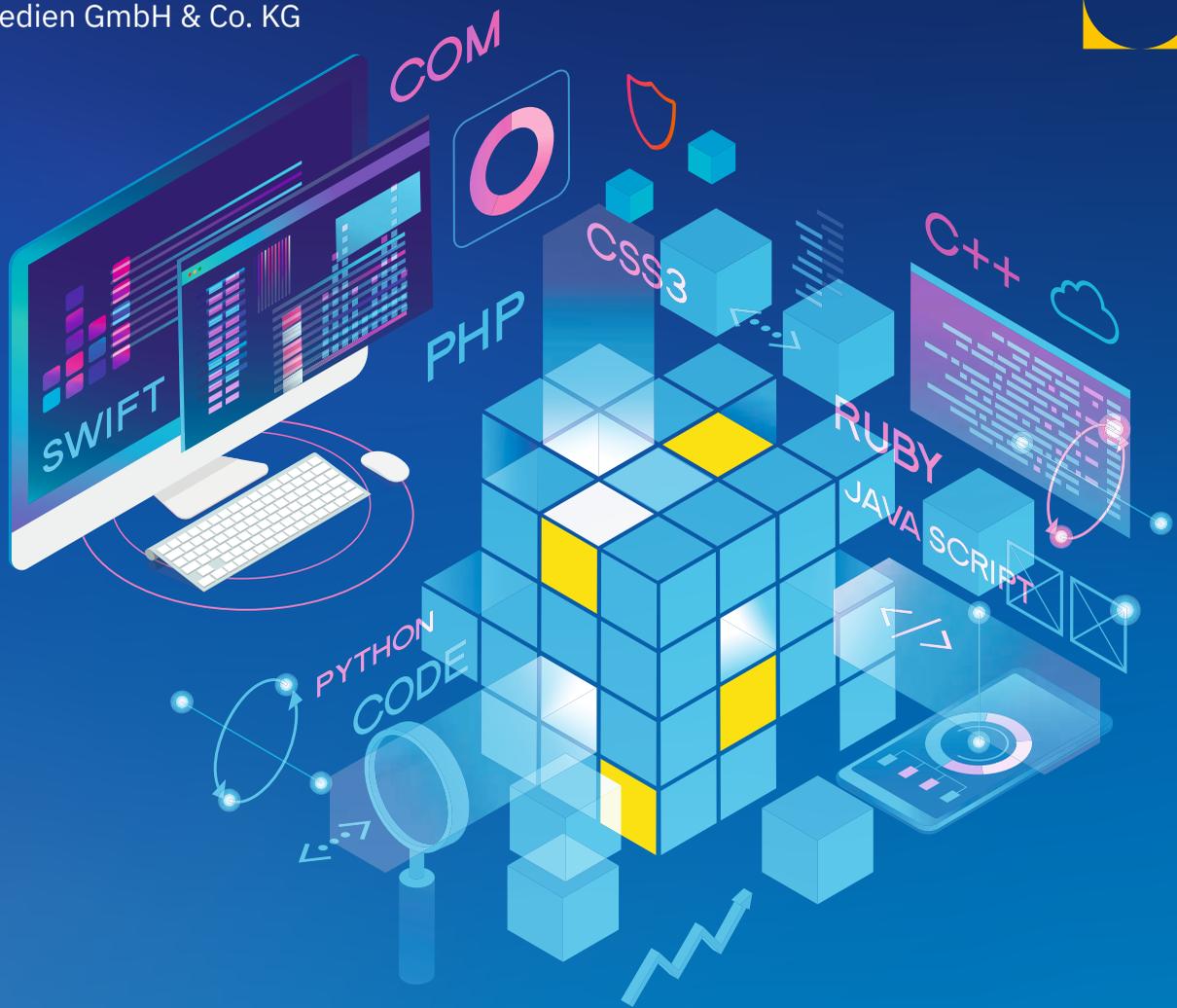


We Are Developers!

Eine Themenbeilage der
Heise Medien GmbH & Co. KG



> JAVASCRIPT

Das Build-Tool Vite.js als flotte Alternative zu webpack

> CLOUD BURSTING

Lastschwankungen mit der hybriden Cloud abfedern

> SYSTEMWECHSEL

Neustart als Developer: Unternehmen locken Quereinsteiger

> HIBERNATE SEARCH > BEHAVIOUR-DRIVEN DEVELOPMENT

Spring-Boot-Anwendungen um eine Volltextsuche erweitern

Ein Praxisbericht zu BDD

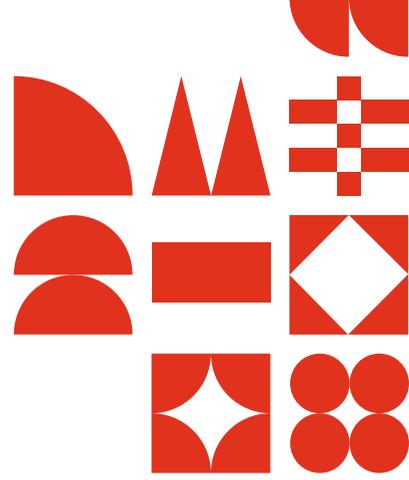


Kommst
root *in*
jeden **Club?**



Sicherheit, IT und Netzwerke
sind deine Passion?
Dann melde dich bei uns:
sipgate.de/techjobs

EDITORIAL



Servus!

Absolventen und Absolventinnen der MINT-Fächer haben zurzeit die Qual der Wahl, offenbar gibt es mehr offene Stellen als fertig ausgebildete Fachkräfte. Das wird noch lange so bleiben, meinen unsere Freunde von der IT-Jobplattform WeAreDevelopers, denn der eigentliche Digitalisierungsschub steht (zumindest für Deutschland) noch aus. Doch wo verorten sich die Young Professionals selbst beim Einstieg in den Beruf? Wir haben nachgehört, und offenkundig ist nicht immer alles Gold, was glänzt. So hat Nils Kasseckert in seinem früheren Team gemischte Erfahrungen mit dem Pair Programming gemacht und schreibt über die Licht- und Schattenseiten des Behaviour-Driven Development. Von seinen Eindrücken können Entscheider in Unternehmen und Berufsanfänger profitieren, denn nicht nur technische, sondern auch kulturelle Herausforderungen entscheiden über den Erfolg von Entwicklungsteams – und ob Mitarbeiter bleiben.

Weitere Autoren der Frühjahrsausgabe unserer Developer-Beilage geben Einblick in ihre teils nebenberuflich betriebenen Projekte. So hat Bernhard Jungwirth aus Wien neben seiner Arbeit bei einer Bundesbehörde eine Webanwendung für das Erstellen von Gesetzessammlungen als Print-on-Demand-Produkt erstellt. In seinem Beitrag führt er vor, wie sich mit überschaubarem Aufwand mit Open-Source-Software wie Hibernate Search und Apache Lucene jedes beliebige Projekt um eine Volltextsuche erweitern lässt. Timo Zander wiederum sieht in dem Build-Framework Vite.js ein besonders flottes Tool mit dem Potenzial für Neuentwicklungen im JavaScript-Ökosystem und möchte es einer breiteren Community schmackhaft machen. Simon Mennig teilt mit uns Fundstücke seiner Masterarbeit zum Softwareengineering in der Cloud: Sein Beitrag zeigt, wie sich bei Lastschwankungen die Ressourcen verteilter Kubernetes-Cluster durch die Technik des Cloud Bursting besser nutzen lassen.

Vielleicht bekommt ja die eine oder der andere jetzt Lust, selbst zur digitalen Feder zu greifen? Dann wäre das Ziel doppelt erreicht. Ich freue mich auf eure Vorschläge an sih@ix.de. Viel Freude beim Lesen und Mitmachen wünscht euch

Silke Hahn



INHALT

4 Build-Tool Vite.js

16 Cloud Bursting

24 BDD – Fluch oder Segen?

30 Neustart als Developer

34 Hibernate Search

Young Professionals schreiben für Young Professionals

Unsere von jetzt an halbjährlich erscheinende Beilage zu c't und iX basiert auf einer Artikelserie von Heise Online, die Young Professionals eine Bühne bietet für erste Fachartikel. Die Autoren und Autorinnen erhalten begleitendes Mentoring beim Schreiben von der Heise-Developer-Redaktion.

Dein erster Fachartikel bei Heise

Unsere Serie soll jungen Menschen auch Mut machen, sich selbst als Autor oder Autorin zu betätigen. Sei es, um Altersgenossen wichtige Erfahrungen mitzuteilen, oder sei es, um ein eigenes Projekt vorzustellen, das einem am Herzen liegt. Oder einfach, weil du schon immer mal einen Fachartikel schreiben wolltest.

Kontakt: developer@heise.de

> Vite.js: Build-Tool mit Tempo

Timo Zander

Vite legt Tempo vor: Das Development- und Build-Tool soll zwanzigmal schneller sein als webpack. Dahinter steht das Team um Vue-Gründer Evan You.



Kaum eine Programmiersprache polarisiert so sehr wie JavaScript. Während es für einige Entwickler die Wunschsprache ist, stellen sich anderen beim bloßen Klang die Nackenhaare auf. Doch dass JavaScript eine ausgesprochen schnelle Entwicklung ermöglicht, beweist Vite – ein neues Projekt des Teams um Evan You, den Gründer des clientseitigen JavaScript-Webframeworks Vue.

Vite (von französisch „vite“ für schnell) [1] soll sich nach der Vorstellung seines Entwicklerteams im JavaScript-Ökosystem vor allem mit Geschwindigkeit und Entwicklungskomfort durchsetzen. Als Development-Server und Build-Tool begleitet Vite JavaScript-Entwickler vom Erstellen des Projekts bis zum Release. Bislang setzten die meisten Entwickler – bewusst oder durch Tools wie die Create React App oder das Vue CLI (Command Line Interface) – auf webpack. Doch je größer der Umfang des Projekts, desto träger wird der webpack-eigene Development-Server. Entwickler warten immer länger, bis vorgenommene Änderungen sichtbar sind. Aus diesem Ärgernis heraus entstand die Idee für Vite. Das Release der stabilen Version 2.0 Mitte Februar 2021 [2] hat die Position von Vite weiter gestärkt.

Probleme von webpack

Webpack ist im Kern ein sogenannter Bundler. Er sammelt alle JavaScript-, CSS- und Bilddateien zusammen und verschmilzt sie zu deutlich weniger Einzelteilen. Liegen beispiels-

weise im Quellcode Hilfsmethoden noch in einer anderen Datei als der API-Aufruf, finden sich im fertigen Build unter Umständen beide in einer einzigen, langen JavaScript-Datei wieder. Der Vorteil davon ist die verringerte Ladezeit für die Endnutzer: Statt Tausender Dateien lädt der Browser nur noch einige wenige. Die Laufzeit des Bundlings ist jedoch abhängig von der Anzahl der Dateien.

Beim Build sind einige Sekunden mehr noch vergleichsweise verkraftbar. Problematisch wird es bei der Entwicklung mit dem Development-Server, denn dieser nutzt ebenfalls das Bundling von webpack. Ändern Entwickler eine einzige Datei, muss webpack ganze Bundles aus Dutzenden Dateien neu generieren. Daher verliert das sogenannte Hot Module Replacement – also das sofortige Austauschen geänderter Komponenten im Browser, ohne dass ein Neuladen nötig wäre – schnell an Schärfe, wenn die Wartezeiten an die zehn Sekunden grenzen. Schon für den Start einer leeren React-Anwendung benötigt Vite nur 430 Millisekunden, während es bei Create React App beziehungsweise webpack mehr als 12 Sekunden dauert (s. Listing 1).

Für JavaScript-Verhältnisse ist webpack ein echter Dinosaurier. Seit der initialen Veröffentlichung im Jahr 2012 konnte es reifen und ist ungeschlagen darin, seltene Randfälle gut zu behandeln. Zum Beispiel sind sogenannte „Deep Imports“ oft die Ursache für auftretende Bugs im GitHub-Repository [3], während webpack in solchen Szenarien selten Probleme hat. Aber auch die Konfigurationsgröße von

Zukunft passiert nicht einfach. Ich schreibe sie.



Seeing beyond



Digitale Innovation bei ZEISS

„Ich bin kein Anzugträger! In ein traditionelles Unternehmen passe ich nicht“, war sich Michael sicher, als ihn auf halber Strecke zur Informatik-Vorlesung die blauen ZEISS Logos von den Werkdächern anstrahlten. Heute, zehn Jahre später, erklärt er die Digitalisierung der Kundeninteraktion. Als Chief Product Owner bei ZEISS. „Heute weiß ich, dass auch für traditionelle Unternehmen, die mit Hardware groß geworden sind, Software einen immer höheren Stellenwert bekommt. Und dass Leute, die anders denken, da auch guttun“, sagt er. Leute die digital und mutig denken. So wie Michael. Coole Cap, grelle Sneaker. Noch immer kein Anzugträger. Muss er auch nicht sein.

Erfahre mehr über seine Geschichte und Jobs im Bereich IT: zeiss.de/karriere

webpack zeigt Flexibilität und zugleich Komplexität [4], daher büßen häufig etwa 90 Prozent der Anwendungsfälle an Geschwindigkeit und Komfort ein.

Alle Zeichen stehen auf Geschwindigkeit

Vite strebt im Kern dasselbe an wie webpack (nämlich die Entwicklung von JavaScript-Projekten mit einem produktiven Build am Ende), allerdings verfolgt es eine andere Philosophie. Entwickler verwenden im Gegensatz zu vielen Endkunden moderne Browser, Vites mitgelieferter Development-Server macht sich das zunutze. So lassen sich ECMAScript-Module direkt per `<script type="module">`-Tag einbinden und jeder zeitgemäße Browser mit dynamic import-Unterstützung kann sie laden, interpretieren und ausführen [5]. Jedes import im JavaScript-Quellcode wandelt der Browser in eine HTTP-Anfrage um, die dann direkt beim Development-Server landet. Über das `--debug`-Flag gibt Vite einen genauen Log aus, der unter anderem auch die Auflösung (`vite:resolve`) von HTTP-Anfragen zeigt (s. Listing 2).

Fragt der Browser jede Datei einzeln an, führt das auch bei steigender Dateianzahl zu reduzierter Geschwindigkeit. Um das zu vermeiden, verfügt Vite über die Fähigkeit, unabhängige Einstiegspunkte zu ermitteln und so Code-Splitting zu praktizieren. Taucht ein JavaScript-Modul beispielsweise auf einer Unterseite auf, ist es auch erst beim Betreten dieser Unterseite erforderlich – nur im Bedarfsfall bearbeitet Vite das Modul und schickt es zum Browser. Das spart vor allem dadurch Zeit, dass die Module on demand von Vite geparkt und kompiliert werden, also nicht wie bei webpack gesammelt vorab.

Ein weiteres Problem ist das Wasserfallszenario: Mit jeder angeforderten Datei kommt im schlimmsten Fall eine Vielzahl weiterer Module hinzu, die der Browser dann anfordern muss. Selbst wenn er diese Anfragen nahezu sofort beantwortet, sorgt vor allem der allgemeine HTTP-Overhead für Zeitverlust (s. Tabelle). Zudem ist die Anzahl laufender Anfragen browserseitig begrenzt. Clevere Optimierungen wie das zuvor erwähnte Code-Splitting können das verhindern, indem sie die Browser-Caching-Funktionalität verwenden

Listing 1: Start einer Vite-App, verglichen mit einer äquivalenten Create-React-Anwendung

```
...
yarn run v1.22.10
$ vite

⚡ Vite dev server running at:

> Network: http://192.168.56.1:3000/
> Network: http://192.168.10.1:3000/
> Network: http://192.168.2.119:3000/
> Local: http://localhost:3000/

ready in 430ms
...

[13:07:00] yarn run v1.22.10
$ react-scripts start
Starting the development server...

[13:07:12] Compiled successfully!

You can now view cra-example in the browser.

Local: http://localhost:3000
On Your Network: http://192.168.56.1:3000

Note that the development build is not optimized.
To create a production build, use yarn build.
...
```

und die Dependencies beim Start des Servers in einer einzigen Datei zusammenfassen. So führt der Browser je Dependency nur noch genau eine Anfrage durch.

Intelligentes Bundling von Dependencies

In der JavaScript-Welt sind ECMAScript-Module nicht die einzige Möglichkeit, Dateien in einer anderen Datei zu nutzen. Alternativ gibt es Methoden wie das synchrone, auf JavaScript-Backends beschränkte CommonJS (via `require(...)` und `module.exports`) oder die sogenannte Universal Module Definition (UMD) [6], die andere Methoden vereinheitlichen soll und im Browser wie auch im Backend funktioniert. Aufgrund der Standardisierung durch das ECMA-Komitee gelten die ECMAScript-Module jedoch als zukunftsweisend. Um dennoch Kompatibilität für Dependencies jeglicher Art in einem Vite-Projekt sicherzustellen, müssen JavaScript-Entwickler sie anpassen und optimieren. Vite führt die Anpassungen allerdings

ausschließlich bei einer veränderten `package.json`- oder Lock-Datei aus, um Zeit zu sparen.

Optimiert werden somit nicht nur alle Abhängigkeiten, die keine ECMAScript-Module exportieren, sondern auch jene, deren Einstiegspunkte `import`-Anweisungen in tiefere Untermodule oder

Listing 2: HTTP-Anfragen mit Auflösung (`vite:resolve`)

```
vite:resolve 0ms react -> C:/dev/vite-example/node_modules/.vite/react.js?v=f9c662d4&es-interop+800ms
vite:resolve 0ms react-dom -> C:/dev/vite-example/node_modules/.vite/react-dom.js?v=f9c662d4&es-interop+9ms
vite:resolve 1ms ./index.css -> C:/dev/vite-example/src/index.css+3ms
vite:resolve 1ms ./App -> C:/dev/vite-example/src/App.jsx+2ms
vite:resolve 0ms /src/App.jsx -> C:/dev/vite-example/src/App.jsx+1ms
vite:load 7ms [fs] /node_modules/.vite/react.js?v=f9c662d4+782ms
vite:resolve 0ms ./logo.svg -> C:/dev/vite-example/src/logo.svg+117ms
vite:resolve 0ms ./App.css -> C:/dev/vite-example/src/App.css+1ms
```

andere Dependencies aufweisen. Beispielsweise fasst der Optimierungsprozess von Vite alle JavaScript-Dateien von Lodash – eine Utility-Sammlung mit zahlreichen Funktionen, die jeweils in eigenen Untermodulen liegen – für die Entwicklung in eine einzige Datei zusammen. Während des Builds findet korrektes Code-Splitting statt: Nur die tatsächlich genutzten Untermodule von Lodash erscheinen auch im Kompilat.

Bytecodes aus Listing 5

Name	Pfad	Protokoll	Größe	Zeit
localhost	/	http/1.1	145 B	12 ms
client	/@vite/client	http/1.1	17.2 kB	11 ms
main.jsx	/src/main.jsx	http/1.1	1.6 kB	7 ms
react.js?v=2a1b6b4a	/node_modules/.vite/react.js	http/1.1	850 B	9 ms
react-dom.js?v=2a1b6b4a	/node_modules/.vite/react-dom.js	http/1.1	2.4 MB	105 ms
index.css	/src/index.css	http/1.1	1.1 kB	63 ms
App.jsx	/src/App.jsx	http/1.1	6.5 kB	70 ms
localhost	/	ws	0 B	3 ms

Leichtgewichtige Dateioptimierungen

Der Development-Server von Vite liefert die eigenen Projektdateien bei einer Anfrage nicht genauso aus, wie sie auf der Festplatte liegen. Denn es gilt nicht nur, einige Randfälle zu behandeln, die die ECMAScript-Module noch nicht unterstützen, sondern auch eine Möglichkeit zu schaffen, dass Plugins – eingebaut oder aus der Community – die Dateien verändern können. Ein Beispiel für ein solches Plug-in ist der TypeScript-Support: Fordert der Browser eine .ts- oder .tsx-Datei an, wird diese in JavaScript kompiliert und zurück-

gesendet. Bemerkenswert ist, dass Vite esbuild zum Übersetzen von TypeScript in JavaScript nutzt. Dieses Tool ist als einer der schnellsten JavaScript-Bundler und -Transpiler in der Lage, TypeScript zwanzig- bis dreißigmal so schnell zu kompilieren wie der offizielle Compiler tsc – allerdings zulasten einiger exotischerer Sprachfeatures (s. Abb. 1). Diese werden beim Build teils falsch interpretiert, verursachen teils aber auch Fehler. esbuild beschreibt allerdings, wie man die tsconfig-Datei konfigurieren muss, um das zu vermeiden [7]. Zusätzlich liefert Vite in seinen TypeScript-Templates solche eine Konfiguration direkt mit aus.

ADVERTORIAL

Miro continues to grow and is opening a brand new hub in Berlin!

Miro is an online collaborative whiteboard platform that enables distributed teams to work together effectively. Miro is perfect for team brainstorming and mindmapping, facilitating agile ceremonies, running design reviews and sprints, and creating a central visual hub for projects.

With over 16 million users, Miro is trusted by top global companies of all sizes, including 95% of the 2020 Fortune 100.

We are launching our brand new hub in Berlin and are thrilled to become a part of the Berlin tech ecosystem. In Berlin we are going to build our Core Product development - Miro Core Product is a huge stream that can offer the most interesting and exciting tasks for frontend and backend engineers: high loaded frontend, developing non-trivial technology solutions for real-time collaboration scenarios, complex react.js development, and tough architecture challenges to support our product stability and high performance through our hyper-growth.

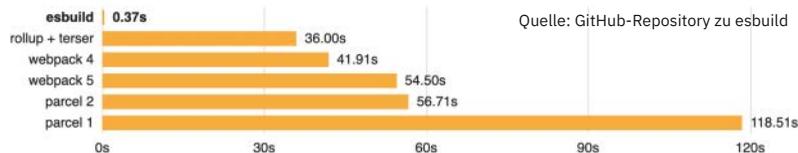
Collaboration is at the heart of what we do and through our work together we hope to create a supportive, welcoming, and innovative environment. We strive to play as a team to win the world and create a better version of ourselves everyday.

If this sounds like something that excites you, we want to hear from you!

Check out our career page (<https://miro.com/careers/>) for all our open roles.

To learn more about us - read our blog - <https://medium.com/@RealtimeBoard>





>> Nach Messungen des esbuild-Autors Evan Wallace ist die Performance dieses JavaScript-Bundlers bis zu 100-mal schneller als die von Alternativen [8] (Abb. 1).

Des Weiteren werden Dependencies in import-Anweisungen von Browsern nicht nativ unterstützt. Denn dort erscheint hinter dem Import kein absoluter oder relativer Dateipfad, sondern lediglich der Name des Pakets. Auch hier greift eine Optimierung, sodass der Pfad für den Browser lesbar wird:

```
- import { helloWorld } from 'dependency-a'
// Die Version entspricht der installierten Paket-Version
+ import { helloWorld } from '/node_modules/  
dependency-a/dist/dependency-a.js?v=1.2.1'
```

Daneben bietet Vite schon jetzt zahlreiche weitere Features wie JSX-Unterstützung (JavaScript XML), die Integration zahlreicher CSS-Präprozessoren wie SASS oder PostCSS und die Möglichkeit von WebAssembly-Modulen. Neben den offiziellen Erweiterungen für Vue oder React hält die stetig wachsende Menge der Plug-ins aus der Community zusätzliche Funktionen bereit [9].

Kompatibel mit Backends aller Art

Auch Webseiten, die auf traditionellen Backends wie Laravel, Ruby on Rails oder Django basieren, nutzen häufig zusätzlich zu den serverseitig gerenderten Seiten dynamische Inhalte mithilfe von JavaScript-Komponenten. Hierbei kann Vite ebenfalls den Entwicklungsprozess verbessern.

Ein eigenes Kapitel der Dokumentation erklärt, wie sich Vite mit wenigen Konfigurationsschritten als Development-Server für das Frontend nutzen lässt [10]. Grundsätzlich lassen sich die Schritte aber als die Konfiguration des Einstiegspunkts und das korrekte Einbinden von Vite und dem JavaScript-Einstiegspunkt im HTML-Template zusammenfassen (s. Listing 3).

In der Entwicklung zeigen alle JavaScript- und CSS-Links auf den laufenden Vite-Server (zum Beispiel localhost:3000), während sie in einem produktiven Umfeld auf die statischen Assets im Server-Verzeichnis verweisen (etwa ./public/assets).

Rasante Vorschaufunktion

Gerade in der JavaScript-Welt ist Hot Module Replacement (HMR) zu einem der meistgeschätzten Features geworden.

Kaum ist eine Änderung gespeichert, schon wird sie binnen weniger Sekunden im Browser sichtbar – im besten Falle. Das Bundling mittels webpack sorgt bei einer Vielzahl von Dateien dafür, dass aus „sofort“ oft mehrere Sekunden werden. Dass das auch anders möglich ist, zeigt Vite. Hier ist die Performance des HMR unabhängig von der Projektgröße.

Beim Optimieren und Aufarbeiten der Dateien erstellt Vite einen schlanken Abhängig-

keitsgraphen. So weiß der Development-Server genau, welche Datei wiederum welche anderen benutzt. Wird nun eine Datei verändert, durchläuft Vite seinen Graph ausgehend von dieser Datei zum Einstiegspunkt hin. Währenddessen prüft es, ob die Import-Metadaten import.meta der Dateien auf dem Weg das Attribut hot besitzen und ob sie dort das Vite-eigene HMR-Interface implementieren. In dem Fall erfolgt die Mitteilung über die Änderung der Datei, wodurch dann der Code des Interface eigenständig zur Ausführung kommt und sich um die Behandlung der Dateiänderung kümmert.

Erreicht Vite beim Traversieren den Einstiegspunkt, ohne ein solches Interface zu finden, lädt es schlicht die komplette Seite im Browser neu. Das ist aber ein seltener Fall, da das import.meta.hot-Interface keine Neuerfindung ist. Im Gegenteil, webpack nutzt mit module.hot eine äquivalente Lösung, mit dem einzigen Unterschied, dass die Webpack-Entwickler nicht auf das für ECMAScript-Module spezifische import.meta gesetzt haben. So ist bei vielen Frameworks die HMR-Unterstützung für Vite bereits vorhanden.

Entwicklung und Build aus einem Guss

Das Bundling für produktive Builds geschieht mit Rollup – einer modularen webpack-Alternative, die in den Augen von Evan You zukunftsfähiger und somit als Unterbau für Vite besser geeignet ist als webpack [11]. Hierdurch unterstützt Vite nicht nur von Haus aus eine Vielzahl existierender Rollup-Plug-ins, sondern kann auch durch die Entscheidung für ein konkret zugrunde liegendes Build-Tool tiefere Integration

Listing 3: Vite und JS-Einstiegspunkt ins HTML-Template einbinden

```
@if (__DEV__)
<!-- Entwicklung -->
<script type="module" src="http://localhost:3000/@vite/client"></script>
<script type="module" src="http://localhost:3000/index.js"></script>
@else
<!-- Produktion -->
<script type="module" src="dist/{{ $manifest['index.js']]['file'] }}"></script>
<link href="dist/{{ $manifest['index.css']]['file'] }}" rel="stylesheet" />
@endif
```

Entkoppelt und dennoch integriert: Microservices für digitale Content-Plattformen

Vom Monolithen zu einem Mix aus standardisierten und anpassbaren Microservices

Digitale Inhalte haben im letzten Jahrzehnt extrem an Bedeutung gewonnen. Was früher Web- oder E-Commerce-Inhalte waren, für Websites, sind heute hochauflösende 4K-Videos, dazugehörige Metadaten, Publikationskanäle, Integrationen und vieles mehr.

Oft wurden Systeme, die diese Inhalte verwalten, als All-in-One-Lösungen entwickelt. Mit den veränderten Konsumgewohnheiten hat sich jedoch auch die Komplexität des Content-Managements verändert. Althergebrachte Alt-systeme erlaubten es Benutzern, Inhalte zu verwalten, anzuzeigen und zu veröffentlichen; allerdings werden nun, mit zunehmender Komplexität, die Rollen aller Beteiligten, zum Beispiel der Content-Redakteure, granularer und verbundener.

Die Entkopplung monolithischer Software in mehrere Services ist nichts Neues, hat aber in den letzten Jahren stark an Zugkraft gewonnen. Die entkoppelte, serviceorientierte oder auf Microservices basierende Architektur eignet sich sehr gut für große Plattformen, bei denen Inhalte mit mehreren eigenen oder fremden Systemen zu integrieren sind.

Aber wie genial Microservices als Konzept auch sein mögen, integrierte Services (ähnlich einem Monolithen) werden noch von vielen bevorzugt. Dies vor allem wegen der einfachen Entwicklung, den einheitlichen Benutzeroberflächen und vor allem der engen Integration der Funktionalitäten. In der realen Welt wäre dies vergleichbar mit dem Kauf eines Laptops in einem Geschäft, anstatt Einzelteile zu kaufen, um diesen selbst zusammenzubauen.

Granular denken

Ein Backend einer modernen Content-Plattform wie Netflix oder Amazon Prime besteht aus vielen Teilen, die jeweils bestimmte Funktionen wie Benutzerverwaltung, Content-Ingestion, Medienverarbeitung, Sicherheit, Suche und mehr ausführen. Diese Teile werden einmal oder wiederholt im gesamten System verwendet. Und fast alle diese Teile sind voneinander entkoppelt. Ein Übergang zur Granularität schafft Vorteile für die Zusammenarbeit mehrerer Entwickler. Ein granulares Setup ist zudem sehr viel besser skalierbar, und lässt sich einfacher erweitern. Langfristig reduzieren Microservices auch die Kosten für Wartung und Betrieb erheblich.

Integrationen, Abhängigkeiten und Aufwand

Bei der Bereitstellung von Inhalten geht es nicht mehr nur um das Übertragen der Inhalte auf die Geräte – mittlerweile geht es um viel mehr als die bloße Veröffentlichung auf einem Frontend. Von der Anmeldung über ein Social-



Media-Konto bis zu Zahlungen über einen präferierten Zahlungsanbieter müssen Streaming-Plattformen zur Personalisierung mit verschiedenen Systemen und Diensten integriert werden.

Ebenso bestehen starke Abhängigkeiten zwischen einigen Teilen eines Systems, da diese wichtige Datenbanken und Korrelationen gemeinsam nutzen, um das System funktionsfähig zu halten. Somit erzeugt die Integration mehrerer Systeme ohne Microservices unwiderrufliche Abhängigkeiten – im Idealfall keine sinnvolle Situation

für Plattformen, die umfangreich gewartet, skaliert oder zukünftig erweitert werden sollen.

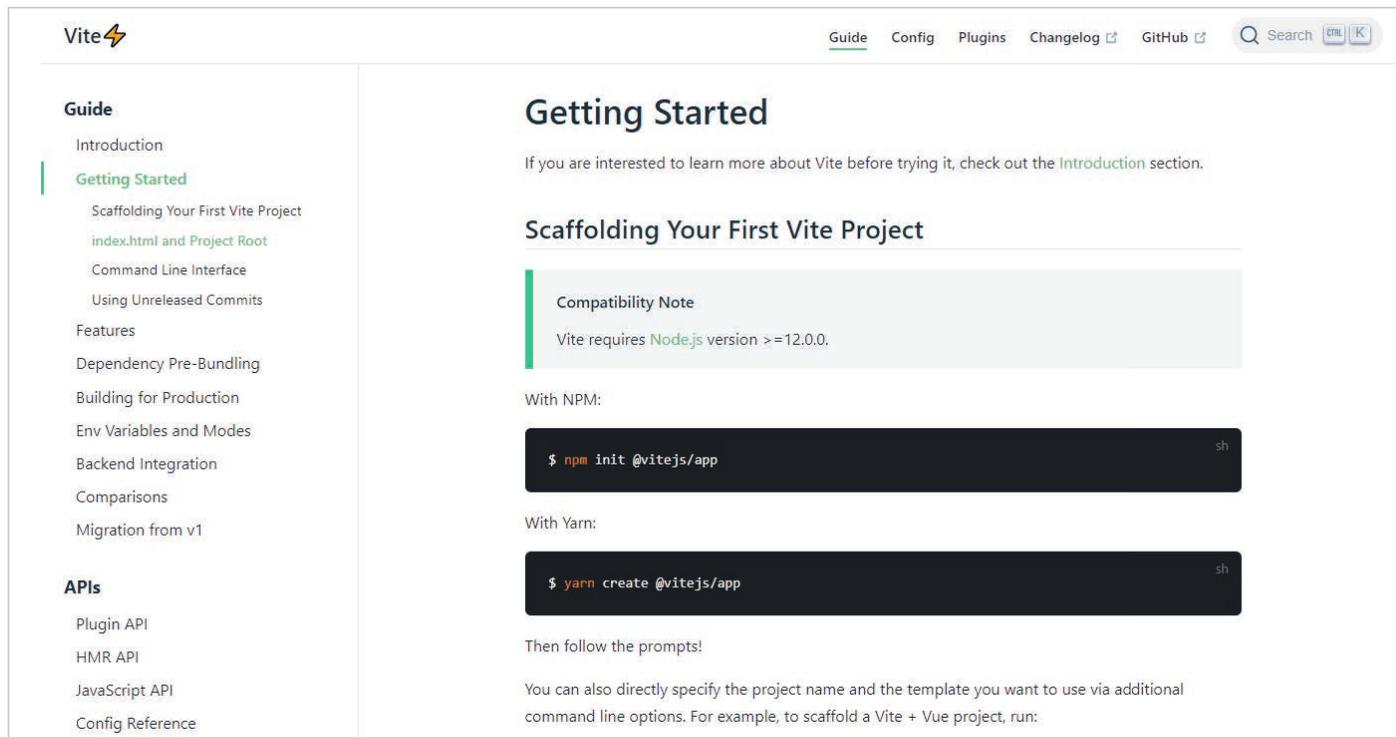
Überbrückung durch eine integrierte und doch entkoppelte Plattform

Was wäre, wenn es eine Möglichkeit gäbe, das Beste aus beiden Welten zu haben? Ähnlich einer Möglichkeit, den Computer zum Zeitpunkt des Kaufs zu konfigurieren und ihn später aufrüsten zu können? Wenn Microservices miteinander integriert wären und dennoch Anpassungen erlauben würden? Wenn eine Möglichkeit für ein standardisiertes UX/UI bestünde? Und eine mögliche kundenspezifische Entwicklung und Integration für neue Microservices möglich wäre? Moderne Software-Tools wie GraphQL, React und Node.js bieten Mittel, um das fehlende Puzzlestück zu ergänzen. Zum Beispiel in Form einer Plattform, die es Unternehmen ermöglicht, einen Mix aus standardisierten und anpassbaren Services zu nutzen. Ein solcher entkoppelter und doch integrierter, auf Microservices basierender Ansatz bietet Vorteile:

1. Gemeinsame einheitliche Libraries und Service-API
2. Agile und kollaborative Entwicklung, kontinuierliche Optimierung
3. Standardisierte, einsatzbereite Dienste
4. An individuelle Arbeitsabläufe anpassbare Dienste
5. Hohe Skalierbarkeit, Flexibilität und Erweiterbarkeit

Unsere jahrzehntelange Erfahrung in Content-Management und -Delivery hat uns wertvolle Einblicke in die Herausforderungen von komplexen Content-Plattformen gegeben. Daher sind wir heute in der Lage, erstklassige, interoperable Lösungen anzubieten, die die Verwaltung, Verarbeitung, und Bereitstellung von Inhalten neu denken.

Axinom verfolgt einen Ansatz, der Services entkoppelt und dennoch standardisiert. So können unsere Partner schnell digitalisieren und neuartige Anwendungen erstellen, die vom Media-Streaming zu Hause bis zum In-Flight-Entertainment im Flieger reichen. Besuchen Sie uns auf www.axinom.com und erfahren Sie, wie Sie schon heute zukunftssicher sein können!



>> **Die Dokumentation von Vite basiert bereits auf VitePress [23] (Abb. 2)**

damit bieten. Die vordefinierte Rollup-Konfiguration von Vite ermöglicht es, dass die meisten Anwendungsfälle ohne weiteres Setup bereits abgedeckt sind. Wer dennoch einen komplexeren Build-Prozess benötigt, kann sämtliche Konfigurationen nach eigenen Anforderungen und Bedürfnissen ändern. Der Hintergrund dabei ist, dass im breit gestreuten JavaScript-Ökosystem oft viel Zeit beim Konfigurieren eines Projekts verloren geht. You möchte das verbessern.

Ein technisch recht ähnliches Tool ist Snowpack, ein ebenfalls auf ECMAScript-Modulen basierender Development-Server mit Build-Funktionalität [12]. Trotz aller Ähnlichkeiten gibt es grundlegende Unterschiede. Durch die Wahlfreiheit bei Snowpack und das Konzept einer möglichst schmalen Standardkonfiguration müssen Entwickler beispielsweise das Bundling beim produktiven Build eigens konfigurieren – für die Mehrheit der Anwender ist das wichtig. Außerdem integriert Vite Rollup als festes Build-Tool tiefer und bietet auf diese Weise mehr Features.

Vielfältige Build-Features

Der Build-Prozess ist nicht bloß eine reine Rollup-Integration; vielmehr umfasst Vite einige weitere Features, die den Build noch komfortabler machen [13]. CSS-Code wird beispielsweise – analog zum JavaScript-Code – in Chunks aufgeteilt,

die Code-Splitting automatisch ermöglichen. Zusammen mit den von Vite generierten `link rel="modulepreload" />`-Links erhöht das die Geschwindigkeit der produktiven Webanwendung merklich. Des Weiteren optimiert der Build-Prozess Abhängigkeitsketten asynchron geladenen JavaScript-Codes und verhindert damit kaskadierende Netzwerkanfragen.

Zudem inkludiert Vite automatisch einen Polyfill für die dynamischen ECMAScript-Module-Importe, um so die Kompatibilität älterer Browser zu gewährleisten. Weitere Polyfills nutzt es allerdings nicht, Nutzer müssen sie eigenständig bei Bedarf einbetten. Neben dem Erstellen herkömmlicher Webseiten bietet der Build auch Unterstützung für Webseiten mit mehreren Einstiegspunkten und Bibliotheken. Standardfunktionen wie Umgebungsvariablen sind ebenfalls vorhanden.

Verfrühtes Release von Vite 1

Im Dezember 2020 hatte You im Repository von Vite eine Erklärung dazu veröffentlicht, welche Fehler aus seiner Sicht bei der Entwicklung von Vite 1 passiert sind und welche Pläne er für die Zukunft von Vite hat [14]. So war die Plug-in-Architektur weitaus weniger modular als geplant, einige Vue-Spezifika waren fest verbaut (statt modular) und bei zahlreichen Schnittstellen fehlte die Trennschärfe. Er schlug mehrere Breaking Changes vor, die allerdings mit dem Ökosystem kol-



Wahre
HELDEN
GESUCHT

Wir suchen:

IT-SYSTEMADMINISTRATOR (m/w/d) | Vollzeit

willy.tel sucht wahre Helden, die uns bei unserer Mission unterstützen,
Hamburg zu verbinden: mit dem Netz der Zukunft.



Helden mit Superkräften scannen
diesen Code oder schauen unter
willytel.de/karriere


TV · INTERNET · TELEFON · MOBILFUNK

lidierten, das sich schon um die erste Version von Vite herum gebildet hatte. Aus diesem Grund markierte er den Stand als „1.0 Release Candidate“, stoppte die Arbeit an Vite 1 und arbeitete mit seinem Team stattdessen an Vite 2 weiter. Auf Twitter verkündete You mittlerweile, dass Vite 1 veraltet sei und Entwickler auf Version 2 setzen sollten [15].

In der Entwicklungslinie 2.x setzt er nun offenbar die von ihm geforderten Änderungen durch. Die Plug-in-API hat das Vite-Team neu entworfen, sodass es einige spezifischere Funktionalitäten entfernen oder auslagern konnte. Die Konfiguration ist mittlerweile deutlicher nach Build und Development-Server getrennt, die Performance gilt als verbessert und verschiedene kleinere Probleme sind behoben. Das Nutzerfeedback seit dem Release der Beta von Vite 2 im Januar 2021 hat ihn offenbar in seiner Zukunftsvision für Vite.js bestärkt. Mit der inzwischen vorliegenden finalen Version ist es auch für jene interessant, die Beta-Versionen meiden.

Bedeutung für die JavaScript-Welt

Gerade durch die vorgenommenen Änderungen in Vite 2 könnte das Tool ein echter Zugewinn für die JavaScript-Welt werden. Schon jetzt, wenige Wochen nach dem ersten Beta-Release, sprießen regelmäßig neue, durch die Community getriebene Plug-ins und Projekte hervor. Neben einer Vite-Integration für Ruby-on-Rails-Projekte [16] gibt es weniger exotische Kontributionen in Form diverser Starter-Templates, Beispielprojekte oder kleinerer Utility-Plug-ins wie der automatischen Bildkomprimierung [17]. Das Wachstum zeigt das Stimmungsbild der Community, die auch in den sozialen Medien positiv antwortet. Angetrieben von der beabsichtigten Framework-Unabhängigkeit wagen sich auch React- und Svelte-Entwickler an Vite. Das zeigen nicht nur einige Issues im GitHub-Repository, sondern auch die Welle an Tutorials und How-tos, die zurzeit das Netz fluten (zum Beispiel auf der Online-Publishing-Plattform Medium [18]).

Nächstes Ziel in Sicht

Der nächste Meilenstein für Vite ist die finale Unterstützung für serverseitiges Rendern. Nach dem initialen Hype um Frontend-Frameworks wie React oder Vue ist die wachsende Popularität des serverseitigen Renderings ein Schritt in Richtung performanterer JavaScript-Webseiten. Gerade clientseitige Frameworks belasten die initiale Ladezeit stark, was unter anderem zu schlechteren Platzierungen in Suchmaschinen führt.

Ende Januar veröffentlichte You mit der Beta 50 eine erste offizielle Implementierung des Features [19]. Zwar existieren aus der Community heraus schon einige Beispiele für server-

seitiges Rendern mit Vite, allerdings strebt You eine integrierte Lösung an, die durch geschickten Einsatz der bestehenden Architektur und Funktionen seinem Kernanliegen – der Geschwindigkeit – treu bleiben soll. Die derzeitige Implementierung ist zwar noch experimentell, sodass die offizielle Dokumentation um Vorsicht bittet und auf mögliche Änderungen hinweist [20]. Allerdings gibt es schon vollfunktionale Beispielprojekte auf der Basis von Vue 3 oder von React [21, 22].

Die Nichttrivialität des Themas macht es zur häufigen Herausforderung, sodass Vite hier das Potenzial hat, eine echte Erleichterung zu werden. Der gesamte Feature-Umfang zielt allerdings nicht bloß auf Komfort und Vereinfachung ab, sondern ermöglicht zusätzlich Anwendungen, die unter webpack so nicht möglich wären. Mit VitePress zeigt das Team beispielsweise eindrucksvoll, wie man Vites „PS“ auf die Straße bringt und liefert Ideen für die Einsatzmöglichkeiten des Tools.

VuePress und VitePress

Im Jahr 2018 hat das Vue-Team die erste stabile Version von VuePress veröffentlicht [23]. Dieses Tool ist eine Kombination aus statischem Seitengenerator (SSG) und Single-Page Application (SPA). Für die Praxis bedeutet das, dass der Build-Prozess alle mit Inhalt gefüllten Seiten in statische HTML-Dateien übersetzt, die dann beim ersten Aufruf mit dem Browser die JavaScript-Logik laden und die Webseite in eine gewöhnliche Single-Page Application verwandeln. Diese basiert dann dem Namen entsprechend auf Vue.js 2, sodass das Tool besonders in der Vue-Community für Blogs oder technische Dokumentationen beliebt ist. Die Hauptvorteile einer solchen zweifaltigen Architektur sind eine bessere Suchmaschinenoptimierung und eine schnelle initiale Ladezeit, ähnlich dem serverseitigen Rendern. Durch den Einsatz der SPA gewinnt man zusätzlich deren Vorteile, nämlich responsive Navigation und die Möglichkeit dynamischer Komponenten.

Da VuePress allerdings ebenfalls auf webpack basiert, plagen es auch die vorgenannten Probleme wie langsames Hot Module Replacement und ein träger Development-Server. Somit erschien es für das Vue-Team wohl naheliegend, mit VitePress eine VuePress-Schwester auf Vite-Basis zu schaffen (s. Abb. 2). Durch die parallele Entwicklung beider Projekte lassen sich Probleme oder fehlende Features, die bei VitePress auffallen, unbürokratisch in Vite verankern, um für zahlreiche weitere Projekte einen Mehrwert zu bieten.

Die Summe der fundamentalen Unterschiede von Vue- und VitePress ist gering: Aus der Version 2 von Vue wird Version 3, Vite ersetzt den webpack-Unterbau, und ein größerer Fokus liegt auf der Performance. Hierdurch werden zusätzliche Verbesserungen möglich wie das Vermeiden des sogenannten „Double Payload“.

Täglich auf dem Programm: Echte Pionierarbeit

Steigen Sie beim Deutschen Zentrum für Luft- und Raumfahrt (DLR) in die IT ein, und gestalten Sie die Zukunft von Luft- und Raumfahrt, Energie, Verkehr, Sicherheit und Digitalisierung mit!

IT beim DLR – woran arbeiten unsere Expertinnen und Experten da eigentlich genau? „Eine“ Antwort gibt es auf diese Frage nicht, denn die IT spielt an jedem unserer 55 Institute und Einrichtungen eine wichtige Rolle. Die ITlerinnen und ITler am DLR programmieren digitale Zwillinge für die Flugzeuginstandhaltung und sorgen für den reibungslosen Betrieb und Test der IT-Bodeninfrastruktur



und der wissenschaftlichen Bodenanlagen in der Raumfahrt. Sie entwickeln Software für eine sichere und effiziente Mobilität der Zukunft, vermitteln Robotern via KI das Wissen über ihre eigenen Fähigkeiten oder unterstützen Forschungsgruppen durch maßgeschneiderte Kommunikationslösungen. Kurz gesagt: immer handelt es sich um echte Pionierarbeit für eine lebenswerte Zukunft.

Als öffentliche Forschungseinrichtung der Bundesrepublik Deutschland meistern wir die Corona-Krise – u.a. mit flächendeckendem Homeoffice – und suchen weiter Verstärkung für die großen Aufgaben unserer Gesellschaft. Bewerbungen können Sie sich wie bisher über das DLR-Karriereportal für alle 30 Standorte. Die Vorstellungsgespräche mit den Instituten und Einrichtungen finden angepasst an die gegebenen Möglichkeiten statt.



Aktuell finden Sie über 100 Stellenausschreibungen für neue IT-Talente unter www.DLR.de/jobs.

Starten Sie Ihre Mission bei uns!

Deutsches Zentrum für Luft- und Raumfahrt (DLR)
Linder Höhe, 51147 Köln
www.DLR.de



Mein Wissen bewegt.

Wir leben in einer Welt im Wandel. In der Fertigung eröffnet uns die Digitalisierung neue Weitsicht.

Meine Antwort für einen effizienteren Faserverbundleichtbau: Mit Virtual-Reality-Anwendungen und digitalen Zwillingen mache ich automatische Produktionsprozesse transparent und Problemstellungen sichtbar.

Beim Deutschen Zentrum für Luft- und Raumfahrt arbeite ich mit Partnern aus der Industrie auf Augenhöhe. Und gestalte mit meinem Wissen die Welt von Morgen.

Sven Torstrick-von der Lieth, Dipl.-Ing. Werkstofftechnik
DLR-Institut für Faserverbundleichtbau und Adaptronik



Was möchten Sie bewegen?
Erfahren Sie hier mehr zur Karriere beim DLR:
DLR.de/jobs/Mein-Wissen-bewegt



Double Payload adé

Das Problem von statisch generierten Seiten mit Single-Page-Application-Funktionalität ist, dass Inhalte, die Nutzer bereits in der statischen HTML-Datei geladen haben, oft durch JavaScript erneut geladen werden. So kommen beim Versand zwischen Server und Nutzer deutlich größere Datenmengen zusammen, was gerade bei schlechter Netzwerkverbindung relevant ist. VitePress bedient sich zur Vermeidung dieses Double Payload eines Features von Vue 3: Statische Elemente kann der Vue-3-Compiler hosten. Das heißt, diese Elemente werden aus der Render-Funktion in Variablen ausgelagert und somit nur einmal erstellt – egal, wie oft die Seite oder Komponente zu rendern ist.

VitePress erkennt im Kompilat die gehosteten Elemente und löscht sie. Es rührt beim Initialisieren der SPA-Funktion im Frontend nur jene Elemente an, die von der Änderung unmittelbar betroffen sind. Der gesamte statische Inhalt, der nicht dynamisch änderbar ist, bleibt unverändert. Hiermit löst VitePress das Problem des Double Payload und erzielt so meistens Dateigrößen von wenigen Kilobyte – die Ladezeit ist somit sehr kurz. Besonders durch die Aufmerksamkeit, die es den vermeintlich kleinen Problemen widmet, könnte VitePress in der Summe deutlich an Fahrt gewinnen und sich von der Konkurrenz abheben.

Flottes Tool mit Potenzial

Entwicklungstools nehmen immer mehr Platz in der Open-Source-Szene ein und dies nicht ohne Grund: Wer regelmäßig JavaScript nutzt, merkt oft erst nach dem Umstieg auf ein flotteres Tool, wie sehr die eigene Arbeit gebremst wurde. Dieses Aha-Erlebnis trägt maßgeblich zur Popularität solcher Projekte bei. Das Potenzial von Vite.js wird gerade durch Projekte wie VitePress deutlich und könnte in näherer Zukunft zu weiteren Neuentwicklungen in der JavaScript-Community führen – ähnlich, wie es auch schon andere bemerkenswerte Technologien wie React, Electron oder TypeScript vorher getan haben. Die Vielseitigkeit macht das Tool für einen Großteil der JavaScript-Entwickler attraktiv.

Quellen

- [1] Vite.js: vitejs.dev/
- [2] Vite.js 2.0 ist erschienen, Meldung bei Heise: heise.de/-5062097
- [3] Deep Imports als Ursache von Bugs: github.com/vitejs/vite/issues
- [4] Konfigurationsvielfalt von webpack: webpack.js.org/configuration/

- [5] JavaScript-Modul dynamic import(): caniuse.com/es6-module-dynamic-import
- [6] Universal Module Definition (UMD): github.com/umdjs/umd
- [7] TypeScript richtig konfigurieren mit esbuild: esbuild.github.io/content-types/#typescript
- [8] Geschwindigkeitsmessungen mit dem JavaScript-Bundler esbuild: github.com/evanw/esbuild
- [9] Offizielle Erweiterungen, unter anderem für Vue und React: vitejs.dev/plugins/#vitejs-plugin-vue
- [10] Vite.js als Development-Server für das Frontend nutzen: vitejs.dev/guide/backend-integration.html
- [11] Rollup.js, eine modulare Alternative zu webpack: rollupjs.org/guide/en/
- [12] Snowpack: vitejs.dev/guide/comparisons.html#snowpack
- [13] Weitere Vite.js-Features für einen komfortablen Build: vitejs.dev/guide/features.html#web-workers
- [14] Vision für Vite.js 2: github.com/vitejs/vite/issues/1207
- [15] Vite.js 1 gilt als veraltet (deprecated), Tweet von Evan You am 23. Januar 2021: twitter.com/youyuxi/status/1352836500329738240
- [16] Vite-Integration für Ruby-on-Rails-Projekte: github.com/ElMassimo/vite_ruby
- [17] Automatische Bildkomprimierung als Utility-Plug-in: github.com/alloc/vite-plugin-compress
- [18] Es gibt zahlreiche Tutorials zu Vite.js, ein Beispiel: medium.com/swlh/vitejs-and-react-eb3f1afc6e6
- [19] Die Beta 50 von Vite.js brachte erste experimentelle Teile der offiziellen Implementierung von Server-Side Rendering (SSR): github.com/vitejs/vite/releases/tag/v2.0.0-beta.50
- [20] Offizielle Dokumentation zum Server-Side Rendering (SSR) in Vite.js: vitejs.dev/guide/ssr.html
- [21] Beispielprojekt auf der Basis von Vue 3: github.com/vitejs/vite/tree/main/packages/playground/ssr-vue
- [22] Beispielprojekt auf der Basis von React: github.com/vitejs/vite/tree/main/packages/playground/ssr-react
- [23] VuePress, zugleich statischer Seitengenerator (SSG) und Single-Page Application (SPA): vuepress.vuejs.org/
- [24] Dokumentation von Vite.js: vitejs.dev/guide/



Timo Zander

ist dualer Student der Angewandten Mathematik und Informatik und arbeitet als App-Entwickler. Er interessiert sich für Open Source, das JavaScript-Universum und aufstrebende Technologien.

Smarte Antwort auf komplexe Fragen der IT-Entwicklung? – InterSystems IRIS!

Die Datenplattform InterSystems IRIS zeigt sich flexibel und unterstützt Developer bei der Entwicklung von innovativen, datengetriebenen Anwendungen. Entscheidend ist dabei die Fähigkeit, Daten in Echtzeit nutzbar zu machen.

Ehrgeizige IT-Projekte verlangen nach intelligenten, zeitsparenden und effizienten Anwendungen. Allein im Jahr 2021 wird die Menge der Daten in Unternehmen laut einer Publikation der International Data Corporation und Seagate Technology¹ voraussichtlich um 42,2 Prozent zunehmen. Allerdings betrachten weltweit 89 Prozent der IT-Leiter ihre bestehenden Datensilos als große Herausforderung für die digitale Transformation ihres Unternehmens, wie Vanson Bourne und MuleSoft² angeben. Kein Wunder also, dass in der modernen IT der schnelle und einfache Zugriff auf alle relevanten Daten über den Erfolg und Misserfolg von IT-Projekten entscheidet.

Zuverlässige Basis für erfolgreiche Entwicklungsarbeit

Die Datenplattform InterSystems IRIS ist darauf ausgelegt, Entwicklern vertikal und horizontal skalierbares Datenmanagement, flexible Interoperabilität und eine offene Analyseumgebung in einer einzigen, hochperformanten Lösung zur Verfügung zu stellen. Zudem lässt sie sich ganz einfach in bestehende Systeme integrieren und ist in der Lage, nahezu alle denkbaren Daten – ganz gleich ob strukturiert oder unstrukturiert - in Echtzeit und mithilfe aller gängigen Programmiersprachen zu verarbeiten.

Einfacher Einstieg ins Machine Learning

InterSystems IRIS erleichtert die Arbeit mit normalisierten und sauberen Daten – sogenannte Healthy Data - und stellt diese für KI- und ML-Projekte bereit. Dadurch werden Entwickler befähigt, innovative Automatisierungsprozesse und Lösungsansätze beispielsweise für Predictive Maintenance ohne großen Aufwand zu implementieren. Das Feature IntegratedML bietet mit einer einfachen, an SQL-Befehle angelehnten Programmierung einen mühelosen Einstieg ins Machine Learning – ideal für Probeläufe von Projekten oder wenn es an eigenen Data Scientists fehlt. So können beispielsweise Datenanomalien in Echtzeit erkannt und intelligente Aktionen wie etwa der vorzeitige Austausch von Verschleißteilen als Antwort auf Echtzeitsignale aus der Fertigung automatisch veranlasst werden.

Datensilos aufbrechen und Entwicklungsvorhaben beschleunigen

Mit dem effizienten Toolset aus Containerisierung, Microservices und API-Management hilft InterSystems IRIS, die vielerorts vorherrschende Komplexität innerhalb der IT-Infrastruktur zu verringern und Entwicklungsprozesse zu vereinfachen und zu beschleunigen. Darüber hinaus lassen sich mithilfe der Datenplattform neue Versionen einer Software besonders einfach bereitstellen. Als hochintegrierte Daten-



drehzscheibe gewährleistet die Datenplattform eine weitreichende Konnektivität zwischen allen Datenquellen und leistet dank freier Skalierbarkeit und durch integrierte Analysefunktionen umfassende Unterstützung bei einer Vielzahl von Datenmodellen. Entwickler eliminieren auf diese Weise vorhandene Datensilos, stellen relevante Informationen übergreifend und zeitnah zur Verfügung und etablieren in Verbindung mit integrierten Analysefunktionen sowie ML- und KI-Algorithmen eine Smart Data Fabric.

Neue Ära für Forschergeist und Experimentierfreude

Digitalisierung, Big Data und smarte Prozessautomatisierung sind keine Zukunftsmusik – sie finden jetzt statt. Genau deshalb kommt es für Entwickler darauf an, dass sie über die geeignete Infrastruktur verfügen. InterSystems IRIS bietet jene und lässt sich leicht unter praxisnahen Bedingungen testen und für den Routineeinsatz optimieren (<https://www.intersystems.com/tryiris>). Der Startschuss für das Rennen um die wegweisenden Anwendungen für morgen ist gefallen. Ready. Set. Code.



InterSystems GmbH
Robert Bosch Str. 16a, 64293 Darmstadt
InterSystems.de
info@intersystems.de



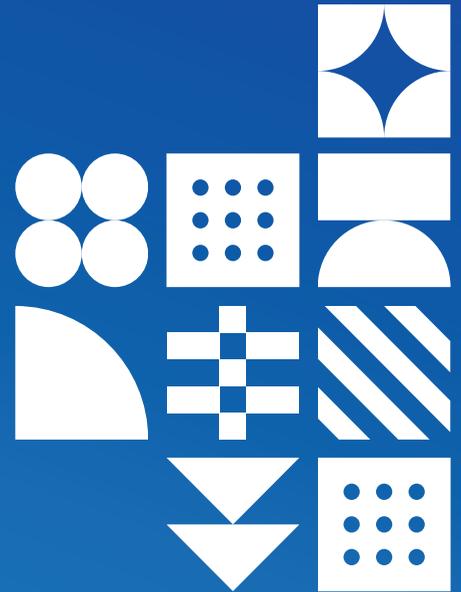
¹ International Data Corporation und Seagate Technology (2020): Rethink Data: Bessere Nutzung von mehr Unternehmensdaten – vom Netzwerkrand bis hin zur Cloud.

² Vanson Bourne und MuleSoft (2020): 2020 Connectivity benchmark report. Insights from 800 IT leaders across the globe

> Wenn die Cloud platzt

Simon Mennig

On Premises betriebene Anwendungen in der Private Cloud lassen sich bei Auslastungsspitzen durch die Rechenkapazität einer Public Cloud entlasten: Wenn die Private Cloud platzt, ist die Public Cloud zur Stelle.



In den vergangenen Jahren hat das Cloudgeschäft ein rasantes Wachstum erlebt. Lag der weltweite Umsatz 2018 noch bei 182,4 Milliarden US-Dollar, wird er für 2022 auf 331,2 Milliarden US-Dollar prognostiziert [1].

Damit steigt auch das Interesse von Unternehmen, die die Cloud für ihre IT-Zwecke nutzen. Jedoch können oder wollen viele Firmen nicht all ihre Daten in eine Public Cloud verlagern (zum Beispiel aufgrund von Compliance-Regeln) und entscheiden sich deshalb für das Modell der hybriden Cloud. Hierbei lassen sich nicht nur die Vorteile mehrerer Technologien vereinen, sondern auch die Unabhängigkeit und Hoheit eines bestimmten Anbieters gewährleisten. Ein weiterer

Vorteil besteht im elastischen Skalieren einzelner Komponenten. Sie bietet die Möglichkeit, die Grundlast auf der eigenen Infrastruktur zu betreiben und aufwendige oder unregelmäßige Lasten in der Public Cloud auszuführen. Aus der Kombination von Public und Private Cloud ergeben sich einige neue Optionen:

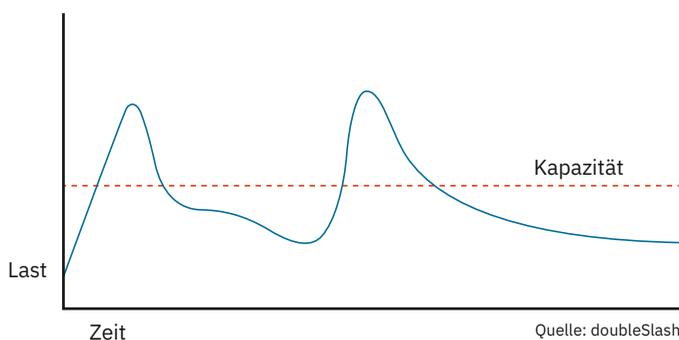
- Die Kontrolle über Speicherung und Verarbeitung von Daten bleibt bei der Organisation.
- Unternehmen können fallweise entscheiden, ob Daten die Private Cloud verlassen dürfen (besonders hinsichtlich personenbezogener und unternehmenskritischer Daten).
- Für Anwendungsfälle mit stark schwankender Last (z. B. bei einem Reiseveranstalter) ist die Elastizität der Public Cloud attraktiv.

Weiterhin ist ein sogenanntes Cloud Bursting denkbar: Dabei lässt sich eine Anwendung On Premises betreiben, aber bei hoher Last erweitert eine Public Cloud die Kapazitäten automatisch.

Was ist eine Hybrid Cloud?

Das NIST (National Institute of Standards and Technology) definiert eine hybride Cloud wie folgt [2]:

„Die Cloud-Infrastruktur [der hybriden Cloud] ist ein Zusammenschluss aus zwei oder mehr unterschiedlichen Cloud-In-



>> Lastschwankungen mit Auslastungsspitzen (Abb. 1)

frastrukturen (privat, gemeinschaftlich oder öffentlich), die grundsätzlich getrennte Einheiten bilden. Allerdings sind sie durch standardisierte oder proprietäre Technologie miteinander verbunden, die die Portabilität von Daten und Anwendungen ermöglicht, wie im Falle von Cloud Bursting für den Lastausgleich zwischen Clouds.“

Meist handelt es sich um die Kombination einer privaten und einer Public Cloud. Kommen mehrere Public Clouds zum Einsatz, spricht man von einer Multicloud. Ein wichtiges Merkmal ist in jedem Fall die Portabilität von Daten und Anwendungen. Der Einsatz von Containern und Kubernetes ermöglicht hierbei einen standardisierten Umgang mit der Anwendung – unabhängig vom Cloud-Anbieter.

Cloud Bursting: Wenn Clouds aus allen Nähten platzen

Der Begriff „Cloud Bursting“ enthält das englische Wort „to burst“, das für zerplatzen und explodieren steht, aber auch „etwas zum Platzen bringen“ heißen kann. Bildlich gesprochen hieße der Ausdruck auf Deutsch etwa „die Cloud sprengen“. Worum genau geht es hier? Im Kern geht es darum, Anwendungen im privaten Teil einer hybriden Cloud bei steigender Last durch die Ressourcen einer Public Cloud zu erweitern: die Private Cloud „platzt“ gewissermaßen und die Public Cloud fängt sie auf. Viele der eingesetzten Technologien sind dabei nicht auf hybride Clouds beschränkt, sondern sie lassen sich auch auf Multicloud-Umgebungen mit mehreren Kubernetes-Clustern übertragen. Damit ist das Thema für alle interessant, die mehrere Cluster parallel betreiben und diese enger miteinander verzahnen wollen, um die verfügbaren Ressourcen besser zu nutzen. Das trifft insbesondere auf Anwendungsfälle mit schwankender Last zu (s. Abb. 1).

Die meiste Zeit über befindet sich die Auslastung unterhalb der Kapazitätsgrenze der privaten Cloudinfrastruktur. Es kommt zu kurzen Spitzen, welche die Kapazität überschreiten. Diese bestandsüberschreitenden Auslastungen soll der öffentliche und kostenpflichtige Teil der hybriden Cloud auffangen. Auf diese Weise lassen sich die benötigten Ressourcen elastisch beanspruchen und die zusätzlichen Kosten für die Nutzung des öffentlichen Teils gering halten.

Fallbeispiel: Entwicklung neuer Medikamente

Cloud Bursting kann beispielsweise bei der optischen Analyse von Wirkstoffen im Rahmen der Entwicklung neuer Medikamente zum Einsatz kommen. Dazu werden Proben mit verschiedenen Wirkstoffkandidaten kombiniert. Anhand von Auf-

nahmen der Proben gilt es, ein neuronales Netz zu trainieren, das bereits in einem frühen Stadium erkennen soll, ob es sich um einen vielversprechenden Wirkstoffkandidaten handelt. Da hierbei von einer schwankenden Auslastung auszugehen ist und teilweise sensible medizinische Daten als Basis dienen, handelt es sich um ein passendes Einsatzgebiet für eine hybride Cloud.

Das Training ist – abhängig von der Menge der Daten und dem Aufbau des Netzwerks – vergleichsweise ressourcenhungrig, damit ist es ein Kandidat für das Cloud Bursting. Ein fertig trainiertes Modell lässt sich zentral im privaten Teil der Cloud speichern. Stößt man das Training eines neuronalen Netzes an, beginnt in der hybriden Cloud die dafür notwendige Arbeit. Abhängig von der Ressourcenauslastung läuft der Job entweder im privaten Kubernetes-Cluster oder im öffentlichen Teil der Cloud. Die Übermittlung des trainierten Modells in das zentrale Repository geschieht gesichert. Dabei spielt es keine Rolle, in welchem Teil der hybriden Cloud das Training letztlich stattgefunden hat.

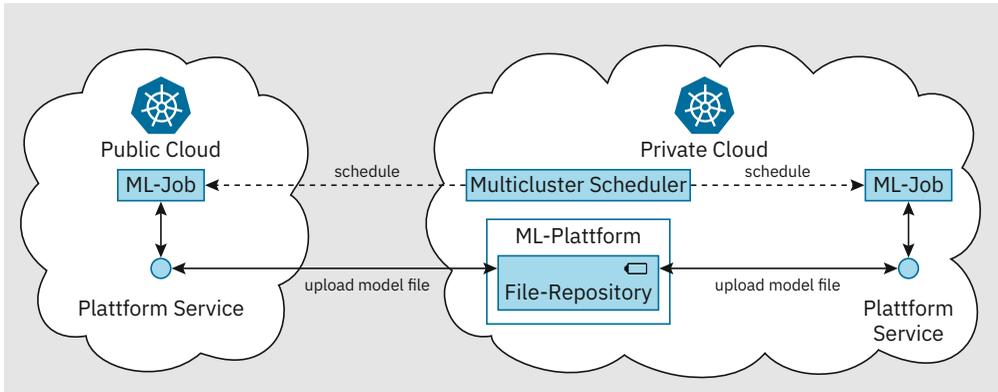
Hands-On: Cloud Bursting mit Kubernetes

Dazu gilt es, eine private und eine Public Cloud aufzusetzen, die jeweils aus einem eigenen Kubernetes-Cluster bestehen [3]. Zusammen bilden die zwei Cluster die hybride Cloud. Damit aus zwei getrennten Umgebungen eine effektive hybride Cloud inklusive Cloud Bursting wird, sollten folgende Anforderungen erfüllt sein:

1. Services in einem Cluster müssen Services im anderen Cluster entdecken und aufrufen können.
2. Die Kommunikation zwischen den Clustern sollte abgesichert sein.
3. Die Freigabe von Anwendungen in der hybriden Cloud lässt sich zentral verwalten.
4. Die Ressourcen beider Cluster sind gebündelt.

Ziel ist es, dass sich die beiden Teile der hybriden Cloud möglichst wie ein einziges, homogenes System verhalten. Unter den Begriffen Cross-Cluster-Kommunikation, Multicloud-Verwaltung und Multicloud-Scheduling lassen sich diese Vorgänge zusammenfassen. Kubernetes selbst bietet hierfür keine Lösung an. Glücklicherweise bietet das Kubernetes-Ökosystem aber für jeden dieser Punkte ein Kubernetes-Projekt, das die jeweilige Funktion zur Verfügung stellt.

Die aufgeführte Beispielanwendung soll das Verwalten und das Scheduling von Multiclouds sowie die Kommunikation über Clustergrenzen hinweg (Cross-Cluster) übernehmen:



Quelle: doubleSlash

kann die Anzahl seiner Nodes dynamisch skalieren.

Anwendungen in Kubernetes lassen sich über einen internen DNS-Server eindeutig identifizieren und aufrufen. Hierfür zuständig ist die Kubernetes-Service-Ressource. Ziel der Cross-Cluster-Kommunikation ist es, diese Funktion auf mehrere Cluster auszuweiten. Eine Möglichkeit ist, ein Service Mesh einzusetzen, das alle Cluster der hybriden Cloud umspannt (s.

>> **Systemüberblick über das Istio-Multicloud-Service-Mesh (Abb. 2)**

- ML-Training in Python (Keras plus TensorFlow) anhand des MNIST-Datensatzes, bei dem es darum geht, handgeschriebene Ziffern zu erkennen [4]
- Ausführen der ML-Jobs dort, wo Kapazitäten frei sind (Cloud Bursting)
- Ablegen der trainierten Modelle in einem zentralen Repository – das benötigt Kommunikation über die Cluster Grenzen hinweg

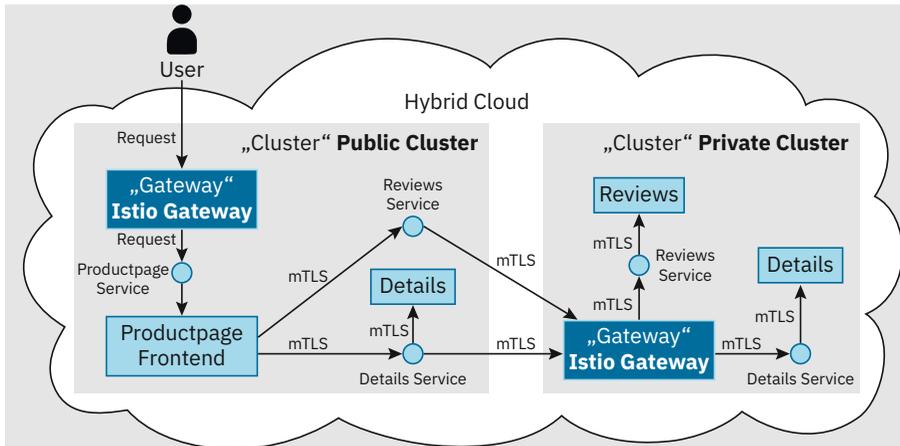
Abb. 2) – für den Use Case kommt Istio als Multicloud-Installation in der Variante als Shared Control Plane zum Einsatz [5]. Das hat den Vorteil, dass keine VPN-Verbindung zwischen den Clustern erforderlich ist. Die Kommunikation zwischen den Clustern geschieht jeweils über ein Gateway und ist über mutual TLS (mTLS) gesichert. In dieser Variante teilen sich beide Cluster die Istio Control Plane. Sie muss folglich immer aktiv sein und wird deshalb typischerweise in der privaten Cloud angesiedelt. Um Probleme mit dem Multicloud-Scheduler zu vermeiden, sollte man Prometheus bei der Installation von Istio deaktivieren. Im Anschluss an die Installation von Istio gilt es, die Sidecar Injection auf Namespace-Ebene zu aktivieren. Dazu labelt man den entsprechenden Namespace in jedem Cluster folgendermaßen:

```
$ kubectl label namespace sample istioinjection= enabled
```

Die Cluster-Umgebung aufsetzen

Als Basis für die Umgebung dienen zwei Kubernetes-Cluster, die beide über Azure Kubernetes Service (AKS) aufgesetzt sind. Grundsätzlich lassen sich die Cluster aber auch selbst betreiben oder zum Beispiel bei Amazon Web Services und in der Google Cloud aufsetzen. Wichtig für ein funktionierendes Cloud Bursting ist die Konfiguration der Ressourcen pro Cluster. Für den privaten Teil wird dem Cluster eine feste Anzahl an Nodes zugewiesen. Der öffentliche Cluster hingegen

Damit eine Anwendung aus beiden Clustern ansprechbar ist, auch wenn sie nur in einem Cluster existiert, muss in jedem Cluster der zugehörige Service eingerichtet sein. Abbildung 3 ist an die Istio-Beispielanwendung Bookinfo angelehnt [6] und zeigt die Kommunikation zwischen verschiedenen Microservices, die sich über zwei Cluster verteilen. Sie besteht aus einem Frontend und zwei Backend-Services, die Details zu Büchern und Reviews bereitstellen. Der Microservice „Details“ ist in beiden Clustern vorhanden. Anfragen lassen sich so zwischen beiden Instanzen verteilen. Der Microservice „Reviews“ ist nur im privaten Cluster ausgerollt. Da die Service-Definition aber auch im öffentlichen Cluster verfügbar ist, leitet das Istio-Gateway alle Anfragen automatisch vom öffentlichen zum privaten Cluster weiter (s. Abb. 3).



Quelle: doubleSlash

>> **Kommunikation zwischen Microservices (Abb. 3)**

Multicluster-Verwaltung

Wie die Einrichtung von Istio vielleicht schon erahnen lässt, fallen viele Aufgaben bei einer hybriden Cloud mehrfach an, da sie in jedem Cluster auszuführen sind. Ziel ist es, das Anlegen von Ressourcen zu vereinheitlichen, um zum Beispiel Namespaces, Services oder Deployments zentral zu verwalten. Bei der Verwaltung mehrerer Cluster ist es dann möglich, solche redundante Aufgaben einzusparen. Das entsprechende Kubernetes-Projekt heißt Kubernetes Cluster Federation [7]. Es ist für den Einsatz von Cloud Bursting nicht zwingend notwendig, erleichtert Entwicklern aber die Arbeit. Der User Guide unterstützt bei der Installation [8]. Nach dem Installieren gibt es einen Host-Cluster, dem beliebige Cluster beitreten können. Sinnvollerweise wird der private Cluster zum Host. Im Host-Cluster können Entwickler zentral Ressourcen anlegen und dann auf ausgewählte Cluster verteilen. Zu den gebräuchlichsten Kubernetes-Objekten existiert jeweils eine Variante mit dem Präfix „Federated“.

Aus einem Service wie dem `blob-storage` wird dann ein `FederatedService`, wie die folgenden Listings veranschaulichen (s. Listing 1 und 2). Neben dem geänderten Namen ist vor allem das Attribut `placement` entscheidend. Durch die Auswahl

von Labels oder die Auflistung von Clusternamen lässt sich hier die Verteilung auf die Cluster bestimmen. Standardmäßig legt man die Ressource in allen Clustern an. Wird der `FederatedService` nun also im Host-Cluster angelegt, resultiert daraus ein entsprechender Service im privaten und öffentlichen Cluster.

Zusammen mit einem Istio-Multicluster-Mesh lässt sich über die Federation die Verteilung von Services vereinfachen. Das gilt insbesondere für das Betreiben von mehr als zwei Clustern, um beispielsweise verschiedene geografische Regionen abzudecken.

Multicluster-Scheduling

Da nun die Kommunikation innerhalb der hybriden Cloud sichergestellt ist, fehlt für das Cloud Bursting noch die Unterstützung eines übergreifenden Schedulers. Dieser entscheidet, in welchem Cluster das Machine Learning letztlich zur Ausführung kommt. Die Funktion steht über den Multicluster-Scheduler von `admiralty.io` bereit [9].

Nach der Installation genügt es, die Annotation `multicluster.admiralty.io/elect: ""` zu einem Deployment, Job oder Pod hinzuzufügen. Anschließend übernimmt der Multicluster-Sche-

ADVERTORIAL

Die Entwicklung von digitalen Bürgerdiensten bringt frischen Wind in die Tool-Landschaft

Die Daten sollen laufen – nicht die Bürgerinnen und Bürger. Deshalb werden im Zuge des Onlinezugangsgesetzes (OZG) insgesamt 575 Verwaltungsdienste als Online-Dienste umgesetzt. BAföG, Elterngeld, Geburtsurkunde – können damit bald von der Couch im Wohnzimmer aus beantragt werden.

Dataport setzt eine Vielzahl dieser Online-Dienste für verschiedene Bundesländer und Kommunen um. Dabei hat sich eins gezeigt, die bewährten Plattformen Java und C#.Net für die Back-End-Verfahren müssen durch weitere Technologie-Stacks ergänzt werden.

TypeScript, Node.js, Angular und weitere Webframeworks machen JavaScript zu einem mächtigen (und mittlerweile ausgereiften) Ökosystem – die Qual der Wahl nehmen die Entwickler von Dataport gerne in Kauf. Die Leichtigkeit von NoSQL, konkret MongoDB, spart bei den passenden Problemstellungen deutlich Aufwand und Entwicklungszeiten ein. Für die Entwicklung von mobilen Apps stellt Kotlin keinen Geheimtipp mehr dar – Entwickler lieben es, weil der Quellcode so kompakt ist. Jetpack Compose für Android sowie SwiftUI und Combine für iOS geben der UI-Entwicklung neuen Schub. Künstliche Intelligenz und Data Science bringen Python als neuen Player ins Spiel. Eine Umfrage bei den 400 Entwicklerinnen und Entwicklern bei Dataport ergab, dass viele über umfangreiche individuelle Kenntnisse verfügen, die jetzt zu

einem soliden Standard zusammengeführt werden müssen. Agile Softwareentwicklung hat sich etabliert, neben Scrum darf es bei kleinen Wartungsteams auch Richtung Kanban gehen. Doch immer bleibt der Weg das Ziel. Zurzeit ist das Skalieren von Scrum für große Entwicklungsvorhaben eine neue Herausforderung für Dataport. Scrum of Scrums, LeSS, SAFe oder Nexus, das ist hier die Frage.

Die Online-Dienste für das OZG sind nur ein, allerdings großer Baustein, mit dem die Digitalisierung in der öffentlichen Verwaltung vorangetrieben wird. Das Aufgabenspektrum reicht von maßgeschneiderten IT-Lösungen für das Bildungswesen, über hochsichere Infrastrukturen für den IT-Betrieb bis hin zu Online-Diensten. Dataport ist das größte staatliche IT-Unternehmen und ein attraktiver Arbeitgeber, der seinen Mitarbeiterinnen und Mitarbeitern ein vielseitiges Aufgabenspektrum und viele Karrieremöglichkeiten bietet. Machen Sie nicht irgendeinen Job – starten Sie jetzt Ihre Karriere bei Dataport:

www.dataport.de/karriere



Dataport AöR
Billstraße 82
20539 Hamburg
040-42846-0

Listing 1: Service blob-storage

```
apiVersion: v1
kind: Service
metadata:
  name: blob-storage
spec:
  ports:
    - port: 10000
      protocol: TCP
      targetPort: 10000
  selector:
    app: blob-storage
  sessionAffinity: None
  type: ClusterIP
```

Listing 2: FederatedService blob-storage

```
apiVersion: types.kubefed.io/v1beta1
kind: FederatedService
metadata:
  name: blob-storage
spec:
  placement:
    clusterSelector: {}
  template:
    spec:
      ports:
        - port: 10000
          protocol: TCP
          targetPort: 10000
      selector:
        app: blob-storage
      sessionAffinity: None
      type: ClusterIP
```

Listing 3: Batch Job für das Multicenter-Scheduling

```
apiVersion: batch/v1
kind: Job
metadata:
  labels:
    app: mnist-ml
    name: ml-job
spec:
  parallelism: 1
  template:
    metadata:
      annotations:
        multicenter.admiralty.io/elect: ""
    labels:
      app: mnist-ml
    spec:
      containers:
        image: smennig/ml-job:mnist
        imagePullPolicy: Always
        name: ml-job
        restartPolicy: Never
        dnsPolicy: ClusterFirst
```

duler und entscheidet, in welchem Cluster ein Pod zum Ausführen des Containers entsteht. Listing 3 zeigt, wie das geht.

Aus technischer Sicht erstellt man einen so annotierten Pod wie gewohnt und lässt ihn anschließend vom Multicenter-Scheduler durch einen sogenannten Proxy Pod ersetzen, der nichts tut. Der Scheduler wählt dann einen Cluster zur Ausführung aus und erstellt darin einen sogenannten Delegate Pod. In ihm kommt die eigentliche Anwendung zur Ausführung.

Deployment der Beispielanwendung

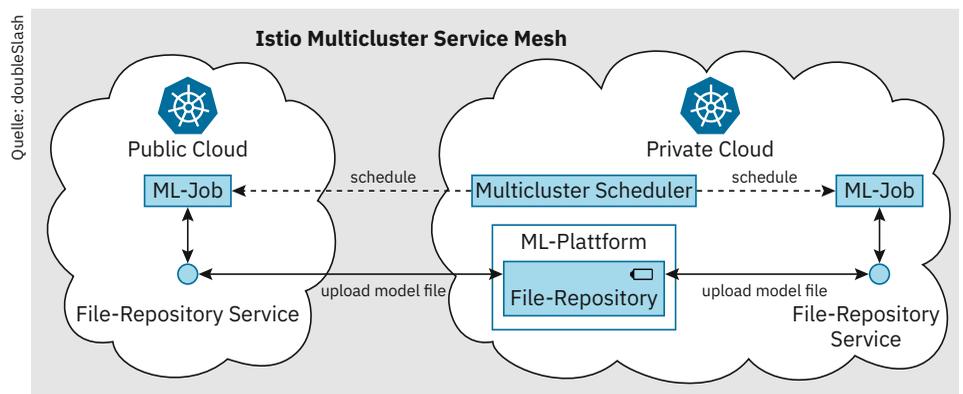
Die Beispielanwendung ist im GitHub-Repository des Autors namens „Hybrid Cloud Burst“ verfügbar. Der Aufbau lässt sich dem Übersichtsbild entnehmen (s. Abb. 4). Im Cluster der privaten Cloud befindet sich ein File Repository für die Ablage von trainierten Modellen. Über den zugehörigen Service ist es aus beiden Clustern erreichbar.

Mit den folgenden drei Befehlen legt man zuerst einen Namespace an, in dem Istio und der Scheduler aktiviert sind (s. Listing 4). Anschließend gilt es, das Deployment und den Service für das File-Repository anzulegen. Da es sich jeweils um Kubernetes-Federation-Ressourcen handelt, lassen sie sich automatisch auf beide Cluster anwenden. Anschließend sollte der private Cluster wie in Listing 5 aussehen. Der öffentliche Cluster sollte dann entsprechend aussehen wie in Listing 6. Nun ist es möglich, einen Job anzulegen, den der Multicenter-Scheduler automatisch einem freien Cluster zuweist. In diesem Fall ist der Pod (mljob-6ymu-

xvrv2-p5bbl) dem Cluster in der Public Cloud zugewiesen. Bei dem Pod im privaten Cluster handelt es sich lediglich um den Proxy-Pod, wie Listing 7 veranschaulicht. Beim Erstellen weiterer Jobs sieht man, dass beide Cluster nach und nach stärker ausgelastet sind. Ab einem gewissen Punkt greift in der Public Cloud das Autoskalieren, dem betroffenen Cluster werden neue Nodes und damit Ressourcen hinzugefügt. Sobald die Trainingsjobs abgeschlossen sind, sinkt die Last erneut. Mit einer gewissen Verzögerung sollte dann auch die Anzahl der Nodes in der Public Cloud wieder zum Ausgangswert skalieren. Über die Mechanismen `nodeSelector` [10] und `nodeAffinity` [11] lässt sich das Verhalten des Schedulers beeinflussen, um zum Beispiel den privaten Cluster für die Ausführung eines Pods zu bevorzugen.

Lastschwankungen abfedern

Mit den entsprechenden Kubernetes-Erweiterungen lässt sich vergleichsweise einfach eine hybride Cloud mit Unterstützung



>> Aufbau des Istio-Multicenters (Abb. 4)

Java: Multitalent für starke IT-Anwendungen in Österreich

Das Bundesrechenzentrum (BRZ) in Österreich entwickelt Anwendungen für seine Kunden in der Verwaltung und setzt dabei zahlreiche Technologien ein. Eine davon ist Java, das vor allem bei vielen aktuellen Anwendungen zum Einsatz kommt. Vorteile wie die vielseitige Anwendbarkeit, die große Community, der plattformunabhängige Einsatz und der hohe Reifegrad haben dazu geführt, dass Java im BRZ hoch im Kurs steht. Knapp 200 Developer sind auf Java-Entwicklung spezialisiert.

BRZ spricht viele Java-Sprachen

Java ist nicht gleich Java, im BRZ in Wien sind eine Reihe unterschiedlicher Frameworks im Einsatz, die je nach Anforderung eingesetzt werden können. Zum Einsatz kommen etwa Spring bei der Dependency Injection und bei der Integration von aspektorientierter Programmierung. Camunda wird für individualisierte und eigenständige Prozessautomatisierungen eingesetzt. Wird ein ausgereifter OR-Mapper benötigt, der auch den JPA-Standard implementiert, greifen die Developer im BRZ zu Hibernate. Für die Integration von DevOps und die automatisierte Erstellung von Applikationsversionen kommt Jenkins zum Einsatz. „Wo möglich, versuchen wir natürlich kundenübergreifend zu arbeiten und einheitliche Technologien einzusetzen. Mit Java, seinen Frameworks und Libraries, können wir aber flexibel auf Anforderungen reagieren und die Vorteile der einzelnen Produkte für unsere Kunden nutzen“, so Sandra Teix, die selbst seit 10 Jahren im BRZ als Java-Developer arbeitet.

Java – eine lebendige Community

„Entwickler/innen sind keine Einzelkämpfer/innen, schon gar nicht im BRZ“ ist Java-Developer Markus Reichstädter überzeugt. Der regelmäßige Austausch im Team ist wichtig. Technologisch setzt man im BRZ nicht nur auf Collaboration-Tools wie Microsoft Teams, sondern auch Wissensmanagement-Lösungen wie Confluence. Der persönliche Austausch erfolgt im wöchentlichen AC/DC-Treffen, kurz für „Architects Circle/Development Continuity Community“. „Dank des projektübergreifenden Austauschs lernen wir als Developer nicht nur laufend dazu, wir sorgen auch für einheitliches Coding im BRZ“, so Reichstädter.

Mit OpenShift in die Cloud-Zukunft

„Java wird uns auch in Zukunft begleiten und eine der wichtigsten Programmiersprachen im BRZ bleiben“ sind Teix und Reichstädter überzeugt. Geplant sind etwa die Umstellung und Modernisierung der Applikationen auf eine Microservice-basierte Applikationsarchitektur und die Verwendung einer OpenShift-Cloud-Plattform. „Wer Lösungen entwickeln will, die den Menschen das Leben leichter machen und Freude bereiten, muss moderne Technologien einsetzen und immer am Ball bleiben. Das ist uns als Java-Developer im BRZ auch persönlich wichtig“ sagen die beiden BRZ-Java-Profis. Die nächsten spannenden Projekte haben sie schon in der Pipeline.

Wir vergeben Topjobs mit Sinn!

Sie möchten **Apps und Services** entwickeln, die ganz Österreich nutzt?

Sie möchten am Ausbau von Österreichs **eGovernment** aktiv beteiligt sein?

Sie wollen **neue Technologien** einsetzen, **Innovationen** umsetzen und kreative Lösungen entwickeln?

Wir sind der **Marktführer im IT-Bereich** des Public Sectors in Österreich und bieten spannende Positionen von **Trainee bis Senior Developer**.

Starten Sie Ihre Karriere im BRZ!

Jetzt gleich online bewerben unter [brz-jobs.at!](https://brz-jobs.at)



Boban Blazevski
Java Developer

Mediha Fejzic
Java Developer

Listing 4: Deployment für den privaten und den öffentlichen Cluster

```
$ kubectl --context ${PRIVATE_CLUSTER_CTX} apply -f k8s/namespace.yaml
$ kubectl --context ${PRIVATE_CLUSTER_CTX} apply -f k8s/blob-storage.yaml
$ kubectl --context ${PRIVATE_CLUSTER_CTX} apply -f k8s/blob-storage-7
service.yaml
```

Listing 5: Privater Cluster nach dem initialen Deployment

```
$ kubectl --context ${PRIVATE_CLUSTER_CTX} get all
NAME                                READY   STATUS    RESTARTS   AGE
pod/blob-storage-d4556c66f-z79f2    2/2     Running   0           2d5h
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
AGE				
service/blob-storage	ClusterIP	10.0.225.183	<none>	10000/TCP
3d6h				
service/kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP
5d23h				

```
NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/blob-storage        1/1     1             1           3d5h
```

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/blob-storage-d4556c66f	1	1	1	3d5h

Listing 6: Öffentlicher Cluster nach dem initialen Deployment

```
$ kubectl --context ${REMOTE_CLUSTER_CTX} get all
NAME                                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)
AGE
service/blob-storage                ClusterIP      10.0.42.74   <none>         10000/TCP
3d6h
service/kubernetes                  ClusterIP      10.0.0.1     <none>         443/TCP
7d6h
```

Listing 7: Scheduling eines ML-Jobs im öffentlichen Cluster

```
./k8s/new-job.sh
job.batch/ml-job-6ymu created

$ kubectl --context ${PRIVATE_CLUSTER_CTX} get pods
NAME                                READY   STATUS    RESTARTS   AGE
blob-storage-d4556c66f-z79f2        2/2     Running   0           2d5h
ml-job-6ymu-xvzv2                    2/2     Running   0           20
```

```
$ kubectl --context ${REMOTE_CLUSTER_CTX} get pods
NAME                                READY   STATUS    RESTARTS   AGE
ml-job-6ymu-xvzv2-p5bb1             2/2     Running   0           27s
```

für Cloud Bursting aufsetzen. Ein elementarer Baustein dafür ist die Cross-Cluster-Kommunikation, das konkrete Beispiel hat sie mit Istio umgesetzt. Die zweite wichtige Komponente ist der Multicloud-Scheduler, der dafür sorgt, dass sich die Ressourcen der hybriden Cloud zu einem gemeinsamen Pool zusammenfassen lassen. Auch wenn es nicht zwingend notwendig ist, erleichtert die Verwendung von Kubernetes Cluster Federation die Verwaltung mehrerer Cluster und ist damit besonders für größere Projekte von Nutzen.

Cloud Bursting kann dabei helfen, die Cloud bei Lastschwankungen besser zu nutzen. Durch die weite Verbreitung von Kubernetes gibt es eine große Auswahl an Cloud-Anbietern, die dafür in Frage kommen. Entwickler können auf ein wachsendes Ökosystem an Erweiterungen zurückgreifen.

Weitere Projekte, die sich ebenfalls mit dem Thema Multicloud und hybride Cloud beschäftigen, sind zum Beispiel Cilium [12], Submariner [13] und Google Anthos [14]. (sih)

Quellen

- [1] Prognose für das Cloudgeschäft bis 2022: gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g
- [2] Definition von Cloud Computing nach NIST: nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
- [3] Kubernetes, Übersicht über die Plattform und das Aufsetzen von Clustern: kubernetes.io/de/docs/concepts/overview/what-is-kubernetes/
- [4] MNIST-Datenbank: yann.lecun.com/exdb/mnist/
- [5] Ein Istio-Mesh auf mehreren Clustern installieren: istio.io/latest/docs/setup/install/multicloud/

- [6] Istio-Beispielanwendung Bookinfo: istio.io/latest/docs/examples/bookinfo/
- [7] Kubernetes Cluster Federation (KubeFed): github.com/kubernetes-sigs/kubefed#kubernetes-cluster-federation
- [8] Installationsleitfaden für KubeFed auf GitHub: github.com/kubernetes-sigs/kubefed/blob/master/docs/userguide.md
- [9] Multicloud-Scheduler: admiralty.io/
- [10] nodeSelector: kubernetes.io/docs/concepts/scheduling-eviction/assign-podnode/#node-selector
- [11] nodeAffinity: kubernetes.io/docs/concepts/scheduling-eviction/assign-podnode/#node-affinity
- [12] Ciliums ClusterMesh: cilium.io/blog/2019/03/12/clustermesh/
- [13] Submariner ermöglicht direktes Netzwerken zwischen Pods und Services in verschiedenen Kubernetes-Clustern: submariner.io/
- [14] Verwaltete Anwendungsplattform Google Anthos: cloud.google.com/anthos/?hl=de



Simon Mennig

hat Wirtschaftsinformatik (B.Sc.) und Informatik mit Schwerpunkt Software Engineering (M.Sc.) studiert. Er arbeitet als Softwareentwickler bei doubleSlash und hat umfassendes Know-how in den Bereichen IoT, Cloud Computing und Java. Das Thema „Cloud Bursting“ war ein Teil seiner Masterarbeit beim IT-Dienstleister doubleSlash.

Cloud Native Full Stack und Apps für Millionen Anwender?

Neu-, Um- und Mitgestalter, die die spannende Herausforderung annehmen, ein riesiges Software-Portfolio in die Cloud zu heben – solche Leute sind gefragt bei DATEV. Seine über 200 Programme und IT-Dienstleistungen will der drittgrößte Anbieter von Business-Software in Deutschland nämlich Stück für Stück neu denken.

Auf Basis moderner, flexibler Cloud-Architektur soll perspektivisch eine digitale Plattform entstehen – ein echtes Megaprojekt. Denn geplant ist keine 1:1-Umsetzung der vorhandenen Lösungen. DATEV will noch stärker den Aspekt Collaboration betonen und nicht nur einen Transfer schaffen, sondern ein durchdachtes Ökosystem bereitstellen, in das sich auch Drittanbieter im Rahmen einer Plattformstrategie für den deutschen Mittelstand einklinken können. Für diese spannende Transformation werden die entsprechenden Skills benötigt.

Neben Cloud-Native-Full-Stack-Entwicklern sucht DATEV aktuell vor allem Software Engineers und Java-Entwickler. Denn zukunftssträchtige Softwareautomatisierungs-Technologien wie MPS halten Einzug in den Tech Stack der Genossenschaft aus Nürnberg. Neben Cloud-Anwendungen rückt auch die App-Entwicklung ins Blickfeld – in den kommenden Jahren sollen auch Produkte und Dienstleistungen für Hunderttausende von Endverbrauchern entstehen. Softwareentwicklung ist bei DATEV über flexible Workstreams organisiert, die Entwicklung folgt agilen Prinzipien. Flexible Teams, die Flex Hubs, unterstützen die „normalen“ Workstreams bei Bedarf an unterschiedlichsten Stellen und übernehmen für die Dauer von drei Monaten jeweils Arbeitspakete.

Mit dem DATEV Lab existiert zudem seit Jahren eine eigene Unit, die sich kompromisslos der Entwicklung neuer Business-Cases widmen kann. Über einen hauseigenen Inkubator werden die vielversprechendsten Ansätze im Haupthaus weiterverfolgt.

DATEV – digitale Transformation im Markt und in der Organisation

Ist das noch die DATEV, die viele als grünes Leucht-Quadrat wahrnehmen? – strahlend an Bahnhöfen und in Airport-Terminals quer verstreut durch die gesamte Republik. Allenfalls bedingt, denn die digitale Transformation treibt das Unternehmen nicht nur in seinen Märkten voran, sondern mit sehr viel Mut und Innovationskraft auch bei sich selbst im Inneren.

Die Rechtsform von DATEV bleibt mit ihren Besonderheiten als verlässlicher Eckpfeiler der Marke erhalten: Als Genossenschaft ist sie keinen Aktionären verpflichtet, man hechelt also nicht von Quartal zu Quartal, sondern kann Entwicklungslinien langfristig und nachhaltig verfolgen. Rund 40.000 Mitglieder insbesondere aus dem Bereich der Steuerberater, Wirtschaftsprüfer und Rechtsanwälte sind Träger dieser Genossenschaft. Damit ist DATEV einer der wesentlichen Unterstützer des deutschen Mittelstandes, genauer gesagt von rund 2,5 Millionen Unternehmen,



vor allem im Bereich der KMUs. Über 400.000 Kunden greifen auf DATEV-Anwendungen zurück. Die mehr als 8.100 Mitarbeiterinnen und Mitarbeiter erwirtschaften Jahr für Jahr einen Umsatz von über 1 Milliarde Euro.

Unter dem Dach des grünen Quadrats schlägt das technische Herz der Genossenschaft. Über 2.000 Entwicklerinnen und Entwickler widmen sich Zukunftstechnologien und sorgen dafür, dass bei DATEV längst mehr als nur Steuersoftware entsteht.

Durch die engen Verbindungen in den Mittelstand ist das Unternehmen entscheidender Treiber der Digitalisierung in Deutschland. Die rund 13,5 Millionen Lohnabrechnungen, die jeden Monat mit Software der Genossenschaft abgewickelt werden, sorgen für einen unvergleichlichen Datenschatz. Big Data, Blockchain & Co. sind technologische Entwicklungen, die immer stärker den Fokus leiten.

Von Wachstum, Expansion und Flexibilisierung

Die Wurzeln von DATEV liegen unbestreitbar in Nürnberg. Mittlerweile streckt die Genossenschaft ihre Fühler aber auch in andere deutsche Städte aus. In der Berliner Dependence werden derzeit Software Engineers Backend für die Arbeit im grünen Quadrat gesucht.

Lust bekommen, Teil von DATEV zu werden?
Weitere Infos findest Du unter www.datev.de/it.



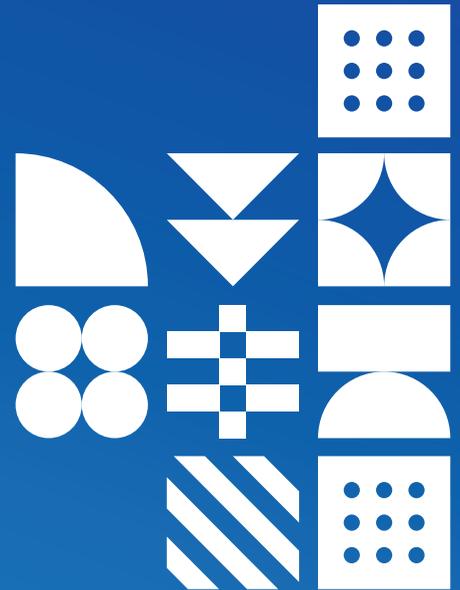
DATEV eG
90329 Nürnberg

www.datev.de

> BDD – Segen oder Fluch?

Nils Kasseckert

Software mit hoher Qualität und im Zeit- und Kostenrahmen zu entwickeln, schließt sich nicht zwangsläufig gegenseitig aus. BDD zeigt, wie es in der Theorie funktionieren kann.



Wahrscheinlich kennen alle Entwickler:innen mindestens eines der folgenden Probleme: Der Zeit- und Kostenrahmen für das Projekt ist zu eng gesteckt und es bleibt keine Kapazität mehr, um sich um die Qualität der Software zu kümmern. Oder nach fast abgeschlossener Entwicklung stellt sich heraus, dass die Software am Kunden vorbei entwickelt worden ist. Damit das nicht mehr geschieht, entstand das sogenannte Behaviour-Driven Development (BDD).

BDD (zu deutsch: verhaltensgetriebene Softwareentwicklung) ist eine aus dem Test-Driven Development (TDD) entstandene agile Form der Softwareentwicklung. Sie wurde 2013 durch Dan North erstmalig beschrieben und hat sich seitdem kontinuierlich von einer reinen Erweiterung des TDDs zu einer eigenen Methodik weiterentwickelt. Wie beim TDD steht bei der verhaltensgetriebenen Entwicklung der Test im Mittelpunkt. Bevor eine Änderung im produktiven Code stattfinden kann, muss zuerst ein Test entwickelt werden.

Worin unterscheiden sich BDD und TDD?

Der Unterschied zwischen den beiden Entwicklungsmethoden liegt in der Definition des Tests. Beim Test-Driven Development kommt in der Regel ein Unit-Test zum Einsatz. Er ist in der jeweils verwendeten Programmiersprache verfasst und für gewöhnlich nur für Entwickler verständlich. Beim Behaviour-Driven Development hingegen wird der Test in der

natürlichen Sprache der jeweiligen Domäne verfasst. Er dient gleichzeitig auch als lebende Dokumentation der Software. Das hat den Vorteil, dass nicht nur Entwickler, sondern auch Product Owner, Tester und andere nicht technikaffine Personen den Test verstehen und bestenfalls sogar verbessern können. Wie angedeutet ist das Ziel beim Einsatz von BDD, eine einfach zu wartende Software mit hoher Qualität nach Wünschen der Kunden zu erhalten.

In der Praxis wird das Behaviour-Driven Development oft mit Techniken des Extreme Programming kombiniert. Extreme Programming lässt sich als eine Sammlung verschiedener Methoden auffassen, die allesamt dieselben Ziele wie die verhaltensgetriebene Softwareentwicklung verfolgen. TDD und BDD stellen einen Teil dieser Methoden dar. In meinem ehemaligen Team findet genau eine solche Kombination statt. Aus diesem Grund präsentiere ich im Folgenden die Methoden, die in einem Scrum-Team innerhalb eines großen Konzerns zum Einsatz kommen. Zu ihnen zählen:

- der Test-First-Ansatz
- das Pair Programming
- die gemeinsame Verantwortung
- Tests als Dokumentation und Planung
- der Kunde vor Ort

Beim Einsatz des Test-First-Ansatzes wird vor der Entwicklung der Funktionalität zuerst ein entsprechender Test ge-

schrieben. Diese Tests sollen sich immer wieder automatisch ausführen lassen. Somit ist es wichtig, dass sie unabhängig voneinander sind und sich nicht gegenseitig beeinflussen. Auch beim Auftreten eines Bugs werden zuerst der entsprechende Test entwickelt und erst im Anschluss daran der Fehler behoben. Das stellt sicher, dass dieser spezielle Fehler ab sofort automatisch die Testung durchläuft und somit in der Produktion nicht mehr auftreten kann. Durch den Test-First-Ansatz lassen sich mehrere Probleme auf einmal lösen. Zum einen stellt er eine hohe Testabdeckung sicher, da für jede Funktionalität zwangsläufig ein Test zu entwickeln ist. Zum anderen kann sich das Team im Allgemeinen darauf verlassen, dass der entsprechende Test die Fehler findet. Der Test sollte vor dem Start der Entwicklung fehlschlagen. Ist das nicht der Fall, ist der Test falsch entwickelt und vor der eigentlichen Implementierung zu ändern. Dieser False-Positive-Fall lässt sich bei einer nachträglichen Testentwicklung kaum entdecken. Ein Test, der immer, unabhängig von den Rahmenbedingungen, erfolgreich ist, ist ein nutzloser Test.

> Ein Test, der immer erfolgreich ist, ist ein nutzloser Test.

Das Pair Programming wird zur Erhöhung der Codequalität und des Bus-Faktors eingesetzt. Der Bus-Faktor ist eine

Kennzahl zur Abschätzung von Projektrisiken. Er beschreibt, wie viele Personen ausfallen können, ohne dass es das Projekt gefährdet. Beim Pair Programming arbeiten in der Regel zwei Personen an einer Funktionalität. Eine Person ist für die Eingabe des Programmcodes verantwortlich und die andere Person für die Kontrolle des eingegebenen Codes. Die Rollen wechseln mindestens täglich. Spätestens einmal pro Woche sollten die Paare getauscht werden. Kürzere Abstände sind möglich, je nach Aufgabe jedoch nicht sinnvoll. Eine Zeile Code entspricht immer dem Konsens zweier Personen. Wenn eine der beiden Personen ausfällt, lässt sich dennoch ohne Weiteres an der Funktionalität weiterarbeiten. Auch verbessert sich in der Regel der Code, wenn man gemeinsam darüber spricht.

In vielen Entwicklungsteams herrscht eine strikte Trennung von Themen und Aufgaben. Tritt ein Bug auf, ist meist sofort klar, wer für diesen Bereich zuständig ist und das erforderliche Wissen zur Behebung des Fehlers besitzt. Ist die betreffende Person jedoch gerade nicht verfügbar, kennt sich niemand anderes beim Thema aus. Das Prinzip der gemeinsamen Verantwortung kann ein solches Problem lösen. Für ein spezielles Aufgabengebiet ist nicht nur eine Person ver-

VOLKSWAGEN
GROUP SERVICES

**DIGITALE TRANSFORMATION
in der AUTOMOBILINDUSTRIE**

[Sichere Arbeitsplätze] +++ [Gute Work-Life-Balance]
[Kooperative Arbeitsatmosphäre] +++ [Mobile Arbeit]
www.volkswagen-groupservices.com/it-jobs

NEUGIERIG ?
Hier geht's zu
unseren IT-Jobstories!

Listing 1: Definition Feature-File

```
# language: de

Funktionalität: Addition
Mein Taschenrechner soll die Addition beherrschen

Szenario: Regular numbers
  Angenommen, ich habe die Zahl 10 in den Taschenrechner eingegeben
  Und ich habe die Taste "+" gedrückt
  Und ich habe die Zahl 5 in den Taschenrechner eingegeben
  Wenn ich auf die Taste "=" drücke
  Dann wird als Resultat 15 auf dem Bildschirm ausgegeben
```

antwortlich, sondern jeder ist für alles zuständig und besitzt auch entsprechendes Wissen. Durch das Pair Programming lässt sich das in der Praxis einfach umsetzen. Bei jedem Wechsel der Partner kommen die noch offenen Aufgaben zur Sprache. Eine Person im Pair behält die Verantwortung für die Aufgabe und ist gleichzeitig für den Wissenstransfer zu ihrem neuen Pairing-Partner zuständig. Die andere Person wechselt in ein neues Pair mit einem anderen Thema. Hierbei ist zu beachten, dass eine Person nie länger als zwei bis drei Wochen in einem Thema bleiben sollte. Ansonsten lässt sich der Wissenstransfer von anderen Themen für diese Person nicht gewährleisten.

Eine zwingende Grundvoraussetzung für den Einsatz dieser Methoden ist ein gutes Auskommen miteinander. Existieren Abneigungen zwischen einzelnen Mitgliedern, ist das Pair Programming nur schwer zu realisieren. Auch muss sich jeder mit dem zu entwickelnden Produkt identifizieren können, damit das Prinzip der gemeinsamen Verantwortung auch gelebt wird und nicht nur auf dem Papier besteht. Weiterhin ist ein ähnliches Wissensniveau hilfreich, da es sonst beim Pair Programming zur Unter- oder Überforderung kommen kann. Auch sollte die Teamgröße nicht mehr als zehn Entwickler betragen. Ansonsten sind der Aufwand zum Austausch und zum notwendigen Wissenstransfer zu hoch.

Behaviour-Driven Development konkret

Beim Behaviour-Driven Development (BDD) dient der Test zugleich als Dokumentation und Planung der Anforderungen. Eine neue Funktion wird nicht mittels Ablaufdiagrammen oder Ähnlichem geplant, sondern schriftlich in der Syntax der verhaltensgetriebenen Entwicklung festgehalten. Hierfür ist es notwendig, dass Kunden bei der Planung vor Ort sind. In einem typischen Entwicklungsteam übernehmen die Product Owner die Rolle der Kunden. Durch die dauerhafte Verfügbarkeit der Kunden lassen sich Fragen und Probleme jederzeit klären und der Arbeitsablauf anpassen.

> Tests dienen zugleich der Dokumentation und Planung der Anforderungen.

Um die beschriebenen Techniken umzusetzen, sind verschiedene Werkzeuge hilfreich. Einen Test im Behaviour-Driven Development kann man mit dem Framework Cucumber und der Beschreibungssprache Gherkin definieren. Hierbei folgt ein BDD-Test immer einem festen Schema von Schlüsselwörtern. Beginnend mit "Gegeben sei" oder "Angenommen" wird zuerst der Kontext definiert, in dem der Test stattfindet. Anschließend werden Aktionen mit dem Schlüsselwort "Wenn" eingeleitet und das erwartete Ergebnis mit dem Schlüsselwort "Dann" geprüft (s. Listing 1).

Es besteht die Möglichkeit, mehrere Schritte desselben Typs zu nutzen. Das Verknüpfen dieser Schritte geschieht durch das Schlüsselwort „Und“. Ein Feature-File bündelt thematisch zusammengehörige einzelne Tests. Es enthält einen Titel und eine kurze Beschreibung der Intention. Die Definition der Schritte kann in verschiedenen Sprachen erfolgen. Englisch gilt als Standardsprache. Soll eine andere Sprache genutzt werden, ist diese in der ersten Zeile anzugeben. Nach dem Schreiben des Feature-Files werden die Schritte programmatisch umgesetzt. Die Umsetzung im Code erfolgt aus einer Kombination von Annotationen und einer speziell für Cucumber angepassten Art von Regex. Eine Schritt-für-Schritt-Implementierung kann auch Variablen enthalten, die automatisch aus der Textdefinition extrahiert werden, wie Listing 2 illustriert.

Mögliche Programmiersprachen für die Umsetzung sind Java, JavaScript, C++ sowie alle auf der Java Virtual Machine (JVM) basierenden Sprachen. Die Verknüpfung zwischen Schritt-für-Schritt-Implementierung und Feature-File findet automatisch durch Cucumber statt. Hierfür ist nur eine einmalige Konfiguration mittels der so genannten CucumberOptions notwendig. Viele Entwicklungsumgebungen bieten mittlerweile sogar mit entsprechenden Plug-ins das Debuggen eines Feature-Files an.

Für die Zusammenarbeit zwischen den verschiedenen Paaren ist Git mit entsprechenden Review-Guidelines und verbindlichen Code-Guidelines notwendig. Auch muss man eine Form des Pairings finden. Hierfür gibt es verschiedene Möglichkeiten. Grundlegend kann man das Zusammenar-

Listing 2: Implementierung des Feature-Files

```
import io.cucumber.java.de.Angenommen;

public class Addition {

    @Angenommen("Ich habe die Zahl {int} in den Taschenrechner eingegeben")
    public void ichHabeDieZahlInDenTaschenrechnerEingeben(int zahl) {

    }
}
```

beiten virtuell oder vor Ort gestalten. Vor Ort hat den Vorteil, dass der direkte Austausch einfacher und die Konzentration höher ist. Für das virtuelle Pairing ist ein geeignetes Tool notwendig. Das Teilen des Bildschirms mittels Teams, Skype oder Slack ist oft unscharf und zeitverzögert. Das erschwert die Zusammenarbeit erheblich. Aus diesem Grund sollten Teams den Einsatz spezieller Pairing-Tools wie "Code With Me" für IntelliJ-Produkte oder das "Live Sharing Extension Pack" für VS Code evaluieren. Mit diesen Plug-ins wird nicht der Screen als Ganzes geteilt. Stattdessen öffnen beide Paare die jeweilige Entwicklungsumgebung. Die gerade programmierende Person ist Host der Session. Die andere nimmt als Beobachter teil und sieht alle ausgeführten Aktionen.

Umsetzung in der Praxis

In der Theorie klingen die vorgestellten Methoden gut, aber wie ist die Erfahrung in der Praxis damit? Hierzu kurz der Aufbau des Teams: Es besteht aus sieben Entwickler:innen, einem Architekten, zwei Tester:innen, zwei Product-Ownern, einer Scrum-Verantwortlichen und einem Personalverantwortlichen. Wir sind ein multikulturelles Team, das auf ver-

schiedene Standorte verteilt ist. Zwei der Entwickler haben ihren Sitz in Frankreich. Einer der Product Owner und die Scrum-Verantwortliche sind quer durch Deutschland verteilt. Das restliche Team ist auf einen Standort konzentriert.

Die ungerade Anzahl von Entwicklern ist Fluch und Segen zugleich. Bei Anwesenheit des gesamten Teams entsteht zwangsläufig ein „Pair“ aus drei Personen. Da bei uns im Unternehmen flexible Vertrauensarbeitszeit herrscht, ist es schwer, einen passenden Slot zu finden, bei dem alle drei Personen verfügbar sind. Im Krankheitsfall oder bei Urlaub einer Person lassen sich jedoch durch die ungerade Anzahl immer noch zuverlässig Paare bilden. Auch kommt es häufig vor, dass die Tester:innen für die automatischen Tests die Unterstützung eines Entwicklers benötigen. Somit ist eine Dreierkonstellation eher selten. Im Allgemeinen erschweren jedoch die unterschiedlichen Arbeitszeiten die Zusammenarbeit. Einige Personen starten zwischen 6.30 und 7 Uhr, andere erst zwischen 9 und 10 Uhr. In den Leerlaufzeiten wird ohne Partner gearbeitet, was dem Prinzip des Pair Programming widerspricht.

Ein weiteres Problem beim Pair Programming ist das vorherrschende unterschiedliche Wissensniveau. Manche Personen programmieren seit vielen Jahren in Java oder

Major Upgrade for the Next Level

FERCHAU 

**Werden Sie Teil unseres Teams.
Jetzt bewerben.**

Was unsere Kunden für ihre Technologie-Lösungen erwarten? Das nächste Level! Das gelingt unserem Team Tag für Tag. Weil bei uns neue Engineering- und IT-Herausforderungen auf die jeweils besten unserer Experten treffen – Experten wie Sie. Unser Deal: Ihr Potenzial und unsere Möglichkeiten. Challenge accepted? Starten wir gemeinsam das nächste Level und entwickeln wir die Zukunft!

FERCHAU GmbH, Zentrale
bewerber@ferchau.com

ferchau.com/go/karriere-it

Connecting People and Technologies
for the Next Level



JavaScript. Andere wiederum haben viele Jahre in einer anderen Sprache programmiert und tun sich mit dem Umstieg schwer. Das führt zu Frustration auf beiden Seiten. Ziel ist es, dass sich das Wissensniveau durch das Programmieren in Paaren langfristig angleicht. Bis dahin ist es jedoch ein weiter Weg.

Auch erschweren kulturelle Unterschiede die Zusammenarbeit. Beim Pair Programming ist es notwendig, über die Lösung zu diskutieren, um den besten Weg zu finden. Ziel ist die langfristige Erhöhung der Codequalität. Jedoch ist zum Beispiel in der chinesischen Kultur eine Diskussion eher unüblich. Chinesische Kollegen sind daher immer der gleichen Meinung wie die Gesprächspartner. Das ist wenig hilfreich. Auch sind die verschiedenen Arbeitsweisen in den verschiedenen Kulturen schwierig aneinander anzupassen. Im schlimmsten Fall entstehen Abneigungen zwischen einzelnen Kolleg:innen. Das ist bei uns passiert. Wenn diese Personen miteinander arbeiten sollen, werden die Aufgaben untereinander aufgeteilt und jeder arbeitet für sich. Das Aufteilen geschieht ohne Wissen des Managements und widerspricht den Prinzipien des Pair Programming sowie der gemeinsamen Verantwortung. Selbstverständlich sind dies nur Einzelfälle und der Großteil des Teams versteht sich gut untereinander. Jedoch ist auch für solche Fälle eine Lösung zu finden.

Bei Themen, die die Kommunikation mit anderen Teams erfordern, kommt es durch den ständigen Wechsel im Pair Programming oft zu Irritationen. Es ist für den anderen Kollegen oft nur schwer nachvollziehbar, warum sich der Ansprechpartner schon wieder geändert hat. Auch muss das Pairing aufgrund der unterschiedlichen Standorte oft virtuell stattfinden. Wegen der Platzierung des Teams in einem Großraumbüro entsteht eine große Geräuschkulisse, und für die Personen vor Ort ist es nicht leicht, sich zu konzentrieren. Aus Kostensicht scheint das Programmieren in Paaren zunächst teuer. Jedoch wird dadurch das Wissen auf viele Personen verteilt, und eine einzelne Person lässt sich ohne große Probleme ersetzen.

Das Behaviour-Driven Development erfordert viel Zeit zur Definition der Tests. Zu Beginn hat das gesamte Team jedes Feature-File gemeinsam geschrieben. Das stellte sich jedoch als zu zeitintensiv heraus. Aus diesem Grund werden nun die Tests vom Paar selbst geschrieben und nach Abschluss der Aufgabe mit dem gesamten Team in einem Meeting durchgesprochen. Dadurch lässt sich zwar viel Zeit einsparen, jedoch leidet hierunter der Test-First-Ansatz. Oft werden nun zuerst die Funktionalität und anschließend der Test entwickelt. Dies hat vor allem zwei Gründe: Zum einen wissen die Entwickler:innen oft zu Beginn noch nicht, aus

welchen UI-Komponenten die Funktionalität besteht oder wie sie allgemein umzusetzen ist. Somit müsste jedes Mal das Feature-File geändert werden. Vor diesem Aufwand schrecken viele zurück. Zum anderen ist leider das verwendete Test-Setup instabil. Das Team steckt mehr Zeit in das Beheben von Fehlern im Test als in die Entwicklung der eigentlichen Funktion. Das führt zunehmend zu der Entscheidung, für kleinere Funktionalitäten keinen Test zu entwickeln, da dies einen zu großen Aufwand darstellen würde. Prinzipiell ist jedoch das Ziel der automatischen Tests, den später folgenden Wartungsaufwand so gering wie möglich zu halten, sodass man sich mehr auf Neuentwicklungen konzentrieren kann. Auch spart eine Firma dadurch Kosten im Bereich Support ein.

Da die Kunden (in unserem Fall die Product Owner) immer direkt vor Ort sind, können wir auftretende Fragen schnell beantworten. Das geschieht in verschiedenen Meetings, an denen

das gesamte Team teilnimmt. Somit ist jedes Teammitglied immer informiert und kann aktiv an der Entscheidungsfindung mitwirken. Nachteilig anzumerken ist jedoch, dass durch Diskussionen und Ähnliches die Meetings sehr lange dauern. Es kommt mitunter vor, dass bis zu 60 Prozent der wöchentlichen Arbeitszeit aus Meetings besteht, worunter die Produktivität erheblich leidet. Daraus folgt oft, dass sich das Prinzip der gemeinsamen Verantwortung nicht umsetzen lässt. Durch die so schon begrenzte Zeit für die Aufgabe haben Teammitglieder oft keine Zeit mehr, gemeldete Fehler zu beheben oder einen Pull Request zu begutachten. Hierfür eingeführte Regeln, wie das Begutachten eines Pull Request während des Pairings oder dass das Beheben von Fehlern die höchste Priorität hat, brachten bisher nicht den gewünschten Erfolg.

Auch lebt nicht jeder die offene Feedbackkultur. Es existieren zwar Meetings, die genau diesen Zweck verfolgen, jedoch werden Probleme oft nicht angesprochen. Das hängt mit den verschiedenen nationalen Kulturen zusammen. Ein paar Entwickler sind zum Beispiel mit dem Arbeiten in Paaren oder der derzeitigen Teststrategie unzufrieden, kommunizieren das jedoch nicht. Solche Themen besprechen sie nur mit Entwicklern, zu denen sie ein gutes Verhältnis haben. Das Management fördert diese geschlossene Feedbackkultur im Rahmen der BDD-Strategie mit Pair Programming. Viele haben nicht den Mut, etwas dagegen einzuwenden und sich aktiv gegen das Management zu stellen. Dieses wiederum kennt nicht den realen Ablauf während des Entwicklungsprozesses und kann somit nicht gesteuern. Selbst wenn die Führungsebene diese Themen aktiv anspricht, kommunizieren die Entwickler:innen die Probleme nicht.

> Pair Programming ist oft eine kulturelle Herausforderung.

BDD: Fluch oder Segen?

Dennoch lässt sich nicht pauschal sagen, dass der Einsatz von Behaviour-Driven Development in Kombination mit Techniken des Extreme Programming unmöglich sei. Es existieren durchaus Teams innerhalb und außerhalb des Unternehmens, in denen dieser Ansatz problemlos funktioniert und gute Ergebnisse zutage bringt. Auch in meinem ehemaligen Team sind durchaus einige Entwickler, in der richtigen Pairing-Konstellation, zufrieden. Die Idee der besseren Codequalität und der Kunden vor Ort funktioniert gut. Außerdem lässt sich durch die automatisierten Tests eine durchgängige Funktionalität der Software gewährleisten. Das ist der Grundstein für kürzere Lieferungszyklen.

Die größte Herausforderung beim Einsatz dieses Konzepts ist jedoch der Faktor Mensch. Multikulturelle Unterschiede sind schwierig zu handhaben. Erschwerend kommen unterschiedliche Sprachen hinzu. Englisch ist zwar in der Informatik Standard, jedoch fühlen sich beide Pairing-Partner in der Muttersprache wohler, und es kommt manchmal zu Missverständnissen. Selbst unterschiedliche Arbeitszeiten sind, wie erwähnt, ein Problem. Nichtsdestoweniger kann das Prinzip gut funktionieren. Hilfreich hier-

für ist jedoch ein nicht interkulturelles Team, das wenig bis gar keine Abhängigkeiten zu anderen Teams besitzt. Auch auf jede einzelne Person kommt es an. Es kann mitunter nicht in dem gewohnten Arbeitstempo gearbeitet werden und es ist zu Beginn eine große Umstellung, jegliche Aufgaben gemeinsam zu erledigen. Auch sollte einem das Vermitteln von Wissen Spaß machen, ohne dass man vor schnell frustriert aufgibt. Dies muss jeder Person bewusst sein, wenn sie in einem Team arbeiten will, das Pair Programming nutzt.



Nils Kasseckert

arbeitet nach seinen Erfahrungen mit dem Pair Programming als Technischer Architekt in der IT-Sparte eines großen deutschen Handelskonzerns.

Seit 2013 betreibt er zusätzlich sein Nebengewerbe „AppSupporter“. Er interessiert sich für Softwarearchitektur, Embedded Systems sowie jegliche neue Technologien.



Ob Fullstack, Frontend oder Backend – wir freuen uns auf dich und deine Ideen zur Beseitigung der Spider-App!

ENTWICKLER (M/W/D)
GESUCHT

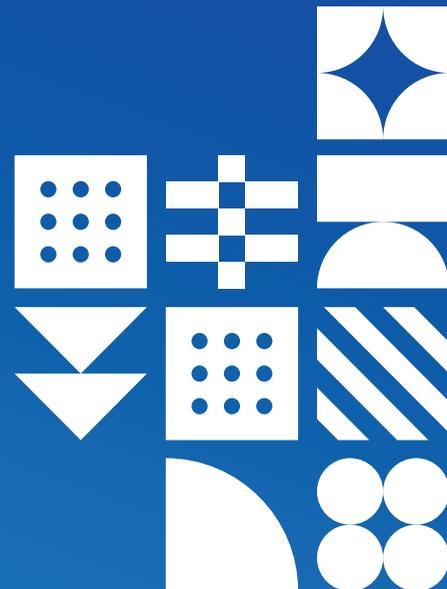
Neugierig?

Dann besuch uns einfach auf wertgarantie-group.com/karriere oder melde dich bei Jacqueline Benjamin unter 0172 7347845.

> Neustart als Developer

Sead Ahmetović und Thomas Limbüchler

IT-Fachkräfte sind händeringend gesucht. Alternative Aus- und Weiterbildungsmöglichkeiten sollen Abhilfe schaffen. Nie waren die Wege in die IT-Branche vielfältiger.



Wer sich mit Softwareentwicklung auskennt, ist längst nicht mehr nur bei Softwareunternehmen wie Microsoft, SAP und Salesforce gefragt. Ob Versicherungen, Banken, Transport, Industrie oder Handelsunternehmen, so gut wie alle Unternehmen digitalisieren ihre Geschäftsprozesse, ihren Vertrieb, die Produktion und auch die Geschäftsmodelle. Dafür benötigen sie zwingend das entsprechende Know-how und folglich die richtigen Mitarbeiter.

Softwareentwickler:innen, Softwarearchitekt:innen, Testing-Fachkräfte, DevOps-Spezialist:innen, Cloud-Expert:innen, IT-Administrator:innen, Data Scientists, IT-Projektmanager:innen, IT-Sicherheitsexpert:innen, IT-Service-Berater:innen sowie Fachleute für Künstliche Intelligenz, Industrie 4.0 und viele weitere Bereiche sind besonders begehrt.

Fachkräftemangel: 86.000 IT-Stellen unbesetzt

Laut einer jährlichen Erhebung des IT-Branchenverbands Bitkom hat sich der Mangel an IT-Spezialist:innen in Deutschland durch die Corona-Krise zwar vorübergehend verringert, liegt aber nach wie vor auf dem zweithöchsten Wert seit Erhebungsbeginn. Im vorletzten Jahr war ein historischer Höchststand von 124.000 unbesetzten Stellen zu verzeichnen. Ende 2020 waren quer durch alle Branchen 86.000 Stellen für IT-Fachkräfte vakant [1] – gedämpft durch die Corona-Krise, so sei angemerkt.

Selbst während der Pandemie leiden Unternehmen unter akutem Mangel an IT-Fachkräften. Die Krise hat aber vielerorts nicht bloß Defizite bei der Digitalisierung aufgezeigt, sondern im letzten Jahr einen „Digitalisierungsaufschub“ ausgelöst, der die Nachfrage nach Informatikerinnen und Informatikern schon bald wieder ansteigen lassen und den Mangel womöglich weiter verschärfen wird (s. Abb. 1).

Um dem Fachkräftemangel zu begegnen, sind laut Bitkom ergänzend zu einer effizienteren Aus- und Weiterbildung die Stärkung von Frauen in der IT und die Förderung qualifizierter Zuwanderung gefragt. Vor allem die qualifizierte Zuwanderung ist eine kontroverse Angelegenheit, denn knapp neun von zehn Unternehmen (87 Prozent) in Deutschland sehen Deutschkenntnisse nach wie vor als ein Muss, was die Lage für viele Unternehmen zusätzlich befeuert. Auch wenn der IT-Fachkräftemangel nicht nur ein deutsches Phänomen ist, könnte die Arbeitgeberseite diesem durch mehr Offenheit bei der Suche nach IT-Kräften entgegenwirken.

Die Suche nach den richtigen Mitarbeitern ist hart. Viele Unternehmen stehen sich jedoch selbst im Weg, weil sie unrealistische Erwartungen haben, die vom Arbeitsmarkt nicht erfüllt werden können, und so bleiben dann viele IT-Stellen für lange Zeit unbesetzt. Jene Organisationen, die im IT-Recruiting flexibler agieren und die eigenen Anstellungskriterien an neue Mitarbeiter etwas lockern, werden daraus mittel- und langfristig auch einen Wettbewerbsvorteil ziehen können. Es liegt im Interesse der Arbeitgeber:innen, neue Mitarbeiter:in-

**>> Deutschland fehlen IT-Fachkräfte:
Anzahl zu besetzender IT-Stellen in der
deutschen Wirtschaft (Abb. 1)**

Quelle: Statista

Deutschland fehlen IT-Experten

Anzahl zu besetzender IT-Stellen in der deutschen Wirtschaft

■ Offene Stellen (in 1.000) ● Unternehmen, die einen Mangel an IT-Kräften feststellen (in %)*



* Basis: 853 Geschäftsführer und Personalleiter von Unternehmen ab 3 Mitarbeitern (exkl. Landwirtschaft und öffentl. Sektor); 2020
Quelle: Bitkom



nen schnell produktiv einsetzen zu können. Daher ist es aufgrund der Arbeitsmarktlage oftmals sinnvoller, auch Bewerber:innen in Betracht zu ziehen, die zwar nicht alle formellen Kriterien erfüllen, diese aber im Unternehmen erlernen können, insbesondere wenn bereits eine entsprechende praktische Vorerfahrung besteht. Das ist immer noch besser, als eine Stelle monate- oder jahrelang unbesetzt zu lassen.

Der anhaltende Engpass auf dem Arbeitsmarkt macht aber nicht nur bestehende IT-Expert:innen zu gefragten Kandidat:innen. Er eröffnet auch Türen für Bewerber:innen, die bislang keine klassische IT-Laufbahn durchschritten hatten. Diese begann lange Zeit für die meisten mit einer Fachausbildung – etwa zum Fachinformatiker – oder mit einem Studium. Nach einem Bericht der Bundesagentur für Arbeit über MINT-Berufe trifft das derzeit auf 85 Prozent der sozialversicherungspflichtig beschäftigten IT-Fachkräfte in Deutschland zu [2].

Neue Rollen ohne "Diplomzwang"

Doch der Bedarf an Fachkräften ist so rasant gewachsen, dass dieser „Diplomzwang“ immer häufiger entfällt und ein

Teil der IT-Fachkräfte heute ihren Job ganz ohne spezifische Fachausbildung ausüben. In Zukunft dürften es noch mehr werden – auch weil neue und derzeit noch unbekannte Berufsfelder entstehen, deren Inhalte noch gar nicht den Weg auf die Lehrpläne der klassischen Ausbildungsstätten gefunden haben. Da sich der gesamte IT-Bereich ständig weiterentwickelt und immer mehr Möglichkeiten bietet, bestehen

NOAH

»ENTWICKLER SIND DIE BESSEREN TESTER?
BEI UNS KANNST DU ES BEWEISEN.«

imbus

Schlechte Software nervt. Unser Job ist es, sie besser zu machen. Bei imbus dreht sich alles um das professionelle Testen und die intelligente Qualitätssicherung von Softwareprodukten. Wir suchen Entwickler, die Software richtig gut machen wollen. Bei uns kannst du als **Junior Softwareentwickler Java oder C# (m/w/d)** oder als Berufseinsteiger in unserem „JumpStart“-Programm loslegen. Oder schau dich einfach auf unserer Jobbörse um.



www.imbus.de

für viele neu geschaffene Rollen oftmals noch gar keine formalen Einstellungskriterien, die es zwingend zu erfüllen gibt. Aufgrund dieser rasanten Entwicklung wird praktische Erfahrung zunehmend wichtiger.

Schon lange verlassen sich die Studierenden nicht bloß auf die Ausbildung, sondern arbeiten nebenher, absolvieren Praktika und setzen eigene Projekte um. Das empfiehlt sich generell für Berufseinsteiger:innen – unabhängig davon, ob man den klassischen Ausbildungsweg oder den Quereinstieg gewählt hat.

Lebenslanges Lernen, da Inhalte sich rasch ändern

Softwareentwickler:innen auf der ganzen Welt erleben Ähnliches im eigenen Alltag. In der IT entwickelt sich ständig viel Neues. Das Studienwissen bietet sicherlich eine solide Basis, um grundlegende Konzepte zu verstehen, oft werden große Teile der praxisbezogenen Ausbildungsinhalte aber schon nach wenigen Jahren obsolet. IT-Fachkräfte müssen ihre Fähigkeiten daher laufend erneuern und in vielen Fällen eigenständig erlernen und vertiefen.

Jobanwärter:innen sollten im Blick behalten, welche spezifischen Fachkenntnisse bei den Unternehmen gerade gefragt sind, damit sie nach ihrem Abschluss direkt einsatzbereit sind. Welche fachlichen Kompetenzen Berufseinsteiger:innen dazu mitbringen müssen, hängt stark vom späteren Einsatzgebiet ab.

Alternative Bildungswege

Unbestritten ist jedenfalls, dass die traditionellen Ausbildungsstätten – Universitäten und Fachhochschulen – den Bedarf der Wirtschaft an IT-Fachkräften schon lange nicht mehr decken können. „Der Bedarf ist enorm und er wird die kommenden Jahre auch nicht abreißen“, sagt Sead Ahmetović, CEO der Jobplattform WeAreDevelopers. „Das akademische System allein kann dieser Nachfrage nicht mehr gerecht werden, denn ein Studium dauert nun mal mindestens drei bis fünf Jahre.“

Das verlangt sowohl ein Umdenken der Personalverantwortlichen in den Unternehmen als auch eine Anpassung bestehender Anforderungskriterien. Weist ein Kandidat oder eine Kandidatin beispielsweise hohe autodidaktische Fähigkeiten auf, ohne ein dezidiertes Studium vorweisen zu können, beweist dies nicht bloß eine ordentliche Portion an Motivation, sondern stellt auch ein wertvolles Asset für neue

Herausforderungen dar. Solche Soft Skills werden in Zukunft stetig gefragter sein als formal bestätigte Hard Skills.

Über die Bedeutung alternativer Bildungswege diskutieren die Expert:innen kontrovers. Beispielsweise funktionieren Coding-Bootcamps vor allem für Quereinsteiger:innen, die einen Karrierewechsel anstreben, aber bereits Berührungspunkte mit Informatik hatten und auch ausreichend Interesse besitzen. Ergebnisse einer US-amerikanischen Absolventenstudie von Course Report bestätigen dies: Der klassische Bootcamper ist 31 Jahre alt, hat sechs Jahre Berufserfahrung und in der Regel einen Bachelorabschluss [3]. Der übliche Weg in ein Coding-Bootcamp läuft also häufig über ein Fach- oder Hochschulstudium und ein paar Jahre im Arbeitsleben. Ganz so einfach und voraussetzungsfrei ist der Wechsel also dann doch nicht.

Thomas Pamminger, CPO und Co-Founder bei WeAreDevelopers, beobachtet solche Werdegänge immer wieder:

„Weil der Bedarf so groß ist, ist vielen Unternehmen das tatsächliche Können und der Cultural Fit von Bewerbern wichtiger als deren formale Qualifikation.“ Wichtig sei dabei, dass Quereinsteiger echte Projekte und Motivation vorweisen können – sei es, dass sie in ihrer Freizeit schon eine eigene App programmiert, ihr Können in Open-Source-Projekten bewiesen haben, aber auch

andere, „unübliche“ Bildungswege eingeschlagen haben. Eine solche Erfolgsgeschichte gibt es auch in den eigenen Reihen von WeAreDevelopers. Einen Berufseinstieg in die IT ohne spezifische Ausbildung hat Sabina Dego geschafft. Dass sie keinen Informatik-Abschluss hatte, störte ihren zukünftigen Arbeitgeber überhaupt nicht. Erste Vorkenntnisse halfen dabei, und sie hatte keinerlei Berührungsängste. Nach einem Erstgespräch und anschließendem Kennenlernen des Teams gab es ein Jobangebot als Quality Assurance Engineer. „Gerade die Bereiche Quality Assurance (QA) und Software Testing bieten gute Möglichkeiten, erste Gehversuche in IT-Teams zu machen, um deren Denk- und Arbeitsweisen kennenzulernen und zu verstehen“, so Pamminger.

Unternehmen locken Quereinsteiger

Manche Unternehmen richten sich sogar mit eigenen Initiativen und Formaten an Quereinsteiger – so etwa Volkswagen. Auch dort wächst der Bedarf an IT-Fachkräften. Nicht nur, weil in jedem Auto heute auch ein Computer steckt. Darüber hinaus helfen Big-Data-Expert:innen, die Marktnachfrage zu analysieren, und Machine-Learning-Spezialist:innen, bestimmte Logistikprozesse zu optimieren. Spieleent-

wickler:innen gestalten virtuelle Entwicklungsumgebungen für den Bau neuer Modelle, Cloud-Expert:innen vernetzen die Produktion. Um sich als Technologiekonzern zu positionieren, hat Volkswagen die „Fakultät 73“ ins Leben gerufen – ein zweijähriges Programm, bei dem Quereinsteiger zu Entwicklern ausgebildet werden. Die Teilnehmer und Teilnehmerinnen lernen zunächst Grundlagen von Programmiersprachen wie Python und Java; später lernen sie dann verschiedene IT-Aufgaben im Konzern kennen. Neben solchen Eigeninitiativen bei großen Unternehmen bieten darüber hinaus auch Start-ups gute Chancen für einen Quereinstieg. Gerade neu gegründete Unternehmen können oftmals nicht mit den finanziellen Kompensationsmöglichkeiten von etablierten Konzernen mithalten und müssen daher in anderen Bereichen punkten. Die Einstellungsverfahren sind zumeist kürzer und das Aufgabengebiet jedes Einzelnen ist breiter gefächert, somit auch abwechslungsreicher und für Quereinsteiger mit mehreren Fähigkeiten und entsprechender Experimentierfreudigkeit bestens geeignet.

Digitalisierungsschub erwünscht

Während der Fachkräftemangel wohl in den nächsten Jahren wieder steigen wird, braucht es die richtigen staatlichen und privaten Initiativen, die in der Wirtschaft die nötige Anerkennung finden. Vor allem sollten sich Unternehmen selbst an der Nase nehmen und ein Mindset-Update vollziehen. Möglicherweise sind die Talente nicht dort, wo sich Unternehmen diese wünschen. Aber die Talente sind da. Ein weiterer „Digitalisierungsaufschub“ – wie derzeit durch die Corona-Krise – darf keine Option sein.

Quellen

- [1] Erhebung des IT-Branchenverbands Bitkom: bitkom.org/Presse/Presseinformation/86000-offene-Stellen-fuer-IT-Fachkraefte
- [2] Bericht der Agentur für Arbeit über MINT-Berufe: statistik.arbeitsagentur.de/DE/Statischer-Content/Statistiken/Themen-im-Fokus/Berufe/Generische-Publikationen/Broschuere-MINT.pdf
- [3] Coding Bootcamps 2020 laut Course Report: course-report.com/2020-guide-to-coding-bootcamps-by-course-report.pdf



Sead Ahmetović

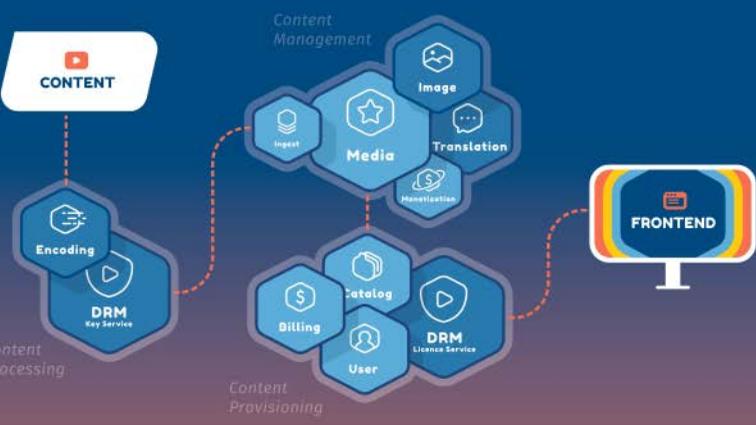
ist CEO und Co-Founder von WeAre-Developers, einem der führenden Karriere-Portale für Software Developer und IT-Fachkräfte in Europa.



Thomas Limbüchler

ist Developer Relations Strategist bei WeAreDevelopers und beschäftigt sich unter anderem mit Veränderungen, die gerade und in der Zukunft in der Karriere von Software Developern passieren.

Die Heise Medien sind seit 2019 mit 10 Prozent an WeAreDevelopers beteiligt.



Die Bausteine für Ihr Media-Backend

Egal, ob Sie vordefinierte Services nutzen oder eigene entwickeln möchten, Axinom Mosaic hilft Ihnen, Ihr Media-Backend zu vereinfachen und zu zentralisieren.

BUILT ON

GraphQL
node.js
RabbitMQ
React
Sentry
Typescript
Microservices



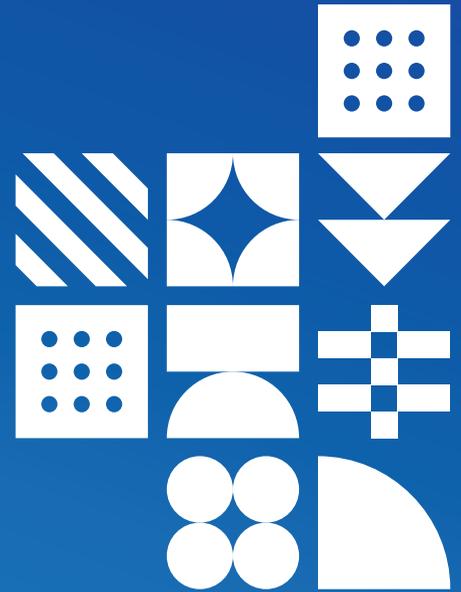
Mehr Infos gibt's auf axinom.com/mosaic




> Volltextsuche mit Spring-Boot

Bernhard Jungwirth

Mit Hibernate Search lässt sich das Datenmodell von Spring-Boot-Anwendungen mit Annotationen und ohne Server einfach um eine Volltext-Suchfunktion erweitern.



Die einfache Auffindbarkeit von Daten wird immer wichtiger. Anwender nutzen eine Suche in Onlineshops oder im Betriebssystem zum Auffinden von Dateien, Apps oder Einstellungen. All diese Suchfunktionen arbeiten mit einem Suchindex, dessen Verwaltung und Aktualisierung im Hintergrund geschieht. Dadurch ist es möglich, priorisierte Suchvorschläge oder eine Klassifikation der Treffer anzuzeigen, um die Suche weiter zu verfeinern. Von diesen Vorteilen können auch eigene Softwareprojekte profitieren und Benutzern die gewohnte Funktionalität anbieten. Je nachdem, welche Datenmenge zu verarbeiten ist, ist es möglich, die Programmibliothek zur Volltextsuche direkt zu integrieren oder den Index an einen Elasticsearch-Server auszulagern.

Hibernate: ein quelloffenes ORM-Framework

Relationale Datenbanken sind noch immer eine zentrale Komponente in modernen IT-Systemen. Objektorientierte Programmiersprachen wie Java verwenden Object-Relational Mapper (ORM), um das Datenmodell der Klassen des Programmcodes in Tabellen der Datenbank zu erstellen. Anwendungen im Java-Framework Spring Boot verwenden als Standardimplementierung von ORM Hibernate [1]. Hibernate ist ein Open-Source ORM-Framework, das in Java implementiert ist und von Red Hat gesponsert wird. Spring Boot vereinfacht die Entwicklung von Webanwendungen mit Java

dadurch, dass eine Konfiguration per Konvention bereits vorgegeben ist. Von Hibernate gibt es neben dem bekannten ORM-Projekt noch weitere Teilprojekte, darunter Hibernate Search. Dieses Projekt erstellt automatisch einen Suchindex zu den Daten, die Hibernate ORM verwaltet. Hibernate Search speichert den Suchindex direkt mit Apache Lucene oder mit dem Suchserver Elasticsearch. Ersteres ist eine Programm-bibliothek zur Volltextsuche und ebenfalls Open Source. Elasticsearch ist ein eigener Suchserver und verwendet als Basis wiederum Apache Lucene. Die Anforderungen der Hochverfügbarkeit und Lastverteilung erfordern den Einsatz von Elasticsearch als eigenem Suchserver. Sind diese Anforderungen nicht erfüllt, genügt die Programm-bibliothek Apache Lucene. Das Beispielprojekt stellt die Verwendung dieser Bibliothek mit Hibernate Search vor. Hier ist es von Vorteil, dass keine eigene Infrastruktur zu betreiben ist.

Neuerungen in Hibernate Search 6

Die Version 6.0 von Hibernate Search ist im Dezember 2020 erschienen und umfasst viele Neuerungen. Bis zur Version 5 war Apache Lucene fester Bestandteil von Hibernate Search, aber mit Version 6.0 ist die API überarbeitet und die Such-Backends sind abstrahiert. Somit können Entwickler Apache Lucene und Elasticsearch gleichwertig verwenden und Aktualisierungen schneller einbinden. Zurzeit ist die aktuelle Version 8.7 von Apache Lucene in den Abhängigkeiten

referenziert, und der Suchserver Elasticsearch lässt sich in den Versionen 5.6, 6.8 oder 7.10 anbinden.

Durch die Abstrahierung der Such-Backends ist auch die Search-DSL (Domain Specific Language) neu entwickelt, da sie ebenfalls auf Apache-Lucene-Abfragen aufbaute. Diese Abfragesprache ist nun typischer, aussagekräftiger und sie umfasst auch Lambda-Ausdrücke. Weitere Details finden sich im Blog von Hibernate [2].

Für die Umstellung eines Projekts von Hibernate Search 5 auf Version 6 gibt es einen eigenen Migrationsleitfaden, da durch die vielen Änderungen beim Versionswechsel einiges anzupassen ist [3].

Spring-Boot-Anwendung für Gesetze und Paragraphen

Die Beispielanwendung ist eine Spring-Boot-Anwendung, die Gesetze und einzelne Paragraphen speichert und als REST-API anbietet. Das Datenmodell dazu besteht aus der Entität `LegalDocument`, die die allgemeinen Daten des Gesetzes umfasst, und darin enthalten ist eine Liste von `Articles` mit dem eigentlichen Gesetzestext des Paragraphen. Der Datenzugriff erfolgt über ein Spring-Data-Repository [4], das für jede En-

Listing 1: Maven-Abhängigkeiten

```
<dependency>
  <groupId>org.hibernate.search</groupId>
  <artifactId>hibernate-search-engine</artifactId>
  <version>${hibernate.search.version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate.search</groupId>
  <artifactId>hibernate-search-mapper-orm</artifactId>
  <version>${hibernate.search.version}</version>
</dependency>
<dependency>
  <groupId>org.hibernate.search</groupId>
  <artifactId>hibernate-search-backend-lucene</artifactId>
  <version>${hibernate.search.version}</version>
</dependency>
```

Listing 2: Konfiguration des Suchindex

```
hibernate.search.backend.directory.type = local-filesystem
hibernate.search.backend.directory.root = C://Tools//Search//Playground
hibernate.search.backend.analysis.configurer = LegalDocumentAnalysisConfigurer
```

tität erstellt wird. Zusätzlich ist Spring-Data-REST [5] in das Projekt eingebunden und exponiert das Datenmodell automatisch als REST-Endpunkt. Über diese Schnittstelle sind die Daten bereits abrufbar. Der vollständige Quelltext zum Projekt ist auf GitHub verfügbar [6].

Anzeige

daskeyboard®

Smart, stylisch und effizient Grenzenlose Möglichkeiten für Programmierer und Vielschreiber

<https://das-keyboard-eu.myshopify.com/>
<https://www.daskeyboard.com/de/>



DK 5QS: Smart, intelligent und individuell

Direkte Informationswiedergabe vom Internet zur Tastatur

- Integrierte „Das Keyboard Q“-Technologie ermöglicht intelligente RGB-Benachrichtigungen
- Ultrahelle RGB-Beleuchtung
- Integrierter RGB-Profil-Editor mit vorkonfigurierten Profilen
- Integrierter **Makro-Editor** zur Wiedergabe von aufgezeichneten Tastenanschlägen
- Gamingmodus mit Full-NKRO
- Mechanische Schüsselschalter mit 100M Zyklen
- Abnehmbare Handballenauflage
- **Q-Lautstärkeregl** ändert Lautstärk & Vorschau von Benachrichtigungen
- Eloxierte Aluminium-Oberfläche
- Unterstützung für Windows und Ubuntu
- Integrierter Mac-Modus.

199,00 €
(inkl. gesetzl. MwSt.)



DK4 Professional für Mac: Hochtechnisch und stylisch

Fühl- und hörbare Klickmomente

- Cherry mechanische Tastenschalter mit Goldkontakten
- Dedizierte Mediensteuerungen mit besonders **großem Lautstärkeregl**
- **SuperSpeed USB 3.0 Hub mit zwei Anschlüssen** – bis zu 5 Gbit/s, zehnmal so schnell wie USB 2.0
- **Instant-Sleep-Knopf**, zur Aktivierung des Ruhezustandes und Senkung des Energieverbrauches
- Mac OS spezifische Tastenfunktionen
- Gelaserte Tastenbeschriftung
- N-Key Rollover (NKRO) über USB
- Magnetische Fußleiste mit Linealfunktion
- Designed für Mac OSX
- Zwei Ausführungen:
Cherry MX „blau“ und „braun“

149,00 €
(inkl. gesetzl. MwSt.)

Damit ist das Datenmodell fertig und die Einträge lassen sich lesen, erstellen und bearbeiten. Als Nächstes kommt der Suchindex für die Volltextsuche in den Paragraphen hinzu.

Die Abhängigkeiten lassen sich über Maven einbinden (s. Listing 1). Auf jeden Fall nötig sind die Such-Engine und die Mapper-Bibliothek, die das Datenmodell mit dem Suchindex verbindet. Zur direkten Speicherung des Suchindex mit Apache Lucene ist dieses als Backend noch dabei.

Die Konfiguration für Hibernate Search befindet sich in der Datei `hibernate.properties` im Ressourcen-Ordner `src/main/resources/`. Die Einstellung für die Eigenschaft `directory.type` ist auf ein lokales Dateisystem gesetzt und zusätzlich ist noch der Pfad angegeben, wo das Programm den Suchindex ablegt (s. Listing 2).

Das Entity-Modell erweitern

Die Java-Klassen für das Datenmodell sind mit der Annotation `@Entity` und die Attribute mit `@Column` erweitert, damit Hibernate ORM sie erkennt. Zur Aufnahme der Daten in den Suchindex kommt die Annotation `@Indexed` zur Klasse hinzu. Damit das Programm die einzelnen Felder in den Index übernimmt, erhalten sie eine der `@*Field`-Annotationen. Das Feld `title` ist als `@FullTextField` annotiert. Das bedeutet, dass für dieses Feld eine Textanalyse ausgeführt wird. Der Text wird in einzelne Tokens gesplittet und anschließend normalisiert. Die Details der Textanalyse folgen weiter unten in diesem Artikel. Für die Attribute Kurztitel (`titleShort`) und Abkürzung (`abbreviation`) wird die `@KeywordField`-Annotation verwendet. Dabei wird der Inhalt des Feldes nicht analysiert, sondern nur normalisiert. Zusätzlich ist beim Attribut `abbreviation` die Eigenschaft `aggregable` aktiviert, damit lässt

Listing 3: LegalDocument.java – Gesetz mit einer Liste an Paragraphen

```
@Entity
@Indexed
public class LegalDocument extends AbstractEntity {

    @FullTextField
    @Column(length = 1024)
    private String title;

    @KeywordField(name = "titleShort")
    private String titleShort;

    @KeywordField(aggregable = Aggregable.YES)
    private String abbreviation;

    @OneToMany(mappedBy = "document", cascade = CascadeType.ALL)
    @JsonBackReference
    private List<Article> articles;

    // getter and setter
}
```

sich eine Aggregation auf dieses Attribut ausführen, wie Listing 3 zeigt.

Die `Article`-Klasse ist ebenfalls mit der `@Indexed`-Annotation versehen. Der Titel des Artikels (`title`) und der eigentliche Gesetzestext (`text`) sind wiederum als `@FullTextField` annotiert und werden somit für den Suchindex ausgewertet. Jeder Paragraph verfügt über ein Datum des Inkrafttretens und Außerkrafttretens. Diese Datumsfelder sind als `@GenericField` annotiert. Das `@GenericField` unterstützt verschiedene Typen in Java, und bei der Suche wird ein exakter Vergleich gezogen. Die Suchabfragen laufen im Datenmodell auf der Ebene der Artikel. Für den Zugriff auf den Suchindex des übergeordneten Gesetzes wird das `LegalDocument` mit `@IndexedEmbedded` annotiert. Zur Suche nach allen Artikeln eines Gesetzes anhand der Abkürzung können Entwickler den Punktoperator `document.abbreviation` verwenden (s. Listing 4).

Aufbau des Suchindex

Jede Änderung einer Entität über die Hibernate Session in der Anwendung aktualisiert den Suchindex automatisch. Beim Starten des Beispielprojekts überträgt ein SQL-Skript die Daten an die Datenbank. Dazu genügt es, in Spring Boot ein Skript mit dem Namen `data.sql` in den Ressourcen-Ordner zu legen. Damit der Index sowohl die neu ergänzten

Listing 4: Article.java – ein Paragraph

```
@Entity
@Indexed
public class Article extends AbstractEntity {

    @GenericField(name = "number", sortable = Sortable.YES)
    private String number;

    private String eli;

    @FullTextField(name = "title", analyzer = "german")
    @GenericField(name = "title_sort", sortable = Sortable.YES)
    private String title;

    @FullTextField(name = "text", analyzer = "german")
    @Column(length = 4096)
    private String text;

    @GenericField
    @Column(columnDefinition = "DATE")
    private LocalDate effectiveDate;

    @GenericField
    @Column(columnDefinition = "DATE", nullable = true)
    private LocalDate expireDate;

    @IndexedEmbedded
    @ManyToOne
    @JoinColumn(name = "document_id", nullable = false)
    @JsonManagedReference
    private LegalDocument document;

    // getter and setter
}
```

Impressum We Are Developers!

Redaktion: iX | heise Developer
Telefon: 0511 5232-387, **E-Mail:** post@ix.de

Herausgeber: Ansgar Heise

Chefredakteur: Dr. Oliver Diedrich

Konzeption und redaktionelle Leitung: Silke Hahn

Autoren dieser Ausgabe:

Sead Ahmetović, Bernhard Jungwirth, Nils Kasseckert, Thomas Limbüchler, Simon Mennig, Timo Zander

DTP-Produktion:

Lisa Hemmerling Heise Medienwerk, Rostock

Korrektorat:

Marei Stade, Heise Medienwerk, Rostock

Titelbild:

© Adobe Stock; Montage: Lisa Hemmerling

Verlag

Heise Medien GmbH & Co. KG,
 Postfach 61 04 07, 30604 Hannover; Karl-Wiechert-Allee 10, 30625 Hannover;
 Telefon: 0511 5352-0, Telefax: 0511 5352-129

Geschäftsführer:

Ansgar Heise, Dr. Alfons Schröder

Mitglieder der Geschäftsleitung:

Beate Gerold, Jörg Mühle

Verlagsleiter:

Dr. Alfons Schröder

Anzeigenleitung (verantwortlich für den Anzeigenteil):

Michael Hanke (-167), E-Mail: michael.hanke@heise.de,
 www.heise.de/mediadaten/ix

Leiter Vertrieb und Marketing:

André Lux

Druck:

Dierichs Druck + Media GmbH & Co. KG, Frankfurter Straße 168, 34121 Kassel

Eine Haftung für die Richtigkeit der Veröffentlichungen kann trotz sorgfältiger Prüfung durch die Redaktion vom Herausgeber nicht übernommen werden. Kein Teil dieser Publikation darf ohne ausdrückliche schriftliche Genehmigung des Verlages verbreitet werden; das schließt ausdrücklich auch die Veröffentlichung auf Websites ein.

Printed in Germany

©Copyright by Heise Medien GmbH & Co. KG

Inserenten

Axinom GmbH	Fürth	S. 9, 33	Infineon Technologies AG	Neubiberg	S. 44
Bundesrechenzentrum	A – Wien	S. 21	InterSystems GmbH	Darmstadt	S. 15
Carl Zeiss AG	Oberkochen	S. 5	Metadot UG Das Keyboard	Hamburg	S. 35
Dataport	Altenholz	S. 19	Miro	USA - San Francisco	S. 7
Datev eG	Nürnberg	S. 23	siggate GmbH	Düsseldorf	S. 22
DLR Deutsches Zentrum für			Volkswagen Group Services GmbH	Wolfsburg	S. 25
Luft- und Raumfahrt e.V.	Weßling	S. 13	WERTGARANTIE Beteiligungen GmbH	Hannover	S. 29
FERCHAU GmbH	Gummersbach	S. 27	willy.tel GmbH	Hamburg	S. 11
imbus AG	Möhrendorf	S. 31			

Die hier abgedruckten Seitenzahlen sind nicht verbindlich. Redaktionelle Gründe können Änderungen erforderlich machen.

Aufrüsten im Homeoffice

NEU




Heft + PDF
mit 29%
Rabatt

Generell portofreie Lieferung für Heise Medien- oder Maker Media Zeitschriften-Abonnenten oder ab einem Einkaufswert von 20 €.
Nur solange der Vorrat reicht. Preisänderungen vorbehalten.



shop.heise.de/ct-hardware-ho21 >

Listing 5: Den Indexer ausführen

```
@PostConstruct
public void initialize() {
    SearchSession searchSession = Search
        .session(entityManager.getEntityManagerFactory().createEntityManager());
    MassIndexer indexer = searchSession
        .massIndexer(Article.class)
        .threadsToLoadObjects(2);
    try {
        indexer.startAndWait();
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
}
```

Daten als auch Einträge, die bereits vorhanden waren, übernimmt, gilt es, einen `MassIndexer` zu erstellen und zu initialisieren.

Als Parameter benötigt der `MassIndexer` die Klassen, die in den Index aufzunehmen sind. Im Beispielprojekt wird nur die Klasse `Article` übergeben, da das `LegalDocument` nicht explizit in den Suchindex kommt. Zusätzlich ist noch zu konfigurieren, wie viele Threads für den Indizierungsvorgang zu verwenden sind (s. Listing 5).

Suchabfragen schreiben

Die Implementierung einer einfachen Suche nach Titel und Abkürzung eines Gesetzes ist in Listing 6 abgebildet. Per `EntityManager` lässt sich eine Hibernate Search Session erstellen und mit ihr die Suche ausführen. Für den Zugriff auf die eingebetteten Felder vom `LegalDocument` kommt als Prefix der Name des Attributs (`document`) zum Einsatz. Der Lambda-Ausdruck übergibt der `where`-Methode das Suchprädikat, und der Suchbegriff ist im `matching`-Aufruf angegeben. Der Aufruf der `fetch`-Methode führt die Suche aus und liefert die Daten als `SearchResult` zurück. Das `SearchResult` umfasst in `result.hits()` die Suchergebnisse, und der Befehl `result.total().hitCount()` fragt die Gesamtanzahl der Suchtreffer ab. Die Steuerung der Paginierung erfolgt über die Parameter `Offset` und `Limit` der `fetch`-Methode (s. Listing 7).

Die Sortierung geschieht per Default nach der Relevanz der Suchergebnisse. Es ist auch möglich, andere Sortierreihenfolgen einzustellen. In der Felddefinition für den Suchindex muss dafür die Sortierbarkeit (`sortable`) aktiviert sein. Im Beispiel ist

Listing 7: Einfache Suche nach Titel und Abkürzung mit Paginierung

```
SearchResult<Article> result = searchSession.search(Article.class)
    .where(f -> f.match()
        .fields("document.title", "document.abbreviation")
        .matching(request.keyword))
    .fetch(0,20);
```

Listing 6: Einfache Suche nach Titel und Abkürzung

```
SearchSession searchSession = Search.session(entityManager);
SearchResult<Article> result = searchSession.search(Article.class)
    .where(f -> f.match()
        .fields("document.title", "document.abbreviation")
        .matching(request.keyword))
    .fetch(20);
return new SearchResponse<>(result.hits(),
    new PageMetadata(20, 0, result.total().hitCount(), null);
```

das beim Feld `number` und `title` der Fall. Ein `@FulltextField` ist nicht sortierbar, da der Text in einzelne Wörter zerteilt ist, aber es lassen sich mehrere Indexfelder für ein Attribut einer Klasse definieren. Listing 8 zeigt zwei Varianten zur Steuerung der Sortierung: Entweder ist das Feld `fix` im Code definiert, oder über eine Komposition lassen sich die Sortierfelder dynamisch hinzufügen. Zur Steuerung der Paginierung und Sortierung über eine REST-Schnittstelle gibt es im Spring-Data-Projekt ein `Pageable`-Interface. Die Suche in Listing 8 verwendet das Objekt, um die Parameter für die Paginierung und die Sortierung der Suchabfrage zu übergeben.

Facettensuche bei Hibernate Search

In vielen Online-Shops sind nach einer Suche passende Kategorien und Preisbereiche aufgelistet. Die Bezeichnung dafür ist Facettensuche und Hibernate Search bietet sie ebenfalls an. Durch die Facettensuche lässt sich das Suchergebnis einfach einschränken und verfeinern, beispielsweise auf eine bestimmte Marke oder einen Preisbereich. Bis Version 5 von Hibernate Search hieß diese Funktion auch noch `Faceting`, mit

Listing 8: Einfache Suche mit einer Sortierung

```
SearchResult<Article> result = searchSession.search(Article.class)
    .where(f -> f.match().fields("document.title", "document.abbreviation")
        .matching(request.keyword))

    // Variante A
    .sort(f -> f.field("title_sort").desc())

    // Variante B
    .sort(f -> f.composite(b -> {
        if(request.page != null) {
            request.page.getSort().get().forEach(s -> {
                b.add(f.field(s.getProperty()).order(s.isAscending() ? SortOrder.ASC : SortOrder.DESC));
            });
        }
    })))
    .fetch(
        request.page.getPageNumber() * request.page.getPageSize(),
        request.page.getPageSize()
    );
```



Finde die besten Developer-Jobs in deiner Region!



Entdecke Top-Entwicklerjobs, die deinen Anforderungen entsprechen – basierend auf Tech Stack, Location und Gehaltsvorstellungen.

Kein Spam.
Keine Recruiter.
Kein Bullshit.

Senior Fullstack Developer

Codalize Berlin, Germany

€ 82k Java Angular Typescript

Senior Frontend Developer

devopify Vienna, Austria

€ 67k Javascript React HTML5

Registriere dich auf Europas führender Job-Plattform für Developer:
wearedevelopers.com



28. 06. – 01. 07.

WEAREDEVELOPERS WORLD CONGRESS 2021

GOES DIGITAL

Hol dir dein Ticket:
wearedevelopers.com/events/world-congress



Listing 9: Einfache Suche mit einer Aggregation

```

AggregationKey<Map<String, Long>> countByAbbreviation = AggregationKey.
of("countByAbbreviation");

SearchResult<Article> result = searchSession.search(Article.class)
    .where(f -> f.match().fields("title", "document.title", "document.abbrevi-
ation"))
    .matching(request.keyword)
    .aggregation(countByAbbreviation, f -> f.terms()
    .field("document.abbreviation", String.class))
    .fetch(100);

List<SearchAggregation> facetAbbreviation = result.aggregation(countByAbbrevi-
ation)
    .entrySet()
    .stream()
    .map(e -> new SearchAggregation(e.getKey(), e.getValue()))
    .collect(Collectors.toList());

return new SearchResponse<>(
    result.hits(),
    new PageMetadata(100, 0, result.total().hitCount()),
    facetAbbreviation);

```

Version 6 wurde sie umbenannt in eine allgemeine Aggregation-DSL. Im Beispiel basiert die Aggregation auf der Abkürzung eines Gesetzes. Dadurch bekommt man einen schnellen Überblick, in wie vielen Paragraphen eines Gesetzes sich der Suchbegriff auffinden lässt. Die Voraussetzung ist ähnlich wie bei der Sortierung: Das Feld muss in der Entität als aggregierbar definiert sein, und ein FullTextField lässt sich nicht aggregieren.

Für eine Suche gilt es, zuerst einen AggregationKey mit einem Namen zu definieren. Die Suche selbst ist aufgebaut wie im vorigen Beispiel, allerdings ist die Aggregation zum Aufruf nach der matching-Methode hinzuzufügen. Nach der Ausführung der Suche lässt sich die Aggregation über das Ergebnis auslesen. Die Aggregation ist ein Map-Objekt, wobei der Schlüssel (Key) das Objekt ist, nach dem aggregiert ist. Im Beispiel (s. Listing 9) handelt es sich um einen String, und der Wert (Value) enthält die Anzahl der Treffer. Die Anwendung der Aggregation kann sich wahlweise auf diskrete Begriffe oder auf kontinuierliche Bereiche beziehen, wie es für Preisbereiche erforderlich ist.

>> Aufruf der Suche über die REST-Schnittstelle mit dem Suchbegriff – und das Ergebnis als JSON-Objekt (Abb. 1)

Listing 10: Textanalyse für deutschsprachige Inhalte

```

@Component("LegalDocumentAnalysisConfigurer")
public class MyLegalDocumentAnalysisConfigurer implements LuceneAnalysisCon-
figurer {

    @Override
    public void configure(LuceneAnalysisConfigurationContext context) {
        context.analyzer("german").custom()
            .tokenizer(
                StandardTokenizerFactory.class)
            .tokenFilter(
                LowerCaseFilterFactory.class)
            .tokenFilter(
                SnowballPorterFilterFactory.class)
                .param("language", "German")
            .tokenFilter(ASCIIFoldingFilterFactory.class);
    }
}

```

Textanalyse für die Volltextsuche

Eine Textanalyse ermöglicht eine effizientere Suche. Beim Indizieren erfolgt sie auf die Daten und bei der Suche entsprechend auf die Suchbegriffe, und zwar in drei Schritten.

Der erste Schritt ist dabei die Anwendung von Zeichenfiltern (character filters), die den Eingabetext von nicht benötigten Zeichen bereinigen. Beispielsweise lässt sich mit dem HTMLStripCharFilter eine HTML-Formatierung des Textes entfernen. Liegt der Text der Paragraphen im HTML-Format vor, ist es möglich, die HTML-Formatierung zu bereinigen und dann gezielt nur den tatsächlichen Inhalt in den Suchindex zu übernehmen. Das Aufsplitten des Textes in einzelne Wörter (tokens) erfolgt im zweiten Schritt mit einem Tokenizer.

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/services/search?q=*wirtschaft*`. The response is a JSON object with the following structure:

```

{
  "result": [
    {
      "id": 1104,
      "version": null,
      "number": "§ 4",
      "eli": "https://www.ris.bka.gv.at/eli/bgbl/i/2011/106/P4/NOR40191855",
      "title": "Nationales Klimaschutzkomitee",
      "text": "(1) Der Bundesminister für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft hat ein Nationales Klimaschutzkomitee einzurichten. (2) Das Nationale Klimaschutzkomitee berät über Grundsatzfragen zur österreichischen Klimapolitik im Lichte der Zielvorgaben des Übereinkommens von Paris, insbesondere über die langfristige Reduktion der Treibhausgasemissionen hin zu einer kohlenstoffarmen Gesellschaft, die Anpassung an unvermeidbare Folgen des Klimawandels sowie über langfristige Szenarien zur Steigerung der Energieeffizienz und des Anteils erneuerbarer Energieträger am Endenergieverbrauch.",
      "effectiveDate": "2017-04-26",
      "expireDate": null,
      "document": {
        "id": 1,
        "version": null,
        "title": "Bundesgesetz zur Einhaltung von Höchstmengen von Treibhausgasemissionen und zur Erarbeitung von wirksamen Maßnahmen zum Klimaschutz (Klimaschutzgesetz - KSG) StF: BGBl. I Nr. 106/2011 (NR: GP XXIV RV 1255 AB 1486 S. 124. BR: AB 8596 S. 861.)",
        "titleShort": "Klimaschutzgesetz",
        "abbreviation": "KSG"
      }
    }
  ],
  "page": {
    "size": 20,
    "totalElements": 6,
    "totalPages": 1,
    "number": 0
  },
  "abbreviations": [
    {
      "value": "KSG",
      "count": 4
    },
    {
      "value": "ASVG",
      "count": 2
    }
  ]
}

```

ONLINE

DEVELOPER-KONFERENZEN UND WORKSHOPS 2021

KONFERENZEN

// heise
devSec()

29.06.2021 · DevSecOps
01.07.2021 · Webanwendungen
heise-devsec.de

Industrial IoT
CONFERENCE

16.06.2021
iiot-conference.de

betterCode()

23.06.2021 · Qt
qt.bettercode.eu
31.08.2021 · WebAssembly
wasm.bettercode.eu
23.11.2021 · .NET 6.0



24.06.2021 · Agile Leader Day
ald.inside-agile.de



15.06.2021 · Vue Day
enterjs.de/vueday.php

Weitere Informationen unter:
heise.de/developer

WORKSHOPS betterCode()

- 11.06.2021: Webanwendungen mit React und TypeScript
- 21.06.2021: Turbo-Einführung funktionale Programmierung
- 24.06.2021: Effizientes DevOps-Tooling mit Go
- 29.06.2021: Praxistag: Best Practices für Modernes C++
- 05.07.2021: C++20-Crashkurs – Überblick zum neuen Standard

Weitere Informationen unter:
bettercode.eu/workshops.php

Workshop-Tickets unter:
bettercode.eu/tickets.php

Veranstalter:



@ heise Developer



dpunkt.verlag

Vom Datenbank- zum Suchindexeintrag: Drei Schritte der Textanalyse

Eingabe für HTMLStripCharFilter	Ergebnis
<code><h1 class="Titel AlignJustify">Text</h1><h2 class="UeberschrPara AlignCenter">Maßnahmen</h2><div class="MarginTop4 AlignJustify"><h3 class="GldSymbol AlignJustify">§&nbsp;2.</h3><p class="Abs AlignJustify">Maßnahmen im Sinne dieses Bundesgesetzes sind solche, ... <p></div></code>	Maßnahmen§ 2. Maßnahmen im Sinne des Bundesgesetzes sind solche, ...
Eingabe für StandardTokenizer	Ergebnis
Maßnahmen § 2. Maßnahmen im Sinne des Bundesgesetzes sind solche, ...	[Maßnahmen, §,2., Maßnahmen, im, Sinne, des, Bundesgesetzes, sind, solche]
Eingabe für LowerCaseFilter	Ergebnis
[Maßnahmen, §,2., Maßnahmen, im, Sinne, des, Bundesgesetzes, sind, solche]	[maßnahmen, §,2., maßnahmen, im, sinne, des, bundesgesetzes, sind, solche]

Im dritten Schritt kommen Token-Filter zur Anwendung, um einzelne Tokens zu normalisieren, transformieren oder zu entfernen. Ein Beispiel dafür ist die Normalisierung der deutschen Umlaute oder die Stammformreduktion: Das Stemming führt Wörter auf einen gemeinsamen Wortstamm zurück und ermöglicht somit auch verschiedene Formen eines Wortes als Suchtreffer.

Tabelle 1 zeigt ein Beispiel für die Transformation des Textes aus der Datenbank mit dem Analyzer für den Suchindex.

In der Standardeinstellung von Hibernate Search mit Apache Lucene erledigt ein Tokenizer die Aufteilung in einzelne Wörter und ein Token-Filter kümmert sich um die Kleinschreibung. Diese Analyse funktioniert für viele Sprachen, ist aber auch nicht optimiert für eine bestimmte.

Die Erstellung einer individuellen Konfiguration für Apache Lucene im Backend wird mit einem `LuceneAnalysisConfigurer` umgesetzt. Bei den deutschsprachigen Gesetzestexten kommt ein Standard Tokenizer zum Einsatz, außerdem die Token-Filter `LowerCaseFilter`, `ASCIIFoldingFilter` und `SnowballPorterFilter`. Letzterer implementiert die Stammformreduktion, die sich mit einem Parameter für Deutsch konfigurieren lässt. Die Konfiguration erhält die Bezeichnung "german" und ist in der `Article`-Klasse beim `@FullTextField` zu platzieren (s. Listing 10).

Aufruf einer Suchabfrage

Der Aufruf einer Suchabfrage erfolgt im Beispiel über eine REST-API, die der `SearchController` implementiert. Je nach Anwendungsfall lässt sie sich vom Frontend oder von einem anderen Service aus aufrufen. Das Resultat ist ein JSON-Objekt, das die Ergebnisliste (`result`), die Paginierungsinformation (`page`) und die Aggregation nach Abkürzungen (`abbreviations`) enthält. Abbildung 1 auf Seite 40 zeigt zeigt die Suche nach dem Begriff "Wirtschaft" samt Ergebnis. Die Suche fördert sechs Einträge zutage, und die Aggregation zeigt, dass davon vier im Klimaschutzgesetz (KSG) sowie zwei im Sozialversicherungsgesetz (ASVG) enthalten sind.

Volltextsuche einfach integriert

Mit Hibernate Search lässt sich ein Spring-Boot-Projekt um eine Volltextsuche erweitern und vielfältig konfigurieren. Das Release 6 von Hibernate Search führt viele Neuerungen ein, und die Anwendung der Such-DSL mit den Lambda-Ausdrücken ist einfach. Da für die Kombination von Hibernate Search und Apache Lucene kein eigener Suchserver nötig ist, lässt sich jedes beliebige Projekt um eine Volltextsuche erweitern. Die Kunden profitieren von den gewohnten Funktionen einer indexbasierten Suche, und der Aufwand für Entwicklung und Wartung bleibt überschaubar. Zur Vertiefung lohnt sich ein Blick in die sehr gute Dokumentation des Projekts [7].

Quellen

- [1] Website des Projekts Hibernate: hibernate.org/orm/
- [2] Release-Meldung zu Hibernate Search 6.0: in.relation.to/2020/12/11/hibernate-search-6-0-0-Final/
- [3] Leitfaden für die Migration zu Version 6.0: docs.jboss.org/hibernate/search/6.0/migration/html_single/
- [4] Spring Data: spring.io/projects/spring-data
- [5] Spring Data REST: spring.io/projects/spring-data-rest
- [6] Quelltext zum Projekt: github.com/j3ernhard/spring-boot-hibernate-search-playground
- [7] Dokumentation zu Hibernate: docs.jboss.org/hibernate/search/6.0/reference/en-US/html_single/



Bernhard Jungwirth

ist Softwareentwickler beim Umweltbundesamt in Wien und Mitentwickler von FlexLex.at – einer Webanwendung zum Erstellen individueller Gesetzessammlungen als Print-on-Demand-Produkt.



~~Vielleicht!~~

~~Könnt~~

LOS GEHT'S!

~~Sollte!~~

~~Würde!~~

29. Juni 2021

THEMEN-HIGHLIGHTS:

- Veränderung beginnt im Kopf und Bauch – Impuls für den Umgang mit Veränderung
- Selbstreflektion – was ist das, wie geht das?
- Bewerbung: Im Jahr 2021 richtig bewerben
- Gehaltsspiegel IT – Was bin ich wert?
- Kostenloser Lebenslaufcheck

+

Virtuelle Ausstellung & Möglichkeit für **direkte Bewerbungsgespräche**



Registrieren Sie sich jetzt kostenlos!

www.it-job-kompakt.de

Organisiert von:

 **Heise Medien**

UNSERE PARTNER

 **accenture**

develop4edu

GULP
experts united

 **Heise Gruppe**

 **Jobware**
BEI WIRTSCHAFTSUNIVERSITÄT WÜRZBURG

Landesamt für Digitalisierung,
Breitband und Vermessung
IT-Dienstleistungszentrum des Freistaats Bayern



NTS RELAX,
WE CARE

PIXIDA

 **STRATO**

VHV GRUPPE /



As Software Engineers,
we bring chips to life.

#beInfineon

Software Engineers at Infineon dedicate their skills to developing products that count, such as security software and systems for self-driving cars. Without them, chips would never be as powerful as they are. You could say that their designs give chips a brain.

Do you want to know how Software Developer Roland keeps the bad guys of the cyber world in check and why Ruth thinks it's a good thing when Software Engineers are lazy?

Find employee stories, a contact person and open jobs at:
www.infineon.com/software-engineering

#beInfineon

Your impact: Write great code for tiny things.

