M.EIC, 2022-23

Large Scale Software Development

Filipe Correia, Daniel Pinho, João Pedro Dias

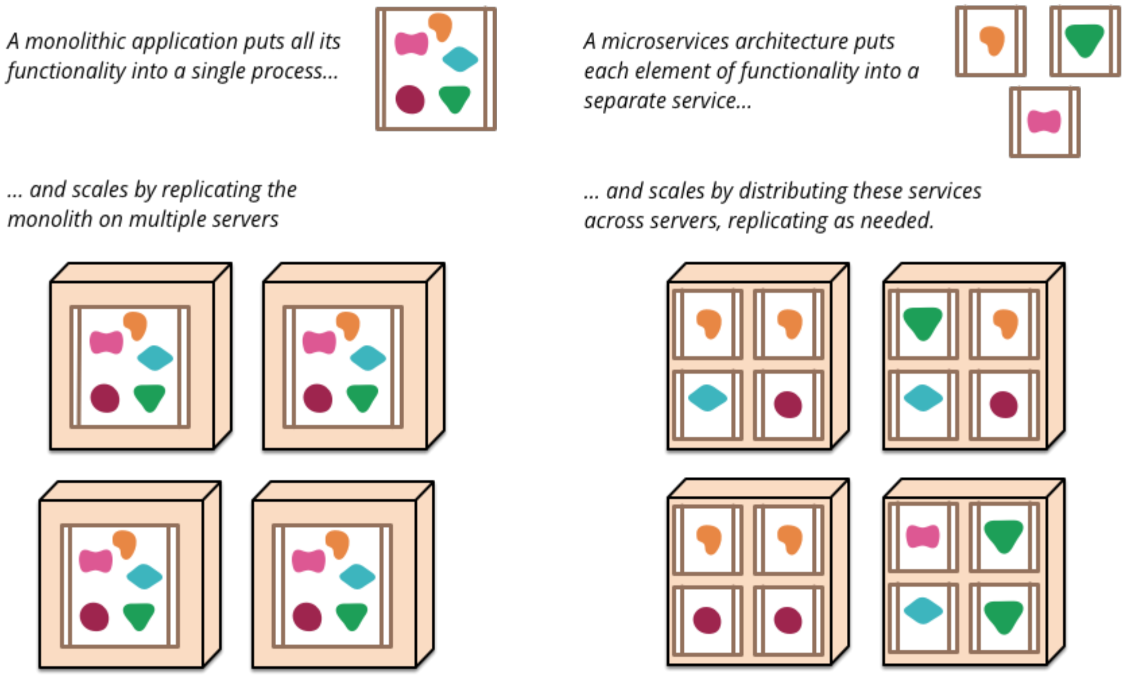
U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DEI
DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA

Microservice Architectures

Microservices



A monolithic application puts all its functionality into a single process…

… and scales by replicating the monolith on multiple servers

A microservices architecture puts each element of functionality into a separate service…

… and scales by distributing these services across servers, replicating as needed.

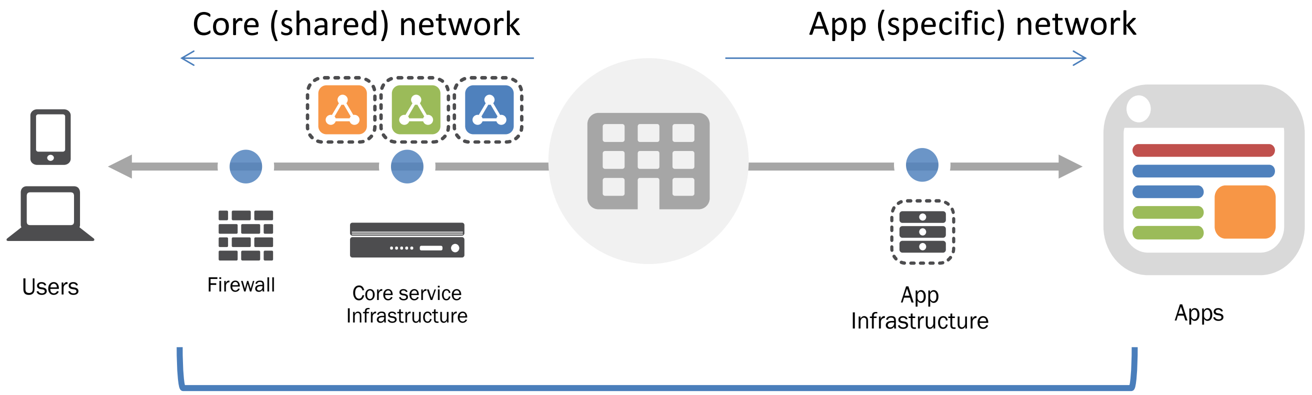
https://martinfowler.com/articles/microservices.html

Why microservices?

- Conway's Law
  "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations"
- Accelerating the pace of change
- Increasing the scale of operation
- Reducing the cost (of change and of operation)

Melvin E. Conway, "How do committees invent." Datamation 14.4 (1968): 28-31.

Monolithic Service Architecture

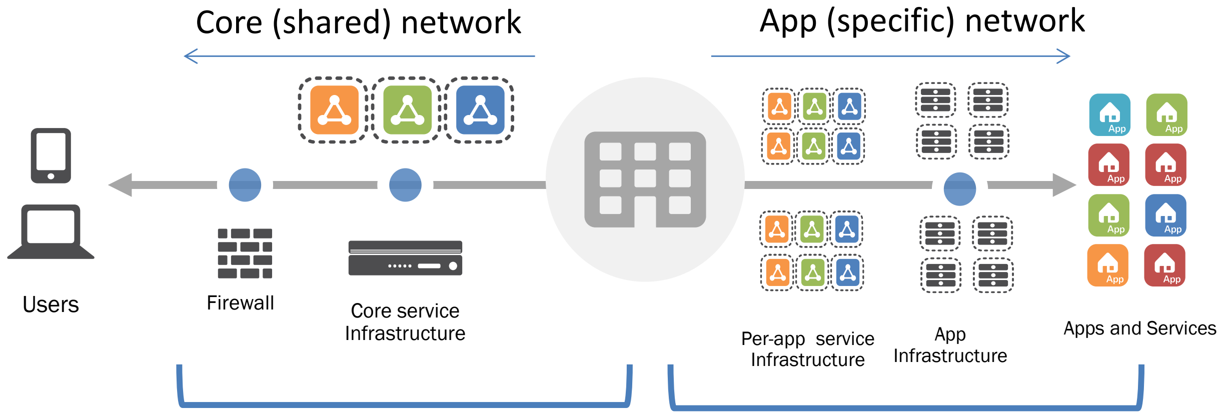


- Low rate of change
- High cost of change
- Low tolerance for disruption

PRIORITY: RELIABILITY

Lori MacVittie, "Pushing the DevOps Envelope - How Microservices Architecture is Expanding DevOps to the Network", 2015

Microservice Architecture



- Low rate of change
- High cost of change
- Low tolerance for disruption

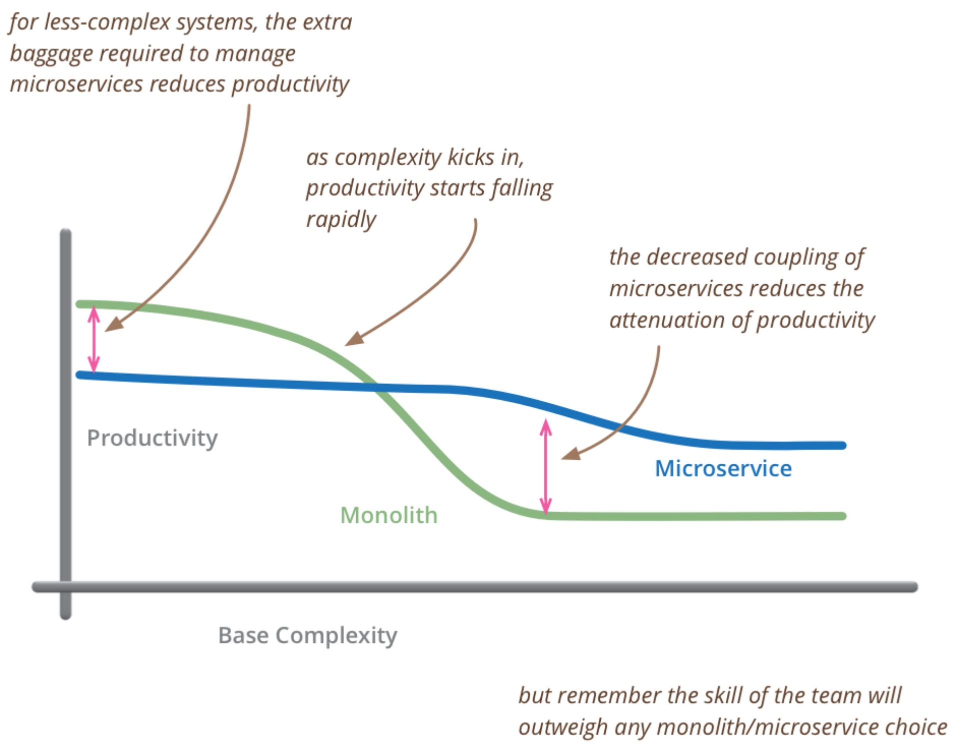
PRIORITY: RELIABILITY

- High rate of change
- Low cost of change
- High tolerance for disruption

PRIORITY: AGILITY

Lori MacVittie, "Pushing the DevOps Envelope - How Microservices Architecture is Expanding DevOps to the Network", 2015

The cost of Microservices



https://martinfowler.com/bliki/MicroservicePremium.html

Microservices Characteristics

- Componentization via Services
- Organized Around Business Capabilities
- Products not Projects
- Smart Endpoints and Dumb Pipes
- Decentralized Governance
- Decentralized Data Management
- Infrastructure Automation
- Design for Failure
- Evolutionary Design

https://martinfowler.com/articles/microservices.html

Componentization via Services

- **Components** are independently replaceable and upgradeable software units.
- Microservices approach componentization by breaking a system down into distinct services.
- Benefits:
    - Services are independently deployable
    - Strong component interfaces (helps keep coupling down)
- Liabilities:
    - Remote calls are more expensive
    - Harder to refactor across service boundaries

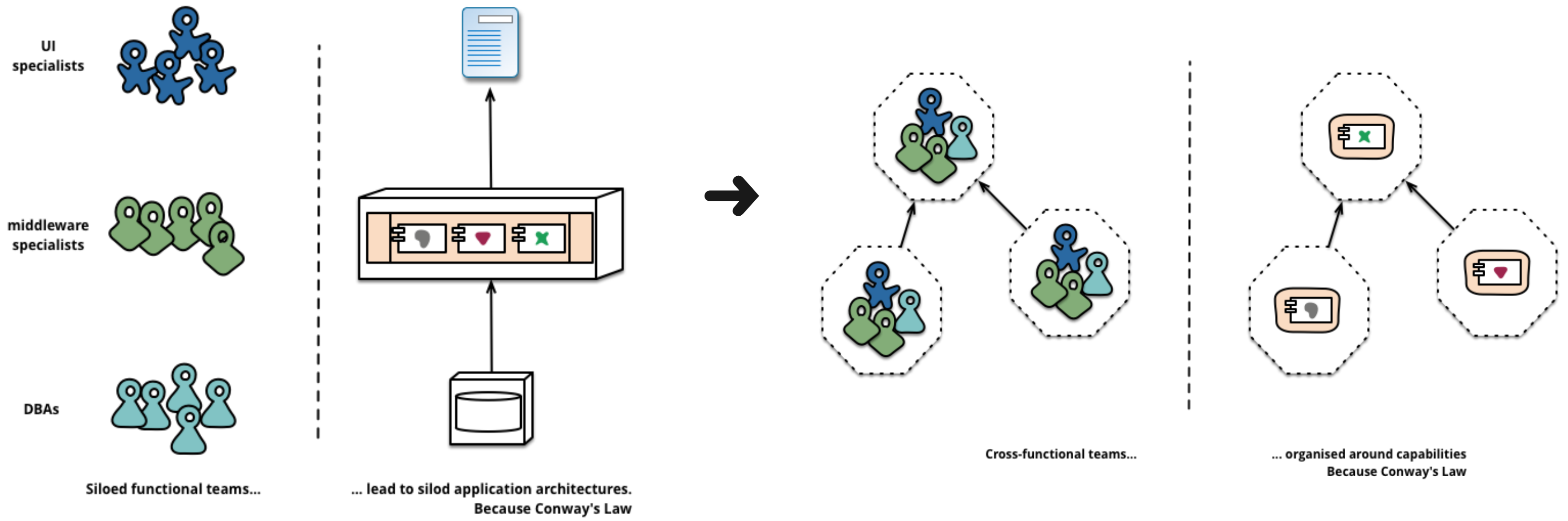
https://martinfowler.com/articles/microservices.html

Organized Around Business Capabilities

- Conway’s Law
- Use cross-functional teams
- “Encapsulate what varies”

https://martinfowler.com/articles/microservices.html

Organized Around Business Capabilities


UI specialists
middleware specialists
DBAs
Siloed functional teams...
... lead to silod application architectures. Because Conway's Law
Cross-functional teams...
... organised around capabilities Because Conway's Law

https://martinfowler.com/articles/microservices.html

Products not Projects

- Putting developers into **day-to-day contact** with how their software **behaves in production**
- **Projects:** Endeavors with a start, middle and end, with the goal of delivering some piece of software
- **Products:** Owned over its lifetime by a team, who takes full responsibility for the software in production

https://martinfowler.com/articles/microservices.html

Smart Endpoints and Dumb Pipes

- Avoid sophisticated communication structures orchestrated by a central tool (e.g., ESBs)
- "*smart endpoints and dumb pipes*"
- Use of lightweight messaging (REST, lightweight message bus)
- Use of coarser-grained communication

https://martinfowler.com/articles/microservices.html

Decentralized Governance

- Use the right tool for the job
- Value **emergent reuse** rather than **imposed standards**
- Evolvable service contracts

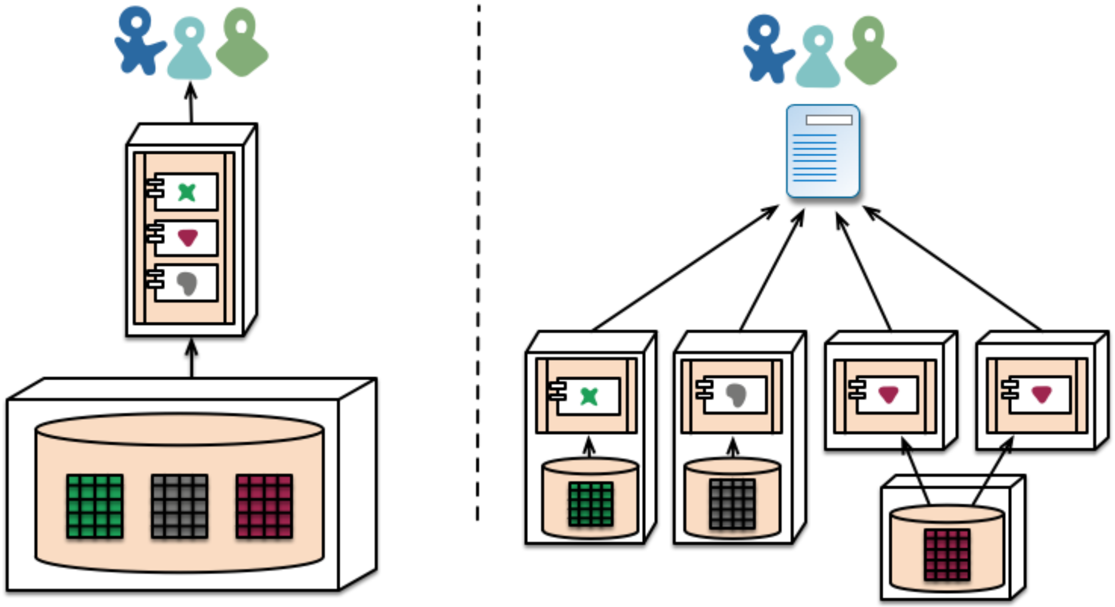
https://martinfowler.com/articles/microservices.html

Decentralized Data Management

- Decentralization of decisions about conceptual models and data storage
- Each service manages its own data
- Eventual consistency – transactionless coordination between services

https://martinfowler.com/articles/microservices.html

Decentralized Data Management


monolith - single database
microservices - application databases

https://martinfowler.com/articles/microservices.html

Infrastructure Automation

- Deployment of **many different services** implies added **operational complexity** of building, deploying and running
- Automate tests!
- Automate deployment!
- Automate infrastructure provisioning!

https://martinfowler.com/articles/microservices.html

Design for Failure

- Design to tolerate the failure of services
- Requires constant attention to how service failures affect the user experience
- Leads to sophisticated **real-time monitoring** setups

https://martinfowler.com/articles/microservices.html

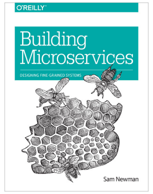
Evolutionary Design

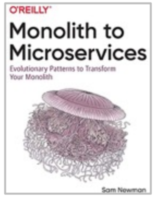
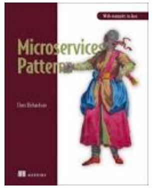
- Value ways to do frequent, fast, and well-controlled changes to software
- Components should be independently replaceable and upgradable
- *“Encapsulate what varies”*

https://martinfowler.com/articles/microservices.html

References

- *Building microservices: designing fine-grained systems*
  Sam Newman, O'Reilly Media, 2015
- *Monolith to Microservices*
  Sam Newman, O'Reilly Media, 2019
- *Microservice patterns*
  Chris Richardson, Manning, 2017
- *How to break a Monolith into Microservices*
  https://martinfowler.com/articles/break-monolith-into-microservices.html
- *Microservices*
  https://martinfowler.com/articles/microservices.html




O'REILLY
Monolith to Microservices
Evolutionary Patterns to Transform Your Monolith
Sam Newman


FEUP.DEI

DEI DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

U.PORTO

DEI - DEPARTAMENTO DE ENGENHARIA INFORMÁTICA
dei.fe.up.pt | secdei@fe.up.pt | +351 225 082 134
Director **João MP Cardoso** | Full Professor
jmpc@fe.up.pt | +351 220 413 284

FEUP - FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Rua Dr. Roberto Frias | 4200-465 Porto | PORTUGAL
www.fe.up.pt | feup@fe.up.pt | +351 225 081 400
@DEI_FEUP