Alle wichtigen Infos auf einen Blick

MODULARISIERTE MONOLITHEN

Begriffsdefinitionen

- Ein System ist in **Module** aufgeteilt.
- Module haben eine **Schnittstelle**.
  - *Implementierung dahinter änderbar, ohne Beeinflussung der Nutzung.*
  - *Beispiele für Module: Java Packages, C++ Namespaces oder Libraries*
- **Microservices** sind Module, die getrennt deployt werden können.
- **Deployment Monolithen** lassen sich nur als Ganzes deployen.
- **Architektur Monolithen** haben keine gute Aufteilung in Module *(getrennte Entwicklung nicht möglich)*.
- **Verteilter Monolith**: Mit Microservices gebauter Architektur Monolith.

Was ist ein modularisierter Monolith?

- Deployment Monolith
- gute Aufteilung in Module
- auch **Modulith** genannt

Alternativen

- Microservices
- Monolith
- Big Ball of Mud
- Mischformen

Werkzeuge

- Spring Modulith
- ArchUnit
- SonarQube
- ...

Vorteile von Modulithen

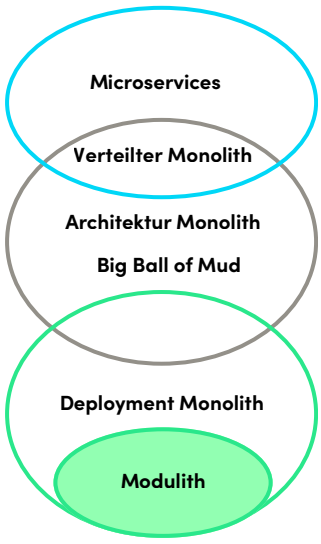
- geringeres Infrastruktur-Investment verglichen mit Microservices
- einfach verstehbar
- einfach änderbar

Nachteile von Modulithen

- Deployment dauert lange *(werden als Ganzes deployt)*
- Rollback auf alte Version schwierig
- Testausführung dauert lange *(werden meist als Ganzes getestet)*

Hohe Relevanz von Modulithen

- Ende des Hype um Microservices und gewonnene Erkenntnisse:
  - Overhead beim Betrieb
  - bei ungeschickter Aufteilung nicht leichter verständlich oder änderbar



Inhalte von **Eberhard Wolff**

SOCREATORY
The Software Creators' Academy