

Browsernative Microservices

Jan Peteler, FH Würzburg-Schweinfurt, jan.peteler@student.fhws.de

AUSGANGSSITUATION

en der Browserplattform erläutern.[1] Oder man könnte die organisatorische Seite betrachten und die Schwerfälligkeit von Standardisierungsprozessen anprangern.[2] Oder aber man nimmt die historische Entwicklung in Augenschein. Die Zeiten statischer HTML/CSS Seiten auf Desktopgeräten ist mithin noch nicht lange her.

cript als Zielsprache betrachtet, die vielen Entwicklern Schwierigkeiten bereitet.[3]

nftige Unsicherheiten abzuwägen.[4, S. 1–3] Die Unsicherheit, die Frameworks immer anhaftet, ist die Lebensdauer derselben.

Mit *Web Components* ist der Entwickler in der Lage, eigene HTML Elemente zu definieren und diese per Templates zu ex- & importieren. Darüber hinaus können sie JavaScript und CSS Funktionalität enthalten und diese vom Rest der Webseite abzukapseln. Ziel dieser Arbeit ist die systematische Erfassung und Risikoabwägung der neuen Technologien sowie eine praktische Erläuterung möglicher Architekturmodelle. Bisher gibt es keine einheitliche Best-practice, wie denn die Komponenten umzusetzen sind, obwohl diese Technologien bereits länger durch Polyfills genutzt werden können. Manch ein Entwickler fürchtet schon eine Flut von schlecht gebauten Komponenten, die wiederum neue Probleme schaffen könnten.[5]

VORGEHENSWEISE

Den Anfang dieser Arbeit soll eine Analyse der aktuellen Situation der Frontend Entwicklung aufzeigen. Dort soll auf die Problematik der bisherigen Architekturmodelle und die Probleme mit monolithischen Frameworks eingegangen werden. Außerdem soll dieser Teil vor Augen führen, warum es bisher nur mit zusätzlicher Abstraktionsebenen möglich war einen modularen Aufbau von Web Applikationen zu ermöglichen.

Im nächsten Abschnitt der Arbeit sollen die neuen Standards aufgeschlüsselt, zugänglich gemacht und auf Anwendungsmöglichkeiten untersucht werden. Explorativ sollen Bestpractices offengelegt werden und mögliche Einbettungen ins Gesamtsystem diskutiert werden. Auch die Probleme, wie sie beispielsweise bei alten Browsern auftreten können, sollen hier behandelt werden.

So we are

This is a blockquote

```
'hello world'
```

Javascript baby

so this is the js world

one more last test

1. This is the list

This is a sub

- ## 2. Another list

- [] oohhhhhhhh

Der letzte Abschnitt soll das Thema mit einer Metaebene abrunden, in der auch die neuen CSS Standards *Houdini* als Teil des Gesamtsystems betrachtet werden sollen.¹

- [1] Y. Katz, „Extend the Web Forward“, 2013 [Online]. Verfügbar unter: <http://yehudakatz.com/2013/05/21/extend-the-web-forward/>

- [2] P. Walton, „Houdini: Maybe The Most Exciting Development In CSS You’ve Never Heard Of“, 2016 [Online]. Verfügbar unter: <https://www.smashingmagazine.com/2016/03/houdini-maybe-the-most-exciting-development-in-css-youve-never-hea>

- [3] D. Berner, „Not An Imposter: Fighting Front-End Fatigue“, 2016 [Online]. Verfügbar unter: <https://www.smashingmagazine.com/2016/11/not-an-imposter-fighting-front-end-fatigue/>

- [4] R. Dodson, „The Case for Custom Elements: Part 2 – Dev Channel – Medium“. 2016 [Online]. Verfügbar unter: <https://medium.com/dev-channel/the-case-for-custom-elements-part-2-2efe42ce9133{\#}.gl5nuty2r>. [Zugegriffen: 01-Dez-2016]

- [5] J. Keith, „Web Components“, 2014 [Online]. Verfügbar unter: <https://adactio.com/journal/7431>

¹<https://drafts.css-houdini.org/>