

IT-SYSTEMTECHNIK · CLOUD-INFRASTRUKTUR & DEVOPS · ZERO TRUST

Hybride Multi-Cloud-Orchestrierung & Zero-Trust-Architektur

Praxisorientiertes Arbeitsblatt für Fachinformatiker Systemintegration, DevOps & Cloud Engineers (Service Mesh, mTLS, SLA-Verfügbarkeit & Konsistenz)

Name: _____

Datum: _____

Klasse/Fach: _____

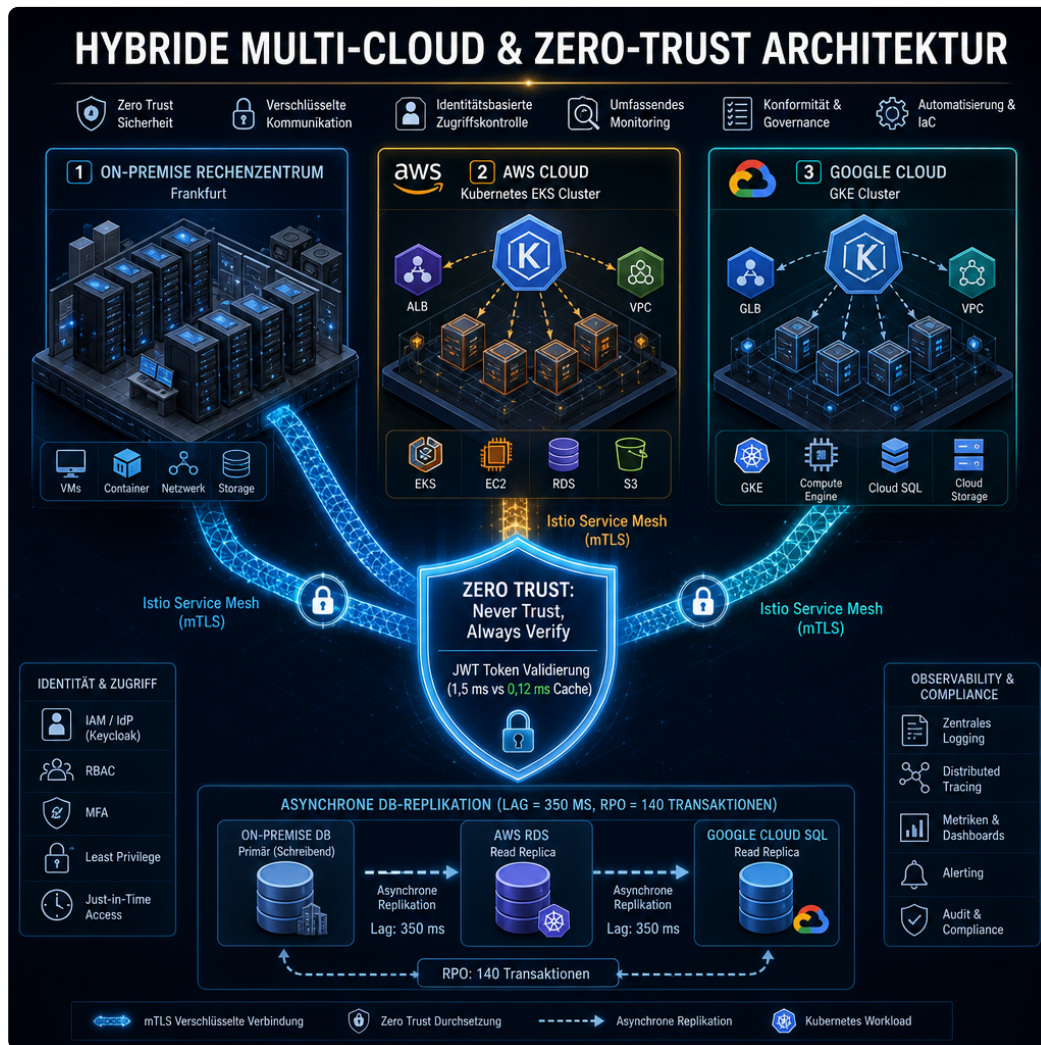


Abb. 1: Hybride Multi-Cloud Architektur – On-Premises Core RZ, AWS & Google Cloud über Istio Service Mesh, mTLS & Zero-Trust Policy Engine

Praxisszenario: „GlobalRetail Multi-Cloud Plattform“

Eine unternehmenskritische Handelsplattform läuft verteilt über drei Standorte: **On-Premise Rechenzentrum** (Frankfurt), **AWS eu-central-1** (EKS) und **Google Cloud europe-west3** (GKE). Alle Microservices kommunizieren verschlüsselt über ein Istio Service Mesh mit Envoy-Sidecars und Zero-Trust-Richtlinien.

- **Tägliche Anfragen:** 5.000.000 API-Calls · 4 Hops je Request
- **Transaktionsrate:** 400 Schreibtransaktionen/s (OLTP)

- **Krypto-Prüfung:** 1,5 ms reine CPU-Zeit je Token-Validierung
- **Replikations-Lag WAN:** 350 ms asynchron (On-Premise → Cloud)

Zero-Trust-Grundsatz (NIST SP 800-207): „Never trust, always verify.“

Kein Vertrauen auf Netzwerkgrenzen – jeder interne Hop erfordert strikte Identitäts- & Rechteprüfung.

mTLS + JWT

Aufgabe 1 Zero-Trust Token-Validierung, Krypto-Overhead & CPU-Laufzeit

4 Punkte

Unter Zero Trust wird jeder interne Serviceaufruf per mTLS verschlüsselt und per JWT autorisiert (1,5 ms CPU-Zeit je Prüfung; 4 Hops je Request).

- Berechne den zusätzlichen Latenz-Overhead pro Endnutzer-Request ohne Caching.
- Wie viele Stunden reine CPU-Rechenzeit fallen im gesamten Cluster täglich für 5.000.000 Requests an?
- Ein Envoy-Token-Cache erzielt 95 % Trefferquote (Lookup: 0,05 ms; 5 % Miss à 1,5 ms). Berechne die neue durchschnittliche Prüfzeit je Hop.

Rechenweg Aufgabe 1: Hop-Multiplikation, Cluster-Tageslast (Sekunden → Stunden) & gewichteter Erwartungswert

Formeln: $t_{lat} = n_{hops} \cdot t_{hop}$ | $t_{ges} = N_{req} \cdot n_{hops} \cdot t_{hop}$ | $t_{avg} = p_{hit} \cdot t_{hit} + p_{miss} \cdot t_{miss}$

Ergebnis: Overhead = _____ ms | CPU = _____ h/Tag | Ø Hop = _____ ms

Aufgabe 2 Multi-Cloud Hochverfügbarkeit (SLA) & Ausfallzeit-Budget

4 Punkte

Ein Monat wird mit 30 Tagen (= 43.200 Minuten = 2.592.000 Sekunden) kalkuliert.

a) Provider A garantiert 99,9 % SLA („Drei Neunen“). Berechne das monatliche Ausfallbudget in Minuten.

b) Zur Absicherung laufen AWS und GCP parallel im Active-Active-Verbund (je A = 99,9 %; Ausfallwahrscheinlichkeit $p = 0,001$). Das Gesamtsystem fällt nur aus, wenn *beide* zeitgleich versagen. Berechne Gesamtverfügbarkeit A_{ges} und das Ausfallbudget in Sekunden/Monat.

Rechenweg Aufgabe 2: Downtime-Budget $D = T \cdot (1 - A)$ & Parallele Ausfallwahrscheinlichkeit $p_{ges} = p_1 \cdot p_2$

Formeln: $D_{min} = 43.200 \cdot (1 - A)$ | $A_{ges} = 1 - (1 - A_1) \cdot (1 - A_2)$

Ergebnis: Ausfallzeit A = _____ min | A_{ges} = _____ % | Budget = _____ s

Aufgabe 3 State Management, Replikations-Lag & RPO/RTO-Analyse

4 Punkte

Die Kundendatenbank repliziert asynchron von Frankfurt in die AWS-Cloud (Replikations-Lag: 350 ms = 0,35 s; Last: 400 Schreibtransaktionen/s).

- Wie viele Transaktionen befinden sich maximal im Flug und gehen bei einem abrupten RZ-Ausfall verloren (RPO in Transaktionen)?
- Erkläre präzise den Unterschied zwischen **RPO** und **RTO**. Welcher wesentliche Nachteil entsteht, wenn RPO = 0 über WAN erzwungen wird?

Rechenweg Aufgabe 3: Transaktionen im Flug = Last · Lag & Fachliche Definition von RPO vs. RTO und Latenzkompromiss

Formeln: $N_{lost} = Rate \cdot Lag$ | $RPO = \text{Datenstand}$ | $RTO = \text{Wiederanlaufzeit}$

Ergebnis: Verlust im Flug = _____ Transaktionen | RPO / RTO = siehe Begründung

Musterlösung Aufgabe 1 Zero-Trust Token-Validierung & Krypto-Overhead

- a) **Latenz-Overhead pro Request:** $4 \text{ Hops} \times 1,5 \text{ ms} = \mathbf{6,0 \text{ ms Latenzzuwachs}}$ je Nutzeranfrage ohne Caching.
- b) **Tägliche Cluster-CPU-Rechenzeit:** $5.000.000 \text{ Requests} \times 4 \text{ Hops} = 20.000.000 \text{ Hops/Tag} \times 1,5 \text{ ms} = 30.000.000 \text{ ms} = 30.000 \text{ s} \div 3.600 \text{ s/h} = \mathbf{8,33 \text{ Stunden (8 h 20 min) reine CPU-Zeit täglich}}$.
- c) **Optimierung mit Envoy Token-Cache:** $t_{\text{avg}} = (0,95 \times 0,05 \text{ ms}) + (0,05 \times 1,5 \text{ ms}) = 0,0475 \text{ ms} + 0,075 \text{ ms} = \mathbf{0,1225 \text{ ms pro Hop}}$ (–91,8 % Overhead-Reduktion).

Musterlösung Aufgabe 2 Multi-Cloud Hochverfügbarkeit (SLA) & Redundanz

- a) **Monats-Ausfallbudget bei 99,9 % SLA:** $43.200 \text{ min} \times (1 - 0,999) = 43.200 \text{ min} \times 0,001 = \mathbf{43,2 \text{ Minuten / Monat}}$.
- b) **Active-Active Verbund (AWS + GCP):** Ausfallwahrscheinlichkeit beider Clouds: $p_{\text{fail}} = 0,001 \times 0,001 = 0,000001 \text{ (} 10^{-6} \text{)}$.
Gesamtverfügbarkeit: $A_{\text{ges}} = 1 - 0,000001 = \mathbf{99,9999 \% („Six Nines“)}$.
Neues monatliches Ausfallbudget: $2.592.000 \text{ s} \times 0,000001 = \mathbf{2,592 \text{ Sekunden / Monat}}$.

Musterlösung Aufgabe 3 State Management, Replikations-Lag & RPO/RTO

- a) **Maximaler Datenverlust (RPO im Flug):** $400 \text{ Transaktionen/s} \times 0,35 \text{ s} = \mathbf{140 \text{ Transaktionen}}$ drohen verloren zu gehen.
- b) **RPO vs. RTO & Replikation:**
 - RPO (Recovery Point Objective):** Maximal tolerierbarer Datenverlust (Zeitfenster des Datenstands; hier 350 ms bzw. 140 Transaktionen).
 - RTO (Recovery Time Objective):** Maximal tolerierbare Ausfallzeit (Dauer bis zum Wiederanlauf nach einem Crash).
 - Technischer Kompromiss bei RPO = 0:** Synchrone Replikation/Quorum (Raft/Paxos). **Nachteil:** Die WAN-Round-Trip-Time (RTT $\approx 15\text{--}30 \text{ ms}$) addiert sich auf *jeden* Schreibvorgang; Durchsatz sinkt massiv!

⚠ Typische Prüfungsfallen in der IT-Systemtechnik & Cloud-Infrastruktur:

- Verwechslung von RPO und RTO:** RPO betrifft *Daten* (Wie viele Daten dürfen verloren gehen? → Stand in Minuten/Transaktionen). RTO betrifft *Zeit* (Wann muss das System wieder laufen? → Ausfallzeit).
- Verfügbarkeits-Kombination:** In Reihe geschaltete Systeme *verringern* die Verfügbarkeit ($A = A_1 \cdot A_2$). Nur echte *redundante* Parallelsysteme verbessern die Verfügbarkeit drastisch ($p_{\text{fail}} = p_1 \cdot p_2$)!

Formelsammlung IT-Systemtechnik & Multi-Cloud-Kennzahlen:

Kennzahl / Begriff	Formel / Definition	Einheit	Praxisbedeutung & Kontext
Latenz-Overhead	$t_{\text{lat}} = n_{\text{hops}} \cdot t_{\text{prüf}}$	ms (Millisekunden)	Akkumulierter Zeitverlust durch mTLS-Handshakes & Tokens
Erwartungswert Cache	$t_{\text{avg}} = p_{\text{hit}} \cdot t_{\text{hit}} + (1 - p_{\text{hit}}) \cdot t_{\text{miss}}$	ms	Effektive Prüfzeit bei aktiviertem Envoy-Token-Cache
Ausfallzeit-Budget (SLA)	$D = T_{\text{gesamt}} \cdot (1 - A)$	min / s pro Monat	Maximal zulässige Downtime lt. Service Level Agreement
Redundante Verfügbarkeit	$A_{\text{ges}} = 1 - \prod (1 - A_i)$	% / Dezimal	Gesamtverfügbarkeit redundanter Active-Active Clouds
In-Flight Datenverlust	$N_{\text{lost}} = \text{Schreiblast} \cdot \text{Replikationslag}$	Transaktionen	Realer Datenverlust bei abruptem Primär-Cluster-Ausfall
RPO (Recovery Point)	Zeitstempel / Datenverlust-Toleranz	Sekunden / Tx	Maximal akzeptierter Datenverlust im Desasterfall
RTO (Recovery Time)	Downtime-Toleranz bis Wiederanlauf	Minuten / Stunden	Maximal akzeptierte Dauer bis zur Wiederherstellung

Unterstützung bei Cloud-Architektur & IT-Systemtechnik gesucht? Finde erfahrene IT-Dozenten auf [Nachhilfe-Vermittlung.com/s/IT-Systemtechnik](https://www.nachhilfe-vermittlung.com/s/IT-Systemtechnik)

© 2026 Nachhilfe-Vermittlung.com · Alle Rechte vorbehalten