

Web-Developer

(Web)-Architekturen





(WEB)-ARCHITEKTUREN

Horst Werner Schneider, Seite 2

Bedeutung von Web-Anwendungen

- Web-Portale als neue Visitenkarte
- Sammeln von Kundeneinformationen
 - Kundenverhalten aus Tracking-Cookies ermitteln (analytisches CRM)
- Self Service Automation
 - Hilfestellung Rund um die Uhr, 24 / 7 Service ohne Personaleinsatz
- Schnittstelle zu den Stakeholdern
 - Bereitstellung von Services für Kunden, Angestellte, Lieferanten, etc..
- Hohe Reichweite, unbeschränkte Verkaufsflächen
- Verwendung offener Standards
- Plattformunabhängig

Terminologie

- Portale

lateinisches Wort „porta“ = Pforte, Tor, Tür, Anwendungen, Prozesse und Dienste werden in einem Anwendungssystem integriert.

- Website

lateinisches Wort „situs“ = Lage, Stellung, virtueller Platz im WWW wo sich mehrere Webseiten und andere Ressourcen befinden,
Synonyme: Webauftritt, Webangebot oder Webpräsenz

- Homepage

Hauptseite einer Website, zentrale Ausgangsseite eines Internetauftritts

- WWW

world-wide-web, dem ein Hypertext und das HTTP-Protokoll zugrunde liegen.

- URL

Uniform Resource Locator, bezeichnet eine Netzwerkressource

Terminologie

- **HTTP(S)**
Hypertext Transfer Protocol (Secure), Übertragungsprotokoll
- **XML**
Extensible Markup Language, Auszeichnungssprache zur Darstellung von hierarchisch strukturierten Daten in Form von Textdateien.
- **JSON**
Javascript Object Notation, kompaktes Datenformat in Form von Textdateien, das für Mensch und Maschine leicht lesbar ist.
- **Landing page**
Spezielle Webseite, auf die ein Interessent (Lead) durch ein Werbemittel oder nach einem Klick in einer Suchmaschine hingeführt wird. Ist auf den Werbeträger und die Zielgruppe optimiert.

Terminologie

- ASP / ASP.NET

Active Server Pages = von Microsoft entwickelte Skriptsprache

ASP.NET = von Microsoft entwickeltes Web Application Framework

- MVC

Model View Controller, Design-Pattern (Entwurfsmuster), Architektur-Muster, strikte Trennung von Business Logik, Oberfläche.

- CMS

Content Management System, Software zur gemeinschaftlichen Verwaltung von Inhalten.

- Framework

Rahmengerüst, definiert die Regeln der Anwendungsarchitektur, ist selbst kein fertiges Programm, die Funktionalität entsteht durch die Registrierung von Implementierungen (z.B. Zugriff auf Dateisystem)

Terminologie

- IS

Informationssysteme

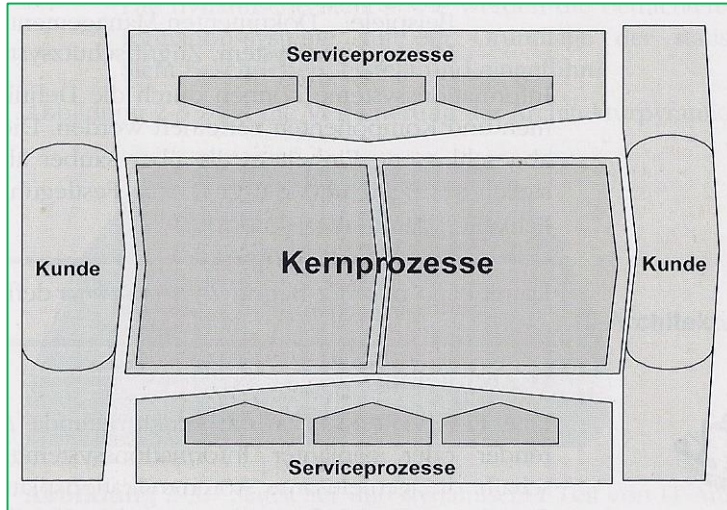
- IT-Architektur

Die IT-Architektur eines Unternehmens stellt die Gesamtheit aller Komponenten, Technologien und organisatorischen Maßnahmen dar, die die im Unternehmen vorkommenden Funktionen, Prozesse und Daten abbilden und deren Zusammenspiel (Vernetzung) ermöglichen.

Die eingesetzten IT-Architekturen müssen die Geschäftsstrategie und daraus abgeleitete Geschäftsprozesse unterstützen!

Die IT-Architektur stellt eine wesentliche Leitlinie für alle Entwicklungen und Entscheidungen im IT-Bereich dar.

IT-Architekturen - Bestandteile



Quelle: Dern (2009), S. 17

Anwendungslandschaft

Gesamtheit aller Informationssysteme eines Unternehmens, die zur Unterstützung der Durchführung von Geschäftsprozessen betrieben werden.

Geschäftsprozesse

Unterteilt in Kerngeschäfts- und Serviceprozesse.

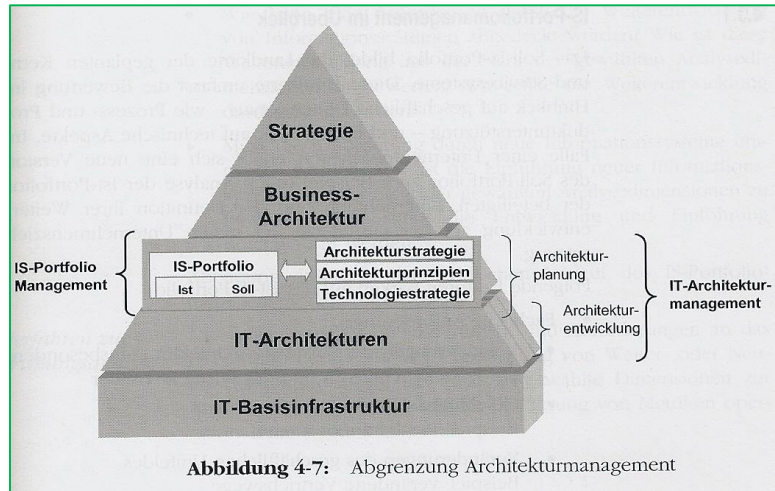
Kerngeschäftsprozesse werden aufgrund der fachlichen Aktivitäten, die vom Kunden ausgelöst werden definiert.
(z.B. Verkauf, Logistik, Produktion)

Serviceprozesse sind Aktivitätenfolgen für den internen Nutzen und dienen zur Unterstützung anderer Geschäftsprozesse.
(z.B. Buchhaltung, Personalwesen, Instandhaltung)

Horst Werner Schneider, Seite 8

Vgl. Dern (2009), S. 17.

Architektur-Management – IS-Portfolio-Management



Quelle: Dern (2009), S. 65

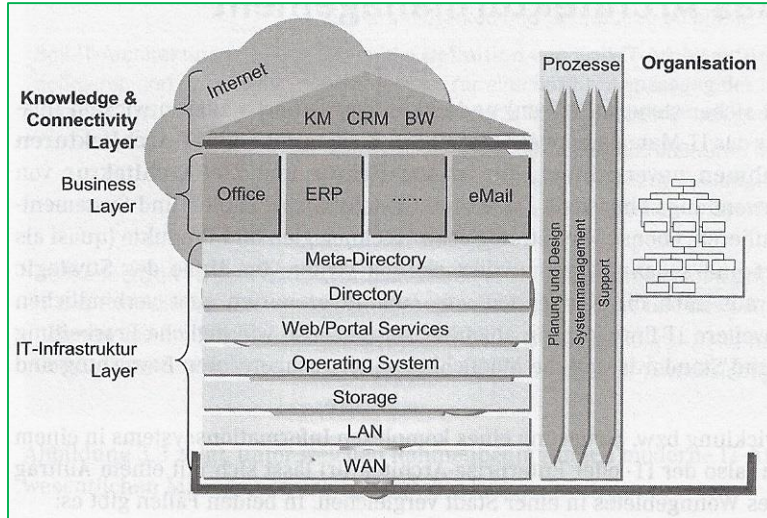
Das IS-Portfolio-Management steht in Wechselwirkung mit dem IT-Architektur-Management, et vice versa

IS-Portfolio-Management

Entwicklung eines Gesamtangebotes / Dienste auf Basis der Geschäftsarchitektur. Die Dienste können auf Basis der bestehenden bzw. zukünftigen IT-Architekturen umgesetzt werden.

„Although everyone wants their systems to reflect modern architectural principles, few take the necessary first step. A detailed understanding of current systems in a prerequisite to modernization.“
(Gartner Group, 2003 [Gartner03-1])

IT-Architekturen als Schichtenmodell



Quelle: Tiemeyer (2011), S. 92

Grundbausteine

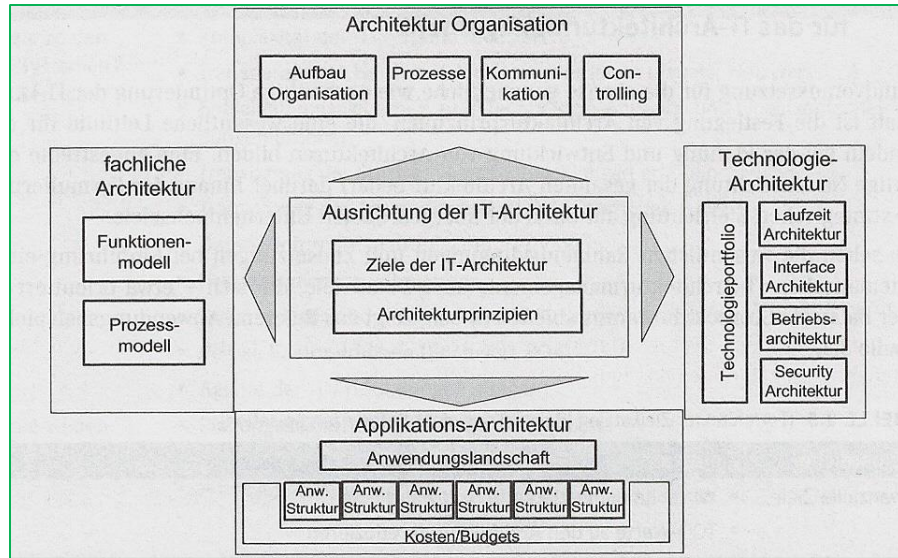
Technologie-Architektur (IT-Infrastruktur – Infrastructure Layer)

Anwendungs-Architektur (Business- bzw. Knowledge Layer)

Geschäfts-Architektur (fachliche Architekturen – Prozesse)

Daten / Informationsarchitektur

Architektur-Management - Ordnungsrahmen

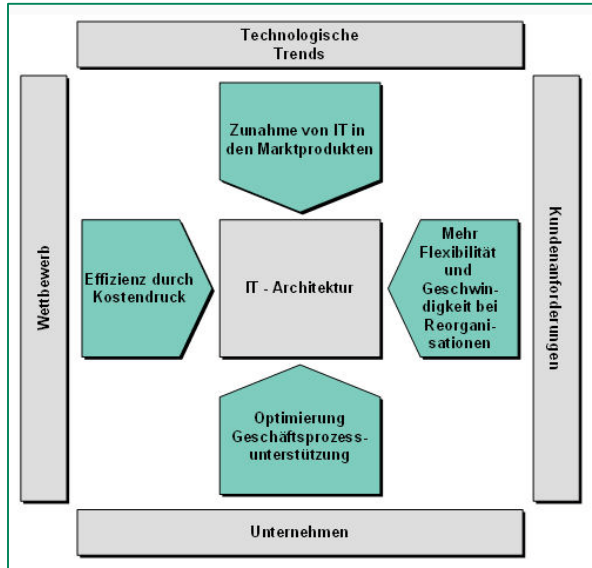


Quelle: Tiemeyer (2011), S.94

Ausrichtung der IT-Architektur

- Fachliche Architektur
- Architektur der Organisation
- Technologie-Architektur / IT Infrastruktur
- Applikations-Architektur

Architektur-Management - Einflussfaktoren



Quelle: Niebuhr (2005)

Hohe Anforderungen an Flexibilität der IT-Architektur
IT-Transformations-Projekte laufen in Echtzeit

IT-Komponenten spielen wichtige Rolle bei
Endkundenprodukten

Wettbewerbsvorteile durch optimale Ausrichtung auf
Geschäftsanforderungen / -prozesse

Architektur-Management – Zielsetzungen

Architektur-Zielsetzungen bilden eine wesentliche Leitlinie für das Handeln bei der Planung und Entwicklung von Architekturen.

Welche Zielbereiche, Zielsetzungen kennen Sie?

Beispiele:

Kundenziele: Nutzungsgrad von LTE im Mobilfunk erhöhen

IT-Services: Daten- / Informationstransparenz erhöhen (z.B. Gemeinden, Bürgernähe)
Komplexität reduzieren

Personalziele: Hohe Mitarbeiterzufriedenheit sichern
IT-Architektur-Know-How an das Unternehmen bilden (Knowledge-Base, Wiki)

Projektziele: Planungssicherheit für IT-Projekte erhöhen
Migrationsplanung optimieren

Architektur-Management – Handlungsprinzipien

Architekturprinzipien basieren auf Grundsätzen und Leitlinien, die für die Umsetzung und beim Management von IT-Architekturen vorgegeben werden.

Abweichungen von diesen Grundsätzen müssen begründet werden.

Die Anforderungen an die IT-Landschaft wirken dadurch **standardisierend**, **komplexitätssteuernd** und **richtungsgebend**.

Beispiele:

Finanzen:	Wirtschaftlichkeit der Architekturen und betroffenen Services, Prozesse, Projekte
Personal:	Klare Zuordnung von Verantwortung für die IT-Architektur
Anwendungen:	Mehrfachnutzung von Anwendungen / Services.
Daten:	Konsistente und Redundanzfreie Datenhaltung
Prozesse:	Standards festlegen und einhalten

Technologie-Architektur

Umfasst technische Systeme (Hardware, Plattformen, Standorte, etc.) inkl. ihren Komponenten, die Konfiguration und das Management der Systeme.

Kriterien:

Einstiegspunkt bei neuen Technologien

First Mover, Early Adopter, Early Follower, Cheapest

Zentralisierung / Dezentralisierungsgrad

Sourcing?

Lease or Buy, Cloud-Services

Einheitlichkeit / Heterogenität / Standards

Applikations-Architektur

Stellt einen Plan der eingesetzten Anwendungssysteme (ERP, E-Mail, CRM, etc.) und deren Interaktion (Schnittstellen) und deren Beziehung zu den Geschäftsprozessen der Organisation bereit.

Die Funktionsbereiche von Unternehmen werden durch die unterschiedlichen Anwendungen der Architekturkomponenten abgedeckt.

Kriterien:

Funktionalität, bezogen auf Prozesse und IS-Portfolio

System-Design, Designprinzipien, Standards

Geschäfts-Architektur

Die Geschäftsarchitektur umfasst die Business Strategie, Organisation und die Geschäftsprozesse.

Kriterien:

Grundstruktur des Geschäfts

inkl. der Geschäftsziele

Organisationsstrukturen

Zuständigkeiten, organisatorische Einheiten, Handlungsbeziehungen zw. den organisatorischen Einheiten.

Geschäfts-Prozesse

Kernprozesse und Serviceprozesse

Ressourcen

Akteure, Sachmittel

Daten-Architektur

Der statische Zusammenhang von Daten im Unternehmen werden in Form von Datenmodellen durch die Datenarchitektur beschrieben (Entity-Relationship-Modell)

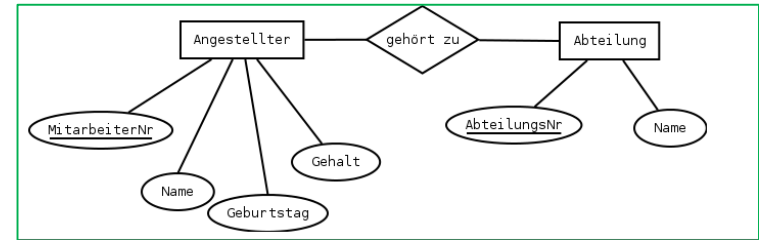
Kriterien

Klassifikation der Daten

Regelung der Datenverwendung

Definition des Datenzugriffs

Festlegen Dateneigner und dessen Verantwortung



Quelle: <http://www.datenbank-grundlagen.de/entity-relationship-modell.html>

Nutzen von IT-Architekturmanagement

Kosteneinsparung durch Vereinheitlichung

Steigerung der Effizienz, Kostenreduktion, Vereinfachung -> geringerer Know-How Bedarf

Ressourcen- und Kostensenkung durch Standardisierung

Konsolidierung der Infrastruktur, Bewertung Softwareplattformen und Eigenentwicklungen

Standardisierung führt zur Steigerung der Flexibilität in Bezug auf Schnittstellen zu externen IT-Systeme!

Entscheidungshilfen bei IT-Projekten

Bewertung von Kosten und Nutzen, Berücksichtigung aller Auswirkungen auf die IT-Landschaft sind wichtige Kriterien bei Projektentscheidungen.

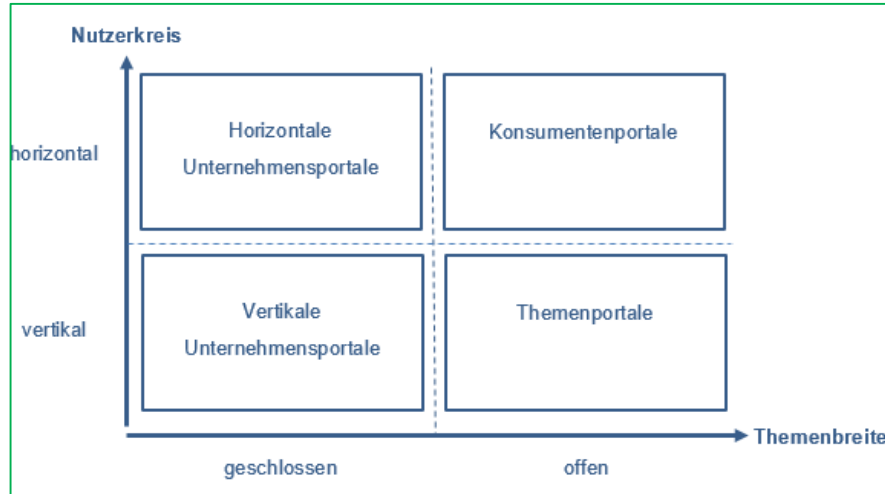
Kosteninformationen als Entscheidungsgrundlagen für das IT-Management

verursachungsgerechte Zuordnung von Kosten

Ausrichtung der Budgetplanung und des Controllings an den IT-Strukturen

Transparenz wird geschaffen -> Auswirkungen bei Veränderungen lassen sich bereits in der Planungsphase beurteilen.

Klassifizierung von Portalen



Quelle: In Anlehnung an Stelzer (2004), S. 14

Definition

Ableitung vom lateinischen Wort „porta“ (Pforte Tür, Tor)

Portale öffnen die unternehmensinterne Welt

Schaffen eine Brücke zur Umwelt des Unternehmens

Vgl. (Großmann/Koschek, 2005), S. 30.

Horst Werner Schneider, Seite 20

Klassifizierung von Portalen

Offene Portale

Nutzung des Portals ist nicht auf bestimmte Benutzer oder Benutzergruppen beschränkt.
Authentifizierung zur Identifizierung und zum Schutz personenbezogener Daten.

Das Portalangebot ist in den meisten Fällen personalisiert.

Geschlossene Portale

Das Portal wird nur für eine definierte Benutzergruppe im Internet oder Intranet geöffnet.
Eine Authentifizierung ist zwingend erforderlich.

Benutzergruppen werden als Stakeholder des Unternehmens zusammengefasst
(Mitarbeiter, Kunden, Lieferanten, Unternehmenspartner).

Klassifizierung von Portalen

Horizontale Ausrichtung von Portalen

Portale mit einem breit gefächertem Informations- und Funktionsangebot.

Verschiedene Applikationen (Katalogdienste, Foren, Service-on-Demand) werden zusammengefasst.

Mit horizontalen Portalen wird das Ziel der Erreichung einer hohen Reichweite verfolgt.

Strategie der horizontalen Integration, d.h. Aufgaben des Kunden die nicht zum Kerngeschäft des Kunden gehören werden übernommen. (Beispiel: Google)

Vertikale Ausrichtung von Portalen

Vertikale Portale stellen spezifische Funktionen für fachliche und technische Anforderungen zur Verfügung.

Sie sind auf bestimmte Themen, Branchen oder Zielgruppen zugeschnitten und werden auch als Themenportale bezeichnet.

Sind von ihrer Art „Insellösungen“ – lassen sich aber über offene Schnittstellen (z.B. RESTful API, SOAP) in horizontale Portale integrieren.

Klassifizierung von Portalen nach Zielgruppen

Es gelten die Größen „Nutzerkreis“ und „Themenbreite“

Nutzerkreis

Unterteilung in Konsumenten (B2C) und Unternehmensportale (B2B, B2E (Business-to-Employee)).

Konsumentenportale

Konsumentenportale sind in der Regel offene, horizontale Portale für die Endverbraucher.

Sie unterstützen den Nutzer bei der Suche von Webinhalten; strukturieren, katalogisieren und personalisieren die Webinhalte (Webtracking).

Machen den Nutzer auf Inhalte aufmerksam, nach denen nicht explizit gesucht wurde - wichtiges Werbemedium!

Prominentester Vertreter: Google mit Google+, Google Docs, Google Maps, etc.

Klassifizierung von Portalen nach Zielgruppen

Themenportale

Themenportale werden auch als vertikale Konsumentenportale bezeichnet.

Informationen stehen im Vordergrund, weshalb sie sich auf spezifische Interessengruppen, Themen, eine Branche oder Produktgruppen beziehen und sehr detaillierte Informationen anbieten.

Der Themenschwerpunkt ist meist vom Portalnamen ableitbar (z.B. Stadt Dornbirn Online – www.dornbirn.at).

Sie werden zur Unterstützung von Marketing- und Sales-Maßnahmen verwendet (z.B. www.prad.de)

Horizontale Unternehmensportale 1/2

Unternehmensportale sind geschlossene Portale.

Durch die Authentifizierung werden relevante Inhalte personalisiert und individuell gestaltet.

Die Geschäftsprozesse des Unternehmens werden in einer einheitlichen Ablaufumgebung zur Verfügung zu stellen.

Sie unterstützen die effiziente, sichere und räumlich sowie zeitlich unabhängige Zusammenarbeit der Stakeholder untereinander.

Vgl. (Stelzer, 2004), S. 18, Vgl. (Großmann, Koschek, 2005), S. 33.

Horst Werner Schneider, Seite 24

Klassifizierung von Portalen nach Zielgruppen

Horizontale Unternehmensportale 2/2

Werden immer öfter als Wissensspeicher ausgebaut, in dem aus individuellem Wissen Unternehmenswissen entsteht (Wikis).

Wissen, Knowledge ist die wichtigste Ressource im Zeitalter von Industrie 4.0 (Informatisierung der Fertigungstechnik, Maschine-zu-Maschine Kommunikation)!

Entwicklung erfolgt nach dem Bottom-up Ansatz, d.h. Start mit eingeschränkten Funktionsumfang, gefolgt von einer iterativen Weiterentwicklung entsprechend den Anforderungen (Anm.: hier greift das Anforderungs-Management!)

Sie entstehen deshalb meist aus Vertikalen Unternehmensportalen.

Vertikale Unternehmensportale 2/2

Richten sich zu Beginn an bestimmte Zielgruppen, Anwendungsgebiete oder Unternehmensbereiche.

Bereitstellung von integrierten Anwendungen.

Unterstützung der Nutzer bei der Aufgabenerledigung bestimmter Geschäftsprozesse.

Klassifizierung von Portalen nach Produktart

Framework-Portale

Framework-Portale besitzen eine eigene Entwicklungsumgebung bestehend aus einem Paket von Modulen, Klassen, Funktionen und Schnittstellen.

Sie bieten die meisten Freiheitsgrade für die Entwicklung.

Sie selbst können weitere Frameworks integrieren (ASP.NET integriert das .NET Framework)

Entwicklungsaufwand wird durch vorgefertigte Funktionsmodule reduziert (z.B. IO, Security, etc.)

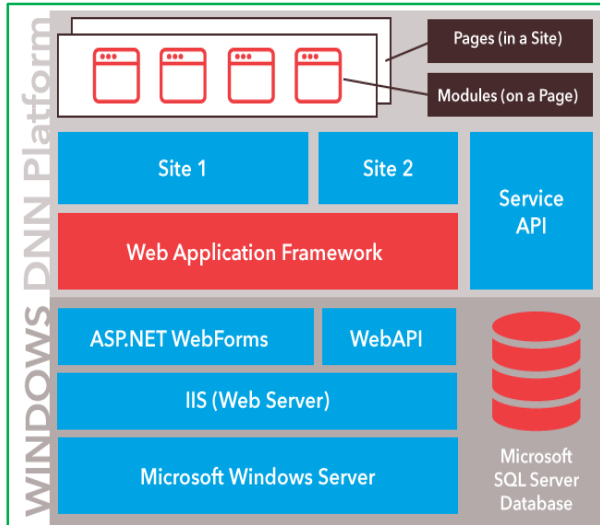
ASP.NET / ASP.NET MVC sind Framework-Portale (Java Pendant: JSP – Java Server Pages)

DNN (ehemals DotNetNuke) bekanntester Web-Content Framework auf .NET Technologie.

DNN Funktionsmodule können über ein CMS (Content Management System) durch den Benutzer administriert werden.

Klassifizierung von Portalen nach Produktart

DNN Plattform



Vgl. <http://www.dnnsoftware.com/Platform/Start/Architecture> (2013), o.S.

Der rot hinterlegte Bereich Web Application Framework steht dem Entwickler zur Anpassung und Erweiterung der Funktionalität zur Verfügung.

Über die Service API lässt sich das Portal mit externen Applikationen verbinden.

Das Content-Management System ermöglicht dem Benutzer die Webseiten und die in die Webseiten eingebundenen Module zu administrieren.

Vorteil:

Hohe Flexibilität bei Entwicklung von webbasierten Softwarelösungen zur Unterstützung der entsprechenden Geschäftsprozesse

Nachteil:

Der gewonnenen Flexibilität steht jedoch der Verlust an Homogenität und Standardisierung gegenüber.

Horst Werner Schneider, Seite 27

Klassifizierung von Portalen nach Produktart

Applikations-Portale

Der Wunsch nach einer heterogenen Landschaft ist in den meisten Fällen ein sehr komplexes Unterfangen.

Hersteller von Applikationen (z.B. SAP, Oracle, Microsoft) bieten deshalb eigene Portallösungen an.

Sie lassen sich leicht in bestehende Informations-Systeme (z.B. ERP-Systeme) des jeweiligen Herstellers integrieren.

Sie bieten Standard-Schnittstellen zu externen Systemen an.

Beispiele:

SAP: SAP Enterprise Portal, Microsoft: Sharepoint Server

Vorteile:

Hohe Flexibilität, Standardisierung und Wiederverwendbarkeit durch Standardisierung und Vernetzungsmöglichkeiten.
Nahtlose Integration der Fachlichkeit – d.h. Unterstützung aller relevanten standardisierten Geschäftsprozesse eines Unternehmens.

Klassifizierung von Portalen nach Produktart

Content-Portale

Content-Portale ähneln in ihrer Art den Informations-Portalen.

Sie fassen einzelne Themen oder verschiedene Themenbereiche aus einer Vielzahl von Quellen strukturiert in einer einheitlichen Oberfläche zusammen.

Informationen werden nicht wie bei Informations-Portalen ausschließlich konsumiert, sondern die Nutzer können Dateien und Webinhalte veröffentlichen und mit anderen Nutzern teilen.

Anwendung in Forschung und Entwicklung, Marketing, Produkt-Management und Verkauf für den Austausch von Informationen im Unternehmen und mit externen Partnern.

Der bekannteste Hersteller eines ECMS (Enterprise-Content-Management-Systems) war Day Software (Schweiz) mit dem Produkt CQ5 – jetzt Adobe Experience Manager!

Nachteile:

Die Integration von Applikationen oder die Abbildung von Geschäftsprozessen über verschiedene Backend-Systeme erweist sich bei Content-Portalen als schwierig.

Web-Architektur - Charakteristika

Besteht häufig aus 4 physische Schichten

Die Benutzeroberfläche wird in einem **Web-Browser** dargestellt.

Der **Web-Server** (z.B. IIS, Apache) verteilt HTML-Dokumente, Multimediaobjekte (z.B. Dateien, Streaming Videos) die durch HTTP Request (z.B. GET / POST) angefragt werden. Er stellt zudem die Kommunikation mit der Anwendungslogik sicher.

Der **Anwendungs-Server** (z.B. ASP.NET MVC, PHP) ist zuständig für die Business Logic (Geschäftslogik).

Der **Datenserver** (z.B. SQL-Server, MySql) ist zuständig für die Verwaltung der persistenten Daten.

Keine permanente Verbindung

Im Gegensatz zu Client-Server Systemen existiert keine permanente Verbindung zw. Web-Client und Web-Server.

HTTP-Protokoll

Bei jeder Benutzeranfrage (Request) wird eine TCP Verbindung (Port: 80) zum Web-Server aufgebaut. Nach der Rücksendung (Response) wird die Verbindung wieder beendet.

Web-Architektur - Charakteristika

Unbegrenzte Anzahl von Benutzern

Kein Einfluss auf Laufzeitumgebung

Die Entwickler haben grundsätzlich keinen Einfluss auf die Laufzeitumgebung des Web-Clients (Ausnahmen: Intranet- und Extranet-Anwendungen).

Einfaches Deployment (Software-Verteilung)

Keine anwendungsspezifische Software muss auf den Clients installiert werden.

Session-Management notwendig

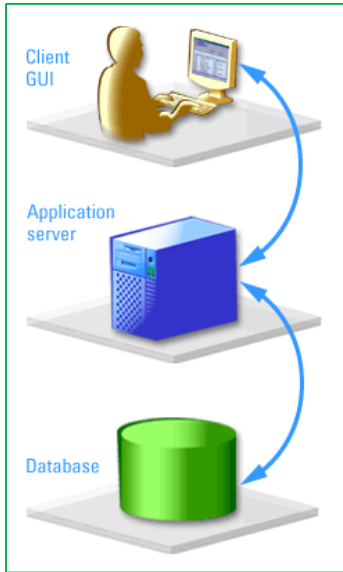
Da keine permanente Verbindung zwischen Web-Client und Web-Server besteht, müssen Zustände während und zwischen Sitzungen (Sessions) gespeichert und verfolgt werden.

Unterstützung unterschiedlicher Browser / Komponenten-Versionen

Unterschiedliche Browser (Safari, IE, Chrome, Firefox, etc.) und Komponenten-Versionen (HTML 4.01, HTML 5, CSS 2.0, CSS 3.0) müssen unterstützt werden.

Horst Werner Schneider, Seite 31

Web-Architektur – n-Tier (Multitier) Softwaresysteme



GUI / Presentation Layer (Präsentationsschicht)

HTTP-Requests (z.B. GET / POST) werden vom Client (Browser) an die zweite Ebene, dem Web-Applikations-Server geschickt.

Web-Application Server (Logikschicht)

Der Anwendung auf dem Webserver enthält die Geschäftslogik, verarbeitet die Requests, validiert die Daten, steuert den Zugriff auf die Datenquelle und sendet die Ergebnisse (HTML-Code) an den Client.

Database (Datenhaltungsschicht)

Verarbeitet die Commands des Web-Servers und liefert die angeforderten Daten an den Web-Server.

Quelle: <http://www.ewebprogrammer.com/java-database-connectivity/module3/ntier-architectural-model.jsp>

Horst Werner Schneider, Seite 32

Web-Architektur – n-Tier (Multitier) Softwaresysteme

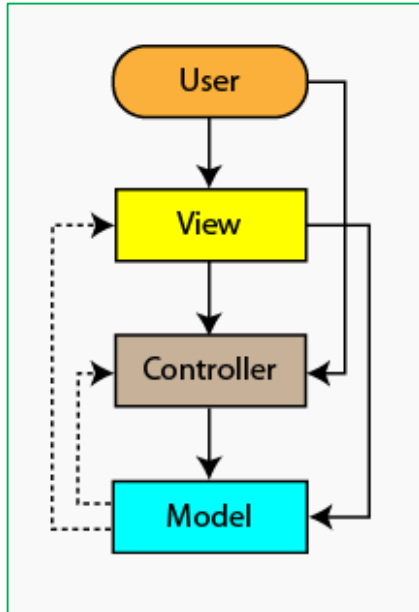
Kommunikation zw. Schichten

Der Datenaustausch zwischen den Schichten ist Teil der Architektur. Mehrere Protokolle werden verwendet: UDP, RESTful API, SOAP, Windows Communication Foundation (WCF), .NET Remoting, ADO.NET, Entity-Framework und andere proprietäre Protokolle.

Vorteile

- Sehr gute **Skalierbarkeit** aufgrund der logisch getrennten Schichten.
- Ermöglicht **verteilte Systemarchitekturen**, d.h. zentrale Datenserver, Logikschicht auf Workgroup-Server und Präsentation auf dem Web-Client
- **Wiederverwendbarkeit** der Schichten
- **Kapselung** der Schichten-Logik
- Möglichkeit der **strikten Trennung** der Schichten, d.h. der Client kann nicht direkt mit dem Datenserver kommunizieren; die Kommunikation erfolgt immer über den Applikations-Server.

Web-Architektur – MVC – Model-View-Controller



Quelle: <http://glossar.hs-augsburg.de/Schichtenarchitektur>

MVC wird als Design-Pattern und als Architekturmodell verstanden.

Ermöglicht die **Wiederverwendbarkeit** der einzelnen Komponenten.

Z.B. wird das Model für mehrere Anwendungen genutzt, nur Controller und View müssen neu implementiert werden.

Beim MVC Model kommen in der Regel nicht-strikte Schichtenarchitekturen zum Einsatz.

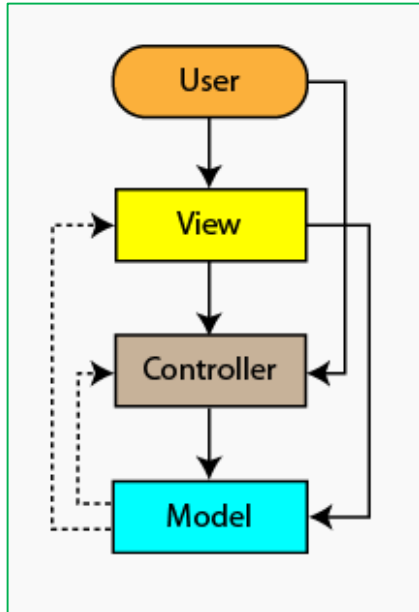
Model (Modell)

Das Model enthält die darzustellenden Daten. Es ist von der Präsentation und Steuerung unabhängig. Es kann die Businesslogik, die Logik für die Internationalisierung der Daten (Zahlenformate, Übersetzungen) und Daten-Validierungslogik enthalten.

Das Model kann die Geschäftslogik enthalten.

Horst Werner Schneider, Seite 34

Web-Architektur – MVC – Model-View-Controller



Quelle: <http://glossar.hs-augsburg.de/Schichtenarchitektur>

View (Präsentation)

Die Präsentationsschicht stellt die Daten aus dem Model dar und nimmt die Benutzerinteraktionen entgegen. In ASP.NET MVC ist das die HTML-Seite. Sie ist nicht für die Weiterverarbeitung der Daten zuständig.

Im View können bereits einfache Validierungen (z.B. `jquery.validate.js`) implementiert werden.

Controller (Steuerung)

Der Controller nimmt die Benutzeraktionen von einer oder mehreren Views entgegen, wertet diese aus und agiert entsprechend. Für jedes View existiert eine eigene Steuerung (Action-Methode).

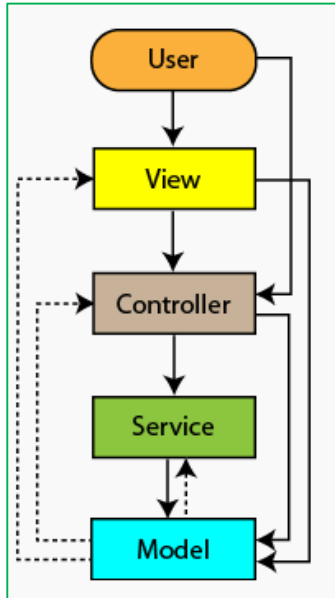
Er ist bei Web-Anwendungen der aktive Vermittler zw. View und Model. Das Observer Muster kommt aufgrund von Request / Response nicht zum Einsatz.

Der Controller instanziert das Model und reagiert auf ein validiertes Model, das fehlerhaft ist (z.B. Redirect auf eine Fehlerseite).

Der Controller implementiert, meist historisch bedingt, die Geschäftslogik.

Horst Werner Schneider, Seite 35

Web-Architektur – MVC – Model-View-Controller



Quelle: In Anlehnung an <http://glossar.hs-augsburg.de/Schichtenarchitektur>

Geschäfts-Logik

Die Geschäfts-Logik muss sich nicht zwingend im Model oder im Controller befinden (siehe Grafik links).

Sie kann in eigene Services (Logikschicht) gekapselt werden, die von MVC strikt getrennt werden. Der Controller greift auf die Geschäfts-Logik mittels Service-Schnittstellen zu.

Die Service-Schicht kann das Model instanziiert und initialisiert (z.B. POJO = Model Klasse).

Sie erkennt Zustandsveränderungen am Model und agiert entsprechend (z.B. CRUD Operationen)

Dadurch kann die Geschäfts-Logik unabhängig von MVC wieder verwendet werden.

Beispiel: Einheitliche Staffelpreis-Berechnung im Auftrags-Modul eines ERP-Systems und im Web-Shop.

MVC entspricht quasi der Präsentationsschicht in einem n-Tier Modell.

Horst Werner Schneider, Seite 36

Web-Architektur – SOA, Serviceorientierte Architekturen

Was ist eine SOA ?

- SOA ist eine IT-Architektur (Prozess-Architektur), bei der die Funktionen der Geschäftsprozesse gekapselt und als Services (Dienste) angeboten werden. Damit stellt SOA die fachlichen Aspekte der Geschäftsprozesse prozessorientiert in den Vordergrund.
- Durch SOA werden IT-Dienste strukturiert.
- Die aus den Prozessen entwickelten Funktionen können mit anderen Funktionen über Standardschnittstellen verbunden werden. Man spricht in diesem Falle von einer Orchestrierung.
- Durch die Orchestrierung können recht flexibel und mit einer größtmögliche Wiederverwendbarkeit von Services niedriger Abstraktionsebenen, Services höherer Abstraktionsebenen geschaffen werden.

Web-Architektur – SOA, Serviceorientierte Architekturen

Was ist eine SOA ?

- SOA wurde 1996 vom Marktforschungs-Unternehmen Gartner Group erstmals als Begriff benutzt.
- Die Services (Dienste) besitzen die folgenden idealtypischen Eigenschaften:
 - Dienst repräsentiert eine fachliche Funktionalität
 - Dienst ist abgeschlossen und kann eigenständig verwendet werden
 - Dienst ist in einem Netzwerk verfügbar (z.B. mittels Webservices)
 - Dienst besitzt eine Schnittstelle
 - Dienst ist plattformunabhängig und in einem Verzeichnis registriert
 - Dienst ist dynamisch gebunden
 - Dienst sollte grobgranular sein, um die Abhängigkeit zwischen verteilten Systemen zu senken.

Horst Werner Schneider, Seite 38

Web-Architektur – SOA, Serviceorientierte Architekturen

Was sind die Nutzen von SOA?

- Höhere Effektivität

SOA stellt die Geschäftslogik in den Vordergrund. Damit wird die bestmögliche Unterstützung der Geschäftsprozesse erreicht.

- Höhere Flexibilität

SOAs erhöhen die Flexibilität der IT und verringern dadurch die Kosten für Wartung, Pflege und Anpassung der IT an neue Anforderungen.

Dadurch wird die fachliche Komplexität reduziert und ein höherer Integrationsgrad kann erreicht werden.

Web-Architektur – SOA, Serviceorientierte Architekturen

Was sind die Nutzen von SOA?

- Höhere Agilität

Ein zentraler Punkt ist die Kapselung der Funktionalität, die als unabhängige Dienste ausgeführt werden können. Damit lassen sich Auswirkungen von Veränderungen stark reduzieren und die einzelnen Dienste lassen sich schneller an neue Anforderungen anpassen.

- Innovation

Die SOA Architektur unterstützt die Entwicklung neuer Geschäftsmodelle und ermöglicht die größtmögliche Freiheit bei der Erschließung neuer Geschäftsfelder. Durch die Wiederverwendbarkeit der SOA-Dienste für unterschiedliche Anwendungen können Zeit und Kosten für die Entwicklung neuer Applikationen gesenkt werden.

Web-Architektur – SOA, Serviceorientierte Architekturen

Was sind die Nutzen von SOA?

- Korrektheit

Die Anwendungslandschaft erfüllt die funktionalen und nicht-funktionalen Anforderungen des gesamten Geschäfts.

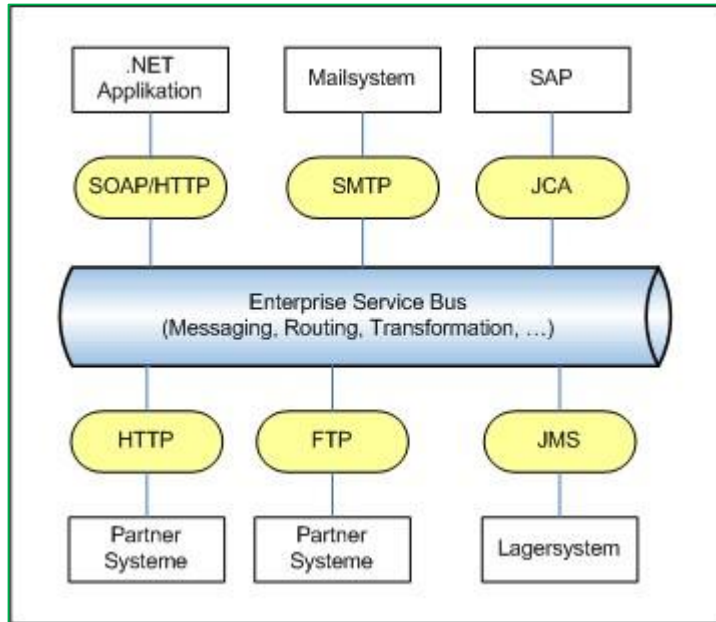
Dateninkonsistenzen werden durch die verbesserte Integration von Anwendungssystemen und durch die Synchronisation der in den einzelnen Geschäftsprozessen verwalteten Informationen vermieden.

- Kostenreduktion

Outsourcing von Teilbereichen führt zu Reduzierung von Strukturkosten

Standardprodukte können eingesetzt werden

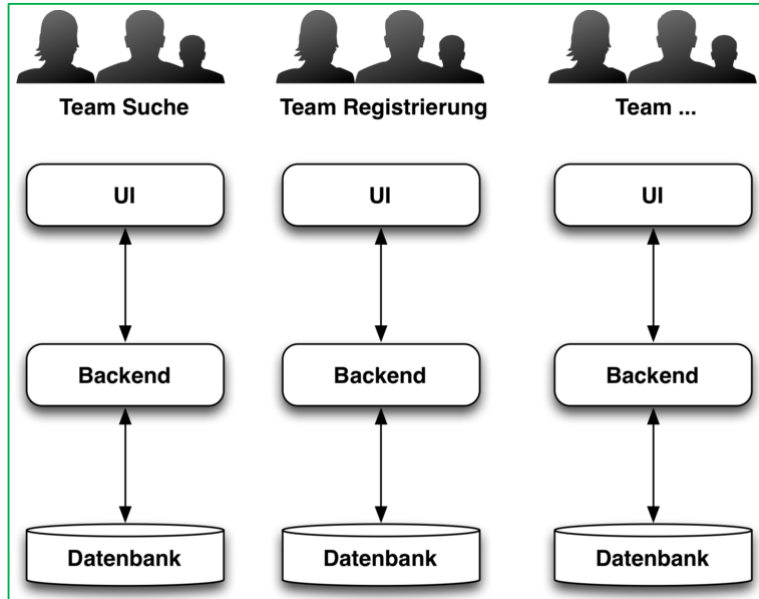
Web-Architektur – Enterprise Service Bus



Der Enterprise Service Bus, kurz ESB, kann als Kommunikationsinfrastruktur verstanden werden, der für die nahtlose Integration der verschiedenen Systeme zuständig ist.

Der ESB verbindet die Bereitsteller und Nutzer von Services, indem die tieferliegenden Kommunikationsprotokolle (HTTP, SOAP, etc...) abstrahiert werden, um den Anwendern einen leichten und standardisierten Zugang zu den verschiedenen Services zu ermöglichen.

Web-Architektur – Microservices



Quelle: Wolff, 2015, <http://www.informatik-aktuell.de/entwicklung/methoden/microservice-architekturen-nicht-nur-fuer-agile-projekte.html>

Ähnlich wie bei SOA wird die Fachlichkeit in der Vordergrund gerückt, d.h. die Architektur orientiert sich nach Fachlichkeiten.

Gesetz von Conway

Eine Organisation kann nur solche Architekturen hervorbringen, die den Kommunikationsstrukturen der Organisation entsprechen.

Aufteilung in Deployment Einheiten

Die Software wird in kleinere Deployment-Einheiten aufgeteilt. Jedes Team sollte seine Fachlichkeit wie z.B. Suche, Registrierung in einer oder mehreren Deployment-Einheiten ausliefern können.

Web-Architektur – Microservices

Deployment Einheiten = Microservices

Merkmale

Jeder Microservice kann eine UI (Frontend) enthalten und Geschäftsprozesse implementieren.

Jeder Microservice repräsentieren eine Geschäftsfunktion (z.B. Suche, Registrierung).

Jeder Microservice kann unabhängig von anderen Microservices getestet und in Produktion gebracht werden. Das ermöglicht Continuous Delivery!

Kommunikation

Die Microservices können über offene Schnittstellen-Standards (z.B. REST) miteinander kommunizieren. Damit kann jeder Microservice in einer anderen Programmiersprache, mit anderen Datenbanken entwickelt werden.

Web-Architektur – Microservices

Vorteile

- Teams können unabhängig voneinander arbeiten
- Microservices sind klein und damit übersichtlich und leicht weiterentwickelbar
- Abhängigkeiten zwischen den Microservices müssen über die API geschaffen werden.
- Unabhängige Skalierbarkeit der Microservices
- Continuous Delivery: Einzelne Microservices können in ein oder mehrere Deployments ausgeliefert werden.
- Microservices sollten gegen den Ausfall einzelner Microservices abgesichert sein – das Gesamtsystem bleibt robust.
- Garantieren eine produktive Entwicklung, da die Microservices wartbar bleiben und die Architektur des Gesamtsystems erhalten bleibt.

Web-Architektur – Microservices

Nachteile

- Die verteilte Architektur erzeugt Komplexität.
Kommunikation über z.B. REST erzeugt Latenzen, Problem der Lastverteilung und Fehlertoleranz.
- Die Vielzahl an Services macht Software-Verteilung und das Testen komplexer.
Test und Deployment muss quasi automatisiert werden.
- Die komplexen Abhängigkeiten eines Deployment-Monolithen verschwinden nicht, sondern existieren auf der Netzwerk-Ebene weiter.
- Redundanter Code
Code kann in mehreren Service vorkommen. Der Code könnte in eine separate Library ausgelagert werden. Dann können aber die Microservices nur noch gleichzeitig veröffentlicht werden.

Web-Architektur – Microservices

Nachteile

- Asynchrone Ausführung

Microservices machen Gebrauch von der asynchronen Programmierung. Das führt zu Problemen, wenn Abläufe synchron oder in Transaktionen ausgeführt werden müssen.

- Testing

Einzelne Microservices können gut getestet werden. In einem dynamischen Umfeld, können die Interaktionen der einzelnen Microservices zu einem subtilen Verhalten führen, das nur schwer greifbar und vorhersehbar ist.