CLOUD NATIVE COMPUTING

CLOUD-NATIVE

Unit:

Cloud Computing

(3) Eine kurze Geschichte der Cloud

creative commons

CLOUD NATIVE COMPUTING

Urheberrechtshinweise

Diese Folien werden zum Zwecke einer praktikablen und pragmatischen Nutzbarkeit im Rahmen der **CCo 1.0 Lizenz** bereitgestellt.

Sie dürfen die Inhalte also kopieren, verändern, verbreiten, mit eigenen Inhalten mixen, auch zu kommerziellen Zwecken, und ohne um weitere Erlaubnis bitten zu müssen.

Eine Nennung des Autors ist nicht erforderlich (aber natürlich gern gesehen, wenn problemlos möglich).

Diese Folien sind insb. für die Lehre an Hochschulen konzipiert und machen daher vom **§51 UrhG (Zitate)** Gebrauch.

Die CCo Lizenz überträgt sich nicht auf zitierte Quellen. Hier sind bei der Nutzung natürlich die Bedingungen der entsprechenden Quellen zu beachten.

Die Quellenangaben finden sich auf den entsprechenden Folien.

KAPITEL 2 + 4

Cloud Computing und Cloud-native

2 Cloud Computing

2.1 Service Modelle

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Software as a Service (SaaS)

2.2 Cloud-Ökonomie

- Eignung von unterschiedlichen Arten von Workloads
- Effekt von Zuteilungsdauer und Ressourcengröße

2.3 Entwicklung der letzten Jahre

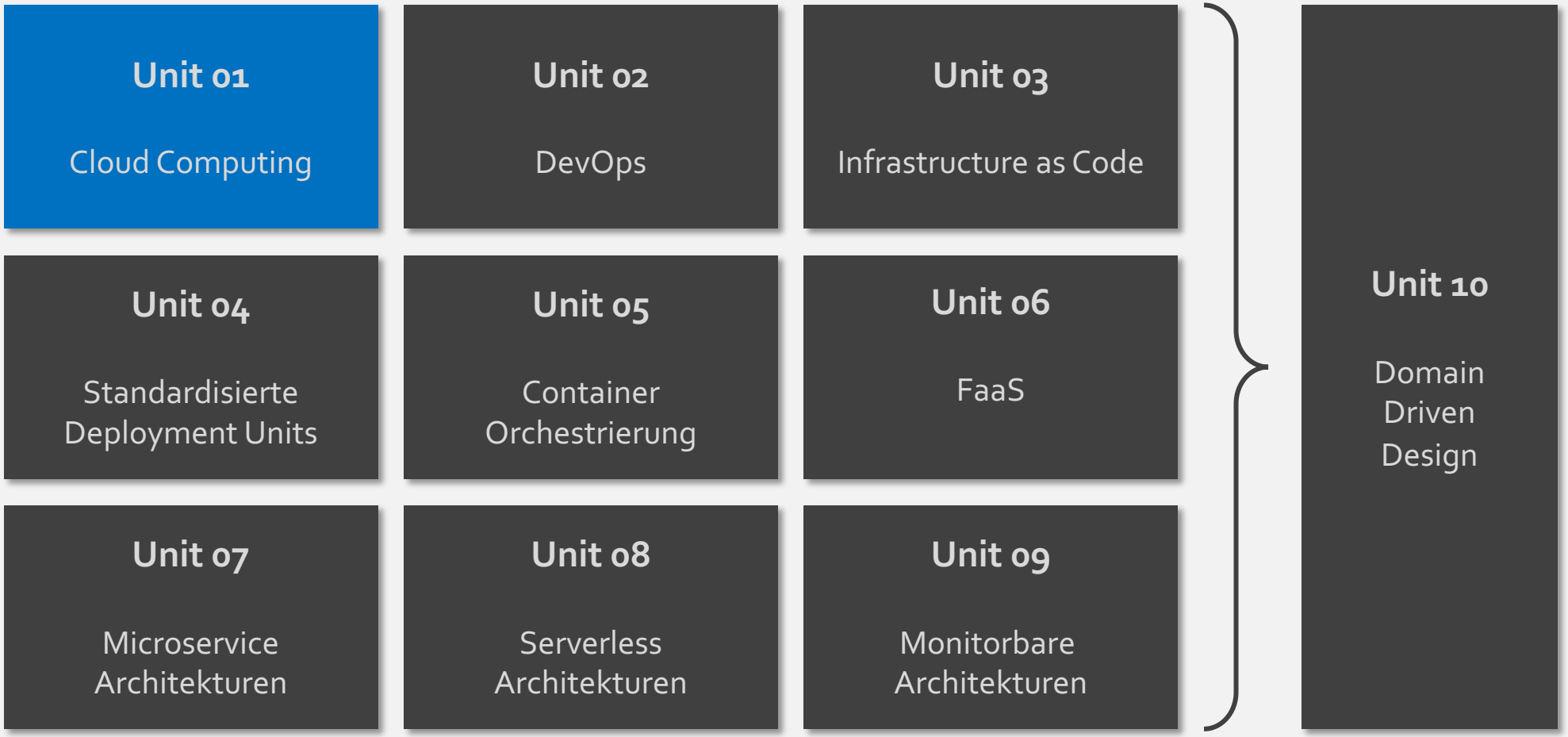
4 Cloud-native

- Definitionen in Industrie und Forschung
- Die Cloud-native Definition dieses Buchs
- Zusammenfassung und Ausblick auf Teil II bis IV

CLOUD NATIVE COMPUTING

INHALTSVERZEICHNIS

Überblick über Units und Themen dieses Moduls

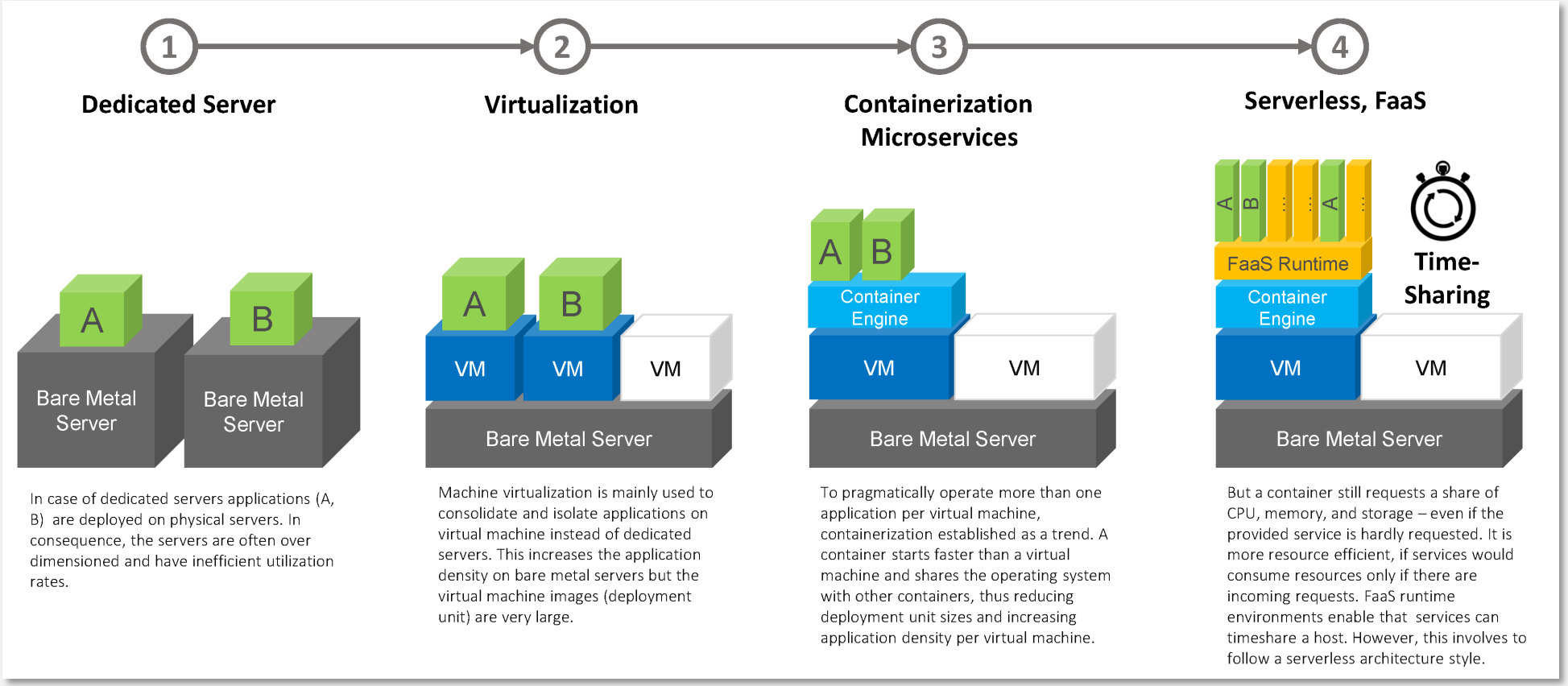


INHALTE

- Cloud Computing
- Cloud-Ökonomie (Crash-Kurs)
- **Eine kurze Geschichte der Cloud**
- Implikationen für die Entwicklung und den Betrieb von Cloud-nativen Anwendungen und Dienste

EINE KURZE GESCHICHTE DER CLOUD

Die Verkleinerung von Deployment Units im Verlaufe der Zeit



Quelle: Kratzke, N. A Brief History of Cloud Application Architectures. Appl. Sci. 2018, 8, 1368.

CLOUD NATIVE COMPUTING

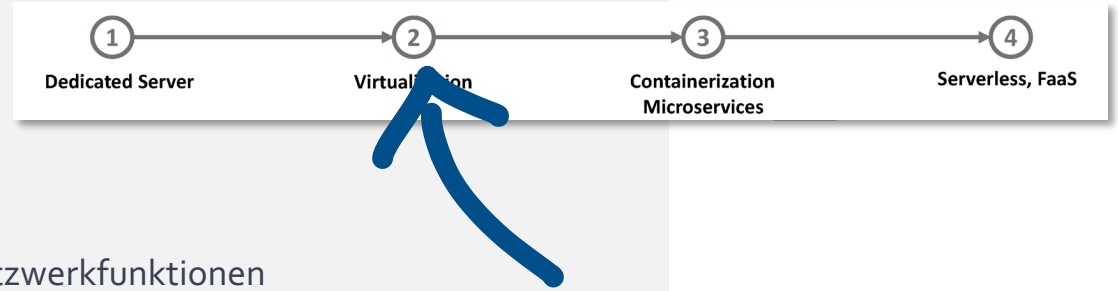
Im Verlaufe der Zeit sind die Größen von Deployment Units geschrumpft!

VM -> Container -> Function

Parallel dazu haben sich Servic-orientierte Architekturen (weiter)entwickelt.

Microservices + Serverless Architectures

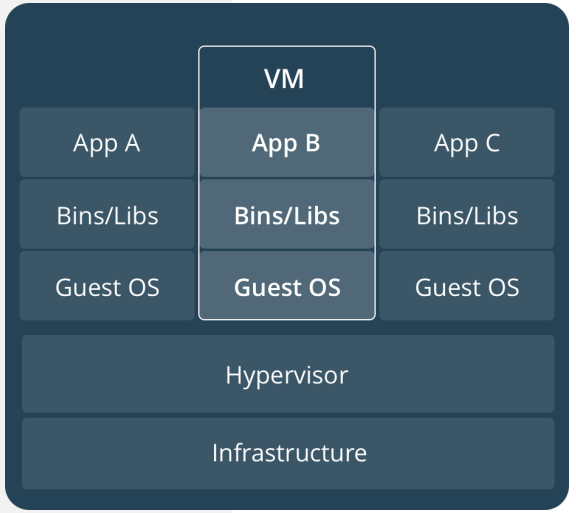
EINE KURZE GESCHICHTE DER CLOUD



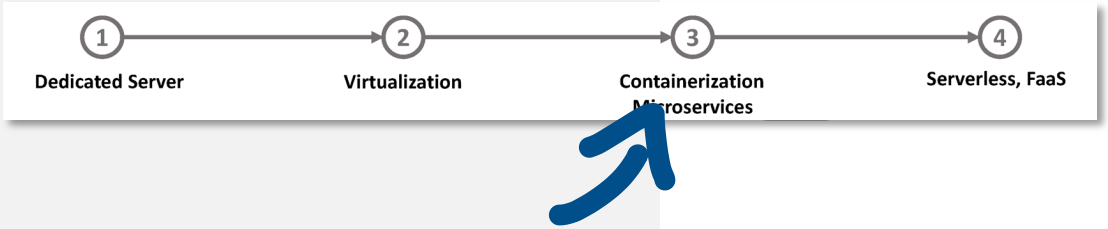
Virtualisierung

Bei der Virtualisierung werden physikalische Ressourcen wie Server, Speicher oder Netzwerkfunktionen in mehrere virtuelle Instanzen aufgeteilt. Virtualisierung ermöglicht es, eine physische Ressource als mehrere virtuelle und voneinander isolierte Ressourcen bereit zu stellen. Diese Technologie hat die Effizienz und Flexibilität von Rechenzentren und anderen IT-Umgebungen erheblich verbessert.

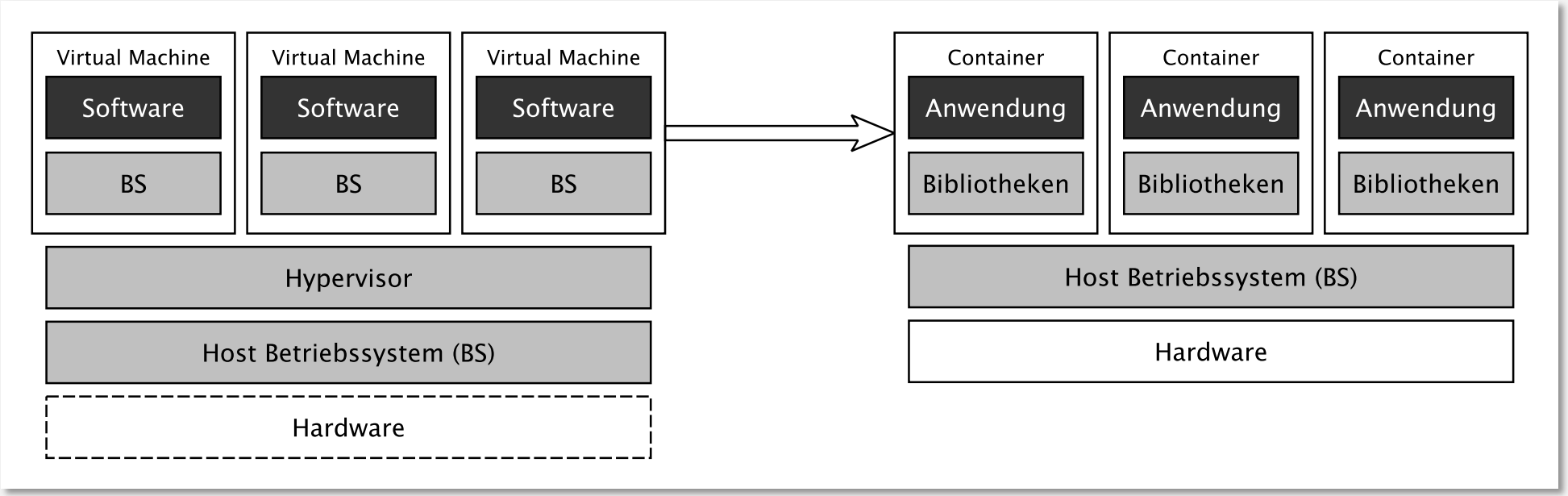
- **Ressourceneffizienz:**
  Durch die Aufteilung physischer Ressourcen in mehrere virtuelle Instanzen kann die Hardware besser ausgelastet werden.
- **Kosteneinsparungen:**
  Durch effizientere Ressourcennutzung können Kosten für Hardware, Energie und Wartung gesenkt werden.
- **Flexibilität und Skalierbarkeit:**
  Virtuelle Ressourcen können schneller als physische Ressourcen hinzugefügt, angepasst oder entfernt werden, was die Anpassung an sich ändernde Anforderungen erleichtert.
- **Einfacheres Management:**
  Durch zentrale Verwaltungstools wird die Komplexität der IT-Infrastruktur reduziert.
- **Isolation und Sicherheit:**
  Virtuelle Maschinen sind voneinander isoliert, was die Sicherheit erhöhen kann.



EINE KURZE GESCHICHTE DER CLOUD



Containerisierung



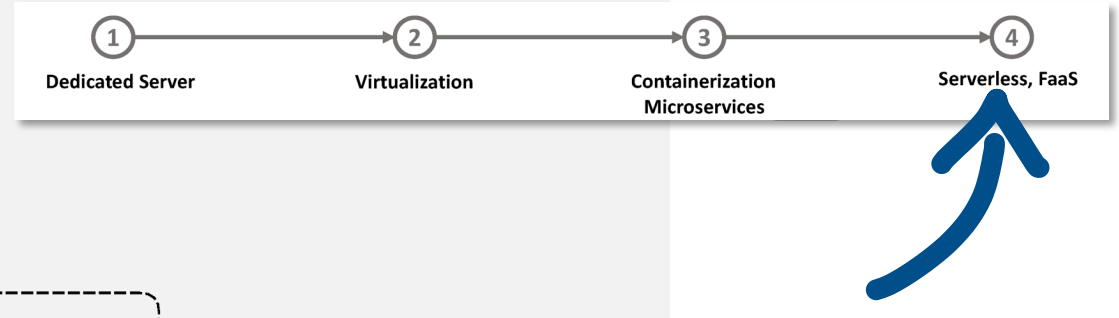
Überwindung der 1-vCPU-Schwelle!

Container nutzen Betriebssystem-Virtualisierung (Prozessisolation) anstelle von Maschinen-Virtualisierung.

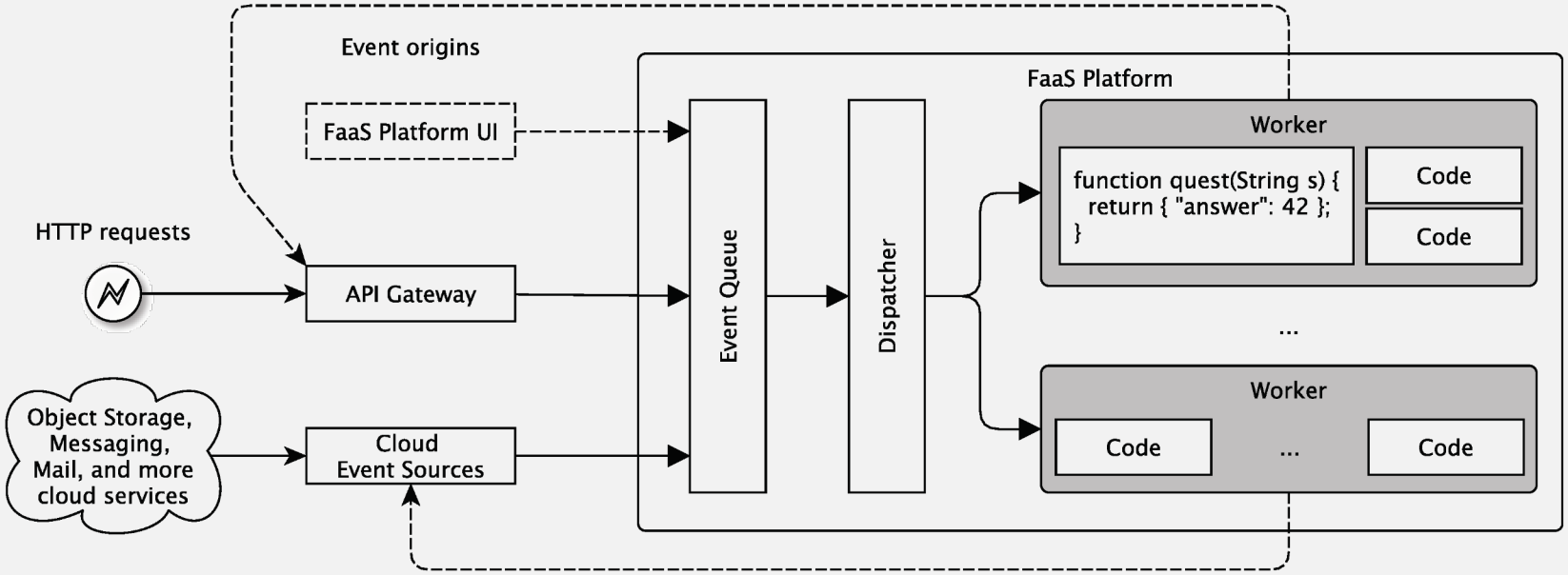
Nachteil:
Container-OS und Host-OS müssen identisch sein (meist Linux).

Vorteil:
Container starten sehr viel schneller und können wesentlich kleiner sein.

EINE KURZE GESCHICHTE DER CLOUD



Serverless Computing aka Function as a Service (FaaS)



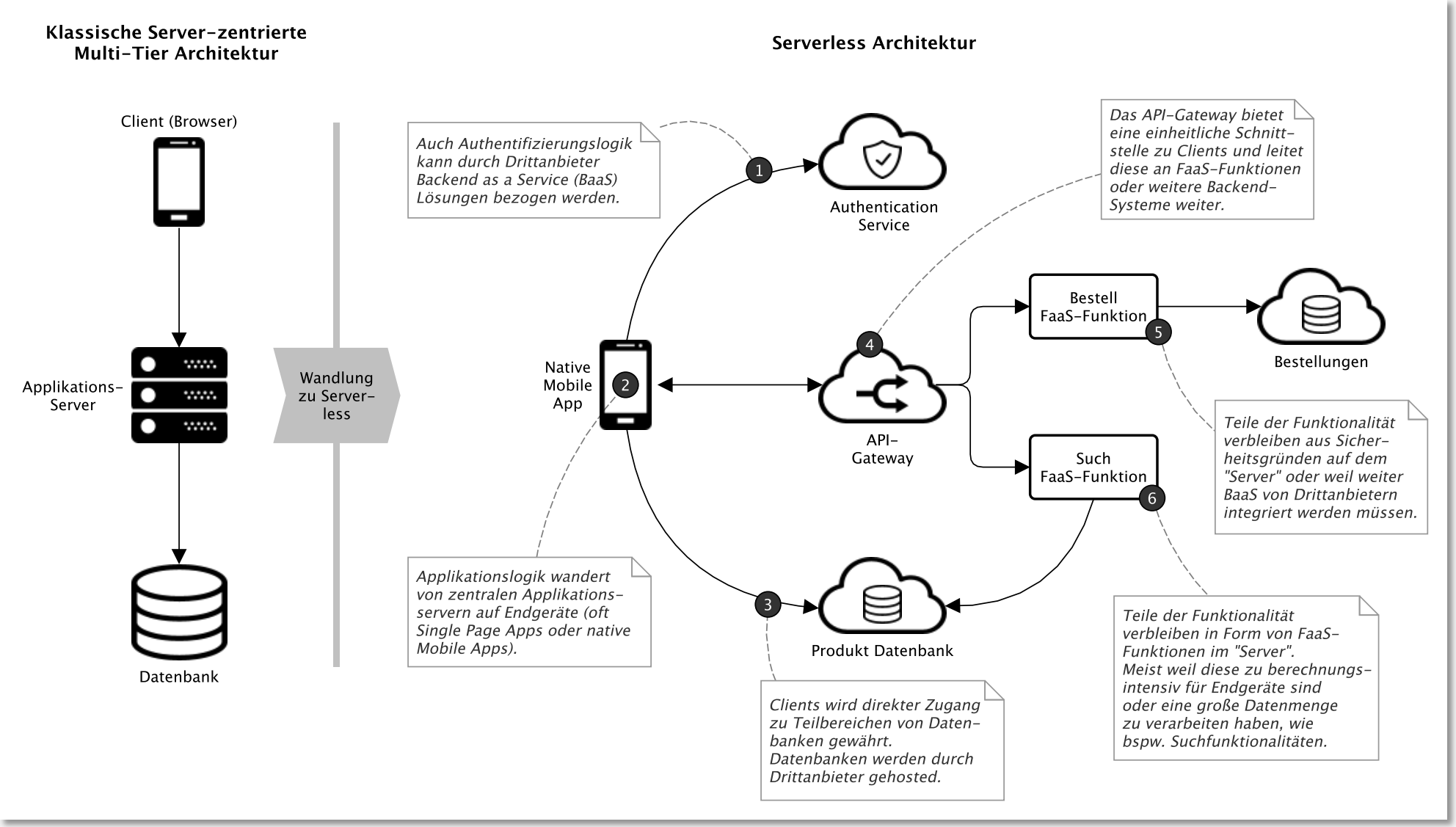
Eine serverlose Plattform ist lediglich ein Ereignisverarbeitungssystem. Ereignisse können z.B. über HTTP gesendet oder von weiteren Ereignisquellen (aus Cloud-Infrastrukturen) empfangen werden. Die Plattform bestimmt dann, welche Funktionen für ein Ereignis registriert sind, sendet das Ereignis an die Funktionsinstanz und wartet auf eine Antwort um diese Antwort an den Requestor weiterzuleiten (im Falle von Request-Response-basiertem Triggering).

Siehe auch Unit 06! FaaS

Quelle: Nane Kratzke: A Brief History of Cloud Application Architectures. In *Applied Sciences*; MDPI, 2018, https://doi.org/10.3390/app8081368

SERVERLESS ARCHITEKTUREN

Containerisierung in Cloud-nativen Anwendungen zu Ende gedacht



AUSBLICK

- Cloud Computing
- Cloud-Ökonomie (Crash-Kurs)
- Eine kurze Geschichte der Cloud
- **Implikationen für die Entwicklung und den Betrieb von Cloud-nativen Anwendungen und Dienste**

CLOUD NATIVE COMPUTING

KONTAKT

Disclaimer

Nane Kratzke

+49 451 300-5549

nane.kratzke@th-luebeck.de

kratzke.mylab.th-luebeck.de