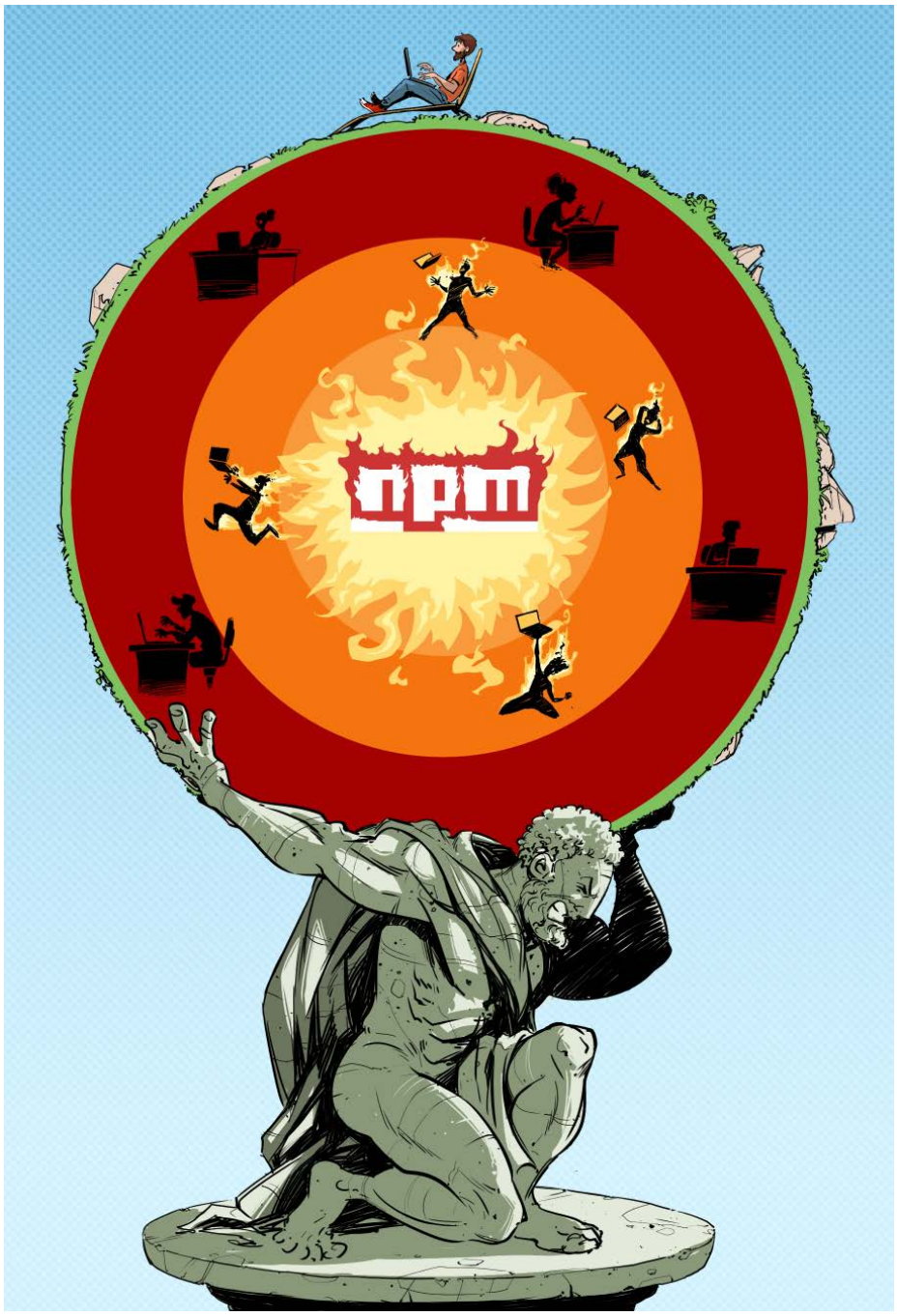


Bild: Albert Hulm

Die Säulen der Erde

Die Architekten und Arbeiter eines JavaScript-Frameworks

Große Open-Source-Projekte sind ein Zusammenspiel vieler Komponenten, die wiederum aus Komponenten bestehen. Eine beispielhafte Reise in die Tiefen des Frameworks Vue.js verrät, auf wessen Schultern die meiste Last großer Projekte ruht, wer hinter Benutzer- und Organisationsnamen steckt und was ein komplexes Open-Source-Gebilde ins Wanken bringen kann.

Von Jan Mahn

Banken nutzen sie, Versicherungen, Autohersteller, Start-ups und Weltkonzerne. Den großen Frontend-Frameworks wie Angular, Vue.js und React begegnet man überall dort, wo Websites mehr tun sollen, als nur statische Informationen darzustellen. Die Frameworks betreiben interaktive Bestell- und Konfigurationsseiten, Kunden- und Mitarbeiterportale, Admin-Oberflächen und Onlinebankingseiten. Veröffentlicht ist der Code der Frameworks unter Open-Source-Lizenz, einsetzen darf man sie auch kommerziell ohne Lizenzkosten.

Um zu verstehen, was so große Open-Source-Projekte zusammenhält, ist ein genauerer Blick angebracht. Auf den nächsten Seiten graben Sie sich mit uns anhand eines Beispielprojekts durch die Schichten aus Abhängigkeiten und stoßen zum Kern der Frage vor, wem Webentwickler die gut gepflegten, dokumentierten und nützlichen Werke wirklich zu verdanken haben – und wo die Probleme liegen, wenn Projekte auf zu wenigen Schultern lasten. Auf dem Weg werden Ihnen große und bekannte Akteure begegnen, unbekannte Einzelkämpfer und lose Grüppchen. Und Sie werden auf menschliche Schicksale und individuelle Karrierepfade stoßen, auf unternehmerische Probleme und die Auswirkungen der großen Weltpolitik auf harmlosen JavaScript-Code.

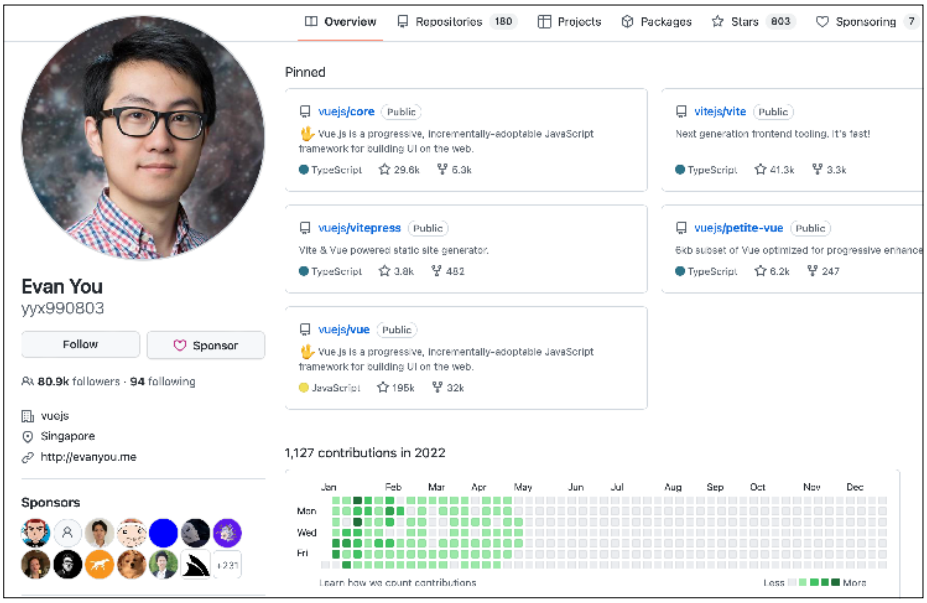
Von hinten aufgerollt

Als Forschungsobjekt haben wir uns das drittgrößte unter den Frontend-Frame-

works vorgenommen: Vue.js (github.com/vuejs/core). Das wird bei GitHub gehostet, eingesetzt wird es unter anderem von dessen Mitbewerber GitLab. In der Liste der Unternehmen, die mit Vue.js arbeiten, finden sich außerdem Schwergewichte wie Facebook, Netflix, Nintendo und Adobe. Bevor Missverständnisse aufkommen: Das bedeutet nicht, dass diese Unternehmen direkt alle ihre Websites mit Vue.js gebaut haben – das Framework kann auch in internen Admin-Oberflächen oder kleinen Anwendungen stecken.

Vue.js unterscheidet sich in einem nicht unwesentlichen Detail von Angular und React: Dahinter steckt kein Weltkonzern. Erfinder und bis heute Maintainer ist Evan You, ein Softwareentwickler Anfang 30. Nach dem Studium wurde Google auf seine JavaScript-Basteleien im Internet aufmerksam und kontaktierte ihn. Zwei Jahre lang bezahlte das Unternehmen You dafür, dass er im „Creative Lab“ Prototypen für neue Bedienkonzepte in JavaScript entwickelte. Weil Googles Angular nicht so recht zur Art seiner Projekte passte, entstand bei You langsam die Idee, selbst ein eigenes kleines Framework zu erschaffen. Den ersten Commit bei GitHub veröffentlichte er im Juli 2013.

In Interviews spricht You immer wieder über seine damalige Motivation, die



Evan You erfand Vue.js für den Eigenbedarf, heute spielt es in einer Liga mit Angular und React. Code für sein Projekt schreibt er immer noch.

prototypisch für Open-Source-Projekte ist: Bei seinem Arbeitgeber Google fehlte ihm ein Projekt, in dem echte Menschen seine Entwicklungen nutzten. Mit Vue.js bekam er schnell, was er vermisste: Kollegen und immer mehr Fremde begannen, die Software auszuprobieren und einzusetzen. Der große Durchbruch kam 2014 mit einem Tweet von Taylor Otwell, seines Zeichens Erfinder und Maintainer des populären PHP-Backend-Frameworks Laravel. Otwell war unzufrieden mit seinen zähen Angular-Lernfortschritten und verkündete per Twitter, jetzt Vue.js zu lernen. Damit stand das kleine Framework schlagartig im Rampenlicht. Die schon damals große La-

Glossar: Software im Team entwickeln

Contributor: Wer etwas zu einem Projekt beiträgt, darf sich Contributor nennen. Selbst wenn er nur einen Tippfehler in der Dokumentation beseitigt hat – für viele ist genau das der Einstieg in Open-Source-Projekte. Die Einstiegshürde ist niedrig, die meisten Softwareprojekte werden heute bei GitHub oder GitLab verwaltet. Mit einem Account dort kann jeder, der einen Fehler findet, beim Verbessern helfen.

Maintainer: Nicht jeder, der etwas am Code ändern will, kann Änderungen direkt vornehmen. Jedes Projekt braucht einen oder mehrere Maintainer, die die Berechtigung haben, eingereichte Änderungen zu akzeptieren und in den Code aufzunehmen. Maintainer kümmern sich auch darum, die geplanten Funktionen für eine neue Version festzulegen und steuern, wohin sich das Projekt entwickelt.

Pull Request und **Merge Request:** Auf den Softwareverwaltungsplattformen GitHub und Bitbucket heißen sie Pull Request, GitLab nennt sie Merge Request. Gemeint ist dasselbe. Wer Änderungen an einer Software plant, legt zunächst einen sogenannten **Fork** an, eine Kopie der Code-Basis. Darin schreibt er all seine geplanten Änderungen und testet sie gründlich. Wenn er überzeugt ist, dass die Änderungen das Projekt voranbringen, stellt er einen Request mit diesen Änderungen an das Originalprojekt. Ein Maintainer prüft die Änderungen und entscheidet, ob sie reif für die Übernahme sind.

Commit: Ein Block aus Änderungen in der Versionsverwaltung Git heißt Commit. Er kann Änderungen in mehreren Dateien enthalten. Bestenfalls teilt man Commits so, dass sie immer nur ein Problem angehen und gibt ihnen präzise Beschreibungen. Das erleichtert später die Nachverfolgung von Änderungen.

Issue: Zum guten Stil in Projekten gehört es, nicht blind loszulaufen und die Community mit einem Pull Request zu überraschen. Erster Schritt ist ein Issue – die Übersetzung „Problem“ greift zu kurz, Issues können auch Wünsche oder Ideen sein. Meist werden sie erst diskutiert, bevor sich jemand an neue Entwicklungen macht und schließlich einen Pull Request anlegt.

Code Review: Bevor Maintainer Code in ihre Software übernehmen, prüfen sie ihn gründlich in einer Code Review. In einigen Projekten passiert das auch mehrstufig, bevor die Maintainer des Gesamtprojekts zustimmen, schauen sich Spezialisten für eine Baustelle den Code an und erstellen eine Review. Wenn bei der Review Probleme auftreten, kann der Code auch noch mal zur Überarbeitung an den Ersteller gehen.

ravel-Community gab Vue.js eine Chance, und die Nutzerzahlen stiegen rasant.

Beide Hände im Code

Auch 2022 ist Evan You noch Maintainer des Vue-Projekts, mittlerweile nicht mehr allein, sondern mit einem Core-Team aus insgesamt 19 Entwicklern. Ein Unternehmen oder eine andere Organisation steht bis heute nicht hinter dem Projekt, unter der Website vuejs.org steht weiterhin „Copyright © 2014-2022 Evan You“. You selbst hat keinen Arbeitgeber im klassischen Sinn, er arbeitet Vollzeit als Vue-Maintainer ohne klassischen Arbeitsvertrag. Um sein Auskommen muss man sich aber wohl keine Sorgen machen, immerhin zählt er auf seiner Homepage Uhrensammeln als Hobby auf. Bezahlt wird er über Firmen, die im Vue-Kontext unterwegs sind und ihr Logo gern unter „Sponsors“ auf der Projektseite sehen. Das sind vor allem Agenturen, die mit Vue.js Projekte für Kunden entwickeln und Anbieter von Vue-Trainings. Außerdem gibt es Vue-Konferenzen auf der ganzen Welt, die das Kernprojekt und dessen Verantwortliche finanziell unterstützen.

Die meisten weiteren Maintainer neben You dagegen arbeiten für andere Unternehmen, darunter zum Beispiel GitLab. Bei einem genaueren Blick auf Evan Yous GitHub-Profil (Benutzername yyx990803) sticht sofort seine regelmäßige Arbeit ins Auge – es vergeht kaum ein Tag ohne Beitrag. Und anders als andere Maintainer, deren Projekte so groß geworden sind, sieht You seine Aufgabe bis heute im Programmieren: Im laufenden Jahr 2022 sind 92 Prozent seiner GitHub-Beiträge Commits, also eigener Code, nur 1 Prozent machen Code Reviews aus.

Dass You selbst so aktiv und eine zentrale Figur ist, kann – wie jede One-Man-Show – langfristig zum Problem werden. Ein gesundheitsbedingter Ausfall wäre ein ernstes Problem für das Projekt. Die Tatsache, dass keine Firma hinter dem Projekt steht, ist dagegen eher kein Problem – wenn auch ungewohnt für Deutsche, die in einem solchen Fall mindestens einen eingetragenen Verein erwarten.

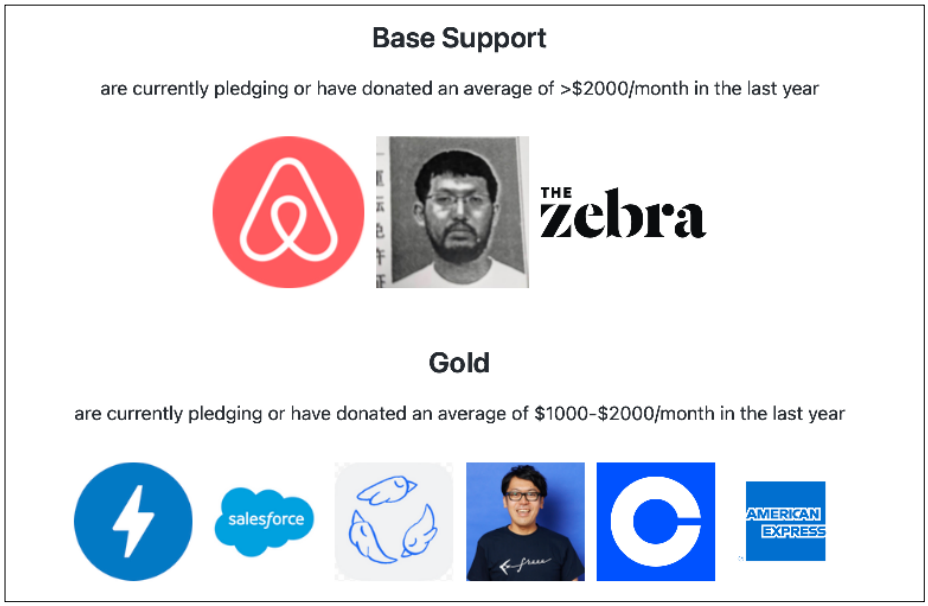
Und darunter?

Auf dem nächsten Teil der Reise gehts von der oberen Ebene des Projekts zu den Abhängigkeiten. Bei JavaScript-Projekten, die den Paketmanager NPM nutzen, findet man sie in Dateien mit dem Namen package.json. Einen besseren Überblick bekommt man mit programmiersprachenunabhängigen Software-Teilelisten, sogenannten SBOMs. Wie die funktionieren und wie man sie auswertet, lesen Sie ab Seite 68. Mithilfe des dort vorgestellten Werkzeugs Dependency-Track haben wir uns auch die Abhängigkeiten von Vue.js angesehen.

Eine kaum zu übersehende Säule, auf der Vue.js steht, sind die zahlreichen Pakete aus dem Babel-Projekt. Babel (babeljs.io) übersetzt modernen und effizienten ECMAScript-Code, eine syntaktisch verbesserte JavaScript-Variante bei Bedarf in alte JavaScript-Syntax, die auch alte Browser verstehen. So können Entwickler alle Annehmlichkeiten der modernen Sprache nutzen, ohne den untoten Internet Explorer 11 abzuhängen oder sich um kleine Unterschiede zwischen Browsern kümmern zu müssen. Perspektivisch könnte Babel irgendwann gänzlich obsolet werden, wenn die Browserhersteller die kleinen Unterschiede beseitigt haben.

Auch hinter Babel steht kein Unternehmen und wie bei Vue gibt es auch keine natürliche Person und keine Rechtsform. Am ehesten kann man das Projekt als „nicht-eingetragenen internationalen Verein ohne festen Sitz“ beschreiben. An dieses Organisationsmodell muss man sich in diesen Teilen der Open-Source-Welt gewöhnen. Die Aufgabe eines Vereinsregisters übernehmen GitHub (für den Code) und die Plattform Open Collective (opencollective.com), auf der die Finanzierung des „Vereins“ verwaltet wird. Damit andere Unternehmen solchen losen Zusammenschlüssen Geld zukommen lassen können, gibt es dort das Modell des sogenannten „Fiscal hosting“: Eine zwischengeschaltete Organisation, in diesem Fall das „Open Source Collective“ nimmt das Geld an, verwahrt es und lässt es dem Projekt zukommen – nach Abzug von Verwaltungskosten. Die Sponsoren bekommen von der in den USA registrierten gemeinnützigen Organisation eine Bescheinigung für ihre Unterlagen.

Organisatorisch scheint Babel gut aufgestellt, sechs Maintainer teilen sich die Arbeit. Auf der Homepage erfährt man, dass zum Beispiel die Wohnungsvermittlerplattform Airbnb als „Base Supporter“ monatlich über 2000 US-Dollar an das Projekt überweist. Bis zu 2000 US-Dollar pro Monat ist das Projekt unter anderem Salesforce, Coinbase und American Express wert. Weitere Unternehmen zahlen kleinere Beträge. Was nach viel Geld klingt, ist aber offenbar nicht so rosig: Die Maintainer sollten Vollzeit von dem Geld bezahlt werden, doch dafür reichte plötzlich das Geld nicht mehr. Im Mai 2021 wandten sie sich mit einem Blogbeitrag an die Community: 333.000 US-Dollar, so rechneten sie vor, bräuchten sie, um sich überschaubare Gehälter von 2000 US-Dollar auszuzahlen – nicht gerade ein Vermögen für IT-Spezialisten. Nur knapp die Hälfte, also rund 150.000 US-Dollar kamen laut Blogbeitrag jährlich zusammen. Was zuerst stabil und solide aussieht,



Obwohl große Akteure wie Salesforce und Airbnb monatlich Geld an Babel zahlen, meldete das Projekt 2021 finanzielle Probleme. Die Maintainer zahlen sich ein überschaubares Gehalt aus.

entpuppt sich leider bei genauerem Hinsehen als allzu bröckelige Säule des Webs.

Tausendsassa

Auf ein ganz anders gelagertes Problem stießen wir beim unauffälligen Paket mit dem Namen „brotli“. Das lädt Vue.js als Abhängigkeit für Entwicklungsumgebungen, es handelt sich um eine Implementierung des Brotli-Kompressionsalgorithmus, der 2014 bei Google erfunden wurde und in Schriftarten im „Web Open Font Format“ (WOFF2) zum Einsatz kommt. Geschrieben hat das Brotli-Paket der Entwickler Devon Govett mit dem NPM- und GitHub-Benutzernamen devongovett, im Hauptberuf Entwickler bei Adobe. Das Paket selbst hat nur eine eigene Abhängigkeit, eine kleine Base64-Bibliothek namens base64-js. Auch die stammt von einem Einzelentwickler, dem Amerikaner Jameson Little (beatgammit). Dessen Base64-Implementierung ist sowas wie der Sockel zahlreicher großer Säulen des JavaScript-Universums. Laut GitHub steckt sie in über 10 Millionen Projekten. Aber auch Devon Govetts brotli ist ein viel gefragtes Paket. Es verbirgt sich in 236 anderen NPM-Paketen und hat laut NPM ganze 1,7 Millionen wöchentliche Downloads. Für ein so bedeutendes Projekt ist es erschreckend schlecht aufgestellt. Vor ziemlich genau fünf Jahren, im Mai 2017 erschien die letzte Version (1.3.2). Das NPM-Paket liegt im privaten Account von Devon Govett, das zugehörige GitHub-Repository hat er auf die GitHub-Organisation foliojs übertragen – ein Projekt, das er mit seinem Kumpel Luiz Américo zusammen betreut. Kein großes Team und kein Unternehmen.

Govett scheint ein Händchen dafür zu haben, sich Arbeit aufzuhalsen: Unfassbare 748 NPM-Pakete hat er in seiner Karriere angehäuft, längst nicht alle bekommen Aktualisierungen, wobei man manches sicher als ausentwickelt betrachten kann. Doch die schiere Menge ist nicht genug – zielstrebig oder unfreiwillig hat er sich auch Programmierherausforderungen gesucht, die elementare Probleme lösen und heute Grundlagen für andere Bibliotheken sind, die wiederum Grundlage für andere sind. Neben der Brotli-Implementierung hat er sich Encoder und Decoder für Audio, Bilder und Videos sowie weitere Kompressionsalgorithmen herausgepickt. Wer einen Beleg dafür sucht, auf welch wackligen Säulen manches in der heutigen Softwareentwicklung steht, wird bei Govett fündig. Von einem Tag auf den anderen zusammenbrechen kann aber auch dieses Gebilde nicht mehr, selbst wenn Govett im hypothetischen Fall die NPM-Welt brennen sehen wollte und bei allen Projekten den Stecker ziehen wollte: NPM verhindert mittlerweile immerhin, dass Projekte über Nacht depubliziert werden. Im schlimmsten Fall könnte er kaputte neue Versionen veröffentlichen, die alten blieben aber erhalten.

Weltpolitik und Code

Die letzte Etappe der Reise führt raus aus dem Unterbau von Vue.js in eine andere Ecke des Vue-Ökosystems. Bei der Recherche stießen wir auf eine besonders bedrückende Diskussion bei GitHub über das Projekt BootstrapVue. Das hat es sich zur Aufgabe gemacht, Komponenten aus dem CSS-Framework Bootstrap in Vue.js bereitzustellen. Die aktuell größte Aufgabe dieser Community: ihr Werk für das vergleichsweise neue Vue.js 3 und Bootstrap 5 vorzubereiten – ein umfangreiches Unterfangen. Und so war die Freude groß, als sich am 14. Dezember 2021 der GitHub-Nutzer xanf zu Wort meldete (zu finden über ct.de/yb7h) und verkündete, dass er mit einem Maintainer-Team diese Herausforderung annehmen wolle. Hinter dem Benutzernamen steckt Illya Klymov, hauptberuflich Frontend-Entwickler bei GitLab. Es folgten immer wieder Status-Updates der Umbauarbeiten und der Umbau schien auf einem guten Weg. Am 1. März 2022 dann der Schock. Wieder meldete sich xanf zu Wort, diesmal mit einer nichttechnischen Nachricht: „Sorry, no deadlines atm - we have Russia invasion in our country (Ukraine) and I'm writing this post from a shelter in Kharkiv, which is under rocket attacka for a few days.“ Seitdem gibt es dennoch Fortschritte beim Projekt, und trotz der Umstände vermeldete der Entwickler zwischendurch, aktuell an der kompliziertesten Komponente, der für Tabellen, zu arbeiten. Auch BootstrapVue organisiert seine Sponsoren über Open Collective und bezahlt Klymov darüber. Der Fall zeigt aber leider, dass Finanzierung und Organisationsstruktur manchmal nicht alles sind. Am Ende stehen auch hinter Organisationsnamen engagierte Menschen mit eigenen Schicksalen und Problemen.

Dass Einzelkämpfer mit ihren privaten GitHub-Repositories das Web zusammenhalten, ist schlicht unvernünftig.

Zu groß zum Scheitern?

Was bleibt nach dieser Reise durch die JavaScript-Paketquellen? Zum einen die Erkenntnis, dass sich Open-Source-Projekte mittlerweile sehr professionell organisieren können und Wege finden, international zusammenzuarbeiten. Um Vue.js selbst muss man sich am wenigsten Sorgen machen, aber schon beim aktuell unverzichtbaren Babel sieht es finanziell problematisch aus. Projekte, die Grundlagentechnik im Verborgenen bauen, haben es auf dem Sponsorenmarkt nicht so leicht wie ein schillerndes Frontend – schließlich liegt es ja in ihrer Natur, dass sie nicht im Mittelpunkt stehen und man von ihrer Existenz bestenfalls nichts mitbekommt.

Richtig problematisch sind die Lebenswerke von Einzelkämpfern, die teils vor vielen Jahren sehr grundlegende Probleme gelöst haben. Die stützen heute mehr oder weniger unfreiwillig die Fundamente des Webs, ohne Kontrolle, ohne Organisationsstruktur und mit vielen offenen Fragen. Hier ist klar die ganze Community gefragt: Für grundlegende Funktionen bräuchte man eine starke und die Zeiten überdauernde Organisation wie die Linux-Foundation, die den Linux-Kernel organisatorisch betreut, oder deren Tochterorganisation, die OpenJS-Foundation, die sich um NodeJS kümmert.

Als Webentwickler mögen diese Zustände gruselig klingen und dazu verleiten, künftig Anwendungen ohne fremde Hilfe und ohne Abhängigkeiten schreiben zu wollen. Richtig zielführend ist aber auch das meist nicht – und keinesfalls automatisch besser oder sicherer. Angesichts der vielen großen Unternehmen auf Nutzerseite wären Sie immerhin nicht allein, wenn irgendwo existenzielle Probleme aufträten. Wichtig ist aber in jedem Fall, sich beim Laden neuer Pakete einen kritischen Blick anzugewöhnen und jederzeit einen Überblick zu haben. Mehr dazu auf den folgenden Seiten. *(jam@ct.de)* c't

Projekte und Akteure: ct.de/yb7h