Kontrollverlust
in
Softwaresystemen?

Holger Tiemeyer

Java Forum Stuttgart, 07.07.2022

Photo by Lucrezia Carnelos on Unsplash

Health

Banking

> 20 Jahre Erfahrung

Design & Entwicklung

Architektur

Automotive

ITECH PROGRESS

Ludwigshafen:
ITech Progress GmbH
Donnersbergweg 4
67059 Ludwigshafen

Nürnberg:
ITech Progress GmbH
Südwestpark 37-41 / 1.OG
90449 Nürnberg

info@itech-progress.com
www.itech-progress.com
+49 621 595 702 0

Alles aus einer Hand:

ITECH ACADEMY

wissen

Training & Academy

- iSAQB CPSA Foundation Level
- iSAQB CPSA Advanced Level
- Extended Trainings
  - SW Modellierung
  - SW Entwicklung (Java & OOP, JEE, Programmieren für Fortgeschrittene mit JAVA)

→ Als Inhouse- und offene Trainings

ITECH PROGRESS

können

Beratung & Coaching

- Softwarearchitektur
  - Enterprise Architecture Management
  - Domain Driven Design
  - Assessment & Dokumentation

anwenden

Entwicklung

- … mit agilen Methoden
- … mit Bereitstellung von Experten
  (SW-Architekten, Scrum Master, Product Owner / Fachanalysten, Entwickler, Tester, UI-Experten)
- … mit Übernahme der Lösungsverantwortung

Photo by Tim Mossholder on Unsplash

Verfall, Rückbildung, Degeneration

Symptome:

- Zunehmende Zerbrechlichkeit
- Starrheit
- Zu viele Abhängigkeiten (schlechte Wiederverwendungs-möglichkeit)

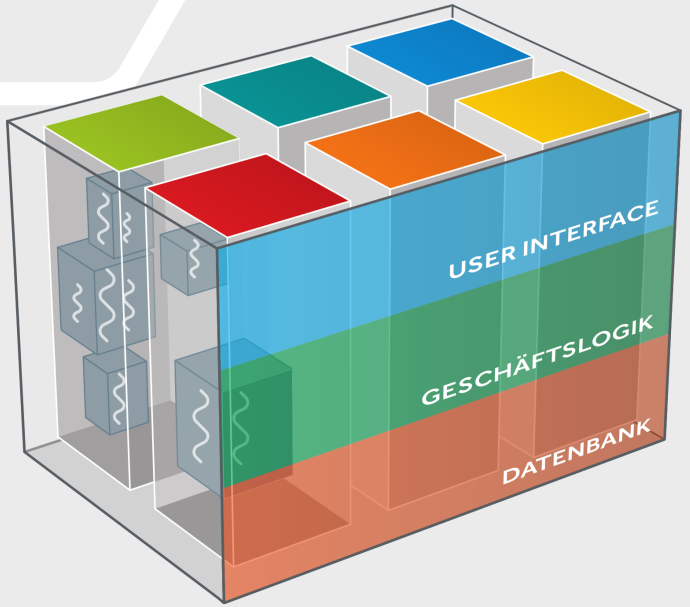
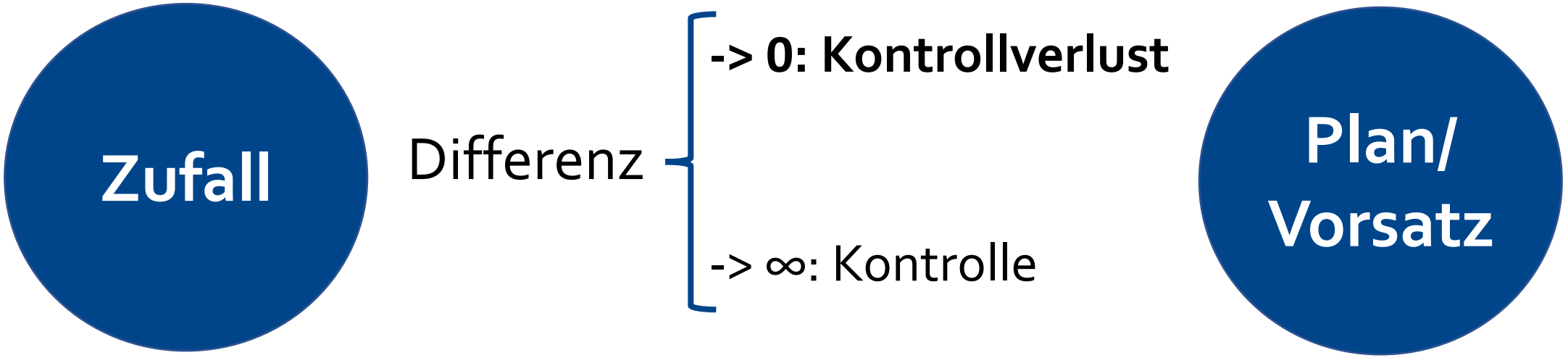


Photo by Markus Spiske on Pexels

Photo by Flora Westbrook on Pexels

Definition: Kontrollverlust, Herkner 2001

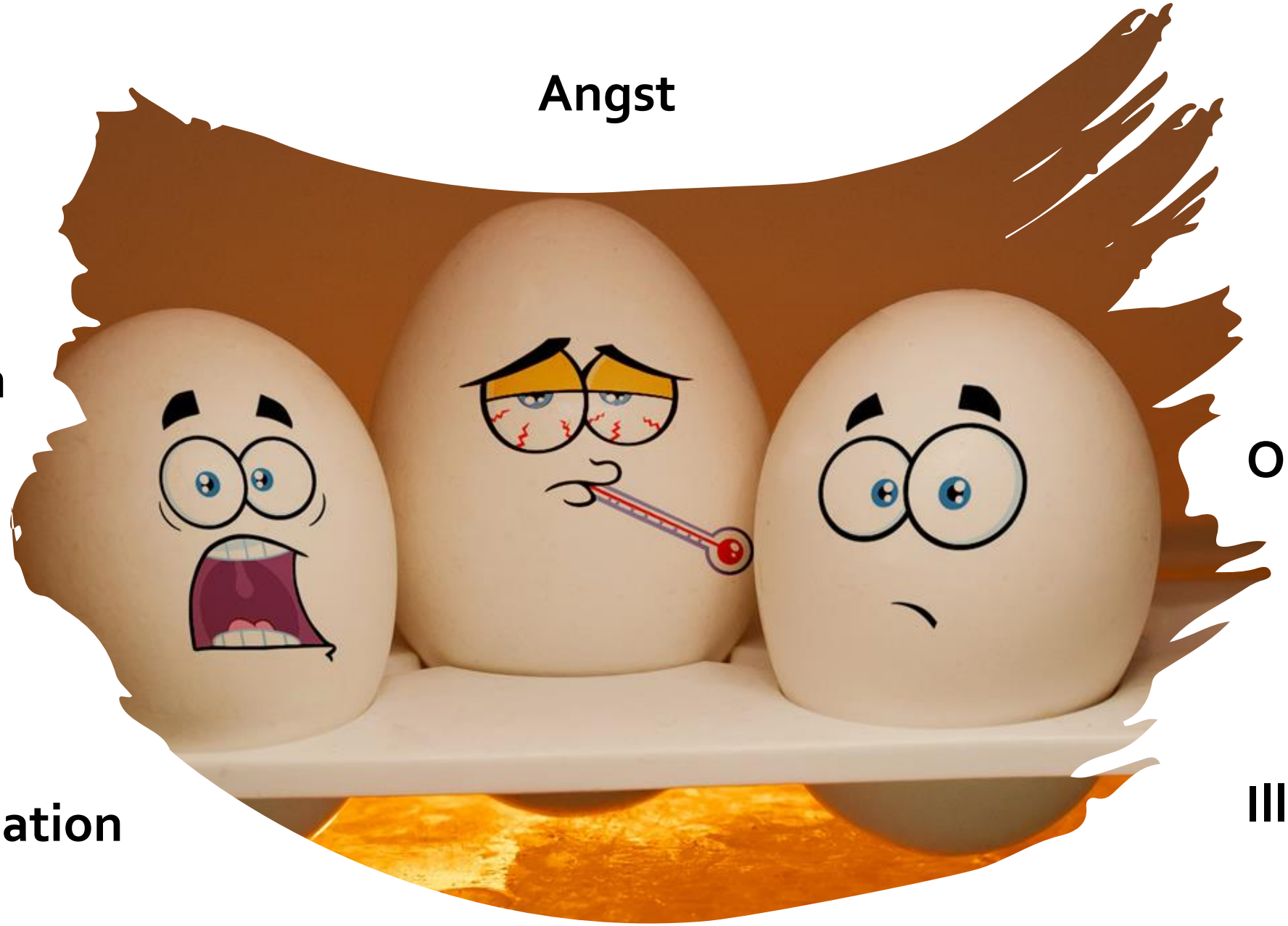


Wahrscheinlichkeit, dass ein erwünschtes Ergebnis ohne eigenes Zutun eintritt.

Wahrscheinlichkeit, dieses Ereignis durch eigenes Handeln herbeiführen zu können.

Angst

Schrecken



Ohnmacht

Resignation

Illusion

Photo by Karan Mandre on Unsplash

Kontrollverlust führt zu schlechten Entscheidungen.

Photo by Tim Johnson on Unsplash

- **Kontrollverlust führt zu schlechten Entscheidungen.**

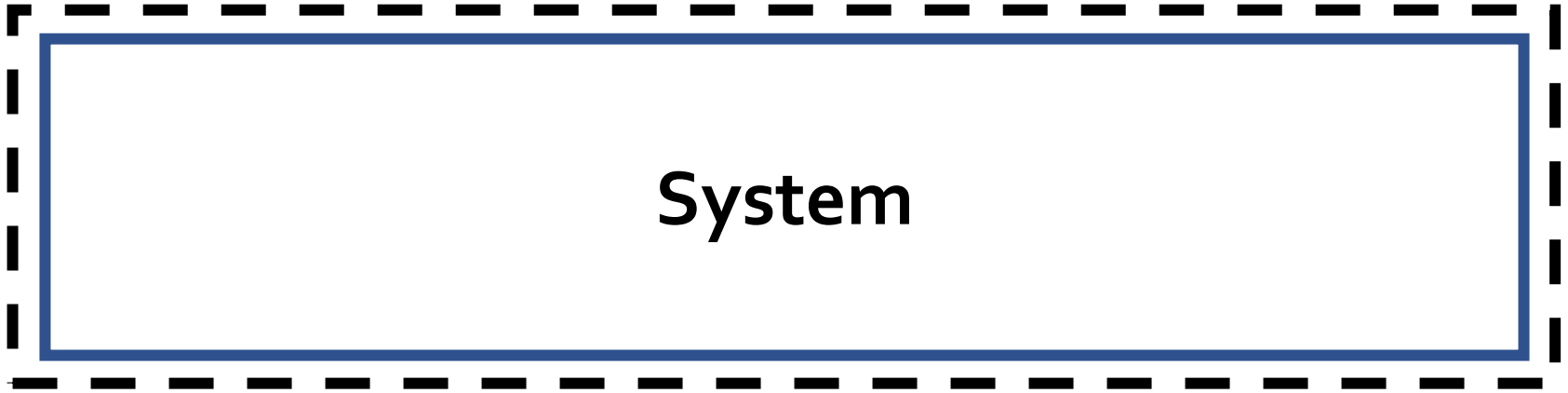
Photo by Clem Onojeghuo on Unsplash

Photo by David Becker on Unsplash

„Unsere Software verrottet!“

Robert C. Martin

Projekt-Scope



1 Projekt == 1 System?

Softwareentropie I

- **Entropie:** Maß für die Informationsdichte, den Informationsgehalt eines (Zeichen-) Systems.

- **Zweites Gesetz der Thermodynamik:**
  In einem **geschlossenen adiabaten System** kann die Entropie nicht geringer werden.
  Sie kann nur **gleich bleiben** oder **erhöht werden**.

Softwareentropie II

Theorie nach Lehman (1985):

1. Ein Computerprogramm, das benutzt wird, wird verändert.

2. Wenn ein Programm verändert wird, wird seine **Komplexität** erhöht – vorausgesetzt, dass niemand aktiv dagegen wirkt.

Komplexität

Photo by Brett Jordan on Unsplash

Photo by Ariel on Unsplash

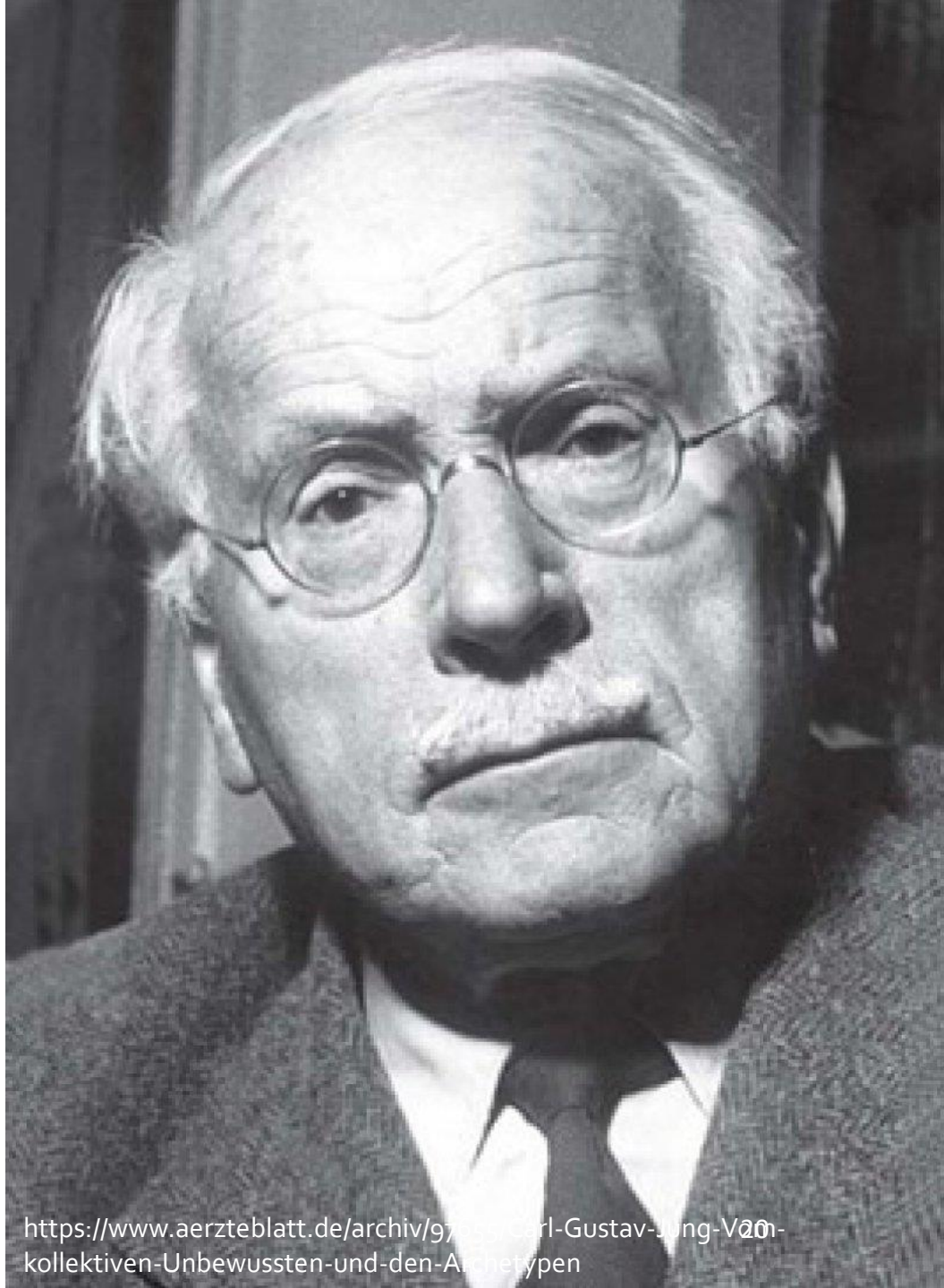
Photo by Manik Roy on Unsplash

“Komplexität ist unser postmodernes Schicksal.”

N. Bolz

C. G. Jung

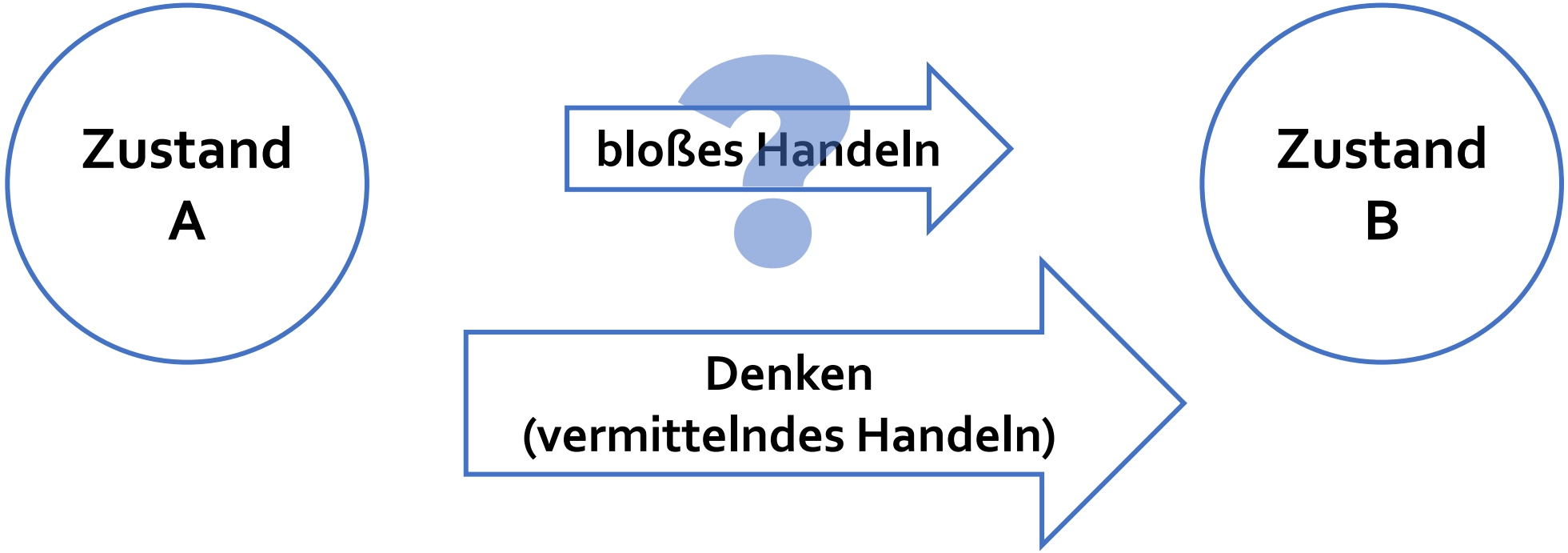
"Bis wir uns das **Unbewusste bewusst machen**,
wird es unser Leben lenken
und wir werden es **Schicksal** nennen."



https://www.aerzteblatt.de/archiv/97[illegible]arl-Gustav-Jung-Vom-kollektiven-Unbewussten-und-den-Archetypen

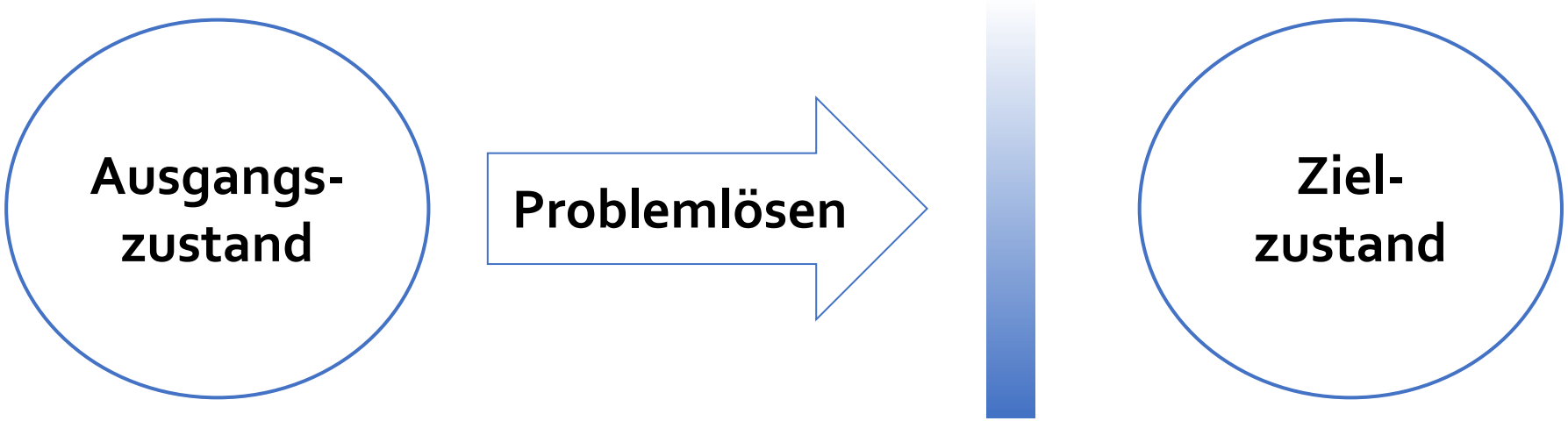
„Ein **Problem** entsteht z.B. dann, wenn ein Lebewesen ein Ziel hat und nicht *weiß*, wie es dieses Ziel erreichen soll."

(Duncker, 1935)



„Unter **Problemlösen** versteht man das Bestreben, einen gegebenen Zustand in einen anderen, gewünschten Zustand zu überführen, wobei es gilt, eine **Barriere zu überwinden**, die sich zwischen Ausgangs- und Zielzustand befindet."

(Duncker, 1935)



Erwerb der Operatoren

Entdecken

Instruktion

Beobachtungslernen

Extraktion der Operatoren:

Analogiebildung

- **Kontrollverlust führt zu schlechten Entscheidungen.**
- **Analogiebildung hilft uns bei der Lösungsfindung.**

Photo by Clem Onojeghuo on Unsplash

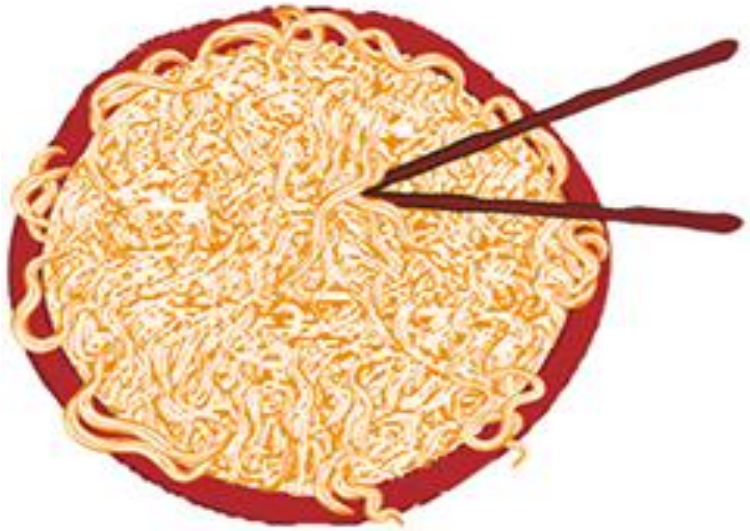
iSAQB® Certified Professional for Software Architecture (Advanced Level)

Entscheidungen treffen!

Photo by Elizabeth Gottwald on Unsplash

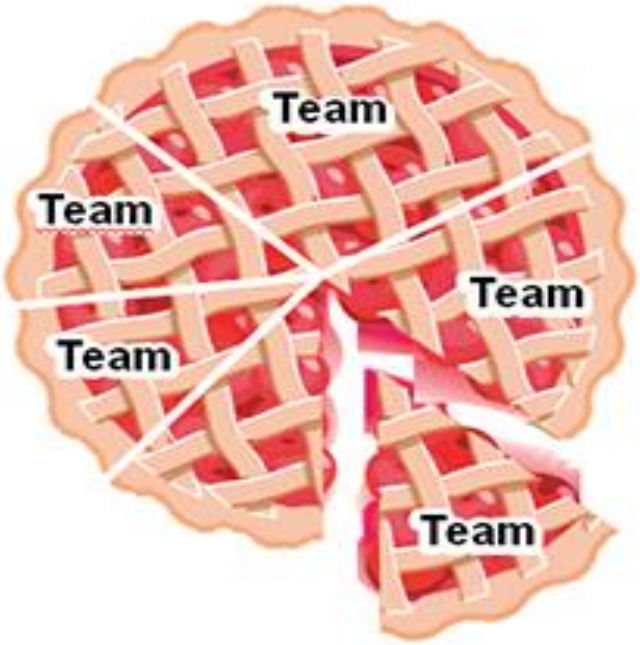
Welcher Architekturstil?

Monolith
- Enge Kopplung



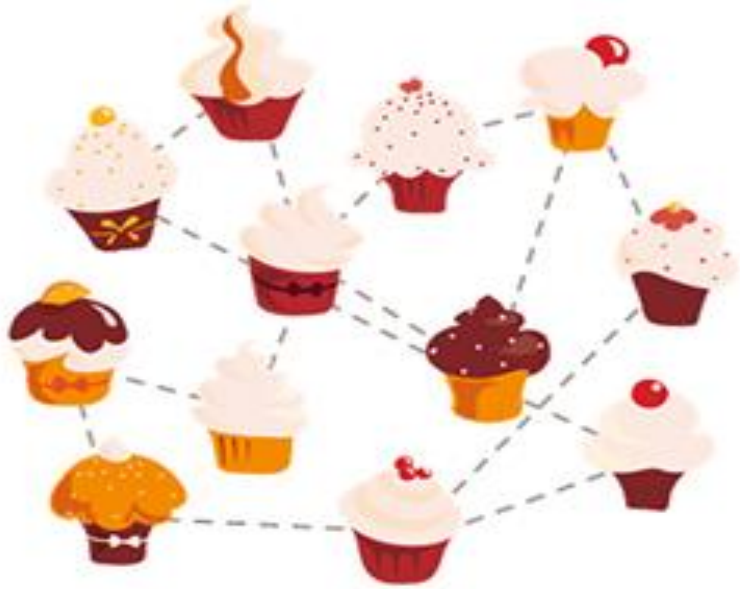
bis in die 1990er

SOA
- Lose Kopplung



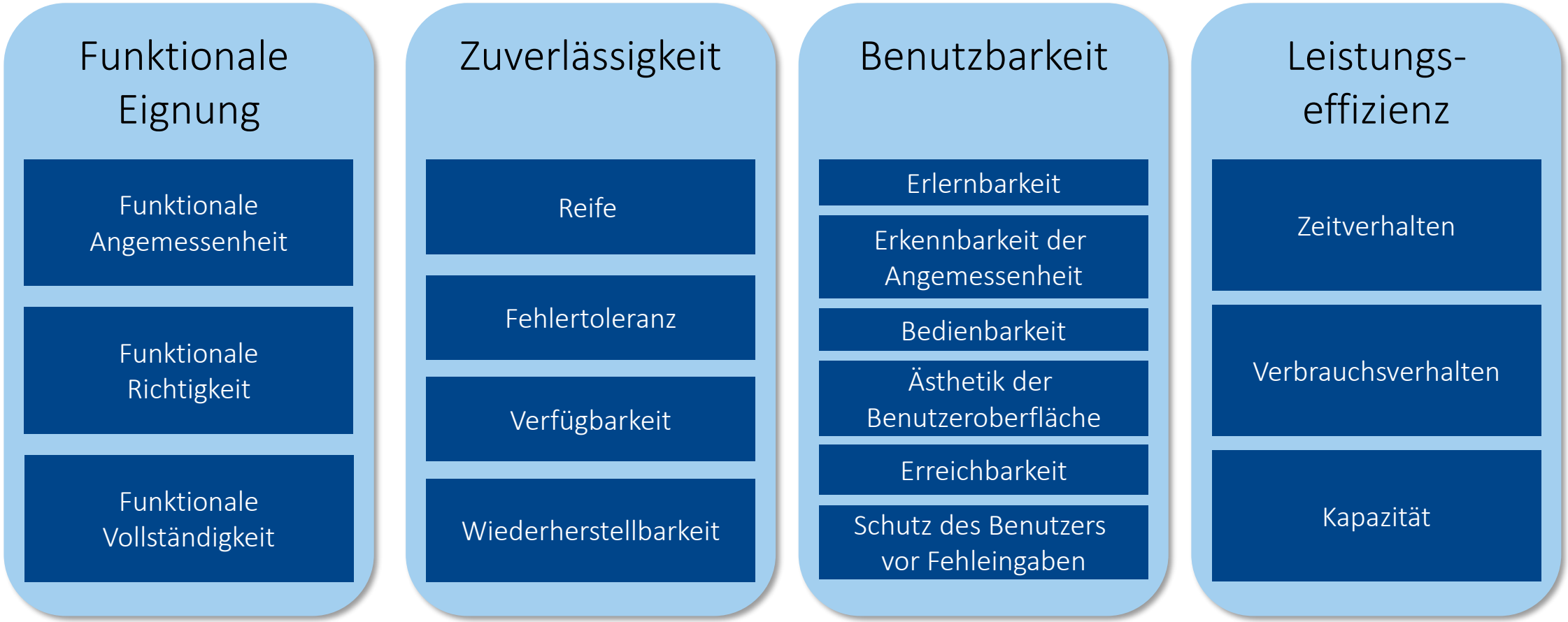
2000er

Microservices
- Entkopplung

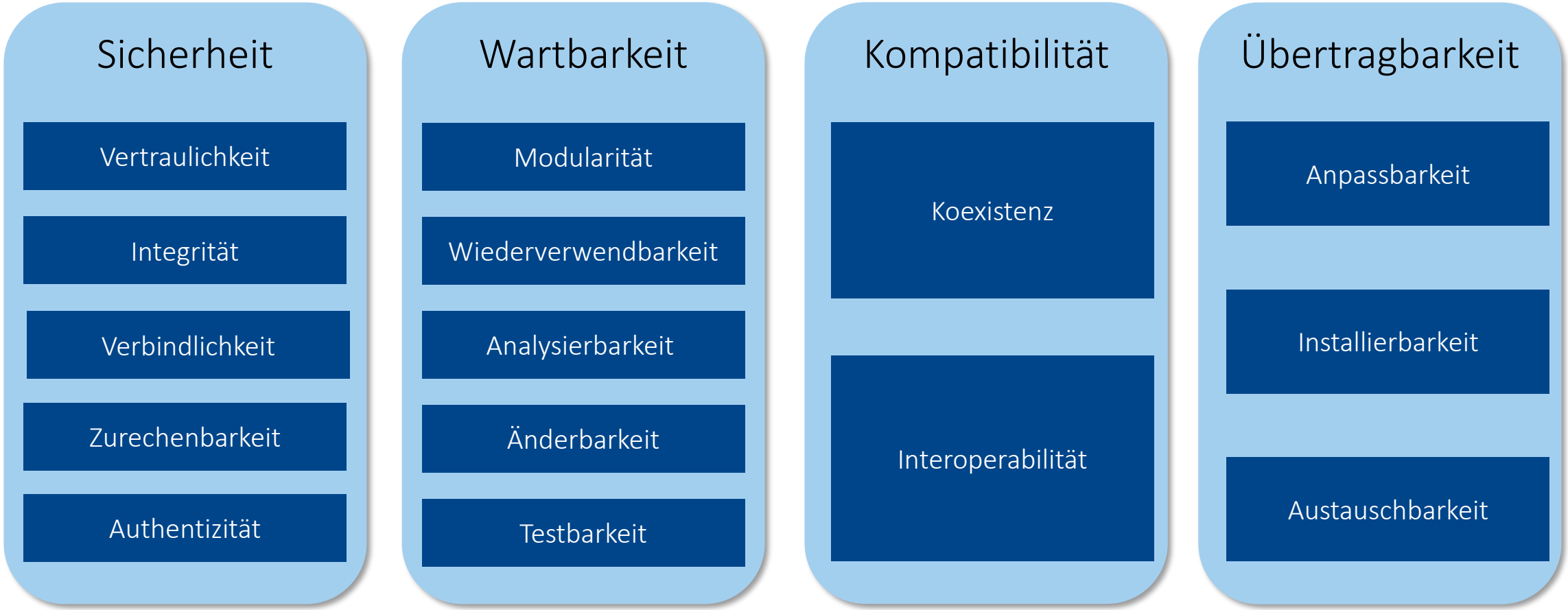


2010er

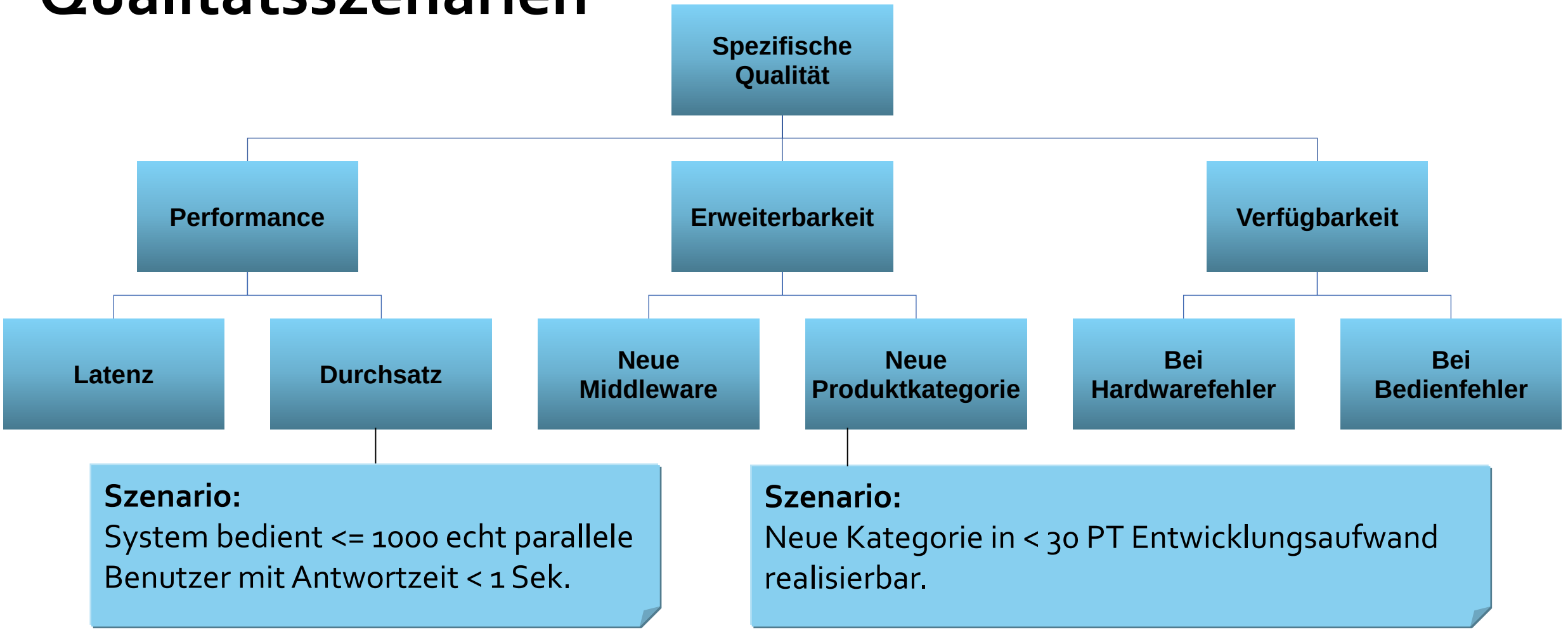
ISO 25010 - Qualitätsmerkmale



ISO 25010 - Qualitätsmerkmale



Qualitätsszenarien



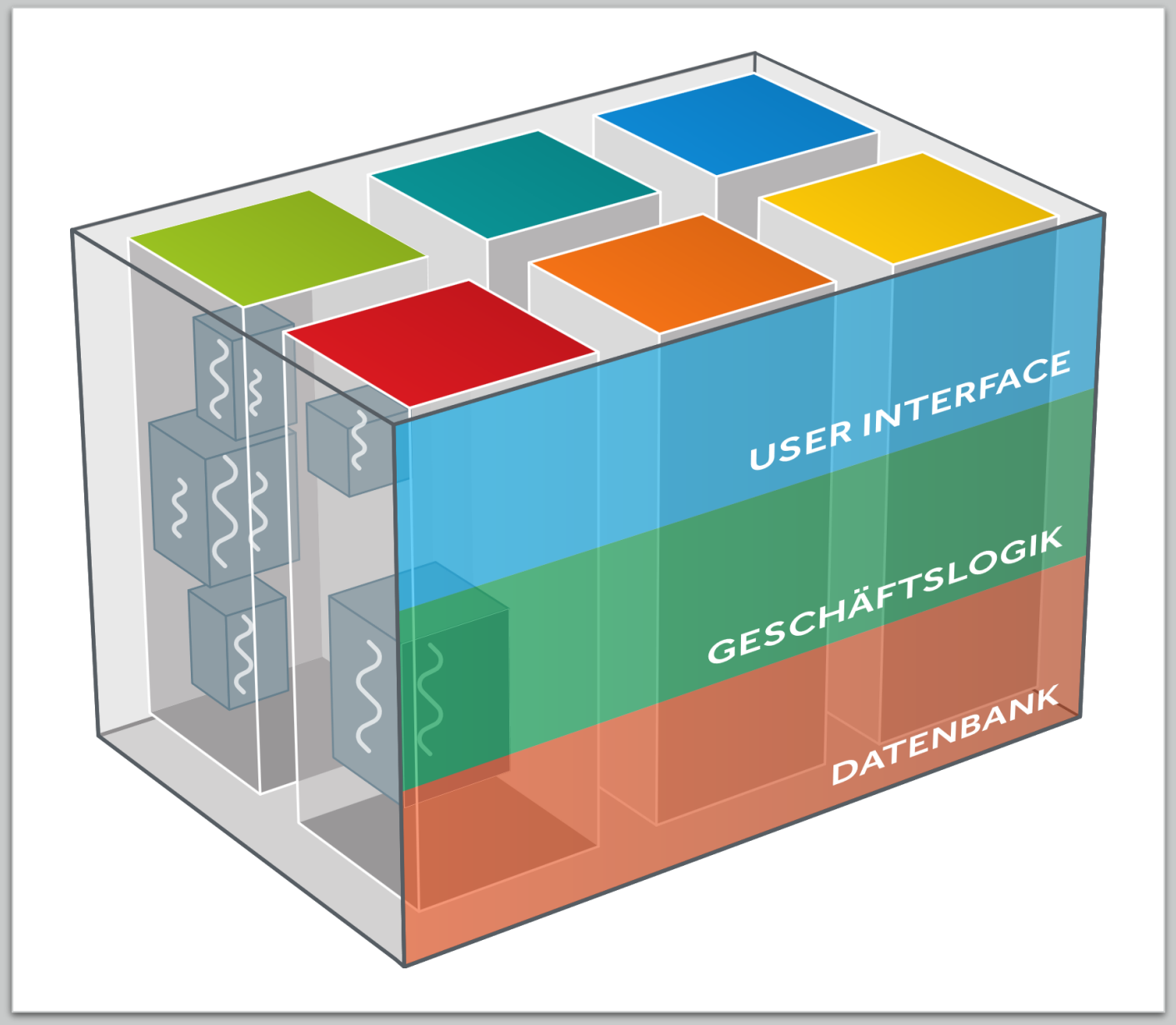
- Kontrollverlust führt zu schlechten Entscheidungen.
- Analogiebildung hilft uns bei der Lösungsfindung.
- Softwarearchitekturen lassen sich in Bezug auf die ISO 25010 Qualitätsnorm optimieren.

Photo by Clem Onojeghuo on Unsplash

iSAQB® Certified Professional for Software Architecture (Foundation Level)

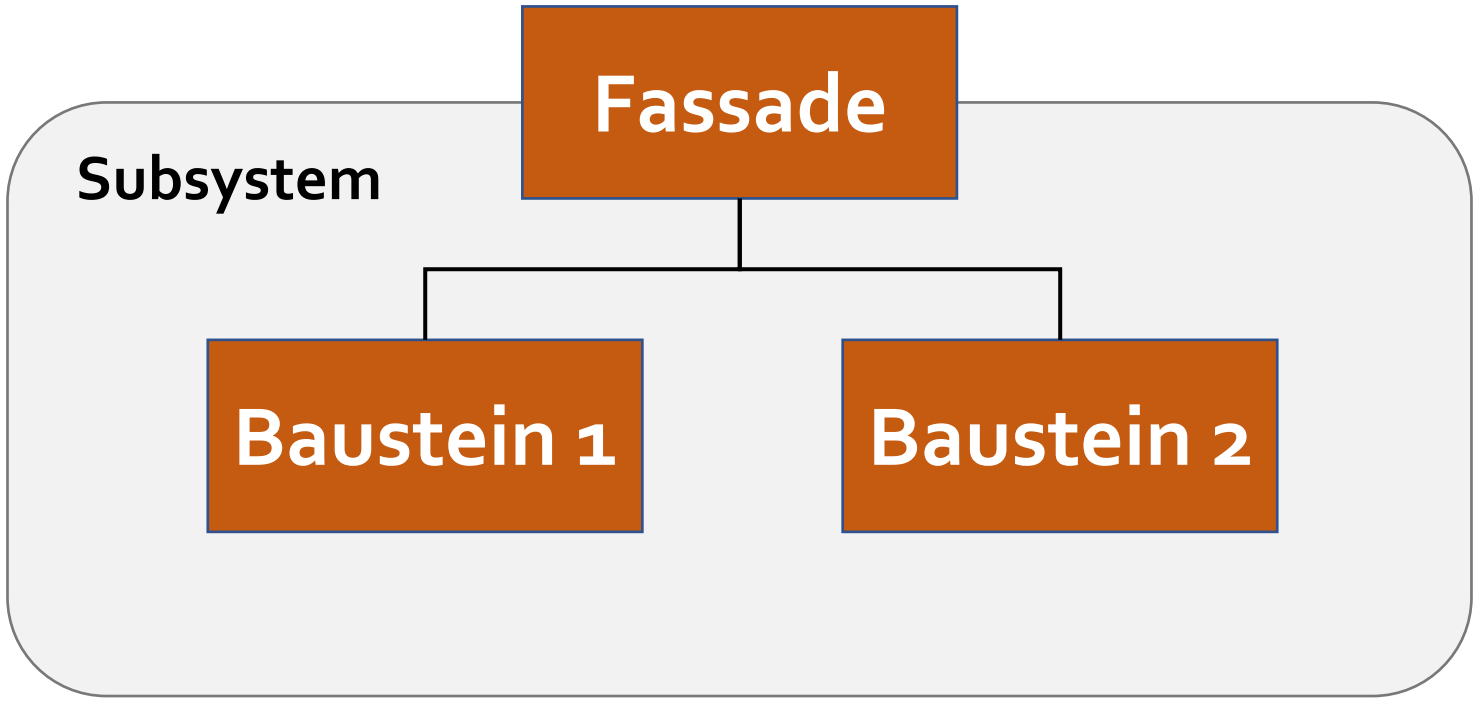
ACID

- Atomarität (Atomicity)
- Konsistenzerhaltung (Consistency)
- Isolation (Isolation)
- Dauerhaftigkeit (durability)



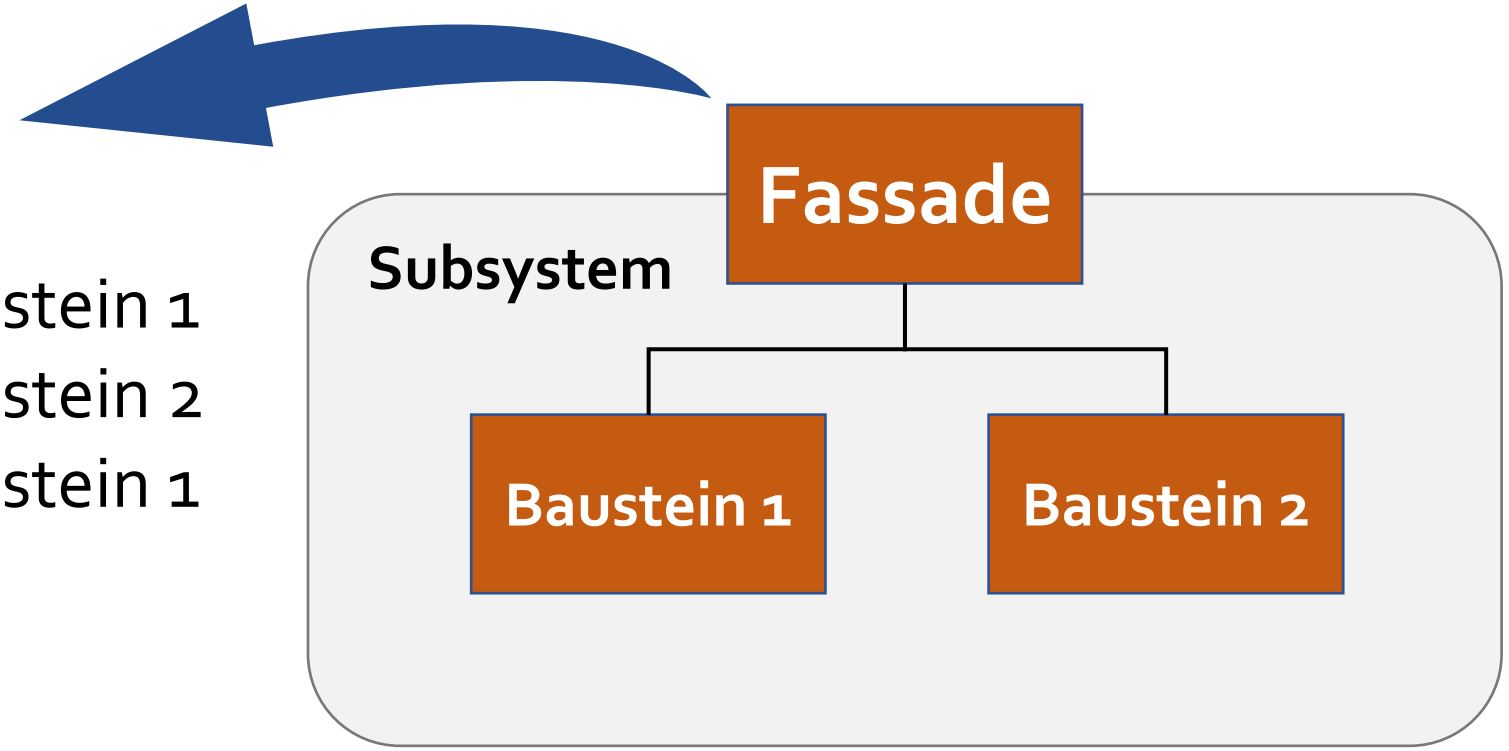
Beispiel: Fassade

Bietet vereinfachte Schnittstelle(n) zu einer Menge von Schnittstellen eines Subsystems.



Transaktion: ACID

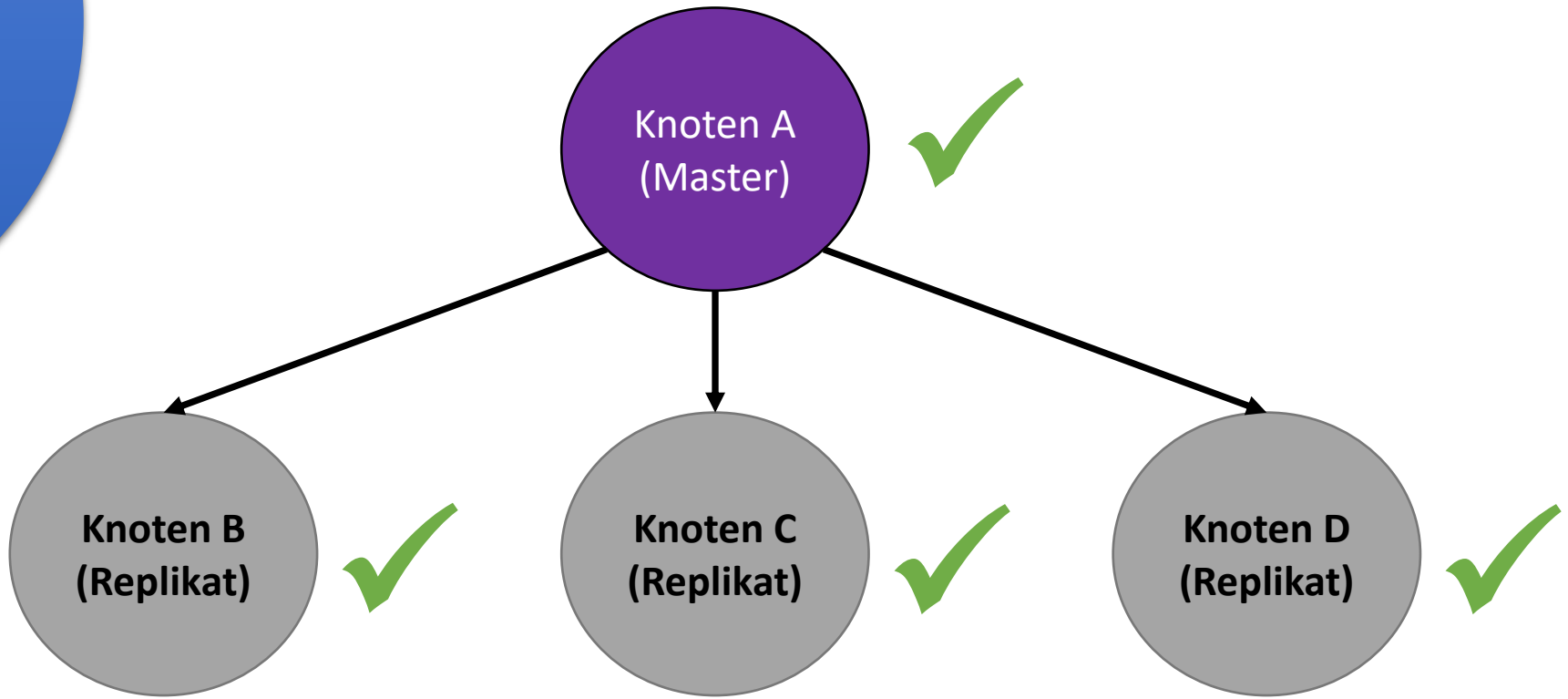
```
Operation 1   ->   Baustein 1
Operation 2   ->   Baustein 2
Operation 3   ->   Baustein 1
…             ->   …
```



Konsistenz (Consistency)

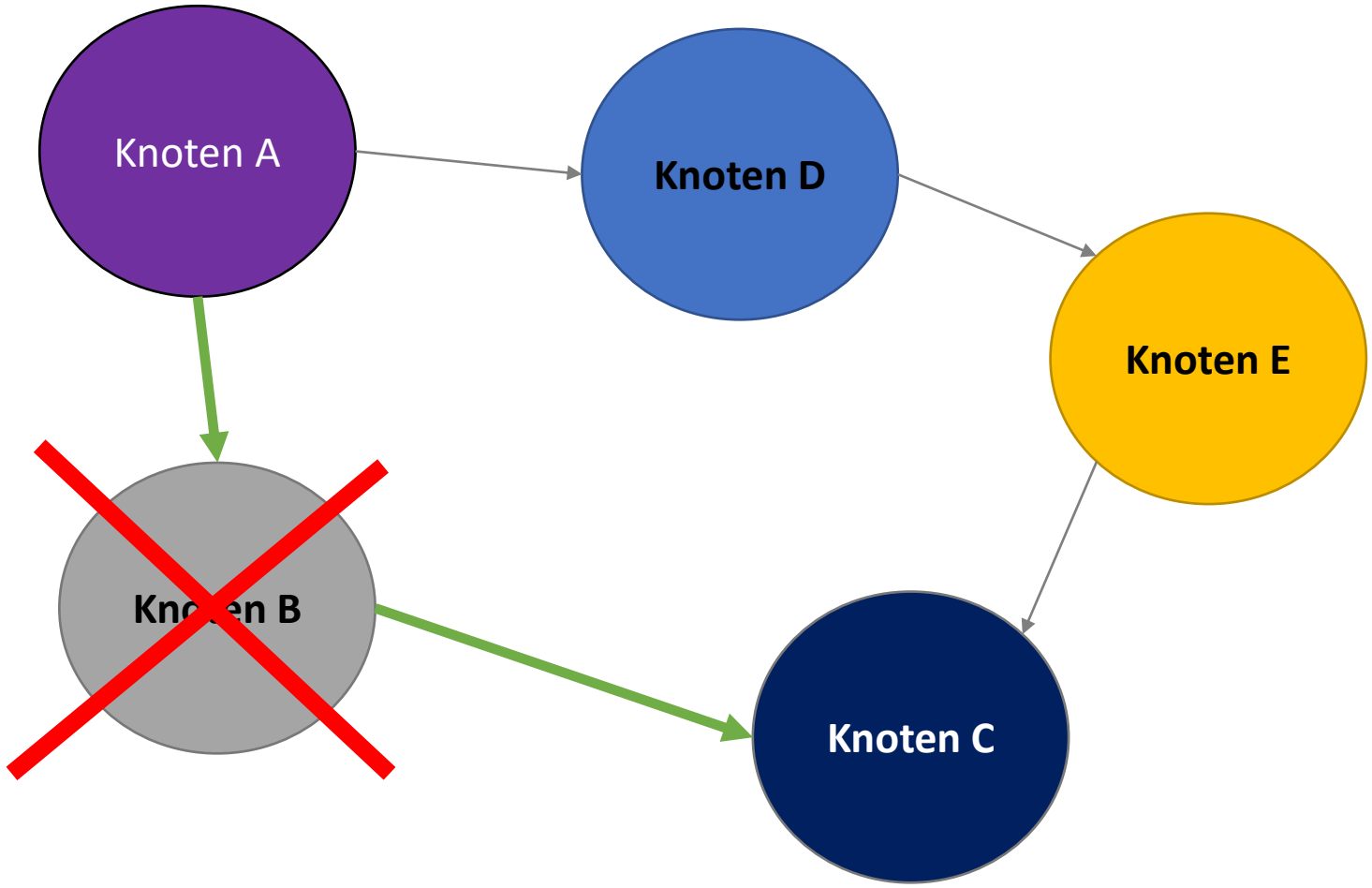
.. in einem verteilten System

Eigenschaften in verteilten Systemen



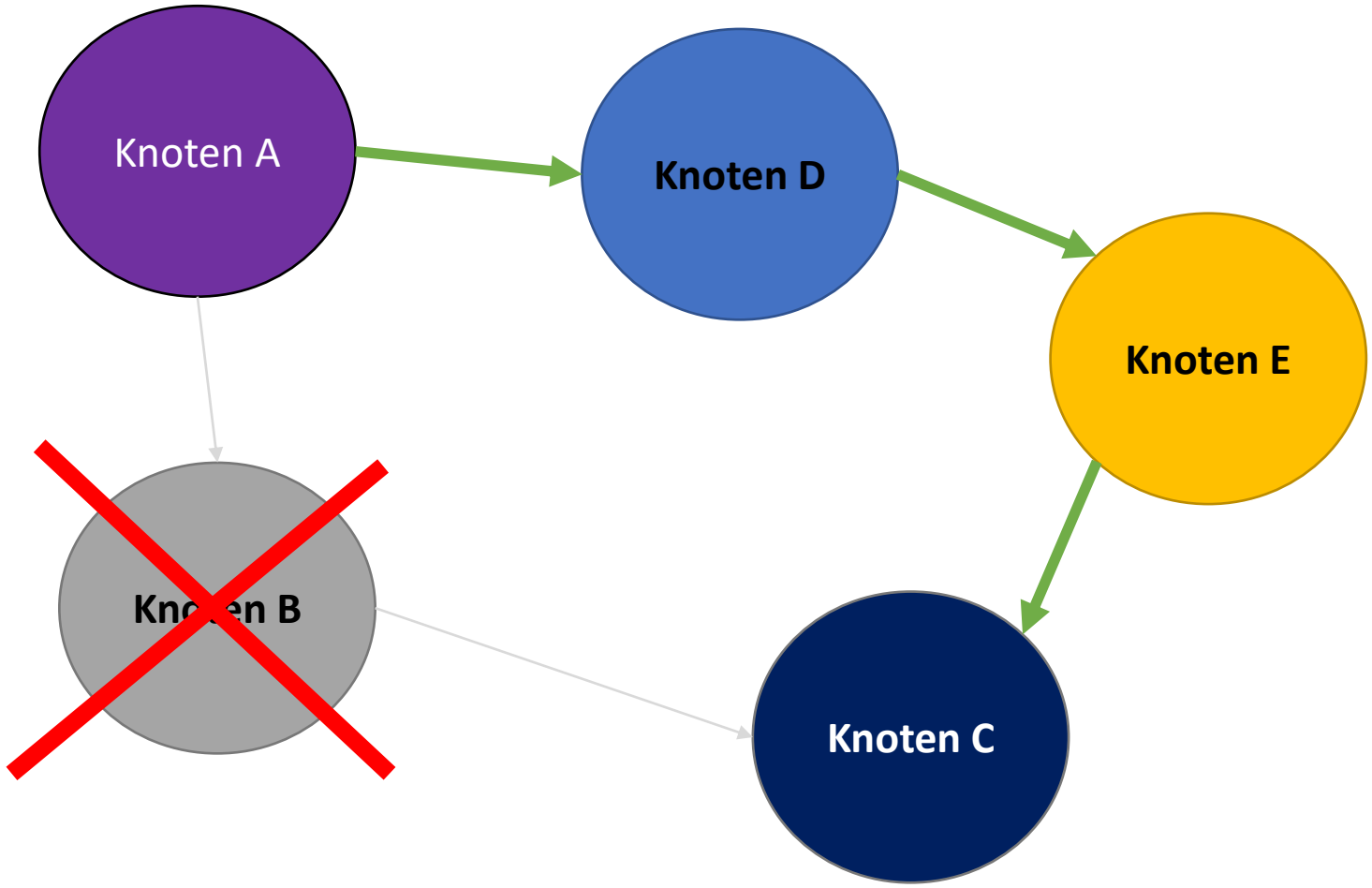
Eigenschaften in verteilten Systemen

Partitions-toleranz (Partition tolerance)



Eigenschaften in verteilten Systemen

Partitions-toleranz (Partition tolerance)



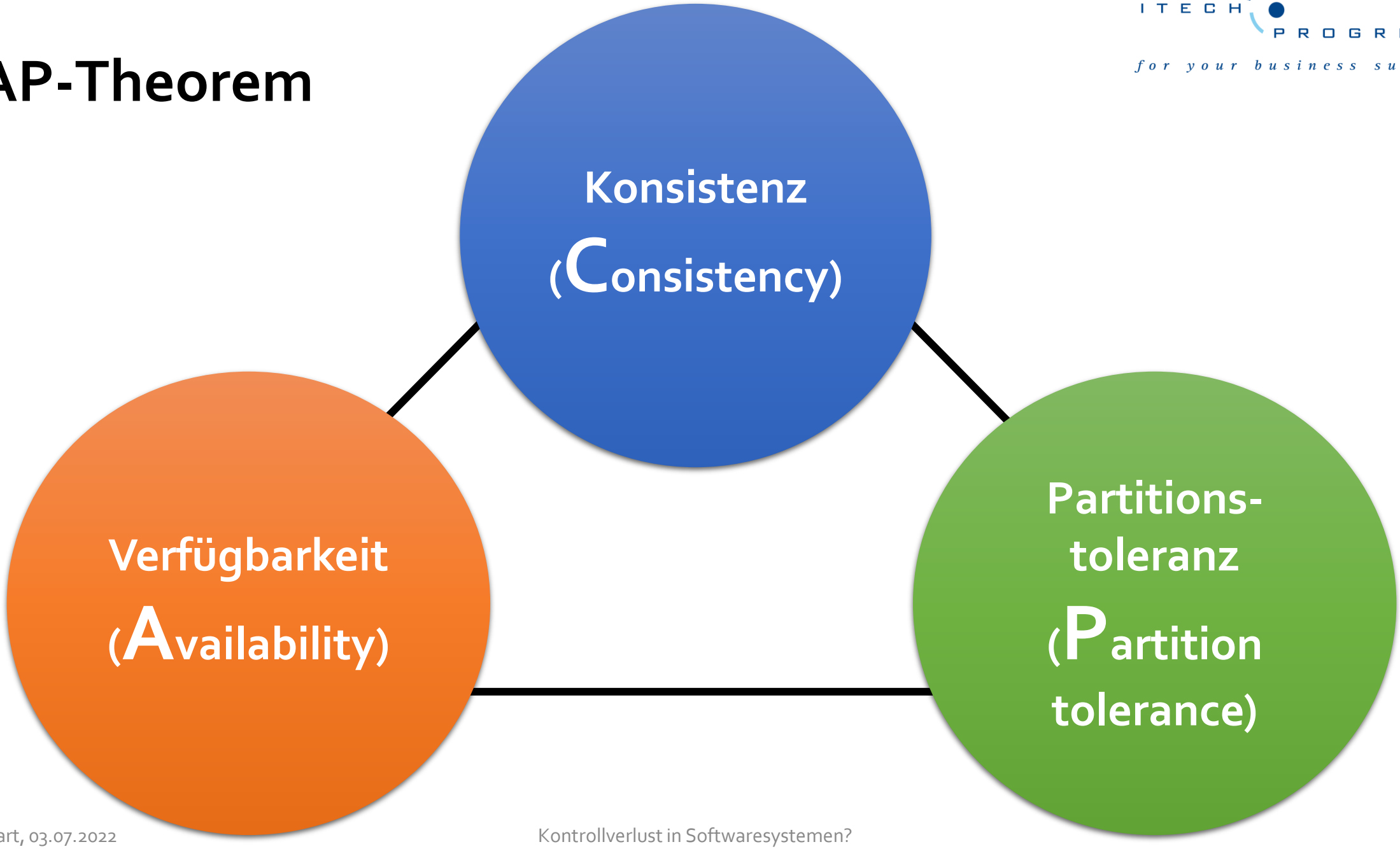
Verfügbarkeit (Availability)

Eigenschaften in verteilten Systemen

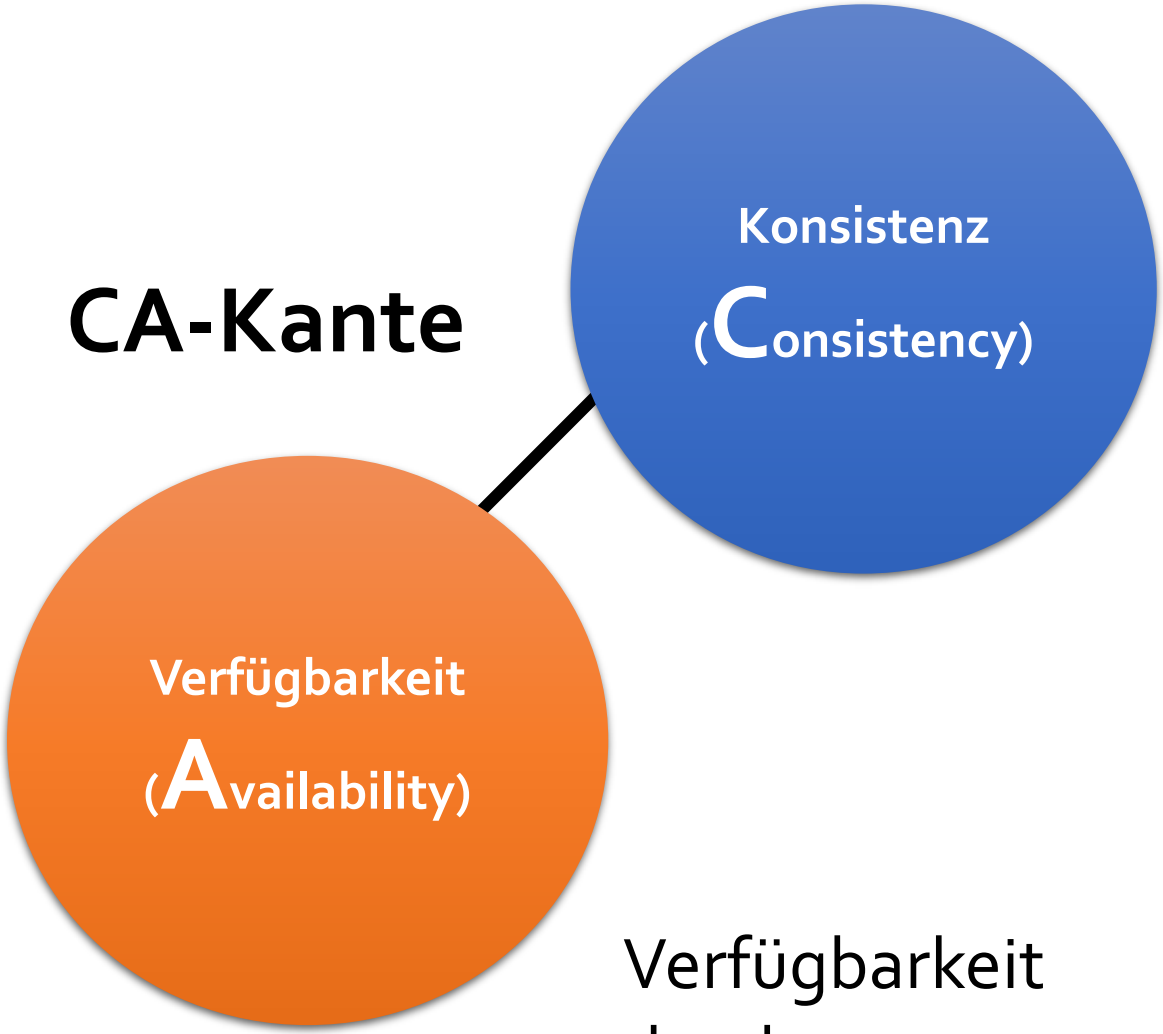
$$\text{Verfügbarkeit} = \frac{MTBF}{(MTBF + MTTR)}$$

- MTBF: Mean Time Between Failure
- MTTR: Mean Time To Recover

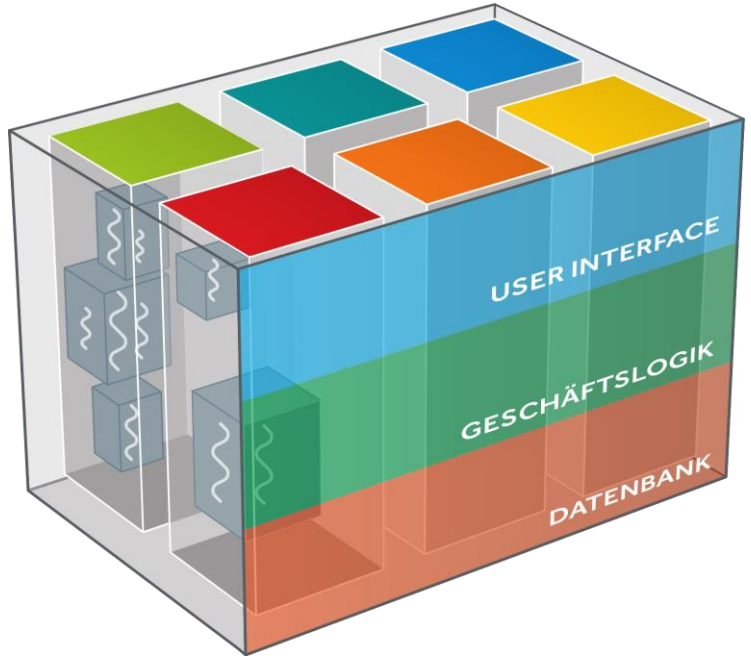
CAP-Theorem


Konsistenz
(Consistency)
Verfügbarkeit
(Availability)
Partitions-
toleranz
(Partition
tolerance)

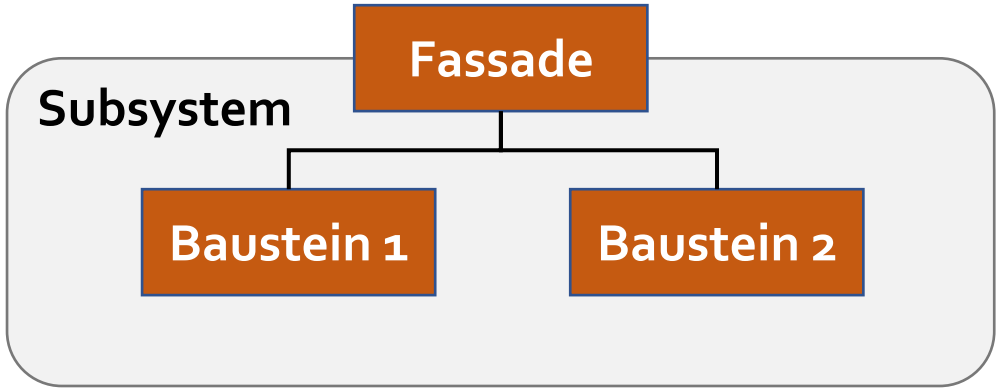
CA-Kante

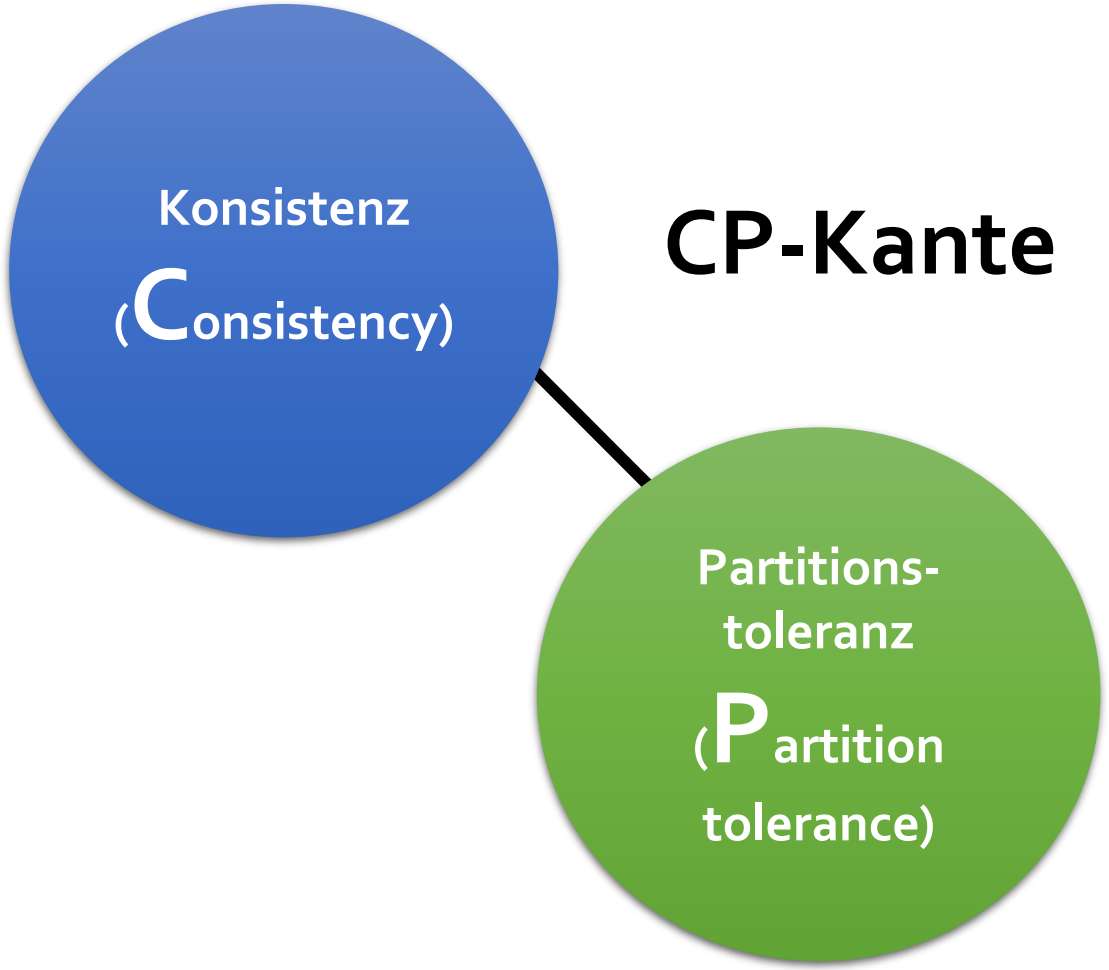


Konsistenz
z.B. durch
Transaktionen

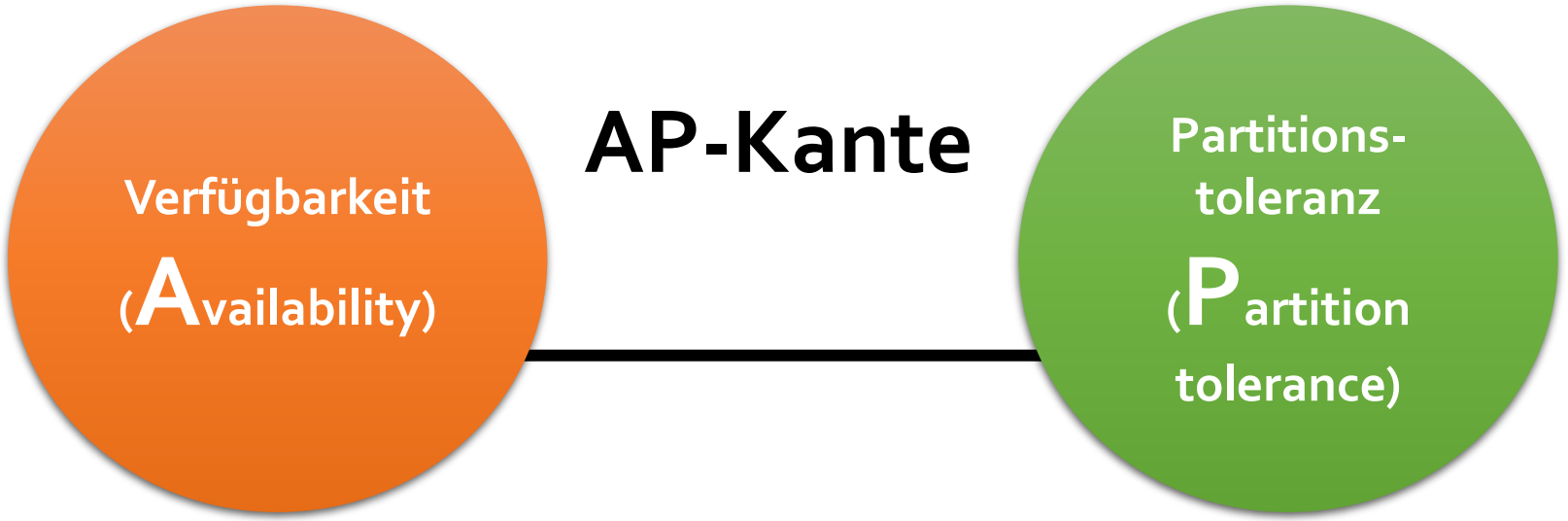


Verfügbarkeit
durch
Maximierung
der MTBF





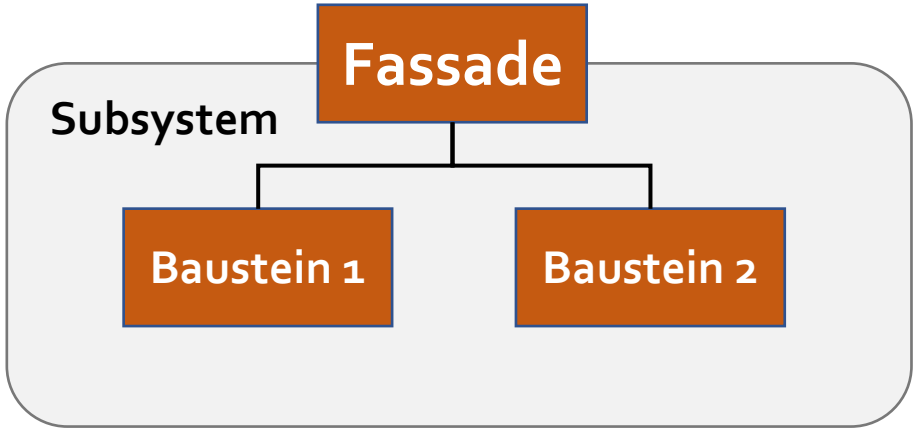
Spezialisierte Sonderfälle
$\Rightarrow$ z.B. Finanzsysteme

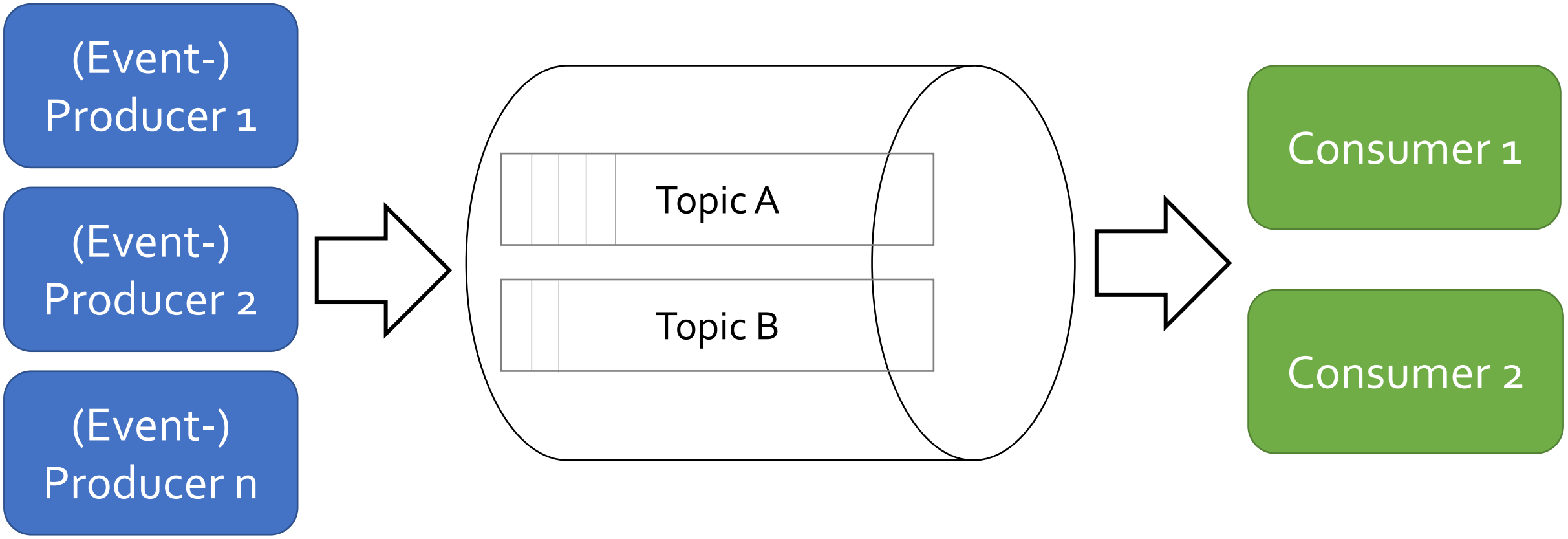


BASE:

- **B**asically **A**vailable
- **S**oft State
- **E**ventual Consistency
  - Optimistische Replikation

```
Operation 1   ->   Event 1, Topic A
Operation 2   ->   Event 2, Topic A
Operation 3   ->   Event 3, Topic B
...           ->   ...
```





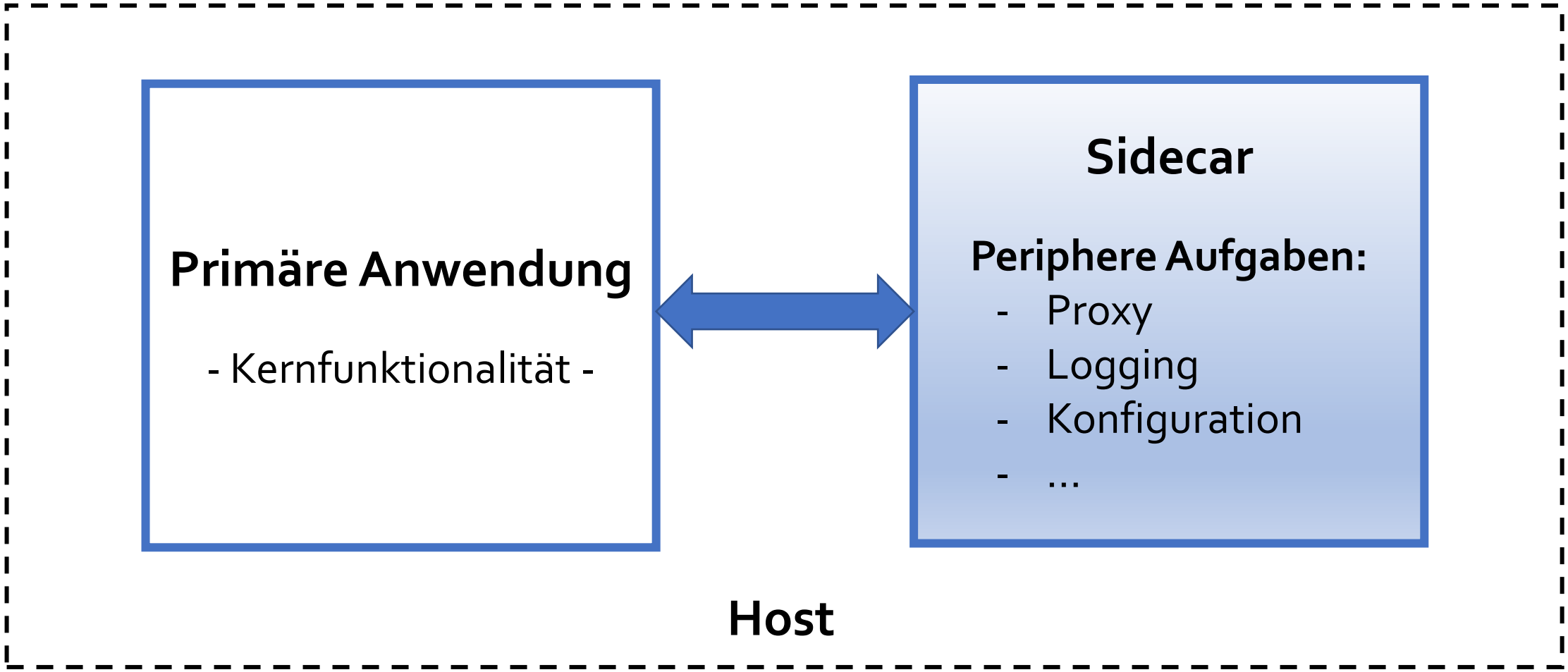
- Kontrollverlust führt zu schlechten Entscheidungen.
- Analogiebildung hilft uns bei der Lösungsfindung.
- Softwarearchitekturen lassen sich in Bezug auf die ISO 25010 Qualitätsnorm optimieren.
- Das CAP-Theorem hilft u.a. bei der Entscheidung ACID vs. BASE in verteilten Systemen.
- Eventual Consistency nimmt uns den Druck einer sofortigen (Daten-)Replikation.

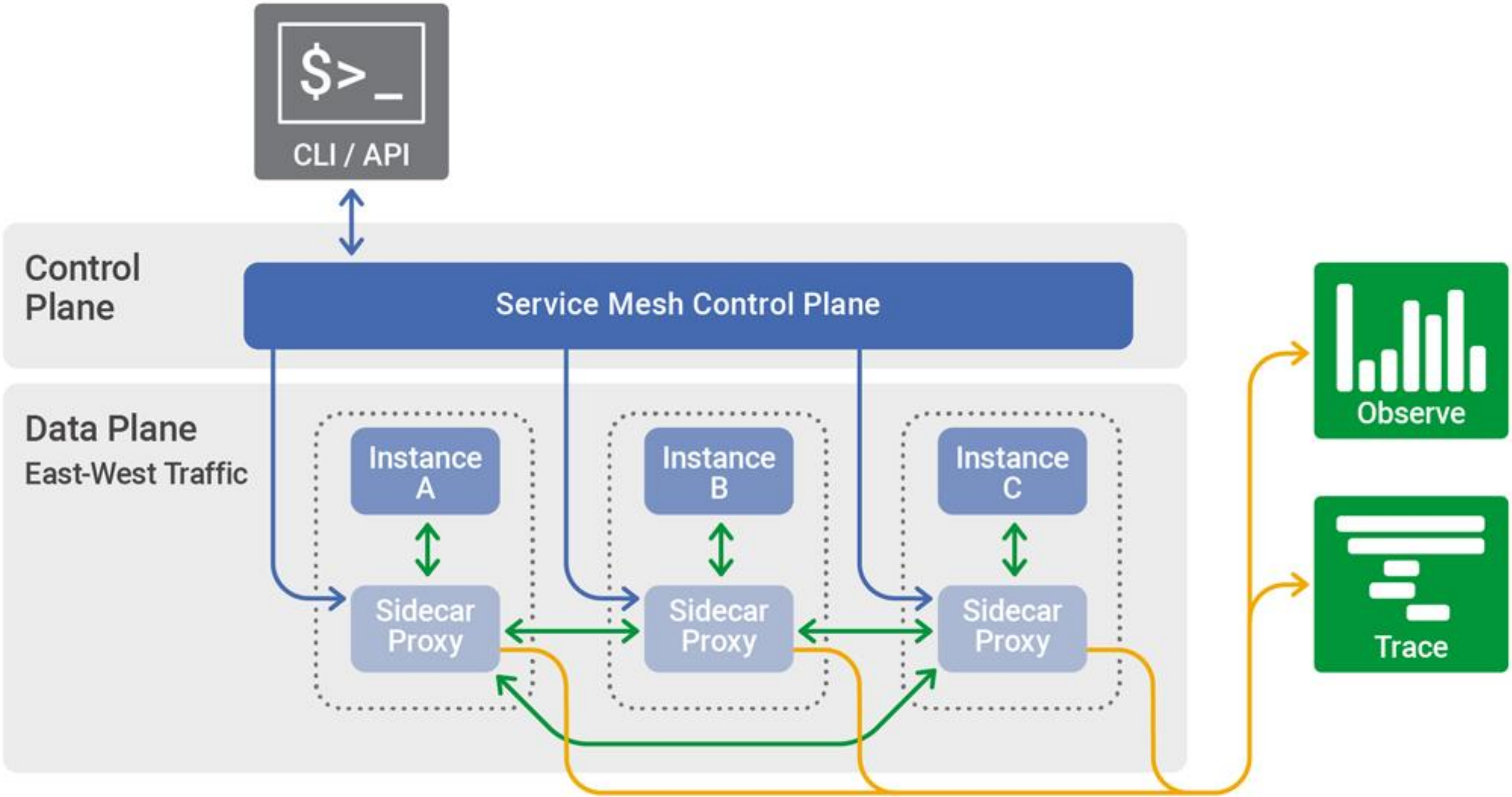
Photo by Clem Onojeghuo on Unsplash

iSAQB® Certified Professional for Software Architecture (Advanced Level)

Photo by hidde schalm on Unsplash

Sidecar-Muster




$>_
CLI / API
Control Plane
Service Mesh Control Plane
Data Plane
East-West Traffic
Instance A
Instance B
Instance C
Sidecar Proxy
Sidecar Proxy
Sidecar Proxy
Observe
Trace

- Kontrollverlust führt zu schlechten Entscheidungen.
- Analogiebildung hilft uns bei der Lösungsfindung.
- Softwarearchitekturen lassen sich in Bezug auf die ISO 25010 Qualitätsnorm optimieren.
- Das CAP-Theorem hilft u.a. bei der Entscheidung ACID vs. BASE in verteilten Systemen.
- Eventual Consistency nimmt uns den Druck einer sofortigen (Daten-)Replikation.
- Service Meshes erlauben die Auslagerung von Querschnittsthemen sowie erweiterten Aspekten in die Infrastruktur.

Photo by Clem Onojeghuo on Unsplash

iSAQB® Certified Professional for Software Architecture (Advanced Level)

The iSAQB®-Community

The iSAQB Community is

the community for all iSAQB alumni,

being ***Serious about Software Architecture,***

led by its community members,

organized in local communities, having their own **Meet-Ups**,

discussing current **trends** and **topics**, **networking**

and **finding new like-minded people,**

supported by the iSAQB

25.07.2022 **ab 18:00 Uhr**

Kultur Kiosk Stuttgart


Architecture Evaluation
Architecture Documentation
Improve
Flexible Architectural Models
Soft Skills
Web Architectures
CPSA-F
Domain Driven Design
Service Oriented Architecture
Agile Software Architecture
Enterprise Architecture Management