CloudCourse GmbH

ENTERPRISE PLATFORM

Die gemeinsame Grundlage für Ihre Softwareprodukte.

CloudCourse orchestriert bewährte Komponenten zu einer Plattform, die zu Ihren Anforderungen passt: sicher, nachweisbar und schnell.

LandingZone · Plattformdienste · Entwicklerportal

Management-Überblick · Zielbild und Service Offering



Ein Fundament statt vieler Einzellösungen

Ohne gemeinsame Basis baut jedes Projekt Netz, Zugriffe und Protokollierung neu auf.

CloudCourse

Heute: jedes Projekt baut sein eigenes Fundament

Projekt A	Projekt B	Projekt C
Netz	Netz	Netz
Zugriffe	Zugriffe	Zugriffe
Protokolle	Protokolle	Protokolle
Pipeline	Pipeline	Pipeline

Dreimal dieselbe Arbeit, dreimal andere Regeln

Abweichungen fallen erst im Audit oder im Vorfall auf

Mit Plattform: einmal bauen, alle nutzen



Regeln gelten einheitlich und lassen sich belegen

1 Schneller starten

Neue Umgebungen entstehen mit fertiger Grundausstattung statt mit individueller Einrichtung je Projekt.

2 Sicher und prüfbar

Regeln werden technisch erzwungen. Jede Änderung folgt einem Weg und hinterlässt Belege für Audits.

3 Unabhängig bleiben

Fähigkeiten statt Produktbindung: Zielplattform und Komponenten bleiben austauschbar.

DER GRÖSSTE HEBEL

Alle Teams nutzen dieselbe Grundlage, und jede Verbesserung kommt bei allen an.

Vier Schichten, ein Änderungsweg

Die Landing Zone trägt die Plattform, die Plattform trägt die Produkte. Die Infrastruktur darunter ist austauschbar.

CloudCourse

Schicht 4
Produkte
Ihre Teams

Anwendungen und Connectoren der Fachbereiche

Gebaut aus Plattform-Bausteinen, entwickelt und betrieben von Ihren Teams

trägt

Schicht 3
Plattform
Gemeinsame Dienste

Plattform: gemeinsame Dienste und Bausteine

Service-Typen
Delivery, Identität, Observability, SIEM, Clearing, Daten, Messaging, E-Mail

Geprüfte Bausteine
Speicher, Laufzeit, Datenhaltung mit Owner und Support-Zusage

Secrets und PKI
Zugangsdaten, Zertifikate, Anwendungs-CA

Entwicklerportal
Servicekatalog, Owner, Dokumentation, Links zu Pipelines

trägt

Schicht 1 und 2
Landing Zone
Kontrolliertes Fundament

Landing Zone: herstellerneutral als Fähigkeiten beschrieben

Netzwerk und Sicherheit
Segmentierung der Umgebungen
Anbindung an On-Premises, Internet
Namensdienst und IP-Adressplanung
Zentrale, geschützte Protokolle

Governance
Organisation und Leitplanken
Identität und Rechte
Kosten und Verantwortlichkeiten
Änderungsweg und Nachweise

Basis
Infrastruktur
Austauschbares Profil

Infrastrukturprofil
Public Cloud, Private Cloud oder eigene Server

AWS · **Azure** · **OpenStack** · **Bare Metal**

Eine Lieferkette für alle Schichten

- Versionsverwaltung
- Merge Request und Review
- Pipeline und Prüfungen
- Evidenz und Freigabe
- Rollout in Produktion

Einziger regulärer Weg für Änderungen an Fundament und Plattform

Die Landing Zone: Fähigkeiten statt Cloud-Produkt

CloudCourse

Dieselben zehn Fähigkeiten lassen sich auf Public Cloud, Private Cloud und eigenen Servern abbilden.

FÄHIGKEITEN: stabil und herstellerneutral

- Mandantenhierarchie
- Leitplanken
- Identität und Zugriff
- Zentrale Protokolle
- Netz-Hub
- Namensdienst
- IP-Adressplanung
- Umgebungsanlage
- Kosten und Zuordnung
- Schlüsselverwaltung

Jede Fähigkeit beschreibt Ziel und Nachweis, nicht den Dienst

ADAPTER-MODULE: je Zielplattform eine Umsetzung hinter derselben Schnittstelle

Infrastruktur als Code · gleiche Lieferkette · gleiche Prüfungen auf jeder Zielplattform

AWS

Public Cloud
Verwaltete Fundament-Dienste

Azure

Public Cloud
Verwaltete Fundament-Dienste

OpenStack

Private Cloud
Mehr Eigenbau-Anteil

Bare Metal

Eigene Server
Mehr Eigenbau-Anteil

Wichtig für die Wahl der Zielplattform:

Wo verwaltete Fundament-Dienste fehlen, bleibt die Fähigkeit gleich, doch Eigenbau-Anteil und Betriebsverantwortung steigen.

Neun Service-Typen, Komponenten bleiben austauschbar

CloudCourse

Die Plattform bietet Dienste als Typ an. Welche bewährte Komponente dahinter steht, ist eine Entscheidung je Einsatz.

Delivery und Quellcode

Einziger regulärer Weg für Änderungen, Review und Auslieferung

Identität und Anmeldung

Zentrale Anmeldung für Entwickler und Dienste

Secrets und Zertifikate

Passwörter, interne Zertifikate und eine eigene CA für Anwendungen

Observability

Metriken und Dashboards über alle Dienste

Security Monitoring (SIEM)

Erkennung und Nachverfolgung von Sicherheitsereignissen

Software Clearing

Pakete und Images werden geprüft, bevor Teams sie nutzen

Datenplattform

Gemeinsame Basis für Analyse und Datenverarbeitung

Streaming und Messaging

Ereignisse und Nachrichten zwischen Anwendungen

E-Mail-Versand

Zentral kontrollierter Mailversand für alle Anwendungen

Geprüfte Bausteine (Opinionated Components)

Speicher · Container-Laufzeit · Datenhaltung (Datenbanken, Caches, Queues, Dateispeicher). Jeder Baustein ist versioniert und hat einen Owner und einen Support-Weg.

Auswahl je Kunde

Das Angebot ist der Service-Typ. Die konkrete Komponente wählen wir passend zu Ihnen.

Ihre Plattform: aus bewährten Komponenten orchestriert

CloudCourse liefert keine Standardlösung, sondern wählt und verbindet Komponenten passend zu Ihren Anforderungen.



Wiederverwendbar
Components und gemeinsame technische Fähigkeiten

Nachvollziehbar
Versionen, Prüfungen, Freigaben und Zuständigkeiten

Integrierbar
Ihre Infrastruktur, Ihre Systeme, passende Werkzeuge

SSDLC: vom Entwurf bis zum überwachten Betrieb

Der Secure Software Development Lifecycle in acht Stufen, mit einer Kontrolle in jeder Stufe.

CloudCourse

BAUEN				PRÜFEN UND FREIGEBEN			BETREIBEN
Planen	**Entwickeln**	**Review**	**Bauen & Prüfen**	**Clearing**	**Testen**	**Ausliefern**	**Betreiben**
Kontrolle Anforderung klären Bedrohungsmodell Baustein aus dem Katalog wählen Verantwortlichen benennen	**Kontrolle** Repository aus Vorlage anlegen Entwicklung in der eigenen Umgebung Secrets nur aus zentraler Verwaltung	**Kontrolle** Merge Request Vier-Augen-Prinzip nach Freigabe-matrix Änderung bleibt nachvollziehbar	**Kontrolle** Tests und statische Code-Analyse Abhängigkeits- und Secret-Scan Fehler stoppen den Lieferweg	**Kontrolle** Image-Scan, SBOM und Signatur Pakete und Images nur aus geprüften Registries Policy-Gate	**Kontrolle** API- und Event-Verträge End-to-End-Tests Dynamische Sicherheitstests Migrationschecks	**Kontrolle** Freigabe Signatur, Smoke-Test Rollback-Plan Dieselbe geprüfte Version in Produktion	**Kontrolle** Geprüfte Bausteine Metriken, Logs, Security Monitoring Alarme gehen an das Team Updates als Vorschlag

Rückkopplung: Alarme, Befunde und Versionsupdates werden neue Merge Requests

Components statt Einzelbau

Versionierte Module und Stacks setzen Architekturentscheidungen um.

Sicherheit mit Wirkung

Befunde führen zu Behebung, Freigabe oder befristeter Ausnahme.

Updates werden gepflegt

Änderungen werden automatisch vorgeschlagen; Tests und Reviews begleiten die Übernahme.

Entwickler konzentrieren sich auf Fachlichkeit. Die Sicherheit steckt im Weg, nicht in einer Checkliste.

Was das Team tut und was die Plattform übernimmt

Die Entwicklerreise vom Antrag bis zum Betrieb, aufgeteilt nach Zuständigkeit.

CloudCourse

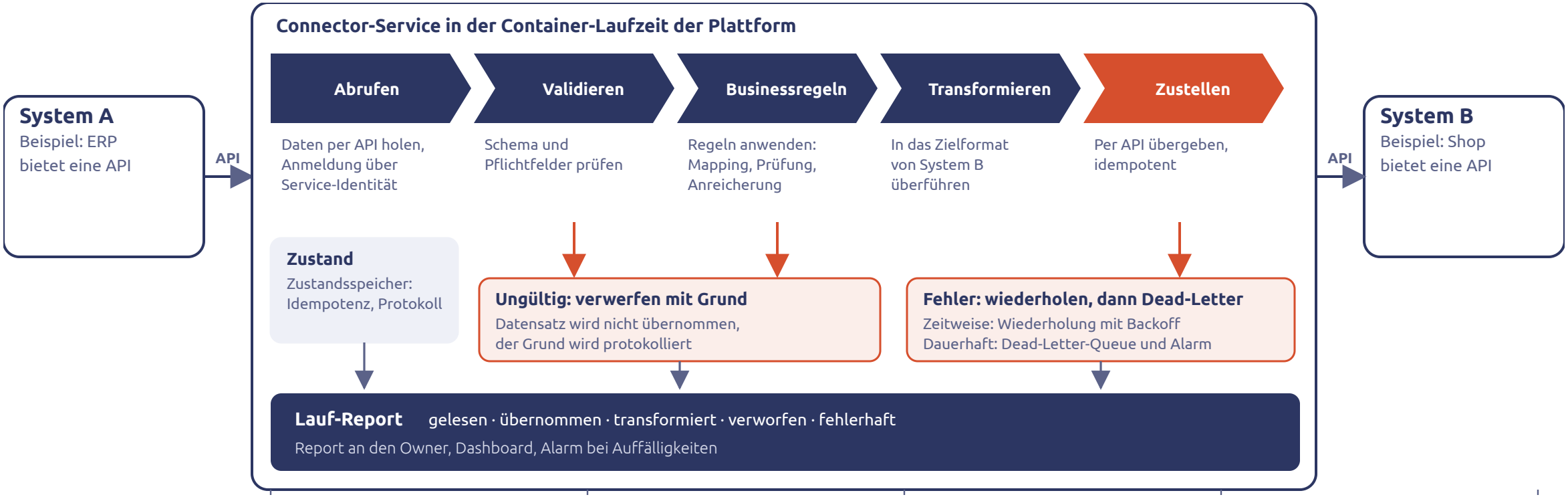
	1 Umgebung anfordern	2 Aus Bausteinen bauen	3 Änderung einreichen	4 Ausliefern	5 Betreiben
Entwicklungsteam	Antrag stellen mit Verantwortlichem und Kostenstelle	Code im Repository schreiben, Bausteine einbinden	Merge Request einreichen, Feedback einarbeiten	Freigabe zur Produktion erteilen	Alarme und Dashboards beobachten, Updates einspielen
Plattform automatisch	Umgebung mit fertiger Grundausstattung: Netz, Zugänge, Logs, Pipeline	Service-Typen und Bausteine bereit: Speicher, Laufzeit, Datenhaltung, Secrets	Pipeline prüft und erzeugt einen Plan, nutzt geprüfte Registries	Rollout über kurzlebige Berechtigungen, dieselbe geprüfte Version	Observability, SIEM, Zertifikate, Updates der Bausteine
Governance und Security	Leitplanken aktiv, Rechte nach geringstem Privileg	Nur freigegebene Bausteine mit Owner und Support-Weg	Vier-Augen-Prinzip nach Freigabematrix	Evidenz und Plan werden gespeichert	Unveränderliche Protokolle, Kosten je Verantwortlichem

Ergebnis: Ein Connector läuft entwickelt, abgesichert und betrieben.

Das Team musste weder Netz noch Zugänge noch Protokollierung selbst aufbauen.

Beispiel: Der Connector im Betrieb auf der Plattform

Ein Connector verbindet zwei Systeme per API, wendet Regeln an, verwirft Fehler und berichtet.



GENUTZTE PLATTFORMDIENSTE

Identität	Secrets	Warteschlange	Datenhaltung	Observability
Service-Identität	Zugangsdaten zu A und B	Wiederholung und Dead-Letter	Zustand und Protokoll	Metriken, Logs, Alarme

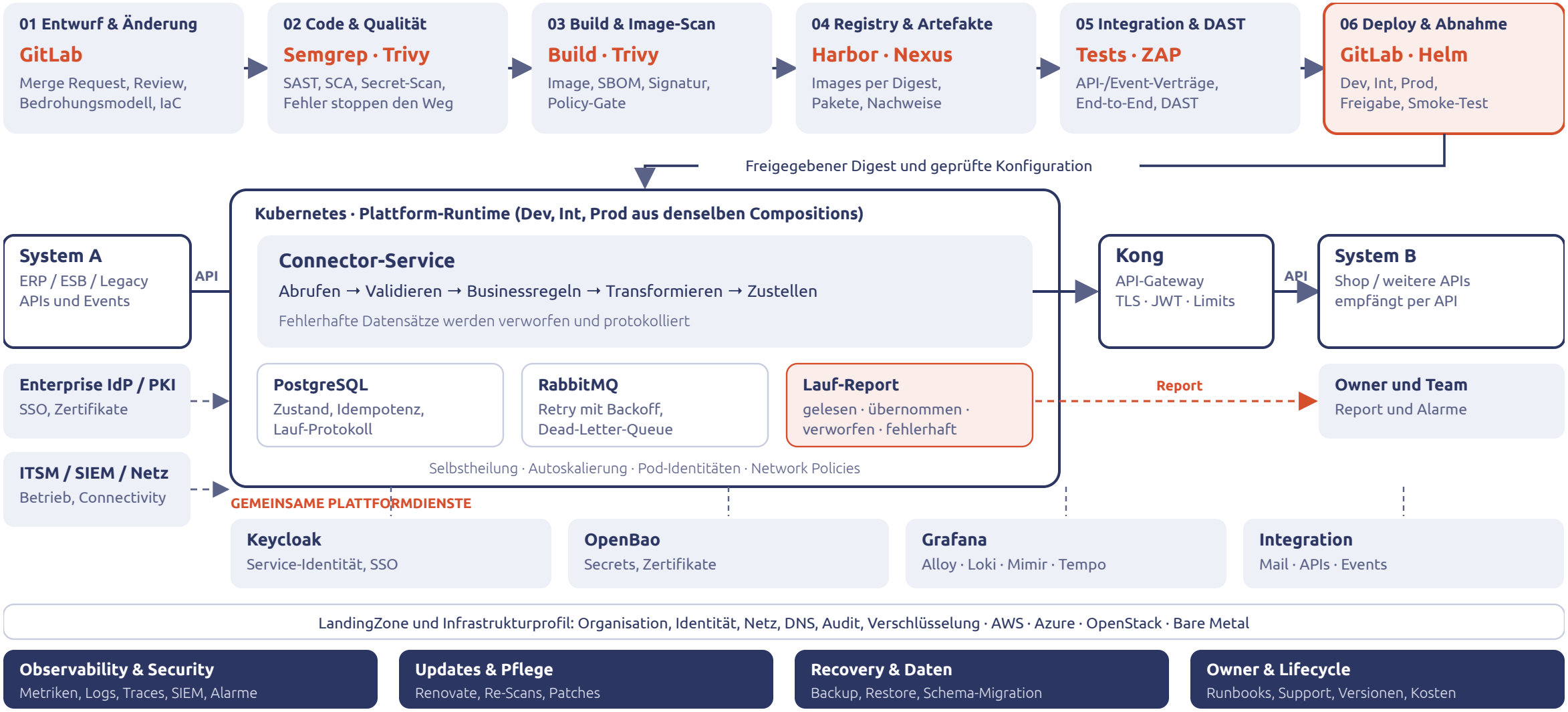
Selbstheilung
Probes und Controller ersetzen ausgefallene Instanzen.

Autoskalierung
Nach Bedarf, Quotas und Kapazität setzen Grenzen.

Security First
Pod-Identitäten, Network Policies, Stateless und 12-Factor.

Der Weg einer Anwendung durch den Toolstack

Vom Commit bis zum überwachten Connector: Lieferweg, Plattform und Umsysteme im Zusammenspiel.



Governance: Die Regel steckt in der Umgebung

Sechs Kontrollziele, jeweils mit einem Nachweis, und ein einheitlicher Änderungsweg.

Nur freigegebene Änderungen

Infrastruktur und Pipelines liegen als Code vor. Änderungen laufen nur per Merge Request.

Nachweis: Versionshistorie, Review-Protokoll

Begrenzte Rechte

Geringstes Privileg, kurzlebige Berechtigungen statt dauerhafter Schlüssel.

Nachweis: Rollenliste, Anmelde-Protokolle

Verbindliche Leitplanken

Organisationsweite Richtlinien begrenzen, was ein Konto technisch darf.

Nachweis: Richtlinien als Code

Unveränderliche Protokolle

Audit- und Netzprotokolle landen in einem abgeschotteten Archiv. Teams können sie nicht löschen.

Nachweis: Zentrales Archiv

Reproduzierbarer Stand

Code, Pipeline und Konfiguration bilden eine Version. Verbraucher halten ihre Version fest.

Nachweis: Versions-Pins je Umgebung

Verantwortung und Kosten

Verantwortlicher und Kostenstelle sind Pflichtangaben je Umgebung.

Nachweis: Kennzeichnung je Umgebung

DER WEG EINER ÄNDERUNG



Für das Fundament selbst gelten strengere Regeln als für Produkte: Wer seine Pipelines ändern kann, steuert alle Umgebungen.

Betrieb und Weiterentwicklung sind Teil des Angebots

CloudCourse

Zuständigkeiten, Schnittstellen und Übergaben werden gemeinsam festgelegt.

CloudCourse / Plattformteam
Architektur und Components
Gemeinsame Integrationen
Vereinbarter Plattformbetrieb

Unternehmens-IT
Identity und Unternehmensnetz
Bestehende Fachsysteme
Übergreifende Vorgaben

Produktteams
Fachliche Software und Daten
Anwendungsverhalten
Endkundennutzen

Beobachten
Signale, Alarme, Servicekontext

→

Verbessern
Ursachen, Components, Versionen

→

Prüfen
Verträge, Integration, Recovery

→

Ausrollen
Review, Freigabe, Migration

Zusagen für Components

Kompatibilität
Semantic Versioning. Ein Bruch außerhalb einer Hauptversion ist ein Fehler des Bausteins.

Pflege
Aktuelle Version je Hauptstrang. Vorheriger Strang erhält Sicherheitskorrekturen.

Abkündigung
Technisch gekennzeichnet, feste Frist, Nachfolger oder Migrationspfad benannt.

Rückweg
Jede Version bleibt abrufbar. Rollback setzt nur die Version zurück.

Verantwortung
Owner und Support-Weg je Component. Ohne Owner keine Veröffentlichung.

Fristen für Sicherheitspflege und Ankündigung legen wir gemeinsam mit Ihnen fest. Support, Wartung, Wiederherstellung, Kosten und Kapazität erhalten einen klaren Leistungszuschnitt.

Nutzen belegen, Start vorbereiten

Fünf Phasen, messbare Ziele und fünf Entscheidungen für den Einstieg.

CloudCourse

EINFÜHRUNG IN PHASEN (überlappend, keine harten Tore)

1 Fundament	2 Lieferkette	3 Bereitstellung	4 Plattformdienste	5 Produkte
Leere, kontrollierte, angebundene LandingZone	Änderungen nur über den definierten Weg	Umgebungen per Antrag mit Grundausstattung	Teams nutzen gemeinsame Dienste	Anwendungen und Connectoren laufen auf der Plattform

Nutzen messen

- **Bereitstellungszeit**
  Vom Antrag bis zur nutzbaren Umgebung
- **Wiederherstellungszeit**
  Dauer bis zur Wiederherstellung
- **Aufwand je Nachweis**
  Audit- und Compliance-Belege
- **Änderungsfehler**
  Anteil fehlgeschlagener Änderungen
- **Pflegeaufwand**
  Updates und Betrieb je Produkt
- **Kosten mit Verantwortlichem**
  Anteil zuordenbarer Kosten

Kostenblöcke

Aufbau: Fundament, Dienste, Components, Migration | **Betrieb: Plattformteam, Support, Pflege**

Verbrauch: Infrastruktur je Umgebung | **Lizenzen: kommerzielle Dienste**

Der Nutzen liegt in vermiedenem Aufwand und Risiko. Wir beziffern ihn gemeinsam mit Ihren Ausgangswerten.

Fünf Entscheidungen für den Start

1. Pilotvorhaben und Abnehmer benennen
2. Plattformprofil für den Start festlegen
3. Support-Zusagen und Fristen festlegen
4. Ausgangswerte für den Business Case erfassen
5. Kostenmodell und Verrechnung abstimmen

Vorhaben besprechen

cloudcourse.de/kontakt

Lassen Sie uns über Ihr Vorhaben sprechen.

CloudCourse GmbH
Heikendorfer Weg 57
24149 Kiel

Telefon: +49 431 24 85 31 08
E-Mail: info@cloudcourse.de
cloudcourse.de/kontakt

Vorhaben besprechen

Impressum

ANGABEN GEMÄSS § 5 TMG
CloudCourse GmbH
Heikendorfer Weg 57, 24149 Kiel

REGISTERGERICHT
Amtsgericht Kiel, HRB 23573 KI

UMSATZSTEUER-ID
DE349562991

VERTRETUNGSBERECHTIGT
Andre Lohmann, Geschäftsführer

Vollständiges Impressum und Datenschutzerklärung:
cloudcourse.de/legal/impressum
cloudcourse.de/legal/datenschutz

© 2026 CloudCourse GmbH. Alle Rechte vorbehalten.