

# MAGISTERARBEIT

Titel der Magisterarbeit

## “Identifizierung des Paradigmas Software-as-a-Service für den Einsatz von CRM On Demand Software”

Verfasser

Georg Hauser, Bakk. rer. soc. oec.

angestrebter akademischer Grad

Magister der Sozial- und Wirtschaftswissenschaften  
(Mag. rer. soc. oec.)

Wien, im März 2009

Studienkennzahl lt. Studienblatt: A 066 926

Studienrichtung lt. Studienblatt: Wirtschaftsinformatik

Betreuerin / Betreuer: o. Univ. Prof. Dr. Dimitris Karagiannis

# Eidesstaatliche Erklärung

“Hiermit versichere ich, die vorliegende Magisterarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben, alle Stellen, die den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen”.

Wien, im März 2009

---

Georg Hauser

# Kurzfassung

Diese Arbeit thematisiert das an Application Service Providing (ASP) angelehnte Nutzungsmodell Software-as-a-Service (SaaS) als Grundlage zur Flexibilisierung der betrieblichen Handlungsweise von Customer Relationship Management (CRM), wobei die Transformationsmöglichkeiten der zentral zugänglichen CRM-Lösungen auf die individuellen Anforderungen kritisch zu hinterfragen ist.

Eine extern vorgehaltene und über den Browser nutzbare On Demand Software korreliert immer mehr mit Trends, wie der technologischen Vernetzung und der Virtualisierung der Leistungserstellung. Die Virtualisierung der Leistungserstellung resultiert daraus, dass die CRM-Bereitstellung aufgrund Kosteneffizienz (geringe Anfangsinvestitionen), Geschwindigkeit (keine Implementierung) und Flexibilität (einfacher Zugang) durch einen losen und zeitlich begrenzten Zusammenschluss mit einem Partnerunternehmen (CRM-Hersteller) erfolgt. Um dieses Ziel zu erreichen werden modulare Plattformen auf der Grundlage von serviceorientierten Technologien, um Techniken und Ansätze des Utility Computings erweitert. Dabei wird die Zielsetzung verfolgt SaaS in hoher Effizienz und unter Ausnutzung entsprechender Skaleneffekte erbringen zu können. Daher zeichnet das breite Spektrum an Betrachtungen bestehend aus IT-Sourcing und -Servicemanagement den vielseitigen Charakter für die Identifizierung des Paradigmas Software-as-a-Service aus. Eine Betrachtung der Begriffe rund um die daraus resultierende Softwarebereitstellung findet dabei ebenso statt wie eine Charakterisierung der technologischen und organisatorischen Rahmenbedingungen des On Demand Nutzungsmodells insbesondere unter Berücksichtigung des Konzepts der Multitenancy. Für die kritische Auseinandersetzung werden die Anforderungen hinsichtlich der Anwendungsmöglichkeiten sowie die Customizingmöglichkeiten von CRM On Demand als Analyseschema zugrunde gelegt und mit den Dimensionen des Begriffs Software-as-a-Service gekoppelt.

# Danksagung

An dieser Stelle möchte ich meinen Dank an all diejenigen aussprechen, die mich während der Verwirklichung dieser Arbeit behilflich waren und mir immer wieder wertvolle Tipps und Anregungen gegeben haben. Vor allem bin ich o. Univ.-Prof. Dr. Dimitris Karagiannis für die akademische Betreuung zum großen Dank verpflichtet.

Die Idee für diese Arbeit ist dem Projekt - “CRM On Demand in Grossunternehmen” entnommen, welches ich am Institut für Wirtschaftsinformatik an der HSG St. Gallen durchgeführt habe. Mein besonderer Dank gilt hier Susanne Glissmann als betreuende Mitarbeiterin, die mir ebenfalls mit akademischen Rat zur Seite stand und so zum Gelingen der Arbeit wesentlich beigetragen hat.

Ein besonderer Dank gilt auch allen Mitgliedern des IWI-Teams, insbesondere Florian Schnabel, Stefan Reitbauer, Bernhard Schindlholzer, Malte Dous, Alexander Vogedes und Holger Schmalz, die mich sowohl physisch als auch psychisch über mehrere Monate hinweg in St. Gallen begleitet haben. Weiters gebührt großer Dank an die SAP AG, Salesforce.com, Oracle und RightNow Technologies, die durch ihre Kooperation einen wesentlichen Beitrag zum Gelingen dieser Arbeit beigetragen haben.

Schlussendlich möchte ich noch meine Eltern für die Unterstützung während meiner gesamten Ausbildungszeit herzlichst danken. Nicht zu vergessen sind auch all meine Freunde, die mich immer wieder motiviert und inspiriert haben.

---

*“Lesen Sie schnell, denn nichts ist beständiger als der  
Wandel im Internet” [Anita Berres]<sup>1</sup>*

---

---

<sup>1</sup>Wenige Jahre nach dem Zusammenbruch der New Economy durch das Platzen der “Dotcom-Blase” hat eine grundlegende Veränderung in der Wahrnehmung des Webs stattgefunden [[o.V06b](#)].

# Abkürzungsverzeichnis

AICPA	American Institute of Certified Public Accountants
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
ASAP	Accelerated SAP
ASP	Application Service Providing
ASPIC	ASP Industry Consortium
B2B	Business-To-Business
B2C	Business-To-Customer
BPEL4WS	Business Process Execution Language for Web Services
CMDB	Configuration Management Database
CRM	Customer Relationship Management
CSS	Cascading Syle Sheets
CSV	Comma-Separated Values
CTI	Computer Telephony Integration
DBMS	Datenbankmanagementsystem
DOM	Document Object Model
DRBD	Distributed Replicated Block Device
DWH	Data Warehouse
EAI	Enterprise Application Integration
EJB	Enterprise Java Beans
ERM	Employee Relationship Management
ERP	Enterprise Ressource Planning
ESB	Enterprise Service Bus

---

ETL	Extract, Transform, Load
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
i.d.R.	in der Regel
ICA	Independent Computing Architecture
IPSec	Internet Protocol Security
ISV	Independent Software Vendor
ITAA	Information Technology Association of America
ITIL	IT Infrastructure Library
JSR	Java Specification Request
MTD	Multi Tenant Data
OASIS	Organization for the Advancement of Structured Information Standards
OLAP	On Line Analytical Processing
PDA	Personal Digital Assistant
PKI	Public-Key-Infrastructure
PRM	Partner Relationship Management
QoS	Quality of Services
RDP	Remote Desktop Protocol
RIAs	Rich Internet Applications
ROI	Return on Investment
SaaS	Software-as-a-Service
SAS 70	Statements on Auditing Standards Nr. 70
SCM	Supply Chain Management
SDL	Salesforce Data Loader
SOAP	Simple Object Access Protocol
SOC	Service Oriented Computing
SOD	Software On Demand
SSL	Secure Socket Layer
TCO	Total Cost of Ownership

---

UMTS	Universal Mobile Telecommunications System
VPN	Virtual Private Network
WSDL	Web Services Description Language
WSDM	Web Service Distributed Management
WSLA	Web Service Level Agreements
XHR	XMLHttpRequest
XHTML	Extensible Hypertext Markup Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformation



# Tabellenverzeichnis

2.1. Systematisierung der Erlösformen [Sch04, S. 12] . . .	9
2.2. Systemverfügbarkeiten in % [Köh07b, S. 128] . . . . .	31
3.2. Gegenüberstellung On Premise und On Demand . . .	39
3.3. ASP Nutzen aus Kundensicht [Die02, S. 96] . . . . .	46
3.5. Unterschiede ASP - SaaS . . . . .	49
4.2. Erfolgsfaktoren von SaaS [Gun07] . . . . .	89
5.1. Auswahlkriterien für ASPs bei Geschäftskunden [KSvHB03, S. 51] . . . . .	94
5.3. Anbieter von CRM On Demand . . . . .	96
5.4. Untersuchungsgegenstand . . . . .	97
5.5. Kennzahlen zu RightNow Technologies . . . . .	99
5.6. Kennzahlen zu Salesforce.com [o.V08a] . . . . .	103
5.7. Kennzahlen zu Siebel CRM On Demand . . . . .	109
C.2. Varianten zur Gewährleistung des Datenschutzes [BIT06c, S. 64 ff.] . . . . .	143

# Abbildungsverzeichnis

1.1. Modell zur Identifizierung von SaaS für CRM On Demand . . . . .	4
2.1. Entwicklung der Unternehmensarchitekturen [Mas05, S. 41] . . . . .	10
2.2. N-Tier Architektur [Fac02, S. 71] . . . . .	12
2.3. Struktur des Vertragsmodell . . . . .	18
2.4. Koordinationsformen zwischen Software und Anwender [Küh07, S. 3] . . . . .	25
2.5. Multi-Tenant Plattform [GSH <sup>+</sup> 07, S. 551] . . . . .	30
2.6. Customizingwizard [GSH <sup>+</sup> 07, S. 557] . . . . .	32
2.7. Webservice Technologie . . . . .	35
3.1. Monetäre Aspekte und Monitoring von Utility Management Services . . . . .	52
3.2. Verschiedene Möglichkeiten von Multi Tenant Datenbanken [JA07] . . . . .	53
3.3. relationales Multi Tenant Datenbankmodell [C <sup>+</sup> 06] . . . . .	55
3.4. Service Delivery und Service Support nach ITIL . . . . .	59
3.5. asynchrones Interaktionsmuster bei SaaS [Bos07, S. 39 ff.] . . . . .	60
4.1. Funktionsübersicht CRM [Sch07a, S. 32] . . . . .	70
4.2. Architektur der operativen CRM-Prozesse [AW04, Sch05] . . . . .	71
4.3. Business Intelligence im CRM [Gei06, S. 127 ff.] . . . . .	75
4.4. Referenzarchitektur für Portal-Software [GH03, S. 160] . . . . .	78
4.5. Nutzung von CRM On Demand . . . . .	82
4.6. Überblick über CRM On Demand . . . . .	83
4.7. Nutzeffekte von CRM On Demand . . . . .	87
5.1. Multi-Tenant Plattform von Salesforce.com [o.V08b] . . . . .	104
6.1. CRM im SaaS-Modell . . . . .	118

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Einführung in das Themengebiet . . . . .	1
1.2. Problem- und Zielstellung . . . . .	2
1.3. Aufbau der Arbeit . . . . .	4
<b>2. Kontext der Arbeit</b>	<b>7</b>
2.1. Internetbasierte Geschäftsmodelle . . . . .	7
2.1.1. Anwendungsarchitektur . . . . .	10
2.1.2. Schichtenmodell von webbasierten Systemen . . . . .	11
2.1.3. SOA - serviceorientierte Architekturen . . . . .	14
2.2. IT-Sourcing . . . . .	15
2.2.1. Vertragsmodell . . . . .	18
2.2.2. Gründe für Outsourcing . . . . .	20
2.2.3. Kritische Merkmale des Outsourcings . . . . .	22
2.3. Begriffsbestimmung Softwarebereitstellung . . . . .	23
2.3.1. Koordinationsformen zwischen Software und Anwender . . . . .	24
2.3.2. Sicherheitsrelevante Anforderungen an den Anbieter . . . . .	26
2.3.3. Eigenschaften einer Multi Tenant Plattform . . . . .	29
2.3.4. Customizing der ausgelagerten Software . . . . .	31
2.3.5. Möglichkeiten der Softwareintegration . . . . .	33
2.3.6. SOAP-basierte Kopplung der Software . . . . .	34
2.4. Konstituierende Merkmale und Implikationen . . . . .	36
<b>3. On Demand-Prinzipien als Geschäftsmodell</b>	<b>38</b>
3.1. Begriffsbestimmung: Software On Demand . . . . .	38
3.1.1. Vorteile gegenüber klassischen Ansätzen . . . . .	40
3.1.2. Nachteile von On Demand Software . . . . .	42
3.2. Markt im On Demand Umfeld . . . . .	43
3.2.1. Entstehung und Verbreitung von ASP . . . . .	43
3.2.2. Einordnung von Software-as-a-Service und Application Service Providing . . . . .	47
3.3. SOC als Basis für On Demand Computing . . . . .	49
3.3.1. Multi Tenant Data (MTD) - Architektur . . . . .	52
3.3.2. On Demand Computing und ITIL . . . . .	55
3.3.3. Rich Internet Applications . . . . .	59
3.4. SaaS - Software-as-a-Service . . . . .	62

3.4.1. Erfolgsfaktoren für SaaS . . . . .	65
3.4.2. Mögliche Problemfelder . . . . .	67
<b>4. Die Komponenten eines CRM-Systems</b>	<b>69</b>
4.1. Grundlagen CRM . . . . .	69
4.2. Operatives CRM . . . . .	70
4.2.1. Marketing Automation . . . . .	72
4.2.2. Sales Automation . . . . .	73
4.2.3. Service Automation . . . . .	74
4.3. Analytisches CRM . . . . .	75
4.3.1. Data Warehouse . . . . .	75
4.3.2. Online-Analytical-Processing . . . . .	76
4.3.3. Data Mining . . . . .	76
4.4. Kollaboratives CRM . . . . .	76
4.4.1. Internet-Portale . . . . .	77
4.4.2. Portal Referenzarchitektur . . . . .	77
4.5. Risiken und kritische Merkmale eines CRM On Premise Betrieb	79
4.5.1. CRM-Software im globalen Umfeld . . . . .	80
4.5.2. Erfolgsfaktoren für CRM On Demand . . . . .	81
4.5.3. Anforderungen an CRM On Demand . . . . .	83
4.5.4. SaaS als Grundlage für CRM On Demand . . . . .	87
<b>5. Analyse von CRM On Demand Software</b>	<b>90</b>
5.1. Zielstellung der Analyse und verwendetes Anwendungsszenario .	90
5.1.1. Anforderungsszenario als Grundlage für die Analyse . . .	91
5.1.2. Begleitende Aspekte bei der Untersuchung . . . . .	93
5.1.3. Vorgehensweise bei der Analyse . . . . .	95
5.2. Untersuchungsgegenstand . . . . .	98
5.2.1. RightNow Technologies . . . . .	98
5.2.2. Salesforce.com . . . . .	102
5.2.3. Siebel CRM On Demand . . . . .	108
5.3. Zusammenfassung der Analyse . . . . .	113
<b>6. Fazit und Ausblick</b>	<b>118</b>
<b>Literaturverzeichnis</b>	<b>122</b>
<b>A. Entwicklung des Outsourcings</b>	<b>139</b>
<b>B. On Premise Einführung</b>	<b>141</b>
<b>C. Varianten zur Gewährleistung des Datenschutzes</b>	<b>143</b>

# 1. Einleitung

Mit zunehmend günstigeren Rahmenparametern, wie breitbandiges Internet und etablierten Sicherheits- und Internettechnologien, avancierte das Internet zu einem elektronischen Distributionskanal für die Übertragung von Softwarefunktionalitäten als Dienstleistung. Somit führen webbasierte Dienste zu einer neuen Ausrichtung des Leistungserstellungs- und angebotsmodells. Diese Arbeit soll einerseits einen konzeptionellen Ansatz für die Transformation von betriebsrelevanten Softwareprodukten zu internetbasierten Services entwickeln, um andererseits die theoretische Basis für Software-as-a-Service als Grundlage von CRM On Demand bieten.

## 1.1. Einführung in das Themengebiet

Was im Zuge der Kommerzialisierung des Internets (1999 bis 2001) als Application Service Providing (ASP) bereits einen bemerkenswerten Aufschwung mit hohen Wachstumsprognosen erfuhr, wird nun als Software-as-a-Service (häufig in den IT-Medien als SaaS abgekürzt) wieder neu aufgegriffen [VH07, S. 256]. Wichtig für diese Entwicklung ist u.a. die stärkere Verbreitung von breitbandigen Internetzugängen, aber auch die immer stärker veränderten Internet-Nutzungsgewohnheiten. Deshalb lässt sich das SaaS-Modell als eine interessante Alternative zum klassischen Lizenzmodell betrachten. IT-Analysten glauben, dass Software-as-a-Service ein Milliardenmarkt werden könnte [Vol06]. Auch die Gartner Group prognostiziert, dass bis 2011 insgesamt 25 Prozent der gesamten Unternehmenssoftware auf Basis des SaaS-Modells entwickelt werden wird [DHAL06, S. 2].

Alle großen Softwarehersteller<sup>1</sup> - von Oracle über SAP bis zu Microsoft - arbeiten verstärkt an entsprechenden Lösungen und vermarkten das SaaS-Modell im Rahmen ihres eigenen Portfolios. Bei Microsoft läuft das Programm unter dem Namen "SaaS on Ramp", SAP vermarktet es unter dem Namen "SAP Business ByDesign", Oracle nennt es "Oracle On Demand" und bei IBM läuft die SaaS-Initiative unter den Namen "Software as a Service". Die CRM-Studie 2007 belegt darüber hinaus, dass zunehmend standardisierte CRM-Lösungen als SaaS auf den Markt kommen werden, da der Anteil der CRM-

---

<sup>1</sup>2006 kündigten z.B. Microsoft als auch SAP an, zusätzlich zum clientfokussierten Geschäftsmodell, neue CRM-Lösungen als browserbasierte Services im On Demand Modell anzubieten.

Anbieter mit On Demand Lösungen von 55,8 % in 2006 auf 65,8 % in 2007 stieg [HHW07, S. 40].

Das Erkenntnisinteresse gilt somit dem IT-gestützten Customer Relationship Management, dass nicht physisch in die vorhandene IT-Infrastruktur des Anwenders eingebettet ist, sondern “on demand” nach dem Uno-actu-Prinzip<sup>2</sup> und nach der Bezahlung einer definierten Nutzungsgebühr genutzt werden kann, wobei die Transformationsmöglichkeiten der zentral zugänglichen CRM-Lösungen auf die individuellen Anforderungen kritisch zu hinterfragen ist. Die dafür notwendigen Bereitstellungs- und Softwarearchitekturen und ihr Einfluss auf die Serviceerbringung werden daher als wichtige Merkmale identifiziert, um die CRM-Software - als eine Form des “outside resource using” - zur Deckung betrieblicher Anforderungen einsetzen zu können.

## 1.2. Problem- und Zielstellung

Im Mittelpunkt dieser Arbeit steht die ganzheitliche Betrachtung einer extern vorgehaltenen, entkoppelt von systemspezifischen Abhängigkeiten und als Service nutzbaren CRM-Software. Diese Sichtweise der Softwarenutzung grenzt sich von einem Softwaredownload mit anschließender lokaler Installation ab und stellt damit ein (aus der Sicht eines Kunden) Outsourcing<sup>3</sup> der Bereitstellung von Anwendungen bzw. ein (aus der Sicht des Anbieters) Geschäftsmodell für die Bereitstellung und den Vertrieb von Anwendungen über das Internet dar. Die als zentrale Unternehmensfunktionen empfundene Anwendungen, wie das IT-gestützte Customer Relationship Management über das Internet für eine Nutzungsgebühr zu beziehen, wirft somit folgende Frage auf.

*Welche organisatorischen und technologischen Anforderungen müssen Softwareanbieter respektive Anwendungsvermieter für die Realisierung des Nutzungsmodells Software-as-a-Service berücksichtigen, damit ein bedarfsgerechtes Angebot für die flexible Softwarebereitstellung ermöglicht werden kann? Zudem stellt sich die Frage, wodurch sich ausgewählte CRM On Demand Lösungen hinsichtlich des integrierten Customizingrahmens unterscheiden?*

Es gibt zahlreiche Kritiker, die im SaaS-Modell die Wiedergeburt von Application Service Providing (ASP) als Software-as-a-Service ([Nau05]; [HR07b]) sehen. Als signifikanter Grund werden hierfür die mit viel Verve gestarteten

---

<sup>2</sup>Die CRM-Software welche als Service angeboten wird ist dadurch gekennzeichnet, dass diese simultan zur Erbringung auch sofort genutzt werden kann [Hal07, S. 74].

<sup>3</sup>Outsourcing wird als Oberbegriff für alle Leistungen verwendet, die extern erbracht oder als Dienstleistung bezogen werden. Im Kontext dieser Arbeit fokussiert sich Outsourcing ausschließlich auf Dienstleistungen aus dem Umfeld der Informationsverarbeitung, also auf IT-Outsourcing.

ASP-Dienstleister während der New Economy genannt, die Unternehmenssoftware webtauglich machten, ohne die Chancen und Grenzen der Softwarebereitstellung zu beachten [App02, S. 53].

- Zum einen wird der Individualisierungsgrad als größtes Hindernis angesehen, da eingeschränkte Möglichkeiten der Adaption an branchenspezifische Rahmenbedingungen den Verlust von Differenzierungsmerkmalen auf sich ziehen kann und mögliche Schwierigkeiten bei der Integration zu redundanter Datenhaltung führen können.
- Zum anderen kann aufgrund geringer Kontrollmöglichkeiten nur schwer nachvollzogen werden, ob die zentralen Faktoren Sicherheit wie auch die Erfüllung gesetzlicher Auflagen gewährleistet werden. Darunter fallen Systemausfallsrisiko, Ausfallsrisiko<sup>4</sup>, das Risiko des Datenverlustes durch Organisationsversagen, wiederrechtliche Weitergabe an Dritte und das Risiko der Abhängigkeit von einem Anbieter während der Vertragslaufzeit.

Es erscheint daher als zweckmäßig, die Fragestellungen zunächst im Kontext des ASP-Modells einzuordnen, um die Basis für eine weitergehende Betrachtung von Software-as-a-Service zu schaffen. Es sind der zeitliche Kontext und die technischen Rahmenbedingungen - wie beispielsweise Internetnutzung und Bandbreite zu beachten. Weiters sind auch betriebswirtschaftliche und soziale Aspekte, beispielsweise Zielsetzung, Verlustängste und Sicherheitsaspekte, mit in die detaillierte Evaluation aufzunehmen [Hal07, S. 76].

Deshalb werden zunächst allgemeine Anforderungen wie Verfügbarkeit, Sicherheit und Anpassungsfähigkeit einer solch gesourceten Leistung unter Berücksichtigung der Webfähigkeit der Software betrachtet und als Grundlage für die internetbasierte Bereitstellung<sup>5</sup> festgehalten. Um dabei den in der Literatur beschriebenen Unterschieden zwischen SaaS und ASP Rechnung zu tragen, werden die Merkmale des Service Oriented Computing (SOC) herausgestellt, um damit das Anwendungs- und Substitutionspotential von SaaS aufzeigen zu können.

Ferner wird ein Kriterienkatalog zur Analyse und Bewertung von On Demand Lösungen herangezogen, um der Frage nachzugehen, in welchem Ausmaß die Parametrisierung ausgewählter CRM On Demand Angebote durchgeführt werden kann. Ziel dieses abschließenden Teils ist es, Merkmale zu definieren, anhand denen die als SaaS realisierte Lösungen im Speziellen von Salesforce.com, RightNow Technologies und Siebel CRM On Demand, beurteilt werden

---

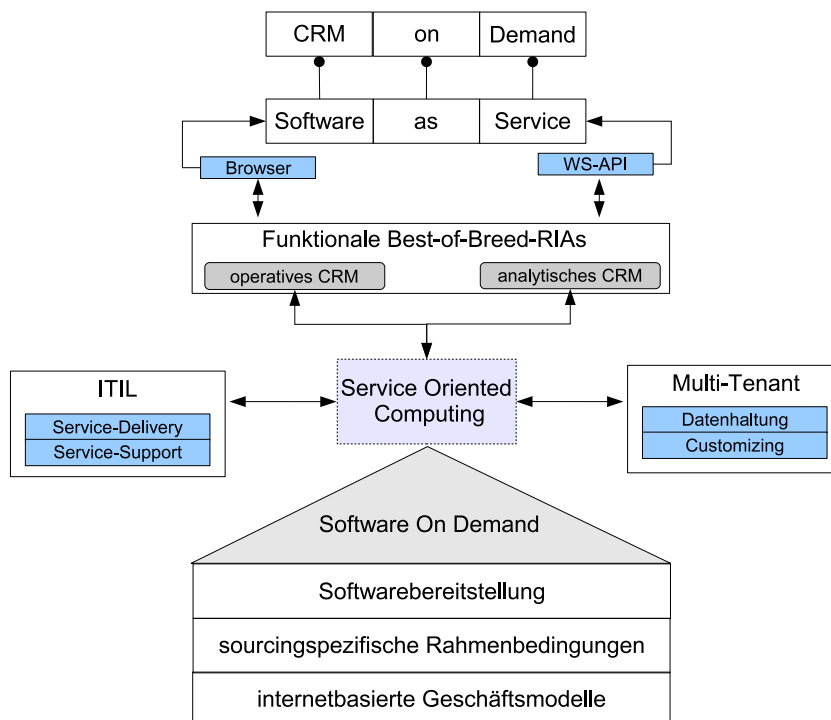
<sup>4</sup>Ist das Ausscheiden des Anbieters vom Markt aufgrund finanzieller Probleme.

<sup>5</sup>Die internetbasierte Bereitstellung umfasst dabei technische und organisatorische Aufgaben, damit die Software in einer skalierbaren, sicheren und angepassten Art und Weise über das Web als Service genutzt werden kann.

können. Dabei werden ausgewählte CRM On Demand Lösungen hinsichtlich der Anwendungsmöglichkeiten und des Customizings analysiert. Die analysierten Informationen - hinsichtlich Anforderungen an die Softwarebereitstellung, Optionen, wie die Software beim Kunden zur Anwendung kommt sowie Formen der Integrationen und Customizing - dienen wiederum für die Aufarbeitung der Informationsasymmetrien zwischen SaaS und ASP.

### 1.3. Aufbau der Arbeit

Von dem zuvor beschriebenen Hintergrund ist das Ziel der vorliegenden Arbeit, mit Hilfe einer Analyse der konstituierenden Merkmale und Implikationen im Kontext der internetbasierten Bereitstellung die impliziten Prämissen für SaaS zu verdeutlichen und unter Berücksichtigung festgelegter Anforderungen, die erwähnten Aspekte kritisch zu hinterfragen und zu prüfen.



**Abbildung 1.1.:** Modell zur Identifizierung von SaaS für CRM On Demand

Dabei bilden internetbasierte Geschäftsmodelle, sowie deren technologische Basis die Grundlage für die Transformation von Softwareprodukten in Form der Software-Services (die als Softwarekomponenten z.B. als Webservices implementiert sein können), die Ausgangsbasis für die systematische Identifizierung von Software-as-a-Service. Diese Idee leistet dem Outsourcing entsprechend Vorschub, da die im Internet verfügbaren Software-Services im Sin-



ne eines Outsourcings integriert werden. Deshalb werden neben den formellen Grundlagen für die Festlegung der konkreten Rahmenbedingungen solcher Sourcingleistungen die Gründe, aber auch die kritischen Merkmale hervorgehoben, um die konstituierenden Merkmale für die anforderungsgerechte Softwarebereitstellung abzuleiten und zu diskutieren. Aufgrund dessen werden im Anschluss dieses Kapitels die Merkmale und Aspekte der Softwarebereitstellung zusammengefasst, um mit einer Einordnung im Kontext des On Demand Modells eine Basis für eine weitergehende Betrachtung von SaaS zu schaffen.

In Kapitel 3 wird mit der Beschreibung des On Demand Geschäftsmodells eine Grundlage für die weitergehende Betrachtung der Softwarebereitstellung geschaffen. Aufgrund dessen, dass der Markt im On Demand Umfeld durch mehrere technologische Entwicklungen geprägt wurde, die mitunter andere Paradigmen verfolgten, wird zunächst die Entstehung und Verbreitung von ASP diskutiert, um in weiterer Folge eine Einordnung von SaaS und ASP vorzunehmen. Dabei werden die im Kapitel 2 erarbeiteten Merkmale der servicezentrierten Softwarebereitstellung im Vergleich denen, des im Allgemeinen als ASP bezeichneten Modells, der anwendungszentrierten Softwarebereitstellung gegenübergestellt. Die anwendungszentrierte Bereitstellung wird hier als eine dezidierte Softwarebereitstellung dargestellt und die servicezentrierte Bereitstellung als ein Servicezugriffsmodell, bei dem Unternehmen die Software über die IT-Infrastruktur des Anbieters nutzen. Deshalb wird eine durch das Service Oriented Computing realisierbare Architekturvariante, auf Basis der Multi-Tenant Architektur, für die Abbildung von mehreren Kunden innerhalb der Software, als erweiterte Gestaltungsoption für die Individualisierung und Optionierung von Software-Services, herangezogen. Dabei werden die Facetten des Service Oriented Computing<sup>6</sup> für die Zerlegung der Systemfunktionalitäten in einzelne Dienste herangezogen, damit die Service-Instanzen der MTD-Architektur - unabhängig des auftretenden Lastaufkommens sowie nach vereinbarten Rahmenbedingungen und Garantien - zwischen Serviceanbieter und -nutzer (Kunde) bereitgestellt werden können. Dies erfordert nicht nur ein automatisiertes Monitoring der zugesicherten Qualitätseigenschaften, sondern auch Managementfunktionalitäten für das Versions-, Konfigurations- und Änderungsmanagement in Korrelation mit dem User-Support. Daher wird mit einer prozessbezogenen Ausrichtung des IT-Servicemanagements - sowohl auf die bereitgestellte Software als auch auf die Nutzersicht, wie dies durch ITIL vorgegeben wird - eine gemeinsame Basis für die Bereitstellung geschaffen. Dabei werden innerhalb des Service-Support-Bereiches die Service-Prozesse behandelt, die bei Abruf eine geschäftliche Wertschöpfung beim Kunden erzielen. Dies betrifft auch komplementäre Dienste wie Support durch einen IT-Helpdesk.

---

<sup>6</sup>“Service Oriented Computing (SOC) ist ein Paradigma, das Services als fundamentales Element für die Erstellung von Applikationen verwendet” [Lie07, S. 8].

Da in den vergangenen Jahren webbasierte Anwendungen den Beigeschmack einer umständlichen Bedienung hatten, werden in weiterer Folge Technologien beschrieben, die die Mechanismen von Desktop- und Webanwendungen vereinen (Stichwort: Rich Internet Applications) und den Anwendungsbereich von Webanwendungen erweitern. So verfügen RIA-Frontends zum Beispiel über den Bedienungskomfort und die Performance einer Desktopanwendung. Die zugrundeliegenden Technologien für die Realisierung von RIA-Frontends werden deshalb als ein weiteres Merkmal zur Identifikation von Software-as-a-Service behandelt, damit die Funktionalitäten einer geschäftskritischen Software geräteunabhängig und direkt durch die Nachfrage im gleichen Umfang wie eine desktopbasierte Software, genutzt werden können. Durch die zugrundeliegenden Technologien des On Demand Computing wird nun die Software in ein servicebasiertes Angebot transformiert, das sich leicht an neue Situationen (z.B. neue Kooperation von Unternehmen) anpassen lässt und die Nutzerfunktionen in einen wirtschaftlichen Kontext stellt [S<sup>+</sup>03, S. 20]. Ferner bedarf es einer stärkeren Integration von betriebs-technischen Aspekten: sogenannte APIs (Application Programming Interfaces). Besonders wichtig sind hierfür Standards für einen sicheren Datentransfer sowie für den Austausch von Daten- und Businessobjekten. SOAP basierte Webservices stellen hierfür ein bereits etabliertes Mittel dar, da sie nicht nur das bekannte Internetprotokoll HTTP zur Übertragung verwenden, sondern auch eine wesentlich geringere Komplexität aufweisen als klassische Middleware wie CORBA oder DCOM.

Die in Kapitel 2 und 3 aufgezeigten Darstellungen implizieren in weiterer Folge, unter Berücksichtigung des Konzept der Multitenancy, die technologischen und organisatorischen Rahmenbedingungen für Software-as-a-Service. In Kapitel 4 werden dabei, ausgehend von den eingangs gestellten Fragen, die Dimensionen des Begriffs Software-as-a-Service für die weitergehende Betrachtung des Customer Relationship Management dargestellt. Folglich werden in Kapitel 5 die Stärken und Schwächen von CRM On Demand, im Speziellen von Salesforce.com, RightNow Technologies und Siebel CRM On Demand evaluiert, um einen Überblick über die unterschiedlichen Customizingrahmen zu erhalten.

## 2. Kontext der Arbeit

Das intendierte Ziel dieser Arbeit ist dem steigenden Investitionsbedarf bei der Entwicklung und Erhaltung von CRM-Systemen durch netzbasierte Bereitstellungskonzepte entgegenzusteueren. Als Sonderform des IT-Sourcings stellt das Outsourcing, unter dem weitreichenden Aspekt der räumlich und zeitlich unabhängigen Nutzung von IT-Lösungen auf Basis von CRM-Standardsoftware, den Rahmen dieser Arbeit dar. Das folgende Kapitel beschäftigt sich daher mit der organisatorischen Verlagerung von Anwendungen auf externe Unternehmen, das als Einflussfaktor für die Modelltheorie von CRM On Demand dient.

### 2.1. Internetbasierte Geschäftsmodelle

Die Besonderheit eines internetbasierten Geschäftsmodells besteht darin, dass die Marktleistung überwiegend mit Internettechnologien realisiert wird [Hal07, S. 76]. Dabei kann die Leistungserstellung analog zum Modell des Wertschöpfungsprozesses für Dienstleistungen in folgende drei Phasen differenziert werden [Hal07, S. 78 ff.]:

- Erstellung der Leistungsbereitschaft,
- Leistungsvereinbarung und
- Leistungserbringung.

Die Erstellung der Leistungsbereitschaft umfasst im Wesentlichen die Entwicklung und Bereitstellung einer Software, auf Basis von Webtechnologien. Eine Einschränkung der relevanten Marktteilnehmer ist nur schwer möglich, da sich theoretisch jegliche Form standardisierter Software für eine heterogene Kundengruppe als webbasierter Service anbieten lässt. Hierzu lassen sich die Anwendungsbereiche in die drei Kategorien “einfach”, “umfassend” und “komplex” einteilen [Nol06, S. 194].

**Einfache** Anwendungsfälle sind solche, die fertig vorkonfiguriert direkt über das Internet nutzbar sind und keinen oder kaum individuellen Anpassungen bedürfen, wie z.B. eine webbasierte Textverarbeitung [Nol06, S. 198].

**Umfassende** Szenarien treten beispielsweise in Form von umfangreichen Variationen aus dem Bereich der einfachen Anwendungen oder als Teilmodule von komplexen Lösungen auf [Nol06, S. 201]. Eine einfache Trennung der Benutzerdaten durch einen Login reicht hier nicht mehr aus, da zur Erfüllung der Aufgaben ein gewisses Maß an Parametrisierungsmöglichkeiten<sup>1</sup> meist Voraussetzung ist (z.B. für einfache CRM-Systeme).

**Komplexe** Lösungen bedürfen einer intensiven Betreuung durch den Anbieter, da die Anwender bei zunehmender Komplexität oft auf kompetente Hilfestellung bei der Einführung und dem Betrieb angewiesen sind [Nol06, S. 202]. Zu den komplexen Lösungen zählen z.B. Anwendungen mit einem integrierenden oder aggregierenden Charakter, wie ERP-Systeme.

Die Umsetzung solcher Geschäftsmodelle ist daher mit der Herausforderung verbunden ein geeignetes Marktsegment zu finden (bzw. zu definieren), in dem genügend Interessenten angesprochen, eine geeignete Finanzierung des Angebots sichergestellt und ein Wettbewerbsvorteil gegenüber anderen Anbietern klar kommuniziert werden kann [Hal07, S. 78]. In weiterer Folge sind im Rahmen der Erstellung der Leistungsbereitschaft auch die Betriebs- und Wartungsarbeiten, sowie die Weiterentwicklung der Software zu berücksichtigen. Die Leistungsvereinbarung erfolgt in der Regel in Form einer Anmeldung, bei der die relevanten Informationen in Abstimmung mit den jeweiligen Kunden festgehalten werden (▷ vgl. Abschnitt: 2.2.1). Die Leistungserbringung erfolgt dann durch die möglich Nutzung der Software. Dabei ist, das Leistungsangebotsmodell als Folge der Standardisierung der Software, nur geringfügig ausgeprägt. Durch Verknüpfungen verschiedener Funktionalitäten kann jedoch eine höhere Werthaltigkeit erzielt werden (z.B. die Verknüpfung mit Google-Diensten). Hierbei sind die Möglichkeiten des Leistungserstellungs- und angebotsmodells auf die verwendete Applikationsarchitektur samt Hostinginfrastruktur zurückzuführen, da der Aufbau (“Architektur”), der im Rahmen dieses Geschäftsmodells angebotenen Software und die zugrundeliegende Hostingarchitektur bestimmt, inwieweit die laufende Anpassung der bestehenden Funktionalitäten überhaupt erfüllbar ist.

Ebenso werden - durch die zunehmende Standardisierung der Browser-Software für die Nutzung des DOM<sup>2</sup> (Document Object Model) sowie über das Eventmodell von JavaScript - Designkonzepte von Desktop-Anwendungen auf web-basierte Anwendungen übertragen, um die Interaktionsfunktionalitäten einer Desktop-Anwendung mit der hohen Reichweite von Webanwendungen zu kombinieren. So sind gewohnte Funktionen (wie z.B. “Drag-and-Drop”, Plausi-

---

<sup>1</sup>Für die Umsetzung kundenspezifischer Anforderungen benötigt man Möglichkeiten für die Parametrisierung der zugrundeliegenden Datenbank, um die Abbildung der Organisationsstruktur samt Workflows berücksichtigen zu können.

<sup>2</sup>Ein XML-Baum, der zum Rendern des Seiteninhaltes genutzt wird.

bilitätsprüfungen von Nutzereingaben und Autocomplete) an der Benutzerschnittstelle mit gleichzeitig verringerten Ladezeiten möglich. Durch die Unterstützung verschiedener Browser können unterschiedlichen Endgeräte, also beispielsweise auch auf PDAs oder UMTS Handys, auf die gewünschten Anwendungen und Daten zugreifen [PBW00, S. 45]. Außerdem erscheinen die Folgekosten für Wartung und Pflege nahezu vernachlässigbar, da lediglich die Clients instand zu halten sind. Da neben der webbasierten Software auch die Daten auf webbasierten Speichermedien, anstelle lokaler Festplatten, gespeichert werden, entfallen zudem Transaktions- und Synchronisationskosten [BMW07, S. 34].

Diese internetbasierten Angebote korrelieren daher immer mehr mit dem Paradigma “Web as a Platform” [BMW07, S. 30 f.]. Dabei kann festgestellt werden, dass eine Aufteilung von transaktionsabhängigen und transaktionsunabhängigen Erlösen eine mögliche Form der Erlösgenerierung in internetbasierten Geschäftsmodellen darstellen kann. In folgender Tabelle soll ein systematischer Überblick über das direkte Erlösmodell gegeben werden.

	<b>Direkte Erlöse</b>
<b>Transaktionsabhängig</b>	<ul style="list-style-type: none"> <li>- Anteil an Geschäftsvolumen</li> <li>- Transaktionsgebühr</li> <li>- Anteil einer Einsparung</li> <li>- Erfolgsprovision</li> </ul>
<b>Transaktionsunabhängig</b>	<ul style="list-style-type: none"> <li>- Einrichtungsgebühr</li> <li>- Grundgebühr/Mitgliedsgebühr</li> <li>- Datenvolumengebühr</li> <li>- Zeiteinheitsgebühr</li> <li>-Gebühr pro Zugang</li> </ul>

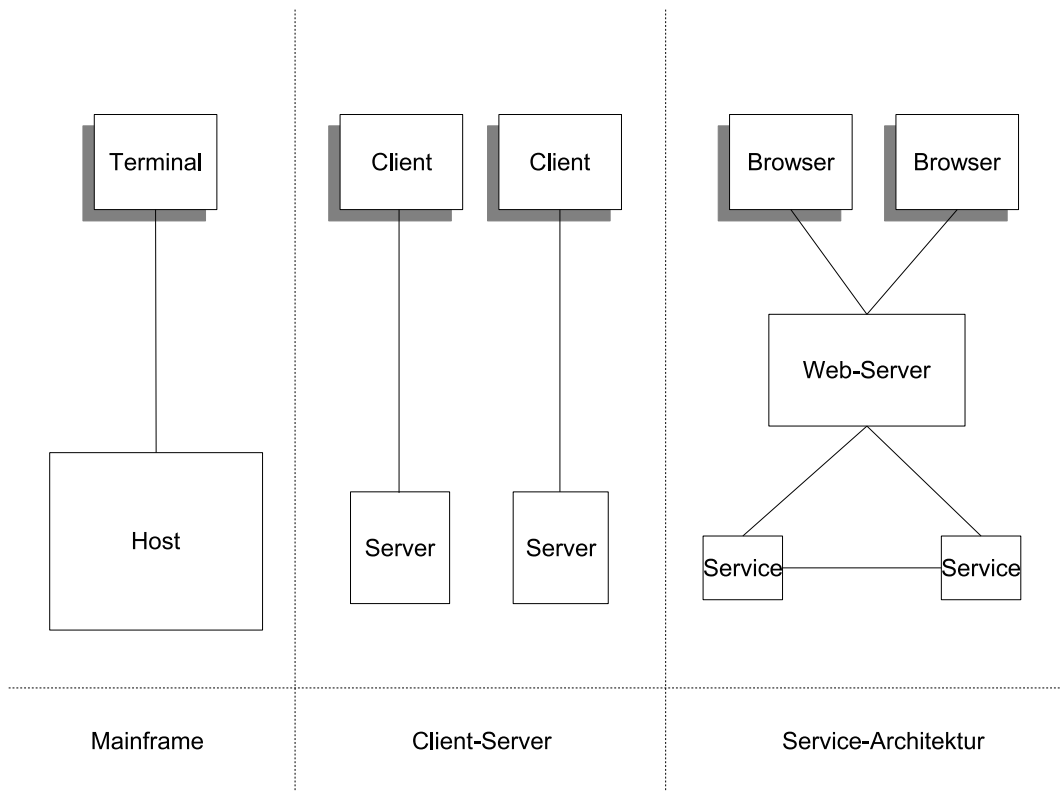
**Tabelle 2.1.:** Systematisierung der Erlösformen [Sch04, S. 12]

Für die Nutzung der Webservices wird eine Transaktionsgebühr erhoben. Dies bedeutet, dass z.B. auf einem bereitgestellten Webshop für jede abgegebene Bestellung eine fixe Transaktionsgebühr entrichtet werden muss [Hal07, S. 80]. Eine Form transaktionsunabhängiger Erlöse wäre die Bereitstellung einer Software gegen einen monatlichen Paketpreis, welcher sich an der prosperierten Userzahl inklusive funktionaler Breite und Datenvolumen (Nutzungsintensität) orientiert. Dieses auch “Subscription-based” genannte Erlösmodell ermöglicht somit eine unlimitierte Nutzung im Rahmen eines pauschalierten Abonnements. Die Besonderheit des in der Regel rein internetbasierten Geschäftsmodells liegt nun darin, dass Programme, deren On Premise Alternative hohe Lizenzgebühren und neben den infrastrukturellen Aufwendungen nicht

planbare Kosten auf sich ziehen, durch eine definierte Nutzungsgebühr eindeutig identifizierbarer und somit kalkulierbarer werden.

### 2.1.1. Anwendungsarchitektur

Folglich wird ein kurzer Überblick über die drei Anwendungsarchitektur-Epochen gegeben [Mas05, S. 40 ff.].



**Abbildung 2.1.:** Entwicklung der Unternehmensarchitekturen [Mas05, S. 41]

**Mainframe-Ära** - ist charakterisiert durch einen Großrechner, welcher alle Applikationen zentral beherbergt und nur über Terminals zugänglich ist. Nach außen hin ist dieser völlig abgeschottet und besteht intern aus einer Reihe paralleler Applikationen, welche über batchbetriebene Datenschnittstellen direkt miteinander verbunden sind.

**Client-Server-Ära** - ist charakterisiert durch die Verteilung von Applikationsteilen, die auf einem zentralen Server angeboten werden. Der zentrale Zugriff erfolgt von einer Arbeitsstation (Client) aus. Ein Server wartet dabei auf Anfragen eines Clients, ein in der Regel am Desktop installierter Client (z.B. SAP R/3: SAPGUI), arbeitet diese Anfrage ab und schickt das Resultat an den Client zurück. Diese Variante kommt bei

traditionellen Softwaresystemen zum Einsatz und bietet alle denkbaren Funktionalitäten und Interaktionsmöglichkeiten. Die Installation eines Clients ohne Webinterface erscheint jedoch für die Benutzer internet-basierter Anwendungen oft als Hemmschwelle die Software einzusetzen, da durch den Einsatz eines Browsers der Anwender deutlich flexibler in der Wahl des Endgerätes und des Einsatzortes wäre. Daher werden häufig Anwendungen, die einen eigenständigen Client besitzen mit einem als Middleware agierenden Terminalserver (ein bekannter Hersteller ist diesem Zusammenhang Citrix) webtauglich gemacht. Der Hauptnachteil dieser Vorgehensweise liegt aber in der benötigten und kostenintensiven Software auf der Clientseite, die die GUI der Anwendung darstellen kann. Dieses Modell ist somit nicht dauerhaft für den Einsatz im Endkundenmarkt geeignet, da die Kunden oftmals nicht bereit sind eine zusätzliche Software zu installieren. Aus diesem Grund ist diese Technologie eher als eine Übergangslösung von traditionellen Anwendungen zu webbasierten Anwendungen zu sehen.

**Service-Architektur-Ära** - ist die völlige Loslösung von Orten und Systemen und wird durch Services charakterisiert, welche von Applikationen erbracht werden. "Ein Service sei in diesem Kontext als eine Funktionalität definiert, die über eine standardisierte Schnittstelle in Anspruch genommen werden kann" [HK06, S. 16]. In einer Service-Architektur werden daher die logischen Systeme, von physischen Implementierungen, durch die Trennung der Logik von konkreten und singulären Systemen, abstrahiert [BIT06e, S. 4]. Durch eine solche Abstraktion lässt sich eine äußerst flexible Infrastruktur für die gemeinsame Nutzung, eine dynamische Zuordnung von Ressourcen und eine zugehörige Prozess- und Organisationsanpassung realisieren [Ede07, S. 71]. Der Anwendung wird also kein bestimmter Server oder Speicher zugeteilt, sondern sie wird, je nach Bedarf, auf einen freien Rechner im Pool weitergeleitet [Sti07, S. 47]. Für die Anbieter liegt der Vorteil in der daraus resultierenden Skalierbarkeit und der Erzielung von Skaleneffekte durch jeden zusätzlichen Kunden, da mit einem gemeinsamen Ressourcenpool die variablen Kosten (z.B. Maintenance und Support) wesentlich geringer ausfallen.

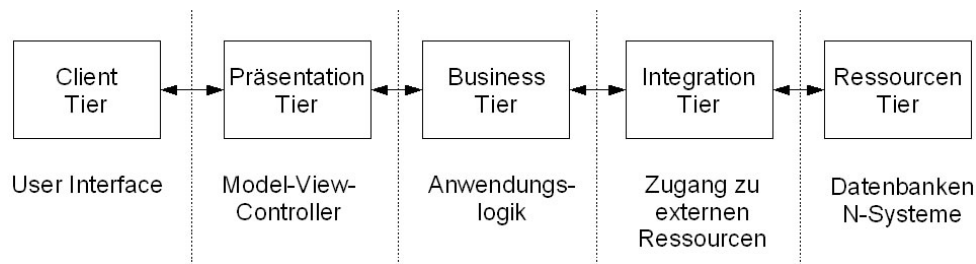
Im Rahmen dieser Arbeit ist die Service-Architektur von großer Bedeutung, da diese eine optimale Plattform für die Service Bereitstellung bietet. Dadurch können Service Provider sowohl die Anwendungsplattform als auch die zugehörige Hardware granular skalieren und Kundenservice-, Vertriebs- und Marketingssysteme auf zuverlässigen und skalierbaren Plattformen anbieten [Wei07].

### 2.1.2. Schichtenmodell von webbasierten Systemen

Der Wunsch nach mehr Modularität, verbesserter Wartbarkeit, Verringerung der Netzbelastung und besserer Skalierbarkeit, führte schließlich zu einer kla-



ren logischen Strukturierung der ursprünglich am Webserver implementierten Anwendungslogik in mehreren Ebenen (oder Schichten bzw. *Tiers*) [Glä03, S. 187]. Das System hat zumindest eine Dreischichtarchitektur auf zu weisen, die die Präsentationsschicht (für die Benutzerverwaltung und die Darstellung der Benutzeroberfläche), die Applikationsschicht (für Ablaufsteuerung, Geschäftslogik und Geschäftsprozesse) sowie die Datenhaltungsschicht trennen [LET06, S. 164 f.].



**Abbildung 2.2.:** N-Tier Architektur [Fac02, S. 71]

Diese Multi-Tier-Architekturen (N-Tier) ermöglichen ein Höchstmass an Flexibilität in der Gestaltung der Skalierbarkeit [Fac02, S. 69 ff.]:

- *Client* - hier findet die gesamte Benutzerwechselwirkung über ein Graphical User Interface (GUI) statt. Der Client hat kaum Applikationslogik implementiert. Je nach Umfang der am Client verankerten Funktionalität spricht man von Thin Clients bzw. Thick Clients. Dabei ermöglichen die Clients einen Zugriff auf das zentrale Backend oder auch auf die zentrale IT-Infrastruktur der Hostingumgebung.
- *Präsentation* - diese Schicht stellt alle notwendigen Informationen zur Verfügung, welche anschließend an das Benutzerinterface geschickt werden und dient daher als Schnittstelle zwischen Anwendungslogik, Webserver und Anwender.
- *Anwendungslogik* - die Anwendungslogik hat die eigentliche Funktionalität der Software implementiert und wird je nach Anzahl der Schichten im Modell auf eine oder mehrere Ebenen verteilt. Diese Schicht enthält einzelne Objekte, die eine Sammlung von Daten und Basisfunktionen, wie auch elementare Regeln, implementiert haben.
- *Integration* - diese Schicht dient zur Reinstanzierung und Speicherung der Objekte aus der Applikationsschicht und beinhaltet Methoden (Middleware-Services) für die Integration verschiedener Objekte mit der darunterliegenden Ressourcenschicht. Dabei stellt die Middleware eine Software-schicht dar, die die Funktionen mittels Kommunikation und Vermittlung für der Programmlogik erbringt. Die Middleware oder auch Enter-



prise Service Bus (ESB) stellen Dienste für den physischen und transparenten Transport von Daten zwischen verteilten Anwendungen bzw. Datenbanken bereit. Dabei werden die Nachrichtenflüsse und die Nachrichtentransformation über Regeln (Transformation Services), durch die Übernahme der Daten von Geschäftsobjekte (Module, Komponenten) und dem daraus anfallenden Abruf anderer Geschäftsobjekte, über definierte Interface Services (Adaptoren, Konnektoren), gesteuert.

- *Datenbank* - diese Ressourcenschicht beinhaltet in der Regel ein Datenbankmanagementsystem (DBMS), welches die Daten speichert und wieder lädt. In der Praxis handelt es sich hier meistens um relationale Datenbanken mit der entsprechenden SQL-Syntax und den Transaktionseigenschaften relationaler Datenbanken [Mas05, S. 121].

Mehrschichtige Architekturen verfolgen, wie eben besprochen, das Prinzip der Teilung von Präsentation, Anwendungslogik, sowie Integration und Datenhaltung in verschiedene Schichten. Dies ermöglicht eine Verschlankung der Applikationen am Front-end, sodass sich auch komplexe Anwendungen auf eingebetteten oder mobilen Geräten mit begrenzten Ressourcen nutzen lassen können. Unabhängig davon mit welchem Client auf eine webbasierte Ressource zugegriffen werden soll, lässt sich die Schichtenarchitektur an folgenden Beispiel charakterisieren:

- Die Anwender stellen mit einem Browser eine Anfrage (Request) über das Netzwerk an einen Webserver.
- Der Request wird dann von einem Load-Balancer<sup>3</sup> an einen Webserver mit geringer Auslastung weitergeleitet.
- Der Webserver bearbeitet den ankommenden Request und stößt die benötigte Applikationslogik auf dem Anwendungsserver an.
- Der Anwendungsserver dient zur Verwaltung allgemeiner technischer Funktionen für das Deployment der Geschäftslogik.
- Entsprechend der Geschäftslogik werden Schnittstellen und Daten unterschiedlicher Ressourcen miteinander gekoppelt, um letztendlich die Datenbasis bereitzustellen.
- Nach Beendigung des Requests werden die Ergebnisdaten via Internet über den Webserver, der Clientebene übermittelt (Response).

---

<sup>3</sup>Ein Load-Balancer dient zur Lastverteilung des ankommenden Traffics auf mehrere Systeme.

Jede Schicht steht in der Regel auch für eine physikalische Ressource. Je umfangreicher und kritischer Größe und Art der Anwendung sind, sind einige Bestandteile als eher optional anzusehen oder entsprechend redundant vorzuhalten, da die zugrundeliegenden Ressourcen, aufgrund unterschiedlicher Lastspitzen, Einfluss auf die nicht funktionalen Eigenschaften der Webanwendung nehmen. Die Hard- und Softwarekomponenten werden, zugunsten einer hohen Verfügbarkeit des Gesamtsystems, mehrmals bereitgestellt, was wiederum einige Ressourcen unausgelastet lässt. Im Rahmen des Webentwicklungsprozesses müssen deswegen Technologien berücksichtigt werden, die die Last auf mehrere Ressourcen verteilen können und entsprechend den Anforderungen skaliert, eine dauerhafte Verfügbarkeit erbringen. Die unterschiedlichen Lastspitzen von Server- und Speichersystemen werden im Rahmen der Ressourcenbereitstellung und -konfiguration durch die Virtualisierungstechnologien ausgeglichen, die es ermöglichen, Ressourcen in einem gemeinsamen Rechnerpool zu nutzen [Sti07, S. 47].

### 2.1.3. SOA - serviceorientierte Architekturen

Um innerhalb eines Unternehmens alle benötigten Ressourcen bei Bedarf zur Verfügung zu haben, zeichnet sich der Trend zu serviceorientierten Architekturen (SOA) immer stärker ab. Diese werden von vielen IT-Experten und -Forschern als Schlüssel für kostengünstige und flexible IT-Systemlandschaften betrachtet [HLP07, S. 21]. Die serviceorientierte Architektur, wird als ein "Systemarchitekturkonzept" gesehen, das den Aufbau einer Anwendungslandschaft aus einzelnen fachlichen Anwendungsbausteinen beschreibt, die jeweils eine klar umrissene Aufgabe wahrnehmen [Krc05, S. 275]. Dazu werden standardisierte Technologien, wie das Hypertext Transfer Protocol (HTTP) und die Extensible Markup Language (XML), verwendet, welche als Webservices implementiert sind. SOA verspricht sich deshalb auch auf der Anwenderseite ein hohes Nutzenpotential, das folgendermaßen zusammengefasst werden kann[RhS05]:

- Neue oder sich ändernde Geschäftsprozesse können durch eine Neukombination bereits bestehender Services besser angepasst werden.
- Das Prinzip der Kapselung führt zur Reduktion der Komplexität.
- Durch standardisierte Schnittstellen wird ein hoher Wiederverwendungsgrad erzielt.
- Darüber hinaus erleichtert die Standardisierung von Schnittstellen das Outsourcing von kompletten Anwendungsfunktionalitäten.

“Die serviceorientierte Architektur dient damit als IT-Architekturstil, der das Konzept der Serviceorientierung durch die Definition umfassender Geschäftsprozesse als modulare Anwendungen unterstützt.

Diese (Teil-)Prozesse können dann entsprechend zusammengesetzt und miteinander verknüpft werden, um damit letztendlich die Gesamtheit der Geschäftsprozesse eines Unternehmens abzubilden und das Gesamtgeschäft eines Unternehmens zu unterstützen” [Bre07, S. 80].

Daraus lassen sich drei Schwerpunkte erkennen:

- Erstens wird die IT an die Geschäftsprozesse angepasst und nicht etwa vice versa.
- Der zweite Aspekt ist die neue Qualität der Prozesse, die der kurzen time-to-market Zeit gerecht werden.
- Drittens erwarten die Unternehmen durch die Einführung von SOA zudem noch eine spürbare Kostensenkung, um dem steigenden Kostendruck Rechnung zu tragen.

Gerade der letzte Aspekt ist jedoch stark von den jeweiligen Gegebenheiten eines Unternehmens abhängig. Auf die Frage, wie sich die relativ hohen Anfangsinvestitionen einer SOA-Ausrichtung mit der Forderung nach einem kurzfristigen Return on Investment verbinden lässt, empfiehlt Beinhauer keine zu hohen Erwartungen an die Kostenersparnis zu stellen.

“Bei der RoI-Betrachtung von SOA-Projekten, sollte man das Augenmerk nicht so sehr auf die Kosten richten, sondern auf den erzielten Nutzen. Ob sich überhaupt Kosten einsparen lassen, ist aus meiner Sicht fraglich. Dass man mit SOA eine neue Qualität bekommt und echte Wertschöpfung erzielen kann, ist hingegen sicher” [Bei05].

## 2.2. IT-Sourcing

Das primäre Ziel beim IT-Sourcing besteht darin, ineffizient ausführbare oder zu teure IT-Services nicht mehr selbst her- und bereitzustellen, sondern auf Dienstleistungen von Best Practice Unternehmen zurückzugreifen, deren Leistungsportfolio prozessspezifische Anforderungen unterstützen und zusätzlich eine höhere Flexibilität beim Softwareeinsatz gewähren. Dabei helfen die immer etablierteren Designprinzipien der serviceorientierten Architekturen (SOA), im Unternehmen klar dargestellte und voneinander abgegrenzt fachliche Funktionen, einzelne Prozessschritte oder Teilprozesse aus dem Unternehmen auszulagern [Mül06b]. Ein einheitliches Verständnis des Dienstleisters für das zugrundeliegende Geschäftsmodell kann zur Deckung der Ansprüche der Kunden einen guten Ausgangspunkt darstellen. Auf einer dafür ausgelegten Infrastruktur wird versucht mögliche Ineffizienzen und Komplexitäten eines On Premise

Betriebes, mit einer je nach Anfrage skalierenden Bereitstellung basierend auf flexiblen Lizenzierungsformen, zu reduzieren.

Im Kontext dieser Arbeit versteht sich daher das IT-Sourcing<sup>4</sup> als eine Methode für die zeitlich begrenzte Externalisierung von Leistungen, Funktionen und Prozessen an einen dafür spezialisierten Anbieter, der über ein dafür ausgelegtes Dienstzugriffsmodell den bedarfsgerechten Abruf ermöglicht (vgl. [Ham07, S. 34]; [SMS06, S. 313]; [Rei05, S. 130]). Dabei kommt es vor allem auf Kriterien wie Optimierung, Kosteneffizienz, Standardisierung und Hochverfügbarkeit an [Ede07, S. 64]. Aus betriebswirtschaftlicher Sicht sprechen besonders die Kostengründe für das Fremdbeziehen von klar definierten Aufgaben [Rei05, S. 130]. Zudem erwarten sich die Unternehmen einen flexiblen Zugriff auf individualisierte Funktionalitäten, die auf einem höheren Qualitätsniveau, ohne Kapazitätsbeschränkungen und über einen standardisierten Zugriffspunkt, erbracht werden, um die Dauer des Einführungsprozesses und damit auch die Time-to-Market - also den Zeitrahmen von der Einführung bis zum produktiven Betrieb - zu verkürzen. Ausgehend von dieser Betrachtungsebene lässt sich das IT-Sourcing nach dem Grad des externen Leistungsbezugs, abhängig von Dauer und Intensität, in unterschiedliche Formen<sup>5</sup> unterteilen.

Im Allgemeinen sind aus der Sicht des Sourcinggegenstandes sowohl das Infrastruktur- als auch das Applikations-Outsourcing maßgeblich [BIT06b, S. 13]. Dabei wird im Rahmen der Leistungsexternalisierung wiederum zwischen kompletten Outsourcing und selektivem Outsourcing unterschieden [Sta01]. Während beim “Komplett Outsourcing” oder auch “Totalen Outsourcing” die Auslagerung der gesamten Informationstechnologie erfolgt, steht beim “selektiven Outsourcing” nur die Auslagerung von eng begrenzten und wohl definierten Aktivitäten im Fokus. Der Anwender bezieht, ohne in den Verantwortungsbereich für die Anwendung und die Infrastruktur mit einbezogen zu sein, die für die Stützung bestimmter Prozesse notwendigen IT-Funktionalitäten. Um dabei die Vorteile von speziellem Domänenwissen auf der Anbieterseite ausnutzen zu können bzw. den sich ständig ändernden Anforderungen, durch Maßnahmen im Rahmen der Flexibilisierung des Softwareeinsatzes, gerecht zu werden, wird der Zugang zu “Best-of-Breed-Funktionalitäten” über spezifische Abrechnungsmodelle als Infrastruktur- und Anwendungsservices im Abonnement ermöglicht [HB06, S. 171]. Neben der Verringerung der Ressourcenauslastung für Implementierung, Betrieb und Support einer Anwendung führt das selektive Outsourcing systembedingt zu einigen Problemfeldern. Vor allem die Thematik rund um die Datensicherheit wird immer wieder als ein zu vernachlässigter Negativpunkt angeführt. Als Beispiel ist eine Studie aus dem Jahre 2003 anzuführen, in der 75 Prozent der befragten Unternehmen angaben, dass

---

<sup>4</sup>“Als Oberbegriff wird Outsourcing oft synonym zu Sourcing benutzt” [JD07].

<sup>5</sup>[JD07] bietet eine gute Übersicht über die unterschiedlichen Outsourcingformen.

das Sicherheitsrisiko als ein besonders wichtiges Entscheidungskriterium angesehen wird [KSvHB03, S. 26].

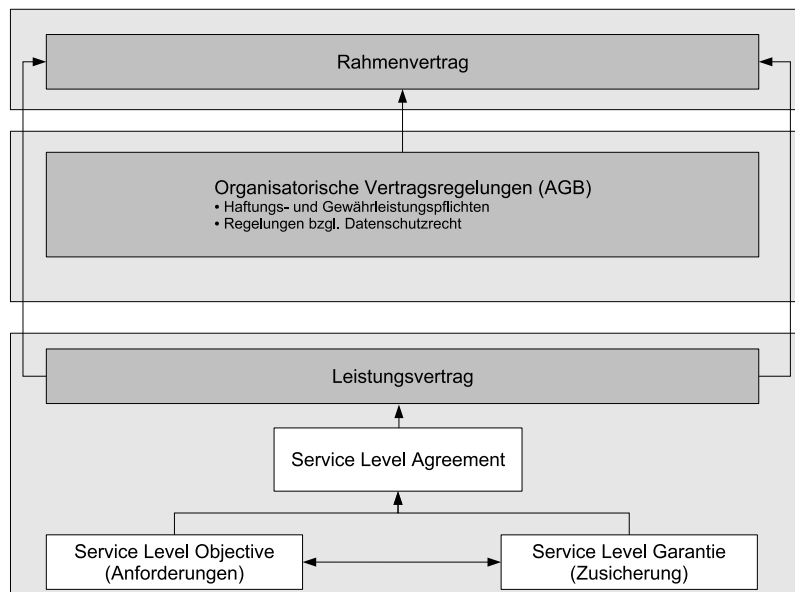
Da im Rahmen eines CRM-Sourcings eine räumlich und zeitlich unabhängige Unterstützung und Durchführung kundenzentrierter Prozesse angestrebt wird, gilt es zunächst folgende Aspekte zu betrachten:

- **Entscheidungen hinsichtlich Eigen- und Fremdproduktion (Make or Buy)**
  - Die bereitgestellten Funktionalitäten der Software müssen für die Durchführung geschäftlicher Aufgaben 24 Stunden am Tag, sieben Tage die Woche und 365 Tage im Jahr (24 x 7 x 365) verlässlich erbracht werden;
  - Ein adäquater Datenschutz muss Vertraulichkeit, Verfügbarkeit und Integrität der verlagerten Daten gewährleisten. Auch die Funktionstüchtigkeit, sowie die Verlässlichkeit der Sourcingleistung muss in einem nachvollziehbaren Preismodell transparent dargestellt sein;
- **Überprüfung ob die Software zur Deckung der Geschäftsziele verantwortungsvoll bereitgestellt werden kann, und**
  - Geschäftsabläufe bedürfen einer adäquaten Abbildung, damit die angebotenen Funktionalitäten die angestrebten Wertschöpfungsbeiträge realisieren;
  - Diverse standardisierte Schnittstellen zum Im- und Export müssen eine automatisierte Verarbeitung von Daten ermöglichen, sowie end-to-end Prozesse unterstützen;
- **ob die Risiken der Bereitstellung angemessen überwacht werden.**
  - Durch eine für den Kunden zugängliche Performance und Überwachungssoftware (Monitoring-Anwendungen) muss ein Tracking der Hostingauslastung ermöglicht werden;
  - Die Anwendung international anerkannte IT-Security-Standards muss nachgewiesen werden können;
  - Zertifizierte Regulatoren, die bei geographischen Unterschieden transparente Datenschutzbestimmungen gewährleisten;

Aufgabe ist es nun, durch die Definition grundlegender Indikatoren eine vertrauensvolle Kooperation zwischen Kunden und Dienstleister zu schaffen [MZK03, S. 447]. Vor allem die Details der Vereinbarung, insbesondere im Hinblick auf die Leistungsmerkmale und Qualitätsziele, liefern eine Basis für das gemeinsame Verständnis des IT-Sourcings [RSG06, S. 195].

### 2.2.1. Vertragsmodell

Um eine erfolgreiche Geschäftsbeziehung zwischen einem Kunden und dem Dienstleister gewährleisten zu können, bedarf es formeller Grundlagen für die Festlegung der konkreten Rahmenbedingungen in Bezug auf die zu erbringenden Leistungen. Dabei umfasst das Vertragswerk einen Rahmenvertrag und einen Leistungsvertrag für die entsprechende Sourcingleistung.



**Abbildung 2.3.:** Struktur des Vertragsmodell

Dazu gehören in der Folge auch die Pflichten der Instandsetzung, die - durch die Einräumung bestimmter Ausfallszeiten für Wartungsarbeiten - zu gewährleisten sind und auch der vertrauliche Umgang mit den Kundendaten. Deshalb muss für die räumlich und zeitlich unabhängige Nutzung, der Dienstleister die Einhaltung diverser Mindestqualitäten, hinsichtlich des Betriebs der Hostingumgebung und regulatorischen Bestimmungen für die Gewährleistung des Datenschutzes in den Allgemeinen Geschäftsbedingungen berücksichtigt haben, welche wiederum den strengen Vorgaben des AGB-Recht unterliegen.

Da der Dienstleister auf organisatorischer als auch auf technischer Ebene mit verschiedenen Anforderungen konfrontiert wird, kann dieser mit einem Zusatzvertrag Detailregelungen und besondere Bedingungen, im Rahmen der Flexibilisierung seines Leistungsportfolios anbieten. Diese zusätzlich als Vertragsbestandteil vereinbarten quantitativen Leistungsmerkmale inklusive deren monetären Ausgleich und auch Qualitätsziele, werden transparent in den Service Level Agreements (kundenorientierte Dienstgütevereinbarung) festgehalten. Eine solche Extrastellung kann z.B. folgende Konkretisierungen beinhalten.:

- die Berücksichtigung und Implementierung kundenspezifischer Governancerichtlinien hinsichtlich Datenhaltung und Datensicherung sowie die Berücksichtigung von Schnittstellen für die System- und Geschäftsintegration,
- die Abstimmung des Funktionsumfangs der komplexen und geschäftskritischen Hostingumgebung, und
- die Priorisierung des Problemmanagements durch kundenorientierte Festlegung der Antwort- und Reaktionszeiten.

Die Service Level Agreements (SLAs) ermöglichen daher ein wichtiges Regelwerk für die Sourcingbeziehung, da alle wesentlichen Aspekte des Leistungsverkehrs zwischen dem Anbieter und den Nutzenden berücksichtigt werden und als Grundlage für die anforderungsgerechte Leistungserbringung dienen [Ber05, S. 12]. Je nach Umfang des Leistungsportfolios und den Ausprägungen der auf der Hostingumgebung zur Verfügung gestellten Software, variieren auch der Umfang und die Detailliertheit der SLAs. Folgende Attribute sind für die Spezifikation einer SLA von Bedeutung [HK06]:

- *Spezifität der funktionalen Parameter*: Mit Hilfe dieser Parameter werden die Anforderungen hinsichtlich Softwareappliances, Backups, Security und Integrationsschnittstellen festgelegt.
- *Service Delivery Point*: Beinhaltet den Zugangspunkt sowie den Ort der Servicedurchführung.
- *Anzahl der Service-Konsumenten*: Diese Angabe ist von besonderer Bedeutung um die benötigten Service-Erbringungskapazitäten zu ermitteln.
- *Preis der Services*: Der Umfang der Services kann sowohl in kompletten Lösungspaketen als auch in einzelnen Modulen erbracht werden und ein fixes oder variables Abrechnungsmodell beinhalten.
- *Service Verfügbarkeit*: Definiert den Zeitrahmen in dem die Services zur Verfügung gestellt werden müssen sowie die Wahrscheinlichkeit, dass diese zu einem bestimmten Zeitpunkt funktionstüchtig sind.
- *Erbringungsdauer des Service*: Darunter wird die Dauer vom aktuellen Serviceabruf bis zur vollständigen Erbringung aller Service-Nutzeneffekte verstanden (Antwortzeit und Datendurchsatz).
- *Zeitrahmen des Service-Supports*: Sind die Zeiten, in denen der Service-Desk besetzt ist sowie die festgelegten Antwortzeiten, um die Service-Konsumenten bei Incidents oder anderen Schwierigkeiten bzgl. der Serviceerbringung zu unterstützen.

- *Service Supportsprache*: Ist die Sprache, in der das Service-Desk-Team bei der Bearbeitung von Incidents und sonstigen servicerelevanten Anfragen mit den Kunden spricht.

Generell wird bei einem selektiven Sourcing die gesamte Wertschöpfungskette auf der Dienstleisterseite durch eine einzige Vereinbarung abgedeckt. Dabei bündelt solch ein Agreement alle für diese Sourcingform relevanten Komponenten hinsichtlich Systemumgebung, Software, Netzwerk und Support zu einer “end-to-end-SLA”, um dem Kunden ein schlüssiges Gesamtkonzept anbieten zu können. Dies hängt wiederum vom Zusammenspiel mehrerer Systemkomponenten ab.

Ein wesentlicher Faktor eines SLAs ist daher, neben der Auswahl geeigneter Kennzahlen zu möglichst exakten Definitionen der Leistung, das Messverfahren zur Überprüfung der Einhaltung von Zielen [SMS06, S. 317]. Wie bei den technischen Aspekten sollte auch bei den organisatorischen Vereinbarungen die Art des Nachweises geklärt werden [Hai07, S. 117]. Hierbei ist insbesondere an Verfügbarkeitsquoten (Availability) und Datendurchsatzquoten zu denken. Die Überwachung und Berichterstattung sollte daher über eine zugängliche Schnittstelle z.B. durch ein Webfrontend ermöglicht werden. Als Beispiel sei das Monitoring Service von Salesforce.com für die transparente Visualisierung der Leistungs- und Verfügbarkeitsdaten anzuführen (siehe <http://trust.salesforce.com/>). Die SLAs werden daher auch als ein wichtiges Kontrollinstrument eingesetzt. Darüber hinaus können entsprechende Malusregeln für den Fall einer signifikanten Unterschreitung der vereinbarten Bereitstellungsziele festgelegt werden. Diese legen fest, wie Kunden zu entschädigen sind, wenn die zugesagten Leistungen ausbleiben [Hai07, S. 117]. Aus diesem Grund müssen die Beteiligten die Kosten abwägen können, die eine Qualitätsunterschreitung nach sich ziehen würde, da z.B. ein Ausfall zu einem hohen finanziellen Schaden führen könnte.

### 2.2.2. Gründe für Outsourcing

Das Bestreben ein Outsourcing-Geschäft zu vollziehen, ist weitgehend durch folgende Frage gekennzeichnet:

“Welche Leistungen erbringen für das Unternehmen einen wesentlichen Beitrag zur Wertschöpfung und welche kostenintensive Leistungsbereiche leisten wenig bzw. kaum was bei” [Mat05, S. 7].

Diese Frage wird durch folgende Faktoren gestützt [A+07, S. 33 f.]:

- *Nicht integrierte IT-Landschaft*: Da die IT-Infrastruktur aus einer gewachsenen IT-Landschaft besteht, kann sich eine schlechte Wartbarkeit und Erweiterbarkeit ergeben. Die notwendigen Veränderungen an den



Geschäftsanforderungen lassen sich nur in komplexen Projekten umsetzen.

- *IT kann Businessveränderungen auf der Infrastrukturseite nicht schnell genug folgen:* Die beschriebene Komplexität der IT-Landschaft hat zur Folge, dass Veränderungen nicht so durchgeführt werden können, wie sie gewünscht wären und daher das Risiko des Scheiterns von Projekten deutlich erhöht wird.
- *Hohe Veränderungsdynamik der IT-gestützten Prozesse:* Die Wettbewerbssituation führt zu einer ständigen Anpassung der IT-Landschaft und wirkt sich daher auf die Dynamik laufender IT-Projekte aus.
- *Starre Kostenstruktur in der Infrastruktur:* Da ein großer Anteil der IT-Kosten durch Personalkosten und hohen Abschreibungen verursacht wird, lässt dieser Fixkostenblock wenig Raum für strategische Investitionen.
- *Hoher Kostendruck führt zum Einsatz teilweise veralteten Technologien:* Der durch den starken Wettbewerb gekennzeichnete Markt wirkt sich auch auf das Budget der IT aus. Anstatt den Lebenszyklus der IT-Systeme aktiv zu managen, werden Ersatzinvestitionen zu spät geplant und getätigt.

Die erhöhte informationstechnologische Komplexität erfordert neben den bereits gebundenen Ressourcen, zunehmend unternehmensinterne Ressourcen, da ein gewünschter Automatisierungsgrad auf die Integration modernster und hochqualitativer Informations- und Kommunikationstechnologie angewiesen ist. Besonders die periodisch anfallenden Kosten wie Raum-, Personal- und Instandhaltungskosten die unabhängig von der Nutzung anfallen, lassen sich problemlos externalisieren [E+06, S. 35]. Dem unternehmensinternen IT-Personal wird dadurch Kapazitätsfreiraum geboten, welchen sie zur Entwicklung und Organisation von strategisch bedeutenden Prozessen und Systemen für die Förderung des Wachstums nutzen können [Nol06, S. 251]. Dadurch lassen sich folgende Vorteile identifizieren (vgl. [FO04, S. 332]; [LLJ03, S. 150]):

- Finanzen
  - Senkung der Basiskosten und der Kosten pro Einheit
  - Geringere Aktiva in der Bilanz
- Strategie
  - Fokussierung auf das Kerngeschäft
  - Zugang zu Kompetenzen, Fähigkeiten und Technologien
  - Erhöhung der organisatorischen Flexibilität

- Innovationsgeschwindigkeit und Umsetzungskompetenz
- Reduzierung technisch-operationeller Risiken
- Technologie und Service
  - Standardisierung der Infrastruktur
  - Standardisierung von Infrastrukturdienstleistungen
  - Optimierung der Prozesse durch Qualitäts- und Leistungsverbesserung

### 2.2.3. Kritische Merkmale des Outsourcings

Neben der Erwartung effiziente Leistungen zu möglichst geringen Kosten beziehen zu können, sind den auslagernden Unternehmen die Risiken einer Verlagerung durchaus bewusst. Datenschutz und Datensicherheit sowie nicht zufriedenstellende Verfügbarkeit und fehlende Transparenz sind die bedeutendsten Gründe die gegen eine Auslagerung sprechen [Str05, S. 35]. Dieser Nachteil wird seit Jahren vorgebracht, wenn es um den Einsatz von Software bei Anwendungsdienstleistern geht. Schließlich liegt es auf der Hand, dass durch die Bindung an einen Anwendungsdienstleister sämtliche Kontrollen verloren gehen, und zwar nicht nur über die eingesetzte Softwarefunktionalitäten, sondern auch über die gesamte darunter liegende Bereitstellungsinfrastruktur.

Umso kritischer das selektive Outsourcing - im Kontext der Softwarebereitstellung für die Abdeckung bestimmter Geschäftsziele - eingeschätzt wird, desto bedeutender sind zuverlässige Kommunikation und Verfügbarkeit der ausgelagerten Tätigkeit. Weiters gelten, je nach Unternehmensstandort, oft unterschiedliche Transparenzverpflichtungen und regulatorische Anforderungen [Gar07, S. 16]. Somit bedarf es einem gemeinsamen Verständnis für die verwendeten Mittel und Methoden, vor allem dann, wenn die vereinbarten Leistungen strategische Differenzierungsvorteile gegenüber dem Wettbewerber versprechen oder der Gesamtumfang der vereinbarten Leistungen ein hohes IT-Budget verschlingt [RSG06, S. 202]. Deshalb kommt der klaren Zuordnung sowie der Definition von Schnittstellen zwischen Kunden und Anwendungsdienstleistern eine besondere Bedeutung zu. Sind dabei die Prozesse des IT-Service-Managements nach Standards wie z.B. ITIL (▷ vgl. Abschnitt: 3.3.2) organisationsübergreifend abgestimmt und an den speziellen Kontext der Geschäftsbeziehung angepasst, wird sich die Kommunikation zwischen den am Dienstleistungsprozess beteiligten Akteuren vereinfachen lassen [Sti07, S. 42].

Eine große Verunsicherung herrscht auch bei der Integration der Dienstleistung in die Anwendungsumgebung des Kunden. Die Nichtberücksichtigung der Integration der extern erstellten Leistungen in die internen Arbeitsabläufe, kann daher die positiven Effekte sogar vollständig zunichte machen.

## 2.3. Begriffsbestimmung Softwarebereitstellung

Im Kontext des selektiven Outsourcings verlangt die Softwarebereitstellung (engl. Providing) technische und organisatorische Aufgaben von einem Anwendungsdienstleister, welcher die technischen Voraussetzungen und Kompetenzen besitzt, um eine Software - zentral als Dienstleistung (engl. Service) verwaltet - so zur Verfügung zu stellen, dass eine gesicherte Nutzung möglich ist [Jac05, S. 15]. Dabei lauten die Kernaspekte dieser Definition folgendermaßen:

- Eine *Software* oder auch eine *Anwendung*, ein *Anwendungssystem* sowie eine *Applikation* wird im Rahmen dieser Arbeit als ein standardisiertes Softwaresystem für einen klar definierten Anwendungsbereich definiert. Für die Sicherstellung eines günstigen Betriebes ist die Vereinigung der Vorteile von Standardsoftware mit den Paradigmen Anpassbarkeit und Erweiterbarkeit entscheidend.
- Die *gesicherte Nutzung* bedeutet die Verpflichtung des Anbieters zur Leistungserfüllung, um die definierte Beanspruchung für die Nutzenden, unabhängig von internen und externen Störfaktoren, stetig zu gewährleisten. Dabei wird durch Kompensierung der Störungs- und Ausfallsrisiken und durch Vorkehrungen für die ständige Aufrechterhaltung, bzw. zügige Wiederherstellung der erforderlichen Bereitstellungskapazität, die vereinbarte Verfügbarkeit der bereitgestellten Software gewährleistet [HK06, S. 38].
- Die *technischen und organisatorischen Aufgaben* erweitern den technischen Betrieb wie Wartungs- und Pflegedienste, Netzwerkmanagement und Supportleistungen durch organisatorische Maßnahmen [Jac05, S. 16]. Anhand festgeschriebener Leistungsvereinbarungen werden dabei, die Risiken einer Auslagerung angemessen überwacht und kontrolliert, um Incidents proaktiv zu verhindern.
- Der Begriff *Dienstleistung* (Serviceleistung) rückt vertragsrechtliche und monetäre Aspekte in Vordergrund und beschreibt eine anwendungszentrierte Gesamtleistung, deren Nutzeffekte - bei der tatsächlichen geschäftsbezogenen Anwendung - eine unmittelbare Verbesserungen der geschäftlichen Wertschöpfung hervorrufen. Dabei bezieht der Kunde, unabhängig von örtlichen oder zeitlichen Restriktionen und ohne in den Verantwortungsbereich für die Software und die Infrastruktur mit einbezogen zu werden, die für die Durchführung bestimmter Prozesse notwendigen Softwarefunktionalitäten. Dies betrifft neben der reinen Softwarebereitstellung und Wartung auch andere damit verbundene Dienstleistungen, wie Beratung und weitergehenden Support [ZE08, S. 49].

- Der *zentrale* Betrieb (Anwendungsfunktionalität und Bereitstellung) und das zentrale Management (Update, Programm- und Datenpflege) erfolgt für alle Benutzer auf einer homogenen Serverplattform [Wal02, S. 50]. Dabei wird die Software, zur Generierung von Kostenvorteilen, auf einer gemeinschaftlich zu nutzenden Systemumgebung über das Internet mehreren Kunden zur Verfügung gestellt.

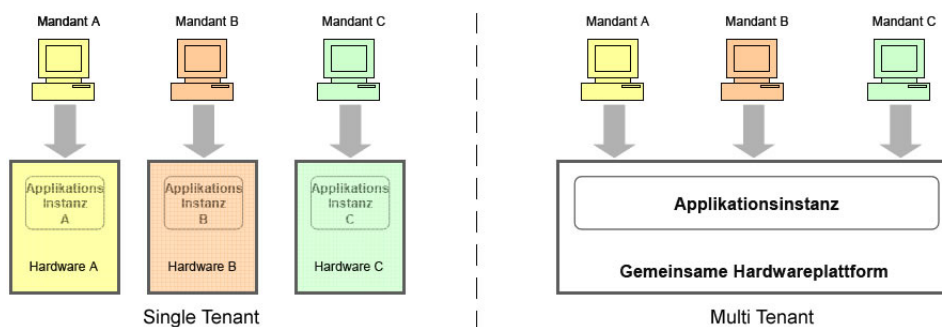
Diese Form des Fremdbezugs von informationstechnologischen Dienstleistungen umfasst die sach- und funktionsgerechte Bereitstellung einer konfigurierbaren Standardsoftware, die vollständig oder überwiegend mit Internettechnologien realisiert wird. Dabei werden die kundenspezifischen Daten (Nutzdaten) in der zugrundeliegenden Datenbank der bereitgestellten Software gespeichert, welche auch gleichzeitig die Daten anderer Unternehmen beherbergt. Damit gewinnen neue Problemkreise stark an Bedeutung, die im Zusammenhang mit der klassischen Nutzung von Software eher eine untergeordnete Rolle spielen. Die Anwender haben zunächst einmal Interesse daran, dass ihre Daten sicher sind, d.h. geschützt vor unbefugter Nutzung (access control, authentication), Verfälschung (integrity) oder gar Zerstörung (availability). Zusätzlich ist, auch eine hohe Verfügbarkeit der Systeme erwünscht bzw. erforderlich, weil in kritischen Fällen sonst erhebliche wirtschaftliche Schäden entstehen können. In diesem Zusammenhang sind die Haftungs- und Gewährleistungsbedingungen der jeweiligen Anwendungsdienstleister ein wichtiger Aspekt für den Anwender (▷ vgl. Abschnitt: 2.2.1). Die Problematik mit der ausgelagerten Datenhaltung wird jedoch von Heinrich durch folgender Sichtweise entkräftet [Hei68, S. 69] zitiert nach [Rie05, S. 51]:

- „Jeder Dienstleister, der Geheimhaltungsbruch begeht, indem er seine Kenntnisse über Daten von Kunden an Dritte weitergibt, setzt seine Existenz aufs Spiel.“
- „Kein Unternehmen kann absolute Geheimhaltung sichern, da notwendigerweise eigene Mitarbeiter in interne Vorgänge eingeweiht sind. Die Sicherung der Geheimhaltung von Daten ist folglich auch gegenüber eigenen Mitarbeitern nicht gegeben. Vielmehr ist die Gefahr der Weitergabe von Betriebsgeheimnissen bei eigenen Mitarbeitern größer, weil sie die Vorgänge im Detail kennen, eine größere ‚Nähe‘ zu ihnen besitzen und möglicherweise ein persönliches Interesse an ihrer Verwertung haben.“

### 2.3.1. Koordinationsformen zwischen Software und Anwender

Die Softwarebereitstellung im Rahmen eines Dienstleistungsmodells inkludiert Verschlüsselung, Übertragung, Speicherung von Daten sowie den Zugriff auf diese Daten und ermöglicht eine auf vertraglicher Basis geregelte Nutzung

von Softwaresysteme über das Internet. Dabei wäre es nahe liegend, als Lösung standardisierte Serviceleistungen anzubieten, welche für den Kunden die kostengünstigste Variante sind [ZS07, S. 259]. Die Softwarebereitstellung zu günstigen Konditionen ist für den Anbieter nur dann wirtschaftlich möglich, wenn er die Software für viele Anwender gleichzeitig vereinheitlichen kann, um die Fixkosten der Providinginfrastruktur auf mehrere Anwender verteilen zu können. Wenn aber die Softwarebereitstellung einen größtmöglichen Kundennutzen und damit Wertschöpfung beim Kunden generieren soll, muss diese Dienstleistung jedoch für jeden Kunden individuell gestaltet werden. Im Sinne der Effizienz ist zwischen Kosten und Kundennutzen ein Kompromiss zu finden [ZS07, S. 259].



**Abbildung 2.4.:** Koordinationsformen zwischen Software und Anwender [Küh07, S. 3]

### Single Tenant

Bei dieser 1:1 Zuordnung wird eine Softwareinstanz, für jeden Kunden (Mandanten) gesondert auf einer dezidierten Hardwareplattform betrieben [Küh07, S. 3]. Das heißt, jeder Kunde hat einen isolierten Hardware- und Softwarestapel zugeordnet und ist komplett losgelöst und unabhängig von anderen Kunden-Instanzen.

Bei der Single Tenancy Variante (traditionelles Einzelinstanz-Modell) können die Anforderungen durch Hardwareergänzungen und Individualprogrammierungen erfüllt werden. Diese dezidierte Bereitstellung verlangt jedoch höhere Anforderungen hinsichtlich des IT-Managements auf der Anbieterseite, da Updates und Patches für jede einzelne Instanz eingespielt werden müssen. Folglich werden keine Skaleneffekte<sup>6</sup> auf der Anbieterseite erzielt, da für jede einzelne Softwarebereitstellung die Entwicklungs- und Bereitstellungskosten zu berücksichtigen sind. Deshalb werden vermehrt standardisierte Bereitstellungsangebote auf einer Plattform gebündelt, um durch ein Umlageverfahren die anfallenden Kosten auf viele Nutzer aufteilen zu können.

<sup>6</sup>Skaleneffekte treten ein, wenn bei definierten Fixkosten der Output gesteigert und die Durchschnittskosten verringert werden.

### Multi Tenant

In der 1:N Bereitstellung (“One-to-Many”) werden Hardwareplattform und Software gemeinsam genutzt, weil eine mehrmandantenfähige Lösung (Multi Tenant), durch die Bündelung der Gesamtbetriebskosten, u.a. für eine Reihe von Skaleneffekten sorgt. Um z.B. eine Kostensenkung für ein ERP-System (Enterprise Resource Planning) von 30%-50% zu erzielen [Rie05, S. 44], müssen sowohl die Lizenzkosten als auch die Wartungs- und Infrastrukturkosten auf alle Mandanten aufgeteilt werden, welche durch Autorisierungsfunktionen sowie Sicherheitsrichtlinien im Sinne der Mandantenfähigkeit sicher voneinander getrennt sind [Küh07, S. 3]. Die Multi Tenant Bereitstellung setzt genau da an:

Die Softwarebereitstellung erfolgt auf einer gemeinsamen Infrastruktur und Codebasis, wobei für jeden einzelnen Mandanten ein eigener Workspace (engl. multi-tenancy enabled service environment) für die isolierte Nutzung instanziiert wird. Es existiert also nur mehr eine physikalische Instanz der Software, die simultan identische Dienste - basierend auf einer gemeinsamen Dateninfrastruktur - leistet. Deshalb führt die optimierte Nutzung der Ressourcen (wie Serversysteme, Datenbanken, Betriebsressourcen sowie die Bündelung der Wartungskosten an eine Codebasis) zur Erzielung von Kostendegressionseffekten. Martin Feidicker [Fei07, S. 20] beziffert die mögliche Kostensenkung mit bis zu 30 % gegenüber dem dezidierten Betrieb, da der Anbieter nicht, wie beim Single Tenancy Modell, notwendige Updates oder Patches für jede einzelne Installation durchführen muss. In weiterer Folge ermöglicht diese Koordinationsform eine sofortige Nutzung der Software (im Extremfall bedarf es lediglich einer Registrierung) [Koh07a]. Da aber viele Kunden differierende Anforderungen an eine Software stellen, um die internen Prozesse optimal zu unterstützen, bedarf es einem Mittelweg in Form von standardisierten, skalierbaren Branchen- und Prozesslösungen, bei einer gleichzeitig schlanken Kostenstruktur [Pie07]. Die Deckung der kundenseitigen Anforderungen erfolgt hierbei über ein mandantenspezifisches Metamodell, das alle individuellen Einstellungen in einer metadatenbasierten Konfigurationsdatenbank speichert (> vgl. Abschnitt: 2.3.4).

#### 2.3.2. Sicherheitsrelevante Anforderungen an den Anbieter

Der Schlüsselfaktor für die Erfüllung der Kundenansprüche ist die Sicherung der Vertraulichkeit [Sew05, S. 13]. Vor allem sensible Daten müssen durch ein entsprechendes Datenschutz- und Security-Konzept vor internen, wie auch vor externen Gefahren geschützt werden. Dabei ist im Sinne des Datenschutzes zu gewährleisten, dass die Übertragungskanäle vor Aushorchversuche oder Missbrauch geschützt werden, sei es durch Verschlüsselung oder Abschottung [Köh07b, S. 218]. Des Weiteren muss neben der Sicherstellung der Vertraulichkeit auch die Integrität der Daten, also deren Richtigkeit und Vollständigkeit,

sichergestellt sein. Es müssen daher entsprechende Sicherungsmaßnahmen für eine zuverlässige Authentifikation der Kommunikationspartner implementiert sein. Für diesen Zweck wurden in den letzten Jahren kryptographische Verfahren entwickelt, die inzwischen standardisiert sind oder Quasi-Standards darstellen. Das SSL-Protokol (Secure Socket Layer) hat für die authentifizierte und vertrauliche Kommunikation zwischen Webserver und WWW-Browser einen hohen Reifegrad erreicht, so dass auf der Basis von Public-Keys und X.509-Zertifikaten eine automatisierte Identifikation von Server und Client (Provider und Anwender) vorgenommen werden kann. Als Voraussetzung für den breiten Einsatz von SSL ist die sogenannte Public-Key-Infrastructure (PKI) zu nennen, die wesentliche Elemente zur Handhabung und Verwaltung von öffentlichen Schlüsseln (Public-Keys) bzw. Zertifikaten enthält. Den Kern bildet dabei ein Trust Center (bzw. Certification Authority), das eine nachprüfbare Verbindung zwischen einem asymmetrischen Schlüsselpaar im Rahmen eines Zertifikates, mit einer digitalen Signatur begläubigt. Durch Überprüfung des SSL-Zertifikates wird die Identität des Empfängers (Provider) sichergestellt und die Datenübertragung verifiziert. Als ein weiterer Ansatz ist zusätzlich der Aufbau von VPNs (Virtual Private Networks) zwischen Provider und Anwender per IPsec zu erwähnen. Da aber IPsec auf der Transportebene (IP-Ebene) arbeitet können sich nur Netzwerkkomponenten gegenseitig authentifizieren, jedoch keine Anwendungen.

Neben der Verschlüsselung der Übertragung muss die Providinginfrastruktur vor DoS-Angriffen (Denial of Service), unbefugter Nutzung und Manipulation entsprechend geschützt werden. Eine Gefahr besteht gerade bei Internetanwendungen, durch DoS Anfragen oder durch Schadprogramme, die gezielt nach Schwachstellen suchen und diese folglich ausnutzen. Dagegen helfen dienen entsprechende Filtermechanismen von Firewall-Systemen, wie Paketfilter und Proxies für die Vorbeugung von Mal- und Spyware. Zusätzlich gebührt auch der Integrität, also der Richtigkeit und Vollständigkeit der Daten, oberste Priorität. Aus diesem Grund ist die Datenspeicherung eng mit den Richtlinien zum Datenschutz und der Sicherheit verknüpft. Es liegt daher in der Pflicht des Anbieters dafür zu sorgen, dass nur ein berechtigter Personenkreis Zugriff auf die Daten erlangen kann.

Ein Anbieter ist zudem verpflichtet<sup>7</sup> die Systemsicherheit zu gewährleisten. Dazu gehört der Schutz vor Datenverlust durch redundante Speicherung und Maßnahmen („Desaster-Recovery-Pläne“), die einen Notfallplan zur Wiederherstellung der Daten im Fall eines Systemausfalls berücksichtigen. Daher muss für die Sicherung des operativen Geschäfts, neben der redundanten Speicherung der Daten, auch eine Backupstrategie für die internen und externen

---

<sup>7</sup>Der Anbieter ist verpflichtet die Zusagen, die er mit dem Vertragsverhältnis abgibt, auch einzuhalten.



Steuerungen des Datenflusses, mit Hilfe von Archivierungsmethoden, berücksichtigt werden. Darüber hinaus, muss dem Kunden über eine standardisierte Schnittstelle ein ständiger Datenexport (z.B. in CSV) gewährleistet werden. Dabei lässt sich der Schutz vor technischen Defekten mit einem geographisch entfernt gelegenen Ausfalls-Rechenzentrum, in dem die Backups und Spiegelungen der eingesetzten IT-Systeme ausgelagert werden, relativ umfassend abdecken. Jedenfalls sollten Anbieter für datenintensiven Branchen international standardisierte Verfahren und Prozesse, für eine sichere Informationsverarbeitung, implementiert haben und sich regelmäßiger Überprüfungen ihrer eigenen Sicherheitsstandards durch externe Security Audits unterziehen [BIT06c, S. 70]. Als Anhaltspunkte können die zugrundeliegenden Methoden für Zertifizierungsrichtlinien (Trust Services) des American Institute of Certified Public Accountants<sup>8</sup> (AICPA) dienen. Trust Services vereinen hierbei die Standards SysTrust und WebTrust des AICPA, um etwaige Risiken der IT unter Kontrolle zu bekommen. Von SysTrust werden vier Prinzipien definiert, die die Zuverlässigkeit des Systems zertifizieren [Mat05, S. 91 f.]:

- *Verfügbarkeit*: Das System ist gemäß des SLAs verfügbar.
- *Sicherheit*: Das System ist gegen unauthorisierte physische und logische Zugriffe geschützt.
- *Integrität*: Die Verarbeitung durch das System wird komplett, fehlerfrei, zeitgerecht und autorisiert durchgeführt.
- *Wartbarkeit*: Die Aktualisierung des Systems verursacht keine Beeinträchtigungen der zu leistenden Aufgaben und gewährleistet weiterhin die Verfügbarkeit, Sicherheit und Integrität des Systems.

WebTrust bezieht sich im Gegensatz zu SysTrust auf Webanwendungen, wobei die bei SysTrust beschriebenen Prinzipien auch bei WebTrust ihre Gültigkeit haben [Mat05, S. 92 f.]:

- *Datenschutz*: Ohne ausdrückliche Genehmigungen dürfen keine Informationen über Personen gespeichert werden.
- *Integrität*: Prüfung der Transaktionen auf Richtigkeit und Vollständigkeit.
- *Datensicherheit*: Schutz vor unberechtigtem Zugriff durch Dritte, via Verschlüsselung und Firewalls.
- *Verfügbarkeit*: WebTrust definiert Maßnahmen für die Gewährung der Systemverfügbarkeiten.

---

<sup>8</sup>vgl. <http://infotech.aicpa.org/Resources/System+Security+and+Reliability/System+Reliability/Trust+Services/>



- *Vertraulichkeit*: Hier werden ebenfalls Maßnahmen für die Gewährleistung der Vertraulichkeit der Daten im Geschäftsverkehr definiert.

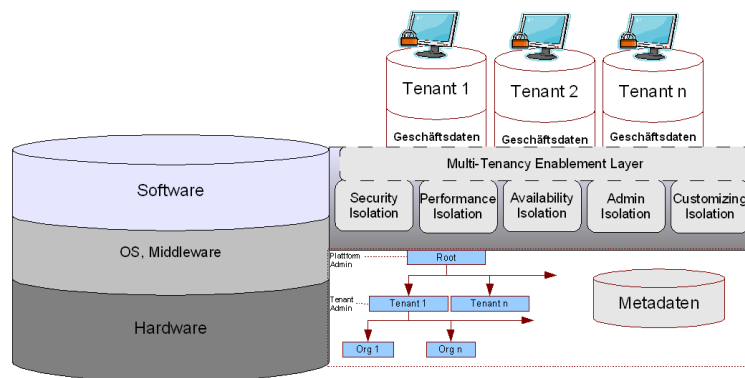
Dabei werden mit den Zertifizierungsrichtlinien geschäftliche, rechtliche sowie technische Anforderungen für den Betrieb, Überwachung, Wartung und Updates einer Bereitstellungsinfrastruktur, unter Berücksichtigung der Risiken und regulatorischen Datenschutzmaßnahmen, festgelegt. In weiterer Folge sei noch TRUSTe<sup>9</sup> als Gütesiegel (Seal-Programm) für die Einhaltung von Datenschutzrichtlinien zu beachten. Anders als bei der Datensicherheit, stellen Richtlinien zum Datenschutz, Verhaltensregeln dar, die beschreiben, wie mit den Daten umgegangen werden muss. Im Rahmen dieses Seal-Programms werden die Safe Harbor Abkommen behandelt, die vor allem für den Datentransfer in die USA von besonderer Bedeutung sind. Die Safe Harbor Principles stellen eine besondere Datenschutzvereinbarung zwischen der Europäischen Union und den Vereinigten Staaten dar. US Unternehmen verpflichten sich gegenüber dem US Department of Commerce und der Federal Trade Commission freiwillig zur Einhaltung bestimmter Datenschutzgrundsätze gemäß der Datenschutzrichtlinien der EU für die Verarbeitung personenbezogener Daten [o.V00].

### 2.3.3. Eigenschaften einer Multi Tenant Plattform

Basierend auf einer gemeinsam nutzbaren Hard- und Softwareinfrastruktur, erfolgt nun in einer dafür ausgelegten Multi-Tenant Plattform (▷ vgl. Abbildung: 2.5) eine mandantenorientierte Virtualisierung des Leistungsportfolios, das über ein Webfrontend als eine nutzerspezifische Softwareinstanz dargestellt wird [GSH<sup>+</sup>07, S. 551]. Im Back-end werden alle Hard- und Softwareressourcen als IT-Betriebsmittel (IT-Utilities) über eine standardisierte Schnittstelle zu einer physikalischen Instanz zusammengefasst und, gemeinsam mit den Metadaten, bei Abruf für jeden einzelnen Tenant (Mandanten), virtualisiert, um die Verbundeffekte einer traditionellen Mehrmandantenarchitektur mit der isolierten Bereitstellung eines Einmandantensystems (Single-Tenant Umgebung) zu verbinden. Da aber die IT-Utilities, aufgrund unterschiedlicher Lastspitzen bei der simultanen Nutzung, einer Plattform Einfluss auf die nicht funktionalen Eigenschaften der virtualisierten Instanzen nehmen, müssen Technologien berücksichtigt werden (▷ vgl. Abschnitt: 3.3), die durch die Lastaufteilung eine dauerhafte Verfügbarkeit garantieren. Dementsprechend muss der Anbieter alle Maßnahmen hinsichtlich Anwendungsentwicklung und Infrastrukturmanagement darauf ausrichten, eine skalierbare Nutzung mandantenspezifischer Instanzen, wie bei einer "isolierten" Hostingumgebung, zu gewährleisten. Deshalb sind folgende Eigenschaften, nach [GSH<sup>+</sup>07], für die Multi-Tenant Infrastruktur von besonderer Bedeutung, zumal die Bereitstellung eine nahezu unbegrenzte Nutzermenge ermöglicht [Rie06]:

---

<sup>9</sup>vgl. <http://www.truste.org/>



**Abbildung 2.5.:** Multi-Tenant Plattform [GSH<sup>+</sup>07, S. 551]

- *Datensicherheit (Security)*: Da die persistenten Daten der einzelnen Mandanten in einer gemeinsamen Dateninfrastruktur gespeichert werden, muss die Vertraulichkeit, Integrität und Verfügbarkeit durch Authentizitätsmechanismen sowie Maßnahmen zur Isolierung der mandantenspezifischen Daten gewährleistet werden.
- *Zuverlässigkeit (reliability)*: Die Zuverlässigkeit eines Systems  $R(t)$  zu einem bestimmten Zeitpunkt ermöglicht die Wahrscheinlichkeit, dass das System bei definierter Beanspruchung eine geforderte Funktionalität zum definierten Zeitpunkt erbringen kann [Köh07b, S. 122].
- *Isolierte Verfügbarkeit (availability)*: Die isolierte Verfügbarkeit beschreibt die Wahrscheinlichkeit der Funktionstüchtigkeit der virtualisierten Instanz, in Abhängigkeit von der Systemzuverlässigkeit  $R(t)$ . Folgende Tabelle ( $\triangleright$  vgl. Tabelle: 2.2) soll einen Überblick über die tatsächlichen Stunden<sup>10</sup> an Ausfälle bei einer bestimmten Verfügbarkeit darstellen.
- *Isolierte Administrationskonsole*: Jeder virtualisierten Arbeitsumgebung bedarf es einer Konsole für administrative Tätigkeiten, wie Userverwaltung, Backups, Monitoring, Konfiguration und Integration.
- *Isoliertes Customizing*: Jede gehostete Softwareinstanz muss in einer selbstadministrierenden Weise anpassbar sein. Die Deckung der kundenseitigen Anforderungen erfolgt hierbei über ein mandantenspezifisches Metamodell, welches individuelle Konfigurationseinstellungen metadatenbasiert beschreibt.

<sup>10</sup>Das Jahr wird mit  $24 \text{ h} \times 364 \text{ Tagen} = 8736 \text{ h}$  berechnet.

Verfügbarkeit ( %)	Stunden Ausfallszeit im Jahr (h)
99	84
99,5	44
99,9	8,5
99,95	4,4
99,99	1
99,999	0,085 (5 Minuten)

**Tabelle 2.2.:** Systemverfügbarkeiten in % [Köh07b, S. 128]

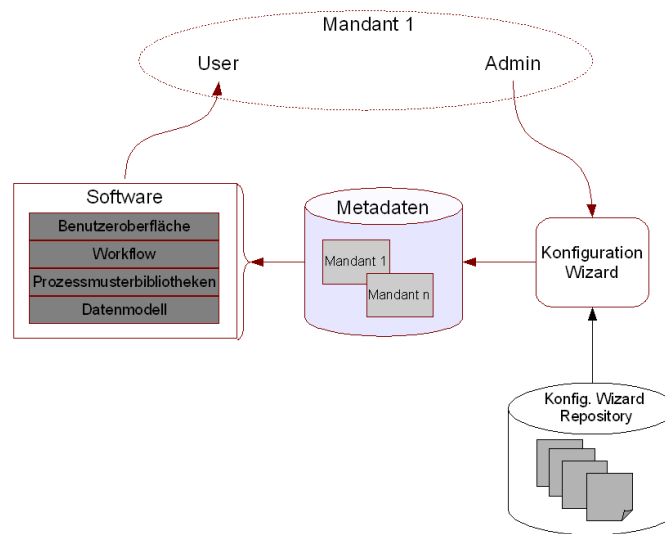
### 2.3.4. Customizing der ausgelagerten Software

Ein weiterer Aspekt der Mehrbenutzerfähigkeit und insbesondere der Mandantenfähigkeit ist die gezielte Adaption und Parametrisierung (Customizing) der bereitgestellten Software an die jeweiligen Kernprozesse der organisatorisch selbstständigen Einheiten und deren Umfeld [Jac05, S. 103]. Dabei entspricht jeder Mandant einem eigenen Unternehmen, das über einen Satz von voreingestellten Datenbanktabellen seine Organisationsstruktur und Arbeitsschritte abbilden kann.

Beim Customizing von Standardsoftware wird daher, ohne den zugrundeliegenden Sourcecode zu verändern, die geeignetste Ablaufalternative aus einer Vielzahl möglicher Varianten bzgl. Funktionen und Prozessen durch Setzen von Umgebungsvariablen konfiguriert. Die Konfigurationsoptionen können, je nach Sichtweise, einen enger oder weiter gefassten Detaillierungsgrad aufweisen [AM04, 87 ff.]. Durch die Parametrisierung soll eine angepasste Sicht auf die jeweiligen Objekte ermöglicht werden, um den zu stützenden Geschäftsprozessen gerecht zu werden [Jac05, S. 20]. Abts unterscheidet in organisatorische Parameter (Definition eines Rollenmodells<sup>11</sup>), Verfahrensparameter (Festlegen von Workflows und Genehmigungsprozessen), beschreibende Parameter (z.B. Kalender, Währungen und Datenfelder) und technische Anpassungen durch Anpassungskonfiguration bzw. -programmierung [AM04, S. 88].

Wie bereits in der Einleitung beschrieben, beinhaltet die Grundidee des Geschäftsmodells On Demand eine fest vorgegebene Funktionalitätsvielfalt, die allgemeine Funktionen und Workflows fertig vorkonfiguriert - als "ready-to-run" Konzept - zur Verfügung stellt, um den Aufwand für das clientseitige Customizing so gering wie möglich zu halten. Plakativ ausgedrückt, soll die

<sup>11</sup>Über eine Rolle wird ein Benutzer mit Rechten auf eine bestimmte Ressource verknüpft um den Zugriff auf Inhalte und Aktionen einzuschränken [Sto06, S. 95].



**Abbildung 2.6.:** Customizingwizard [GSH<sup>+</sup>07, S. 557]

Software schnell, einfach, flexibel, innovativ und preiswert sein. Daher werden von der Anbieterseite - im Rahmen des Systemcustomizing - bereits homogene Kundenanforderungen mandantenunabhängig berücksichtigt. Darüber hinaus soll ein implementierter Customizingrahmen jedem einzelnen Mandanten individuelle Anpassungsvorgaben, bei denen bereitgestellten Standardobjekten mit Hilfe spezifischer Parameter umgesetzt werden. Darüber hinaus wird der Fokus auf eine bedarfsgerechte Auswahl und Parametrisierung der im On Demand Ecosystem verfügbaren Module (Prozessmusterbibliotheken) gelegt. Da der zugrundeliegende Sourcecode nicht änderbar ist bzw. kein direkter Zugriff besteht, erfolgt das Customizing durch einen Konfiguration Wizard.

Dabei werden die Systemänderungen durch den Wizard validiert und in einem abstrakten Satz von Instruktionen (Anweisungen), als mandantenspezifische Metadaten, erfasst. Die dadurch entstandenen Metadaten beschreiben die Konfigurationen, die mittels eines Laufzeitinterpreters deployed werden. In der Regel werden folgende Bereiche berücksichtigt [CC06]:

- *Benutzeroberfläche* - für die Erfüllung individueller Anforderungen hinsichtlich der Corporate Identity (CI), können benutzerspezifische Darstellungen in Bezug auf Farbschemata, Anzeigenreihenfolge und andere grafische Elemente berücksichtigt werden.
- *Workflow und Business Rules* - mit einer Workflowengine können Aktionen auf der Grundlage individueller Prozesse durch Aktionsregeln und Genehmigungsprozesse automatisiert werden.
- *Datenmodellerweiterungen* - in Anlehnung an vorhandenen Objektklassen und Relationen können neue Objekte und Strukturrelationen erstellt werden.

- *Prozessmusterbibliotheken* - sind branchenspezifische Funktionsbausteine, die in einem Repository<sup>12</sup> bereitgestellt werden und mittels Wizard vereinfacht in das Ecosystem integriert werden können.

### 2.3.5. Möglichkeiten der Softwareintegration

In weiterer Folge muss auch beachtet werden, dass die spezifischen Funktionalitäten (durch einen Remote-Zugriff per Internet) gewünschte Geschäftsoperation ausführen, um bestimmte Geschäftsfunktionen zu stützen. Hier bedarf es einer nahtlosen Verzahnung mit der internen Systemlandschaft des Kunden. Folglich müssen die entfernten Anwendungen eine modulare und offene Architektur aufweisen, um mit clientseitigen Anwendungen integriert werden zu können. Dafür werden drei Möglichkeiten für eine Integration angeboten [SZC<sup>+</sup>07, S. 561 ff.]:

#### User Interface (UI) Integration

Da jede Anwendung ihre eigene Benutzerschnittstelle samt Zugriffssteuerung hat sind üblicherweise verschiedene Benutzeridentitäten und Kennwortinformationen notwendig. Hier kommt die Idee der Mashups ins Spiel, die über Zugriffsmethoden eine beliebige Kombination von Quelldiensten ermöglicht. Dabei werden für eine bestimmte Aufgabenstellung öffentlich zugängliche APIs von Webanwendungen miteinander kombiniert und dadurch neue Anwendungen geschaffen. Um herauszufinden, welche APIs bereits miteinander verbunden wurden und welche Services dabei entstanden sind, bietet die Mashup-Matrix von Programmableweb<sup>13</sup> einen sehr guten Überblick.

#### Prozess Integration

Ein Geschäftsprozess der durch die bereitgestellte Software gestützt wird löbt in der Regel einen weiteren Prozess aus, der durch eine andere Anwendung gestützt sein kann. Zum Beispiel sollte ein, durch die CRM Anwendung gestützter, Auftragsprozess einen Erfüllungsprozess in einer ERP-Anwendung auslösen. Folglich müssen die Geschäftsprozesse - über Firewallgrenzen hinweg - durchgängige end-to-end Geschäftsprozesse unterstützen, um eine virtuelle Sicht auf das ganze Unternehmen zu erhalten.

#### Daten Integration

Die Geschäftsdaten sind Referenzdaten zur Beschreibung grundlegender Elemente eines Unternehmens. Daten über Produkte, Preisstrukturen, Verträge, Kunden sowie Lieferanten werden kollektiv als Stammdaten bezeichnet.

---

<sup>12</sup>vgl. Salesforce Appexchange <http://www.salesforce.com/appexchange>

<sup>13</sup>vgl. <http://www.programmableweb.com/matrix>

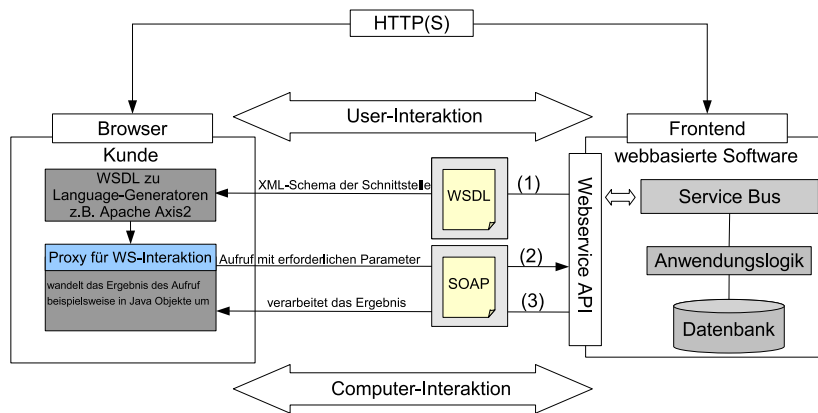
Für die Bewahrung der Konsistenz sind die Stammdaten in der Regel einem internen ERP-System zugeordnet. Die initialen Daten müssen über eine zugängliche API synchronisiert werden können, damit sichergestellt wird, dass die Datenbestände, in der remote-zugänglichen Anwendung mit dem internen ERP-System, aufeinander abgestimmt sind.

### 2.3.6. SOAP-basierte Kopplung der Software

Je stärker sich die Welt der Anwendungen in den Onlinebereich verschiebt, desto notwendiger gestaltet sich der Bedarf nach einem standardisierten Format für die Integration von webbasierten Anwendungen innerhalb und außerhalb des Unternehmens. Dabei kann, ohne ein gemeinsames Verständnis über technische Realisierungsdetails, die entfernte Softwarenutzung im Rahmen der Softwarebereitstellung als ein proprietäres und nicht integrierbares Lösungsmodell dargestellt werden. Daher stehen und fallen mit den bereitgestellten Anwendungen die eine API zur Verfügung stellt, die Integrationsmöglichkeiten [vG08, S. 36]. Aus diesem Grund gilt es im Vorfeld genau zu prüfen, welche Möglichkeiten die API im Rahmen eines Integrationsvorhabens ermöglicht. Im Idealfall bietet eine API zu einer bereitgestellten Anwendung mindestens dieselben Datenmanipulationsmöglichkeiten, wie sie durch die Bedienung durch das User Interface bereitgestellt werden [vG08, S. 36]. Das heißt, dass die Daten beliebig gelesen und geschrieben werden können, sofern sich die zugängliche API an normierten Standards<sup>14</sup>, z.B. in Bezug auf semantische Schnittstellendefinitionen orientiert. Die etablierten Industriestandards SOAP und WSDL können die Komplexität der Schnittstellen-Infrastruktur verringern und auf allen Ebenen, also von der Datenintegration bis hin zur Geschäftsprozessintegration, durch jede denkbare Integration eine ganzheitliche Sicht auf Informationen ermöglichen [Jac07]. Dafür geeignet sind XML-orientierte Beschreibungssprachen, die die Grundlage zur Spezifizierung von Schnittstellen liefern, um über Rechenzentrumsgrenzen hinweg gewünschte Funktionsaufrufe, betriebssystem- als auch programmiersprachenunabhängig, durchführen zu können. Es erfolgt über das Web ein zustandloses Interaktionsmodell, das mittels einem nachrichtenbasierten Interaktionsprotokoll (SOAP), synchrone oder asynchrone Funktionsaufrufe der regional entfernten Software mit XML-Daten ermöglicht. Dabei liefert die Schnittstelle der entfernten Software (Webservice-API), anhand der WSDL, eine Beschreibung der Funktionalitäten (Porttypen, Operationen) und des Datenschemas (Nachrichtentypen, Datentypen) ab und spezifiziert wiederum in XML die Funktionen und die zu unterstützenden Datenformate. Deswegen ist ein Webservice autonom. Das heißt, dass eine Nachricht die von einem adressierten Webservice verarbeitet wird, nicht beeinflusst werden kann, da die Details, durch die Bindung an eine konkrete Sprache, nicht einsehbar sind

---

<sup>14</sup>Siehe Web Services Interoperability Organization (WS-I) <http://www.ws-i.org/>



**Abbildung 2.7.:** Webservice Technologie

Die in der WSDL definierten Nachrichten bzw. Operationsaufrufe (definierte Nutzdaten - also der eigentliche Inhalt) erleichtern die Abwicklung des Datenaustausches per SOAP, der durch die Einbindung in HTTP-Aufrufe mit SSL abgesichert werden kann. Zusätzlich sollte das Webservice-Framework, neben der Unterstützung mehrerer Transportprotokolle, in der Lage sein, als Server zu fungieren, Klassen als Webservice anzubieten sowie einen Client-Teil zur Verfügung zu stellen, um auf Services zugreifen zu können [SD07, S. 55]. Mit Hilfe von Tools können aus WSDL-Spezifikationen Client-Zugriffsklassen generiert werden und aus Schema-Spezifikationen Data Bindings erstellt werden. Unter Verwendung eines `wsdl2language`-Generator (z.B. `org.apache.axis.wsdl.WSDL2-Java`) können z.B. Java-Zugriffsklassen erstellt werden, die zusätzlich als `Proxy`<sup>15</sup> verwendet werden. Dabei werden die einzelnen WSDL-Parameter wie Datentypen, Nachrichten und Schnittstellen in die entsprechenden Java-Objekte umgewandelt. Durch den Proxy werden dann die Parameter im richtigen Format in einen SOAP-Call an den Webservice übermittelt. Außerdem verarbeitet der Proxy das Ergebnis des SOAP-Calls und wandelt es beispielsweise wieder in ein Java-Objekt um.

Durch die Verwendung der bereits bestehenden und weit verbreiteten XML-Standards wird nun die notwendige Interoperabilität im Bereitstellungskontext

<sup>15</sup>Der Proxy oder auch SOAP-Engine ermöglicht die Interaktion mit dem Webservice durch die Konfiguration der HTTP-Verbindung und liefert Angaben über spezielle Verbindungsparameter für den Service. Dies kann eine Java-Klasse mit einer Methode sein, die alle für den Aufruf des Service erforderlichen Parameter übergeben bekommt.



ermöglicht, da das Konzept der SOAP-basierten Webservices die Koppelung mit einer standardisierten Schnittstelle (WSDL) ermöglicht und damit die Effizienz der ausgelagerten Funktionalitäten fördert. Auf der Basis einer ausgereiften Integrations-Middleware und einer dafür ausgerichteten Architektur, kann die Software besser und kontrollierter in die interne Anwendungsumgebung integriert werden [Jac07]. Vor allem dann, wenn die kundenseitige Architektur über entsprechend standardisierte XML-Schnittstellen verfügt und einen integrierten Workflow Host (z.B. MS Biztalk Server) für die prozessgesteuerte Koppelung mit einer XML-basierten Workflowsprache, wie z.B. BPEL4WS (Business Process Execution Language for Web Services), berücksichtigt, kann anhand spezifizierter Bedingungen, die über eine URL zugängliche Webservice-API (WSDL) für die technologische Stützung der end-to-end Geschäftsprozesse integriert werden.

Es wird nun eine neue Form der Nutzung von Software möglich, indem Kundensysteme auf die angebotenen Dienste bei Bedarf auch aus der Ferne zugreifen können [B<sup>+</sup>06, S. 19]. In diesem Zusammenhang ermöglicht vor allem das Designprinzip einer serviceorientierten Architektur die Nutzung bereitgestellter Software ohne auf die Vorteile einer integrierten Unternehmenssoftware verzichten zu müssen.

## 2.4. Konstituierende Merkmale und Implikationen

Die Begrifflichkeiten im Umfeld internetbasierter Geschäftsmodelle werden in der Regel nicht eindeutig verwendet, bzw. nicht eindeutig voneinander abgegrenzt, da gleiche oder ähnliche Sachverhalte keine klaren Unterscheidungsmerkmale zulassen. So wird neben den Begriffen ASP und SaaS, vielfach auch der Ausdruck Software On Demand (SOD) - im Zusammenhang mit Service Oriented Computing (SOC) - und Utility Computing<sup>16</sup> - verwendet. Erstere Begrifflichkeiten bezeichnen dabei die jeweilige Ausprägung mit der die Softwarenutzung als Dienstleistung im Sinne von SOD praktiziert wird, und die für das SOC das zugrundeliegende Realisierungskonzept liefert. Daher wird für diese Arbeit SOD als Muster für alle softwarebezogenen Bereitstellungskontexte herangezogen, wobei der konstituierende Begriff Software, wie eine CRM-Anwendung, als Service angeboten wird und dessen Nutzung somit dem uno-actu-Prinzip folgt: Produktion und Konsumption der Leistung fallen zusammen [Hal07, S. 74]. Das bedeutet nun, dass Hersteller von CRM-Software den Softwareentwicklungsprozess speziell für die Bereitstellung über das Web ausrichten, so dass On Demand über zentrale Server (Utility Computing Um-

---

<sup>16</sup>“Utility Computing is a collection of technologies and business practices that enables computing to be delivered seamlessly and reliably across multiple computers. Moreover, computing capacity is available as needed and billed according to usage, much like water and electricity are today” [RW04, S. 6].



gebung) eine webbasierte CRM-Software samt allgemeinen Funktionen und vorkonfigurierten Workflows, basierend auf einer definierten Nutzungsgebühr, für die sofortige Nutzung zur Verfügung steht. Eine weitergehende Betrachtung verdeutlichen hierfür die impliziten Prämissen [Hal07, S. 74 f.]:

### **Hosting**

Die CRM-Software wird direkt vom Anbieter bzw. von einem Dienstleister betreut und gewartet. Der Nutzer wird nicht dazu aufgefordert die Software selbst zu installieren oder zu betreiben, sondern bleibt von den üblichen Wartungsaufgaben befreit. Damit die Software jeweils in Abhängigkeit des aktuell auftretenden Lastaufkommens in einer gleichbleibenden Verfügbarkeit und Qualität, bereitgehalten werden kann, ermöglichen dabei die Technologien des Service Oriented Computing (SOC) ein autonomes Systemverhalten.

### **Publikation**

Der Softwareanbieter stellt eine CRM-Software über das Internet zur Verfügung und nutzt damit die gängigen Vorteile, wie das Aufsetzen auf bereits etablierte Standards oder die hohe Verbreitung des Internets. Ziel ist stets eine Vereinheitlichung, die sowohl die Implementierung als auch die Wartung im laufenden Betrieb verbessern soll.

### **Mandantenfähigkeit**

Der Definition folgend, besteht die Idee der Multi-Tenant Architektur darin, die CRM-Software mehreren Kunden anzubieten und auf die Anforderungen des Kunden individuell anzupassen. Eine zentrale Codebasis sorgt für einen entsprechenden Veredelungs- und Individualisierungsgrad.

### **Veredelung**

Der Anbieter kann sich bei der Erbringung seiner Leistung auf andere, ihm zugängliche Webservices beziehen und diese zu einer höherwertigeren kombinieren.

### **Entgelt**

Bei der Inanspruchnahme der CRM-Software findet nicht das klassische Kaufmodell Anwendung, welches den einmaligen Erwerb von Nutzungsrechten vorsieht. Vielmehr fließt der tatsächliche Nutzungsgrad bei der Kalkulation mit ein.

# 3. On Demand-Prinzipien als Geschäftsmodell

Mit der Beschreibung des On Demand Geschäftsmodells wird eine Grundlage für die weitergehende Betrachtung der Softwarebereitstellung geschaffen, um durch die Abgrenzung zu ASP, die Erfolgchancen für SaaS abzuleiten. Dabei werden die zu konstituierenden Merkmale durch eine des Service Oriented Computing realisierbaren Architekturvariante analysiert um damit die Handelbarkeit der Softwarebereitstellung als auch -nutzung zu erweitern.

## 3.1. Begriffsbestimmung: Software On Demand

Im Folgenden gilt es den Begriff “Software On Demand” zu erläutern. Der Zusatz “On Demand” umfasst zunächst ein temporäres, auslastungsorientiertes und den Anforderungen entsprechendes Nutzungsrecht, wobei die Software nicht mehr im konventionellen Sinne käuflich ist, sondern mittels transaktionspezifischem Nutzungsmodell bereitgestellt wird. Dieses Konzept, erlaubt für einen vergleichsweise kleinen organisatorischen Aufwand einen Zugriff auf die bereitgestellte Software, da Updates, Wartung und Sicherheitsfragen zentral vom Anbieter gelöst werden [LET06, S. 164 f.]. Der Gedanke besteht in der Entlastung des Anwenders von Sekundäraufgaben (wie der Installation, Wartung, Administration und der Aktualisierung von Software) und damit in der Kostenreduzierung. Dabei muss bei der Erzielung eines hohen Nutzenpotenzials die zugrunde liegende Hardware- und Softwareinfrastruktur, in Bezug auf die geforderte Flexibilität und Geschwindigkeit mit der sich Märkte und deren Produkte verändern, immer eine am Unternehmen und dessen Entwicklung orientierte Nutzung erlauben.

Folgende Tabelle soll daher in Anlehnung an ( [UII04, S. 47 f.]; [BIT06d]) einen Überblick über das On Demand Modell bieten.

On Premise	On Demand
<i>Hardwareausgaben</i> (Web Server, Datenbank Server, Portal Server, Application Server)	In der Nutzungsgebühr enthalten. Es fallen nur Betriebskosten für die Clientrechner an.

### 3.1. BEGRIFFSBESTIMMUNG: SOFTWARE ON DEMAND

<i>Software-Lizenzen und -Verwaltung</i> (Softwarelizenz, Datenbanklizenz, Portalplattformlizenz, Updates und Upgrades)	In der On Demand Gebühr enthalten. Updates werden regelmäßig eingespielt und für gewöhnlich werden Vierteljährig große Upgrades durchgeführt.
<i>Konzept/Implementierungskosten</i> ( Installation und technische Implementierung)	Kurzer Rollout, da die On Demand Lösung sofort nach Vertragsabschluss genutzt werden kann.
<i>Netzwerk-Infrastruktur</i> (Switches, Router )	Für die Nutzung der Funktionalitäten ist nur eine Internetverbindung notwendig.
<i>Administrationskosten</i> (Wartung und Monitoring sowie Pflege der Systemlandschaft unter den Faktoren Hochverfügbarkeit, Backups, Archivierung, Nutzeradministration )	Ist in der Nutzungsgebühr inkludiert.
<i>IT-Training</i> (Weiterbildungs- und Ausbildungssysteme)	Best Practices werden über Communities bzw. direkt vom Anbieter bereitgestellt. Je nach Anwendungsfall werden Tutorials und Onlinehilfen für verschiedene Benutzertypen wie Standardbenutzer, Administratoren und Manager angeboten.
<i>IT-Support</i> (Problemlösungsmanagement durch IT-Helpdesk)	In der Nutzungsgebühr inkludiert

**Tabelle 3.2.:** Gegenüberstellung On Premise und On Demand

Insbesondere IBM<sup>1</sup> gilt als einer der Initiatoren des On Demand Begriffs [SJ02]. Geprägt durch IBM CEO Sam Palmisano sollen Serviceleistungen für Kunden auf Abruf zur Verfügung stehen, damit diese auf neue Geschäftsanforderungen, bzw. auf veränderte Rahmenbedingungen, reagieren können [Bar05]. Somit ist Software On Demand ein Sourcingmodell, das eine - dem Geschäftsverlauf angemessene - Kostensituation schafft, in dem es die betriebswirtschaftliche Funktionalität für die Ausführung geschäftlicher Aufgaben in den Vordergrund rückt und gleichzeitig die Dominanz von rein systemtechnischen oder technologischen Aspekten reduziert. Wegen der aktuellen Entwicklungen in Bezug auf die geforderte Flexibilität und Geschwindigkeit, mit der sich Märkte und deren Produkte verändern, ermöglicht die flexible Inanspruchnahme von On Demand Software eine am Unternehmen und an dessen Entwicklung orientierte Nutzung sowie im Voraus berechenbare monatliche Kosten pro Arbeitsplatz. Deshalb fördert diese Geschäftsmodell implizit die Unverbindlichkeit,

<sup>1</sup>Für weitere Informationen siehe <http://www-5.ibm.com/e-business/de/>

mindert das Risiko und ermöglicht einen Wechsel, falls sich die Entscheidung als Fehlgriff herausstellen sollte [S<sup>+</sup>06, S. 2]. Die daraus resultierende Reduzierung der direkten Softwarekosten und deren Umwandlung in ein nutzungsorientiertes Vergütungsmodell<sup>2</sup> führen in weiterer Folge zur Senkung der Total Costs of Ownership (TCO) [Hal07, S. 82].

#### 3.1.1. Vorteile gegenüber klassischen Ansätzen

Während die im On Premise<sup>3</sup> Betrieb ausgelegte Softwarelizenzierung notwendige Installationen samt Update und Upgrademechanismen bei gleichzeitigem Management der Infrastrukturlandschaft (Server und Datenbanken) vorsieht, führt der zunehmende Kosten- und Flexibilisierungsdruck, unter dem weitreichenden Aspekt der Implementierungs- und Infrastrukturverlagerung, zur Etablierung neuer Alternativlösungen. Hinzu kommt, dass durch den Kostendruck auf die zentralen IT-Organisationen, bei gleichzeitiger Optimierung der Prozessunterstützung, On Premise Rollouts mit zahlreichen unvorhergesehenen Herausforderungen einher gehen [Ste07], da

- die einzuführende Anwendung sich komplexer darstellt als erwartet;
- zu wenige Mitarbeiter für die Aufrechterhaltung der geschäftskritischen Performance zur Verfügung stehen;
- Anwenderakzeptanz und -kompetenzen unterschätzt werden; und
- die Folgekosten für Wartung, Systemsicherheit und unvorhersehbare Ausfälle nur schwer kalkulierbar sind.

Gerade weil die Komplexität der verwendeten Anwendungen immens angestiegen ist und sich demgegenüber die Halbwertszeit aktueller Hard- und Software immer schneller verringert, vereinfachen gemietete und ausgelagerte Dienstleistungen den Betrieb und Unterhalt von Software samt Infrastruktur [Gab07]. Da auch das Geschäftsmodell der Anbieter von der sicheren Bereitstellung abhängig ist, kann davon ausgegangen werden, dass die Sicherheitsstandards wesentlich höher sind, bzw. die Verfügbarkeit wesentlich leichter gewährleistet werden kann, als bei einer internen Infrastruktur. Aus Kundensicht stellen daher die geringe Kapitalbindung, die Möglichkeit zur nachhaltigen Kostenreduktion und die Komplexitätsreduzierung - durch Fremdvergabe der gängigen Betriebs- und Entwicklungsaufgaben - einen großen Vorteil dar [Hal07, S. 81 f.]. Dabei haben die Anwender bei einer Fixgebühr (Flatrate), unabhängig von der Häufigkeit und Dauer der Nutzung, immer gleich bleibende Fixkosten in

---

<sup>2</sup>Im Internet existieren auch On Demand Angebote ohne einen expliziten Vergütungsspekt. Beispielsweise kann die Nutzung einer Suchmaschine als "On Demand Prinzip" aufgefasst werden.

<sup>3</sup>Softwarebetrieb im eigenen Rechenzentrum.

Form von linearen Nutzungsgebühren, welche durch folgende Parameter die Preisgestaltung bestimmen (vgl. [Küc01, S. 173]; [Mül07, S. 2]):

- Skalierbarkeit der Software in funktionaler Dimensionierung;
- Definition der Service Levels in Bezug auf Reaktionszeiten, Verfügbarkeiten, Riskmanagement, Disaster Recovery und Backupstrategien;
- Anzahl der User;
- Umfang des Customizingrahmens;

Je nach Anzahl der User und dem Umfang der genutzten Funktionen können zirka 60 bis 140 Euro pro Nutzer und Monat vom Anbieter verrechnet werden. Verglichen mit einem Basispreis von etwa 1500 bis 3500 Euro pro Nutzer- zuzüglich eines jährlichen Wartungsaufschlags von etwa 15 Prozent der Kaufsumme einer Software und den internen Kosten für den Betrieb - wird in der Regel die On Demand Software günstiger sein, wenn beim Anwender, im Fall des Kaufs, erst die Infrastruktur für die Lösung geschaffen werden müsste [Kub07]. Würden sowohl personelle als auch in der IT-Infrastruktur vorhandene Ressourcen genutzt werden und die längerfristige Planung keine großen funktionalen Upgrades vorsehen, so könnte längerfristig gerechnet, der On Premise Betrieb der Software preiswerter sein. Um dieser Preisfrage nachzugehen wurden schon unter dem Schlagwort ASP (Application Service Providing) unterschiedliche Beispielrechnungen für unterschiedliche Firmengrößen und Anwendungsszenarien durchgeführt, die natürlich auch in den Ergebnissen divergierten [Gro07, S. 38]. So führte zum Beispiel Morgan Doyle eine detaillierte Kalkulation der Gesamtkosten (TCO) an einem konkreten, aber anonymisierten Unternehmen aus Großbritannien mit 500 Anwendern durch und kam zu dem Ergebnis, dass mit einer On Demand Lösung Kosteneinsparungen in der Höhe von mehr als 1,7 Millionen Euro realisiert werden könnten [Doy99, S. 3]. Die größten Einsparungspotentiale wurden laut Morgan Doyle, im Wegfall der Anschaffungs-, Implementierungs<sup>4</sup>- und Wartungskosten<sup>5</sup> realisiert, die unter anderem zur Steigerung der Unternehmensperformance sowie zur verbesserten Produktivität und Infrastruktur mit geringen Ausfallszeiten führte [Doy99, S. 3]. Der TCO pro Anwender und Jahr sank von ca. 6000 Euro auf ca. 2700 Euro. Daher wird, neben den Kosteneinsparungen, vor allem der Innovationsfähigkeit von On Demand Software eine große Bedeutung beigemessen, da die Umwandlung von hohen und fixen Investitionskosten in kleinere, monatlich zu entrichtende variable Entgelte, den Einsatz von innovativen, aber ansonsten sehr kostenintensiven, Lösungen erlaubt.

<sup>4</sup>Dabei sind nach Holm [Hol98] die Kosten für Installation, Integration, Administration und Pflege der geschäftskritischen Anwendung grundsätzlich signifikant höher als die Anschaffungskosten [Tam03, S. 182].

<sup>5</sup>Triple Tree geht davon aus, dass ein Großteil des TCOs auf die Administration und das Management der Applikation entfallen, was sich in hohen Personalkosten widerspiegelt [G+06, S. 9 f.].

#### 3.1.2. Nachteile von On Demand Software

Die Entscheidung eine geschäftskritische Anwendung über den Browser zu nutzen, birgt neben den Vorteilen, wie ortsunabhängige Verfügbarkeit und ein optimiertes Kosten- und Nutzenverhältnis auch potentielle Schwächen und Risiken. Die Nachteile werden, aus Kundensicht im Bereich Security und Akzeptanz gesehen [Hal07, S. 81]. Vor allem wegen der Abhängigkeit von einem Anbieter während der Vertragslaufzeit und den geringen Kontrollmöglichkeiten lässt sich schwer einschätzen, inwieweit der Anbieter in der Lage ist, eine adäquate Sicherheitspolitik zu betreiben. Die Bedenken hinsichtlich der Abhängigkeit von der Internetanbindung sind nicht unbegründet, da die zur Verfügung stehende Netzinfrastruktur ein unabdingbares Glied in der Leistungskette des On Demand Computings darstellt. Die Bandbreite ist zwar, dank der flächendeckenden Verfügbarkeit, weit weniger einschränkend als früher [vG07]. Jedoch sind die Antwortzeiten von den ausgelagerten Systemen immer vom augenblicklichen Status der Netzwerkverbindung abhängig.

Für einen wertsteigernden Beitrag genießen daher Sicherheit und Qualität stets oberste Priorität. Dabei beschreiben die Service Level Agreements (SLAs), basierend auf einem ausgewogenen Verhältnis zwischen Kosten und Nutzen, zunächst die Randbedingungen und die einzelnen Konditionen, unter denen das Vertragsverhältnis zwischen Anbieter und Kunde geschlossen wird. Je nach angesprochener Zielgruppe und dem Charakter der Anwendung, variieren auch der Umfang und die Detailliertheit der SLA. Im Rahmen der Bereitstellung einer komplexen und geschäftskritischen Umgebung, sollte das Vertragswerk für die Spezifizierung der Serviceleistungen - in Anlehnung an einen erwartungskonformen Betrieb - nach allgemeinen Industriestandards (Verfügbarkeit 99 %) erfolgen und mit einer zuvor festgelegten Höhe an etwaigen Vertragsstrafen, bei Nichteinhaltung der SLAs, ergänzt werden [Köh07b, S. 122]. An dieser Stelle kann es leicht zu einem Interessenskonflikt zwischen Anbieter und Kunde kommen, da der Anbieter in der Regel die gewünschte Servicespezifizierung und auch die monetäre Kompensation in die preisliche Kalkulation einfließen lässt, während der Kunde sich mit einem günstigen Serviceangebot - in Erwartung einer maximalen Betriebszeit und geringen Lock-In-Effekten (Switching Cost) - nicht zu sehr einer Abhängigkeit vom Anbieter verfallen möchte.

Neben den rechtlichen Bedenken sind auch noch organisatorische Ungereimtheiten zu lösen, denn so positiv es auch erscheinen mag, dass die Software per Internet einen flexiblen Zugriff ermöglicht, so darf auch nicht die Kehrseite der Internetnutzung verschwiegen werden. Hierbei sind einerseits verschiedene Browsertypen und -versionen sowie gegenfalls zu installierende Plug-ins zu beachten. Andererseits bedarf es einer genauen Evaluierung der Schnittstellen für hybride Konnektivitäten, da sonst die Gefahr von Redundanzen, semantischer

Integrität und Inkonsistenzen besteht. Daher gilt im Rahmen der Vertragsvereinbarungen zu klären, wer die Verantwortung für Daten übernimmt, wenn diese verloren gehen, beschädigt werden oder in fremde Hände gelangen. Als nicht unwesentlicher Punkt ist auch der Datenrückfluss zu gewährleisten. Das heißt, nach Beendigung der vertragsbedingten Dienstleistung müssen Maßnahmen berücksichtigt sein, die eine problemlose Datenrückfuhr garantieren können.

### 3.2. Markt im On Demand Umfeld

Der Markt im On Demand Umfeld ist durch mehrere Entwicklungsschritte geprägt. Im Zuge dessen gestaltet sich der Markt für das On Demand Geschäftsmodell sehr dynamisch; d.h. die Softwarehersteller portieren ihre Produkte nicht einfach nur in das World Wide Web, sondern fahren dabei auch unterschiedliche Strategien, die gemäß zu unterschiedlichen Leistungsdarstellungen führen. Zudem sind der zeitliche Kontext und die technischen Rahmenbedingungen, wie beispielsweise Internetnutzung und Bandbreite, zu beachten [Hal07, S. 76]. Hier lässt sich auch SaaS einreihen, mit der Idee, Geschäftslösungen in einer schnell modifizierbaren Weise als Service von Dritten zu beziehen. Dabei suggerieren ASP und SaaS die gleichen Wertversprechen, da SaaS unter Nutzung aller Möglichkeiten und Vorteile der internetbasierten Software- und Systemarchitekturkonzepte, auf die gewonnenen Erfahrungen des ASPaufbaut [Vor04]. Für eine Abgrenzung des Begriffs Software-as-a-Service (SaaS) ist an dieser Stelle angebracht auf den verbreiteten Ausdruck des Application Service Providing (ASP) einzugehen.

#### 3.2.1. Entstehung und Verbreitung von ASP

Die ersten ASPs, die während der e-business Euphorie rund um die Jahrtausendwende auf den Markt getreten sind, stammten hauptsächlich aus dem Bereich der Internet Service Provider, Systemintegratoren und Softwarehersteller, die durch strategische Partnerschaften die internetbasierte Bereitstellung von hauptsächlich geschäftskritischer und proprietärer Software forcierten [Göt06, S. 96]. Je nach strategischer Ausrichtung wurden hierfür die adressierten Hosting- und Dienstleistungsangebote an verschiedenen Zielgruppen ausgerichtet. Dabei wurden hauptsächlich Geschäftsanwendungen, die nicht ursprünglich für die Internetnutzung ausgelegt waren, mit Hilfe einer "Access Infrastructure Software" webtauglich gemacht [NDS07, S. 546]. Daraus lässt sich ableiten, dass die Anbieter maßgeblich als Intermediatoren eines umfangreichen Anwendungsportfolios (wie z.B. Anwendungen mit einem integrierten oder aggregierenden Charakter) auf dem Softwaremarkt aufgetreten sind. Die Entstehung und Verbreitung von ASP - als (aus der Sicht eines Kunden) Outsourcing der Bereitstellung von Anwendungen bzw. als (aus der Sicht eines Dienstleistungsanbieters) Geschäftsmodell für die Bereitstellung von Anwen-



dungen für unternehmensexterne Kunden - ist auf der raschen und fortschreitenden Entwicklungen der Internettechnologie in den 90iger Jahren zurückzuführen [Jac05, S. 232]. Dabei wurde Application Service Providing durch das unternehmensübergreifende ASP Industrie Konsortium (*ASPIC*) und das ASP Konsortium der Information Technology Association of America (*ITAA*)<sup>6</sup> folgendermaßen definiert [Mus01]:

*Application-Service-Providing (ASP) beinhaltet die Bereitstellung (Providing) einer Anwendung (Application) als Dienstleistung (Service) für eine Vielzahl von Kunden, die für eine bestimmte Benutzungsgebühr per Netzverbindung auf die Anwendung Zugriff haben. Der Dienstleister sorgt für die Software-Lizenz, die Wartung und die Aktualisierung der Software und stellt einen Benutzersupport zur Verfügung.*

Die Variante des Thin-Client- oder Server-Based-Computing war zunächst die meist favorisierte Methode der ASP. Proprietäre Übertragungsprotokolle, wie Remote Desktop Protocol (RDP) von Windows oder Independent Computing Architecture (ICA) von Citrix, ermöglichen dabei einem Terminalclient die Transformation der Ein- und Ausgabe durchzuführen, so dass sich die Interaktion mit der Anwendung so verhält, als wäre dies "vor Ort" zugänglich. Das Terminalserver-Modell bot schon zuvor die Möglichkeit, bestimmte Standardsoftware über das Netz an mehrere Kunden zu verteilen. Oft wurde aber die Tauglichkeit für solch einen Betrieb überschätzt. Darüber hinaus musste auch berücksichtigt werden, dass die per Terminalclient zugängliche Software auch Einschränkungen hinzunehmen hat bzw. hatte:

- Nutzbarkeit der Software ist nur durch einen bestimmten Terminalclient möglich.
- Die Lizenzen für die Benutzung der Terminals müssen von den Kunden getragen werden bzw. müssen vom Client lfd. upgedatet werden.
- Zahlreiche Anwendungen sind im Mehrbenutzermodus des Terminal Servers nicht lauffähig.
- Bei wachsender Benutzerzahl entstehen hohe Anforderungen an die Serverressourcen.
- Die Bindung an ein eigenes Netzwerkprotokoll (ICA und RDP) erfordert eine hohe Bandbreite für die performante Übertragung von multimedialen Komponenten sowie Druckerdatenströme.

---

<sup>6</sup>ITAA: <http://www.itaa.org/> Das Konsortium wird jetzt unter dem Namen Software as a Service (SaaS) SITTA weitergeführt: <http://www.siaa.net/default.asp>



Bei schmalbandigen Verbindungen sind hier deutliche Verzögerungen bei der Bedienung aufgetreten, was zu Akzeptanzproblemen auf Seiten der Anwender führte. Der schwache integrative Charakter führte auch dazu, dass die gesamte Breite der integrierten Anwendungen auf den Server ausgelagert werden musste. Dieser Aspekt steht allerdings in einem gewissen Widerspruch zur Idee des ASP, da schlussendlich viele Systeme parallel betrieben werden mussten [Jac05, S. 103]. Entweder verteuerte sich der Betrieb oder die Application Service Provider konnten keine Skalenerträge mehr erwirtschaften was in der Regel zu fehlenden Finanzierungsmöglichkeiten, hinsichtlich der Aufrechterhaltung des Leistungsbetriebs führte. Zudem wurden, ohne die Chancen und Grenzen der Softwarebereitstellung zu beachten, Softwarelösungen angeboten, deren Softwarearchitektur zwar eine mandantentaugliche Bereitstellung im Sinne einer Mehrfachbenutzbarkeit ermöglichte, sich schlussendlich aber zu komplex gestaltete [App02, S. 52 f.]. In der scheinbaren Einfachheit dieses Ansatzes lag auch die größte Schwäche des ASP-Modells, da die Dienstleister aufgrund von fehlendem Expertenwissen bzgl. des Leistungsumfangs mit vielen individuellen Problemen konfrontiert wurden ([Hel07]; [VSTT06, S. 276]).

*"In particular, enterprise-class applications designed for licensing in a client/server architecture have not been ideally suited to be delivered in an on-demand model, leading to a 'customization trap' in attempting to exploit application efficiencies across a 'one-to-many' dimension" [H<sup>+</sup>04, S. 4].*

Beispielsweise ist das SAP R/3 (ehemaliges Hauptprodukt des deutschen Softwarehauses SAP) ein grundlegend mehrmandantenfähiges System für ERP das aber im Rahmen des Customizings mandantenübergreifende Datenbanktabellen für die Parametrisierung<sup>7</sup> verwendet [Hot02, S. 63 ff.]. Dies hatte zur Folge, dass Individualisierungen bzw. Updatemechanismen die mandantenspezifischen Parametrisierungen zum Teil manipulierten. Daher wurden für solch explizite Systeme die Möglichkeiten für des Customizings bewusst restriktiv gehalten oder dezidiert im Single Tenant Modus bereitgestellt, damit eine vermeintliche Fehlbedienung und ein daraus resultierender Programmfehler nicht die Kapazitäten des Client-Server-Systems einschränkte oder gar das Gesamtsystem beeinträchtigte.

Erfahrungen rund um ASP seit der e-business Euphorie zur Jahrtausendwende haben nun deutlich gemacht, dass die Inanspruchnahme derartiger Softwarefunktionalitäten sehr wesentlich davon abhängt, ob ASP Anbieter ihren Kunden neben den funktionalen Anwendungsservices auch verbindlich darstellen können, wie sie die bedarfsgerechte Bereitstellungsqualität im Sinne einer

---

<sup>7</sup>Dabei werden die Einstellungen der einzelnen Mandanten in der für das Customizing vorgesehenen Datenbank gespeichert und zur Laufzeit (entsprechend dieser Parametrisierung) ausgeführt.

Service-Verfügbarkeit und -Sicherheit gewährleisten können [BIT06d, S. 9]. Es geht daher bei einem erfolgreichen ASP-Angebot nicht nur darum, Software über das Internet anzubieten, sondern dem Kunden durch weitere Dienstleistungen einen zusätzlichen Mehrwert zu verschaffen [Die02, S. 94]. Dabei ergeben sich, je nach Zielgruppe, unterschiedliche Erwartungshaltungen an den Anbieter. Folgende Tabelle (▷ vgl. Tabelle: 3.3 über den ASP Nutzen aus Kundensicht) gibt einen, nach Häufigkeiten der Nennung sortierten Überblick über den zu erwartenden Nutzen von ASP.

Nutzen	Unternehmen ( < 500 MA )	Unternehmen ( > 500 MA )
Reduzierung der Abhängigkeit von der IT-Abteilung	+	+
Kosteneinsparung und bessere Kostenkontrolle	+	+
Einfachere und schnellere Softwareeinführung samt anwendungszentrierter Dienstleistungen	-	+
Bereitschaft zum Test neuer Anwendungen	+	+
Vorantreiben der Softwarestandardisierung	-	+
Flexibilität, um mit IT-Anforderungen Schritt zu halten	+	+
Zugang zu bisher unerschwinglicher Software	+	-

**Tabelle 3.3.:** ASP Nutzen aus Kundensicht [Die02, S. 96]

Die in der Tabelle dargestellten Erwartungen wurden durch die zunehmende Etablierung von Internetstandards und dem Aufkommen von komponentenbasierter Software, unter Berücksichtigung der Webfähigkeit der Software als webbasiertes ASP, immer mehr erfüllt. Im Kontext der Webarchitektur bedeutet dies, dass die Zugriffsmethoden über Protokolle wie HTTP und Datenstrukturen durch Seitenbeschreibungssprachen (HTML, XML) so gestaltet sind, dass die Software entkoppelt, über eine standardisierte Schnittstelle (Browser und API), als webbasierter Service genutzt werden kann. [Wil06, S. 3]. Damit wurden einige der identifizierten Schwachstellen im ASP-Modell, wie beispielsweise die Überladung der Kundenlösungen, durch individuelle Anpassungen und Entwicklungen oder die hohen Nutzungsgebühren, berücksichtigt und beseitigt [Hal07, S. 82]. Dennoch sind weitere Maßnahmen einzuleiten, um die Thematik der Individualisierung sowie die Leistungsfähigkeit und Ska-

lierbarkeit solch standardisierter Leistungsangebote zu beantworten. Deshalb lässt sich die Dimension des Begriffs Software-as-a-Service aus einer Synthese der methodischen Auseinandersetzung der Softwarebereitstellung mit einem skalierenden Service für die Reduzierung der technologischen Komplexität und zur Stützung spezifischer Aufgaben darstellen.

### 3.2.2. Einordnung von Software-as-a-Service und Application Service Providing

Da die Grundidee von Software-as-a-Service (SaaS) die gleichen Prinzipien beinhaltet wie ASP, wird - in Anlehnung an Gartner Analyst Ben Pricing [Pri04, S. 2] - SaaS als eine konsequente Weiterentwicklung des ASP-Modells dargestellt [Las05, S. 72].

*“The term ‘ASP’, with its historical connotations of ‘dot-com hype, boom and bust’, is slowly being replaced by the terms software-as-a-service (SAAS) and, mostly importantly ‘on demand’” [Ben Pricing]*

Allerdings darf SaaS nicht als ein neues, revolutionäres und alles veränderndes Geschäftsmodell betrachtet werden. Vielmehr handelt es sich bei diesen Modell um eine evolutionäre Entwicklung, in dem die Hersteller die Verantwortung für die Implementierung und den Betrieb (Administration und Anwendungsmanagement) der eigenen Technologien, in Form eines Outsourcing-Vertrags übernehmen [MB07]. Genau dieses Prinzip wird auch dem On Demand Konzept zugrundegelegt, wo die Softwarehersteller ihre Anwendungsfunktionalitäten über das Internet, als Service, bereitstellen. Dabei stellt SaaS eine feingranulare Softwarebereitstellung dar, die aufgrund der technologiegetriebenen Entwicklung andere Paradigmen verfolgt und unterschiedliche Technologien als ASP verwendet (vgl. [Las05, S. 72]; [Hal07, S. 75]).

Die Abgrenzung zum klassischen ASP liegt hierbei in der Rolle des Anbieters von Softwareservices und in der Software an sich, da die Bereitstellung einer - speziell auf Skalierbarkeit und Mehrbenutzerbetrieb, entwickelte Webanwendung - direkt vom Hersteller erfolgt [Blo05]. Dadurch sind die Softwarehersteller für den reibungslosen Betrieb der Software verantwortlich, und dafür, eine schnelleres und besseres Service bereitzustellen, als es heute mit klassischer Software mit Maintenance-Vertrag möglich ist [Kub07].

Als zentrale Unterschiede werden die Möglichkeit der Abbildung von mehreren Kunden innerhalb der Software (Mehrmandantenfähigkeit) [vG07] und die Bereitstellung respektive Veröffentlichung der Software über das Internet, auf Basis von Webtechnologien, genannt [Hal07, S. 75]. Es ist an dieser Stelle auch darauf hinzuweisen, dass eine Abgrenzung zwischen SaaS und ASP im Hinblick auf einen stärker ausgeprägten Customizing-Aspekt denkbar ist. Es werden in

einer gemeinsamen Datenbankinfrastruktur alle kritischen Daten der Kunden, samt den zur Verfügung gestellten Objekten, Entitäten und deren Korrelationen gespeichert, was wiederum als Grundlage für die Individualisierung und Optionierung von CRM On Demand dient.

Um den soeben beschriebenen Unterschieden zwischen SaaS und ASP Rechnung zu tragen, werden die Merkmale von ASP, im Vergleich zu SaaS, in folgender Tabelle nochmals verdeutlicht.

	<b>ASP</b>	<b>SaaS</b>
<b>Software-zugriff</b>	Spezielle Zugriffsarchitektur, bspw. Terminalserver	webbasierter Zugriff per Browser
<b>Vorraussetzung</b>	PC mit installiertem Terminalclient oder internettauglichem Endgerät	Internettaugliches Endgerät
<b>Zielstellung</b>	Kostenreduzierung und Softwarestandardisierung durch Outsourcing der Software	Kostenreduzierung, Flexibilität, Standardisierung und kurze Softwareeinführung ("time-to-market")
<b>Geschäftsmodell</b>	Horizontales Geschäftsmodell: Der Anbieter agiert in der Regel als Intermediator zwischen Software- und Infrastrukturanbieter.	Vertikales Geschäftsmodell: Softwarehersteller vertreiben speziell für das SaaS-Modell entwickelte Software.
<b>Zentrale Bereitstellung</b>	Bereitstellung der Software erfolgt durch dezidierte Server bei einem ASP oder dessen Hostingpartner.	Bereitstellung erfolgt im Multi-Tenant Modell direkt vom Softwarehersteller oder dessen Hostingpartner.
<b>Abbildung von Kunden</b>	Nutzung kundenspezifischer Software (1:1-Beziehung).	Nutzung einer Software für alle Kunden (1:N-Beziehung)
<b>Software</b>	i.d.R. komplexe, spezifisch angepasste Software.	i.d.R. standardisierte Geschäftsanwendungen, welche spezifisch für das SaaS-Modell entwickelt werden.

	<b>ASP</b>	<b>SaaS</b>
<b>Nutzungs- gebühr</b>	monatlicher Fixpreis oder transaktionsbasierte Abrechnung.	monatlicher Fixpreis oder transaktionsbasierte Abrechnung.
<b>Kosten</b>	Standardleistungen werden durch die monatlichen Mietgebühren abgedeckt.	Gebühren richten sich nach dem Funktionsumfang und sind an der tatsächlichen Verwendung (Transaktion) oder der Anzahl der User (Subskription) angelehnt.
<b>Vertrags- bindung</b>	mittelfristige Vertragslaufzeiten (häufig 1 bis 3 Jahre).	unverbindliche Vertragslaufzeit (monatlich kündbar).

**Tabelle 3.5.:** Unterschiede ASP - SaaS

Die in der Tabelle dargestellten Eigenschaften von ASP und SaaS verdeutlichen, dass beide eine Ausprägung der On Demand-Idee darstellen und sich nur in wenigen Punkten unterscheiden. Dennoch befreit SaaS nun ASP aus der technologieorientierten Umklammerung der Softwarebereitstellung, in dem die Software, durch die Kopplung von konkreten und singulären Systemen, eine mandantenspezifische Virtualisierung ermöglicht. Insofern ist SaaS nicht nur als Erweiterung von ASP zu verstehen, sondern als ein serviceorientierter Ansatz mit Ausrichtung des Leistungserstellungs- und Angebotsmodells auf das Internet.

### 3.3. SOC als Basis für On Demand Computing

Da eine optimierte Hosting-Infrastruktur für das On Demand Geschäftsmodell auch von geschäftskritischer Bedeutung ist, wird durch die Konzentration auf die Kapazitäts- und Ressourcenverfügbarkeit von Server- und Speicher-Ressourcen, im Rahmen des an geschäftskritische Anwendungen (Verfügbarkeit 99,9 %) angelehnten Managements der Ressourcenbereitstellung und -konfiguration, die Bereitstellung der webbasierten Anwendung durchgeführt. Dabei ist die Standardisierung der Prozesse in der Leistungserstellung für webbasierte Dienstleistungen als ein wesentlicher Erfolgsfaktor festzuhalten. Mit

den standardisierten Prozessen wird definiert, wie die Hostinginfrastruktur entwickelt, angepasst und betrieben werden muß, um den Kundenanforderungen zu entsprechen. Ein Weg zur Gestaltung der Leistungserstellung stellt die Orientierung an Referenzmodellen ▷ vgl. Kapitel 3.3.2 für IT-Prozesse dar [WBK07, S. 4].

Darüber hinaus gilt es die Herausforderung zu berücksichtigen, ein variantenreiches Leistungsangebot mit möglichst hoher Effizienz und unter Ausnutzung entsprechender Skaleneffekte zu erbringen. Dafür ist eine robuste, allzeit verfügbare dynamische sowie vollständig redundante Multi-Tenant Infrastruktur (mandantenfähige Infrastruktur) erforderlich. Der operative Aufgabenbereich umfasst dabei die Auslastung von Server- und Speichersystemen, die Automatisierung typischer, wiederholt anfallender Managementaufgaben und eine statische sowie dynamische Zuordnung von Ressourcen [Ede07, S. 71].

Die zugrundeliegenden Technologien des serviceorientierten Computings (SOC) eröffnen daher Optionen zur Anpassung und zu globalen Leistungstiefenbestimmung, um den wachsenden Forderungen nach bedarfsorientierten, anpassbaren Leistungen begegnen zu können [WBK07, S. 5]. Durch die gezielte Gestaltung von Servicearchitekturen können - in Analogie zu einem Baukastenprinzip - die Softwareprodukte komponentenorientiert gestaltet werden [WBK07, S. 5]. Die für CRM On Demand relevante CRM-Software, kann daher aus einer Sammlung gekoppelter Funktionsbausteine bestehen, deren funktionaler Ablauf nach einer Service Orchestration Engine koordiniert wird und über den Multi-Tenancy Laufzeitinterpreter mandantenabhängig virtualisiert (Service-Instanziierung) wird. Die in kleinen, lose gekoppelten Bausteinen (Services) angebotenen Funktionalitäten, erlauben den nutzenden Unternehmen durch deren Kombination, schnell und agil auf veränderte Anforderungen zu reagieren. Im Sinne der "Mass Customization" lassen sich so die Skaleneffekte einer standardisierten Softwarebereitstellung nutzen, aber trotzdem individuelle Lösungen anbieten [BEHb07, S. 5]. Dabei würde sich die CRM-Anwendung im Java Umfeld als eine Sammlung J2EE-bezogener Artefakte<sup>8</sup> (Servlets, EJBs, Portlets, BPEL-Prozesse) in Enterprise Application Archives (Web Application Archive für Servlets und Java Archive für Enterprise Java Beans) betrachten lassen, die auf mindestens einem J2EE-Container, einer BPEL-Prozess-Engine und einem Portal-Server ausgeführt werden [Bre07, S. 84 f.]. Diese Laufzeitumgebung wird aus einer Kombination von Infrastrukturkomponenten ausgeführt, die wiederum aus Server-, Speicher-, Netzwerk- und entsprechenden Softwarekomponenten besteht. Diese Betriebsmittel, bzw. IT-Utilities, nehmen aufgrund unterschiedlicher Lastspitzen Einfluss auf die nicht funktionalen Eigenschaften der bereitgestellten Anwendung. Würde beispielsweise die CRM-Software auf einem zu stark belasteten Netzwerk laufen, wird die - in den

---

<sup>8</sup>Diese IT-Artefakte sind Servicebausteine, die durch Orchestrierungsmaßnahmen über ein SOAP-Interface - via HTTP - mit dem Webserver kommunizieren.

standardisierten SLAs - vereinbarte Dienstgüte nicht mehr eingehalten werden können. Würde man andererseits dem CRM-System so viele IT-Utilities bereitstellen, so dass diese jeder Last - unter Einhaltung der SLA - standhalten könnte, dann wären wahrscheinlich viele Ressourcen unausgelastet. Ein dynamischer Ressourcenzuteilungsmechanismus könnte daher auf die unterschiedlichen Lastspitzen, im Sinne einer serviceorientierten Ausrichtung, Einfluss nehmen.

Für die dynamische Verwendung von Ressourcen müssen die IT-Utilities über eine standardisierte Schnittstelle, als zustandsbehaftete Utility Services, entkoppelt werden und in einem Ressourcenrepository ähnlich einer UDDI<sup>9</sup> (Universal Description Discovery and Integration), auffindbar sein. Hierfür wurden die ersten Standards hinsichtlich Web Service Distributed Management (WSDM) von der Organization for the Advancement of Structured Information Standards (OASIS<sup>10</sup>) spezifiziert. Die Zielstellung des WSDM-Ansatzes besteht darin, IT-Utilities auf der Grundlage entsprechender XML-Spezifikationen zu managen und unter Verwendung der Webservice-Technologie den für das Servicemanagement benötigten Datenaustausch auf eine gemeinsame Grundlage zu stellen [Sch07b, S. 82 f.]. Genau diese Standardisierung der Schnittstellen ermöglicht die Orchestrierung der IT-Utilities zu größeren Einheiten. Dabei erfolgt die Zustandsbeschreibung während der Laufzeit in XML und wird durch eine Quality of Services (QoS)-Komponente innerhalb des Workloadmanagements überwacht. Die QoS-Komponente wird durch die Web Service Level Agreements (WSLA<sup>11</sup>) entsprechend konfiguriert. WSLA bilden dabei die definierten Service Level Objectives der IT-Utilities, unter Zuhilfenahme eines XML Schemas in einem XML-Dokument ab und ermöglichen die Formalisierung der SLA-Regeln mittels Boolescher Logik [Pas04, S. 9]. Dadurch werden die XML-basierten Zustandsbeschreibungen der eingesetzten Utilities mit mehreren logischen Ausdrücke über die Booleschen Operatoren mit der WSLA verknüpft und innerhalb des Workloadmanagements berechnet. Dabei ist das von der Multi-Tenancy Laufzeitumgebung verwaltete Serviceportfolio über Service Level Requirements (beispielsweise Verfügbarkeit: 99,99 % und Datendurchsatz: 0,148 Sekunden) definiert, das während der Laufzeit durch das Workload-Management (SLA-Manager), gemäß den definierten Zielen, gesteuert wird.

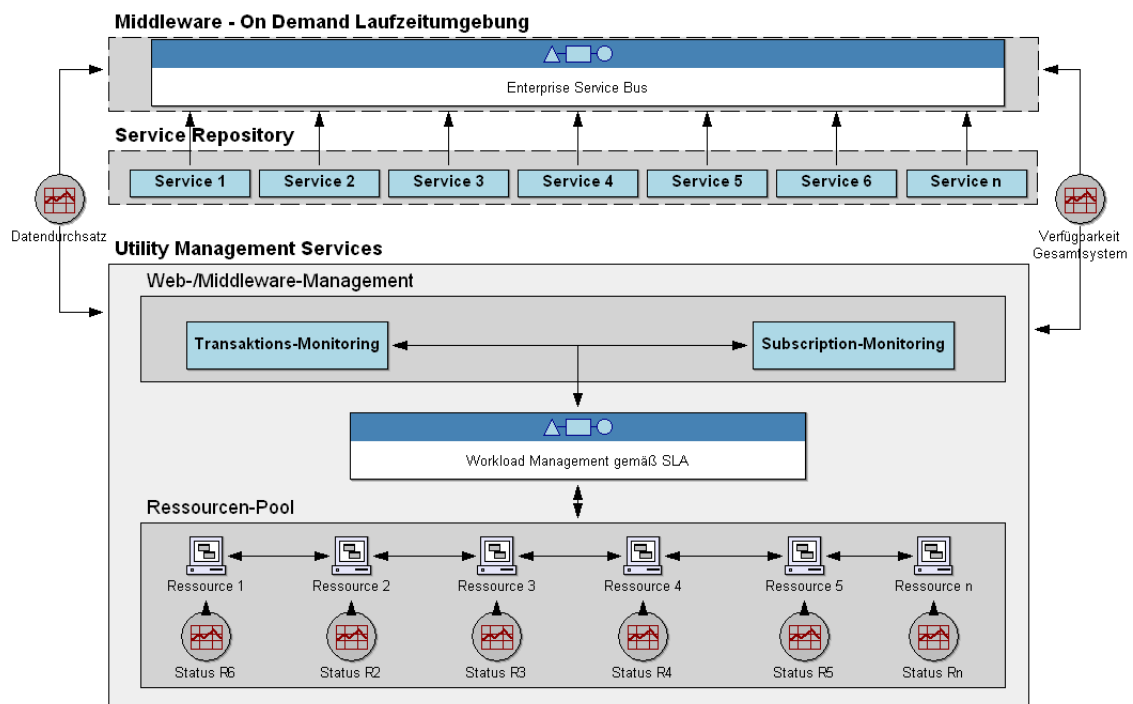
Nach einer erfolgreichen Authentifizierung am On Demand System wird eine automatische Instanzierung der zentralen CRM Anwendung, durch Bereitstellung und Konfiguration aller Ressourcen (wie Anwendungs- und Datenbankserver, Speicher, Netzwerk und Metadaten) ausgelöst. Zeitgleich erfolgt durch

---

<sup>9</sup>Die UDDI dient als logisch zentralisiertes aber physisch dezentralisiertes Repository von Webservices, in dem die vorhandenen Webservices registriert und katalogisiert werden.

<sup>10</sup>[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsdm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsdm)

<sup>11</sup>WSLA ist ein von IBM definiertes Service Level Management-Framework, basierend auf einer XML-Syntax zur Beschreibung von SLAs.



**Abbildung 3.1.:** Monetäre Aspekte und Monitoring von Utility Management Services

die Überwachung der Laufzeitumgebung die transaktionsorientierte Bemessung der tatsächlich verwendeten Ressourcen aus dem Ressourcenpool.

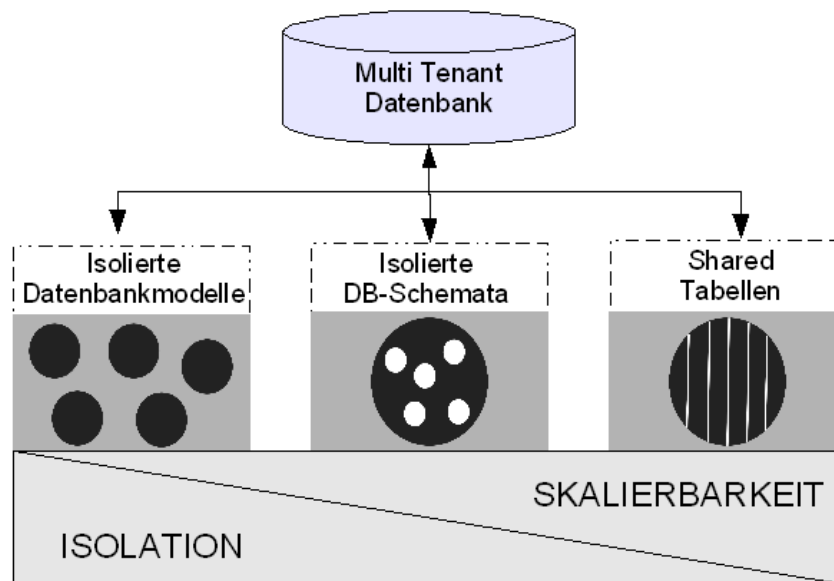
Das SOC-Konzept liefert folglich ein Framework für die Anpassung der IT-Infrastruktur und Applikationen an die adressierten Anforderungen eines Unternehmens, was eine wichtige Grundlage für das Liefermodell Software-as-a-Service hinsichtlich Leistung und Qualität bildet (vgl. [McK07];[Hou07]).

#### 3.3.1. Multi Tenant Data (MTD) - Architektur

Die soeben beschriebene Bereitstellungsmethode durch On Demand Computing beherbergt das Management für mehrere organisatorisch abgeschlossene Einheiten auf einer mandantenübergreifenden Anwendungsumgebung. Die Architektur, die sich dabei hinter einer mandantenfähigen Anwendung verbirgt, wird durch Process-, Business-, Security- und Metadata-Services charakterisiert [CC06]. Die Process-Services definieren, mit der Schnittstelle zur Präsentationsschicht, die Prozessfunktionalitäten und bedienen sich für die Aufgabenerledigung an den Business-Services, die den Zugriff auf die mandantenabhängigen Datenquellen durchführen. Hierfür sind die Security-Services für die Rechteverwaltung und Zugriffskontrollen verantwortlich. Bei einem zentralen Authentifizierungssystem werden von den Security-Services die Nutzer



innerhalb eines Mandanten logisch zusammengefasst. Dabei wird für die Autorisierung ein Rollenkonzept verwendet, durch das ein befugter Benutzer über eine Rolle mit mandantenabhängigen und mandatenübergreifenden Daten und Objekten verknüpft wird [Sto06, S. 95]. In der Regel sind hierfür die Rollen in einer Hierarchie angeordnet. Dadurch erben alle Rollen implizit dieselben Rechte für alle darunterliegenden Rollen. Je nachdem welchen Rollen ein Benutzer zugeordnet ist, werden ihm die entsprechenden Berechtigungen erteilt bzw. verweigert.



**Abbildung 3.2.:** Verschiedene Möglichkeiten von Multi Tenant Datenbanken [JA07]

Die zugrundeliegende Datenbankinfrastruktur kann dabei auf drei mögliche Implementierungen einer MTD-Datenbank basieren:

#### **Isoliertes Datenbankmodell:**

Bei dieser Form der Multi Tenant Architektur wird für jeden einzelnen Mandanten eine separate Datenbank angelegt, in der mandantenabhängige Daten und Konfigurationen gespeichert werden. Bei dieser mandantenorientierten Datenhaltung wird eine isoliert abgeschlossene Einheit gewährleistet, dessen Datenpräsentation und Konfigurationen individuell regelbar ist, da das Datenmodell beliebig erweitert werden kann. Jedoch bringt diese Datenisolierung einen Nachteile bzgl. Robustheit und Verfügbarkeit mit sich, da nur eine geringe Anzahl von Mandantendatenbanken auf einem Datenbankserver unterstützt werden können [CC06]. Als ein weiterer Nachteil lässt sich der hohe administrative Aufwand identifizieren, da bei Wartungs- bzw. Weiterentwicklungsarbeiten die Datenmodelländerungen in mehreren logischen Datenbanken durchgeführt

werden müssen. Dieses Datenbankmodell ist eher eine Option für Kunden, die bereit sind Extrakosten für Hardware und Wartung zu zahlen um eine hohe Sicherheit und ein anpassbares System zu erhalten. Daher wird solch ein isoliertes Datenbankmodell nur dann angeboten, wenn von Seiten des Mandanten eine individuelle Datenhaltung aufgrund der Corporate Governance Richtlinien notwendig ist.

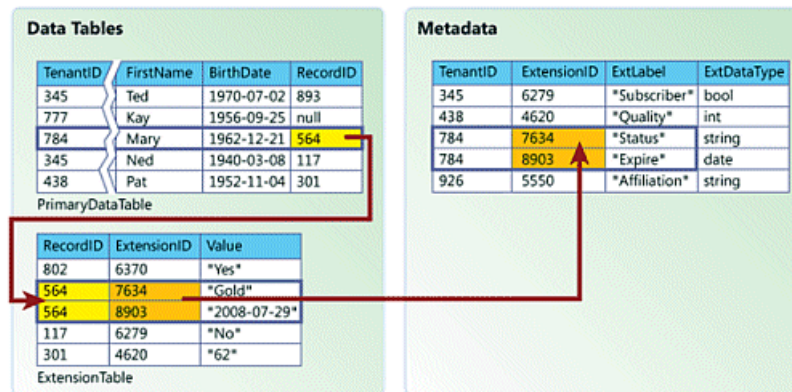
#### **Isoliertes Datenbankschemata:**

Ein anderer Ansatz wäre die Auslegung einer logischen Datenbank für die mandantenübergreifenden Daten und Objekte, in der jeder Mandant ein eindeutig identifizierbares Set an Tabellen zugeordnet hat. Die über Parameter gesteuerte und definierte Vorgehensweise ist einfach zu implementieren und führt in Summe, auch zu niedrigeren Hardwarekosten. Dieser Ansatz eignet sich einerseits besonders gut, wenn auf Anbieterseite nicht genügend Hardwareressourcen zur Verfügung stehen und andererseits, wenn die Kunden wenig Konfigurationsbedarf (wie z.B. beschränktes Customizing auf die Umbenennung von Feldern) haben. Der Kunde findet innerhalb der Softwareinstanz nur ein fixes Set an variablen Spalten für die benutzerdefinierte Konfiguration vor. Zu beachten ist auch, dass durch fehlerhafte oder manipulierte Datenbankzugriffssteuerungen keine zuverlässige Datenisolation mehr garantiert werden kann. Es müssen daher beachtliche Vorkehrungen hinsichtlich strenger Datentrennung getroffen werden, damit die Gefahr von fehlerhaften Zugriffen auf andere Mandanten minimiert wird. Zum Beispiel können über eine Trusted Database Connection oder eine Datenverschlüsselung entsprechende Vorkehrungen getroffen werden.

#### **Shared Tabellenmodell:**

Beim dritten Ansatz werden alle Mandanten über alle Tabellen in einer Datenbank zusammengefasst. Dieses Modell bietet die niedrigsten Hardware- und Entwicklungskosten, weil dadurch die Anzahl der Mandanten pro Datenbankserver auf ein Vielfaches maximiert werden kann. Dabei werden die Daten mit Mandanten IDs sowie Rollen-Suffixe zu einem Tenant View Filter verknüpft, um damit die Sichtbarkeit und die Attribute der mandantenübergreifenden Daten und Objekte zu steuern. Weil alle Daten der Mandanten in einem globalen Set an Tabellen zusammengefasst werden, sind die Standardobjekte in einer Objektdatenstruktur (z.B. Lead, Account, ...) durch Record IDs für alle Mandanten fest definiert. Bei der Konfiguration können die Standardobjekte um mandantenrelevante Datenfelder (Entitäten) ergänzt und von verschiedenen Entitäten genutzt werden. Dabei werden die Nutzdaten der neu definierten Entitäten in einer eigenen Extension Tabelle gespeichert und mit der entspre-

chenden Record ID in Relation gesetzt. Ein durch die Metadaten<sup>12</sup> durchgeführtes Mapping der Extension ID mit den Tenant IDs führt schlussendlich zur Definition des Datentyps und der Kennzeichnung der benutzerdefinierten Entitäten innerhalb eines mandantenübergreifenden Objekts (▷ vgl. Abbildung: 3.3).



**Abbildung 3.3.:** relationales Multi Tenant Datenbankmodell [C+06]

Diese Form der Erweiterung kann auch ohne zusätzliche Extension Tabelle erfolgen. Bei dem von Salesforce.com patentierten Ansatz "Custom Entities and Fields in a Multi-Tenant Database System" [CS05] erfolgt die Speicherung von benutzerdefinierten Entitäten innerhalb der Objektdatenstruktur durch vordefinierte und generische Spalten, die wiederum verschiedene Entitätstypen repräsentieren.

Diese MTD-Datenbank ist für die Softwarebereitstellung an eine unbegrenzte Anzahl von Kunden unentbehrlich, da Wartung und Weiterentwicklung die geringsten Probleme verursachen. Dennoch bedarf es entsprechenden Restore- bzw. Backupmechanismen auf der Ebene eines Mandanten, da die Daten nicht mehr so einfach wiederhergestellt werden können.

### 3.3.2. On Demand Computing und ITIL

Der Lebenszyklus einer digitalisierbaren Dienstleistung beschränkt sich ausschließlich auf den Zeitraum der Service-Erbringung und den gleichzeitigen Service-Verbrauch [HK06, S. 28 f.]. Der Zyklus beginnt mit dem Service-Abruf durch den Nutzer (z.B. beim Mausklick auf den webbasierten Login) und endet, nachdem die erbrachten Leistungen für die Bewältigung geschäftlicher Aufgaben genutzt und die gewünschten Effekte erzielt wurden (z.B. wenn nach einer erfolgreichen Kampagne in der bereitgestellten CRM-Anwendung ein Lead bearbeitet werden konnte). Nur innerhalb dieses Zeitabschnitts ist die Software

<sup>12</sup>Dabei erfolgt, durch einen Metadaten getriebenen Ansatz, die Konfiguration mit Hilfe von deklarativen Constraints (z.B. eindeutige Datenfelder).

für den Nutzenden existent, wirksam und nutzbar. Alle vorgehenden Service-Prozesse betreffen die Vorbereitung und den Betrieb der Hostingumgebung, wie die Disposition und Aktivierung aller benötigten Utilities sowie das Management der Bereitstellung. Erst durch den Abruf der On Demand Software wird, die individuelle "Service-Produktion" gestartet, der Service erbracht und dem jeweiligen Nutzer bereitgestellt [HK06, S. 28].

Um den Prozess der Softwarebereitstellung erfolgreich durchführen zu können, müssen sowohl Anfangs- als auch Endpunkt und alle Zwischenschritte bekannt sein [ZS07, S. 263]. Die von OCG (Office of Government Commerce) [o.V94] veröffentlichte Information Technology Infrastructure Library (ITIL) versteht sich als eine Richtlinie für die Gestaltung und das Management des Service-Supports und Service-Delivery-Prozesse, die vorrangig den Aktionsbereich der Anbieter abdecken [HK06, S. 6]. ITIL liefert hierfür ein Rahmenwerk zur Gestaltung eines effektiven Servicemanagements unter Berücksichtigung des Lebenszyklus der betroffenen IT-Services. Um ein effektives Servicemanagement tatsächlich realisieren zu können, müssen einige grundlegende Voraussetzungen erfüllt sein. Einerseits ist es wichtig, dass sich der Anbieter seines eigenen Leistungsprogramms bewusst ist und mögliche Potenziale, aber auch Lücken erkennt [ZS07, S. 260]. Der Anbieter muss, mit Blick auf die bedarfs- und nutzergerechte Bereitstellung der On Demand Software - durch die Analyse der Kundenanforderungen - die effiziente und rationelle Steuerung der Software und der zugrundeliegenden IT-Utilities planen, damit die Software durchgehend abrufbereit ist und durch eine optimale Auslastung die Erwirtschaftung hoher Erträge möglich wird. Andererseits muss der Softwareanbieter genaue Kenntnisse über die Bedürfnisse und Werttreiber seiner Kunden haben, um zu erkennen welche Services dem Kunden nutzen [ZS07, S. 260]. Besondere Bedeutung kommt dabei der Analyse des Nutzerverhaltens zu, da durch Verbesserungspotenziale ein passender, individueller Service für die heterogenen Nutzergruppen angeboten werden kann. Dabei können die Anbieter durch Produktdiversifikation (Aufnahme neuer Services mit horizontaler, vertikaler oder lateraler Beziehung zum existierenden Angebot) der hohen Individualität der Internetnutzer gerecht werden. Der durch geänderte Kundenbedürfnisse hervorgerufene Anpassungsdruck stellt sicher, dass ein Unternehmen sein Serviceportfolio stets neu aufstellt und die Anforderungen optimal erfüllt [ZS07, S. 262]. Nebenbei muss der Anbieter den dauerhaften Betrieb in einer Weise sicherstellen, die den Qualitätserwartungen der Kunden entspricht.

So werden zwingende Themen - wie SLA Management, Capacity Management, Availability Management, Financial Management und Continuity Management - adressiert, um die Erwartungen der Kunden zu steuern und die Bereitstellung in der definierten Qualität durch, technische und organisatorische Rahmenbedingungen, für eine erwartungskonforme Bereitstellung zu planen und zu erbringen [Sch07b, S. 79 f.]. Hierfür werden alle Aktivitäten auf die

Überwachung und Steuerung der erforderlichen Ressourcen ausgerichtet, damit eine erwartungskonforme Bereitstellung der Software erfolgt und der Kunde die Software in vollem Umfang für seine Zwecke nutzen kann. Alle Abläufe in der Prozesskette müssen die Bereitstellung in Echtzeit überwachen und steuern, so dass bei Bedarf verzuglos korrigierend eingegriffen werden kann [HK06, S. 31]. Die Prozesse des Service Delivery dienen also dazu sicherzustellen, dass die bereitgestellte Software von den registrierten Benutzern jederzeit und mit der vereinbarten Qualität benutzt werden kann.

- **Service Level Management** - beinhaltet die vertragliche Fixierung und Überwachung zugesicherter Leistungseigenschaften gegenüber dem Kunden.
- **Capacity Management** - dient zur zeitgerechten Bereitstellung durch situations- und lastabhängiger Nachführung von Ressourcen.
- **Availability Management** - dient zur Sicherstellung der durchgehenden Verfügbarkeit.
- **Financial Management** - ermittelt die Kosten der On Demand Nutzung und erstreckt sich bis zur Investitionsplanung für die erforderlichen IT-Ressourcen.
- **Continuity Management** - beinhaltet Methoden zur fortwährenden Analyse des Qualitäts- und Leistungsverhaltens eines produktiven Betriebes um proaktiv reagieren zu können.

Weiters unterliegen webbasierte Lösungen, im Gegensatz zu konventioneller Software, keinen statischen und langen Releasezyklen mehr, da diese inkrementell und durch enge Zusammenarbeit mit den Nutzern durchgeführt werden [HSSJ07, S. 79 f.]. Anbieter und Nutzer profitieren dabei gleichzeitig von den verkürzten Releasezyklen. Daher ist das Ziel eines jeden Anbieters, dass sich die Nutzer mit den Services identifizieren und sich eine Community um die angebotene Software bildet. Als Beispiel können folgende Community-Plattformen, im Kontext CRM On Demand aufgeführt werden:

- **Salesforce.com:** <http://community.salesforce.com/>
- **Netsuite:** <https://usergroup.netsuite.com/users/index.php>
- **Oracle CRM On Demand:** <http://www.oracle.com/siebel/forums.html>
- **RightNow Technologies:** <http://community.rightnow.com/customer/>
- **Salesnet:** <http://www.salesnetdeveloper.com/>

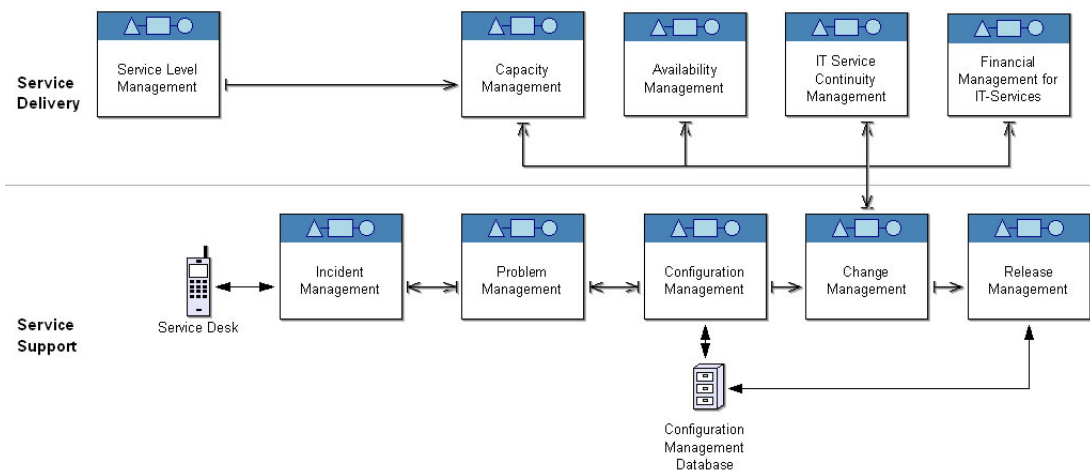
- **SAP Community Network:** <https://www.sdn.sap.com/irj/sdn>

Durch diese Form der Kundenbindung werden die Nutzer, in Anlehnung an die Open-Source-Bewegung, explizit zum Bestandteil des Entwicklungsprozesses, da sie durch direktes Feedback die Serviceverbesserung aktiv mitgestalten können. Mit dieser veränderten Rolle der Kunden wird jedoch die Qualitätssicherung nicht ersetzt, sondern nur die Funktionalität der Services validiert und präzisiert. So müssen die kontinuierlichen Updates und Upgrades der Software-Services nach einem disziplinierten Build-, Deployment- und Supportprozess erfolgen. Für die Gewährleistung eines reibungslosen Betriebs unter Berücksichtigung spezifischer Rahmenbedingungen, ergeben sich daher als operative Aufgabenstellungen folgende Aspekte, die im Rahmen des Service Supports in fünf verschiedene Prozesse unterteilt werden [Sch07b, S. 78 ff.]:

- **Incident Management** - behandelt über den Service-Desk, den Weblog oder das Forum die Schnittstelle zwischen Anwender und Anbieter und dient so als Kommunikationsplattform für Anfragen, Verbesserungsvorschläge, Ankündigungen und Schulungen.
- **Problem Management** - behandelt die Identifizierung grundsätzlicher Fehlerursachen mit Hilfe einer systematischen Analyse der Problemstellung, um eine geeignete Problemlösung zu spezifizieren, die durch eine adäquate Änderung der Prozesse bzw. der IT-Utilities umgesetzt wird.
- **Configuration Management** - geplante und ungeplante Änderungen bedürfen der exakten Kenntnis aktueller Bestandteile der Bereitstellungsinfrastruktur. Eine CMDB (Configuration Management Database) unterstützt diese Aufgabe.
- **Change Management** - sichert die erforderlichen, konsistenten und kontrollierten Änderungen über alle Funktionsschichten hinweg, damit die Softwarebereitstellung weiterhin gewährleistet wird.
- **Release Management** - beinhaltet die Qualitätssicherung, die Dokumentation des Änderungsprozesses und entsprechende Restorevarianten. Grundlage des Release Managements ist wiederum die CMDB, in der die Informationen über die durchgeführte Änderung gespeichert werden.

Folgende Abbildung soll einen Überblick über den ITIL-basierten Zyklus des Service Supports und Service Delivery inkl. der Funktion des Service Desks geben, der die Prozesse miteinander verbindet.

Diese Prozesse sollen die Risiken reduzieren und die Reaktionsfähigkeit bzw. -sensibilität bei einer garantierten und qualitativ gleichbleibenden Bereitstellung ermöglichen, mit dem Ziel eine enge Beziehung zu den Kunden aufzubauen.



**Abbildung 3.4.:** Service Delivery und Service Support nach ITIL

ITIL umfasst daher die Orientierung an einem allgemein anerkannten Standard zur Gestaltung der Prozesse [Köh07b, S. 34]. Hierbei werden innerhalb des Service-Support-Bereichs die Service-Prozesse behandelt, die bei Abruf eine geschäftliche Wertschöpfung beim Kunden erzielen. Dies betrifft auch komplementäre Dienste, wie Support durch einen IT-Helpdesk. Dabei wird die traditionelle Sicht, IT-Utilities und Anwendungen als mehr oder weniger separate Einheiten zu betrachten, durch eine ganzheitliche Betrachtung der Utility- und Anwendungs-Services, im Rahmen des Service Delivery abgelöst. Daher sind die Bereiche Service Support und Service Delivery eng miteinander gekoppelt und beinhalten Best Practices, welche für die Beschreibung und den Betrieb der On Demand Plattform wichtig sind.

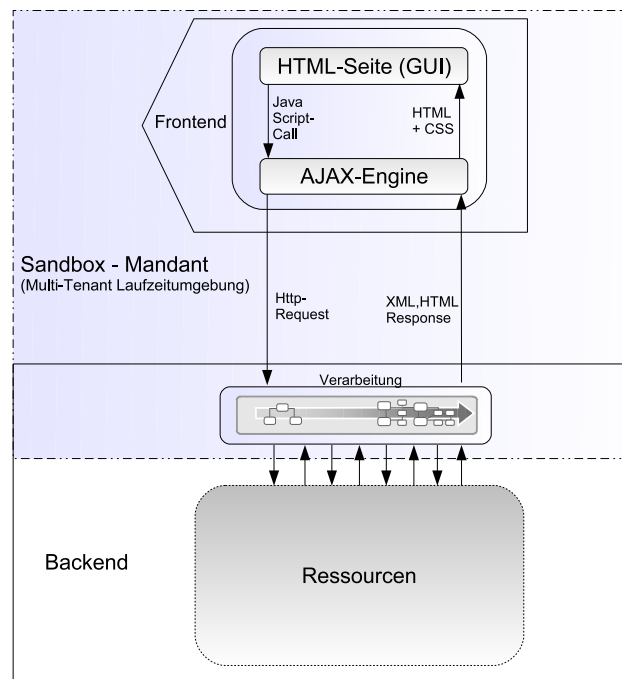
### 3.3.3. Rich Internet Applications

Ein entscheidendes Merkmal für die Etablierung von internetbasierten Geschäftsmodellen sind die Rich Internet Applications (RIAs), welche durch asynchrone Technologien wie die XMLHttpRequest-API (XHR), einen optimierten Datenfluss zwischen Client und Server garantieren und sich durch visuell und funktionell überzeugende Benutzeroberflächen auszeichnen. Insbesondere die Ausprägung des Service Clients aus kompilierten Bytecodes (Flex) oder aus zu interpretierenden Skripten (XHTML/DOM, CSS und JavaScript) ermöglicht GUI-Komponenten mit Rich-Client Funktionalitäten, welche einen hohen Bedienkomfort bei SaaS-Interaktionsvarianten bieten [SK08, S. 1]. Im Zusammenspiel dieser Techniken entsteht ein neues Nutzungsgefühl, welches oftmals als „reichhaltig“ umschrieben wird (d.h., dass eine Vielzahl von Funktionen bei einer minimalen Antwortzeit geboten wird) und sich von der hierarchischen Hypertextnavigation vergangener Jahre unterscheidet [Lan07, S. 14]. Der Begriff „Rich“ steht dabei für leistungsfähige Anwendungen die nicht installiert wer-



den, sondern über traditionelle Internetzugriffsmodelle den selben Umfang an Funktionalitäten wie desktopähnliche Anwendungen mit Rich-Clients liefern [NH05, S. 2 f.].

Dabei werden ausschließlich interaktive, mediengestützte Internettechnologien - entsprechend definierter Security Levels nach dem Sandbox-Prinzip - ausgeführt. Das bedeutet, dass das XMLHttpRequest-Objekt nur vom Server aus Zugriffe auf die in den Metadaten definierten Ressourcen der jeweiligen Mandanten (Domain-Sandbox) machen kann und die mandantenspezifische sowie virtualisierte Softwareinstanz über ein Webservice-Backend in einer isolierten Laufzeitumgebung ausgeführt wird. Somit kann die Benutzerschnittstelle der RIA, durch abgesicherte, asynchrone Ladevorgänge die Abarbeitung von benutzerdefinierten Prozessschritten, sowie die Ausführung von komplexen Businessfunktionen, ohne Page Reloads ermöglichen. Deshalb erfordert die internetbasierte Bereitstellung geschäftskritischer Anwendungen eine flexible und komponentenbasierte Architektur, die eine klare Trennung zwischen den fluiden Services und dem stabileren Backend-System beinhaltet [HSSSJ07, S. 87].



**Abbildung 3.5.:** asynchrones Interaktionsmuster bei SaaS [Bos07, S. 39 ff.]

Die Technologien für die Realisierung von RIAs lassen sich in zwei Kategorien einteilen. Möglichkeiten für die Realisierung von RIA wären die Implementierung einer clientseitigen JavaScript-Engine, die keine weiteren Plug-Ins im



Browser benötigt, oder der Einsatz von Techniken, die eine Installation von Plug-Ins voraussetzt. Die derzeit bekanntesten Technologien für Rich Internet Applications sind:

- AJAX (Asynchronus JavaScript and XML; keine weiteren Voraussetzungen)
- Adobe Flex (setzt Flash Runtime Engine voraus)

## Asynchronous JavaScript And XML

Das Aufbrechen des starren Interaktionsmusters Request/Response zwischen Webbrowser und Webserver, wird durch Asynchronous JavaScript And XML (AJAX) als ein Sammelbegriff von mehreren Standards zusammengefasst. Die AJAX Frameworks unterstützen die Cross-Plattform und Cross-Browserfunktionalitäten durch eine plattform und browserneutrale Abstraktionsschicht auf Basis von XHTML/DOM, CSS und JavaScript. Die Frameworks lassen sich grundsätzlich in clientseitige und serverseitige Frameworks unterteilen, wobei die clientseitigen und JavaScript basierenden Bibliotheken per XMLHttpRequest mit den serverseitigen Services kommunizieren. Über dieses Objekt ist es der AJAX Engine möglich XML Daten zum Webserver zu senden und zu empfangen und damit dynamische HTML Seiten aufzubauen. Es handelt sich daher um daher ein Verwendungsmuster von Webtechnologien, das durch asynchrone Serverkommunikation geprägt ist und sich durch folgende Aspekte auszeichnet [GGA06, S. 5 f.]:

- eine standardisierte Präsentation, die XHTML und CSS nutzt;
- eine dynamische Anzeige und Interaktion unter Nutzung des Document Object Modells des Browsers;
- der Datenaustausch und die Datenmanipulation erfolgen durch XML und XSLT;
- eine asynchrone Datenabfrage durch Nutzung des browserintegrierten XMLHttpRequest-Objekts; und
- die Verwendung von JavaScript als Bindeglied.

Die Aspekte von AJAX ergeben dabei neue Möglichkeiten [GGA06, S. 6 f.]:

- **Kleine serverseitige Ereignisse:** Kleine Teilanfragen an den Server erlauben eine Anpassung des DOM ohne einen vollständigen Reload auszulösen.
- **Asynchronität:** An den Server gerichtete Anfragen blockieren den Browser nicht und der Nutzer kann weiterhin agieren.

- **onAnything:** Anfragen an den Server können durch fast alle Aktionen wie Mausaktionen und Tastenanschläge ausgelöst werden. Durch jedes dieser vom Nutzer ausgelösten Ereignisse kann eine asynchrone Anfrage an den Server geschickt werden.

Zusammengefasst, liefern die AJAX-Technologien eine kontextspezifische und bedienfertige Benutzerschnittstelle auf Basis von XHTML, CSS und JavaScript [Bau07, S. 13 f.].

### Adobe Flex

Adobe Flex stellt eine Entwicklungsumgebung von designtechnisch komplexen RIAs dar, die insbesondere wegen der Komponenten für die Visualisierung von Data-Charts, Data-Grids, Cubes und Report Views - auf Basis von HTML, CSS, JavaScript, AJAX und Flash - von besonderer Bedeutung sein können (▷ vgl. <http://examples.adobe.com/flex2/inproduct/sdk/dashboard/dashboard.html>). Die wichtigsten Bestandteile von Flex sind die deklarative Sprache MXML sowie ActionScript. Dabei wird der Rich-Client auf Basis von XML und Actionscript kompiliert, welcher - ähnlich wie bei Java Applets - in einer Sandbox (Flashplayer ab Version 9) ausgeführt wird. Die größten Vorteile von flashbasierten Rich-Clients ergeben sich aus [Bau07, S. 12 f.]:

- einer kompletten Unabhängigkeit der Darstellung in unterschiedlichen Systemumgebungen, sofern die Flash Runtime Environment in Form des Flash Players verfügbar ist.
- einem großen Spielraum bei der Verwendung von Designelementen, was sich darauf zurückführen lässt, dass Flash zu Beginn als Werkzeug zur Generierung von Animationen gedacht war.
- einem Sicherheitsvorteil, da Flash Applikationen nur in einer kompilierten \*.swf Datei präsentiert werden.

Darüber hinaus ermöglicht die betriebssystemübergreifende Laufzeitumgebung AIR (Adobe Integrated Runtime), die als SaaS zugängliche Software direkt aus dem Netz zu laden und außerhalb eines Browsers auszuführen.

## 3.4. SaaS - Software-as-a-Service

Software-as-a-Service beschreibt nun eine gänzlich via Internet betreibbare Software, welche vom Anbieter entsprechend vereinbarter Rahmenbedingungen als Service bereitgestellt wird. Dabei umfasst die Serviceorientierung

- die Identifikation von Kundengruppen, auf deren Basis die Leistungserstellung auf die Anforderungen ausgerichtet wird,

- die auf diese Kunden ausgerichtete Bereitstellung von Software als Dienstleistung
- sowie die Planung und Kontrolle der Dienstleistungsqualität [WBK07, S. 4 f.].

Die Software, welche On Demand auf den Servern des Softwareherstellers für die Nutzung durch servicerelevante Systemfunktionen verteilt wird, wird folgendermaßen definiert:

Die Software ist eine browserbasierte Anwendung, die einen nutzer- und wertschöpfungsorientierten Service liefert.

Dabei liefert der Service eine “ajaxifizierte Browserschnittstelle”, die per Verschlüsselung den Zugriff, die Übertragung und die Speicherung von Daten auf der laufend aktualisierbaren Anwendungsumgebung des Softwareanbieters, bei gleichzeitiger Überwachung und Steuerung der IT Utilities, gewährleistet. Für den Einsatz im geschäftlichen Umfeld werden die relevanten Leistungs-, Service- und Sicherheitslevels im Rahmen der vertraglichen Vereinbarung, garantiert. Zusätzlich wird das SaaS-Modell in der Regel durch integrierte Servicekomponenten, wie Self-Monitoring, für die proaktive Steuerung des Anwendungsmanagements erweitert [ZE08, S. 47]. Des weiteren kommt eine globale Serviceinfrastruktur, als Grundlage für mandantenspezifische Support-Services, zum Tragen. Zudem ist die Software im SaaS-Modell schnell einsetzbar, kann einfach erweitert werden und führt gleichzeitig zu einer Vereinheitlichung und Standardisierung, die wiederum eine rationelle und unternehmensübergreifende Zusammenarbeit sowie die Integration aller Akteure ermöglicht. Der Reifegrad reicht hierbei von standardisierten Software-Services ohne Anpassungsmöglichkeiten bis hin zu parametrisierbaren Serviceangeboten, wobei immer eine angemessene Balance zwischen einfacher Bedienbarkeit und Komplexität der Anforderungen festgestellt werden kann. Dies erlaubt es, die Lösung ohne aufwändige und kostspielige Investitionen zu evaluieren und zu nutzen [ZE08, S. 50]. Die Anpassung an Kundenanforderungen muss so einfach gestaltet sein, dass Anwender ein auf sie angepasstes Trialsystem nutzen können, um im weiteren Prozess - ohne Brüche - die notwendigen Anpassungen bis hin zu einem produktiv einsetzbaren System durchführen können [ZE08, S. 47]. Für die Sicherstellung einer einfachen Bedienung, lässt sich durch den integrierten Customizingrahmen eine weitgehend anpassbare Arbeitsumgebung für die mandantenspezifische Nutzung samt rollenorientierter Ausrichtung konfigurieren, um den Nutzern jeweils genau die Funktionen bereitzustellen, die in ihrem Aufgabenkontext relevant sind [ZE08, S. 48].

Für den Anbieter stellt zunächst die sicherere und standardisierte Bereitstellung der Software den Kern der Wertschöpfung dar, damit eine zentral zugängliche Software für eine große Anzahl von Anwendern eingesetzt werden kann

[Hal07, S. 78]). Dadurch ist, auf der Seite des Anbieters, das Anwendungsmanagement nur auf eine Systemumgebung auszurichten. Aus diesem Grunde sehen sich die SaaS-Anbieter gezwungen, den integrierten Customizingrahmen für die einzelnen Kundeninstanzen so auszurichten, dass dadurch die Update- und Upgradefähigkeiten nicht beeinflusst werden [Hel07]. Aus diesem Grund werden die gesamten Abläufe des Customizings stark vereinfacht, um die daraus resultierenden Komplexitäten - nicht nur im Rahmen der Einführung, sondern auch bei den notwendigen Anpassungen im Betrieb - für die Anwender, möglichst einfach zu halten [ZE08, S. 48]. Um aber auch hier einen größtmöglichen Kundennutzen und damit eine Wertschöpfung beim Kunden zu generieren, ist es naheliegend, die Software auf der Grundlage einer Multi-Tenant Architektur für die mandantenspezifische Nutzung bereitzustellen.

Ohne die Technologien des Service Oriented Computing ist das SaaS-Modell nur schwer realisierbar, da die Multi-Tenant Architektur, in einem betriebswirtschaftlich relevanten Maßstab, hohe Anforderung an die Leistungsfähigkeit der zugrunde liegenden Technologie stellt. Von Bedeutung ist hier vor allem die modifikationsfreie Erweiterung durch die Verwendung von offenen Standards, die letztlich den Konflikt zwischen mandantenspezifischen Anpassungen und Standardentwicklungen löst [ZE08, S. 49]. Dabei etabliert die Serviceorientierung [VSTT06] eine ortsunabhängige und den Anforderungen abgestimmte Nutzung, welche anhand eines Metadaten-getriebenen-Ansatzes viele einzelne mandantenspezifische Schemata in einem physischen Schema vereinigt und wohldefinierte Funktionalitäten für zugesicherte Qualitätskriterien zur On Demand Nutzung anbietet. Das Zusammenspiel von Webservicetechnologien und Multi-Tenant Architekturen ermöglicht es, die Benutzeroberfläche von der darunterliegenden Applikationsplattform zu entkoppeln und die individuellen Softwareinstanzen als mandantenspezifische Benutzeroberflächen (Workspaces) darzustellen. Ebenso lassen sich, entsprechend der unterschiedlichen Kommunikationskanäle und -geräte Benutzeroberflächen für mobile Szenarien (z.B. PDAs) bereitstellen [ZE08, S. 50].

Das Zusammenspiel von serviceorientierten Technologien auf einer gemeinsamen Infrastruktur und einer Codebasis für die Multi-Tenant Bereitstellung liefert nun eine gut durchdachte Systemplattform zur Individualisierung und Optionierung von SaaS. Dadurch erzielt das Software-as-a-Service-Paradigma immer mehr an Akzeptanz am Markt. Vor allem das enorme Wachstum des Vorreiters Salesforce.com spricht Bände und nicht zuletzt SAPs Launch des SaaS-Angebot Business by Design hat viele Kritiker davon überzeugt, dass diese Art und Weise der Softwarenutzung, über ein großes Zukunftspotential verfügt [vG08, S. 36]. Gemessen an den Userzahlen, entwickelt Salesforce.com darüber hinaus eine hohe Anziehungskraft für alle möglichen Softwarefirmen (Independent Software Vendors), die z.B. über die Salesforce-API Anwendungen entwickeln, welche sich vollständig innerhalb der Salesforce-Arbeitsumgebung

bedienen lassen und vom Nutzer erwerbbar sind. Durch den Netzwerkeffekt ist die Attraktivität der Plattform Salesforce.com so hoch, dass sich die Multi-Tenant Plattform von Salesforce als Entwicklungs-, Vertriebs- und Betriebsumgebung für Applikationen von Dritten immer stärker etabliert. Über eine Art Marktplatz für Salesforce-Anwendungen (mit dem Namen Appexchange) werden so bereits über 700 Anwendungen von Drittfirmen bereitgestellt, die die Funktionalitäten von Salesforce nutzen und erweitern [vG08, S. 37]. So wundert es nicht, dass zunehmend auch Unternehmensanwendungen, die mit der eigentlichen Idee eines netzbasierten CRM-System nichts mehr viel zu tun haben, den Weg als Service auf diese Plattform finden.

### 3.4.1. Erfolgsfaktoren für SaaS

Sind es die monetären Aspekte einer SaaS-Lösung die durch den Wegfall kostenintensiver Lizenzen sowie Folgekosten für Wartung und Pflege, die Liquidität eines Unternehmens schonen und eine erhöhte Kostentransparenz im Rahmen des IT-Controllings liefern? Sind es die Aspekte hinsichtlich einer konsolidierten Managementschicht einer Flexibilität im Rahmen der Internationalisierung oder bei der Vernetzung neuer Standorte? Oder wird einfach ein Dienstleister benötigt der aufgrund steigender Anforderungen die Betreuung einer leistungsfähigen IT-Infrastruktur samt Software übernimmt und nebenbei zusätzliche Serviceleistung erbringt? Jedenfalls erlaubt SaaS - als eine weitere Facette des Sourcing von IT-Services - die Möglichkeiten zur Bündelung von technischer Dienstleistungen und Prozessdienstleistungen besser hinzubekommen und damit wesentlicher auf die Kundenbedürfnisse einzugehen. Die zugrundeliegenden Virtualisierungstechniken auf Basis modularer Servicearchitekturen ebnen immer mehr die Voraussetzungen für Plattformen, auf denen sich standardisierte Funktionsbausteine zu individualisierten IT-Services zusammenfügen lassen [BEHb07, S. 5]. Die damit verknüpften Technologien setzen ganz auf die mit den industriellen Massenfertigungen verknüpften Kennzeichen, nämlich Spezialisierung, Modularisierung und Zusammenarbeit. Vor allem die Entkopplung der Ressourcen-Infrastruktur, die durch ein besonderes Maß an Kontrolle und Koordination - in Anlehnung an das prozessorientierte IT-Servicemanagement Framework ITIL - gekennzeichnet ist, ermöglicht eine große Anzahl mandantenspezifischer Softwareinstanzen, basierend einer zentralen Codebasis, zu erzeugen. Dadurch werden wiederverwendbare Elemente zielgerichtet definiert und standardisiert eingesetzt. Dies erhöht die Konsistenz und Qualität der Servicelieferung, da der Einsatz von wiederverwendbaren Elementen zu vorhersagbaren Kosten und Resultaten führt [ZS07, S. 263]. Gleichzeitig kommt es mit der Standardisierung und Automatisierung der Prozesse bei der Erstellung von Standardsoftwareprodukten zu einer Dekomposition der Wertschöpfungskette [WBK07, S. 3]. Dadurch werden immer mehr - als webbasierter Service - angebotene Standardsoftwareprodukte, Produkt- sowie Prozessinnovationen ermöglichen. Diese Produkt- und Prozessinnovationen

basieren auf folgenden Überlegungen:

- Integration dezentral strukturierter Organisationseinheiten bei gleichzeitig geringen Anlaufinvestitionen.
- Homogenisierung und agile Stützung der Prozesse in einer verteilten Umgebung ohne zusätzliche up-front Kosten für Hardware und Software.
- Zeitgleiche Aufwandsverringerung sowie Reduzierung der Kosten und Komplexität des Infrastrukturmanagement im Rahmen der Anbindung von Organisationseinheiten mit eingeschränkten Infrastrukturen.
- Daraus resultierende Freisetzung von Kapazitäten zur Stärkung der Kernkompetenzen.
- Sofortige Inbetriebnahme und Steigerung der Effizienz.
- Reduzierung der Projektkosten mit gleichzeitig schnellem Return on Investment (ROI).
- Steigerung der Kostentransparenz basierend auf linearen Nutzungsgebühren und folglich eine bessere Prognostizierbarkeit sowie Planbarkeit der benötigten Kosten.

Jedenfalls gilt die (nahezu) risikolose Möglichkeit neue Softwarefunktionalitäten zu verhältnismäßig geringen Initialkosten im Unternehmensumfeld einzusetzen als einer der Erfolgsfaktoren dieser Nutzungsform. Damit wird eine hoch entwickelte und im klassischen Lizenzmodell oft sehr kostenintensive Software durch eine kalkulierbare Mietgebühr stets aktualisiert und in der neuesten Version bereitgestellt [Tam03, S. 48]. Da die Möglichkeit besteht die Software vor der Einführung im Rahmen einer kleinen Anwendergruppe, zu testen, können vor dem Vertragsabschluss die technischen Rahmenbedingungen sowie der Customizingrahmen mit den eigenen Anforderungen abgestimmt werden. Auf diesem Weg entsteht eine Vertrautheit mit den Umgangsformen der bereitgestellten Software, so dass der Rollout einfach zu realisieren ist. Die als Service abonnierte Software kann nach einem erfolgreichen Testlauf, in kürzester Zeit in den produktiven Betrieb aufgenommen werden und das Unternehmen so schneller am Markt agieren.

SaaS spielt vor allem dann seine besondere Stärke aus, wenn im Bereich Collaboration (die über Abteilungs- und Unternehmensgrenzen hinweg durchgeführte Zusammenarbeit), temporäre oder dezentral strukturierte Organisationseinheiten ein zentrales System für einen fach- und standortübergreifenden Wissensaustausch benötigen. Deshalb korreliert SaaS mit Trends, wie der ansteigenden unternehmerischen Vernetzung und mit der Virtualisierung der Leistungserstellung, die wiederum von der ökonomischen Prämisse der Konzentration auf betriebliche Kernkompetenzen angetrieben wird [BMW07, S. 34].



### 3.4.2. Mögliche Problemfelder

Obwohl heute die Erfüllung sicherheitsrelevanter Anforderungen im Rahmen des SaaS-Geschäftsmodells oberste Priorität genießt, ist die Skepsis im Zusammenhang mit der Übertragung von geschäftsrelevanten Daten an einen Anbieter nach wie vor unverändert. Die Hindernisse resultieren insbesondere aus der erschwerten Kommunikation des tatsächlichen Mehrwerts. Ferner ist als kritisch anzusehen, dass, bedingt durch das Scheitern von ASP, viele Interessenten die SaaS-Nutzung bereits im Vorfeld mit einer gewissen Skepsis betrachten [Hal07, S. 81]. Als ein signifikanter Grund werden hierfür die fehlenden Sicherheitsmechanismen und die Schwierigkeiten ehemaliger ASP-Anbieter herangezogen. Aus diesem Grund ist es für die Anbieter von entscheidender Bedeutung, den Kunden ihre Argumente glaubwürdig zu vermitteln. Dabei können beispielsweise Zertifizierungen wie Trust Services (▷ vgl. Abschnitt: 2.3.2) dienen, die einen Überblick über die physische und umgebungsbezogenen Sicherheitsmaßnahmen auf Anwendungs-, Anlagen- und Netzwerkebene verschaffen oder die Vorlage einer SAS 70<sup>13</sup> Bescheinigung (Statements on Auditing Standards Nr. 70), die regelmäßige Kontrollen der Teilgebiete Systembetrieb, Datenmanagement und Informationssicherheit transparent darlegen. Da auch viele SaaS-Lösungen direkt von amerikanischen Softwareherstellern bereitgestellt werden, stellt die Gewährleistung der Rechtssicherheit (vgl. Tabelle: C.2) für die grenzüberschreitenden Nutzung, im Rahmen der Datenübermittlungen oder -verarbeitungen, eine weitere Hürde dar [BIT06c, S. 64]. Vor allem im Zeitalter weltweit verteilter Produktions- und Absatzketten werden regulatorische Maßnahmen immer mehr zur unabdingbaren Voraussetzung.

Als ein weiteres Problem für jede Art von Serviceerbringung im Netz wird im Regelfall der Individualisierungsgrad hinsichtlich der Funktionalitäten und Möglichkeiten des Prozessmanagements angesehen. Sehr zum Leidwesen von SaaS, führt diese Erkenntnis vor allem im großbetrieblichen Umfeld zu starker Skepsis, da eingeschränkte Möglichkeiten der Adaption, bzw. mögliche Schwierigkeiten bei der Integration, eine Restrukturierung bereits etablierter Prozesse herbeiführen würde. Andererseits kann eine gut durchdachte Systemplattform die nötige Grundlage für eine Individualisierung und Optionierung eines SaaS-Angebots liefern, wie am Beispiel von Salesforce.com in Erfahrung gebracht werden konnte. Es gibt auch Ansätze, bei denen Softwarehersteller wie z.B. Oracle im Rahmen des eigenen SaaS-Portfolios versuchen, mit vorkonfigurierten Lösungen, basierend auf branchenspezifischen Referenzmodellen, dem - vor allem im Mittelstand als kostenaufwändig wahrgenommenen Customizingprozess - entgegenzuwirken.

---

<sup>13</sup>Dieser Standard wurde vom AICPA (American Institute of Certified Public Accountants) im Jahre 1992 mit dem Ziel veröffentlicht, die internen Steuerungs- und Kontrollprozesse der Service Anbieter offenzulegen und zusätzlich eine Aussage über deren Qualität vorzunehmen [Mat05, S. 1].

Beim Thema Abhängigkeit sollten Kunden die Vorteile und die Risiken einer internetbasierten Softwarenutzung sorgfältig abwägen, denn wenn z.B. eine SAP-Lösung eingeführt wird und im Rahmen des Customizings auf individuelle Anforderungen parametrisiert wird, ist auch eine nicht unerhebliche Abhängigkeit von SAP gegeben [Lix08]. Der Unterschied zu einer SaaS-Anwendung besteht darin, dass der SAP-Nutzer die Kontrolle über die Daten behält. Daher sind die Antworten folgender Frage wichtig [Lix08]:

- Wer übernimmt die Verantwortung, wenn Daten z.B. durch Phishing-Attacken verloren gehen, beschädigt werden oder in fremde Hände geraten?
- In welcher Form werden die Daten zurückgeführt, wenn der Vertrag mit dem SaaS-Anbieter gekündigt wird?
- Können die überführten Daten danach problemlos auf den kundeneigenen Systemen oder auf denen von Dritten weiterverarbeitet werden?

Die weit verbreitete Angst, dass SaaS-Lösungen in puncto Datensicherheit riskant sind, ist also keineswegs unbegründet. Allerdings besteht bei kundeninternen Lösungen diese Gefahr ebenso, weil in der Regel schon die eigenen Mitarbeiter für Unsicherheit sorgen [Lix08]. Jedenfalls ist eine sorgfältige Auswahl des Anbieters samt der Software eine wesentliche Voraussetzung für einen erfolgreichen Einsatz von SaaS. Daher empfiehlt es sich, vor dem erwogenen Einsatz die Referenzen und Partnerschaften auf etwaige prestigeträchtige Hersteller-Kunde-Beziehungen zu begutachten und das gesamte Leistungsportfolio des Anbieters mit den eigenen Erwartungen abzustimmen [Rie05, S. 88].



# 4. Die Komponenten eines CRM-Systems

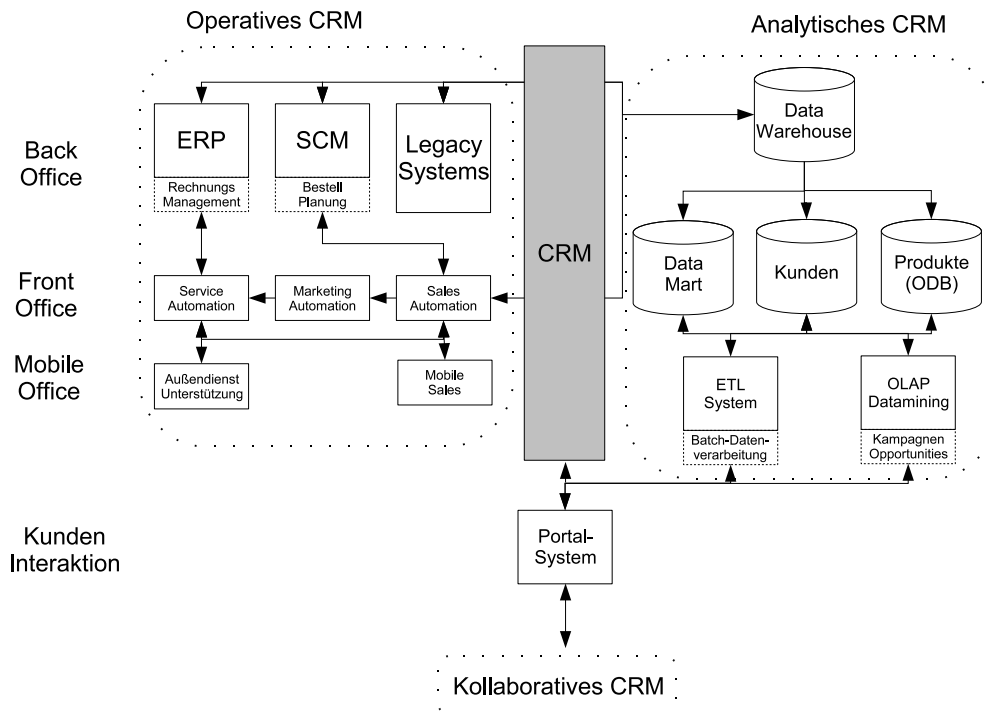
Da sich die spätere Analyse auf CRM On Demand bezieht, vermittelt dieses Kapitel zunächst einen Einstieg in das Thema Customer Relationship Management und behandelt die Grundlagen von CRM Systemen, worunter hier die organisatorische und technische Realisierung von CRM On Demand in Unternehmen verstanden werden soll.

## 4.1. Grundlagen CRM

Unter CRM wird die kundenorientierte Ausrichtung eines Unternehmens verstanden, wobei mittels zugrundeliegenden Informations- und Kommunikationstechnologien langfristige und profitable Kundenbeziehungen aufgebaut werden sollen [Mey02, S. 8]. Die relevante Zusammenführung aller Kundeninformationen sowie deren Integration in eine zentral zugängliche Datenbankanwendung ermöglicht eine strukturierte und gegebenenfalls automatisierte Erfassung sämtlicher Kundenkontakte und -daten. Die notwendige Zusammenführung aller Kundeninformationen sowie deren Analyse und Nutzung bedürfen daher der intensiven IT-Unterstützung folgender CRM-Komponenten [SJ04, S. 17 ff.]:

- Synchronisation und operative Unterstützung der Anwendungen aus Marketing, Vertrieb und Service (Operatives CRM);
- Einbindung aller Kommunikationskanäle zwischen Kunden, Abnehmer und Unternehmen (Kollaboratives CRM);
- sowie die Auswertung aller relevanten Informationen für die Wissensgewinnung und Wissensnutzung (Analytisches CRM).

In Anlehnung an Meyer wird unter einem CRM-System die Zusammenführung verschiedener Softwaretools verstanden, die im Rahmen des operativen, kollaborativen und analytischen CRM eingesetzt werden [MW02, S. 12]. Die einzelnen Komponenten samt deren Funktionalitäten sind in der ▷ Abbildung: 4.1 dargestellt. Als Komponenten eines integrierten CRM-Systems werden Daten, Funktionsmodule, Organisationen, Hardware und Kommunikationsinfrastruktur genannt [Mey02, S. 13]. Demzufolge beschränkt sich ein CRM-System nicht



**Abbildung 4.1.:** Funktionsübersicht CRM [Sch07a, S. 32]

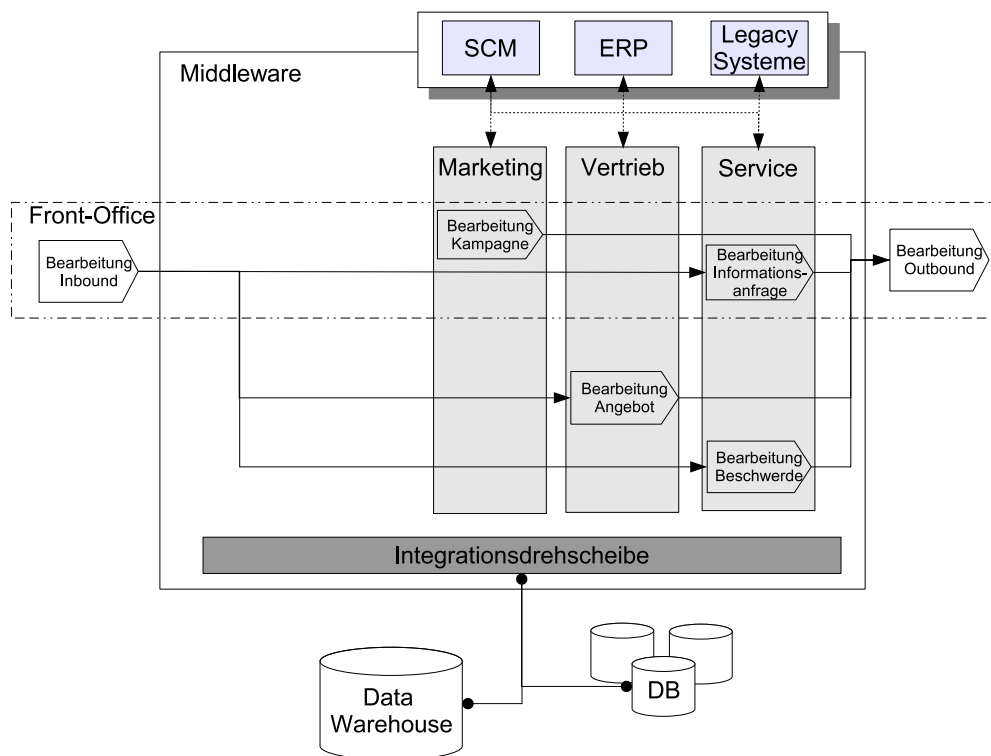
ausschließlich auf die eingesetzten Softwaretools. Ein Grund für diese Auffassung ist, dass CRM im Überschneidungsbereich verschiedener Unternehmensbereiche angesiedelt ist. Deshalb lassen sich verschiedene Ausprägungsformen von CRM kategorisieren, nämlich die Basisversion durch Unterstützung operativer CRM-Prozesse, die Nutzung des Informationspotentials der Kundenbeziehungen durch Ausbau der Basisversion zum analytischen CRM und die Integration der Kunden und Prozesspartner in das kollaborative CRM.

## 4.2. Operatives CRM

Das operative CRM unterstützt Anwendungen im Front Office und bildet damit den Kern jeder CRM-Systemumgebung. Die zentralen Kernkomponenten werden durch spezielle Module des Marketings, Vertriebs und Services gebildet und über eine Middleware mit dem Back Office gekoppelt. Die im Back-Office angesiedelten Unternehmensanwendungen (ERP, SCM oder Legacy Systeme) werden im Sinne einer ganzheitlichen Informationsinfrastruktur durch Enterprise Application Integration (EAI)-Mechanismen<sup>1</sup> integriert.

Diese Ebene wird als Kernstück einer jeden CRM-Integration betrachtet,

<sup>1</sup>EAI ist die unternehmensinterne Integration von Systemen.



**Abbildung 4.2.:** Architektur der operativen CRM-Prozesse [AW04, Sch05]

weil sich dadurch Daten und Funktionen über verschiedene Applikationen hinweg integrieren und ohne manuelle Eingriffe mehrfach verwenden lassen. Daher versteht sich EAI im Sinne von CRM als eine "Integrationsdrehzscheibe", die Prozesse des operativen CRM über mehrere Back-Office-Systeme automatisiert, um die Kernprozesse des operativen CRM bestmöglich zu unterstützen [Mar04].

Bevor explizit auf die operativen Funktionalitäten eingegangen wird, werden kurz die Basisfunktionalitäten, die den einzelnen Kernkomponenten des CRM-Systems zur Verfügung stehen, erläutert [AS02, S. 28 f.]:

- Als typische Basisfunktionen werden die Bereitstellung von Arbeitsoberfläche und Arbeitsumgebung sowie die Administrator-Funktionen, für die Verwaltung des CRM-Systems, gesehen.
- Das chronologische Aktivitätenmanagement soll die Anwender bei der Planung durchzuführender Aktivitäten (mit Hilfe von Kalenderfunktionen, Tätigkeitslisten und Priorisierungsverfahren) unterstützen.
- Die Mehrsprachigkeit und Internationalisierung des Systems wird durch die Verwendung unterschiedlicher Sprachen, Adress-, Titel- und Zahlenformate sowie Währungen und Zeitzonen gewährleistet.

- Korrespondenzunterstützung kann durch Formatvorlagen und einer Anbindung an externe Textverarbeitungsprogramme sowie Mailprogramme erfolgen.
- Funktionalitäten des Wissensmanagements - wie Suchfunktionen, Indizierungen, Selektionsverfahren, Kataloge, Baumstrukturen und Verzeichnisse - sollen den Umgang mit umfangreichen Informationsbeständen erleichtern.
- Workflowmanagement-Systeme sollen definierte Aktivitäten in automatisierter Form ausführen.

Aufbauend auf diese Basisfunktionalitäten eines CRM-Systems werden unterschiedliche Modultypen eingesetzt, die folgend beschrieben werden.

### 4.2.1. Marketing Automation

Die Aufgabe von CRM im Bereich Marketing ist die Konzeption, Ausgestaltung und Durchführung der kundenbezogenen Kommunikation des Unternehmens, welche auf die verschiedenen Phasen des Kundenlebenszyklus abgestimmt ist [SJ04, S. 18 f.]. Für das Unternehmen ergibt sich durch organisierte und aufbauende Kommunikationsbeziehungen die Notwendigkeit dem Informations- und Lösungsbedarf eines Kunden in kontinuierlicher Form gerecht zu werden. Beispielsweise unterstützen die Kampagnenmanagementapplikationen die Marketingmitarbeiter bei der Kampagneplanung, -optimierung, -durchführung und -erfolgsanalyse [Gei06, S. 142]. Folgende Bereiche lassen sich für das Marketing identifizieren [AS02, S. 29 f.]:

- Analysefunktionalitäten ermöglichen, primär durch Selektions- und Filterverfahren, die Untersuchung von Branchen-, Markt- oder Kundendaten hinsichtlich bestimmter Eigenschaften sowie die Bildung von Kundensegmenten bzw. Kundengruppen.
- Das Territorien-Management ermöglicht die Berücksichtigung territorialer Eigenschaften, was z.B. bei der Bildung von Kundengruppen notwendig ist. Eine Einteilung kann mehrere Ebenen (z.B. Land, Region, Stadt sowie Stadtteil) umfassen und durch die Berücksichtigung geographischer Aspekte bei der Einsatzplanung besonders die wirtschaftliche Gestaltung des Außendienstes fördern.
- Lead-Generierung und Lead-Management sind für die Identifizierung und dem Management von Kaufinteressenten wichtig. Diese Kontakte können anhand festgelegter Kriterien qualifiziert und klassifiziert werden. Abhängig von Bedeutung und Potential eines Kontaktes werden unterschiedliche Folgeaktivitäten zur Intensivierung der Beziehung ausgelöst.

- Funktionalitäten des Kampagne-Managements bieten Analyse- und Selektionsfunktionen zur Definierung und Abgrenzung von Zielgruppen. Von besonderer Bedeutung sind hier die Schnittstellen zu Groupware-Lösungen und die Telefonintegration (CTI).
- Prognosefunktionalitäten generieren - aufbauend auf Analyseergebnissen - Prognosen, Trendvorhersagen und Szenarien. Auf dieser Grundlage lassen sich Zielerrechnungsgrade und Sollzustände für CRM-Aktivitäten festlegen. Die Messung des Erfolges ist nach der Soll-Ist Zustände mit den Zielvorgaben und den realisierten Ergebnissen möglich.

Wurde durch Analysen und Marketingaktivitäten der Kontakt zu Interessenten bzw. potentiellen Kunden hergestellt, folgen typischerweise Vertriebsaktivitäten, die den Erwerb eines Produktes fördern sollen.

### 4.2.2. Sales Automation

Sales Automation-Applikationen beinhalten Funktionen, Informationsangebote und Dienste, die primär Mitarbeiter bei der Planung und Durchführung der Verkaufsaktivitäten unterstützen und auf Innen- und Außendienst ausgerichtet sind. Dabei bildet das Kontaktmanagement, das sämtliche Interaktionen des Kunden speichert, die Basis für die Kundenbeziehung. Typische Funktionalitäten der Sales Automation sind [AS02, S. 30 ff.]:

- Verkaufsteamfunktionalitäten beinhalten Funktionalitäten für eine verteilte Aufgabenbearbeitung (Rollenmodelle, Zugriffsregelungen, Workflows und Möglichkeiten der Zeiterfassung).
- Produktkonfiguratoren ermöglichen auf Basis detaillierter Produktinformationen die Konfiguration, Modifikation und Kombination von Produkten.
- Angebotsmanagementfunktionalitäten basieren auf der Spezifikation des Angebotsgegenstands mittels Produktkonfiguration. Bei der Erstellung kundenspezifischer Angebote können im Rahmen der Preiskalkulation relevante steuerliche Regelungen, Rabattsysteme, Finanzierungsmöglichkeiten und unterschiedliche Verfahren der Zahlungsabwicklung berücksichtigt werden.
- Bestellaufnahmefunktionalitäten ermöglichen Aufträge oder Bestellungen direkt aufzunehmen, welche wiederum über Schnittstellen mit den Back-Office-Systemen gekoppelt sind. Vollständigkeits- und Konsistenzprüfungen sichern eine korrekte Erfassung der benötigten Daten.
- Mobile Computing unterstützt Mitarbeiter im Außendienst durch den Einsatz mobiler Systemkomponenten (wie Notebooks oder Handhelds)

und ist primär im Vertriebs- und Servicebereich von Bedeutung. Es sind vor allem Zugriffsmöglichkeiten auf unternehmensinterne Informationen, Daten und Funktionen notwendig, um dem Kunden die relevanten Funktionalitäten der entsprechenden Module vor Ort zur Verfügung zu stellen. Für eine aktuelle Informationsversorgung der Mitarbeiter ist die kontinuierliche Synchronisation zwischen Datenbeständen unternehmensinterner und mobiler Systemkomponenten erforderlich.

Konnte ein Kunde durch erfolgreiche Vertriebsaktivitäten zum Erwerb eines Produktes motiviert werden, ist in der After-Sales-Phase der Kundenservice sowie die Kundenbetreuung von Bedeutung.

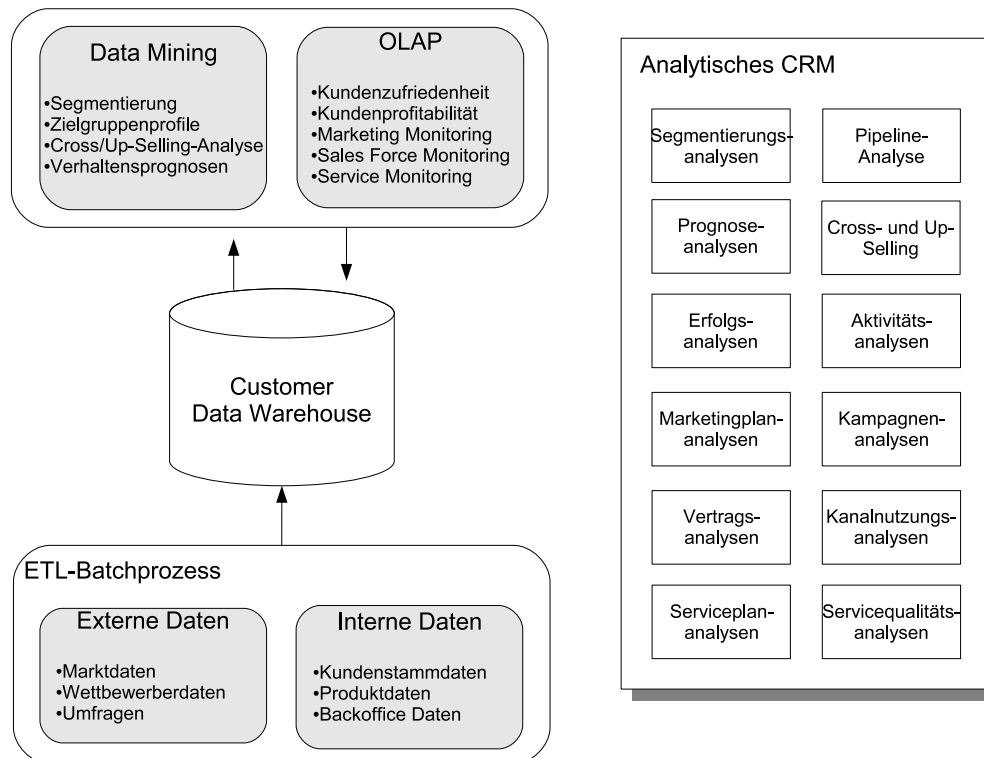
### 4.2.3. Service Automation

Service Automation-Applikationen dienen zur Umsetzung unternehmerischer Serviceaufgaben. Besonders hilfreich sind diese bei der Bearbeitung von Kunden-Inbounds im Bereich Reklamation bzw. Beschwerde und bei Aktivitäten des Kundenservices [AS02, S. 31]. Funktionen wie Case Management, Serviceauftragsabwicklung, Problemlösung, Beschwerdebearbeitung und -auswertung sind dabei typische Service Automation-Applikationen [Gei06, S. 142], die folgende Funktionalitäten beinhalten [AS02, S. 31]:

- Funktionalitäten zur Verwaltung von Service-Level-Agreements (SLAs) ermöglichen Inhalte von Rahmenverträgen mit Kunden, hinsichtlich unternehmerischer Service- oder Supportleistungen, im System zu hinterlegen und bei der Klassifizierung eingehender Reklamationen automatisiert zu überprüfen, ob die Bearbeitung im Leistungsumfang des Rahmenvertrags enthalten ist.
- Funktionalitäten zur Unterstützung des Qualitätsmanagements beinhalten Auswertungs- und Analysemöglichkeiten, um anhand vorliegender Beschwerden häufige Fehlerquellen oder Schwachpunkte von Produkten identifizieren zu können.
- Funktionalitäten zur Unterstützung des Kundendienstes sind auf die Integration von Beschwerdemanagement und Kundendienstaktivitäten am Kundenstandort ausgerichtet, wobei die Außendienstmitarbeiter bei Serviceaktivitäten durch mobile Systemkomponenten unterstützt werden.
- Funktionalitäten zur Unterstützung von Informationsanfragen von Kunden sollen die Bereitstellung und Übermittlung entsprechender Informationsmaterialien unterstützen.

## 4.3. Analytisches CRM

Unter das analytische CRM fallen alle Aktivitäten der Datensammlung und -auswertung. Entsprechend den chronologisch oder inhaltlich festgelegten Zyklen und Ad-hoc Berichten generieren die Berichtsfunktionen, unter Einbeziehung definierter Kennzahlen, Indikatoren und Analyse Kriterien von systemübergreifenden Datenbeständen, automatisch Auswertungen [AS02, S. 29].



**Abbildung 4.3.:** Business Intelligence im CRM [Gei06, S. 127 ff.]

Die durch das analytische CRM gewonnenen Erkenntnisse werden wiederum im Customer Data Warehouse gespeichert und können entweder direkt oder später, für operative Maßnahmen, genutzt werden [Her01, S. 11].

### 4.3.1. Data Warehouse

Die Aufgabe eines Data Warehouses (DWH) im Sinne von CRM besteht darin, die Kundendaten, Produktdaten, Marktdaten sowie Back-Office-Daten in eine einheitliche Systemumgebung zu integrieren, damit eine geeignete Grundlage für strategische Analysen geschaffen werden kann. Ausgehend von den operativen Daten des CRM werden weitere interne und externe Informationsquellen auf der Speicherebene eines Data Warehouses aufgebaut bzw. mehrere Data Marts errichtet [BS00, S. 2 f.]. Durch einen Extract-, Transform-

Load-Prozess (ETL)<sup>2</sup> wird gewährleistet, dass die Daten aus unterschiedlichen Datenquellen in einer einheitlichen Datenbank vereinigt werden. Die dauerhafte Sammlung von Informationen steht dann, themenorientiert nach Objekten (Produkte, Kunden, Märkte etc.), für zeitbezogene Anfragen zur Verfügung.

### 4.3.2. Online-Analytical-Processing

Um eine flexible und dynamische Analyse zu gewährleisten, erfolgt mit dem OLAP-Ansatz auf der Modellierungsebene eine mehrdimensionale Abbildung der DWH-Datenbasis [BS00, S. 2]. Da OLAP-Werkzeuge durch Multidimensionalität charakterisiert werden, spricht man von der Technologie eines interaktiven Zugriffs, die eine Vielzahl von Sichten und Darstellungsweisen ermöglicht. Zur bildlichen Darstellung werden Würfel verwendet, die die relevanten Kennzahlen anhand unterschiedlicher Dimensionen (z.B. Kunden, Regionen, etc.), mehrdimensional betrachten und bewerten [Pre02, S. 21]. Dadurch haben die Anwender die Möglichkeit, sich einen schnellen, direkten und interaktiven Zugriff auf möglichst verschiedenen Datenbestände zu verschaffen [Col01, S. 5 ff.].

### 4.3.3. Data Mining

Ebenfalls zum analytischen CRM zählt das Data Mining, innerhalb dessen umfangreiches Datenmaterial anhand von automatisierten Methoden durchsucht wird. Dadurch lassen sich aus großen Datenmengen bisher unbekannte, verständliche und potentiell nützliche Muster, Regeln und Abhängigkeiten herausfinden [Her01, S. 5 f.]. Data Mining beschreibt auf der Grundlage von Methoden aus den Bereichen der Statistik, Neuronale Netze und maschinellen Lernen den Prozess des Entdeckens von Wissen in strukturierten Massendaten [Pre02, S. 23]. Die durch das Data Mining entdeckten und als relevant identifizierten Informationen, Muster und Beziehungen werden wiederum im DWH gespeichert [Pre02, S. 23]. So wird eine Grundlage zur Informationsgewinnung und Entscheidungsvorbereitung geboten, die sowohl eine quantitative als auch qualitative Basis liefert [Her01, S. 13 f.].

## 4.4. Kollaboratives CRM

Über standardisierte Schnittstellen und ein Web-Frontend sollen kollaborative Geschäftsprozesse zwischen Partner-Unternehmen oder/und Kunden unterstützt werden [BE01, S. 203]. Dabei werden bei der B2B-Kooperation funktionelle Komponenten aus den Bereich des operativen CRM über ein Extranet-Portal personalisiert zur Verfügung gestellt. Bei der B2C-Kooperation werden

---

<sup>2</sup>Ist die Überführung heterogener Daten in ein Data Warehouse, was in drei Schritten passiert, die als Extraktion, Transformation und Laden bezeichnet werden.



hauptsächlich Self-Service-Funktionalitäten (beispielsweise FAQs, Diagnose- bzw. Entscheidungsbäume und Lösungsverfahren für eine eigenständige Problembehebung) angeboten.

#### 4.4.1. Internet-Portale

Das Hauptziel für den Einsatz von Portalen ist das Zusammenführen von verschiedenartigen Inhalten unter einer einheitlichen Benutzungsoberfläche. Im Kontext von CRM soll das Internet-Portal als individuelle Kommunikationschnittstelle zwischen Unternehmen, Kunden und Interessenten dienen [AS02, S. 33]. Von großer Bedeutung sind daher Personalisierungsfunktionen die es jedem Benutzer erlauben, auf die für ihn interessantesten und wichtigsten Information zuzugreifen [Glä03, S. 199]. Unter Berücksichtigung der relevanten Zugriffsrechte wird der Aufbau und die Oberfläche (GUI) der Client-Seite an die Interaktionsmöglichkeiten des jeweiligen Anwenders angepasst. Typische Funktionsbereiche von Portalen sind [AS02, S. 33 f.]:

- Funktionalitäten des Wissens- und Informationsmanagements
- Funktionalitäten zur individuellen Dialogführung und
- Kunden-Funktionalitäten

Mit diesen Funktionalitäten wird die vollständige Unterstützung sämtlicher Phasen der Anbieter-Kunden-Beziehung angestrebt.

#### 4.4.2. Portal Referenzarchitektur

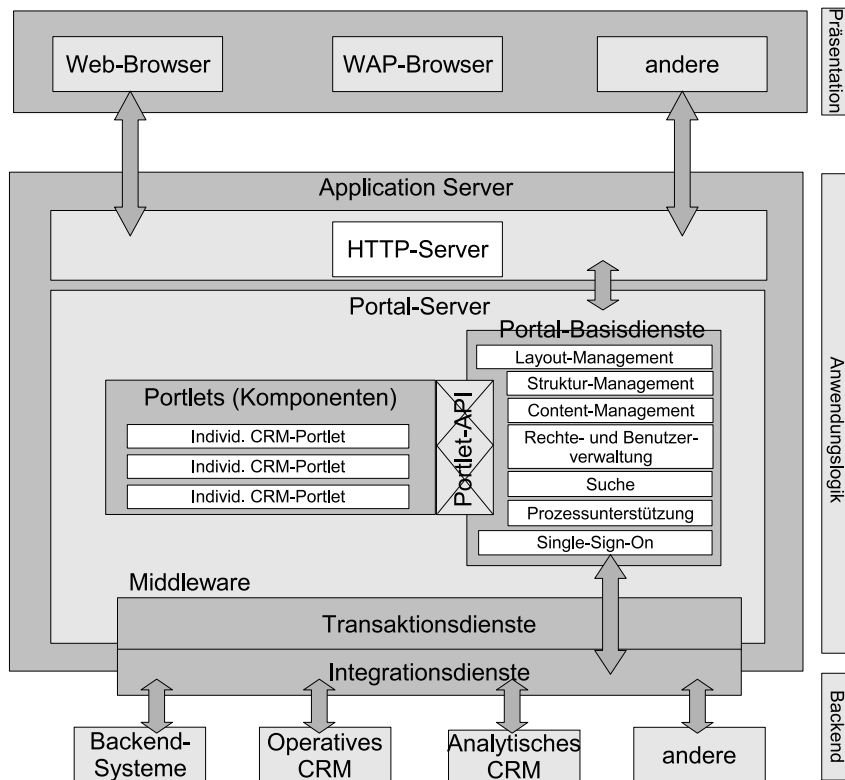
In der folgenden Beschreibung wird die Portal-Referenzarchitektur von Gurzki und Hinderer als Grundlage für ein kollaboratives CRM-Portal herangezogen [GH03, S. 158 ff.].

Die Aufteilung der Schichten erfolgt in Präsentation, Anwendungslogik und Backend. Endgeräte der Portal-Nutzer, wie z.B. Web-Browser, werden der Client-Schicht zugeordnet. Die Anwendungslogik besteht aus den Bereitstellungsdiensten und aus dem Portal-Server. Über EAI-Funktionen gewährleisten die Integrations- und Transaktionsdienste die Datenaggregation zwischen der Backend-Schicht und der Anwendungslogik.

Die Portlets, die aus allgemeinen Portlets-Anwendungsklassen instanziiert werden - wie das Portlet-Standard JSR 168<sup>3</sup>, werden über die Portlet-API

---

<sup>3</sup>JSR 168: Java Portlet Specification ist ein akzeptierter Standard, welcher die Schnittstelle, minimale Funktionen und den Lebenszyklus einer Portalkomponente (Portlet) beschreibt. Für weitere Informationen siehe folgenden Link: <http://www.jcp.org/en/jsr/detail?id=168>.



**Abbildung 4.4.:** Referenzarchitektur für Portal-Software [GH03, S. 160]

vom Portal aufgerufen. Das Struktur-Management liefert dabei den strukturellen Aufbau für die Personalisierungsfunktionalitäten. Zusätzlich wird im Struktur-Management definiert, welches Portlet fest definiert ist und welches optional und damit durch den Nutzer konfiguriert werden kann. Dieser Basisdienst wird auch der Personalisierungs-Engine zugeordnet, die regelbasierte Anwendungen und Inhalte zielgruppenspezifisch anbietet. Die vom User aufgerufenen Portal-Seiten werden durch das Layout-Management, entsprechend dem Endgerät gerendert. Dabei werden strukturelle Vorgaben, Berechtigungen sowie Layout-Vorgaben des Portal-Betreibers und/oder des jeweiligen Nutzers im Bezug auf Farbschemata, Platzierung der Visualisierungskomponenten und anderen grafischen Elementen (wie z.B. Bilder), berücksichtigt. Durch den Basisdienst "Suche" werden verschiedene heterogene Datenbestände durchsucht. Die Benutzerverwaltung kann durch die Portal Software für die Kunden realisiert werden, wobei für die internen User auf eine externe Benutzerverwaltung, in Form eines unternehmensweiten Verzeichnisdienstes, zurückgegriffen wird. Der Single-Sign-On-Dienst meldet die Nutzer des Portals automatisch an allen im Portal integrierten Systemen an. Zum Schluss werden, mit dem Basisdienst der Prozessunterstützung, Workflow-Mechanismen durch eine Workflow-Engine zwischen den integrierten Anwendungen gesteuert.

## 4.5. Risiken und kritische Merkmale eines CRM On Premise Betrieb

In dem gleichen Maß, in dem CRM-Software für Unternehmen zu einem zwingenden Mittel wird um wettbewerbsfähig zu bleiben, stellt die kostengünstige Einführung und der leistungsstarker Betrieb einen immer wichtigeren Indikator dar. Daher stellt sich nur die Frage, ob das CRM-System als On Premise oder als On Demand Anwendung eingesetzt werden soll.

Eine On Premise Implementierung bis zur Produktivschaltung kann dabei häufig mehr als drei Monate in Anspruch nehmen, wobei ein Großteil dieser Zeit für die Installation der Hardware und deren Konfiguration benötigt wird [Ede07, S. 66 f.]. Des Weiteren bedarf es organisatorischer und humankapitalintensiver Anstrengungen, damit das CRM-System unterbrechungsfrei und zuverlässig verwendet werden kann. Dafür müssen die Server in voller Leistungsfähigkeit zur Verfügung stehen. Zudem kann die CRM-Software, je nach Dauer der Einführung, bereits kurz nach Ende der Implementierung den veränderten Anforderungen nicht mehr entsprechen. Deshalb wird der Softwarelebenszyklus, von der Anforderungsanalyse über die Implementierung bis zum Einsatz, mehrmals durchlaufen, da die Lösung immer wieder neu entstehenden Anforderungen gerecht werden muss. Aktualisierungen sind auch ein Bestandteil der komplex aufgebauten Struktur von IT-Utilities - wie Betriebssysteme, DBMS-Software, Webserver, Firmware, Sicherheitshardware und -software, usw. - die die Betriebsumgebung von Datacentern ausmachen. Die fortwährende Verfügbarkeit neuester Hardware und aktueller Software ist daher ein nicht zu unterschätzender Kostenfaktor, der ex ante nicht abzusehen ist, da einerseits wieder Implementierungskosten anfallen und andererseits, bei fehlenden Know-How, zu Lasten der Systemstabilität führen kann. Sowohl die bloßen Betriebskosten des Datacenters als auch die Energie- und Gebäudekosten tragen dazu bei, dass i.d.R. mehr als 80 % des IT-Budgets für den Betrieb aufgewendet werden müssen [Ede07, S. 66].

Als Folge der Globalisierung setzen unternehmensorientierte Zentralisierungsansätze auf eine zunehmende Flexibilisierung in Bezug auf übergreifende Informationsversorgung, damit die oft weltweit verstreuten Mitarbeiter von einer zentralen Datenhaltung und den Möglichkeiten des weltweiten Zugriffs profitieren. Da für ein globales CRM wesentliche geschäftsrelevante Segmentierungskriterien in Bezug auf Kunden, Produkte, Lieferanten und Vertriebskanäle vorausgesetzt werden, sind abgesehen von der langen Verzögerung - hinsichtlich des Aufbaus der zugrundeliegenden dezentralen Infrastruktur - auch verhältnismäßig hohe Kosten für die Bereitstellung der notwendigen Server-Kapazitäten und Softwarelizenzen einzuplanen. Neben dem langen Roll-Out ist - überall dort, wo eine entsprechende IT-Infrastruktur und das Know-How nicht gegeben sind und diese dadurch teuer zur Verfügung gestellt werden müssten - die organisa-

torische Integration der CRM-Prozesse über die Grenzen der Abteilungen oder des Unternehmens hinweg, zusätzlich einer aufwändigen Wartung, zeitraubenden Aktualisierungen und umfassender Betreuung ausgesetzt. Dabei kann die Neugestaltung der kundenzentrierten Wertschöpfungskette bedeuten, die dafür notwendige Technologie zur durchgängigen Unterstützung der primären Funktionen von Marketing, über Vertrieb, bis hin zum Service, an spezialisierte Unternehmen bzw. direkt an die Softwarehersteller abzugeben, um eine Reduktion des notwendigen Investitionsvolumens in der Infrastruktur sowie in den laufenden Betriebskosten zu erreichen und einen einheitlichen Standard zu finden, der im globalen Einsatz ein einheitliches CRM-System für alle Niederlassungen gewährleistet.

### 4.5.1. CRM-Software im globalen Umfeld

Da die Wettbewerbsfähigkeit von Unternehmen unter anderem von der Organisationsstruktur abhängig ist, sind mittlerweile viele Unternehmen dezentral organisiert, um flexibel auf die sich immer schneller ändernden Anforderungen der Absatzmärkte reagieren zu können [Rie05, S. 46]. Würde die Optimierung aus einer lokalen Perspektive erfolgen, könnten die Unterschiede zwischen den einzelnen Standorten verstärkt werden. Es bedarf daher einem global zugänglichen CRM-System, das die Prozessprinzipien für eine kundenzentrierte Geschäftsorientierung konsequent umsetzt und die lokal verteilten Stammdaten zu einer globalen Stammdatenstruktur konsolidiert.

Im Vorfeld der Entscheidungsfindung zur Auswahl eines CRM On Demand Systems ist es daher wichtig festzulegen, welche eigenen Zielsetzungen das Unternehmen mit dieser Nutzungsform verfolgt. Dazu ist es notwendig die Kriterien, die ein entsprechendes System erfüllen muss, mit den Zielsetzungen (vor allem im Rahmen einer Internationalisierung) in Einklang zu bringen. So sind die inhomogenen und verteilten Organisationseinheiten sowie deren vorhandenen internen IT-Ressourcen und die Notwendigkeit einer kurzen Implementierungszeit besonders zu berücksichtigen. Hier sind unterschiedliche Reifegrade und Arbeitsweisen der jeweiligen Organisationseinheit wie auch kulturelle Unterschiede zu bedenken. Plakativ ausgedrückt, soll eine schnelle und den Anforderungen gerechte Einführung ermöglicht werden, wobei potenzielle Kostenvorteile nicht in den Vordergrund zu stellen sind, wohl aber die geringe Kapitalbindung, die eine Flexibilität hinsichtlich der Art und der Dimension des Einsatzes ermöglicht.

Vor allem für die zentrale Verkaufsleitung ergeben sich dadurch Fakten in Form von Statistiken und Reports, über die dezentral verteilten Organisationseinheiten. Deshalb muss die zentrale Datenhaltung für operative und analytische Zwecke rollenbasiert zugänglich sein. Aspekte, wie die Überbrückung verteilter Standorte, die Beherrschung von kunden- und vertriebsintensiven

Prozessen und die Integration von länderspezifischen Informationsquellen, sind neben den niedrigen Initialkosten für die IT-gestützte Zentralisierung und der Möglichkeit das System bei unerfüllten Erwartungen nicht weiter verwenden zu müssen, ausschlaggebend.

### 4.5.2. Erfolgsfaktoren für CRM On Demand

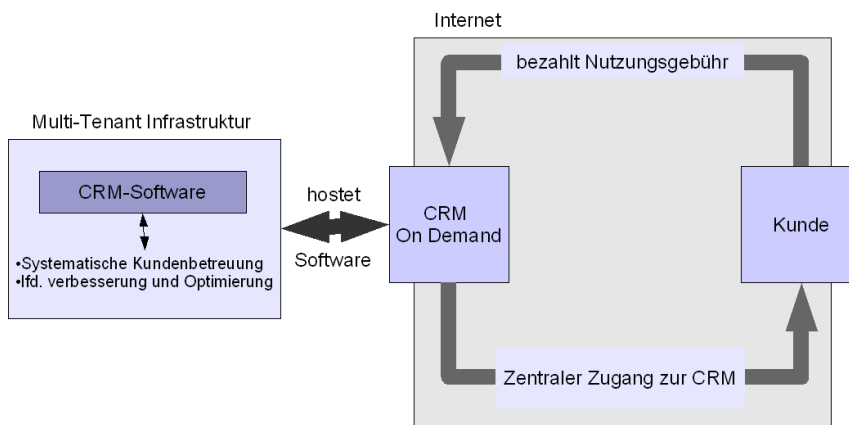
Mit den Begriffszusatz “On Demand” wird der Bereich Customer Relationship Management (CRM), um ein neues Geschäftsmodellen für Unternehmenssoftware erweitert. Dieses Geschäftsmodell basiert nicht mehr auf dem Verkauf eines CRM-Systems, sondern der zentralen Bereitstellung sowie der Bezahlung bei Benutzung. . Es sind daher grundsätzlich wirtschaftliche Argumente die CRM On Demand, im Bereich der eher kostspieligen und durch kurze Innovationszyklen geprägten CRM-Softwarelösungen, attraktiv machen.

CRM On Demand ist somit eine Möglichkeit, die Kosten, die durch eine On Premise Lösung entstehen würden, besser in den Griff zu bekommen, und die betriebliche Handlungsweise flexibler zu gestalten. Damit sind potenzielle Vorteile verbunden wie z.B. [Sti07, S. 47]:

- **Kosteneinsparungen:** CRM kann im On Demand-Modell effizienter genutzt und besser ausgelastet werden, da die Kosten für Administration und Support unter den verschiedenen Nutzern der On Demand Plattform geteilt werden. Außerdem erscheinen die Folgekosten für Wartung und Pflege nahezu vernachlässigbar, da in diesem Fall nur die Clients (PC) und die Netzverbindung instand zu halten sind.
- **Geringere Investitionsrisiken:** Der On Demand Anbieter trägt die Kosten für die Software sowie die IT-Infrastrukturkomponenten und übernimmt damit auch das Investitionsrisiko, während der Anwender nur für die aktuelle Nutzung oder den tatsächlichen Verbrauch zahlt.
- **Bessere Performance und Sicherheit:** Spezialisierte Anbieter, für die der On Demand Betrieb das Kerngeschäft darstellt, haben in der Regel mehr Ressourcen für die Sicherstellung von systemkritische Anforderungen (Performance, Security) zur Verfügung.

Die Auswahl der CRM-Software kann nach betriebswirtschaftlichen Funktionalitäten sowie nach der Eignung für die betrieblichen Anforderungen getroffen werden, ohne weitergehend Rücksicht auf mögliche Kompatibilitätsprobleme mit der unternehmenseigenen Infrastruktur nehmen zu müssen. Zudem werden die kundenzentrierten Daten auf Basis webbasierter Speichermechanismen und unter Berücksichtigung datenschutztypischer Anforderungen, auf einer zentral

#### 4.5. RISIKEN UND KRITISCHE MERKMALE EINES CRM ON PREMISE BETRIEB

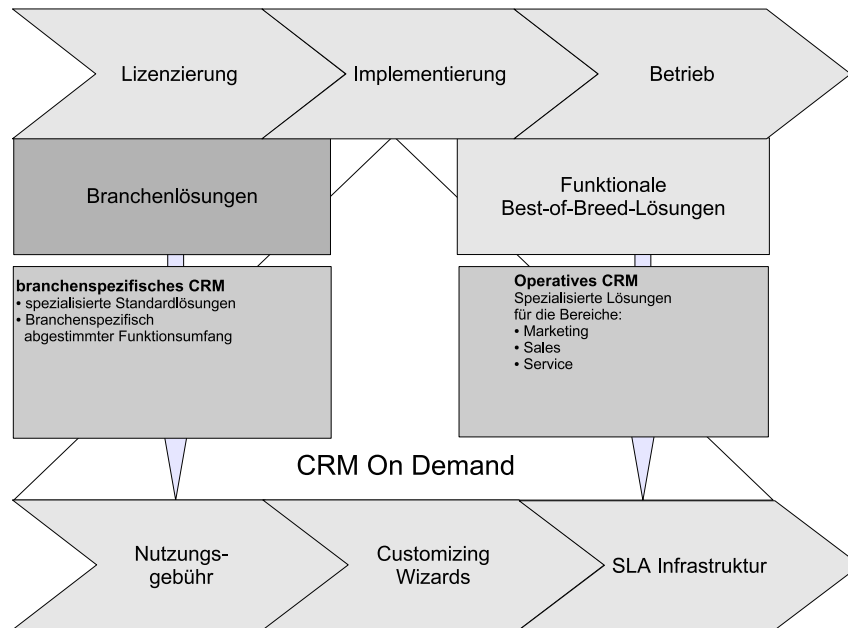


**Abbildung 4.5.:** Nutzung von CRM On Demand

zugänglichen CRM-Plattform gespeichert. Dadurch entfallen die Aufwendungen für Transaktions- und Synchronisationskosten, da mit einer global bereitgestellten CRM-Lösung eine konsolidierte Datenbasis für das Forecasting und Reporting ermöglicht wird. Ein CRM-Projekt im On Demand-Modell wird damit besonders für solche Unternehmen interessant, die mit einer dezentralen Vertriebsorganisation arbeiten und ihr Kontakt- und Opportunity-Management über Grenzen und Zeitzonen hinweg vereinfachen möchten [App02, S. 55]. Daraus folgernd ermöglicht die Flexibilisierung der betrieblichen Handlungsweise die Reaktionsgeschwindigkeit - hinsichtlich technologischer Vernetzung von geografischen Standorten und deren notwendigen Adaptionen für ein globales und konsistentes CRM bei gleichzeitiger Modernisierung der IT-Landschaft - signifikant zu steigern, da die CRM-Software unabhängig von Ort und Zeit für die technologische Unterstützung der kundenzentrierten Prozesse sofort produktiv eingesetzt wird. Deshalb korreliert eine extern vorgehaltene und über den Browser nutzbare On Demand Software immer mehr mit Trends, wie der technologischen Vernetzung und der Virtualisierung der Leistungserstellung. Die Virtualisierung der Leistungserstellung resultiert daraus, dass die CRM-Bereitstellung aufgrund Kosteneffizienz (geringe Anfangsinvestitionen), Geschwindigkeit (keine Implementierung) und Flexibilität (einfacher Zugang) durch einen losen und zeitlich begrenzten Zusammenschluss mit einem Partnerunternehmen (CRM-Hersteller) erfolgt.

Auch wenn zwischen den einzelnen Angeboten Unzulänglichkeiten (etwa in Bezug auf die Modifizierbarkeit) bestehen können, erhalten Unternehmen über die Anbieter von CRM On Demand in der Regel Zugriff auf Best-of-Breed-Lösungen, welche die funktionalen Bereiche für Marketing, Sales und Service autonom in einem Webportal darstellen und über Webserviceschnittstellen eine entsprechende Datenintegration für analytische Vorgehensweisen liefern. Dabei variieren, je nach Anwendungsziel, die Möglichkeiten: von der Vertriebssteuerung über das Tele- und Internetmarketing bis hin zur kompletten Vollintegration

#### 4.5. RISIKEN UND KRITISCHE MERKMALE EINES CRM ON PREMISE BETRIEB



**Abbildung 4.6.:** Überblick über CRM On Demand

tion von weiteren Systemen. Einige Anbieter stellen auch, basierend auf branchenspezifischen Referenzmodellen, vorkonfigurierte Lösungen (“Best Practices”) auf Basis unterschiedlicher Module für verschiedene Branchen bereit. Aber auch zwischen diesen beiden Ausprägungen sind zahlreiche Abstufungen denkbar [ZS07, S. 266]. Die linearen Nutzungsgebühren sind langfristig nicht in den Vordergrund zu stellen, wohl aber die geringe Kapitalbindung, die eine Flexibilität hinsichtlich der Art und der Dimension des Einsatzes ermöglicht. Da sich der Anbieter neben den lästigen Begleitthemen auch der IT annimmt, (wie z.B. gesetzeskonformer Betrieb und Sicherung von Daten, Testen und Einspielen von Patches sowie Updates, Abwehr von Sicherheitsbedrohungen etc.) wird dem nutzenden, respektive auslagernden Unternehmen geholfen die für das Kerngeschäft identifizierten Aktivitäten zu fokussieren. Mit wenig zeitlicher Absorption für die Aufgaben entlang des gesamten Lebenszyklus der CRM-Lösung, bleibt mehr Zeit für die Optimierung der Prozesse und Organisationseinheiten, um im nationalen wie im weltweiten Wettbewerb bestehen zu können.

#### 4.5.3. Anforderungen an CRM On Demand

Die gesamte Wartung, Installation, Administration und Sicherheit des On Demand Angebots liegt in der Verantwortung des Anbieters, der sein CRM-Portfolio als Service über das Internet anbietet. Der entscheidende Unterschied zum firmeninternen Betrieb ist der Umstand, dass die kritischen Daten meh-



rerer Firmen (Mandanten) gemeinsam auf dem Server des Anbieters liegen. Aufgrund der Spezialisierung und der dafür notwendigen Kapazitäten ist der Sicherheitsstandard in den Rechenzentren der Anbieter selbstverständlicher als bei einer internen Infrastruktur. Der Anbieter ist nämlich gegenüber seiner gesamten Anwenderbasis zu einem unterbrechungsfreien Service verpflichtet.

Über die offensichtlichen Vorteile hinaus, dass keine besonderen Tätigkeiten, keine Planung, keine Tests bzw. keine Priorisierung durchgeführt werden müssen, ist als Ziel einer erfolgreichen CRM On Demand-Einführung die Stimmigkeit zwischen [Krc05, S. 288]:

- der Unternehmenskultur,
- den Organisationsprinzipien,
- der tatsächlichen Organisationsstruktur im Unternehmen
- und den spezifischen Chancen,

zu beachten. Im Vorfeld der Entscheidungsfindung zur Auswahl eines On Demand Systems ist es somit wichtig festzulegen, welche eigenen Zielsetzungen das Unternehmen mit dieser Nutzungsform verfolgt. Der Reifegrad von CRM On Demand reicht hierbei von der Stützung operativer CRM-Prozesse ohne Anpassungsmöglichkeiten bis hin zu parametrisierbaren CRM-Modulen. Daher benötigt das Management eines Unternehmens einerseits eine klare Vorstellung von den Zielvorgaben und andererseits eine offene Haltung gegenüber innovativen Geschäftsprozessen, die sich durch den Einsatz der CRM-Software ergeben können. Es ist vor allem zu bedenken, dass in Abhängigkeit des integrierten Customizingrahmens die Abbildung von kundenindividuellen Prozessen mit Schwierigkeiten verbunden sein könnte. Dazu ist es notwendig die Kriterien, die ein entsprechendes CRM-System erfüllen muss, mit den Zielsetzungen vor allem im Rahmen eines integrierten CRM-Ansatzes durch die Verzahnung des Front-Office mit den Back-Office, in Einklang zu bringen. Daher ist die tatsächliche Integration von CRM On Demand in die betrieblichen Prozesse des anwendenden Unternehmens entscheidend für den Nutzen. Um hier gegenüber Anbietern und Anwendern keine starken Divergenzen hinsichtlich der Erwartungshaltung aufkommen zu lassen, ist ein gut durchdachter Customizingrahmen für die Anpassung und der Integration der zugänglichen CRM-Kernkomponenten Voraussetzung. Dabei lässt sich der Customizingrahmen in folgender Topologie anordnen:

- **wenig Customizing:** Die nötigen Parameter für den Gebrauch einer relativ spezifischen Software nötigen Parameter lassen sich parametrisieren. Dabei soll die Auswahl bestimmter Felder und die Strukturierung des grundlegenden Fensterlayouts anpassbar sein.



- **mittleres Customizing:** Zusätzlich lassen sich neben benutzerdefinierten Feldern und Objekten, Validierungsregeln, Seitenlayouts sowie auch Workflows einrichten und verwalten, ohne dabei den Anbieter kontaktieren zu müssen. Außerdem sind weitere Module auf Anfrage (On Demand) zuschaltbar.
- **hohes Maß an Customizing:** Die Nutzung des vorhandenen Datenmodells ist für das Design von komplexen Strukturen möglich. Zudem lassen sich Datenmodelle von Fremdanwendungen integrieren. Daher sind standardisierte Schnittstellen auf der technische Ebene sowohl für Massendatenanforderungen und -übertragungen als auch für Datenmanipulationsmöglichkeiten notwendig. Das bedeutet, dass die Daten beliebig gelesen und geschrieben werden können, um jede denkbare Integrationsanwendung realisieren zu können [vG08, S. 36]

Im Zentrum von CRM On Demand stehen somit über ein Webportal zugängliche CRM-Front-Office-Systeme, die meistens über eine modulare Architektur verfügen und auf gemeinsame zentrale Kernkomponenten aufbauen [AS02, S. 41]. Durch Modulkonfiguration, -kombination und -modifikation können unternehmensspezifische Anforderungen bei der individuellen Systemgestaltung des Front-Offices von CRM On Demand berücksichtigt werden. Im Vergleich zur individuellen On Premise Lösung kommen hier immer wieder Bedenken bezüglich der Konfigurierbarkeit, Anpassungsfähigkeit und Datensicherheit auf [ZS07, S. 261]. Ein Customizing Wizard sorgt dafür, dass die geeignetste Ablaufalternative aus einer Vielzahl möglicher Varianten von Funktionen und Prozessen per Umgebungsvariablen konfiguriert werden kann. So müssen für die CRM-Module Sales, Service und Marketing folgende Aspekte für die Parametrisierung berücksichtigt werden können:

- Wiedergabe der Organisationsstruktur:
  - Abbildung der Organisationsschemata durch ein spezifisches Zugriffsmo-  
dell;
  - die Erstellung von Profilen, die festlegen mit welchen Objekten und  
Feldern gearbeitet werden darf;
  - ein Freigabemodell für Rollen, das pro Datensatztyp festlegt, wer die  
Daten sehen kann und wie das Layout der einzelnen Datenfeldern  
aussehen soll;
  - Zusammenfassung verschiedener Rollen zu einer Gruppe für Grup-  
penzugriffsrechte;
- die Einrichtung und Verwaltung von Workflows für komplexe Geschäftspro-  
zesse und Genehmigungsverfahren ermöglichen:

#### 4.5. RISIKEN UND KRITISCHE MERKMALE EINES CRM ON PREMISE BETRIEB

---

- Regeleinstellungen, die bei Geschäftsaktionen ausgelöst werden (z.B. E-Mail Benachrichtigung), wenn z.B. ein Datensatz gespeichert oder erstellt wird;
- automatische Feldaktualisierung durch interne Triggermechanismen, wenn eine Workflow-Regel oder ein Genehmigungsprozess ausgelöst wird;
- die Definition von Ampelfunktionen mit individueller Merkmalsverknüpfung, die auf Schwellenwerte basieren;
- den Zugriff auf Daten verschiedener Datenbestände über APIs gewährleisten:
  - funktions- und leistungsfähige Kopplung mit bereits im Unternehmen eingesetzten Standard-Textverarbeitungssystemen (MS Office) und E-Mail-Clients (Lotus Notes, MS Outlook) durch standardisierte Konnektoren;
  - Middleware Konnektoren für Outbound-Messaging, die standardisierte SOAP-Calls an externen Systemen auslösen z.B. SAP, Oracle, JD Edwards, PeopleSoft, und Microsoft Dynamics;
  - integrierte Validierungsfunktionen (ETL-Funktionen, Mapping, Matching) bei Import- und Export-Wizards, um die Datenqualität für vorhandene Datenquellen - wie Kundenstammdaten, Artikelstammdaten, Kontaktpersonen, Auftragsdaten, etc. - zu gewährleisten, die als CSV-Dateien exportiert oder bei Updates importiert werden;
  - Datenintegration, wie z.B. Vertrags-Dokumente (DOC, PDF, ...) über ein Dokumentenmanagement System;
- einen entsprechenden Customizingrahmen anbietet:
  - für benutzerdefinierte Objekte wie Felder und Schaltflächen um spezifische Seitenlayouts für die Optimierung der Nutzungseigenschaften erstellen zu können;
  - für die Definierung individueller Such- und Lookup-Dialoge;
  - für die Anpassung des Systems auf unterschiedliche internationale Standards und Sprachen;
  - Computer Telephony Integration (CTI);

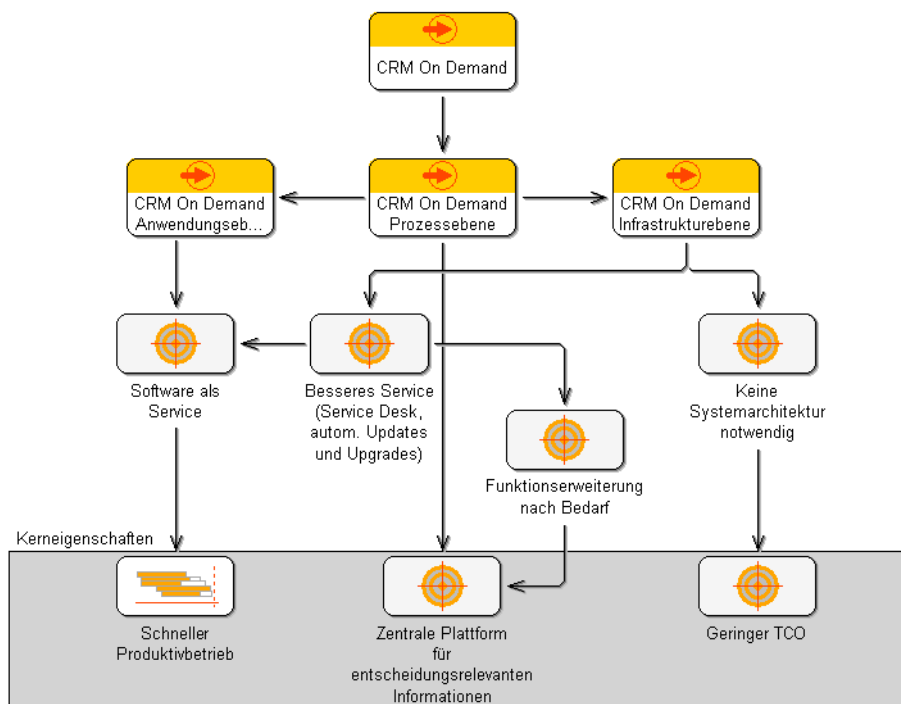
Darüber hinaus verfügen On Demand Systeme, typischerweise im Customizingrahmen, über standardisierte Schnittstellen, um z.B. den Bestellstatus eines ERP-Systems an CRM On Demand weiterzugeben oder Daten von Umsatz- und Marktentwicklungen auf einer Karte von Google Maps zur weiteren Datenrepräsentation anzeigen zu lassen [vG08]. Von besonderer Bedeutung ist

daher das Verhältnis oder die Gewichtung der Hauptkomponenten des CRM-Systems und ERP-Systems sowie die, durch die von der Wettbewerbssituation definierte Strategie und IT-Architektur des Unternehmens für Management- und Controllingzwecke.

Die Absicht von CRM On Demand ist in weiterer Folge die Senkung der Gesamtkosten und die Gewährleistung der Flexibilität. Zum einen, ist dies eine geeignete Methode im Optimierungsprozess von Konsolidierungsmaßnahmen. Zum anderen kann durch den Einsatz einer webbasierten CRM-Zentralisierung eine übergreifende Informationsversorgung über die gesamte Unternehmensstruktur hinweg erzielt werden.

#### 4.5.4. SaaS als Grundlage für CRM On Demand

Wie bereits im vorigen Kapitel erläutert, beschreibt der Begriff Software-as-a-Service (SaaS) einen fundamentalen Wandel in der Bereitstellung von Software-Services - von einem technologieorientierten Ansatz hin zu einem geschäftsorientierten Ansatz [Bre07, S. 86]. Dabei wird aus der Sicht des Softwareanbieters die CRM-Software in seinem Rechenzentrum zentral betrieben und entkoppelt, über eine einzelne Systemplattform, für alle Kunden zum Service umdeklariert.



**Abbildung 4.7.:** Nutzeffekte von CRM On Demand

SaaS - als Grundlage für CRM On Demand - wandelt somit den Eigen-

#### 4.5. RISIKEN UND KRITISCHE MERKMALE EINES CRM ON PREMISE BETRIEB

---

tumsbegriff, da nicht mehr die Eigentumsverhältnisse an einer Software relevant sind, sondern eine, gegen Entrichtung entsprechender Gebühren, zeitlich abgegrenzte Nutzungsmöglichkeit, die eine dem Geschäftsverlauf entsprechende Nutzung ermöglicht. Zudem macht sich, durch ein tiefes Verständnis der Serviceorientierung, eine starke Kundenfokussierung bemerkbar, da SaaS-Lösungen den sich ändernden Voraussetzungen und Trends angepasst werden, um den Kundenwünschen und -bedürfnissen besser zu entsprechen [ZS07, S. 261]. SaaS-Lösungen werden also kontinuierlich und proaktiv verbessert, um eine Erhöhung der Kundenzufriedenheit und Kundenbindung zu erreichen [ZS07, S. 260]. So kommt das Anwendungs- und Substitutionspotential von Software-as-a-Service, wie:

- schneller Produktivbetrieb,
- zentrale Plattform für entscheidungsrelevante Informationen und
- geringer TCO,

unter dem Aspekt einer schnellen und dynamischen Anpassung an globalisierte Wertschöpfungsketten bei einem flexiblen und selektiven Outsourcing, im Rahmen von CRM On Demand, zu tragen. Eine SaaS-basierte CRM-Software umfasst somit die zeitunabhängige Erbringung der definierten Nutzeffekte von CRM On Demand an einen Anwender auf Basis folgender Erfolgsfaktoren:

Einloggen und arbeiten	Die CRM-Software steht zur Verfügung, sobald die SaaS-Anwendung aufgerufen wurde, die Registrierung abgeschlossen ist und der User sich erfolgreich eingeloggt hat.
Von jedem Arbeitsplatz aus erreichbar	Mit einer SaaS basierten CRM-Software kann von jedem Arbeitsplatz, der ans Internet angeschlossen ist, gearbeitet werden.
Betriebssystem- und Plattformunabhängig	Ob mit Safari auf Mac, mit IE auf Windows oder Firefox auf Linux, Software as a Service funktioniert mit jedem Webbrowser der die aktuellen W3C-Standards unterstützt.
Problemlose Zusammenarbeit	Durch kollaborative Funktionen kann über Unternehmensgrenzen hinweg und im Kontext des betroffenen Prozesses eine problemlose Zusammenarbeit realisiert werden.
Updates und Bugfixes	Die CRM-Software wird vom Softwareanbieter laufend aktualisiert

#### 4.5. RISIKEN UND KRITISCHE MERKMALE EINES CRM ON PREMISE BETRIEB

---

Hohe Verfügbarkeit	Da der CRM-Hersteller seine Software als webbasierten Service bereitstellt, wird dieser entsprechende Aufwände betreiben und deutlich bessere Sicherheitsinfrastrukturen betreiben als typische Anwenderunternehmen, um dafür zu sorgen, dass die CRM-Software möglichst ausfallsicher läuft.
Hohe Skalierbarkeit und Flexibilität	Für den Bedarf weiterer Zugänge können einfach weitere User freigeschaltet werden. Genauso kann der funktionale Umfang im Rahmen eines vertraglichen Upgrades problemlos erweitert werden.

**Tabelle 4.2.:** Erfolgsfaktoren von SaaS [Gun07]

# 5. Analyse von CRM On Demand Software

Die in den vorigen Kapitel beschriebene Form der zentralen Bereitstellung standardisierter Anwendungen stellt eine Reihe von Anforderungen an eine CRM On Demand Software. In diesem Kapitel wird untersucht, inwieweit verfügbare CRM On Demand Lösungen diese Anforderungen erfüllen. Dabei werden ausgewählte CRM On Demand Lösungen vorgestellt, diskutiert und nach deren Schwerpunkten klassifiziert. Danach erfolgt eine Beurteilung der Durchführung von operativen Prozessen des Kampagnenmanagements und des Leadmanagements unter Berücksichtigung des Communication Managements. Dabei sind Informationen über den Systembetrieb, SLAs, der Customizingrahmen, die Integrations- und Erweiterungspotenziale, das Datenmanagement und das Usability die Grundlage des Analyseprozesses.

## 5.1. Zielstellung der Analyse und verwendetes Anwendungsszenario

Die bereits durchgeführte Darstellung der Zusammenhänge zwischen Software On Demand und webbasierenden Anwendungsdienstleistungen, hinsichtlich deren Ausrichtung auf organisatorische und technische Infrastrukturen, wird nun durch SaaS-basierende CRM-Angebote evaluiert. Ausgehend von typischen Anforderungen hinsichtlich Individualisierung von standardisierten SaaS Angeboten werden im Rahmen eines fiktiven Anwendungsszenarios ausgewählte CRM Suites analysiert und bewertet, um der Frage nachzugehen, inwieweit CRM On Demand ein individuelles Customizing ermöglicht oder wo die Grenzen liegen.

Dabei bilden sowohl die bisher betrachteten Aspekte, hinsichtlich der Anforderungen an solche Systeme, als auch die für CRM-Systeme relevanten Komponenten die Grundlage für den Bewertungsprozess, welcher insbesondere auf die Durchführung des Kampagnenmanagements und des Leadmanagements unter Berücksichtigung des Communication Managements ausgerichtet ist.

Zusammengefasst soll bei einem international aufgestellten Unternehmen, welches neben dezentralen Organisationseinheiten auch auf mehrschichtige Vertriebsstrukturen zurückgreift, der manuelle und administrative Verwaltungs-

aufwand beim Lead- und Kampagnenmanagement - bei gleichzeitiger Optimierung des Vertriebscontrollings - verringert werden. Vor allem im Hinblick auf eine zentral zugängliche Webplattform zur Aggregation und Verwaltung von externen Datenlieferanten, sollen die analytischen Kernkomponenten einer CRM On Demand Lösung - basierend auf integrierten Analysetypen - ein zentrales Forecasting und Reporting im Zuge des globalen Pipelinemanagements stützen. Dabei gilt es die CRM-Prozesse - basierend auf den definierten Anforderungen - auf der Grundlage standardisierter Salesprozesse über Unternehmensgrenzen hinweg zu homogenisieren und allen im Vertrieb, respektive im Salesprozess, involvierten Rollen ein hochverfügbares und über unterschiedliche Devices zugängliches State-of-the-Art-System für die Sales Force Automation zu bieten. Deshalb stellt ein einfaches Usermanagement und die Anforderungen an Multilingualität von solch On Demand Systemen ein weiteres Kriterienmerkmal dar. Jedenfalls soll einer aus dem fiktiven Anwendungsszenario abgeleiteter Kriterienkatalog einen transparenten Einblick in den Customizingrahmen ausgewählter On Demand Software geben.

### 5.1.1. Anforderungsszenario als Grundlage für die Analyse

Neben den benötigten Funktionen für die Umsetzung eines dezentralen Leadmanagement beinhaltet dieses fiktive Anwendungsszenario folgende Anforderungen:

1. Bei einem automatischen Lead Eingang durch ein über die Unternehmenswebseite bereitgestelltes Anfrageformular, soll über einen Webservice-Request eine Aktion im On Demand System ausgelöst werden, die anhand vorgegebener Regeln einen Genehmigungsprozess für die Ablehnung oder Qualifizierung eines Leads durchführt. Qualifizierte Leads sollen automatisch, mit Hilfe einer Zuordnungsregel, nach geographischen Kriterien und Produktkategorien einem potentiellen Vertriebsmitarbeiter korrekt zugewiesen werden. Dabei sollen, durch die Kopplung der On Demand Lösung mit einem Produktkonfigurator, die daraus erfassten Leads auf Basis von Routing- und Fallbackregeln qualifiziert werden, damit z.B. ein zentral erfasster Lead - anhand geschäftsrelevanten Segmentierungskriterien in Bezug auf Produkte sowie geographischer und altersspezifischer Merkmale - einen workflowbasierten Weiterleitungsprozess an einen entsprechenden Vertriebsmitarbeiter auslöst. Basierend auf den speziellen Kriterien für den betreffenden Lead muss über "if-then-Regeln" sichergestellt werden, dass Leads korrekt zugeordnet werden und die Workflow-Regel das Feld des Lead-Status automatisch mit einem bestimmten Wert "Web-to-Lead" aktualisiert. Nach einer Zuordnung müssen die Benutzer per E-Mail mit einer entsprechenden E-Mail-Vorlage benachrichtigt werden und gleichzeitig soll mit Hilfe automatischer Antwortregeln, basierend auf beliebigen Attributen des eingehenden Leads,

eine Vorlage für ein Antwortschreiben generiert werden. Dafür bedarf es einer MS-Office Integration: vor allem einer Integration mit MS-Word, das Informationen aus den Datensätzen der On Demand Lösung über Verknüpfungsfelder verwendet.

2. Die Weiterleitungsregeln müssen zusätzlich mit dem Standardwert "Web-to-Lead abgelehnt" für die Feldaktualisierung angepasst werden und zusätzlich eine Vertretungsregel berücksichtigen, die beispielsweise Standardinhaber für alle Leads festlegt, falls diese über die Zuordnungsregeln nicht korrekt zugewiesen wurden. Dasselbe gilt auch für die Lead-Qualifizierung: Wurde ein Lead vom System abgelehnt, so muss dieser manuell - durch eine dafür berechnigte Rolle - nochmals überprüft werden und gegenfalls an einen relevanten Vertriebsmitarbeiter weitergeleitet werden. Dieser kann wiederum in zweiter Instanz den Lead mit einer Begründung ablehnen und natürlich auch annehmen. Aus Gründen des Qualitätsmanagements dürfen bei allen endgültig abgelehnten Leads keine Feldaktualisierungen mehr durchgeführt werden, da diese für die Duplikatenüberprüfung herangezogen werden müssen.
3. Des weiteren bedarf es der Lead-Erfassung als Teil einer Kampagne. Aus Kampagnen gewonnene Leads müssen mit der entsprechenden Kampagne verknüpft werden können. Dabei sollen vordefinierte Templates für das Berichtswesen umfangreiche Extraktions- und Analysemöglichkeiten für das Tracking von Leads nach verschiedenen Prioritäten ermöglichen. Via Channel Analysen soll nachvollzogen werden, welche Leads durch welche Kampagnen erfasst wurden. Dabei soll auch nachvollziehbar sein, warum qualifizierte Leads abgesprungen sind oder abgelehnt wurden.
4. Um die generierten Leads effektiv im Vertrieb weiterbearbeiten zu können, müssen alle Grundfunktionen, die für das Lead-Management notwendig sind, vorhanden sein. In weiterer Folge sind im Rahmen der CRM-Prozesse, die Survey-Funktionalitäten für die strukturierte Qualifizierung und Verwaltung von Leads notwendig. Eine fortlaufende Auswertung des Erfolges der Aktionen soll die kontinuierliche Verbesserung der kundengerichteten Aktivitäten ermöglichen. Die ermittelten Kundendaten sollen, auf der Grundlage einer homogenen Kundenklassifizierung und mit Hilfe eines integrierten Opportunitymanagements, effizient verwaltet werden können.
5. Zur Führung des Benutzers durch komplexe Prozesse braucht man Assistenten, dynamische Masken und eine Bedienoberfläche, die trotz der Browserbasiertheit eine optimale User Experience bietet. Abgesehen von den üblichen Merkmalen der Corporate Identity (CI), sollen die Aspekte der individuellen Anpassungen für die Datenmasken und Listen beachtet werden. Besonders wichtig sind die vorgegebenen Möglichkeiten bzgl. der



Zugriffsrechte und Sichtbarkeit von Daten, die Anpassbarkeit auf internationale Sprachen und Währungen sowie die Verwaltung und Einrichtung von Workflows.

### 5.1.2. Begleitende Aspekte bei der Untersuchung

Ausgehend von den relevanten Anforderungen an CRM On Demand aus dem Kapitel 4.5.3, lässt sich zunächst ein qualitatives Anforderungsbündel definieren, das u.a. folgende Merkmale von Reussner in den Vordergrund stellt [RH06, S. 279 ff.]:

- *Sicherheit*: Das System muss vor unautorisierter Benutzung, Datennahme, Manipulation oder Zerstörung geschützt sein [RH06, S. 280].
- *Volle Integration/Schnittstellen*: Die kundeninternen Anwendungen müssen in der Lage sein, lesend und schreibend auf die ausgelagerte Software zugreifen zu können. Webservice-APIs bzw. standardisierte Konnektoren für defacto Systemstandards wie SAP, ORACLE, Microsoft und Lotus Notes ermöglichen dabei die Einbindung der als “Blackbox” wahrgenommenen Softwarefunktionalitäten und Datenbestände in die Unternehmensarchitektur des Kunden.
- *Kompatibilität*: Heterogene Datenquellen (z.B. DOC, PDF, XLS) müssen, neben CSV- oder XML-Formaten in das System überführbar sein, um Datenaktualität zu gewährleisten. Umgekehrt sind Exporte als standardisierte Datenquellen notwendig. Wichtige Einträge sind dabei Kundenstammdaten, Artikelstammdaten, Auftragsdaten, Aktivitätenhistorie sowie Kontaktpersonen.
- *Echtzeitfähigkeit*: Die Daten müssen überall und jederzeit sofort verfügbar sein.
- *Modifizierbarkeit*: Die Modifizierbarkeit oder Änderbarkeit beschreibt, mit welchem Aufwand das CRM-System “gecustomized” werden kann. Dieser Aspekt ist sehr stark von der eingesetzten Architektur des Systems abhängig.
- *Vermeidung von Medienbrüchen*: Ein Medienbruch entsteht zum Beispiel, wenn via Internet eingehende Leads (Interessenten) manuell in das CRM-System überführt werden müssen. Webseiten mit entsprechenden Kontaktformularen müssen über eine Schnittstelle mit dem System verbunden sein, damit hinterlassene Kontaktdaten automatisch im On Demand System gespeichert werden können.

Zudem gaben - in Anlehnung an eine Studie über die Auswahlkriterien von ASP aus dem Jahre 2003 - zwei Drittel der Befragten an, sicherheits- und servicerelevante Aspekte als sehr wichtig zu erachten:

## 5.1. ZIELSTELLUNG DER ANALYSE UND VERWENDETES ANWENDUNGSSZENARIO

---

Auswahlkriterien für ASPs bei Geschäftskunden	Prozentangabe für "sehr wichtig"
Sicherheit/Datenschutz	69,7
Service/Support	66,67
Verfügbarkeit	63,6
Service Level Agreements	60,6
Ökonomische Stabilität des Anbieters	54,5
Preis	42,4
Referenzen	18,2
Andere	3,0

**Tabelle 5.1.:** Auswahlkriterien für ASPs bei Geschäftskunden [[KSvHB03](#), S. 51]

Deshalb ist die funktionale Eignung einer CRM-Software nur eine Seite der Medaille. Da die Kunden- und Unternehmensdaten in die Hand eines externen Dienstleisters gegeben werden, ist bei der Wahl des CRM On Demand Anbieters zunächst dessen Seriösität und Professionalität zu evaluieren [Koc07, S. 43]. Als ein anderes wesentliches Entscheidungskriterium ist die klar strukturierte Beschreibung des Services unabdingbar. Dabei können die Service Level Agreements helfen, in denen die Bereitstellung, die Sicherheit und die gesetzlichen Auflagen geregelt werden. Daher werden die Referenzen und das präsenste Auftreten samt Marktdurchdringung der einzelnen Anbieter mit ihrem Know-how und ihren langjährigen Erfahrungen als Grundlage für den Auswahlprozess herangezogen.

### 5.1.3. Vorgehensweise bei der Analyse

Die Märkte für CRM On Demand Software entwickeln sich sehr dynamisch. Glaubt man Analysten von Gartner, werden mehr als eine Milliarde US-Dollar und damit zwölf Prozent des prognostizierten Gesamtumsatzes von 7,4 Milliarden US-Dollar auf CRM On Demand Lösungen entfallen [HR07a]. Dabei werden bis 2011 11, 4 Milliarden US-Dollar erwartet, wobei der Markt für CRM

On Demand Lösungen doppelt so schnell wachsen wird, wie der herkömmliche CRM-Markt [HR07a]. Dementsprechend umfangreich gestaltet sich das derzeitige Angebot an CRM On Demand Lösungen. Daher soll folgende Tabelle (▷ vgl. Tabelle 5.3) einen aktuellen Überblick über das Angebot von CRM im On Demand Umfeld geben. Das Anbieterspektrum reicht dabei von kleinen und mittelständischen Unternehmen mit Speziallösungen bis zu den großen und bekannten Anbietern der CRM-Software.

Bei der Auswahl der zu untersuchenden Systeme wurde zunächst ein Anwendungsszenario (▷ vgl. Abschnitt 5.1.1) spezifiziert, anhand dessen die On Demand Lösungen auf den Umfang der Grundfunktionen hin, verglichen wurden. Dazu gehören Angaben über den Hersteller, eine Übersicht über die verwendeten Technologien und eine Einordnung der Anwendung hinsichtlich der möglichen Einsatzbereiche und Funktionalitäten. Zudem wurden Studien von Analysten für die funktionale Evaluierung ausgewertet sowie Websites der Softwareanbieter und Weblogs konsultiert. Unter Berücksichtigung eines Forrester Berichts über Hosted Sales Force Automation aus dem Jahr 2005 [Her05, S. 7] wurden dann vier CRM On Demand Anbieter ausgewählt (Salesforce.com, RightNow Technologies, Siebel und Netsuite) und anhand deren User-Frontend und Backend klassifiziert und bewertet.

Eine zusätzliche Restriktion stellte das Angebot an Informationsmaterial und einer Demo- oder Testversion dar. Dabei zeigte sich, dass der von Netsuite bereitgestellte Testzugang kein Admin-Backend hatte, um entsprechend des festgelegten Szenarios, eine Parametrisierung durchführen zu können. Des-

## 5.1. ZIELSTELLUNG DER ANALYSE UND VERWENDETES ANWENDUNGSSZENARIO




---

Anbieter	Produkt	URL
Salesforce	Salesforcesfa	<a href="http://www.salesforce.com">http://www.salesforce.com</a>
RightNow	RightNow 8	<a href="http://www.rightnow.com">http://www.rightnow.com</a>
Oracle	Siebel CRM On Demand	<a href="http://www.crmondemand.de">http://www.crmondemand.de</a>
SAP	SAP CRM On Demand	<a href="http://www.sap.com/crmondemand/index.epx">http://www.sap.com/crmondemand/index.epx</a>
Netsuite	NetSuite CRM+	<a href="http://www.netsuite.com">http://www.netsuite.com</a>
Sage	SageCRM.com	<a href="https://de.sagecrm.com/">https://de.sagecrm.com/</a>
Microsoft	Dynamics CRM Live	<a href="http://www.crmlive.com">http://www.crmlive.com</a>
CAS	GenesisWorld OnDemand	<a href="http://www.cas.de">http://www.cas.de</a>
CRM-Now	CRM-Now/PS und PG	<a href="http://www.crm-now.de">http://www.crm-now.de</a>
Etelos	Etelos CRM	<a href="http://www.eteloscrm.com">http://www.eteloscrm.com</a>
24 SevenOffice	24SevenOffice CRM	<a href="http://www.24sevenoffice.com">http://www.24sevenoffice.com</a>
SugarCRM	Sugar On-Demand	<a href="http://www.sugarcrm.com">http://www.sugarcrm.com</a>
Different Solutions	Vtiger CRM Hosting	<a href="http://www.different-solutions.de">http://www.different-solutions.de</a>
myfactory.com	Team.ON!	<a href="http://www.myfactory.com/">http://www.myfactory.com/</a>
Wice	WICE OnDemand	<a href="http://www.wice.de/">http://www.wice.de/</a>
Kana	KANA OnDemand	<a href="http://www.kana.com/services.php?tid=4">http://www.kana.com/services.php?tid=4</a>
Entellium	eSalesforce	<a href="http://www.entellium.com/">http://www.entellium.com/</a>
Salescenter	SalesCenter 3-D	<a href="http://www.salescenter.com/">http://www.salescenter.com/</a>

**Tabelle 5.3.:** Anbieter von CRM On Demand

halb konnte das für die Arbeit definierte Anwendungsszenario und in weiterer Folge der beispielhafte Workflow der Lead-Distribution innerhalb der Netsuite-Instanz nicht deployed werden. Aus diesem Grund wurden die CRM-Services von Netsuite für die Evaluierung nicht weiter herangezogen. Folgende Tabelle soll nun einen Übersicht über die ausgewählten CRM On Demand Lösungen geben.

## 5.1. ZIELSTELLUNG DER ANALYSE UND VERWENDETES ANWENDUNGSSZENARIO

			
Hauptsitz	Bozeman, USA	San Francisco, USA	San Mateo, USA
Release	7.5.5	Summer Edition 06	11
Preis	transaktionsabhängig	ab € 61,25	ab € 55,-
Hosting Architektur	Multi Tenant	Multi Tenant	Multi od. Single Tenant
Softwarebetrieb	USA und für europäische Kunden in London	USA	Vor der Übernahme durch Oracle waren IBM und T-Systems Hostingdienstleister
Referenzen	Lufthansa Cargo, British Telecom, Cisco Systems, British Airways, Toshiba,	Motorola, Merill Lynch, Tim Warner, Dell Swisscom,	Caterpillar, T-Systems, Nasdaq, Baxter, Bayer,
Branchenlösung	branchenspezifisch: Media, Automotive, Kreditwesen, Telekommunikation,	branchenübergreifend	branchenspezifisch: Automotive, Telekommunikation, Produktion, Finanzen,
Mehrsprachenfähigkeit	19 Sprachen	11 Sprachen	5 Sprachen
Mehrwährungsfähigkeit	vollständig	vollständig	vollständig
Stärkstes Modul	Servicemodul	Salesmodul	Reportingmodul

**Tabelle 5.4.:** Untersuchungsgegenstand

In weiterer Folge wurde an den ausgewählten Systemen untersucht, wie die Menüstruktur zur Funktionsauswahl, die Datenmasken, Tabellen und Dialogfelder verändert werden können. Anhand verschiedener Funktionen zur Bearbeitung von Stamm- und Bewegungsdaten wurden auch die Aspekte wie das

Look and Feel, die Navigation und Orientierung, die Bedienung und Personalisierung der Funktionen sowie die Darstellung der Daten evaluiert. Darüber hinaus wurden die vorhandenen Möglichkeiten hinsichtlich der Softwareintegration - insbesondere SAP R/3 und Groupware Lösungen wie MS Outlook sowie Lotus Notes - begutachtet. Schließlich erfolgte eine Betrachtung des Systembetriebs und des Datenmanagements einschließlich der SLAs und der Leistungsbeschreibung.

Der Zeitraum der Erhebungsmethode samt deren zugrundeliegenden Kriterien erstreckte sich von September 2006 bis November 2006. Anschließend erhielten alle Anbieter Einsicht in die Bewertungsergebnisse und konnten dazu auch Stellung beziehen.

## 5.2. Untersuchungsgegenstand

### 5.2.1. RightNow Technologies

Das in Bozeman (Montana/USA) ansässige Unternehmen RightNow Technologies hat sich auf die Bereitstellung von Anwendungsfunktionalitäten für die Bereiche Service, Vertrieb und Marketing spezialisiert. Als Highlight des Serviceangebotes ist das integrierte Servicemodul zu erwähnen, das sich in unterschiedlichsten vertikalen Märkten bewährt hat [oRT07c]. So wird die On Demand Lösung nicht nur im Hightechsektor und in der Telekommunikation, sondern auch im Konsumgütersektor, im öffentlichen Dienst, im Reisesektor sowie im Finanzwesen eingesetzt [Sch07d, S. 3]. Von den ca. 1700 Kunden sind 50 % Großunternehmen (mit mindestens 1 Milliarde Dollar Umsatz), die über das Internet auf RightNow CRM On Demand zurückgreifen, und kombiniert mit dem SOA-Ansatz, an ihre vorhandenen Unternehmenssysteme ankoppeln. (vgl. [o.V08d]; [oRT08]). Für die Inbetriebnahme der kundenspezifischen Instanz erfolgt lt. Auskunft von RightNow Technologies eine einmalige Einrichtungsgebühr. Der lfd. Betrieb wird folglich verbrauchsorientiert abgerechnet. Die CRM-Lösung kann für 19 Sprachen parametrisiert werden. Dazu zählen neben Deutsch auch Chinesisch (vereinfachtes, traditionelles Kantonesisch sowie traditionelles Mandarin), Tschechisch, Dänisch, Niederländisch, Englisch, Finnisch, Französisch, Italienisch, Japanisch, Koreanisch, Norwegisch, Polnisch, brasilianisches Portugiesisch, Spanisch und Schwedisch [oRT07a].

#### **Funktionsumfang:**

Für den definierten Testzeitraum wurde von RightNow eine CRM-Suite als Service bereitgestellt, bei der sich - bis auf das Nichtvorhandensein einer Leadmanagementkonsole - der Funktionsumfang erwartungsgemäß gestaltet [oRT07b]. Mit Hilfe eines Workarounds konnte auch die Problematik der fehlenden Funktionalitäten für das Leadmanagement umgangen werden, in dem die automa-

<b>Factsheet</b>	<b>RightNow Technologies</b>
RightNow CRM Suite - Version	Release: 7.5.5
Bereitstellung für Evaluierung	Okt. - Nov. 2006
Zielgruppe für CRM-System	Das Angebot gilt für Firmen jeder Größe
Kunden insgesamt	1700 Kunden
Anzahl der Nutzer (insgesamt)	keine Angaben

**Tabelle 5.5.:** Kennzahlen zu RightNow Technologies

tische Weiterleitung von Kundenkontakten an die dafür relevanten Verkaufsteams im Opportunitymanagement als Statusinformation "New Lead" parametrisiert wurde. Somit können auch über Webformulare erfasste Kunden und Interessenten in Echtzeit analysiert und den passenden Zielsegmenten zugeordnet werden. Die Auswertung relevanter Informationen über durchgeführte Kampagnen stützt eine ausgerichtete Kommunikation mit den erfassten Leads. Darüber hinaus beinhaltet die Weboberfläche eine interaktive Parametrisierung, in der z.B. Informationen wie Tabellen, Diagramme und Felder für Auswertungen zusammengestellt werden können. Neben den vorkonfigurierten Reports können aktuelle und prognostizierte Daten erfasst, analysiert und mit historischen Daten verglichen werden, um aktuelle Einblicke in die Vertriebspipeline, zur Neukundenquote sowie zu anderen Leistungskriterien zu erhalten. Diese können in gängigen Formate wie CSV, XML und PDF exportiert werden. Somit kann die zentrale Vertriebsorganisation den Erfolg ihrer dezentral organisierten Vertriebseinheiten analysieren und ermitteln, welche Geschäfte abgeschlossen wurden oder gefährdet sind.

### **Architektur:**

Das Hosting der On Demand Lösung basiert auf der Grundlage einer Multi-Tenant Architektur, die wiederum auf der Basis eines LAMP-Stacks (Linux, MySQL, Apache, PHP und andere Open Source Komponenten<sup>1</sup>) beruht. Eine Einteilung der Funktionen in die Verwendungsbereiche Marketing, Vertrieb und Service, in Verbindung mit einer Windows Forms ähnlichen Smart Client-GUI auf Basis einer SOAP-Architektur, lassen die Oberfläche sehr übersichtlich erscheinen. Bei dieser Vorgehensweise wird mit Hilfe der Smart Client Technologie von Microsoft das Backend über Webservices mit den GUI-Bibliotheken gekoppelt, in denen die benötigten Assemblies dynamisch vom Remotewebsserver downgeloadet werden. Zur Bewahrung der vereinbarten Performance und

<sup>1</sup>vgl. [http://opensource.rightnow.com/licenses\\_8x.php](http://opensource.rightnow.com/licenses_8x.php)

Verfügbarkeit wird die zugrundeliegende MySQL-Datenbank samt den kundenspezifischen DB-Schemas, über die DRBD-Komponenten (Distributed Replicated Block Device) nach Bedarf mit den replizierende Datenbank-Server erweitert.

### **Customizing:**

Ohne jegliche Programmierkenntnisse, können über die Administration Felder, Registerkarten (Tabs) und eingebette Analysefunktionen verschoben oder umbenannt werden. Obwohl das Datenmodell keine neuen Datenobjekte zulässt, bietet die Datenbankstruktur ausreichend vorbelegte Felder, welche individuell eingesetzt und verknüpft werden können. Zusätzlich zur Definition der Felder können auch Zugriffsrechte beim Customizing der Datenstrukturen definiert werden. Über Einstellmöglichkeiten in den Feldeigenschaften und über zugängliche PHP-Skripte sowie CSS-Dateien ist die Anpassung der Konsolen möglich. Über die bereitgestellten PHP-Skripte kann in Abhängigkeit von definierten Eigenschaften eine Individualisierung von Maskenobjekte vorgenommen werden. Weiters umfasst die Administration vordefinierte und rollenbasierte Profile, die sich für die gewünschten Einsatzbereiche entsprechend differenzieren lassen.

### **Integration:**

Auf der Grundlage der Designprinzipien einer serviceorientierten Architektur (SOA) wird eine nahtlose Kopplung der RightNow-Daten und -Funktionalitäten mit anderen Systemen und Anwendungen gewährleistet. Dabei wird durch das RightNow Connect-Framework die Datenintegration in Echtzeit über eine XML-API und die Applikationsintegration über Iframes (Custom Tabs) ermöglicht. In weiterer Folge ist eine Verbindung zu anderen vorkonfigurierten Anwendungen, durch individuelle oder definierten Konnektoren, möglich [oRT07a].

### **Workflow:**

Durch die integrierte Workflow-Komponente von RightNow Technologies können Aufgaben und Aktionen strukturiert definiert und anhand festgelegter Regeln und Ablauffolgen automatisch ausgelöst werden. Dabei können jedem Objekt beliebig viele Aktionen und Bedingungen durch if-then-else-Sequenzen zugewiesen werden, welche Vertretungshierarchien ebenfalls unterstützen. Darüber hinaus verfügt die Workflow-Komponente über graphische Unterstützungselemente für die Workflow-Definitionen und Eskalationsfeatures, die auf jedem Workflowschritt aufsetzen können.



### **Datenmanagement:**

Für die Durchführung von Importprozessen verfügt RightNow über ein integriertes Importmodul. Damit kann jeder durchgeführte Import auch wieder rückgängig gemacht werden.

### **SLA und Verfügbarkeit:**

Eine 99,5 prozentige Verfügbarkeit wird durch die Nutzungsbedingungen garantiert. RightNow betreibt drei Rechenzentren in Santa Clara (USA), Secaucus (USA), London (Europa) und eine Backup-Zentrale in Bozeman (USA), wobei die Datenhaltung entsprechend den Safe Harbor Principles erfolgt. Wartungsarbeiten werden vorzeitig angekündigt und auf einer extra dafür eingerichteten Website<sup>2</sup> publiziert. Zusätzlich können Supportleistungen in unterschiedlichen Paketen - wie Basic Support (werktags), Preferred Support (24 x7) und Premier Support (24 x 7) - für die Unterstützung des laufenden Betriebs bezogen werden.

### **Updatezyklen:**

Updates in regelmäßigen Abständen sind bei RightNow nicht explizit erkennbar. Die Veröffentlichung eines Minor Releases ist jedoch einmal jährlich und die Veröffentlichung eines Major Releases alle zwei Jahren vorgesehen.

### **Stärken:**

- **Marktposition:** RightNow Technologies ist ein bereits seit 1997 ein etablierter Anbieter von Service-, Vertriebs- und Marketinglösungen [oRT07b].
- **Verfügbarkeit:** Die Verfügbarkeit beträgt 99,5 %<sup>3</sup>.
- **Spezifische Branchenlösungen:** Die vorkonfigurierten Branchen-Vertriebsmethodiken der RightNow Vertriebsautomatisierung werden als On Demand Lösung für Medienunternehmer (Media Sales Edition), für die Automobile Branche (Automotive Dealer Edition), für Finanzdienstleister (Commercial Lending Edition) und Telekommunikationsdienstleister (Telecom Services Edition) angeboten.
- **Offene Architektur:** Die technische Architektur basiert auf offenen Standards und nutzt keine proprietären Technologien. Zusätzlich kooperiert RightNow auch mit der WS-I (Web Services Interoperability Organization), um die Interoperabilität von Webservices über alle Platt-

---

<sup>2</sup>vgl. <http://sitestatus.rightnowtech.com/cgi-bin/maintenance.py>

<sup>3</sup>vgl. <http://www.rightnowtech.de/support/hostgraph.html>

formen, Betriebssysteme und Programmiersprachen hinweg zu fördern [o.V09].

- **Viele Referenzimplementierungen:** Es existieren viele Referenzimplementierungen in kleinen bis großen Unternehmen.

### **Schwächen:**

- **Schwieriges Customizing:** Die zugrundeliegende Konfigurationsplattform ermöglicht keine intuitive Parametrisierung. Die Inanspruchnahme von Zusatzleistungen des Anbieters ist daher unabdingbar.
- **Intransparente Preisstrukturen:** Die Preisgestaltung orientiert sich, neben einer Bereitstellungsgebühr, an einem/dem Übertragungsvolumen.

### **5.2.2. Salesforce.com**

Das Unternehmen Salesforce.com wurde 1999 gegründet und gilt im On Demand Segment mit seinen Services für das Customer Relationship Management als Marktführer [KNS05, S. 4]. “Sie vergleichen sich gerne mit mehr verbraucherorientierten E-Business-Unternehmen wie eBay, Amazon.com und Google, deren Online Services von Endverbrauchern häufig in Anspruch genommen werden und bei jenen als sichere und solide Services gelten” [Mül06a]. Das Unternehmen sieht sowohl kleine als auch große Firmen unterschiedlichster Branchen als Zielgruppe. Dabei bietet Salesforce.com seinen Kunden eine zentral zugängliche Software in Form von mehreren Editionen<sup>4</sup> zur Sales Force Automation (SFA) und zur weiteren Support-bezogenen Funktionalitäten im Rahmen des Kundenbeziehungsmanagements als Service an. Zudem strukturiert Salesforce.com sein Programm in den Bereichen Entwicklung und Beratung. In der Entwicklung werden von Partnern, aber auch von Kunden selbst bereitgestellte Module und Anwendungen in die Multi-Tenant Infrastruktur integriert, um das Nutzungsspektrum der CRM-Suite zu erweitern. Im Beratungsbereich greift auch ein Netzwerk von Partner, über die bei Bedarf Implementierungsberatungen abgewickelt werden können.

### **Funktionsumfang:**

Der Zugriff auf Salesforce.com erfolgt über das Salesforce.com Portal, welches als Informations- und Mietplattform unter <https://login.salesforce.com/> zur Verfügung steht. Dabei werden unterschiedliche Produktausführungen angeboten:

- Unlimited Edition

---

<sup>4</sup>Der Service wird als Unlimited Edition, Enterprise Edition, Professional Edition, Group Edition mit Google AdWords und Salesforce Personal Edition angeboten.

<b>Factsheet</b>	<b>Salesforce.com</b>
Salesforce.com - Version	Release: Summer Edition 06
Bereitstellung für Evaluierung	Okt. - Nov. 2006
Zielgruppe für CRM-System	Das Angebot gilt für Firmen jeder Größe
Kunden insgesamt	41.000 Kunden
Anzahl der Nutzer (insgesamt)	1.000.000 Abonnenten

**Tabelle 5.6.:** Kennzahlen zu Salesforce.com [o.V08a]

- Enterprise Edition
- Professional Edition
- Group Edition
- Salesforce Personal Edition und
- Developer Edition

Für die Evaluierung wurde die Summer Edition 06, welche den 20. Release darstellt, in der Enterprise Edition bereitgestellt. Der Funktionsumfang der Trial Version unterstützt die zu erwartenden Aspekte: Kontakte, Leads, Opportunities, Kampagnen- und Pipelinemanagement. Diese Funktionen sind durch die webbasierte Infrastruktur, über mehrere Abteilungen und Standorte hinweg, nutzbar und ermöglichen durch die Online Leaderfassung, auf der Basis benutzerdefinierter Geschäftsregeln, einen optimierten Lead-Fluss für Vertrieb und Marketing. Dabei können doppelt erfasste Leads mit vorhandenen Leads, Kontakten und Kunden zusammengeführt werden und auch manuell für andere Anwender freigegeben werden. Zusätzlich wird durch eine Kampagnen- bzw. Lead-Quellenverfolgung ein globaler Überblick bzgl. der regionalen Leaderfassung möglich. Das Leadmanagement unterstützt auch die Definition von benutzerdefinierten Prozessen für die Etablierung konsistenter Methoden. Im Rahmen der CRM-Suite können auch analytische Szenarien für Lead- und Kampagnenberichte erstellt und in benutzerdefinierten Dashboards dargestellt werden.

### **Architektur:**

Die Multi-Tenat Architektur ist Java-basiert und bietet per Webservices Zugriff auf eine zentrale Oracle Datenbank. Laut Bither bestand die Datenbankarchitektur 2006 bei Salesforce.com aus einem Cluster (Rechnerverbund) von

vier Servern aus dem Hause Sun-Microsystems mit einer Oracle Datenbank [Bit06a]. Die Abbildung: 5.1 zeigt die Architektur von Salesforce.com. Der



**Abbildung 5.1.:** Multi-Tenant Plattform von Salesforce.com [o.V08b]

Apex-Builder dient dabei zur einfachen Erstellung von individualisierten Masken, Datenfeldern etc. So gesehen, belegen die individuellen Implementierungen der einzelnen Unternehmen virtuelle Partitionen, die das selbe Daten-, Sicherheitsmodell und die gleiche Oberfläche nutzen. “Auf diesen Partitionen werden die Metadaten gespeichert, die für jedes Unternehmen Geschäftsregeln, genutzte Felder, benutzerdefinierte Objekte und Schnittstellen zu anderen Systemen festlegen” [oS07b]. Neben den Metadaten für die virtuellen Instanzen enthalten die virtuellen Partitionen auch den mit der Programmiersprache Apex<sup>5</sup> entwickelten, benutzerdefinierten Code [oS07b]. Dadurch wird sichergestellt, dass sich mögliche Probleme, die durch den Code eines Unternehmens entstehen, nicht auf andere Kunden übertragen und dass sich Codefehler eines Objekts nicht auf andere Bereiche der gleichen Kundenanwendung auswirken [oS07b]. Die Apex-Code-Sprache ist sehr stark an Java angelehnt, welche durch eine integrierte Abfragesprache, auf Basis der Sforce Object Query Language, den Zugriff auf einzelne Objekte oder Objektmengen enthält und dadurch Manipulationsmöglichkeiten für die im Salesforce.com-Account gespeicherten Datensätze mittels INSERT, UPDATE, DELETE ermöglicht [HM07, S. 241 f.]. Zudem können Methoden im Apex-Code anderen Systemen als öffentliches API zur Verfügung gestellt werden oder im AppExchange Directory<sup>6</sup> von Salesforce.com - eine offene und communitygetriebene Plattform auf der Basis einer CRM-Suite - möglichen Benutzern zugänglich gemacht werden [HM07, S. 11]. Interessant erscheint das Konzept von Apex dahingehend, dass auf diese Weise auch verschiedene Drittanbieter ihre Module, bereits in Salesforce.com integriert, bereitstellen können. So können verschiedenartige Anwendungen über AppExchange in das Gesamtsystem von Salesforce.com integriert werden.

### Customizing:

Bereits im Standard kann die Struktur des Datenmodells sehr flexibel und vielfältig angepasst werden. Dabei ist das Anpassen von Seitenlayouts, die

<sup>5</sup>vgl. <http://www.salesforce.com/us/developer/docs/apexcode/index.htm>

<sup>6</sup>vgl. <http://www.salesforce.com/appexchange/>

Erstellung und Konfiguration neuer Datenfelder - die dann auch befüllt, berechnet oder befragt werden können - und die Anlage von Berichten und Dashboards möglich. Salesforce.com unterscheidet hierbei zwischen dem "Anpassen" von standardmäßig zur Verfügung gestellten Objekten und dem "Aufbauen" von neuen, benutzerdefinierten Objekten, Anwendungen und Custom S-Controls [HM07, S. 17]. Dabei stehen verschiedene Grundelemente zur Verfügung [HM07, S. 76 ff.]:

- **Custom Object** - Ein benutzerdefiniertes Objekt erlaubt die Definition eigener Datenbehälter, ähnlich einer Tabelle in einer relationalen Datenbank.
- **Tabs** - Tabs dienen zur Aufnahme und zur grafischen Repräsentation von benutzerdefinierten Objekten.
- **Dokumentenbibliothek** - In der Dokumentenbibliothek werden Textfragmente, Bilder und Informationen für die CRM-Suite verwaltet.
- **S-Control** - Ein S-Control ermöglicht die Definition von wieder verwendbaren Fragmenten einer Anwendung. Auch hier kann der Anwendung mehr Funktionalität, durch benutzerspezifische Quellcodes, hinzugefügt werden.
- **Modul** - im Modul, welches z.B. die vertriebsrelevanten Funktionalitäten beherbergt, werden Benutzerobjekte, Dokumente, Tabs und S-Controls zusammengefasst und unter einem Namen präsentiert.

Die Parametrisierung erfolgt im Wesentlichen durch "Drag and Drop"-Funktionen. Um aber die Limitierungen eines reinen webzentrierten Customizings zu umgehen, wird zusätzlich ein Apex Toolkit<sup>7</sup> für Eclipse angeboten, das einen Schema-Explorer (oder auch Object Browser) und eine Editiermöglichkeit für die S-Controls zur Verfügung stellt.

### **Integration:**

Der direkte API-Zugriff erfolgt über die bereitgestellten WSDL-Files. Die Salesforce WSDL kann leicht über *Setup* ▷ *Integrate* ▷ *Apex API* gefunden werden und erlaubt individuelle Nutzungsmöglichkeiten. Dabei hat man die Auswahl zwischen einer Unternehmens-WSDL, Partner-WSDL und Apex-WSDL. Die Partner-WSDL ist für Kunden, Partner und ISVs gedacht, die eine Integration erstellen möchten, welche unabhängig von den jeweiligen benutzerdefinierten Objekten und Feldern über mehrere Salesforce-Organisationen hinweg funktionieren soll. Die Apex-WSDL ist für Unternehmen gedacht, die Apex in einer anderen Umgebung ausführen oder kompilieren möchten. Darüber hinaus

---

<sup>7</sup><http://wiki.apexdevnet.com/index.php/Tools>

werden Desktopkonnektoren für die Integration in Microsoft Outlook, Lotus Notes, Microsoft Excel und Microsoft Word sowie systemeigene Konnektoren für die Integration in Oracle- und SAP-Systeme angeboten [oS08].

### **Workflow:**

Salesforce.com bietet Trigger zur Automatisierung von Prozessen an, die in Abhängigkeit der definierten Workflowregeln ausgelöst werden, sobald bestimmte Geschäftsaktionen ( z.B. die Erfassung und Speicherung von Kundendaten) durchgeführt wurden [KNS05, S. 8]. Beim erfolgreichen Auslösen einer Workflow-Regel lassen sich benutzerdefinierte Benachrichtigungen, Aufgaben sowie Feldaktualisierungen automatisieren. So können eingehende Leads dem richtigen Vertriebsmitarbeiter im Status "Web-to-Lead" zugewiesen werden. Bis auf das Fehlen einer grafischen Workflow-Modellierung, werden alle für das Anwendungsszenario benötigten Funktionalitäten zur Verfügung gestellt.

### **Datenmanagement:**

Die Konsolidierung von Datenbeständen aus verschiedenen Quellen wird, nach Salesforce.com, durch den Sforce Data Loader (SDL) ermöglicht. Der SDL beinhaltet einen Wizard, der den Anwender dabei unterstützt den Ladevorgang zu starten und eine Rückmeldung über Erfolg bzw. Misserfolg liefert [HM07, S. 65]. Zudem lassen sich die auf Salesforce.com in Echtzeit gespeicherten Daten als CSV-Datei für Sicherungs- oder Speicherzwecke exportieren. Dabei ist der Kunde für die Einhaltung von handels- und steuerrechtlichen Aufbewahrungsfristen verantwortlich.

### **SLA und Verfügbarkeit:**

Die Software Salesforce.com, die man über das Internet als webbasierten Service nutzen kann, wird in einem Rechenzentrum betrieben, welches nach dem Auditing Standard SAS 70 Type II<sup>8</sup> zertifiziert ist. Die, nach einer Transaktion im Rechenzentrum von Salesforce.com, gespeicherten Daten, werden automatisch gesichert (stündlich, täglich, wöchentlich) und entsprechend regulatorischen Maßnahmen - wie das Safe Harbor Abkommen und das TRUSTe Privacy Program (▷ vgl. Kapitel 2.3.2 über sicherheitsrelevante Anforderungen an den Anbieter) - abgelegt. Für die Hosting-Infrastruktur und die ein- und ausgehende Kommunikation des Rechenzentrums, wird eine Verfügbarkeit von 24 Stunden am Tag, und das 7 Tage die Woche, 365 Tage im Jahr angeboten, jedoch nicht zugesichert. Es finden sich also keine Zusagen hinsichtlich einer garantierten Verfügbarkeit Stattdessen werden Verfügbarkeitshinweise und frühzeitige Ankündigungen überWartungsaktivitäten auf einer

---

<sup>8</sup>vgl. <http://www.salesforce.com/platform/service-delivery/security/>

Webseite (▷ <http://trust.salesforce.com/>) publiziert. Dabei beträgt die regelmäßige Systemverfügbarkeit im Monatsmittel 99,9 %.

### Updatezyklen:

Salesforce.com wird regelmäßig gewartet und in der Regel alle drei Monate upgedatet. Darunter fallen z.B. die Durchführung notwendiger Anpassungen aufgrund rechtlicher Änderungen bzw. die Vornahme kleinerer Programmverbesserungen und Erweiterungen innerhalb des aktuellen Wartungsstandes.

### Stärken:

- **Marktführerposition:** Von der Hostinginfrastruktur über Datenbank- und Integrationsdienste bis hin zur Entwicklungsumgebung und zum Marktplatz für Applikationen bietet Salesforce.com derzeit die funktional umfassendste Plattform für Vertrieb, Service & Support, Marketing, Analysen und Partner Relationship Management an [Göl08].
- **Große Community:** Neukunden profitieren von den mehr als 40.000 anderen Unternehmen, die in die umfassende Sammlung bewährter Vorgehensweisen und praktischer Onlinere Ressourcen eingebracht wurden. Auf der eigens für diesen Zweck eingerichteten Seite (<http://success.salesforce.com/>) tauscht die Community ihre Erfahrungen aus; Interessenten erhalten Hilfe bei der Auswahl der richtigen Optionen und bestehende Kunden werden zur Übernahme der bewährten Vorgehensweisen ermutigt.
- **Sehr breite und tiefe Funktionalität:** Der On Demand Service von Salesforce.com inkludiert integrierte Anwendungen für die Vertriebs- und Marketingautomatisierung sowie für Service & Support [oS07a]. Darüber hinaus erweitert die AppExchange Plattform mit über 800 Anwendungen, das Servicespektrum von Salesforce.com über die CRM-Grenzen hinweg. Allerdings müssen diese Bausteine separat erworben werden.
- **Gut durchdachte Systemplattform:** Salesforce.com bietet durch die Multi-Tenant Architektur eine optimale Grundlage für die Individualisierung und Optionierung der integrierten Services. Die Kopplung an unternehmenseigene Applikationen ist leicht umzusetzen, da auf die Einhaltung offener Standards, wie Webservices über SOAP, geachtet wurde. Schließlich unterstützt die herstellerunabhängige Integrationsebene die Integrationsmethoden auf Benutzerschnittstellen-, Daten-, Funktionen- und Prozessebene.
- **AppExchange als Servicemarktplatz:** Durch die Vielzahl der Nutzer der Plattform und den daraus resultierenden Netzwerkeffekten, hat



Salesforce.com darüber hinaus eine hohe Anziehungskraft für alle möglichen Softwarefirmen und ISVs (Independent Software Vendors), die ihre eigenen Services auf der Grundlage von Salesforce.com bereitstellen. Branchenspezifische Services oder individualisierbare Services für Spezialprobleme sind damit für den Nutzer “on demand” erreichbar.

- **Enge Integration von operativem und analytischem CRM:** Die Funktionen für das analytische und operative CRM sind durch benutzerdefinierte Berichte und Dashboards eng integriert und können durch ETL-Services erweitert werden.
- **Transparente Servicebereitstellung:** Unter [▷ http://trust.salesforce.com](http://trust.salesforce.com) sind Statusinformationen über die Hostingplattform abrufbar. Dadurch lässt sich die inhaltliche Transparenz erhöhen.

### Schwächen:

- **keine individualisierbaren SLAs:** Neben dem Fehlen von Zusagen hinsichtlich der garantierten Verfügbarkeit, können auch keine individuell abgestimmten SLAs vereinbart werden.
- **Spezifische Branchenlösungen:** Die vorkonfigurierten<sup>9</sup> Lösungen zur Salesforce Automation werden standardisiert und branchenunabhängig angeboten.
- **Corporate Design:** Ein Branding der CRM On Demand Lösung von Salesforce.com ist nur begrenzt umsetzbar.

### 5.2.3. Siebel CRM On Demand

Das 1993 gegründete Unternehmen Siebel Systems galt bis zur 2006 stattgefundenen Übernahme durch Oracle, als der führende Anbieter von CRM-Standardsoftware und weltweiter Marktführer mit mehr als 3500 Kunden und 2 Mio Nutzern [Gei06, S. 168]. Als Hauptprodukte galten die eBusiness Applications, welche Funktionalitäten für das CRM, Partner Relationship Management (PRM), Employee Relationship Management (ERM) sowie das 2004 gestartete CRM On Demand Programm, umfasste [Gei06, S. 168]. Bis zur Übernahme durch Oracle zeigte sich IBM für den Betrieb des CRM On Demand von Siebel verantwortlich. Inzwischen wird die Lösung im Rechenzentrum von Oracle betrieben. Mit über 20.000 laufenden Servern, 4 Petabyte Festplattenspeichern und einer Fläche von über 7.600 Quadratmetern, gilt das Rechenzentrum von Oracle in Austin als eines der sichersten [oOC09b].

---

<sup>9</sup>Dementsprechend umfangreich ist der Customizingrahmen für die Individualisierung von Salesforce.com.



<b>Factsheet</b>	<b>Siebel CRM On Demand</b>
Siebel CRM On Demand - Version	Release: 11
Bereitstellung für Evaluierung	Okt - Nov. 2006
Zielgruppe für CRM-System	KMUs
Kunden insgesamt	3500
Anzahl der Nutzer (insgesamt)	2.000.000 Abonenten

**Tabelle 5.7.:** Kennzahlen zu Siebel CRM On Demand

### **Funktionsumfang:**

Der Funktionsumfang gestaltet sich mit branchenspezifischen Modulen als vielseitig. Neben branchenspezifischen Vorlagen der wichtigsten Datenfelder, sind auch branchenspezifische Datenobjekte und Registerkarten vorhanden. Dabei erfolgt eine Einteilung in Basisfunktionen für Vertrieb, Service, Marketing und Reportingfeatures. Neben den zu erwartenden Funktionen für das Lead-, Opportunity-, und Kampagnenmanagement, um potenzielle Interessenten einzuordnen und zu Verwalten sowie die Effektivität von Kampagnen zu kontrollieren, steht auch eine grafische Engine für Ad-hoc-Analysen und ein gehostetes Data Warehouse zur Verfügung [oOC06, S. 17]. Hierfür beinhaltete der von Oracle bereitgestellte Testzugang für Siebel CRM On Demand 35 vordefinierte Berichte mit 250 Einzelanalysen, welche mit Hilfe interaktiver Dashboards eine Visualisierung der geschäftliche Informationen samt historischen Trends in Echtzeit ermöglicht. Dabei können die, für die einzelnen User entsprechend ihrer Funktion personalisierten Dashboards tieferegehende Analysen anstoßen, um kritische Kenngrößen zu finden. Um nicht permanent auf das Internet angewiesen zu sein, erlaubt ein Offline-Client das Bearbeiten und Erstellen von Datensätzen, welche im Onlinemodus mit Siebel CRM On Demand synchronisiert werden können. Über die persönlichen Einstellungen kann jeder User aus unterschiedlichen Sprachen wählen und auch sonstige individuelle Anpassungen an Layouts, Berichten oder Registerkarten vornehmen. Siebel CRM On Demand unterstützt Deutsch, Französisch, Spanisch, Italienisch, Portugiesisch, Japanisch, Koreanisch und Chinesisch [o.V06a, S. 6 f.]. Zudem wird eine vollständige Unterstützung aller Währungen (über 160) geboten. Auch Grundfunktionen wie ein Kalender, eine Mail-Integration für Outlook und Lotus Notes stehen zur Verfügung und ermöglichen eine bidirektionale Synchronisierung von Kontaktinformationen, Terminen und Aufgaben, wobei bei dem bereitgestellten Testaccount keine Mail-Integration vorgesehen war.

### Architektur:

Seit dem Wechsel des Siebel CRM On Demand Hostings bei IBM zu Oracle's Technologie Stack, wird neben der typischen Multi-Tenant Bereitstellung auch eine "Single Tenant Enterprise Edition"<sup>10</sup> angeboten [Bea08]. Diese basiert auf der Grundlage der Oracle Fusion Middleware, welche neben einer einheitlichen Infrastruktur für Grid Computing, mit service- und eventorientierten Designprinzipien (SOA und Event-Driven Architektur) eine dezidierte Bereitstellung der Siebel Instanz ermöglicht. Dabei ermöglichen die, der Oracle Fusion Middleware, zugrundeliegenden Technologien, auf Basis des Oracle Enterprise Managers, eine dezidierte Virtualisierung der javabasierten Service- und Utilityinfrastruktur, um die Update- und Upgradefähigkeit der einzelnen Instanzen gewährleisten zu können.

### Customizing:

Durch die integrierte Bereitstellung von Funktionen für die Modifizierung von Prozessen, Daten und die Benutzerschnittstellen, wird die Möglichkeit geboten die Dateneingabe zu vereinfachen und verschiedene Layouts für unterschiedliche Aufgaben und Rollen, durch anpassbare Registerkarten und Objekte, zu integrieren [o.V08c]. So können beliebig viele Felder vom Typ Integer, String, Prozent, Datum/Uhrzeit, Währung, URL, Telefon, E-Mail und Auswahlliste für eigene Kundenfelder und Berichte erstellt werden. Zur Aufwertung des standardisierten Siebel CRM On Demand können daher über die eingebetteten Anpassungsfunktionen folgende Individualisierungsschritte vorgenommen werden:

- Umbenennung der Registerkarten und Objekte innerhalb der mandantenspezifischen Instanz.
- Übergabe von Daten an externe Systeme durch die Erstellung von Hyperlinks.
- Definition des Rollenmanagements respektive der Datenzugriffsrechte, in Abhängigkeit von Benutzerberechtigungen und Bildschirm-Layouts.
- Die Ausführung von benutzerdefinierten Analysen und interaktiven Übersichten anhand angepasster Felder.

Darüber hinaus kann der Anwender Personalisierungsfunktionen nutzen, wie z.B. die Reihenfolgen und Sortierung von Listenansichten.

---

<sup>10</sup>vgl. [http://www.oracle.com/corporate/press/2008\\_feb/scod-privatehosting.html](http://www.oracle.com/corporate/press/2008_feb/scod-privatehosting.html)

### **Integration:**

Die Siebel CRM On Demand Lösung, die bereits auf der Oracle Fusion Middleware Technologie aufsetzt, bietet anhand Anbieterinformationen eine State-of-the-Art Architektur, hinsichtlich SOAP-basierten Webservices, für eine anwendungsübergreifende Bereitstellung von End-to-End-Geschäftsprozessen [o.V08c]. Inwieweit die Kopplung eines Produktkonfigurators möglich ist, konnte aber nicht ermittelt werden, da der bereitgestellte Testzugang keine Webservice Schnittstelle beinhaltete.

### **Workflow:**

Die Abbildung des definierten Prozessablaufs für die Lead Qualifizierung und Lead Weiterleitung konnte - unabhängig von dem nicht vorhandenen Möglichkeiten für die Integration des Produktkonfigurators - entsprechend umgesetzt werden. Auf der Grundlage von konfigurierbaren Geschäftsregeln wurden die Kriterien "Produktinteresse" und "Region" für die Neuordnung von Interessenten implementiert. Zusätzlich wurde die Option zum Akzeptieren oder Zurückweisen bei der Interessentenweiterleitung mit berücksichtigt, um eine problemlose Neuordnung gewährleisten zu können. Im Zuge dessen wurde eine Konvertierungsfunktion entsprechend parametrisiert, zugeordnete Interessenten anhand deren Lead Status in Opportunities zu konvertieren, was daraus resultiert, dass man die Verkaufschancen bewerten kann. Dazu kommt ein eigener, anpassbarer Salesprozess der, abgestimmt auf die betreffende Phase des Verkaufsgeschehens, mit Hilfe von Anweisungen und Aufgabenvorlagen direkt in den Vertriebszyklus einbettet werden konnte.

### **Datenmanagement:**

Mit einem integrierten Wizard können CSV-Dateien importiert und auch exportiert werden. Damit Adressdubletten bei der Neuerfassung oder beim Datenimport erkannt werden, greift Siebel CRM On Demand auf eine integrierte Dublettenkontrolle zurück [Sch07c, S. 10]. Die Dublettenprüfung kann zwischen der Importdatei und dem vorhandenen Adressenbestand nach individuellen Kriterien durchgeführt werden [Sch07c, S. 26].

### **SLA und Verfügbarkeit:**

Die Nutzungsbedingungen vermeiden die Angabe konkreter Zahlen, es wird jedoch formuliert, dass eine Verfügbarkeit entsprechend des allgemein anerkannten Industriestandards angestrebt wird. Für supportspezifische Anforderungen werden zwei Supportpakete angeboten:

- Standard Customer Care: Telefon Support 12 x 5

- Gold Customer Care: Telefon Support 24 x 7 und eine Reaktionszeit innerhalb von 4 Stunden

### Updatezyklen:

Der Hersteller wirbt mit etwa zehn Updates innerhalb von zwei Jahren, was auf einen Zyklus von zwei bis drei Monate schließen lässt.

### Stärken:

- **Marktführerposition:** Oracle ist mit dem Siebel Portfolio der weltweite Marktführer für CRM-Standardsoftware. Die CRM- und Analysefunktionen, welche als Service über das Internet als Siebel CRM On Demand zugänglich sind, profitieren auch von der technologischen Marktführerschaft Oracles im Kontext von Unternehmensanwendungen.
- **Sehr breite und tiefe Funktionalität:** Siebel CRM On Demand bietet integriert in ein kompaktes Webinterface, eine umfassende Sammlung von Funktionalitäten zum Automatisieren, Vereinfachen und Verwalten von vertriebs- und kundenrelevanten Informationen.
- **Enge Integration von operativem und analytischem CRM:** Die bereitgestellten Anwendungsfunktionalitäten für analytische und operative CRM-Szenarien beinhalten umfangreiche und interaktive Reports.
- **Starker Branchenfokus:** Die branchenspezifischen Siebel CRM On Demand-Editionen<sup>11</sup> zeichnen sich durch sofort einsetzbare Prozesse, Datenmodelle, Objektmodelle und Geschäftslogiken aus, die auf die Anforderungen der Automobil-, Versicherungs-, Hochtechnologie-, Finanz- und Medizin-Branchen zugeschnitten sind [oOC09a].
- **Nutzungsgebühren:** Siebel CRM On Demand kann ab € 55,- pro Anwender und Monat genutzt werden.

### Schwächen:

- **Customizingrahmen:** Unternehmensspezifische Anpassungen sind aufgrund rudimentärer Anpassungsmöglichkeiten schwer umsetzbar. Felder, welche im initialen Setup als Required definiert sind, können nicht verändert werden, bzw. können benutzerdefinierte Felder nicht mehr gelöscht werden. Zusätzlich gibt es auch Einschränkungen hinsichtlich der möglichen Feldtypen, wie z.B. berechenbare Felder oder lookup Felder. Außerdem erschweren die fehlenden Drag & Drop-Funktionalitäten den Customizingprozess.

---

<sup>11</sup>vgl. <http://crmondemand.oracle.com/de/products/industrysolutions/index.htm>

- **SLA:** In den Nutzungsbedingungen werden keine konkreten Angaben hinsichtlich der Verfügbarkeit gemacht. Nach Rücksprache mit Oracle Deutschland werden jedoch 99,5 % Verfügbarkeit zugesichert.
- **Browserkompatibilität:** Die Weboberfläche von Siebel CRM On Demand ist nur unter IE ab 5.5 und FF nutzbar.
- **Vertragslaufzeit:** Der On Demand Vertrag ist nur jährlich kündbar.

### 5.3. Zusammenfassung der Analyse

Als Ergebnis der Analyse in Bezug auf die Erwartungen und den betrachteten Kriterien lässt sich folgendes festhalten:

- Im Rahmen eines typischen CRM-Szenarios bieten die als webbasierte Anwendungsservice zugänglichen Produktausführungen die, für den in Kapitel 5.1.1 beschriebenen Anwendungsfall, notwendigen Basisfunktionalitäten. Dabei ist der Anwendungsservice von Salesforce.com, RightNow Technologies und Siebel CRM On Demand aus der jeweiligen Sicht der Anbieter mandantenfähig. Zudem bietet Oracle, seit der Übernahme von Siebel, auch eine dezidierte Bereitstellung von Funktionen für CRM, wenn eine tiefer eingehende Geschäftsprozessorientierung berücksichtigt werden soll bzw. die gesetzlichen Bestimmungen zu einer separaten Datenhaltung verpflichten.. Inwieweit sich die Softwarebereitstellung auf dezidierten Hardware-Plattformen durchsetzt bleibt aber abzuwarten, da die mehrmandantenfähige Lösungen geringere Wartungs- und Entwicklungskosten auf sich ziehen. Denn aus der Sicht einer mehrmandantenfähigen Hostingarchitektur (Multi-Tenant Architektur) kann auch ein instanzierter Service als eine datentechnisch und organisatorisch abgeschlossene Einheit, angeboten werden. Eine strenge Trennung der mandantenabhängige Daten und Objekte gewährleistet bei Salesforce.com und RightNow Technologies die Unabhängigkeit zwischen Versionen und Konfigurationen, mit dem Vorteil, dass die Kernfunktionen jederzeit erweitert und korrigiert werden können.
- Der Zugriff auf die jeweiligen CRM On Demand Lösungen erfolgt über einen Browser, welcher über eine sichere Verbindung auf Basis eines 128-Bit Schlüssels verfügt. Insbesondere die Ausprägung von Salesforce.com als Rich Internet Application (RIA), durch clientseitige Präsentations- und Kommunikationstechnologien wie AJAX und Flex, ermöglicht ein schnelles und intuitives Rich User Interface und somit eine kontextspezifische und bedienfertige GUI. Die Navigation und die Benutzeroberfläche von Siebel erfolgt ebenfalls auf der Grundlage von GUI-Komponenten

(JavaScript-Bibliotheken) mit Rich-Client-Funktionalitäten, die eine konsistente Menüführung ermöglichen, über die Hauptfunktionen schnell abgerufen und im Kontextbereich übersichtlich visualisiert werden können. Der Zugriff auf RightNow erfolgt auf der Grundlage der Smart Client-Patterns von Microsoft (Smart Client Software Factory), welche auf Windows Forms-Bibliotheken aufbaut. Hier wird eine an MS-Outlook angepasste GUI innerhalb einer clientseitigen Sandbox ausgeführt.

- Die für die Vertriebsaufgaben zugeschnittenen Funktionen werden unabhängig vom Branchenfokus bereits vorkonfiguriert und zugeschnitten auf standardisierte Geschäftsanforderungen bereitgestellt. Dadurch können die bereitgestellten Systeme nach Vertragsabschluss sofort in den produktiven Betrieb aufgenommen werden, wobei, sich die vertraglichen Vereinbarungen bezüglich der Haftung und Verfügbarkeit als eher unverbindlich gestalten. Besonders Salesforce.com garantiert weder eine bestimmte Verfügbarkeit noch eine Kompensationen für mögliche Ausfälle. Daher muss jeder Anwender für sich selbst entscheiden, ob ein SLA notwendig ist, denn Salesforce.com - mit über 1 Million Usern - bietet in der Regel eine Verfügbarkeit von 99,9 %. Anbieter wie RightNow Technologies und Oracle stellen zwar CRM On Demand zu standardisierten Konditionen bereit, bieten aber Möglichkeiten hinsichtlich einer Individualisierung der Service Level Agreements.
- Es kommt bei allen Anbietern das Modell des registrierten Anwenders (Names User Modell) zur Anwendung, welches beinhaltet, dass jeder Anwender einen eigenen personalisierten Zugang (Lizenz) benötigt. Die personalisierte Zugriffssteuerung erfolgt, entsprechend definierter Rollen, innerhalb eines Rollenmodells, das die Sichtbarkeiten der Unternehmensdaten steuert. Die Rollen werden zusätzlich mit Profilbeschreibungen hinsichtlich objekt- und feldrelevanten Zugriffsrechten ergänzt. Alle evaluierten Anbieter liefern in der Grundaufführung bereits Beispiel-Profile wie z.B. Administrator, Marketing und Vertrieb, welche einem Benutzer zugewiesen werden.
- Für den manuellen Datenaustausch werden CSV-Dateien für Excel verwendet. Abgesehen von RightNow Technologies, sorgen bei Salesforce.com und Siebel die integrierten Plug-ins für Office- und Groupwarelösungen (z.B. spezielle Schaltflächen in MS Office und Lotus Notes) für direkte Zugriffsmöglichkeiten. Zusätzlich können erstellte Aufgaben, Termine und Mails bidirektional synchronisiert werden.
- Weitere Unterschiede zwischen den einzelnen Lösungen sind im Bereich Workflowautomatisierung festzustellen, wobei RightNow Technologies als einziger Anbieter über eine grafische Workflow-Engine für die modell- und regelbasierte Anpassung der vertriebsrelevanten Geschäftsprozesse

bietet. Die integrierten Werkzeuge zur Workflowsteuerung innerhalb von Salesforce.com und Siebel CRM On Demand ermöglichen ebenfalls ein durchgängiges Bearbeitungsmanagement auf der Grundlage von festgelegten Prozessschritten und Regeleinstellungen. Damit sorgen die integrierten Mechanismen zur Steuerung von Geschäftsabläufen für klare und auf unterschiedlichen Phasen strukturierte Vertriebsprozesse.

- Bei Salesforce.com und RightNow Technologies erfolgt die Kopplung mit verschiedenen Daten- und Serviceprovider via HTTP und SOAP. Dabei stehen vor allem Nutzungsformen im Stil von Mapping-Mashups im Vordergrund, um z.B. CRM-Daten mit geografischen Daten für die interaktive Darstellung zu bündeln. Darüber hinaus werden bei Salesforce.com und RightNow Technologies systemeigene Konnektoren für die bidirektionale Synchronisierung zwischen den CRM-Kundendaten und ERP-Stammdaten angeboten. Über den AppExchange-Marktplatz von Salesforce.com gibt es zusätzlich vorgefertigte Integrationsappliances für Salesforce.com.
- Der Einsatz von Tools und Frameworks zur schnellen Integration mit ERP und anderen Back-end-Systemen erschließt sich dem Anwender nicht in jedem Fall ohne weitere Nachfrage bei Oracle. Gleichwohl basiert Siebel CRM On Demand auf State-of-the-Art Ansätzen, die zumindest unter Einbeziehung von Oracle, eine Integration mit dem IT-Backend des Nutzers ermöglichen. Da aber die bereitgestellte Trial-Version keine Webservice-Schnittstellen berücksichtigt hatte, konnten die Anforderung hinsichtlich der Integration mit einer Drittapplikation nicht erfüllt werden.
- Die evaluierten Systeme können über die integrierten Werkzeuge für die Systemgestaltung (Customizingrahmen) in unterschiedlichen Ausprägungen, entsprechend des beschriebenen Anwendungsfalls und den daraus resultierenden Anforderungen angepasst werden. Ausgehend von der Fragestellung “Wodurch unterscheiden sich die ausgewählten CRM On Demand Lösungen hinsichtlich des integrierten Customizingrahmens?” lässt sich in Anlehnung an die festgelegte Customizingtopologie (▷ vgl. Kapitel 4.5.3) folgendes Ergebnis festhalten:
  - Salesforce.com bietet ein hohes Maß an Customizing: Der Nutzer kann aufgrund des zugrundeliegenden Metadaten-Frameworks das Datenmodell, die Geschäftslogik sowie die Benutzeroberfläche als ein individuelles Deployment auf einer virtuellen Partition anpassen und erweitern. Darüber hinaus bietet Salesforce.com Webservices in Form von APIs, eine proprietäre Programmiersprache (Apex) mit wieder verwendbaren Prozesskomponenten und eine umfangreiche

Dokumentation sowie IDEs für alle wichtigen Entwicklungsumgebungen.

- RightNow Technologies bietet ein mittleres Maß an Customizing: Dem Nutzer ist es möglich Änderungen sowohl an den Objekten (Entitäten) und Attributen als auch im Erscheinungsbild der Anwendung vorzunehmen, wobei für tiefgreifende Änderungen ein programmatischer Eingriff auf Basis von PHP notwendig ist.
- Siebel CRM On Demand bietet ein mittleres Maß an Customizing: Dem On Demand Kunden kann Änderungen sowohl an der Struktur als auch im Erscheinungsbild über notwendige Parameter vornehmen. Zusätzlich ist es möglich in einem sinnvollen Rahmen weitere Parameter anzulegen. Im Unterschied zu RightNow Technologies können aber Prozesskomponenten nicht wiederverwendet werden, da kein programmatischer Zugriff innerhalb des Customizingrahmens durchgeführt werden kann. Individuelle Wünsche sind durch Zusatzleistungen des Anbieters realisierbar und außerdem sind weitere Module auf Anfrage zuschaltbar.

Prinzipiell ist aus technischer Sicht nichts gegen eine extern vorgehaltene CRM-Software, welche als SaaS im Rahmen eines On Demand Vertrags deklariert und genutzt wird, einzuwenden, solange die definierten Geschäftsziele unterstützt werden und die Datenschutzgarantien angeboten werden. Es gilt dafür immer einen entsprechenden Nachweis für die Sicherheit der Datenhaltung bei den SaaS-Anbietern einzuholen. Hierfür konnte von den evaluierten Anbietern in Erfahrung gebracht werden, dass laufende Security Audits<sup>12</sup> für die regelmäßige Überprüfung der Sicherheitsstandards herangezogen werden. Bei der Übernahme von Gewährleistungen, hinsichtlich der Verfügbarkeit und den Disaster-Recovery-Prozeduren, besteht aufgrund fehlenden formellen Service-Level-Vereinbarungen jedoch noch Handlungsbedarf, um auch risikoscheue Kunden von den CRM On Demand Möglichkeiten überzeugen zu können. Je “anonymer” das Vertragsverhältnis ausfällt, desto unverbindlicher gestalten sich die vertraglichen Vereinbarungen zu Fragen der Haftung und Verfügbarkeit, wobei mit der üblichen - zu 99,5 Prozent garantierten - Verfügbarkeit [DW07] die Evaluierungsphase ohne Serviceausfall stattgefunden hat.

Darüber hinaus haben die untersuchten Anbieter und Lösungen verschiedene Zielsetzungen, insbesondere in Bezug auf ihre Zielgruppen. So bietet die zugrundeliegende Multi-Tenant Architektur von Salesforce.com ein Metadaten getriebenes Customizingmodell, welches eine permanente Erweiterung und Anpassung der servicebasierten CRM-Lösung ermöglicht. Zusätzlich wird durch

---

<sup>12</sup>SysTrust SAS 70 Type II gilt als Bescheinigung für unternehmensinterne Kontrollen hinsichtlich Sicherheit und Hochverfügbarkeit.



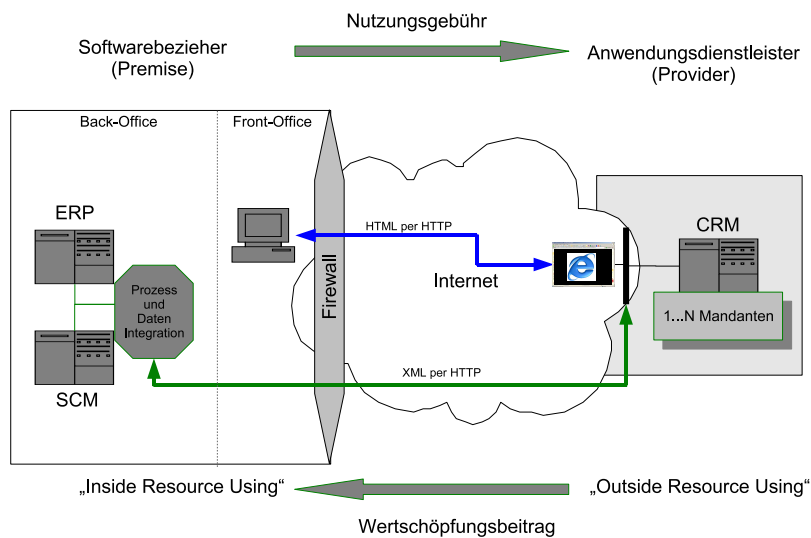
das Apex-Skript, eine stark an Java angelehnte prozedurale Programmiersprache, eine integrierte Abfrage- und Manipulationsmöglichkeit für die CRM-Funktionalitäten geboten, deren Methoden wiederum als öffentliche API zur Verfügung gestellt werden. In weiterer Folge lassen sich - durch die Verwendung der API - Anwendungen bei Salesforce.com integrieren, die sich durch den Nutzer vollständig innerhalb der Salesforce-Arbeitsumgebung bedienen lassen [vG08, S. 37]. Durch den Netzwerkeffekt ist die Attraktivität dieser inzwischen als “AppExchange” bezeichneten Servicebibliothek so hoch, dass zahlreiche Drittanbieter sich auf individualisierbare Lösungen für diese Umgebung fokussieren, was wiederum unterschiedlichste Zielgruppen anspricht. So wundert es nicht, dass zunehmend auch Unternehmensanwendungen, die mit der eigentlichen Idee eines netzbasierten CRM-Systems nicht mehr viel zu tun haben, den Weg als Service auf diese Plattform finden.

Die aus der Multi-Tenant Hosting Architektur abgeleitete SmartHosting Umgebung von RightNow bietet deutliche Vorteile für Großunternehmen. Kunden können frei entscheiden, wann sie zu einer neueren Version wechseln möchten, in dem Upgrades auf einer integrierten Staging-Umgebung zunächst als kundenspezifische Pilotinstanz genutzt werden können [oRT06]. Zusätzlich werden die kundenspezifischen Daten in dezidierten Datastores gespeichert. Oracle adressiert mit Siebel CRM On Demand neben kleinen und mittleren Unternehmen auch Konzerne. Vor allem expandierende Unternehmen, die in ihren Niederlassungen schnell auf zentral gesteuerte Prozesse zugreifen müssen.

## 6. Fazit und Ausblick

Zur anfänglichen Fragestellung hinsichtlich der Identifizierung des Paradigmas Software-as-a-Service als Grundlage für CRM On Demand, lässt sich folgende Aussage treffen:

Bei Software-as-a-Service wird eine beim Softwarehersteller betriebene CRM-Anwendung in Form eines Services über das Internet zur On Demand Nutzung bereitgestellt, wobei der Anwender lediglich einen Service Client, in der Regel einen Web-Browser, benötigt.



**Abbildung 6.1.:** CRM im SaaS-Modell

Diese Alternative zum klassischen Lizenzmodell unterliegt den Voraussetzungen und Rahmenbedingungen einer auf dem Grundgerüst des ASP-Modells etablierten Sourcing-Initiative, die mit den assoziierten Technologien des Service Oriented Computings (SOC) die Anwendungsfunktionalitäten einer extern vorgehaltenen CRM-Implementierung bereitstellt. Die CRM-Software wird dabei direkt vom Hersteller oder von einem Dienstleister selbst betreut und gewartet, und auf Basis von Webtechnologien schnell und unkompliziert geliefert. Diese übernehmen die Verantwortung für den Betrieb der Software, Hardware,

---

des Rechenzentrums sowie der Infrastruktur und garantieren durch die Einhaltung von Service Levels eine entsprechende Systemperformance und Systemverfügbarkeit. Dabei ist zu berücksichtigen, dass die erreichbare Qualität der Servicebestandteile - wie Sicherheit, Flexibilität und Skalierbarkeit - erheblich höher ist als die einer eigenrealisierten Lösung. Zur Deckung des starken Bedarfs für die Kollaboration innerhalb von Unternehmensnetzwerken, ermöglichen die zentral zugänglichen Anwendungsfunktionalitäten digitale, serviceorientierte und globale Wertschöpfungsnetze. Die einzige Voraussetzung für solch eine effiziente Steuerungsmöglichkeit ist eine leistungsfähige, globale Infrastruktur auf IP-Basis. Dabei orientiert sich die Preisgestaltung, mit Blick auf Flexibilität und Transparenz für die leistungsbeziehenden Unternehmen, stets an der Nutzung, also an einer entsprechenden Pauschale pro Nutzer im Unternehmen. Daher ist SaaS ein Kernbestandteil von CRM On Demand in Bezug auf Flexibilisierung der Standardfunktionalitäten bei gleichzeitiger Sicherung des Investitionsschutzes, da die betriebliche Handlungsweise hinsichtlich der Aspekte Globalisierung und Kernkompetenz flexibler gestaltet wird.

Je erfolgskritischer das zu nutzende System aber für das Unternehmen ist, umso abwegiger scheint die Überlegung, sich auf die Anwendungs- und Substitutionspotentiale von Software-Services aus dem Netz einzulassen. Ein weiteres Hindernis stellen oft die kundeninternen Richtlinien bzgl. Datensicherheit und Datennutzung dar. Zur Erreichung eines wertsteigenden Beitrags gelten daher die Aspekte Sicherheit und Qualität, im Hinblick auf die Kontrolle über die Daten sowie die Abhängigkeiten von den Netzzugänglichkeiten und vom Betreiber als wichtige Indikatoren. Während aber in den meisten Unternehmen die Serververfügbarkeit bei 99 % liegt, können die SaaS-Anbieter mit einer Verfügbarkeit von 99,9 % aufwarten [DW07]. Außerdem ist beim SaaS-Ansatz die Datensicherheit durch regulatorische Maßnahmen, wie Safe Harbor und kontinuierliche Backups, sichergestellt. Dennoch sind die Hypothesen bezüglich der Unzugänglichkeiten der Datensicherheit gegenüber einer On Premise Software, im Rahmen dieser Arbeit, nicht vollständig zu verwerfen. Jedenfalls steht fest, dass die am Markt angebotenen Lösungen schnell einsatzfähig sind. Ein Unternehmen, das den Einsatz einer CRM-Software mit der Zielstellung verfolgt Wettbewerbsnachteile zu vermeiden statt sie herauszuarbeiten, kann daher signifikant von On Demand Prinzipien profitieren. Ausschlaggebend dafür sind beispielsweise niedrige Initialkosten und die Möglichkeit das System bei unerfüllten Erwartungen nicht weiter zu verwenden.

Die untersuchten Lösungen haben auch gezeigt, dass bei einem expliziten On Demand Angebot wie bei Salesforce.com der Fokus auf eine bedarfsgerechte Auswahl und Anpassung der bereitgestellten Funktionalitäten gelegt wird. Dementsprechend ist durch das Customizing über die Metadaten die datentechnisch und organisatorisch abgeschlossene Einheit eines Mandanten anpassbar. Es handelt sich hierbei also um die metadatenbasierte Individualisierung

---

einer zentralen Softwareinstanz, die durch jeden Mandanten virtuell strukturiert wird und daher beim Mandanten A anders aussieht als bei Mandanten B. Dem gegenüber führt das "ready-to-run" Konzept von SaaS und On Demand aber zu einer zunehmend kritischen Betrachtung der umfangreichen Customizingmaßnahmen, da diese als zu zeit- und kostenaufwendig wahrgenommen werden. Daher wird von Anbieterseite weitgehend versucht den gesamten Ablauf des reinen webzentrierten Customizings durch unterstützende Wizards und Paramterisierungsoptionen, stark zu vereinfachen. Denn je weniger Aufwand für das Customizing erforderlich ist, desto geringer sind die damit verbundenen Kosten für den Kunden, was sich in dessen Zufriedenheit mit der Software und einem hohen Return on Investment (ROI) niederschlägt.

Die Arbeit hat gezeigt, dass das in der Praxis anzutreffende Schlagwort SaaS mit zwei Varianten von Bereitstellungsarchitekturen in Verbindung gebracht wird:

- Multitenancy - Ausgehend von der Tatsache, dass die operativen, analytischen und kollaborativen Anwendungsfunktionalitäten weitgehend als Standard zur Verfügung stehen und die zugehörigen Geschäftsprozesse im Bereich der Vertriebsunterstützung dadurch stärker standardisiert sind, teilen sich bei diesem Modell alle Anwender eine einzige, gemeinsame Infrastruktur und Codebasis. Es ist deshalb auf Seiten der Anbieter von entscheidender Bedeutung, ein granulares Serviceangebot anzubieten, um die Individualisierung und Optionierung standardisierter Software-Services zu ermöglichen.
- Isolated tenancy - Hier wird jedem Kunden eine eigene System- und Datenbankinstanz zugeordnet jedoch im Vergleich zu einer Single-Tenant Architektur mit den Unterschied, dass alle Kunden (Tenants) mit der gleichen Softwareversion arbeiten, sprich die gleiche Codebasis verwenden. Man verbindet auf diesem Level die Effizienz einer traditionellen Mehrmandantenarchitektur mit der isolierten Bereitstellung eines Einmandantensystems. Die Umsetzung der kundeneigenen Anforderungen in diesem Isolated Tenancy Modell erfolgt in Anlehnung an einen Customizing-Rahmen, in dem bestimmte Anpassungen hinsichtlich Layout, Struktur und Felder durchgeführt werden können.

Als Randbedingungen des Software-as-a-Service-Pardigmas lassen sich daher folgende Aspekte für CRM On Demand herausstellen:

- Der Ansatz eine Software, wie ein CRM-System, nicht selbst zu betreiben, sondern einem Dienstleister, respektive den Softwareanbietern selbst diese Aufgabe zu überlassen und via Internet darauf zuzugreifen ist unabhängig von der Bereitstellungsarchitektur möglich.

- 
- Das metadatengetriebene Customizingmodell zur schnellen und selbstständigen Anpassung an neue Anforderungen wird nur bei Salesforce angeboten. Gleichwohl basieren die Lösungen von RightNow Technologies und Siebel auf diesen Ansätzen und ermöglichen zumindest unter Einbeziehung der Entwickler, eine Individualisierung der Applikation.
  - Auch wenn von den Möglichkeiten, die sich durch die Verwendung von Webservice-APIs ergeben, ausgegangen wird, besteht trotzdem die Gefahr von Insellösungen, insbesondere wenn sich verschiedene Systeme zwar auf Applikations- und Prozessebene integrieren lassen, aber nicht auf der Benutzerebene.
  - Wichtige Punkte in den SLAs (wie Verfügbarkeit, Antwortzeit, usw.) finden sich in den Nutzungsbedingungen für die untersuchten On Demand Angebote nur ansatzweise wieder.
  - Probleme des Datenschutzes und der Rechtssicherheit lassen sich bei der grenzüberschreitenden Nutzung trotz regulatorischer Maßnahmen nicht immer eindeutig klären.
  - Kundenzufriedenheit ist im On Demand Umfeld sehr wichtig, da die Lock-In-Effekte geringer sind als bei On Premise Lösungen.
  - Für den SaaS-Markt charakteristisch ist auch, dass ein hohes Maß an Vertrauen von Seiten der Anwender erforderlich ist. Die Nachfrage an Software-Services ist im Vergleich zum angloamerikanischen Raum aber geringfügig.
  - Potentielle Kostenvorteile durch das On Demand Nutzungsmodell sind langfristig nicht in den Vordergrund zu stellen, wohl aber die geringe Kapitalbindung, die eine Flexibilität hinsichtlich der Art und der Dimension des Einsatzes ermöglicht. Entscheidend für den Nutzen ist der Umfang der tatsächlichen Integration der Systeme in die betrieblichen Prozesse.

Die Entwicklung der SaaS-Branche ist auch für CRM-Systeme immer noch im Wachstum begriffen. Es ist zu erwarten, dass durch die etablierte technologische Basis eine vielfältigere Nutzungsweise möglich wird. Insbesondere durch die Integration von Java und Flash auf mobilen Devices und durch die Etablierung von Technologien für die Offline-Nutzung, wird die Breite möglicher Anwender zunehmen.

# Literaturverzeichnis

- [A<sup>+</sup>07] C. A. Albayrak *et al.*, “IT-Outsourcing im Kontext global tätiger Unternehmen,” *HMD - Praxis Wirtschaftsinformatik*, vol. 254, pp. 27–38, April 2007.
- [AM04] D. Abts and W. Müller, *Grundkurs Wirtschaftsinformatik - eine kompakte und praxisorientierte Einführung (Ausbildung und Studium)*. Vieweg, April 2004.
- [App02] J. Appleby, “CRM über ASP - Erfolgsrezept mit drei Buchstaben,” *IT Management*, no. 10, pp. 50–54, Oktober 2002. [Online]. Available: <http://www.it-verlag.de/htdocs/itm/index.html>
- [AS02] M. Amber and J. Schumacher, “CRM-Systeme und Basistechnologien,” in *M. Meyer (Hrsg.) CRM-Systeme mit EAI - Konzeption, Implementierung und Evaluation*. Braunschweig/Wiesbaden: Vieweg, Juni 2002, pp. 21–59.
- [AW04] M. Amberg and M. Wiener, “Formen des IT Offshoring,” Lehrstuhl für Betriebswirtschaftslehre, insbes. Wirtschaftsinformatik III, Universität Erlangen-Nürnberg, Arbeitspapier 4, 2004. [Online]. Available: [http://www.international-outsourcing.de/CSF-Tool/docs/AP4\\_FormenITOffshoring\\_2004\\_WienerMartin.pdf](http://www.international-outsourcing.de/CSF-Tool/docs/AP4_FormenITOffshoring_2004_WienerMartin.pdf)
- [B<sup>+</sup>06] J. Bizer *et al.*, “Ubiquitäres Computing und Informationelle Selbstbestimmung,” Institut für Wirtschaftsinformatik der Humboldt-Universität, Berlin, Technikfolgenabschätzung, Juni 2006. [Online]. Available: <http://amor.rz.hu-berlin.de/~spiekers/TAUCIS.pdf>
- [Bar05] G. Barnet, “The road to on-demand: a user’s guide,” Ovum, Tech. Rep., Juni 2005. [Online]. Available: [www-03.ibm.com/servers/uk/pdf/ondemand/the\\_road\\_to\\_on\\_demand\\_users\\_guide\\_2.pdf](http://www-03.ibm.com/servers/uk/pdf/ondemand/the_road_to_on_demand_users_guide_2.pdf)
- [Bau07] T. Baumgärtner, “Zeiterfassungssoftware mit Adobe Flex,” ITP Bakkalaureatsarbeit, Wirtschaftsuniversität Wien, 2007. [Online]. Available: [http://epub.wu-wien.ac.at/dyn/virlib/bakkWI/mediate/epub-wu-01\\_bf8.pdf?ID=epub-wu-01\\_bf8](http://epub.wu-wien.ac.at/dyn/virlib/bakkWI/mediate/epub-wu-01_bf8.pdf?ID=epub-wu-01_bf8)

- [BE01] R. Buck-Emden, *mySAP CRM*, ser. 1. Auflage. Galileo Press, Dezember 2001.
- [Bea08] B. Beal, “Oracle adds single-tenant option to CRM On Demand,” SearchCRM.com, Februar 2008. [Online]. Available: [http://searchcrm.techtargt.com/news/article/0,289142,sid11\\_gci1301294,00.html](http://searchcrm.techtargt.com/news/article/0,289142,sid11_gci1301294,00.html)
- [BEHb07] W. Brenner, N. Ebert, A. Hochstein, and F. Übernickel, “IT-Industrialisierung: Was ist das?” *Computerwoche*, no. 5, p. S. 5, April 2007.
- [Bei05] W. Beinhauer, “Die SOA-Vision liegt in weiter Ferne,” *Computerwoche*, Oktober 2005. [Online]. Available: <http://www.computerwoche.de/zone/soa/567938/>
- [Ber05] T. G. Berger, “Konzeption und Management von Service-Level-Agreements für IT-Dienstleistungen,” Ph.D. dissertation, TU Darmstadt, April 2005. [Online]. Available: [http://deposit.d-nb.de/cgi-bin/dokserv?idn=975446169&dok\\_var=d1&dok\\_ext=pdf&filename=975446169.pdf](http://deposit.d-nb.de/cgi-bin/dokserv?idn=975446169&dok_var=d1&dok_ext=pdf&filename=975446169.pdf)
- [Bit06a] B. Bither, “Conversation with Salesforce.com VP of Development Regarding Recent Outage,” Bill Bither’s Insights, Februar 2006. [Online]. Available: <http://www.atalsoft.com/cs/blogs/billbither/archive/2006/02/03/9650.aspx>
- [BIT06b] BITKOM, “Ausrichtung der IT bei BPO,” Arbeitskreis Deinstleistungs-Management und -Standards (AK DMS), Berlin, Leitfaden 6, September 2006. [Online]. Available: [http://www.bitkom.org/files/documents/BITKOM\\_Leitfaden\\_IT-Ausrichtung\\_fuer\\_BPO.pdf](http://www.bitkom.org/files/documents/BITKOM_Leitfaden_IT-Ausrichtung_fuer_BPO.pdf)
- [BIT06c] —, “Compliance in IT-Outsourcing-Projekten,” Arbeitskreis Outsourcing, Berlin, Leitfaden zur Umsetzung rechtlicher Rahmenbedingungen, Juli 2006. [Online]. Available: [http://www.bitkom.org/files/documents/BITKOM-Leitfaden\\_Compliance.pdf](http://www.bitkom.org/files/documents/BITKOM-Leitfaden_Compliance.pdf)
- [BIT06d] —, “Leitfaden zur Umsetzung von ASP Geschäftsmodellen,” Arbeitskreis Application Service Providing, Berlin, Leitfaden 1.0, März 2006. [Online]. Available: <http://www.bitkom.org/files/documents/ASP-Lizenzmodellev1-1.pdf>
- [BIT06e] —, “Virtualisierung - Überblick und Glossar,” Arbeitskreis Server- und Betriebskonzepte, Berlin, Leitfaden 1.0, Juli 2006.

- [Online]. Available: [http://www.bitkom.org/files/documents/BITKOM-Virtualisierungsglossar\\_2006\\_07\\_24\\_ST.pdf](http://www.bitkom.org/files/documents/BITKOM-Virtualisierungsglossar_2006_07_24_ST.pdf)
- [Blo05] N. Blozan, "Service-Oriented Architectures and Software as a Service," ebizQ, 2005. [Online]. Available: [http://www.ebizq.net/topics/biz\\_opt/features/5899.html](http://www.ebizq.net/topics/biz_opt/features/5899.html)
- [BMW07] O. Bohl, S. Manouchehri, and U. Winand, "Unternehmerische Wertschöpfung im Web 2.0," *HMD - Praxis Wirtschaftsinformatik*, vol. 255, pp. 27–36, Juni 2007.
- [Bos07] A. Bosch, "Ajax - Grundlagen und Funktionsweise," *HMD - Praxis Wirtschaftsinformatik*, vol. 255, pp. 37–48, Juni 2007.
- [Bre07] G. Breiter, "Utility Computing als integraler Bestandteil der serviceorientierten Architektur," in *Kircher, H. (Hrsg.) IT - Technologien, Lösungen und Innovationen*. Springer Berlin Heidelberg, März 2007, pp. 78–100.
- [BS00] C. Bange and H. Schinzer, "Data Warehouse und Business Intelligence - Grundlagen entscheidungsorientierter Informationssysteme," Competence-Site.de, 2000. [Online]. Available: [http://www.competence-site.de/bisysteme.nsf/BBE71828E145670BC125695400483F6B/\\$File/grundlagen%20-%20dw%20und%20bi-systeme.pdf](http://www.competence-site.de/bisysteme.nsf/BBE71828E145670BC125695400483F6B/$File/grundlagen%20-%20dw%20und%20bi-systeme.pdf)
- [C+06] F. Chong *et al.*, "Multi-Tenant Data Architecture," MSDN - Microsoft Developer Network, 2006. [Online]. Available: <http://msdn2.microsoft.com/en-us/library/aa479086.aspx>
- [CC06] F. Chong and G. Carraro, "Architecture Strategies for Catching the Long Tail," MSDN - Microsoft Developer Network, 2006. [Online]. Available: <http://msdn2.microsoft.com/en-us/library/aa479069.aspx>
- [Col01] K. Collins, "Analytical CRM: Driving Profitable Customer Relationships," Gartner Research, Stamford, Analysis, März 2001.
- [CS05] W. Craig and W. Simon, "Custom Entities and Fields in a Multi-Tenant Database System," Salesforce.com, INC., San Francisco, Patentscope PCT/US2005/010915, Oktober 2005. [Online]. Available: <http://www.wipo.int/pctdb/en/wo.jsp?wo=2005098593&IA=WO2005098593&DISPLAY=DESC>
- [DHAL06] R. P. Desisto, J. Holincheck, G. Alvarez, and B. J. Lheureux, "Predicts 2007: Software as a Service Provides a Viable Delivery Model," Gartner, Tech. Rep. G00144023, November 2006.



- [Die02] W. Dieckmann, “Die Strategische Option für Unternehmen der Energiewirtschaft,,” in *Köhler-Frost, W. (Hrsg.), Grundlagen des Application Service Providing*. Berlin: KS-Energy-Verlag, 2002, pp. 91–111.
- [Doy99] M. Doyle, “TCO Discovery Audit for XYZ Publishing Company,” MorganDoyle Limited, Whitepaper, Juni 1999. [Online]. Available: [http://www.morgandoyle.co.uk/white\\_papers/Total\\_Cost\\_of\\_Ownership.pdf](http://www.morgandoyle.co.uk/white_papers/Total_Cost_of_Ownership.pdf)
- [DW07] A. Dubey and D. Wagle, “Delivering Software as a Service,” The McKinsey Quarterly, Tech. Rep., Mai 2007. [Online]. Available: <http://www.mckinseyquarterly.com/PDFDownload.aspx?L2=4&L3=43&ar=2006>
- [E<sup>+</sup>06] T. Eymann *et al.*, “Ökonomische Bewertung der Dienstauswahl in Service-Netzen,” in *Barth, T., Schüll, A., (Hrsg.) Grid Computing. Konzepte, Technologien, Anwendungen*. Vieweg, 2006, pp. 33–66.
- [Ede07] W. Ederer, “Utility Computing - das Rechenzentrum als Unternehmen,” in *IT - Technologien, Lösungen und Innovationen*. Springer Berlin Heidelberg, März 2007, pp. 64–78.
- [Fac02] A. L. Factor, *Analyzing Application Service Providers*. Prentice Hall PTR, 2002.
- [Fei07] M. Feidicker, “Outsourcing: Maßgeschneiderte Services oder besser doch Lösungen von der Stange?” *Computerwelt*, no. 11, Juni 2007.
- [FO04] J. Frischmuth and C. Oecking, “Outsourcing als strategische Unternehmensentscheidung,” in *B. Kuhlin, H. Thielmann (Hrsg.), Real-Time Enterprise in der Praxis*. Berlin: Springer, 2004, pp. 331–344.
- [G<sup>+</sup>06] K. Green *et al.*, “Software as a Service update - spurring innovation, enhancing business value and enabling business processes,” Tripple Tree, Industry Analysis, September 2006. [Online]. Available: <http://www.triple-tree.com/research/technology.html>
- [Gab07] A. Gabriel, “Beim rechten Namen nennen,” e-commerce Magazin, Jänner 2007. [Online]. Available: <http://www.e-commerce-magazin.de/index.php3?page=01-07/outsourcing.html>

- [Gar07] U. Gartz, “Mit Weitblick & Augenmass,” E-Commerce Magazin Nr. 6, September 2007.
- [Gei06] M. Geib, *Kooperatives Customer Relationship Management in Finanzdienstleistungsnetzwerken*. Deutscher Universitätsverlag, April 2006.
- [GGA06] J. Gehland, B. Galbraith, and D. Almaer, *Ajax. Eine pragmatische Einführung in Web 2.0*. München: Hanser Fachbuchverlag, 2006.
- [GH03] T. Gurzki and H. Hinderer, “Eine Referenzarchitektur für Software zur Realisierung von Unternehmensportalen,” in *U.Reimer, A. Abecker, S. Staab, G. Stume (Hrsg.) WM 2003: Professionelles Wissensmanagement - Erfahrungen und Visionen, Beiträge der 2. Konferenz Professionelles Wissensmanagement*, ser. LNI, vol. 28. GI, April 2003, pp. 157–160. [Online]. Available: [http://www.gurzki.de/publications/wm2003/WM2003\\_Gurzki\\_Hinderer\\_Referenzarchitektur\\_Portalsoftware.pdf](http://www.gurzki.de/publications/wm2003/WM2003_Gurzki_Hinderer_Referenzarchitektur_Portalsoftware.pdf)
- [GK06] O. Günther and M. Klafft, “Internettechnologien im Kontext moderner Wertschöpfungsnetze - Beispielanwendungen am Point of Sale,” in *IT - Information Technology*, no. 48. Oldenbourg Wissenschaftsverlag, April 2006, pp. 193–199.
- [Göl08] A. Göldi, “Das nächste it-schlachtfeld,” netzwertig.com, Mai 2008. [Online]. Available: <http://netzwertig.com/2008/05/14/platform-as-a-service-das-naechste-it-schlachtfeld/>
- [Glä03] L. Glässer, *IT-Lösungen im E-Business*. Publicis Corporate Publishing, 2003.
- [Gro07] W. Grohmann, *Von der Software zum Service: Neue Formen der Software-Nutzung*. München: H. K. P. Consulting, Juni 2007.
- [GSH<sup>+</sup>07] C. J. Guo, W. Sun, Y. Huang, Z. H. Wang, and B. Gao, “A Framework for Native Multi-Tenancy Application Development and Management,” in *E-Commerce Technology and the 4th IEEE International Conference on Enterprise Computing, E-Commerce, and E-Services, 2007. CEC/EEE 2007. The 9th IEEE International Conference on*. IEEE Computer Society, July 2007, pp. 551–558.
- [Göt06] M. Götzfried, “Service-orientierte Anwendungsintegration im Intra- und Internet,” Ph.D. dissertation, Wirtschaftswissenschaftliche Fakultät, Universität Passau, 2006. [Online]. Available: <http://www.opus-bayern.de/uni-passau/volltexte/2006/79/>

- [Gun07] A. V. Gunten, “Warum sich Software as a Service durchsetzen wird,” *ondemandnotes.com*, August 2007. [Online]. Available: <http://www.ondemandnotes.com/>
- [H<sup>+</sup>04] F. Hoch *et al.*, “Whitepaper Software as a Service: changing the paradigm in the software industry,” *SIIA und TRIPLETREE*, August 2004. [Online]. Available: [www.triple-tree.com/research/technology/saas\\_aug04.pdf](http://www.triple-tree.com/research/technology/saas_aug04.pdf)
- [Hai07] P. Haitz, “Festgeklopft - Service Level Agreements bei Hosting-Unternehmen,” *iX - Magazin für professionelle Informationstechnik*, no. 4, pp. 112–119, April 2007.
- [Hal07] S. Hallek, “SaaS-Geschäftsmodelle im Web 2.0,” in *Grob, H.L., Vossen, G. (Hrsg.) Entwicklungen im web 2.0 aus technischer, ökonomischer und sozialer Sicht*, no. 51. Westfälische Wilhelms-Universität Münster: Institut für Wirtschaftsinformatik, Oktober 2007, pp. 73–83. [Online]. Available: <http://www.wi.uni-muenster.de/aw/download/hybride-systeme/Hybrid%2051.pdf>
- [Ham07] N. Hamlett, “IT Outsourcing Impacts on Enterprise Architecture,” *IT Professional*, vol. 9, no. 2, pp. 34–40, March-April 2007.
- [HB06] O. Hinz and M. Bernhardt, “Interaktive Preisfindung als zwischenbetriebliche Prozessintegration auf Basis von Webservices,” *Wirtschaftsinformatik*, vol. 48, no. 3, pp. 169–177, Juni 2006.
- [Hei68] L. J. Heinrich, *Gemeinsame Computerbenutzung in der Industrie*. Wiesbaden: Verlag Gabler, 1968.
- [Hel07] K. Helling, “Was Sie über SaaS wissen müssen,” *Computerwoche, SaaS-Perspektiven 19*, 2007.
- [Her01] D. Herzog, “Analytisches Customer Relationship Management mit Data Mining,” Deloitte & Touche, Präsentation, 2001. [Online]. Available: [http://www.competence-site.de/crm.nsf/AttachShow!OpenFrameset&attachfile=/crm.nsf/87F6AAAA66E8A509C1256B650043C66F/\\$File/Analytisches\\_CRM\\_Herzog.pdf](http://www.competence-site.de/crm.nsf/AttachShow!OpenFrameset&attachfile=/crm.nsf/87F6AAAA66E8A509C1256B650043C66F/$File/Analytisches_CRM_Herzog.pdf)
- [Her05] L. Herbert, “The Forrester Wave?: Hosted Sales Force Automation, q1 2005,” Forrester Research, Inc., Tech Choices, April 2005. [Online]. Available: [http://www.rightnow.com/pdf/Forrester\\_wave.pdf](http://www.rightnow.com/pdf/Forrester_wave.pdf)
- [HHW07] H. Hippner, O. Hoffmann, and K. D. Wilde, “CRM-Studie 2007,” Katholische Universität Eichstätt-Ingolstadt, Lehrstuhl

- für ABWL und Wirtschaftsinformatik, Tech. Rep., August 2007. [Online]. Available: [http://www.competence-site.de/crm.nsf/35FDCFC9F9041162C125733A0071764F/\\$File/crm-studie\\_2007\\_leseprobe.pdf](http://www.competence-site.de/crm.nsf/35FDCFC9F9041162C125733A0071764F/$File/crm-studie_2007_leseprobe.pdf)
- [HK06] P. G. Huppertz and M. Kresse, *IT Service Management Advanced Pocket Book. ITIL-Service - Kern des Ganzen*. Serview GmbH, 2006, vol. Bd. 6.
- [HLP07] L. Heuser, S. Lacher, and S. Perlmann, “Flexible Prozessgestaltung als Basis innovativer Geschäftsmodelle - Von der Service-Orientierten Architektur zur Vision des Business Webs,” *Wirtschaftsinformatik (1)*, pp. 19–28, 2007. [Online]. Available: <http://www.aifb.uni-karlsruhe.de/Forschungsgruppen/BIK/wi2007/papers/wi-2007-1-004.pdf>
- [HM07] A. Holubek and C. Metzger, *Salesforce.com Entwicklerhandbuch. On-Demand-Anwendungen mit Apex*. entwickler.press, 2007.
- [HN01] H. R. Hansen and G. Neumann, *Wirtschaftsinformatik*. Stuttgart: UTB, 2001.
- [Hol98] W. K. Holm, “Total Cost of Ownership: Kennzahlen oder Konzept?” in *Information Management*, no. 2, 1998, pp. 19–22.
- [Hot02] R. Hotter, “Möglichkeiten und Grenzen webbasiert-verteilter Wertschöpfung,” in *Picot, Arnold (Hrsg.) Web Services - Bausteine für das e-Business*. Münschen: Hüthig, Februar 2002, pp. 57–68.
- [Hou07] D. Houlding, “From SOA to SaaS,” Dr.Dobb’s Portal, März 2007. [Online]. Available: <http://www.ddj.com/dept/webservices/197700752>
- [HR07a] K. Hoffmann-Remy, “Gartner prognostiziert für CRM-Markt 14 Prozent Wachstum,” IT-BUSINESS.de, September 2007. [Online]. Available: <http://www.it-business.de/index.cfm?pid=2279&pk=92359>
- [HR07b] —, “Software as a Service mausert sich zum Hype,” IT-BUSINESS, April 2007. [Online]. Available: <http://www.it-business.de/index.cfm?pid=2431&pk=64235&fk=242#>
- [HSSSJ07] V. Hoyer, C. Schroth, K. Stanoevska-Slabeva, and T. Janner, “Web 2.0-Entwicklung - ewige Beta-Version,” *HMD - Praxis Wirtschaftsinformatik*, vol. 255, pp. 78–87, Juni 2007.

- [JA07] D. Jacobs and S. Aulbach, “Ruminations on Multi-Tenant Databases,” in *BTW Proceedings of Datenbanksysteme in Business, Technologie and Web*, ser. LNI, A. Kemper, H. Schöning, M. Jarke, T. Seidl, C. Quix, and C. Brochhaus, Eds., vol. 103, GI-Fachbereichs für Datenbanken und Informationssysteme. Aachen: GI, März 2007, pp. 514–521. [Online]. Available: <http://www.btw2007.de/vortraege/JacobsAulbach.pdf>
- [Jac05] I. Jackewitz, “Evolutionary Application Service Providing (eASP),” Ph.D. dissertation, Universität Hamburg, aug 2005.
- [Jac07] A. Jacobs, “Soa als Grundlage für Software as a Service? (SaaS),” SOA-Expertenrat, April 2007. [Online]. Available: <http://www.computerwoche.de/soa-expertenrat/?p=170>
- [JD07] H. Jouanne-Diedrich, “Die ephorie.de IT-Sourcing-Map. Eine Orientierungshilfe im stetig wachsenden Dschungel der Outsourcing-Konzepte,” ephorie.de - das Management Portal, Mai 2007. [Online]. Available: <http://www.ephorie.de/it-sourcing-map.htm>
- [Küc01] P. R. Küchler, “Vermarktungsstrategien für Application Service Providing: Ideen zum Verkauf? Auch wenn das Marktsegment boomt?” in *Köhler-Frost, W. (Hrsg.), Application service providing: die neue Herausforderung für Unternehmen*. Berlin: KS-Energy-Verlag, 2001, pp. 168–176.
- [Küh07] K. Kühnel, “Software-as-a-Service,” NTT Europe Online GmbH, Whitepaper, 2007.
- [KNS05] R. Korb, F. Naujoks, and W. Schwetz, “Crm-software-zertifizierung 2005 salesforce.com,” HEWSON GROUP, Naujoks & Collegen, schwetz consulting, Tech. Rep., 2005 2005. [Online]. Available: <http://www.schwetz.de/Zertifizierung/pdf/050916%20Salesforce%20Zertifizierung%202005.pdf>
- [Koc07] T. F. Koch, “Software as a Service (SaaS) im Unternehmen unter den Aspekten Datenschutz und Datensicherheit,” Abschlussarbeit Bachelor of Science, Fachhochschule Brandenburg, Brandenburg, Juni 2007. [Online]. Available: <http://www.webpunkt.net/uni/Abschlussarbeit.pdf>
- [Koh07a] D. Kohler, “SaaS? der schlanke Weg zur Business Software,” Netzwoche, 2007. [Online]. Available: [www.io-market.com/\\_pdf/io-market\\_Netzwoche33\\_DanielKohler.pdf](http://www.io-market.com/_pdf/io-market_Netzwoche33_DanielKohler.pdf)
- [Köh07b] P. T. Köhler, *ITIL - Das IT-Service-Management Framework*, ser. 2., Auflage. Springer-Verlag, Dezember 2007.

- [Krc05] H. Krcmar, *Informationsmanagement*, 4th ed. Berlin: Springer, 2005.
- [KSvHB03] T. Klostermann, T. Specht, C. van Husen, and G. Bischoff, "Application Service Providing: A Market Study of Application Service Providing in the European Market," Institute for Human Factors and Technology Management der Universität Stuttgart, Stuttgart, Marktstudie, 2003.
- [Kub07] A. Kubsch, "Software as a Service wird die Branche verändern," computerwoche.de, November 2007. [Online]. Available: [http://www.computerwoche.de/produkte\\_technik/software/1848725/index.html](http://www.computerwoche.de/produkte_technik/software/1848725/index.html)
- [Lan07] A. Langer, "Evaluierung von AJAX-basierten frameworks für das Web 2.0," Studienarbeit, Technische Universität Chemnitz, März 2007. [Online]. Available: [http://archiv.tu-chemnitz.de/pub/2007/0048/data/Studienarbeit\\_Ajax\\_AndreLanger.pdf](http://archiv.tu-chemnitz.de/pub/2007/0048/data/Studienarbeit_Ajax_AndreLanger.pdf)
- [Las05] A. Lassila, "Taking a service-oriented perspective on software business: how to move from product business to online service business," in *IADIS International Journal on WWW/Internet*, vol. 4, no. 1, Porto, Oktober 2005, pp. 70–82. [Online]. Available: [http://www.iadis.org/ijwi/files/vol4\\_issue1/6\\_Lassila.pdf](http://www.iadis.org/ijwi/files/vol4_issue1/6_Lassila.pdf)
- [LET06] R. Lemm, V. Erni, and O. Thees, "Effiziente Kommunikation, Koordination und Kooperation im Netzwerk holz dank IFIS - ein neues integriertes und webbasiertes Informationssystem," *Schweizerische Zeitschrift für Forstwesen*, vol. 157, no. 12, pp. 162–170, Mai 2006. [Online]. Available: [ftp://ftp.wsl.ch/downloads/waldnutzung/dokumente/proj001/szf\\_5\\_2006.pdf](ftp://ftp.wsl.ch/downloads/waldnutzung/dokumente/proj001/szf_5_2006.pdf)
- [Lie07] D. Liebhart, *SOA goes Real - serviceorientierte Architekturen erfolgreich planen und einführen*. Carl Hanser Verlag München, 2007.
- [Lix08] C. Lixenfeld, "SaaS: Lösungen aus der Leitung," Computerwoche.de, Februar 2008. [Online]. Available: [http://www.computerwoche.de/hp\\_cw\\_mittelstand/loesungen/1854899/index.html](http://www.computerwoche.de/hp_cw_mittelstand/loesungen/1854899/index.html)
- [LLJ03] P. Lassig, H.-J. Lamberti, and C. Jochum, "Scoring- und beidseitige Due-Diligence-Prozesse - im rahmen der Lieferantenauswahl beim Infrastruktur-Outsourcing," *Wirtschaftsinformatik*, vol. 45, no. 2, pp. 147–156, 2003. [Online]. Available: [http://www.wirtschaftsinformatik.de/wi\\_artikel.php?sid=1120](http://www.wirtschaftsinformatik.de/wi_artikel.php?sid=1120)

- [Mar04] W. Martin, "EAI 2004 - Das prozess-orientierte Unternehmen," *IT-Research.net*, Mai 2004.
- [Mas05] D. Masak, *Moderne Enterprise Architekturen*. Springer Berlin Heidelberg, 2005.
- [Mat05] B. Mathews, "Prüfung outgesourcter Informatikleistungen - Prozessoptimierung im IT-Security Bereich," Master's thesis, Universität Zürich, September 2005. [Online]. Available: [www.isaca.ch/files/Diplom\\_PruefungOutsourcingIT.pdf](http://www.isaca.ch/files/Diplom_PruefungOutsourcingIT.pdf)
- [MB07] C. Müller and M. Bereszewski, "Saas - sourcing-spielart mit potenzial," *Informationweek.de*, Februar 2007. [Online]. Available: <http://www.informationweek.de/services/showArticle.jhtml?articleID=197004458>
- [McK07] J. McKendrick, "Is SOA Software as a Service, delivered internally?" *ZDnet.com*, März 2007. [Online]. Available: <http://blogs.zdnet.com/service-oriented/?p=825>
- [Mey02] M. Meyer, "CRM und CRM-Systeme - Grundlagen und Überblick," in *M. Meyer (Hrsg.) CRM-Systeme mit EAI - Konzeption, Implementierung und Evaluation*. Braunschweig/Wiesbaden: Vieweg, Juni 2002, pp. 3-20.
- [Mül06a] D. Müller, "Deutschland ist ein hartes Pflaster für CRM-on-Demand-Anbieter," *ZDNet.de*, Mai 2006. [Online]. Available: <http://www.zdnet.de/itmanager/kommentare/0,39023450,39143172,00.htm>
- [Mül06b] T. Müller, "SOA! Aber wann?" *SOA Blog*, Juli 2006. [Online]. Available: <http://www.explore-soa.de/soa/de/soablog/SOA%20Nutzen>
- [Mül07] C. Müller, "Oracle on Demand? Eine Standortbestimmung," *Competence-Center.de*, November 2007. [Online]. Available: [http://www.portale-competence-center.de/offshore.nsf/6E3ED5F8794D84E0C125738A003CE1C7/\\$File/software\\_as\\_a\\_service.pdf](http://www.portale-competence-center.de/offshore.nsf/6E3ED5F8794D84E0C125738A003CE1C7/$File/software_as_a_service.pdf)
- [Mus01] D. Muse, "ASPIC names new officers, agrees to work with ITAA," *internetnews.com*, 2001. [Online]. Available: <http://www.internetnews.com/xSP/article.php/771921>
- [MW02] M. Meyer and S. Weingärtner, "Enterprise Application Integration - Grundlagen," in *M. Meyer (Hrsg.) CRM-Systeme mit EAI - Konzeption, Implementierung und Evaluation*. Braunschweig/Wiesbaden: Vieweg, Juni 2002, pp. 199-230.



- [MZK03] M. Meyer, R. Zarnekow, and L. Kolbe, "IT-Governance - Begriff, Status quo und Bedeutung." *Wirtschaftsinformatik*, vol. 45, no. 4, pp. 445–448, 2003.
- [Nau05] F. Naujoks, "Die Wiedergeburt von ASP: Software as a Service," *crmmanager.de*, July 2005. [Online]. Available: [http://www.crmmanager.de/magazin/artikel\\_708\\_die\\_wiedergeburt\\_von\\_asp\\_software\\_as\\_a\\_service.html](http://www.crmmanager.de/magazin/artikel_708_die_wiedergeburt_von_asp_software_as_a_service.html)
- [NDS07] I. Nassi, J. Das, and M.-C. Shan, "The Challenges of Application Service Hosting," in *Lecture Notes in Computer Science - Webengineering*, vol. 46. Springer Berlin / Heidelberg, Juli 2007, pp. 545–549.
- [NH05] T. Noda and S. Helwig, "Rich Internet Applications," *Best Practice Reports*, p. 10, November 2005. [Online]. Available: <http://www.uwebc.org/opinionpapers/docs/RIA.pdf>
- [Nol06] R. Nolting, *Netzbasiertes Outsourcing in kleinen und mittleren Unternehmen*, band 3185 ed., ser. Voks- und Betriebswirtschaft. Peter Lang Verlagsgruppe, 2006, no. 5.
- [oOC06] o.V. Oracle Corporation, "Siebel CRM On Demand Sales," *crmondemand.oracle.com*, 2006. [Online]. Available: [http://crmondemand.oracle.com/ocom/groups/public/@crmondemand/documents/webcontent/7369\\_de.pdf](http://crmondemand.oracle.com/ocom/groups/public/@crmondemand/documents/webcontent/7369_de.pdf)
- [oOC09a] —, "Branchenspezifische Oracle CRM On Demand-Editionen," *crmondemand.oracle.com*, 2009. [Online]. Available: [http://crmondemand.oracle.com/de/products/industrysolutions/6080\\_EN\\_DE](http://crmondemand.oracle.com/de/products/industrysolutions/6080_EN_DE)
- [oOC09b] —, "On Demand - Warum Software als Service?" *crmondemand.oracle.com*, 2009. [Online]. Available: [http://crmondemand.oracle.com/de/why-oracle/6168\\_EN\\_DE](http://crmondemand.oracle.com/de/why-oracle/6168_EN_DE)
- [oRT06] o.V. RightNow Technologies, "RightNow führend bei On-Demand-CRM-Lösungen für Enterprise-Kunden," *rightnowtech*, January 2006. [Online]. Available: <http://www.rightnowtech.de/news/article.php?id=6112>
- [oRT07a] —, "RightNow Connect für unternehmensweite Integrationsfunktionen," *rightnowtech*, Mai 2007. [Online]. Available: <http://www.rightnowtech.de/news/article.php?id=7711>
- [oRT07b] —, "RightNow Technologies - Unabhängiges Marktforschungsunternehmen stuft RightNow als führenden Anbieter von



- SFA-Lösungen (Sales Force Automation),” rightnowtech, Mai 2007. [Online]. Available: <http://www.rightnowtech.de/news/article.php?id=7687>
- [oRT07c] —, “Unabhängiges Marktforschungsunternehmen stuft Right-Now als führenden Anbieter von CSM-Software (Customer Service Management) ein,” Rightnowtech, June 2007. [Online]. Available: <http://www.rightnowtech.de/news/article.php?id=7748>
- [oRT08] —, “Software as a Service: Perfekte Kundenerlebnisse mit SaaS und SOA,” Competence Site, June 2008. [Online]. Available: [http://www.competence-site.de/crm.nsf/B0A79007C3045CFCC125730E004F3F4A/\\$File/crm\\_saas\\_soa\\_kosteng%C3%BCnstige\\_integration.pdf](http://www.competence-site.de/crm.nsf/B0A79007C3045CFCC125730E004F3F4A/$File/crm_saas_soa_kosteng%C3%BCnstige_integration.pdf)
- [oS07a] o.V. Salesforce, “Integration von Web Services On-Demand,” Salesforce.com - Success on Demand, Mai 2007. [Online]. Available: <http://www.salesforce.com/de/company/news-press/press-releases/2007/05/070529.jsp>
- [oS07b] —, “Mandantenfähige Plattformen,” Salesforce.com - Success on Demand, 2007. [Online]. Available: <http://www.salesforce.com/de/platform/why-ondemand/multi-tenant-platforms/index.jsp>
- [oS08] —, “Integration,” salesforce.com, März 2008. [Online]. Available: <http://www.salesforce.com/de/platform/integration/>
- [o.V94] o.V., “ITIL - IT Infrastructure Library,” OGC - Office of Government Commerce, ITIL-Framework, 1994. [Online]. Available: <http://www.ogc.gov.uk/index.asp?id=2261>
- [o.V00] —, “Safe Harbor Overview,” Department of Commerce - Export Portal U.S. Department of Commerce, November 2000. [Online]. Available: [http://www.export.gov/safeharbor/sh\\_overview.html](http://www.export.gov/safeharbor/sh_overview.html)
- [o.V06a] —, “Information erschliesst das wahre Potenzial ihrer Kundenbasis - umfassendes Kundenmanagement für Beste Ergebnisse,” Oracle CRM On Demand, 2006. [Online]. Available: [http://crmondemand.oracle.com/ocom/groups/public/@crmondemand/documents/webcontent/7350\\_de.pdf](http://crmondemand.oracle.com/ocom/groups/public/@crmondemand/documents/webcontent/7350_de.pdf)
- [o.V06b] —, “Nichts ist beständiger als der Wandel,” *ONEtoONE*, no. 09, pp. 6–11, August 2006. [Online]. Available: [http://www.onetoone.de/downloads/themadesmonats/2006/OtO\\_0906\\_Interaktive\\_Trends.pdf](http://www.onetoone.de/downloads/themadesmonats/2006/OtO_0906_Interaktive_Trends.pdf)

- [o.V08a] —, “Kundenbasis von Salesforce.com,” Salesforce.com, März 2008. [Online]. Available: <http://www.salesforce.com/de/customers/all-customers.jsp>
- [o.V08b] —, “Mandantenfähige Plattformen,” Salesforce.com, März 2008. [Online]. Available: <http://www.salesforce.com/de/platform/why-ondemand/multi-tenant-platforms/>
- [o.V08c] —, “Oracle gibt Verfügbarkeit von Siebel CRM On Demand Release,” CRM Forum, August 2008. [Online]. Available: <http://www.crmforum.de/news-archiv/archiv5579.html>
- [o.V08d] —, “RightNow Technologies - Customer Experience Management, CRM, Lösungen für Service, Marketing und Vertrieb,” Competence Site, April 2008. [Online]. Available: <http://www.competence-site.de/offerer.nsf/anbieter/r1148-rightnow-technologies>
- [o.V09] —, “RightNow Technologies - Einbindung von WS-I,” rightnowtech, 2009. [Online]. Available: <http://www.rightnowtech.de/support/development.html>
- [Pas04] A. Paschke, “Regelbasiertes SLA-Management - Ein regelbasierter Ansatz zum automatisierten IT-Dienstleistungsmanagement,” Internetbasierte Geschäftssysteme (IBIS), Technische Universität München, Technical Report, Juni 2004. [Online]. Available: <http://ibis.in.tum.de/staff/paschke/docs/RBSLA.pdf>
- [PBW00] A. Picot, A. Buttermann, and R. Walters, “Erfolgsfaktoren für Application Service Providing,” *Information Management & Consulting*, no. 15, pp. 45–51, 2000.
- [Pie07] G. Piech, “Unklare Mietverhältnisse,” DV-Dialog - IT-Mittelstand, Juni 2007. [Online]. Available: <http://www.dv-dialog.de/startseite/detailansicht-brennpunkt/artikel///67/unklare-mietverhaeltnisse.html#>
- [Pre02] S. Preuschoff, “Business Intelligence? Gegenstand, Ansätze und Technologien,” H. Nohr (Hrsg.) Arbeitspapiere Wissensmanagement, Fachhochschule Stuttgart, Tech. Rep. 3, 2002. [Online]. Available: <http://www.iuk.hdm-stuttgart.de/nohr/Km/KmAP/BusinessIntelligence.pdf>
- [Pri04] B. Pricing, “ASP Transforms into On Demand and Software-as-a-Service,” Gartner Research, Stamford, Analysis, November 2004.

- [Rei05] M. Reinicke, "Eine ökonomische Bewertung der Dienstausswahlverfahren in serviceorientierten Overlaynetzen," in *Wirtschaftsinformatik 2005*. Physica-Verlag HD, 2005, pp. 129–148. [Online]. Available: [http://www.iw.uni-karlsruhe.de/catnets/fileadmin/publications/wi-paper\\_cr.pdf](http://www.iw.uni-karlsruhe.de/catnets/fileadmin/publications/wi-paper_cr.pdf)
- [RH06] R. Reussner and W. Hasselbring, *Handbuch der Software-Architektur*. Heidelberg: Dpunkt Verlag, 2006.
- [RhS05] J.-P. Richter, H. haller, and P. Schrey, "Serviceorientierte Architektur," Informatiklexikon der Gesellschaft für Informatik e.V. (GI), 2005. [Online]. Available: <http://www.gi-ev.de/service/informatiklexikon/informatiklexikon-detailansicht/meldung/118/>
- [Rie05] R. Riedl, *Application Service Providing - Entwicklung eines Modells zur Qualitätsmessung*. Wiesbaden: Deutscher Universitäts-Verlag, 2005.
- [Rie06] A. Riegler, "SaaS - Software as a Service," Monitor, Dezember 2006. [Online]. Available: <http://www.monitor.co.at/index.cfm?storyid=8792>
- [RSG06] A. Rüter, J. Schröder, and A. Göldner, *IT-Governance in der Praxis*. Berlin: Springer, 2006.
- [RW04] J. W. Ross and G. Westerman, "Preparing for utility computing: The role of IT architecture and relationship management," in *IBM Systems Journal*, vol. 43, no. 1. IBM, 2004, pp. 5–19.
- [S+03] A. Schmietendorf *et al.*, "Aktuelle Ansätze für Web Service basierte Integrationslösungen," Arbeitsgruppe Softwaretechnik - Software Measurement Laboratory SIMLab, Otto-von-Guericke-Universität Magdeburg, Preprint, 2003.
- [S+06] W. Schwetz *et al.*, "Virtual roundtable: CRM-on-Demand - Wann kommt der Durchbruch?" Competence Site CRM, 2006. [Online]. Available: <http://www.competence-site.de/crm/crm-on-demand-roundtable>
- [Sch02] J. Schulze, *CRM erfolgreich einführen*. Berlin: Springer, 2002.
- [Sch04] A. C. Schwickert, "Geschäftsmodelle im Electronic Business ? Bestandsaufnahme und Relativierung," *Arbeitspapiere WI*, no. 2, p. 16, 2004. [Online]. Available: [http://www.econbiz.de/archiv/gi/ugi/winformatik/geschaeftsmodelle\\_eb.pdf](http://www.econbiz.de/archiv/gi/ugi/winformatik/geschaeftsmodelle_eb.pdf)
- [Sch05] J. Schumacher, "Outsourcing von CRM-Prozessen," Ph.D. dissertation, RWTH Aachen, 2005.

- [Sch07a] M. Scheibenpflug, "Die Kunst der Ganzheitlichkeit," Computerwoche, CRM Initiative 2007 3, Jänner 2007.
- [Sch07b] A. Schmietendorf, "IT-Management serviceorientierter Architekturen," *HMD - Praxis Wirtschaftsinformatik*, vol. 253, pp. 74–83, Februar 2007.
- [Sch07c] W. Schwetz, "CRM-Systeme auf dem Prüfstand," Schwetz Consulting, CRM-Software Zertifizierung 2007 / 2008, November 2007. [Online]. Available: [http://www.schwetz.de/Zertifizierung/pdf/B&R\\_Zertbericht\\_07-08\\_051107\\_final.pdf](http://www.schwetz.de/Zertifizierung/pdf/B&R_Zertbericht_07-08_051107_final.pdf)
- [Sch07d] —, "im Gespräch mit Constantin Schaible," Competence Center CRM, Juni 2007. [Online]. Available: [http://www.eai-systeme.de/crm.nsf/1764D0E0583CCE88C1257301006CB131/\\$File/einterview\\_crm-l%F6sungen\\_b2b\\_vs\\_b2c\\_schaible.pdf](http://www.eai-systeme.de/crm.nsf/1764D0E0583CCE88C1257301006CB131/$File/einterview_crm-l%F6sungen_b2b_vs_b2c_schaible.pdf)
- [SD07] A. Schatten and M. Demolsky, "Leistungsfähige Web Services," *Infoweek.ch*, no. 2, pp. 55–57, Februar 2007.
- [Sew05] S. Sewera, "Referenzmodelle im Rahmen von IT-Governance Cobit ITIL MOF," Wirtschaftsuniversität Wien, Seminararbeit, 2005. [Online]. Available: <http://www.wu-wien.ac.at/~koch/lehre/inf-sem-ss-05/referenzmodelle.pdf>
- [SJ02] J. G. Spooner and S. Junnarkar, "IBM talks up 'computing on demand'," CNet News.com, Oktober 2002. [Online]. Available: [http://www.news.com/IBM-talks-up-computing-on-demand/2100-1001\\_3-963807.html](http://www.news.com/IBM-talks-up-computing-on-demand/2100-1001_3-963807.html)
- [SJ04] R. Schnauffer and H.-H. Jung, *CRM - Entscheidungen richtig treffen*. Springer, Berlin, 2004.
- [SK08] M. Steger and C. Kappert, "Services mit einem kontextbezogenen Gesicht," *Javamagazin*, no. 3, Februar 2008. [Online]. Available: [http://it-republik.de/zonen/magazine/ausgaben/psfile/source\\_file/19/Onlinearti47a74020cb4e8.pdf](http://it-republik.de/zonen/magazine/ausgaben/psfile/source_file/19/Onlinearti47a74020cb4e8.pdf)
- [SMS06] T. Schmidt-Melchior and S. Schmidt, *Flexible Servicemodelle - die atmende IT durch adaptives Outsourcing*, ser. Innovation durch IT. Springer Berlin Heidelberg, 2006.
- [Sta01] P. Stahlknecht, "Aktueller Stand und Entwicklungstendenzen im IT-Outsourcing und im Application Service Providing," *Vortrag am Institut für Wirtschaftsinformatik der Phillips-Universität Marburg am 23.01.2001*, pp. 1–21, Jan 2001. [Online]. Available: <http://www.uni-marburg.de/fb02/bwl06/studium/hauptstudium/betranwendungssysteme/download/asp>

- [Ste07] P. Stemmler, “ASP, On demand & Outsourcing: Aus ASP wird SaaS,” e-commerce Magazin, April 2007. [Online]. Available: <http://www.e-commerce-magazin.de/index.php3?page=04-07/ondemand.html>
- [Sti07] A. Stiehler, “Systematisierung und Handelbarkeit von IT-Dienstleistungen,” Berlecon Research, Berlin, Berlecon Research/INTERDIG, Februar 2007. [Online]. Available: [http://www.berlecon.de/studien/downloads/Interdig\\_Report1.pdf](http://www.berlecon.de/studien/downloads/Interdig_Report1.pdf)
- [Sto06] T. Stober, “Virtuelle Portale - Mandantenfähigkeit für Websphere Portal Server.” *it - Information Technology*, vol. 48, no. 2, pp. 90–98, 2006.
- [Str05] W. Streitberger, “Framework for the Negotiation of Electronic Contracts in E-Business on Demand,” in *Proceedings of the Seventh IEEE International Conference on E-Commerce Technology CEC05*, 2005, pp. 370–373.
- [SZC<sup>+</sup>07] W. Sun, K. Zhang, S.-K. Chen, X. Zhang, and H. Liang, “Software as a Service: An Integration Perspective,” in *Service-Oriented Computing - ICSOC*, ser. Lecture Notes in Computer Science, B. J. Krämer, K.-J. Lin, and P. Narasimhan, Eds., vol. 4749. Wien: Springer, September 2007, pp. 558–569.
- [Tam03] G. Tamm, “Neztbasierte Dienste: Angebot, Nachfrage und Matching,” Ph.D. dissertation, Humboldt-Universität, Berlin, 2003. [Online]. Available: <http://www.edoc.hu-berlin.de/dissertationen/tamm-gerrit-2003-05-09/PDF/Tamm.pdf>
- [Ull04] T. Ullerich, *TCO-Modell für SAP-Systeme am Beispiel mySAP CRM mit SAP Enterprise Portal*. Galileo Press, 2004.
- [vG07] A. von Gunten, “Mit Software als Service zum Erfolg,” InfoWeek, 2007. [Online]. Available: [http://www.infoweek.ch/archive/ar\\_single.cfm?ar\\_id=18503&ar\\_subid=2&sid=0](http://www.infoweek.ch/archive/ar_single.cfm?ar_id=18503&ar_subid=2&sid=0)
- [vG08] —, “Plattformen als Service,” scribd.com, Februar 2008. [Online]. Available: [http://www.scribd.com/full/2170011?access\\_key=key-23vfh8s1ve84i42fsz2c](http://www.scribd.com/full/2170011?access_key=key-23vfh8s1ve84i42fsz2c)
- [VH07] G. Vossen and S. Hagemann, *Unleashing Web 2.0: From Concepts to Creativity*. Burlington: Morgan Kaufmann, July 2007.
- [Vol06] M. Volland, “Software-Lego gehört die Zukunft,” Computerwoche.de, Dezember 2006. [Online]. Available: <http://www.computerwoche.de/itevolution/news/584445/>

- [Vor04] J. Vorisek, "Business Drivers for Application Servicing and a Software-as-a-Service Model," in *ICEB*, 2004, pp. 511–516.
- [VSTT06] B. Vassiliadis, A. Stefani, J. Tsaknakis, and A. Tsakalidis, "From application service provision to service-oriented computing: A study of the IT outsourcing evolution," *Telematics and Informatics*, vol. 23, pp. 271–293, November 2006.
- [Wal02] G. Walter, "Application Service Providing mit Schwerpunkt auf Organisation und Technik," Master's thesis, Technische Universität Wien, März 2002. [Online]. Available: [http://www.ifs.tuwien.ac.at/~bruckner/students/Application\\_Service\\_Providing.pdf](http://www.ifs.tuwien.ac.at/~bruckner/students/Application_Service_Providing.pdf)
- [WBK07] S. M. Walter, T. Böhm, and H. Krcmar, "Industrialisierung der IT - Grundlagen, Merkmale und Ausprägungen eines Trends," *HMD - Praxis Wirtschaftsinformatik*, vol. 256, pp. 1–11, August 2007.
- [Wei07] W. Weigend, "Kommentar über SOA als Grundlage für Software-as-a-Service (SaaS)," SOA-Expertenrat - COMPUTERWOCHE, Mai 2007. [Online]. Available: <http://www.computerwoche.de/soa-expertenrat/?p=170#more-170>
- [Wil06] E. Wilde, "Modulare und Offene Komponenten zur Wissensverwaltung," 11.Europäische Jahrestagung der Gesellschaft für Medien in der Wissenschaft (GMW06), September 2006. [Online]. Available: <http://dret.net/netdret/docs/wilde-gmw06-modular.pdf>
- [ZE08] P. Zencke and R. Eichin, "SAP Business ByDesign - Die neue Mittelstandslösung der SAP," *Wirtschaftsinformatik*, vol. 1, pp. 47–51, 2008. [Online]. Available: [http://www.wirtschaftsinformatik.de/wi\\_artikel.php?sid=2172](http://www.wirtschaftsinformatik.de/wi_artikel.php?sid=2172)
- [ZS07] T. Ziemann and D. Schepanek, "Industrialisierung der Service-Leistungserbringung," *Zeitschrift der Unternehmensberatung (ZUb)*, no. 06, pp. 259–266, 2007. [Online]. Available: [http://zubdigital.de/aid/zub\\_20070607/inhalt.html](http://zubdigital.de/aid/zub_20070607/inhalt.html)

# A. Entwicklung des Outsourcings

In den letzten vier Jahrzehnten hat sich das IT-Outsourcing sukzessive weiterentwickelt. Grundsätzlich können nach [Nol06, S. 205 ff.] drei “Wellen” unterschieden werden, die den Entwicklungsvorgang beschreiben:

- Das technologiezentrierte Outsourcing der sechziger Jahre leitete die erste Welle des Outsourcings ein [VSTT06, S. 275]. Insbesondere große Unternehmen fingen an einige technische Komponenten, wie z.B. Mainframes (Hauptrechner) auszugliedern. Das Outsourcing dieser Welle wurde daher mit der TSO, der Time Sharing Option in Verbindung gebracht. So wurde die Benutzung der Rechenleistung durch mehrere Benutzer ermöglicht, da über diese Option eine bestimmte CPU (Central Processing Unit)-Zeit zugeordnet werden konnte.
- Die zweite Welle des Outsourcings hatte einen geschäftszentrierten Charakter und fand ab den achtziger Jahren statt. Zusätzlich zu den technischen Komponenten, wurden zunehmend auch Unternehmensfunktionen (z.B. Buchhaltung), Geschäftsprozesse (Business Process Outsourcing) sowie weitere Leistungen (wie z.B. die Systementwicklung und -überwachung) ausgelagert. Während in den Unternehmen vor allem maßgeschneiderte Softwarelösungen eingesetzt wurden, nahm in der zweiten Welle die Nutzung standardisierter Software stark zu. Insbesondere das Aufkommen der Client-Server-Architektur führte zu einer Verlagerung der Rechenleistung vom Client auf zentrale Server und ermöglichte daraufhin das Modell des Server Based Computing. Dabei wurde der Fernzugriff auf die Anwendungen über einen Terminalserver durchgeführt und Ende der Neunziger von den ersten Application Service Provider angeboten [VSTT06, S. 275].
- Durch den Erfolg des Internets und der darauf folgenden Entwicklung von webbasierter Software, wurde die dritte Welle des IT-Outsourcings ausgelöst. Diese ist industriezentriert und zeigt eine signifikante Weiterentwicklung von der ersten Welle, die sich durch das sogenannte “centralized computing” auszeichnet und der zweiten Welle, die sich durch das sogenannte “distributed computing” auszeichnet, bis hin zu dem “remote computing”, welches durch eine webzentrierte Infrastruktur geprägt ist. Zusätzlich wurde mit dem Architekturmuster des Model-View-Controller-Prinzips die Verschlinkung der Applikationen am Frontend

---

ermöglicht, sodass sich auch komplexe Anwendungen auf mobilen Geräten mit begrenzten Ressourcen nutzen lassen. Dafür eignet sich insbesondere die CRM-Software, da die mobile Technologie für den Außendienst (▷ vgl. Kapitel: 4.2.2) viel Freiraum zur Prozessinnovation bietet [GK06].

Vor allem das Management und die Lieferung von geschäftsprozessunterstützenden Anwendungen durch Service Provider erfuhr im Zuge der Verbreitung und der steigenden Akzeptanz des Internets einen bemerkenswerten Aufschwung, da fortan eine räumlich und zeitlich unabhängige Softwarenutzung möglich war.



## B. On Premise Einführung

Dabei wird die Einführung eines CRM-Systems in Anlehnung an die etablierte Einführungsmethode “*Accelerated SAP (ASAP)*”, folgendermaßen dargestellt [BE01, S. 249 ff.] :

- Vorbereitung und Überprüfung des Standortes durch eine IST-Aufnahme der bestehenden Organisation. Diese Phase findet statt, um zu prüfen inwieweit die Software die funktionalen Anforderungen des Kunden abdeckt. Das Ziel dieser *Einsatzuntersuchung* dient dazu um:
  - Geschäft und Anforderungen des Kunden zu verstehen
  - Geschäftsprozesse des Kunden nachzuvollziehen
  - die bestehende Systemlandschaft und den Customizing-Grad der eingesetzten Anwendungen zu verstehen
  - Daten über Umfang, Ressourcen und Kosten des Projekts zu sammeln
  - Hardware- und softwarespezifische Anforderungen zu erarbeiten
  - Mit den relevanten Entscheidungsträgern abstimmen
- Mit Hilfe des Business Blueprints wird anhand der ermittelten Geschäftsprozessanforderungen ein Sollkonzept erstellt auf dem erarbeitet wird, wie der betriebswirtschaftliche Ablauf im CRM-System abgebildet werden soll. Des weiteren muss sichergestellt werden, dass ausreichende Kapazitäten für Verarbeitung, Speicherung und Sicherung vorhanden sind, damit die geplante CRM-Software an den laufenden Anforderungen des Unternehmens skaliert werden kann.
- Die Implementierungsphase dient der Umsetzung der im Business Blueprint definierten Anforderungen. Mit zwei Arbeitspaketen wird die Realisierung durchgeführt. Wird bei der Auswahl der Systemkomponenten festgestellt, dass nicht alle Anforderungen abgebildet werden können, wird das System den Anforderungen durch Parameterveränderungen in Datenbanktabellen, Veränderungen in Templates oder durch Manipulation des Quellcodes, angepasst [Sch02, S. 194 ff.]. Das Customizing, beginnt im Rahmen der Konfigurationsmöglichkeiten, bei länderspezifischen Einstellungen wie z.B. Sprache, Währung usw. [HN01, S. 527]. Es folgen Anpassungen in der Organisations- und Datenstruktur sowie in

---

der Installation und im Umfang der notwendigen Systemkomponenten [SJ04, S. 158 f.]. Mit Hilfe von APIs wird der Datentransfer zwischen den notwendigen Systemen realisiert. Die *Baseline-Konfiguration* zeichnet sich daher aus, dass die Organisationsstrukturen vollständig und die Geschäftsprozesse zu 60 % konfiguriert sind. Dadurch kann nach der Abnahme des Baseline-System systematisch an den Details weitergearbeitet werden. Bei der *Detail-Konfiguration* werden die durch die Erprobungsphase (Evaluation der Unterstützungsmöglichkeiten sowie die Auswirkungen auf Bearbeitungs- und Kooperationsprozesse) ermittelten Änderungen am System durchgeführt.

- Gegenstand der Produktionsvorbereitung und Qualitätssicherung, auch Pilotphase genannt ist neben Tests, Benutzerschulungen und dem Aufsetzen von Systemmanagements auch der Anwender-Support, damit alle offenen Fragen entgültig geklärt werden. Darüber hinaus müssen auch die Funktionalitäten des Datenaustausches und der Datenverteilung zwischen den einzelnen Komponenten getestet werden [SJ04, S. 160]. Was zu diesem Zeitpunkt an Funktionalität - besonders an Datenbasis - vorhanden ist entscheiden für die Akzeptanz und damit für den implizierenden Erfolg der Einführung. Es müssen also im Rahmen des Initial Loads alle Tabellen und Auswahlmenüs mit Inhalten gefüllt werden [SJ04, S. 160].
- Die Liveschaltung in das Produktivsystem (Rollout) beinhaltet die Aufgabe einen Benutzer-Support einzurichten und aufzubauen, der nicht nur in den ersten kritischen Tagen, sondern langfristig zur Verfügung steht. Diese Phase dient außerdem dazu, die Systemtransaktionen zu überwachen und die Gesamtleistung des Systems zu optimieren. Am Ende dieser Phase wird dann das Projekt abgeschlossen.

## C. Varianten zur Gewährleistung des Datenschutzes

Datenübertragung innerhalb der EU:	Der Datentransfer bzw. die Datenverarbeitung innerhalb der EU stellt aufgrund der EU-Richtlinie 95/46 EG und den daraus resultierenden Datenschutzgesetzen kein Problem dar.
Angemessenheitsentscheidung der EU-Kommission:	Darüber hinaus gibt es von der Europäischen Kommission einen Bescheid über Nicht EU-Staaten, in welchen ein, den EU-Richtlinien angemessenes, Datenschutzniveau herrscht. Dazu zählen z.B. die Schweiz, Argentinien sowie die Kanalinseln Guernsey und Jersey.
U.S. "Safe Harbor" - Programm:	Im rechtskonformen Raum bewegt man sich auch bei US-Unternehmen, die sich am "Safe Harbor" - Übereinkommen beteiligen. Dieses Regelement gilt bisher nur in den Vereinigten Staaten.
EU-Standardvertragsklauseln:	Eine weitere Möglichkeit des gesetzeskonformen Verhaltens sind Standardvertragsklauseln, in deren sich der Anbieter - ähnlich dem Safe-Habor Übereinkommen - zur Einhaltung europäischer Datenschutzrichtlinien verpflichtet, so dass auch ein Datentransfer in das nicht-europäische und nicht-amerikanische Ausland möglich ist.
Verbindliche Unternehmensregelungen:	Verbindliche Unternehmensregelungen, zum Teil Data Protection Code of Conduct genannt, bieten auch eine Variante zur Einhaltung des europäischen Datenschutzniveaus. Ein Konzern kann sich dadurch ein eigenes Regelwerk für den Umgang mit personenbezogenen Daten, (wie z.B. Kundendaten) aufstellen, wenn diese von einer Datenschutzbehörde genehmigt wird.

**Tabelle C.2.:** Varianten zur Gewährleistung des Datenschutzes [BIT06c, S. 64 ff.]

---