



# Center for Code Excellence Trends – Eine Plattform für Trends in Softwaretechnologien

Johannes Kroß<sup>1</sup> · Peter Bludau<sup>1</sup> · Alexander Pretschner<sup>1,2</sup>

Online publiziert: 6. Juli 2020  
© Der/die Autor(en) 2020

## Zusammenfassung

Neue Softwaretechnologien entstehen täglich. Da deren Vielfalt wächst und deren Lebensdauer aber abnimmt, müssen Unternehmen und Fachleute Trends erkennen und Technologien bewerten können, um fundierte Designentscheidungen für nachhaltige Softwaresysteme zu treffen. Die Wahrnehmung neuer Trends außerhalb der eigenen Fachkompetenz kann jedoch schwierig sein und die Entscheidung für bestimmte Technologien kann durch persönliche Vorurteile beeinflusst werden. In diesem Beitrag stellen wir unsere Plattform vor, um Trends und Beziehungen von Softwaretechnologien zu visualisieren. Basierend auf öffentlichen Stellenanzeigen, Open-Source-Projekten und Online-Diskussionen erkennen wir (i) automatisch neue Technologien, Kompetenzen und Techniken, (ii) lernen Beziehungen zwischen ihnen und (iii) bewerten ihre Trendentwicklung. Wir demonstrieren die Ergebnisse unseres Ansatzes, indem wir exemplarische Vergleiche und Zusammenhänge aufzeigen. Unsere Plattform soll Organisationen und Einzelpersonen eine unabhängige Bewertung und Informationsquelle zur Entscheidungsunterstützung bieten.

## Motivation und Bedarf

Open-Source-Plattformen und ein wachsender Anteil an Software in Unternehmen treiben die Entstehung, aber auch Kurzlebigkeit von Softwaretechnologien voran. So wurden beispielsweise 2019 über 44 Mio. neue Software-Repositories auf GitHub erstellt [4]. Diese steigende Vielfältigkeit hat unterschiedliche Auswirkungen auf Unternehmen und Fachkräfte.

Unternehmen, insbesondere, wenn Softwaretechnik nicht zu der Kernkompetenz zählt, müssen in der Lage sein, Trends in Softwarethemen zu erkennen, sowie deren Potenziale richtig einzuschätzen und einzusetzen. Dies gilt sowohl für Investitionsentscheidungen in Drittanbieter-Software als auch für eigene Softwareentwicklungen.

Im letzteren Fall ist es weiterhin wichtig, Frameworks je nach Anwendungsfall passend auszuwählen, um eine möglichst zukunftsfähige Software sicherzustellen und ein veraltetes System zu vermeiden. In ähnlicher Weise müssen auch Fachkräfte mit aktuellen Entwicklungen in Softwaretechnologien, Themen und Techniken Schritt halten und ihre Kompetenzen verstärken. Die Entscheidung für eine Technologie oder einen Toolstack, beispielsweise, basiert oftmals lediglich auf einer persönlichen Präferenz oder einem Vorurteil gegenüber anderen bzw. neuen Technologien, kann aber tiefgreifende Auswirkungen auf den Erfolg und die Qualität von Software haben. Ebenso schwer kann es sein, neue Technologien zu identifizieren und deren Reife und Anwendungsmöglichkeiten ausreichend zu evaluieren, vor allem, wenn Domänenwissen erforderlich wäre.

Plattformen wie Google Trends [5] erlauben es, Trends durch automatisierte Analysen zu visualisieren. Jedoch fällt es schwer, neue Softwaretechnologien, Themen oder Techniken zu entdecken und in Bezug zu ähnlichen Themen zu setzen. Diese Herausforderungen adressieren wir mit unserer Plattform Center for Code Excellence (CCE) Trends [1, 2]. In diesem Aufsatz stellen wir unsere Plattform vor und zeigen auf, wie sich Toptrends zu einem Softwarethema erkennen lassen, Technologien auf verschiedene Arten vergleichen lassen und Beziehungen zwischen Technologien und Softwarethemen visualisiert werden können. Mit den CCE Trends möchten wir Unternehmen und Fachkräften

---

Johannes Kroß  
kross@fortiss.org

Peter Bludau  
bludau@fortiss.org

✉ Alexander Pretschner  
alexander.pretschner@tum.de

<sup>1</sup> fortiss, Landesforschungsinstitut des Freistaats Bayern, Guerickestr. 25, 80805 München, Deutschland

<sup>2</sup> Technische Universität München, Boltzmannstr. 3, 85748 Garching, Deutschland

eine unabhängige Informationsquelle und Wissensbasis für Themen der Softwaretechnik zur Verfügung stellen, die als Anregung für Weiterbildungsmaßnahmen dient und als Entscheidungshilfe unterstützt.

## Verwandte Ansätze

Um Trends von Technologien, Techniken und Themen in der Softwaretechnik aufzuzeigen, gibt es eine Reihe von manuellen Ansätzen. Zu den wohl prominentesten manuellen Ansätzen zählt das Technologieradar von ThoughtWorks [10]. Dieses Radar stellt Trends und Veränderungen der Softwarelandschaft in den vier Bereichen Techniken, Tools, Plattformen und Sprachen bzw. Frameworks vor. Die Trends des regelmäßig erscheinenden Radars basieren auf den persönlichen Einschätzungen und Bewertungen mehrerer Experten. In Anlehnung daran gibt es eine Vielzahl an weiteren Unternehmen, die ihre eigenen Einschätzungen und Bewertungen von Softwaretrends veröffentlichen, beispielsweise Zalando [12] und Munich RE [7].

Im Gegensatz dazu gibt es automatisierte Lösungen, um Trends zu visualisieren und zu vergleichen. Sowohl Google [5] als auch Stack Overflow [9] bieten beispielsweise die Möglichkeiten, Trends basierend auf den Nutzeraktivitäten ihrer eigenen Plattform zu erkunden. In ähnlicher Weise gibt es eine Reihe von Webplattformen, die Statistiken von Begriffen auf diversen sozialen Medien anbieten. Zuletzt gibt es die Initiative DB-Engines [6], die strukturierte Informationen zu Datenbanklösungen aufbereitet und ein monatliches Ranking der aktuellen Popularität veröffentlicht. Das Ranking basiert auf einer Vielzahl von Datenquellen, wie zum Beispiel der Nachfrage auf Suchmaschinen, Diskussionsplattformen und sozialen Netzwerken.

## Ansatz und Design

Unser Ziel ist es, eine Plattform zu schaffen, die neue Technologien identifiziert, quantitativ über die Zeit bewertet und weiterführende Informationen zum Technologieumfeld, wie zum Beispiel Beziehungen zu ähnlichen Technologien und Toolstacks, bereitstellt. Zu diesem Zweck haben wir eine Reihe von Komponenten entwickelt, die in Abb. 1 dargestellt sind und nachfolgend besprochen werden.

Die *CCE-Trends*-Plattform greift hauptsächlich auf zwei Komponenten zurück. Die Komponente *Technology Provider* bietet Webschnittstellen für unsere Wissensbasis von Technologien, während die Bewertungen von Technologien über die Schnittstelle des *Technology Score Provider* angeboten werden. Die Datenbasis zur Bereitstellung der Informationen beider Komponenten stellen wir nachfolgend im Detail vor.

## Erstellung einer Wissensbasis

Um Technologien und Softwarethemen zu verstehen und Beziehungen zu lernen, bilden wir Informationen in einem Wissensgraphen bzw. einer Ontologie ab (*Technology Graph Calculation*), wobei Informationen als Knoten und Beziehungen (inkl. Metainformationen) als Kanten dargestellt sind. Als Knoten unterscheiden wir derzeit zwischen Programmiersprachen, Technologien und Kategorien, die wiederum Technologien in verschiedene Gruppen (z.B. Cloud Computing, Machine Learning) einordnen. Als Kanten differenzieren wir Ähnlichkeits-, Ko-Verwendungs- und Kategorisierungsbeziehungen sowie Kanten zur Verknüpfung von Metainformationen zu Technologien wie beispielsweise Synonym-, Beschreibungs-, Bild- und Referenzbeziehungen.

Diese Informationen werden durch die Verarbeitung unterschiedlicher Datenquellen erhoben. Zunächst verwenden wir verschiedene APIs, um Metainformationen zu den einzelnen Begriffen zu sammeln. Hierfür nutzen wir *Github Explore* [3], *Wikipedia* [11] und *StackOverflow* [8], wobei Beschreibungstexte, Synonyme, Bilder und weiterführende Ressourcen zu den Begrifflichkeiten abgefragt und verknüpft werden. Zudem identifizieren wir, durch die grammatikale Analyse der Beschreibungstexte, Anhaltspunkte zur Kategorisierung von Technologien, sowie Informationen zu etwaigen Konkurrenzprodukten bzw. Ähnlichkeiten und weitere Beziehungen zu anderen Technologien.

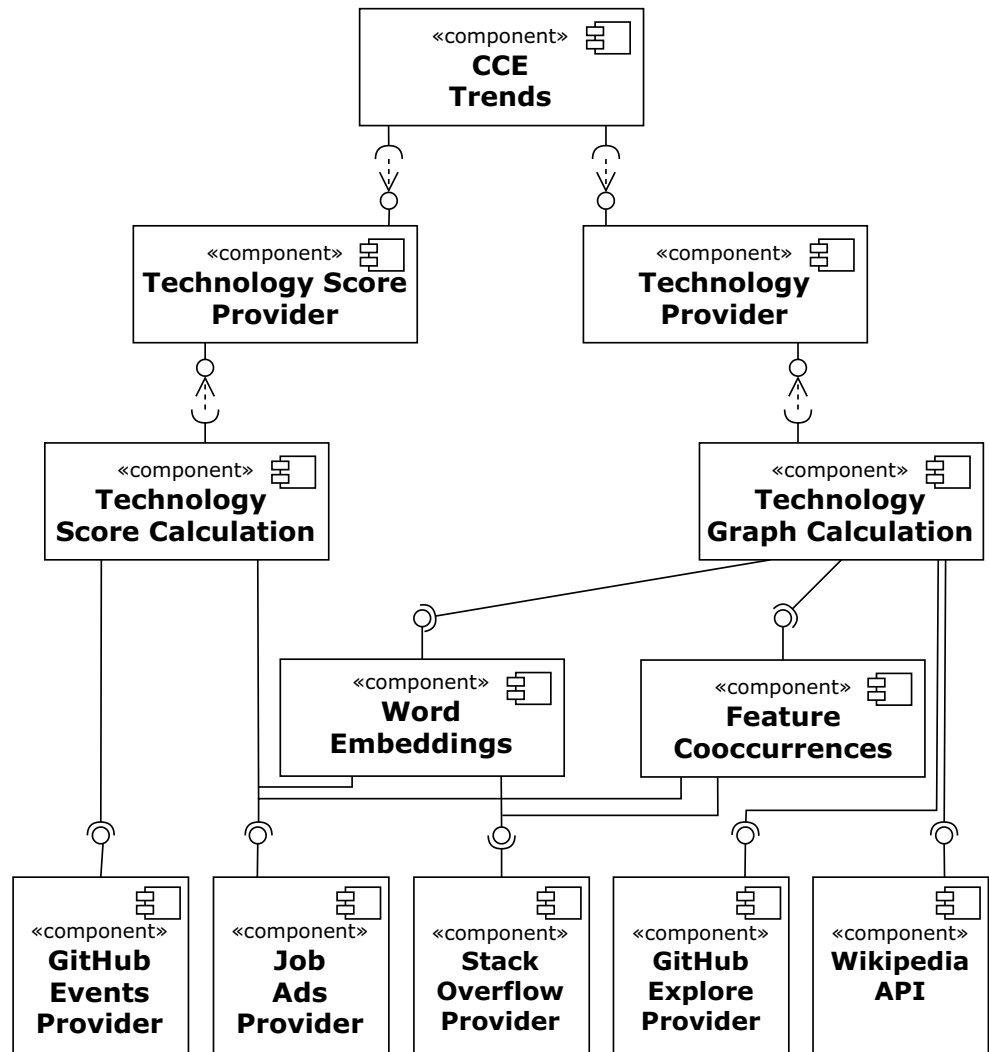
Ähnlichkeitsbeziehungen zwischen Technologien werden weiterhin aus öffentlichen Stellenanzeigen (*Job Ads Provider*) und Diskussionen in Internetplattformen (*StackOverflow Provider*) gewonnen, wobei die Stellenanzeigen aus der StackOverflow-Stellenbörse extrahiert werden und die Diskussionsfragen als Datendownload zur Verfügung stehen. Dabei lernen wir eingebettete Repräsentationen (*Word Embeddings*) des gesamten Korpus an Texten. Einzelne Worte werden als Vektoren repräsentiert, die im Anschluss durch Distanzberechnungen die Nähe und Ähnlichkeit zweier Begriffe im Vokabular ausdrücken.

Wir berechnen zudem eine Co-Occurrence-Matrix auf denselben Datenquellen (*Feature Cooccurrences*). Diese stellt jedem Wort jedes andere Wort in allen Dokumenten gegenüber und gibt an, wie häufig Begriffe zusammen in Dokumenten vorkommen. Hieraus können wir die Koverwendungsbeziehung und Toolstack-Beziehungen ableiten.

## Bewertung von Softwaretechnologien

Im Fokus der *CCE-Trends*-Plattform steht es, Technologien, deren Beliebtheit und Reife vereinfacht identifizieren und einordnen zu können. Dazu berechnen wir eine Bewertung für jede Technologie (*Technology Score Calculation*). Derzeit verwenden wir Eventdaten von Open-Source-Pro-

**Abb. 1** Komponenten, Schnittstellen und Abhängigkeiten der CCE Trends



jekten der Plattform GitHub (*GitHub Events Provider*) und, wie auch bei der Erstellung der Wissensbasis, öffentliche Stellenanzeigen von StackOverflow als Grundlage für diese Bewertung.

Stellenanzeigen repräsentieren die Nachfrage nach einer Technologie auf dem Arbeitsmarkt und Kompetenzen, die gesucht werden. Die Eventdaten dagegen repräsentieren das Angebot und Interesse an einer Technologie und damit vorhandene Kompetenzen. Durch die Verwendung beider Datenquellen möchten wir wichtige Bereiche des Marktes abdecken.

Um Bewertungen und Trends zu erhalten, werden die Datenquellen in Zeitreihen analysiert. Texte der Stellenanzeigen werden bereinigt, Vorkommnisse von Technologien gezählt, gefiltert und normalisiert. Dabei werden auch Synonymbeziehungen aus der Wissensdatenbank beachtet, um verschiedene Schreibweisen und Abkürzungen zu konsolidieren. Die Zeitreihen stellen dadurch die relative Häufigkeit einer Technologie in der Gesamtheit der Stellenanzei-

gen dar. Aus den Eventdaten von GitHub-Projekten verwenden wir Events, die neue Commits, Beitragende, Forks und Favoriten eines Projekts betreffen. Um zu einer Technologie die zugehörigen Projekte zu bestimmen, verwenden wir die Tags der jeweiligen Projekte. Alle Open-Source-Projekte, die einen Tag einer bestimmten Technologie enthalten, werden zusammengefasst, und die zugehörigen Eventdaten werden summiert, normalisiert und als Zeitreihen zur Bewertung dieser Technologie herangezogen. Jede Zeitreihe repräsentiert dann das Interesse an einer Technologie.

Die Trendberechnung selbst basiert auf den Zeitreihen der jeweiligen Datenquelle und betrachtet die Entwicklung einer Technologie über die Zeit durch einfache Trendanalyseverfahren. Die finale Bewertung einer Technologie ist eine Kombination aus relativem Vorkommen und Trend. Dadurch werden übermäßige Trendschwankungen bei unbedeutenden Technologien sowie hohe relative Vorkommen bei etablierten Technologien ohne erkennbaren Trend abgeschwächt.

## CCE Trends

Den Startpunkt zum Zugriff auf die Plattform bildet eine Suchmaske, in der Softwarethemen und konkrete Technologien vorgeschlagen werden und gesucht werden können. Davon ausgehend werden nachfolgend beispielhaft einzelne Hauptfunktionen der CCE Trends vorgestellt.

### Relevanz von Technologien für ein Softwarethema

Wird nach einem Softwarethema bzw. einer -Kategorie (z.B. Cloud-Anbieter, Deep Learning, Distributed Ledgers) gesucht, werden zugehörige Technologien auf einem Radar gezeigt. Je höher die Bewertung einer Technologie ist, desto zentrierter wird sie auf dem Radar platziert. Wie bereits angesprochen ergibt sich die Bewertung einer Technologie dabei aus der Relevanz und deren Entwicklung über die Zeit (Trend). Um die Relevanz von Technologien sinnvoll auf dem Radar zu platzieren, wurden initial alle

verfügbaren Technologien auf einer Skala von 0 bis 100 eingestuft.

Abb. 2 zeigt das Radar mit einem Ausschnitt an Technologien für die Kategorie *Programmiersprachen*. Beispielsweise besitzt die Sprache *PHP* eine niedrigere Bewertung als *Python*. Die illustrierte Bewertung basiert hierbei bei beiden verfügbaren Datenquellen, sowohl der Nachfrage auf dem Arbeitsmarkt als auch dem Angebot auf Open-Source-Plattformen. Nutzer haben die Möglichkeit, die Bewertung auch je Datenquelle zu filtern bzw. einzusehen.

Die Visualisierung durch das Radar soll es Nutzern auf diese Weise ermöglichen, sich einen ersten Überblick zu verschaffen. Manager oder auch Nutzer mit oberflächlichem Technologiewissen können eine Einschätzung zu ihrem jeweiligen Interessengebiet erhalten, um beispielsweise im Unternehmen eingesetzte Technologien und Kompetenzen zu benchmarken. Hochschulen können die Plattform auch für die Modernisierung des Lehrangebots nutzen, indem sie für die praktische Anwendung möglichst weitgenutzte, für



Abb. 2 Radar für die aktuelle Relevanz von Technologien (Ausschnitt) einer Kategorie. Relevantere Technologien werden zentrierter platziert

ein gewisses Thema passende und moderne Technologien einsetzen.

Um weitergehende Informationen zu den Technologien zu erfahren, wie zum Beispiel Beschreibungen und weiterführende Links, können eine oder mehrere Technologien ausgewählt und in einer Detailansicht untersucht werden.

Die nachfolgenden Abschnitte erklären weitere einzelnen Funktionen der Detailseiten.

### Zeitlicher Verlauf der Relevanz von Technologien

Auf den Detailseiten wird der zeitliche Verlauf der Relevanz der jeweiligen Technologie visualisiert, um die Bedeutung der Bewertung und auch den Trend der Technologien genauer zu erläutern.

Steht in einem Unternehmen oder in einer Hochschule beispielweise die Entscheidung über den Einsatz einer Programmiersprache für den Bereich von Server- bzw. Webanwendungen bevor, können mehrere Technologien verglichen werden, die als aufstrebend wahrgenommen werden oder für die Mitarbeiter bereits Kompetenzen besitzen. Abb. 3 stellt hierfür einen exemplarischen Vergleich typischer und neuerer Programmiersprachen für Webanwendungen dar.

Der zeitliche Verlauf der Relevanz in Open-Source-Plattformen (Abb. 3a) zeigt deutlich auf, dass die Sprachen *PHP* und *Ruby* im Jahr 2015 sehr dominierend waren, der Trend jedoch kontinuierlich abnehmend ist. *Rust* und *TypeScript*, zwei neuere Programmiersprachen, besaßen im Jahr 2015 eine noch deutlich niedrigere Relevanz, aber steigen seither kontinuierlich und weisen beinahe beide eine höhere Relevanz in Open-Source-Plattformen als *PHP* und *Ruby* heutzutage auf. Betrachtet man jedoch den zeitlichen Verlauf basierend auf Stellenanzeigen (siehe Abb. 3b), wird ersichtlich, dass *PHP* nach wie vor auf dem Arbeitsmarkt gefragt ist, was möglicherweise auch an der notwendigen Wartung und Weiterentwicklung bestehender Systeme liegen kann, die Nachfrage nach der Sprache *TypeScript* jedoch auch hier stetig zunimmt. Dass die Nachfrage auf dem Arbeits-

markt dem Angebot auf Open-Source-Plattformen „hinterhinkt“, ist im Allgemeinen wohl nicht überraschend.

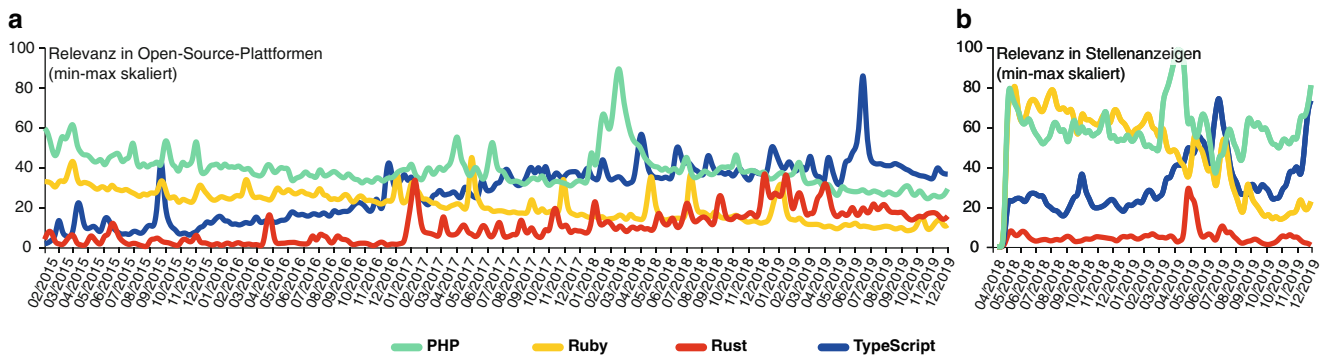
Da wir Stellenanzeigen erst seit Anfang 2018 erheben ist der zeitliche Verlauf, im Vergleich zu Open-Source-Plattformen, entsprechend limitiert. Im Gegensatz zu dem Radar, wird die Relevanz für die zeitliche Darstellung min-max skaliert, damit Trends für den Nutzer einfacher und deutlicher erkennbar sind und ein Vergleich mehrerer Technologien möglich wird.

### Regionale Verteilung der Relevanz

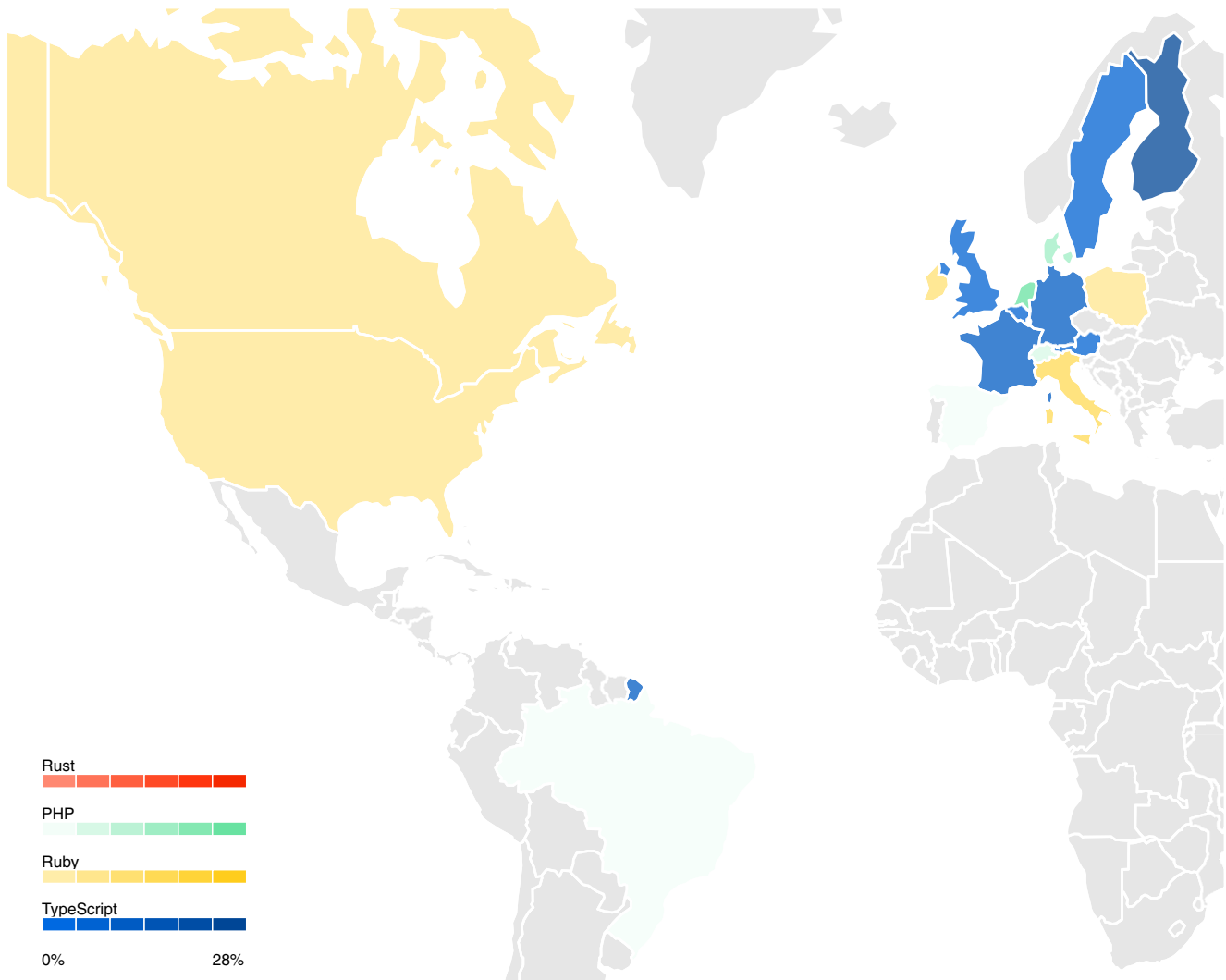
Neben dem allgemeinen zeitlichen Verlauf lässt sich zusätzlich die regionale Verteilung der Nachfrage darstellen, um Technologiebewertungen auch in unterschiedlichen Arbeitsmärkten vergleichen zu können. Hierfür wird das prozentuale Vorkommen in Stellenanzeigen pro Land als Heatmap auf einer Weltkarte visualisiert. Bei einem Vergleich mehrerer Technologien erhält ein Land nur die Farbe und Temperatur der Technologie, die die höchste Relevanz für dieses Land besitzt. Für jedes Land sind trotzdem auch die Bewertungen aller Technologien einsehbar.

Ein Ausschnitt für die bereits erwähnten Programmiersprachen ist in Abb. 4 dargestellt. *Ruby* gefolgt von *TypeScript* sind beispielsweise am stärksten in den Vereinigten Staaten gefragt, während in Deutschland *TypeScript* gefolgt von *PHP*-Kompetenzen am häufigsten in Stellenanzeigen gesucht werden. Mit Bezug auf unsere Beispielszenarien in Unternehmen und Hochschulen können Nutzer durch unsere plattformähnliche Technologien, z. B. je nach Thema oder auf Vorschlag durch Kollegen, vergleichen und letztendlich, zum Beispiel *TypeScript* genauer evaluieren, testweise praktisch einsetzen und ggf. diese Sprache in Zukunft für passende Projekte adaptieren.

Des Weiteren lässt sich, ähnlich zu Abb. 3, der zeitliche Verlauf der Relevanz auch pro Land einsehen, wodurch ein Vergleich der Entwicklung der Relevanz einer Technologie in verschiedenen Ländern möglich wird.



**Abb. 3** Min-max normalisierte Zeitreihen der Relevanz von Technologien. (a) Relevanz in Open-Source-Plattformen. (b) Relevanz in Stellenanzeigen



**Abb. 4** Regionale Verteilung, die das prozentuale Vorkommen auf dem Arbeitsmarkt pro Land zeigt. Länder sind mit der Farbe der jeweils dominierenden Technologie eingefärbt

Bei der regionalen Verteilung sind zwei Einschränkungen zu beachten. Die regionale Verteilung ist aktuell nur für die Bewertung einer Technologie basierend auf der Nachfrage auf dem Stellenmarkt möglich, da sich nur hier eindeutige regionale Zugehörigkeiten bestimmen lassen. Daneben ist die Anzahl an verfügbaren Stellenanzeigen pro Land gering, da im Moment lediglich eine Quelle bzw. eine Plattform für Stellenangebote untersucht wird. Entsprechend sind die Aussagekraft und Repräsentativität hierfür aktuell noch kritisch zu betrachten. Beide Limitationen planen wir in naher Zukunft zu minimieren.

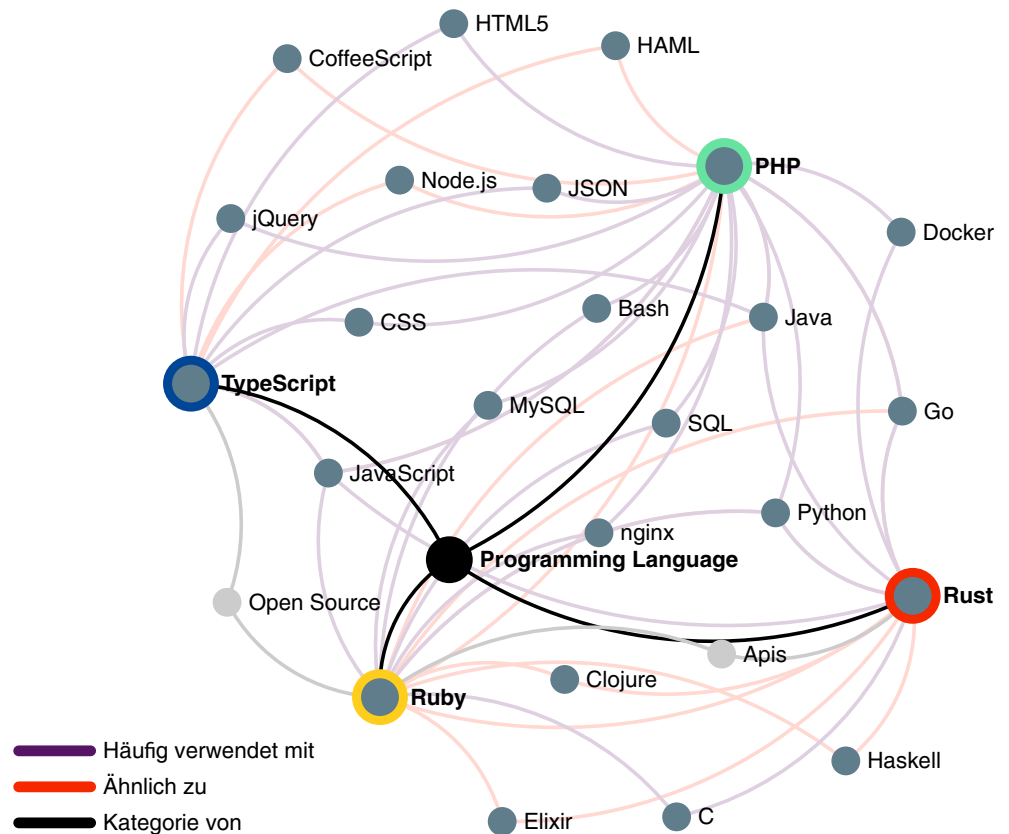
### Abhängigkeiten zwischen Technologien und Softwarethemen

Um einen Teil unserer Wissensbasis zu visualisieren, zeigen wir für jede Technologie zugehörige Softwarethemen bzw.

Kategorien, ähnliche Technologien und Technologien, die häufig zusammen verwendet werden, an. Bei einem Vergleich mehrerer Technologien veranschaulichen wir deren gemeinsamen Abhängigkeiten in einem Graphen.

Abb. 5 stellt diesen Vergleich anhand der bereits erwähnten Programmiersprachen dar. Die Darstellung der Abhängigkeiten soll Nutzern der CCE Trends ermöglichen, weitere ähnliche Technologien einzusehen und kennenzulernen, die eventuell noch unbekannt sind. Zum Beispiel könnten Nutzer auf die Sprache *Elixir* aufmerksam werden, die ähnlich zu *Ruby* und *Rust* ist. Daneben können komplementäre Technologien, die häufig zusammen verwendet werden, gefunden werden. Beispielsweise wird *Rust* häufig mit *C-Bibliotheken* verwendet und Anwendungen mit *Docker* in Container gebündelt. Zuletzt werden gemeinsame Softwarethemen bzw. -kategorien angezeigt (z.B. *Programmiersprachen*).

**Abb. 5** Gemeinsame Beziehungen zwischen PHP, Ruby, Rust und TypeScript



Ein limitierender Faktor bei den Abhängigkeiten liegt noch in der Qualität der Ergebnisse. Häufig herrscht eine gewisse Unschärfe bei den Beziehungen *häufig verwendet mit* und *ähnlich zu*. Die Ursachen hierfür sind die aktuell nur gegebene Möglichkeit für unüberwachtes Lernen dieser Beziehungen, da keine Evaluationsgrundlage vorhanden ist, und die Diversität der Verwendung der Begriffe in Stellenanzeigen. Je nach Stellenanzeige werden beispielsweise sowohl komplementäre Technologien (z.B. TypeScript und Node.js) als auch ähnliche Technologien (z.B. Rust und Java) in gleichem Kontext aufgeführt.

**Fazit und Ausblick**

Mit den CCE Trends möchten wir eine Übersicht anbieten, um neue Technologien zu identifizieren, eine unabhängige, quantitative Bewertung über die Zeit zu erhalten und weiterführende Informationen zu ähnlichen Technologien und Toolchains zu beziehen. Unser mittelfristiges Ziel ist es, Nutzern auch ohne technologische Expertise eine Informationsquelle und Entscheidungshilfe bereitzustellen, um selbstständig Fragen zu beantworten wie zum Beispiel, welche Cloud-Anbieter im Moment wie sehr gefragt sind, ob die Programmiersprache Rust wirklich häufig eingesetzt

wird oder welche Toolstacks für maschinelles Lernen häufig zusammen mit Scikit Learn genutzt werden.

In Zukunft möchten wir weitere Datenquellen und Metriken (z.B. Erlernbarkeit) hinzufügen und auch Lehrmaterialien für Technologien zur Verfügung stellen. Ebenso planen wir direktes Feedback der Nutzer zu integrieren, um die Technologiebeziehungen durch unser unüberwachtes Lernen zu evaluieren. So möchten wir langfristig auch überwachtes Lernen ermöglichen und unsere Datenqualität verbessern.

**Danksagung** Das CCE wird durch das Bayerische Staatsministerium für Wirtschaft, Landesentwicklung und Energie gefördert.

**Funding** Open Access funding provided by Projekt DEAL.

**Open Access** Dieser Artikel wird unter der Creative Commons Namensnennung 4.0 International Lizenz veröffentlicht, welche die Nutzung, Vervielfältigung, Bearbeitung, Verbreitung und Wiedergabe in jeglichem Medium und Format erlaubt, sofern Sie den/die ursprünglichen Autor(en) und die Quelle ordnungsgemäß nennen, einen Link zur Creative Commons Lizenz beifügen und angeben, ob Änderungen vorgenommen wurden.

Die in diesem Artikel enthaltenen Bilder und sonstiges Drittmaterial unterliegen ebenfalls der genannten Creative Commons Lizenz, sofern sich aus der Abbildungslegende nichts anderes ergibt. Sofern das betreffende Material nicht unter der genannten Creative Commons Lizenz steht und die betreffende Handlung nicht nach gesetzlichen Vorschriften

ten erlaubt ist, ist für die oben aufgeführten Weiterverwendungen des Materials die Einwilligung des jeweiligen Rechteinhabers einzuholen.

Weitere Details zur Lizenz entnehmen Sie bitte der Lizenzinformation auf <http://creativecommons.org/licenses/by/4.0/deed.de>.

## Literatur

1. fortiss CCE Trends. <https://cce.fortiss.org/trends>. Zugegriffen: 26. Nov. 2019
2. fortiss Center for Code Excellence. <https://cce.fortiss.org/>. Zugegriffen: 26. Nov. 2019
3. GitHub github/explore—community-curated topic and collection pages on GitHub. <https://github.com/github/explore>. Zugegriffen: 4. Dez. 2019
4. GitHub The State of the Octoverse. <https://octoverse.github.com/>. Zugegriffen: 28. Nov. 2019
5. Google Google Trends. <https://trends.google.com/trends/>. Zugegriffen: 3. Dez. 2019
6. solid IT. DB-Engines—Knowledge Base of Relational and NoSQL Database Management Systems. <https://db-engines.com/>. Zugegriffen: 26. Nov. 2019
7. Munich R Tech Trend Radar 2019 – Die wichtigsten technologischen Trends. <https://www.munichre.com/topics-online/de/digitalisation/tech-trend-radar-future-technologies-2019.html>. Zugegriffen: 3. Dez. 2019
8. Stack Exchange Stack Exchange API. <https://api.stackexchange.com/>. Zugegriffen: 3. Dez. 2019
9. Stack Exchange Stack Overflow Trends. <https://insights.stackoverflow.com/trends>. Zugegriffen: 3. Dez. 2019
10. ThoughtWorks Technology Radar—An opinionated guide to technology frontiers. <https://www.thoughtworks.com/radar>. Zugegriffen: 3. Dez. 2019
11. Wikipedia Wikipedia—The Free Encyclopedia. <https://www.wikipedia.org/>. Zugegriffen: 3. Dez. 2019
12. Zalando Zalando Tech Radar. <https://opensource.zalando.com/tech-radar/>. Zugegriffen: 3. Dez. 2019