Architecting for Innovation

Stefan Tilkov, @stilkov
innoQ Deutschland GmbH

innoQ

Problems Some People Have

Building features takes too long

Technical debt is well-known and not addressed

Deployment is way too complicated and slow

Architectural quality has degraded

Scalability has reached its limit

“-ility” problems abound

Replacement would be way too expensive

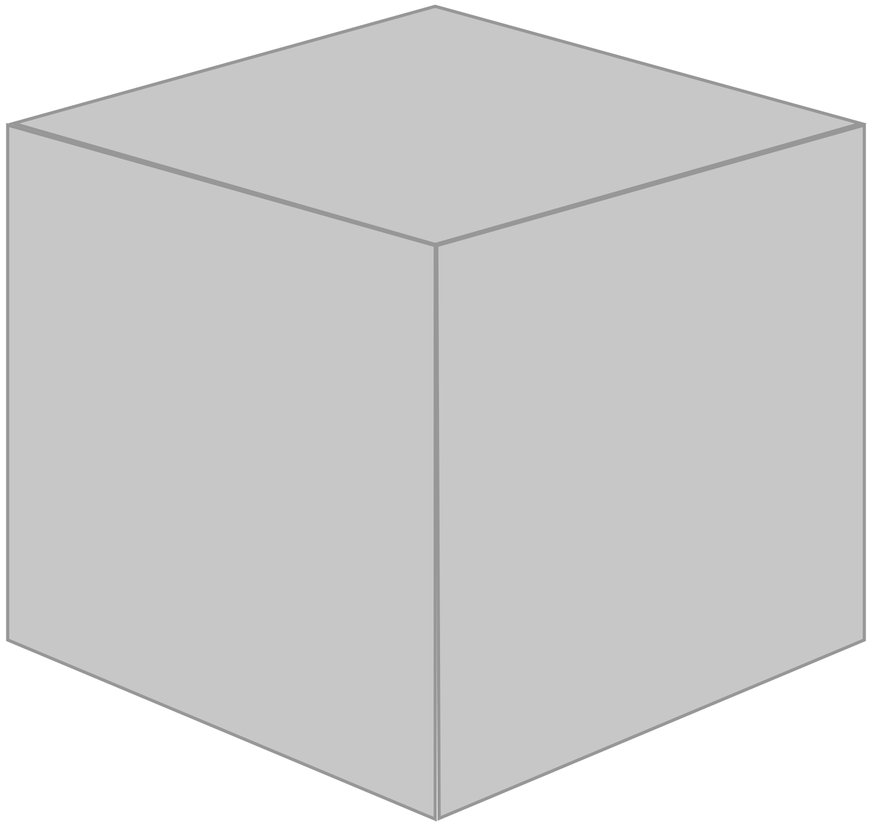
If you barely manage to keep systems running …

… how can you drive innovation?

Microservices

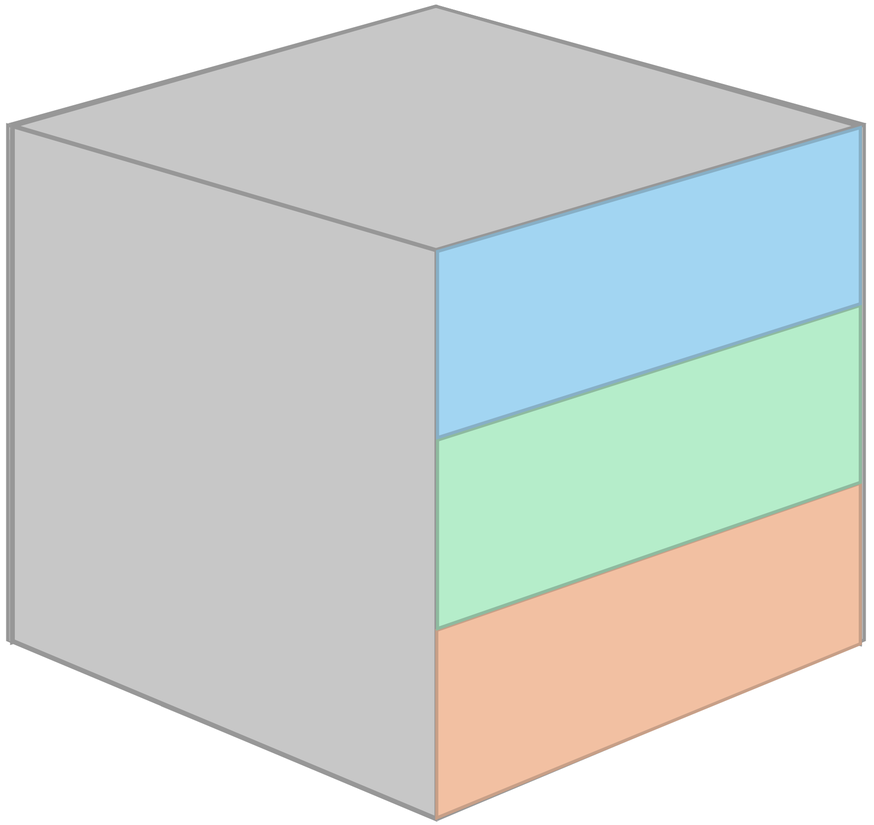
(a.k.a. “the solution to every problem”)

“Monolithic” Applications

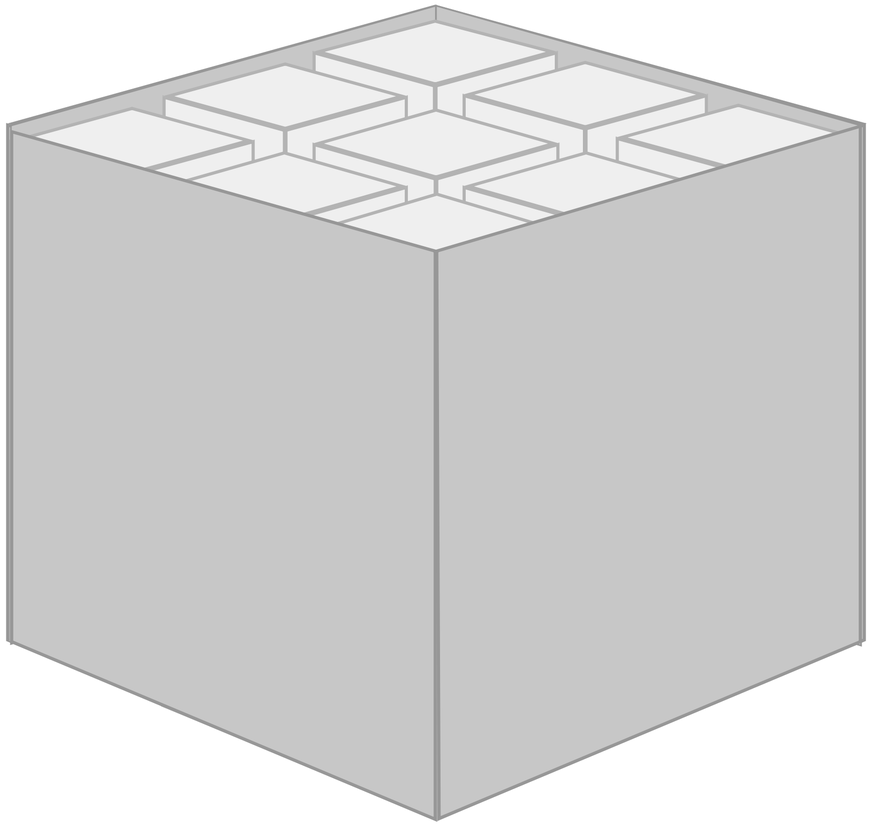


See https://www.innoq.com/en/links/self-contained-systems-infodeck/

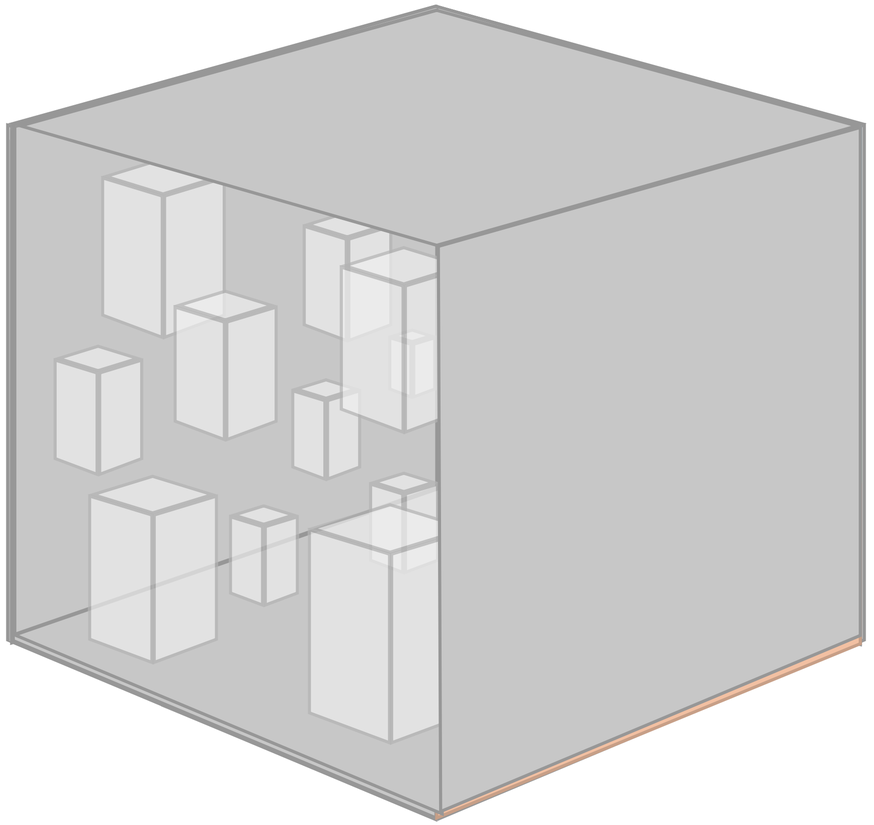
Layers



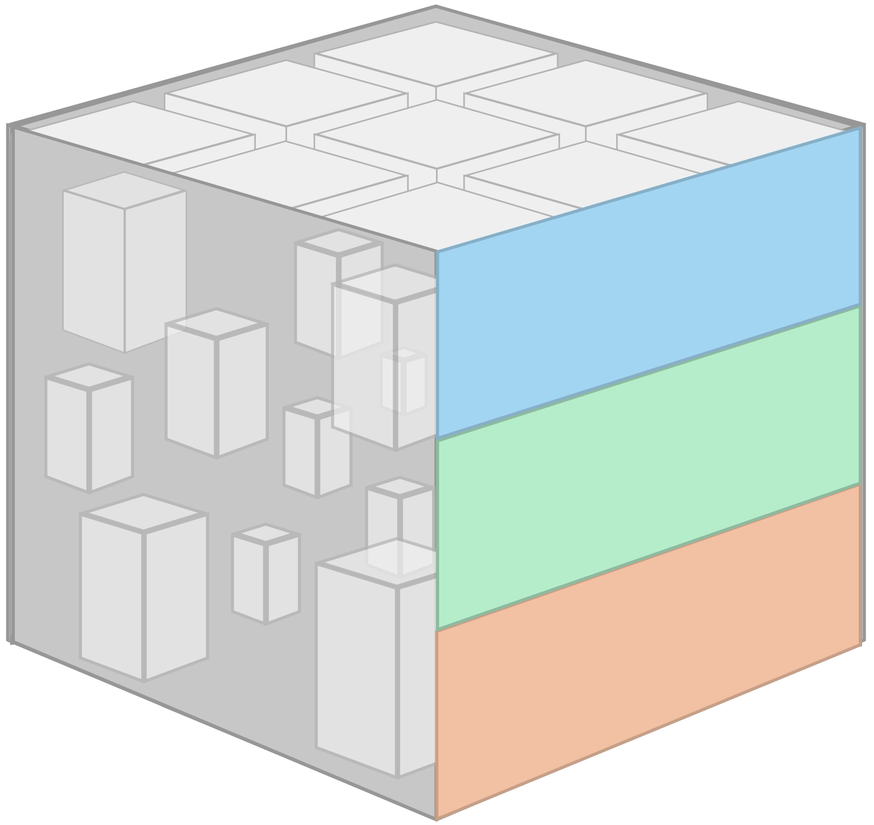
Contexts



Internal components



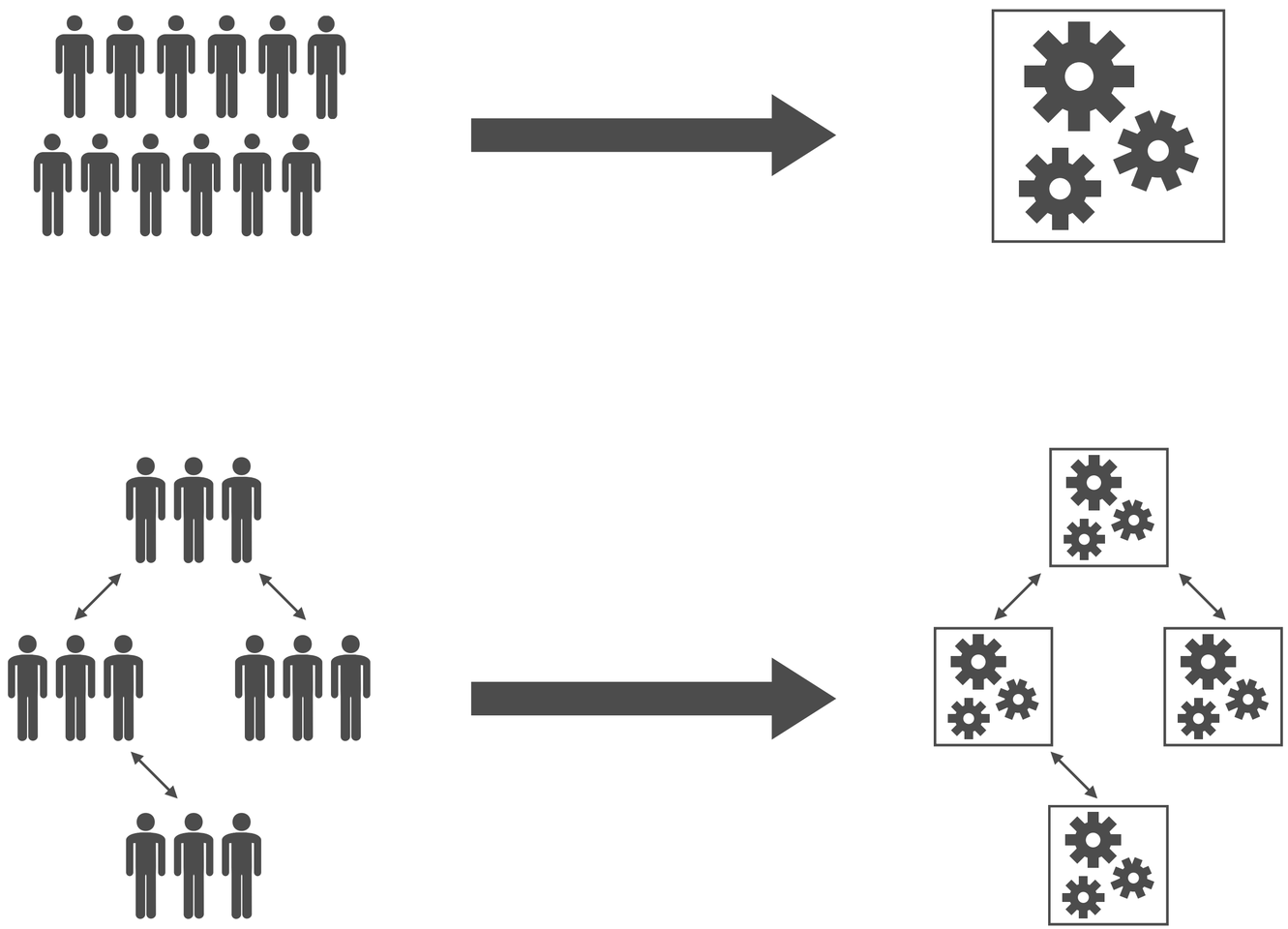
Multi-dimensional complexity



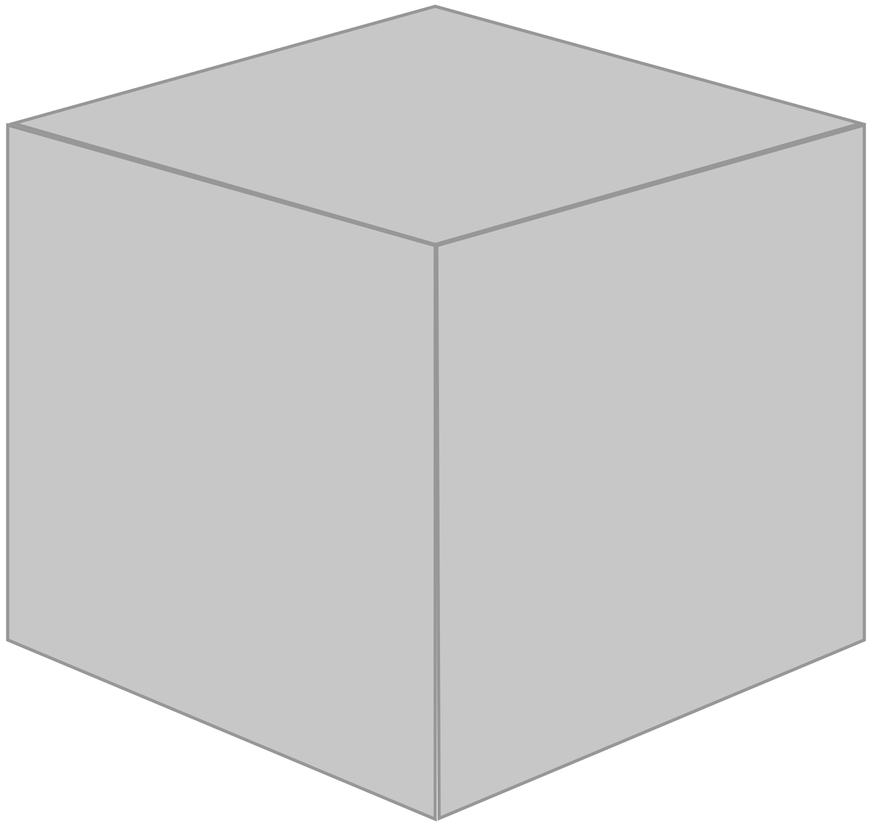
„Organizations which design systems [...] are constrained to produce designs which are copies of the communication structures of these organizations.“

— Melvin Conway, 1968

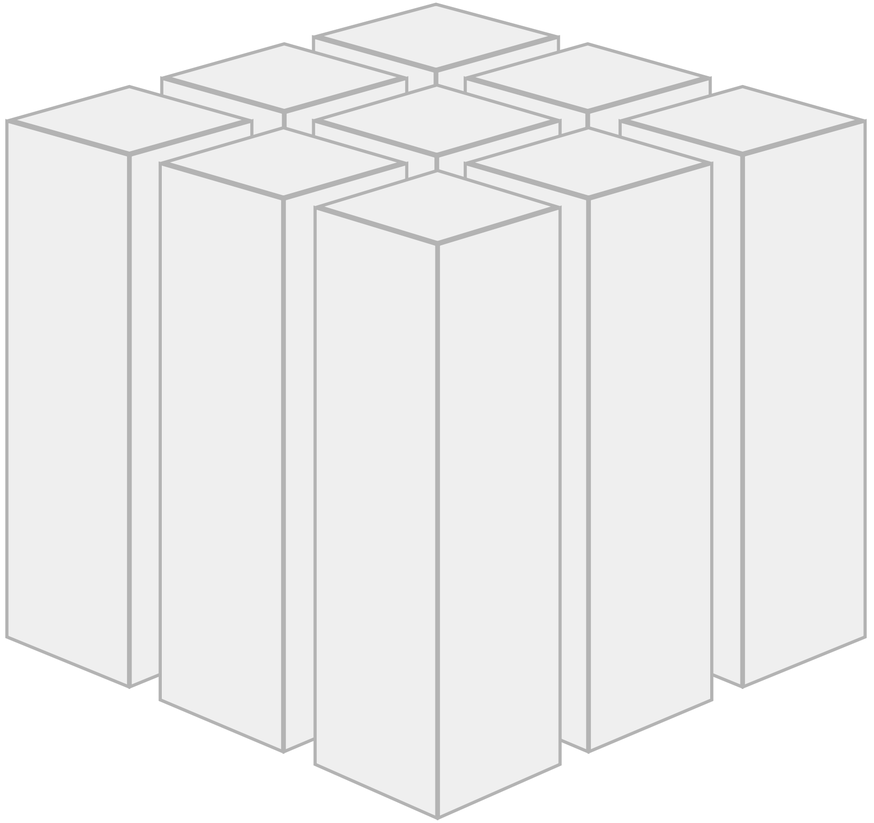
Conway’s Law illustrated



So let’s start with this …



… and cut it apart: *Microservices*

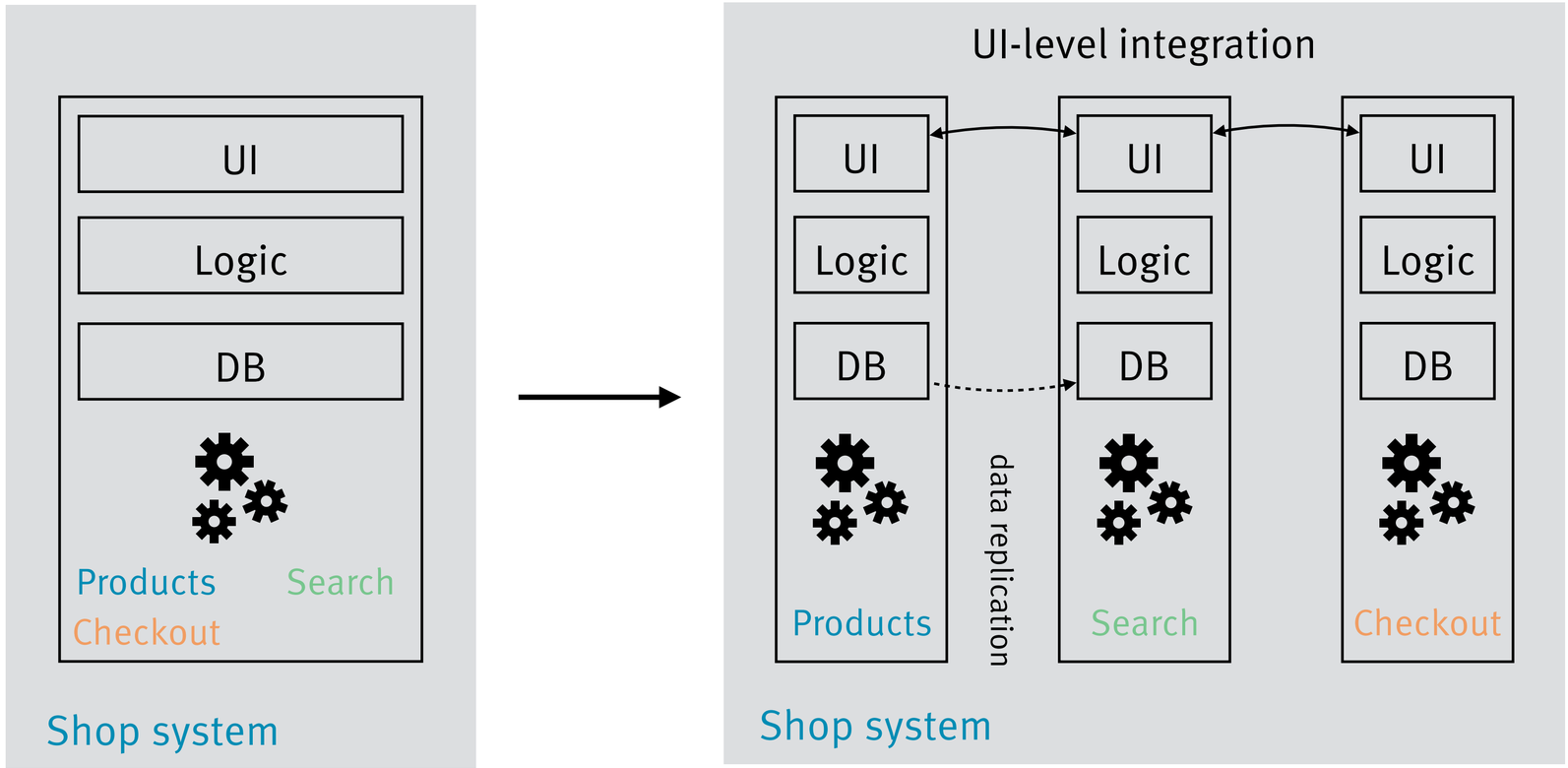


“Flavors” of Microservices

- “Microservice” is a very ill-defined term and means different things to different people
- Three mainstream interpretations of the term:
  - “Proper” microservices (where “micro” means “really, really small”)
  - Collaborating “service graphs” (Netflix-style), medium-sized
  - Self-contained systems (SCS), complete autonomous systems (see http://scs-architecture.org)

Self-contained Systems(*) (SCS)

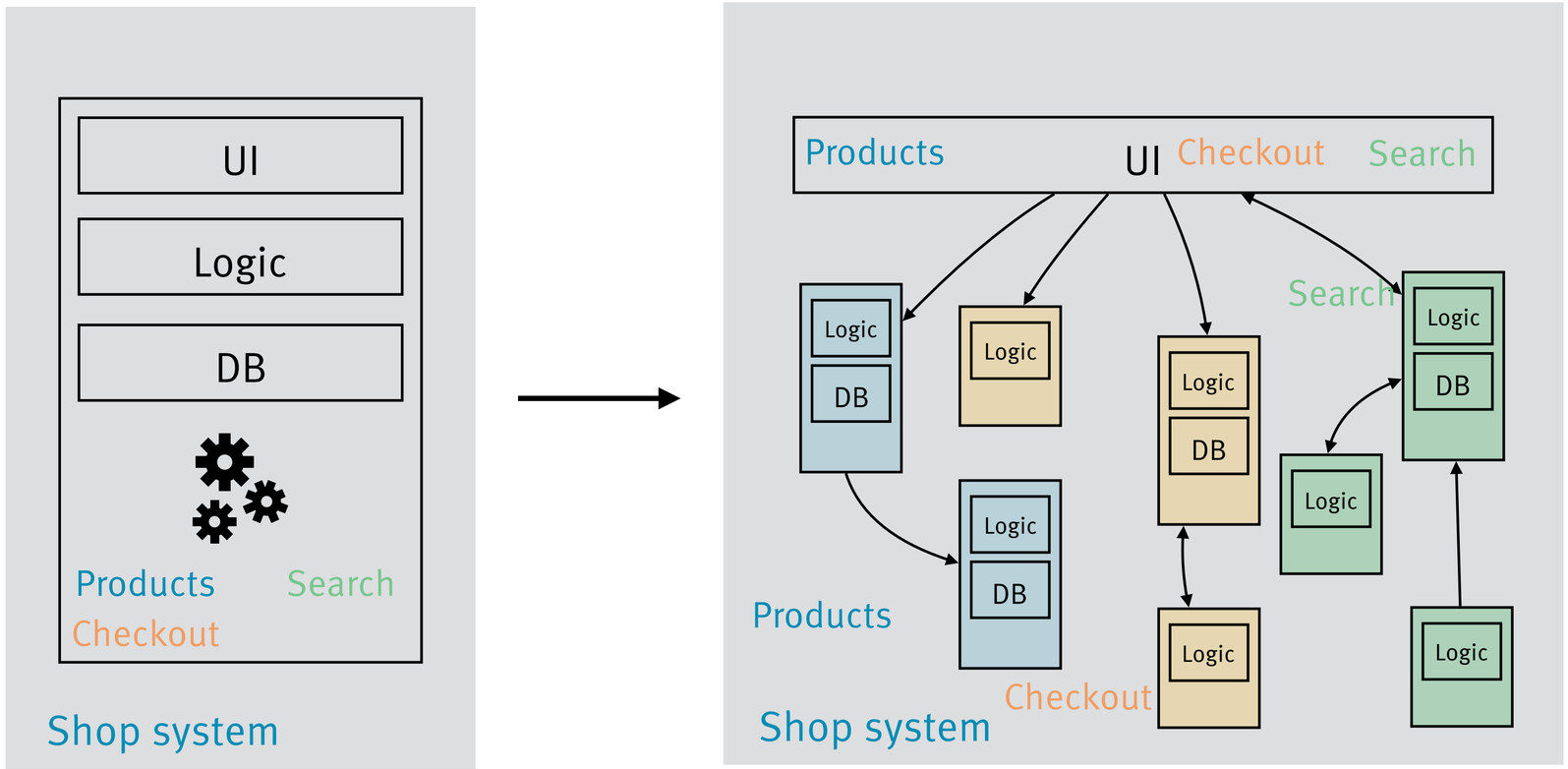
Business domain broken down into vertical slices



(*) see: https://scs-architecture.org

Finer-grained Microservices

Business domain broken down into UI and services



Some equations

1 Service = 1 Deployment Unit

(“can be deployed independently”)

1 Service = 1 Team

(“one team is responsible for building it”)

1 Service = 1 Bounded Context

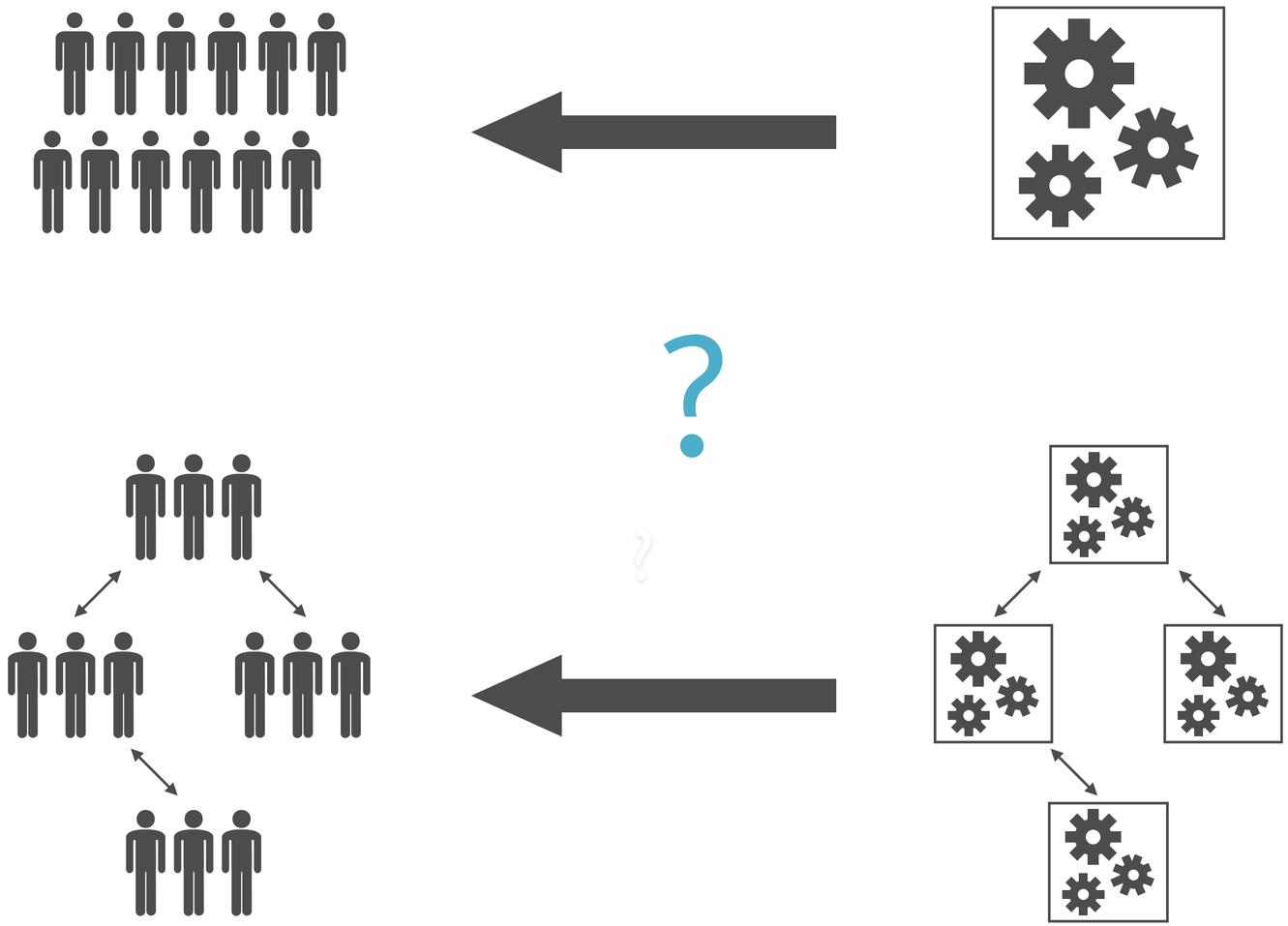
(“does only one thing, but does it well”)

1 project ≠ 1 system

(“modularize from the start”)

Conway’s Law revisited –

Can architecture have an impact on organisation?

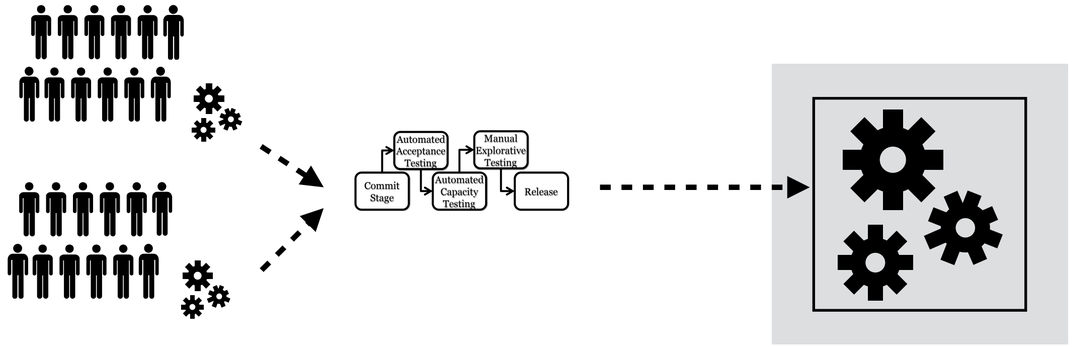


Continuous Delivery

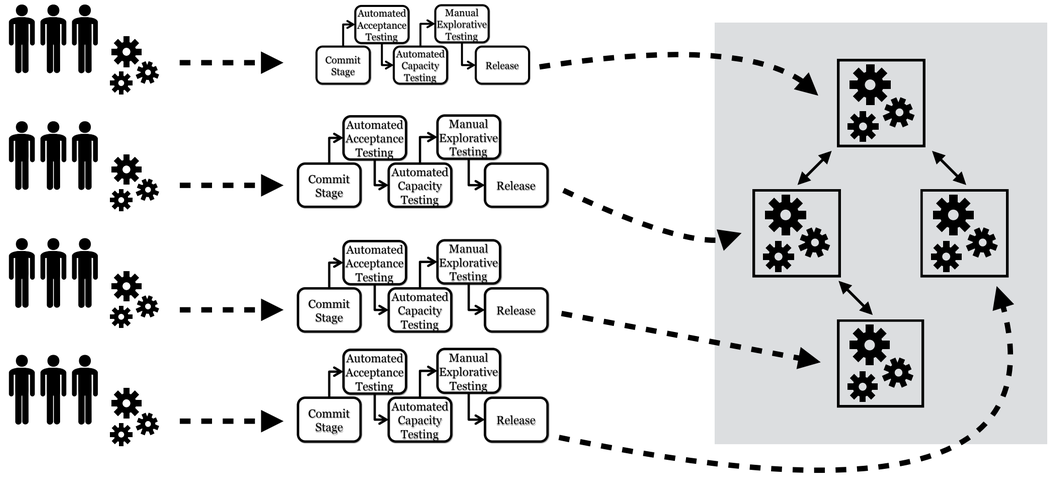
- “If it hurts, do it more often”
- Enable pushing changes into production as quickly as possible
- Continuous Deployment vs. Continuous Delivery
- Requires high degree of automation

Delivery pipeline

Delivering monoliths vs. microservices



- whole system is released and deployed in one go
- requires significant coordination within the team(s)
- large, infrequent releases



- separate deployment pipelines per team
- services can be released and deployed independently
- smaller and more frequent releases

Prepare to change your DC strategy …

- Move from hardware towards software
- Automation and self-service
- New economics in networking, storage, memory, CPU
- Not for the faint of heart
- Essentially, become a Cloud provider

… or use services of someone who has done so.

- Dramatic change in acceptance of Public Cloud offerings
- Regional support
  (e.g. AWS in Frankfurt a.M., Germany)
- Improvements in legal aspects
  (e.g. Microsoft/T-Systems trustee arrangement)
- No silver bullet

Benefits for Innovation

1. Isolation
2. Autonomy
3. Indidual Scalability
4. Resilience
5. Speed
6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

1. Isolation
2. Autonomy
3. Scalability
4. Resilience
5. Speed

- Local effects of changes
- Reduced risk of side-effects
- High cohesion

Benefits for Innovation

1. Isolation
2. Autonomy
3. Scalability
4. Resilience
5. Speed

- Independent decisions
- Separate schedules
- Meeting avoidance

Benefits for Innovation

1. Isolation
2. Autonomy
3. Scalability
4. Resilience
5. Speed

- Scale up or down locally
- Cost-efficiency

Benefits for Innovation

1. Isolation
2. Autonomy
3. Scalability
4. Resilience
5. Speed

- Less centralization
- Reduced effects of failures
- Higher overall availability

Benefits for Innovation

1. Isolation
2. Autonomy
3. Scalability
4. Resilience
5. Speed

- Reduced delivery latency
- Higher throughput
- Faster changes

Benefits for Innovation

- Business strategies
- Technical implementation

6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

- DevOps
- “BizDev”?
- “BizDevOps”?
- Metrics for measuring results

6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

- Diversity as asset
- No *forced* single choices
- Build, buy, outsource

6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

- Reduced need for 10-year decisions
- Replace instead of reuse
- Focus on immediate needs

6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

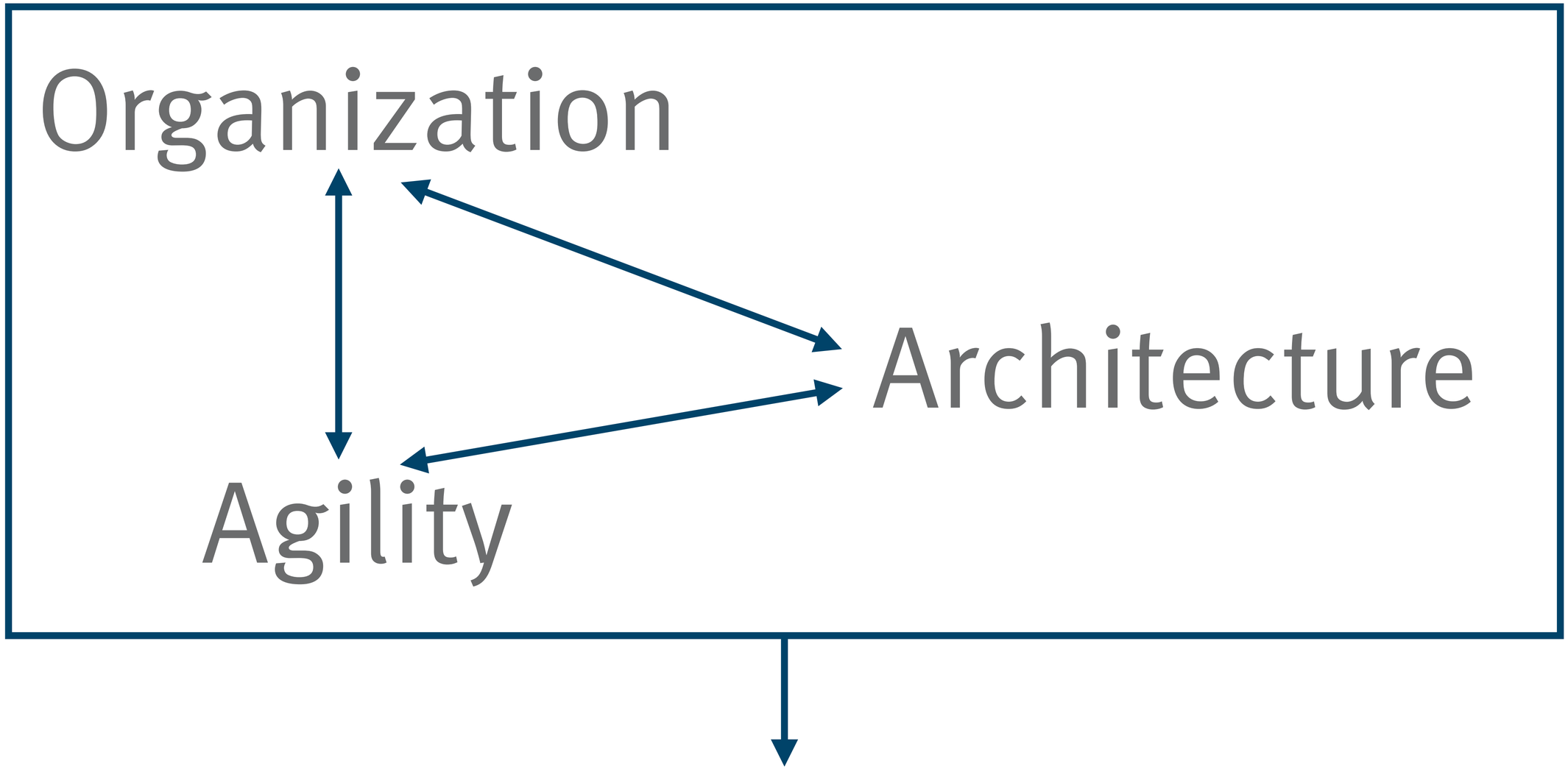
- Supports for partners
- Internal and external collaboration

6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Benefits for Innovation

1. Isolation
2. Autonomy
3. Indidual Scalability
4. Resilience
5. Speed
6. Experimentation
7. Rapid Feedback
8. Flexibility
9. Replacability
10. Ecosystem

Summary


Organization
Architecture
Agility
Innovation

Thank you.

Questions?

Comments?

@stilkov

Stefan Tilkov

stefan.tilkov@innoq.com

Phone: +49 170 471 2625

innoQ

www.innoq.com

innoQ Deutschland GmbH

Krischerstr. 100
40789 Monheim am Rhein
Germany
Phone: +49 2173 3366-0

Ohlauer Straße 43
10999 Berlin
Germany
Phone: +49 2173 3366-0

Ludwigstr. 180E
63067 Offenbach
Germany
Phone: +49 2173 3366-0

Kreuzstraße 16
80331 München
Germany
Phone: +49 2173 3366-0

innoQ Schweiz GmbH

Gewerbestr. 11
CH-6330 Cham
Switzerland
Phone: +41 41 743 0116