

## AUSGANGSSITUATION

Softwareentwicklung für den Browser ist ohne Zweifel eine komplexe Angelegenheit. Die Gründe für diesen Umstand sind so vielfältig, dass darüber eine ganze Forschungsarbeit geschrieben werden könnte. Je nach Blickwinkel könnte man von der technischen Seite argumentieren und beispielsweise Designschwächen von JavaScript oder die unflexiblen Browserplattformen als Bremsklotz der Webentwicklung benennen.[1] Oder man könnte die organisatorische Seite betrachten und die vielen beteiligten Organisationen benennen, die für Standardisierungsprozesse im Internet zuständig sind.[2] Oder aber man nimmt die historische Entwicklung in Augenschein, die dazu geführt hat, dass die technische Entwicklung der rasanten Evolution im Internet nicht Schritt gehalten hat. Die Zeiten statischer HTML/CSS Seiten auf einfachen Desktopgeräten ist mithin noch gar nicht so lange her.

Dennoch tragen alle diese Facetten dazu bei, dass die Entwicklung und Wartung von komplexen Webapplikationen, wie sie heute Standard sind, mit enormen Zeit- und Geldaufwand verbunden sind. Die Anzahl der Frameworks, Werkzeuge und Bibliotheken mit Javascript als Zielsprache ist nahezu unüberblickbar und wandelt sich in einer Geschwindigkeit, die vielen Entwicklern Schwierigkeiten bereitet.<sup>1</sup> Auch die Konsumenten und Nutzern der Webdiensten bekommen diese Probleme zumindest teilweise spüren. Einerseits spürbar in Form von "Downtimes" der Services. Andererseits subtil wie etwa durch lange Ladezeiten wegen aufgeblähtem Quellcode. Frederic Filloux zeigte in einem Blogpost anschaulich, dass sich in einem Zeitungsartikel der britischen Zeitung "The Guardian" zu jedem lesbaren Buchstaben des Artikels über 100 Zeichen Code addieren.[3]

Aus diesen Grund wurde schon 2013 das "Extensible Web Manifesto" proklamiert. Darin wird, kurz gefasst, eine Öffnung der Webplattformen avisiert, um Webprogrammierung teilweise von Browserstandards zu entkoppeln.[^fn2] Drei Jahre später sind diese Ziele in W3C Spezifikationen konkretisiert worden. Einige dieser neuen Standards werden bereits nativ in (einigen) Browsern unterstützt.

## ZIELSETZUNG

Viele populäre Frameworks für den Browser als Zielplattform weisen heutzutage eine monolithische Architektur auf. Der Grund dafür liegt unter anderem darin, dass es bisher nicht möglich war einzelne Browserelemente in ihrem Aussehen und Verhalten zu isolieren. Mit dem neuen W3C Spezifikation "Web Components", die mehrere Substandards unter sich vereint, Ziel dieser Arbeit soll ein In jüngster Zeit haben einige Vorschläge für eine "low-level API" Ziel dieser Arbeit ist die systematische Erfassung der neuen Technologien sowie eine praktische Erläuterung möglicher Architekturmodelle.

<sup>1</sup>Ein fiktives Gespräch mit einem JavaScript Neuling <https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f>

houdini <https://www.w3.org/standards/techs/components>

## VORGEHENSWEISE

Den Anfang dieser Arbeit soll eine Analyse der aktuellen Situation der Frontend Entwicklung aufzeigen. In ihr sollen aktuelle Architekturmodelle analysiert werden. Außerdem soll dieser Teil vor Augen führen, warum es bisher nur mit zusätzlichen Schichten der Abstraktion möglich war einen modularen Aufbau von Web Apps zu ermöglichen. In diesem Teil sollen auch die Probleme die der Entwicklung und Wartung dieser Systeme aufzeigen.

Im nächsten Abschnitt der Arbeit sollen die

<sup>2</sup>: <https://extensiblewebmanifesto.org/>

title: Microservices im Browser durch Web Components  
 subtitle: Kombinierte W3C Spezifikationen für plattformübergreifende Browsermodule  
 author: Jan Peteler, FHWS Würzburg-Schweinfurt, [jan.peteler@student.fhws.de](mailto:jan.peteler@student.fhws.de)  
 date: November 2016  
 abstract: ...

- [1] Y. Katz, "Extend the web forward," 2013.
- [2] P. Walton, "Houdini: Maybe the most exciting development in css you've never heard of," 2016.
- [3] F. Filloux, "Bloated html, the best and the worse," 2016.

<sup>2</sup>Ein fiktives Gespräch mit einem JavaScript Neuling <https://hackernoon.com/how-it-feels-to-learn-javascript-in-2016-d3a717dd577f>