Ajax

asynchronous JavaScript and XML

Christian Marienfeld | 2016

History

- 1996

  the iframe tag was introduced by Internet Explorer to load or to fetch content asynchronously

- 1998

  Microsoft Outlook Web App team implemented the first component XMLHTTP by client script

- 1999

  XMLHTTP ActiveX control in Internet Explorer 5, which was later adopted by other browsers as the XMLHttpRequest JavaScript object

- 2004/05

  Google made a wide deployment of standards

which was later adopted by other browsers as the XMLHttpRequest JavaScript object

- 2004/05

  Google made a wide deployment of standards-compliant, cross browser Ajax with Gmail and Google Maps

- 2005

  The term "Ajax" was publicly stated on 18 February 2005 by Jesse James Garrett in an article titled "Ajax: A New Approach to Web Applications", based on techniques used on Google pages

- 2006

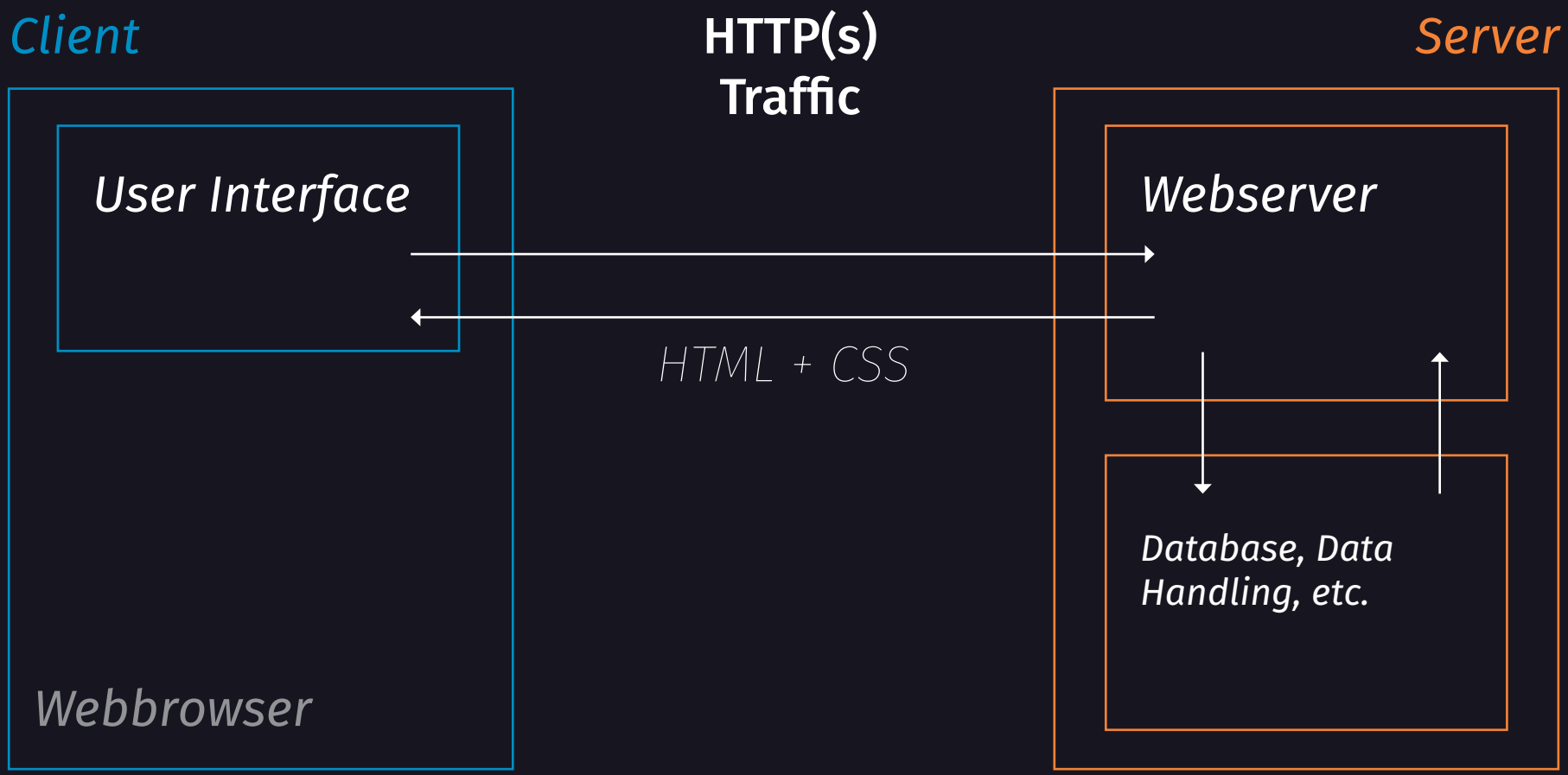
  World Wide Web Consortium (W3C) released the first draft specification for the XMLHttpRequest object in an attempt to create an official Web standard.
  The latest draft of the XMLHttpRequest object was published on 30 January, 2014

Ajax is not a technology,
but a group of technologies!

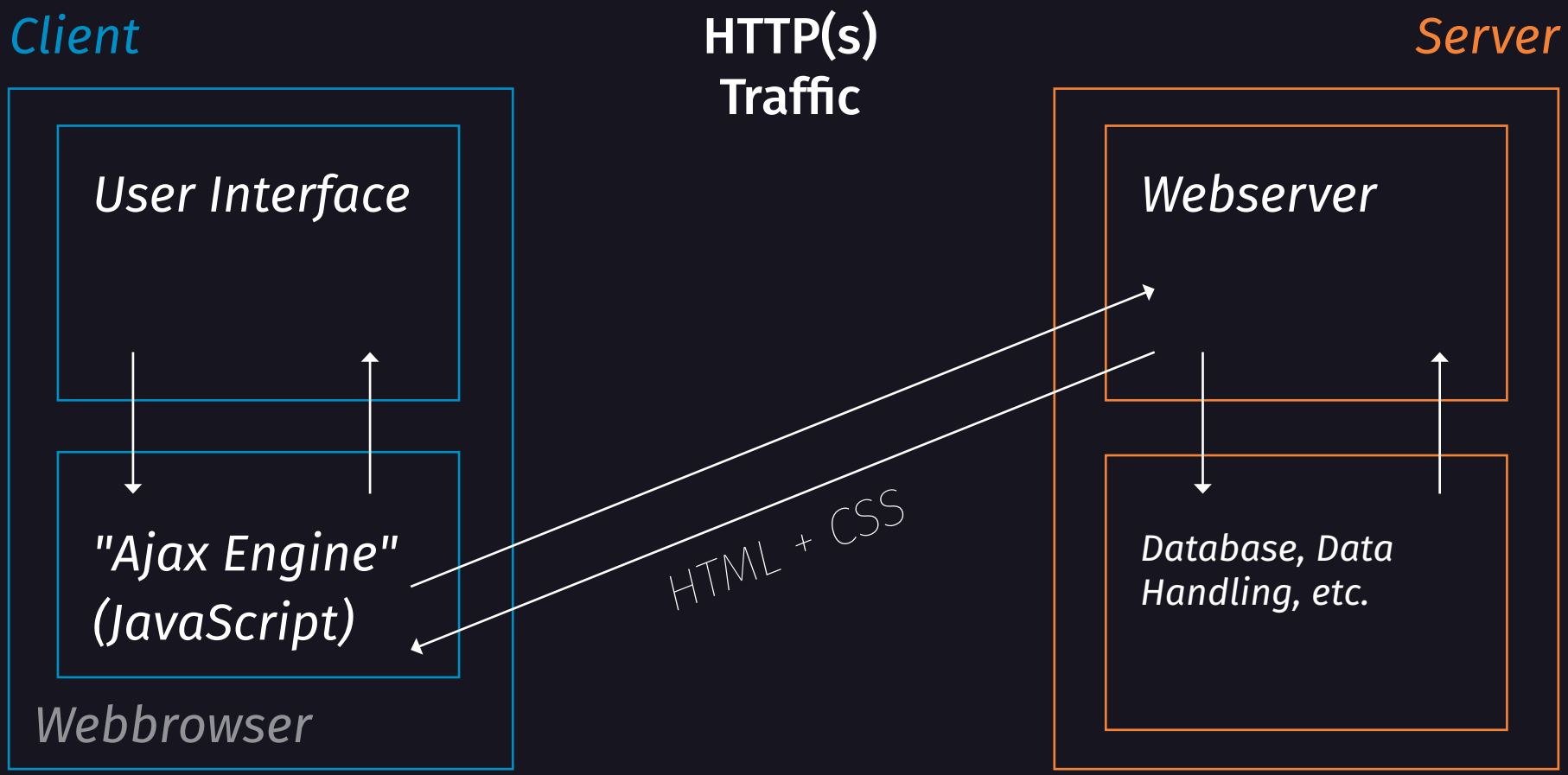
Technologies

- HTML + CSS — *for presentation*
- DOM — *for dynamic display*
- JSON or XML — *for the interchange of data*
- XMLHttpRequest — *object for asynchronous communication*
- JavaScript — *to bring these technologies together*

Classic Model



Ajax Model


Client
HTTP(s)
Traffic
Server
User Interface
Webserver
"Ajax Engine"
(JavaScript)
HTML + CSS
Database, Data
Handling, etc.
Webbrowser

Pro

- Kein Neuladen aufgebauter Seiten
- Kein Browser-Plugin wird benötigt

Contra

- Umfangreiche Tests erforderlich
- Serverseitige Browsererkennung
- Verwendung der „Zurück“-Schaltfläche
- Lesezeichen setzen
- Rückmeldung/ Latenzzeit
- Barrierefreies Internet
- Suchmaschinen/Deep Links

Verbreitung

	seit Version	
IE	5.0 / 7.0	
Firefox	1.0	Auch alle anderen auf Gecko basierenden Browser.
Safari	1.2	Apples Browser Safari basiert auf WebKit, einem KHTML Fork.
Chrome	0.2	Google Chrome basiert bis Version 27 auf Apples WebKit, seit Version 28 kommt die WebKit-Abspaltung Blink zum Einsatz.

Und Wie jetzt?

```
XMLHttpRequest
```

Methods

```
open // create a connection
```

```
send // send a request to the server
```

Und Wie jetzt?

```
XMLHttpRequest
```

Attributes

```
responseText // holds loaded data as a string
```

```
responseXml // holds an XML loaded file
```

```
status // 200 is OK - 404 page is not found
```

Und Wie jetzt?

```
XMLHttpRequest
```

Attributes

```
onreadystatechange // property for event handling
```

```
readyState // the state of availability of data
```

```
0: not initialized.
1: connection established.
2: request received.
3: answer in process.
4: finished.
```

1. CREATE AN INSTANCE

```
var xhr;
if (window.XMLHttpRequest) {   // Standard object

    xhr = new XMLHttpRequest();   // Firefox, Safari, ...

} else if (window.ActiveXObject) {  // Internet Explorer

    xhr = new ActiveXObject("Microsoft.XMLHTTP");
}
```

or...

```
try {

    // Trying IE
    xhr = new ActiveXObject("Microsoft.XMLHTTP");

} catch(e) {   // Failed, use standard object

  xhr = new XMLHttpRequest();

}
```

2. WAIT FOR THE RESPONSE

```
xhr.onreadystatechange = function() {

    if (xhr.readyState == 4) {
       // Received, OK
    } else {
      // Wait...
    }

}
```

3. MAKE THE REQUEST ITSELF

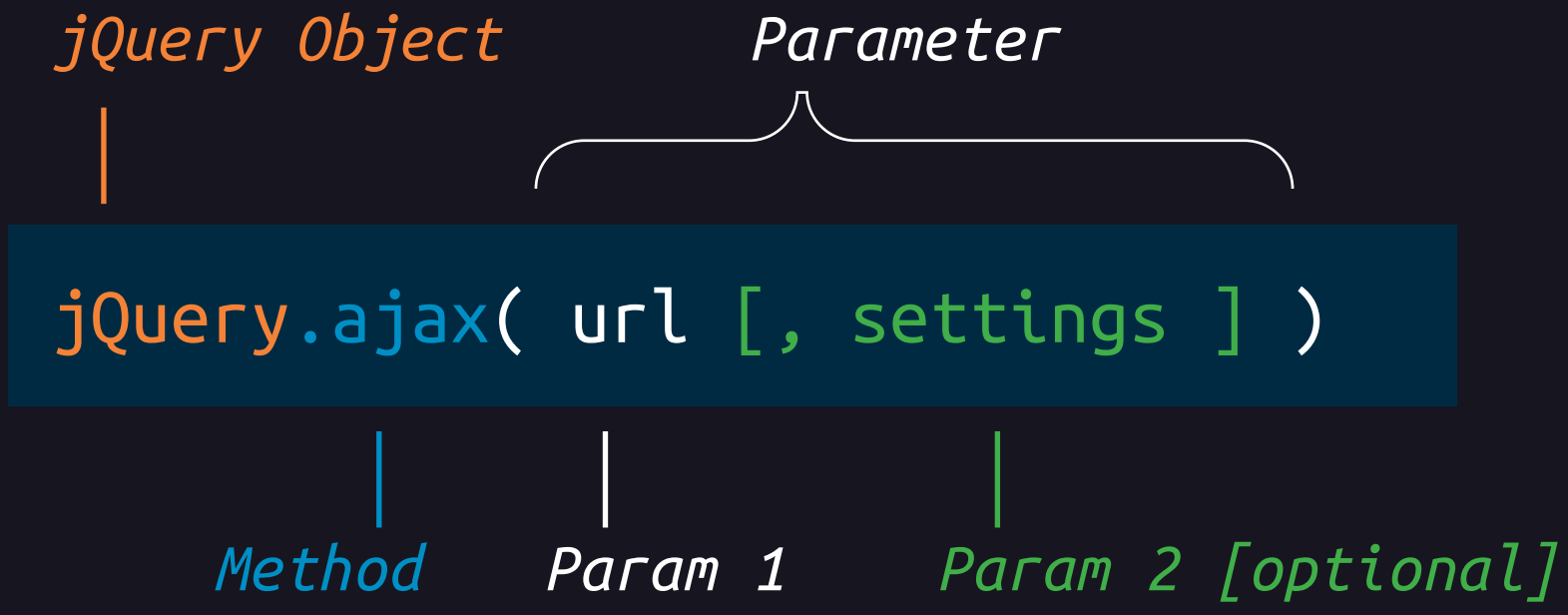
```
xhr.open("GET", "http://www.test.de/somefile.xml", true);
xhr.send(null);
```

```
function submitForm() {
    var xhr;
    try {
        xhr = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (e) {
        try {
            xhr = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e2) {
            try {
                xhr = new XMLHttpRequest();
            } catch (e3) {
                xhr = false;
            }
        }
    }
    xhr.onreadystatechange  = function() {
        if(xhr.readyState  == 4) {
            if(xhr.status  == 200) {
                document.ajax.dyn="Received:"  + xhr.responseText;
            } else {
                document.ajax.dyn="Error code " + xhr.status;
            }
        }
    };
    xhr.open(GET, "data.txt",  true);
    xhr.send(null);
}
```

jQuery Help!

```
jQuery.ajax( url [, settings ] )
```

jQuery Syntax



jQuery Methods

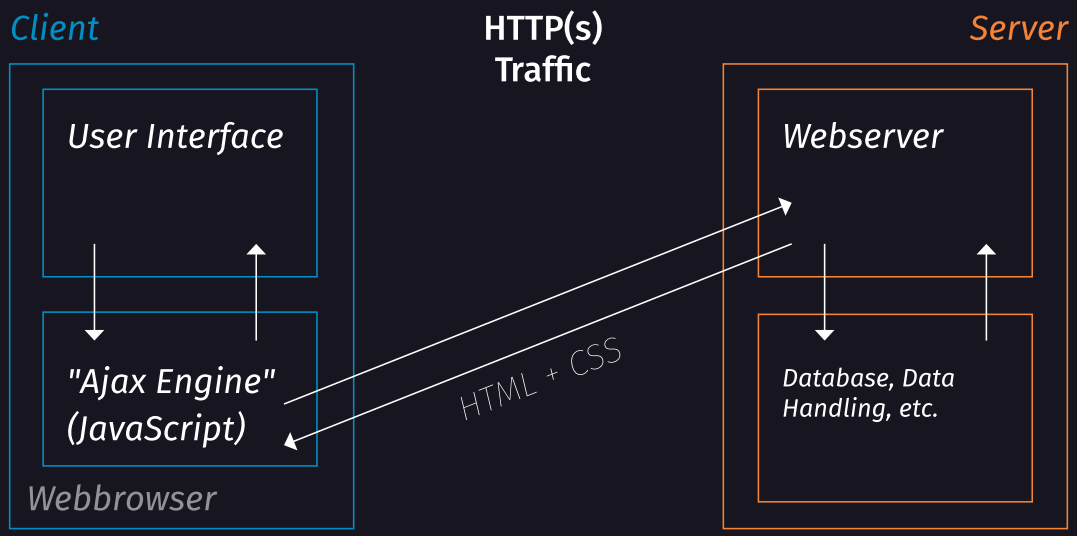
```
jQuery.ajax( url [, settings ] )
```

```
jQuery.load( url [, data ] [, complete ] )
```

```
jQuery.post( url [, data ] [, success ] [, dataType ] )
```

```
jQuery.getJSON( url [, data ] [, success ] )
```

Zusammenfassung



Use a Framework!

```
jQuery.ajax( url [, settings ] )
```

```
function submitForm() {
    var xhr;
    try {
        xhr = new ActiveXObject("Msxm
    } catch (e) {
        try {
            xhr = new ActiveXObject("
        } catch (e2) {
            try {
                xhr = new XMLHttpRequ
            } catch (e3) {
                xhr = false;
            }
        }
    }
    xhr.onreadystatechange  = functio
        if(xhr.readyState  == 4) {
            if(xhr.status  == 200) {
                document.ajax.dyn="Re
            } else {
                document.ajax.dyn="Er
            }
```