

2 Rechner und deren Vernetzung

2.1 Hardware

Unter Hardware versteht man vereinfacht ausgedrückt alle physischen Komponenten und Geräte, aus denen sich ein Computer oder Rechnernetzwerk zusammensetzt.

Ein typischer PC-Arbeitsplatz besteht aus folgenden Hardwarekomponenten:

- Prozessor
- Hauptspeicher
- externe Speicher (z. B. Festplatte, DVD; siehe Abschnitt 2.1.2)
- Dateneingabegeräte (z. B. Tastatur, Maus, Scanner)
- Datenausgabegeräte (z. B. Bildschirm, Drucker)

Darüber hinaus lässt sich ein solcher Arbeitsplatz um eine Netzwerkschnittstelle (z. B. Netzwerkkarte, Modem) ergänzen, wodurch der Computer an ein Kommunikationsnetz angeschlossen werden kann (siehe Abschnitt 2.5). Neben stationären Arbeitsplätzen haben in den letzten Jahren portable Geräte wie Notebooks und Personal Digital Assistants (PDA), etwa in Verbindung mit Mobiltelefonen, zunehmend an Bedeutung gewonnen.

Die Arbeitsweise eines Rechners kann man sich wie folgt vorstellen: Zunächst werden Daten eingegeben, z. B. über die Tastatur oder über optische Lesegeräte. Diese werden dann verarbeitet und anschließend etwa auf einem Bildschirm, Drucker oder einem externen Speichermedium ausgegeben. Dieser Ablauf wird als *Eingabe-Verarbeitung-Ausgabe-Prinzip* (EVA-Prinzip) bezeichnet.

Abbildung 2.1/1 stellt die Grundform einer *Zentraleinheit* dar, die aus einem Prozessor (Central Processing Unit, CPU), einem Hauptspeicher und einem Ein-/Ausgabewerk besteht. Das Ein- und Ausgabewerk dient als Schnittstelle für Geräte zur Dateneingabe und -ausgabe, welche ebenso wie externe Speicher nicht Bestandteil der Zentraleinheit sind. Der Prozessor setzt sich wiederum aus einem Rechenwerk und einem Steuerwerk zusammen.

Zur Verdeutlichung der Befehlsverarbeitung in einer Zentraleinheit stellen wir uns vor, dass das gesamte auszuführende Programm (Befehle und Daten) im Hauptspeicher geladen ist. Das *Steuerwerk* holt sich mittels einer Speicheradresse den ersten Befehl, interpretiert ihn und veranlasst das *Rechenwerk*, ihn auszuführen. Das Rechenwerk entnimmt die Daten aus dem Hauptspeicher, führt die Operation aus und legt das Ergebnis wieder im Hauptspeicher ab (z. B. durch Überschreiben der „alten“ Position mit der „neuen“). Anschließend lädt das Steuerwerk den nächstfolgenden Befehl aus dem Hauptspeicher, um wie oben beschrieben zu verfahren.

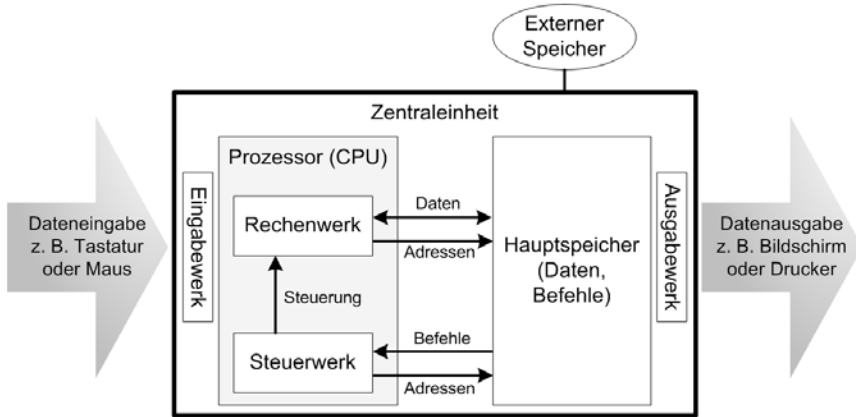


Abb. 2.1/1 Grundform einer Rechnerarchitektur

Rechnerarchitekturen, die entsprechend Abbildung 2.1/1 aufgebaut sind und die den vorgenannten Ablaufzyklus der Befehlsverarbeitung realisieren, werden nach dem Mathematiker und Kybernetiker John von Neumann, der diese Prinzipien der sog. speicherprogrammierten Rechner 1946 formulierte, als *Von-Neumann-Rechner* bezeichnet. Ein Nachteil der Von-Neumann-Architektur besteht darin, dass aufgrund des gemeinsamen Speichers Daten und Befehle nur sequenziell verarbeitet werden können.

Neben der von-Neumann-Architektur gibt es einige Konzepte und Erweiterungen, die in heutigen Rechnersystemen standardmäßig benutzt werden. So ist in der sogenannten *Harvard-Architektur* der Hauptspeicher in einen Befehls- und einen Datenspeicher unterteilt, sodass das Steuer- und das Rechenwerk Befehle und Daten parallel aus dem Hauptspeicher laden können. In den letzten Jahren werden PCs zumeist mit *Mehrkernprozessoren* ausgestattet. Hierbei befinden sich auf einem Chip mehrere parallel arbeitende Hauptprozessoren.

2.1.1 Zentraleinheit

2.1.1.1 Prozessor

Hersteller von Mikroprozessoren drücken deren Leistungsfähigkeit i. d. R. durch die Maßzahl Gigahertz (GHz) aus, welche die Taktfrequenz des Prozessors angibt. Sie determiniert, wie viele Milliarden Befehle pro Sekunde ausgeführt werden können, erlaubt jedoch keinen direkten Rückschluss auf die Verarbeitungsgeschwindigkeit, denn die interne Verarbeitungsgeschwindigkeit eines Prozessors ist auch davon abhängig, wie schnell z. B. die einzelnen Einheiten Rechenwerk, Steuerwerk und Hauptspeicher operieren und wie schnell zwischen Rechenwerk und Steuerwerk sowie zwischen Prozessor und Hauptspeicher kommuniziert werden kann. Diese Komponenten sind mit

sog. *Bussen*, die man sich als mehradrige Kabel vorstellen kann, verbunden. Man unterscheidet zwischen dem Adressbus und dem Datenbus, wobei sich die Bitangabe (als weiteres Leistungsmerkmal eines Prozessors) auf die gleichzeitig über einen Bus übertragbare Datenmenge bezieht („Bit“ als Maßeinheit für die Datenmenge bei digitaler Speicherung leitet sich von „binary digit“ (Binärziffer) ab). Ein Adressbus ist unidirektional und überträgt ausschließlich Speicheradressen. Der Datenbus dagegen ist bidirektional und transferiert Daten oder Befehle zwischen dem Prozessor und dem Hauptspeicher. Heutige Prozessoren verfügen zumeist über einen 64-Bit-Daten- und einen 32-Bit-Adressbus. Busse dienen zudem auch der externen Kommunikation des Prozessors, z. B. mit Speichermedien wie einer Festplatte.

2.1.1.2 Hauptspeicher

Der Hauptspeicher eines Informationsverarbeitungssystems besteht aus dem Arbeitsspeicher und einem Festwertspeicher.

Der *Arbeitsspeicher* (Random Access Memory, RAM) setzt sich aus direkt adressierbaren Speicherzellen zusammen, die als Speicherworte bezeichnet werden. Bei einem PC besteht ein Wort i. d. R. aus 4 Byte (1 Byte entspricht 8 Bit). Hauptspeicherkapazitäten werden in Megabyte (1 MB = 2^{20} Byte) oder Gigabyte (1 GB = 2^{30} Byte) angegeben. Handelsübliche Rechner besitzen heute eine Arbeitsspeicherkapazität von 2 bis 4 GB.

Alle Programme müssen zum Zeitpunkt ihrer Ausführung vollständig oder partiell (mit dem aktuell auszuführenden Teil) im Arbeitsspeicher zur Verfügung stehen. Im zweiten Fall bietet das Betriebssystem die *virtuelle Speichertechnik* an. Dabei lagert es automatisch Programmteile, die nicht mehr in den Arbeitsspeicher geladen werden können (da z. B. andere Programme ebenfalls zur schnellen Abarbeitung im Hauptspeicher abgelegt sein müssen), auf der Festplatte aus und bringt sie nur bei Bedarf in den Arbeitsspeicher, wodurch sich dieser logisch, jedoch nicht physisch vergrößert. Das Ein- und Auslagern auf Festplatte bezeichnet man auch als *Paging*, da Programme und Hauptspeicher aus mehreren gleich großen Seiten (*Pages*) bestehen. Im Gegensatz zu Festplatten, die Daten auch nach Abschalten des Rechners halten, verliert der Arbeitsspeicher bei einer Unterbrechung der Stromzufuhr alle Informationen, die sich gerade in ihm befinden.

Ein *Festwertspeicher* (Read Only Memory, ROM) kann nur gelesen, nicht jedoch verändert werden. Festwertspeicher werden i. d. R. vom Hersteller beschrieben und dienen u. a. der Aufbewahrung grundlegender Teile des Betriebssystems, auf die beim Einschalten des Rechners automatisch zugegriffen wird (z. B. hardwarenahe Programme zur Ansteuerung des Bildschirms oder zur Kommunikation mit der Tastatur).

2.1.2 Externe Speicher

Ein externer Speicher ist speziell dazu geeignet, größere Datenmengen langfristig aufzubewahren und transportabel zu machen. Die wichtigsten externen Speichermedien sind Festplatten, optische Speicher und Speicherkarten:

Eine herkömmliche *Festplatte* (auch als Magnetplatte bezeichnet) weist i. d. R. mehrere übereinander gestapelte Kunststoff- oder Aluminiumscheiben auf, die mit einer magnetisierbaren Schicht überzogen sind. Daten werden in Form von Bitketten in konzentrischen Spuren durch Magnetisierung dargestellt. Der vielfach fest im PC-Gehäuse installierte Plattenstapel dreht sich mit konstanter Geschwindigkeit. Auf die Daten greifen Schreib-Lese-Köpfe zu, die radial auf die gewünschte Spur positioniert werden und dann warten, bis der Sektor mit den zu verarbeitenden Daten „vorbeikommt“. Magnetplatten für PC besitzen heute i. Allg. eine Speicherkapazität von mehreren hundert Gigabyte. In Großrechnersystemen werden Kapazitäten von mehreren Terabyte ($1 \text{ TB} = 2^{40} \text{ Byte}$) erreicht.

Eine relativ neue Festplattentechnologie stellt die sog. *Solid State Disk* dar. Im Gegensatz zu Magnetplatten enthalten Solid State Disks keine beweglichen Komponenten, sind robuster und erlauben schnelleren Zugriff. Ein Nachteil besteht in dem vergleichsweise höheren Preis pro Speichereinheit.

Sollen z. B. Sicherungskopien von Datenbeständen gelagert oder zwischen nicht vernetzten Rechnern ausgetauscht werden, so greift man auf transportable Speichermedien zurück. Weite Verbreitung haben hier insbesondere optische Speichermedien und Speicherkarten gefunden.

Bei *optischen Speichern* zeichnet man die Daten mit einem Laserstrahl auf der unterhalb einer transparenten Schutzschicht liegenden Speicherschicht auf, wobei deren Oberfläche verändert wird. Diese Strukturen sind wiederum mittels Laserstrahl abtastbar. Da das Laserlicht eine kurze Wellenlänge hat und sehr genau positioniert werden kann, besitzen optische Speicher eine hohe Kapazität. Es werden verschiedene Techniken unterschieden: CD-ROMs (Compact Discs Read Only Memory) haben eine Speicherkapazität von bis zu 700 MB (dieser Wert entspricht 80 Audiominuten). Varianten sind die CD-R (CD-Recordable) zum einmaligen und die CD-RW (CD-Recordable) zum wiederholten Beschreiben von CDs. Die für komplexe audiovisuelle Anwendungen konzipierte DVD (Digital Versatile Disc, häufig auch digitale Video Disc) hat in jüngster Vergangenheit die CD größtenteils substituiert. Ihre Speicherkapazität beträgt rund 5 GB. Weiterentwicklungen umfassen DVD-R/+R (DVD-Recordable) zum einmaligen und DVD-RW/+RW zum wiederholten Beschreiben. Den Nachfolger der DVD stellt die Blu-ray Disc dar, welche eine Speicherkapazität von knapp 30 GB aufweist.

Speicherkarten (auch als Flash Card oder Memory Card bezeichnet) sind transportable Speichermedien im Miniaturformat, welche auf der Flash-Speichertechnik (digitale Speicherchips) basieren und ohne permanente Stromversorgung Daten speichern können. Verbreitete Speicherkartenvarianten sind bspw. die Secure Digital Memory Card (SD) und die CompactFlash-

Card (CF). *USB-Massenspeicher* nutzen zumeist ebenfalls die Flash-Speichertechnik, kommunizieren aber über den Universal Serial Bus. Die am häufigsten verwendete Form ist der USB-Stick.

2.1.3 Datenein- und -ausgabegeräte

Zu den wichtigsten Dateneingabegeräten zählen neben Tastatur und Maus der Touchscreen, optische Belegleser und Lesegeräte zur Erfassung von elektromagnetischen Wellen.

Bei einem *Touchscreen* deutet der Benutzer auf ein Objekt auf dem Bildschirm und optische oder magnetische Sensoren registrieren die Berührung sowie die Positionierung (z. B. bei Geldautomaten). Ein *optischer Belegleser* erfasst genormte Daten, z. B. Bar- und OCR(Optical-Character-Recognition)-Code, indem etwa mit einem Lesestift die einzugebende Vorlage abgetastet wird, um Hell-Dunkel-Unterschiede zu erkennen und auszuwerten. Optische Eingabegeräte benutzt man z. B. an Scannerkassen in Supermärkten oder in Kreditinstituten zum Einlesen von Formularen. Eine Variante optischer Belegleser sind Scanner, welche die Vorlage in Bildpunkte zerlegen und als Graubild oder farbig erfassen (z. B. Fotos und Grafiken). Zu Eingabegeräten zählen ebenfalls sog. *RFID(Radio Frequency Identification)-Lesegeräte*, die der Erfassung von elektromagnetischen Wellen dienen, welche von an Gegenständen oder Lebewesen befindlichen RFID-Transpondern ausgesendet werden (siehe Abschnitte 1.1.3 und 4.3.3.3). Sind zum Beispiel in Bibliotheken Bücher mit RFID-Transpondern ausgestattet, so können ganze Stapel auf einmal aus- oder eingebucht werden, ohne dass die Bücher einzeln aufgeschlagen werden müssen, um einen Barcode zu scannen.

Das für die betriebliche Informationsverarbeitung wichtigste Ausgabe-medium ist der *Bildschirm* (Monitor). Er dient der Datenausgabe und unterstützt die Eingabe, da auf dem Bildschirm z. B. auch Masken zur Datenerfassung und Symbole (Icons) zur Aktivierung von Programmen dargestellt werden. Ein weiteres wichtiges Ausgabegerät ist der *Drucker*, der es erlaubt, Arbeitsergebnisse auf Papier zu bringen. Verwendung finden sowohl Tintenstrahl- als auch Laserdrucker. Der Tintenstrahl drucker setzt Zeichen und Grafiken aus Einzelpunkten zusammen und spritzt sie als schnell trocknende Tinte auf das Papier. Beim Laserdrucker wird die Seite als Ganzes im Drucker aufgebaut und mittels Toner auf das Papier übertragen.

2.2 Software

Software bildet die Voraussetzung für den Betrieb eines Rechners und bezeichnet allgemein in einer Programmiersprache geschriebene Programme, die nach Übersetzung auf einem Rechner ausführbar sind. Man unterscheidet nach dem Kriterium der Nähe zur Hardware bzw. der Nähe zur Anwendung zwischen Systemsoftware einerseits und Anwendungssoftware andererseits (vgl. Abbildung 2.2/1).

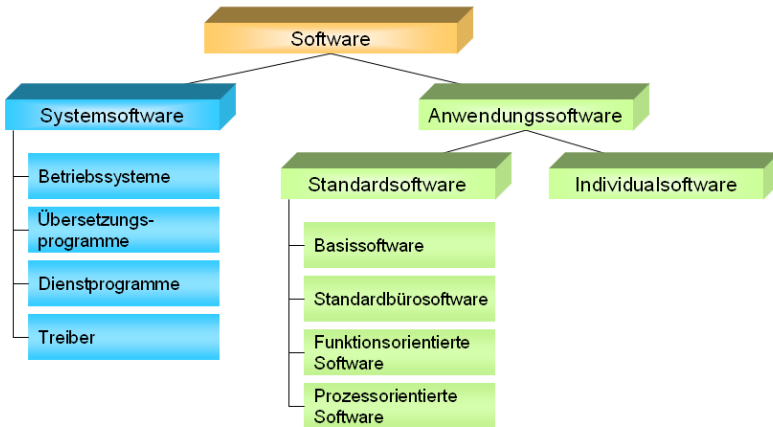


Abb. 2.2/1 Klassifizierung von Software

Eine zentrale Anforderung an *Systemsoftware* besteht darin, die Hardware einfacher nutzbar zu machen. Beispielsweise wäre es unwirtschaftlich, in jedem Anwendungsprogramm eine eigene Druckersteuerung vorzusehen, die Vorkehrungen für den Fall trifft, dass kein Papier mehr verfügbar ist. Darüber hinaus sind vielfältige weitere Verwaltungs- und Überwachungsleistungen zu erbringen, die im Rahmen der Systemsoftware unter dem Begriff Betriebssystem zusammengefasst werden.

Die *Anwendungssoftware* gliedert sich in zwei Klassen: Als *Standardsoftware* bezeichnet man Programme, die nicht für einen einzelnen Anwender, sondern für eine Vielzahl von Kunden mit gleichen oder ähnlichen Aufgaben produziert werden. Demgegenüber wird *Individualsoftware* (z. B. zur Steuerung einer Gepäckbeförderungsanlage) speziell auf den Bedarf eines Benutzers hin entwickelt und kann häufig ohne Anpassungen nicht von anderen Anwendern (andere Abteilungen oder Unternehmen) eingesetzt werden.

2.2.1 Systemsoftware

Die Systemsoftware umfasst neben dem Betriebssystem Übersetzungsprogramme (für verschiedene Programmiersprachen), Dienstprogramme (erfahrungsgemäß häufig gebrauchte Programme, z. B. zum Sortieren von Daten) sowie Treiber zur Kommunikation mit Peripheriegeräten.

2.2.1.1 Betriebssysteme

Das Betriebssystem (Operating System) hat die Aufgabe, die zunächst unabhängigen Komponenten (z. B. Zentraleinheit, Drucker, Tastatur) bei der Bewältigung eines Benutzerauftrags zu koordinieren. Betriebssysteme bilden die Schnittstelle zwischen einem Benutzer bzw. Anwendungsprogramm einerseits und der Hardware andererseits. Sie haben folgende Anforderungen zu erfüllen:

- Bereitstellen eines Systems zur Dateiverwaltung (siehe Kapitel 3)
- Verwaltung der Hardwarebetriebsmittel (Prozessor, Hauptspeicher, Peripheriegeräte)
- Administration der Benutzeraufträge und Überwachung der Programmabläufe
- Vorhalten einer grafischen (Graphical User Interface, GUI) oder textuellen Benutzungsschnittstelle

Moderne Betriebssysteme unterstützen das sog. Multitasking sowie teilweise auch den Multiuser-Betrieb. Durch *Multitasking* ist der Rechner in der Lage, Programme parallel auszuführen. Beispielsweise ist es möglich, einen Text zu bearbeiten, während die Maschine dann, solange sie auf die nächste Eingabe wartet, im Hintergrund eine Kalkulation durchführt. Darüber hinaus spricht man von *Multithreading*, wenn es ein Betriebssystem zulässt, dass in einem Programm ein Prozess aus mehreren Teilprozessen (Threads) besteht, die parallel ausgeführt werden können. *Multiuser-Betrieb* liegt vor, wenn von einem zentralen Rechner mehrere Terminals und damit mehrere Anwender quasi-parallel bedient werden. Beim *Singleuser-Betrieb* wird hingegen nur ein Nutzer versorgt.

Für Personal Computer werden zurzeit am häufigsten Betriebssysteme der Firma *Microsoft* (MS) verwendet, die sich zu einer Art inoffiziellen Standard entwickelt haben. Windows 7 gestattet bspw. Multitasking und bietet eine gemeinsame Nutzung von Ressourcen im Netz (siehe Abschnitt 2.6). *Unix-Systeme* erlauben den Multitasking- und den Multiuser-Betrieb. Zudem verfügen einige Varianten über eine integrierte Softwareentwicklungsumgebung. Der Terminus Unix suggeriert eine Einheitlichkeit, die aber so am Markt nicht auffindbar ist. Es existieren verschiedene Versionen und herstellereigene Implementierungen (z. B. Solaris von Sun Microsystems, AIX von IBM, MacOS X von Apple). Eine Besonderheit im Umfeld der Unix-Derivate sind *Linux-Betriebssysteme*, deren Quellcode im Gegensatz zu kommerziellen Systemen jedermann frei zugänglich ist (siehe auch Open-Source-Software, Abschnitt 5.1.1.2). Dies bietet z. B. Spezialisten die Möglichkeit, eigene Modifikationen vorzunehmen, das Programm auf Sicherheitsbedrohungen hin zu überprüfen und sich an der Weiterentwicklung des Betriebssystems zu beteiligen.

2.2.1.2 Programmiersprachen und Übersetzungsprogramme

Ein Rechner einschließlich Betriebssystem wird installiert, um den Anwender bei der Lösung seiner Fachaufgabe (z. B. Buchhaltung, Planung) zu unterstützen. Daher muss nun, aufbauend auf der Betriebssystemschnittstelle, ein AS konstruiert werden, das dies leistet. Die Gestaltung derartiger AS (wie auch des Betriebssystems selbst) erfolgt mittels Programmiersprachen. Unter einer Programmiersprache versteht man eine formale Sprache, mit der eine auf einer Hardware ablauffähige Software entwickelt werden kann. Sie wird häufig nach *Programmierparadigmen* klassifiziert. Die eindeutige Zuordnung

zu einem Paradigma gestaltet sich schwierig, da einzelne Programmiersprachen Merkmale verschiedener Ansätze aufweisen.

Mit *imperativen Sprachen* legen Programmierer fest, wie eine Aufgabe durchzuführen bzw. ein Problem zu lösen ist. Programme bestehen aus einer Menge von Anweisungen (Befehlen) und Ablaufstrukturen, die eine sequenzielle oder parallele Ausführung der Anweisungen, welche Zustände ändern, festlegen. Der Zustandsraum wird von konstanten und veränderlichen Werten aufgespannt. Untermengen bilden *prozedurale* und *objektorientierte* Programmiersprachen.

Bei *prozeduralen Sprachen* werden Daten- und Befehlsstrukturen getrennt entworfen. Die Programmierung erfordert entsprechende Kenntnis und Erfahrung. So ist es z. B. möglich, die Codierung in einer an der Fachsprache des jeweiligen Problembereichs ausgerichteten und damit weitgehend maschinenunabhängigen Form vorzunehmen (man spricht von einer problemorientierten Programmiersprache). Verbreitete prozedurale Programmiersprachen sind z. B. BASIC, C, COBOL und FORTRAN.

Objektorientierte(OO)-Programmiersprachen sehen hingegen ein Programm als eine Sammlung von Objekten an, die miteinander in Verbindung stehen und zum Zwecke der Problemlösung Nachrichten austauschen. Gleichartige Objekte gehören zu einer Klasse. Für jede Klasse wird festgelegt, welche Zustände die Objekte annehmen können und welche Änderungen des Objektzustands ausgeführt werden sollen, wenn Nachrichten eintreffen. Die Änderungen werden durch Methoden ausgelöst, die direkt an die zugehörigen Objekte gebunden sind. OO-Sprachen unterstützen den Abstraktionsprozess durch die Klassifikation und Kapselung von Objektdaten mit ihrer Verarbeitung. Durch das Konzept der Vererbung bieten sie die Möglichkeit zur Wiederverwendung von Softwarekomponenten und damit zu einer Produktivitätssteigerung der Softwareentwicklung. Ein Beispiel eines betriebswirtschaftlichen Objekts ist eine Rechnung, die (in der Nomenklatur klassischer Systeme) eine Datenstruktur besitzt (z. B. Rechnungskopf mit dem Empfänger der Ware und Rechnungspositionen mit Materialnummer, Auslieferungsmenge und Preis). Gleichzeitig verbindet der Anwender z. B. mit dem Begriff Rechnungsposition die Verfahren, die zugelassen sind, um eine solche zu erzeugen bzw. zu verändern (z. B. Fakturierung). Zur Bereitstellung von OO-Konzepten im Bereich der Programmiersprachen werden zwei unterschiedliche Ansätze verfolgt. Der eine besteht darin, Programmiersprachen auf Basis des objektorientierten Paradigmas zu entwickeln; sie werden als objektorientierte Sprachen bezeichnet. Beispiele hierfür sind Java (siehe Abschnitt 2.3.1) und C# (gesprochen „C sharp“). Alternativ ist die Erweiterung traditioneller Programmiersprachen um OO-Konzepte zu beobachten. Vertreter dieser sog. Hybridsprachen sind z. B. C++ (die objektorientierte Erweiterung von C) und VISUAL BASIC (enthält seit Version 4.0 Konzepte der Objektorientierung) [Eicker/Nietsch 99].

Von den imperativen lassen sich die *deklarativen* Programmiersprachen unterscheiden. Eine wichtige Eigenschaft besteht darin, dass der Benutzer nicht mehr formulieren muss, WIE ein bestimmtes Problem zu lösen ist, sondern lediglich angibt, WAS gelöst werden soll. Zu den deklarativen Sprachen zählen Abfragesprachen für Datenbanksysteme wie beispielsweise SQL (Structured Query Language).

Die Hardware eines Rechners ist jedoch nicht unmittelbar in der Lage, Anweisungen einer Programmiersprache (*Quellcode*) zu „verstehen“. Sie müssen zunächst mithilfe eines Übersetzungsprogramms in Maschinensprache, d. h. in eine binäre Darstellung mit der Trennung von Befehlen und Daten, übersetzt werden (*Objektcode*), welche die Hardware unmittelbar verarbeiten kann.

Die Übersetzung eines Quellcodes in einen Objektcode erfolgt durch einen Compiler oder einen Interpreter. *Compiler* übersetzen das gesamte Quellprogramm „in einem Stück“ (Batch). Sie prüfen vor der Übertragung das vorliegende Programm auf Syntaxfehler, z. B. ob nach einer „Klammer auf“ auch die „Klammer zu“ folgt. Im nächsten Schritt wird das Programm übersetzt (kompiliert). Es ist dann jedoch noch nicht lauffähig, sondern muss erst um Hilfsprogramme (z. B. zur Ein- und Ausgabesteuerung), die in Bibliotheken abgelegt sind, erweitert werden. *Interpreter* erzeugen dagegen keinen archivierbaren Objektcode. Vielmehr wird jeder Befehl einzeln abgearbeitet, d. h. immer wieder neu übersetzt und sofort ausgeführt. Dieses Vorgehen bietet Vorteile für die interaktive Programmentwicklung. So kann man etwa die Richtigkeit einzelner ausgeführter Programmschritte verfolgen.

2.2.1.3 Dienstprogramme und Treiber

Dienstprogramme ermöglichen die Abwicklung systemorientierter, häufig wiederkehrender, anwendungsneutraler Aufgaben. Dazu zählen etwa Sortier- und Suchroutinen sowie systemorientierte Hilfsprogramme. Letztere erfüllen Funktionen wie etwa (benutzungsfreundliches) Kopieren von Dateien, Datensicherung oder Änderung der Speicherorganisation (z. B. dahin, dass die durchschnittliche Zeit zum Auslesen eines erfragten Inhalts minimal wird). Sie sind als Bestandteil sowohl innerhalb der Systemsoftware (z. B. Programme zur Defragmentierung in MS Windows) als auch der Basissoftware (z. B. Norton Utilities) zu finden.

Transaktionssysteme sind spezielle Hilfsprogramme, welche die Abwicklung von formalisierten und meist kurzen Verarbeitungsvorgängen im Dialog sichern. Bei der Buchung eines Fluges durch ein Reisebüro ist es bspw. notwendig, dass zwischen der Suche nach einem freien Flug und dessen Buchung niemand auf diesen zugreifen kann. Unter Umständen wird hier auch mit verteilten Datenbanken (siehe Abschnitt 3.1.3) gearbeitet, sodass das Transaktionssystem die konsistente Eingabe aller Informationen in allen Datenbeständen gewährleisten muss, bevor der Vorgang abgeschlossen wird.

Bei Auftreten eines Fehlers muss jede Eintragung rückgängig gemacht werden.

Auch die in einer Software implementierten Treiber können als Dienstprogramme interpretiert werden. Unter einem Treiber (Driver) versteht man ein Programm, das als Übersetzer zwischen den Protokollen (siehe Abschnitt 2.6.1) verschiedener Funktionseinheiten oder einer Programm- und einer Funktionseinheit fungiert. Zum Beispiel werden die von einem Rechner an einen Drucker gesendeten Signale durch den Treiber vorher in ein dem Drucker verständliches Format umgewandelt.

2.2.2 Anwendungssoftware

Ein *Anwendungsprogramm* (engl. „Application Software“) wird verkürzt oft als „Anwendung“ bezeichnet. In der Alltagssprache hat sich außerdem die etwas unglücklich gewählte Bezeichnung „Applikation“ eingebürgert. Hier- von ist des Weiteren der Begriff „App“ abzugrenzen, welcher eine Kurzform von Application darstellt. Mit Apps sind i. d. R. Anwendungsprogramme für Smartphones (siehe Abschnitt 2.4) gemeint, welche über einen Onlineshop (bspw. dem App Store der Firma Apple) bezogen und direkt auf dem Smartphone installiert werden können.

2.2.2.1 Standardsoftware

Standardsoftware umfasst Produkte, die für den Massenmarkt konzipiert wurden. In der Regel werden sie mit Selbstinstallationsroutinen ausgeliefert und ermöglichen oft nur geringe, bei komplexeren Produkten (z. B. funkti- onorientierter Software) jedoch auch größere Anpassungen (Customizing, siehe Abschnitt 5.1) an die individuellen Bedürfnisse.

2.2.2.1.1 *Basissoftware*

Basissoftware stellt grundlegende Funktionalitäten zur Verfügung, die unab- hängig von spezifischen Arbeitsgebieten genutzt werden. Wesentliche Ele- mente sind E-Mail (inkl. Adressverwaltung), Virens Scanner, Komprimierungs- programme (zur Minimierung der Größe einer Datei und damit der Übertra- gungsdauer zwischen Sender und Empfänger) und Browser.

Als Browser werden allgemein Programme bezeichnet, die eine Suche nach Dateien und deren Platzierung in der Verzeichnishierarchie ermöglichen (z. B. Windows Explorer). Die Visualisierung erfolgt i. d. R. über Baum- strukturen. Wird ein Browser darüber hinaus zur audiovisuellen Darstellung von HTML-Seiten im World Wide Web (WWW) (siehe Abschnitt 2.6.2) verwendet, so spricht man von einem Webbrowser (z. B. Mozilla Firefox). Der Benutzerzugriff erfolgt durch die Angabe einer URL (Uniform Resource Locator, bspw.: <http://www.is-frankfurt.de>).

2.2.2.1.2 *Standardbürosoftware*

Zur Standardbürosoftware zählen Programme zur Textverarbeitung (etwa MS Word), zum Erstellen von Präsentationen (z. B. MS PowerPoint), zur Tabel-

lenkulation (z. B. MS Excel), zur Datenbankverwaltung (siehe Kapitel 3) und Editoren für Webseiten (auch HTML-Editoren, siehe Abschnitt 2.6.2).

Darüber hinaus sind am Markt integrierte Standardbürosoftwarepakete verfügbar, die Textverarbeitung, Tabellenkalkulation, grafische Bearbeitung und auch eine Datenbank unter einer einheitlichen Benutzungsoberfläche anbieten (z. B. MS Office oder OpenOffice).

2.2.2.1.3 Funktionsorientierte Software

Als funktionsorientierte Standardsoftware werden Lösungen bezeichnet, die aus betriebswirtschaftlicher Sicht eine Funktion oder funktionsübergreifend mehrere Anwendungsbereiche (z. B. Vertrieb, Materialwirtschaft, Produktion, Finanzwesen und Personalwirtschaft) und deren Prozesse unterstützen. In letzterem Fall spricht man von funktionsübergreifender integrierter Standardsoftware, welche sich in Module gliedert, die auf eine gemeinsame Datenbasis zugreifen [Keller 99]. Dieser Aufbau bietet aus Sicht des Anwenders den Vorteil, dass er Software nur für die von ihm benötigten Problemstellungen betreiben muss. Er kann also z. B. Module für die Durchlaufterminierung und den Kapazitätsausgleich im Rahmen der Produktionsplanung und -steuerung erwerben, ohne die Werkstattsteuerung anschaffen zu müssen (siehe Abschnitt 4.4.1.3). Der modulare Aufbau ermöglicht zudem eine schrittweise Einführung neuer Systeme und somit ein langsames Ablösen von Altsystemen. Die Anpassung einer solchen Standardsoftware an spezifische Einsatzbedürfnisse in Unternehmen erfolgt durch das Customizing (siehe Abschnitt 5.1), ohne dass eine Veränderung des Quellprogramms stattfinden muss. Darüber hinaus werden auch Schnittstellen für individuelle Erweiterungen angeboten.

Eine Ausprägung funktionsorientierter und funktionsübergreifender Software sind sog. *ERP-Systeme* (Enterprise-Resource-Planning-Systeme). Der Begriff ist sehr verbreitet, aber unglücklich gewählt, da diese Systeme gerade beim Umgang mit knappen Ressourcen, z. B. Produktionsengpässen, oft Schwächen aufweisen. Die beiden weltweit größten Anbieter kommerzieller ERP-Systeme sind SAP und Oracle.

2.2.2.1.4 Prozessorientierte Software

Die Grenze zwischen prozessorientierter und funktionsübergreifender Software ist fließend. In erstgenannten Systemen sind Prozesse quer durch unterschiedliche Funktionsbereiche eines Unternehmens zu integrieren. Die Realisierung erfolgt häufig unter Verwendung von zentralen Datenbanken. Sog. *Workflow-Management-Systeme* (WMS, siehe Abschnitt 4.2.2) unterstützen durch verschiedene Funktionalitäten die Beschreibung und Modellierung von Geschäftsprozessen, z. B. Vorgänge zur Erstellung und Abgabe eines Angebots in der chemischen Industrie.

Systeme zur Unterstützung verteilten Arbeitens sind bspw. *Workgroup-Support-Systeme*. Diese werden im Gegensatz zu den WMS zumeist bei der Bearbeitung einer relativ unstrukturierten Aufgabe eingesetzt. Die Kooperati-

on basiert auf Netzwerkarchitekturen mit zugehörigen Kommunikationssystemen, wie:

- Konferenzplanungssystemen: Terminvereinbarung, Ressourcenverwaltung (Besprechungsräume, Präsentationsgeräte),
- Computerkonferenzsystemen: Diskussionen zwischen räumlich getrennten Personen (z. B. Videokonferenzsystem),
- Gruppenentscheidungsunterstützungssystemen (Mehrbenutzerumgebungen, z. B. zur gezielten Kompromissfindung bei Verhandlungen) und
- Mehrautorensystemen (Co-Authoring): Werkzeuge zur gleichzeitigen Bearbeitung von Dokumenten (Texten, Plänen, Konstruktionszeichnungen, Grafiken) durch mehrere Teammitglieder.

2.2.2.2 Individualsoftware

Unter Individualsoftware versteht man AS, die für eine spezielle betriebliche Anforderung mit der zugehörigen Hard- und Softwareumgebung individuell angefertigt wurden. Die Individualsoftware wird entweder selbst produziert oder fremdbezogen (zu Kriterien für diese Entscheidung siehe Abschnitt 6.3). Die Eigenentwicklung kann sowohl von der IV-Abteilung als auch von den entsprechenden Fachabteilungen, dort i. d. R. mit deklarativen Sprachen (siehe Abschnitt 2.2.1.2), durchgeführt werden. Aufgabe ist hier, die Entwicklung von Anwendungssoftware als Einzelfertigung technisch und finanziell zu beherrschen (siehe Abschnitt 5.1).

Wegen der hohen Entwicklungskosten von Individualsoftware ist ein zunehmender Trend hin zu Standardsoftware zu beobachten. Demgegenüber wird der Einsatz von Individualsoftware u. a. durch ein unzureichendes Funktionspektrum der Standardsoftwarelösungen für bestimmte Problemstellungen (z. B. zur Steuerung von Aluminiumwalzwerken) begründet.

2.3 Ausgewählte Integrationstechniken

2.3.1 Komponentenarchitekturen

Die zunehmende Modularisierung von AS, die aus Gründen der erhöhten Wiederverwendbarkeit und leichten Veränderbarkeit der Bausteine verfolgt wird, lässt die Grenze zwischen Individual- und Standardsoftware schwinden. Komponentenbasierte Anwendungssysteme werden aus einzelnen Bausteinen individuell zusammengestellt. Eine *Softwarekomponente* ist ein Codebaustein mit Schnittstellen, Attributen (Eigenschaften) und Verhalten (Funktionalitäten). So können die Komponenten zwar als Standardsoftware bezeichnet werden, da sie jedoch erst in einer spezifischen Zusammenstellung die gewünschte Funktion erfüllen, ist das resultierende AS keine Standardsoftware im eigentlichen Sinne mehr.

Die Integration der Komponenten zu AS erfolgt in Komponentenarchitekturen (auch: Komponentenframeworks). Diese spezifizieren einerseits, wie

Schnittstellen der Komponenten aufgebaut sein müssen. Andererseits bieten sie eine Plattform als Laufzeitumgebung (Funktionalität zur Ausführung von Maschinencode), die den Betrieb des AS – also das konsistente Zusammenspiel der Komponenten – steuert und verwaltet sowie u. a. Sicherheitsmechanismen, Datenbankverbindungen, Benutzungsschnittstellen und die Speicherverwaltung bereitstellt. Aus Entwicklersicht besteht die Aufgabe darin, die Anwendungslogik (auch: Geschäftslogik oder Prozesslogik) in Form spezifischer Kombinationen vorgefertigter Bausteine zu programmieren (oder fremd zu beziehen, siehe Abschnitt 6.3). Es ist nicht erforderlich, die Komponenten auf dem gleichen Rechner zu betreiben.

Zwei weit verbreitete Architekturen bzw. Plattformen zur Entwicklung komponentenbasierter AS sind die Java Platform Enterprise Edition (Java EE) von Sun Microsystems und das .NET-Framework von Microsoft.

Java EE bietet ein Komponentenmodell auf Basis der Programmiersprache Java an: Die Anwendungslogik wird in Enterprise Java Beans (EJB) gekapselt. Als Laufzeitumgebung bietet Java EE einen sog. Container, der die Komponenten verwaltet und z. B. die Kommunikation mit Benutzungsschnittstellen ermöglicht.

Das *.NET-Framework* stellt ein mit Java EE vergleichbares Konzept dar, lässt jedoch die Entwicklung in zahlreichen Programmiersprachen zu. Dafür ist dieses Framework jedoch nicht plattformneutral wie Java, sondern auf eine Anwendung im Umfeld von Microsoft-Betriebssystemen ausgerichtet.

2.3.2 Webservices und Serviceorientierte Architekturen

Mit Webservices wird das Konzept der komponentenbasierten Softwareerstellung weiterentwickelt hin zu weltweit verteilten und völlig voneinander losgelösten Anwendungsmodulen. Webservices lassen sich verstehen als autonome, gekapselte Dienste, die eine genau definierte Funktion erfüllen und über das Web als Teile übergreifender Wertketten verwendet werden können. Haben zwei AS eine Webservice-Schnittstelle, so können sie über standardisierte Internetprotokolle (siehe Abschnitt 2.6.1) miteinander kommunizieren. Auf diese Weise ist es u. a. möglich, Geschäftsprozesse abzuwickeln, die durch mehrere AS ausgeführt werden, z. B. die Buchung einer Reise, die individuell aus Flug, Hotel und Mietwagen in den AS unterschiedlicher Anbieter zusammengesetzt wird (siehe Abschnitt 1.1.1).

Auf Basis von Webservices lassen sich wiederum Serviceorientierte Architekturen (SOA) definieren. SOA bezeichnet eine Systemarchitektur für eine plattform- und sprachenneutrale Nutzung und Wiederverwendung verteilter Dienste, die von unterschiedlichen Besitzern verantwortet werden. Der Entwurf von Services orientiert sich hierbei nicht mehr (wie z. B. Softwarekomponenten im Rahmen einer Komponentenarchitektur) vorrangig an technischen Gesichtspunkten, sondern vielmehr an der Funktionalität im Hinblick auf die zu unterstützenden Funktionen und Prozesse. Ein Service kann bspw. die im Rahmen der Auftragsdatenerfassung durchzuführende Bonitätsprüfung

des Kunden über eine Kreditauskunftei sein. Ziel ist es, dass einem Unternehmen ermöglicht wird, als Antwort auf geänderte geschäftliche Anforderungen durch die Reorganisation von Services schnelle und kostengünstige Anpassungen der AS-Landschaft vorzunehmen, ohne jede benötigte Funktionalität neu und selbst implementieren zu müssen [Buhl u. a. 08]. Als zentrale Infrastrukturkomponente für die Kommunikation zwischen Anbietern und Nutzern von Webservices dient im Rahmen einer SOA häufig ein sog. Enterprise Service Bus (ESB).

2.3.3 Kommunikationsstandards

Die Grundlage jeglicher Interaktion und Koordination betrieblicher Aufgaben und Prozesse ist eine effiziente Kommunikation. Damit diese funktionieren kann, müssen sich Sender und Empfänger einer Nachricht ex ante auf eine gemeinsam genutzte Sprache bzw. einen Kommunikationsstandard einigen.

Zum Informationsaustausch hat sich die *Extensible Markup Language* (XML, siehe auch Abschnitt 2.6.2) als Standardstrukturierungssprache etabliert. XML ist eine textbasierte Meta-Auszeichnungssprache, die es ermöglicht, Daten bzw. Dokumente bezüglich Inhalt und Darstellungsform derart zu beschreiben und zu strukturieren, dass sie – v. a. auch über das Internet – zwischen einer Vielzahl von Anwendungen in verschiedensten Hardware- und Softwareumgebungen ausgetauscht und weiterverarbeitet werden können. Dokumente zur Unterstützung von Geschäftsprozessen (wie etwa Bestellung oder Fakturierung), die in XML geschrieben sind und bei denen man sich auf eine inhaltliche Struktur geeinigt hat, können so (z. B. im Rahmen des Electronic Data Interchange, EDI) von Systemen verschiedener Geschäftspartner automatisch verarbeitet werden [Weitzel u. a. 01].

Bei *Electronic Data Interchange* (EDI) handelt es sich um den transaktionsbegleitenden Datenaustausch von strukturierten und standardisierten Geschäftsdokumenten zwischen den Computersystemen der Handelspartner mit einem Minimum an personellen Eingriffen. Das Hauptziel von EDI ist die effiziente Kommunikation mit Geschäfts- bzw. Kooperationspartnern, unabhängig von den verwendeten IV-Systemen. Da eine EDI-Kommunikation im Idealfall vollautomatisch abläuft, müssen für die korrekte Interpretation und Weiterverarbeitung einer Nachricht ihre Syntax und Semantik genau festgelegt werden. Auf der Syntaxebene wird definiert, welche Zeichen für eine Nachricht verwendet werden dürfen und nach welchen Regeln diese Zeichen zu komplexeren Konstrukten zusammengefasst werden können. Auf der semantischen Ebene wird den Zeichen bzw. Zeichengruppen eine inhaltliche Bedeutung zugeordnet. Für den EDI-Einsatz gibt es (neben XML) verschiedene *Nachrichtenstandards*, die nach ihrer Reichweite bezüglich Wirtschaftszweig und nationaler Gültigkeit unterschieden werden können. Die wichtigsten Nachrichtenstandards sind ANSI X.12 und EDIFACT.

2.4 Rechnerklassen

Für die Gestaltung der betrieblichen Rechner- und Netzinfrastruktur sind neben dem Mikrocomputer (PC) weitere Rechnerklassen relevant, von denen die wichtigsten im Folgenden vorgestellt werden.

Der *Großrechner* (Host oder auch Mainframe) bietet durch seine großen Rechen- und Speicherkapazitäten eine hohe Verarbeitungsgeschwindigkeit im Multiuser-Betrieb an. In größeren Unternehmen werden oft mehrere Hosts in einem Netz verbunden, z. B. um hohe Leistungsbedarfe der Anwender befriedigen zu können oder eine gewisse Sicherung gegenüber Systemausfällen zu erhalten. Neuinstallationen von Großrechnersystemen werden zunehmend zugunsten von PC-Netzen in *Clustern* verworfen, wenn diese entsprechende Leistungen zu erbringen versprechen.

Workstations sind prinzipiell als selbstständige Arbeitsplatzrechner konzipiert, deren Leistungsfähigkeit zunächst unterhalb von Großrechnern einzuordnen ist. Die Leistung von Workstations kann durch die Verwendung im Verbund – typischerweise vernetzt zu einem Local Area Network (LAN, siehe Abschnitt 2.5.3.1) – in sog. Workstation-Farmen zur Lastverteilung auf momentan freie Kapazitäten erhöht werden.

Als eine weitere Rechnerklasse werden häufig *Netzwerkcomputer* (NC) und *Thin Clients* diskutiert. Dies sind preisgünstige Rechner mit einer geringeren Leistungsfähigkeit, die man speziell für den (Client-)Betrieb in Netzen (siehe Abschnitt 2.5.2) konzipiert hat. NC bzw. Thin Client nutzen über das Netz AS, die auf einem entfernten Server ablaufen. Im Idealfall kommt ein solches System ohne Festplatten aus. Durch die zentrale Administration (z. B. in einem Rechenzentrum) werden zudem die Kosten für die Pflege der Systeme reduziert.

Mobile Endgeräte werden in einer großen Vielfalt und in unterschiedlichen Formen angeboten, entsprechen jedoch in ihrer Funktion den beiden nachfolgenden Beispielen: Personal Digital Assistants (PDAs) sind meist etwa handflächengroße Geräte mit multiplen Fähigkeiten, die sich mit der Funktionsfähigkeit eines Notebooks annähernd vergleichen lassen. PDAs können i. d. R. über Kabel oder Bluetooth mit einem Netzwerk verbunden werden und verfügen über einen Browser zur Darstellung von Internetseiten. Smartphones sind hybride Geräte, die PDA-Funktionalitäten in ein Mobiltelefon integrieren. Im Unterschied zu PDAs steht bei der Konzeption von Smartphones die daten- oder sprachorientierte Kommunikation im Vordergrund.

Embedded Systems sind spezialisierte Computer, welche Teil eines größeren Systems oder eines Gerätes darstellen und gewisse Aktivitäten in ihrer Umgebung steuern. Charakteristisch für diese Systeme ist, dass sie nicht in erster Linie als Computer wahrgenommen werden. Sie sind i. d. R. derart spezialisiert, dass sie kein Betriebssystem benötigen, sondern nur Anwendungsprogramme zur Erfüllung ihrer Funktion beinhalten. Viele Geräte für den alltäg-

lichen Gebrauch sind bereits mit solchen Systemen ausgestattet. Ein Beispiel für Embedded Systems sind Antilockiersysteme (ABS) für Kraftfahrzeuge.

2.5 Rechnernetze und Netzarchitekturen

An sich unabhängig arbeitsfähige Rechner können über Kommunikationspfade miteinander zu einem Rechnernetz verbunden werden, um mehrere Entscheidungsträger (Menschen oder Maschinen) in gemeinsame verteilte Dispositions- oder Planungsprozesse einzubinden. Beispiele sind verschiedene Formen der zwischenbetrieblichen Integration (z. B. elektronischer Datenaustausch im Rahmen des Supply-Chain-Managements (SCM) in Abschnitt 4.8) oder der Zugriff auf externe Datenbanken (z. B. bei der Patentrecherche). Mit dem Einsatz von Rechnernetzen werden verschiedene Ziele verfolgt, so z. B. die bessere Ausnutzung von Kapazitäten sowie der parallele Zugriff auf im Netz verfügbare Daten, Programme oder Hardwareressourcen.

Eine relativ neue Technologie stellt das sogenannte *Grid Computing* dar, bei dem Nutzer oder AS Zugriff auf einen großen Pool von heterogenen, vernetzten IT-Ressourcen haben. IT-Ressourcen können in diesem Zusammenhang z. B. Server, Speicher, CPUs, Datenbanken oder Services sein [Foster/Kesselman 99, Berman u. a. 03].

2.5.1 Komponenten von Rechnernetzen

Die wichtigsten Komponenten eines Rechnernetzes sind:

- die Rechner selbst, einschließlich der physischen Netzwerkanbindung (Netzwerkkarte oder Modem) sowie der jeweiligen Betriebs-, Netz- und Anwendungssoftware
- die Verbindungs- und Kommunikationskomponenten in und zwischen Netzen (Hubs, Switches, Router)
- die Datenübertragungswege
- die Protokolle

Verbindungs- und Kommunikationskomponenten bezeichnen spezielle Geräte, deren Aufgabe in der Einbindung von Rechnern in Netze, der Verknüpfung von Netzen sowie hierauf aufbauend der intelligenten Weiterleitung von Datenpaketen liegt. Man bezeichnet sie häufig als Vermittlungsstelle oder Vermittlungsknoten. Hubs und Switches sind die zentralen Punkte in einem lokalen Netzwerk (siehe Abschnitt 2.5.3.1), die Rechner miteinander verbinden. Die Verbindung erfolgt über eine Reihe von Anschlüssen, sog. Ports. Die in einen Port eingehenden Datenpakete werden innerhalb des Hub kopiert und anschließend an alle weiteren Ports und damit Rechner übertragen. Router können unterschiedliche Netztypen miteinander verbinden; bspw. kann ein lokales Netzwerk an das Internet angeschlossen werden.

Daten werden auf Datenübertragungswegen (Leitungen oder Funkstrecken) übermittelt. Die gängigsten Übertragungskanäle sind verdrehte Kupferkabel,

Glasfaserkabel, Radiowellen (Mobilfunk, Wireless LAN (WLAN) und Bluetooth sowie Infrarot- und Laserwellen (optischer Richtfunk)).

Protokolle definieren sämtliche Vereinbarungen und Verfahren, die zur Kommunikation zwischen Rechnern beachtet werden müssen. Die in der Praxis am weitesten verbreitete Protokollfamilie TCP/IP (Transmission Control Protocol/Internet Protocol) spielt v. a. im Internet eine große Rolle (siehe Abschnitt 2.6.1).

2.5.2 Client-Server-Konzept als Kooperationsmodell

Die Kommunikation zwischen Rechnern setzt die Existenz eines geeigneten Kooperationsmodells voraus, das im Hinblick auf die Partner eine eindeutige Rollenverteilung festlegt und die gemeinsamen Protokolle spezifiziert. Im Client-Server-Konzept versuchen auf der Benutzerseite sog. Clients, von einem bestimmten Rechner im Netz (Server) angebotene Dienste (z. B. Daten und Transaktionen eines AS) in Anspruch zu nehmen. Aufgaben des Clients sind die Präsentation der entsprechenden Daten und die Interaktion mit dem Benutzer. Dieses Kooperationsmodell lässt sich auch mehrstufig umsetzen, so können etwa Datenbank- und Applikationsserver auf unterschiedlichen Rechnern implementiert werden, um die Arbeitslast zu verteilen. Die Clients nehmen einen Dienst des Applikationsservers in Anspruch, der wiederum die benötigten Daten von einem Datenbankserver erfragt.

In großen Netzwerken dienen verschiedene Rechner häufig als Clients und Server zugleich, was als Peer-to-Peer-Kommunikation (Kommunikation unter Gleichgestellten) bezeichnet wird (siehe Abschnitt 4.4.5.2).

2.5.3 Netzklassen

2.5.3.1 Lokale Netze

Befinden sich die miteinander vernetzten Rechner in einem Büro, einem Haus oder einem Betriebsgelände, so spricht man von einem lokalen Netz (Local Area Network, LAN). Dieses wird häufig von unternehmenseigenen Netzabteilungen betrieben. In nicht kabelgebundenen LANs (Wireless Local Area Network, WLAN) können mobile Endgeräte wie Notebooks mittels Funktechnik über stationär installierte „Access Points“ in einem Netz kommunizieren. Sie sind in der Regel an ein (kabelgebundenes) LAN angeschlossen.

2.5.3.2 Weitverkehrsnetze

Geografisch weit auseinander liegende lokale Rechner oder Rechnernetze können über Weitverkehrsnetze (Wide Area Network, WAN) miteinander verbunden werden. Wir unterscheiden zwischen geschlossenen WANs mit Zugangssicherungsverfahren für spezielle Benutzergruppen und öffentlichen WANs wie dem Internet (siehe Abschnitt 2.6). Als technische Infrastruktur

nutzt man Kabel- und Funkverbindungen, die innerhalb verschiedener (Netz-)Dienste Anwendung finden.

Integrated Services Digital Network (ISDN) ist ein digital arbeitender Telekommunikationsdienst auf Basis des herkömmlichen Telefonnetzes zur Übertragung digitalisierter Sprache und Daten in Bitform. Durch die gleichzeitige Nutzung verschiedener Kanäle können Telefongespräche, Telefax und Datenübertragung parallel abgewickelt werden. Mit *Asymmetric Digital Subscriber Line* (ADSL) wird ein Verfahren der Datenübertragung bezeichnet, das im Vergleich zu ISDN eine vielfach höhere Übertragungsrate zwischen Vermittlungsstelle und Teilnehmerendeinrichtung (z. B. PC, Workstation) ohne Änderung der Kabelinfrastruktur zur Verfügung stellt.

Die Nutzung von *Mobilfunknetzen* basiert auf eigens dafür entwickelten Systemen. Das weltweit erfolgreichste Mobilfunksystem ist das Global System for Mobile Communications (GSM). Die (mobilen) Funknetze der ersten und zweiten Generation bauen auf den Architekturen traditioneller Telefonnetze auf und sind daher vor allem für den leitungsvermittelten Sprachdienst konzipiert. Durch den enormen Erfolg des Internets erhöhte sich auch die Nachfrage nach paketvermittelnden Technologien im Mobilfunk (auf Paketvermittlung wird in Abschnitt 2.6.1 näher eingegangen). General Packet Radio Service (GPRS) stellt einen Zwischenschritt hin zu einer flexiblen und leistungsfähigen Datenübertragung in Mobilfunknetzen dar. Universal Mobile Telecommunications System (UMTS) ist die Technologie der dritten Generation, die mobilen Endgeräten durch neue Übertragungsverfahren, wie dem High Speed Downlink Packet Access (HSDPA), breitbandige Datenübertragung ermöglicht, sodass auch multimediale Inhalte, wie etwa Videoclips, übertragen werden können.

Als *Hochleistungsnetz* oder Backbone werden zentrale Übertragungstrecken bezeichnet, die Daten aus unterschiedlichen Subnetzen bündeln und weiterleiten. Sie verfügen über hohe Übertragungskapazitäten und garantieren den reibungslosen nationalen bis transkontinentalen Datenverkehr.

2.6 Weltweite Vernetzung: Das Internet

Das Internet bezeichnet den Zusammenschluss tausender lokaler Netzwerke, bestehend aus Millionen Rechnern, die Informationen über die Protokollfamilie TCP/IP (siehe Abschnitt 2.6.1) austauschen. Darüber hinaus bietet es eine Reihe von Diensten und Techniken, die nicht nur seine Funktionalität sichern, sondern auch vielfältige Impulse für private Netzbetreiber geben. Die Entwicklung des Internets kennzeichnet das Bestreben, durch Verbindung von Netzen den jederzeitigen Zugriff auf weltweit verfügbare Informationsressourcen preiswert zu eröffnen, um Kooperationsvorteile zu ermöglichen.

2.6.1 Protokollfamilie TCP/IP

Die Protokollfamilie TCP/IP setzt sich aus zwei Teilen, dem Transmission Control Protocol (TCP) und dem Internet Protocol (IP), zusammen. Das TCP zerlegt Nachrichten, z. B. eine E-Mail, in verschiedene Datenpakete und versieht jedes Datenpaket mit der IP-Adresse des Senders und Empfängers.

IP-Adressen sind Zifferncodes, die zur Identifikation von Informationsstandorten dienen. In der aktuell noch verbreiteten Protokollversion IPv4 haben sie eine Länge von 32 Bit (4 Byte), werden in Form von vier durch Punkte getrennten Dezimalzahlen angegeben und können Werte aus dem Intervall von 0 bis 255 annehmen. Für Menschen ist es i. Allg. leichter, mit Namen anstelle von Zahlenkolonnen umzugehen. Der sog. Domain Name Service (DNS) übersetzt diesen Namen in die zugehörige IP-Adresse (z. B. `www.wiwi.uni-frankfurt.de` in `141.2.196.151`).

Abbildung 2.6.1/1 verdeutlicht die *Datenübertragung im Internet*. Die Pakete werden an einen Router geschickt (z. B. an den des Internetproviders), dessen Aufgabe in der IP-gesteuerten Weiterleitung der Informationen liegt. Innerhalb des Routernetzwerks versuchen z. B. Telefongesellschaften, momentane Belastungstaler in der verfügbaren Streckeninfrastruktur aufzufüllen, indem ein Paket über den am wenigsten ausgelasteten Weg in Richtung Ziel geleitet wird. Jedes Datenpaket einer Nachricht kann einen anderen Weg im Internet nehmen (man spricht von einem Packet Switching Network). Am Ziel werden die Pakete – gesteuert durch TCP – in die ursprüngliche Reihenfolge gebracht.

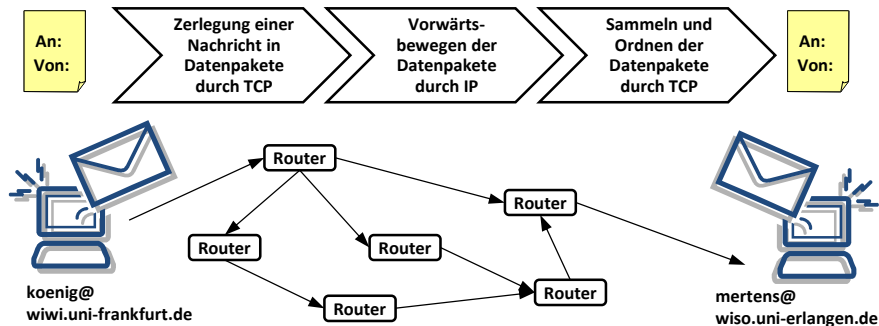


Abb. 2.6.1/1 Datenübertragung im Internet

Jeder Dienst über das TCP/IP-Protokoll nutzt fest im Netzwerkprotokoll spezifizierte Ports zur Kommunikation. Dieser Zusatz erlaubt es, dass mehrere AS über eine Internetverbindung gleichzeitig Daten austauschen können. Anhand der Portnummer erkennt das System, für welches AS die ein- und ausgehenden IP-Pakete bestimmt sind. Die Kombination aus IP-Adresse und Port ermöglicht die eindeutige Identifizierung des Dienstes auf einem spezifizierten Rechner.

Da die Anzahl der verfügbaren IPv4-Adressen nahezu ausgeschöpft ist, soll eine neue Protokollversion, die als IPv6 bezeichnet wird, die 32-Bit-Version

des IPv4 schrittweise ablösen. Mit der Version 6 werden 128 Bits für die Adressierung verwendet, was einer Anzahl von $3,4 * 10^{38}$ Adressen entspricht. Im Gegensatz zu IPv4 bezeichnet man die Adressen bei IPv6 in Form von acht durch Doppelpunkte getrennten 16-Bit-Werten in hexadezimaler Schreibweise (z. B. 2BA:0:66:899:0:0:459:AC39). Neben der Erweiterung des Adressraums soll IPv6 das Routing vereinfachen und zu einer höheren Datensicherheit beitragen sowie die Reservierung von Ressourcen, etwa für eine dauerhafte Verbindung, ermöglichen.

2.6.2 Dienste und Technologien der Vernetzung

Das Internet verfügt heute über eine Vielzahl von Diensten, die es einem Anwender ermöglichen, Informationen zu empfangen bzw. diese zu senden. Zu den populärsten Diensten zählt das World Wide Web (WWW).

Das *Hypertext Transfer Protocol* (HTTP) ist das Standardprotokoll des WWW. Über dieses Protokoll werden die Webseiten übertragen. Der Webbrowser stellt einen HTTP-Client dar, der Anfragen generiert und diese an einen Webserver sendet. Der Server enthält einen sog. HTTP-Daemon, der auf HTTP-Anfragen wartet und diese verarbeitet. Ein solches System bearbeitet diese anschließend und sendet das gewünschte Dokument zurück.

Zentraler Baustein von Webanwendungen sind in der *Hypertext Markup Language* (HTML) geschriebene Dokumente. Der Begriff Hypertext bezeichnet die Verknüpfung von Wörtern oder Textabschnitten mit anderen Informationsquellen. Durch Anklicken eines solchen Verweises (Link) kann das referenzierte Dokument aufgerufen werden. HTML-Editoren, wie etwa Microsoft Expression Web oder Adobe Dreamweaver, sind Programme zur Aufbereitung von Texten und Grafiken, die im WWW veröffentlicht werden sollen.

Durch den Erfolg des WWW sind die Grenzen des HTML-Konzepts vielfach sichtbar, da z. B. die inhaltliche Struktur der ausgegebenen Daten nicht expliziert und damit deren Weiterverarbeitung erschwert wird. Eine Lösung dieses Problems bringt die *Extensible Markup Language* (XML, siehe Abschnitt 2.3.3). Diese Metasprache eröffnet die Möglichkeit, Daten im Netz so zu beschreiben, dass auch die zugrunde liegende Datenstruktur an ferne Anwender und deren IV-Systeme übermittelt wird. XML erlaubt es, die Inhalte und ihre Struktur von der Darstellung (Layout) zu trennen, sodass ein Dokument für unterschiedliche Endgeräte jeweils grafisch angemessen visualisiert werden kann (z. B. PC-Monitor vs. Handy-Display).

Für client-seitige Anwendungen ermöglicht Java die Entwicklung von *Applets*, die als portable Programme vom Server auf den Client übertragen und dort im Browser ausgeführt werden. Dies ermöglicht auch die Verlagerung der Ressourcenbeanspruchung (Prozessor, Speicher) vom stark beanspruchten Server auf die Client-Rechner.

Neben dem WWW und HTTP existieren weitere anwendungsbezogene Dienste auf der Grundlage von TCP/IP wie z. B. FTP (File Transfer Protocol)

für die Dateiübertragung oder Voice-over-IP zur Übertragung von digitalisierten Sprachinformationen über das Netz.

Sozio-technische Weiterentwicklungen des WWW hin zu einer aktiven Einbindung von (ursprünglich zumeist nur passiv konsumierenden) Internetnutzern v. a. in die Inhaltsgenerierung werden unter dem Begriff *Web 2.0* zusammengefasst. Zentrale Techniken sind in diesem Zusammenhang die asynchrone Datenübertragung zwischen Server und Browser mittels Ajax (Asynchronous JavaScript and XML) sowie die XML-basierten Datenspezifikationen RSS (Real Simple Syndication) und Atom, welche Nutzern das gezielte Abonnieren von Web-Inhalten ermöglichen. Vor dem Paradigma des Web 2.0 unterstützen *Social-Software-Anwendungen* wie z. B. Wikis, Weblogs oder Social-Bookmarking-Dienste die menschliche Kommunikation, Interaktion und Zusammenarbeit [Kollmann/Häsel 07, Back u. a. 08].

2.6.3 Intranets und Extranets

Die beschriebenen Internettechniken sowie die vielfach kostenfreie Verfügbarkeit entsprechender Software werden zunehmend interessant für den breiten Einsatz im Unternehmen – in sog. Intranets und Extranets.

Intranets sind geschlossene Netze auf der Basis von TCP/IP und den darauf aufsetzenden Protokollen und Diensten. Der Aufbau von Intranets ist insbesondere aus Gründen der Integration mit den Diensten im Internet attraktiv, sodass Anwender beide Netze mit der gleichen Oberfläche benutzen können. Häufig bietet man interne Handbücher, Rundbriefe, Adressverzeichnisse, Organisationsrichtlinien und nicht-öffentliche Teilekataloge in Intranets an. Bestehen Schnittstellen zwischen einem geschlossenen Netz und dem Internet, so werden üblicherweise *Firewalls* implementiert, die den internen Bereich vom öffentlichen Netz abschotten.

Ein Extranet bezeichnet demgegenüber ein geschlossenes Netz von über das Internet verbundenen Unternehmen mit entsprechenden Zugriffsrechten (z. B. die Zulieferunternehmen eines Automobilherstellers oder die eines Produzenten mit seinen Logistikpartnern). Wie Intranets basieren Extranets auf der Nutzung von Internettechniken. Eine verbreitete Technik sind *Virtual Private Networks* (VPNs), in welchen über ein Tunneling-Protokoll Informationen beim Übergang vom privaten LAN in das öffentliche Netz verschlüsselt und beim Eintreffen am Empfangspunkt entsprechend decodiert werden. Darüber hinaus kann diese Technik auch in Intranets zum Einsatz kommen.

2.6.4 Rechner- und Netzinfrastrukturen

Unternehmen und andere netzbetreibende Organisationen setzen, logisch gesehen, aus den vorgestellten Bausteinen ihre Rechner- und Netzinfrastruktur zusammen und verbinden diese mit dem Internet. Bei großen Betrieben verläuft die Entwicklung einzelner Beschaffungs- und Erweiterungsentscheidungen im Zuge der zunehmenden Integration von Betriebswirtschaft und Technik in vielen Fällen ausgehend von zentralen Großrechnern zu de-

zentralen Architekturen. Dabei führen kurzfristige Einflüsse bisweilen dazu, dass derartige Systemstrukturen unkoordiniert wachsen. Einen Beitrag zur gezielten Entwicklung kann die Anwendung von IV-Architekturmodellen leisten (siehe Abschnitt 6.1.2.2).

PRAKTISCHES BEISPIEL

Die *comdirect bank AG* ist Marktführer unter den Online-Brokern Deutschlands und die führende Direktbank für Anleger. Wertpapiergeschäfte (Brokerage) und andere Bankgeschäfte (z. B. Kreditvergabe und Beratung, siehe auch Abschnitt 4.4.3) unter einem Dach stellen für mehr als 1,3 Millionen Kunden ein komplettes Leistungsangebot dar. Im Brokerage werden 90 % aller Aufträge über das Internet abgewickelt. Im Zentrum steht die Website www.comdirect.de mit dem Transaktionssystem *Direct Brokerage*, über das die Kunden der Direktbank Wertpapieraufträge erteilen können.

Neben dem *Direct-Brokerage-System* stellt die *comdirect bank* ihren Kunden den größten europäischen Börseninformationsdienst im Internet, den sog. *Informer* der Firma *Interactive Data Managed Solutions* in Frankfurt am Main, zur Verfügung (integriert in <http://www.comdirect.de>). Dieses System ist über Standleitungen mit den Zentralsystemen der *Commerzbank* verbunden, z. B. für das Konsolidieren der Konten (vgl. Abbildung 2.6.4/1), und bietet z. B. kostenlose aktuelle Informationen zum weltweiten Börsengeschehen und unterschiedliche Oberflächen für Anfänger und Experten. Der *Informer* bediente in 2008 über das Internet ca. 540 Mio. Seitenabfragen (page impressions) pro Quartal. In der Spitzenlast beantwortet das System ca. 250.000 Datenbankabfragen pro Minute.

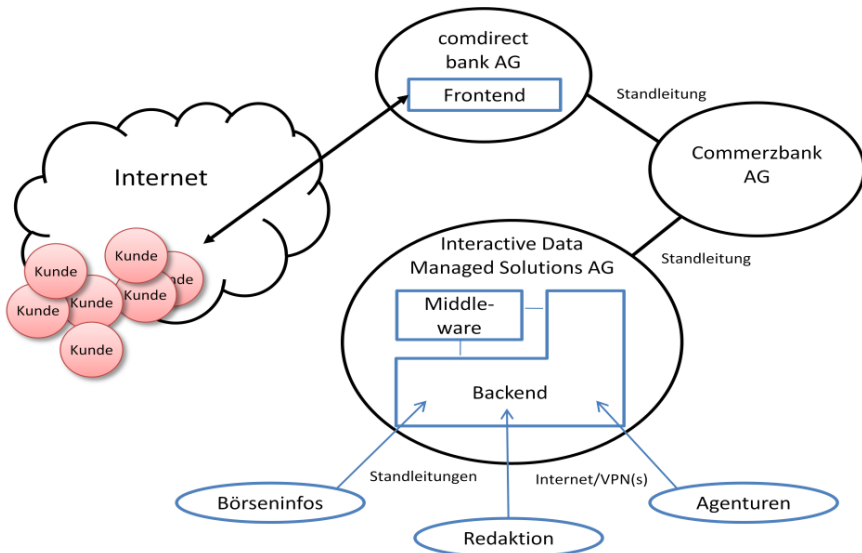


Abb. 2.6.4/1 Ausschnitte aus der Rechner- und Netzarchitektur der *comdirect bank AG*

Die Leistungen werden von den beiden Häusern gemeinschaftlich erbracht. *Interactive Data Managed Solutions AG* liefert alle im öffentlichen Bereich des *Informer-Angebots* zugänglichen Informationen über das Netzwerk der *Commerzbank AG* an die *comdirect bank AG*. Dort werden die Informationen, angereichert und zu voll-

ständigen Seiten zusammengesetzt, an den Kunden ausgeliefert. Neben zwischen den Häusern redundant ausgelegten Standleitungen mit jeweils 34 Mbit verfügt die *comdirect* hierfür über zwei 200 MBit-Anbindungen an das Internet. Die Anwender greifen auf Frontend-Server zu, welche die Anwendungslogik enthalten und sich zur Seitenerstellung zusätzlicher Backend-Funktionen bedienen (z. B. standardisierte Kursabfragen oder individuelle Marktübersichten). Der Datenaustausch zwischen den Anwendungsebenen wird hierbei über sog. *Middleware* (siehe Abschnitt 4.2.4) ermöglicht.

Die Anwendungssoftware läuft unter dem Betriebssystem *Linux* (siehe Abschnitt 2.2.1.1) und ist komplett eigenerstellt, um die hohen Durchsatzanforderungen zu befriedigen. Da die Dienste *Direct Brokerage* und *Informer* im Webbrowser integriert sind, ist für den Anwender die Trennung der beiden Systeme nicht erkennbar. Das Rechenkongingent der *comdirect bank* ist in der Spitze zu etwa 50 % ausgelastet. Die Transaktionsleistung wird von einem nach Bedarf ausbaufähigen Zusammenschluss von momentan 250 Intel-basierten Hochleistungsservern erbracht, die ebenfalls über ein Breitbandnetz kommunizieren.

2.7 Sicherheit

Funktionierende IV-Systeme sind Voraussetzung für einen reibungslosen Geschäftsbetrieb. Aus diesem Grund sind Sicherheitsmaßnahmen in Bezug auf die Nutzung von Hard- und Software unerlässlich. Aus Organisations-sicht ist das Personal zu sensibilisieren und zu schulen. Um sich gegen einen Stromausfall abzusichern, müssen Notstromaggregate genutzt werden. Der Einsatz doppelter Systeme, der im Ganzen oder durch einzelne Komponenten erfolgt, wie z. B. gespiegelte Festplatten oder doppelte Festplattencontroller, vermindert das Ausfallsrisiko. Ferner sollte ein Katastrophenhandbuch ausgearbeitet werden, welches das Vorgehen im Notfall (z. B. Brand im Rechenzentrum) beschreibt und die notwendigen Schritte umfasst, um basierend auf den bisherigen Sicherheitsmaßnahmen den Betrieb der IV bzw. Unternehmung binnen kürzester Zeit fortzuführen.

Im Hinblick auf die Sicherheit in Informations- und Kommunikationsnetzen ist ein Schutz vor Viren u. ä. schädlichen Programmen notwendig. Um unerwünschte Werbe-Mails (Spam) und die damit verbundenen Sicherheitsrisiken zu vermeiden, müssen Mail-Filter definiert werden. Externe Angreifer werden durch Firewalls abgehalten. Ebenso ist die Software durch Zugriffskontrollen zu sichern, indem eine Identifikation der Benutzer, die Überprüfung der Benutzerrechte sowie die Protokollierung der Aktivitäten stattfinden. Dies dient sowohl dem Datenschutz als auch der Sicherung von Geschäftsgeheimnissen.

Kryptografische Verfahren dienen dazu, die Geheimhaltungs- und Authentifizierungsziele zu erreichen. Man kann hierbei symmetrische und asymmetrische Verschlüsselungsmethoden unterscheiden.

Bei der *symmetrischen Verschlüsselung* wird eine Nachricht durch den Sender mit einem Schlüssel chiffriert und beim Empfänger durch die umgekehrte

Anwendung desselben Schlüssels dechiffriert. Ein Problem ist, dass zuvor Sender und Empfänger den Schlüssel über einen sicheren Kanal transportieren müssen.

Bei der *asymmetrischen Verschlüsselung* dagegen hält jeder Kommunikationsteilnehmer ein eng aufeinander bezogenes Schlüsselpaar (bestehend aus einem öffentlichen Schlüssel und einem privaten Schlüssel), wobei sich der eine Schlüssel nicht ohne Weiteres aus dem anderen (bspw. durch Umkehrung) herleiten lässt. Wenn ein Absender A eine Nachricht an Person B schicken will, chiffriert er diese mit dem öffentlichen Schlüssel von B. Die Dekodierung der Nachricht wird von B dann mittels seines geheimen Schlüssels durchgeführt. Bei diesem Verfahren muss folglich keine Verabredung im Sinne eines „abhörsicheren“ Austauschs von Schlüsseln getroffen werden. Ein Beispiel findet sich in [Buchmann 03].

In Bezug auf externe Dienstleister sind notwendige Leistungen vertraglich durch Service Level Agreements (siehe dazu auch Abschnitt 5.1.1.3) festzuschreiben, um das notwendige Maß an Sicherheit, Verfügbarkeit, Wiederanlauf und Reaktionszeiten zu gewährleisten.

2.8 Literatur zu Kapitel 2

- | | |
|---------------------|--|
| Back u. a. 08 | Back, A., Gronau, N. und Tochtermann, K. (Hrsg.), Web 2.0 in der Unternehmenspraxis, München 2008. |
| Berman u. a. 03 | Berman, F., Fox, G. und Hey, A., Grid Computing: Making the Global Infrastructure a Reality, New York 2003. |
| Buhl u. a. 08 | Buhl, H. U., Heinrich, B., Henneberger, M. und Krammer, A., Service Science, WIRTSCHAFTSINFORMATIK 49 (2008) 2, S. 129-132. |
| Buchmann 03 | Buchmann, J., Einführung in die Kryptographie, 3. Aufl., Heidelberg 2003. |
| Eicker/Nietsch 99 | Eicker, S. und Nietsch, M., Standards zum objektorientierten Paradigma, WIRTSCHAFTSINFORMATIK 41 (1999) 4, S. 358-370. |
| Foster/Kesselman 99 | Foster, I. und Kesselman, C., The Grid: Blueprint for a New Computing Infrastructure, San Francisco 1999. |
| Kollmann/Häsel 07 | Kollmann, T. und Häsel, M. (Hrsg.), Web 2.0 – Trends und Technologien im Kontext der Net Economy, Wiesbaden 2007. |
| Keller 99 | Keller, G., SAP R/3 prozessorientiert anwenden. Iteratives Prozess-Prototyping mit Ereignisgesteuerten Prozessketten und Knowledge Maps, 3. Aufl., München 1999. |
| Weitzel u. a. 01 | Weitzel, T., Harder, T. und Buxmann, P., E-Business und EDI mit XML, Heidelberg 2001. |