

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Geht nicht gibt's nicht!

HTML5-Features im Jetzt

Yann Massard

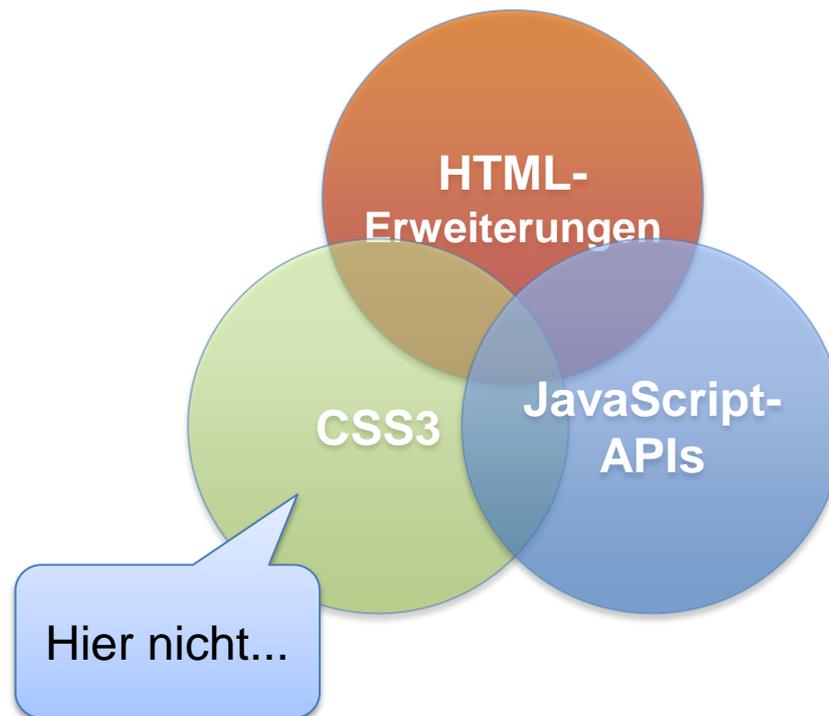
BTC AG

Agenda

- Client-Side Graphics
- Client-Server-Kommunikation
- Client-seitige Storage-Lösungen



HTML5?



HTML5?

- **1997:** HTML 4.0 (W3C Recommendation)
 - Danach keine funktionale Weiterentwicklung mehr
 - Ebenfalls keine Weiterentwicklung von **CSS**
- **2004:** Gründung der WHATWG
 - Web Hypertext Application Technology Working Group
 - Erster Vorschlag für **HTML5:** „Web Applications 1.0“
- **seit 2007:** W3C beteiligt sich an HTML5-Entwicklung
- **2014:** geplante Verabschiedung von HTML5

Agenda

- Client-Side Graphics
- Client-Server-Kommunikation
- Client-seitige Storage-Lösungen



Client-Side Graphics

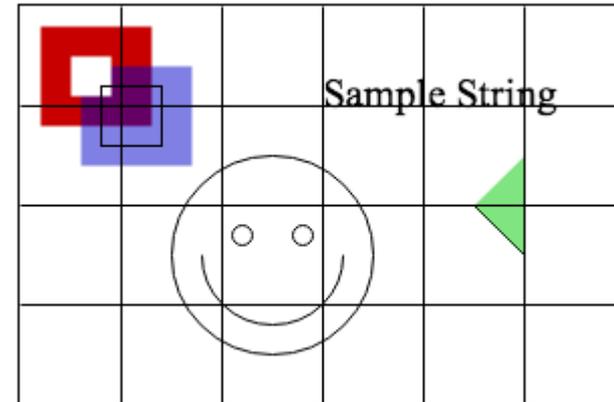
- Ziel:
 - Generierung und Interaktion mit graphischen Inhalten im Browser
- Bisher:
 - Bilder generieren auf Server-Seite + Image Maps
 - Flash
- HTML5:
 - Canvas
 - SVG



Client-Side Graphics



- Canvas
 - Zeichnen auf Graphik-Puffer
 - Mittels JavaScript
 - 2D/3D



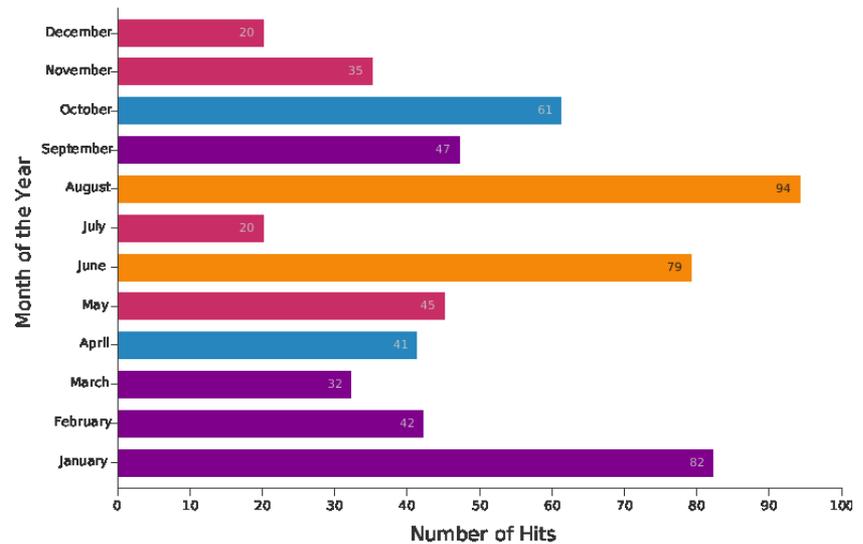
```
<canvas id="myCanvas" width="200" height="100"/>
```

```
var c = document.getElementById('canvas');
var ctx = c.getContext("2d");
ctx.fillRect (10, 10, 55, 50);
```

Client-Side Graphics



- Scalable Vector Graphics
 - XML-Modellierung visueller Elemente
 - Manipulation wie sonstige DOM-Elemente



```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect width="50" height="100" style="fill:rgb(100,140,255)"/>
</svg>
```

Client-Side Graphics

Canvas	SVG
Raster-basiert	Vektor-basiert
Primitives Event-Handling	Events für dargestellte Objekte
Manipulation per JavaScript	Deklaratives Erstellen Manipulation per JavaScript + CSS
Bessere Performance bei <ul style="list-style-type: none"> • Geringer Fläche • Vielen dargestellten Objekten 	Bessere Performance bei <ul style="list-style-type: none"> • Großer Fläche • Wenigen dargestellten Objekten
Mehr Möglichkeiten <ul style="list-style-type: none"> • 3D mit WebGL • Video/Webcam • ... 	



Client-Side Graphics

- Problem:

Canvas /
SVG:

				
9+	2+	4+	3.1+	9+
3.6+	7+	5.1+	11.6+	
3/2011	10/2006	1/2010	3/2008	6/2006

Windows XP: IE6
 Windows Vista: IE7
 Windows 7: IE8

Client-Side Graphics

- Lösung:

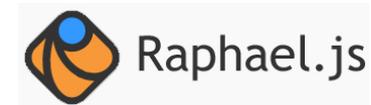


- Fallback auf **VML**
- VML war erste Vektorgraphik-Implementierung für Browser
- Seit **Internet Explorer 5 (!)**

Client-Side Graphics

- Libraries

- Raphael.js ([http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/))
 - SVG-Library mit Fallback auf VML



- dojo.gfx (<https://dojotoolkit.org/>)
 - Vector Graphics Library
 - Abstrahiert
 - Canvas (favorisiert!)
 - SVG
 - VML
 - Silverlight



- <https://code.google.com/p/explorercanvas/>
- <http://flashcanvas.net>

Client-Side Graphics

- Demos



- Raphael.js ([http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/))
 - SVG-Library mit Fallback auf VML



Raphael.js

- dojo.gfx (<https://dojotoolkit.org/>)
 - Vector Graphics Library
 - Abstrahiert
 - Canvas (favorisiert!)
 - SVG
 - VML
 - Silverlight

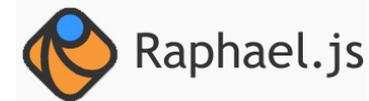


- <https://code.google.com/p/explorercanvas/>
- <http://flashcanvas.net>

Client-Side Graphics

- Demos

- Raphael.js ([http://raphaeljs.com/](http://dmitrybaranovskiy.github.io/raphael/))
 - SVG-Library mit Fallback auf VML



- dojo.gfx (<https://dojotoolkit.org/>)
 - Vector Graphics Library
 - Abstrahiert
 - Canvas (favorisiert!)
 - SVG
 - VML
 - Silverlight



- <https://code.google.com/p/explorercanvas/>
- <http://flashcanvas.net>

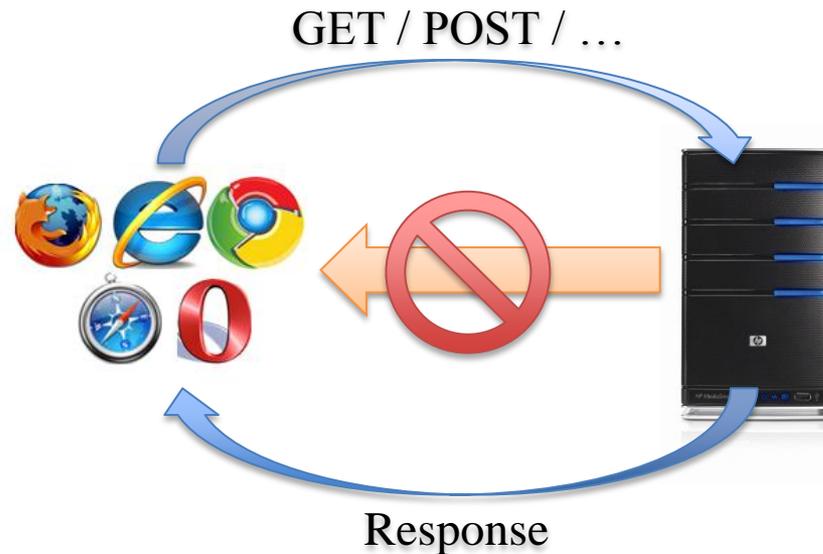
Agenda

- Client-Side Graphics
- Client-Server-Kommunikation
- Client-seitige Storage-Lösungen



Client-Server-Kommunikation

- Problem:
 - HTTP-Kommunikation geht immer vom Client aus
 - Echtzeit-Nachrichten vom Server?



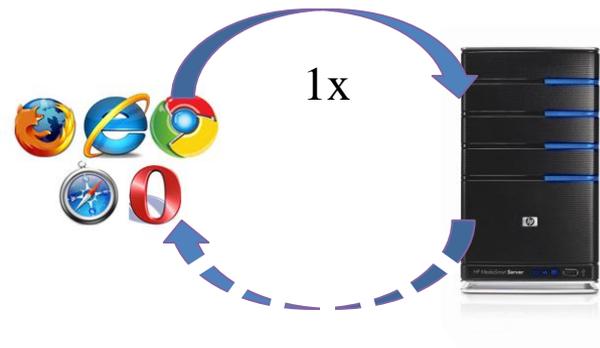
Client-Server-Kommunikation

- Bisher:

- Long-Running Requests



- HTTP Streaming



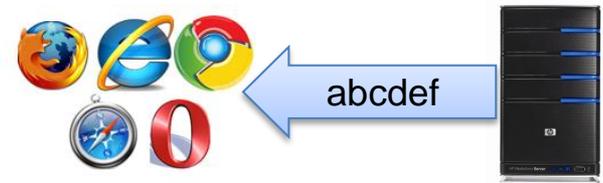
Client-Server-Kommunikation

- HTML5:

- Web Sockets 
 - **Bidirektionale** Kommunikationsverbindung (TCP)
 - Zeichenketten / binäre Daten



- Server-Sent Events
 - **Unidirektionale** Kommunikationsverbindung
 - Zeichenketten
 - Automatischer Reconnect
 - Message IDs



Client-Server-Kommunikation

- Vorteile **beider** Technologien gegenüber den alten:
 - Permanente Verbindung
 - Kein HTTP-Overhead
 - Wirklich real-time (im Vergleich zu long-polling)
- Vorteile von **Web Sockets** gegenüber SSE:
 - Bi-Direktional
- Vorteile von **SSE** gegenüber Web Sockets:
 - Einfachere API
 - Automatische Wiederverbindung
 - Event-IDs
 - Einfache HTTP-Verbindung (kein HTTP Upgrade)

Client-Server-Kommunikation

- Problem:

					
Web Sockets:	10+ 10/12	6+ 8/11	14+ 9/11	6+ 7/12	12+ 6/12
Server-Sent Events:	-	6+ 8/11	6+ 9/10	5+ 7/11	9+ 6/06

• IE	<10 :	~15%
• Safari	< 6 :	~2.3%
• Firefox Version	< 6 :	~1%
• Opera	< 12:	~0.1%

Client-Server-Kommunikation

Web Sockets:

Server-Sent Events:

				
10+ 10/12	6+ 8/11	14+ 9/11	6+ 7/12	12+ 6/12
-	6+ 8/11	6+ 9/10	5+ 7/11	9+ 6/06

- Lösungsmöglichkeiten:

- **Polyfill**

- Server-Sent Events:

<https://github.com/Yaffle/EventSource>



8+ 3+

- Web Sockets:

<https://github.com/gimite/web-socket-js>



10+

- **Abstraktion** über Kommunikationstechnologien

- <https://github.com/Atmosphere/atmosphere>



6+ 3+

- <http://cometd.org/>

Client-Server-Kommunikation

Web Sockets:

Server-Sent Events:



- Demo:

- Polyfill

- Server-Sent Events:

<https://github.com/Yaffle/EventSource>



8+ 3+

- Web Sockets:

<https://github.com/gimite/web-socket-js>



10+

- Abstraktion über Kommunikationstechnologien

- <https://github.com/Atmosphere/atmosphere>



6+ 3+

- <http://cometd.org/>

Client-Server-Kommunikation

Web Sockets:

Server-Sent Events:

				
10+ 10/12	6+ 8/11	14+ 9/11	6+ 7/12	12+ 6/12
-	6+ 8/11	6+ 9/10	5+ 7/11	9+ 6/06

- Demo:

- **Polyfill**

- Server-Sent Events:

<https://github.com/Yaffle/EventSource>



8+



3+

- Web Sockets:

<https://github.com/gimite/web-socket-js>



10+

- **Abstraktion** über Kommunikationstechnologien

- <https://github.com/Atmosphere/atmosphere>



6+



3+

- <http://cometd.org/>

Agenda

- Client-Side Graphics
- Client-Server-Kommunikation
- Client-seitige Storage-Lösungen



Clientseitige Storage-Lösungen

- Daten auf der Client-Seite persistieren



Clientseitige Storage-Lösungen

- Wozu?
 - Offline-Applikationen
 - Daten-Caching
 - Entlastung des Servers
 - Usability
 - ...

Clientseitige Storage-Lösungen

- Bisher:

- Cookies

- bis 4kb
 - Overhead in jedem Request



- Flash 8 persistent storage

- max. 100kb (default-Wert)



- IE5+: userData

- 64kb - 128kb



- DOM storage

- Nur Firefox und Safari
 - 5MB



Clientseitige Storage-Lösungen

- HTML5
 - Application Cache
 - HTML, JavaScript, CSS
 - Bilder, Videos
 - sonstige statische Ressourcen
 - Offline Storage
 - Laufzeitdaten
 - Flach (Key-Value)
 - Strukturiert (Datenbank)



Clientseitige Storage-Lösungen

- Bisher:
 - Cookies
 - bis 4kb
 - Overhead in jedem Request
 - Flash 8 persistent storage
 - max. 100kb (default-Wert)
 - IE5+: userData
 - 64kb - 128kb
 - DOM storage
 - Nur Firefox und Safari
 - 5MB



Clientseitige Storage-Lösungen



8+



4+



4+



4+



11+

- HTML5: Web Storage

- Speichern von Strings (Key-Value)

- Scopes

- Session

(Daten gehen nach Ablauf der Session verloren)

- **Local**

(Daten bleiben persistent auf dem Client)

- Limits

- 5MB (W3C, Chrome, Firefox, Safari, Opera)

- 10MB (IE)

} oder mehr...

```
localStorage.setItem('key', 'value');  
localStorage.getItem('key');
```

Client-Server-Kommunikation

- Problem:



Clientseitige Storage-Lösungen

- Lösung: Polyfill-Libraries

- Store.js

- Abstraktion über

- HTML5 Web Storage

- userData

- DOM-Storage (globalStorage)

- <https://github.com/marcuswestin/store.js>



6+



3.5+



5+



4+



10+



- Weitere:

- lawnchair (<http://brian.io/lawnchair>)

- PersistJS (<https://github.com/jeremydurham/persist-js>)

Clientseitige Storage-Lösungen

- Demo



- Store.js

- Abstraktion über

- HTML5 Web Storage

- userData

- DOM-Storage (localStorage)

- <https://github.com/marcuswestin/store.js>



6+



3.5+



5+



4+



10+

- Weitere:

- lawnchair (<http://brian.io/lawnchair>)

- PersistJS (<https://github.com/jeremydurham/persist-js>)

Clientseitige Storage-Lösungen

- Clientseitige Datenbanken



Clientseitige Storage-Lösungen



-



-



4+



4+



11+

- Web SQL

- SQL-Datenbank im Browser
- Erfüllt ACID-Regeln (Transaktionen)
- Daten bleiben persistent auf dem Client
- *W3C: Deprecated!*



Clientseitige Storage-Lösungen



10+



10+



23+



-



15

- IndexedDB

- No-SQL-Datenbank
 - Object Stores
 - Indizes
- Erfüllt ACID-Regeln (Transaktionen)



```
var request = indexedDB.open("todos", 1);
request.onupgradeneeded = function(e) {...}
request.onsuccess = function(e) {
    todoDb = e.target.result;
}
request.onerror = function(e) {...}
```

```
var store =
    todoDb.transaction("todos", "readonly").objectStore("todos");
store.get(123).onsuccess = function(e) {
    var todo = e.target.result;
    ...
}
```

Clientseitige Storage-Lösungen

- Problem



BROWSER SUPPORT

									
	4+	4+	4+	11+	8+	6+	3+	—	11+
<i>Web Storage - name/value pairs</i> <small>↗</small>	4+	4+	4+	11+	8+	6+	3+	—	11+
<i>IndexedDB</i> <small>↗</small>	23+	10+	—	15	10	—	—	—	14
<i>Web SQL Database</i> <small>↗</small>	4+	—	4+	11+	—	6+	3+	—	11+

Quelle (angepasst): <http://www.html5rocks.com/en/features/storage>

Clientseitige Storage-Lösungen

- Lösung: Polyfill-Libraries
 - <http://nparashuram.com/IndexedDBShim/>
 - IndexedDB-API
 - Fallback-Implementierung über WebSQL
 - Weitere:
 - <http://jensarps.github.com/IDBWrapper/>
 - <https://bitbucket.org/ytkyaw/ydn-db/wiki/Home>

Clientseitige Storage-Lösungen

BROWSER SUPPORT

									
	4+	4+	4+	11+	8+	6+	3+	—	11+
<i>Web Storage - name/value pairs</i> ↗	4+	4+	4+	11+	8+	6+	3+	—	11+
<i>IndexedDB</i> ↗	23+	10+	—	15	10	—	—	—	14
<i>Web SQL Database</i> ↗	4+	—	4+	11+	—	6+	3+	—	11+

Clientseitige Storage-Lösungen

- Polyfill-Libraries



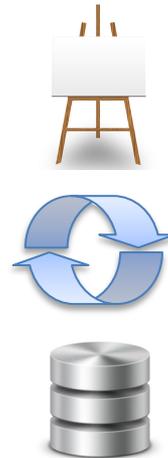
- <http://nparashuram.com/IndexedDBShim/>

- IndexedDB-API
- Fallback-Implementierung über WebSQL

- Weitere:

- <http://jensarps.github.com/IDBWrapper/>
- <https://bitbucket.org/ytkyaw/ydn-db/wiki/Home>

Fazit



Geht schon...!

2.– 5. September 2013
in Nürnberg



Herbstcampus

Wissenstransfer
par excellence

Vielen Dank!

Yann Massard

BTC AG



10 Jahre

Kunden verstehen, Prozesse neu denken, IT weiterentwickeln

Menschen beraten – BTC AG

BTC Business Technology Consulting AG ist seit zehn Jahren an der Seite ihrer Kunden – jetzt und in der Zukunft.

Erfahren Sie mehr auf www.btc-ag.com/10jahre.

