

Die JavaScript (R)Evolution

Artur Lojewski, Dr. Frank Gerhardt

Java Forum Stuttgart, 2. Juli 2009

TIOBE Software: Tiobe Index - Mozilla Firefox

File Edit View History Bookmarks Tools Help 1

http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

TIOBE Programming Community Index for June 2009

June Headline: JavaScript has never been so popular

The TIOBE Programming Community index gives an indication of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. The popular search engines Google, MSN, Yahoo!, Wikipedia and YouTube are used to calculate the ratings. Observe that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found [here](#).

Position Jun 2009	Position Jun 2008	Delta in Position	Programming Language	Ratings Jun 2009	Delta Jun 2008	Status
1	1	=	Java	20.147%	-0.74%	A
2	2	=	C	16.779%	+1.27%	A
3	3	=	C++	10.594%	-0.21%	A
4	4	=	PHP	9.675%	-0.53%	A
5	5	=	(Visual) Basic	7.943%	-1.84%	A
6	7	↑	Python	4.756%	-0.14%	A
7	8	↑	C#	4.536%	+0.48%	A
8	9	↑	JavaScript	4.021%	+1.09%	A

Done

One active download (39 minutes remaining)

Überblick

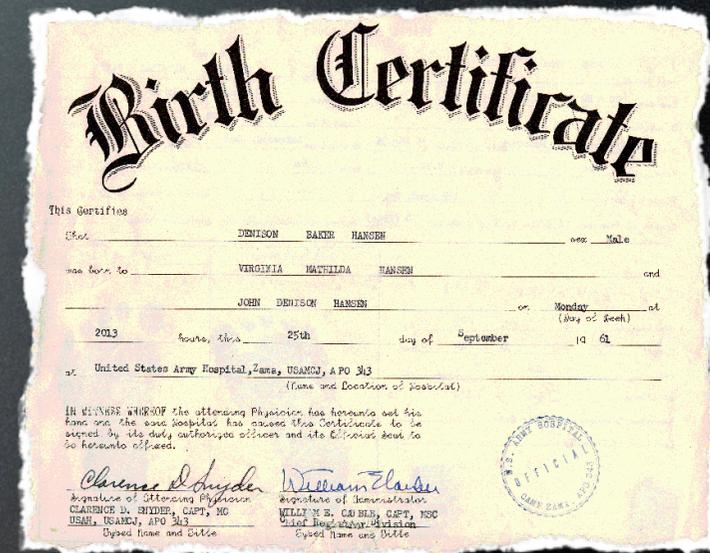
- Einführung
- Die Sprache
- Security
- Tools und Bibliotheken
- Praxis-Beispiel
- Ausblick

Zurück in das Jahr 1995...

- Januar: offizielles Java 1.0 Release
- Web: Netscape / Microsoft
 - statische HTML-Seiten
 - es fehlt etwas...
- Lösung: Die Script-Sprache JavaScript wird geboren!

Geburtsurkunde

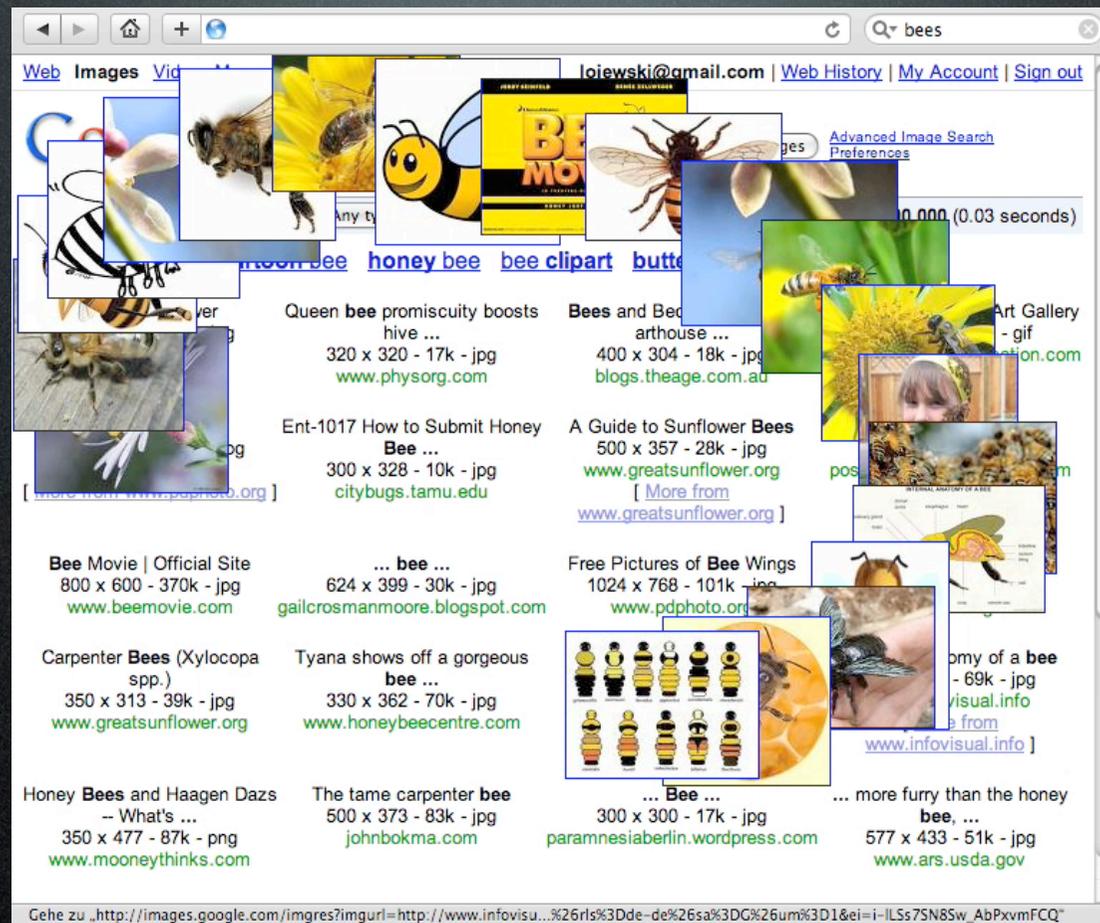
- Name: Mocha ↗ LiveScript
↗ JavaScript/ECMAScript
- Geburtsdatum: Mai, 1995
- Geburtsort: Netscape, CA
- Vorfahren: Scheme, Self
- Erzeuger: Brenden Eich
- Geburtsdauer: **10 Tage!**



JavaScript in der Anfangszeit

- Java-Applets sind gescheitert...
- Missbrauch von JavaScript
 - Laufzeitfehler im Browser
 - Security issues - People turn off JS
- Web-Designer übernahmen das Ruder...

JavaScript rules!



JavaScript Historie

http://en.wikipedia.org/wiki/Browser_wars



browser war #1



browser war #2

1.0 Mar '96 1.2 Jun '97 1.4 '03 1.6 Nov '05 1.8 Jun '08



Aug '96 Oct '98 Nov '00 Oct '06
1.1 1.3 1.5 1.7

↑
ECMA 262 1st/2nd ed. ECMA 262 3rd ed./ES3

~~ECMA 262 4rd ed. "ES4"~~



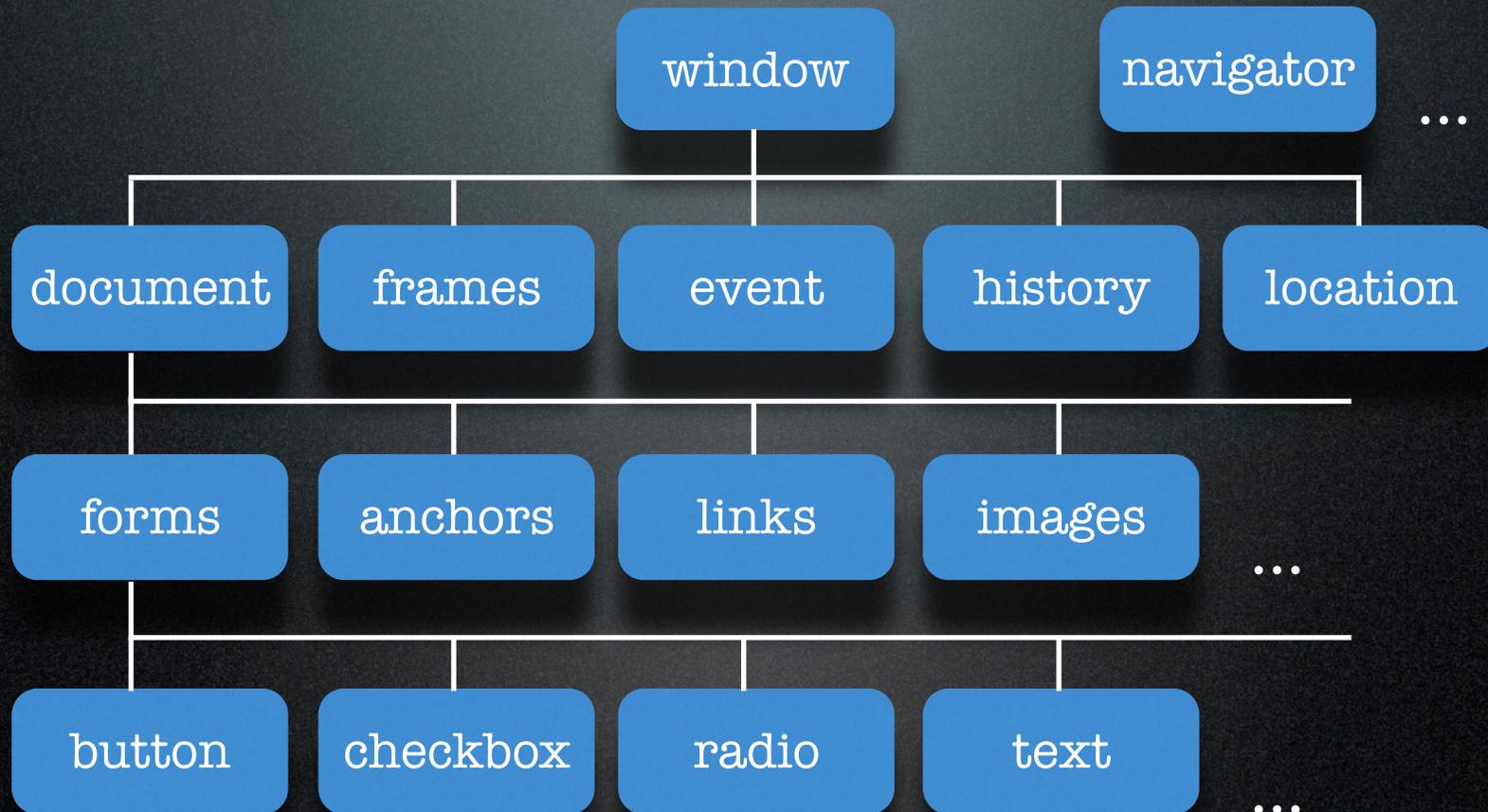
ECMA 262 5rd ed./ES5 finalized

ECMAScript "Harmony" 6th ed.?

JavaScript / DOM

- Document Object Model:
 - Schnittstelle für Programme zu Inhalt, Struktur und Stil von Dokumenten
 - cross-platform
 - sprachunabhängig
- API-Zugriff
 - Fenster, Bilder, Links, etc.
 - Ereignisse und Timer
 - W3C als Instanz für Spezifikationen DOM 1, 2, 3

Object Hierarchy



if implements return false final finally
abstract typeof function else float export
in with goto super enum void
boolean short synchronized try
instanceof short static throws extends
break var for this switch
null Die Sprache do while
byte package debugger delete private
int const while transient
case char default continue
catch long class volatile
throw native public protected
interface new try true import double

Semicolon Insertion



Was passiert hier?

```
return; // return 'undefined'  
{  
  status: true  
};
```



```
return {  
  status: true  
};
```

JavaScript / Sprache

- Scripting Sprache (HTML)
- Cross-plattform
- Weakly typed `var i = 1; i = 'aString'`
- Prototype-based object construction
- First-class Funktionen

Imperativ & Strukturiert

- Sprach-Syntax ähnelt Java / C
 - `if()` / `switch()` / `while()` / etc.
- Kein block-level scoping wie in C!
 - ab JS 1.7 möglich mit "let"

```
var x = 5;
var y = 0;
let (x = x + 10, y = 12) {
    print(x+y + "\n");    =>27
}
print(x+y + "\n");      =>5
```

Dynamisch

- Dynamisch Typisiert
 - Typen sind mit Values assoziiert, nicht Variablen!
 - Values können später andere Typen bekommen
 - Unterstützt Prüfung auf Typen
- Objekte als assoziierte Arrays
 - `obj.x <=> obj["x"]`
 - Properties können zur Laufzeit zugefügt, gelöscht und verändert werden!
 - Laufzeit-Evaluierung mit `"eval()"`

Funktional

- Funktion == first class Objekt
 - Funktionen sind selber Objekte!
 - haben Properties
 - dienen als Parameter
- Innere Funktionen und Closures
- Innere Funktionen werden beim Aufruf der äusseren Funktion kreiert
- Variablen der äusseren Funktion bleiben auch nach Aufruf vorhanden, solange innere Funktion da ist.

Closure

... ist eine Programmfunktion, die beim Aufruf ihren Definitionskontext reproduziert, selbst wenn dieser Kontext außerhalb der Funktion schon nicht mehr existiert. Closures „konservieren“ also ihren Kontext.

Quelle: wikipedia

Prototypen

- Prototypen anstatt Klassen
 - definieren Objekt-Eigenschaften (Methoden, Vererbung)
- Funktionen als Konstruktor
 - Bsp.: `var obj = new Funktion();`
- Funktionen als Methoden

Verschiedenes

- Laufzeitumgebung ist im Browser
- Funktionen mit mehreren Argumenten (arguments object)
- Arrays und Objekt-Literale können einfach erzeugt werden
- Regular-Expressions werden unterstützt

JavaScript / Bad Parts

- keine standard class libraries
- kein Modul-Konzept (encapsulation)
- Globale Variablen
- Kein Block-Scope
- Semicolon insertion (JSLint)
- Sicherheit: eval()
- Arrays sind Objekte (Performance)
- 2 Vergleichs-Operatoren == und ===
- new-Operator, weglassen bedeutet this wird an global object gebunden

JavaScript Implementierungen

- SpiderMonkey/TraceMonkey - C, Firefox/Mozilla
- Rhino - Java (JDK 6), Mozilla
- V8 - C++, Chrome/Google
- SquirrelFish - C++, WebKit/Apple
- JScript - C++, IE/Microsoft
- ...

Rare rhinos found in Java, Dez. 2008



ca. 1 MB!



The Javan rhino (left), which is distinguished by its small size, single horn and loose skin folds, is likely the most endangered large mammal on the planet, according to WWF. -- PHOTO: AFP

Security



ECMAScript 5 / Strict Mode



- eingeschränkter Betriebs-Kontext
- verhindert unsichere Aktionen:
Zugriff auf globales Objekt, ...
- schaltet bestimmte Features ab:
`eval("var a = false");` `with() {}`
- "use strict": globaler Scope od. Funktion
- für bessere Fehlerüberprüfung und
Programmsicherheit!

JavaScript Security / Google Caja

- JavaScript kennt 3 Sicherheitsstufen
 - aus | darf alles | sandbox
- Caja ("Box") = Capabilities Attenuate JavaScript Authority
- Sicherer als Sandbox!
- Caja ist Untermenge von JavaScript

JavaScript Security / Google Caja

- prüft Zugriff/Modifikation auf Daten mittels zusätzlicher Methoden
- Daten sind nie alleine, sondern immer durch Zugriffsmethoden eines Objektes gekapselt
- alle Skripte gehen zur Laufzeit durch einen Filter (safety-checker)

JavaScript Security / Google Caja

- Caja lässt zu, dass gesicherte und ungesicherte Web-Applikationen im gleichen Kontext ablaufen (Mashups)
- Nachrichten zwischen Web-Applikationen können untereinander sicher ausgetauscht werden
- Caja schreibt JS/HTML/CSS neu

Hype-Alarm!

- AJAX
- Web 2.0
- Gmail

- CHEF: Das will ich auch!

Put the client
on the client!

Tools & Frameworks



JavaScript / JSON

- JSON: JavaScript Object Notation
- Daten-Austausch Format
- leicht zu lesen / schreiben (Mensch)
- leicht zu parsen / generieren (Comp.)
- Untermenge von JavaScript
- sprachunabhängig

JavaScript / JSON

- Übertragung von Daten zwischen Computern
- Ansammlung von name/value-Paaren
- XMLHttpRequest / eval()
- Oh, kein XML... ;-)

JSON vs. XML

JSON

```
{"menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      {"value": "New", "onclick": "CreateNewDoc()"},  
      {"value": "Open", "onclick": "OpenDoc()"},  
      {"value": "Close", "onclick": "CloseDoc()"}  
    ]  
  }  
}}
```

XML

```
<menu id="file" value="File">  
  <popup>  
    <menuitem value="New" onclick="CreateNewDoc()" />  
    <menuitem value="Open" onclick="OpenDoc()" />  
    <menuitem value="Close" onclick="CloseDoc()" />  
  </popup>  
</menu>
```

JavaScript Verifier / JSLint

- Prüft JavaScript Programm auf potentielle Probleme
 - Syntax, Variablen, Komma, etc.
- JavaScript Source, HTML oder JSON
- Generiert Report zu jeder Funktion mit verständlichen Meldungstext

JavaScript Bibliotheken

- GWT
- Dojo
- script.aculo.us
- JQuery
- YUI
- prototype



Beispiel: Dojo Toolkit

- JavaScript für Web-Applikationen (Charts, Calendars, File Uploads, ...)
- klein (ca. 26 KB) und schnell
- build-system / module loading
- BSD Lizenz / Academic Free License
- Dojo Foundation: Unterstützung von grossen IT-Firmen!

Dojo Form Widgets Test

http://demos.dojotoolkit.org/demos/form/

Job Application Form

This is just a little demo of dijit's form widgets

Name And Address As you type in the text below, notice how your input is auto corrected and also the auto completion on the state field.

Name *

Address *

City * State Zip *

DOB *

Desired position Custom checkboxes and radio buttons...

Position IT Marketing Business Hours Full time Part time

Education and Experience slider and spinner ...

Education level high school college masters PhD Work experience (years, 0-40)

Self description Rich text editor that expands as you type in text



Dear Audience,

what you see right now is a web page created via the **Dojo toolkit**. Dojo is written in JavaScript, which is the glue language of the web.

You're using JavaScript all the time when you're surf in the web, so it might be the right time to look closer to JavaScript and related technologies.

Best Regards,

your JS spokesperson

JavaScript in Eclipse 4 (e4)

e4 wird andere Sprachen für die Entwicklung von Plug-ins unterstützen, an erster Stelle JS

<http://wiki.eclipse.org/E4/JavaScript>

```
{
  "Bundle-SymbolicName": "sample.jsbundle",
  "Bundle-Version": "1.0",
  "Bundle-ScriptPath": "script.js",
  "Import-Package": "a.resource;version=[1.0.0,2.0.0)",
  "Export-Package": "sample.resource;version=1.0.0",
  "Require-Bundle": "some.other.bundle",
}
<extension
  point="org.eclipse.equinox.http.registry.servlets">
  <servlet
    alias="/jsHello"
    class="org.eclipse.e4.languages.javascript.registry.JavaScriptFactory:some.func">
  </servlet>
</extension>
```

http://wiki.eclipse.org/E4/UI/Toolkit_Model/org.eclipse.e4.emf.javascript

SWT Portierung auf Dojo

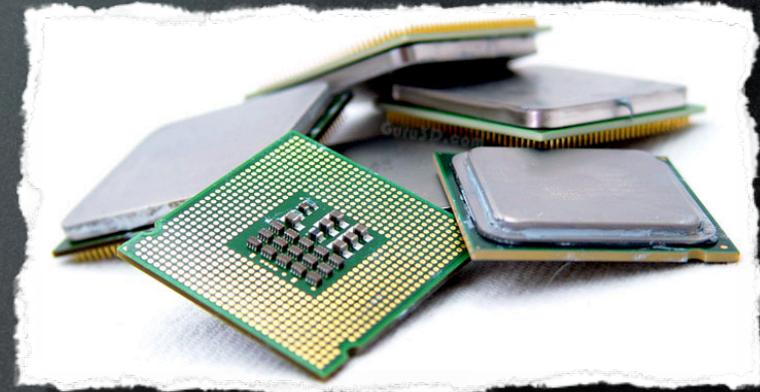
- SWT unterstützt heute > 10 Plattformen (Win, Mac, GTK, ...)
- Für e4 wird eine Portierung auf RIA-Technologien entwickelt (Flash, Silverlight, Dojo)
- Der Dojo-Port funktioniert im Prinzip wie GWT
 - Programmierung mit SWT Java API
 - Compilierung (mit GWT Compiler) nach JS
 - Zur Laufzeit JavaScript und Dojo Widgets

JavaScript Server serverjs

- Motivation: JS auf Client und Server
- Ziele
 - Standard-Bibliothek
 - Mechanismus um Module zu laden
 - Package & Distribution

JavaScript und Threads?

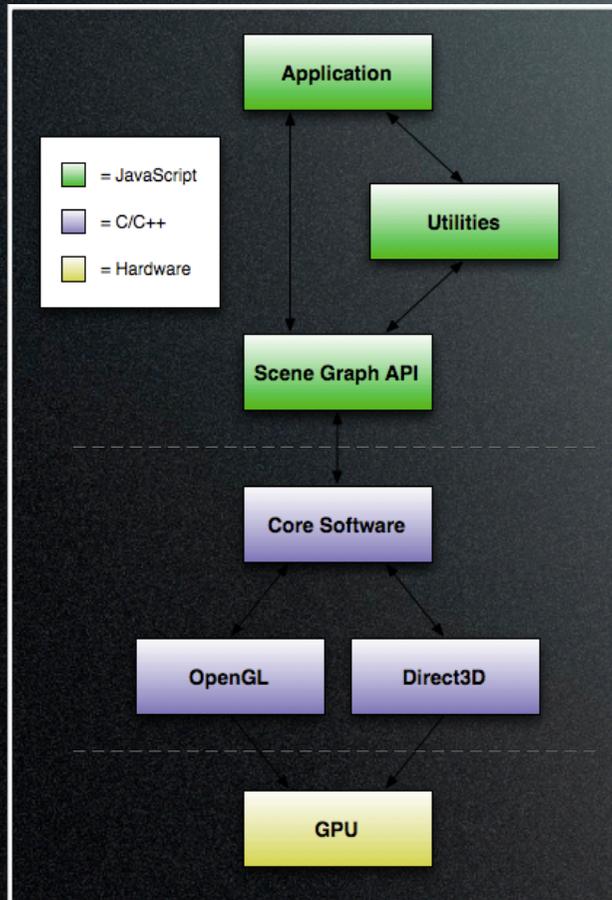
- JavaScript ist/wird nicht multi-threading fähig - OK
- Google Gears Worker Pools (shared-nothing background threads)
- DOM workers (Mozilla)
- Don't Panic - kreative Entwickler...



Google O3D

- JavaScript-API für interaktive 3D-Web-Applikationen
- browser-plugin (Windows/Mac)
- converter / loader für 3D-Inhalte
- für JavaScript-Entwickler mit 3D-Grafik Kenntnissen / Spiele-Entwickler

O3D Architektur



Ausblick

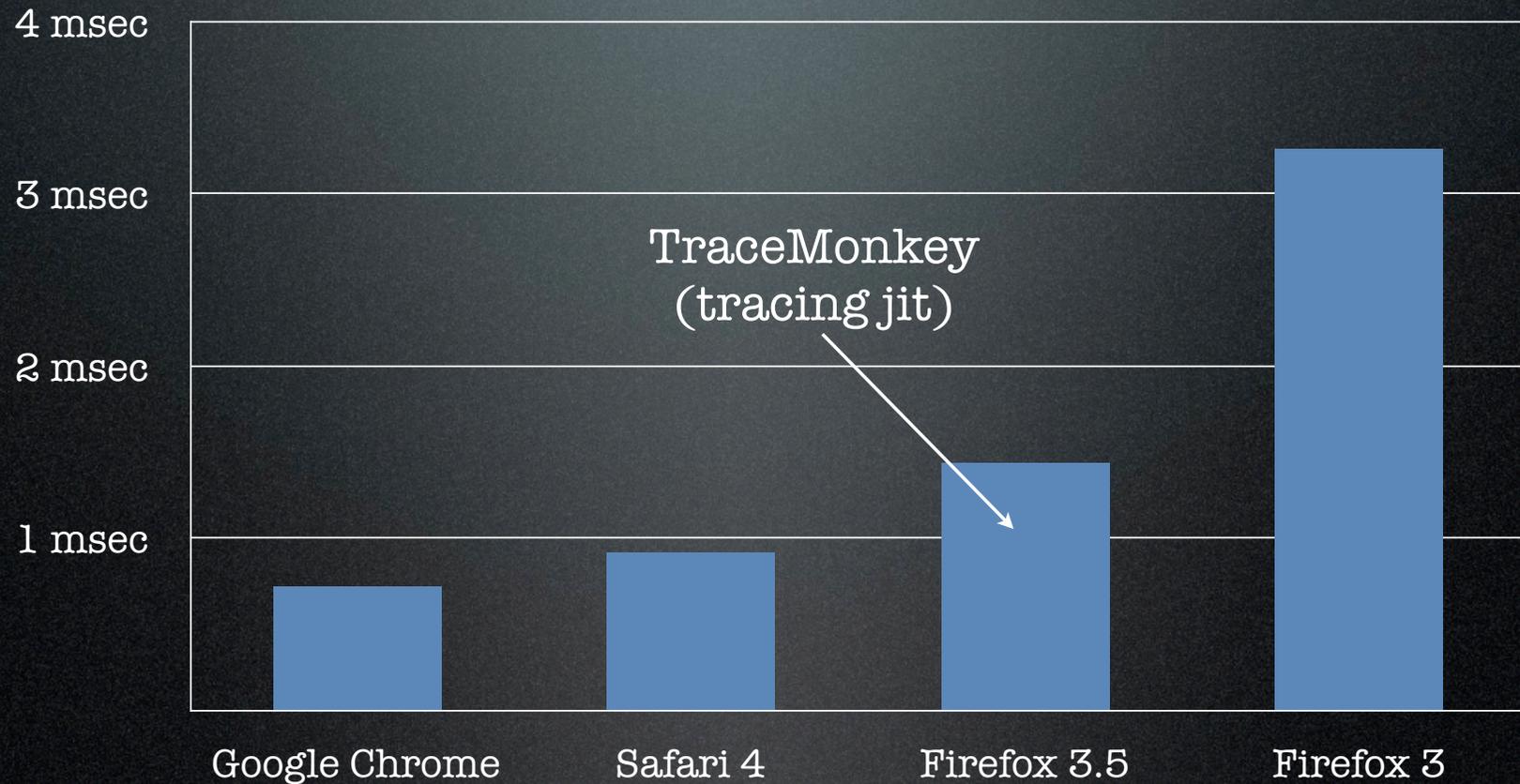


JavaScript + HTML 5

- JavaScript + HTML 5
 - video/audio element
 - channel messaging (cross-document)
 - offline storage
- Flash, Flex
- Silverlight
- JavaFX

Newsflash von gestern...

■ SunSpider JavaScript Benchmark (kleiner ist besser)



Quelle: cnet uk, 1. Juli 2009

Praxis-Beispiel

JavaScript in der Datenbank

- Der Client verwendet JavaScript
- Der Server verwendet JavaScript
- Zur Kommunikation werden REST und JSON verwendet
- Und was ist mit der Datenbank?
- Will man sich damit beschäftigen, JSON auf Tabellen zu mappen?



apache CouchDB relax

Home

Home

Search ↗

Downloads

Screenshots

License ↗

Road Map

Documentation

Introduction

Overview

Wiki ↗

Books ↗

Get Involved

Mailing Lists

Issue Tracking

Source Code

Coding Standards

Committers

Planet ↗

The CouchDB Project

Apache CouchDB is a distributed, fault-tolerant and schema-free document-oriented database accessible via a RESTful HTTP/JSON API. Among other features, it provides robust, incremental replication with bi-directional conflict detection and resolution, and is queryable and indexable using a table-oriented view engine with JavaScript acting as the default view definition language.

CouchDB is written in Erlang, but can be easily accessed from any environment that provides means to make HTTP requests. There are a multitude of third-party client libraries that make this even easier for a variety of programming languages and environments.

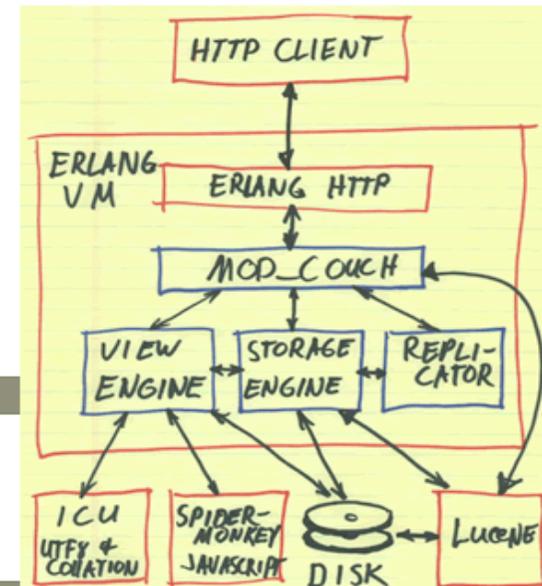
See the [introduction](#) and the [technical overview](#) for more information.

How to Get Started

Read the [introduction](#) and the [technical overview](#) to learn more about the design and basic concepts of CouchDB. To give it a try, download the CouchDB code from the [Subversion repository](#) and follow the included build instructions.

How to Get Involved

Join the [mailing lists](#) and participate in the discussions around the development of CouchDB. If you encounter a problem and have an idea how to fix it, please start by making a patch and filing it with our [issue tracking system](#).





- Cluster of unexpensive computing hardware
- Dokumenten-orientierte Datenbank für Ressourcen im Sinne von REST
- Speicherung von JSON Dokumenten u.a.
- Abfrage über Views, die in JavaScript definiert sind
- Schema-frei, einfache Evolution
- Peer-2-peer Replikation, offline-fähig, ideal für Cloud Computing
- Eventual Consistency, keine Transaktionen, Motto: relax

Syntax
JavaScript
HTML5 vs. Flash
HTML5 vs. Silverlight
Scripting
AJAX
Client
Security
JSON
Browser
Tools
Fragen?
Libraries
Tools
Funktional
Server
Performance
multi-threading
Zukunft
Frameworks
testing
mag-ich-nicht