

**Konzipierung und prototypische Entwicklung einer E-mail Integration  
auf Basis der Microsoft Exchange Web Services im Service Desk der  
CAFM Software GEBman 10**

Bachelorarbeit  
zur Erlangung des Abschlusses  
Bachelor of Engineering  
im Studiengang Informationstechnik

eingereicht von:  
Zilewitsch, Paul

1. Gutachter: Dr. rer. nat., Dipl.-Chem. Hansi Schilling
2. Gutachter: B.Sc. Sebastian Schulze

Tag der Themenübergabe: 22.04.2016

Tag der Einreichung: 18.07.2016

# **Autorenreferat**

ZILEWITSCH, Paul: Konzipierung und prototypische Entwicklung einer E-mail Integration auf Basis der Microsoft Exchange Web Services im Service Desk der CAFM Software GEBman 10, Berufsakademie Sachsen, Staatliche Studienakademie Dresden, Studiengang Informationstechnik, Bachelorarbeit, 2016.

XXX Seiten, XXX Literaturquellen , XXX Anlagen.

# Inhaltsverzeichnis

Inhaltsverzeichnis . . . . .	IV
Abbildungsverzeichnis . . . . .	VI
Abkürzungsverzeichnis . . . . .	VII
Tabellenverzeichnis . . . . .	VIII
1 Einleitung . . . . .	1
1.1 Die Software GEBman10 von der KMS Computer GmbH . . . . .	1
1.2 Modul Service Desk . . . . .	1
1.3 Motivation der Bachelorarbeit . . . . .	2
1.4 Vorgehen bei der Bachelorarbeit . . . . .	2
2 Servicedesk nach ITIL v3 . . . . .	3
2.1 Begriffsabgrenzung . . . . .	3
2.2 Unterschied zu Helpdesk . . . . .	5
2.3 Anforderungen eines Servicedesk . . . . .	6
2.3.1 Aufgaben . . . . .	6
2.3.2 Kategorien . . . . .	7
2.4 Vergleich verschiedener Servicedesk-Lösungen . . . . .	8
3 Der Servicedesk in GEBman10 . . . . .	10
3.1 Aktuelle Umsetzung . . . . .	10
3.2 Anforderungen für die Erweiterung . . . . .	10
4 Microsoft Exchange Server . . . . .	11
4.1 Grundlagen . . . . .	11
4.2 Exchange Web Services . . . . .	13
4.2.1 Funktionsweise . . . . .	13
4.2.2 Derzeitige Verwendung in GEBman10 . . . . .	14
5 Konzipierung . . . . .	15
5.1 Analyse aus Kapitel 1 und Kapitel 2 (UML-Diagramme etc.) . . . . .	15
5.2 Zielsetzung . . . . .	15
5.3 Sicherheitsaspekte . . . . .	15
6 Umsetzung . . . . .	17
6.1 Erweiterung des bestehenden Service Desk Moduls . . . . .	17
6.2 Erläuterung der wichtigsten Klassen und Methoden . . . . .	17
6.3 Fehlschläge/Erfahrungen . . . . .	17

7	Fazit . . . . .	18
7.1	Erweiterungsmöglichkeiten . . . . .	18
7.2	Schlussbemerkung . . . . .	18
	Anhangsverzeichnis . . . . .	19
	Anhang . . . . .	20
	Literaturverzeichnis . . . . .	24

## Abbildungsverzeichnis

Abbildung 1: ITIL Service Lifecycle . . . . .	4
Abbildung 2: Single Point of Contact . . . . .	5
Abbildung 3: Aufbau einer klassischen RPC Verbindung . . . . .	12
Abbildung 4: RPC over HTTP Prinzip . . . . .	12
Abbildung 5: EWS Funktionsumfang . . . . .	14
Abbildung 6: Sicherheitsprobleme . . . . .	15

## Abkürzungsverzeichnis

1	CAFM	Computer-Aided Facility Management
2	ITIL	Information Technology Infrastructure Library
3	HTTP	Hypertext Transfer Protocol
4	HTTPS	Hypertext Transfer Protocol Secure
5	API	Application Programming Interface
6	REST API	Representational State Transfer Application Programming Interface
7	SOAP	Simple Object Access Protocol
8	XML	Extensible Markup Language
9	SPoC	Single Point of Contact

## **Tabellenverzeichnis**

# **1 Einleitung**

## **1.1 Die Software GEBman10 von der KMS Computer GmbH**

Die KMS Computer GmbH wurde 1990 gegründet und konzentriert sich heute vorrangig auf den Vertrieb von entwickelter Software im Bereich Computer-Aided Facility Management System (CAFM). Computer-Aided bedeutet „computergestützt“ und Nävy beschreibt Facility Management präzise „als strategische Management-Disziplin, die die Analyse, Dokumentation und Optimierung aller kostenrelevanten Vorgänge rund um Gebäude und ihre Anlagen und Einrichtungen (Facilities) unter besonderer Berücksichtigung von Arbeitsplatz und Umfeld der Nutzer umfaßt.“<sup>1</sup> Seit 2011 entwickelt und vertreibt die KMS Computer GmbH die webbasierte Software GEBman 10. Es handelt sich bei GEBman 10 um eine CAFM-Software für Kommunen, Industrie und Gebäudeverwalter. Übersichten geografischer Informationen oder die Analyse von Sachdaten können individuell auf die Kunden abgestimmt werden. Außerdem ist es ein Werkzeug zur Verwaltung und ein Arbeitsmittel zur Unternehmensführung oder der finanziellen Planung. Die Anwendung kann als Desktop-Installation, als interne Weblösung oder als Cloud-Lösung betrieben werden. Aber auch mobile Lösungen einzelner Module sind bereits in Verwendung und werden stetig weiterentwickelt. Es wird schon jetzt deutlich, dass GEBman 10 mannigfaltig ist und mit über 40 Modulen auch in vielen Branchen zum Einsatz kommt. Gerade in sehr speziellen Bereichen wie beispielsweise Außenbeleuchtung oder Baumverwaltung kann es zu den unterschiedlichsten und ungewöhnlichsten Problemen kommen. Eine Grundvoraussetzung für die effiziente Lösung von Problemen, ist das Festhalten der genauen Vorkommnisse. Hierbei kann das Modul Service Desk in GEBman 10 durchaus hilfreich sein.

## **1.2 Modul Service Desk**

Im Modul Service Desk ist es möglich, Meldungen jeglicher Art aufzugeben. Ist beispielsweise eine Außenbeleuchtung komplett ausgefallen, kann ein Benutzer eine Störmeldung bezüglich der Außenbeleuchtung aufgeben. Dabei trägt er die genaue Problemstellung und das Gebäude in die Meldung ein, an dem dieses Problem auftritt. Ein anderer Benutzer hat nun die Möglichkeit, auf die Störmeldung Einsicht zu nehmen und die beispielhafte defekte Außenbeleuchtung zu reparieren. Oder aber er fragt nach der genauen Ursache nach und antwortet somit auf die Störmeldung. Probleme können genau spezifiziert und archiviert werden. Dadurch ist eine effizientere Lösung des Problems bei einem erneuten Auftreten möglich.

---

<sup>1</sup> J. Nävy (2006), Facility Management, Grundlagen Computerunterstützung Systemeinführung Anwendungsbeispiele, S.VII



### **1.3 Motivation der Bachelorarbeit**

Ziel der Bachelorarbeit ist es, das Modul Service Desk um eine E-Mail Integration zu erweitern. Mit dieser Erweiterung soll es möglich sein, über den E-Mail Verkehr auf Meldungen im Service Desk zu antworten oder neue Meldungen zu erstellen. Hierfür muss ein Konzept erstellt werden, um anschließend eine gute Implementierung zu erreichen.

Des Weiteren soll die Lösung für den Service Desk von GEBman10 mit anderen Softwarelösungen und verglichen werden, um Verbesserungsmöglichkeiten zu identifizieren.

### **1.4 Vorgehen bei der Bachelorarbeit**

Zu Beginn wird im Punkt 2 der Begriff Service Desk in einen Kontext gebracht, allgemeine Anforderungen bestimmt und einige Service Desk - Softwarelösungen verglichen. Im Anschluss im Punkt 3 wird der gegenwärtige Service Desk in GEBman10 durchleuchtet. Es werden außerdem die endgültigen Anforderungen an die E-Mail Integration beschrieben. Der Punkt 4 konzentriert sich auf den Microsoft Exchange Server, der einen wesentlichen Bestandteil der E-Mail Integration bildet. Auch wird an dieser Stelle die derzeitige Verwendung der Exchange Web Services in GEBman10 näher betrachtet. Durch eine Analyse der bis dato errungenen Erkenntnisse, kann im Punkt 5 eine Konzipierung erfolgen. Bei der Umsetzung im Punkt 6 wird auf die Erweiterung des Moduls Service Desk eingegangen und Erfahrungswerte erläutert. Abschließend kann dann im Punkt 7 eine Fazit gezogen werden und die mögliche Verbesserungen des Service Desk in GEBman10 genannt werden.

## 2 Servicedesk nach ITIL v3

### 2.1 Begriffsabgrenzung

Für die Klärung des Begriffs „Servicedesk“ ist es sinnvoll, sich auf die Information Technology Infrastructure Library - kurz ITIL - zu beziehen. ITIL ist zwar keine Norm, die in der IT-Branche eingehalten werden muss, dennoch bezieht man sich im IT-Service Management ausschließlich auf ITIL. Bereits 1989 wurde die Central Computer and Telecommunication Agency (CCTA) von der britischen Regierung beauftragt, Geschäftsprozesse und ihre Abhängigkeiten zu beschreiben und schriftlich festzuhalten.<sup>2</sup> Ziel war es, Abläufe in der Unternehmenswelt darzustellen und dadurch die IT-Betriebskosten zu reduzieren. Im Laufe der Jahre wurden die ersten Ausarbeitungen überarbeitet und ergänzt. Die ITIL Edition 2011 ist die derzeitige neuste Fassung und stellt ein Update der 2007 veröffentlichten Version ITIL v3 dar.<sup>3</sup> Auch bestimmte Normen leiten sich aus dem ITIL-Rahmenkonzept ab. Der internationale Standard ISO/IEC 20000 beispielsweise basiert auf der Version ITIL v2 und definiert die Minimalanforderungen des IT-Service-Managements für Organisationen.<sup>4</sup> ITIL kann deshalb als "Quasi-Standard" gesehen werden. Es ist ein Best-Practice Leitfaden, der beschreibt "Was zu tun ist, aber nicht wie". Das macht deutlich, dass ITIL durchaus Handlungs- und Interpretationsspielraum zulässt, aber dennoch in einem vorher definierten Rahmen greifbar sein muss. Beschrieben wird ITIL v3 in fünf Büchern, auf die später noch eingegangen wird:

- Service Strategy
- Service Design
- Service Transition
- Service Operation
- Continual Service Improvement

ITIL kann als Framework gesehen werden, mit dem Abläufe im Bereich IT Service beschrieben werden können. Genauer gesagt, spricht man IT Services Management, kurz ITSM. Hier werden alle Methoden erläutert, die nötig sind, um die bestmögliche Unterstützung von Geschäftsprozessen durch die IT-Organisation zu erreichen.<sup>5</sup> In ITIL ist ein Prozess definiert als „Satz von in Wechselbeziehung oder Wechselwirkung stehenden Tätigkeiten (und Mitteln), die Eingaben in Ergebnisse umwandelt. Zu den Mitteln können Personal, Einrichtungen und Anlagen, Technologie und Methodologie gehören. Eingaben für einen Prozess sind üblicherweise Ergebnisse anderer Prozesse.“<sup>6,7</sup>

<sup>2</sup> Vgl. Olbrich, A. (2008): ITIL kompakt und verständlich, S.1.

<sup>3</sup> Vgl. Vorlesung ITIL, S.25 5.Semester.

<sup>4</sup> Vgl. Buchsein, R./Victor, F./Günther, H./Machmeier, V. (2008): IT-Management mit ITIL® V3, S. 5ff.

<sup>5</sup> Vgl. Ebel, N. (2008): ITIL® V3 Basis-Zertifizierung, S.27 ff.

<sup>6</sup> Müller, A. (2015): Vorlesung ITIL, Skript S.18, 5.Semester.

<sup>7</sup> DIN EN ISO 9000:2005, Qualitätsmanagementsysteme, S. 23.

Jeder IT Service Management-Prozess hat eine charakteristische Zielrichtung und wird durch Funktionen unterstützt. Eine Funktion besteht aus einer Gruppe von Personen und deren Werkzeuge, die dafür verwendet werden, ein oder mehrere Prozesse oder Aktivitäten zu stützen.<sup>8</sup> Außerdem bewirkt das Zusammenspiel verschiedener IT Service Management-Prozesse, dass dem Kunden die notwendigen IT Services zur wirkungsvollen Unterstützung seiner Geschäftsprozesse geliefert werden. Der Service Lifecycle in Abbildung 2 veranschaulicht genau diese Kernprozesse und Kernfunktionen, die ein IT-Prozess während seiner gesamten Lebensdauer besitzt. Die einzelnen Teilbereiche decken sich mit den zuvor aufgeführten Büchern von ITIL v3.<sup>9</sup>

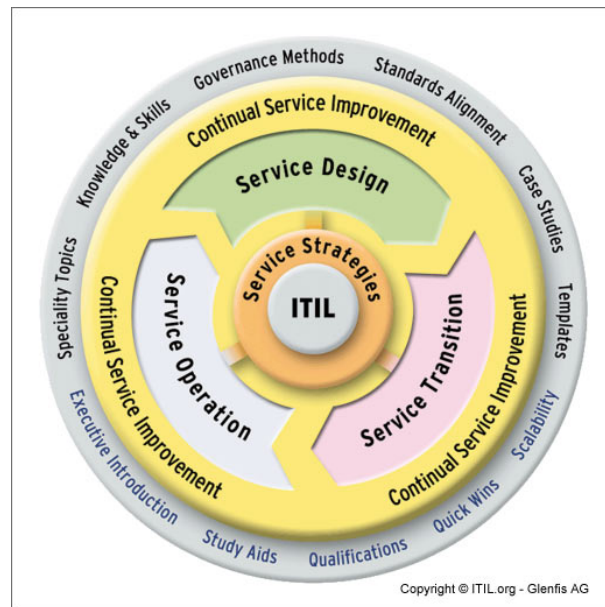


Abbildung 1: ITIL Service Lifecycle, Quelle: [http://os.ital.org/osMedia/site/t1media/JPEG/01\\_ital\\_imap.jpg](http://os.ital.org/osMedia/site/t1media/JPEG/01_ital_imap.jpg)

Auf alle Teilbereiche einzugehen, wäre zu zeitaufwendig, ist aber auch gar nicht nötig. Der Servicedesk ist nämlich Bestandteil der Service Operation und somit die richtige Anlaufstelle für die Begriffsklärung.

„Service Operation beschreibt den Abschnitt des Lebenszyklus, der von den Kunden primär wahrgenommen wird.“<sup>10</sup> In der Service Operations-Phase werden die Prozesse und Funktionen beschrieben, die einen stabilen und bestmöglich IT Service garantieren sollen. Bei dieser Verbindung von IT Organisation und Kunde wird besonders auf den Kunden eingegangen. Der Servicedesk ist hierbei „die zentrale Anlaufstelle, der Single Point of Contact (SPoC) zwischen Anwender und der IT-Organisation“<sup>11</sup>. Wie der Name schon verrät, kommt der Anwender nur über diese Schnittstelle in Kontakt mit der IT. Hier werden Meldungen der Anwender üblicherweise erfasst, kategorisiert und eingetragen. Der Servicedesk ist nicht nur eine Kommunikationsunterstützung, sondern bietet gleichzeitig eine Auskunft für bereits bekannte Probleme. Dadurch kann bei häufig auftretenden Service-Unterbrechungen schneller gehandelt werden.

<sup>8</sup> Vgl. Cannon, D./Wheeldon, D. (2007): Service Operation, S. 233.

<sup>9</sup> Vgl. Ebel, N. (2008): ITIL® V3 Basis-Zertifizierung, S.27 ff.

<sup>10</sup> Vgl. Ebel, N. (2008): ITIL® V3 Basis-Zertifizierung, S.439.

<sup>11</sup> Vgl. Ebel, N. (2008): ITIL® V3 Basis-Zertifizierung, S.439.

Auch Supportanfragen, Beschwerden, Verbesserungsvorschläge oder Änderungswünsche können in den Servicedesk eingetragen werden. Diese einzelnen Managementbereiche von ITIL v3 (Incident -, Problem -, Configuration -, Change -und Release Management) laufen im Service-desk zusammen, sodass der Benutzer nicht mehr entscheiden muss, in welchen Bereich sein Problem/Vorfall eingeordnet werden muss. Das ist dann Aufgabe des Supports. Die nachfolgende Abbildung verdeutlicht dieses Vorgehen.

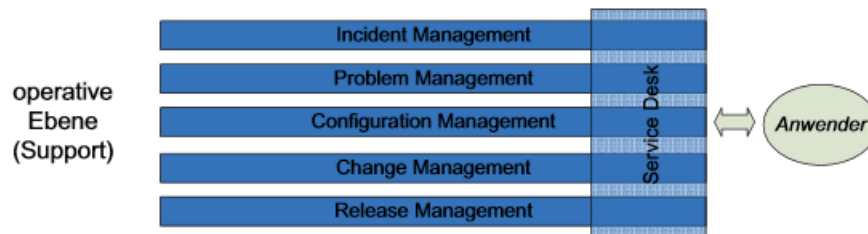


Abbildung 2: Single Point of Contact, Quelle: <http://edoc.hu-berlin.de/conferences/dfn2006/fischlin-roger-105/PDF/fischlin.pdf>

## 2.2 Unterschied zu Helpdesk

Bei der Begriffsabgrenzung zwischen Helpdesk und Servicedesk ist Vorsicht geboten. In mehreren Quellen ist zu finden, dass Helpdesk (auch User-Help-Desk) lediglich ein veralteter Begriff für den Servicedesk sei.<sup>12,13</sup> Im Internet heißt es beispielsweise auf Wikipedia: „Der Artikel Helpdesk und Servicedesk überschneiden sich thematisch.“<sup>14</sup> In anderen Literaturquellen tritt der Begriff Helpdesk erst gar nicht auf oder wird dem Servicedesk gleichgesetzt<sup>15</sup>. Im Zweifelsfall sollte man sich direkt auf das Buch ITIL Service Operation beziehen. In dem heißt es übersetzt unter dem Stichwort Help Desk: „Eine Anlaufstelle für Anwender, um Incidents zu erfassen. Ein Help Desk ist in der Regel eher technisch orientiert als ein Service Desk und stellt keinen Single Point of Contact für die gesamte Interaktion bereit. Der Begriff „Help Desk“ wird häufig auch als Synonym für Service Desk verwendet.“<sup>16</sup>

In den weiteren Ausführungen gilt deshalb der Helpdesk als Synonym für den Servicedesk. Auch im Punkt 2.3.3 „Vergleich verschiedener Servicedesk-Lösungen“ wird diese Begriffsüberschneidung bzw. Gleichbedeutung noch einmal deutlich.

<sup>12</sup> Vgl. Victor, F./Günther, H. (2005): Optimierte IT-Management mit ITIL, S.24.

<sup>13</sup> Meier, A./Myrach, T. (2004): IT-Servicemanagement, S.26.

<sup>14</sup> <https://de.wikipedia.org/wiki/Servicedesk>, Stand 08.11.2013

<sup>15</sup> Vgl. Olbrich, A. (2008): ITIL Kompakt und verständlich, S.19.

<sup>16</sup> Vgl. Cannon, D./Wheeldon, D. (2007): Service Operation, S. 233. Übersetzung entnommen aus: Ebel, N. (2008): ITIL® V3 Basis-Zertifizierung, S. 699.

## 2.3 Anforderungen eines Servicedesk

### 2.3.1 Aufgaben

Im folgenden sollen die wichtigsten Aufgaben eines Servicedesk aus Sicht der ITIL v3 erläutert werden. Hierfür wird zunächst stichpunktartig die Kernaussage festgehalten, um sie anschließend zu erläutern. Dabei beziehen sich die Kernaussagen auf der Ausarbeitung Olbrichs aus „ITIL Kompakt und verständlich“<sup>17</sup> und sind eine leicht abgewandelte Form vom ITIL v3 Band Service Operation<sup>18</sup>

- „Einheitlichen, zentralen Kommunikationsschnittstelle (SPoC) mit konkreten Ansprechpartner“

Der Kunde hat mehrere Möglichkeiten den Support zu kontaktieren. Schreibt der Kunde eine E-Mail an den Support, könnte diese ausgewertet und den Servicedesk eingetragen werden. Ebenso könnte er selbst einen Eintrag über ein Web-Frontend erstellen. Oder aber der Support erstellt einen solchen Eintrag im Servicedesk, wenn der Kunde zum Telefon greift. Es ist aber Voraussetzung, dass eine einheitliche und zentrale Kommunikationsschnittstelle bereitgestellt wird.

- „Aufnahme, Dokumentation und Auswertung aller Vorfälle“

Wenn alle Vorfälle ordnungsgemäß aufgenommen und dokumentiert wurden, kann schneller reagiert werden, wenn sich ein Problem wiederholt. Dass alle Vorfälle auch ausgewertet werden sollten, ist verständlich und kann eventuell dazu beitragen, Folgeprobleme frühzeitig zu erkennen.

- „Überwachung, Nachverfolgung und Eskalation von laufenden Supportvorgängen. Frühzeitiges Erkennen von Bedürfnissen und Problemsituationen“

Wie im Punkt zuvor erwähnt können Probleme frühzeitig erkannt werden, indem Vorfälle genauestens ausgewertet werden. Das ist aber längst nicht die einzige Möglichkeit, Bedürfnisse der Kunden zu erkennen. Gute Mittel für vorausschauende Handlungen sind bspw. Monitoring-Systeme oder log-Files. Sie liefern technische Informationen, die - nach einer Auswertung - Aufschluss über die aktuelle Lage des Kunden geben und in den Servicedesk integriert werden könnten.

- „Überprüfung der Einhaltung des Dienstleistungsgegenstands anhand von Service-Level-Agreements“

Mithilfe des Servicedesks kann kontrolliert werden, ob die vereinbarten Leistungen zwischen Auftraggeber und Beauftragter eingehalten wurden, wenn alle Vorfälle dokumentiert wurden.

---

<sup>17</sup> Olrich, A. (2008): ITIL kompakt und verständlich, S.19 f.

<sup>18</sup> Vgl. Cannon, D./Wheeldon, D. (2007): Service Operation, S. 110.

- „Reporting – Beauskunftung gegenüber den Usern (Kunden) und dem Management. Informationen über den aktuellen Status von Vorgängen, geplanten Änderungen und verschiedenen Nutzungsmöglichkeiten“

Der Servicedesk dient außerdem dazu, stets mit dem Kunden im Kontakt zu stehen. So können Information an den Kunden weitergeleitet oder auf der anderen Seite aktuelle Vorgänge, Status etc. des Kunden verfolgt und analysiert werden.

- „Überprüfen der Kundenzufriedenheit, Stärkung der Kundenbeziehung. Kontaktpflege. Aufspüren neuer Geschäftschancen“

Nicht zuletzt kann der Servicedesk auch als Instrument für einen ständigen Kontakt zum Kunden eingesetzt werden. Der Kunde hat dadurch den Eindruck, permanent mit der Support und somit der Firma verbunden zu sein. Das kann das Verhältnis zum Kunden stärken oder gar neue Geschäftsmöglichkeiten eröffnen.

### 2.3.2 Kategorien

Grundsätzlich gibt es drei Kategorien bezüglich der Architektur eines Servicedesks:

- Lokaler Servicedesk
- Zentraler Servicedesk
- Virtueller Servicedesk

Der lokale Servicedesk zeichnet sich dadurch aus, dass er innerhalb eines Bereiches selbständig agiert. Mehrere Unternehmensstandorte oder verschiedene Bereiche eines Unternehmens haben jeweils einen eigenen Servicedesk. Dieser kann präzise auf die Probleme und Prozesse vor Ort angepasst werden. Jedoch gestaltet sich eine Zusammenarbeit mehrere Bereiche gerade wegen dieser individuellen Servicedesks schwierig.

Beim zentralen Servicedesk ist diese bereichsübergreifende Zusammenarbeit keine Hürde, da es einen Servicedesk gibt, der für alle Bereiche eine gleichmäßige Zuständigkeit besitzt. Die Prozesse und Abläufe aller Benutzer sind hier identisch. Zwar sind die Kommunikationsmöglichkeiten hier sehr umfangreich, jedoch wächst die Informationsmenge rasant an. Eine kunden-nahe Betreuung wird durch den hohen Organisationsumfang deutlich erschwert.

Bei der Kompromisslösung dieser beiden Architekturmodellen stößt man auf den virtuellen Servicedesk. Die Informationseingabe der Benutzer kann durchaus an verschiedenen Standorten erfolgen ganz wie beim lokalen Servicedesk. Doch alle Daten werden gesammelt und zentral verwaltet, was dem zentralen Servicedesk entspricht. Entscheidend ist hierbei, dass es einheitliche Prozesse und Abläufe an den einzelnen Standorten geben muss. Individuelle Servicedesks

wären zu aufwendig in der Verwaltung. Auch so ist der Ressourcen -und Organisationsaufwand gegenüber den anderen Modellen enorm und bedarf guter Planung.<sup>19 20</sup>

## 2.4 Vergleich verschiedener Servicedesk-Lösungen

Nach diesem sehr theoretischem Ansatz die Anforderungen eines Servicedesks zu klären, wird nun auf Beispiele in der Praxis eingegangen. Wichtig sind dabei nicht Kriterien wie das äußere Erscheinungsbild oder die Kosten. Ziel soll es sein durch einen Vergleich gängiger Softwarelösungen, Verbesserungsmöglichkeiten der eigene Servicedesk-Funktionalität in GEBman10 zu ermitteln. Dabei wird auf die drei folgenden Punkte Wert gelegt:

- Funktionalität:

Die Funktionalität ist wohl das fundamentalste Kriterium. Hier ist entscheidend, welche Möglichkeiten dem Benutzer gegeben werden bspw. Meldungen/Tickets anzulegen, zu zuweisen, zu suchen oder zu filtern. Wichtig ist aber auch, welche Informationen in welcher Darstellungsform erhalten sind (Diagramme etc.) und welche Daten erfasst werden müssen bzw. können.

- Bedienbarkeit:

Aus diesem Blickwinkel wird untersucht, welche Bedienelemente zur Verfügung stehen. Eine Bewertung nach intuitiver Bedienbarkeit ist schwierig vorzunehmen, da das immer eine subjektive Ansicht enthält.

- Anpassbarkeit:

Inwieweit kann man bspw. die grafische Oberfläche vom Benutzer geändert und auf die eigenen Bedürfnisse angepasst werden.

Nach den Recherchen auf mehreren Review und Ranking Websites zum Thema Service Desk / Help Desk, sind drei Softwarelösungen wiederholt erwähnt und gut bis sehr gut bewertet worden.<sup>21</sup> Diese drei webbasierten Anwendungen werden nun kurz vorgestellt, um sie anschließend ihre Stärken und zu ermitteln.

- Freshdesk:

---

<sup>19</sup> Vgl. Olbrich, A. (2008): ITIL kompakt und verständlich, S.21.

<sup>20</sup> Vgl. Cannon, D./Wheeldon, D. (2007): Service Operation, S. 111 f.

<sup>21</sup> siehe Literaturverzeichnis

- Desk.com:

- Zendesk:

Aktuelle benutzt der Support von der KMS Computer GmbH die Software SysAid für den Service Desk. Durch eine Vielzahl von Einträgen und Erfahrungen der Mitarbeiter im Support, ist es sinnvoll auch diese Anwendung mit in den Vergleich einfließen zu lassen.

- SysAid:

Vgl. IT Notfallmanagement unter Incident Management



### **3 Der Servicedesk in GEBman10**

#### **3.1 Aktuelle Umsetzung**

#### **3.2 Anforderungen für die Erweiterung**

Qualität ist ein Maß für das Erfüllen von Anforderungen. Die Qualität des Servicedesk kann deshalb nur gesichert werden, wenn die Anforderungen möglichst genau definiert werden. Zu den zu erfüllen Anforderungen sollte aber noch festgehalten werden, welche Anforderungen nicht erfüllt werden sollen. Letzteres wird häufig nicht beachtet, ist jedoch ein wesentlicher Schritt für das Sicherstellen der Anforderungen.<sup>22</sup>

hier muss noch einmal auf alle Punkte von 2. eingegangen werden, um abzugrenzen, was eine Anforderung

---

<sup>22</sup> Vorlesungsreihe Qualitätsmanagement: Hans-Jörg Günther, 6.Semester

## 4 Microsoft Exchange Server

### 4.1 Grundlagen

Der Microsoft Exchange Server ist eine serverseitige Anwendung, die den Nachrichtenaustausch und die Zusammenarbeit im Unternehmen erleichtern soll.<sup>23</sup> Im Juni 1996 wurde die erste Version von Microsoft Exchange veröffentlicht. Sie löste das Mailsystem MS Mail ab, da dieses für einen Gebrauch mit über 500 Postfächern nicht ausgelegt und somit für größere Unternehmen nicht mehr sinnvoll war. Dieser Wechsel der Software wurde passend durch den Namen Exchange beschrieben, da es so viel heißt wie „Austausch“.<sup>24</sup> Ziel des Microsoft Exchange Servers ist es, Nachrichten zu verarbeiten und zu verwalten. Obwohl Exchange ein Mailserver ist, können neben E-Mails auch Termine angelegt oder Aufgaben vergeben werden. Somit wäre es denkbar, Exchange als zentrale Anlaufstelle der Unternehmenskommunikation einzusetzen.

Microsoft Exchange ist klar in den Bereich Groupware einzuordnen. Groupware-Software wird vor allem zur Unterstützung der internen als auch der externen Unternehmenskommunikation genutzt. Ellis, Gibbs und Rein beschreiben die bekannteste Form von Groupware als ein „computer-based message system, which supports the asynchronous exchange of textual messages between groups of users“<sup>25</sup>. Also ein Computersystem für den asynchronen Austausch von Textnachrichten innerhalb einer Gruppe. Meistens sind dies Arbeitsgruppen im Unternehmensumfeld. Microsoft Exchange erfüllt dieses Kriterium. Vorausgesetzt die Benutzer verfügen über eine Client-Software.

Um auf die Inhalte des Microsoft Exchange Servers zugreifen zu können, benötigt jeder Benutzer eine Client-Software. Verwaltung des Postfaches, Zugriff auf öffentliche Ordner und natürlich auch Empfangen und Senden von E-Mails sind die Hauptziele einer Client-Software. Im Anhang auf Seite 20 findet man eine Abbildung, die eine umfassende Übersicht verschiedener Clients darstellt. Diese Übersicht ist nicht Vollständigkeit, bildet aber dennoch die am häufigsten verwendeten Client-Systeme ab.

Am bekanntesten ist sicher der Microsoft Outlook-Client. Aus diesem Grund wird die Verbindung eines Clients mit dem Exchange Server am Beispiel des Outlook-Clients erläutert. Outlook kommuniziert nach dem RPC-Prinzip mit dem Exchange Server. Die Variante RPC über TCP/IP gilt zwar seit Exchange 2013 als veraltet, ist aber für das nähere Verständnis durchaus wichtig, da alle Weiterentwicklungen RPC im Hintergrund weiter nutzen. RPC bedeutet "Remote Procedure Call" und wird verwendet, um eine Verbindung zu einem Dienst eines Servers herzustellen. Als Übertragungsprotokoll fungiert TCP/IP (Transfer Control Protocol/Internet Protocol).<sup>26</sup> Um die Kommunikation von RPC über TCP/IP zu verstehen, ist die nachfolgende

<sup>23</sup> Joos, T. (2013): Microsoft Exchange Server 2013, S.26.

<sup>24</sup> Vgl. Übersetzer

<sup>25</sup> Ellis, C. A./Gibbs, S. J./Rein, G. L. (1991): Groupware - Some Issues and Experiences. Communications of the ACM, 34(1), S. 38-58.

<sup>26</sup> Vgl. <http://www.msxfaq.de/verschiedenes/rpc.htm>

Abbildung hilfreich.

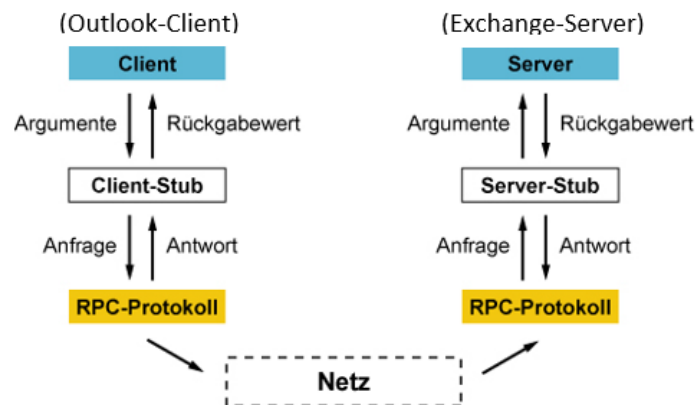


Abbildung 3: Aufbau einer klassischen RPC Verbindung, in Anlehnung an:  
<https://technet.microsoft.com/de-de/library/e8feb37e-f3a9-4f26-bed0-6583d8a110ed>

Der Client ruft eine Prozedur mit spezifischen Argumenten (Eingabeparameter) auf. Der Client-Stub aktiviert eine gleichnamige Prozedur und wandelt die Argumente in ein plattformunabhängiges Datenformat um. Die Daten werden dann über das Netz mithilfe des TCP/IP-Protokolls an den Server geschickt. Dort erhält der Server-Stub die Anfrage und wandelt die Argumente in das lokale Format des Servers um. Nun ruft der Server die gewünschte Prozedur mit den Eingabeparametern auf und der Rückgabewert kehrt in den Server-Stub zurück. Nach einer erneuten plattformunabhängigen Umwandlung der Datenformate, wird die Antwort an den Client-Stub geschickt. Im letzten Schritt erhält der Client die in das lokale Format umgewandelte Antwort (Rückgabewerte) auf seine Anfrage.<sup>27</sup>

Ab Exchange 20013 wird der Datenaustausch zwischen dem Outlook-Client und dem Exchange Server standardmäßig über RPC/HTTP (auch Outlook Anywhere genannt) geregelt. Wie der Name schon vermuten lässt, läuft die gesamte Kommunikation bei dieser Variante über HTTP bzw. HTTPS. Somit kann über das Internet auf den Exchange Server zugegriffen werden. Outlook Anywhere ist besonders für Mitarbeiter geeignet, die von zu Hause auf ihr Exchange-Postfach zugreifen möchten, da sie sich nicht im Unternehmensnetzwerk befinden müssen.<sup>28</sup>

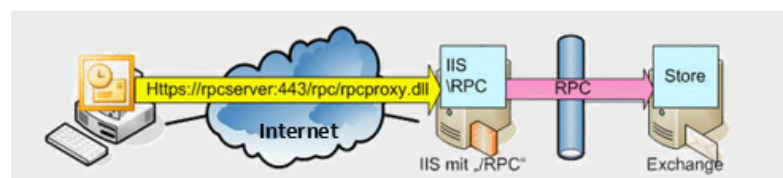


Abbildung 4: RPC over HTTP Prinzip, in Anlehnung an:  
<http://www.msxfaq.de/clients/oagrundlagen.htm>

In der Abbildung 4 sieht man deutlich, dass der Outlook-Client über HTTPS mit dem IIS kommuniziert, auf dem ein virtuelles RPC-Verzeichnis existiert. Der IIS ist eine Microsoft-

<sup>27</sup> Vgl. Schneider, U. (2012): Taschenbuch der Informatik, S.406 f.

<sup>28</sup> Vgl. Joos, T. (2013): Microsoft Exchange Server 2013, S.33, S.254.

Webserverplattform, die Webanwendungen und Dienste bereitstellen und verwalten kann.<sup>29</sup> Der IIS wiederum, baut über einen RPC-Proxy eine Verbindung mit dem Exchange Server auf. Wichtig ist hierbei, dass die Kommunikation des Clients sich auf HTTP oder HTTPS beschränkt.<sup>30</sup>

## 4.2 Exchange Web Services

Die Grundlagen von Microsoft Exchange Server sind geklärt. Doch wie können nun Programmierer über Quellcode auf den Exchange Server zugreifen. Hierfür hat Microsoft im Laufe der Jahre eine Reihe von Programmierschnittstellen bereitgestellt. Eine Schnittstellen zur Anwendungsprogrammierung wird als API (englisch application programming interface) bezeichnet. Es ist nicht nötig, alle Programmierschnittstellen zu erläutern, da eine Vielzahl der APIs keine Verwendung mehr finden oder nicht mehr unterstützt werden. Zu Informationszwecken befindet sich jedoch im Anhang auf Seite 21 eine Übersicht einiger Programmierschnittstellen ab dem Jahr 1992. Ab Exchange 2007 setzt Microsoft immer mehr auf den Exchange Web Services (kurz EWS) als Programmierschnittstelle und baut diese seitdem weiter aus.<sup>31</sup> Redmond schreibt im Handbuch Exchange 2010: „Abgesehen von Windows PowerShell liegt der Schwerpunkt für die meisten Entwickler jetzt auf der API EWS (Exchange-Webdienste), die in Exchange Server 2007 eingeführt wurde.“<sup>32</sup> Er macht deutlich, dass EWS die erste Anlaufstelle für Entwickler ist. Eine weitere Programmierschnittstelle namens REST API erschien 2015 mit Office 365. Der REST API wird aber keine weitere Beachtung geschenkt, da sie lediglich in der Office 365-Umgebung Anwendung findet.<sup>33</sup>

### 4.2.1 Funktionsweise

Standardmäßig bildet das Simple Object Access Protocol (kurz SOAP) eine Grundlage für Web-Services. Schneider beschreibt SOAP als „ein RPC-Mechanismus, bei dem die übertragenen Daten im XML-Format codiert werden.“<sup>34</sup> Schill und Springer schreiben genauer: „Das objektorientierte Kommunikationsprotokoll SOAP ermöglicht die Kommunikation zwischen heterogenen Diensten unter interner Nutzung des Hypertext Transfer Protocol (HTTP) und mit Kodierung der Parameter in der eXtensible Markup Language (XML).“<sup>35</sup> Auch die Exchange Web Services funktionieren nach diesem Prinzip.<sup>36</sup>

Mit Exchange 2007 wurde erstmals der Exchange Web Services bereitgestellt und löste nach und nach andere APIs ab. Die nachfolgende Abbildung zeigt, wie sich EWS seit 2007 entwickelt

<sup>29</sup> Vgl. Volodarsky, M./Cheah, B./Londer, O./Hill, B./Schofield, S./Aguilar Mares, C./Meyer, K./ Microsoft IIS Team (2009): Microsoft Internetinformationsdienste (IIS) 7.0 – Die technische Referenz, S.3.

<sup>30</sup> Vgl. <http://www.msxfaq.de/clients/oaggrundlagen.htm>

<sup>31</sup> <http://www.msxfaq.de/code/ews.htm>

<sup>32</sup> Vgl. Redmond, T. (2011): Microsoft Exchange Server 2010, S.634.

<sup>33</sup> <https://msdn.microsoft.com/de-de/library/office/dn776319%28v=exchg.150%29.aspx>

<sup>34</sup> Vgl. Schneider, U. (2012): Taschenbuch der Informatik, S.407.

<sup>35</sup> Vgl. Schill, A./Springer, T. (2012): Verteilte Systeme. S.69.

<sup>36</sup> Vgl. <https://msdn.microsoft.com/en-us/library/office/dd877045%28v=exchg.140%29.aspx>

hat.

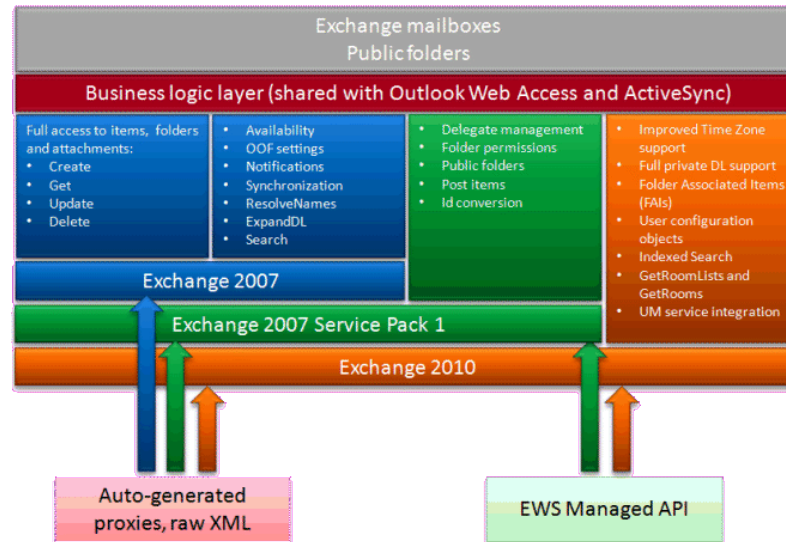


Abbildung 5: EWS Funktionsumfang, Quelle: <http://www.msxfaq.de/code/ews.htm>

EWS bietet einen großen Funktionsumfang und ist somit eine hervorragende Möglichkeit auf die Daten des Exchange Servers zuzugreifen. Aus diesem Grund haben sich die Entwickler der KMS Computer GmbH für diese Programmierschnittstelle entschieden.

#### 4.2.2 Derzeitige Verwendung in GEBman10

GEBman10 unterstützt Exchange Server 2007, Exchange Server 2007\_SP1, Exchange Server 2010 sowie Exchange Server 2010\_SP1. In Groupware kann....Auswahl getroffen werden. Die Implementation erfolge über EWS bzw. mit der Hilfe von der Microsoft Exchange Web Services Managed API 2.2.

Diese bereits vorhandenen Funktionalitäten werden als Grundlage für die folgenden Konzipierung und anschließende Implementierung genutzt.

## 5 Konzipierung

### 5.1 Analyse aus Kapitel 1 und Kapitel 2 (UML-Diagramme etc.)

### 5.2 Zielsetzung

### 5.3 Sicherheitsaspekte

Immer wieder vernachlässigen Entwickler die Sicherheit ihrer Implementierungen. Das liegt meistens an mangelnder Zeit, da Releases einen festen Zeitplan verfolgen, den es einzuhalten gilt. Es kann aber auch sein, dass die Implementierung nicht aus dem Blickwinkel der Sicherheit betrachtet wird. "Hauptsache es funktioniert erst einmal", wird dann häufig als Argument genutzt. Natürlich hat das wenig mit Sicherheit zu tun. Dabei können es Entwickler mit wenig Aufwand, Angreifern deutlich schwerer machen. Deswegen werden im nachfolgendem zwei Sicherheitsprobleme für die Umsetzung des Konzepts in GEBman 10 besprochen. Die Abbildung XX zeigt zwei kritische Bereiche, die genauer erläutert werden müssen.

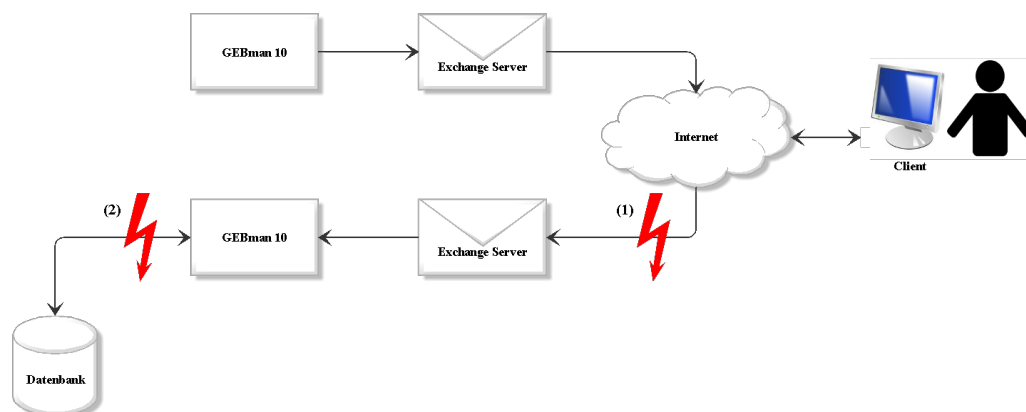


Abbildung 6: Sicherheitsprobleme, Quelle:eigene Darstellung

Der rechte rote Blitz - gekennzeichnet mit der (1) - symbolisiert das erste Problem.

Beim zweiten roten Blitz der Abbildung XX - gekennzeichnet mit der (2) - wird uns die Arbeit nicht wie beim ersten kritischen Bereich abgenommen. GEBman 10 synchronisiert in regelmäßigen Abständen die Nachrichten vom hinterlegten Exchange Server. Entsprechend ihrer ID werden die Nachrichten in die Datenbank von GEBman 10 gespeichert. Nun könnte ein Angreifer beispielsweise versuchen, in den Mail-Body versteckte SQL -Anweisungen oder JavaScript-Code einzuschleusen. Werden die SQL-Anweisungen in die Datenbank geschrieben, ohne sie vorher zu validieren, könnte der Angreifer Informationen über Daten in der Datenbank erlangen. Im schlimmsten Fall könnte er sie zerstören. Diese Angriffsmethode nennt sich SQL-Injection und zielt darauf ab, die normalen SQL-Statements mittels Sonderzeichen zu manipulieren. Selbst einfache Zeichen wie -"können bewirken, dass alles, was hinter den beiden Bindestrichen steht, ignoriert wird. In T-SQL sind die beiden Bindestriche das Zeichen

für einen Kommentar. Bei JavaScript-Code haben wir ein ähnliches Problem mit bestimmten Zeichen. Wird zum Beispiel die einfache Zeichenfolge «script>alert('hallo')<script> in die Datenbank geschrieben, passiert erst einmal gar nichts. Die SQL-Statements werden hierdurch nicht manipuliert. Doch wird diese Zeichenfolge beispielsweise als Antwort auf eine Meldung geladen, erkennt der Browser möglicherweise JavaScript-Code anstatt einfachen Text. Das hat in unserem Beispiel zur Folge, dass der Browser eine kleine Nachricht meldet (siehe Abbildung). Auch hier muss vor dem Einfügen der Zeichenfolge, eine Validierung vollzogen werden. Welche Zeichen genau gefiltert werden müssen, wird im Punkt 5 Umsetzung erläutert. Man sollte sich jedoch nicht darauf verlassen, dass die Schutzmechanismen des Microsoft Exchange Servers alle Zeichen und Zeichenfolgen als Bedrohung erkennen. Zeichen wie - oder »" können durchaus im normalen Schriftverkehr gebräuchlich sein. Es gibt für Angreifer noch mehr Angriffsmöglichkeiten beispielsweise zwischen Client und Internet/Server über Man-in-the-Middle etc.. Darauf soll aber in dieser Arbeit nicht weiter eingegangen werden, da das ein sehr umfangreiches Thema ist und eine eigenständige Bachelorarbeit bilden könnte. Eins sollte jedem klar sein: Hat ein Cracker genügend Zeit und Ressourcen, ist ein System, das über das Internet kommuniziert, äußerst schwer vollkommen zu sichern. Durch die Analyse aus Punkt 2 und Punkt 3 konnte ein genaues Ziel gesetzt werden. Die Funktionsweise der Exchange Web Services sind geklärt und auch Sicherheitsaspekte wurden berücksichtigt. Die Konzipierung ist somit abgeschlossen und die Implementierung steht nichts mehr im Wege.

## **6      Umsetzung**

<https://www.microsoft.com/en-us/download/details.aspx?id=42951>

hier die Managed API herunterladen

### **6.1    Erweiterung des bestehenden Service Desk Moduls**

### **6.2    Erläuterung der wichtigsten Klassen und Methoden**

### **6.3    Fehlschläge/Erfahrungen**



## **7      Fazit**

### **7.1    Erweiterungsmöglichkeiten**

### **7.2    Schlussbemerkung**

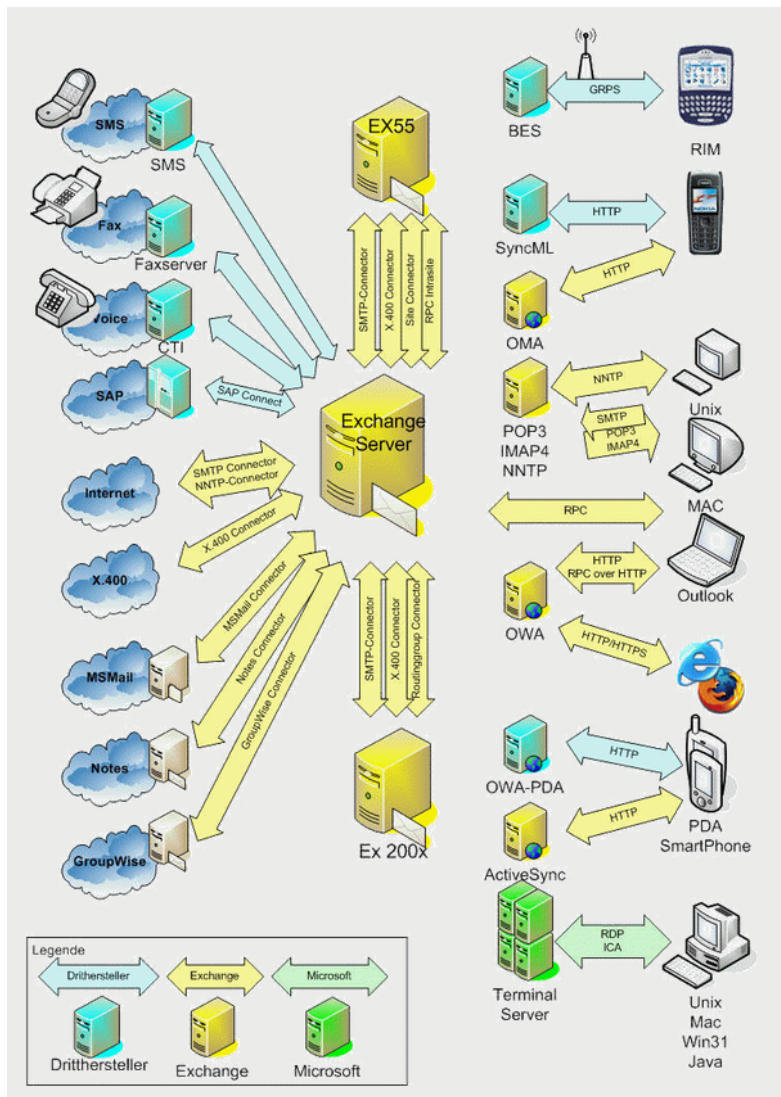
## **Anhangsverzeichnis**

# Anhang

Anhang 1

Blatt 1

## Einige Exchange Verbindungen



Quelle: <http://www.msxfaq.de/basics/excomm.htm>

## Übersicht verschiedener APIs

Jahr	APIs Quelle	Beschreibung
1992	■ SimpleMAPI 1.0 MS-Mail 3.0	Diese Schnittstelle wurde mit Microsoft Mail 3.0 erstmals veröffentlicht. Das war der Startschuss von MAPI und es war sicher nicht abzusehen, dass es so ein Langläufer wird. Damals gab es noch andere MAPI-Schnittstellen wie VIM (Vendor Independent Messaging von Lotus Notes, und CMC Common Messaging C??? von Novell/WordPerfect)
1995	■ Extended MAPI Windows95	Durch Windows 95 wurde auch der neue "Exchange Client" verfügbar, welcher nun der Nachfolger von MS-Mail sein sollte. Dieser Client wurde auch mit Exchange Server genutzt, ehe Outlook 97 dann den Wechsel bedeutet (Nicht mit dem Exchange Server) zu verwechseln.
1996	■ Directory API ■ Electronic Forms Designer ■ Exchange Developer Kit Gateway ■ OLE Messaging 1.0 (CDO 1.0) ■ Common Messaging Calls 1.0 Exchange 4.0	Exchange 4.0 hat nun auch im Backend die alte Microsoft Mail Datenbank (MAILDATA) ersetzt und entsprechend sowohl für den Server als auch auf dem Client neue APIs mitgebracht
1997	■ Active Messaging SDK 1.1 (CDO 1.1) Exchange 5.0	
1997	■ CDO 1.2 ■ CDO Rendering (CDOHTML) ■ Event Service Agents Exchange 5.5	Exchange 5.5 hat Exchange weiter für das Internet geöffnet. Erstmals konnte per "OWA" auf das Postfach zugegriffen werden. Zudem gab es nun auch den "Event Agent", welcher auf dem Server eingestellte Ordner überwachte und Skripte beim Eintreffen neuer Nachrichten ausführt.
1997	■ CDONTS Windows NT4 Option Pack	Durch das Windows NT4 Option pack war es nun mit dem darin enthaltenen IIS4 und dem SMTP-Server möglich, auch Mails mit Windows zu verarbeiten. CDONTS war dazu die erste API, um Nachrichten auf dem Server zu senden
1998	■ CDO 1.2.1 ■ Routing Objects ■ HTML Form Converter Exchange 5.5 SP1	
2000	■ Antivirus API (VSAPI) Exchange 5.5	In der Zeit hat sich Sybari den Grundstock für ihr Unternehmen gelegt, indem Sie einen Weg gefunden haben, Viren in Exchange zu finden und zu stoppen, ehe diese in der Datenbank landen. Erst durch die VSAPI, die Microsoft nachgereicht hat, konnten alle Hersteller dann über eine offizielle API gehen.
2000	■ CDO for Exchange 2000 (CDOEx) ■ CDO for Exchange Management (CDOExM) ■ CDO for Workflow ■ Exchange Installable File System ■ Exchange OLEDB provider (ExOLEDB) ■ WSS Forms ■ FrontPage Extensions ■ OWA URL commands ■ Backup and Restore API (a.k.a. ESEdbcli2) ■ Queue Viewer API ■ Store Events ■ WebDAV ■ WMI classes for Exchange management ■ Workflow Designer Exchange 2000	Das Jahr 2000 war auch der Start von Exchange 2000 und einer ganz neuen Plattform für Nachrichten und Zusammenarbeit. Entsprechend gibt es einen ganzen Satz neuer APIs die heute noch genutzt werden. Einige hingegen werden eher selten eingesetzt, z.B.: die Queue Viewer API
2000	■ CDO for Windows 2000 (CDOSYS) ■ SMTP Transport Events Windows 2000	Durch die Installation von Windows 2000 und dem darin enthaltenen SMTP-Server kommen ebenfalls einige APIs mit, die auch in Exchange vorhanden sind.
2001	■ Antivirus API 2.0 (VSAPI 2.0) Exchange 2000	
2003	■ Antivirus API 2.0 (VSAPI 2.3) Exchange 2003	Im Hinblick auf APIs hat Exchange 2003 nicht viel Neuerungen mitgebracht. Es ist ja auch eher eine kleine aber wichtige Weiterentwicklung von Exchange 2000.
2007	■ AutoDiscover ■ EWS ■ OAB per HTTP	Mit Exchange 2007 wurden erstmals nicht nur Autodiscover eingeführt sondern neben der "alten" WebDAV-Schnittstelle auch ein richtiger Webservice.
2015	■ RestAPI	Aktuell ist nur in Office 365 eine Vorabversion einer Rest-API auf die Inhalte eines Exchange Store verfügbar.

Quelle: <http://www.msxfaq.de/code/wege.htm>





## Literaturverzeichnis

Widl, M (2015): Microsoft Office 365. Das umfassende Handbuch, 3., aktualisierte Auflage, Bonn

Joos, T. (2013): Microsoft Exchange Server 2013 - Das Handbuch, Köln

<http://venturebeat.com/2014/01/22/help-desk-software-heres-some-of-the-best-and-most-interesting/>,  
22. Mai 2014

<http://web.appstorm.net/roundups/communication-roundups/10-online-support-and-help-desk-apps/>,  
20. März 2015

<http://www.capterra.com/help-desk-software/> , Oktober 2015

<http://www.pcmag.com/article2/0,2817,2489457,00.asp> , 22. März 2016