Netzbasierte Informationssysteme

Angereicherte Webdokumente (SVG, CDF, Web 2.0, AJAX)

Prof. Dr.-Ing. Robert Tolksdorf
Freie Universität Berlin
Institut für Informatik
Netzbasierte Informationssysteme
mailto: tolk@inf.fu-berlin.de
http://www.robert-tolksdorf.de

SVG

Scalable Vector Graphics (SVG)

Freie Universität Berlin

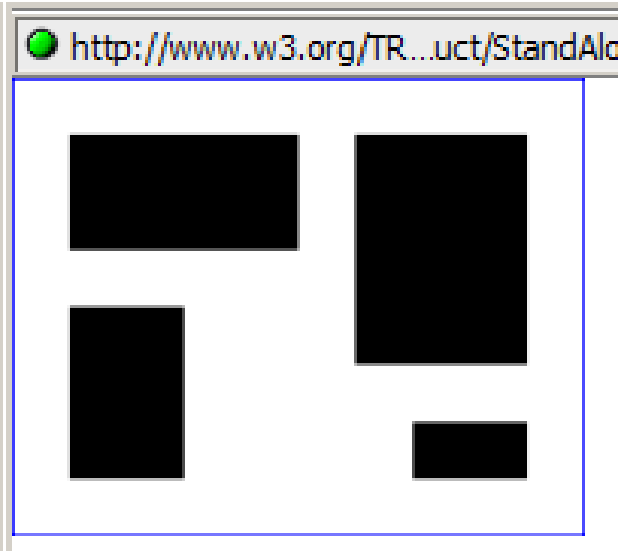
- Scalable Vector Graphics (SVG) 1.1
  - Zweck: XML Sprache zur Beschreibung von 2D Vektorgrafiken
  - Status: W3C Recommendation 14 January 2003
  - Quelle: http://www.w3.org/TR/SVG/

SVG

- Drei Klassen von Objekten
  - Zeichnungsobjekte
  - Bildobjekte
  - Text
- Zeichnungen können dynamisch und interaktiv sein
- SVG ist modularisiert und kann in Profilen definiert werden
- SVG im Web
  - Eigenständige Dokumente (image/svg+xml)
  - Eingebettete Referenz (<object> bei HTML)
  - Eingebettetes SVG (in XML Sprachen)
  - ...

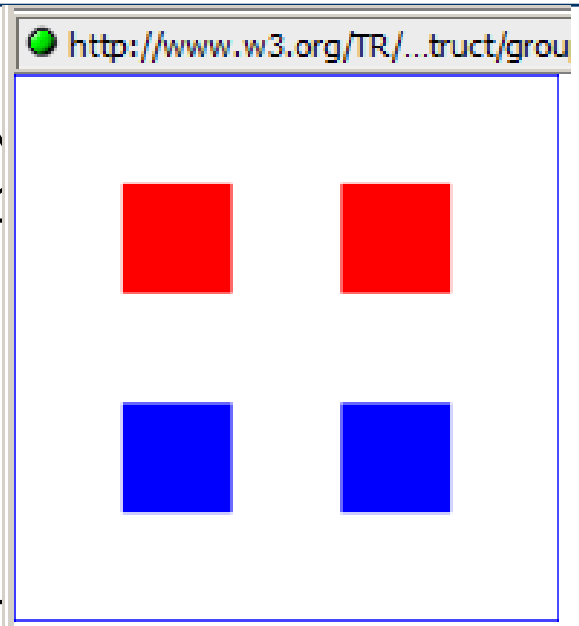
Beispiel [aus Standard]

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.
 "http://www.w3.org/Graphics/SVG/1.1/DTD/
<svg width="5cm" height="4cm" version="1.1
    xmlns="http://www.w3.org/2000/svg">
  <desc>Four separate rectangles
  </desc>
    <rect x="0.5cm" y="0.5cm" width="2cm" height="1cm"/>
    <rect x="0.5cm" y="2cm" width="1cm" height="1.5cm"/>
    <rect x="3cm" y="0.5cm" width="1.5cm" height="2cm"/>
    <rect x="3.5cm" y="3cm" width="1cm" height="0.5cm"/>
  <!-- Show outline of canvas using 'rect' element -->
  <rect x=".01cm" y=".01cm" width="4.98cm" height="3.98cm"
        fill="none" stroke="blue" stroke-width=".02cm" />
</svg>
```



Elementgruppen

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN
 "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11
<svg width="5cm" height="5cm" version="1.1"
    xmlns="http://www.w3.org/2000/svg">
  <desc>Two groups, each of two rectangles
  </desc>
  <g id="group1" fill="red" >
    <rect x="1cm" y="1cm" width="1cm" height="1
    <rect x="3cm" y="1cm" width="1cm" height="1cm" />
  </g>
  <g id="group2" fill="blue" >
    <rect x="1cm" y="3cm" width="1cm" height="1cm" />
    <rect x="3cm" y="3cm" width="1cm" height="1cm" />
  </g>
  <!-- Show outline of canvas using 'rect' element -->
  <rect x=".01cm" y=".01cm" width="4.98cm" height="4.98cm"
      fill="none" stroke="blue" stroke-width=".02cm" />
</svg>
```

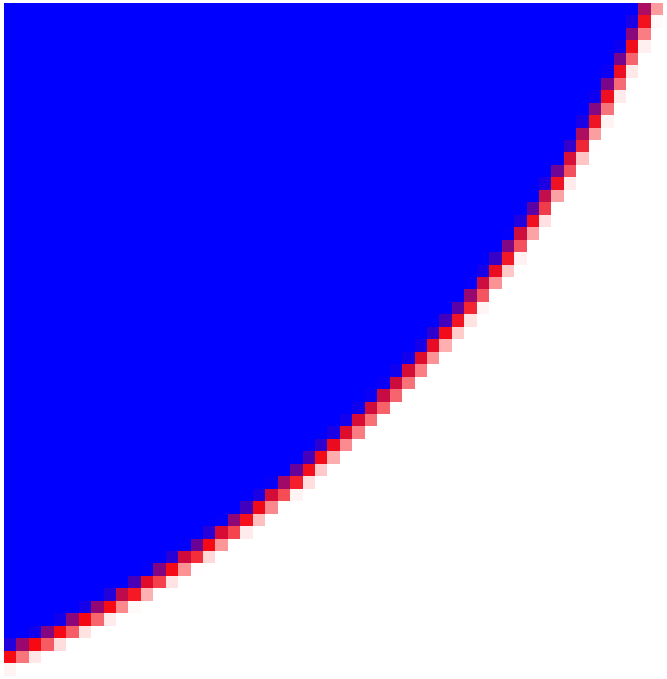


- Elemente
  - <rect>
  - <circle>
  - <ellipse>
  - <line>
  - <polyline>
  - <polygon>

```
<ellipse
      transform="translate(900 200) rotate(-30)"
      rx="250" ry="100"
      fill="none" stroke="blue"
      stroke-width="20"  />
```

SVG und CSS

```
<svg width="3in" height="3in">
 <desc>A blue circle with a red outline</desc>
  <g>
    <circle cx="110" cy="120" r="100"
     style="fill: blue; stroke: red"/>
  </g>
</svg>
```



SVG und CSS

- Überführung visueller Eigenschaften in Stildefinitionen:
  - <ellipse cx="110" cy="60" rx="100" ry="50"
    fill="blue" stroke="black" stroke-width="4" />
  - <ellipse cx="110" cy="60" rx="100" ry="50"
    style="fill:blue;stroke:black;stroke-width:4" />
    [http://www.selfsvg.info/?section=4.8]
- Mit externen Stylesheets:

[http://www.carto.net/papers/svg/samples/styles.shtml]

SVG und CSS

Freie Universität Berlin

- SVG übernimmt CSS2 Eigenschaften:
  - Schriften:
    'font', 'font-family', 'font-size', 'font-size-adjust', 'font-stretch', 'font-style', 'font-variant', 'font-weight'
  - Text:
    'direction', 'letter-spacing', 'text-decoration', 'unicode-bidi', 'word-spacing'
  - Etc:
    'clip', 'color, 'cursor', 'display', 'overflow', 'visibility'

- http://www.w3.org/TR/SVG/styling.html

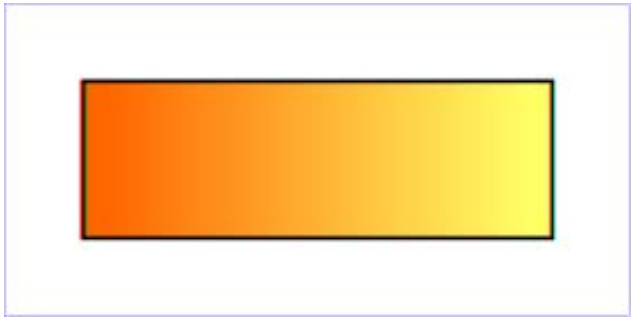
SVG und CSS

Freie Universität Berlin

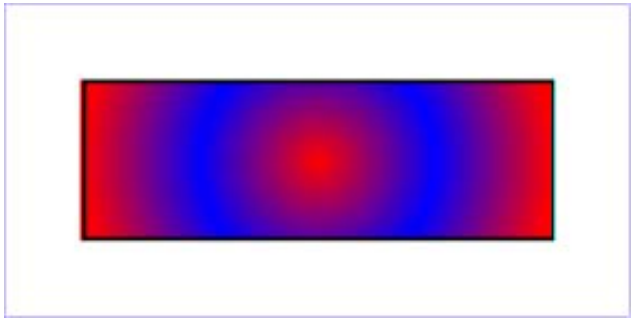
- SVG definiert folgende neue Stileigenschaften
  - Bildkomposition:
    'clip-path', 'clip-rule', 'mask', 'opacity'
  - Filter:
    'enable-background', 'filter', 'flood-color', 'flood-opacity', 'lighting-color'
  - Verlauf:
    'stop-color', 'stop-opacity'
  - Interaktivität:
    'pointer-events'
  - Farbe:
    'color-interpolation', 'color-interpolation-filters', 'color-profile', 'color-rendering', 'fill', 'fill-opacity', 'fill-rule', 'image-rendering', 'marker', 'marker-end', 'marker-mid', 'marker-start', 'shape-rendering', 'stroke', 'stroke-dasharray', 'stroke-dashoffset', 'stroke-linecap', 'stroke-linejoin', 'stroke-miterlimit', 'stroke-opacity', 'stroke-width', 'text-rendering'
  - Text:
    'alignment-baseline', 'baseline-shift', 'dominant-baseline', 'glyph-orientation-horizontal', 'glyph-orientation-vertical', 'kerning', 'text-anchor', 'writing-mode'

Farbübergänge / Gradienten

```
<g>
  <defs>
    <linearGradient id="MyGradient">
      <stop offset="5%" stop-color="#F60" />
      <stop offset="95%" stop-color="#FF6" />
    </linearGradient>
  </defs>
  <!-- The rectangle is filled using a linear gradient paint server -->
  <rect fill="url(#MyGradient)" stroke="black" stroke-width="5"
        x="100" y="100" width="600" height="200"/>
</g>
```

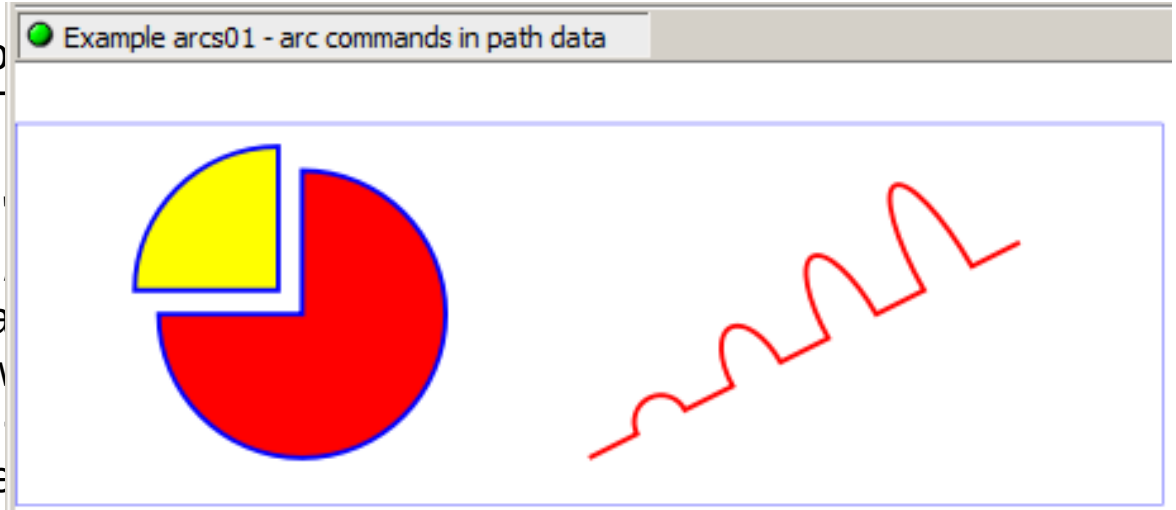


```
<radialGradient id="MyGradient" gradientUnits="userSpaceOnUse"
                cx="400" cy="200" r="300" fx="400" fy="200">
    <stop offset="0%" stop-color="red" />
    <stop offset="50%" stop-color="blue" />
    <stop offset="100%" stop-color="red" />
  </radialGradient>
```



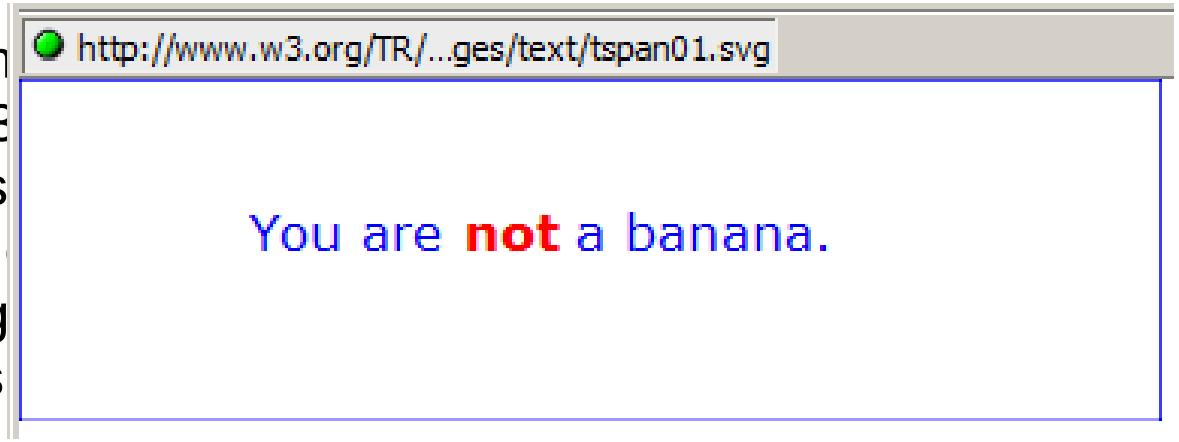
Pfade

```
<?xml version="1.0" standalone="no
<!DOCTYPE svg PUBLIC "-//W3C//DT
 "http://www.w3.org/Graphics/SVG/
<svg width="12cm" height="5.25cm"
    xmlns="http://www.w3.org/2000/
 <title>Example arcs01 - arc comma
 <desc>Picture of a pie chart with tw
      a picture of a line with arc blips
 <rect x="1" y="1" width="1198" he
      fill="none" stroke="blue" stroke-width="1" />
 <path d="M300,200 h-150 a150,150 0 1,0 150,-150 z"
      fill="red" stroke="blue" stroke-width="5" />
 <path d="M275,175 v-150 a150,150 0 0,0 -150,150 z"
      fill="yellow" stroke="blue" stroke-width="5" />
 <path d="M600,350 l 50,-25
        a25,25 -30 0,1 50,-25 l 50,-25
        a25,50 -30 0,1 50,-25 l 50,-25
        a25,75 -30 0,1 50,-25 l 50,-25
        a25,100 -30 0,1 50,-25 l 50,-25"
      fill="none" stroke="red" stroke-width="5"  />
</svg>
```



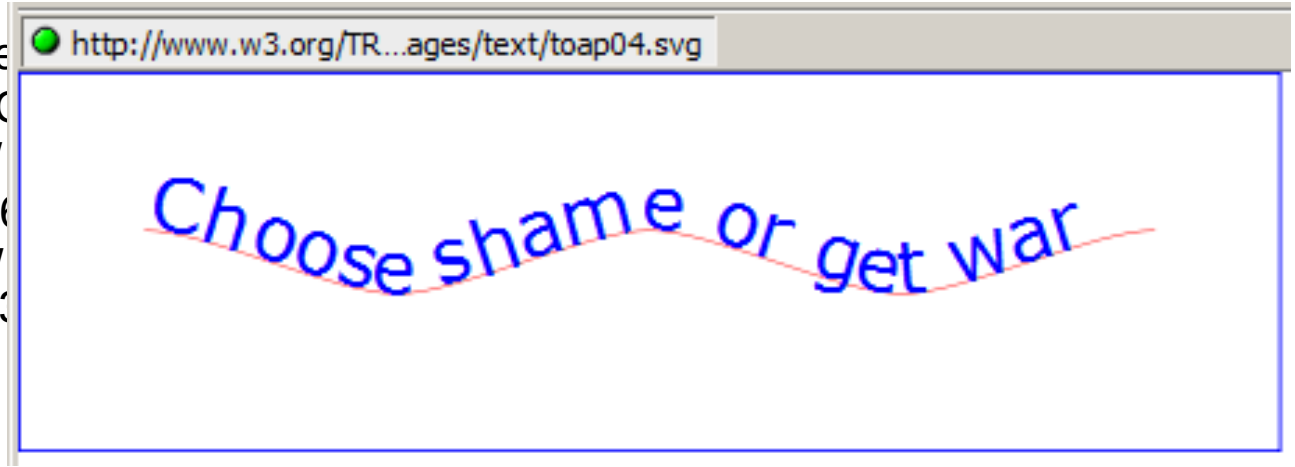
Text

```
<?xml version="1.0" standalon
<!DOCTYPE svg PUBLIC "-//W3
 "http://www.w3.org/Graphics
<svg width="10cm" height="3
    xmlns="http://www.w3.org
  <desc>Example tspan01 - us
   attributes</desc>
  <g font-family="Verdana" font-size="45" >
   <text x="200" y="150" fill="blue" >
     You are
       <tspan font-weight="bold" fill="red" >not</tspan>
     a banana.
   </text>
  </g>
  <!-- Show outline of canvas using 'rect' element -->
  <rect x="1" y="1" width="998" height="298"
       fill="none" stroke="blue" stroke-width="2" />
</svg>
```



Text

```
<?xml version="1.0" standalone
<!DOCTYPE svg PUBLIC "-//W3C
 "http://www.w3.org/Graphics/
<svg width="12cm" height="3.6
    xmlns="http://www.w3.org/
    xmlns:xlink="http://www.w3
  <defs>
    <path id="MyPath"
          d="M 100 125
             C 150 125 250 175 300 175
             C 350 175 450 125 500 125
             C 550 125 650 175 700 175
             C 750 175 850 125 900 125" />
  </defs>
  <desc>Example toap04 - text on a path layout rules</desc>
  <use xlink:href="#MyPath" fill="none" stroke="red"  />
  <text font-family="Verdana" font-size="60" fill="blue" letter-spacing="2" >
    <textPath xlink:href="#MyPath">
      Choose shame or get war
    </textPath>
  </text>
  [...]
</svg>
```

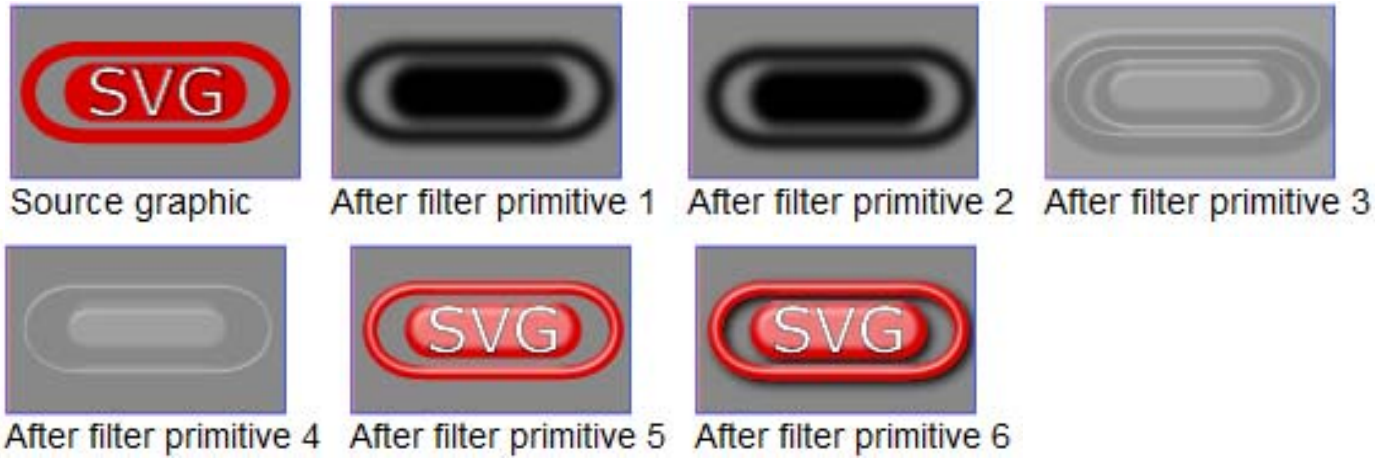


Filter – Transformation von Grafiken

```
[...]<defs>
    <filter id="MyFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="200"
     height="120">
      [...]
    </filter>
  </defs>
  <rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
  <g filter="url(#MyFilter)" >
      <g>
      <path fill="none" stroke="#D90000" stroke-width="10"
            d="M50,90 C0,90 0,30 50,30 L150,30 C200,30 200,90 150,90 z" />
      <path fill="#D90000"
            d="M60,80 C30,80 30,40 60,40 L140,40 C170,40 170,80 140,80 z" />
      <g fill="#FFFFFF" stroke="black" font-size="45" font-family="Verdana" >
        <text x="52" y="76">SVG</text>
      </g>
    </g>
  </g>
</svg>
```

Filter

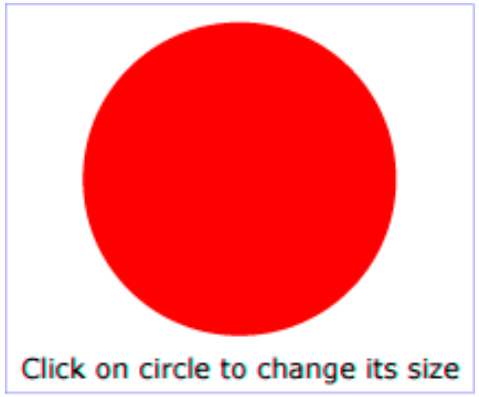
```
<filter id="MyFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="200" height="120">
  <desc>Produces a 3D lighting effect.</desc>
1. <feGaussianBlur in="SourceAlpha" stdDeviation="4" result="blur"/>
2. <feOffset in="blur" dx="4" dy="4" result="offsetBlur"/>
3. <feSpecularLighting in="blur" surfaceScale="5" specularConstant=".75"
                       specularExponent="20" lighting-color="#bbbbbb" result="specOut">
     <fePointLight x="-5000" y="-10000" z="20000"/>
   </feSpecularLighting>
4. <feComposite in="specOut" in2="SourceAlpha" operator="in" result="specOut"/>
5. <feComposite in="SourceGraphic" in2="specOut" operator="arithmetic"
          k1="0" k2="1" k3="1" k4="0" result="litPaint"/>
6. <feMerge>
   <feMergeNode in="offsetBlur"/>
   <feMergeNode in="litPaint"/>
  </feMerge>
</filter>
```



Scripting
SVG Objekte sind per DOM erreichbar

Freie Universität Berlin

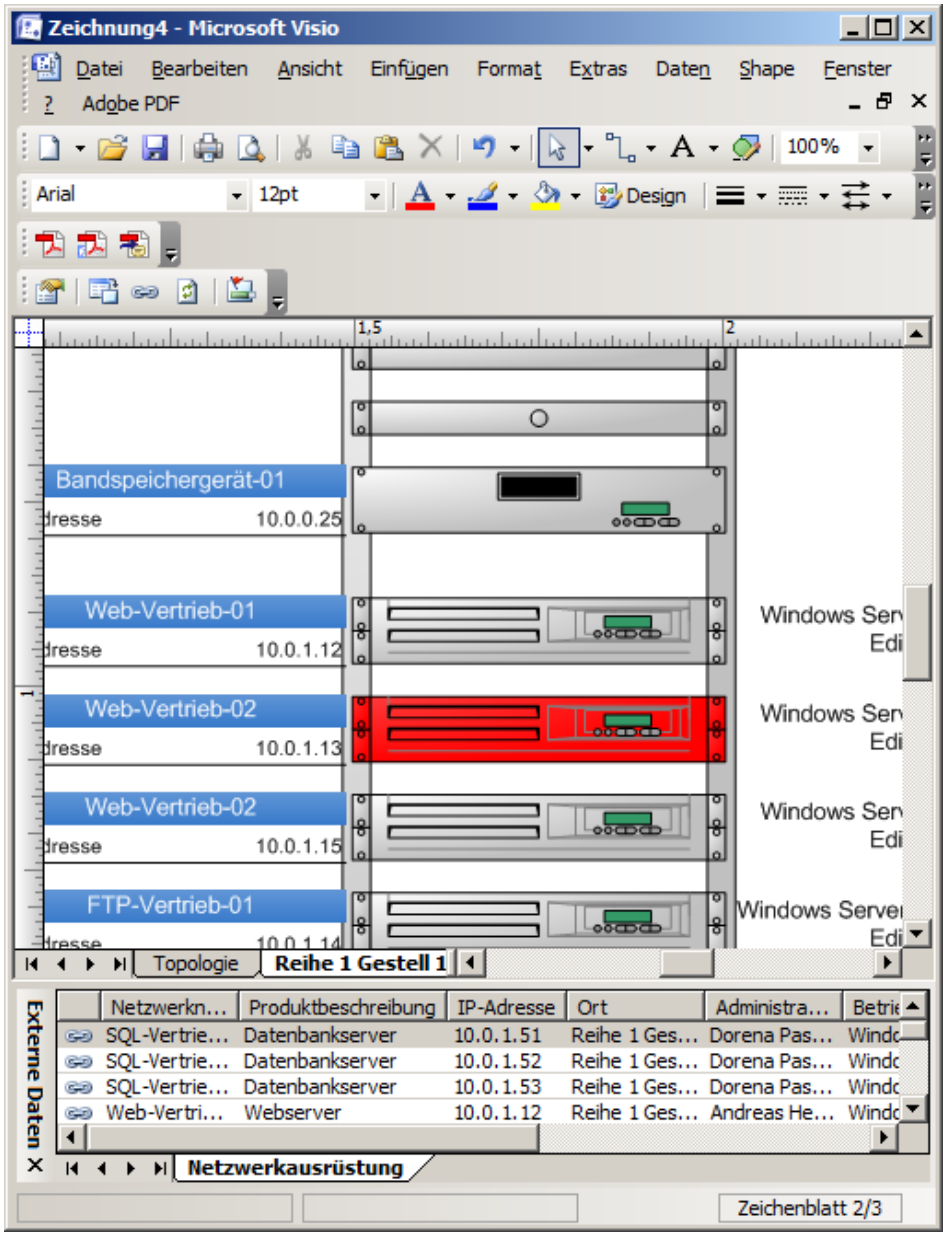
```
[...]
  <!-- ECMAScript to change the radius with each click -->
  <script type="text/ecmascript"> <![CDATA[
    function circle_click(evt) {
      var circle = evt.target;
      var currentRadius = circle.getAttribute("r");
      if (currentRadius == 100)
        circle.setAttribute("r", currentRadius*2);
      else
        circle.setAttribute("r", currentRadius*0.5);
    }
  ]]> </script>
  <!-- Outline the drawing area with a blue line -->
  <rect x="1" y="1" width="598" height="498" fill="none" stroke="blue"/>
  <!-- Act on each click event -->
  <circle onclick="circle_click(evt)" cx="300" cy="225" r="100" fill="red"/>
  <text x="300" y="480"
        font-family="Verdana" font-size="35" text-anchor="middle">
    Click on circle to change its size
  </text>
</svg>
```

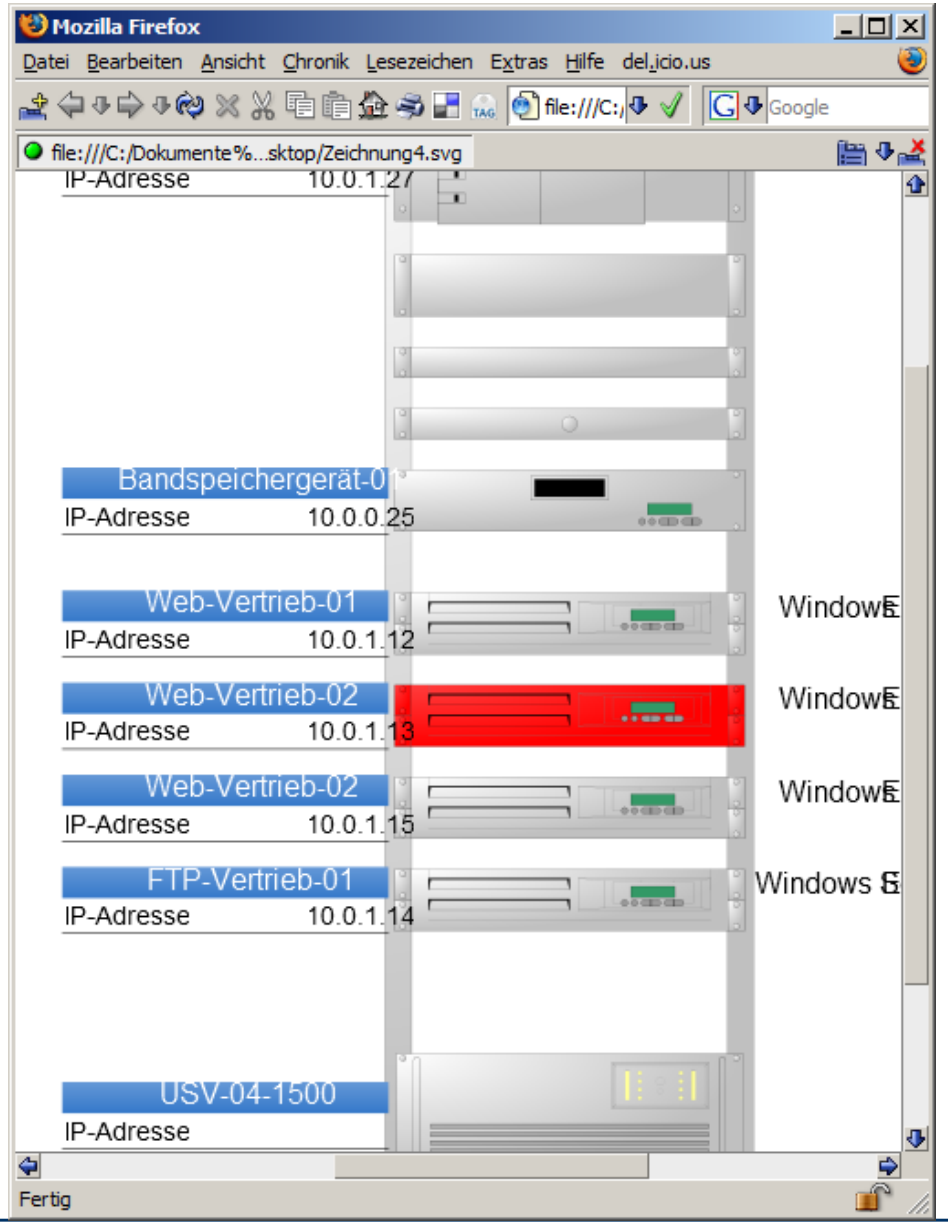


Was noch

- Linienarten
- Farben
- Masken, Clipping
- Schriften
- Verlinkung
- Animationen

Erzeugung von SVG





Rich Web Clients

W3C Rich Web Clients Activity

Freie Universität Berlin

- Das Web ist Plattform für Anwendungen jenseits reiner Informationssysteme
  - Online-Shops
  - Anwendungen wie Texterstellung
  - Lernanwendungen
  - Campus Management
  - ...

Rich Web Clients

- Verlagerung von Anwendungsteilen zusammen mit GUI zum Browser/Clienten
  - Applets
  - Javascript
  - …
- Teile der Anwendung „laufen“ im Browser
- Sie kombinieren verschiedene Medien zu einer Oberfläche

W3C Rich Web Clients Activity

- Drei Arbeitsgruppen
  - Compound Document Formats Working Group
    - „Compound Document“: Zusammengesetztes Dokumenten aus mehreren Teilen in unterschiedlichen Sprachen
    - „develop specifications which combine ***selected*** existing document formats from the W3C and elsewhere, and which specify the runtime behaviour of such combined documents“
  - Web APIs Working Group
    - „develop specifications that enable improved client-side application development on the Web“
  - Web Application Formats Working Group
    - develop specifications that enable improved client-side application development on the Web. This includes the development of languages for applications, especially user interfaces
    - XUL, XAML

Zusammengesetzte Dokumente

Freie Universität Berlin

- Compound Document by Reference Framework 1.0
  - W3C Working Draft 22 November 2006
  - http://www.w3.org/TR/CDR/
- „Compound Document“: Dokument das aus mehreren Teildokumenten in unterschiedlichen Auszeichungssprachen zusammengesetzt ist
  - XHTML + SVG + MathML
  - XHTML + SMIL
  - XHTML + XForms
  - XHTML + VoiceML
- Problem: Jeweilige Spezifikationen decken nicht alle Aspekte solcher Kompositionen ab
- Compound Document by Reference Framework, CDRF ist generischer Rahmen für die Verarbeitung solcher Dokumente

XML Namensräume

- Eine XML-Sprache wird durch eine URN bezeichnet
- Darin sind alle Sprachelemente eindeutig und definiert
- Mit XML-Namensräumen können in einem Dokument Elemente aus unterschiedlichen XML-Sprachen kombiniert werden wobei die Elementnamen eindeutig interpretierbar bleiben
- Möglichkeit 1:
  <element xmlns="URN"> legt fest, daß <element> ... </element> und alle Tags darin aus der durch URN bezeichneten XML-Sprache stammen
  - <math xmlns="http://www.w3.org/1998/Math/MathML">
    ...
    <math>

XML Namensräume

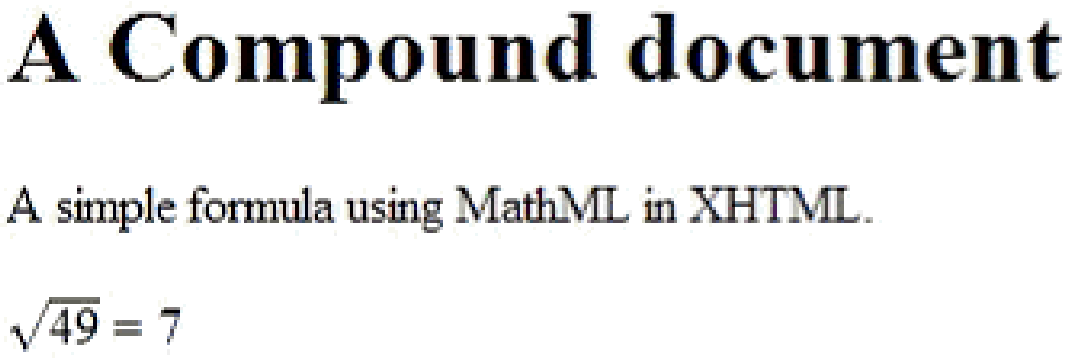
- Möglichkeit 2:
  <element xmlns:name="URN"> definiert ein Präfix name, das allen Tags aus der durch URN bezeichneten XML-Sprache vorangestellt werden
  - <xhtml:html
    xmlns="http://www.w3.org/1999/xhtml"
    xmlns:mathml="http://www.w3.org/1998/Math/MathML">

     <mathml:math>...</math>

Compound Document by Inclusion (CDI)

Freie Universität Berlin

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xhtml:html xmlns:xhtml="http://www.w3.org/1999/xhtml">
  <xhtml:body>
    <xhtml:h1>A Compound Document</xhtml:h1>
    <xhtml:p>A simple formula using MathML in XHTML.</xhtml:p>
    <mathml:math xmlns:mathml="http://www.w3.org/1998/Math/MathML">
      <mathml:mrow>
        <mathml:msqrt>
          <mathml:mn>49</mathml:mn>
        </mathml:msqrt>
        <mathml:mo>=</mathml:mo>
        <mathml:mn>7</mathml:mn>
      </mathml:mrow>
    </mathml:math>
  </xhtml:body>
</xhtml:html>
```



Compound Document by Reference (CDR)

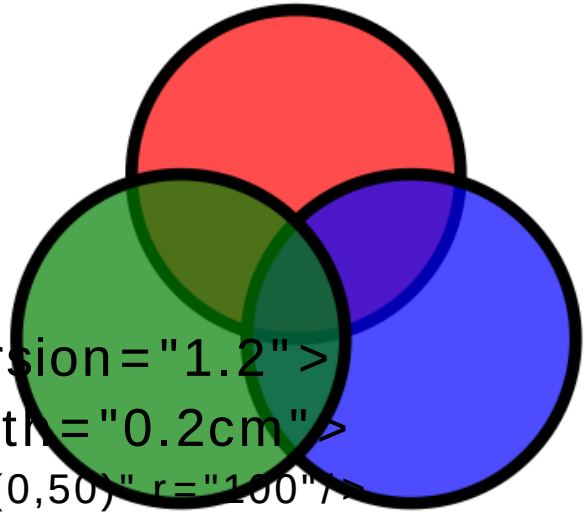
Freie Universität Berlin

- main.xhtml:

```
<?xml version="1.0"?>
<html xmlns="http://www.w3.org/1999/xhtml">
 <head><title>circles</title></head><body>
  <object height="350" width="600" type="image/svg+xml"
  data="circles.svg"/>
 </body></html>
```

- circle.svg:

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg"
   viewBox="0 0 100 100" baseProfile="tiny" version="1.2">
 <g fill-opacity="0.7" stroke="black" stroke-width="0.2cm">
  <circle fill="red" cx="6cm" cy="2cm" transform="translate(0,50)" r="100"/>
  <circle fill="blue" cx="6cm" cy="2cm" transform="translate(70,150)" r="100"/>
  <circle fill="green" cx="6cm" cy="2cm" transform="translate(-70,150)" r="100"/>
 </g>
</svg>
```



Wie werden Objekte referenziert?

Freie Universität Berlin

- Existierende Mittel nutzen
- XHTML:

```
<object type="..." data="...">
  <param name="param1" value="true" />
  <param name="param2" value="123" />
</object>
```

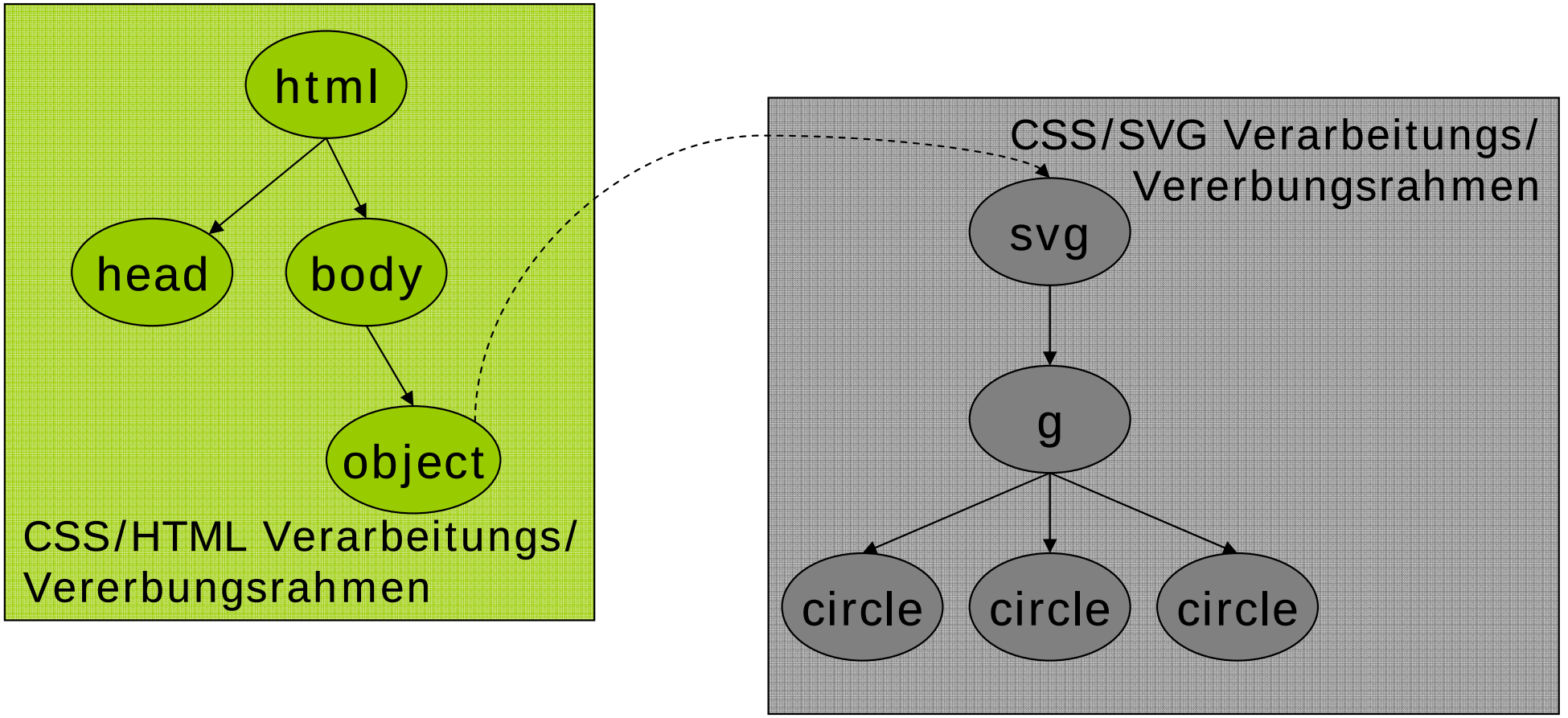
- SVG:

```
<svg xmlns="http://www.w3.org/2000/svg">
   <foreignObject>
      <metadata>
        param=value;  param=value
      </metadata>
   </foreignObject>
</svg>
```

- SMIL:

```
<ref src="http://www.example.com/herbert.face">
  <param name="mood" value="surly" valuetype="data"/>
  <param name="accessories" value="baseball-cap,nose-ring"
   valuetype="data"/>
</ref>
```

Wie sieht das Dokumentenmodell aus?



- Angleichung xhtml mobile, SVG mobile etc.?

DOM Zugang vom Kind aus?

Freie Universität Berlin

- Window Object 1.0
  - W3C Working Draft 07 April 2006
  - http://www.w3.org/TR/Window
  - Definiert window Objekt das den Rahmen eines eingebetteten Dokuments DOM-seitig definiert

```
interface Window {
    // name attribute of referencing frame/iframe/object, or name
   passed to
    // window.open
    attribute dom::DOMString name;
    // global object of containing document
    readonly attribute Window parent;
    // global object of outermost containing document
    readonly attribute Window top;
    // referencing <html:frame>, <html:iframe>, <html:object>,
   <svg:foreignObject>,
    // <svg:animation> or other embedding point, or null if none
    readonly attribute dom::Element frameElement;
};
```

DOM Zugang vom Kind aus?

- Über frameElement Zugang zum DOM des “Elterndokumente”

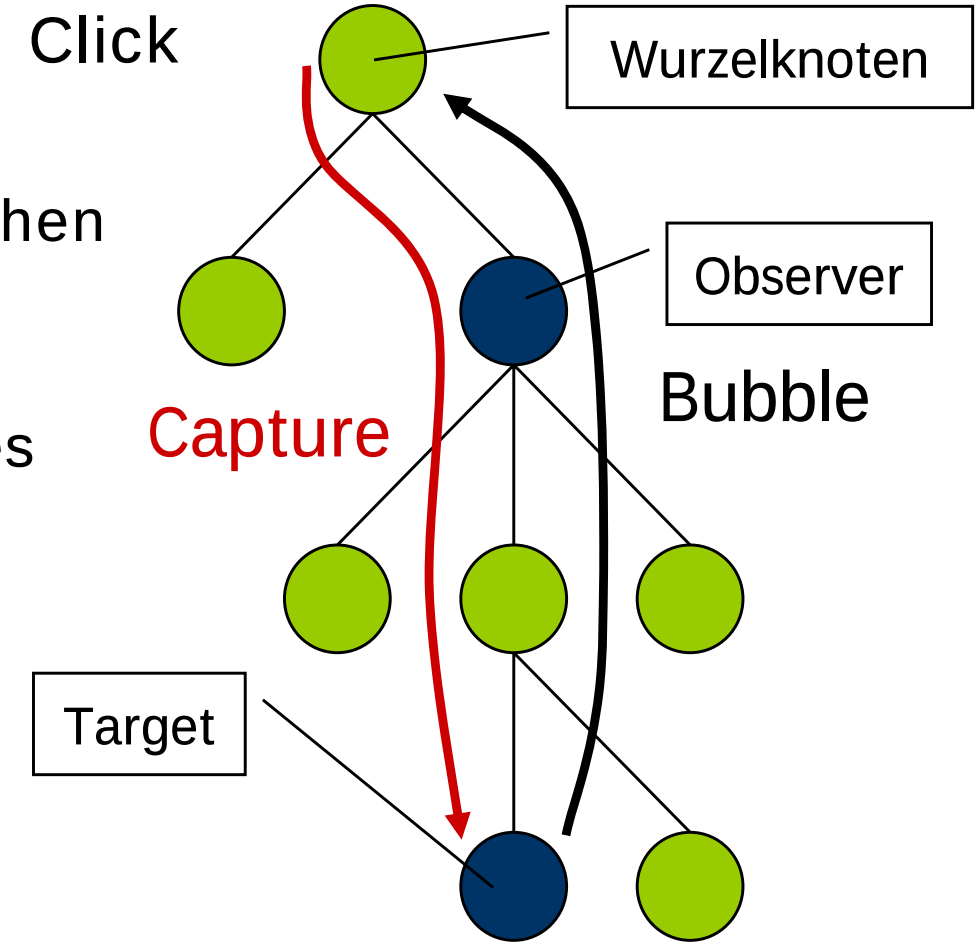
```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg“
   height="20" version="1.1" width="20">
   <title>child-svg</title>
     <rect fill="blue" height="20"
          onload="alert('Child has seen: ' +
   document.defaultView.frameElement.ownerDocument.title)“
          width="20" x="10" y="10" />
</svg>
```

DOM Zugang zum Kind?

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>parent-xhtml</title>
  </head>
  <body>
    <object data="child.svg"
      onload="alert('Parent has seen: ' +
        this.contentDocument.getElementsByTagNameNS(
          'http://www.w3.org/2000/svg',
          'title') [0].textContent)"  />
  </body>
</html>
```

Was ist mit Events?

- Event: Asynchron auftretendes Ereignis, das auf ein Element (target) abzielt, z.B. Click
- Verarbeitung:
  - Capture-Phase: Herunterreichen des Ereignisses
  - Bubble-Phase: Heraufreichen des Ereignisses
- Ereignisse können jeweils
  - verarbeitet werden
  - gestoppt werden
- Verarbeitung ist *Action*
- *Handler* spezifiziert Action
- *Listener* bindet Ereignis an einem Ort mit Handler



Was ist mit Events?

- Vom Kind- zum Elterndokument über frameElement weiterreichbar

```
var x = document.createEvent("CustomEvent");
x.initCustomEventNS("http://example.org/test", "test", true,
    false, null);
window.frameElement.dispatchEvent(x);
```

- Abbildung von sprachspezifischen Events auf DOM3 Events
- Gleiches Verhalten muss gesichert sein

Web 2.0 / AJAX

Web 2.0

- Tim O‘Reilly „What is Web 2.0“
  [http://www.oreilly.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html]
- Das Web ist die Anwendungsplattform
  - Netscape als Software Hersteller
  - Web Browser und Server sind Teil der Web Plattform
  - Google als Software Hersteller?
  - Wertschöpfung durch Dienste auf der Web Plattform
- Netzwerkeffekte sind die Quelle wirtschaftlicher Dominanz im Web 2.0
  - Ebay Angebote stammen von den Nutzern selber
  - „Architecture of participation“

Web 2.0 Muster

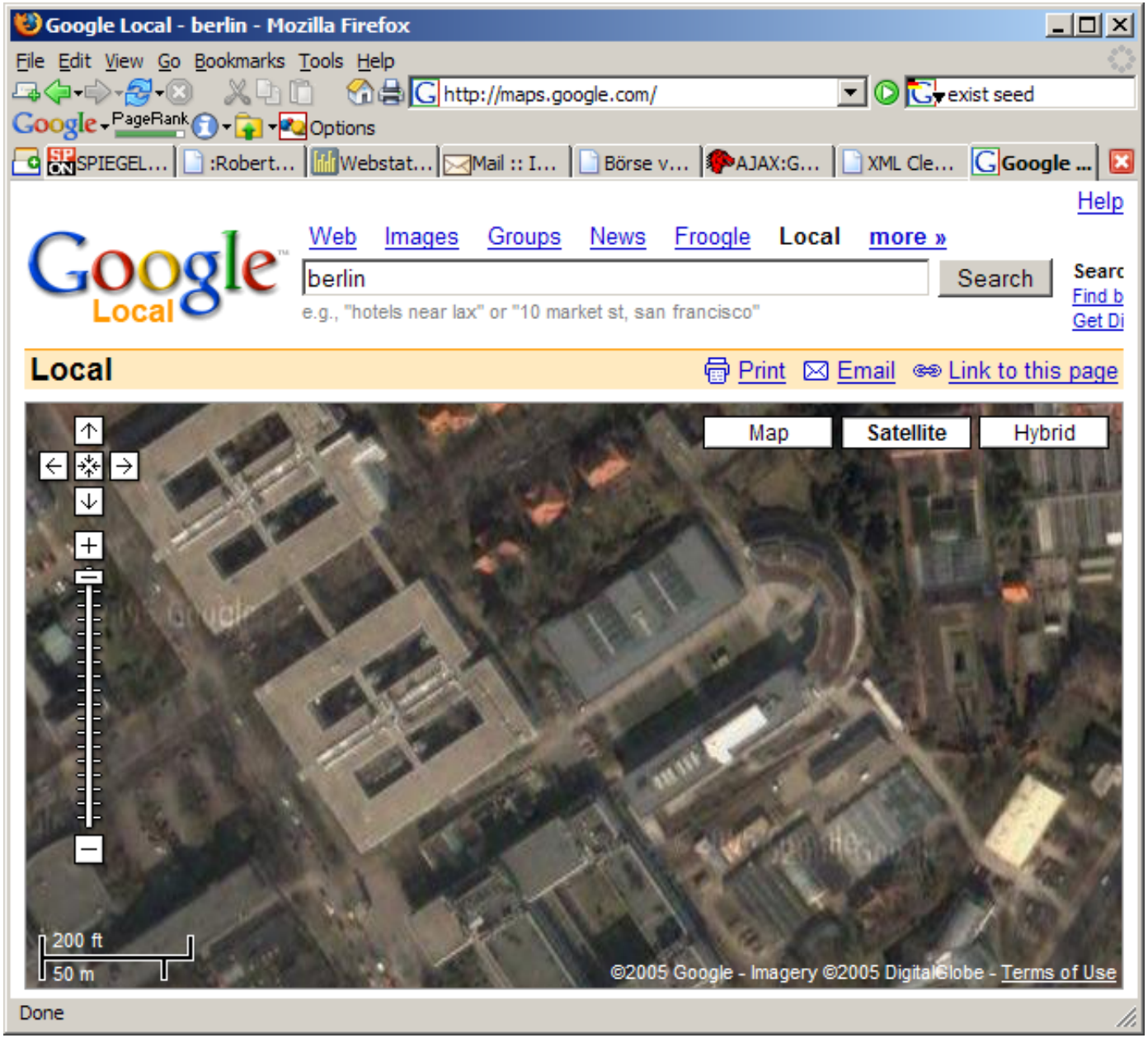
- „The Long Tail“: Nischenangebote summieren sich zum größten Anteil des Web
- „Data is the Next Intel Inside“: Anwendungen werden von den verwendeten Daten getrieben
- „Users Add Value“: Erweiterbarkeit durch Nutzerbeteiligung
- „Network Effects by Default“: Automatische Integration von Nutzerdaten sichert automatische Netzwerkeffekte
  Only a small percentage of users will go to the trouble of adding value to your application. Therefore: Set inclusive defaults for aggregating user data as a side-effect of their use of the application.
- „Some Rights Reserved“: Intergrationsmöglichkeit nicht durch Rechte beschränken
- „The Perpetual Beta“: Dienste sind keine Artefakte sondern im fortlaufend überarbeiteten Betrieb
- „Cooperate, Don't Control“: Integration von Diensten soll möglich sein
- „Software Above the Level of a Single Device“: Anwendungen geräteübergreifend entwerfen

Web 2.0

- Tim Berners-Lee: Das ist genau die Idee des Web!
  [Interview, http://www-128.ibm.com/developerworks/podcast/dwi/cm-int082206.txt]
- LANINGHAM: You know, with Web 2.0, a common explanation out there is Web 1.0 was about connecting computers and making information available; and Web 2 is about connecting people and facilitating new kinds of collaboration. Is that how you see Web 2.0?
  BERNERS-LEE: Totally not. Web 1.0 was all about connecting people. It was an interactive space, and I think Web 2.0 is of course a piece of jargon, nobody even knows what it means. If Web 2.0 for you is blogs and wikis, then that is people to people. But that was what the Web was supposed to be all along. And in fact, you know, this Web 2.0, quote, it means using the standards which have been produced by all these people working on Web 1.0. It means using the document object model, it means for HTML and SVG and so on, it's using HTTP, so it's building stuff using the Web standards, plus Java script of course. So Web 2.0 for some people it means moving some of the thinking client side so making it more immediate, but the idea of the Web as interaction between people is really what the Web is. That was what it was designed to be as a collaborative space where people can interact.

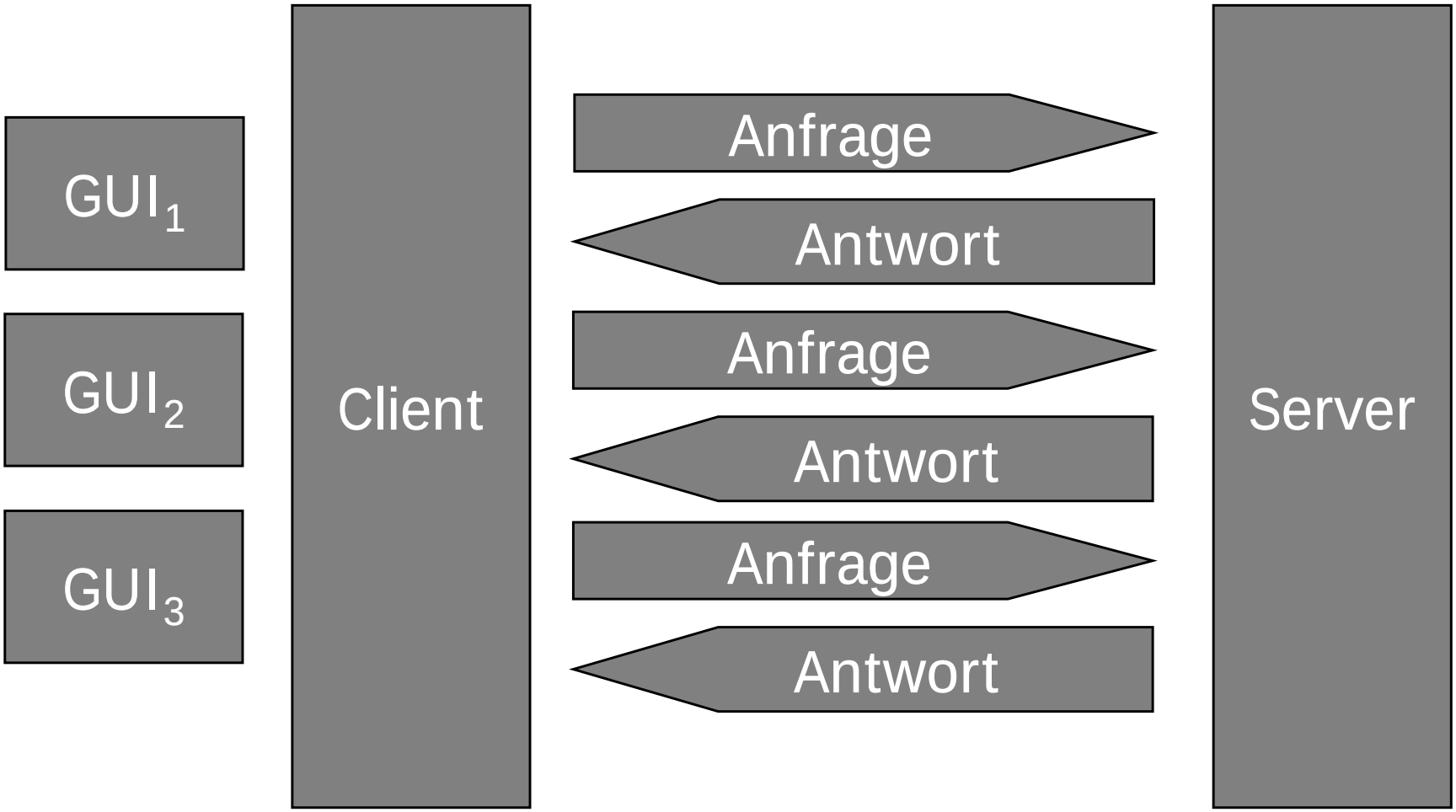
Google Maps

- Anwendung: Durchschauen von Karten und Bildern
- Arbeitet so als sei es eine Desktop-Anwendung mit nativen Mitteln der Oberfläche (vgl: Google Earth)



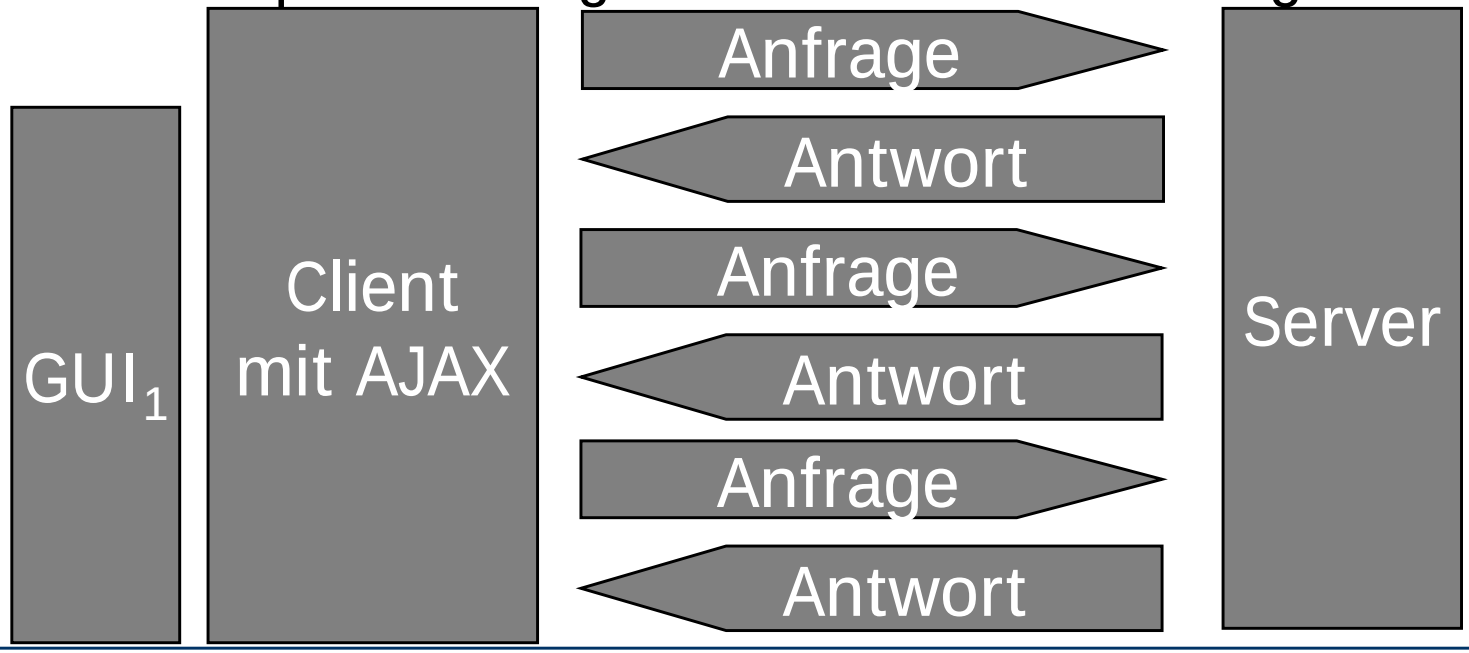
Freie Universität Berlin

- Problematik: Durch die Anfrage/Antwort Interaktion per HTTP wird bei jeder Nutzeraktion ein komplett neues GUI im Browser erzeugt



AJAX

- Asynchronous JavaScript and XML (AJAX) realisiert dies durch Kombination von
  - Präsentationssprachen XHTML und CSS
  - Interaktion und Modifikation im Browser mit DOM
  - Datenaustausch mit XML
  - Datentransfer durch asynchrone HTTP-Anfragen
  - Javascript als Integration dieser Technologien



XMLHttpRequest Object

Freie Universität Berlin

- The XMLHttpRequest Object
  - W3C Working Draft 27 September 2006
  - http://www.w3.org/TR/XMLHttpRequest/
  - “Special thanks to the Microsoft employees who first implemented the XMLHttpRequest interface, which was first widely deployed by the Windows Internet Explorer browser.”

- Definiert Schnittstelle und Verhalten eines Objektes das in Browsern für Scriptsprachen zugänglich sein soll
- Dieses Objekt kann HTTP Anfragen bearbeiten
- Die Anfragen können asynchron zur weiteren Scriptverarbeitung ausgeführt werden

„IDL“ des XMLHttpRequest Objekts

```
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;
};
```

Anfragezustände

- Anfrage durchläuft 5 Zustände
  - readonly attribute unsigned short readyState;
  - 0, Uninitialisiert
    XMLHttpRequest Objekt erzeugt
  - 1, Open
    Objekt initialisiert (open() Methode aufgerufen)
  - 2, Sent
    HTTP Anforderung gesendet (send() Methode aufgerufen)
  - 3, Receiving
    Antwort wird empfangen
  - 4, Loaded
    Antwort ist empfangen
- Bei jedem Zustandswechsel wird eine Methode aufgerufe, der “Handler”
  - attribute EventListener onreadystatechange;

Eine URL überprüfen

- Mit XMLHttpRequest Objekt HEAD Methode absetzen
- Nach Beendigung HTTP Antwortcode holen:

```
function fetchStatus(address) {
 var client = new XMLHttpRequest();
 client.onreadystatechange = function() {
  // in case of network errors this might not give reliable results
  if(this.readyState == 4)
   returnStatus(this.status);
 }
 client.open("HEAD", address);
 client.send();
}
```

- Hallo sagen:

```
function ping(message) {
 var client = new XMLHttpRequest();
 client.open("POST", "/ping");
 client.send(message);
}
```

Objekt initialisieren

- In Zustand 0: Initialisierung des Objekts
- open(method, url, async, user, password)
  - method: HTTP-Methode (inkl. WebDAV):
    GET, POST, HEAD, PUT, DELETE, PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, UNLOCK, VERSION-CONTROL, REPORT, CHECKOUT, CHECKIN, UNCHECKOUT, MKWORKSPACE, UPDATE, LABEL, MERGE, BASELINE-CONTROL, MKACTIVITY, ORDERPATCH, ACL
  - url: URL
  - async: Soll die Anfrage asynchron ablaufen (Default: ja)
  - user, password: Eventuell Nutzer/Password
- Zustand danach 1, open

Anfrage vorbereiten

- In Zustand 1: Setzen von Anfrage-Headern
- setRequestHeader(header, value)
  - HTTP Header setzen
  - Verschiedene Einschränkungen
    - Nichts tun bei Accept-Charset, Accept-Encoding, Content-Length, Expect, Date, Host, Keep-Alive, Referer, TE, Trailer, Transfer-Encoding, Upgrade
    - Vorhandenen Wert überschreiben bei uthorization, Content-Base, Content-Location, Content-MD5, Content-Range, Content-Type, Content-Version, Delta-Base, Depth, Destination, ETag, Expect, From, If-Modified-Since, If-Range, If-Unmodified-Since, Max-Forwards, MIME-Version, Overwrite, Proxy-Authorization, SOAPAction, Timeout
    - Sonst vorhandenen Wert kombinieren
      - client.setRequestHeader('X-Test', 'one');
client.setRequestHeader('X-Test', 'two');
      - X-Test: one, two

Anfrage stellen

- In Zustand 1: Absenden der Anfrage
- void send();
  void send(in DOMString data);
  void send(in Document data);
  - Absenden der HTTP Anfrage mit entsprechenden Header
  - Eventuell mit Inhalt als Anfragebestandteil
    - Content-Type setzen
    - application/xml sonst
  - Folgen von Umleitungen (Antworten 301, 302, 303, 307 und Location:/URI: Header)
- Zustand
  - Nach Absetzen der Anfrage: 2, Sent
  - Vor Empfang des Antwortinhalts: 3, Receiving
  - Nach Empfang des Antwortinhalts: 4, Loaded

Antwort-Header abfragen

- In Zustand 3 oder 4: Abfrage der Antwort-Header
- getAllResponseHeaders()

```
var client = new XMLHttpRequest();
client.open("GET", "test.txt", true);
client.send();
client.onreadystatechange = function() {
 if(this.readyState == 3) {
  print(this.getAllResponseHeaders());
 }
}
```

- Ausgabe:
  - Date: Sun, 24 Oct 2004 04:58:38 GMT
  - Server: Apache/1.3.31 (Unix)
  - Keep-Alive: timeout=15, max=99
  - Connection: Keep-Alive
  - Transfer-Encoding: chunked
  - Content-Type: text/plain; charset=utf-8
- getResponseHeader(header)

Antwort abfragen

- In Zustand 3 oder 4: Abfrage des Antwort-Inhalts (-fragments)
  - readonly attribute DOMString responseText;
- In Zustand 4: Abfrage des Antwort-Inhalts
  - readonly attribute Document responseXML;

- In Zustand 3 oder 4: Abfrage des HTTP Antwortcodes
  - readonly attribute unsigned short status;
  - readonly attribute DOMString statusText;

Antwort-Code berücksichtigen

- „main()“

```
var client = new XMLHttpRequest();
client.onreadystatechange = handler;
client.open("GET", "test.xml");
client.send();
```

- Management

```
function handler() {
 if(this.readyState == 4 && this.status == 200) {// so far so good
  if(this.responseXML != null &&
     this.responseXML.getElementById('test').firstChild.data)
   // success!
   test(this.responseXML.getElementById('test').firstChild.data);
  else
   test(null);
 } else if (this.readyState == 4 && this.status != 200) {
  test(null); // fetched the wrong page or network error...
 }
}
```

- Verarbeitung

```
function test(data) {// taking care of data
}
```