

Kritischer Blick

Die Cloud bringt viele Vorteile, z.B.:

- Geringe Verbauungstiefe
 - Hohe Agilität
 - Niedrige Kosten (für die Arbeit)
- Flexible Architektur
- „Cloud Native“ Entwicklung: Relativ einfache Regeln und Heuristiken (Siehe zB. *12 Factor App* und *Cloud Native Application Reifegrad Modell*) führen zu gut zugeschnittener Software

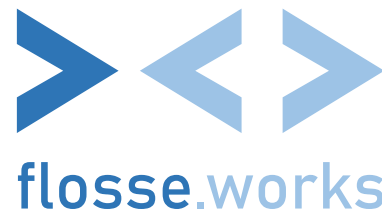
Kritischer Blick: Nachteile



... und wie alles auch Nachteile, z.B.:

- Verteilte Systeme sind komplex
- Parallele Programmierung ist nicht einfach
- Typischerweise schlechte SLAs bei gleichzeitig hohen (Infrastruktur-)Kosten
- Vendor Lock-In

Komplexität



- Viel Abstraktion, viel Indirektion
 - Mehr Gelegenheit für Fehler
 - Erschwerte Fehlersuche
- Viele Komponenten, viele Köche
 - Wer/was ist zuständig?
- Durchwachsene Qualität auch von Kernkomponenten
 - Heise: [Kubernetes – Containerland ist abgebrannt](#)

- Schadenfreude bei vielen Experten: Amazon Prime Video baut einen Teil ihrer Serverless Software zurück und spart 90% Kosten
 - Link: [Prime Video Tech Blog: move to a monolith](#)
- Aber: Architektur ist gleich geblieben, Änderung schnell vollzogen
 - Link: [Heise: Services neu zuschneiden macht noch keinen Monolithen](#)

Vendor Lock-In vs. Cloud Exit Strategy



KPMG Blog:

Why a "Cloud Exit Strategy" is essential to enable the future

- SLAs halten nicht immer, der Cloud-Markt ist tough
- Dienste werden abgekündigt
- Gesetzliche Anforderungen können sich ändern (GDPR)
- Aber: „Cloud Exit“ meint hier eigentlich Migrierbarkeit – insbesondere zwischen Clouds

- Die Cloud bietet viel Potential
- ... aber auch viele Fallstricke
- Nicht verunsichern lassen, auch hier kochen alle bloß mit Wasser und wollen etwas verkaufen
- => Gut überlegen, welche Anforderungen man hat, und nach diesen entscheiden.