HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

# DIPLOMARBEIT

Herr Ing.
**Philipp Nagele**

## Konzept und Realisierung einer IT-Servicemanagement Lösung für geografisch verteilte Systeme

Mittweida, 2016

Fakultät: **Angewandte Computer- und Biowissenschaften**

# DIPLOMARBEIT

# Konzept und Realisierung einer IT-Servicemanagement Lösung für geografisch verteilte Systeme

Autor:
**Herr Ing. Philipp Nagele**

Studiengang:
**Technische Informatik**

Seminargruppe:
**KT11wIA-F**

Erstprüfer:
**Prof. Dr.-Ing. Uwe Schneider**

Zweitprüfer:
**Prof. Dr. habil. Thomas Haenselmann**

Einreichung:
**Mittweida, 14.01.2016**

Verteidigung:
**Mittweida, 2016**

## Bibliografische Beschreibung:

Nagele, Philipp: Konzept und Realisierung einer IT-Servicemanagement Lösung für geografisch verteilte Systeme, 70 Seiten, 30 Abbildungen, Mittweida, Hochschule Mittweida, Fakultät Angewandte Computer- und Biowissenschaften,

Diplomarbeit, 2016

Herr Ing. Philipp Nagele

## Referat:

Im Rahmen dieser Diplomarbeit wird das IT-Service Management und im Speziellen der Bereich Servicebetrieb, für den Einsatz in Unternehmen mit großen, geografisch verteilten IT-Landschaften, nach aktuellen Standards durchleuchtet und bewertet. Auf der Basis der gewonnenen Erkenntnisse wird ein Konzept für eine Lösung erstellt, welche den Servicebetrieb für die angesprochenen Unternehmen, hinsichtlich Bearbeitungszeit von Störungsbehebungen und Einsatz von Arbeitskraft, optimieren soll. Das erarbeitete Konzept wird anschließend als Prototypensystem implementiert und bewertet.

# I. Inhaltsverzeichnis

# II. Abbildungsverzeichnis

# III. Tabellenverzeichnis

# IV. Abkürzungsverzeichnis

ASP ..... *Active Server Pages, Seite 62*
BPM ..... *Business Process Management, Seite 6*
CCTA ..... *Central Computer and Telecommunicatios Agency, Seite 10*
COBIT ..... Control Objectives for Information and related Technology, Seite 6
eTOM ..... *Enhanced Telecom Operations Map, Seite 8*
GUI ..... *Graphical User Interface, Seite 25*
HTTP ..... *Hyper Text Transfer Protocol, Seite 18*
HTTPS ..... *Hyper Text Transfer Protocol Security, Seite 31*
IPsec ..... *Internet Protocol Security, Seite 19*
ITIL ..... *Information Technology Infrastructure Library, Seite 6*
ITSM ..... *Information Technology Service Management, Seite 1*
JRE ..... *Java Runtime Environment, Seite 34*
LAN ..... *Local Area Network, Seite 17*
MAC ..... *Media Access Control, Seite 52*
OGC ..... *Office of Government Commerce, Seite 10*
OOTB ..... *Out Of The Box, Seite 3*
RDBMS ..... *Relationales Datenbank Management System, Seite 36*
RMI ..... *Remote Method Invocation, Seite 18*
SOA ..... *Service Orientierte Architektur, Seite 37*
SQL ..... *Structured Query Language, Seite 33*
SSL ..... *Secure Socket Layer, Seite 31*
TLS ..... *Transport Layer Security, Seite 37*
TMF ..... *Telemanagement Forum, Seite 8*
TQM ..... *Total Quality Management, Seite 6*
UDDI ..... *Universal Description Discovery and Integration, Seite 41*
VB ..... *Visual Basic, Seite 66*
VPN ..... *Virtual Private Network, Seite 19*
WAN ..... *Wide Area Network, Seite 17*
WMI ..... Windows Management Instrumentation, Seite 18
WSDL ..... *WebService Description Language, Seite 41*
XML ..... *Extensible Markup Language, Seite 39*

# 1 Einleitung

## 1.1 Motivation

Die Motivation, welche zur Auswahl und letztendlich zur Bearbeitung dieses Themengebiets führte, entstammt der beruflichen Tätigkeit des Autors. Durch mehrjährige Tätigkeit im Umfeld IT Service Management (ITSM), stieg das Interesse an einer möglichen individuellen ITSM- Lösung, welche den Prozess der Störungsbehandlung in komplexen und geografisch verteilten Systemen optimiert.

## 1.2 Problemstellung

In modernen Unternehmen, welche national bzw. international über mehrere Standorte hinweg aufgestellt sind, entstehen für die IT-Service Management Verantwortlichen, ständig neue Herausforderungen, durch steigende Anforderungen an die Verfügbarkeit der IT-Systeme und Applikationen. Die schnelle und unkomplizierte Behebung von lokal begrenzten sowie auch großflächigen Störungen, gewinnt dadurch mehr und mehr an Wichtigkeit. Zur Sicherstellung des IT-Betriebs wird ständig versucht, die Prozesse und Vorgänge, welche diesen ermöglichen bzw. zur Störungsbehebung eingesetzt werden, zu optimieren.

Oft reichen die Mittel und Wege, welche zur Störungsbehebung durch den IT-Support eingesetzt werden können nicht aus, um spezielle Anforderungen an die Bearbeitungszeit von Störungen zu erfüllen. Ein weiteres Problem entsteht durch die erforderliche Inter-Standort Kommunikation. Längst nicht jede Technologie eignet sich für sichere und reibungslose Kommunikation und das Ausführen von Steuerungsvorgängen über Intranet- und Firewallgrenzen hinweg. Zur Sicherstellung eines geradlinigen und homogenen Störungsbehebungs-Workflows, muss also eine Technologie gewählt werden, welche den unkomplizierten und auch globalen Austausch von benötigten Informationen ermöglicht.

Des Weiteren ist es heutzutage, durch die zunehmende Komplexität von großen IT-Landschaften, im Bezug auf die eingesetzte Software und Hardware, von Vorteil, den Zugriff auf ein IT Service Management System plattformunabhängig zu gestalten, um einem möglichst breiten Spektrum von Systemen den Zugriff darauf zu ermöglichen.

Speziell in mittelgroßen bis großen Unternehmen, welche über eine entsprechend große Anzahl von Applikationen und IT-Benutzern verfügen, kommen vermehrt Thin-Clients[1] in Kombination mit Terminal Servern zum Einsatz. Daraus resultiert eine Steigerung der Komplexität, der zu betreuenden IT-Umgebungen, weil auftretende Problemfälle schwerer zu lokalisieren sind, als in Umgebungen mit klassischen Fat-Clients[2] und lokal installierten Applikationen.

Eine nicht zu unterschätzende Thematik im IT-Support-Bereich, ist die Übermittlung bzw. die Einholung, der tatsächlich zur Behebung einer Störung benötigten Informationen. Es kommt nicht selten zu Verzögerungen, wenn ein IT-Benutzer schlichtweg nicht in der Lage ist, dem IT-Support alle benötigten Informationen schnell und korrekt zukommen zu lassen. Dies muss dann meist durch erneute Kommunikation, oder aber auch durch Aufbau einer Remote Session nachgeholt werden, was wiederum Mehraufwand und letztendlich Kosten verursacht.

## 1.3 Zielsetzung

Eines der Ziele dieser Diplomarbeit ist, das IT-Service Management und im Speziellen den Teilbereich Servicebetrieb, für den Einsatz in Unternehmen, mit großen geografisch verteilten IT-Landschaften, nach aktuellen Standards zu betrachten und zu erörtern. Konkret bedeutet das, dass zuerst die benötigten Grundlagen, welche zum Verständnis und Know-how Aufbau, für den Bereich IT-Service Management von Nöten sind, durchgearbeitet und erörtert werden müssen, um das Thema zielführend bearbeiten zu können. Es soll natürlich, wie schon beschrieben, ganz speziell auf die Anforderungen, welche sich aus der geografischen Verteilung der Systeme und der dadurch gesteigerten Komplexität selbiger ergeben, eingegangen werden. Des Weiteren soll geprüft werden, wie und wodurch der Ablauf von Störungsbehebungen, in IT-Umgebungen mit einer komplexen Struktur optimiert werden kann. Als Beispiel soll eine geografisch verteilte IT-Umgebung dienen, welche sowohl klassische Fat-Clients mit lokal installierten Applikation sowie auch Thin-Clients in Kombination mit Terminal Servern, als Arbeitsstationen beinhaltet.

In weiterer Folge werden die ITSM Produkte zweier kommerzieller Anbieter betrachtet. Hierbei soll der Focus auf dem Funktionsumfang und der Skalierbarkeit der Produkte liegen. Es soll geprüft werden, inwieweit diese Produkte die speziellen Anforderungen von komplexen, geografisch verteilten IT-Landschaften abdecken können, beziehungsweise, wie gut diese Produkte sich dazu eigenen, einen Teil zu einer Gesamtlösung beisteuern zu können, welche spezielle Anforderungen an Bearbeitungszeit von Störungen erfüllen muss. Wesentliche zu betrachtende Parameter sind hierbei die Vernetzbarkeit mit anderen

[1] Ein Thin-Client ist ein Computer der stark auf die Hilfe anderer Computer oder seines Servers angewiesen ist, um seine eigentlichen Computeraufgaben zu erfüllen. (siehe https://de.wikipedia.org/wiki/Thin-Client)
[2] Fat-Client bezeichnet im Allgemeinen vollwertig ausgestattete, leistungsfähige Desktop-Computer. (siehe https://de.wikipedia.org/wiki/Fat_Client)

Produkten und welche Funktionen und Schnittstellen Out Of The Box (OOTB) schon vorhanden sind.
Auch die Art und Weise, wie ein Mitarbeiter des IT-Supports an die zur Problemlösung benötigten Informationen kommen kann und wie der entsprechende Workflow, den die Produkte dazu vorsehen aussieht, soll betrachtet werden. Der Informationsaustausch zwischen IT-Support und IT-Benutzern, soll bezüglich seiner Effizienz geprüft und dadurch mögliche Unzulänglichkeiten aufgezeigt werden. Die dadurch gewonnenen Erkenntnisse, sollen zur Erarbeitung eines Konzepts dienen, welches den ITSM-Servicebetrieb für die angesprochenen Unternehmen, hinsichtlich Bearbeitungszeit von Störungsbehebungen und Einsatz von Arbeitskraft optimieren könnte. Anschließend soll auf der Basis des erarbeiteten Konzepts eine ITSM-Lösung als Prototypensystem implementiert werden. Anhand dieses Prototypensystems kann dann die Umsetzbarkeit des erarbeiteten Konzepts geprüft werden.

## 1.4 Nicht-Ziele dieser Diplomarbeit

Nachdem in Punkt 1.2, die Zielsetzung für diese Diplomarbeit formuliert wurde, werden nun auch noch die Nicht-Ziele dieser Diplomarbeit definiert. Durch die Definition der Nicht-Ziele, kann eine klare Abgrenzung zwischen dem möglichen und dem tatsächlichen Ausmaß der zu bearbeitenden Teilbereiche gezogen werden. Im Hinblick auf die zu erwartenden Ergebnisse dieser Arbeit, im Bezug auf die Analyse bestehender Lösungen und Konzeptionierung einer individuellen Lösung, soll dadurch von vornherein Klarheit geschaffen werden. Folgende Nicht-Ziele wurden definiert:

- **Gesamtbetrachtung ITSM:** Bei der Betrachtung und Analyse der beiden kommerziellen ITSM Lösungen, werden nur die für den Servicebetrieb relevanten Komponenten betrachtet und bewertet. Des Weiteren werden zur genauen Betrachtung nur die Teilbereiche Incident- und Problem Management herangezogen.

- **Design, Performance und Stabilität:** Im Zuge der Prototypenerstellung, wird nur auf die funktionalen Anforderungen, nicht aber auf die Nicht-funktionalen Anforderungen eingegangen. Soll heißen, dass Performance, Usability und Design-Aspekte der einzelnen Systeme und Softwarekomponenten, nicht bei der Prototypenerstellung berücksichtigt werden.

- **Vollständigkeit**: Die Testimplementierung wird nur als Prototypensystem durchgeführt, um die Funktion und Machbarkeit des erstellten Konzeptes zu demonstrieren. Diese Implementierung stellt also keinen Anspruch auf Vollständigkeit. Auf die Verteilung und Bereitstellung, der zur Implementierung benötigten Softwarekomponenten, wird in dieser Arbeit nicht eingegangen.

- **Non-Microsoft Devices:** Bei der geplanten Implementierung als Prototypensystem wird davon ausgegangen, dass es sich bei den durch die spezielle ITSM-Lösung zu

betreuenden Endgeräten, ausschließlich um Geräte mit Microsoft Windows als Betriebssystem handelt. Die Ausnahme bilden nur die Thin-Clients, welche jedoch im Endeffekt via Remote Session auf Windows Terminal Server zugreifen.

## 1.5 Methodik

Grundsätzlich setzt diese Diplomarbeit auf den vorhandenen Best Practices und De-Facto-Standards, welche für den Bereich IT Service Management schon vorhanden sind, auf. Es wird untersucht, inwieweit bestehende Softwarelösungen kommerzieller Anbieter, welche ebenfalls auf diesen Standards fußen, dazu geeignet sind, spezielle Anforderungen an eine ITSM-Lösung zu erfüllen. Diese speziellen Anforderungen werden unter anderem in einem Unternehmen, welches IT-Dienstleistungen für Kunden im Finanzsektor erbringt, erhoben. Zur Prüfung ob und in welcher Form Verbesserungsbedarf besteht, werden reale Problemfälle hinsichtlich Bearbeitungsdauer und Problemlösungseffizienz analysiert. Sämtliche aus der Analyse gewonnen Erkenntnisse, fließen anschließend in die Konzepterstellung und Implementierung der angestrebten Lösung ein.

## 1.6 Verwendete Softwareressourcen

Es folgt mit *Tabelle 1-1* eine Auflistung der zur Erstellung dieser Diplomarbeit eingesetzten Softwareressourcen.

| **Aufgabe** | **Software** |
|---|---|
| Betriebssystem | Microsoft Windows 7 x64 Professional |
| Textverarbeitung | Microsoft Word 2007 Professional |
| Grafikerstellung | Microsoft Visio 2007 Professional |
| Webserver | Microsoft IIS 10.0 Express |
| Framework | Microsoft .NET Framework 4.5 |
| Datenbank | Microsoft SQL Server Compact (SQL CE) |
| Softwareentwicklung | Microsoft Visual Studio 2013 Professional |

Tabelle 1-1: Auflistung verwendeter Softwareressourcen

# 2 Grundlagen des IT-Service Management

## 2.1 Einleitung

In diesem Kapitel geht es vorrangig darum, die Grundlagen und die Bedeutung von ITSM zu erläutern. Hierzu wird erklärt, was unter dem Begriff ITSM zu verstehen ist und welche Normen, Frameworks und Standards es diesbezüglich gibt. Nach einem Überblick über die diversen Möglichkeiten, ITSM zu realisieren, wird auf eine Variante detaillierter eingegangen. In weiterer Folge werden die für diese Diplomarbeit relevanten Teilbereiche des ITSM genauer betrachtet, um eine Wissensbasis für die weitere Bearbeitung des Themas zu schaffen.

## 2.2 Definition von ITSM

Es gibt keine genormte oder standardisierte Definition für das, was sich hinter der Abkürzung ITSM, welche für **I**nformation **T**echnology **S**ervice **M**anagement steht, verbirgt.

Die Information Technology Infrastructure Library (ITIL) beschreibt ITSM, als eine Zusammensetzung aus den beiden Bereichen Service Support und Service Delivery. Innerhalb dieser Bereiche werden die Service-Prozesse behandelt, die den direkten Nutzersupport und die Betriebsunterstützung betreffen und das tägliche Geschäft abhandeln. Hier fällt die Bearbeitung der vom Nutzer gemeldeten Störungen hinein sowie das Warten von Applikationssoftware und Betriebssystemkomponenten, möglichst ohne fühlbare momentane und zukünftige Unterbrechung des normalen Arbeitsablaufes. [TKö07]

In einschlägiger Literatur, welche die ISO-Standardisierung von ITSM (ISO/IEC20000) beschreibt, ist folgende Definition zu finden: ITSM bedeutet die Integration von Prozessen, Technologien und Methoden, die unter Berücksichtigung des Kunden notwendig sind, um die bestmögliche Unterstützung der Geschäftsprozesse im Unternehmen durch die IT zu erreichen. ITSM ist die Basis für eine gut funktionierende IT-Infrastruktur mit allen benötigten IT Services. [Sch10]

Die wohl treffendste und ausführlichste Beschreibung, was denn nun ITSM bedeutet, findet sich in der Definition laut [Kla15] wieder. ITSM ist die Integration von Prozessen und Technologien unter Berücksichtigung des Benutzers. Es ist die Basis für eine gut funktionierende IT-Infrastruktur mit allen benötigten IT-Services. Eine solche IT-Infrastruktur basiert auf erprobten und bewährten Prozessabläufen und orientiert sich an den

Erfordernissen der Benutzer. Es muss sich ständig an deren Bedürfnisse anpassen, die, wie in einem Regelkreis, in das Prozessmodell einfließen und fortlaufend zu Verbesserungen führen. Die Information Technology Infrastructure Library (ITIL) ist ein solcher De-Facto-Standard, der mit einem entsprechenden Prozessmodell arbeitet.

IT-Services sind Aktiva, mit denen der maximale Nutzen für ein Unternehmen erzielt werden soll, ohne die Kostenkontrolle aus den Augen zu verlieren. Dargestellt wird es in einem Prozessmodell, in das alle Prozesse und Abläufe einfließen.
Ein ITSM-Prozessmodell liefert umfassende, lückenlose Beschreibungen von den IT-Prozessen des Unternehmens. Es definiert den kritischen Pfad im Prozessablauf und legt die Verantwortungsbereiche für die einzelnen Service-Elemente fest. Dazu gehört das Business Process Management (BPM) mit dem Geschäftsprozesse beschrieben und optimiert werden können. Darüber hinaus gibt ITSM der IT-Organisation standardisierte Prozesse vor und unterstützt mit dem Total Quality Management (TQM) die Verbesserung von Qualitätsstandards. [Kla15]

# 2.3 Normen, Frameworks und Standards

## 2.3.1 COBIT

COBIT ist eines der sogenannten kostenlosen und frei zugänglichen Frameworks, den sogenannten Public Domain Frameworks. COBIT als Begriff, steht für **C**ontrol **O**bjectives for **I**nformation and related **T**echnology. COBIT ist im Prinzip eine Dokumentensammlung in welche etliche Standards aus mehreren Quellen eingeflossen sind und kann somit als lebender Standard bezeichnet werden. Das Ziel dieses Standards besteht darin, ein Model zur Steuerung und Kontrolle der gesamten IT eines Unternehmens bereitzustellen.
Die Entwicklung von COBIT begann im Jahr 1994. Eine erste Version erschien 1996 und weitere Versionen 1998, 2000 und 2006. Die aktuelle Version ist derzeit V5.0 (Apr 12). Seit der Version V5.0 wird COBIT nur noch als Akronym verwendet.

Um die Hierarchie von COBIT zu erläutern, eignet sich am Besten eine grafische Übersicht wie sie in *Abbildung 2-1* dargestellt ist. Die allgemeine Überlegung ist die, dass sich die Gesamtstruktur in 3 Ebenen gliedert. Auf der niedrigsten Ebene befinden sich die Aktivitäten, die für die Erzielung der gewünschten Resultate benötigt werden. Auf der mittleren Ebene können diese Aktivitäten zu logischen Einheiten gruppiert werden, welche nun wiederrum eine Überwachung durch spezifische Kontrollen zulassen. Diese logischen Einheiten werden Prozesse genannt. Auf der obersten Ebene werden die formierten Prozesse zu Domänen konsolidiert, welche nun idealerweise den Organisationsanforderungen der diversen IT-Bereiche entsprechen. [Gol06]

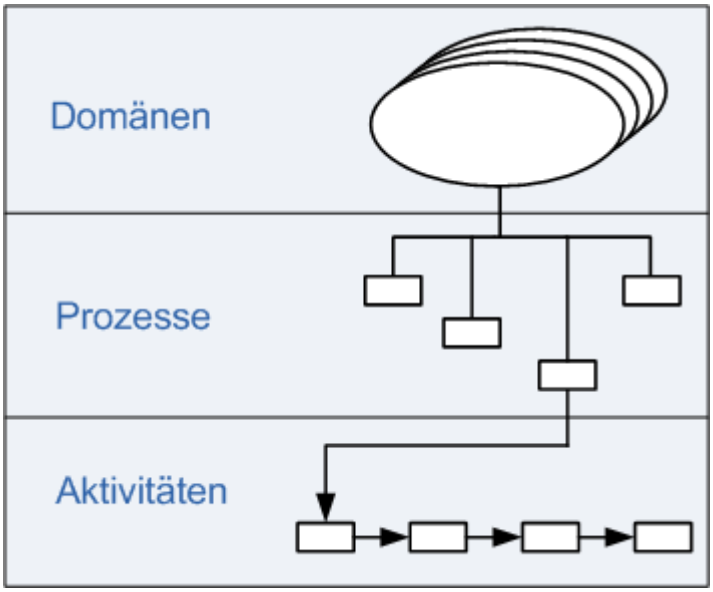


Abbildung 2-1: Hierarchie von COBIT ( [Gol06] , Seite 26)

COBIT verteilt alle seine Prozesse auf die folgenden 4 Domänen, welche zueinander in Abhängigkeit stehen und sich untereinander durch Wechselwirkungen beeinflussen. Dadurch entsteht sozusagen ein Regelkreis, der die ständige Optimierung der Vorgänge erlaubt.

- Planung & Organisation
- Akquisition & Implementierung
- Delivery & Support
- Monitoring & Evaluierung

Die Domäne **Delivery & Support** beinhaltet alle Prozesse, die zur Erbringung der IT-Services vonnöten sind. Die beiden Prozesse dieser Domäne, die für diese Diplomarbeit von Relevanz sind, sollen hier noch kurz beschrieben werden.

- **Anwenderunterstützung:** Der Prozess stellt eine Schnittstelle zwischen Anwender und Support zur Verfügung, über welche jegliche IT-Probleme, denen sich ein Anwender gegenübersieht, angenommen und bearbeitet werden können. Dieser Prozess betrachtet somit den Servicedesk inklusive Incident Management.

- **Problemmanagement:** Der Prozess Problemmanagement ist logisch betrachtet der Nachfolgeprozess des Anwendungsunterstützungsprozesses. Das Problem-management stellt sicher, dass eingemeldete Problemfälle professionell und zeitnah bearbeitet werden.

Zieht man nun ein kurzes Resümee über COBIT bzw. die Einsatzgebiete und Einsatzmöglichkeiten dieses Frameworks, bleibt zu sagen, dass es sich um einen prozessgetriebenen Standard handelt, welcher sich gut für die Anwendung auf dynamische IT-Umgebungen eignet.

### 2.3.2 eTOM

Bei eTOM handelt es sich um ein Prozessmodell welches unter anderem zu den frei zugänglichen Frameworks zu zählen ist. Die Abkürzung eTOM steht für **E**nhanced **T**elecom **O**perations **M**ap und wurde und wird vom Telemanagement Forum (TMF) entwickelt und gepflegt. Auf Grund des hohen Abstraktionsgrades und der universellen Einsetzbarkeit von eTom lässt sich dieses ursprünglich für die Telekommunikationsbranche entwickelte Modell, problemlos auf unterschiedliche Branchen anwenden. [Krc05] eTOM kann als Full Enterprise Framework bezeichnet werden, das nicht nur eine isolierte Prozessbetrachtung zulässt, sondern auch die Zusammenhänge zwischen den Vorgängen innerhalb des eigenen Unternehmens, wie auch die Verbindungen zu anderen Unternehmen verdeutlicht. Dieses Modell ist in 4 unterschiedliche Level eingeteilt, wobei Level 0 die Grobansicht widerspiegelt und der Detailgrad der Prozessfelder bis hin zu Level 3 stetig zunimmt. Level 3 beinhaltet dann schon konkrete Empfehlungen für die Umsetzung von bestimmten IT-Prozessen. Das eTOM-Modell, stützt sich auf 3 Hauptgruppen von Prozessen:

- **Operations:** Dieser Bereich umfasst die betriebswirtschaftlichen Funktionen, die direkt mit der Leistungserbringung verbunden sind.

- **Strategy, Infractructure and Product:** Die Domäne „Strategy, Infractructure and Product“ (SIP) beinhaltet die Vorgänge, die mit der Leistungskonzeption und der Überwachung verbunden sind.

- **Enterprise Management:** Das „Enterprise Management“ beschreibt die allgemeinen Verwaltungstätigkeiten einer IT-Organisation.

Ein Manko von eTOM, ist die fehlende Detailbeschreibung der einzelnen Arbeitsschritte. Dieser Umstand wird von den Verfassern jedoch damit gerechtfertigt, dass eTOM durch seinen hohen Abstraktionsgrad eine ganzheitliche Sicht auf ein IT-Unternehmen liefert und somit als Rahmenwerk für den Einsatz von ITIL verwendet werden kann.

### 2.3.3 ISO/IEC 20000

ISO/IEC 20000 steht für eine Norm, die einen globalen Standard für IT-Service Management Prozesse bereitstellt und eine international gültige Zertifizierung von IT-Unternehmen

ermöglicht. Diese Norm wurde von der ISO[3] am 15.12.2006 offiziell veröffentlicht. ISO/IEC 20000 beschreibt einen Referenzstandard für alle IT-Organisationen, welche IT-Dienstleistungen für interne sowie auch externe Kunden erbringen. Hierzu spezifiziert diese Norm alle Mindestanforderungen und Prozesse, die zur Erbringung einer Leistung in einer bestimmten Qualität, notwendigerweise durch eine Organisation erbracht werden müssen. Des Weiteren wird eine gemeinsame Terminologie gefördert, welche einen wesentlichen Beitrag für die problemlose Kommunikation zwischen Service Provider und Service Receiver leistet. In ISO/IEC 20000 wurde auch der integrierte Prozessansatz aus dem Service Management Framework von ITIL übernommen. (siehe *Abbildung 2-2*) [Sch10]

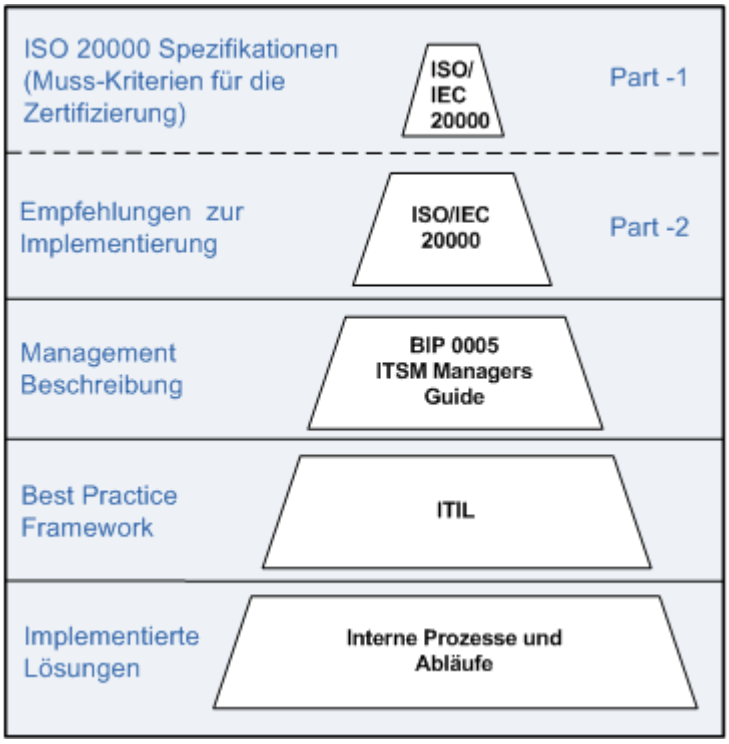


Abbildung 2-2: Struktur von ISO/IEC 20000 ( [Sch10] , Seite 124)

Der Standard ISO/IEC 20000 besteht aus zwei Teilen:

- **"Service Management: Specification":** Dieser Teil enthält die formelle Spezifikation des Standards. Hier sind alle verbindlichen Vorgaben der Prozessgruppen Service Delivery, Control, Release, Resolution und Relationship enthalten, die eine IT-Organisation erfüllen muss, um gemäß der Konformitätsbewertung zertifiziert zu werden.

- **"Service Management: Code of Practice":** Hier ist der "Code of Practice" des IT Service Managements beinhaltet. Er stellt einen umfassenden Leitfaden mit Empfehlungen für die praktische Umsetzung der Kriterien des ersten Teils dar.

[3] Internationale Organisation für Normung (siehe https://de.wikipedia.org/wiki/Internationale_Organisation_f%C3%BCr_Normung)

Wie man durch die Integration von ITIL als Best Practice Framework unschwer erkennen kann, liegt der eigentliche Nutzaspekt von ISO/IEC 20000 bei der Nachweiserbringung für die Umsetzung von Maßnahmen. ISO/IEC 20000 liefert also einen messbaren Qualitätsstandard, der die Einhaltung von Normen und Sicherheitsaspekten im IT-Umfeld gewährleistet.

### 2.3.4 ITIL

Das zu den Public Domain Frameworks gehörende ITIL Rahmenwerk verdient eine genauere Betrachtung, weil viele der am Markt verfügbaren ITSM Lösungen in irgendeiner Weise darauf aufsetzen. Zu diesem Zweck wird ITIL in einem separaten Unterkapitel behandelt.

## 2.4 IT Service Management nach ITIL

### 2.4.1 Die Entwicklung von ITIL

Die Kurzbezeichnung ITIL steht für **I**nformation **T**echnology **I**nfrastructure **L**ibrary und ist ein eingetragenes Warenzeichen der Office of Government Commerce (OGC).Im Jahr 1989 beauftragte die Britische Regierung die heutige OGC, welche jedoch damals noch als Central Computer and Telecommunications Agency (CCTA) bekannt war, mit der Durchführung einer Studie, welche sämtliche aktiven Geschäftsprozesse in der IT ganzheitlich analysieren und beschreiben sollte. Im Rahmen dieser Studie wurden umfangreiche Ist-Aufnahmen und Befragungen in repräsentativen IT-Dienstleistungsunternehmen sowie in Rechenzentren, bei Lieferanten und Kunden durchgeführt. Das Resultat der Studie war eine, einige hundert Bücher umfassende, Bibliothek. Daraus entstand der Name „IT Infrastructure Library“ - kurz ITIL. Im Zuge mehrfacher Konsolidierungsläufe schrumpfte das Gesamtwerk auf ca. 70 Bücher. Heutzutage werden jedoch hauptsächlich nur sechs Bücher, welche den Kern von ITIL bilden, verwendet. Alle Bücher über ITIL, welche von der OCG verfasst wurden, sind nach wie vor öffentlich zugänglich. Es gibt aber mittlerweile etliche kommerzielle Werke diverser Herausgeber. [Olb08] Die bis Dato letzte und aktuellste Version der ITIL-Publikationen, wurde am 29. Juni 2011 unter dem Titel „ITIL 2011 Edition“ veröffentlicht und ist eigentlich nur ein Update der ITIL V3 Version. Diese Ausgabe besteht nunmehr aus einer ausführlichen Einführung sowie fünf Bänden für die Kernbereiche des IT-Service Managements. [And11]

### 2.4.2 Die Definition von ITIL

ITIL ist ein Rahmenwerk, welches die wesentlichen IT-Servicefunktionalitäten beschreibt. Somit beschreibt ITIL welche IT-Services durch Firmen oder öffentliche Institutionen zu

implementieren sind, um ein erfolgreiches IT-Service-Management durchzuführen. ITIL beschreibt aber nicht wie diese Funktionalitäten zu implementieren sind. Wie ITIL im Einzelnen implementiert werden kann, hängt sehr stark von den jeweiligen Gegebenheiten und vom Bedarf ab. [TKö07]

### 2.4.3 ITIL - Flexibilität durch Kommunikation

In der, bis vor Kurzem vor Allem technisch geprägten IT-Welt, findet mehr und mehr ein Wandel in Richtung Dienstleistungen statt. Schlagworte wie Outsourcing[4] und Outtasking[5] sind aus der modernen IT-Welt nicht mehr wegzudenken. Zunehmend werden bisher oft selbst erbrachte Services, an dritte ausgelagert. Zum einen um Kosten zu sparen und zum anderen um immer das Beste, am Markt verfügbare, Know-how nutzen zu können. Durch die Inanspruchnahme einer vom Dienstleister erbrachten Dienstleistung durch den Kunden entsteht also eine Geschäftsbeziehung. Hier steht dann ganz klar die Dienstleistung, also der zu erbringende Service, im Mittelpunkt. Die Qualität hängt hierbei sehr stark von einer guten Kommunikation zwischen den Geschäftspartnern ab. Nur durch einen ständigen Kommunikationsfluss (siehe *Abbildung 2-3*), kann eine stetige Verbesserung der zu erbringenden Dienstleistung erreicht werden.

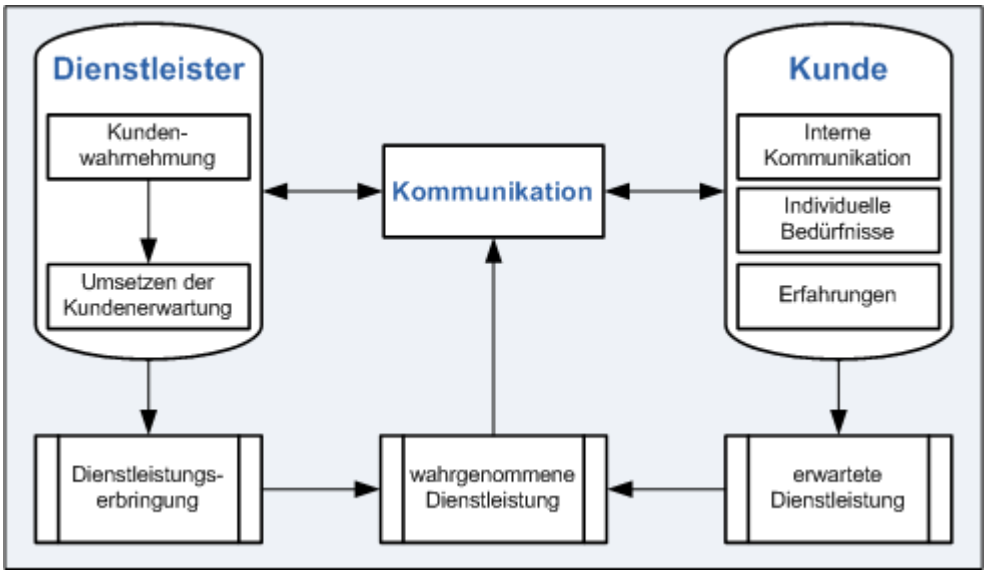


Abbildung 2-3: Kommunikationsfluss bei ITIL ( [Olb08] , Seite 3)

In *Abbildung 2-3* wird der Kommunikationsfluss zwischen den beteiligten Stellen verdeutlicht sowie die Wahrnehmungsebene dargestellt.

[4] Outsourcing bezeichnet in der Ökonomie die Abgabe von Unternehmensaufgaben und Strukturen an externe oder interne Dienstleister. (siehe https://de.wikipedia.org/wiki/Outsourcing)
[5] Beim Outtasking übernehmen externe Dienstleister einzelne Aufgaben. (siehe https://de.wikipedia.org/wiki/Outsourcing)

Aus der ITIL-Perspektive betrachtet, handelt es sich bei dem Modell der erhöhten Kundenorientierung um das tatsächliche IT-Service-Management. Das Ziel sind klar definierte Schnittstellen mit konkreten Ansprechpartnern, Zuständigkeiten und Verantwortlichkeiten, um ein Höchstmaß an Qualität und Kundenzufriedenheit sicherzustellen. Eine Dienstleistung nach ITIL ist daher weit mehr, als die bloße Erbringung einer Leistung. [Olb08] Die Vorteile die sich durch ITIL-konforme Dienstleistung ergeben, lassen sich also wie folgt noch einmal kurz zusammenfassen:

- ITIL hat eine unterstützende Wirkung bei der effektiven und zielorientierten Gestaltung von Prozessen, Rollen und Aufgaben und unterstützt die Entscheidungsträger bei strategischen Maßnahmen. Durch ITIL wird eine erhöhte Flexibilität und Handlungsfreiheit bei dynamischen Vorgängen erreicht.

- Die Umsetzung in die Praxis wird beschleunigt und optimal an die jeweiligen Businessanforderungen ausgerichtet. Dies leistet einen wichtigen Beitrag zur Zukunftssicherung des Unternehmens.

- ITIL sorgt auch dafür, dass weniger bzw. keine Missverständnisse auf Grund von unterschiedlich interpretierbaren Begrifflichkeiten entstehen. Wenn z.B. von „Incidentmanagement“ oder „Eventmanagement“ gesprochen wird, dann ist nach ITIL klar definiert, was darunter zu verstehen ist.

- Die Kommunikation wird intern wie auch extern verbessert. Durch den erhöhten Informationsfluss können Probleme und Änderungsbedarf schon im Ansatz erkannt werden. Laufzeiten verkürzen sich dadurch.

- ITIL ermöglicht eine verbesserte und transparente Sicht über die gesamten Arbeitsabläufe und garantiert somit eine qualitätsgesicherte Leistungserbringung. Die Kunden- und Mitarbeiterzufriedenheit kann dadurch maßgeblich erhöht werden.

- Der Fokus von ITIL liegt auf dem ganzheitlichen Nutzen für Kunden und Unternehmen. Servicekosten werden gesenkt und die Qualität der IT-Serviceerbringung wird gesteigert.

- ITIL ist zwar durch seinen komplexen Aufbau eher im Umfeld von Großunternehmen zu sehen, bietet inhaltlich jedoch auch für kleine und mittelständische Unternehmen interessante Ansätze.

### 2.4.4 Die Struktur von ITIL v3

Der Aufbau von ITIL v3 stellt die ganzheitliche Sicht auf das Service Management in den Mittelpunkt. Diese Fokussierung ist ganz klar den jüngsten Entwicklungen in der IT und der Technik sowie der Innovationsfähigkeit moderner Unternehmen geschuldet. Der Kern des ITIL v3 Frameworks wird durch die 5 Kerngebiete gebildet, welchen wiederum jeweils ein eigener, einheitlich strukturierter Band gewidmet ist:

- Service Strategies
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

Zur Vermittlung eines besseren Praxisbezugs wird das ITIL v3 Framework durch ein umfangreiches Complementary Set mit verschiedenen Themenschwerpunkten ergänzt. Durch diese Ergänzungen wurde dem Vorwurf, dass ITIL ein zu theoretisches Modell zu Grunde liege, entgegengewirkt.

In *Abbildung 2-4*, wird der Aufbau des ITIL v3 Prozessmodells ersichtlich. Im Zentrum stehen die IT Strategien, die auf alle Prozesse Auswirkungen haben. Die drei operativen Prozessgebiete Service Design, Service Operation und Service Transition, sind ringförmig um die Strategien angeordnet. Sämtliche Prozessgebiete werden vom Continual Service Improvement umspannt, wodurch die stetige Präsenz des kontinuierlichen Verbesserungsprozesses über alle Prozessgebiete hinweg zum Ausdruck gebracht wird. Das ergänzende Complementary Set umfasst schließlich das gesamte Prozessmodell und enthält zu jedem Prozessgebiet praxisbezogene Vorlagen, Empfehlungen und Beispiele. [Olb08]

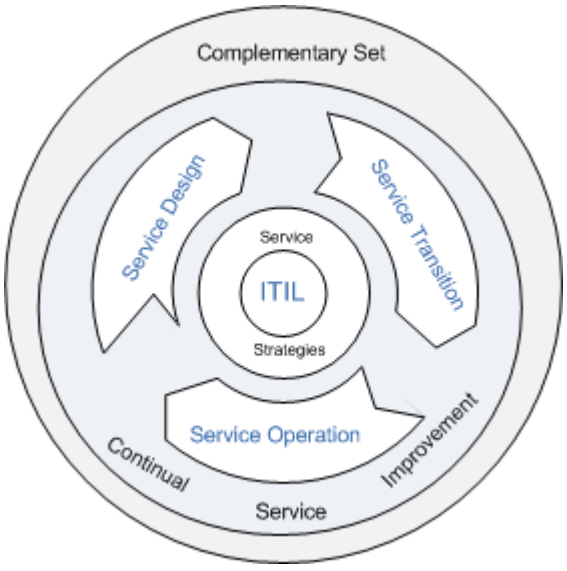


Abbildung 2-4: Prozessmodell von ITIL v3 ( [Olb08] , Seite 145)

## 2.4.5 Wichtige ITIL Service Operation Funktionen

### 2.4.5.1 Service Desk

Der Service Desk ist die wichtigste Anlaufstelle für alle IT-Benutzer eines Unternehmens oder einer Organisation, wenn es um Problemfälle, Dienstunterbrechung oder um Service-Anfragen geht. Der Service Desk stellt sozusagen einen zentralen Kommunikationspunkt für IT-Benutzer dar, über welchen alle Prozesse abgehandelt werden. Problemfälle können hierbei auf mehrere Arten, wie zum Beispiel telefonisch, mittels Textnachrichten, via Web Interfaces oder durch automatisch generierte Event Reports, gemeldet werden. Um eine effiziente Arbeitsweise des Service Desks zu gewährleisten, ist dieser normalerweise von anderen, den Servicebetrieb betreffenden Funktionen, getrennt. Der Service Desk ist von elementarer Wichtigkeit wenn um das IT-Tagesgeschäft eines Unternehmens geht. Daher sollte der Wert eines effektiven Service Desks nicht unterschätzt werden, weil dieser oftmals bestehende Defizite in IT-Umgebungen ausgleichen kann. Andererseits kann ein schlecht funktionierender Service Desk, ein schlechtes Licht auf die gesamten IT-Services einer Organisation werfen. Wie genau und in welcher Stärke ein Service Desk aufgebaut ist, variiert mit der Größe und Komplexität der IT-Landschaft und der Anzahl der zu servicierenden Benutzer. [Off13]

### 2.4.5.2 Technical Management

Das Technical Management repräsentiert eine logische Einheit, welche detailliert alle technischen Fähigkeiten und Ressourcen enthält, die zur Unterstützung des laufenden Betriebs einer IT-Infrastruktur benötigt werden. Eine wichtige Aufgabe des Technical Management besteht darin, die Konstruktion, Prüfung, Freigabe und die Verbesserung der einzelnen IT-Dienstleistungen sicherzustellen. Grundsätzlich sind das Know-how und die Funktionen des Technical Management nicht in einer einzelnen Abteilung gebündelt, sondern auf mehrere technische Fachabteilungen aufgeteilt. [Off13]

## 2.4.6 Wichtige ITIL Service Operation Prozesse

### 2.4.6.1 Event Management

Das Event Management in ITIL basiert auf Ereignissen, welchen jeweils eine definierte Bedeutung zu Grunde liegt. Diese Events sind für die kontinuierliche Bereitstellung von IT-Services und das Management von IT-Infrastrukturen von wesentlicher Bedeutung. Events sind Benachrichtigungen wie Statusmeldungen, Fehler oder Warnungen und werden in der Regel von Monitoring Tools generiert. Die Effizienz des Service Operation hängt davon ab, wie gut der Status der IT-Infrastruktur überwacht und wie effizient Abweichungen vom Normalbetrieb, erkannt werden können. Dazu können grob 2 Arten des Monitorings unterschieden werden. Zum einen gibt es aktive Monitoring-Tools, welche ständig die

Verfügbarkeit der zu überwachenden Services prüfen und zum anderen gibt es passive Monitoring-Tools, die Alarmsignale von Services registrieren und aufarbeiten. [Off13]

#### 2.4.6.2 Incident Management

Ein Incident ist eine nicht geplante Unterbrechung eines IT Service oder eine Qualitätsminderung eines IT Service. Auch ein Ausfall eines Configuration Items ohne bisherige Auswirkungen auf einen Service ist ein Incident. Der Anwender der IT Services ist auf einen reibungslosen Betrieb dieser Dienstleistung angewiesen. Wenn er die IT Services nicht mehr wie gewohnt nutzen kann oder wenn er in seiner Arbeit beeinträchtigt ist, meldet er sich beim Service Desk der IT Organisation. Incident Management ist der Prozess, der für die Verwaltung des Lebenszyklus aller Incidents verantwortlich ist. Wichtigstes Ziel des Incident Management ist eine schnellstmögliche Wiederherstellung des IT Service für die Anwender. Dieser Prozess zur Störungsbehebung ist dreistufig als First-, Second- und Third Level ausgelegt. [Gle15]

#### 2.4.6.3 Request Fulfillment

Der Prozess Request Fulfillment beschreibt die Art und Weise, wie mit Service Requests verfahren wird. Service Requests sind Benutzeranfragen, bei denen es um Beratungsanfragen oder die Anforderung einer Standardaktion wie den Zugriff auf eine neue Serviceressource für den Benutzer geht. Da es sich bei Service Requests, meist um nicht betriebskritische Vorgänge handelt, werden diese in einem separaten Prozess, nämlich dem Request Fulfillment, unabhängig von Incident- und Change Management abgehandelt. [Off13]

#### 2.4.6.4 Problem Management

Laut ITIL beschreibt ein „Problem“, die unbekannte Ursache für einen oder mehrere Incidents. Die Hauptaufgabe des Problem Management ist die Verhinderung von Problemen und der daraus resultierenden Incidents sowie die Eliminierung von wiederkehrenden Incidents und die Minimierung der Auswirkungen, von nicht zu vermeidenden Incidents. Das Problem Management beinhaltet alle, zur Ursachendiagnose von Incidents, benötigten Aktivitäten. Das Problem Management stellt sicher, dass erarbeitete Problemlösungen implementiert und durch Change- bzw. Release Management auf die betroffenen Systeme verteilt werden [Off13] .

#### 2.4.6.5 Access Management

Der Access Management Prozess behandelt die Zuweisung und Kontrolle von Benutzer-Zugriffsrechten für zur Verfügung stehende Ressourcen. Tätigkeiten im Access Management werden in der Regel von allen technischen Abteilungen durchgeführt und stellen somit keine separate Funktion dar. Die zentrale Anlaufstelle für Access Management Anfragen ist jedoch meist der Service Desk. [Off13]

### 2.4.7 Fazit

Für die IT-Verantwortlichen eines mittelständischen bis großen Unternehmens, empfiehlt es sich sehr, über die ITIL-Prozesse und Funktionen Bescheid zu wissen, um nicht eventuell sehr viel Zeit in etwas zu investieren, was es ohnehin schon gibt. Man muss sich jedoch auch vor Augen halten, dass ITIL keine fertige Gebrauchsanweisung ist, sondern lediglich vorgibt, welche Maßnahmen umzusetzen sind. So kann es bei Einsatz von ITIL nicht selten zu Unzulänglichkeiten, zwischen dem was ITIL vorgibt und dem was real benötigt wird, kommen. Die detaillierte Konzeptionierung und diverse Ergänzungen zum ITIL-Modell, bleiben den IT-Verantwortlichen bei der Umsetzung einer Implementierung also nicht erspart. [Pil10]

# 3 Anforderungsanalyse

## 3.1 Einleitung

In diesem Kapitel werden die Anforderungen an eine Service Operation Lösung, für den Einsatz in Unternehmen mit geografisch verteilten IT-Umgebungen, erörtert und analysiert. Hierzu werden zuerst die Besonderheiten, mit welchen man sich bei der Servicierung einer solchen Umgebung konfrontiert sieht, erarbeitet. Anschließend folgt eine Beschreibung bzw. Analyse von 100 realen Problemfällen - sprich Incidents - welche an den Service Desk eines großen IT-Dienstleisters für Kunden im Finanzsektor, gemeldet wurden. Das Hauptaugenmerk liegt hierbei bei der Analyse, wie und wie effizient die Problematik zwischen Kunden und Support vermittelt wurde und durch welche Mechanismen die Lösung herbeigeführt werden konnte. Die Ergebnisse dieser Analyse fließen dann in die Erstellung eines Katalogs von speziellen Anforderungen an eine Lösung für den Servicebetrieb ein. Für einen Abgleich mit dem Stand der Technik in diesem Umfeld, werden auch noch aktuelle ITSM Lösungen zweier kommerzieller Anbieter evaluiert. Die Lösungen werden auch auf Tauglichkeit hinsichtlich der möglichen Erfüllung, der zuvor definierten speziellen Anforderungen, geprüft und bewertet.

## 3.2 Besonderheiten geografisch verteilter IT-Landschaften

Im Vergleich zu der klassisch konservativen IT-Umgebung, in welcher sämtliche IT-Ressourcen inklusive IT-Betreuer am jeweiligen Standort direkt anzutreffen waren, sieht man sich bei modernen Unternehmen mit entsprechender IT-Infrastruktur mit einem weit komplexeren Aufgabenbereich konfrontiert. Im Speziellen davon betroffen sind IT-Umgebungen, welche sich nicht nur auf den LAN- bzw. Intranet Betrieb beschränken, sondern auch über deren Grenzen hinweg Services konsumieren und auch bereitstellen. Trotz der oftmals geografisch weit verteilten IT-Landschaften, werden Kernkompetenzen und vor Allem auch der Anwenderbetreuung, aus Kosten- und Effizienzgründen zentralisiert gehalten. Die Besonderheiten und die Unterschiede die sich dadurch ergeben, sollen hier nun aufgegriffen, wiedergegeben und vermittelt werden.

### 3.2.1 Firewalls und Protokolle

Um Kommunikationsverbindungen zwischen geografisch verteilten Standorten realisieren zu können, ist die Überwindung von WAN - Wide Area Network - Strecken unerlässlich. Somit werden die Kommunikationsdaten auch über öffentliche Netzwerke, die im Allgemeinen als

Internetze bezeichnet werden können, transportiert. Während das eigene Interne Netzwerk einer Organisation oder eines Unternehmens als quasi sicher und vertrauenswürdig bezeichnet werden kann, gilt das umso weniger für die öffentlichen Netzwerke. Ein Schutz gegen unerwünschte und schadhafte Einflüsse, auf den Netzwerkverkehr in einen und aus einem geschützten Netzwerkbereich, wird durch den Einsatz von Firewalls (siehe *Abbildung 3-1*) realisiert. Eine Firewall stellt quasi eine Kontrollstelle beim Übergang zwischen zwei Netzbereichen dar, durch welche gesteuert werden kann, welche Protokolle und Ports zur Kommunikation zwischen den beiden Netzbereichen verwendet werden dürfen und können.

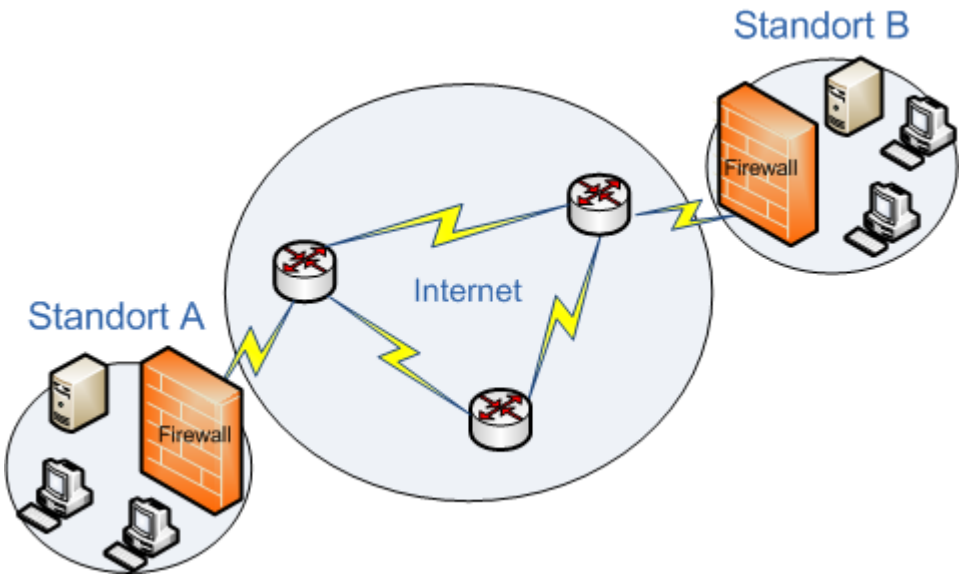


Abbildung 3-1: Schematische Darstellung für Firewall Einsatz

Es gibt diverse Protokolle wie zum Beispiel das Hypertext Transfer Protocol (HTTP), welche standardmäßig eine Freischaltung zur Kommunikation durch eine Firewall hindurch haben. Im Falle von HTTP ist die Freischaltung für die Kommunikation von Webbrowsern und Webservern, also dem klassischen „Internetsurfen“, erforderlich. Hat man jedoch Applikationen, welche den standortübergreifenden Datenaustausch über Methoden wie z.B. die Windows Management Instrumentation (WMI) von Microsoft, oder native Remote Method Invocation (RMI) in JAVA realisieren, sind zusätzliche Freischaltungen auf allen am Kommunikationsfluss beteiligten Firewalls erforderlich. Dadurch wird die Flexibilität nicht unbedeutend eingeschränkt.

### 3.2.2 Performance und Latenzzeiten

Im Vergleich zu Applikationen die für den LAN-Betrieb entwickelt wurden, müssen bei der Entwicklung und Implementierung von Applikationen, die über WAN-Strecken hinweg Daten senden und empfangen, die Nachteile und Schwachstellen der Kommunikationswege im Auge behalten werden. Bei WAN-Lösungen lässt sich das, bei LAN-Lösungen übliche Konzept der synchronen Kommunikation schlecht umsetzen, weil dadurch zu große Latenzzeiten entstehen würden. Hierfür eignet sich das Konzept der Asynchronen

Kommunikation, bei welchem während Wartezeiten, andere Aufgaben abgearbeitet werden. Dadurch können Latenzzeiten verborgen bzw. verschleiert werden. Bei Kommunikation über WAN-Strecken wird eine Punkt-zu-Punkt Verbindung aufgebaut, welche über etliche Knotenpunkte führen kann, dadurch entsteht eine gewisse Unstetigkeit und somit Unzuverlässigkeit der Kommunikationsverbindung.

### 3.2.3 Abhörsichere Datenübertragung

Ein sehr hoher Prozentsatz der Daten, die in Unternehmensnetzwerken tagtäglich von A nach B übertragen werden, sind entweder aus Privatsphäregründen, oder weil es um schützenswerte Inhalte geht, als sensibel zu betrachten und müssen somit gegen unautorisierte Zugriffe geschützt werden. Für den Datenverkehr in Intranets also in abgegrenzten vertrauenswürdigen Netzen, ist dieses Thema weit weniger brisant, als für den Datenaustausch über WAN-Strecken hinweg. Muss man sich in Intranet-Umgebungen zumindest einmal physikalisch Zugriff für eine Angriffsmöglichkeit schaffen, fällt diese Schranke für die externe Kommunikation praktisch weg.

Für die standortübergreifende Übertragung von sensiblen Daten muss also ein passendes Konzept und die entsprechende Technologie gewählt werden. Hierfür kann z.B. auf Technologien wie IPsec[6] oder auf den Einsatz einer VPN-Lösung[7] zurückgegriffen werden. Bei einer Verschlüsselung von Datenverkehr über WAN-Strecken muss jedoch berücksichtigt werden, dass dadurch mit Performanceeinbußen bei der Datenübertragung gerechnet werden muss. Das Ver- und Entschlüsseln der Daten, verursacht je nach Stärke der Verschlüsselung eine bestimmte Verzögerung, welche sich auf die Übertragungszeit auswirkt.

### 3.2.4 Anwendersupport

Die gesammelten Vorgänge und Tätigkeiten, welche für die Unterstützung von IT-Benutzern im Bezug auf Fehleranalysen, Bereitstellung neuer Ressourcen und Problembehebungen angeboten werden, sind dem Anwendersupport zuzuordnen. Das Personal, welches diesen Bereich repräsentiert ist jedoch in der Regel in nur einem Standort gebündelt, weil zum einen etliche Support-Tätigkeiten durch Fernzugriffe durchgeführt und zum anderen dadurch Kosten gespart werden können. Oftmals ist es jedoch unerlässlich, dass qualifiziertes IT-Personal vor Ort ist. Diese Fälle müssen bei geografisch verteilten IT-Umgebungen entweder durch die Anreise eines Mitarbeiters, oder durch Outsourcing der Tätigkeit an lokale Dienstleister abgedeckt werden.

[6] Internet Protocol Security ist eine Protokoll-Suite, die eine gesicherte Kommunikation über potentiell unsichere IP-Netze ermöglicht. (siehe https://de.wikipedia.org/wiki/IPsec)
[7] Virtual Private Network. (siehe https://de.wikipedia.org/wiki/Virtual_Private_Network)

### 3.2.5 Proaktive Störungsbehebung

Wie schon unter Punkt 3.2.1 angedeutet, besteht meist eine Limitierung der erlaubten Kommunikationsprotokolle und Ports zwischen zwei Standorten, welche durch den Einsatz von Firewalls verursacht wird. Dies wirkt sich natürlich auch auf alle Applikationen aus die nicht via HTTP kommunizieren. Als Beispiel soll nun ein Serviceprozess an einem entfernten Standort dienen, der einen Fehler verursacht und schon durch ein Monitoring-Tool als fehlerhaft erkannt wurde. Um den Service wieder in den Normalbetrieb zu versetzen muss eine Steueraktion gesetzt werden. Diese Aktion kann in sicheren Intranets relativ unkompliziert durch einen Remoteprozeduraufruf durchgeführt werden. Müssen jedoch zusätzlich noch WAN-Strecken und somit Firewalls überwunden werden, ist die gesonderte Freischaltung des genutzten Übertragungsprotokolls erforderlich.

## 3.3 Auswertung realer Problemfälle

Als Teil der Anforderungsanalyse wurden, wie schon unter Punkt 3.1 kurz beschrieben, 100 Incidents hinsichtlich wie und wie effizient die Problematik zwischen Kunden und Support vermittelt wurde und wie die Lösung des Problems herbeigeführt werden konnte, analysiert und ausgewertet. Die IT-Landschaft, welche als Einsatzgebiet für diese Feldstudie diente, setzt sich aus vielen unterschiedlichen Konzepten zusammen. Angefangen vom normalen Client-Server-Betrieb[8], bei dem ein Benutzer mit einem Fat-Client arbeitet und nur Dateiressourcen von einem Server bezieht, bis hin zum Thin-Client in Kombination mit Terminal Servern in geografisch entfernten Standorten. Mit einem Volumen von durchschnittlich 2000 gemeldeten Incidents pro Monat, kann eine Auswertung von ca. 5% des Gesamtvolumens, als repräsentativ angesehen werden.

Die angesprochenen 100 Incidents wurden zur Auswertung aus dem Incident-Managementsystem, des unter *Punkt 3.1* beschriebenen IT-Dienstleisters, exportiert und als HTML-Report gespeichert. Um die Problemfälle analysieren zu können, wurden folgende Kriterien definiert:

- **Initialer Informationsgehalt:** Wurden alle, für den Supportmitarbeiter benötigten Information, zur korrekten Bearbeitung des Incidents bereits bei der Erstellung übermittelt?

- **Zusatzinformation durch Logs:** Konnte sich der Bearbeiter des Incidents die restlichen Informationen, welche zur Lösungsfindung im aktuellen Problemfall benötigt werden, durch Auswertung von vorhandenen Logdateien beschaffen?

[8] Der Client-Server-Betrieb wird durch das Client-Server-Modell beschrieben. (siehe https://de.wikipedia.org/wiki/Client-Server-Modell)

- **Kommunikation:** Wurden zusätzliche, zur Lösung benötigte Informationen, Schriftlich oder Mündlich übermittelt?

- **Wartezeiten:** War die Wartezeit auf die durch den Support benötigten Informationen, länger als 60 Minuten?

- **Lösung durch First-Level-Support:** Konnte der Incident durch den First-Level-Support[9] gelöst werden oder wurde dieser an den Second-Level-Support[10] weiter eskaliert?

- **Bereinigung durch Standardaktion:** Konnte das Problem durch Ausführung einer bereits dokumentierten standardisierten Aktion gelöst werden?

Anhand dieser Kriterien wurde nun jeder der 100 Incidents betrachtet und ausgewertet. Dazu wurde der jeweilige, im HTML-Report protokollierte Incident-Verlauf, mit den zuvor definierten Kriterien abgeglichen. Alle Einträge im Incident-Verlauf sind mit einem Zeit- und Datumsstempel versehen sowie mit einer kurzen Notiz über Art und Ersteller der Information gekennzeichnet. Auf Grund dieser Informationen kann die Erfüllung der definierten Kriterien geprüft werden. Incidents, welche einen nicht eindeutig auswertbaren Incident-Verlauf aufwiesen, wurden gestrichen und durch neu ausgewertete Incidents ersetzt.

Die Ergebnisse der Auswertung der einzelnen Incidents lassen sich grafisch durch eine Art Flussdiagramm[11] (siehe *Abbildung 3-2* auf der nächsten Seite) darstellen. So kann die Häufigkeit, mit der die einzelnen Kriterien erfüllt wurden, gut sichtbar gemacht werden.

[9] Der First-Level-Support ist die erste Anlaufstelle für alle eingehenden Unterstützungsfragen. (siehe https://de.wikipedia.org/wiki/First-Level-Support)
[10] Der Second-Level-Support unterstützt den First-Level-Support, sowohl durch Weiterbildung am Arbeitsplatz als auch durch Übernahme komplexerer Anfragen. (siehe https://de.wikipedia.org/wiki/First-Level-Support)
[11] Ein Flussdiagramm ist eine grafische Darstellung der logischen Schritte eines Problems oder Programmablaufs mit speziellen Symbolen.
(siehe http://www.itwissen.info/definition/lexikon/Flussdiagramm-flow-chart.html)

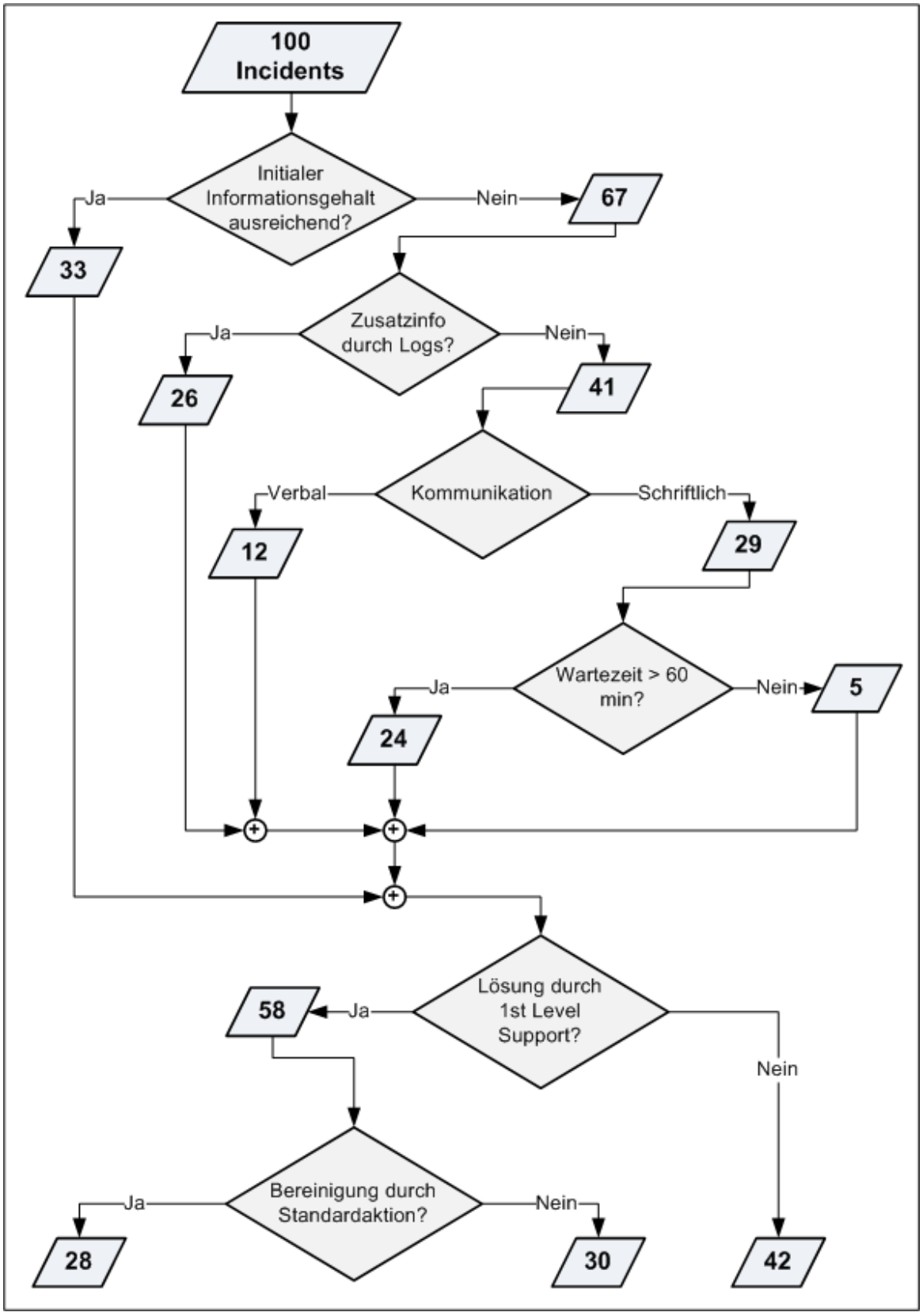


Abbildung 3-2: Incident Auswertung

Bei der Betrachtung des Resultats, dargestellt in *Abbildung 3-2*, sieht man, dass bei 67% der Incidents ein unzureichender Informationsgehalt vorhanden war, sodass eine Weiterbearbeitung erst nach Einholung weiterer Informationen möglich war. Des Weiteren

konnten 26% der Incidents, ohne auf weitere Information durch betroffene Benutzer angewiesen zu sein, weiter analysiert werden. Bei 24% der Incidents kam es während der Wartezeit auf weitere Informationen, zu einer Verzögerung von über einer Stunde. Die Lösungsquote des First-Level-Supports beträgt immerhin 58%, wobei 28% der Fälle durch dokumentierte Standardaktionen gelöst werden konnten.

Nicht zuletzt durch die visuelle Aufbereitung, können durch solch eine Auswertung die Schwachstellen eines Systems aufgezeigt und somit die Prozesse identifiziert werden, welche noch Optimierungsbedarf aufweisen. Die im konkreten Fall identifizierten wesentlichen Schwachstellen, sollen nun beleuchtet und die notwendigen Verbesserungen bzw. Änderungen erörtert werden.

- **Initialer Informationsgehalt:** Die Übermittlung von Informationen ist stets der erste Schritt in Richtung Problemlösung. Der Informationsgehalt dieser Aktion ist meist schon von entscheidender Bedeutung dafür, ob, durch wen und wie schnell ein gemeldetes Problem gelöst werden kann. In der getätigten Incident-Auswertung, wurde dieses Ziel in zwei Drittel der Fälle nicht erreicht. Es hat also in den schlechten Fällen, zumindest eine Kerninformation gefehlt, welche unbedingt zur weiteren Bearbeitung notwendig gewesen wäre. Wie kann hier eine Verbesserung der Situation erreicht werden? Ein erster Schritt in die richtige Richtung wäre sicherzustellen, dass auch Laien in der Lage sind, alle wichtigen Informationen zu beschaffen und zu übermitteln. Dies könnte z.B. durch eine simple Anwendung, welche dem Benutzer per Tastendruck wesentliche Informationen über Benutzerdaten, Systeminformationen und dergleichen ausgibt, realisiert werden. Auf jeden Fall sollte aber die Anzahl der Pflichtfelder für die Incident-Meldung erhöht und eventuell auch noch spezialisierter auf unterschiedliche Problembereiche abgestimmt werden.

- **Zusätzliche Informationen durch Logs:** Sind die im ersten Schritt bereitgestellten Informationen für den Supportmitarbeiter nicht ausreichend um den Incident weiterführend bearbeiten zu können, müssen weitere Informationen eingeholt werden. Idealerweise wurden diese bereits in irgendeiner Form protokolliert und können sogleich durch den Supportmitarbeiter, ohne weitere Kontaktaufnahme mit dem betroffenen Benutzer, abgerufen werden. In der Incident-Auswertung hat sich gezeigt, dass bei gut einem Viertel der Incidents, zusätzliche Informationen aus bestehenden Protokollierungsmechanismen gewonnen werden konnten. Zur Steigerung der guten Fälle, müsste die Protokollierung der einzelnen Vorgänge ausgeweitet und gesteigert werden. Der Implementierung der Vorgangs-protokollierung muss also praktisch fast ident viel Aufmerksamkeit beigemessen werden, wie der Implementierung des Prozesses selbst. Der dadurch gesteigerte Arbeitsaufwand bei der Implementierung, könnte sich jedoch durch Senkung der Problem-Wiederherstellungszeiten bezahlt machen. Des Weiteren, ist die dauerhafte und schnelle Verfügbarkeit der protokollierten Daten, von wesentlicher Bedeutung. Hierbei sollten, wenn möglich keine Verzögerungen entstehen.

- **Kommunikation:** Müssen noch offene Fragen schlussendlich doch direkt mit dem betroffenen Benutzer geklärt werden, geschieht dies in der Regel schriftlich oder verbal. Der Vorteil bei der verbalen Kommunikation liegt auf der Hand. Offene Fragen können sofort im Gespräch geklärt werden, dadurch entstehen keine weiteren Wartezeiten. Manche Informationen lassen sich jedoch, aus verschiedenen Gründen, besser schriftlich vermitteln. In diesem Punkt bleibt also zu sagen, dass hier von Fall zu Fall differenziert werden muss, wie am Besten kommuniziert werden sollte.

- **Bereinigung durch Standardaktion:** Wie viele Incidents, durch den First-Level-Support behoben werden können, hängt typischerweise davon ab, wie gut die einzelnen Mitarbeiter geschult wurden. Aber auch davon, wie häufig ein Incident durch standardisierte und dokumentierte Vorgänge gelöst werden kann. Der, in der Incident-Auswertung erreichte Prozentsatz weist mit 28% einen ausbaufähigen Wert auf. Um in diesem Bereich eine Steigerung zu erreichen, müssen Lösungswege dokumentiert und standardisiert zur Durchführung angeboten werden. Eine wichtige Voraussetzung für diese Tätigkeit ist jedoch, eine generell höhere Standardisierung der sich im Einsatz befindlichen Prozesse.

Abschließend bleibt zu bemerken, dass durch eine relativ triviale Auswertung einer repräsentativen Anzahl von Incidents, verblüffend gut, die Möglich- und Unzulänglichkeiten eines bestehenden Systems, sichtbar gemacht werden können. Die Ergebnisse bilden eine solide Basis, für zukünftige Qualitätsverbesserungen am bestehenden System.

## 3.4 Spezielle Anforderungen an eine Lösung für den Servicebetrieb

Einer der wichtigsten Vorgänge beim Softwaredesign, ist die Definition und Klarstellung, über welche Funktionalitäten und Eckdaten ein System, nach Abschluss der Entwicklungsphase verfügen muss. Diese Forderungen werden in zwei Untergruppen, nämlich funktionale und nicht-funktionale Anforderungen, unterteilt. Funktionale Anforderungen stehen für die gewünschten Funktionalitäten während nicht-funktionale Anforderungen die Qualität bestimmen, in welcher die geforderten Funktionalitäten zu erbringen sind. Ein gutes Software-Produkt muss im Endeffekt zumindest alle definierten Anforderungen erfüllen können.

Nachfolgend werden nun einige wesentliche, spezielle Anforderungen an eine ITSM-Servicebetrieb- Lösung für den Einsatz in Unternehmen mit geografisch verteilten Systemen definiert. Das Ziel ist es, ein Konzept zu schaffen welches darauf abzielt, Problemlösungsvorgänge im Bezug auf Bearbeitungszeit und Personaleinsatz zu optimieren. Des Weiteren soll das Konzept problemlos auf komplexe und geografisch verteilte IT-Umgebungen anwendbar sein. In die Erstellung der Anforderungen, fließen

sowohl die bis dato in dieser Arbeit aufbereiteten und durchgearbeiteten Fakten sowie auch die gewonnen Erkenntnisse der getätigten Incident-Auswertung mit ein.

### 3.4.1 Funktionale Anforderungen

Die zu entwickelnde Lösung für den ITSM-Servicebetrieb, wird im Zuge der Auflistung der einzelnen Requirements, einfachheitshalber kurz „Lösung“ genannt.

**REQ01:**
Der Zugriff auf die Lösung muss für die Problemmelder via Webbrowser möglich sein. Es darf keine zusätzliche Installation einer Softwarekomponente, für den Zugriff der Problem-Melder auf die Lösung erforderlich sein.

**REQ02:**
Die Lösung muss zumindest für Incidentmanagement und Auftragsmanagement (Request Fullfilment) einen Workflow beinhalten. Des Weiteren muss die Möglichkeit gegeben sein, über eine spezielle Administrations- und Customizing GUI-Maske, weitere Prozesse hinzuzufügen und die bestehenden Workflows abzuändern.

**REQ03:**
Für alle möglichen Vorgänge müssen jeweils ein Muster-Template und eine Hilfefunktion vorhanden sein, um den Benutzern die Anlage zu erleichtern und somit den Prozess zu beschleunigen.

**REQ04:**
Eine zentrale Datenbank muss als Teil der Lösung implementiert werden. In dieser Datenbank müssen sämtliche protokollierten Daten gespeichert und abgerufen werden können.

**REQ05:**
Die Lösung muss einen Software Client beinhalten, welcher auf jedem zu betreuenden Zielsystem einmalig installiert wird. Diese Software wird für die Protokollierung lokaler Informationen, für die Kommunikation mit der zentralen Datenbank und für die Ausführung von Steuerkommandos eingesetzt.

**REQ06:**
Welche Informationen der Software Client protokollieren soll, muss an zentraler Stelle konfigurierbar sein und bei Bedarf als Konfigurationsset an die Clients verteilt werden können.

**REQ07:**
Die Kommunikation zwischen den einzelnen, an der Lösung beteiligten Software-komponenten, muss so realisiert werden, dass sie auch zwischen geografisch entfernten Standorten, verlässlich durchgeführt werden kann.

**REQ08:**
Für Anwender muss die Lösung eine GUI-Maske bieten, in welcher für alle unterstützen Prozesse, Anfragen und Problemmeldungen erstellt und an den Support gesendet übermittelt werden können. Die Anmeldung muss an dieser GUI-Maske mit dem jeweiligen LAN-Benutzerkonto des Anwenders möglich sein.

**REQ09:**
Die Lösung muss für die Supportmitarbeiter eine GUI-Maske bieten, in welcher die gemeldeten Problemfälle bearbeitet werden können. Zusätzlich muss für die Supportmitarbeiter eine GUI-Maske vorhanden sein, in welcher sämtliche gesammelten Protokollinformationen, abgerufen werden können. Es muss die Möglichkeit vorhanden sein, nach Stichworten, wie Benutzername oder Zielsystem zu suchen, um dadurch das gewünschte Informationsset zu erhalten. In dieser GUI-Maske müssen auch die schon implementierten Steuervorgänge zur Auswahl stehen und per Mausklick ausgeführt werden können. Die Anmeldung an die GUI-Maske muss mit dem jeweiligen LAN-Benutzerkonto des Supportmitarbeiters möglich sein.

### 3.4.2 Nicht-funktionale Anforderungen

**REQ10:**
Die Lösung muss entsprechend skalierbar konzipiert werden, sodass bis zu 1000 Clientsysteme problemlos eingebunden werden können.

**REQ11:**
Der Schutz der Daten und der Kommunikationswege muss durch Einsatz von Verschlüsselungstechnologien gewährleistet sein.

**REQ12:**
Die Menüsprache der Lösung muss veränderbar sein. Die Standardsprache muss Deutsch sein. Optional muss zumindest Englisch wählbar sein.

**REQ13:**
Die Latenzzeit für Datenbankabfragen darf im Regelfall 10 Sekunden nicht übersteigen.

**REQ14:**
Die Menüführung für die Servicebetrieb-Prozesse muss so gestaltet sein, dass alle geforderten Informationen vollständig angegeben werden müssen, bevor eine Speicherung und Weiterleitung erfolgen kann.

**REQ15:**
Die Lösung muss ausfallsicher implementierbar sein. Das bedeutet, dass die Lösung von zumindest 2 unabhängigen Stellen gleichzeitig gehostet werden können muss.

# 3.5 Evaluierung kommerzieller Lösungen

Bei der Selektion einer geeigneten ITSM-Lösung für eine zu betreuende IT-Umgebung, kann heutzutage aus einer Vielzahl von kommerziellen und auch Freeware Produkten gewählt werden. Wesentlich dabei ist, welche Funktionen benötigt werden, wie hoch die dadurch entstehenden Kosten sind bzw. ob überhaupt Kosten dadurch entstehen dürfen und welche Größenordnungen eine Lösung abdecken kann. Auf Grund der Tatsache, dass in dieser Diplomarbeit eine Lösung für Unternehmen mit komplexen geografisch verteilten IT-Umgebungen erarbeitet werden soll, sind hierfür eher die aktuelleren Produkte, von am Markt etablierten Herstellern, von Interesse. Nachfolgend werden nun die Lösungen zweier kommerzieller Anbieter evaluiert und dahingehend bewertet, wie gut sie sich die dazu eignen, die erarbeiteten speziellen Anforderungen an eine Lösung für den ITSM-Servicebetrieb, abzudecken. Bei der Auswahl der beiden, zu bewertenden ITSM-Produkte, wurde gezielt nach Lösungen gesucht, welche sich grundsätzlich für den Einsatz in geografisch verteilten IT-Umgebungen eignen. Ein weiterer wichtiger Faktor bei der Auswahl, war die ITIL-Konformität der Produkte, sodass die Ausgangsbasis für eine Bewertung der beiden Lösungen gleich ist.

## 3.5.1 BMC Remedy IT Service Management Suite

### 3.5.1.1 Kurzbeschreibung

Der US-amerikanische Softwarehersteller BMC Software Inc. ®, bietet mit dem *BMC Remedy Action Request System 8.1*, eine Sammlung von Arbeitsablauf-Management-Werkzeugen, für den Einsatz in mittleren und großen Unternehmen. Die einzelnen Applikation und Funktionen sind hierbei in Module unterteilt und können somit individuell, auf den Bedarf abgestimmt, erworben und installiert werden. Die *BMC Remedy IT Service Management Suite* ist ein solches Applikations-Modul. [BMC152]

### 3.5.1.2 BMC Remedy AR System Architektur

*BMC Remedy AR System* basiert auf einer mehrschichtigen Client-Server-Architektur. Die Benutzeranwendungsschicht, das sogenannte *Client Tier*, beinhaltet die *AR Systems Clients*. Die Clients dienen zur Ein- und Ausgabe von Daten für Benutzer, es gibt aber auch die Möglichkeit, mit einem Client Applikationen zu erstellen oder auch zu konfigurieren.

Im *Mid Tier* befinden sich die Komponenten und Dienste, welche auf einem Webserver laufen und dem Benutzer die Anzeige und Verwendung von Web Applikationen ermöglichen. Die *Server Tier* beinhaltet *die BMC Remedy AR System* Server, welche die Workflow-Prozesse und die Zugriffe auf Datenbanken und andere Datenquellen kontrollieren. Ebenso werden in dieser Schicht serverseitige Applikationen bereitgestellt.
Im *Data Tier* befinden sich die Datenbankserver bzw. weitere Datenquellen. Auf diese Daten wird von den *BMC Remedy AR System* Servern zugegriffen, somit dient diese Schicht als

Datenspeicher für das System. In *Abbildung 3-3* ist diese Schichten-Architektur zum besseren Verständnis grafisch dargestellt.

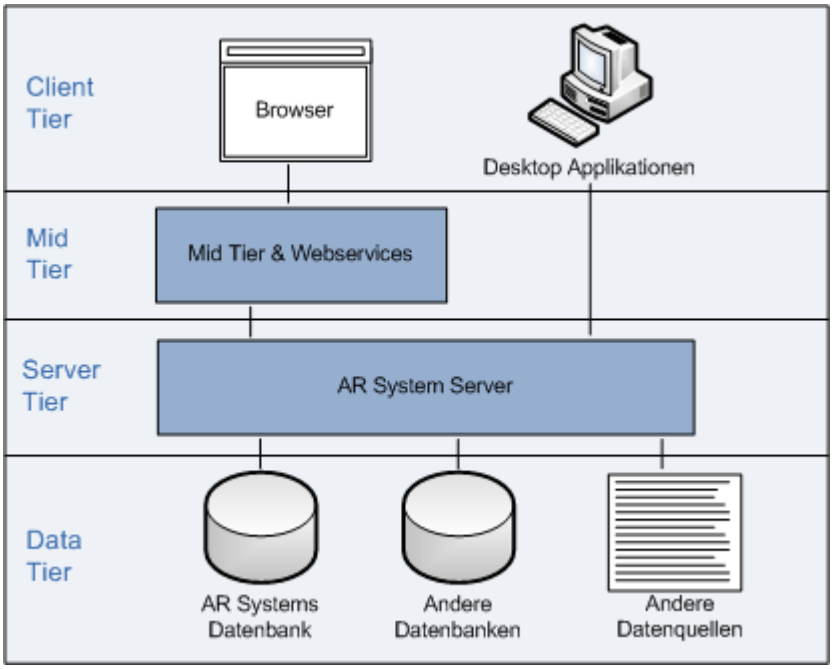


Abbildung 3-3: BMC Remedy System Architektur ( [BMC15] )

Wie in *Abbildung 3-3* ersichtlich, stellen die AR System Clients eine direkte Benutzerschnittstelle, für den Zugriff auf die AR System Server, bereit. Die *Mid Tier* Schicht ermöglicht den Zugriff via Webbrowser.

#### 3.5.1.3 Funktionsumfang

Der Einsatz von *BMC Remedy ITSM Suite* ermöglicht die Verwendung folgender Produkte und deren Funktionen:

- **BMC Service Desk:** Dieses Produkt beinhaltet BMC Incident Management und BMC Problem Management. Die Funktionen der Beiden, basieren auf den ITIL® Best Practices und ermöglichen eine individuelle Gestaltung und Automatisierung der zugehörigen Workflows. Als Kommunikationsplattform zwischen Problemmelder und Support bietet BMC Service Desk, die Konsole für Incidents bzw. Problems. Hierüber können Informationen und Dateien ausgetauscht werden. Der Support ist jedoch stets auf die Interaktion mit dem Problemmelder angewiesen.

- **BMC Asset Management:** Erlaubt das vereinfachte Durchführen von IT-Assets, indem assistentengestützte Berichte erzeugt und unkompliziert ausgewertet werden können.

- **BMC Change Management:** Unterstützt die Verantwortlichen bei der Planung, Verfolgung und Durchführung von Softwareänderungen und sorgt für eine Risikominimierung. Realisiert wird dies durch erweiterte Funktionen wie Konflikterkennung und visuelle Change-Kalender.

- **BMC Knowledge Management:** Ermöglicht die Entlastung von IT-Personal, durch Sammlung von Informationen über bereits bearbeitete Problemfälle, welche bei Bedarf schnell und unkompliziert abgefragt werden können.

Durch die zur Verwendung angebotenen Funktionen lassen sich grundsätzlich die Standardanforderungen, an eine Lösung für ITSM-Servicebetrieb abdecken.

#### 3.5.1.4 Voraussetzungen und Installation

Die in jedem Einsatzfall benötigte primäre Softwarekomponente, *BMC Remedy AR System* Server, kann auf UNIX, Linux oder auch auf einem Microsoft Windows System installiert werden. Bei der für das System benötigten Datenbank kann zwischen Microsoft SQL Server, Oracle, IBM DB2 und Synbase gewählt werden. Die bereitgestellte Datenbank kann im Prinzip auf einem beliebigen System installiert werden, wichtig ist nur, dass eine Verbindung zum *AR System* Server besteht. Bei der *AR System* Server Installation legt der Installer alle initial benötigten Datenbankinstanzen und Tabellen an.

Die optionale Komponente, *BMC Remedy Mid Tier*, kann ebenso wie der System Server auf UNIX, Linux oder Microsoft Windows installiert und betrieben werden. Die *Mid Tier* Komponente, kann auf ein und demselben System mit dem System Server, oder auf einem via Netzwerk verbundenen System installiert werden. Für den Clientzugriff auf die *Mid Tier* Funktionalitäten, muss nur ein vom System unterstützter Webbrowser verfügbar sein.

Für die Installation weiterer Komponenten werden je nachdem bestimmte Komponenten vorab benötigt. Die *Tabelle 3-1* listet die BMC-systemseitigen Minimum-Installations-Voraussetzungen, für den Einsatz der *BMC Remedy ITSM Suite*, auf.

| BMC Applikation | Vorausgesetzte Installationen |
|---|---|
| Remedy ITSMS | AR System Server<br>Approval Server<br>Assignment Engine<br>Atrium Integrator Spoon<br>Full Text Search (FTS) , jedoch nur bei Einsatz von BMC Knowledge Management |

Tabelle 3-1: Voraussetzung für Remedy ITSMS Installation ( [BMC151] )

#### 3.5.1.5 Skalierbarkeit

Grundsätzlich rechtfertigt sich der Einsatz dieser Lösung, auf Grund einer doch nicht unerheblichen, für den Betrieb notwendigen Infrastruktur, erst ab einer gewissen Größe der zu verwaltenden IT-Umgebung. Der Hersteller gibt standardmäßig drei Sizing-Szenarien vor, von denen die SMALL-Variante schon für 800 gleichzeitige Benutzer ausgelegt ist. Die Varianten MEDIUM und LARGE, können 2000 bzw. 5000 gleichzeitige Benutzer bedienen. Diese Tatsache bestätigt, dass der Fokus deutlich auf Großkunden gerichtet ist. Die Verteilung der einzelnen Software-Komponenten kann an sich frei gewählt und beliebig erweitert werden. Es ist jedoch darauf zu achten, dass die möglichen Latenzzeiten, die auf den Kommunikationswegen anfallen, sich nicht negativ auf die Gesamt-Performance auswirken. So könnte z.B. ein AR System Server in einer Niederlassung in New York stehen und ein weiterer in Tokyo. Die Nutzdaten werden im Hintergrund, zwischen den beiden Niederlassungen vom System durch Transfers und Updates aktuell gehalten. Aus Client-Sicht verhält sich das System jedoch wie eine einzelne Instanz.

#### 3.5.1.6 Bewertung hinsichtlich der speziellen Anforderungen

Anhand der in *Punkt 3.4* festgelegten Requirements kann bewertet werden, wie gut und wie vollständig *BMC Remedy ITSMS* die gestellten, speziellen Anforderungen an eine ITSM-Lösung für den Servicebetrieb, grundsätzlich erfüllen kann.

- **Zu REQ1:** Kann vollständig erfüllt werden, weil bei Einsatz von *Remedy Mid Tier*, der Zugriff via Webbrowser ohne zusätzliche Installationen möglich ist. Ein installierter Webbrowser wird als vorhanden angenommen.

- **Zu REQ2:** Kann vollständig erfüllt werden, weil über eine Administrations-Client-Software, alle Templates und Prozesse, die für den Servicebetrieb vorhanden sind, individuell angepasst werden können.

- **Zu REQ3:** Kann im Prinzip erfüllt werden, weil alle Masken individuell angepasst werden können.

- **Zu REQ4:** Kann im Prinzip erfüllt werden, weil *BMC Remedy ITSMS* mit verschiedenen Datenbanktypen kommunizieren und deren Daten ins System einbinden kann.

- **Zu REQ5:** Kann vom System nicht erfüllt werden, solch eine Komponente wird Out-of-the-Box nicht angeboten. Diese Variante müsste erst programmiert und über eine Schnittstelle an das System angebunden werden.

- **Zu REQ6:** Kann vom System OOTB nicht erfüllt werden, dies müsste separat programmiert und über eine Schnittstelle an das System angebunden werden.

- **Zu REQ7:** Kann vollständig erfüllt werden, weil *BMC Remed*y so konzipiert wurde, dass die einzelnen Komponenten nicht am selben Standort stehen müssen.

- **Zu REQ8:** Kann vollständig erfüllt werden, weil Active Directory Integration durch den Einsatz der Komponente *BMC Atrium Single Sign-On* ermöglicht wird.

- **Zu REQ9:** Kann nur zum Teil vom System erfüllt werden, weil dieses Requirement auf **REQ5** aufbaut und dieses nicht OOTB erfüllt werden kann.

- **Zu REQ10:** Kann vollständig erfüllt werden, dies wird durch die vom Hersteller angebotene MEDIUM-Variante bestätigt. Diese Variante ermöglicht schon eine entsprechende Anzahl von eingebundenen Clientsystemen, wenn auch so eine Funktion nicht OOTB vorgesehen ist.

- **Zu REQ11:** Kann vollständig erfüllt werden, weil die Kommunikation zwischen Clients und Mid Tier via HTTPS, also sprich SSL-verschlüsselt stattfindet.

- **Zu REQ12:** Kann vom System vollständig erfüllt werden. Der Umstand, dass die System-Standardsprache Englisch ist, wird in dieser Bewertung nicht berücksichtigt.

- **Zu REQ13 bis REQ15:** Diese können prinzipiell Alle vom System erfüllt werden, sind jedoch schon sehr spezialisiert und hängen teilweise auch davon ab, wie das System dimensioniert wird.

## 3.5.2 TOPdesk

### 3.5.2.1 Kurzbeschreibung

Das internationale Softwareunternehmen TOPdesk, mit Spezialisierung auf den Bereich IT-Servicemanagement, bietet mit der gleichnamigen Softwarelösung *TOPdesk* in der Version 5.6, ein modular zusammenstell- und erweiterbares ITSM Software-Produkt. Alle Versionen in denen *TOPdesk* angeboten wird, arbeiten zu 100% webbasiert und bauen auf den ITIL Kriterien auf. Per Definition eignet sich dieses Softwareprodukt für den Einsatz in Unternehmen jeder Größenordnung. [TOP15]

### 3.5.2.2 Architektur von TOPdesk

Die an einer implementierten *TOPdesk*-Lösung beteiligten Komponenten, bilden eine mehrschichtige Client-Server-Architektur. Die Client-Schicht beinhaltet die System-Zugangsvarianten für *TOPdesk*-Endanwender, *TOPdesk*-Bearbeiter und *TOPdesk*-Administratoren. Die Client-Schicht liefert somit die unterschiedlichen Möglichkeiten für den Systemzugriff.

Die Applikations-Schicht stellt der Client-Schicht alle Services und Komponenten, welche vom System angeboten werden, zur Verfügung. Dazu zählen unter anderem die Meldung von Incidents und Problems sowie Administrationsschnittstellen für Konfigurations-änderungen am System.

Die Ressourcenschicht wird von ein oder mehreren Servern bzw. durch eine Anhäufung von Services gebildet. In der Regel beinhaltet diese Schicht zumindest einen Datenbank-Server bzw. einen Datenbank-Service. Ein möglicher Architekturaufbau, ist in *Abbildung 3-4* schematisch dargestellt.

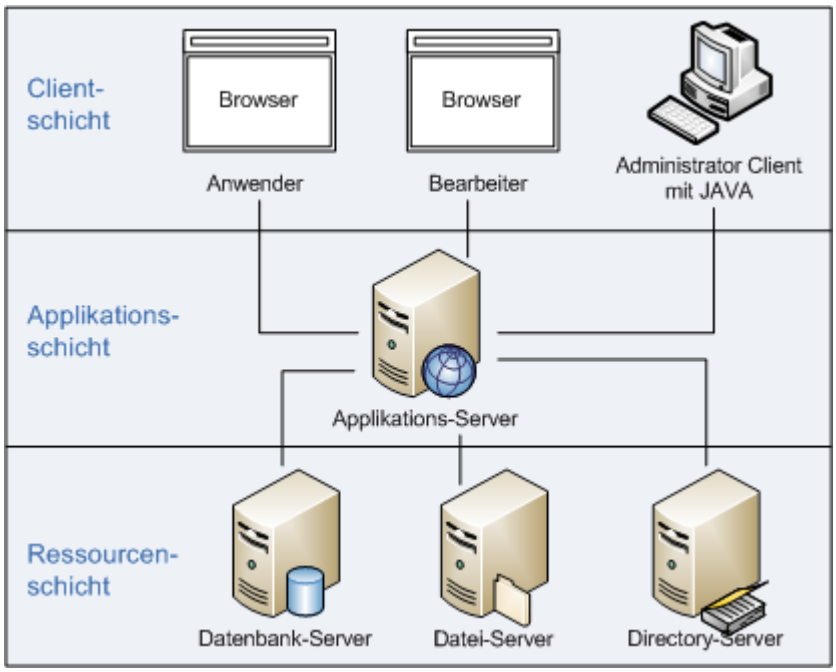


Abbildung 3-4: TOPdesk System Architektur (Quelle nicht öffentlich)

Zu *Abbildung 3-4*, bleibt noch zu erwähnen, dass Datenbank, Datei- und andere Ressourcen nicht zwingend von einem eigenen Server kommen müssen, sondern auch als Service auf ein und derselben Instanz, wie der Applikation-Server laufen können.

### 3.5.2.3 Funktionsumfang

Alle in *TOPdesk*, in der Version 5.6 angebotenen Funktionalitäten, sind in Module gegliedert. *TOPdesk 5.6* kann in den Editionen, *Lite*, *Professional* und *Enterprise* bezogen

werden, wobei alle Module, die wesentliche ITSM-Kernfunktionalitäten bereitstellen, schon in der Professional Edition beinhaltet sind. Folgende Module sind in der Professional Edition von *TOPdesk 5.6* standardmäßig enthalten:

- Incidentmanagement
- Grafisches Configurationmanagement
- SelfServiceDesk
- Problemmanagement
- Vertragsmanagement und SLM
- Operationsmanagement
- Reservierungsmanagement
- Changemanagement
- Vorrat und Bestellungen
- Ereignis- und Aktionsmanagement
- Computer-Telefonie-Integration (CTI)
- Mobile-Interface

Zusätzlich gibt es noch für alle Module die Möglichkeit, Informationen mit Fremdsystemen über die von *TOPdesk* angebotene XML-Schnittstelle, in Echtzeit zu synchronisieren. Alles in Allem, bietet *TOPdesk 5.6* grundsätzlich die benötigten Basis-Funktionalitäten für das ITSM einer IT-Umgebung. Als Kommunikationsplattform zwischen Problemmelder und Support bietet TOPdesk eine Webkonsole für Incidents bzw. Problems. Mittels dieser Webkonsole können Informationen und Dateien ausgetauscht werden. Der Support ist zur Einholung der benötigten Informationen auf den Problemmelder angewiesen.

#### 3.5.2.4 Voraussetzungen und Installation

TOPdesk bietet für die Installation von *TOPdesk 5.6*, Installationspakete für Windows-, Linux- und UNIX-Systeme an. Empfohlen wird aber generell auf ein 64-Bit-System zu setzen, weil die Performance des Systems proportional zum verfügbaren Arbeitsspeicher steigt und fällt. Für die Installation ist die Bereitstellung einer Datenbankinstanz zwingend erforderlich. Bei der Wahl des Datenbank-Servers kann entweder auf Microsoft SQL Server oder auf Oracle gesetzt werden, beide Varianten werden unterstützt. Eine weitere Voraussetzung für die serverseitige Installation, ist das Vorhandensein der vom Hersteller vorgegebenen Java Runtime Environment (JRE) Version.

Auf der Clientseite wird für Anwender- und Bearbeiter-Stationen nur ein vom Hersteller unterstützter Webbrowser benötigt. Für Administrations-Stationen ist zusätzlich die vom Hersteller vorgegebene Java Runtime Environment (JRE) Version, vorab zu installieren. Positiv zu erwähnen ist der Umstand, dass die Erst-Implementierung von *TOPdesk*, in der *Lite*-Edition erfolgen kann und dann in weiterer Folge unkompliziert auf die höheren Versionen aufgestockt werden kann.

#### 3.5.2.5 Skalierbarkeit

*TOPdesk 5.6* bietet mit den 3 Editionen, *Lite*, *Professional* und *Enterprise* für kleine, mittlere und große Betriebe eine passende ITSM-Lösung. Begrenzt werden die Funktionalitäten und die Größenordnung praktisch nur durch das gewählte Lizenzmodel und die eingesetzten Hardware- wie Software-Ressourcen. Da die Kommunikation zwischen Client und Server über einen Webbrowser abgewickelt wird, ist die standortübergreifende Kommunikation theoretisch ohne gröbere Anpassungen möglich.

#### 3.5.2.6 Bewertung hinsichtlich der speziellen Anforderungen

Anhand der in Punkt 3.4 festgelegten Requirements, folgt nun auch für *TOPdesk 5.6* Enterprise die Bewertung, wie gut und wie vollständig, die gestellten speziellen Anforderungen an eine ITSM-Lösung für den Servicebetrieb, grundsätzlich erfüllt werden können.

- **Zu REQ1:** Kann vollständig erfüllt werden, weil *TOPdesk 5.6 Enterprise* zu 100% webbasiert arbeitet. Anwender und Bearbeiter greifen somit via Webbrowser auf das System zu. Ein installierter Webbrowser wird als vorhanden angenommen.

- **Zu REQ2:** Kann im Prinzip erfüllt werden, weil über eine Administrations-Client-Software, alle Templates und Prozesse für den Servicebetrieb angepasst werden können.

- **Zu REQ3:** Kann im Prinzip erfüllt werden, ist jedoch OOTB nicht in dieser Form vorhanden.

- **Zu REQ4:** Kann im Prinzip erfüllt werden, weil *TOPdesk 5.6 Enterprise* mit verschiedenen Datenbanktypen kommunizieren und deren Daten ins System einbinden kann.

- **Zu REQ5:** Kann vom System nicht erfüllt werden, solch eine Komponente wird Out-of-the-Box nicht angeboten. Muss selbst programmiert und über eine Schnittstelle an das System angebunden werden.

- **Zu REQ6:** Kann vom System OOTB nicht erfüllt werden, dies müsste separat programmiert und eingebunden werden.

- **Zu REQ7:** Kann vollständig erfüllt werden, weil *TOPdesk 5.6 Enterprise* so konzipiert wurde, dass die einzelnen Komponenten nicht am selben Standort stehen müssen.

- **Zu REQ8:** Kann vollständig erfüllt werden, weil eine Integration in bestehende IT-Landschaften mit z.B. Microsoft Active Directory, standardmäßig vorgesehen ist.

- **Zu REQ9:** Kann nur zum Teil vom System erfüllt werden, weil dieses Requirement auf **REQ5** aufbaut und dieses nicht OOTB erfüllt werden kann.

- **Zu REQ10:** Kann vollständig erfüllt werden, dies wird durch die vom Hersteller angebotene Enterprise Edition abgedeckt. Jedoch ist eine Client-Software welche eine direkte Anbindung ans System ermöglicht, nicht im Funktionsumfang von *TOPdesk 5.6 Professional* enthalten.

- **Zu REQ11:** Kann vollständig erfüllt werden, weil die Kommunikation zwischen Clients und Applikations-Server, via SSL doppelt gesichert stattfindet.

- **Zu REQ12:** Kann vom System vollständig erfüllt werden.

- **Zu REQ13 bis REQ15:** Diese können eigentlich Alle vom System erfüllt werden, sind jedoch bereits sehr spezialisiert und hängen teilweise auch davon ab, wie das System dimensioniert wird.

Bei der Betrachtung des Erfüllungsgrades der speziellen Anforderungen, durch die beiden evaluierten kommerziellen ITSM-Lösungen, bleibt festzustellen, dass Dieser eigentlich schon ganz gut, aber dennoch ausbaufähig ist. Um den unter *Punkt 3.4* definierten speziellen Anforderungen gerecht zu werden, müssten die beiden untersuchten Produkte durch weitere Produkte bzw. Eigenentwicklungen ergänzt werden. Um eine maßgeschneiderte Lösung zu erhalten bietet sich die Erarbeitung eines eigenen Konzepts an, weil dadurch besser und vor Allem explizit auf die speziellen Anforderungen eingegangen werden kann. Für Unternehmen ab einer gewissen Größenordnung kann dies durchaus Sinn machen. Im nächsten Kapitel wird nun ein solches, auf die speziellen Anforderungen zugeschnittenes, Konzept ausgearbeitet.

# 4 Konzept

## 4.1 Einleitung

In diesem Kapitel soll nun erörtert und bestmöglich geklärt werden, welche Komponenten für eine Lösung für den Servicebetrieb, nach den speziellen Anforderungen benötigt werden bzw. wie diese Komponenten eingesetzt werden müssen. Unter Berücksichtigung der bisherigen Erkenntnisse dieser Diplomarbeit und mit dem Anspruch, die in Kapitel 1 formulierte Zielsetzung zu erfüllen, sollen Bearbeitungszeit von Störungsbehebungen und Einsatz von Arbeitskraft, durch Vermeidung von Abhängigkeiten zwischen Störungsmelder und Störungsbearbeiter schon in der Designphase optimiert werden. Des Weiteren sollen die, auf Grund ihrer Eignung für das Einsatzgebiet, zum Einsatz kommenden Technologien festgelegt werden. Für ein besseres Verständnis der Materie, sollen die eingesetzten Technologien zum Teil noch grundlegend erklärt bzw. eine kurze Einführung in die Arbeitsweise der jeweiligen Technologie gegeben werden. Am Ende dieses Kapitels soll ein ausgearbeitetes Konzept, als Vorlage für die Realisierung einer solchen Lösung, stehen.

## 4.2 Grobkonzept

### 4.2.1 Software-Komponenten

Um den Blick dafür zu schärfen, wie das Konzept grundsätzlich aufgebaut sein muss, müssen zuerst alle, am Gesamtsystem beteiligten Software-Komponenten, betrachtet werden. Die Tatsache, dass das Konzept auch auf inhomogene Systeme angewendet werden können soll, impliziert zum Beispiel schon eine Verwendung offener, unabhängiger Standards. Nachfolgend nun eine Auflistung und Diskussion der Software-Komponenten:

- **Datenspeicher:** Für die Haltung, Bereitstellung und Sicherung der Nutzdaten soll ein RDBMS[12] - Relationales Datenbank Management System - zum Einsatz kommen. Das Datenbanksystem muss auch als Cluster erweiterbar sein. Anbieter solcher Datenbanksysteme bieten die Möglichkeit, Verbindungen von den unterschiedlichsten Softwaresystemen zur Datenbank aufzubauen.

- **Webserver:** Für die Präsentation der Inhalte, Formulare und Steuerfunktionen kommt eine Web-Server-Komponente zum Einsatz. Die Komponente verfügt über die Möglichkeit, SSL-verschlüsselte Kommunikation anzubieten und durchzuführen.

[12] Siehe https://de.wikipedia.org/wiki/Relationale_Datenbank

Diese Komponente kann auch als Teil der Applikations-Server-Komponente realisiert sein.

- **Appliaktionsserver:** Diese Komponente wird für serverseitige Aufgaben, wie Ausführung von Programmcode, Benutzer-Authentifizierung, Bereitstellung und Verarbeitung von Daten, Kommunikation mit anderen Komponenten sowie die Bereitstellung von Diensten eingesetzt.

- **Benutzerschnittstelle:** Die Benutzer-Schnittstelle wird durch einen Webbrowser realisiert. Grundsätzlich können hier praktisch alle gängigen Webbrowser zum Einsatz kommen.

- **Clientsoftware:** Hierbei handelt es sich um die Software-Komponente, welche Kommunikations- und Steueraufgaben auf den zu betreuenden Systemen übernimmt. Diese Komponente hat entgegen seiner Namensgebung jedoch auch Serveraufgaben zu erledigen.

### 4.2.2 Kommunikationsstrecken

Die Tatsache, dass sich das Gesamtsystem über mehrere geografisch verteilte Lokationen erstrecken können soll, schränkt die Auswahl der möglichen Kommunikations-Technologien ein. Die Kommunikation zwischen den einzelnen, am System beteiligten Komponenten erfolgt also per Definition, teilweise über WAN-Strecken und somit auch über Firewall-Grenzen hinweg. Wie bereits unter *Punkt 3.2.1* erörtert, bietet sich für die Kommunikation zwischen den einzelnen Systemen bzw. Komponenten demnach das HTTP-Protokoll bzw. das HTTP-Protokoll in Verbindung mit SSL- oder TLS-Erweiterung zu HTTPS[13] an. HTTPS bietet zusätzlichen Schutz für die zu übertragenden Daten.

### 4.2.3 Architekturmodell

Bei der Auswahl eines, für die zu erfüllenden Aufgaben, passenden Architektur-Modells, fiel die Entscheidung zu Gunsten einer sogenannten „Service Orientierten Architektur“. Abgekürzt mit SOA wird nicht etwa eine eigenständige neue Technologie beschrieben, sondern viel eher ein Konzept zur Vernetzung von verteilten IT-Services. Der Kernnutzen einer sogenannten SOA liegt darin, unterschiedliche Systeme lose untereinander zu koppeln und dadurch ein flexibles und leicht zu wartendes System zu erzeugen. Unter einer SOA versteht man also eine Systemarchitektur, die unterschiedliche und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht. [Fin09] [Bur07]

[13] Siehe https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure

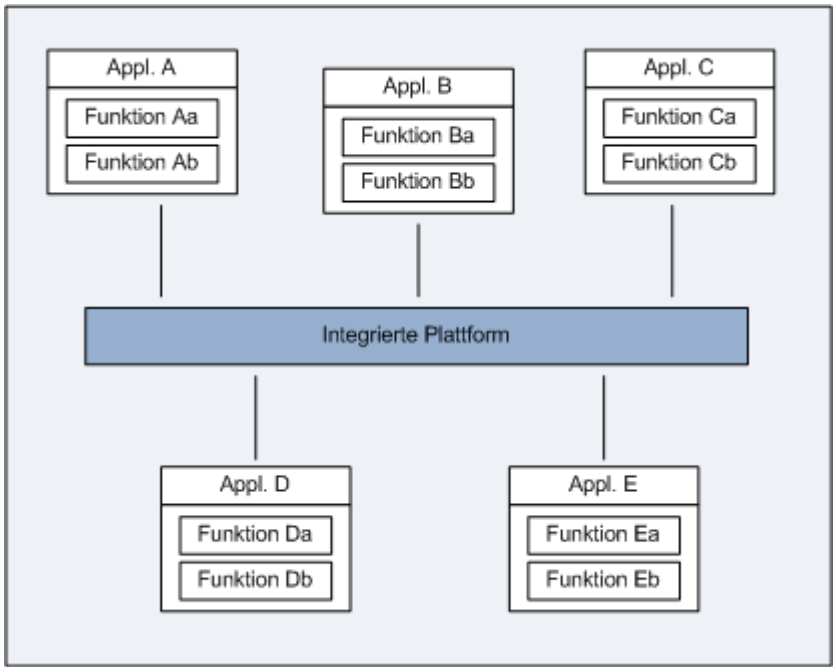


Abbildung 4-1: Schema einer SOA ( [Bur07] , Seite 7)

Im, in der *Abbildung 4-1* dargestellten Schema einer SOA, sind die einzelnen, an der Architektur beteiligten Komponenten als Applikationen mit deren Funktionen dargestellt. Die integrierte Plattform dient als Kommunikations- und Servicebus für die Komponenten. Somit kann jede Komponente auf die gebotenen Funktionen der anderen Komponenten zugreifen, ohne direkt mit der entsprechenden Komponente kommunizieren zu müssen.
Daraus ergeben sich die folgenden Vorteile:

- **Flexibilität:** Dadurch, dass keine direkte Abhängigkeit der einzelnen Komponenten besteht, bleibt die Architektur stets flexibel und Änderungen können ohne großen Aufwand durchgeführt werden.

- **Effizient:** Durch mehrfache Verwendung ein und desselben Services, werden redundante Implementierungen vermieden. Im Idealfall wird eine Funktion nur ein einziges Mal implementiert.

- **Skalierbar:** Als Folge der eingesetzten integrierten Plattform, welche neue Komponenten bei Bedarf flexibel einbinden kann, bleibt die gesamte Architektur skalierbar.

- **Offen:** Eine SOA basiert auf keiner festgelegten Technologie, jedoch passen viele moderne Middleware-Systeme[14] in das Konzept und erfüllen die Anforderungen

[14] Middleware, bezeichnet in der Informatik anwendungsneutrale Programme, die so zwischen Anwendungen vermitteln, dass die Komplexität dieser Applikationen und ihre Infrastruktur verborgen werden. (siehe https://de.wikipedia.org/wiki/Middleware)

einer SOA. Vor allem XML-basierte Technologien bieten ein hohes Potenzial, SOA erfolgreich umzusetzen.

Eine geeignete Technologie für die Umsetzung einer SOA liefern WebServices. Was sich hinter dem Begriff „WebServices“ verbirgt und warum Diese das Mittel der Wahl für die Konzepterstellung sind, wird im folgenden Punkt, 4.2.4, gesondert geklärt.

## 4.2.4 WebServices

### 4.2.4.1 Grundlagen

Was WebServices grundlegend auszeichnet, ist die Tatsache, dass WebServices einen relativ simplen Weg darstellen, vorhandene Daten und Funktionen aus Applikationen zur Verfügung zu stellen bzw. von anderen Anbietern zu konsumieren. Ein WebService wird nicht direkt von einem Nutzer konsumiert oder gesteuert, sondern ist als Kommunikation zwischen Computersystemen zu sehen. Das bedeutet, dass ein WebService nach der Implementierung, in der Regel automatisch seinen Dienst verrichtet. Ein Anwender, welcher vor einer Arbeitsstation sitzt, bemerkt also nicht, dass er im Hintergrund die Dienste eines WebServices konsumiert. Diese Tatsache spiegelt wiederum das Ideal einer SOA wieder.

Auf Grund der Tatsache, dass WebServices auf offenen Standards basieren, bestehen keine Abhängigkeiten zu bestimmten Plattformen. Durch WebServices werden vorhandene Informationen aus bestehenden Systemen webfähig aufbereitet. Durch das Anbieten einer standardisierten Schnittstelle, können unterschiedliche Geräteklassen WebServices implementieren und plattformübergreifend agieren. [Fin09]

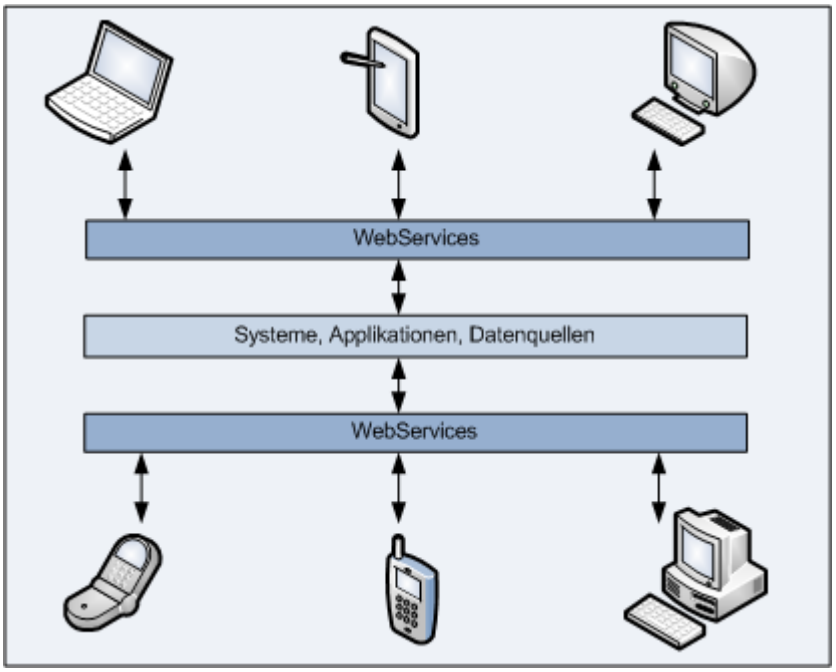


Abbildung 4-2: Nutzung von WebServices ( [Bur07] , Seite 39)

*Abbildung 4-2* zeigt die Anbindung unterschiedlichster Systeme an einen Ressourcenpool, unter Zuhilfenahme der WebService Technologie.

### 4.2.4.2 Grundkonzept von WebServices

Technisch betrachtet, besteht eine WebService-Infrastruktur aus drei Bestandteilen. In der *Abbildung 4-3*, sind Diese und die Interaktionen zwischen Diesen, grafisch dargestellt.

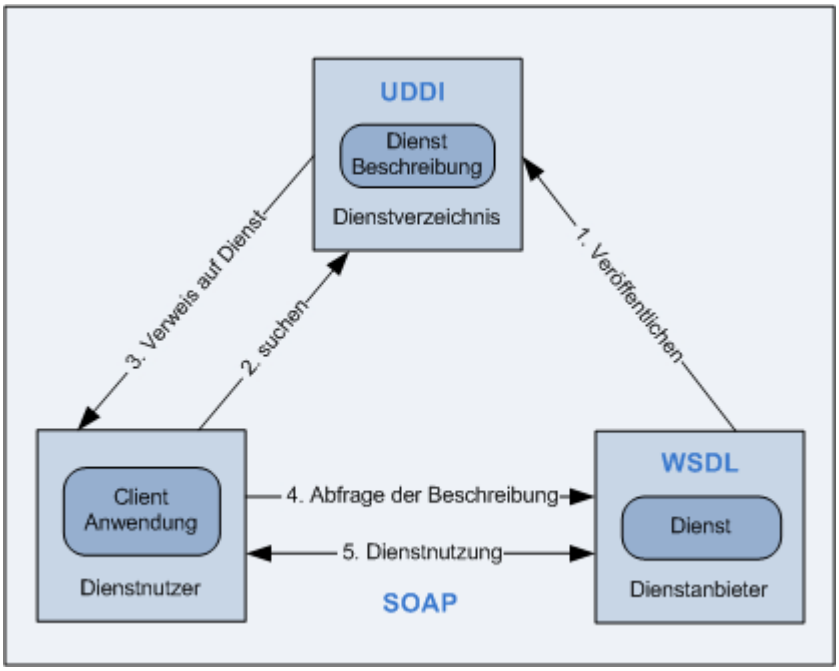


Abbildung 4-3: Funktionsweise einer WebService-Infrastruktur ( [Bur07] , Seite 41)

Das objektorientierte Kommunikationsprotokoll SOAP, ermöglicht die Kommunikation zwischen heterogenen Diensten, unter interner Nutzung von HTTP und mit Kodierung der Parameter in XML. Die Schnittstellen der Dienste werden mit der Beschreibungsnotation *Web Services Description Language* (WSDL) spezifiziert, und der Bindevorgang wird global über den Verzeichnisdienst *Universal Description, Discovery and Integration* (UDDI) realisiert. [Sch07]

### 4.2.4.3 WSDL

Die *Web Service Description Language* (WSDL) dient der Beschreibung der Schnittstellen von Diensten, welche die Basis für entfernte Aufrufe darstellen. Dies erfolgt bei WSDL in zwei Teilen. In einem abstrakten Teil werden die verwendeten Interaktionsmuster, die ausgetauschten Nachrichten und die dafür notwendigen Datentypen, unabhängig von Kommunikationsmechanismen und Transportprotokollen definiert. Ergebnis der Definition sind Schnittstellen, die verschiedene abstrakte Methoden enthalten. In einem konkreten Teil, werden diese Methoden dann an konkrete Transportmechanismen und Nachrichtenformate gebunden. Eine WSDL-Datei liefert also eine Beschreibung über Dienste (Methoden), Ein-

und Ausgabeparameter, Informationen über das verwendete Transportprotokoll und Informationen wo der Service zu finden ist. Struktur und Syntax von WSDL basieren auf einem XML-Schema. [Sch07]

#### 4.2.4.4 SOAP

SOAP ist ein Protokoll zum Austausch von Nachrichten und Dokumenten auf der Basis einer XML-basierten Kodierung. Es legt die einheitliche Kodierung der Aufrufe und ihrer Parameter zur Übertragungszeit fest. Neben den vordefinierten Kodierregeln, die für die Mehrzahl der Anwendungen ausreichen, können erweiterte Kodierregeln für spezielle Datentypen, beispielsweise von Herstellern von Software-Werkzeugen, festgelegt werden. Für SOAP wurde dabei kein Interaktionsschema fest vorgegeben, etwa Request/Response, sondern es können beliebige Dokumente und damit auch Dienstaufrufe und Aufrufresultate ausgetauscht werden. Dabei können verschiedene Interaktionsschemen festgelegt werden. SOAP ist nicht an ein bestimmtes Transportprotokoll gebunden, sondern kann in beliebige Protokolle eingebettet werden. Damit entsteht die Möglichkeit der Einbettung von SOAP in HTTP, die auch häufig genutzt wird. Durch die Verwendung von HTTP ist auch eine Kommunikation über Firewalls hinweg möglich, auch wenn Diese nur Web-basierte HTTP-Kommunikation zulassen (Port 80). Natürlich muss dann trotzdem eine zusätzliche Sicherheitsüberprüfung bei der Annahme und Ausführung von Aufrufen erfolgen, etwa durch den Einsatz von Filtern auf Applikationsebene. [Sch07]

#### 4.2.4.5 Aufbau von SOAP-Nachrichten

Eine SOAP-Nachricht besteht zumindest aus einem Envelope-Element, welches die Hülle der SOAP-Nachricht bildet und einem zugehörigen Body-Element. Ein Header-Element kann optional noch vor dem Body-Element stehen. In Diesem können Meta-Informationen untergebracht werden. Im Body-Element sind die eigentlichen Nutzdaten untergebracht. Innerhalb des Body-Elements können sowohl Informationen zum Datenaustausch, als auch Anweisungen für einen entfernten Prozeduraufruf stehen (siehe *Abbildung 4-4*). [Ros06]

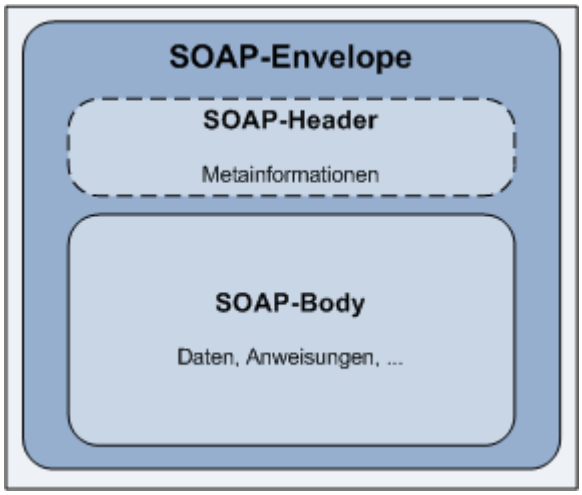


Abbildung 4-4: SOAP-Struktur ( [Sch07] , Seite 73)

#### 4.2.4.6 UDDI

UDDI ist ein weltweiter Verzeichnisdienst für Web Services, der über eine Web-Service-Schnittstelle sowie eine global bekannte Adresse (www.uddi.org) angesprochen werden kann. UDDI kann also mit den „Gelben Seiten“ verglichen werden, weil es quasi eine Suche nach einem Dienst erlaubt, ohne Informationen über den Dienstanbieter zu kennen. Auf Grund der Tatsache, dass das in dieser Diplomarbeit angepeilte Konzept auf unternehmensinterne Kommunikation abzielt und die einzelnen Dienste daher bekannt sind, wird UDDI nicht tiefer erläutert.

#### 4.2.4.7 Ablauf der WebService-Nutzung

Wenn ein, auf einem Client laufendes Programm, einen Dienst (Methode) eines Webservices in Anspruch nehmen will, muss der Methodenaufruf zunächst an ein sogenanntes Stub-Objekt übergeben werden. Das Stub-Objekt übersetzt die gelieferten Daten aus der ursprünglichen Programmiersprache in das XML-Format. Die Struktur der XML-Daten muss dem XML-Schema entsprechen, welches durch die vom WebService erzeugte WSDL-Datei vorgegeben wird. Anschließend wird aus den XML-Daten eine Nachricht im SOAP-Format generiert. Durch diese Vorgehensweise wird der Aufruf einer Methode mit zugehöriger Ergebnisrückgabe ermöglicht. Das Stub-Objekt leitet also eine SOAP-Anfrage an den Server der den WebService anbietet. Auf Seiten des Servers wird die SOAP-Nachricht von einem sogenannten Skeleton-Objekt entgegengenommen. Das Skeleton-Objekt übersetzt die SOAP-Nachricht in einen für den Server ausführbaren Programmcode und veranlasst somit den entsprechenden Methodenaufruf, welcher vom Client gefordert wird. Der Rückgabewert der Methode wird dann wiederum vom Skeleton-Objekt in eine SOAP-Nachricht verpackt und als SOAP-Antwort an das Stub-Objekt übermittelt. Dieses übersetzt die SOAP-Antwort und gibt Diese an das Client-Programm zurück. [Ken15]

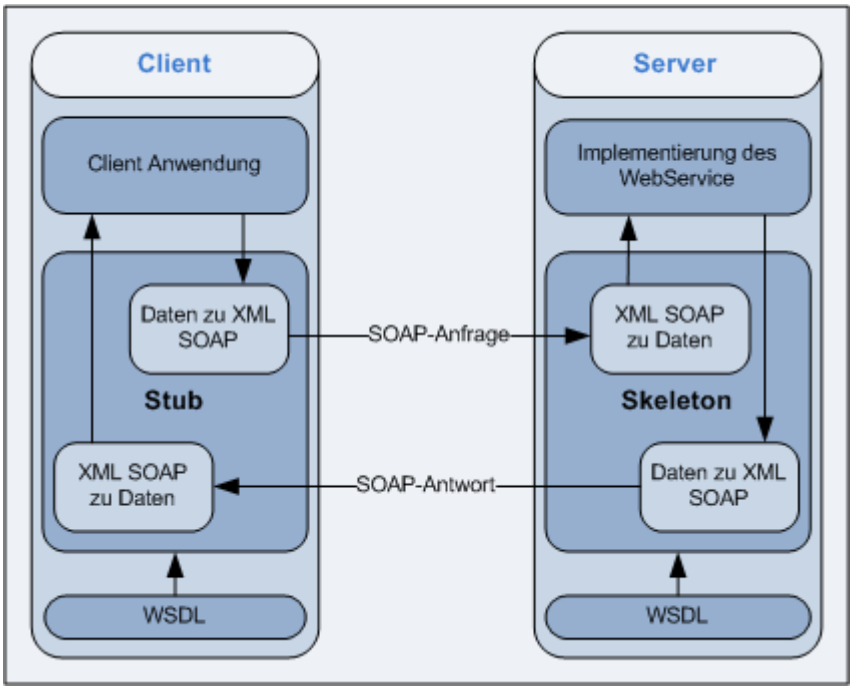


Abbildung 4-5: WebService Dienstnutzung ( [Ken15] )

Die, in *Abbildung 4-5* dargestellten, Stub- und Skeleton-Objekte, sind in der Regel automatisch generierte Proxy-Klassen, welche auf Basis der WSDL-Datei erzeugt werden. Die Benennung Proxy-Klasse kommt daher, dass solch eine Klasse die Methoden für die aufrufenden Anwendungen so darstellt, als wären diese lokal vorhanden. Eine Proxy-Klasse wird, wie schon gesagt, in der Regel automatisch generiert. Diese Aufgabe kann, unter Zuhilfenahme von diversen Tools, durchgeführt werden.

### 4.2.5 Workflow für den Servicebetrieb

Eines der zentralen Ziele dieser Diplomarbeit, ist die Modellierung einer ITSM-Lösung, welche den Bearbeitungsprozess von Störungsbehebungen und den damit einhergehenden Einsatz von Arbeitskraft optimieren könnte. Oftmals lohnt es sich, auf der Suche nach möglichen Optimierungen, unkonventionell erscheinende Wege zu gehen, um zu einer Lösung zu gelangen. Dazu soll nun der „übliche“ Workflow für den ITSM-Servicebetrieb erweitert bzw. ergänzt werden. Wie das grobe Konzept für den geplanten Workflow nun tatsächlich aussehen wird, wird nachfolgend skizziert und grob beschrieben. Dazu wird der gesamte Vorgang in Funktionsblöcke (siehe *Abbildung 4-6*) zerlegt.

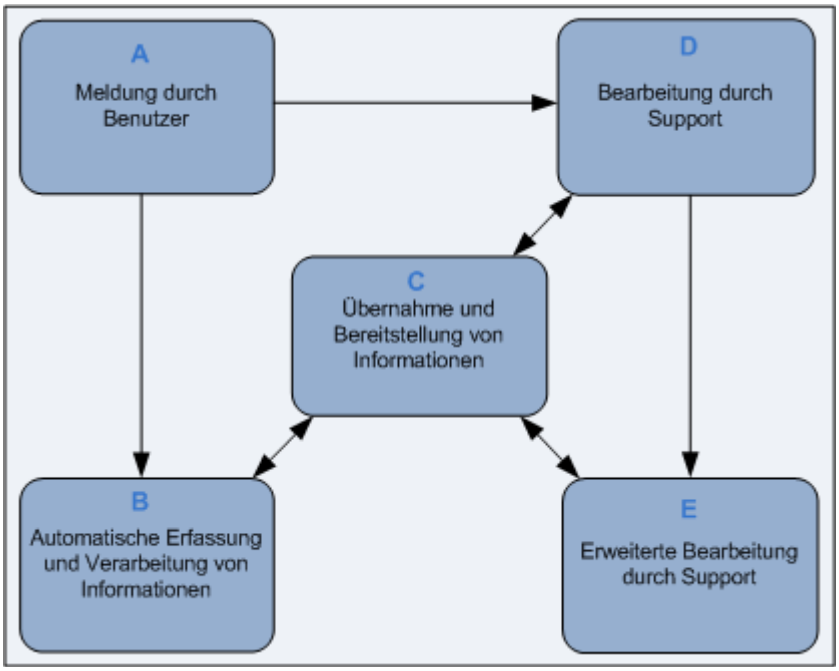


Abbildung 4-6: Workflow Blöcke

In *Abbildung 4-6* ist eine grobe Unterteilung des Workflows dargestellt. Die Beschreibung der Blöcke lautet wie folgt:

- **Block A:** Dieser Block steht für alle Benutzeraktionen, die für die Erstellung einer Anfrage oder Problemmeldung notwendig sind. Auch sind in diesem Block alle systemseitigen Schritte, welche dem Benutzer die Durchführung des Anliegens ermöglichen, angesiedelt.

- **Block B:** In diesem Block läuft die proaktive Datenerfassung. Im Zuge des Erstellungsprozesses einer Problemmeldung, wird automatisch die proaktive Datenerfassung angestoßen, welche nützliche Informationen über Zielsystem, Zielbenutzer und mögliche Problemursachen liefert. Realisiert wird dieser Vorgang unter Zuhilfenahme der WebService-Technologie.

- **Block C:** Dieser Block steht für die Übernahme und Bereitstellung der von den Zielsystemen erfassten Nutzdaten. Alle Informationen die zur Problembearbeitung benötigt werden, werden hier zentral in einer Datenbank gehalten.

- **Block D:** Einfache Aufgaben, wie die schriftliche Kommunikation zwischen Anwender und Support, werden in diesem Block abgewickelt. In diesem Block wird auch über Weiterbearbeitung oder Eskalation an eine technisch tiefere Ebene entschieden.

- **Block E:** Die erweiterte Bearbeitung umfasst alle Schritte und Tätigkeiten des Supports, welche ein aktives Eingreifen am Zielsystem bedeuten. Dazu gehört zum Beispiel das Anstoßen von Steuervorgängen.

# 4.3 Feinkonzept

## 4.3.1 Aufbau Gesamtsystem

An dieser Stelle ist es nun soweit, das System als Gesamtes und das Zusammenspiel der beteiligten Komponenten zu betrachten und zu beschreiben. Das erklärte Ziel dieser Diplomarbeit ist es, eine ITSM-Lösung für den Servicebetrieb zu konzeptionieren. Die Lösung soll sich dadurch auszeichnen, dass die beteiligten Komponenten relativ unabhängig voneinander, über mehrere, geografisch verteilte Standorte, angesiedelt werden können. Des Weiteren soll die Lösung flexibel und vor Allem auch skalierbar sein, also beliebig erweiterbar. Auch der Workflow und Informationsfluss zwischen Anwender und Support soll optimiert werden, bzw. der Zugriff auf erweiterte Informationen für den Support unkompliziert möglich sein. Ein weiterer wichtiger Punkt, der gefordert ist, ist eine größtmögliche Plattformunabhängigkeit, welcher die Verwendung offener Standards voraus setzt. Diese geforderten Eckdaten sowie die speziellen Anforderungen, angegeben unter *Punkt 3.4.1*, bilden nun die Basis, für das Feinkonzept.

Das Konzept sieht also eine SOA vor, in welcher WebServices die Exekutive darstellen. Die zu servicierenden Zielsysteme können sich sowohl am selben Standort wie der Support, als auch unabhängig davon an einem entfernten Standort befinden. Die, das System und die Nutzdaten, für die Nutzer bereitstellenden Service-Komponenten, können natürlich auch in entfernten Standorten beheimatet sein. Ermöglicht wird das dadurch, dass die Kommunikation ausschließlich über HTTP inklusive Sicherheitserweiterungen und in HTTP eingebettete Protokolle wie SOAP stattfindet. Als interaktives Zugriffsmittel auf das System wird für Anwender und Support nur ein Webbrowser benötigt. Die programmatischen und automatisierten Inter-Standort-Zugriffe und Abfragen erfolgen systemseitig via SOAP-Nachrichten. Somit ist das System nicht auf spezielle Firewall-Freischaltungen zwischen den Kommunikationspartnern angewiesen. Die mögliche, ausfallsichere Gestaltung sowie die nahezu uneingeschränkte Skalierbarkeit des Systems, werden durch eine logische Kapselung, der am System beteiligten Server-Komponenten erreicht. Diese Komponenten bilden in ihrer Gesamtheit eine logische Service-Komponente (siehe *Abbildung 4-7*). Kommt mehr als eine Service-Komponente im Gesamtsystem zum Einsatz, gleichen die Service-Komponenten beständig alle ihre Daten, quasi als Multi-Master-System, untereinander ab. Der Abgleich erfolgt dann, wie die Kommunikation zwischen Zielsystem und Service-Komponente, über einen WebService. Um nicht unkontrolliert dauerhaft Daten zwischen den Service-Komponenten zu übermitteln, können Intervalle für den Abgleich festgelegt werden.

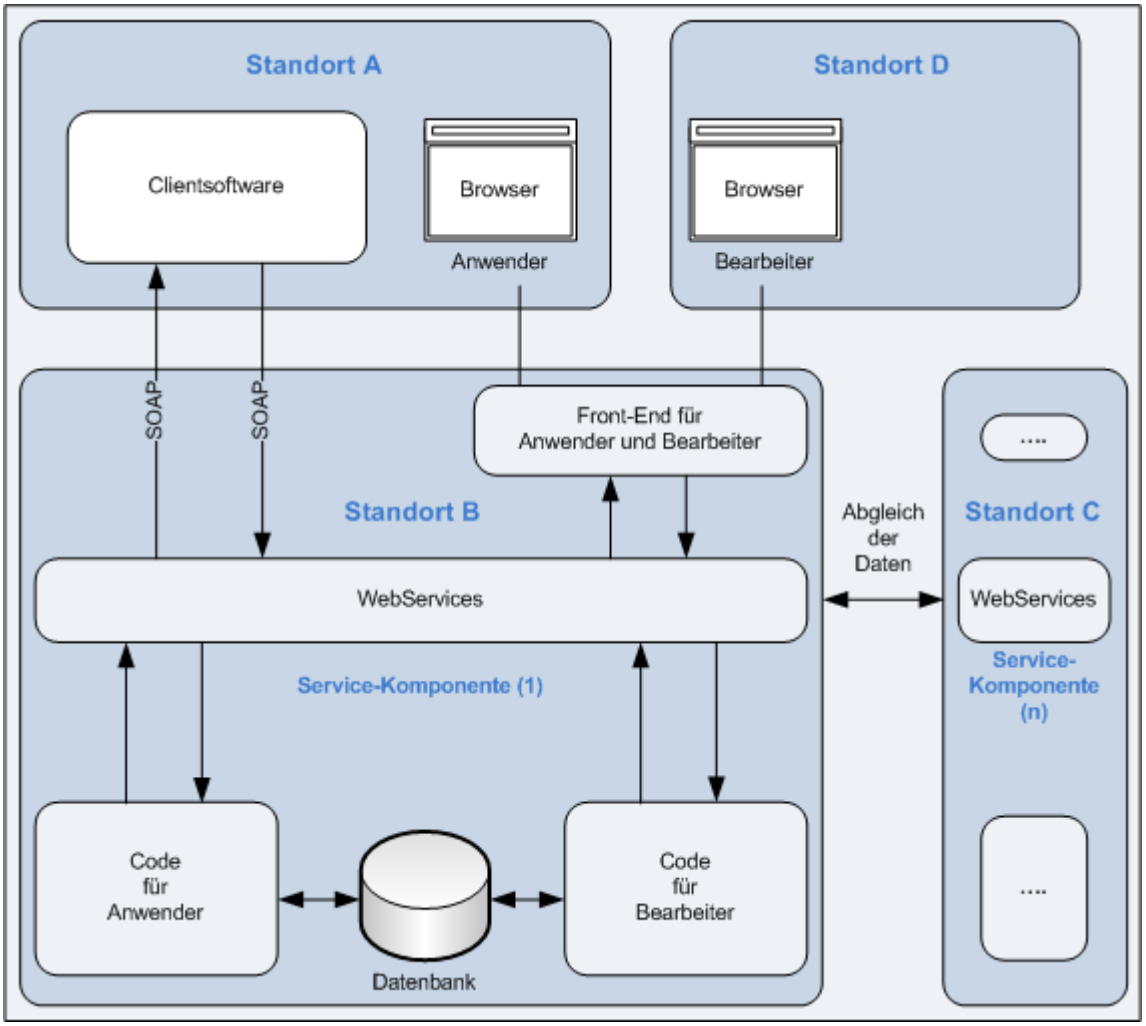


Abbildung 4-7: Schema Gesamtsystem

In *Abbildung 4-7*, ist ein möglicher Aufbau des Gesamtsystems exemplarisch dargestellt. Die standortübergreifende Kommunikation und das Zusammenspiel der Komponenten sind hier schon angedeutet. Die Komponente *Clientsoftware* steht für eine Anwendung, welche auf jedem Zielsystem vorhanden ist und mit der *WebServices*-Komponente SOAP-Nachrichten austauscht. Die *WebServices*-Komponente stellt die unterschiedlichen Methoden zur Verfügung. Das *Front-End* steht für die GUI-Darstellung der Inhalte für Anwender und Bearbeiter im Webbrowser. Die Kacheln, *Code für Anwender* und *Code für Bearbeiter*, stehen für die Methoden und Klassen, welche keine sogenannten „Webmethoden“ sind. Theoretisch könnte also bei diesem beispielhaften Aufbau, Standort B zeitweise komplett ausfallen und die Dienste würden für die Standorte A und D weiterhin verfügbar sein, weil diese von Standort C, in gleichem Maße angeboten werden wie von Standort A. Ermöglicht wird dies dadurch, dass die Clientsoftware über eine Liste der verfügbaren Systeme und IP-Adressen verfügt, welche durch einen WebService aktuell gehalten wird. Ist ein System nicht verfügbar, wird das Nächste in der Liste ausgewählt. Kommt ein neues System hinzu, wird die lokale Liste aktualisiert. Im Folgenden werden nun, die am System beteiligten Komponenten, deren Aufgaben und die Kommunikationswege untereinander, genau beschrieben.

## 4.3.2 Clientsoftware

Die Clientsoftware ist eine Applikation, die auf jedem zu betreuenden Zielsystem vorhanden ist und für die Sammlung und Übermittlung von Daten sowie zur Übernahme und Ausführung von Steuerbefehlen zuständig ist. Da in heterogenen Systemumgebungen unterschiedliche Clients eingesetzt werden, muss diese Software für mehrere Plattformen entwickelt werden. Jedoch können Diese dann auf die gleichen WebServices zugreifen, somit hält sich der Entwicklungsaufwand serverseitig in Grenzen. Wie schon unter *Punkt 4.2.4.6* beschrieben, wird die Kommunikation zwischen den unterschiedlichen Plattformen und dem gewünschten WebService durch die Generierung und Einbindung einer Proxy-Klasse ermöglicht. Die Proxy Klasse übernimmt die Kommunikation mit dem WebService. Je nachdem welche Aufgaben die Clientsoftware am Zielsystem ausführen soll, muss auch ein passendes Dienstkonto gewählt werden, unter welchem diese ausgeführt wird. Grundsätzlich wird in diesem Konzept aber von einem Dienstkonto mit administrativen Berechtigungen für das Zielsystem ausgegangen. Um die Vorgänge im Folgenden anschaulicher zu machen, werden auch schon fiktive Methodenbezeichner wie z.B. „*Get_System*“ verwendet.

### 4.3.2.1 Registrierung des Zielsystems

Stets die erste Tätigkeit nach dem Applikationsstart, ist die Registrierung des Zielsystems mitsamt dem angemeldeten Benutzer am Gesamtsystem. Beziehungsweise, wenn es sich bei dem Zielsystem um einen Terminal-Server handelt, auf dem mehrere Benutzer gleichzeitig arbeiten, werden alle angemeldeten Benutzer erfasst und gesendet. Die Registrierung erfolgt über einen WebService, welcher die Daten erfasst und zur Speicherung in der Datenbank weitergibt (siehe *Abbildung 4-8*).

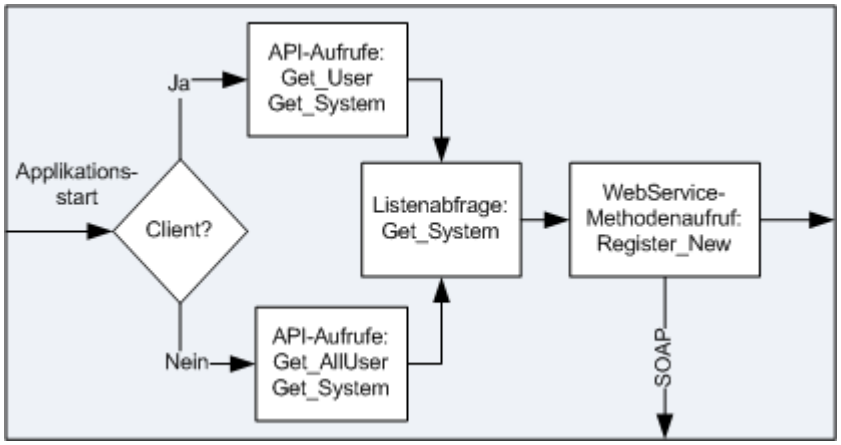


Abbildung 4-8: Registrierung des Zielsystems

Somit sind die Informationen, welche Benutzer und Zielsysteme momentan aktiv sind und vor Allem welche Benutzer am welchen Systemen aktiv sind, in der Datenbank verfüg- und

abrufbar. Bei Beendung der Applikation wird wiederum ein WebService aufgerufen, welcher die zuvor registrierten Informationen wieder aus der Datenbank entfernt (siehe *Abbildung 4-9*). Somit wird der Umfang der zu speichernden Informationen gering gehalten.

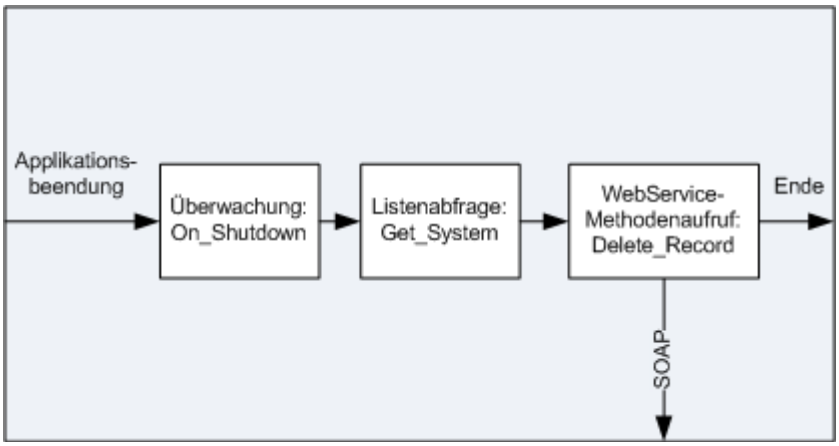


Abbildung 4-9: Löschung der Datenbankeinträge

### 4.3.2.2 Periodische Abfragen

Eine wesentliche Aufgabe der Clientsoftware, ist die Annahme von Steuerinformationen sowie die Ausführung selbiger. Die Informationen über Konfigurationsänderungen und die Anweisungen, was lokal am Zielsystem von der Clientsoftware durchzuführen ist, werden von der Service-Komponente geliefert. Diese Anweisungen werden aber nicht via Push-Benachrichtigung an die einzelnen Zielsysteme übermittelt, sondern von den Zielsystemen im Poll-Verfahren angefordert. Das bedeutet, dass die Zielsysteme in bestimmten Zeitabständen eine WebService-Methode aufrufen, welche dann wiederum anstehende Konfigurationsänderungen oder auch Anweisungen an das Zielsystem zurückliefert. Dieser Vorgang ermöglicht somit in weiterer Folge die proaktive Datenerfassung, welche im Zuge einer Problemmeldung durch einen Benutzer, im Hintergrund von der ITSM-Lösung angestoßen wird. Der Zeitabstand für die periodischen Abfragen muss also so gewählt werden, dass die Deltazeit zwischen dem Eröffnen einer Problemmeldung und dem Eintreffen, der dann in Folge vom Zielsystem gesendeten Informationen, in der Datenbank nicht zu groß wird. Vor Allem ist auch die Ausführung von Steueraktionen am Zielsystem davon abhängig, wie schnell die Kommandos vom Zielsystem entgegengenommen werden und somit auch ausgeführt werden können. Es bietet sich also eine Aufteilung der Abfragen in einen kurzen Zyklus, in welchem akute Kommandos entgegen genommen werden und einen längeren Zyklus, in dem Konfigurationsänderungen abgefragt werden. Der kurze Zyklus wird sich also im Sekundenbereich, der längere Zyklus im Minuten- bzw. Stundenbereich abspielen. Eine anschauliche Darstellung dieser Vorgänge wird in *Abbildung 4-10* gezeigt.

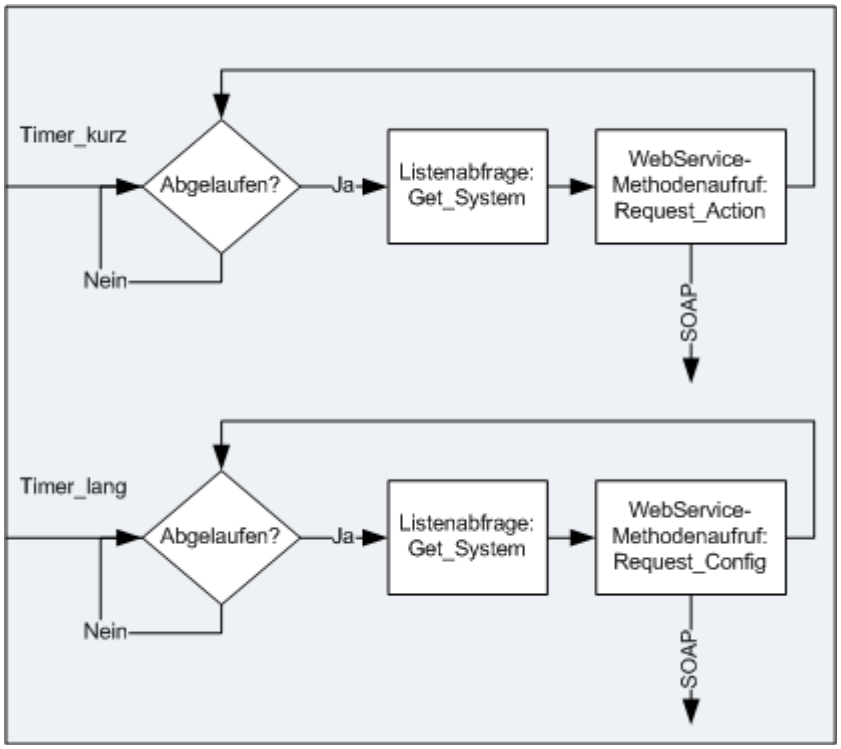


Abbildung 4-10: Periodische Abfragen

#### 4.3.2.3 Ausführung von lokalen Aktionen

Die Clientsoftware läuft zwar im Prinzip komplett eigenständig, wird jedoch von der Service-Komponente ferngesteuert. Die Ausführung von bestimmten Aktionen auf dem lokalen Zielsystem erfordert, je nach Art der Aktion, bestimmte Berechtigungen. Für z.B. Schreibzugriffe oder steuernde Eingriffe, wie das Stoppen und Starten von Prozessen sind administrative Berechtigungen am Zielsystem notwendig. Der Vorteil bei dieser Variante der Fernsteuerung ist, dass die Service-Komponente nicht direkt die Aktionen am Zielsystem auslöst und somit auch nicht über entsprechende Berechtigungen verfügen muss. Dieser Teil wird lokal am Zielsystem, quasi eigenständig ausgeführt. Als Beispiel für das Ausführen einer lokalen Aktion, soll nun das Einlesen und Übermitteln einer Protokolldatei, einer beliebigen lokalen Anwendung dienen. Der Vorgang wird manuell von einem Supportmitarbeiter angestoßen. Über die Service-Komponente wird dann die Protokolldatei vom Zielsystem angefordert. Und zwar indirekt über den zyklischen WebService-Methodenaufruf „Request_Action“, welcher vom Zielsystem ausgeführt wird. Das Zielsystem liest die Protokolldatei lokal ein und sendet die Informationen mit dem WebService-Methodenaufruf „Send_Log“ an die Service-Komponente. Die Service-Komponente legt die Daten in der Datenbank ab und der Supportmitarbeiter kann anschließend auf die geforderten Informationen zugreifen (siehe *Abbildung 4-11*).

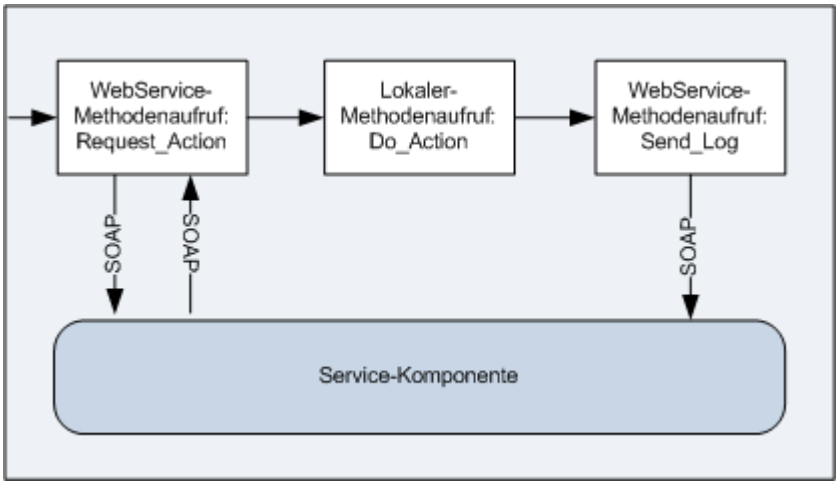


Abbildung 4-11: Ausführung lokale Aktion

### 4.3.3 Front-End

Das Front-End der ITSM-Lösung wird durch die beiden browserbasierenden Portale für Anwender und Bearbeiter gebildet. Für Anwender bietet sich hier nach erfolgreicher Anmeldung die Möglichkeit, einen der angebotenen Servicebetrieb-Workflows zu starten. Hierzu zählt z.B. der Workflow zur Eröffnung eines Incidents(siehe *Punkt 2.4.6.2*). Ist ein neuer Incident vollständig erfasst und wird durch den Eröffner gespeichert und gesendet, wird auf Grund der Angaben im Hintergrund vom System, die proaktive Datenerfassung gestartet. Welche Daten von der proaktiven Datenerfassung eingesammelt werden, hängt wie schon erwähnt, von den Angaben des Incident-Eröffners ab. Wird also z.B. angegeben, dass ein Fehler in Applikation X aufgetreten ist, wird die entsprechend vorhandene Protokolldatei, eingesammelt (siehe *Abbildung 4-12*).

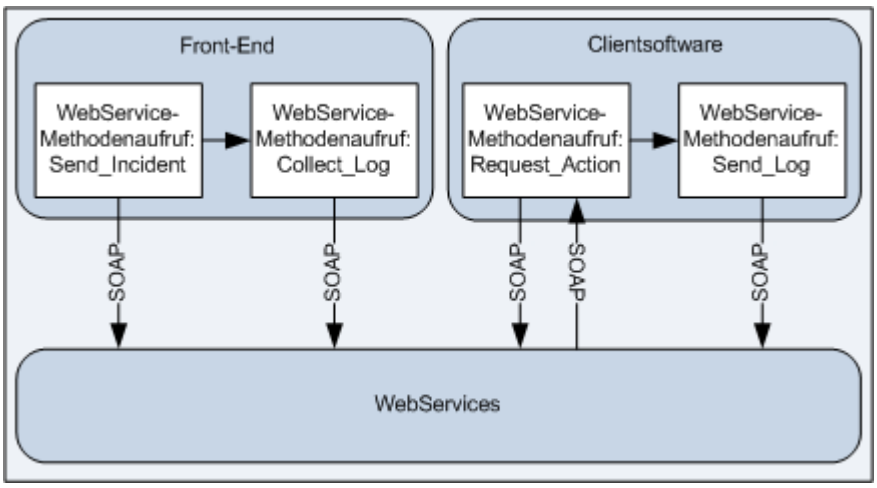


Abbildung 4-12: Incident-Eröffnung

Für Bearbeiter oder auch Supportmitarbeiter öffnet sich nach erfolgter Anmeldung ein Management-Portal mit Zugriff auf sämtliche Problemmeldungen inklusive der Möglichkeit, Daten zur erweiterten Problemanalyse abzufragen und Steueraktionen (siehe *Punkt 4.3.2.3*) auszuführen (siehe *Abbildung 4-13*).

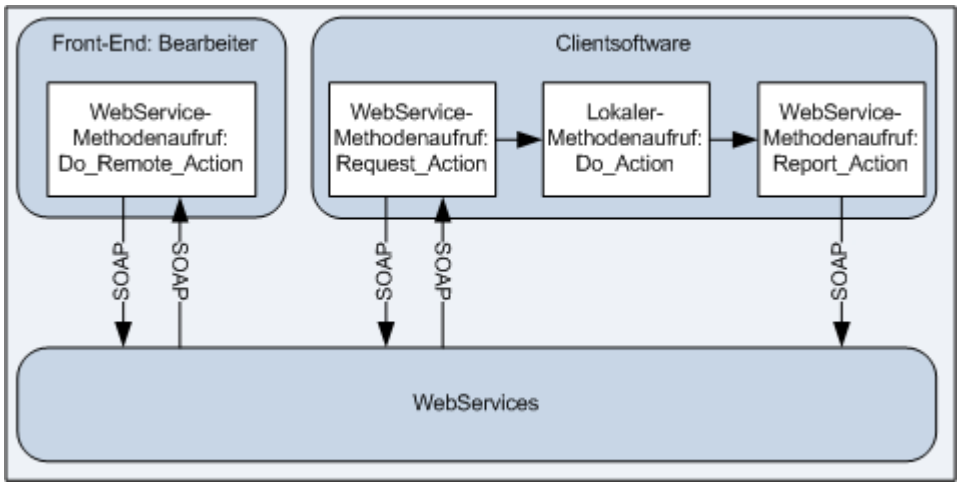


Abbildung 4-13: Steueraktion durch Bearbeiter

Wird via Front-End ein Workflow gestartet oder eine Abfrage durchgeführt, wird im Hintergrund auf eine entsprechende WebService-Methode zugegriffen, somit kann das Front-End unabhängig von der eigentlichen Service-Komponente gewartet und getauscht werden, bzw. auf einer anderen Systemplattform laufen, welche ansonsten mit der restlichen Service-Komponente nicht kompatibel wäre. So könnte zum Beispiel aus Sicherheitsgründen ein LINUX-Webserver als Front-End eingesetzt werden, welcher im Hintergrund die WebService-Methoden eines Microsoft Internet Information Servers aufruft. Wesentlich ist aber die Erhaltung größtmöglicher Flexibilität durch diesen Architekturaufbau.

### 4.3.4 Code für Anwender und Bearbeiter

Die Bereiche *Code für Anwender* und *Code für Bearbeiter* stehen, wie schon unter Punkt 4.3.1 erwähnt, für sämtliche Methoden und Klassen, welche nicht als Webmethoden definiert sind. Dazu zählen Methoden, die von den WebServices im Hintergrund aufgerufen werden. Diese Basismethoden übernehmen Aufgaben wie Schreib- und Lesezugriffe auf die Datenbank und die Anbindung an Drittsysteme wie z.B. die Anbindung an Microsoft Active Directory. Die eigentliche, programmiertechnische Umsetzung der Funktionen, die von der ITSM-Lösung geboten werden erfolgt hier, weil die WebServices selbst quasi nur immer granulare Einzelfunktionen enthalten und nicht selbst komplexe Vorgänge ausführen. Um die beiden Plattformen für Anwender und Bearbeiter voneinander unabhängig zu halten, werden auch die Codebereiche getrennt.

## 4.3.5 WebServices

Dreh- und Angelpunkt, praktisch einer jeden systemseitig durchgeführten Aktion, sind die in die ITSM-Lösung integrierten WebServices. Ein WebService sollte in der Regel so schmal wie möglich gehalten werden und seinerseits nicht auf die Funktionalitäten eines anderen WebServices angewiesen sein, um die eigenen Funktionalitäten bereitstellen zu können. So erfolgt auch in der Konzeptionierung für die ITSM-Lösung eine granulare Aufteilung der Aufgaben auf mehrere WebServices. Bei der nachfolgenden Auflistung der benötigten WebServices samt Beschreibung ihrer Funktionalität, erfolgt noch eine Unterteilung in WebService für die Clientsoftware, WebServices für das Anwenderportal und WebServices auf welche das Portal für die Bearbeiter bzw. Supportmitarbeiter zugreift. Alle WebServices können natürlich auch von allen Beteiligten in Anspruch genommen werden, diese Aufteilung dient also ausschließlich der besseren Übersicht. Die Benennung der einzelnen WebServices erfolgt mittels der, ab *Punkt 4.3.2* eingeführten, fiktiven Methoden-bezeichnungen.

### 4.3.5.1 WebServices für die Clientsoftware

- **Register_New(System,User, [Users]):** Dieser Dienst wird für die Registrierung eines neuen Zielsystems, bzw. zur Überprüfung des Registrierungsstatus aufgerufen. Wie schon unter *Punkt 4.3.2.1* beschrieben, werden die Informationen über Benutzer und System, via Parameter an den Dienst übergeben. Der Dienst ruft seinerseits eine Basismethode außerhalb des Dienstes auf, welche die Speicherung in der Datenbank durchführt. Optional könnte vor Speicherung der Daten in der Datenbank noch eine Tabelle abgefragt werden, in der alle eintragungsberechtigten Systeme gelistet sind. Die Prüfung könnte z.B. auf MAC-Adressen erfolgen.

- **Request_Config(System):** Dieser Dienst liefert dem anfragenden Zielsystem, die zentral im System hinterlegten Informationen, für das im Parameter angegebene Zielsystem zurück. Diese Informationen beinhalten z.B. die Adressen des Front-Ends, welche dann via lokaler Listenabfrage (siehe auch *Abbildung 4-10*) am Zielsystem verwendet werden.

- **Request_Action(System):** Dieser Dienst liefert dem anfragenden Zielsystem die zentral im System hinterlegten, akuten (Steuer-)Informationen, für das im Parameter angegebene Zielsystem zurück. Diese Informationen sind kurzfristig erstellte Anforderungen, welche unter anderem von einem Supportmitarbeiter, über das Bearbeiterportal erzeugt wurden. Genauer noch, wurden diese Anforderungen von einem WebService erzeugt. Auf Grund dieser Informationen führt das Zielsystem lokale Aktionen aus (siehe *Punkt 4.3.2.3*) und meldet das Ergebnis via WebService zurück.

- **Send_Log(System, [User]):** Dieser Dienst ermöglicht einem Zielsystem die Übermittlung einer zuvor via WebService angeforderten Datei. Dabei wird via Parameter das Zielsystem und optional der Benutzer übergeben. Die übermittelten Daten werden wiederum zentral zur Ansicht durch den Support gespeichert.

- **Report_Action(System):** Dieser Dienst ermöglicht dem Zielsystem die Ausführung einer zuvor via *Request_Action* aufgetragenen lokalen Aktion zu bestätigen oder gegebenenfalls, Informationen über Fehler bei der Ausführung der lokalen Aktion zurückzumelden. Die Informationen werden dann vom zentralen System übernommen und verarbeitet.

- **Delete_Record(System , User, [Users]):** Durch die Verwendung dieses Dienstes kann sich ein Zielsystem inklusive Benutzer vom zentralen System abmelden. Dies erfolgt durch Löschung der Datenbankeinträge für Benutzer und System, welche via Parameter übergeben werden.

Ein typischer Ablauf der Dienstnutzung durch die Clientsoftware würde dann wie in *Abbildung 4-14* dargestellt aussehen:

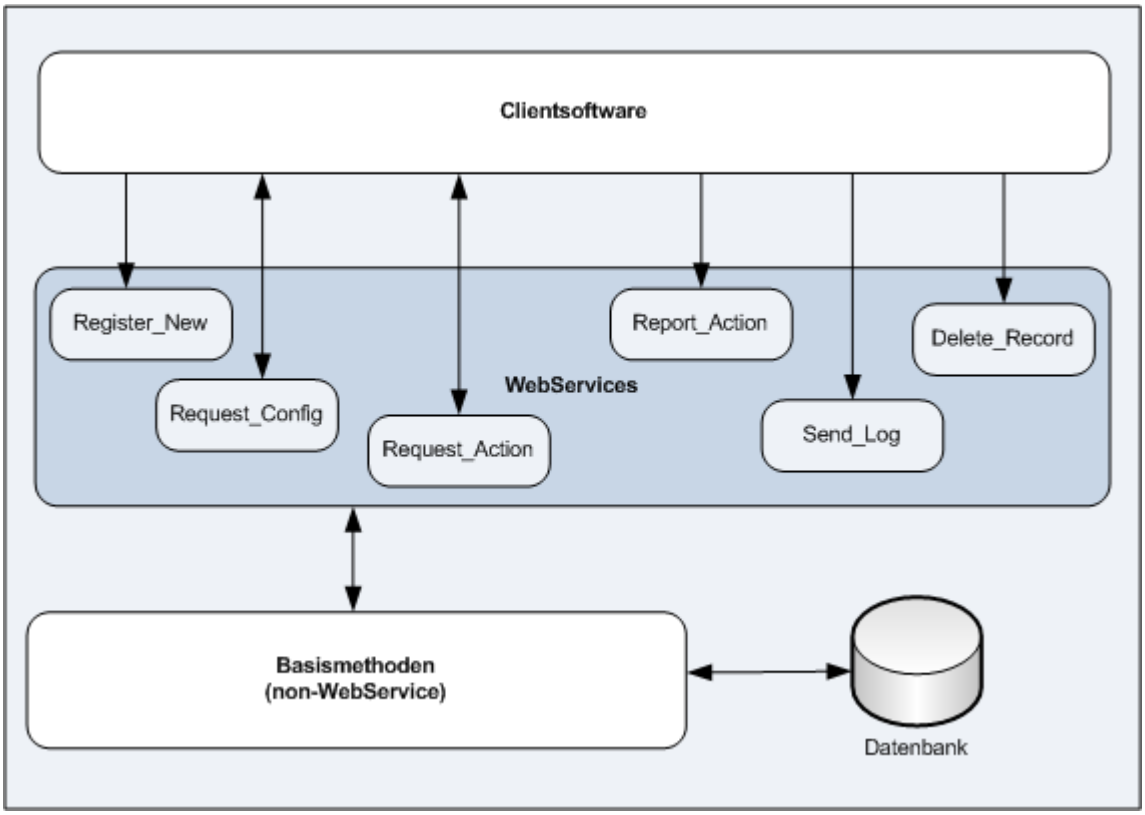


Abbildung 4-14: WebServices für Clientsoftware

*Abbildung 4-14* zeigt, dass nur *Request_Config* und *Request_Action* Informationen an die Clientsoftware zurückgeben und dass alle WebServices für die Verarbeitung auf Basismethoden zurückgreifen.

#### 4.3.5.2 WebServices für das Anwenderportal

- **Login_Customer(User, Password):** Dieser Dienst wird zur Authentifizierung von Anwendern genutzt. Benutzername und Passwort werden mittels Parameter an den Dienst übergeben. Der Dienst ruft eine entsprechende Basismethode auf, welche überprüft, ob Benutzername und Passwort valide sind. Eine mögliche Anbindung an z.B. *Microsoft Active Directory* erfolgt durch die Basismethode. Das Ergebnis der Prüfung wird dann an den Dienstaufrufer, in diesem Fall das Anwenderportal, zurückgesendet.

- **New_Incident():** Dieser Dienst liefert alle Auswahlmöglichkeiten, welche bei der Neueröffnung eines Incidents getroffen werden können bzw. müssen. Diese Informationen werden aus einer Datenbank geholt und sind somit stets aktuell. Auf Grund dieser Daten wird das Formular für einen neuen Incident für den Anwender aufgebaut.

- **Validate_Incident(User, System):** Durch diesen Dienst wird eine Vollständigkeitsprüfung der Angaben in einem neuen Incident ausgeführt. Es wird geprüft, ob es sich bei den Angaben des Erstellers um einen gültigen Benutzer bzw. um ein gültiges Zielsystem handelt. Hierzu wird vom Dienst via Basismethode die Datenbank abgefragt. Die Parameter *System* und *User* gehen dabei aus dem Incidentformular hervor.

- **Send_Incident(Data):** Dieser Dienst übermittelt den erstellten und validierten Incident zur Speicherung in der Datenbank. Die Daten werden in Parameter *Data* mitgegeben. Zusätzlich ruft dieser Dienst den WebService zur proaktiven Datenerfassung auf.

- **Request_Extended(System, User, Data):** Dieser Dienst löst die proaktive Datenerfassung für das angegebene Zielsystem aus. Hierzu wird eine Basismethode aufgerufen, welche auf Basis der angegeben Parameter, *System*, *User* und *Data* eine Anweisung für das Zielsystem hinterlegt. Das Zielsystem ruft, wie schon unter *Punkt 4.3.2.2* und *Punkt 4.3.5.1* beschrieben, regelmäßig den Dienst *Request_Action* auf und erhält somit die Aufgabe.

- **View_Incidents(User):** Dieser Dienst liefert alle, dem im Parameter *User* angegebenen Anwender, zugeordneten Incidents zur Ansicht und Bearbeitung.

- **Update_Incident(Incident):** Dieser Dienst nimmt Änderungen und Updates wie z.B. Kommunikationseinträge, für den in Parameter *Incident* angegebenen Incident entgegen und führt via Basismethode die Änderungen durch.

- **New_Task():** Funktionalität für neue Aufträge, analog zu *New_Incident*.

- **Send_Task(Data):** Funktionalität analog zu *Send_Incident*.

- **View_Tasks(User):** Funktionalität analog zu *View_Incidents*.

- **Update_Task(Task):** Funktionalität analog zu Update_Incident.

Ein typischer Ablauf der Dienstnutzung durch das Anwenderportal würde dann wie in *Abbildung 4-15* dargestellt aussehen:

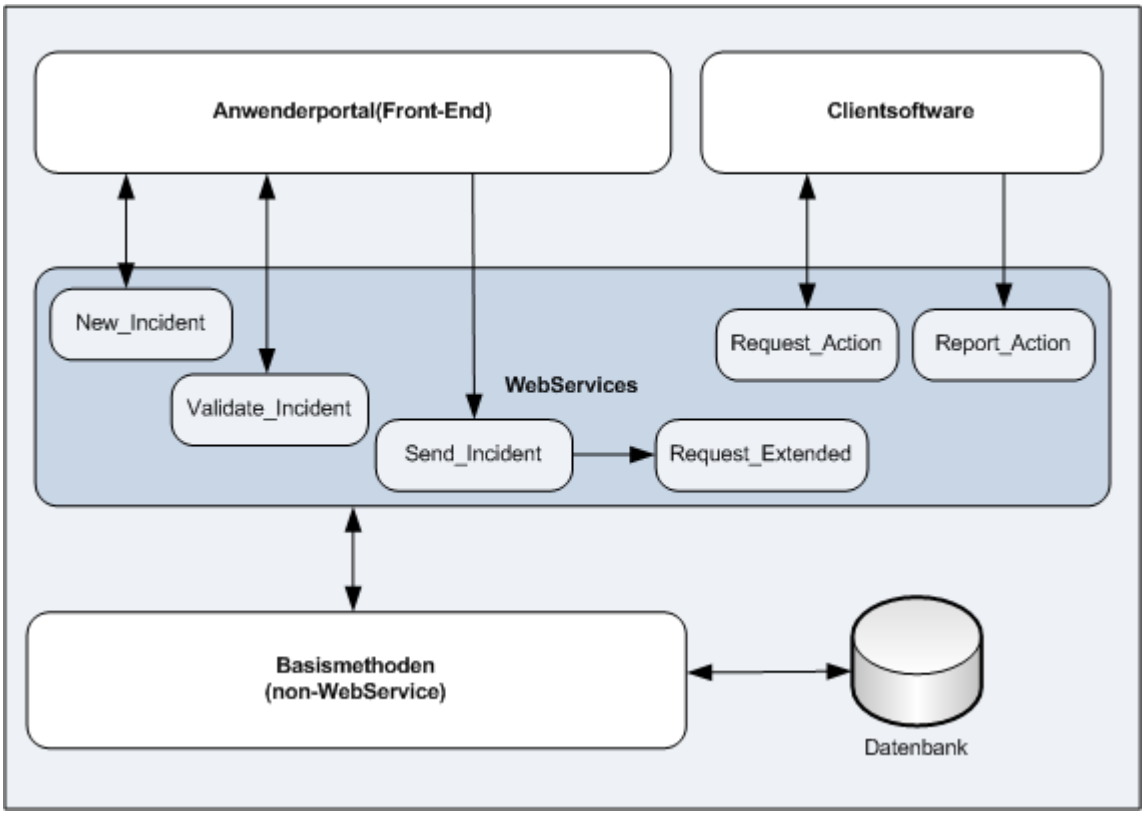


Abbildung 4-15: WebServices für Anwenderportal

In *Abbildung 4-15* ist ersichtlich, dass *Request_Extended* direkt von *Send_Incident* aufgerufen wird. Grundsätzlich sollten WebServices nicht direkt die Funktion eines anderen WebService verwenden, bei dieser Variante macht der direkte Aufruf aber Sinn, weil somit unmittelbar die proaktive Datenerfassung angestoßen wird.

#### 4.3.5.3 WebServices für das Bearbeiterportal

- **Login_Operator(User, Password):** Dieser Dienst wird zur Authentifizierung von Bearbeitern genutzt. Benutzernamen und Passwort werden mittels Parameter an den Dienst übergeben. Der Dienst ruft eine entsprechende Basismethode auf, welche überprüft, ob Benutzername und Passwort valide sind. Eine mögliche Anbindung an z.B. *Microsoft Active Directory*, erfolgt durch die Basismethode. Das Ergebnis der Prüfung wird dann an den Dienstaufrufer, in diesem Fall das Bearbeiterportal, zurückgesendet.

- **View_Jobs(User):** Dieser Dienst liefert alle, dem in Parameter *User* angegebenen Bearbeiter, zugeordneten Incidents und Tasks zur Ansicht und Bearbeitung.

- **Update_Job(JobID):** Dieser Dienst nimmt Änderungen und Updates wie z.B. Kommunikationseinträge, für den in Parameter *JobID* angegebenen Incident oder Task entgegen und führt via Basismethode die Änderungen durch.

- **Search_SystemOrUser([System], [User]):** Dieser Dienst liefert Ergebnisse für die Suche nach System- bzw. Benutzerinformation, welche in der zentralen Datenbank gespeichert sind. Die Parameter *System* und *Benutzer* sind optional, so kann entweder nach Benutzer, nach System, oder nach der Kombination gesucht werden.

- **Request_Extended(System, User, Data):** Dieser Dienst wird standardmäßig auch vom Anwenderportal genutzt, für die Beschreibung siehe *Punkt 4.3.5.2*.

- **Do_EndSystemTask(System, [User], Task):** Dieser Dienst löst die Ausführung einer Steueraufgabe für das angegebene Zielsystem aus. Hierzu wird eine Basismethode aufgerufen, welche auf Basis der angegeben Parameter, *System* und *Task* eine Anweisung für das Zielsystem hinterlegt. Der Parameter *User* ist optional und kann für spezielle, userbezogene Aufgaben genutzt werden. Das Zielsystem ruft wie schon unter *Punkt 4.3.2.2* und unter *Punkt 4.3.5.1* beschrieben, regelmäßig den Dienst *Request_Action* auf und erhält somit die Aufgabe.

- **Submit_Solution(Tag, [JobID], Data):** Dieser Dienst ermöglicht das Speichern wiederverwendbarer Informationen, die zur Problemlösung zukünftiger Fälle dienlich sein könnten, in der zentralen Datenbank. Mit den Parametern *Tag* und *JobID* werden Datum und Titel sowie die Verknüpfung zu einem bestehenden Job mitgegeben. Mit dem Parameter *Data* werden die Informationen mitgeliefert.

- **Search_Solution([Tag], [JobID], Text):** Dieser Dienst wird zur Abfrage der Informationsdatenbank eingesetzt. Mittels der Parameter *Tag*, *JobID* und *Text* kann nach bestehenden Informationen gesucht werden.

- **Modify_Workflow(Workflow, ConfigData):** Dieser Dienst ermöglicht das Anpassen bestehender Workflows. Dadurch können z.B. die Informationen, die bei der Eröffnung eines Incidents angegeben werden müssen, festgelegt werden.

Ein typischer Ablauf der Dienstnutzung durch das Bearbeiterportal würde dann wie in *Abbildung 4-16* dargestellt aussehen:

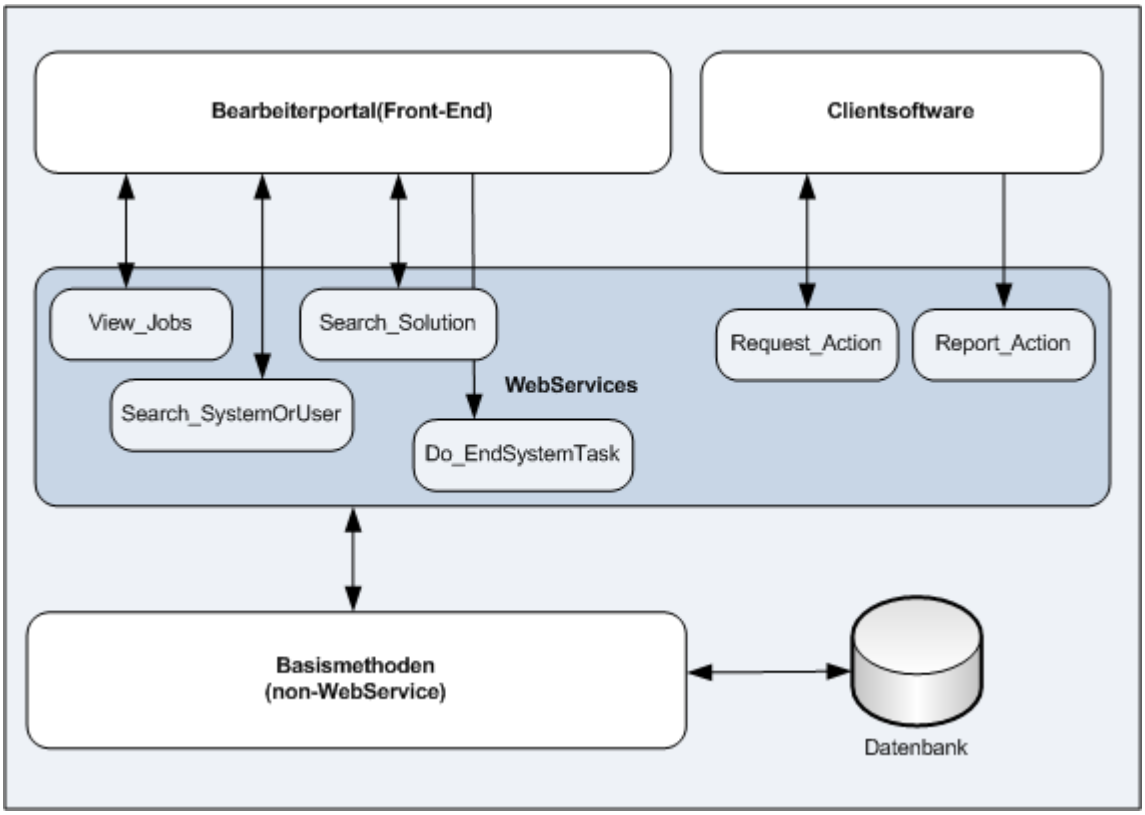


Abbildung 4-16: WebServices für Bearbeiterportal

## 4.3.6 Datenbank

Die Datenbank soll, wie schon im Grobkonzept unter *Punkt 4.2.1* erwähnt, durch eine RDBMS Lösung bereitgestellt werden. RDBMS-Systeme basieren auf Tabellen und zeichnen sich durch folgende Merkmale aus:

- Sehr einfaches Datenmodell
- Garantierte Datenkonsistenz (bei normalisierten Modellen), da Änderungen nur an einer Stelle vorgenommen werden.
- Beliebige Einstiegspunkte in die Datenbank, da alle Relationen gleichwertig sind und es somit keine Hierarchie gibt.
- Verknüpfung über Inhalte.

- Hohes Maß an Datenunabhängigkeit.
- Standardisierte einheitliche Datenbank-Manipulationssprache (SQL[15])

Zusätzlich hat ein RDBMS den Vorteil, dass durch Zerlegung der Objekte in Einzelteile und durch mengenorientierte Verarbeitung, komplizierte Abfragen auf große Datenmengen möglich sind. [Mut15]

Die Nutzdaten werden also in Tabellen organisiert, somit sollte nun eine vernünftige Aufteilung der Nutzdaten auf mehrere Tabellen erfolgen. Sinn macht hierbei eine Speicherung nach Themenkreisen bzw. nach Funktionalitäten. In *Abbildung 4-17* wird eine solche Aufteilung der Nutzdaten in unterschiedliche Tabellen gezeigt.


Abbildung 4-17: Datenbankaufbau

Tabellennamen werden im technischen Sprachgebrauch auch als Entitäten bezeichnet, somit kann eine Datenbasis auch als Entitätsmenge bezeichnet werden. Welche Informationen sich hinter den einzelnen Entitäten in *Abbildung 4-17* verbergen, wird nun nachfolgend aufgelistet.

[15] SQL ist eine Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf basierenden Datenbeständen. (siehe https://de.wikipedia.org/wiki/SQL)

- **Aktive Zielsysteme & Benutzer:** In dieser Tabelle werden alle aktiven Zielsysteme und Benutzer vorgehalten. Die Eintragung- und Löschungen erfolgen auf Initiative der Clientsoftware auf den Zielsystemen.

- **Aktionen für Zielsysteme:** In dieser Tabelle sind die abzuarbeitenden Aufgaben für die einzelnen Zielsysteme hinterlegt. Erzeugt werden die Aufgaben z.B. durch den WebService *Request_Extend* (siehe unter *Punkt 4.3.5.3*), welcher durch das Supportpersonal angestoßen wird.

- **Konfigurationen für Zielsysteme:** In dieser Tabelle sind die globalen Konfigurationseinträge für alle Zielsysteme sowie die speziellen Einzelkonfigurationen für die Zielsysteme hinterlegt. Abgeholt werden die Informationen unter Verwendung des WebService *Request_Config* ( siehe unter *Punkt 4.3.5.1*).

- **Berechtigte Anwender & Bearbeiter:** Mit den Informationen dieser Tabelle wird der Zugriff auf das Anwender- bzw. das Bearbeiterportal gesteuert, bzw. welche Tätigkeiten und Aktionen von wem konkret durchgeführt werden dürfen. In dieser Tabelle stehen also, die den Benutzerkonten zugewiesenen, Berechtigungslevel.

- **Berechtigte Zielsysteme:** Durch diese Tabelle kann gesteuert werden, welche Zielsysteme sich registrieren dürfen. Eine mögliche Filtervariante wäre z.B. die Prüfung auf MAC-Adressen[16].

- **Alle Tasks:** In dieser Tabelle werden alle offenen und bereits abgeschlossenen Tasks gespeichert. Auf diese Daten wird vom Anwender- und vom Bearbeiterportal zugegriffen.

- **Alle Incidents:** In dieser Tabelle werden alle offenen und bereits abgeschlossenen Incidents gespeichert. Auf diese Daten wird vom Anwender- und vom Bearbeiterportal zugegriffen.

- **Servicebetrieb Workflows:** In dieser Tabelle sind die Konfigurationen für die einzelnen Workflows, wie z.B. für Incident- und Taskmanagement gespeichert. Auf Grund dieser Vorgaben baut das Anwenderportal die Formulare für die Erstellung von neuen Incidents oder Tasks auf.

- **Bekannte Lösungen:** Diese Tabelle dient als Speicherort für gelöste Probleme. Die Einträge enthalten Problemlösungsinformationen und haben Schlagwörter (Tags)

[16] Die MAC-Adresse ist die Hardware-Adresse jedes einzelnen Netzwerkadapters, die als eindeutiger Identifikator des Geräts dient. (https://de.wikipedia.org/wiki/MAC-Adresse)

und ehemalige JobIDs als Assoziationen (siehe *Submit_Solution* unter *Punkt 4.3.5.3*).

- **...**: Hierdurch wird die dynamische Erweiterbarkeit der Datenbank um neue Tabellen angedeutet.

Mit der Festlegung der Datenbankstruktur kann dieses Kapitel nun abgeschlossen werden. Wie eine Lösung auf Basis des erstellten Konzepts realisiert und implementiert werden kann bzw. könnte, soll im nächsten Kapitel gezeigt werden.

# 5 Implementierung als Prototypensystem

## 5.1 Einleitung

Ziel dieses Kapitels ist die teilweise Umsetzung des in Kapitel 4 erarbeiteten Konzepts. Hierdurch soll die Machbarkeit der Konzeptionierung geprüft werden. Um das Zusammenspiel der beteiligten Komponenten testen zu können, werden diese jeweils durch die Erstellung eines Prototypen realisiert. Ein Prototyp steht für ein lauffähiges Stück Software, welches als Teilkomponente des Zielsystems interagiert und somit Rückschlüsse auf die Eignung eines Lösungsansatzes ermöglicht. Auf Grund der begrenzt zur Verfügung stehenden Software und Hardwareressourcen, wurden alle beteiligten Komponenten des Prototypensystems, physikalisch auf einer Hardwareplattform realisiert und getestet. Des Weiteren wurde bei den, zur Entwicklung verwendeten Werkzeugen, auf nicht kommerziell lizensierte Software zurückgegriffen. Bei der Benennung der Methoden im Programmcode, wurde auf die verwendeten, fiktiven Methodenbenennungen, wie z.B. *Register_New* oder *Send_Log*, aus Kapitel 4 zurückgegriffen. Dadurch lassen sich die umgesetzten Prototypen besser erklären.

## 5.2 Softwarekomponenten

Bezugnehmend auf das in *Abbildung 4-7* dargestellte Schema des Gesamtsystems, erfolgt nun in *Tabelle 5-1* eine Auflistung der einzelnen Komponenten laut Konzept und die Zuordnung der jeweiligen Prototypen-Softwarekomponente.

| Komponente | Prototypen-Softwarekomponente |
|---|---|
| WebServices | WebService basierend auf Microsoft.NET Framework 4.5 und bereitgestellt durch Microsoft IIS 10.0 Express |
| Datenbank | Microsoft SQL Server Compact 4.0 SP1 x64 ENU |
| Front-End | Webseite basierend auf Microsoft.NET Framework 4.5 und bereitgestellt durch Microsoft IIS 10.0 Express |
| Clientsoftware | Windows Service, entwickelt in VB.NET, basierend auf Microsoft.NET Framework 4.5 |

Tabelle 5-1: Komponenten/Prototypen-Softwarekomponenten

## 5.3 WebServices

Die WebService-Komponente wurde, durch ein in der Entwicklungsumgebung (siehe Punkt 1.6) aufgesetztes ASP.NET-Projekt, realisiert. Durch die Standardfunktionalitäten der Entwicklungsumgebung konnte der serverseitige Skeleton (siehe *Punkt 4.2.4.7*) des WebService automatisiert generiert werden. In das bestehende Programmgerüst wurden dann die einzelnen WebService-Methoden sowie auch die, nach außen hin nicht sichtbaren, Basis-Methoden einprogrammiert. Die Unterscheidung zwischen den Methoden erfolgt durch den in *Abbildung 5-1* dargestellten Bezeichner *WebMethod*.

```
<WebMethod()> _
Public Function RequestAction(ZielSystem As String) As String
    Return ReadFromDataBase(Actions_For_System, ZielSystem)
End Function
```

Abbildung 5-1: Beispielcode einer WebService-Methode

Für den Prototypentest wurden WebService-Methoden auf Basis der folgenden, in Kapitel 4 gelisteten, fiktiven Methoden *Register_New*, *Request_Action*, *Report_Action*, *Request_Extended*, *Send_Log* und *New_Incident* implementiert. Durch die Implementierung dieser Methoden konnten die grundsätzlichen Kommunikationswege sowie die proaktive Datenerfassung getestet werden. Die Datenbankanbindung wurde durch Basis-Methoden realisiert. In *Abbildung 5-1* ist der Aufruf eines Datensatzes aus der Tabelle *Actions_For_System,* via Basis-Methodenaufruf *ReadFromDataBase*, beispielhaft angedeutet. Für die Bereitstellung der WebService-Methoden wurde eine lokale Internet Information Server 10.0 Express Installation genutzt.

## 5.4 Datenbank

Die, zur Datenhaltung verwendete Datenbank, basiert auf einer lokalen Microsoft SQL Server Compact Installation. Diese Variante des Microsoft SQL Servers kann maximal 4 Gigabyte große Datenbanken erzeugen, ist lizenzfrei in der Verwendung und somit kostenlos. Die Datenbank wird als Datei mit dem Format *.sdf abgespeichert. Die Datenbank, die für den Test der Prototypen angelegt wurde, beinhaltet drei Tabellen. Die erste Tabelle wird mit der Entität *Act_Sys_Usr* bezeichnet und beinhaltet (siehe *Tabelle 5-2*) die registrierten Zielsysteme und Benutzer, welche aktuell aktiv sind. Die zweite Tabelle wird mit der Entität *Actions_For_System* bezeichnet und beinhaltet (siehe *Tabelle 5-3*) die geplanten Aufgaben, welche ein Zielsystem durchzuführen hat. Die dritte Tabelle wird mit der Entität *All_Incidents* bezeichnet und enthält (siehe *Tabelle 5-4*) alle Incidents und deren zugehörige Informationen.

| Act_Sys_Usr | |
|---|---|
| System[nvarchar]: | User[nvarchar]: |
| System1 | User1 |
| System2 | User2 |
| System … | User … |

Tabelle 5-2: Datenbank-Tabelle Act_Sys_User

In *Tabelle 5-2* sind die beiden Spalten *System* und *User* mit dem Datentyp nvarchar[17] vorbelegt.

| Actions_For_System | | | |
|---|---|---|---|
| System[nvarchar]: | Action[nvarchar]: | Requester[nvarchar]: | Incident[nvarchar]: |
| System1 | Act1 | Support_Usr_1 | Inc00000001 |
| System2 | Act2 | Support_Usr_2 | Inc00000002 |
| System … | Act … | Support_Usr_ ... | Inc0000000… |

Tabelle 5-3: Datenbank-Tabelle Actions_For_System

*Tabelle 5-3* zeigt die Tabelle *Actions_For_System* in der die einzelnen Aktionen auch einem Auslöser und bzw. oder einem Incident zugeordnet werden können.

| All_Incidents | | | |
|---|---|---|---|
| Incident[nvarchar]: | Starter[nvarchar]: | Supporter[nvarchar]: | Infotext[nvarchar]: |
| Inc00000001 | User1 | Support_Usr_1 | Text |
| Inc00000002 | User2 | Support_Usr_2 | Text |
| Inc0000000… | User … | Support_Usr_ ... | Text |

Tabelle 5-4: Datenbank-Tabelle All_Incidents

[17] Unicode-Zeichenfolgendaten variabler Länge.
(siehe https://de.wikipedia.org/wiki/Microsoft_SQL_Server)

In *Tabelle 5-4* sind die einzelnen Incidents und deren Informationen als Datensätze abrufbar. Das sich, aus den Tabellen der Datenbank ergebende Datenmodell, würde dann wie in *Abbildung 5-2* dargestellt aussehen.

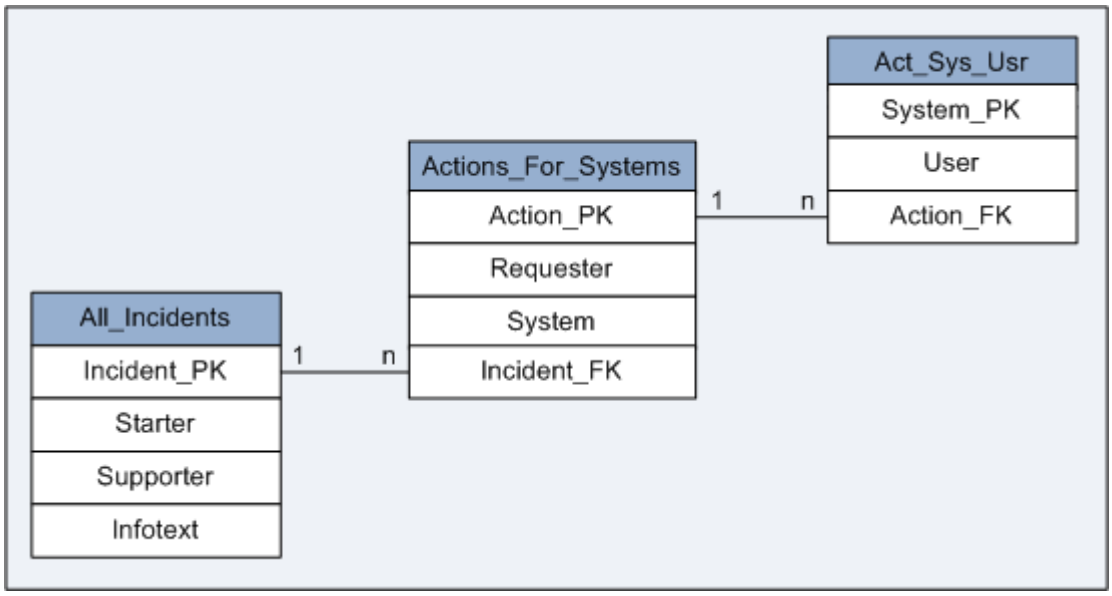


Abbildung 5-2: Datenmodell mit Relationen

Die beiden in *Abbildung 5-2* dargestellten *1 zu n* -Relationen stellen die Abhängigkeiten der Tabellen untereinander dar. Daraus ist ersichtlich, dass ein Incident mehrere Aktionen haben kann, eine Aktion aber immer nur genau einem Incident zugeordnet sein kann. Auch kann eine Aktion mehreren Systemen zugeordnet werden, aber ein System kann immer nur genau eine Aktion haben. Der Zugriff auf einen Datensatz in der Datenbank wird durch eine Basis-Methode realisiert. Ein solcher Zugriff ist in *Abbildung 5-3* dargestellt. Hier wird der Datensatz für eine Neuregistrierung eines Zielsystems samt Benutzer, in die Tabelle *Act_Sys_Usr* der verwendeten Datenbank geschrieben.

```
Private Sub WriteToDB(ByVal Act_Sys_Usr As String, _
                      ByVal Act_System As String, _
                      ByVal Act_User As String)

    DBcmd = New SqlClient.SqlCommand _
    ("Insert Into " & Act_Sys_Usr & _
     "(System, User) Values('System1', 'User1')", _
    DBcon)
    If DBcon.State = ConnectionState.Closed Then
        DBcon.Open()
    End If
    DBcmd.ExecuteNonQuery()
    DBcon.Close()
End Sub
```

Abbildung 5-3: Beispielcode für Datensatz schreiben

## 5.5 Front-End

Der Prototyp für das Front-End wurde nur rudimentär durch eine einfache Webseite, mit der Möglichkeit einen Incident zu eröffnen, realisiert. Für die Bereitstellung der Webseite wurde eine lokale Internet Information Server 10.0 Express Installation genutzt. Für den Test wurde auf die Implementierung einer Benutzeranmeldungslogik für das Anwenderportal verzichtet. Im Wesentlichen werden durch den Prototypen nur die Grundparameter für die Eröffnung eines Incidents abgefragt und ohne weitere Prüfung an das Testsystem weitergeleitet. In *Abbildung 5-4* ist der Vorgang der Incident-Eröffnung abgebildet. Bei Betätigung der Schaltfläche *Incident speichern und senden*, wird im Hintergrund die WebService-Methode *New_Incident* (siehe *Punkt 4.3.5.2*) aufgerufen. Dadurch wird in der Datenbank und genauer noch in der Tabelle *All_Incidents* (siehe *Tabelle 5-4*), ein neuer Datensatz für den Incident erzeugt. Zeitgleich wird die WebService-Methode *Request_Extented* (siehe *Punkt 4.3.5.2*) aufgerufen und somit die proaktive Datenerfassung angestoßen. Dadurch werden die entsprechenden Daten in der Tabelle *Actions_For_System* (siehe *Tabelle 5-3*) abgelegt. Für den Prototypentest werden als Aktionen jedoch nur Dummy-Einträge übergeben, welche von der Clientsoftware interpretiert werden und eine definierte Reaktion auslösen.

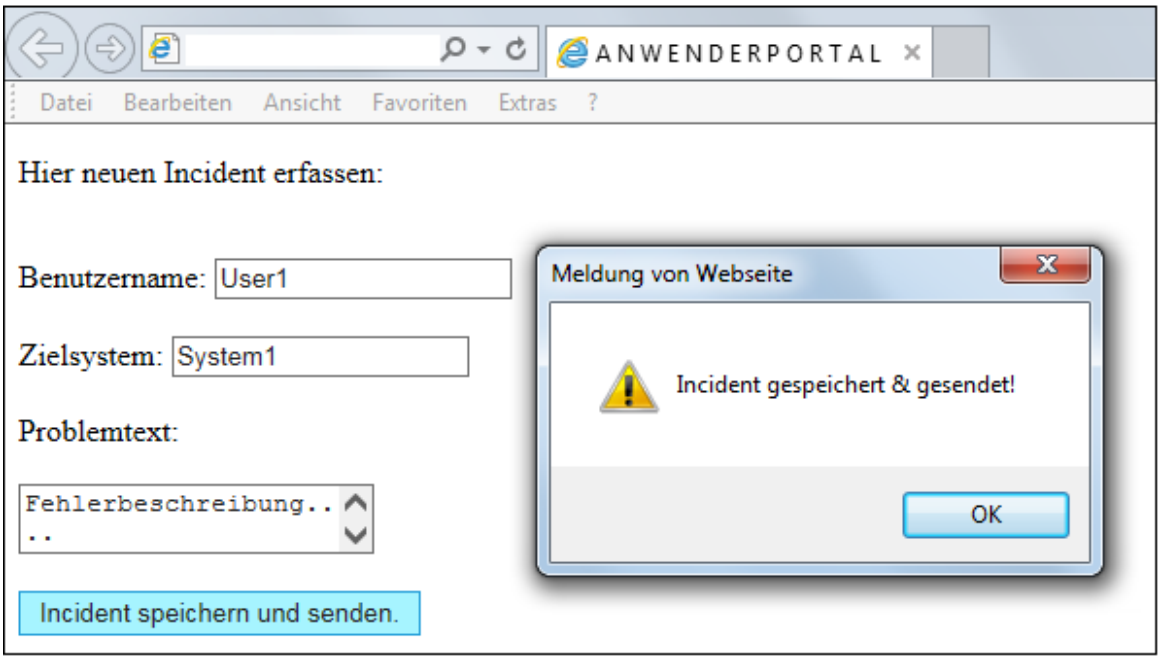


Abbildung 5-4: Incident-Eröffnung

## 5.6 Clientsoftware

Die Clientsoftware wurde durch einen, in VB.NET entwickelten Windows-Service realisiert, welcher unter dem Dienstkonto *LocalSystem*[18] betrieben wird. Dieses Konto verfügt über vollständige Administrationsberechtigungen für das lokale System. Ein Windows-Service bietet sich für die Umsetzung dieser Komponente an, weil er so konfiguriert werden kann, dass er im Zuge des Betriebssystemstarts automatisch gestartet wird und seinen Programmcode ausführt. Ein weiterer Vorteil sind die standardmäßig vorhandenen *OnStart()* und *OnStop()* Methoden, welche das einfache Einfügen für auszuführenden Programmcode bei Stop-Service und Start-Service erlauben.

Somit sind für alle Vorgänge, die beim Starten und Stoppen des Dienstes abzuarbeiten sind, die Codegerüste schon vorhanden. Von der Methode *OnStart()* wird der WebService *Register_New* aufgerufen. Von der Methode *OnStop()* wird der WebService *Delete_Record* aufgerufen. Für den sich periodisch wiederholenden Vorgang *Request_Config* kommt ein Timer zum Einsatz. Der Timer namens *Timer_Action* hat ein Intervall von 10 Sekunden und nach Ablauf eines Intervalls, wird die Methode *On_Timer_Action_Elapsed* automatisch ausgeführt, welche unter anderem den Programmcode für den WebService-Methodenaufruf *Request_Action* beinhaltet. Der Aufruf einer WebService-Methode sieht im Programmcode der Clientsoftware wie in *Abbildung 5-5* dargestellt aus. Hier wird abgefragt, welche Aktionen für das Zielsystem ausständig sind. Die auszuführende Aktion wird vom WebService zurückgeliefert und in der Variablen *strAction* gespeichert.

```
Dim ITSM_WebService As New WebServiceReference.WebServiceFunctionsSoapClient
strAction = ITSM_WebService.RequestAction(strSystemName)
```

Abbildung 5-5: WebServiceaufruf Request_Action

Um der Clientsoftware den Zugriff auf die WebServices zu ermöglichen, wurde die WSDL-Datei, welche die Nutzung der WebServices beschreibt, eingebunden. Durch diese Einbindung wurde automatisch eine Proxy-Klasse, auch Stub-Objekt genannt, generiert (siehe *Punkt 4.2.4.7*). Dieser sogenannte Dienstverweis ließ sich in der verwendeten Entwicklungsumgebung (siehe *Punkt 1.6*) unkompliziert, unter Angabe des Speicherortes der WSDL-Datei, erstellen.

[18] Siehe https://msdn.microsoft.com/en-us/library/windows/desktop/ms677973(v=vs.85).aspx

# 6 Ergebnis

## 6.1 Einleitung

In diesem Kapitel wird das Ergebnis der Konzeptionierung sowie die Umsetzung selbiger als Prototypensystem, diskutiert und bewertet. Dazu wird geprüft, inwieweit das ausgearbeitete Konzept geeignet ist, die unter *Punkt 3.4* definierten Anforderungen zu erfüllen. Das umgesetzte Prototypensystem soll zeigen, ob das erarbeitete Konzept grundsätzlich umsetzbar ist.

## 6.2 Ergebnis der Konzeptionierung

Wie schon zuvor in Kapitel 3 die beiden kommerziellen Lösungen, wird das in Kapitel 4 ausgearbeitete Konzept an den, zuvor in Kapitel 3 definierten speziellen Anforderungen, gemessen. Mit dem Unterschied, dass bei dieser Bewertung nicht ein fertig umgesetztes Softwareprodukt evaluiert und bewertet wird, sondern ein Konzept, welches auf den zuvor definierten Anforderungen fußt. Somit können diverse Anforderungen auch nur auf generelle Machbarkeit geprüft bzw. bewertet werden. Nichtsdestotrotz liefert ein Abgleich mit den Anforderungen Informationen über die Qualität des Konzepts.

- **Zu REQ1:** Kann vollständig erfüllt werden, weil das Konzept für das Anwenderportal, welches als Front-End für die Problemmelder dient, auf die Verwendung eines Webbrowsers setzt.

- **Zu REQ2:** Kann vollständig erfüllt werden, weil das Konzept ein gesondertes Portal für Supportmitarbeiter vorsieht, in welchem benutzerabhängig Rechte für Konfigurationsänderungen erteilt werden können.

- **Zu REQ3:** Kann vollständig erfüllt werden, weil das Konzept die Implementierung dieser Funktionalitäten nicht explizit ausschließt.

- **Zu REQ4:** Kann vollständig erfüllt werden, weil im Konzept zumindest eine zentrale Datenbank vorgesehen ist. Diese kann durch die Verwendung des entsprechenden WebService von mehreren Stellen mit Daten befüllt werden.

- **Zu REQ5:** Diese Anforderung kann, durch die im Konzept vorgesehene Clientsoftware, vollständig erfüllt werden. Die Kommunikation mit der zentralen Datenbank erfolgt dabei via WebService.

- **Zu REQ6:** Kann vollständig erfüllt werden. Konfigurationen können zentral hinterlegt werden. Die Abholung der Konfigurationseinstellungen erfolgt durch die Zielsysteme via WebService (siehe *Punkt 4.3.5.1*, *Request_Config*).

- **Zu REQ7:** Kann vollständig erfüllt werden, weil das Konzept für die Kommunikation zwischen den einzelnen Komponenten, ausschließlich die Nutzung von WebServices vorsieht.

- **Zu REQ8:** Kann vollständig erfüllt werden, weil die Anwender via Anwenderportal auf das System zugreifen. Die Anbindung für die Authentifizierung mittels LAN-Konto kann mittels Basismethode realisiert werden.

- **Zu REQ9:** Kann grundsätzlich erfüllt werden. Das Konzept sieht die notwendigen Komponenten für eine Implementierung dieser Funktionalitäten vor.

- **Zu REQ10:** Kann vollständig erfüllt werden, weil das Konzept bei Bedarf eine problemlose Aufstockung auf zwei oder mehr Servicekomponenten (siehe *Punkt 4.3.1*) vorsieht.

- **Zu REQ11:** Kann vollständig erfüllt werden, weil das Konzept den Einsatz von Verschlüsselungstechniken, wie z.B. SSL, nicht ausschließt.

- **Zu REQ12:** Kann vollständig erfüllt werden, weil das Konzept eine solche Implementierung nicht ausschließt.

- **Zu REQ13:** Kann vollständig erfüllt werden, weil das Konzept eine Anpassung der Anzahl der Servicekomponenten (siehe *Punkt 4.3.1*) an die Benutzeranzahl vorsieht und die Datenbank so strukturiert werden kann, dass die Anforderung eingehalten wird.

- **Zu REQ14:** Kann vollständig erfüllt werden, weil das Konzept eine solche Implementierung nicht ausschließt.

- **Zu REQ15:** Kann vollständig erfüllt werden, weil das Konzept auch eine Verteilung auf mehrere, geografisch verteilte Standorte (siehe *Punkt 4.3.1*) vorsieht.

Zusammenfassend kann gesagt werden, dass das erarbeitete Konzept den Anforderungen und der Zielsetzung zum Großteil gerecht wird. Es deckt die notwendigen Funktionalitäten, welche eine ITSM-Servicebetrieb-Lösung für die IT-Verwaltung von Unternehmen mit großen, geografisch verteilten IT-Landschaften benötigt werden, ab. Darüber hinaus zeigt das Konzept wie proaktiv Informationen, die zur schnellen Beseitigung von Problemfällen genutzt werden, gesammelt und dem

Supportpersonal zur Verfügung gestellt werden können. Des Weiteren sind auch aktive Eingriffe in betroffene Systeme in Form von Steuerbefehlen im Konzept vorgesehen, welche es ermöglichen, aufgetretene Probleme teilweise auch ohne Kommunikation oder Aufbau einer Remote-Session zu lösen. Durch den Einsatz von WebServices zur Kommunikation zwischen den beteiligten Komponenten, bleibt eine auf Basis des Konzepts umgesetzte Lösung weitgehend unabhängig von Firewall-Konfigurationen.

## 6.3 Bewertung des Prototypensystems

Durch den Aufbau einer prototypischen Implementierung des erarbeiteten Konzepts sollte gezeigt werden, dass solch eine ITSM-Lösung durchaus realisiert werden kann. Durch die Realisierung aller Teilkomponenten konnten die Kommunikationsvorgänge, wenn auch nur lokal begrenzt, getestet und damit die Machbarkeit geprüft werden. Wie schon eingangs bei den Nicht-Zielen dieser Diplomarbeit (siehe *Punkt 1.4*) erwähnt, wurden die nicht-funktionalen Anforderungen im Zuge der Implementierung nicht berücksichtigt. Auch wurde bei der Implementierung nur auf Windows-Systemen getestet.

Die wesentlichen Vorgänge, wie das Erfassen eines Incidents inklusive proaktiver Datenerfassung sowie Bereitstellung der Informationen in der zentralen Datenbank und die Erteilung von Steuerkommandos an die Clientsoftware inklusive Rückmeldung des Ergebnisses an die Servicekomponente, konnten, wenn auch nur rudimentär, erfolgreich getestet werden. Eine tatsächliche Verwendung des Systems für geografisch verteilte Standorte wurde mittels des Prototypensystems nicht getestet. Da die Kommunikation aber ausschließlich HTTP-basierend durchgeführt wird, ist dieser Anwendungstest nicht zwingend erforderlich um die Funktionalität bewerten zu können.

## 6.4 Allgemeine Erkenntnisse

Eine moderne IT-Umgebung eines mittleren bis großen Unternehmens, lässt sich ohne die Einbeziehung von ITSM-Leitfäden wie ITIL oder COBIT eigentlich nicht mehr vernünftig steuern. Je mehr die Anforderungen an die zu erfüllenden Serviceziele steigen, desto geringer ist die Chance, diese vollständig mit vorgefertigten Softwareprodukten abdecken zu können. Für Unternehmen mit geografisch verteilten IT-Landschaften machen sicherlich flexible ITSM-Lösungen Sinn, welche auf loser Kopplung und auf Webstandards basieren. Solche Implementierungen können ohne Weiteres als zukunftssicher angesehen werden. Die WebService-Technologie mit ihren unterschiedlichen Implementierungsformen, bietet die passenden Werkzeuge für die Umsetzung einer solchen Lösung, weil sie flexibel einsetzbar ist und theoretisch barrierefreie Kommunikation mit entfernten Systemen ermöglicht.

# 7 Zusammenfassung und Ausblick

In der vorliegenden Diplomarbeit wurde ein Konzept für eine ITSM-Lösung für den Servicebetrieb, welche sich speziell für den Einsatz in Unternehmen mit großen, geografisch verteilten IT-Landschaften eignet und auch Möglichkeiten für eine effizientere Abarbeitung von Problemfällen ermöglicht, erarbeitet. Zur Überprüfung der generellen Umsetzbarkeit des Konzepts, wurde anschließend ein Prototypensystem implementiert.

Als Vorarbeiten für die Konzepterstellung wurden zuerst die Grundlagen, Begriffe und Standards im Zusammenhang mit ITSM aufgearbeitet. Anschließend wurden im Zuge der Anforderungsanalyse die Besonderheiten geografisch verteilter IT-Landschaften beleuchtet. Eine Auswertung realer Problemfälle nach den Gesichtspunkten der Kommunikations- und Problemlösungsabläufe lieferte weitere Erkenntnisse, welche schlussendlich in die Erstellung eines Anforderungskatalogs einflossen.

Das ausgearbeitete Konzept fußt hauptsächlich auf der Verwendung der WebService-Technologie, welche es erlaubt, unterschiedliche Systeme, mehr oder weniger beliebig skalierbar, zu einem Kommunikationsverbund zusammenzuschalten. WebServices benötigen als Kommunikationsmedium grundsätzlich nur HTTP und verwenden zur Definition der Kommunikationsmethoden offene Standards. Ein Punkt im Konzept, der es wohl wert ist hervorgehoben zu werden, ist die angedachte proaktive Datenerfassung. Eine Umsetzung dieser Funktionalität, mit allen daran beteiligten Prozessen und Komponenten, mag in der Planungs- und Implementierungsphase spürbaren Mehraufwand verursachen, welcher sich aber durch die, im laufenden Betrieb entstehenden Vorteile, auf kurz oder lang amortisieren dürfte.

Das Einsatzgebiet für Lösungen, die auf dem in dieser Diplomarbeit ausgearbeiteten Konzept aufbauen, umfasst im Grunde genommen alle denkbaren IT-Konstellationen. Ein, dem Konzept gerechtes Einsatzgebiet, wäre jedoch im Umfeld von Unternehmen mit komplexen und geografisch verteilten IT-Umgebungen zu finden. Speziell für solche Unternehmen würde sich eine spezifische Implementierung wohl auf längere Sicht lohnen, weil dadurch Kosten und Personal im ITSM-Umfeld spürbar optimiert werden könnten.

In den letzten Jahren ist eine gut funktionierende IT-Abteilung für Unternehmen sukzessive wichtiger geworden. Dieser Trend wird sich wohl auch in den nächsten Jahren ungebrochen fortsetzen. Um den ständig wachsenden Anforderungen gerecht zu werden, wird man nicht umhinkommen, Konzepte zur Optimierung der IT-Abläufe umzusetzen, um auf Dauer wettbewerbsfähig zu bleiben.

# V. Literaturverzeichnis


[Bur07] Herbert Burbiel: SOA & Webservices in der Praxis.
© Franzis Verlag, 85586 Poing, 2007. - ISBN 978-3-7723-7627-6

[Cha10] Klaus Chantelau, Rene Brothun: Multimediale Client-Server-Systeme.
© Springer-Verlag, Berlin Heidelberg, 2010. - ISBN 978-3-540-79748-7

[Fin09] Patrick Finger, Klaus Zeppenfeld: SOA und WebServices.
© Springer-Verlag, Berlin Heidelberg, 2009. - ISBN 978-3-540-76990-3

[Gol06] Wolfgang Goltsche: COBIT kompakt und veständlich.
Friedr.Vieweg & Sohn Verlag | GWV Fachverlage GmbH, Berlin, 2006. - ISBN 10 3-8348-0141-0 / ISBN 13 978-3-8348-0141-8

[Krc05] Helmut Krcmar: Informationsmanagement.
© Springer-Verlag, Berlin Heidelberg, 2005. - ISBN 3-540-230015-7

[Off13] Office of Government Commerce: ITIL - Service Operation.
The Stationery Office Ltd, London, 2013. - ISBN 978-0113314034

[Olb08] Alfred Olbrich: ITIL kompakt und verständlich.
Vieweg+Teubner, Wiesbaden, 2008. - ISBN 978-3-8348-0492-1

[Pil10] Lionel Pilorget: MIIP: Modell zur Implementierung der IT-Prozesse.
Vieweg+Teubner, Wiesbaden, 2010. - ISBN 978-3-8348-1308-4

[Ros06] Joachim Rossberg, Rickard Redler: Pro Scalable .NET 2.0 Application Designs.
© Springer-Verlag, New York, 2006. - ISBN 1-59059-541-6

[Sch07] Alexander Schill, Thomas Springer: Verteilte Systeme.
© Springer-Verlag, Berlin Heidelberg, 2007. - ISBN 13 978-3-540-20568-5

[Sch10] Helmut Schlegl: Steuerung der IT im Klinikmanagement.
© Springer-Verlag, Wiesbaden, 2010. - ISBN 978-3-8348-0882-0

[TKö07] Peter T. Köhler: ITIL - Das IT-Servicemanagement Framework.
© Springer-Verlag, Berlin Heidelberg, 2007. - ISBN 10 3-540-37950-9


Onlinequellen:


[And11] Martin Andenmatten: blog.itil.org
[http://blog.itil.org/2011/06/itil/itil%C2%AE-2011-edition-mehr-als-nur-ein-update/]
Zugriff: 14. August 2015.

[BMC15] BMC Software: BMC Support Central
[https://docs.bmc.com/docs/display/public/ars81/BMC+Remedy+AR+System+client+server+architecture]
Zugriff: 20. August 2015.

[BMC151] BMC Software: BMC Support Central
[https://docs.bmc.com/docs/display/public/itsm81/Installing+BMC+Remedy+AR+System]
Zugriff: 25. September 2015.

[BMC152] Inc. BMC Software: BMC
[http://www.bmc.com/]
Zugriff: 22. 12 15.

[Gle15] Glenfis AG : http://os.itil.org
[http://os.itil.org/de/vomkennen/itil/serviceoperation/serviceoperationprozesse/incidentmanagement.php]
Zugriff: 25. August 2015.

[Ken15] Kenn Scribner:
[http://www.codeguru.com/csharp/csharp/cs_graphics/threading/article.php/c8193/Working-With-Asynchronous-NET-Web-Service-Clients.htm]
Zugriff: 27. Oktober 2015.

[Kla15] Dipl.-Ing. (V.i.S.d.P.) Klaus Lipinski: ITWissen Das große Online-Lexikon für Informationstechnologie ()
[http://www.itwissen.info/definition/lexikon/IT-service-management-ITSM.html]
Zugriff: 5. August 2015.

[Mut15] Peter Mutschke:
[http://www.gesis.org/fileadmin/upload/forschung/publikationen/gesis_reihen/iz_arbeitsberichte/ab5.pdf]
Zugriff: 16. November 2015.

# Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Steinach am Brenner, 04.01.2016

Nagele Philipp