

The background of the slide is a grayscale image of a circuit board. It features a central dark horizontal band. Above and below this band, there are intricate patterns of circuit traces, including straight lines, right-angle turns, and circular pads. Four large black circles are positioned along the top edge of the dark band, and four more are along the bottom edge, connected to the circuit traces.

Structured monoliths

a fallacy or a good compromise between
microservices first vs. monolith first approaches?

Timo Verhoeven

Vienna Software Seminar on DevOps and Microservice APIs, 28.08.2019

Context

- GuideCom – software solutions for banking and „human resources“
- Monolithic Java web apps
- Mostly on-premises
- SaaS: on the rise

Qu

„Don't do this first!“
Stefan Tilkov [STil1]

Monolith

- Full-stack in-house framework
- Fast reliable development
- Proven deployment
- Known costs – €

„Don't do this first!“
Martin Fowler [MFow1]

Microservices

- Scalability, reliability, elasticity
- Deployment frequency
- Lead time for changes
- Experimentation
- Unknown costs – €€

„Accelerate:
State of DevOps“
[DORA1]

[STil1] <https://martinfowler.com/articles/dont-start-monolith.html>

[MFow1] <https://martinfowler.com/bliki/MonolithFirst.html>

[DORA1] <https://cloud.google.com/devops/state-of-devops>

Structured monolith

- Monolith first
- ... structured to easily cut out microservices later

- Short TTM
- Scalable later
- Costs?

Obstacles?

Transactions

- Segmentation inside transactional call tree
- Broken transactional behavior

- Refactor
- Distributed Transactions
- Sagas / Process Controllers

Unreliable infrastructure

- Former in-process calls can fail
- Upfront error handling
- Refactor

Type systems

- Enums, discriminated unions, immutability, JSON
- Available and well supported on target platform?

- Use only lowest common denominator
- Introduce translation layer

Reactive programming

- Segmentation between reactive producer and consumer
- Refactor
- Remoting protocol with reactive streams semantics

Remote endpoint security

- Authentication
- Authorization
- Validation
- Denial of service attacks

- Transparent infrastructure-level solutions
- Refactor

Shared services in multitenant environments

- Security implications
- Logging implications
- Debugging implications

- Avoid shared services
- Cater for above needs

Network throughput

- Segmentation between BLOB operations
- Pointer passing vs. Network throughput

- Refactor

Latency

- Segmentation of chatty code
- In-process call (1 ns) vs. remote invocation (500,000 ns)
- CPU cache eviction – mechanical sympathy

- Refactor
- Extra Caching

Is it worth it?

- Additional upfront decisions and efforts needed
- Later: Refactor, refactor, refactor



Structured monoliths

a fallacy or a good compromise between
microservices first vs. monolith first approaches?

Timo Verhoeven

Vienna Software Seminar on DevOps and Microservice APIs, 28.08.2019