

<title>noch unbekannt</title>

Thema der Masterarbeit von Kjartan Ferstl

Zugang und Interesse

Wir sind im Unternehmen viel im Bereich Web-Entwicklung unterwegs und überlegen umfangreiche Desktopapplikationen als Webapplikation zu implementieren. Durch technische Einschränkungen sind solche Entwicklungen meist aufwendiger zu implementieren und durch die Tatsache, dass es sich beim JavaScript eine nicht typensichere Sprache ist, auch wesentlich Fehleranfälliger. Für uns als Unternehmen stellt sich daher die Frage wie die Web-Landschaft in fünf oder sogar zehn Jahren aussieht und wie riskant es ist, sich auf Frameworks wie AngularJS, Backbone oder Knockout zu stützen.

Thema kurz beschrieben

JavaScript gibt es seit 1995, damals es unter dem Namen LiveScript integriert in den Netscape Navigator 2.0 Browser. Vor dieser Programmiersprache bestand das Web hauptsächlich aus statischen Webseiten und vereinzelt integrierten Java-Applets. JavaScript ermöglichte es erstmals Elemente in den HTML Code der Seite einzufügen und andere zu löschen. Mit dem JSON-Datenformat (Douglas Crockford, 2001) und der Möglichkeit im Hintergrund mit Webserver zu kommunizieren (AJAX, Google Maps, 2005) öffneten sich neuen Türen für dynamische Webseiten. Seitdem haben sich zahlreiche JavaScript-Bibliotheken wie jQuery, Backbone oder AngularJS etabliert, die die Entwicklung von Interaktiven Webseiten stark beschleunigen.

Web-Anwendungen heute oftmals schon Desktop-Applikationen, vom Klassiker der Email-Anwendung bis hin zur kollaborativen Tabellenkalkulationsanwendung oder Mehrspieler-Onlinespielen. Der Web-Browser wird hierbei immer mehr zur nächsten Abstraktionsebene, bzw. einer plattformunabhängigen virtuellen Maschine.

Durch die Plattform Node.JS ist es außerdem möglich Serveranwendungen in JavaScript zu schreiben und in Performancevergleichen sind diese sogar oft schneller als herkömmliche kompilierte Serverapplikationen.

Beispiel: <http://www.haneycodes.net/to-node-js-or-not-to-node-js/>

JavaScript soll Funktionalitäten für die Fortschritte des dynamischen Webs mitbringen und zudem den Ansprüchen verschiedenster Applikationen gerecht werden. Hierbei werden noch einige interessante Problemstellungen auftauchen, wie zum Beispiel die Tatsache dass JavaScript bis heute rein single-threaded ist. So wird JavaScript unter Umständen noch einige Hürden nehmen müssen die andere heute präsente Programmiersprachen bereits gemeistert haben.

Mit dieser Arbeit will ich die folgenden Fragen beantworten:

- Wo liegen die Stärken und Schwächen von JavaScript und Varianten davon?
- Wie könnten die Lösungen für aufkommende Themen aussehen und wie wirken sich diese auf die Entwicklung von JavaScript aus?
- Wie entwickeln sich andere Sprachen und Frameworks in Bezug auf das Web?

- Bleibt die Plattformunabhängigkeit und schnelle Verfügbarkeit der treibende Faktor des Webs?
- Wie wird sich das Web in den nächsten Jahren entwickeln?

Randthemen die ich anschneiden will

Plattformunabhängigkeit

Eine der treibenden Kräfte des Webs ist die Plattformunabhängigkeit. Wie Plattformunabhängig wird das Web in Zukunft wirklich sein, wenn man bereits heute vielmals nur Browserversionen der letzten zwei Jahre unterstützt und für mobile Geräte zusätzlichen Entwicklungsaufwand investiert. Für eine Desktopapplikation sind fünf Jahre meist kein relevantes Alter, bei Webapplikationen können drei Jahre ohne update zum Problem werden.

Zugänglichkeit

Applikationen im Internet benötigen meist keine zusätzliche Installation und sind binnen Sekunden am Endgerät verfügbar. Mit der zunehmenden Verbreitung von Software über das Internet und den Märkten für die verschiedenen Plattformen (Google Play Store, Windows Store, usw.) ist auch herkömmliche Software einfach zugänglich.

Bytecode im Browser

Wir JavaScript, als momentan einzige universelle dynamische Sprache im Browser eines Tages durch Bytecode ersetzt?

<http://www.infoq.com/news/2013/05/javascript-browser-bytecode>