Article: Are they good or bad

Microservices: Are they good or bad?

Cons

Main argument: Complexity

- **Management overhead**
  - Every service in production must be monitored, deployed, tested, and documented
  - Ongoing maintenance for each service
  - Monorepos can reduce this overhead, but you trade it for dependency management problems
- **Latency** (especially in heavily microservice-based architectures)
  - Services spread across multiple physical locations / availability zones, so network packets travel farther
  - Load balancers, routers, firewalls, and other network/security equipment add latency between availability zones
  - Possible mitigation: direct service-to-service communication
- **Resilience**
  - More moving parts means more potential points of failure
- **Distributed responsibility**
  - Ownership is spread across services, and a single team can become a bottleneck

Advantages

- **Scalability**
  - Each service becomes independently scalable (scaling the platform piece by piece)
- **Flexibility**
  - Every platform changes over time; adding new functionality is relatively easy
  - Testing can become easier with clear contracts between services
- **Reusability**
  - Services can be shared across the system (DRY)

Other points

- **WebAssembly** changes how we build software, with support for multiple languages
- **Communication** between services is a key concern throughout

Resources

- https://levelup.gitconnected.com/microservices-are-they-good-or-bad-the-truth-is-it-depends-d648811d5ac8