

MODUL 150

E-BUSINESS-APPLIKATIONEN ANPASSEN

ARBEITSBLATT 2

Markus Nufer

Handlungsziele

Mit diesem Aufgabenblatt werden die Handlungskompetenzen gemäss **Handlungsziel 1** in der Modulidentifikation¹ vertieft und abgeschlossen. Der Modulbaukasten mit den Modul-Identifikationen wird durch die Ausbildungsbetriebe definiert.

HZ1 «Aufbau der Applikation, Transaktionskonzept, Applikationsumgebung und Rahmenbedingungen (Sicherheit, Performance, Verfügbarkeit, Transaktionsvolumen, usw.) erfassen.»

Mit den einzelnen Aufgaben im Aufgabenblatt sollen die **Kenntnisse und Handlungsfähigkeiten** gemäss den Vorgaben «Handlungsnotwendige Kenntnisse 1.1, 1.2 und 1.3» erworben werden.

- 1.1 Kennt mögliche Architekturen von Web-Applikationen.
- 1.2 Kennt wesentliche Elemente zur Gewährleistung der Sicherheit bei Web-Applikationen.
- 1.3 Kennt Kriterien, welche die Performance und Verfügbarkeit einer Web-Applikationen beeinflussen.

Inhalt

Handlungsziele.....	1
Thema.....	2
Architekturen von Web-Applikationen.....	2
Weg zur Anwendungsarchitektur	2
Software-Architektur.....	3
Plattform-Architektur.....	4
Kommunikationsarchitektur	6
Kommunikation und Kommunikationsservices	8
Das Transaktions-Konzept.....	13
Das ACID Prinzip	14
Integrität einer Datenbank	15
Sicherheit von E-Business Anwendungen	16
Arbeitsblatt.....	17
Zielsetzung	17
Arbeitsform.....	17
Zeitbudget	17
Aufgabe für die Lernenden.....	17
Arbeitsergebnis (Werkstück) Kompetenznachweis	18
Ergänzender Lesestoff.....	18

¹ Die Handlungsziele sind in der Modul-Identifikation der ICT Berufsbildung Schweiz definiert:
<https://cf.ict-berufsbildung.ch/modules.php?name=Mbk&a=20101&cmodnr=150&noheader=1>

Thema

Mit dieser Aufgabe wird der Aufbau von E-Business-Applikationen behandelt und vertieft. Es geht dabei einerseits um die innere Struktur von solchen Anwendungen und andererseits um die Anforderungen an die Infrastruktur, welche für den Betrieb von E-Business Anwendungen nötig ist.

Architekturen von Web-Applikationen

Mit der Bearbeitung des Arbeitsblatt 1 wurde klar, dass die E-Business Anwendungen die verschiedenen Unternehmen bei Ihrer strategischen Aufgabe «Gewinn erzielen» unterstützen. Die Vorgaben an die Anwendungsarchitektur und die einzelnen Anwendungen entstehen als Ergebnis aus dem Strategie-Prozess des Unternehmens.

Zur gesamten Architektur eines ICT Systems gehören neben der Software / Anwendungen auch die Hardware. Oft werden in der Plattformarchitektur auch die Software Service-Systeme berücksichtigt: Datenbank, Anwendungsplattform, Kommunikation

Weg zur Anwendungsarchitektur

Die wirtschaftliche Bereitstellung (Entwicklung / Beschaffung und Betrieb) der benötigten ICT-Unterstützung kann als Grundauftrag für die Informatikorganisation verstanden werden. Damit dieser Auftrag möglichst effektiv und effizient abgewickelt werden kann ist die IT-Architektur ein zentraler Hebel, um diese betriebswirtschaftliche Zielsetzung zu erreichen.

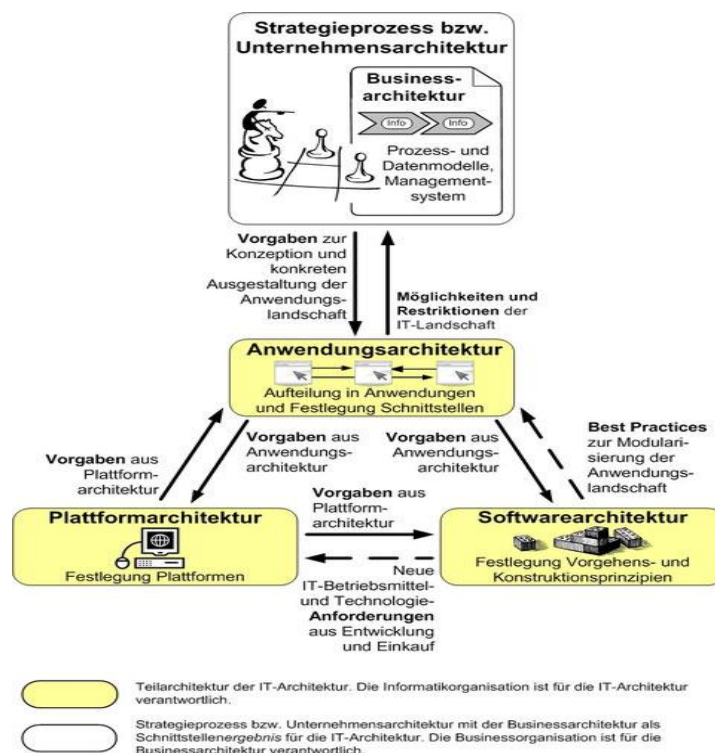


Abb. Entwicklung einer Anwendungsarchitektur

Zur Anwendungsarchitektur (Application Architecture) gehören die WEB- und E-Business Anwendungen und wie damit die Geschäftsabwicklung resp. die Geschäftsprozesse² unterstützt

² Vgl Module 254 «Geschäftsprozesse beschreiben»

werden. In vielen Organisationen und Unternehmungen gehört die Anwendungsarchitektur als Bestandteil in die Enterprise Architecture.

Die Vorgaben aus dem Strategieprozess gliedern sich in Funktionale und Nichtfunktionale Anforderungen.

Unter «**Funktionalen Anforderungen**» oder fachliche Architektur kann verstanden werden:

- User-Interface
- Transaktionsabwicklung (Auftragserfassung, Auftragsbearbeitung, Kundenverwaltung resp. Lieferantenverwaltung)
-

Die Funktionalen Anforderungen resp. die daraus entwickelte fachliche Architektur bestimmt die Anforderungen an die technische Architektur.

Zu den «**Nichtfunktionalen Anforderungen**» resp. der technischen Architektur gehören:

- Verfügbarkeit
- Sicherheit
- Stabilität
- Logging
- Audithing
- Referenzdaten
- Persistenz (Beständigkeit)
- Transaktions-Management

In der Detaillierung der Anwendungsarchitektur wird der Aufbau der Teilsysteme und Softwareschichten anhand ihrer elementaren Bausteine und Beziehungen. Bausteine sind insbesondere Softwarekomponenten und Objektklassen, die über Schnittstellen miteinander in Beziehung stehen.

Software-Architektur

Die **Softwarearchitektur**³ gehört als Teilbereich zur IT Architektur. Sie beschreibt die grundlegenden Komponenten und deren Zusammenspiel innerhalb eines Softwaresystems. Die Softwarearchitektur wurde erst in den 1990er Jahren ein unabhängiges Teilgebiet der Softwaretechnik.

Die **Software-Architektur** zerlegt ein Gesamtsystem in seine Komponenten und bereitet die Informationen auf, die zwischen diesen Komponenten ausgetauscht werden müssen. Ein Ziel dabei ist es, die Kopplung der Komponenten möglichst zu minimalisieren. Darüber hinaus wird deren Verteilung auf die zu verwendenden Hardware- und Software-Ressourcen abgebildet. Das heisst: In der Software-Architektur sind der Bauplan und die Konstruktionsregeln eines Software-Systems enthalten.

Die Softwarearchitektur betrieblicher Anwendungssysteme umfasst die Systemsoftware OS mit den Datenbanksystemen, die Middleware mit dem Applikations-Subsystem (Transaktionssystem) als Verbindung zwischen den verschiedenen Anwendungen resp. den Anwendungen und den Subsystemen wie beispielsweise Datenbank- oder Kommunikationssystem. Als Basis für die Konstruktionsregeln einer Software-Architektur dienen die Grundformen Client-Server-, WEB- oder Serviceorientierte Architektur.

Die Systemsoftware und die Middleware wird heute fast ausschliesslich aus Standardprodukten zusammengestellt.

³ Quelle: <https://de.wikipedia.org/wiki/Softwarearchitektur>

Plattform-Architektur

Eine Plattform – auch Schicht oder Ebene genannt – bezeichnet in der Informatik eine einheitliche Grundlage, auf der Anwendungsprogramme ausgeführt und entwickelt werden können. Für die Komponente, welche die Plattform nutzt, ist die Komponente darunter nicht sichtbar. Daher kann dieselbe Komponente über eine Plattform auf verschiedenen „Untergründen“ betrieben werden. Es gibt eine Vielzahl von Plattformen im Informatikbereich.⁴

Die Plattformarchitektur beschreibt die für die Realisierung und den Betrieb von Anwendungen zur Verfügung stehende technische Infrastruktur. Sie ist in einer modernen ICT-Architektur weitestgehend von der Anwendungsarchitektur entkoppelt und mit der Software-Architektur eng verzahnt. Die Software-Architektur nimmt eine eigentliche Brückenfunktion zwischen Anwendung und Plattform ein.

Die Anforderungen aus der Anwendungsarchitektur führen zur technischen Architektur eines Systems, oder eben zur Plattformarchitektur. Ziel der Plattform-Architektur ist es, das Zusammenspiel der Systemkomponenten unter verschiedenen Aspekten sicher zu stellen. Diese Plattform soll die verschiedenen Services (Core Services) integrieren und als Service Host für heterogene Daten und Informationen dienen.

Als **Core Services** (Kerndienste) werden die Mehrwert-Dienste einer Plattform bezeichnet. Diese stehen allen Partnern über vordefinierte Schnittstellen und mit standardisierten Prozessen zur Verfügung. Es handelt sich dabei entweder um generische domänenspezifische Dienste (Georeferenzierung, Geokodierung, Routenplanung, Nachbarschaftssuche etc.) oder um Querschnittsdienste (Authentifizierung, Autorisierung, Service Management, Service Level Agreement, Service Monitoring, Service Billing, Onboarding, Offboarding, etc.), also um Dienste, die grundsätzlich relevant sind.

Beispiele dieses hart umkämpften Markts der Plattformen zeigen sich bei **den IaaS- und PaaS-Plattformanbietern** deutlich. Diese Dienste müssen hochgradig effizient und kostengünstig erbracht und gleichzeitig mit hoher Dynamik weiterentwickelt werden. Dabei kommen innovative softwaretechnische Methoden zum Einsatz (Serviceorientierte Architekturen, modellbasierter Werkzeuge, Micro Services, introspektiver Architekturen, datenzentrierte Big Data Architekturen, etc.).

Als Beispiel einer standardisierten käuflichen Plattform dient die «IBM TRIRIGA Application Platform».

IBM TRIRIGA Application Platform

Unternehmenssoftware-Anwendungen sind n-schichtig, wobei "n" eine beliebige Anzahl von Verarbeitungsschichten bzw. -ebenen darstellen kann. IBM® TRIRIGA Application Platform⁵ ist ein logisches vierschichtiges System: Web, Anwendung (und Prozess), Tools und Datenbank.

Durch die Aufteilung einer Anwendung in logische Schichten werden auch die Funktionen der einzelnen Schichten voneinander getrennt. Jede physische Schicht beschreibt die tatsächliche Hardwarekonfiguration, die einzelnen Geräte und die Funktion jedes Geräts.

⁴ Quelle: [https://de.wikipedia.org/wiki/Plattform_\(Computer\)](https://de.wikipedia.org/wiki/Plattform_(Computer))

⁵ Quelle: https://www.ibm.com/support/knowledgecenter/de/SSHEB3_3.5.1/com.ibm.tap.doc/ins_overview/c_ovw_platform_architecture.html

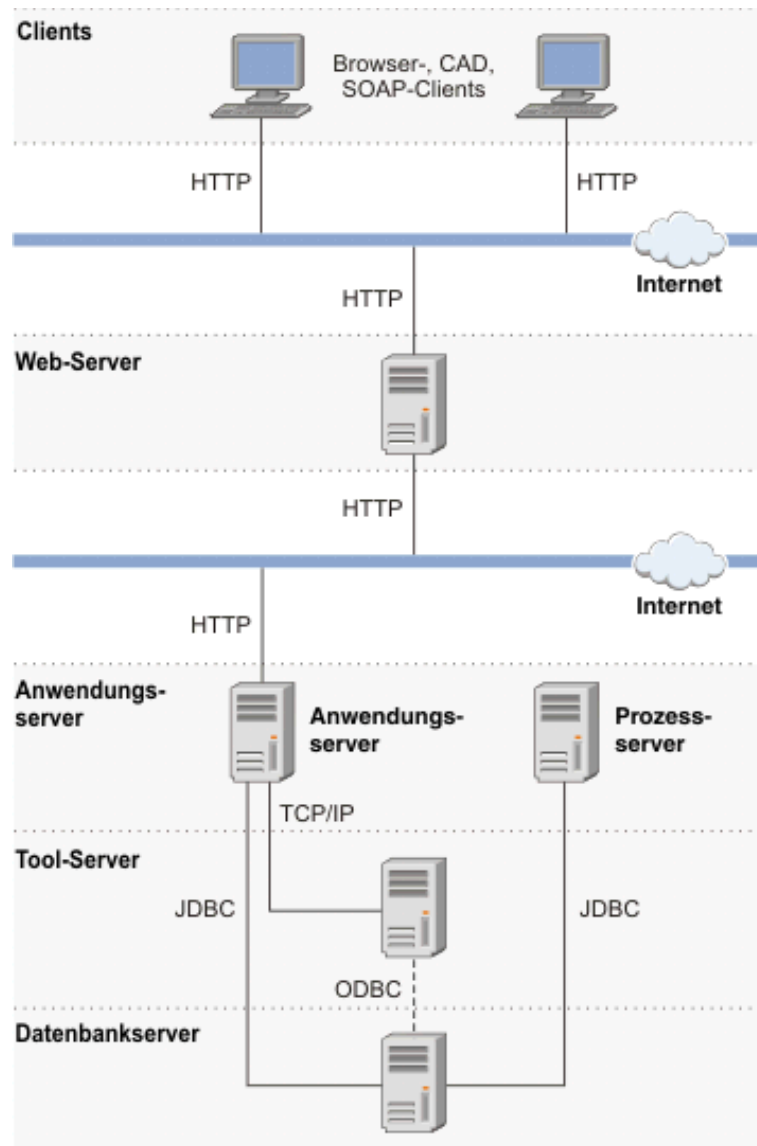


Abb. Diagramm der IBM TRIRIGA Plattformarchitektur

Die folgenden logischen funktionsbasierten Server werden in IBM TRIRIGA Application Platform verwendet:

Web-Server

Empfängt HTTP-Anforderungen für Webinhalte. Dieser Server wird auch als Front-End-Server bezeichnet.

Anwendungsserver

Führt die benutzerdefinierte Geschäftslogik mit WebLogic Server oder WebSphere Application Server aus.

Prozessserver

Führt die Hintergrundverarbeitung und Analyse mit WebLogic Server oder WebSphere Application Server durch.

Tool-Server

Führt die Berichterstellung durch.

Datenbankserver

Enthält die relationale Datenbank und ein unterstütztes Datenbankmanagementsystem wie Oracle Database, IBM DB2-Server oder Microsoft SQL Server.

Kommunikationsarchitektur

Wie im Arbeitsblatt 1 schon grundsätzlich behandelt, wird das Thema Kommunikation in der Kommunikationsarchitektur definiert. In diesem Arbeitsblatt lernen wir die typischerweise verwendeten Bausteine dazu kennen.

In Netzen wird zwischen zwei Schichtenmodellen unterschieden: dem Internet (DoD-Schichtenmodell⁶) und dem OSI-Schichtenmodell unterschieden

Das Internet Schichtenmodell (DoD)

Es besteht aus insgesamt vier Schichten:

- **Anwendungsschicht - Application Layer**
Kommunikation zwischen Applikationen über das Internet.
Beispiel-Protokolle: Hypertext Transfer Protocol - HTTP, File Transfer Protocol - FTP, Simple Mail Transfer Protocol - SMTP.
- **Transportschicht - Transport Layer**
Schicht zur Ermöglichung von gesichertem Datentransport mit Flusskontrolle (keine Überflutung des Empfängers, Wiederholung der Sendung bei Timeout, zuverlässiger Bytestrom, Absicherung der Übertragung durch Sequenznummern und Prüfsummenbildung mit Empfangsquittungen, Festlegung von Portnummern für Anwendungen) oder ungesicherter Datenübertragung.
Beispiel-Protokolle: TCP - Transport Control Protocol (gesicherter Transport mit Kontrolle), User Datagram Protocol - UDP (ungesicherter Transport).
- **Internetschicht - Internet Layer**
Schicht zur Erzeugung und Versendung der Datenpakete mit Hilfe der IP-Adresse des Absenders und Empfängers, Wegsteuerung eines Paketes, Festlegung der Lebensdauer (Time-To-Live - TTL).
Beispiel-Protokolle: Internet Protocol - IP (Datenübertragung), Internet Control Message Protocol - ICMP (für Kontrolle und Fehlersuche)
- **Netzzugangsschicht - Network Access Layer**
Schicht für die Datenübertragung von direkt miteinander verbundenen Rechnern unter Berücksichtigung der Auflösung einer logischen IP-Adresse in eine MAC-Adresse (MAC - Media Access Control - Kennung der Netzkarte), von Übertragungsmedium, Zugriffsprotokoll (Ethernet, Token Ring, ...), usw.
Beispiel-Protokolle: Address Resolution Protocol - ARP (Auflösung der IP in MAC-Adresse und umgekehrt), Ethernet, Token Ring

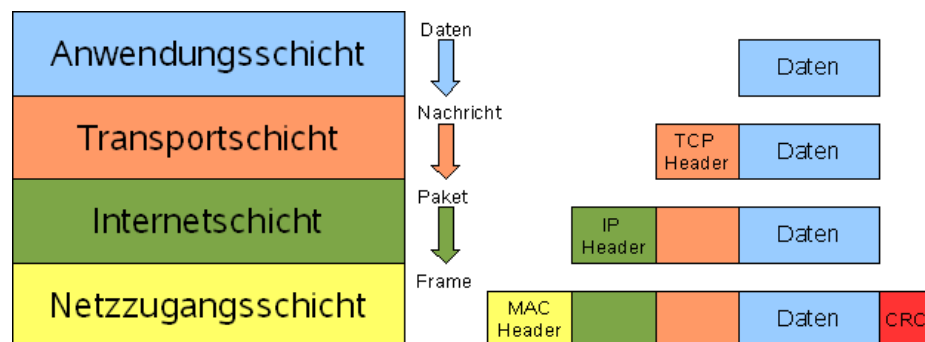


Abb. Internet Schichtenmodell

⁶ DoD-Schichtenmodell: Abk. DoD für US Department of Defense

Das OSI Layer-Modell

Das OSI-Modell (englisch Open Systems Interconnection Model) ist ein Referenzmodell für Netzwerkprotokolle als Schichtenarchitektur. Es wird seit 1983 von der International Telecommunication Union (ITU) und seit 1984 auch von der International Organization for Standardization (ISO) als Standard veröffentlicht. Seine Entwicklung begann im Jahr 1977.

OSI (Open Source Interconnection) 7 Layer Model				
Layer	Application/Example	Central Device/Protocols	DOD4 Model	
Application (7) Serves as the window for users and application processes to access the network services.	End User layer Program that opens what was sent or creates what is to be sent Resource sharing • Remote file access • Remote printer access • Directory services • Network management	User Applications SMTP	G A T E W A Y	Process
Presentation (6) Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.	Syntax layer encrypt & decrypt (if needed) Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation	JPEG/ASCII EBDIC/TIFF/GIF PICT		
Session (5) Allows session establishment between processes running on different stations.	Synch & send to ports (logical ports) Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.	Logical Ports RPC/SQL/NFS NetBIOS names		
Transport (4) Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.	TCP Host to Host, Flow Control Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing	Filters TCP/SPX/UDP	Can be used on all layers	Host to Host
Network (3) Controls the operations of the subnet, deciding which physical path the data takes.	Packets ("letter", contains IP address) Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting			Internet
Data Link (2) Provides error-free transfer of data frames from one node to another over the Physical layer.	Frames ("envelopes", contains MAC address) (NIC card — Switch — NIC card) (end to end) Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control	Switch Bridge WAP PPP/SLIP		Network
Physical (1) Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.	Physical structure Cables, hubs, etc. Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts	Hub	Land Based Layers	

Abb. OSI Layer-Modell (Quelle: <http://www.escotal.com/osilayer.html>)

Auch das DoD Internet Architecture Model⁷ ist auch eine Netzwerk-Schichtenarchitektur. In diesem Vierschichten-Modell werden die einzelnen Aufgaben bei der Datenübertragung in aufeinander aufbauende Schichten eingeteilt (siehe auch TCP/IP-Referenzmodell). Für jede Schicht gibt es eine Reihe von Protokollen (siehe Internetprotokollfamilie), die die Aufgaben der jeweiligen Schicht auf unterschiedliche Weise lösen. Das Modell wurde Ende der 1960er Jahre von der DARPA für das Verteidigungsministerium der Vereinigten Staaten (Department of Defense, kurz: DoD) entwickelt und bestand zunächst aus vier Schichten.

Nr.	Schicht	Beispiele
4	Process	Telnet, SMTP, FTP
3	Host-to-Host	TCP, UDP
2	Internet	IP, IPX
1	Network Access	Ethernet, Token Ring, V.24

Abb. DoD Schichtenmodell

Das OSI Schichtenmodell beschreibt die Kommunikation wesentlich präziser als das DoD.

Eine lesenswerte Übersicht ist im Artikel⁸ über das OSI-Schichtenmodell in der Wikipedia verfügbar.

- **Anwendungsschicht**

Die Anwendungsschicht stellt die Anwendungs- und Dienstprogramme bereit, über die der

⁷ <https://de.wikipedia.org/wiki/DoD-Schichtenmodell>

⁸ https://de.wikipedia.org/wiki/OSI-Modell#Die_7_Ebenen

Endbenutzer die gewünschten Funktionen anstossen kann. Sie stellt den Anwendungen verschiedene Funktionalitäten zur Verfügung

- **Darstellungsschicht**

Die Darstellungsschicht bringt die Daten aus der Anwendungsschicht in ein einheitliches rechnerunabhängiges Datenformat, mit dem der Informationsaustausch über das Netz erfolgt, d.h. wandelt systemabhängige Datendarstellung in eine unabhängige Form um, und sorgt für Datenkompression und Verschlüsselung

- **Kommunikationsschicht, Sitzungsschicht**

Die Kommunikationsschicht, auch Sitzungsschicht genannt, baut die Kommunikationsbeziehung zwischen dem Rechner und dem WEB-Sbserver auf und sorgt für die Prozesskommunikation zwischen den zwei Systemen. Diese Schicht behandelt auch Sitzungsabbrüche

- **Transportschicht**

Die Transportschicht teilt die zu übertragenden Daten, in mehrere Pakete auf. Jedes Paket enthält die Protokollinformationen (Header) der übergeordneten Schichten und wird um das TCP (Transmission Control Protokoll) dieser Schicht erweitert, welches die Reihenfolge der Pakete und den Sende- und Empfangs-Port festlegt. Sie sorgt so für die Stauvermeidung und optimiert den Durchsatz.

- **Vermittlungsschicht**

sorgt für die Weitervermittlung von Datenpaketen einschliesslich der Wegesuche (Routing), Netzadressen

- **Sicherungsschicht**

Codiert die zu übertragenden Datenpakete und gewährleistet eine weitgehend fehlerfreie Übertragung, regelt den Zugriff auf das Übertragungsmedium, sorgt für Prüfsummen und Quittierungs- und Wiederholungsprozeduren bei Fehlern oder Verlusten

- **Bitübertragungsschicht**

stellt mechanische, elektrische und weitere funktionale Hilfsmittel zur Verfügung, um physikalische Verbindungen zu aktivieren bzw. deaktivieren, sie aufrechtzuerhalten und Bits darüber zu übertragen.

Hier werden die verwendeten Übertragungsmedien (z.B. Kupferkabel, Glasfaserkabel, Infrarot- oder Funkübertragung) beschrieben sowie die Schnittstellen mit Spannungspegeln, Steckverbindern und Datenübertragungsraten.

Kommunikation und -services

Als «Digitale Kommunikation» könnte auch ICT Kommunikation bezeichnet werden, wenn da nicht auch noch die Telefonie wäre. Im Rahmen des Modules E-Business Applikationen Anpassen steht die ICT Kommunikation im Vordergrund. Zunehmend drängt sich aber mit der Schnittstelle Mensch-E-Business Applikation auch die Telefonie beim Sprachinterface dazu. Einige Begriffe sollen der Vollständigkeit halber im Zusammenhang mit digitaler Kommunikation gestreift werden⁹:

Mensch-Maschine-Kommunikation beschäftigt sich mit der menschlichen Kommunikation und den Schnittstellen zwischen Technik und Mensch.

Computervermittelte Kommunikation beschäftigt sich mit der Wechselwirkung zwischen den Kommunikationsmedien und ihren Nutzern.

Netzbasierte Kommunikation behandelt Methoden mit deren Menschen über Computernetzwerke kommunizieren.

Kommunikationstechnik beschäftigt sich mit der technischen Gestaltung von Kommunikationsmedien.

⁹ Referenz: https://de.wikipedia.org/wiki/Digitale_Kommunikation

Wirtschaftsinformatik beschäftigt sich mit der Ausgestaltung der Digitalisierung in betrieblichen Prozessen und deren ökonomischen Wirkung

Vernetzung von Geräten spielt vor allem in Bereichen wie Smart Homes oder Smart Cities eine wichtige Rolle.

Die Digitalisierung führt in der Arbeitswelt zu einer Neuausrichtung der Unternehmen und der Geschäftsprozesse sowie eine Veränderung der klassischen Arbeitsplätze. Dies bedeutet vor allem die Vernetzung von einzelnen Prozessen durch den Einsatz von neuen Technologien und damit eine drastische Veränderung der Arbeitsweise. Viele Menschen fürchten sich vor der Digitalisierung, weil nicht abzusehen ist was das für ihren Arbeitsplatz bedeutet. Ebenso wie die industrielle Revolution die menschliche Arbeit nicht abgeschafft hat, so wird es auch die digitale Revolution nicht tun.

Immer wieder ist von der digitalen Revolution -in Anlehnung an die industrielle Revolution- die Rede, eine zentrale Rolle spielt dabei die Kommunikation. Sie wird so in einen geschichtlichen Zusammenhang gestellt und kann besser verstanden. Es scheint, als würde in der digitalen Welt alles mit einer riesigen Geschwindigkeit ablaufen. Oft ist davon die Rede, dass in der ICT Veränderungen statt in einem Jahr in einem Monat ablaufen. Das Smartphone benötigte nach seiner Erfindung gerade einmal 10 Jahre um von der Hälfte der Menschheit genutzt zu werden.

Die Kommunikation zwischen den Schichten

In Schichtenmodellen lassen sich komplexe und aufwändige Abläufe logisch abbilden. Ein Schichtenmodell besteht aus einzelnen Schichten, die in einer bestimmten Reihenfolge durchschritten werden müssen. Jeder Service einer Schicht wird gemäss den definierten Regeln (Protokolle) aufgerufen. Diese Regeln geben an, wie die Daten zu verarbeiten sind. Der Datenaustausch erfolgt zwischen Dienste, welche über Schnittstellen zur Verfügung stehen. Die Protokolle und Services der einzelnen Schichten können unabhängig von den weiter weg liegenden Schichten angepasst, geändert oder ausgetauscht werden. Einzig die Kommunikationsschnittstelle zwischen 2 Schichten darf nur in Absprache geändert werden.

Die Kommunikation zwischen zwei Anwendungen (Layer 7) erfolgt

- a) zur darunter liegenden Schicht
- b) logisch mit dem Protokoll zur

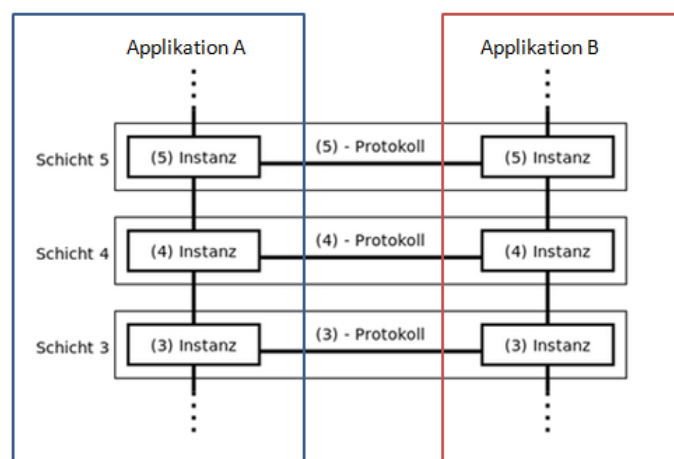


Abb. Kommunikation zwischen den verschiedenen Schichten

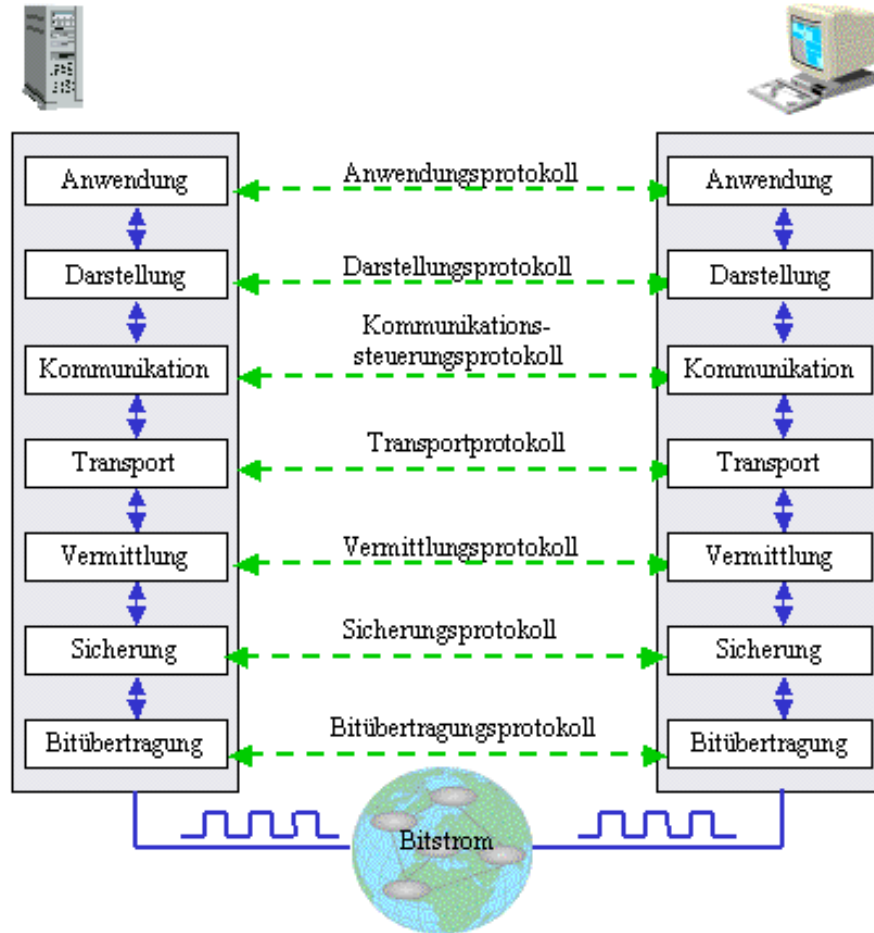


Abb. Praxisbeispiel: Die Kommunikation zweier Anwendungen auf zwei Rechnern

Protokolle

Dieser Abschnitt geht vertiefter auf die verschiedenen Kommunikations-Services und Protokolle ein. Er erhebt keinen Anspruch darauf vollständig zu sein. Heute existieren über 600 Netzwerk-Protokolle.

Zur Abbildung von Prozessen wurden eine Vielzahl von Ansätzen wie bspw. die BPML (Business Process Modeling Language) Spezifikation entwickelt. Die Spezifikation mit der grössten Unterstützung von international agierenden Unternehmen stellt die BPEL4WS (Business Process Execution Language for Web Services) dar.

Eine Übersicht der Web Service Spezifikationen finden sich auf <http://www.osmoticweb.com>

WSDL¹⁰

Die Web Services Description Language (WSDL) ist eine plattform-, programmiersprachen- und protokollunabhängige Beschreibungssprache für Netzwerkdienste (Webservices) zum Austausch von Nachrichten auf Basis von XML. WSDL ist ein industrieller Standard des World Wide Web Consortiums (W3C).¹¹

WSDL ist eine Metasprache, mit deren Hilfe die angebotenen Funktionen, Daten, Datentypen und Austauschprotokolle eines Webservice beschrieben werden können. Es werden im Wesentlichen die Operationen definiert, die von aussen zugänglich sind, sowie die Parameter und Rückgabewerte dieser Operationen. Im Einzelnen beinhaltet ein WSDL-Dokument funktionelle Angaben zu:

- der Schnittstelle

¹⁰ Weitere Informationen:

https://www.ibm.com/support/knowledgecenter/de/SSKM8N_8.0.0/com.ibm.etools.mft.doc/ac34640_.htm

¹¹ Quelle: https://de.wikipedia.org/wiki/Web_Services_Description_Language

- Zugangsprotokoll und Details zum Deployment
- Alle notwendigen Informationen zum Zugriff auf den Service, in maschinenlesbarem Format
- Nicht enthalten sind hingegen:
 - Quality-of-Service-Informationen
 - Taxonomien/Ontologien zur semantischen Einordnung des Services

UDDI

Universal Description, Discovery and Integration (UDDI) ist ein Begriff aus dem Umfeld der serviceorientierten Architektur (SOA) und bezeichnet einen standardisierten Verzeichnisdienst, der die zentrale Rolle in einem Umfeld von dynamischen Web Services spielen sollte.

Der Verzeichnisdienst besitzt eine SOAP-Schnittstelle. Er enthält Unternehmen, ihre Daten und ihre Services. Dabei kann man in UDDI zwischen drei Arten von Informationen unterscheiden: den "White Pages", einer Art Telefonbuch, den "Yellow Pages", also die elektronische Entsprechung der gelben Seiten und den Schnittstellenbeschreibungen in den sog. "Green Pages".

XML-RPC¹²

XML-RPC ist ein einfaches Protokoll für entfernte Funktionsaufrufe. XML-RPC ist leicht zu verstehen und anzuwenden. Die Spezifikation enthält Beispielen und eine FAQ, sie wurde von Dave Winer von Userland verfasst. Der Footprint einer XML-RPC Implementierung kann auf wenige Kilobyte beschränkt implementiert werden und ist auch für Mobile Web Services interessant.

XML-RPC ermöglicht entfernte Funktionsaufrufe über das HTTP Protokoll. Für entfernte Funktionsaufrufe gibt es die Bezeichnung Remote Procedure Call oder kurz RPC. XML dient zur Codierung der Übergabeparameter an die entfernten Procedures. Daher der Name XML-RPC. XML-RPC ist sprachunabhängig, Server und Clients können in beliebigen Sprachen wie Java, Perl oder Ruby implementiert werden.

Ein Aufruf wird per HTTP-POST an den Server geschickt. Dieser wertet das im Body enthaltene XML Dokument aus und verwendet dessen Inhalt als Parameter für den Aufruf der gewünschten Funktion. Das Ergebnis wird wieder in XML gepackt und zum Client übertragen. Neben primitiven Datentypen wie String, Double und DateTime werden auch komplexe Typen unterstützt, die aus den primitiven Typen zusammengesetzt werden können.

WEB Services¹³

Webservices sind im weitesten Sinne Kommunikationsmethoden oder Protokolle zwischen zwei E-Business Anwendungen resp. elektronischen Geräten (Clients) über das World Wide Web (WWW). Webservices gibt es in zwei unterschiedlichen Arten: Als Simple Object Access Protocol (SOAP) und als Representational State Transfer (REST). Zwischen SOAP und RESTful Webservices gibt es erhebliche Unterschiede.

Bei der Entscheidung zwischen REST und SOAP gibt es zwei Schlüsselthemen, die berücksichtigt werden müssen:

1. Die Typen von Clients, die unterstützt werden. Beispielsweise bieten REST-Dienste eine effektive Möglichkeit zur Interaktion mit leichtgewichtigen Clients (Smartphones).
2. Das Unternehmen legt oft die zu verwendenden Services als Standard fest..

SOAP legt eine Standard-Kommunikationsprotokoll-Spezifikation (Regelwerk) für den XML-basierten Nachrichtenaustausch fest. SOAP verwendet verschiedene Transportprotokolle wie HTTP und SMTP. Das Standardprotokoll HTTP erleichtert es dem SOAP-Modell, ohne Änderungen am SOAP-Protokoll über Firewalls und Proxies zu tunneln. SOAP kann manchmal langsamer als Middlewaretechnologien wie CORBA oder ICE sein, da es ein ausführliches XML-Format hat.

¹² Quelle und weiterführende Infos: <https://www.oio.de/public/xml/xml-rpc.htm>

¹³ Quelle: <https://www.computerweekly.com/de/tipp/REST-versus-SOAP-Den-passenden-Webservice-auswaehlen>

- Die Web Services Description Language (WSDL) beschreibt ein gemeinsames Regelwerk zur Definition von Nachrichten, Bindungen, Operationen und Speicherorten des Service. WSDL kann man sich ähnlich wie einen Vertrag vorstellen, der die Schnittstelle beschreibt, die der Dienst anbietet.
- SOAP erfordert weniger Installationscode als das REST-Servicedesign (zum Beispiel Transaktionen, Sicherheit, Koordination, Adressierung und Vertrauen). Die meisten Anwendungen in der Praxis sind nicht einfach und unterstützen komplexe Operationen, bei denen es darauf ankommt, den Gesprächszustand und die Kontextinformationen aufrechtzuerhalten. Mit dem SOAP-Ansatz müssen Entwickler keinen Installationscode in die Anwendungsschicht schreiben.
- SOAP Webservices, wie zum Beispiel JAX-WS, sind für die asynchrone Verarbeitung und Aufrufe nützlich.
- SOAP unterstützt verschiedene Protokolle und Technologien, darunter WSDL, XSDs und WS-Addressing.

REST beschreibt eine Reihe von Architekturprinzipien, mit denen Daten über eine standardisierte Schnittstelle (zum Beispiel HTTP) übertragen werden können. REST enthält keine zusätzliche Messaging-Schicht und konzentriert sich auf Designregeln für die Erstellung von stateless (zustandslosen) Services. Ein Client kann über die eindeutige URI auf die Ressource zugreifen und erhält eine Repräsentation der Ressource zurück. Mit jeder neuen Ressourcen-Repräsentation transferiert der Client seinen Zustand (state). Beim Zugriff auf RESTful-Ressourcen mit dem HTTP-Protokoll dient die URL der Ressource als Ressourcen-Identifikator und GET, PUT, DELETE, POST und HEAD sind die Standard-HTTP-Operationen, die auf dieser Ressource ausgeführt werden.

- RESTful Webservices sind stateless (zustandslos). Sie können diese Bedingung testen, indem Sie den Server neu starten und prüfen, ob Interaktionen weiter vorhanden sind.
- Für die meisten Server bieten RESTful Webservices eine gute Caching-Infrastruktur über eine HTTP-GET-Methode. Dies kann die Performance verbessern, wenn die Information, die der Service zurückgibt, nicht häufig geändert wird und nicht dynamisch ist.
- Service-Produzenten und -Konsumenten müssen den Kontext und die Inhalte, die weitergegeben werden, verstehen, weil es kein einheitliches Regelwerk zur Beschreibung der REST-Webservices-Schnittstelle gibt.
- REST ist nützlich für Geräte mit eingeschränktem Profil wie zum Beispiel für Mobilgeräte, bei denen der Overhead zusätzlicher Parametern geringer ist (zum Beispiel Header).
- REST-Services lassen sich leicht in bestehende Websites integrieren und werden mit XML dargestellt, so dass die HTML-Seiten leicht das Gleiche konsumieren können. Es besteht wenig Bedarf, die bestehende Site-Architektur zu überarbeiten. Damit sind Entwickler produktiver, weil sie nicht alles von Grund auf neu schreiben, sondern nur die vorhandene Funktionalität ergänzen müssen.
- Eine REST-basierte Implementierung ist im Vergleich zu SOAP einfach.

Message Oriented Middleware (MOM) bezeichnet eine Middleware, welche auf der asynchronen Kommunikation, also der Übertragung von Nachrichten (Messages) beruht und drei verschiedene Kommunikationsprotokolle unterstützt. Das Format für die Nachrichten ist nicht festgelegt, in der Praxis hat sich jedoch [XML](#) als beliebtes Format etabliert.

Aufgrund des vergleichsweise selbsterklärenden und im Gegensatz zu Nachrichten im Binärformat leicht menschenlesbaren Formats ist es beim Einsatz von XML einfach, auch die Kommunikation zwischen Middleware-Systemen zu ermöglichen. Um die Kommunikation zu ermöglichen, kann ein XSLT-Prozessor als Übersetzer zwischengeschaltet werden, der mit Hilfe eines Transformations-Stylesheets Nachrichten von der XML-basierten Sprache des Quellsystems in die Sprache des Zielsystems übersetzt. Als Protokoll wird häufig SOAP eingesetzt.

XPath und XQuery sind Abfragesprachen, mit denen Teile von XML-Dokumenten abgefragt und extrahiert werden können. Sie sind keine Voraussetzung für einen ESB, werden aber in ESBs häufig

im Kontext der Routingdienste eingesetzt. Routingregeln, die auf Steuerdaten oder Inhalten von Nachrichten beruhen, sind oft als XPath- bzw. XQuery-Ausdrücke über diese Nachrichten formuliert.

Java Message Service (JMS) ist eine standardisierte Programmierschnittstelle, um aus Java-basierten Anwendungen Nachrichten über einen Message Bus versenden und empfangen zu können. Sie ist keine Voraussetzung für einen ESB. ESBs, die stark in der Java-Welt verankert sind, zum Beispiel, weil sie selbst in Java implementiert sind, bieten oft Standard-Adapter und Standard-Endpunkte an, damit Dienste den ESB über JMS nutzen können. JMS ist der am weitesten verbreitete Standard für MOMs und wird von fast allen Herstellern unterstützt.

Advanced Message Queuing Protocol (AMQP) ist ein offener Standard, der ein binäres Netzwerkprotokoll auf Anwendungsebene für eine Message-orientierte Middleware (MOM) darstellt.

Im Mai 2010 wurde der Draft (Entwurf) der Version 1 veröffentlicht. Um der grossen Verbreitung von JMS Rechnung zu tragen, sind alle JMS Funktionen in dem Protokoll eingearbeitet. Dies ermöglicht es den Entwicklern, weiter die JMS Schnittstelle zu nutzen, während sich MOMs untereinander mit AMQP verständigen können.

Das **Internet Printing Protocol (IPP)** wurde zur Anbindung von Druckdiensten über das Internet entwickelt. Es lässt sich auf allgemeine Ressourcen erweitern, da der HTTP-Standard selbst die wichtigsten Operationen zur Verfügung stellt.

Das Transaktions-Konzept

Zuerst geht es um die Klärung des Begriffs: Was ist eine Transaktion?

Eine Menge von logisch zusammengehörenden Operationen (zumeist in ein Programm eingebunden) die der Abwicklung eines Geschäftsvorfalles dienen werden als Transaktion bezeichnet. Dabei können verschiedenen Typen beachtet werden:

- **Klassische Datenbanktransaktionen** beziehen sich (Lesen bzw. Schreiben von Datenobjekten) auf dasselbe Datenbanksystem
- **Verteilte Transaktionen** erlauben Operationen in verschiedenen Datenbanksystemen z.T. auch an unterschiedlichen Standorten. Das Rückabwickeln einer solchen Transaktion ist dementsprechend anspruchsvoller.

Verallgemeinerter Transaktionsbegriff: Transaktionen sind beliebige Service-Aufrufe (semantisch reiche Operationen) und sind nicht unbedingt Datenbank zentriert. Somit geht die **Bedeutung des Transaktionskonzepts** über den Einsatz im Rahmen von DBs hinaus und stellt ein zentrales Paradigma der Informatik dar. Das Transaktionskonzept ist der Schlüssel zur zuverlässigen Nutzung verteilter Systeme

Eine Transaktion hat grundsätzlich einen atomaren Charakter: Sie wird ganz oder gar nicht ausgeführt („alles oder nichts“). Unter einer Transaktion wird eine Folge von Datenbankoperationen verstanden, die die Daten von einem konsistenten Zustand in einen neuen konsistenten Zustand überführen und entweder ganz oder gar nicht ausgeführt werden.

Bei einem konkurrierenden Zugriff (concurrency) werden Transaktionen untereinander getrennt und bewahren so die Konsistenz der Datenbank.

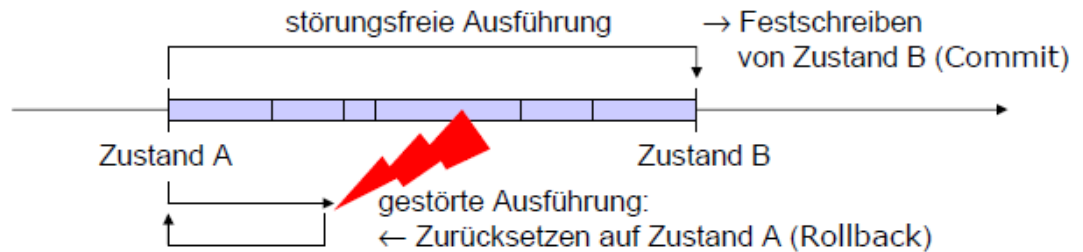


Abb. Transaktion mit erfolgreichem Ausgang und einem „Roll-Back“

Um den Anwendungsentwickler bei der Umsetzung dieser Anforderungen zu unterstützen wurde das ACID Konzept entwickelt und implementiert. Der Entwickler muss lediglich die Transaktionsgrenzen festlegen, von allen anderen Massnahmen wird er bei der Umsetzung befreit:

- BOT Beginn der Transaktion
- EOT Ende der Transaktion

Das ACID Prinzip

Das ACID Prinzip beschreibt die Eigenschaften von Verarbeitungsschritten in Datenbankmanagementsystemen (DBMS) und verteilten Systemen und steht für **A**tomicity, **C**onsistency, **I**solation und **D**urability¹⁴

1. **Atomicity (Atomarität, "Alles oder Nichts")**

Die Ausführung einer Transaktion soll aus Sicht des Benutzers ununterbrechbar verlaufen, so dass sie entweder vollständig oder gar nicht ausgeführt wird. Dies bezieht sich vor allem auf die im Rahmen der Transaktion auszuführenden Änderungen der Datenbank. Tritt während der Ausführung einer Transaktion ein Fehler auf (Programmfehler, Hardware-Fehler, Absturz des Betriebssystems usw.), der die ordnungsgemäße Fortführung verhindert, werden seitens des DBS sämtliche bereits erfolgten Änderungen der Transaktion zurückgesetzt. Durch eine sog. Undo-Recovery werden die "Spuren" der unterbrochenen Transaktion vollständig aus der Datenbank entfernt. Um diese Fehlerbehandlung zu ermöglichen, führt das DBS ein Logging durch, d. h., zu den erfolgten Änderungen werden geeignete Informationen auf einer Protokoll- oder Log-Datei dauerhaft mitgeschrieben.

2. **Consistency (Konsistenz)**

Die Transaktion ist die Einheit der Datenbankkonsistenz. Dies bedeutet, dass sie die Datenbank von einem konsistenten in einen wiederum konsistenten (nicht notwendigerweise unterschiedlichen) Zustand überführt. Von besonderer Bedeutung ist dabei die Einhaltung der logischen Konsistenz, so dass die Inhalte der Datenbank einem möglichst korrekten Abbild der modellierten Wirklichkeit entsprechen. Hierzu können beim Datenbankentwurf semantische Integritätsbedingungen (zulässige Wertebereiche, Schlüsseleigenschaften usw.) definiert werden, welche vom DBS automatisch zu überwachen sind. Das DBS garantiert somit, dass am Ende einer jeden Transaktion sämtliche Integritätsbedingungen erfüllt sind. Änderungen, welche zu einer Verletzung der Integritätsbedingungen führen, werden abgewiesen, d. h., sie führen zum Zurücksetzen der Transaktion. Voraussetzung für die logische ist die physische Konsistenz der Datenbank, d. h. die korrekte interne Repräsentation und Speicherung der Daten im Datenbanksystem.

3. **Isolation**

Datenbanksysteme unterstützen typischerweise eine grosse Anzahl von Benutzern, die gleichzeitig auf die Datenbank zugreifen können. Trotz dieses Mehrbenutzerbetriebs wird garantiert, dass

¹⁴ Quelle: <https://dbs.uni-leipzig.de/buecher/DBSI-Buch/HTML/kap13-2.html>

dadurch keine unerwünschten Nebenwirkungen eintreten, wie z. B. das gegenseitige Überschreiben desselben Datenbankobjektes.

4. Dauerhaftigkeit (Durability)

Das DBS garantiert die Dauerhaftigkeit bzw. Persistenz erfolgreicher Transaktionen, deren Operationen vollständig ausgeführt wurden. Dies bedeutet, dass Änderungen dieser Transaktionen alle künftigen Fehler überleben, insbesondere auch Systemabstürze oder Externspeicherausfälle. Hierzu sind gegebenenfalls die Änderungen seitens des DBS im Rahmen einer Redo-Recovery zu wiederholen. Dafür sind wiederum geeignete Logging-Massnahmen erforderlich, insbesondere sind vor Abschluss einer Transaktion die für die Recovery benötigten Informationen zu protokollieren.

Die ACID Eigenschaften werden als Aufgabe der Laufzeitumgebung (Teil der Plattform-Architektur) umgesetzt:

- **Concurrency Control** dient der Isolation von Transaktionen (Serialisierung)
- **Recovery** dient der Atomarität und Persistenz (Fehlererholung, Crash Recovery)

Integrität einer Datenbank

Verschiedene Bedingungen müssen erfüllt sein, damit die Integrität einer Datenbank gewährleistet werden kann. Die **Referentielle Integrität** stammt aus der Relationstheorie beschreibt wie die Beziehungen zwischen Datenobjekten kontrolliert sind.

Neben der referentiellen Integrität sind auch andere Datenqualitäts- und -Konsistenz-Aspekte zu beachten: Werte Bereichsintegrität, Eindeutigkeit von Schlüsselbegriffen, Konsistenz.

Unter dem Begriff **Konsistenz** sichern Datenbanksysteme vor allem im Mehrbenutzerbetrieb (wie das bei E-Business Applikationen der Fall ist) die Konsistenz von Daten auf Transaktionsebene (alle oder keine Updates, z. B. bei technischem Abbruch) und gegen Updates konkurrierender Benutzer/Transaktionen.

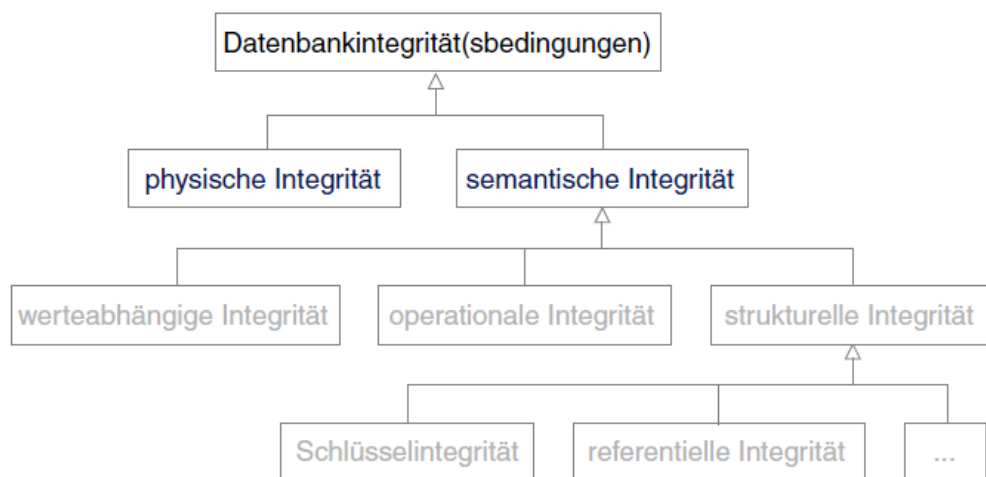


Abb. Datenbankintegrität [Quelle: Weber 2003]

Sicherheit von E-Business Anwendungen

E-Business Anwendungen sind gegenüber Angriffen von aussen sehr exponiert, da je nach Einsatzbereich grössere Betrugsmöglichkeiten offen sind. Nicht nur beim eBanking lauern Gefahren, sondern je nach dem auch bei ganz rudimentären eCommerce Anwendungen.

Wie immer sind nicht nur technische Gefahren vorhanden, oft stellen auch ganz einfache, nicht zu Ende gedachte Geschäftsprozesse ein Risiko dar. Technisch betrachtet bestehen erhebliche Sicherheitsrisiken bei der Speicherung, Übermittlung und Verarbeitung der für E-Commerce und E-Business relevanten Daten.

Die Realisierung von E-Business Anwendungen und deren geeigneter Einsatz in betrieblicher Umgebung stellt eine Herausforderung dar. Bei der Erarbeitung von geeigneten Lösungen wie auch beim Betrieb und beim Benutzen derselben. E-Business-Lösungen verlangen ein umfassendes, aufgabengerechtes Informationssicherheitskonzept und die Implementation geeigneten technischen Lösungen. Die Benutzer der E-Business Anwendung müssen diese ebenfalls respektieren.

Ausgangspunkt von Sicherheitsüberlegungen sind die Bedrohung, die potenziellen Ziele und Objekte von Angriffen und das mögliche Vorgehen allfälliger Angreifer. Grundbedrohungen sind die Verletzung der Vertraulichkeit, der Integrität und der Verfügbarkeit von Daten. Abgeleitete Bedrohungen richten sich vor allem auf die Verletzung der Authentizität, der Verbindlichkeit und der Zweckbindung von Informationen.

Die ICT Massnahmen können verschiedenartig sein, hier einige Beispiele:

- Schutz vor Angriffen aus dem Internet durch Firewalls
- Verschlüsselung der Passwörter auf verschiedenen Ebenen
- Aufteilung des internen Netzes in Netze unterschiedlicher Vertrauenswürdigkeit
- Private Adressräume im lokalen Netz
- Teilung des Netzes in verschiedene physische Netze
- **Demilitarisierte Zone (DMZ, auch Demilitarized Zone)**
- Aufteilung von Mail- und Webserver auf getrennte Maschinen

Die geforderte Sicherheit kann nicht durch den Einsatz von punktuellen Einzelmassnahmen erreicht werden. Entsprechend der spezifischen Sicherheitsbedrohungen und der Risiken und ihrem Ausmass ist die Definition und Umsetzung einer entsprechenden Sicherheitspolitik / Sicherheitsarchitektur nötig. Auf der **strategischen Ebene** gehören Grundsatzentscheidungen basierend auf der Unternehmenspolitik dazu. Auch die Rahmenbedingungen für die Gewährleistung der Informationssicherheit und der damit verbundenen Informationsverarbeitungsprozesse sind festzulegen. Zudem müssen die organisatorischen Elemente definiert werden.

Die **Informationssicherheitspolitik** regelt den Bezugsrahmen, den Geltungsbereich, die Sicherheitsziele, die Grundbedrohungen etc. ... Sie zeigt auf, wie die diversen Gesetze und Vorschriften eingehalten werden können und wie der Erhalt der Handlungsfähigkeit, z.B. Datenschutzgesetz, Urheberrecht etc. möglich ist. Sie stellt die Voraussetzungen für die interne Revision dar.

Die **Sicherheitsarchitektur** regelt die technische Umsetzung der Strategie und zeigt auf, wie die Sicherheit aus technischer Sicht gewährleistet werden kann. Dazu gehören:

- die Feststellung der momentanen Bedrohungssituation
- die Identifikation der Schutzobjekte
- eine sorgfältige Risikoanalyse
- (Antizipation der gefährlichen Ereignisse und Situationen)
- die Bewertung der Risiken (Schadenswert und -häufigkeit)
- die Suche nach adäquaten Massnahmen und deren Bewertung (Kosten/Nutzen)
- die Auswahl und Realisierung durch Management (Restrisikobestimmung)
- das Controlling und Reporting

Arbeitsblatt

Zielsetzung

Die Lernenden setzen sich mit der Anwendungs-Architektur sowie dem Umfeld / ICT-Infrastruktur von E-Business Anwendungen auseinander. Sie können die ausgewählte Anwendung in der Unternehmensarchitektur einordnen. An der Architektur der E-Business Anwendung werden Anforderungen aus und die Massnahmen basierend auf den Themen Sicherheit, Performance, Verfügbarkeit, Stabilität und Durchsatz erläutert.

Arbeitsform

Dies ist eine Einzelarbeit oder Partnerarbeit (zu zweien).

Jedes Team wählt in Absprache mit der Lehrperson eine oder zwei E-Business-Applikation aus der/den im AB01 erstellten E-Business Application MAP und vertieft sich in deren Architektur.

Ein spezieller Fokus wird dem Thema Sicherheit gewidmet. Zudem werden die Fragen im Zusammenhang mit den Anforderungen an die Performance und Verfügbarkeit und deren Messung behandelt (Service Level).

Zeitbudget

3 Lektionen plus selbständiges Studium und Arbeitsleistung, Interview im Ausbildungsbetrieb.

Aufgabe für die Lernenden

Aufgabe 1

Die Lernenden dokumentieren die Anwendungsarchitektur (System Kontextdiagramm) und Softwarearchitektur (Schichtenmodel und deren Inhalte) der ausgewählten E-Business Applikation (eine aus dem Arbeitsblatt 1).

Aufgabe 2

Die Lernenden zeigen detailliert die Schnittstellen zu den anderen E-Business-Applikationen auf und beschreiben die Infrastruktur-Anforderungen bezüglich der Kommunikationsarchitektur.

Aufgabe 3

Die Lernenden erarbeiten die Anforderungen der E-Business Anwendung an deren Sicherheit (Identifikation, Datenaustausch).

Aufgabe 4

Die Lernenden leiten aus dem Ergebnis der Aufgabe 3 die Anforderungen an die Stabilität, die Performance und die Verfügbarkeit der ICT Infrastruktur ab.

Arbeitsergebnis (Werkstück) Kompetenznachweis

Das Team erstellt ein Dokument mit den Erläuterungen und eine Präsentation.

Die Unterlagen werden gemäss den Instruktionen der LP abgegeben. Die LP kann einzelne Teams zur Präsentation auffordern.

Bei der Abgabe als Mail gilt: (Name des PDF-Files: *150_A2 Name1_Name2_Klasse.pdf*)

Für die Präsentation wählt das Team eine zweckmässige Art und wählt allenfalls eine mehrstufige Darstellung. Die formalen Kriterien einer Präsentation werden ebenfalls bewertet.

- a) graphische Darstellung der Anwendungs-Architektur (Erstellung mit einem SW-Tool)
- b) Dokumentation der Schnittstellen und Beurteilung anhand des OSI Layer-Modells und der Security Anforderungen (primär aus dem Ergebnis der Aufgabe 2 und 3).
- c) ICT Infrastruktur für den Betrieb der E-Business-Applikation basierend auf dem Ergebnis der Aufgabe 4.

Ergänzender Lesestoff

https://de.wikipedia.org/wiki/Enterprise_Service_Bus

<http://teambblue.unicomsi.com/products/system-architect/#>

<https://de.wikipedia.org/wiki/OSI-Modell>

<https://de.wikipedia.org/wiki/Internetprotokollfamilie>

https://de.wikipedia.org/wiki/Cold_Standby

<https://de.wikipedia.org/wiki/Failover>