

INTERVIEW

„Die Lösung muss zum Team passen“

Welche Webtechnologie hat Zukunft? Wie setzt man ein Projekt heute um? Und was ist eine verteilte Webanwendung? Antworten darauf gibt Dr. Florian Rappl im Interview.

Entwickler stürzen sich in der Regel kopfüber in neue Technologien. Alles Neue ist gut und das Alte ein Fall für die Müllhalde. Diesen Eindruck könnte man gewinnen, wenn man sich die aktuelle Szene bei der Webentwicklung ansieht. Laufend kommen neue Frameworks und Technologie-Stacks heraus, werden gierig aufgegriffen, um dann wieder in der Versenkung zu verschwinden.

Ein bisschen anders sieht das Dr. Florian Rappl. Er ist Solution Architect für IoT- und verteilte Webanwendungen, Microsoft MVP und Track Chair auf der Developer Week für den Bereich Web Frontend.

Im Interview gibt er Tipps, wie man mit der schnelllebigen Technologie im Web-Bereich umgehen sollte.

Wenn man sich deine Website so ansieht, gewinnt man den Eindruck, dass du sehr vielseitig interessiert bist. Welche Hobbys hast du denn?

Florian Rappl: Früher habe ich gerne Sport gemacht – vor allen Dingen Mountainbiking und Laufen. Mittlerweile widme ich meine Freizeit eigentlich nahezu vollständig meinen Töchtern.

Beruflich beschäftigst du dich mit verteilten Webanwendungen. Die Frage hört sich etwas komisch an, aber was genau ist darunter zu verstehen? Eine Webanwendung wird doch in der Regel von einem Webserver aus bedient.

Rappl: Klassischerweise ist das sicherlich so – dem sogenannten Monolithen. Um bestimmte Bedingungen (beispielsweise Ent-

wicklung in mehreren Teams, dynamische Versionierung oder Skalierung von Features, gezielten Rollbacks etc.) erfolgreich zu erfüllen, müssen diese jedoch sehr gut umgesetzt werden – nicht nur, was den reinen Code, sondern auch, was den Aufbau der Entwicklungsprozesse betrifft. In der Praxis ist das kaum zu erreichen. Einen einfachen Weg bieten hier alternative Muster – angeführt von der aktuell sehr populären Microservice-Architektur. Bei verteilten Webapplikationen werden die Daten daher nie von nur einem System ausgespielt, sondern immer von mehreren aggregiert, zusammengeführt oder ausgewertet. Die größten Schwierigkeiten solcher Systeme liegen daher in deren Ausfallsicherheit, der geforderten Datenkonsistenz und dem erforderlichen Wartungsaufwand.

Wer die Webszene verfolgt, stellt fest, dass ein Trend dem anderen folgt. Angular, React, Single Page Applications, Progressive Web Apps und viele mehr. Was würdest du jemandem raten, der ein Projekt beginnen will?

Rappl: Das Wichtigste vorab: Es gibt kein Allheilmittel. Ein Vue-Entwickler wird immer erzählen, wie einfach Vue ist, und für React-Entwickler gibt es sowieso nichts Besseres als React. Microfrontends sind auch nicht besser als Monolithen für klassische Single Page Applications. Von daher wäre mein Ratschlag: Keep it simple. Alle Wege führen nach Rom – sicherlich mancher einfacher als andere. Die Lösung muss je-

doch für das umsetzende Team passen. Braucht man wirklich eine Single Page Application? Warum muss es ein ganzes Framework sein? Ist eine Aus-

● Dr. Florian Rappl

Florian Rappl arbeitet als Solution Architect für IoT- und verteilte Webanwendungen. Sein Hauptinteresse liegt in der Schaffung innovativer Architekturen, die in Entwicklung und Nutzung ska-



lierbar sind. Dr. Rappl forscht auch an Benutzerinteraktionsmodellen, die die Mensch-Maschine-Interaktion verbessern. Er hat mehrere Preise gewonnen und ist Microsoft MVP im Bereich der Entwicklungstechnologien.

lieferung als Progressive Web App wirklich sinnvoll? Ich persönlich fange daher gerne mit den allgemeinsten Technologien an. Bei mir ist zum Beispiel TypeScript gesetzt, da es meine Produktivität in jedem JavaScript-basierten Web Framework erhöht.

Welche Technologien haben in dem Bereich voraussichtlich eine Zukunft?

Rappl: Ich setze nach wie vor stark auf TypeScript und React im Frontend sowie Node.js und .NET Core im Backend. Wenn wir beim Frontend bleiben, wird in naher Zukunft Svelte aufgrund der interessanten Herangehensweise dazustoßen. Progressive Web Apps werden uns sicherlich auch noch länger beschäftigen, obwohl ich hier noch Luft nach oben sehe. Außerdem vermute ich, dass im Bereich der Frontend-Architektur Microfrontends in den nächsten drei Jahren deutlich an Schwung gewinnen.

Du bist Microsoft MVP. Nun hat Microsoft bei den Technologien für das Web Frontend außer ASP.NET Web Forms nicht viel anzubieten. Wie siehst du das?

Rappl: Microsoft spielt mit Technologien wie TypeScript schon eine wichtige Rolle im Frontend-Bereich. Auch Tools wie VS Code darf man nicht vergessen – immerhin hat sich der Editor in sehr kurzer Zeit zum De-facto-Standard entwickelt. Ansonsten ist mit Blazor eine spannende neue Technologie am Start, die für so manche Anwendungsfälle geeignet sein könnte. Hier sind etablierte C#- und .NET-Entwickler zunächst die primäre Zielgruppe.

Apropos Projekt Blazor: Was ist das und glaubst du, dass es die Situation für Entwickler verbessern kann, die mit C# entwickeln?

Rappl: Ja und nein. Sicherlich eröffnen sich hier spannende Anwendungsfälle – als Wunderwaffe für alles würde ich Blazor jedoch nicht bezeichnen. Microsoft setzt im Moment stark auf diese neue Technologie und versucht sie auch entsprechend zu verbreiten. Meiner Meinung nach bedeutet dies aber nicht, dass wir Blazor für jeden Anwendungsbereich und ohne Einschränkung benutzen können. Ein aktuelles Problem ist, dass Blazor in mehreren Varianten auftritt: Zum einen kann man eine Blazor-Applikation serverseitig ausrollen, zum anderen könnte sie per WebAssembly nur im Browser laufen. Der spannendste Anwendungsfall wäre meiner Meinung nach jedoch ein hybrider Modus, wie er bereits von Technologien wie beispielsweise Meteor ausprobiert wurde. Aktuell versucht Microsoft Blazor auch als eine Art Xamarin beziehungsweise React-Native-Alternative anzubieten.

Du bist der Track Chair für Web Frontend der Developer Week 20. Nach welchen Kriterien hast du die Sessions ausgewählt?

Rappl: Ich wollte eine Story erzählen – eine Story, die für die Frontend-Entwicklung im Jahr 2020 sinnvoll ist. Ich habe die Sessions nach den Kriterien Relevanz, Kreativität, Anspruch und Umfang bewertet. Dabei wurden thematische Duplikate exklusiv betrachtet. Am Ende haben sich die Sessions mit der höchsten Punktzahl durchgesetzt. Sicherlich hätte ich noch

gerne die ein oder andere Session gesehen – es waren einfach so viele tolle Einreichungen dabei – aber ich bin mit dem Endergebnis doch sehr zufrieden.

Deine Affinität zum Web zeigt auch ein Open-Source-Projekt von dir: AngleSharp. Was ist das und wie kamst du darauf, die Bibliothek zu entwickeln?

Rappl: Die ursprüngliche Idee von AngleSharp war, ein natives HTML-basiertes UI-Framework rauszubringen – ohne Chrome oder sonstige nicht MSIL-basierte Abhängigkeiten.



Recht früh hat sich hier jedoch gezeigt, dass .NET ein tatsächlich HTML5-konformer Parser fehlt, der auch ein dynamisches Document Object Model (DOM) erstellen kann. Am Ende kam AngleSharp dabei heraus. Hier handelt es sich um eine Bibliothek, die, genau wie moderne Browser, aus HTML ein DOM ableiten kann. Hierdurch sind alle Manipulationen und Dokumentinformationen identisch zu modernen Browsern wie Chrome, Edge oder Firefox. Über Plug-ins kann man weitere Funktionalitäten wie die Auswertung von CSS oder JavaScript hinzuschalten.

Ein weitere deiner beruflichen Beschäftigungen gilt dem Internet der Dinge. Wie schätzt du dessen Zustand ein? Hype? Realität? Schon vorbei?

Rappl: Ich denke, der initiale Hype ist schon vorbei, was allerdings gut ist. Mittlerweile sehen wir hier auch immer wieder sinnvolle Lösungen, zum Beispiel im Bereich der Heimautomatisierung. Ansonsten sind vor allem in diesem Bereich Themen wie Security und Scalability sehr relevant. Für mich ist das ein spannendes Feld, in dem man viel lernen und sich so einiges anschauen kann. Was tatsächlich noch spannender ist, ist die industrielle Benutzung – häufig als IIoT (Industrial Internet of Things) bezeichnet. Hier kann in Produktionsketten ein echter Mehrwert erzeugt werden.

Und zum Schluss sag uns bitte noch, was dir spontan zu diesen Begriffen einfällt:

Rappl: TypeScript: Geschwindigkeit und Komfort.
C#: Mächtig und modern.
Monolith: Kontrollierbar.
Microservices: Komplex ...