



Software Engineering Seminar (SoSe 2016)

Capacity and Stability (Anti-)Patterns for Microservice Architectures

Recently, the microservice architectural style is gaining more and more attraction when (re)developing enterprise applications. As opposed to monolithic architectures, microservice architectures comprise an assembly of services loosely coupled via APIs (e.g., REST). One of their design principles is “design for failure”. That means that these architectures/services should be designed to cope with failures (e.g., unavailability, poor QoS) of hardware and services. A couple of patterns or guideline have been proposed to implement this characteristic, e.g., fail fast, timeouts, circuit breakers.

This seminar paper should *i.)* provide a brief introduction into the microservice architectural style, *ii.)* present common capacity/stability anti-patterns and patterns, *iii.)* and have a look into technologies implementing selected patterns.

In addition to studying and summarizing the research literature, it is a mandatory part of this seminar to gather and share hands-on experience with the available tooling infrastructure. The provided references are to be considered a starting point and it is expected to extend the literature search and present a coherent view on the current state of the art in this area.

References

- [1] Martin Fowler. Microservices resource guide. <http://martinfowler.com/microservices/>.
- [2] Netflix Inc. Netflix Open Source Software Center. <https://netflix.github.io/>.
- [3] Sam Newman. *Building Microservices—Designing Fine-Grained Systems*. O’Reilly Media, 2015.
- [4] Michael Nygard. *Release It!: Design and Deploy Production-Ready Software (Pragmatic Programmers)*. Pragmatic Bookshelf, 2007.

Contacts

André van Hoorn (van.hoorn@informatik.uni-stuttgart.de)
Reliable Software Systems (RSS) Group
Institute for Software Technology (ISTE)
University of Stuttgart