



□ Prof. Dr. Nane Kratzke

[nane.kratzke@fh-luebeck.de]

ist Professor für Informatik an der Fachhochschule Lübeck, Projektleiter von Forschungsprojekten und Autor zahlreicher wissenschaftlicher Buch-, Zeitschriften- und Konferenzbeiträge sowie Sprecher auf Cloud Computing-Konferenzen. Sein Forschungsinteresse gilt der modellbasierten Systementwicklung und Cloud-nativen Systemen. Insbesondere die Beherrschbarkeit dieser Technologien für kleine und mittlere Unternehmen ist ihm ein besonderes Anliegen. Zuvor war er als Berater, Softwarearchitekt und Software Engineer zumeist für hochkritische Systeme tätig.

## objektspektrum themenspecial: microservices und docker

### Microservices und Docker – Eine Renaissance bewährter aber lange übersehener Technologien?

Microservices und Container sind derzeit der Trend in der Softwareentwicklung. Das ist eigentlich erstaunlich. Weder das Service-orientierte Paradigma noch Container-Technologien sind wirklich neu. Service-orientierte Architekturen (SOA) werden seit mindestens Mitte der 1990er-Jahre thematisiert. Dass Container-Technologien wie z. B. Docker eine äußerst skalierbare und performante Alternative zu Hypervisoren sein können, haben Forscher der Princeton Universität bereits 2007 auf der „European Conference on Computer Systems“ zeigen können. Container-Technologien sind somit schon seit fast einer Dekade im Fokus der Aufmerksamkeit. Zugegeben, die Aufmerksamkeit war nicht sonderlich breit gestreut. Die technischen Basis-konzepte der Betriebssystem-Virtualisierung von Containern sind sogar noch weitaus betagter. Dieses Themenspecial zeigt, warum ausgerechnet solche etablierten – wenn auch lange übersehenen – Technologien und Ansätze die IT-Welt momentan ordentlich umkrepeln.

Sogenannte „Cloud-native Systeme“ werden explizit für die Cloud entwickelt. Es handelt sich häufig um automatisch skalierende und somit elastische Systeme. Diese Skalierbarkeitserfordernisse legen nahe, auf „monolithische Ansätze“ in Cloud-nativen Architekturen zu verzichten. Es kann sonst schwierig werden, die Vorteile des Cloud Computing zu nutzen.

Aber auch für andere (klassischere) Systeme sind bewährte Cloud-Native-Prinzipien nicht von Nachteil. Die Systeme werden dadurch meist fehlertoleranter, robuster und wartbarer. Selbst wenn man

nie beabsichtigt, solche Systeme jemals in der Cloud auszubringen.

Dabei gilt es aber einige Prinzipien zu beachten und zu verstehen. Unsere Beiträge im Bereich *Microservices* beleuchten wesentliche architekturelle, organisatorische, benutzungstechnische und unternehmenskulturelle Aspekte. Unsere *Container*-Beiträge fokussieren natürlich Docker und sein umfassendes und schnell wachsendes Ökosystem. Sie beleuchten vor allem die technologische Komponente.

Aber auch andere Aspekte werden behandelt, z. B. wie Vendor Lock-In systematisch umgangen werden kann. Grau ist

alle Theorie. Spannend sind daher auch die Beiträge, die konkrete *Anwendungen und Fallstudien* beleuchten.

Unsere Autoren zeigen, wofür man Microservices und Container auch einsetzen kann. Das sind nicht immer nur die offensichtlichen Anwendungsfälle des Cloud Computing. Und natürlich berichten unsere Autoren über konkrete Erfahrungen mit den in diesem Themenspecial vorgestellten Methoden und Technologien.

#### Microservices

Dipl.-Inf. Eberhard Wolff führt uns hierzu ein wenig in die Geschichte. Er beleuchtet

Ansätze wie SOA, die auf den ersten Blick sehr ähnlich zu Microservices sind, er zeigt aber auch Unterschiede zu Microservices auf und denkt Microservices stringent weiter. Während die Servicegröße bei SOA vielleicht als Makroservice beschrieben werden könnte, sind Microservices sehr viel kleiner. Der historische Rückblick lehrt häufig, dass ein Pendel erst von einem Extrem ins andere Extrem schwingt, um sich dann irgendwo in einer vernünftigen Mitte einzupendeln. Der Autor führt den Leser so zu Self-contained Systems (SCS) – vielleicht ein neuer Trend.

Dass SCS Aufmerksamkeit verdienen, findet zumindest auch **Richard Attermeyer**. Er widmet sich unter anderem dem Thema der Frontend-Architekturen und vertieft damit einen spannenden Aspekt von SCS. Insbesondere die weitere Differenzierung der Endgeräte in den kommenden Jahren erfordert eine Frontend-Architektur, die eine kontextabhängige Geräternutzung systematisch unterstützt. Mit der „Context-Aware Frontend Architecture“ (CAFA) wird der Autor hier erfrischend konkret und liefert dem Leser sicher neue Ideen und Impulse.

Microservices lösen viele Probleme. Microservices sind aber natürlich nicht „for free“. Um die Vorteile von Microservice-Architekturen wirklich voll ausspielen zu können, sind meist auch Anpassungen in den Bereichen DevOps, Organisation und Governance erforderlich. **Dr. Henning Sternkicker** und **René Meyer** beleuchten, wie das Potenzial von Microservice-Ansätzen technisch wie organisatorisch voll ausgeschöpft werden kann. Wichtig ist dabei, Entwicklungsteams die notwendigen Verantwortlichkeiten für agile Methoden konsequent zuzugestehen. Gerade dies stellt in vielen Unternehmen aber häufig einen enormen Transformationsprozess dar.

### Container

Container-Technologien (insbesondere Docker) haben sich in den letzten Jahren zunehmend mehr zum technischen Rückgrat von Microservice-Ansätzen entwickelt. **Prof. Dr. René Peinl** beschäftigt sich

in seinem Beitrag vor allem mit Docker und seinem vielschichtigen Ökosystem. Die Handhabung einzelner Docker-Container ist recht einfach (und daher so beliebt bei Entwicklern). Die Komplexität steigt jedoch mit großen Umgebungen und vor allem bei elastischen Anwendungsfällen.

Man versucht dies u. a. durch automatisierte Orchestrierung beherrschbar zu machen. Doch hier merkt man noch die Kinderschuhe. Zwar ist eine Vielzahl an Open-Source-Werkzeugen entstanden, die das eine oder andere Problem mildern. Jedoch gibt es (noch) kaum umfassende Lösungen aus einem Guss. Der Autor hat dennoch für den Leser einige wertvolle Empfehlungen für bewährte Technologie-Stacks. Ferner liefert der Beitrag einen hervorragenden Überblick über die Docker-Landschaft. Mehr Tools rund um Docker auf weniger Seiten findet man aktuell nur schwer.

**Dipl.-Inf. Peter-Christian Quint** betrachtet Container-Technologien aus einem anderen, häufig übersehenen Blickwinkel. Er zeigt, wie Container-Cluster – mit Bedacht eingesetzt – Vendor Lock-In im Cloud Computing nachhaltig lösen können. Vendor Lock-In-Befürchtungen sind immer noch ein Hauptgrund, warum sich Unternehmen gegen die (public) Cloud entscheiden. Der Autor gewährt so ganz nebenbei interessante Einblicke in ein aktuelles Forschungsprojekt der Fachhochschule Lübeck, das vor allem für kleinere und mittlere Unternehmen von Interesse sein kann.

### Anwendungen und Fallstudien

Container-Technologien und Service-Ansätze sind nicht nur für die Systeme und deren Architekturen nutzbar. Man kann die gleichen Ansätze auch zur Qualitätssicherung einsetzen. **Adam Trujillo** zeigt, wie man Testumgebungen auf Basis von Service-Virtualisierung und Containern nutzen kann, um Testabläufe und Testumgebungen sowohl flexibel und problemangemessen definieren, als auch zwischen Abteilungen friktionsfrei teilen zu können. Aus dem Entwickleralltag kennt man

bislang meist ein „Entweder-oder“ und selten ein „Sowohl-als-auch“.

**Dipl.-Wirt.-Inf. (BA) Jochen Christ** zeigt in seinem Beitrag, dass Microservices auch ohne Docker funktionieren können. Am Beispiel des paydirekt-Systems wird belegt, wie schnell man auch in kritischen und komplexen Systemen neue Funktionen realisieren kann. Spannend ist das deswegen, weil wir alle doch ein besonders scharfes Auge haben, wenn es um unser Geld und unsere Konten geht.

Noch beeindruckender sind die Reduktionen bei Release-Zyklen. Während bei Banken oft nur zwei bis drei Zahlungssystem-Releases pro Jahr inklusive „Wochenend-Downtimes“ üblich sind, hat man sich bei paydirekt zu zweiwöchentlichen Releases entschieden und kann so wesentlich agiler auf Feedback von Kunden, Händlern und Banken reagieren. Der Beitrag zeigt, wie eine Microservice-basierte Architektur dies (auch ohne Docker) ermöglicht.

Ich wünsche Ihnen eine interessante Lektüre bei diesem Themenspecial zu „Microservices und Docker“ und einige Erkenntnisse und Anregungen. Service-basierte Ansätze und Container-Technologien mögen nicht die neuesten Ansätze sein. Man darf sie – in die IT-Historie blickend – getrost als bewährt (wenn auch lange übersehen) bezeichnen.

Genau das könnte der Grund sein, warum diese Ansätze in den letzten Jahren eine Art „Renaissance“ erlebt haben. „Alter Wein in neuen Schläuchen“ darf einem da durchaus in den Sinn kommen. Man kann sich aber auch auf den Standpunkt stellen, dass ein guter Wein offenbar endlich seine Reife erlangt hat. Vieles deutet meiner Meinung nach auf letztere Interpretation hin. Aber bilden Sie sich lieber Ihre eigene Meinung. ■

Ihr

