich eine wichtige Schaltfläche, betitelt mit dem aktuellen Projektnamen. Hier gelangt der Anwender, wie es in den Anforderungen spezifiziert wurde, ohne Umwege zu dem Projektauftrag (siehe FA-16: Projektauftrag).

Im Inhaltsbereich wird die eigentliche Information des aktuellen Prozessschrittes angezeigt beziehungsweise abgefragt. Da die Schritte sehr unterschiedlich sind variiert die Darstellung von Tabellen, Formularen, Auflistungen, Text bis hin zu Grafiken.

Projekte

Eine Übersicht über die verfügbaren Projekte erreicht der Anwender über den zweiten Punkt im Hauptmenü. Dort werden, je nach Berechtigungen des angemeldeten Kontos, die Projekte in einer Tabelle aufgelistet. Es bestehen die Möglichkeiten zum **Laden**, **Hinzufügen**, **Bearbeiten**, **Kopieren** und **Löschen** eines Projektes. In der Tabelle werden der Name und Beschreibung angezeigt.

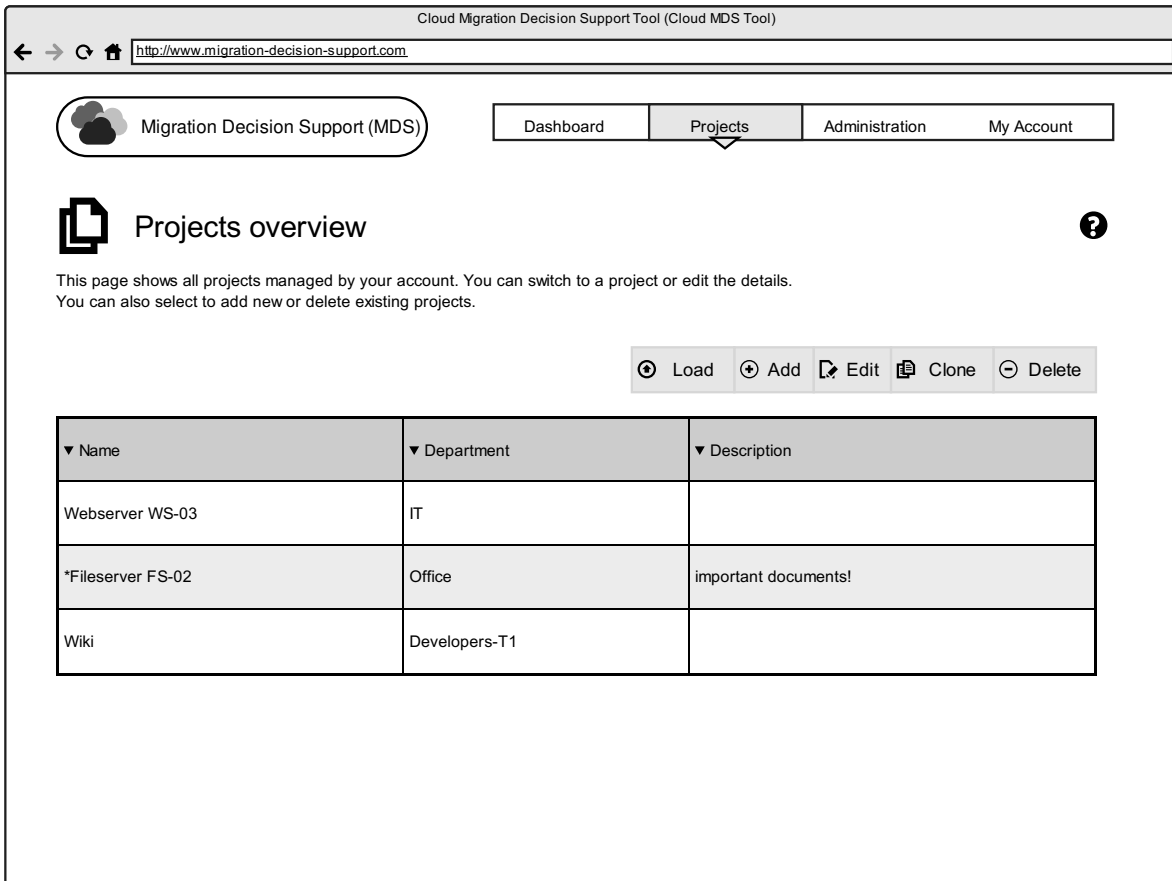


Abbildung 4.4: Mockup: Projektübersicht

Projektauftrag

Der Projektauftrag ist, über die im Abschnitt **Dashboard** erwähnte Schaltfläche, jederzeit im Arbeitsbereich erreichbar. Der Auftrag beschreibt das Projekt und enthält die nötigen Informationen, die den Benutzer bei der Entscheidungsfindung unterstützen sollen.

The mockup shows a web browser window with the URL <http://www.migration-decision-support.com>. The page title is "Cloud Migration Decision Support Tool (Cloud MDS Tool)". The navigation menu includes "Dashboard", "Projects", "Administration", and "My Account". A progress bar shows five steps: 1. Project order, 2. Analyze IT, 3. Migration scenarios, 4. Execute migration, and 5. Monitoring. The "Project order" step is currently active.

The main content area is titled "Project order" and includes a help icon. Below the title, there is a text box explaining the page's purpose: "This page shows information about the currently selected project. Enter the project goals, specifications and item covered by the order."

The form is divided into sections:

- General information**:
 - Name:
 - Reference:
 - Description:
 - Attach files:

At the bottom of the form, there are "Cancel" and "Save project" buttons. The "Save project" button is highlighted and has a "saved" status indicator (a gear icon with a checkmark).

On the left side, there is a sidebar with a "General information" section containing "Goals", "Specifications", and "Items". Below the sidebar, there is a "Previous step" button and a tooltip that says "Directing to the last step".

Abbildung 4.5: Mockup: Projektauftrag

Administration

Im Administrationsbereich können die **Basisdaten** angepasst und **Benutzer** verwaltet werden. Das Mockup in Abbildung 4.6 zeigt eine beispielhafte Eingabemaske zur Bearbeitung der Kriterien. Die bearbeitbaren Elemente wurden in der Anforderungen FA-11: Erweiterbarkeit festgelegt und umfassen deshalb zum Entwicklungszeitpunkt:

- **Kriterien:** Die Gruppen sind fest, jedoch können Einträge der Gruppen hinzugefügt werden.
- **Infrastrukturen:** Die Auswahl der Anbieter, Dienst- und Bereitstellungstypen ist zum Entwicklungszeitpunkt aktuell, jedoch soll die Webanwendung zukunftssicher sein und daher ist ein hinzufügen oder bearbeiten der Infrastrukturen möglich.

The screenshot displays the 'Administration -> Master data' interface. At the top, there's a navigation bar with 'Dashboard', 'Projects', 'Administration', and 'My Account'. The main content area is titled 'Administration -> Master data' and includes a sub-header 'Criteria'. Below this, a form allows editing a criterion named 'TCO analysis'. The form fields are: Name (TCO analysis), Type (Textfield / Name-Value), Parent criteria (Financial criteria), and Description (total cost of ownership). A 'Save item' button is present, and a 'saved' status is shown. A sidebar on the left contains 'Master data' with sub-items 'Criteria', 'Target platforms', and 'Users'. At the bottom left, there are 'Next step' and 'Previous page' buttons.

Abbildung 4.6: Mockup: Administrationsbereich

Benutzerkonto

Jeder angemeldete Benutzer kann seine persönlichen Einstellungen bearbeiten. Dazu gehören:

- Name
- E-Mail Adresse
- Beschreibung
- Passwort
- Spracheinstellung

Die Einstellungen sind für die Anmeldung und für den Administrator zur Verwaltung gedacht. Weitere Einstellungen sind zum Stand der Entwicklung nicht erforderlich, jedoch können diese in der Webanwendung erweitert werden.

The mockup shows a web browser window titled "Cloud Migration Decision Support Tool (Cloud MDS Tool)" with the URL "http://www.migration-decision-support.com". The navigation menu includes "Dashboard", "Projects", "Administration", and "My Account". The main heading is "My account / settings" with a user profile icon and a help icon. Below the heading is the text: "Here you can change your account information, user settings and login data." There are three settings panels:

- Account settings:** Includes fields for "Name" (User), "E-Mail address" (user@somewhere.com), and "About you" (Test user). Buttons: "Cancel", "Save settings", and a "saved" status indicator.
- User settings:** Includes a "Language" dropdown menu set to "English / German". Buttons: "Cancel", "Save settings", and a "saved" status indicator.
- Password settings:** Includes fields for "Current password", "New password", and "Confirm new password", all masked with "***". Buttons: "Cancel", "Change password", and a "saved" status indicator.

Abbildung 4.7: Mockup: Benutzerkonto

4.4 Datenmodell (Domain Model)

Aus der vorangegangenen Analyse der Anforderungen und dem konzeptionellen Entwurf soll ein umfassendes Datenmodell erstellt werden. Hierbei müssen die vielseitigen Einsatzgebiete und Anwendungsfälle berücksichtigt werden.

Um die zuvor erörterten (Nicht-) Funktionalen Anforderungen mit dem Entscheidungsunterstützungstool umzusetzen ist eine Modellierung der gesamten Aufgabenstellung erforderlich. Dies wird mit Hilfe von der Unified Modeling Language (UML), genauer gesagt einem Klassendiagramm, ermöglicht. Das so genannte *Domain Model* (siehe Abbildung 4.8) beschreibt als konzeptuelles Modell die Zusammenhänge einer Problemlösung.

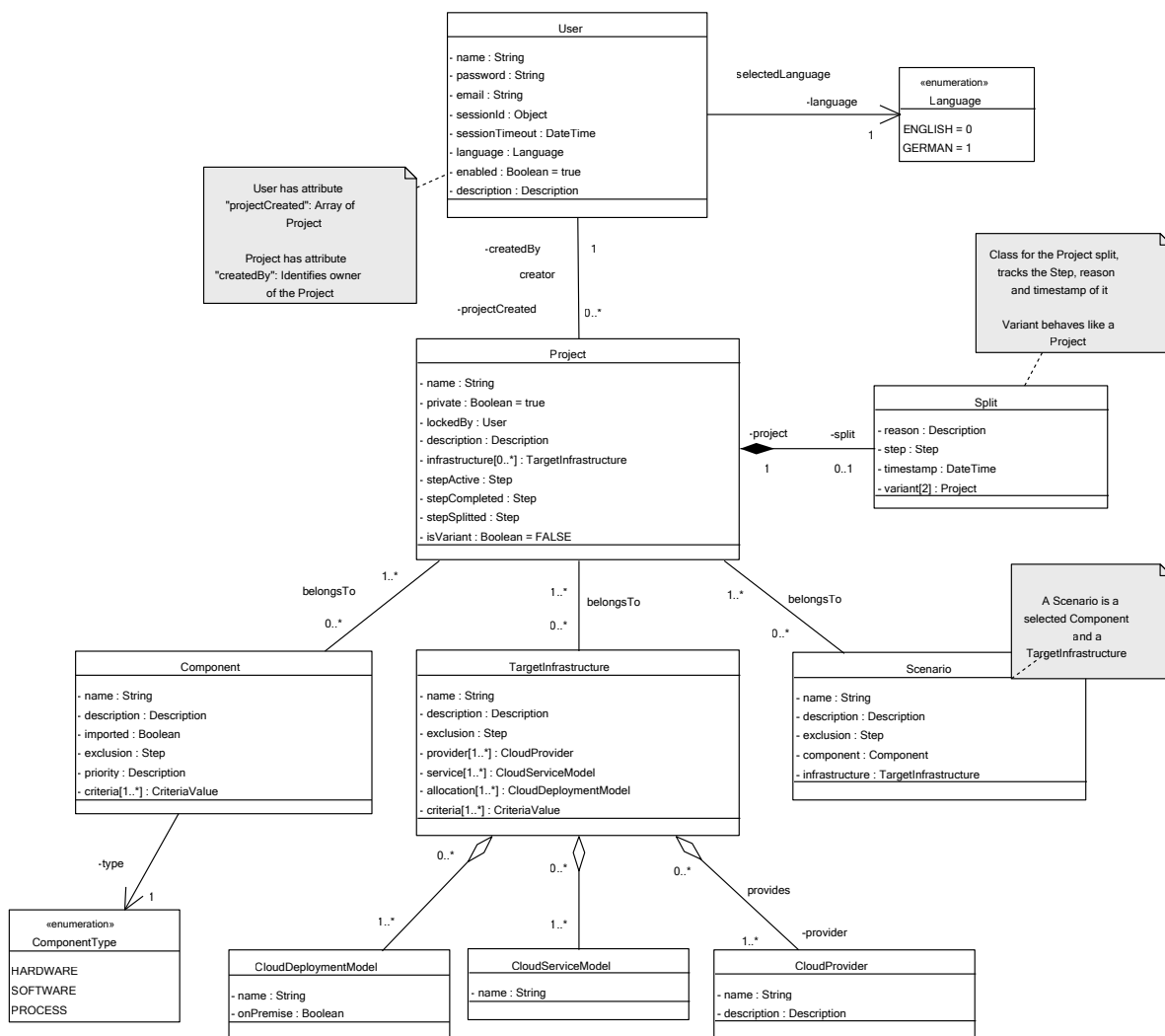


Abbildung 4.8: Domain Model der Webanwendung

Darüber hinaus gibt es Elemente welche nicht im Domain Model dargestellt sind, aber in den einzelnen Klassen vertreten sind. Die folgenden Unterkapitel beschreiben diese Elemente.

4.4.1 Der Schritt

Die Schritte im Migrationsprozess werden als Aufzählung gehandhabt (siehe Abbildung 4.9). Dabei wurden alle sinnvollen Einzelschritte im Vorgehensmodell übernommen, welche auch eine Repräsentation in der Anwendung haben. Die Abfolge der Schritte ist linear und durch die Nummerierung ist ein Vergleich möglich.

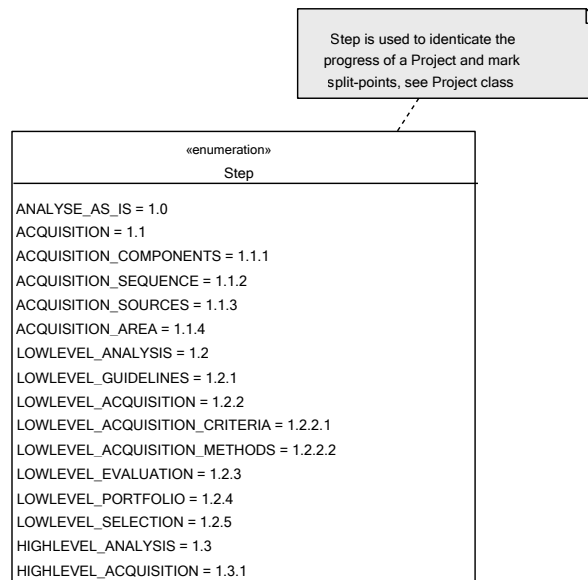


Abbildung 4.9: Schritte im Migrations-Prozess (Step)

4.4.2 Das Kriterium

Ein Kriterium stellt einen Wert dar, um verschiedene Komponenten des Systems zu bewerten (siehe Abbildung 4.10). Dabei ist der **Typ** eines Kriteriums entweder eine Zahl, ein boolescher Ausdruck oder ein Text. Dies ist notwendig um die Kriterien, und damit die Gegebenheiten einer Infrastruktur, später vergleichen zu können. Ein **Gewicht** kann eine zusätzliche Priorisierung des Kriteriums darstellen.

Die einzelnen Gruppen, die auch im Vorgehensmodell beschrieben sind, fassen die Kriterien thematisch zusammen. Es sind beliebig viele Kriterien in einer Gruppe vertreten, wobei jedoch ein Kriterium nur genau einer Gruppe zugehörig ist.

Vergleichbare Elemente, wie Komponenten und Ziel-Infrastrukturen, haben die Kriterien und einen zugehörigen Wert. Über diesen Satz an Kriterien können die Vergleiche stattfinden.

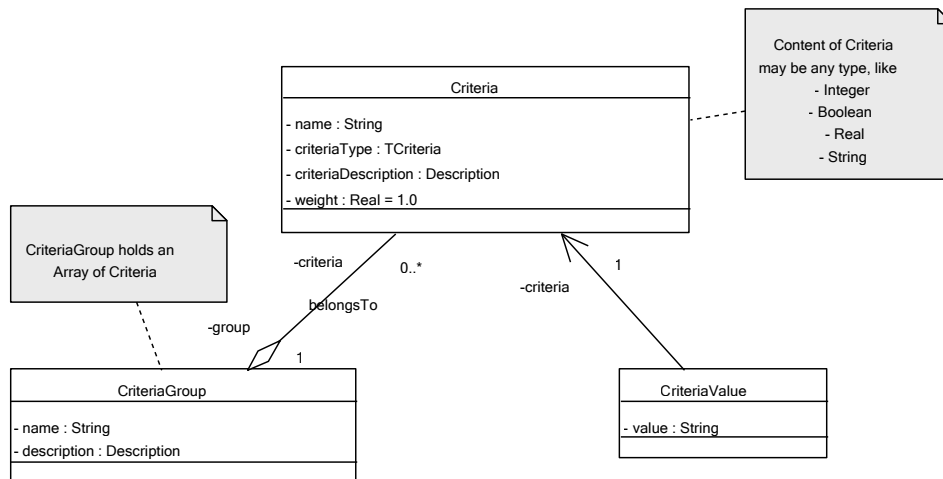


Abbildung 4.10: Modell der Kriterien (Criteria)

4.4.3 Die Ausschließung

Nach dem MIGRATE! Vorgehensmodell ist es an verschiedenen Stellen möglich Migrationskandidaten oder Komponenten auszuwählen. Die dabei ausgeschlossenen Elemente werden durch die Klasse der Ausschließung gekennzeichnet. In den folgenden Schritten nach der Ausschließung erscheint das Element nicht mehr im Prozess. Festgehalten werden jeweils:

- **Schritt** der Aktion
- **Grund** als Beschreibung
- **Zeitpunkt**

Somit kann nachvollziehbar dokumentiert werden, warum der Migrationskandidat oder Komponente nicht weiter im Prozess betrachtet wird.

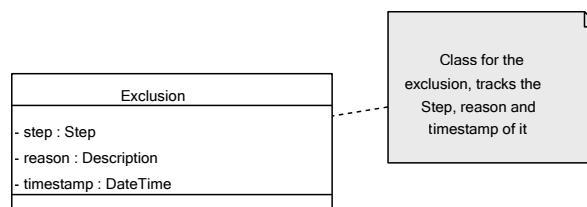


Abbildung 4.11: Modell der Ausschließung (Exclusion)

4.4.4 Die Beschreibung

Eine Beschreibung ist ein allgemeines Feld für verschiedene Elemente, um einen **Text** daran anzuhängen (siehe Abbildung 4.12). Beispielsweise können damit mehr Details für einzelne Komponenten beschrieben, Entscheidungen begründet oder Dokumente angehängt werden. Bei den folgenden Klassen ist es möglich eine Beschreibung hinzuzufügen. Der Informationsgehalt sollte sich auf Details zu dem jeweiligen Objekt beziehen:

- Benutzer
- Projekt
- Komponente
- Infrastruktur
- Szenario
- Cloud Anbieter
- Kriterium
- Kriterium Gruppe

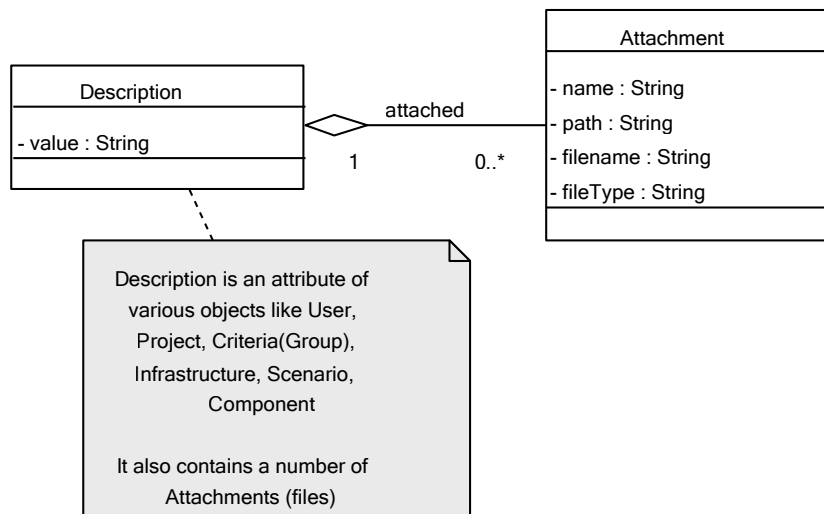


Abbildung 4.12: Model der Beschreibung (Description)

Bei bestimmten Klassen ist es sinnvoll, einen **Grund** anzugeben um die Entscheidung, eventuell für andere Personen oder zur besseren Nachvollziehbarkeit, schlüssig zu erklären. Dies dient auch der Dokumentation im Entscheidungsprozess, um die Begründung hierfür festzuhalten. Die Klassen welche einen Grund erfordern sind die **Aufteilung (Split)** und die **Ausschließung (Exclusion)**.

Zu jeder Beschreibung kann eine beliebige Anzahl von **Anhängen** hinzugefügt werden. Es gibt eine Einschränkung für den Dateityp damit nur Dokumente, Tabellen oder Grafiken angehängt werden können.

4.5 Einbindung des bestehenden Modells

In einer vorangegangenen Arbeit im Rahmen des MIGRATE! Projekts wurde bereits ein Datenmodell erstellt. Die Grundlage bildet das Eclipse Modeling Framework (EMF). Mit dem Vorteil, direkt aus den Modellen Java Code zu generieren, können die bereits vorhandenen Modelle besser importiert werden.

Ein wichtiger Bestandteil des EMF ist das Ecore-Metamodell. Damit lassen sich Objektmodelle beschreiben, welche von verschiedenen Formaten importiert werden können. Als Export stehen ebenso mehrere Dateiformate zur Auswahl. Das Ecore-Metamodell besteht aus den Entitäten:

- **EClass:** steht für eine Klasse mit Namen, besitzt Attribute und Referenzen
- **EAttribute:** Ein Attribut besteht aus einem Namen und einem Typ
- **EReference:** Beschreibt eine Beziehung zwischen zwei Klassen
- **EDataType:** Steht für den Typ eines Attributs

Mit Hilfe des Metamodells lassen sich für unseren Anwendungsfall die einzelnen Komponenten modellieren. Ein Beispiel das im Rahmen des MIGRATE! Projekts entstanden ist, zeigt die Abbildung 4.13. Es wird zuerst die Klasse *Nodes* beschrieben, welche allgemeine Angaben zum Name und dem maximalen Wattverbrauch enthält. Diese Attribute werden von den Klassen *ServerNode*, *ClientNode* und *NetworkNode* geerbt. Dazu gibt es noch einen übergreifenden Raum, in welchem die Knoten untergebracht sind, die Kühlung und Stromversorgung.

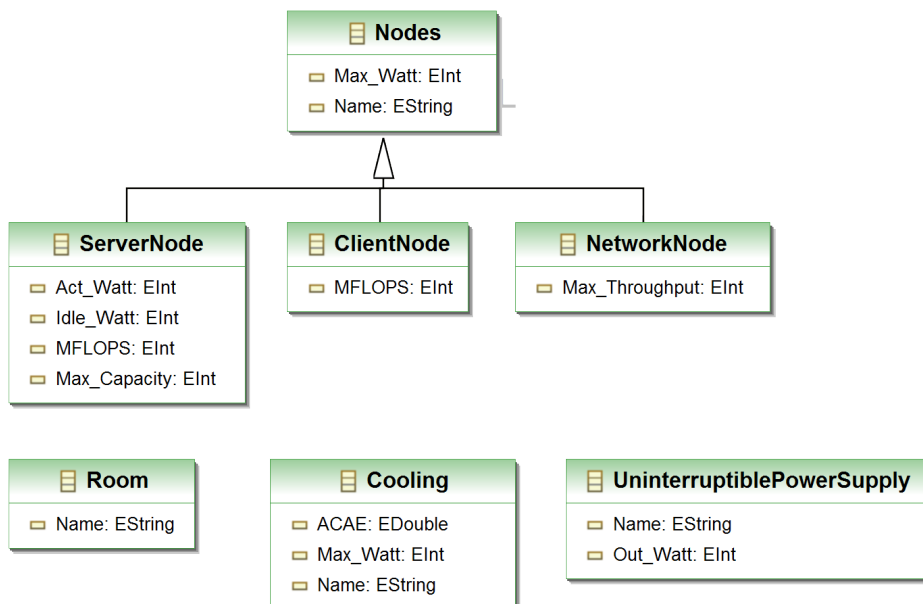


Abbildung 4.13: Beschreibung der Objektmodelle als Ecore Diagramm

Zu dem genannten Modell kann man die passenden Objekte erzeugen. Eine mögliche Zusammenstellung ist in Abbildung 4.14 skizziert. Das Beispiel stammt aus dem **MIGRATE!** Projekt, es zeigt eine Topologie aus einem **Netzwerk** und fünf **Server**. Diese Objekte können über die Eclipse Umgebung importiert werden. Damit sind die Klassen beziehungsweise die Objekte direkt verfügbar und es ist möglich, diese in das Entscheidungsunterstützungstool zu laden.

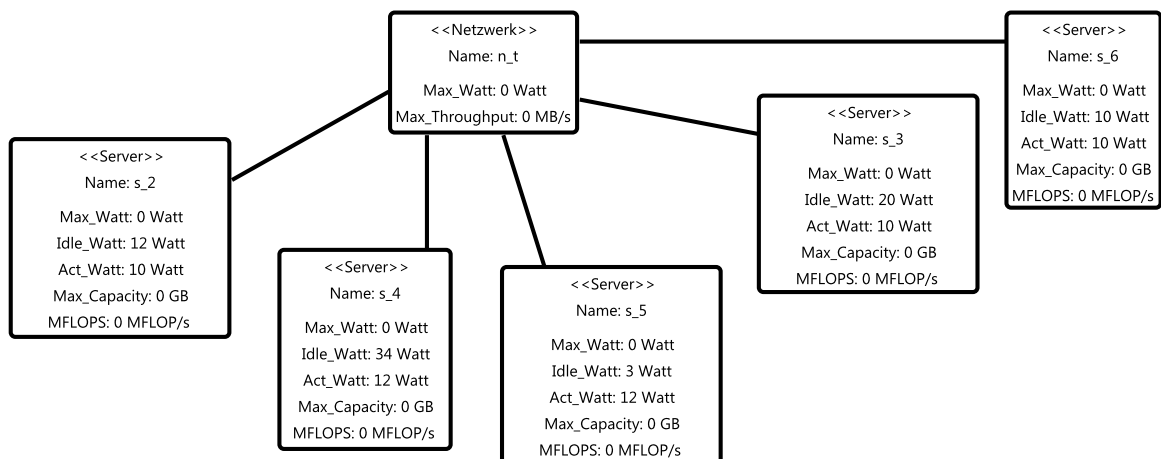


Abbildung 4.14: Beispieltopologie aus dem EAM-Modell

Die Attribute der Objekte werden als Text in das Beschreibungsfeld einer Komponente übernommen. Der Name kann direkt in das entsprechende Feld gespeichert werden und der Wert des *imported*-Attributes auf *TRUE* gesetzt werden. Als Referenz werden die Quelle und der Zeitpunkt des Imports notiert.

Implementierung des Tools

Dieses Kapitel begleitet den praktischen Teil der Arbeit – die prototypische Implementierung des Entscheidungsunterstützungstools. Der Entwurf aus dem vorangegangenen Kapitel wird in ein benutzbares Werkzeug umgesetzt. Die Anwendung wird erläutert und ein Testszenario aufgezeigt.

5.1 Prototypische Implementierung

Die Implementierung der Webanwendung wurde selbstständig und als Top-Down Prozess gestaltet. Nach der Grundlagenrecherche (Kapitel 2) und einem Überblick zu dem Stand der Technik (Kapitel 3) wurde die Anforderungsanalyse durchgeführt. Die funktionalen Anforderungen wurden zusammen mit dem Betreuer, der als Schnittstelle zwischen Entwickler und Kunden gesehen werden kann, erarbeitet. Als Hilfestellung wurden die Punkte der FA in Kategorien aufgeteilt:

- Allgemein: Grobe Funktionen des EUS-Tools , Rahmenbedingungen
- Hilfe: Ein Hilfesystem für unerfahrene Nutzer, Dokumentation
- Projekte: Projektverwaltung und Aufteilung
- Auswertung: Hilfestellung zur Entscheidung
- Dokumentenverwaltung: Anhänge für Beschreibungen, Entscheidungen, Komponenten
- Anpassungen: Erweiterbarkeit des Tools
- Datensatz: Stammdaten und Beispielprojekt
- Layout: Darstellung der Navigation, Aufbau der Seite
- Mehrsprachigkeit: Grundlegende Funktion für Mehrsprachigkeit
- Benutzer: Anmeldung zu einem Benutzerkonto und Mehrbenutzerfähigkeit

Diese Punkte werden im Folgenden nochmals aufgefasst um die Methoden für die Implementierung besser zu erklären.

Dank bereits vorhandenen Kenntnissen in der Software- und Webentwicklung waren die NFA schnell festgelegt. Sie orientieren sich an gängigen Standards und Methoden zur Softwareentwicklung.

Die einzelnen Kategorien wurden mit Berücksichtigung von gängigen Standards und Best-Practices im Gebiet der Online-Informationssysteme umgesetzt.

Allgemein: Der *Ablauf* der Webanwendung war im Vorfeld durch das Vorgehensmodell bestimmt. Die Reihenfolge der Aktivitäten und dazugehörigen Vor- und Nachbedingungen wurden gemäß der Vorgabe nachgebildet. Das erforderliche *Datenmodell* bildet die nötigen Klassen ab, welche im Rahmen der Anforderungsanalyse (siehe Kapitel 4) erörtert wurden. Im Hintergrund der Anwendung muss eine persistente Speicherung der Daten möglich sein. Die dazugehörige Umsetzung wurde durch das verwendete Framework erleichtert (Annotationen der Klassen). Während der Entwicklung wurde die Anwendung mit einer MySQL-Datenbank getestet, dies ist aber über eine Konfigurationsdatei änderbar.

Hilfe: Das Hilfesystem ist in Form einer unaufdringlichen Grafik auf der jeweiligen Webseite zu finden. Hier wird abhängig vom Inhalt ein entsprechender Eintrag aufgerufen, welcher die Bedienung der Anwendung erklärt und auf Probleme hinweist.

Projekte: Ein Projekt ist eine abgeschlossene Einheit. Der Anwender kann somit verschiedene Situationen, Szenarien oder Entscheidungen ausprobieren und mit dem EUS evaluieren. Dem angemeldeten Benutzer werden die Projekte angezeigt für welche er Zugriff hat. Zu bestimmten Zeitpunkten, wenn die Informationsabfrage für den Schritt vollständig ist, kann ein Projekt aufgeteilt werden. Die dabei entstehenden Teilprojekte (genannt *Variante*) können bei der Auswertung verglichen werden. Somit kann eine direkte Gegenüberstellung von Auswirkungen der Entscheidung visualisiert werden.

Auswertung: Zur Evaluation der Entscheidungen ist es möglich eine Gewichtung für einzelne Komponenten zu vergeben. Dieser Faktor beeinflusst die Darstellung des Elements in der *Portfolio*-Ansicht. Durch diese Ansicht können die Komponenten identifiziert werden, welche bedeutsam für den Unternehmenserfolg sind und deren Energieeffizienz.

Dokumentenverwaltung: Die Möglichkeit zum Anhang verschiedener Dokumente an Beschreibungen oder Komponenten wird über ein Formular umgesetzt. Bei der Abfrage der Informationen kann der Anwender ein Dokument auswählen und über die Upload-Funktion auf dem Webserver hinterlegen. Dieses Dokument wird dann als Referenz dem zuvor bearbeiteten Element angeheftet.

Anpassungen: Durch die Erweiterbarkeit der Webanwendung soll diese zukunftssicher und individualisierbar sein. Als administrativer Benutzer können im Administrationsbereich die Kriterien und Infrastrukturen festgelegt werden. Als Entwickler kann der dokumentierte Quellcode der Anwendung angepasst werden. Sollte beispielsweise eine

Änderung im Vorgehensmodell oder neue Attribute der Komponenten erforderlich sein, so kann dies bei Vorlage der Quelldateien implementiert werden.

Datensatz: Bei Einrichtung der Anwendung ist bereits ein Basisdatensatz mit Kriterien und Infrastrukturen hinterlegt. Zudem kann ein Benutzer den Import von vordefinierten Komponenten anstoßen. Das ermöglicht eine schnelle Einfeldung in die Webanwendung und erlaubt dem Benutzer mit dem Komponenten zu experimentieren.

Layout: Das Layout ist gemäß dem konzeptionellen Entwurf (siehe Kapitel 4.3) umgesetzt. Als Technik werden Templates eingesetzt. Die einzelnen Bereiche der Webseite (siehe Abbildung 4.1: Mockup: Grundgerüst) sind als getrennte Java Server Pages (JSP) definiert. Somit erreicht man eine hohe Modularität und kann die Darstellung als Entwickler leichter anpassen.

Mehrsprachigkeit: Dem Anwender stehen die Sprachen Englisch und Deutsch zur Auswahl. Das Spring Framework bietet eine Internationalisierung an. Die Textelemente befinden sich in einer *.properties*-Datei, in den JSP-Dateien werden diese Sprachneutral referenziert. Zur Laufzeit werden dann mit den Sitzungsinformationen des Anwenders die passenden Textelemente angezeigt.

Benutzer: Es gibt die Möglichkeit ein eigenes Benutzerkonto zu erstellen und sich damit anzumelden. So ist eine getrennte Bearbeitung von Projekten möglich, sowie der parallele Betrieb. Es darf keine Beeinflussung von anderen Benutzern geben, wenn mehrere angemeldet sind. Für ein Projekt wird daher eine *locked* Markierung gesetzt, wenn ein Anwender gerade daran arbeitet. Damit kann kein anderer Benutzer das Projekt öffnen.

Die einzelnen Klassen wurden mit einem UML Diagramm modelliert und damit die Verbindungen zwischen den Elementen definiert. Als Beispiel sei hier die Klasse der Komponente veranschaulicht (siehe Abbildung 5.1).

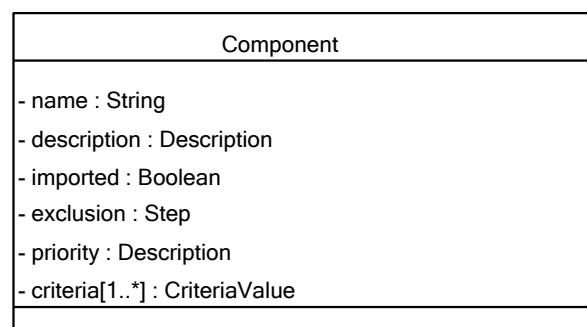


Abbildung 5.1: Modell der Komponente (Component)

Die Komponente ist identifizierbar durch den *Namen* und eine eindeutige *Nummer* (ID, hier nicht abgebildet). Wie bei anderen Klassen auch gibt es hier die Möglichkeit eine *Beschreibung* hinzuzufügen. Das kann ein Text sein oder ein Dokument welches dem Objekt mit angehängt werden kann. Der boolesche Wert *imported* zeigt ob die Komponente ursprünglich importiert

worden ist. Das dient dem Anwender als Hinweis, woher die Komponente stammt. Während des Prozesses gibt es die Möglichkeit verschiedene Elemente auszuschließen. Durch den Eintrag *exclusion* kann der Schritt festgehalten werden, ab wann das Element in dem Migrationsprozess nicht mehr beachtet werden soll. Für die *Gewichtung* der Komponente kann ein weiterer Eintrag als Beschreibung (Text, Dateianhang) hinterlegt werden. Die umfangreichste Information zu einer Komponente sind die **Kriterien**. Aus der vordefinierten Auswahl kann hier ein Kriterium und ein dazugehöriger Wert eingetragen werden. Die Struktur der Komponenten-Klasse ist im folgenden Listing 5.1 aufgeführt:

```
package com.migrationdecisionsupport.classes;

/* import for persistence annotations */

@Entity
@Table(name = "COMPONENT")
public class Component implements Serializable {

    private static final long serialVersionUID = 2074962091118813715L;

    @Id
    @Column(name = "ID")
    @GeneratedValue
    private Integer id;

    @Column(name = "NAME")
    private String name;

    @Column(name = "DESCRIPTION")
    private Description description;

    @Column(name = "IMPORTED")
    private Boolean imported = false;

    /* ... following the other attributes */
}
```

Listing 5.1: Aufbau der Komponenten-Klasse

Durch die Annotationen wird die Klasse als Komponente markiert und die Attribute mit dem Tabellenlayout in der Datenbank verknüpft. Zudem sind die üblichen Get- und Set-Methoden implementiert, um die Attribute zu speichern und auszulesen.

Eine Schnittstelle zur Datenbank (siehe Listing 5.2) ermöglicht das Speichern, Auslesen, Auflisten und Löschen eines Objekts (die sogenannten Create, Read, Update, Delete (CRUD) Operationen). Die Implementierung der Schnittstelle wird durch ein Data Access Object (DAO) realisiert, das mit Hilfe einer *SessionFactory* und Hibernate den Datenzugriff ermöglicht.

```
package com.migrationdecisionsupport.dao;

import org.hibernate.SessionFactory;
import com.migrationdecisionsupport.classes.Component;
```

```
public interface ComponentDAO {  
    // CRUD interface  
    public void save(Component c);  
    public List<Component> getAll();  
    public Component get(Integer id);  
    public void delete(Component c);  
}
```

Listing 5.2: Datenbankinterface der Komponenten-Klasse

5.1.1 Entwicklungsumgebung

Laut der Ausschreibung der Arbeit war eine Bedingung, dass Java als Programmiersprache verwendet wird. Java ist eine objektorientierte Sprache, welche mit dem Java Development Kit (JDK) und der Java Runtime Environment (JRE) die Java-Technologie bilden. Damit wurde eine Entwicklungsumgebung geschaffen, welche den Anforderungen verschiedenster plattformübergreifender Programme gerecht werden kann. Diese Arbeit befasst sich mit dem Themengebiet der Java-Webanwendungen. Neben der bereits erwähnten JDK und JRE benötigt man zur komfortablen Entwicklung eine Integrated Development Environment (IDE). Beliebte Umgebungen sind beispielsweise Eclipse¹ oder NetBeans². Die hier verwendete IDE basiert auf Eclipse und enthält bereits mehrere Erweiterungen und Anpassungen für das verwendete Framework (siehe Kapitel 5.1.2).

5.1.2 Das Spring Framework

Eine Einheitliche Umgebung für die Entwicklung von Java Anwendungen bietet das Spring Framework³. Diese Anwendungsplattform wurde von Johnson [Joh03] eingeführt und bietet einen umfassenden Funktionsumfang für die moderne Java Programmierung. Komplexe Methoden und Konfigurationen werden vermieden, Stattdessen soll sich die Entwicklung auf das wesentliche konzentrieren. Zur Modellierung der Geschäftsobjekte können einfache Java Plain Old Java Object (POJO)s verwendet werden. Die wichtigsten Funktionen [JHD⁺13] die das Framework mitbringt sind:

- Dependency Injection mit Extensible Markup Language (XML) und Annotationen zur Konfiguration
- Aspekt-orientierte Programmierung
- Deklarative Transaktionen, Caching, Validierung und Formatierung
- Abstraktionen für Java Enterprise Edition (EE) Spezifikationen wie Java Database Connectivity (JDBC) und Java Persistence API (JPA)

¹<http://www.eclipse.org/>

²<http://netbeans.org/>

³<http://www.springsource.org/>

5 Implementierung des Tools

- Unterstützung für andere Frameworks wie Hibernate
- Web Framework um Model View Controller (MVC) Anwendungen und Dienste zu erstellen
- Testfunktionen (Unit und Integration Tests)

Der modulare Aufbau von Spring ermöglicht es die benötigten Komponenten selektiv für die Zielanwendung einzubinden. Als Entwicklungsumgebung steht die auf Eclipse basierende Spring Tool Suite (STS) zur Verfügung. Diese bietet bereits angepasste Ansichten, Integration eines Apache Tomcat Webservers, graphische Werkzeuge, Hilfestellungen und Unterstützung bei der Entwicklung.

In Abbildung 5.2 sind die einzelnen Module des Spring Framework aufgelistet. Für die Entwicklung des Migration Decision Support (MDS) Tools werden die folgenden Module verwendet:

Web Eine Web- und Servlet-Komponente, für die Verarbeitung von HTTP-Anfragen, Formularen und deren Darstellung

Data Access Zugriff auf eine Datenbank mittels JDBC und der Abbildung von Objekten über ORM. Zur Absicherung der Datenspeicherung werden Transaktionen verwendet.

Core Container Grundfunktionen des Frameworks, Verwaltung von Beans und Context, Anwendungslogik

Test Unit Tests zur Überprüfung der Funktionalität

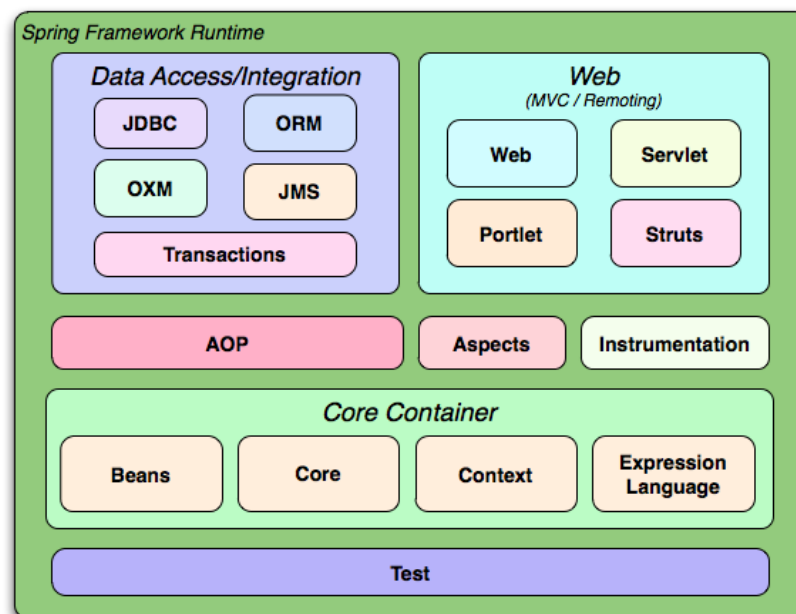


Abbildung 5.2: Übersicht des Spring Framework, verfügbare Module

5.1.3 Aufbau der Webanwendung

In einem Projekt mit dem Titel *mds-tool* werden die Klassen, Tests, Abhängigkeiten, Bibliotheken und Dateien zusammengefasst verwaltet. In den folgenden Unterkapiteln werden **Abhängigkeiten** und **Daten** der Anwendung näher erklärt.

Maven Abhängigkeiten

Um die zunehmende Komplexität von Java-Abhängigkeiten und dem Erstellungsprozess eine einheitliche Form zu geben wurde das Apache Maven⁴ Build-Management-Tool entwickelt. Es unterstützt den Entwickler in den Phasen der Kompilierung, Testen und Verteilen der Anwendung. Die Unterstützung der Eclipse IDE ist sehr gut und erlaubt damit, Komplikationen bei den genannten Phasen zu beseitigen. Die zentrale Einstellungsdatei ist die `pom.xml` in welcher die Abhängigkeiten als Artefakt beschrieben werden. Zur Demonstration zeigt das Listing 5.3 die Abhängigkeit zu den MVC Komponenten des Spring Frameworks. Die Gruppe, Artefakt-Nummer und Version-Nummer erzeugt eine eindeutige Referenz auf ein Java Klassenbibliothek. Diese wird automatisch geladen und beim Ausführen oder Übersetzen verwendet.

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

Listing 5.3: Abhängigkeit in einer `pom.xml` Datei

Daten der Webanwendung

Die eigentlichen Daten für die Webanwendung befinden sich in Ordner `/src/main/webapp/`. Dieser ist noch unterteilt in die Ordner:

- `resources`: Grafiken, Cascading Style Sheets (CSS), und JavaScript (JS) Dateien
- `WEB-INF`: Spring XML Konfigurationen, JSP Views

Unter dem Ressourcen-Ordner findet man sämtliche Grafiken der Webanwendung. Zur Vereinfachung der Gestaltung wurde das Bootstrap⁵ verwendet, welches eine Sammlung vieler Gestaltungsvorlagen für Typographie, Formulare, Schaltflächen, Tabellen, Navigationselemente und das komplette Seitenlayout anbietet. Die dazu benötigten Dateien (CSS, JS und Grafiken) liegen in einem separaten Ordner. Zusätzlich wird noch die jQuery⁶ Bibliothek

⁴<http://maven.apache.org/>

⁵<http://twitter.github.io/bootstrap/>

⁶<http://jquery.com/>

eingebunden. Diese bietet umfangreiche JavaScript Funktionen und viele Möglichkeiten für eine dynamische Webanwendung.

Für das Spring Framework ist die `web.xml` Datei im `WEB-INF` Ordner von großer Bedeutung. Dort ist die Weiche für die Verarbeitung der Anfragen definiert, in unserem Fall ist die Architektur nach dem MVC Stil implementiert. Das Servlet das die Anfragen bearbeitet ist das *DispatcherServlet* (siehe Listing 5.4) des Spring Web Moduls.

```
<!-- Dispatcher Servlet is the point for fetching and replying requests -->
<ervlet>
  <!-- bean file is named 'servlet-name'-servlet.xml -->
  <servlet-name>mds</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <load-on-startup>1</load-on-startup>
</servlet>
```

Listing 5.4: DispatcherServlet in der `web.xml` Datei

Die Anfragen werden von dem *FrontController* an den jeweiligen Controller weitergeleitet (siehe Abbildung 5.3), welcher über die Annotation *@RequestMapping* die Verarbeitung einer Ressource übernimmt. Dort wird das entsprechende Modell erzeugt und über den *FrontController* an das *ViewTemplate* weiter gereicht. Dieser erzeugt in unserem Fall über das Apache Tiles Framework⁷ eine verarbeitete Ansicht des gewählten Templates.

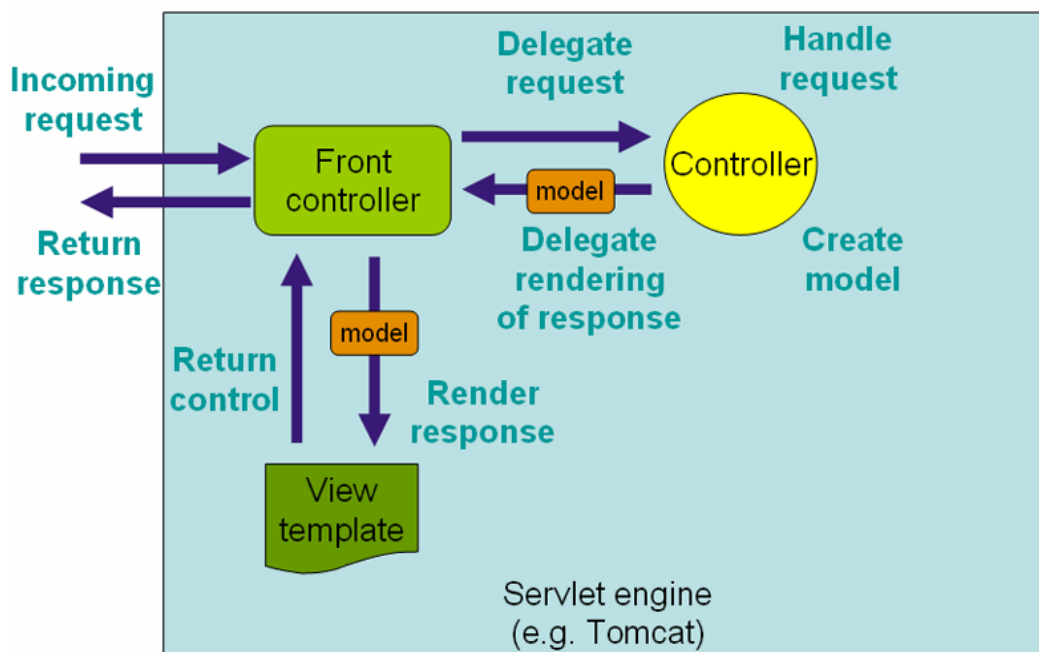


Abbildung 5.3: Ablauf einer Anfrage im Spring Web MVC Modul

⁷<http://tiles.apache.org/>

5.2 Ein Testszenario

Das Testszenario basiert auf den Stammdaten (siehe FA-12: Stammdaten) und beinhaltet ein vordefiniertes Projekt mit Komponenten, einer Infrastruktur und einem Migrationsszenario. Der grobe Aufbau der Komponenten ist in Abbildung 5.4 dargestellt. Es ist ein Raum modelliert, welcher Arbeitsplatzrechner und verschiedene Server(-dienste) enthält. Die Dienste haben unterschiedliche Prioritäten und Anforderungen, was ein mögliches Migrationsszenario beschreibt.

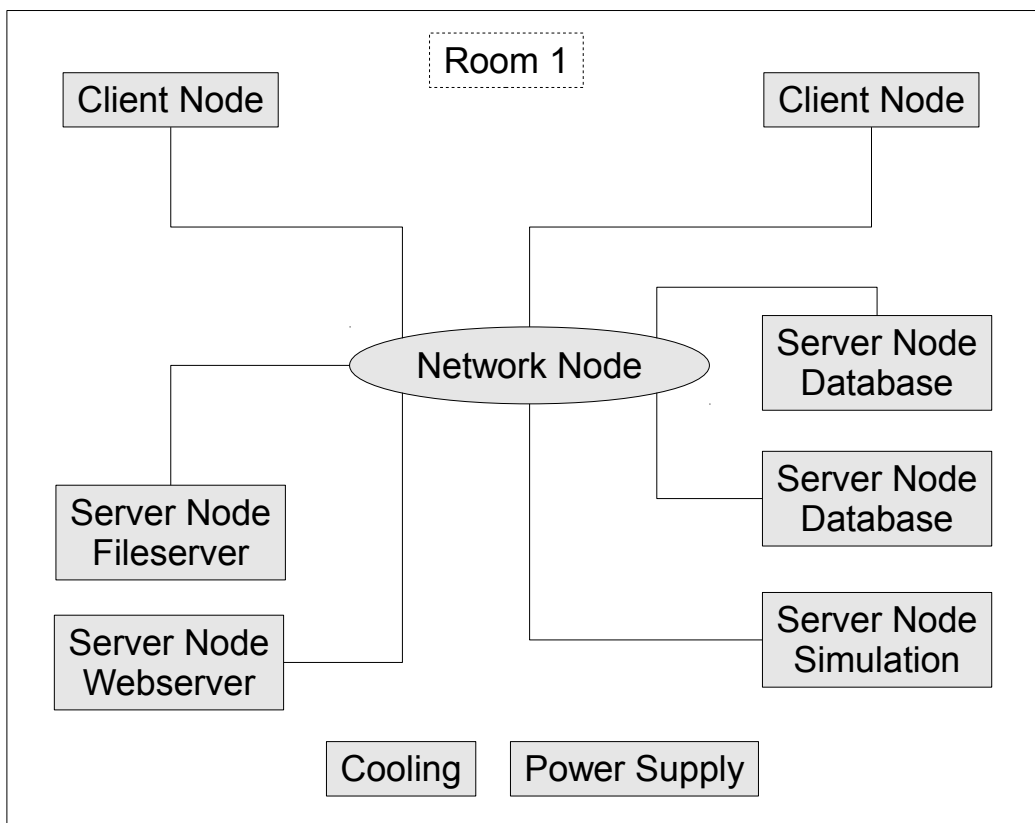


Abbildung 5.4: Aufbau des Testszenarios

Das Testszenario kann von einem Benutzer geladen werden, um eine vorgefertigte Auswahl an Komponenten verfügbar zu haben. Darauf aufbauend können natürlich Anpassungen und Erweiterungen vorgenommen werden. Mit diesem Testszenario soll vor allem der Einstieg in das Entscheidungsunterstützungstool erleichtert werden.

Zusammenfassung und Ausblick

Dieses abschließende Kapitel fasst die gewonnenen Erfahrungen und Erkenntnisse zusammen, die aus der Erstellung des Tools entstanden sind. Anschließend wird eine mögliche zukünftige Entwicklung der Thematik betrachtet

6.1 Zusammenfassung

Die verschiedenen Aspekte rund um die Cloud und Migration von Anwendungen werfen eine Vielzahl an Entscheidungen auf. Möchte man diese Entscheidungen nachvollziehbar und dokumentiert treffen, so bietet sich die Einbeziehung eines Entscheidungsunterstützungssystems an. Grundlegende Überlegungen, worauf bei einer Migration geachtet werden sollte, sind die Basis für die Benutzung eines EUS. Doch auch hier gibt es verschiedene Ansätze wie die Entscheidung unterstützt werden kann – vom reinen abfragen von Informationen und deren Ausgabe bis hin zur komplexen Informationsbeschaffung, Bewertung und Präsentation des Datenbestands. Einige Systeme können daraus bereits schließen, ob bei den eingegebenen Daten die Wahl dafür oder dagegen sprechen sollte.

Das hier entwickelte Werkzeug bietet eine Bestandsaufnahme und organisiert diese Informationen in einem Modell. Durch den Arbeitsfluss des MIGRATE! Vorgehensmodells ist der Ablauf bereits vorgegeben. Das Werkzeug bietet die passenden Eingabefelder und stellt die Informationen übersichtlich dar. Mit Funktionen wie Ausschließungen, Gewichtungen und detaillierten Beschreibungen kann der Anwender den Prozessablauf absolvieren. Wenn die erforderlichen Daten eingegeben sind kann der Anwender selbstständig eine Entscheidung über das weitere Vorgehen bei dem Migrationsvorhaben treffen.

6.2 Ausblick

Auch zukünftig werden Entscheidungen rund um die Cloud für Unternehmen interessant sein. Durch die Modellierung der IT-Infrastruktur kann ein Überblick der Komponenten

die Grundlage von Migrationsentscheidungen sein. Idealerweise kann eine **Anbindung** der Modellierung an das Entscheidungsunterstützungssystem für eine bessere Integration sorgen. Damit wäre die Informationserfassung deutlich einfacher, fehlerfreier und vollständiger.

Ein wichtiger Punkt ist auch die **Bewertung** der Komponenten. Diese könnte ebenso durch die Kopplung mit der Modellierung vereinfacht werden oder automatisiert erfolgen. Hier gibt es noch Spielraum für eine flexible Priorisierung oder Gewichtung der Kriterien. Andere EUS verwenden mathematische Methoden um die Verarbeitung der eingetragenen Bewertungen zu übernehmen.

Nach der Evaluation folgt die Entscheidung – diese liegt in der entwickelten Anwendung alleine beim Benutzer. Hier können Frameworks wie das (MC²)² (siehe Kapitel 3.3.1) dabei helfen, automatisch uninteressante Komponenten auszuschließen und dem Benutzer eine **Entscheidungsempfehlung** vorzuschlagen. Diese Methoden sind jedoch sehr komplex und benötigen eine intensive Auseinandersetzung mit dem Thema. Die angewendeten Methoden sollten eine **Mehrdimensionale Problemstellung** beachten [ASL₁₃], denn die Auswahl beruht auf unterschiedlichen Faktoren wie Anwendungsverteilung, Migrations- und laufende Kosten, Dienstgüte oder Sicherheit der Daten.

Mit Hilfe von Simulationen und **Migrationswerkzeugen** könnte der Anwender seine Entscheidung zeitnah erproben. Wird beispielsweise entschieden, dass eine Komponente der Datenbankschicht in die Cloud migriert werden kann, könnte ein entsprechendes Werkzeug dieses Vorhaben umsetzen. Es gibt bereits verwandte Arbeiten an dem betreuenden Institut, welche diese Thematik untersuchen.

Die Migrationsszenarien decken vielseitige **Einsatzgebiete** ab. In einer Arbeit von Strauch [SABL₁₃] werden die Anwendungsdaten mit dem Einsatz von Cloud Data Patterns migriert. Dabei könnte man im Entscheidungsunterstützungstool beachten ob das Szenario beispielsweise eine Hochverfügbarkeit, geographische Replikation, Synchronisation oder eine Auslagerung der Datenbankschicht als Ziel haben. Je nach Anwendungsfall gibt es konsequenterweise unterschiedliche Prioritäten und Abhängigkeiten der Migrationsaspekte. Eine Hilfestellung durch eine Webanwendung könnte bei der Informationssammlung, Auswertung und Auswahl der Zielinfrastrukturen hilfreich sein.

Literaturverzeichnis

- [ABLS13] V. Andrikopoulos, T. Binz, F. Leymann, S. Strauch. How to adapt applications for the Cloud environment. *Computing*, 95:493–535, 2013.
- [AGW05] E. Ackermann, R. Gimnich, A. Winter. Ein Referenz-Prozess der Software-Migration. *Softwaretechnik-Trends*, 25(4):20–22, 2005.
- [ASL13] V. Andrikopoulos, S. Strauch, F. Leymann. Decision Support for Application Migration to the Cloud: Challenges and Vision. In *Proceedings of the 3rd International Conference on Cloud Computing and Service Science, CLOSER 2013, 8-10 May 2013, Aachen, Germany*, S. 149–155. SciTePress, 2013.
- [BHJ⁺12] D. Betts, A. Homer, A. Jezierski, M. Narumoto, H. Zhang. *Moving Applications to the Cloud on the Microsoft Azure Platform*. Microsoft, 2012.
- [CSM⁺05] T. Constantinou, Y. Sazeides, P. Michaud, D. Fetis, A. Sez nec. Performance implications of single thread migration on a chip multi-core. *ACM SIGARCH Computer Architecture News*, 33(4):80–91, 2005.
- [DIN94] DIN. 8402.(1992) DIN 66272, Norm (1994-10-00), 1994.
- [Gar10] D. Garcia. CS10 The Beauty and Joy of Computing. In *CS10 The Beauty and Joy of Computing*. 2010.
- [GS12] F. Gao, S. Schneider. Cloud-Frameworks: An Information Systems Perspective. In *Proceedings of ConLife Academic Conference*. 2012.
- [GVB11] S. K. Garg, S. Versteeg, R. Buyya. Smicloud: A framework for comparing and ranking cloud services. In *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, S. 210–218. IEEE, 2011.
- [Hö11] T. Höllwarth. *Cloud Migration*. mitp, 2011.
- [HL95] F. Halsall, D. Links. Computer Networks and Open Systems. *Addison-Wesley Publishers*, S. 112–125, 1995.
- [HP13] Hewlett-Packard. Enterprise Application Migration to the Cloud. Internet, 2013.

- [ISO01] ISO. IEC 9126-1: Software Engineering-Product Quality-Part 1: Quality Model, 2001.
- [Jac13] M. Jacob. Entscheidung. In *Management und Informationstechnik*, S. 15–35. Springer Fachmedien Wiesbaden, 2013.
- [JHD⁺13] R. Johnson, J. Hoeller, K. Donald, C. Sampaleanu, R. Harrop, T. Risberg. *Spring Framework Reference Documentation*, 3.2.2.release Auflage, 2013.
- [Joh03] R. Johnson. *Expert one-on-one J2EE design and development*. Programmer to programmer. Wrox, 2003.
- [JPF05] H. Jungermann, H. Pfister, K. Fischer. *Die Psychologie der Entscheidung: eine Einführung*. Spektrum Akademischer Verlag GmbH, 2005.
- [Kee93] R. L. Keeney. *Decisions with multiple objectives: preferences and value trade-offs*. Cambridge University Press, 1993.
- [KHGS10] A. Khajeh-Hosseini, D. Greenwood, I. Sommerville. Cloud migration: A case study of migrating an enterprise it system to iaas. In *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, S. 450–457. IEEE, 2010.
- [KNT09] M. Klems, J. Nimis, S. Tai. Do Clouds Compute? A Framework for Estimating the Value of Cloud Computing. In C. Weinhardt, S. Luckner, J. Stößer, Herausgeber, *Designing E-Business Systems. Markets, Services, and Networks*, Band 22 von *Lecture Notes in Business Information Processing*, S. 110–123. Springer Berlin Heidelberg, 2009.
- [Kun11] V. Kundra. *Federal cloud computing strategy*. The White House, 2011.
- [KV10] R. L. Krutz, R. D. Vines. *Cloud security: A comprehensive guide to secure cloud computing*. Wiley, 2010.
- [KW10] B. Kosch, H. Wagner. Alles im grünen Bereich – Mit Green IT zu Energieeffizienz und Nachhaltigkeit. In D. Spath, W. Bauer, S. Rief, Herausgeber, *Green Office*, S. 205–212. Gabler, 2010.
- [Ley09] F. Leymann. Cloud Computing: The Next Revolution in IT. In D. Fritsch, Herausgeber, *Photogrammetric Week 09*, S. 3–12. Wichmann Verlag, 2009.
- [LFM⁺11] F. Leymann, C. Fehling, R. Mietzner, A. Nowak, S. Dustdar. Moving Applications to the Cloud: An Approach based on Application Model Enrichment. *International Journal of Cooperative Information Systems*, 20(3):307–356, 2011.
- [Lic63] J. C. Licklider. Memorandum for Members and affiliates of the Intergalactic Computer Network. *Apr*, 23:350–1, 1963.
- [LKN⁺09] A. Lenk, M. Klems, J. Nimis, S. Tai, T. Sandholm. What’s inside the Cloud? An architectural map of the Cloud landscape. In *Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD ’09*, S. 23–31. IEEE Computer Society, Washington and DC and USA, 2009.

- [Lou10] P. Louridas. Up in the air: Moving your applications to the cloud. *Software, IEEE*, 27(4):6–11, 2010.
- [M⁺65] G. E. Moore, et al. Cramming more components onto integrated circuits, 1965.
- [MCP09] A. Molla, V. Cooper, S. Pittayachawan. IT and eco-sustainability: Developing and validating a green IT readiness model. In *International Conference of Information Systems*, S. 1–18. AIS Electronic Library, 2009.
- [Mel10] I. Melzer. Service-orientierte Architektur. In *Service-orientierte Architekturen mit Web Services*, S. 9–31. Spektrum Akademischer Verlag, 2010.
- [MG11] P. Mell, T. Grance. The NIST definition of cloud computing (draft). *NIST special publication*, 800:145, 2011.
- [Mic09] Microsoft. Data Layer Guidelines. *Microsoft Application Architecture Guide, 2nd Edition*, Chapter 8:1–13, 2009.
- [MKL09] T. Mather, S. Kumaraswamy, S. Latif. *Cloud security and privacy: an enterprise perspective on risks and compliance*. O'Reilly Media, 2009.
- [Muro8] S. Murugesan. Harnessing Green IT: Principles and Practices. *IT Professional*, 10(1):24–33, 2008.
- [NM90] J. Nielsen, R. Molich. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, S. 249–256. ACM, 1990.
- [PM10] F. Pavel, A. Mattes. Cloud-Computing: Großes Wachstumspotenzial. *DIW Wochenbericht* 48, 48:10–16, 2010.
- [Pur10] G. Purdy. ISO 31000: 2009—setting a new standard for risk management. *Risk Analysis*, 30(6):881–886, 2010.
- [RDMB12] A. Robrecht, R. Dörbecker, C. Möller, T. Böhmman. Entscheidungsunterstützung für Cloud Computing–ITIL Service Lifecycle. *HMD*, 288:24–32, 2012.
- [RG05] A. W. Rainer Gimnich, Uwe Kaiser. Reengineering Prozesse (RePro 2005) Software-Migration. In *Workshop der Fachgruppe Software Reengineering*. 2005.
- [Saa94] T. L. Saaty. *Fundamentals of decision making and priority theory: with the analytic hierarchy process*. Rws Publications, 1994.
- [SABL13] S. Strauch, V. Andrikopoulos, T. Bachmann, F. Leymann. Migrating Application Data to the Cloud Using Cloud Data Patterns. In *Proceedings of the 3rd International Conference on Cloud Computing and Service Science, CLOSER 2013, 8-10 May 2013, Aachen, Germany*, S. 36–46. SciTePress, 2013.
- [SB03] B. Shneiderman, S. Ben. *Designing The User Interface: Strategies for Effective Human-Computer Interaction, 4/e (New Edition)*. Pearson Education India, 2003.
- [SN12] G. Stone, P. Noel. *Cloud Risk Decision Framework*, 2012.

- [SSW10] K. Stanoevska-Slabeva, T. Wozniak. Cloud Basics – An Introduction to Cloud Computing. In K. Stanoevska-Slabeva, T. Wozniak, S. Ristol, Herausgeber, *Grid and Cloud Computing*, S. 47–61. Springer Berlin Heidelberg, 2010.
- [SWH10] H. M. Sneed, E. Wolf, H. Heilmann. Softwaremigration in der Praxis. *dpunkt.verlag, Heidelberg*, 2:287, 2010.
- [VRMCL08] L. M. Vaquero, L. Rodero-Merino, J. Caceres, M. Lindner. A break in the clouds: towards a cloud definition. *SIGCOMM Comput. Commun. Rev.*, 39(1):50–55, 2008.
- [Web10] D. M. Weber. *Cloud Computing - Evolution in der Technik, Revolution im Business*. BITKOM-Leitfaden, 2010.
- [YH95] K. P. Yoon, C.-L. Hwang. *Multiple attribute decision making: an introduction*. 104. Sage, 1995.
- [ZC82] M. Zeleny, J. L. Cochrane. *Multiple criteria decision making*, Band 25. McGraw-Hill New York, 1982.
- [Zim80] H. Zimmermann. OSI reference model–The ISO model of architecture for open systems interconnection. *Communications, IEEE Transactions on*, 28(4):425–432, 1980.

Alle URLs wurden zuletzt am 23. Juli 2013 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben.

Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet.

Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens.

Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht.

Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Unterschrift:

Stuttgart, 24. Juli 2013

Declaration

I hereby declare that the work presented in this thesis is entirely my own.

I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations.

Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before.

The electronic copy is consistent with all submitted copies.

Signature:

Stuttgart, July 24th, 2013