



7363 - Web-basierte Anwendungen

Eine Vertiefungsveranstaltung
mit Schwerpunkt auf XML-Technologien



AJAX

Asynchronous JavaScript and XML

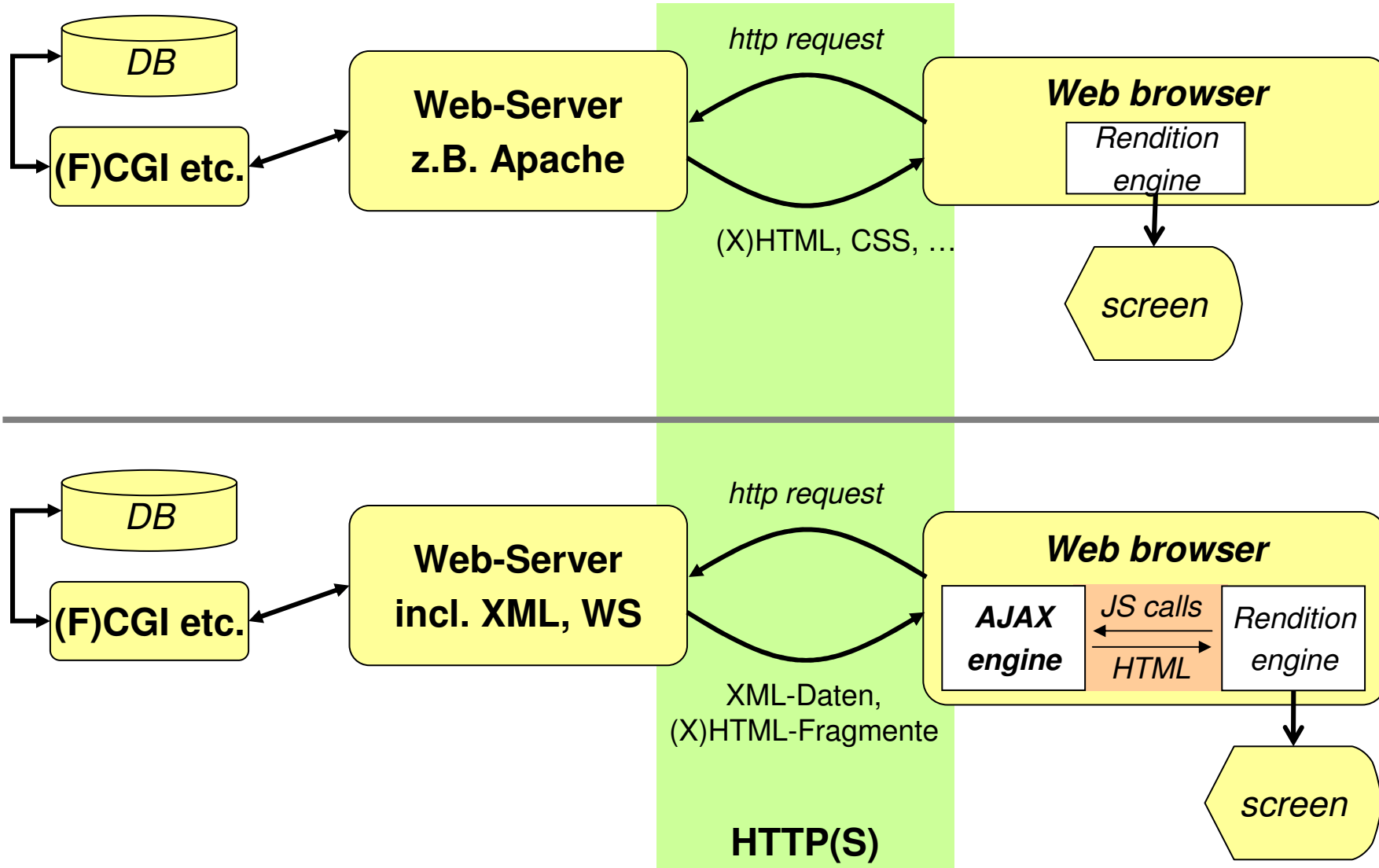


Einführung

Interaktivere Benutzerschnittstellen im Web
mit Ajax

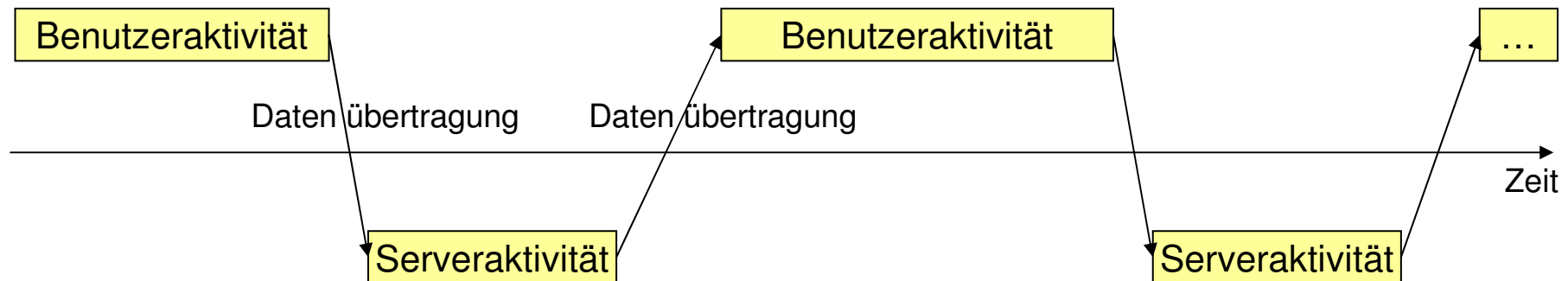
- Ajax – Eine Mischung bekannter Techniken:
 - Präsentation von Information auf der Basis von Standard, insbesondere von **XHTML** und **CSS**
 - Dynamische Anzeigen und Interaktion mit den Inhalten mittels **DOM** (*Document Object Model*)
 - Datenrepräsentation und –transformation mit **XML** and **XSLT**
 - Asynchroner Datenaustausch mit **XMLHttpRequest**
 - und **JavaScript**, um all dies zu verbinden
-

WBA: Ajax vs. traditionelle Interaktionen





Ajax: Synchrone Datenübertragung



Traditionelle WBA wechseln zwischen Benutzer- und Server-Aktivitäten

Anwender empfinden die entstehenden Wartezeiten als störende Unterbrechungen ihres Arbeitsflusses.



Ajax: Alternative Implementierungen



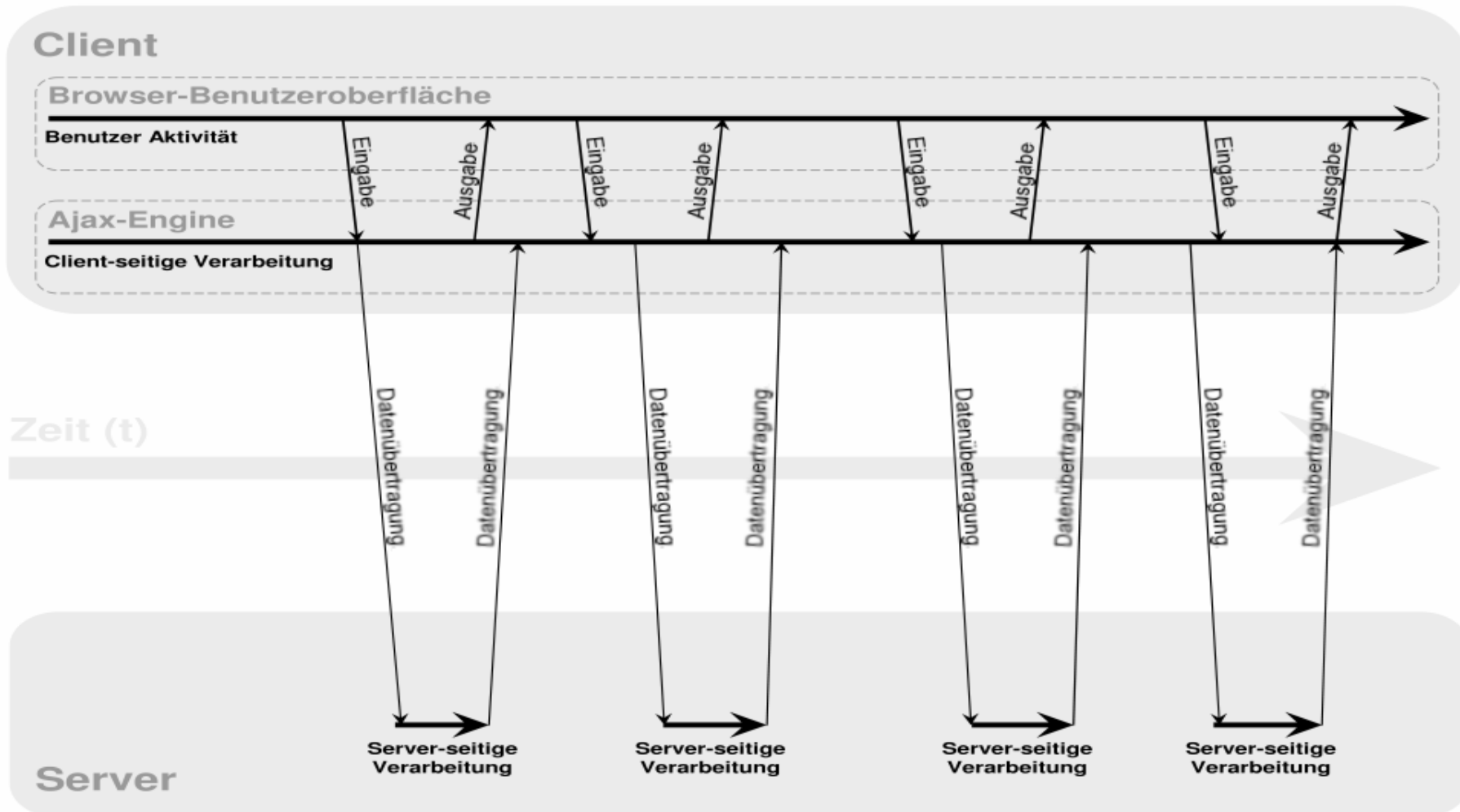
- Direkte Ajax-Implementierung
 - *Client* besitzt API zur XML-basierten Kommunikation mit dem Server (XMLHttpRequest, evtl. mit SOAP-Inhalten)
 - Datenaustausch effizient und flexibel,
 - Transformation erforderlich, komplexer *client*
- Indirekte Ajax-Implementierung
 - Client tauscht HTML-Fragmente mit Server aus
 - Client aktualisiert Darstellung mittels DOM
 - Beispiele:
 - Einfügen / Aktualisieren von Listeneinträgen,
 - Positionen eines Warenkorbs, einer Bestellung etc.
 - Ein solcher Mechanismus ist bereits in Rails implementiert.



Ajax



Ajax-Modell einer Web-Anwendung (asynchrone Datenübertragung)



Quelle: de.wikipedia.org



- Beispielanwendungen
 - *Google Maps*: <http://maps.google.de>
 - „Gleitende“ Verschiebung des Sichtbarkeitsfensters auf die Karte,
 - vorausschauendes Nachladen der nächsten Kacheln
 - Ähnliche Wirkung mit Java Applets: Stadtplan Wiesbaden.
 - Gegenbeispiel: <http://stadtplan.frankfurt.de>
 - *Google Suggest*: <http://www.google.com/webhp?complete=1&hl=en>
 - Sofort angebotene Auswahlliste von Suchbegriffen allein aufgrund der bisher eingetippten Zeichen.



- **XMLHttpRequest: Details**

- Aktuelle Quelle: W3C-Entwurf vom 27. September 2006,
<http://www.w3.org/TR/XMLHttpRequest/>

- Code-Beispiel:

```
var xmlhttp = new XMLHttpRequest();
if (xmlhttp) {
    xmlhttp.open('GET', 'beispiel.xml', true);
    xmlhttp.onreadystatechange = function () {
        if (xmlhttp.readyState == 4) {
            alert(xmlhttp.responseText);
        }
    };
    xmlhttp.send(null);
}
```

- **Kleine JS/DOM-Demo**



cd Ajax-XMLHttpRequest

«interface»

XMLHttpRequest

- onreadystatechange: *Function*
- readyState: *short*
- responseText: *DOMString*
- responseXML: *Document*
- status: *short*
- statusText: *DOMString*

- + *abort()* : *void*
- + *getAllResponseHeaders()* : *DOMString*
- + *getResponseHeader(DOMString)* : *DOMString*
- + *open(DOMString, DOMString)* : *void*
- + *open(boolean, DOMString, DOMString)* : *void*
- + *open(DOMString, boolean, DOMString, DOMString)* : *void*
- + *open(DOMString, DOMString, boolean, DOMString, DOMString)* : *void*
- + *send(DOMString)* : *void*
- + *send(Document)* : *void*
- + *setRequestHeader(DOMString, DOMString)* : *void*

Quelle: de.wikipedia.org



Ajax: XMLHttpRequest



```
interface XMLHttpRequest {
    attribute EventListener onreadystatechange;
    readonly attribute unsigned short readyState;
    void open(in DOMString method, in DOMString url);
    void open(in DOMString method, in DOMString url,
              in boolean async);
    void open(in DOMString method, in DOMString url, in boolean async,
              in DOMString user);
    void open(in DOMString method, in DOMString url, in boolean async,
              in DOMString user, in DOMString password);
    void setRequestHeader(in DOMString header, in DOMString value);
    void send();
    void send(in DOMString data);
    void send(in Document data);
    void abort();
    DOMString getAllResponseHeaders();
    DOMString getResponseHeader(in DOMString header);
    readonly attribute DOMString responseText;
    readonly attribute Document responseXML;
    readonly attribute unsigned short status;
    readonly attribute DOMString statusText;
};
```

Quelle: <http://www.w3.org/TR/XMLHttpRequest/>



- Problemgebiete
 - Verletzung des seitenorientierten Aufbauprinzips
 - Durch das dynamische Verhalten von AJAX-Anwendungen funktionieren *Back button* und Lesezeichenverwaltung des Browsers nicht mehr (bzw. nicht mehr wie erwartet).
 - Problem analog zu früheren Problemen mit Frames
 - Wahrnehmung dynam. Änderungen innerhalb einer Seite durch die Anwender??
 - Aktueller Artikel dazu: Frank Puscher, Klarheit trotz Ajax, c't 2/2007.

Auswege:

- Beschränkung von AJAX-Funktionen auf (kleine) Funktionsgruppen innerhalb einer nach wie vor als „Seite“ wahrgenommenen Einheit
- Verwendung von *back button* und Lesezeichen zwischen diesen „Seiten“, Verzicht auf diese Elemente innerhalb einer Gruppe.
- Verlagern von Ajax-Aktivitäten in unsichtbare „iframes“ in statischer HTML-Seite
- Dynam. geänderte Seitenbestandteile (vorübergehend) farblich kennzeichnen.
- Barrierefreies Internet?
 - Auch mit Ajax angereicherte Seiten sollten sich vorlesen lassen können

- Problemgebiete (Forts.)
 - *Polling*-Problem
 - Web Server unterliegen dem C/S-Modell – sie können den Client nicht „zurückrufen“!
 - Asynchrones Verhalten des XMLHttpRequest-Objekts wird durch Nebenläufigkeit (*multi-threading*) erreicht. Diese zusätzlichen Threads existieren länger als bei normalen C/S-Anfragen und binden Ressourcen auf Client-Seite entsprechend länger.
 - Clients können durch ungeschicktes Vorgehen (häufiges *polling*) neue, erhebliche Serverlasten verursachen. Auch kann sich die Anzahl gleichzeitig offener TCP-Verbindungen des Servers erhöhen.
 - Gelegentlich störend:
 - Download der für Ajax benötigten JS-Bibliotheken: Dauer hinderlich?
 - JS aktiviert? Benötigte JS-Funktionen durch *Client* freigegeben?
 - Code-Weichen für Client-Abhängigkeiten, insb. MSIE vs. Firefox & Co.
 - Konkurrenz-Standard: DOM Level 3 Load & Save Spec., REC 7. April 2004
 - <http://www.w3.org/TR/DOM-Level-3-LS>



- Ajax und Mitbewerber
 - Für die Umsetzung sogenannter *Rich Internet Applications* (RIAs) gibt es neben Ajax auch weitere Optionen:
 - Flash
 - Proprietäre, heute weit verbreitete Technik von Adobe Systems, Inc.
 - Ähnliche Möglichkeiten und Probleme
 - Konkurrierender W3C-Standard:
 - DOM Level 3 Load & Save Specification, 7. April 2004 (Status: REC), siehe <http://www.w3.org/TR/DOM-Level-3-LS>
 - Noch wenig verbreitet. Unterstützung durch Browser?

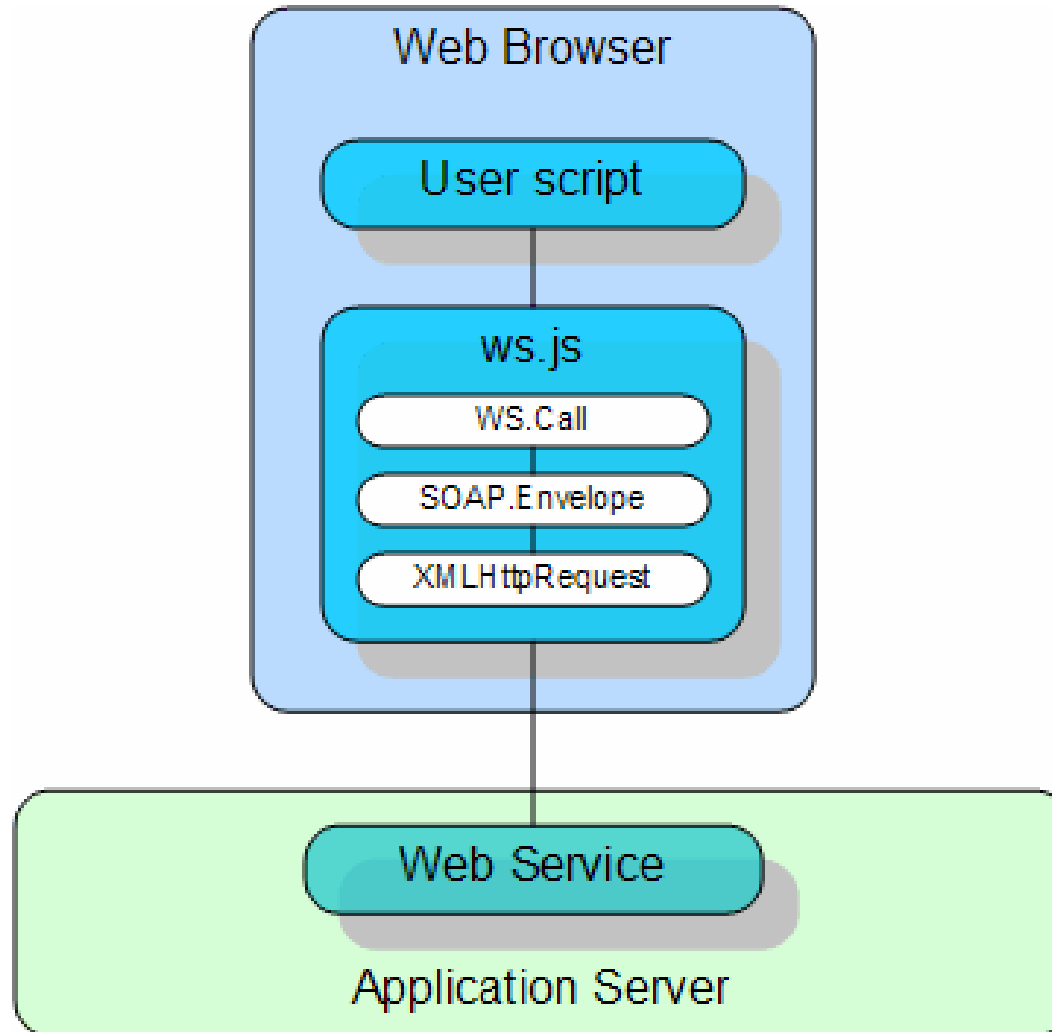


Ajax und SOAP



- Ajax und SOAP?
 - ... sollten doch ideal zueinander passen ☺
- Lösungsansatz:
 - SOAP-Dokument („*envelope*“) wird *client*-seitig erzeugt.
 - Es wird mittels XMLHttpRequest an einen WS-Provider *per http binding* übertragen.
 - Aus Sicht des *WS providers* handelt es sich um eine normale *WS consumer*-Anfrage, die mit einem geeigneten *envelope* beantwortet wird.
 - *Client*-seitig ist dann zwischen SOAP und der JavaScript-Ebene zu vermitteln.
 - Diese Vermittlung JS \leftrightarrow SOAP überlässt man Bibliotheksfunktionen, z.B. denen von „ws.js“ von IBM:
 - <http://www-128.ibm.com/developerworks/webservices/library/ws-wsajax/>





Quelle: <http://www-128.ibm.com/developerworks/webservices/library/ws-wsajax/>



- [1] [http://de.wikipedia.org/wiki/Ajax_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung))
– Sehr kompetenter, gut verständlicher und weiterführender Übersichtsartikel!
- [2] Jesse James Garrett: *Ajax: A New Approach to Web Applications*. Adaptive Path LLC, 18. Februar 2005,
<http://www.adaptivepath.com/publications/essays/archives/000385.php>
– Der Artikel, der die Bezeichnung AJAX nachhaltig prägte.
- [3] <http://www.openajax.net/>
– Beispiel- und Linksammlung
- [3] Drew McLellan: *Very Dynamic Web Interfaces*. 9. Februar 2005,
<http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>
– Ein Artikel u.a. mit Details zum Umgang mit XMLHttpRequest



- Mögliches Einsatzgebiet
 - *Ergebnislisten-Interface, Suche nach Name*
 - Eingabefeld für Name, „Suchen“-Button
 - Tabelle oder Textfeld mit den ersten ca. 10 Treffern
 - Mit jedem eingetippten Zeichen wird die Trefferliste aktualisiert:
 - Anzeige = Die ersten 10 Namen, die mit den eingetippten Zeichen beginnen!
 - *Ergebnislisten-Interface, Suche nach Verein/Ort*
 - Analog zur Namenssuche
- Hinweise
 - Rein als Anregung zu verstehen, keine Sonderwertung, kein Pflichtpunkt!
 - Realisierung sollte mit den Ajax-Möglichkeiten von **Rails** möglich sein.
 - Indirekte Ajax-Implementierung genügt hier (Einfügen auf HTML-Ebene).
 - Selbst die großen Stadtmarathon-Seiten bieten diesen Komfort bisher nicht!