

Cloud-native Apps

„80% der Cloud-Dienste sind nicht standardisiert“

Dank vieler Vorzüge erlebt Cloud-native einen Hype. Das Konzept hat aber auch Nachteile.

Wer zur Entwicklung Cloud-nativer Applikationen Dienste eines Cloud-Anbieters einsetzt, spart viel Arbeit, begibt sich aber auch in Abhängigkeiten. Professor Nane Kratzke vom Fachbereich Elektrotechnik und Informatik an der FH Lüneburg, erklärt, wie sich das Risiko eines Vendor-Lock-ins minimieren lässt, etwa mit Hilfe von Standards.

com! professional: Laut Ihren Forschungen hat das Interesse am Thema Cloud-native Applikationen in den vergangenen zwei Jahren stark zugenommen. Wie erklären Sie sich das?

Nane Kratzke: Ich denke, das liegt unter anderem daran, dass sich das Nutzungsverhalten von Cloud-Ressourcen in den vergangenen zehn Jahren verändert hat. In den ersten Jahren mussten sich Software-Entwickler und -Architekten erst auf die neue Umgebung Cloud und deren Eigenschaften einstellen. Anfangs wurden einfach klassische 3-Tier-Web-Applikationen auf die Cloud-Infrastruktur ausgerollt, da hat sich eigentlich nur der Ort verändert, wo die Applikationen betrieben wurden, nicht ihr Aufbau. Firmen wie Netflix sind hier einen wesentlichen Schritt weitergegangen, um Cloud-Ressourcen besser und horizontal skalierbarer nutzen zu können. Sie setzen auf Konzepte wie Microservices und Cloud-native Applikationen. Ihr Erfolg hat sicher dazu beigetragen, dass diese Begriffe so populär geworden sind.

com! professional: Andererseits warnen Sie davor, dass man sich damit in eine starke Provider-Abhängigkeit begibt.

Kratzke: Die Cloud-Betreiber bieten sehr viele Dienste an, die nach außen gar nicht in Erscheinung treten, die aber unter der Haube den Software-Entwicklern sehr viel händische Arbeit abnehmen – sowohl in der Entwicklung als auch im Betrieb solcher Applikationen. Ich nenne solche Angebote gerne „Klebedienste“, weil sie zum einen Einzeldienste komfortabel zu höherwertigen Applikationen verbinden, aber eben auch den Kunden stark an den jeweiligen Provider binden.

com! professional: Beschleunigen aber nicht gerade diese Services die Entwicklung Cloud-nativer Applikationen so enorm?

Kratzke: Ja, wenn Time-to-Market die oberste Priorität ist, dann ist das sicher die richtige Strategie. Man sollte sich eben

nur wirklich bewusst sein, dass man diese Bindung eingeht. Mein Eindruck ist, dass sich viele Firmen darüber nicht im Klaren sind, insbesondere wenn es die erste Applikation ist, die sie in dieser Form erstellen.

com! professional: Warum ist es überhaupt schlecht, sich an einen Provider zu binden?

Kratzke: Es muss nicht notwendigerweise schlecht sein, ist aber eben mit einem Risiko verbunden, das man kennen sollte. Aktuell mag der gewählte Cloud-Service-Provider die beste Lösung sein, aber es ist kaum vorhersagbar, ob das in einigen Jahren immer noch so ist. Vielleicht gerät er in wirtschaftliche Schwierigkeiten, die rechtlichen Rahmenbedingungen ändern sich, Service Level Agreements werden einseitig abgesenkt, andere Provider bieten bessere Konditionen. Diese Liste ist vermutlich endlos.

Dr. Nane Kratzke

Professor am Fachbereich
Elektrotechnik und
Informatik an der FH Lüneburg
www.fh-luebeck.de



Foto: Nane Kratzke / FH Lüneburg

com! professional: Helfen Standards gegen Vendor-Lock-ins?

Kratzke: Standards existieren im Cloud-Umfeld eher auf den unteren Ebenen des Cloud-Stacks, im Bereich Infrastructure as a Service. Je höher man im Stack kommt, desto mehr Dienste gibt es, die derzeit noch überhaupt nicht von einem Standard abgedeckt werden. Hinzu kommt, dass die Cloud-Provider ihr Angebot an proprietären Services mit enormer Geschwindigkeit ausbauen. Wir haben uns das in unserem Forschungsprojekt CloudTRANSIT am Beispiel Amazon Web Services rückblickend angesehen. Im Jahr 2006 betrieb AWS drei Dienste, zwei davon waren inhaltlich durch heutige Standards abgedeckt gewesen. Heute betreibt AWS mehr als 50 Dienste und nur etwa zehn davon sind durch heutige Standards inhaltlich erfasst – und ich rede hier nicht einmal von der Standard-Konformität dieser Dienste. Nehmen wir andere Provider, ändern sich zwar die absoluten Zahlen, aber die

„Cloud-Provider bauen ihr Angebot an proprietären Services mit enormer Geschwindigkeit aus.“

Verhältnisse bleiben etwa gleich. Anders gesagt: Wenn Sie heute einen Cloud-Dienst unreflektiert nutzen, ist dieser mit 80-prozentiger Wahrscheinlichkeit nicht von heutigen Cloud-Standards inhaltlich erfasst und damit auch kaum durch einen standardisierten Service ersetzbar.

com! professional: Wie erklären Sie sich das?

Kratzke: Standardisierungsprozesse sind einfach langsam. Amazon hat es geschafft, pro Jahr durchschnittlich zwei neue Dienstkategorien zu kreieren, während in den Standards im selben Zeitraum nur eine Kategorie hinzukam. Die Entwicklung neuer Dienste läuft also doppelt so schnell ab wie die Standardisierung, die Schere wird dadurch immer größer.

com! professional: Gibt es andere Möglichkeiten, sich vom Cloud-Provider unabhängiger zu machen?

Kratzke: Man kann Cloud-native Applikationen beispielsweise nicht direkt auf der Infrastruktur des Cloud-Anbieters betreiben, sondern auf einer Selfservice-Automation-Plattform wie Docker Swarm oder Kubernetes. Diese Plattformen erfahren daher auch zunehmendes Interesse. Wenn diese auch noch Open Source sind, haben Sie so etwas wie eine verteilte virtuelle Maschine für die Cloud, die unter Ihrer eigenen Administrations-Hoheit steht. Diese Plattformen können Sie nicht nur über Cloud-Provider hinweg migrieren, sondern auch zeitgleich bei unterschiedlichen Providern betreiben. Sie könnten also einen Teil Ihrer Applikation in der Amazon-Cloud laufen lassen, einen anderen in der Private Cloud.

com! professional: Muss man die Plattformunabhängigkeit bereits während der Entwicklung in die Applikationen einbauen?

Kratzke: Anwendungen im Nachhinein von einem Cloud-Vendor-Lock-in zu befreien, ist sehr viel schwieriger, als wenn man dies von Anfang an bei der Entwicklung berücksichtigt. Instagram etwa betrieb seine komplette Infrastruktur auf AWS, als es 2012 von Facebook gekauft wurde. Es kostete die komplette Instagram-Entwicklerrmannschaft fast ein Jahr, den Umzug in die Facebook-Rechenzentren zu bewerkstelligen.

com! professional: Ist es denn immer besser, eine Applikation Cloud-native zu entwickeln?

Kratzke: Das kann man in meinen Augen nicht pauschal beantworten. In dem Moment, wo ich Dienste anbieten will, die Hunderttausende oder gar Millionen von Kunden adressieren, ist es sicher sinnvoll, die Systeme von vornherein auf eine hohe horizontale Skalierbarkeit hin zu entwickeln. Habe ich eine überschaubare, definierte Nutzerzahl – wie bei unserem E-Learning-System an der Hochschule Lüneburg –, dann lässt sich das sehr gut auch mit einer klassischen Software-Architektur abbilden. Sicher schadet auch in so einem Fall die Flexibilität und Skalierbarkeit einer Cloud-nativen Applikation nicht, vermutlich würde man sie aber nie nutzen. Der Fall läge vollkommen anders, wenn man das System perspektivisch als Dienst für alle deutschen Hochschulen bereitstellen wollen würde, dann reden wir von Millionen Nutzern.

com! professional: Wie sieht es mit den Kosten aus? Ist es nicht günstiger, eine Applikation in der Cloud zu betreiben?

Kratzke: Nicht unbedingt. Prognosen für Cloud-Kosten sind schwierig und ein Kapitel für sich. Als „Daumenregel“ gilt: Applikationen in der Cloud sind aus Kostensicht immer dann sinnvoll, wenn die Zugriffszahlen kaum vorhersehbar sind oder in eng begrenzten Zeiträumen sehr hohe Spitzenlasten auftreten. Bei relativ statischen Workloads und geringen Skalierungserfordernissen ist es meist günstiger, die Infrastruktur im eigenen Haus zu betreiben. Das ist aber wie gesagt eine Daumenregel: Da sollte man jeden Fall einzeln betrachten.

com! professional: Cloud-native ist also nicht in jedem Fall die sinnvollste Strategie?

Kratzke: Wenn man Applikationen grundsätzlich nach den Cloud-nativen Prinzipien konzipiert, macht man vermutlich nichts falsch. Man muss sich allerdings auch mehr Gedanken bei der Entwicklung machen und hat unter Umständen viele Aufwände, die im normalen, überschaubaren Betrieb womöglich kaum etwas bringen. Hier ist es wichtig zu wissen, welche Perspektiven man mit einem System verfolgt. Es ist in etwa so, als wenn man jeden Mittelklasse-Pkw für die Höchstgeschwindigkeit eines Formel-1-Rennwagens konzipieren würde. Es schadet nicht, aber der zusätzliche Engineering-Aufwand wäre für den Fahralltag vermutlich unerheblich.

Thomas Hafen/js
js@com-professional.de

