



Architektur-Frameworks in der Cloud destillierte Prinzipien und Best Practices

Alexander Kaserbacher, embarc

JavaLand 2023

Dienstag, 21.03.2023



Alexander Kaserbacher



ak@embarc.de



@alexksbr



@alexksbr@mastodon.online



linkedin.com/in/alexksbr



embarc.de



Cloud-Charakteristika



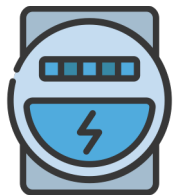
Self-Service: Management von Ressourcen automatisiert per Web-Oberfläche oder API



Resource-Pooling: Sammeln von verschiedenen Ressourcen auf der selben Hardware (Virtualisierung)



Elastizität: Flexible Skalierung „nach oben“ und „nach unten“



(Public Cloud) Pay-Per-Use: Nur für Ressourcen bezahlen, die auch benutzt werden



Wir packen unsere Koffer...



- Was muss ich alles einpacken?
- Wie kann ich sicherstellen, dass ich nichts Wichtiges vergesse?

Die drei Architektur-Ratgeber



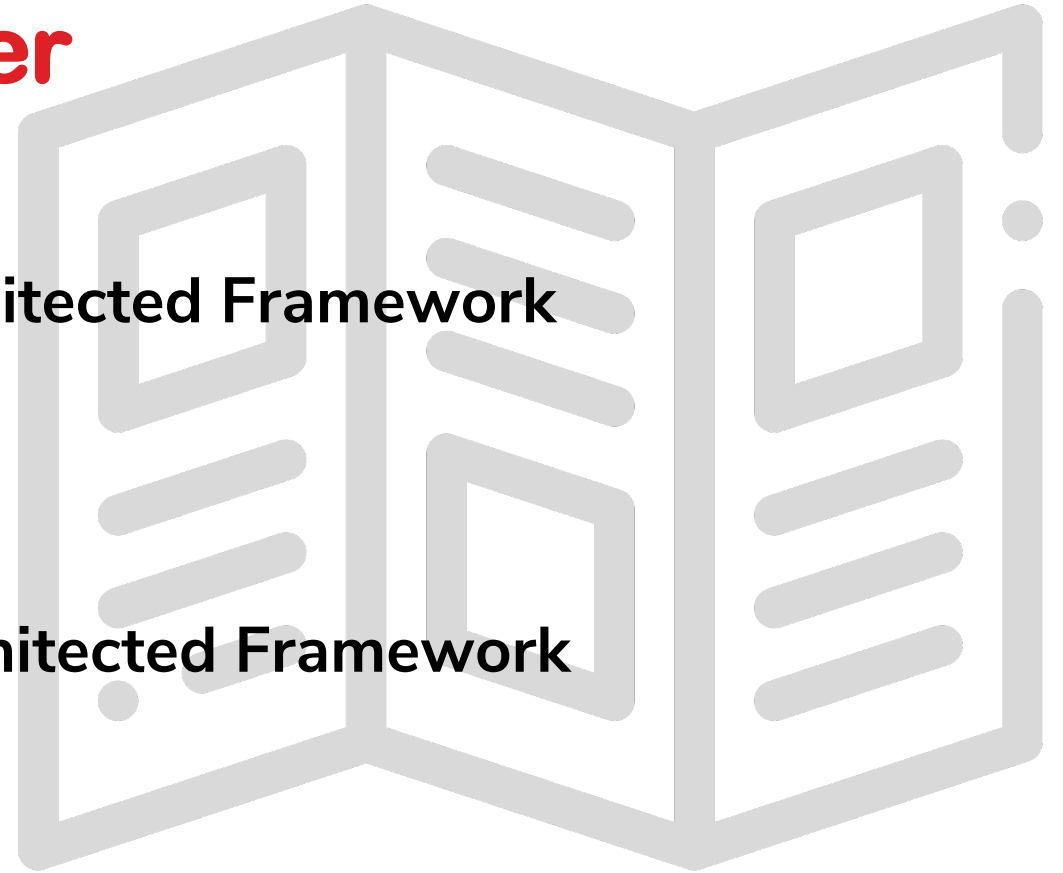
AWS Well-Architected Framework



Azure Well-Architected Framework



Google Cloud Architecture Framework



Verallgemeinerte Struktur



Pillars

(5-6 pro Framework)



Die Pillars



Operational
Excellence



Security



Reliability



Performance
Efficiency



Cost
Optimization



Sustainability
(AWS Well-
Architected
Framework)



System Design
(Google Cloud Architecture Framework)



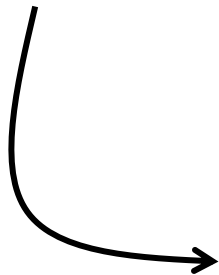
Verallgemeinerte Struktur



Pillars
(5-6 pro Framework)



Designprinzipien
(4-7 pro Pillar)



Best Practices
(~ 50 pro Pillar)



Wir packen unsere Koffer...



Designprinzip

Pack wettergerechte Kleidung ein!



Best Practice

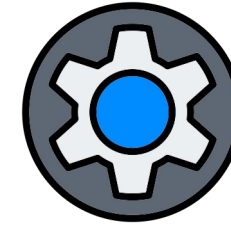
Nimm eine Zahnbürste mit!



Ein Designprinzip



Etabliere „Everything as code“:
Applikationen, Infrastruktur,
Konfiguration, betriebliche Vorgänge

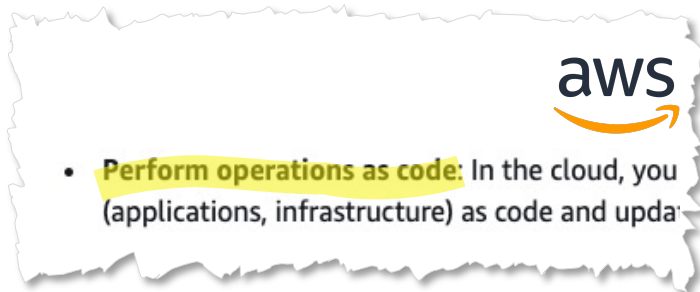


Operational
Excellence

Wie?

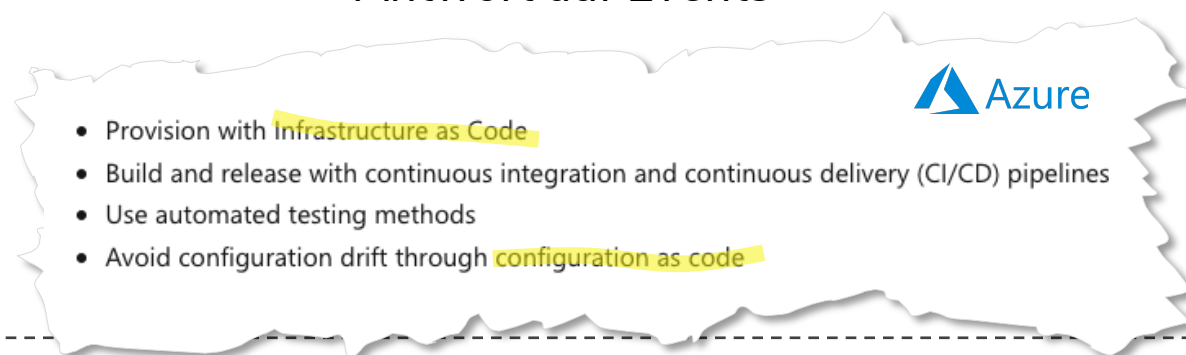
- Infrastructure as Code: Terraform, Pulumi, AWS CloudFormation, Azure Resource Manager ...
- Skripte für betriebliche Vorgänge
- Bei bestimmten Events automatisch ausgeführt

Aus den Frameworks...



Warum?

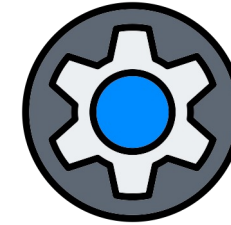
- Verwendung von Versionskontrolle möglich
- Automatisches Erstellen von Umgebungen
- Konsistenz zwischen Umgebungen
- Fehlerrate verringern
- Skripte für betriebliche Vorgänge: Konsistente Antwort auf Events



Best Practice



Nutze mehrere Umgebungen und provisioniere sie automatisiert



Operational Excellence

Aus den Frameworks...



- Create your cloud resources automatically, including the deployment or test environments for your CI/CD pipeline.
- Treat infrastructure changes like you treat application changes. For example, ensure changes to the configuration are reviewed, tested, and can be audited.
- Have a single version of the truth for your cloud infrastructure.
- Replicate your cloud environment as needed.
- Roll back to a previous configuration if necessary.

Dynamically provision environments. Once you have your IaC configurations defined, you can provision environments more efficiently. This agility can be enormously helpful when you're testing your solution. For example, you could quickly provision a duplicate of your production environment that can then be used for security penetration tests, load testing or help a developer track down a bug.

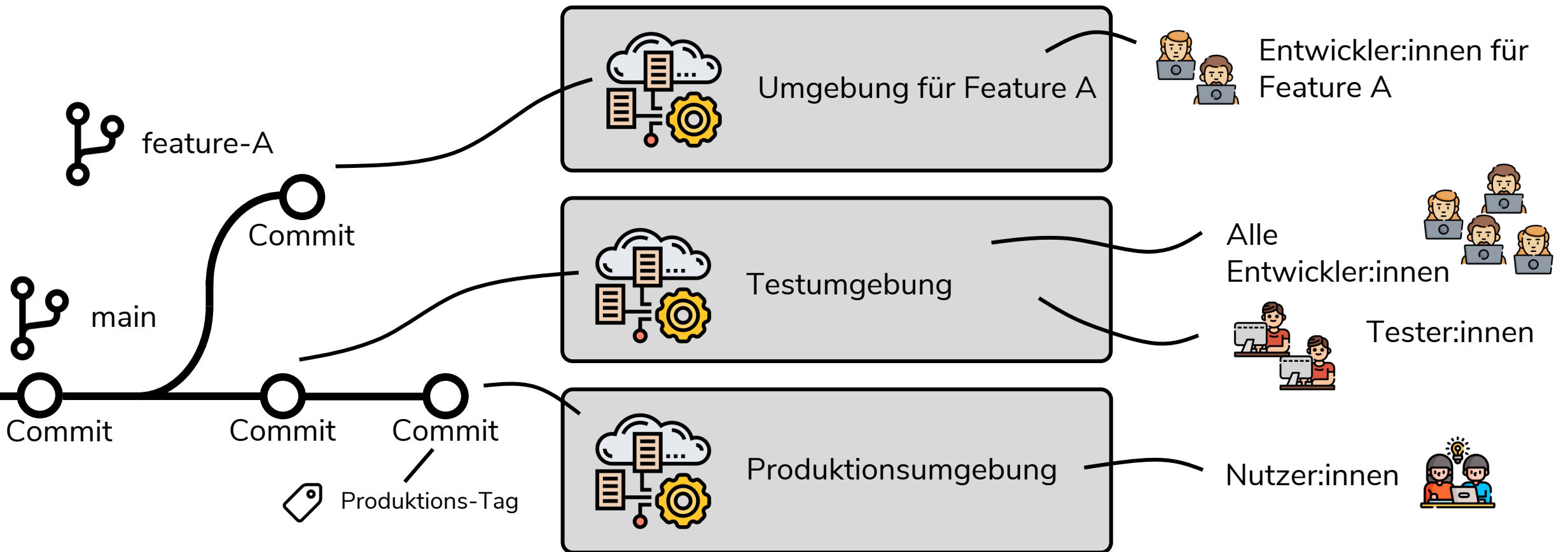


OPS05-BP08 Use multiple environments

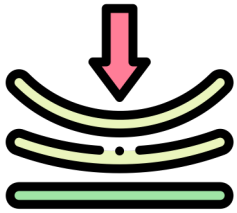
[PDF](#) | [RSS](#)

Use multiple environments to experiment, develop, and test your workload. Use increasing level of confidence your workload will operate as intended when deployed.

Git-Branches und Environments



Ein Designprinzip



Erwarte Ausfälle und entwirf dein System entsprechend resilient



Reliability

Wie?

- Unabhängig deploybare, fachlich orientierte Services
- Zustandslose Services
- „Graceful“ degradation bei Service-Kommunikation

Warum?

- Zuverlässigkeit ist auch ein Thema des Systementwurfs

Aus den Frameworks...



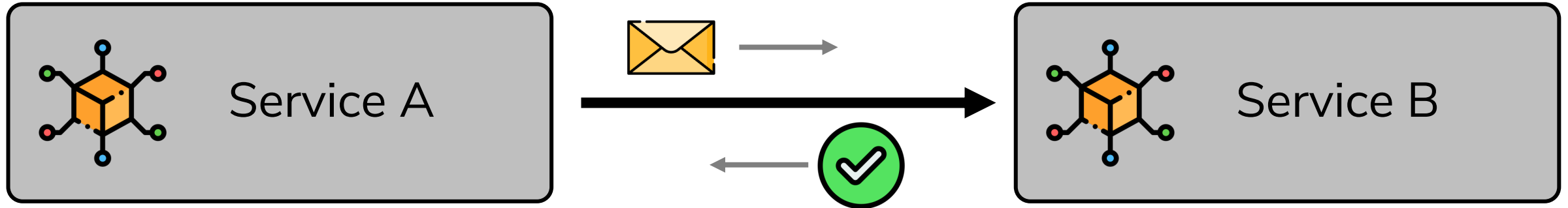
Automatically recover from failure: By monitoring a workload for key metrics, recovery should be a measure of business value, not of the technical aspects of the system. Automated recovery processes that work around or repair the failure.



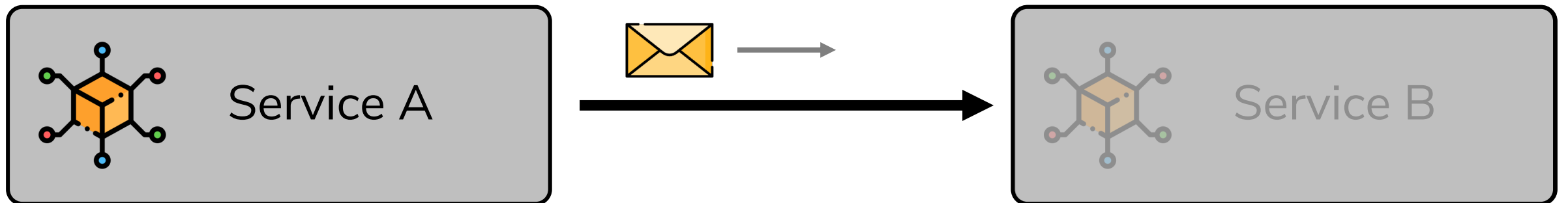
Design for failure

Failure is impossible to avoid in a highly distributed and multi-tenant environment.

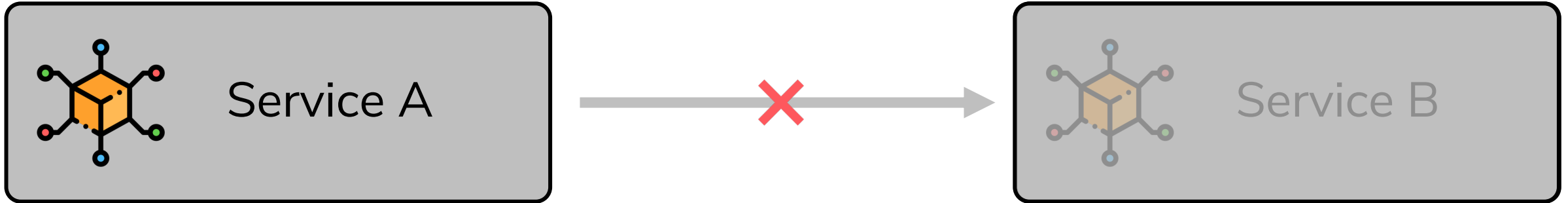
Ein normaler Aufruf



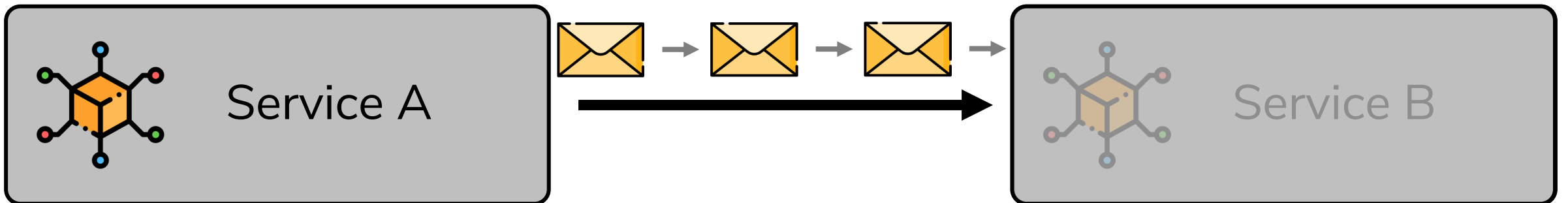
Ein normaler Aufruf geht schief



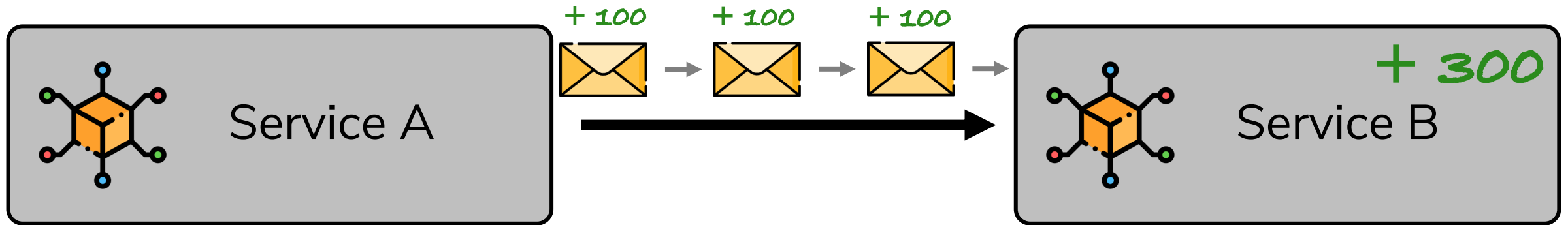
Circuit Breaker



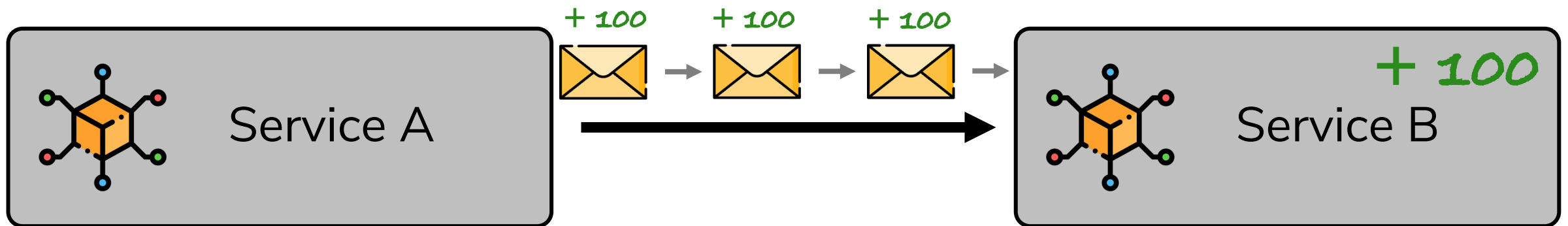
Retries



Die Gefahr von Retries



Idempotenz



Best Practice



Mach Nachrichten und Befehle idempotent



Reliability

Aus den Frameworks...



Your system architecture should **make actions idempotent** - if you perform them multiple times in succession, it should produce the same results as a single invocation.

RELO4-BP04 **Make all responses idempotent**

[PDF](#) | [RSS](#)

An idempotent service promises that each request is completed exactly once.



Design tasks and messages to be idempotent, where possible. An operation should be idempotent, meaning that if it is repeated multiple times and produce the same result. This can ensure that the system is reliable.

Best Practice



Klassifiziere Daten nach Vertraulichkeit



Security

Aus den Frameworks...



Automatically classify your data

Perform data classification as early in the data management lifecycle. Usually, data classification efforts require only a few categories, such as



SEC07-BP01 Identify the data within your workload

[PDF](#) | [RSS](#)

You need to understand the type and classification of data your workload is processing, the associated business processes



All important data should be classified and encrypted with an encryption standard. Classify information storage objects. Use encryption to make sure the contents of files cannot

Klassifizierungsschema

z.B. nicht verschlüsseln, allgemein zugänglich

- **Public** – Daten für allgemeinen, öffentlichen Zugriff
- **Internal** – Nicht-sensitive Daten, die nicht öffentlich zugreifbar sind
- **Confidential** – Sensitive Daten, die unternehmensintern verteilt werden dürfen
- **Restricted** – Hochsensitive Daten, die auch unternehmensintern nur in Ausnahmefällen verteilt werden dürfen

z.B. verschlüsseln, intern zugänglich nur für Administratoren nach Genehmigung

z.B. nicht verschlüsseln, intern zugänglich für Entwickler und Administratoren

z.B. verschlüsseln, intern zugänglich nur für Administratoren

Referenz: <https://cloud.google.com/architecture/framework/security/data-security#dataautoclass>

Services zur automatischen Klassifizierung



Amazon Macie



Google Cloud Data
Loss Prevention



Azure Information
Protection



Datadog Sensitive
Data Scanner

Ein Designprinzip



Analysiere und optimiere Kosten kontinuierlich



Cost Optimization

Wie?

- Kosten-Monitoring aufsetzen und Kosten kategorisieren
- Kosten-Limits und Alarmer konfigurieren
- Regelmäßige Cost-Reviews durchführen
- Return on Investment berechnen

Warum?

- Um Kosten zu sparen und unsere Systeme effizienter zu gestalten
- Opportunitätskosten optimieren
- Kostentransparenz

Aus den Frameworks...



Continuously monitor and optimize cost management

To provision resources dynamically and to scale with demand.



- Monitor and control cost: Best practices, tools, and techniques to track and optimize Google Cloud.

Best Practice



Nutze Tags um Kostenstrukturen abzubilden



Cost Optimization

Aus den Frameworks...



Track and allocate cost using labels

Labels are key-value pairs that you can use to tag projects and resources. To cat granularity, establish a labeling schema that suits your orgar



- Use resource tag policies to build reports. Tags can be used to identify the owners of sy

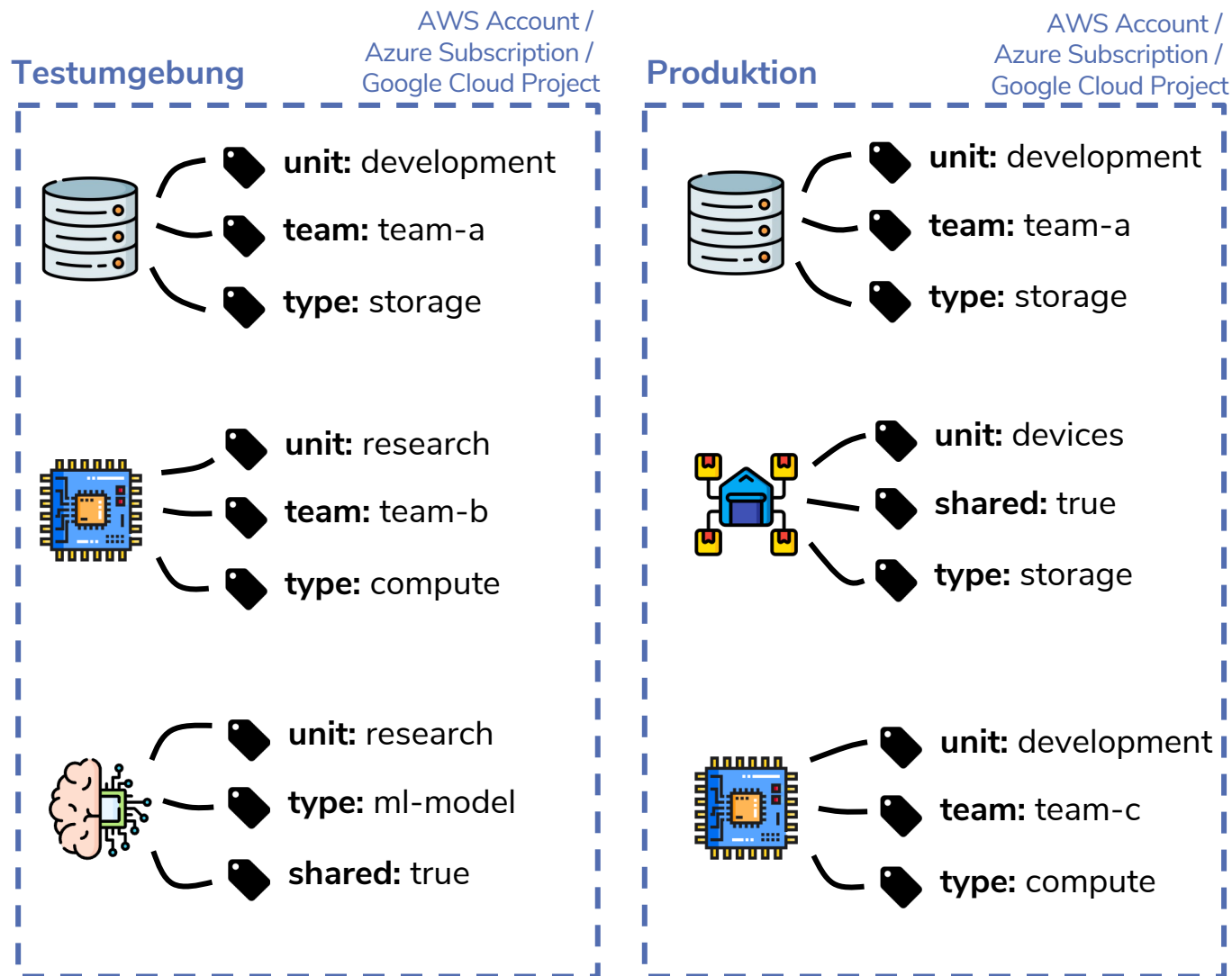
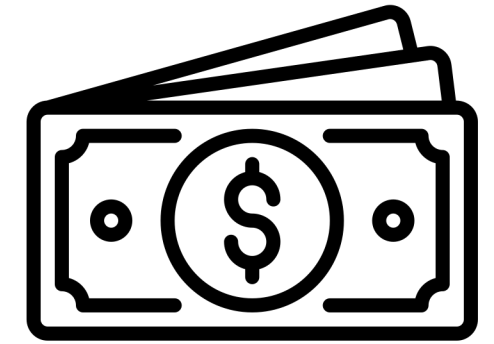


COST03-BP05 Add organization information to cost and usage

[PDF](#) | [RSS](#)

Define a tagging schema based on organization, and workload attributes, and cost allocation categories. Implement tagging across all resources. Use Cost

Kostenstruktur mit Tags abbilden



- Bestimmte Tags verpflichtend machen
z.B. Ressourcen müssen „unit“ und „type“-Tags haben
- Tags erleichtern Kostenanalyse
 - Wie viel kosten unsere „shared“ Ressourcen?
 - Wie teuer sind die Storage-Ressourcen von Team B?
 - Haben wir Machine Learning-Kosten außerhalb der Research-Abteilung?

Einsatz der Frameworks ...

... als Review-Tool



Anwendung gegen Best Practices halten

... als Unterstützung bei Architekturentscheidungen

The screenshot shows the AWS Well-Architected Tool interface. At the top, there's a navigation bar with the AWS logo, 'Services', a search bar, and user information for 'alexkaserbacher'. A blue banner at the top left contains a 'New feature' notification about integrations with AWS Trusted Advisor and AWS Service Catalog AppRegistry.

The main content area is titled 'AWS Well-Architected Framework' and is part of a 'Review workload' session. The breadcrumb trail is: Well-Architected Tool > Workloads > MyOldWorkload > AWS Well-Architected Framework > Review workload. The current focus is on 'REL 10. How do you use fault isolation to protect your workload?'. Below the title is an 'Ask an expert' button.

The 'REL 10' section includes a paragraph: 'Fault isolated boundaries limit the effect of a failure within a workload to a limited number of components. Components outside of the boundary are unaffected by the failure. Using multiple fault isolated boundaries, you can limit the impact on your workload.' Below this is a radio button option: 'Question does not apply to this workload'. Underneath, it says 'Select from the following' and lists several options with checkboxes:

- Deploy the workload to multiple locations
- Select the appropriate locations for your multi-location deployment
- Automate recovery for components constrained to a single location
- Use bulkhead architectures to limit scope of impact
- None of these

On the right side, there's a 'Helpful resources' panel with an 'Ask an expert' button and a list of video links related to AWS re:Invent 2018 and 2019 topics, such as 'Architecture Patterns for Multi-Region Active-Active Applications (ARC209-R2)' and 'Introducing The Amazon Builders' Library (DOP328)'. Below the resources is a section titled 'Deploy the workload to multiple locations' with a paragraph explaining the concept of distributing workload data and resources across multiple Availability Zones and AWS Regions.



Leitfragen

1 Trifft die Best Practice auf unseren Kontext und unser System zu?

wenn ja

wenn nein

aussortieren

2

Je mehr der folgenden Fragen du mit ja beantwortest, desto weiter oben wird die Best Practice einsortiert



- ✓ Ist es recht offensichtlich, dass wir die Best Practice nicht erfüllen?
- ✓ Ist uns die Pillar dieser Best Practice sehr wichtig?
- ✓ Haben wir im Team über dieses Thema nie explizit gesprochen? (*Auch wenn wir das Gefühl haben, dass wir die Best Practice erfüllen*)
- ✓ Behandelt die Best Practice ein Thema, das immer wieder in Diskussionen aufkommt?
- ✓ Sind im geplanten Workshop genug beteiligte Stakeholder anwesend, um sinnvoll über die Best Practice diskutieren zu können?

Einsatz der Frameworks ...

... als Review-Tool



Anwendung gegen Best Practices halten

... als Unterstützung bei Architekturentscheidungen

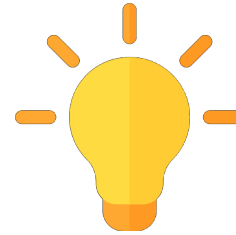


Als Unterstützung bei Architekturentscheidungen

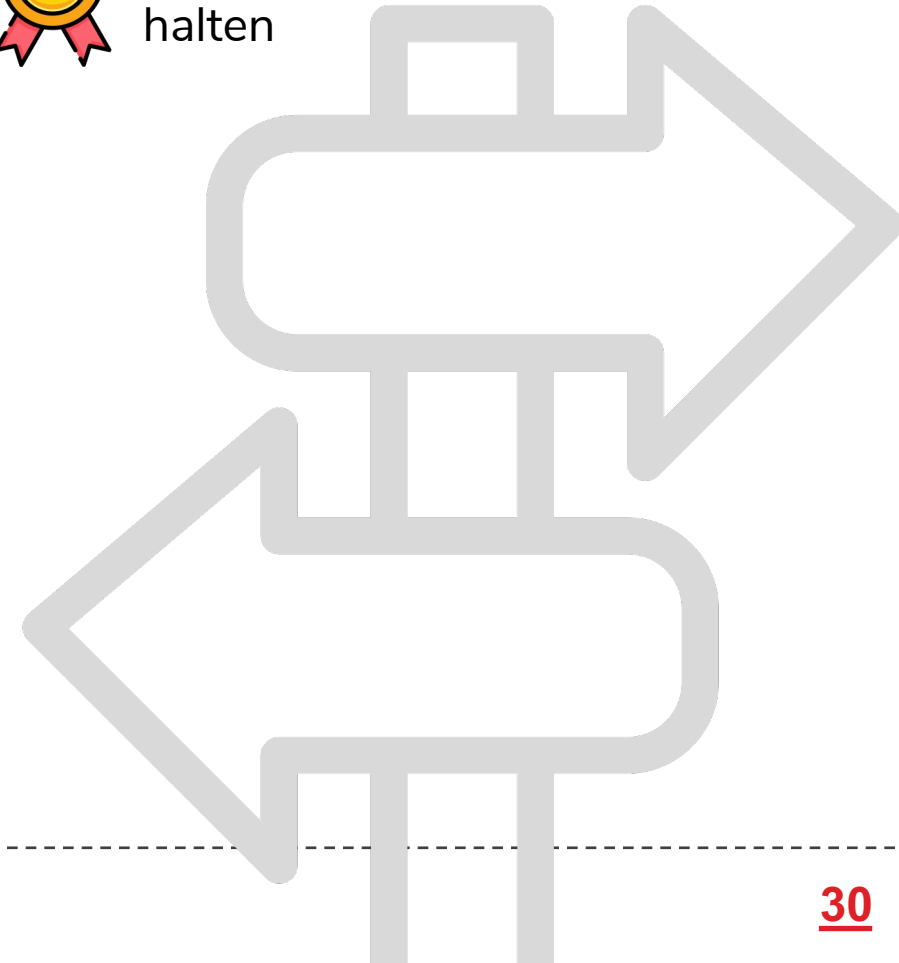
Lösungsoptionen


Fragestellung

 Designprinzipien
+ ausgewählte Best Practices



Gegen mehr Best Practices
halten





OPS08-BP06 Alert when workload outcomes are at risk

~~On AWS, you can use [Amazon CloudWatch Synthetics](#) to create canary scripts to monitor your endpoints and APIs by performing the same actions as your customers. The telemetry generated and the [insight gained](#) can enable you to identify issues before your customers are impacted.~~

On Kubernetes, you can use Kuberhealthy ...



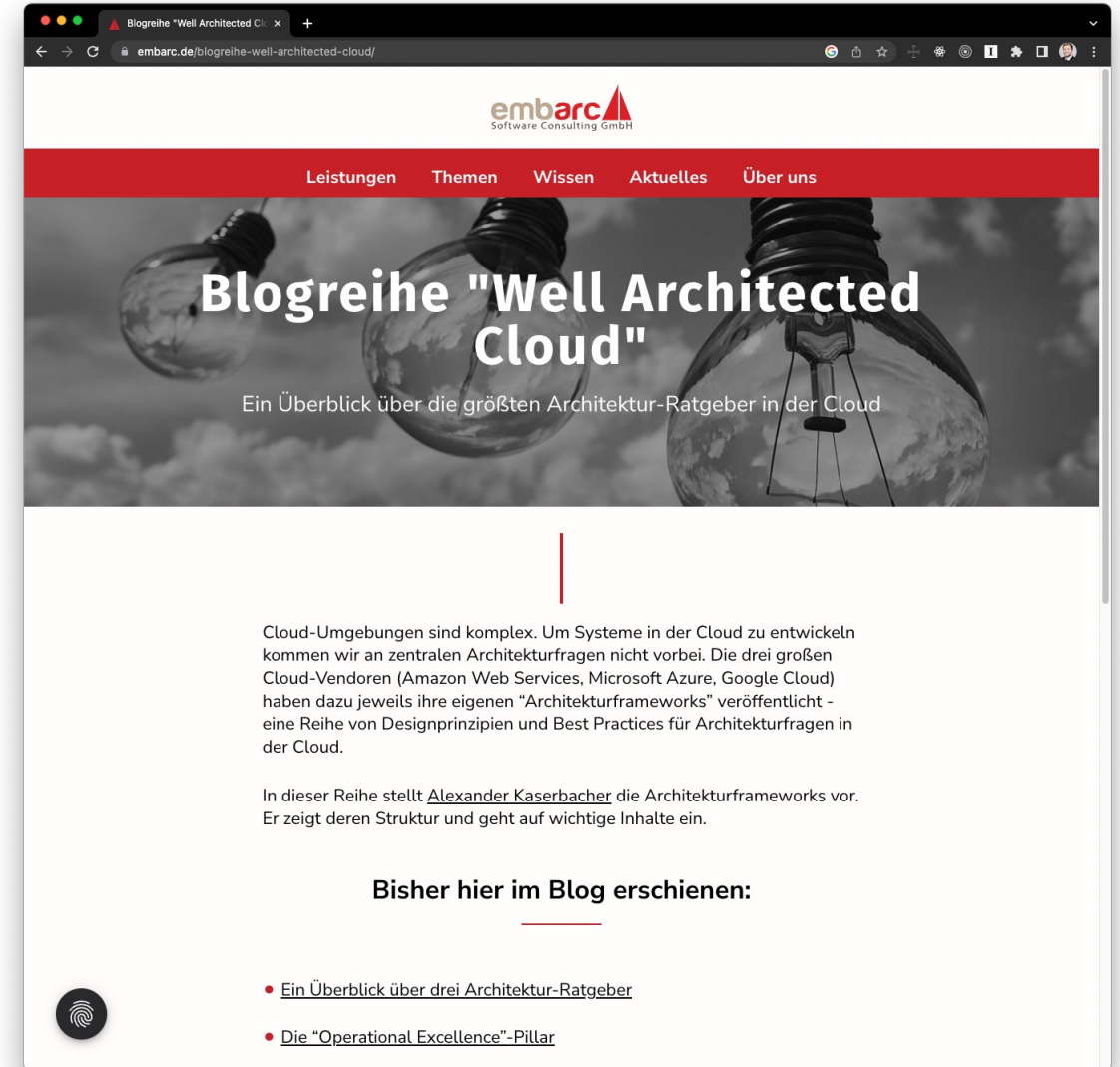
<https://www.cncf.io/projects/kuberhealthy/>

Blogreihe „Well Architected Cloud“

Blogbeiträge

- Überblick
- Pillar „Operational Excellence“
- Pillar Security
- Pillar Reliability
- Pillar „Performance Efficiency“
- Pillars „Cost Optimization“ und Sustainability
- Methodischer Umgang





➔ embarc.de/blogreihe-well-architected-cloud



Vielen Dank.

Ich freue mich auf Eure Fragen!

→ embarc.de/architektur-frameworks-ak-javaland-2023/

 ak@embarc.de
 [@alexksbr](https://twitter.com/alexksbr)
 [linkedin.com/in/alexksbr](https://www.linkedin.com/in/alexksbr)
 [@alexksbr@mastodon.online](https://mstdn.social/@alexksbr)

Folien und Links →

