

**Zeitschrift:** Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

**Herausgeber:** Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

**Band:** 92 (2001)

**Heft:** 1

**Artikel:** Dynamische Websites

**Autor:** Riniker, Werner

**DOI:** <https://doi.org/10.5169/seals-855653>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 15.04.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Dynamische Websites

Wieso ist das World Wide Web so ungeheuer dynamisch? So dynamisch, dass es sogar zum Sinnbild einer neuen Wirtschaft, der New Economy, geworden ist? Die Gründe dafür sind vielfältig und auf komplexe Weise miteinander verknüpft. Mit Dynamik in diesem Sinn hat der nachfolgende Artikel jedoch nur am Rande zu tun, obwohl selbstverständlich alle, die auf dem Internetsektor arbeiten, von ihr betroffen sind. Schwerpunkt dieses Artikels sind die grundlegenden Techniken der Applikationsentwicklung von dynamischen Websites, welche heute und vermutlich noch auf längere Zeit die Praxis prägen. Als dynamisch wird in diesem Bericht die Komposition und Darstellung (Rendering) bezeichnet, welche ereignisgesteuert (Event Driven) zur Laufzeit erfolgt. Eine Website gilt dann als dynamisch, wenn ihre HTML-Pages nicht fertig codiert vorliegen, sondern parametrisiert und gezielt für jeden Online-Request generiert werden.

Der technisch interessierte Leser wird vieles bereits kennen, einiges als neu erfahren und anderes in einem neuen Licht betrachten. Der Web-Applikationsingenieur hingegen wird zu den Themen schon diverse Bücher besitzen, welche viel detaillierter auf die einzelnen Komponenten eingehen. Einige wichtige Links sind am Schluss dieses Artikels zu finden.

## Die 4-Stufen-Architektur

### Die Komponenten

Das in den 80er-Jahren dominierende Client-Server-Prinzip (2-Tier) mit dem Fat-Client wurde durch das Thin-Client-Prinzip abgelöst. Der Thin-Client mit Web-Browser benötigt eine skalierbare 4-Tier(4-Stufen)-Architektur, mit der sich die entsprechenden dynamischen Websites bauen lassen (Bild 1).

### Funktion und Aufgaben

Jede der vier Stufen beschreibt eine Funktion und hat verschiedene Aufgaben zu erfüllen:

- Der Browser stellt für den Benutzer das User-Interface dar und ist für die

Eventbehandlung, die Seitendarstellung und die Bearbeitung des User-Inputs zuständig.

Der Web-Server ist für den Benutzer die Resource (URL), an die er seine Anfragen (Requests) stellt. Er leitet diese an die nächste Stufe weiter.

Der Applikations-Server bearbeitet die Anfragen auf Grund der implementierten Business-Logik und veranlasst die nötigen Abfragen an die Datenbank. Er sendet die Antwort mit den Datenbankresultaten als HTML-Code dem Web-Server zurück.

Der Datenbank-Server verwaltet alle zur Applikation gehörenden Daten und liefert die Resultate entsprechend den Abfragen zurück.

## Technologien

Verschiedenste Produkte und Technologien lassen sich auf der jeweiligen Stufe einsetzen:

1. Internet Explorer, Netscape, Opera
2. Apache, Internet Information Server, Netscape
3. ASP, Coldfusion, C#, Java (Servlets, JSP), PHP, Python, Oscript (Livelink), Web-Objects
4. 4D, DB2, Filemaker, MS-SQL, MySQL, Oracle (4D, Filemaker und Oracle mit eigenem Web-Server)

## Multiple-Tier-Architektur

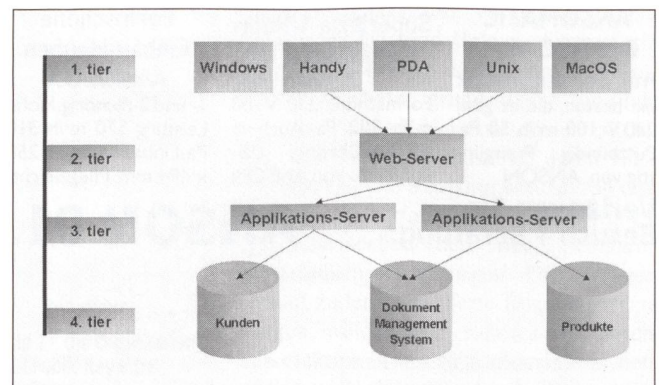
Die 4-Tier-Architektur kann wie folgt funktional weiter aufgeteilt werden und wird damit physikalisch besser skalierbar:

- 1 Browser
- 2 Firewall: schützt gegen Attacken
- 3 Load-Balancing: nimmt Lastverteilung vor
- 4 Web-Server
- 5 Caching: führt Zwischenspeicherung von oft verlangten Seiten aus
- 6 Page-Generator: generiert die Seiten aus Struktur, Inhalt und Darstellung
- 7 Business-Logic-Server: erzeugt Abfragen und generiert den Inhalt auf Grund der Business-Cases
- 8 Integration-Server/Transaction-Manager: stellt Verbindung und Kontrolle zu bestehenden Systemen her
- 9 Security-Manager: kontrolliert die Zugangs- und Zugriffsrechte zur Website bzw. zum Web-Content (Web-Datenbestand)
- 10 Query-Prozessor: führt komplexe SQL-Statements auf Grund einer Business-Anfrage aus
- 11 Datenbank-Server

### Adresse des Autors

Werner Riniker, e-Content Management  
Three Way AG, 5210 Windisch  
w.riniker@3way.ch

Bild 1 Die generelle 4-Stufen-Architektur für dynamische Websites



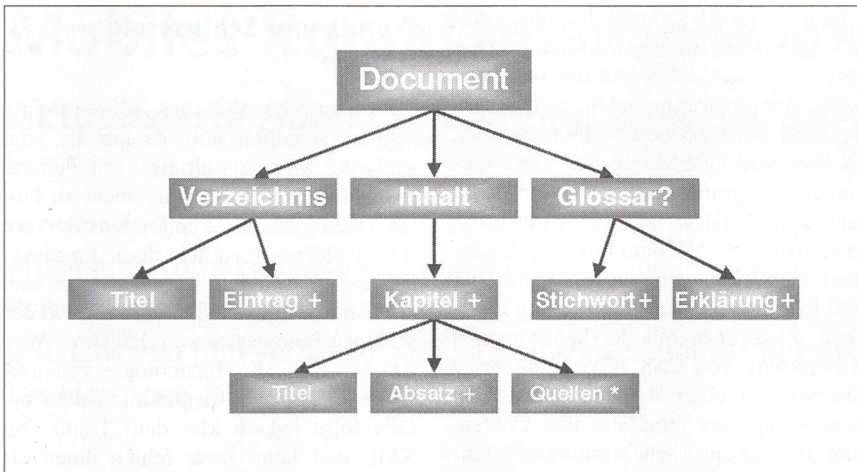


Bild 2 Baumdarstellung (DOM) eines Dokumentes mit DTD-Syntax

DTD (Document Type Definition) und DOM (Document Object Model) stehen in engem Zusammenhang zueinander. Die Syntax hat folgende Bedeutung: ? = kein oder ein Element; + = ein oder beliebig viele Elemente; \* = kein oder beliebig viele Elemente; kein Sonderzeichen = ein Element. Der XML-Schema-Standard, der die DTD ablösen wird, besitzt eine XML-Syntax und ermöglicht zudem die Zuweisung von Datentypen.

- 12 Administrations-Server: verwaltet die Applikationsdaten
- 13 Web-Content-Management-System (WCMS): verwaltet Inhalt und Darstellungsinformationen

**Bemerkungen zur Multiple-Tier-Architektur**

- Traffic-Filtering und Load-Balancing sind für die Sicherheit und optimale Verfügbarkeit der Server zuständig.
- Das Caching dient zur Verbesserung des Antwortverhaltens und ist ähnlich komplex wie das Caching auf CPU-Ebene.
- Der Applikations-Server hat sich natürlich nicht aufgelöst, sondern wurde in vier spezialisierte Funktionseinheiten unterteilt (Page-Generator, Business-Logic-Server, Integration-Server/Transaction-Manager, Security-Manager).
- Während der Integration-Server mit «fremden» Systemen (ERP, PDS usw.)

kommuniziert, bedient der Query-Prozessor die applikationseigene Datenbank.

- Zudem wird die Architektur durch ein Web-Content-Management-System (WCMS) ergänzt, welches die Verwaltung von Inhalten (z.B. News, Produktebeschreibungen usw.) erlaubt.

**Technologien**

Firewalls und Load-Balancing-Systeme sind oft dedizierte «Boxen» oder Softwarelösungen, welche vornehmlich auf einem spezialisierten Linux-Kernel basieren. Die Funktionsebenen fünf bis neun sind meistens Komponenten einer umfassenden Web-Entwicklungsplattform wie Coldfusion Enterprise Server von Allaire, Orion von Evermind, Silverstream von Silverstream, Total E-server von Bluestone, Weblogic Server von Bea, Web-Objects von Apple oder Websphere von IBM und nicht zu vergessen der unter Open Source entwickelte Enhydra

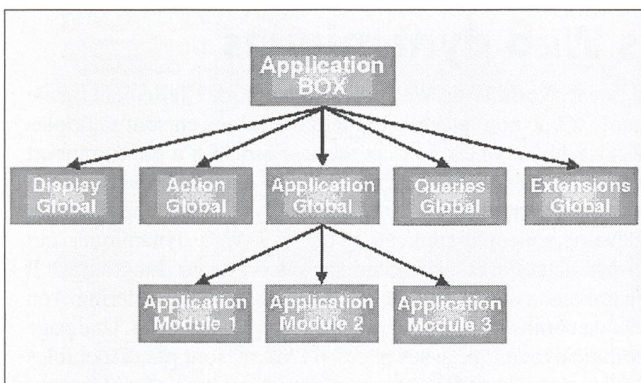


Bild 3 Aufgabenorientierte Applikationsaufteilung kombiniert mit wiederverwendbaren globalen Prozeduren charakterisieren das Fusebox-Programmiermodell.

Server. Die meisten von ihnen benutzen die Funktionen einer Enterprise-Java-Beans(EJB)-Klassenbibliothek. Der Integration-Server verfügt über die applikationsspezifischen Datenbankabfragen, die mittels ODBC (Open Database Connection), JDBC (Java Database Connection), XQL (Extensible Query Language) oder «native» (bei proprietären DB-Schnittstellen) an die Zielsysteme adressiert werden. Zur Leistungsverbesserung können zudem noch datenbankspezifische Features (z.B. Stored Procedures, Caching) benutzt werden.

**Web-Content-Management-Systeme (WCMS)**

**WCMS reduzieren Unterhaltskosten**

Immer mehr Unternehmungen stellen immer mehr Informationen auf ihrer Website zur Verfügung, um personalintensive Support- und Verkaufsanfragen ihrer Kundschaft besser bewältigen zu können. Neben der Menge nimmt auch die Komplexität und Intensität der Informationsbearbeitung zu. Das Management einer Website benötigt deshalb dringend spezialisierte Tools, um den täglichen Anforderungen gewachsen zu sein. Der Unterhalt von grösseren Websites oder Intranets mittels eines HTML-Editors ist zunehmend unrealistisch.

**Anforderungen an ein WCMS**

Ein WCMS sollte endbenutzertauglich sein und folgende Anforderungen erfüllen:

1. Möglichkeit der Bedienung mittels Web-Browser
2. Rollenbasierte (z.B. Autor, Leser, Redaktor) Zugriffsrechte auf Komponenten- und Funktionsebene
3. Minimaler HTML-Code-Anteil im Content-Manager
4. HTML-Code-Kenntnisse von Autoren nicht gefordert
5. Freigabemechanismus

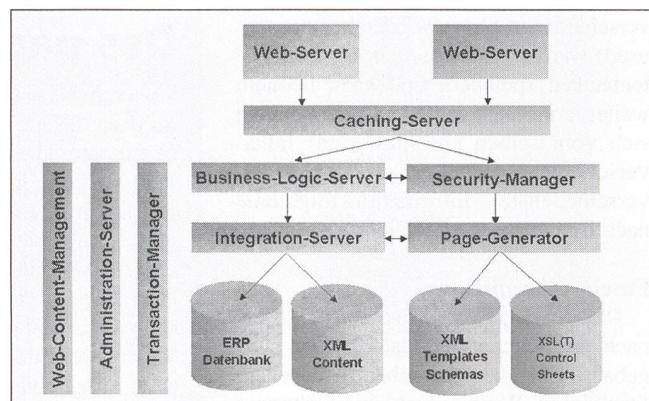


Bild 4 Komponenten einer umfassenden Web-Applikationsplattform mit XML-Technologie

**WCMS-Lösungen**

Neben generell verfügbaren Tools wie Spectra von Allaire verfügen Firmen, die auf dynamische Websites spezialisiert sind, über eigene WCMS, welche als Teil der Gesamtlösung auf die Kundenbedürfnisse zugeschnitten werden. Auch die Three Way AG besitzt auf der Basis der Applikationsumgebung Coldfusion von Allaire eine WCMS-Lösung (Innowaytor).

**Die Trennung von Struktur, Inhalt und Darstellung**

Wie bereits von Frank Jung im *Bulletin SEV/VSE 9/2000* beschrieben, basiert die XML-Technologie auf dem Grundsatz der Aufspaltung eines Dokumentes in Strukturangaben, Inhalt und Darstellung. Jedes XML-Dokument kann als DOM (Document Object Model) geparkt (gelesen, analysiert und auf Fehler überprüft) und mit XSL (Extensible Stylesheet Language) verarbeitet werden. Die Validierung gegen die referenzierte DTD (Document Type Definition) sichert die Vollständigkeit und Gültigkeit eines Dokumentes. Dieser universelle und logische Ansatz der Dokumentenbehandlung erlaubt die dynamische Wiederverwendung (Reuse) von Inhalten in neuen Strukturen und Darstellungen.

Dass in einer HTML-Page alles miteinander «gemischt» wird, mag ein Grund für den schnellen Erfolg von HTML sein, wirkt sich aber auf den Unterhalt von grösseren Sites katastrophal aus. Deshalb wird schon lange versucht, Websites unabhängig von XML wartungsfreundlicher zu machen. Noch schlimmer als die üblichen HTML-Websites sind Word- und andere Applikationen, wo Grafiken und Text ineinander «verwurstet» werden.

**Vorteile der Separierung**

Durch die Separierung und Strukturierung des Inhalts (Content) kann dieser an verschiedenen Orten wieder benutzt (re-used) werden. Er lässt sich leicht in Datenbanken speichern und kann bequem weiterverarbeitet werden. Zudem lassen sich vom selben (strukturierten) Inhalt verschiedene Layouts generieren und verschiedenste Informationsfragmente nach Bedarf herausdestillieren.

**Fusebox-Architektur**

Das Fusebox-Programmiermodell ist nach dem Sicherungskastenprinzip aufgebaut. Ein Haus hat überall dieselben Steckdosen. Wenn jedoch eine Sicherung für den Estrich defekt ist, hat die Küche immer noch Strom. Das Fusebox-Pro-

grammiermodell verlangt Templates mit klar getrennten Funktionen (Action, Display usw.) und die Schaffung von logischen Applikationsmodulen, welche alle mit dem Master-Modul verbunden sind. Es lässt sich unabhängig von einer speziellen Programmiersprache einsetzen und wurde als Erstes für Coldfusion implementiert. Mit dem Fusebox-Ansatz lässt sich eine weitgehende Isolierung von Inhalt, Logik und Darstellung erreichen. Zusammen mit der konsequenten Anwendung von CSS (Cascading Style Sheets) und einer datenbankgestützten Verwaltung von Templates und Content führt dies zu einer sehr wartungseffizienten Website. Innowaytor von Three Way AG arbeitet nach diesem Prinzip.

**XML/Java-Architekturkonzept**

Das nachfolgende Konzeptmodell enthält die wichtigsten Komponenten, welche eine Web-Applikationsplattform heute bieten muss. Anstelle der XML-, XSL(T)-Datenbanken können auch relational angelegte Datenbestände und HTML/CSS-Templates verwendet werden. Ausserdem kann der Business-Logic-Server statt mit EJB auch in C++ implementiert sein.

Durch die volldynamische Page-Generierung mittels XSL(T)-Transformern wird eine hohe Flexibilität erreicht. Demgegenüber werden an das Leistungsvermögen der zu Grunde liegenden Hardwarearchitektur hohe Anforderungen gestellt. Das Clustering von Systemen und das Caching von früher aufgerufenen Pages kann allfällige Performance-Engpässe beheben helfen.

**Trends und Schlussfolgerungen**

Dynamische Websites sollten heute selbstverständlich sein, da nur sie eine saubere Site-Verwaltung mit einem effizienten Content-Management zu bieten vermögen. Mit Applikations-Servern wie Coldfusion unter dem Fusebox-Programmiermodell lassen sich grosse, einfach verwaltbare Sites bauen, wie der schlanke Innowaytor von der Three Way AG beweist. Die Technologie von umfangreichen Web-Applikationsplattformen folgt jedoch klar dem Trend von XML und Java. Zwar fehlen ihnen oft noch benutzerfreundliche Frontends, und die Integration der Module lässt noch zu wünschen übrig. Ein Grund dafür mag die extreme Entwicklungsdynamik der Web-Technologie sein, die ein Reifen der Systeme zurzeit noch verunmöglicht. Doch mit Enhydra und Cocoon bestehen Open-Source-Projekte, die immer wieder neue Techniken für Web-Application-Server entwickeln und die Lösungsansätze vorwegnehmen, welche in kommerziellen Systemen erst viel später ihren Niederschlag finden.

**Links**

- [www.allaire.com](http://www.allaire.com)
- [www.fusebox.com](http://www.fusebox.com)
- [www.javamagazin.de](http://www.javamagazin.de)
- [xml.apache.org](http://xml.apache.org)
- [www.xslt.com](http://www.xslt.com)
- [www.w3c.org](http://www.w3c.org)
- [www.3way.ch](http://www.3way.ch)

**Des pages Web dynamiques**

Comment se fait-il que le World Wide Web soit si dynamique? Tellement même qu'il est devenu symbole d'une nouvelle économie. Les raisons en sont multiples et ont entre elles des relations complexes. Cependant, l'article n'a qu'un rapport marginal avec le dynamisme bien que celui-ci concerne à vrai dire tous ceux qui travaillent dans le domaine Internet. L'essentiel de cet article concerne les techniques de base du développement d'application de sites Web dynamiques qui caractérisent la pratique actuelle et probablement encore pour longtemps. Il considère comme dynamique la composition et la présentation (le «rendering») en fonction du temps, sous la commande des événements («event driven»). Une page Web est considérée comme dynamique si ses pages HTML ne sont pas disponibles sous forme codée d'avance mais générées de manière paramétrée et ciblée pour chaque Online Request.