

# CLOUD NATIVE APPLICATION REFACTORING

Dank Refactoring, also eines Neuaufbaus, von bestehenden Anwendungen das Potential von Cloud Native voll ausschöpfen.



## Problem

Geschäftskritische Anwendungen sind ein fundamentaler Bestandteil der digitalen Transformation eines Unternehmens und ihre Modernisierung daher eine strategische Notwendigkeit.

Viele geschäftskritische Anwendungen können mit den sich ständig ändernden Anforderungen oft nicht mehr Schritt halten. Nicht nur die Bereitstellung neuer Funktionalitäten stellt eine Herausforderung dar, sondern auch steigende Wartungs- und Betriebskosten oder Abhängigkeiten von Altsystemen. Zusammen mit dem möglichen Verlust von zentralem Anwendungs-Know-how kann dies schnell existenzbedrohend werden.

## Hauptmerkmale

- **Architekturworkshops** – Zustandsanalyse & Zieldefinition.
- **Codereviews**
- **Zerlegung in Microservices** – Mittels Microservices spezifische Geschäftsfunktionen abbilden, unabhängig von anderen Services entwickeln, bereitstellen und skalieren.
- **Datenmanagement anpassen** – Datenhoheit gewährleisten und Datenabhängigkeit zwischen Services vermeiden.
- **API-Definition und -Aufbau** – Sicherstellung, dass Services nahtlos miteinander interagieren können.
- **Sicherheit & Compliance** – Gewährleistung der Sicherheit entlang der gesamten Software-Lieferkette.

## Lösung

Der Übergang von einer monolithischen zu einer Cloud Native Architektur ist ein komplexer Prozess, der einer sorgfältigen Planung und schrittweisen Umsetzung bedarf. Folgende Strategien und Vorgehensweisen können dabei zielführend sein:

- **Strangler-Fig Pattern** - schrittweise Ersetzung spezifischer Funktionen eines Monolithen durch neue Microservices.
- **Dekomposition nach Geschäftsfunktion** - Zerlegung eines Monolithen in kleine, unabhängige Teile auf Basis von Geschäftsfunktionen oder Diensten.
- **Datenbank-Refactoring** – Überarbeitung der Datenhaltung.
- **API-First** - Sicherstellung, dass Dienste nahtlos miteinander interagieren können.

## Vorteile

- **Erhöhte Flexibilität und Agilität** – Stärkung der eigenen Innovationskraft und Verkürzung der Markteinführungszeiten sowie der Einarbeitungszeiten für neue Mitarbeiter.
- **Kostenoptimierung und Umweltfreundlichkeit** – Cloud-Ressourcen werden effizienter genutzt, was zu Kosteneinsparungen und einer geringeren Umweltbelastung führen kann.
- **Bessere Sicherheit** – Ermöglicht eine präzise Sicherheitskontrolle auf der Ebene einzelner Dienste, und im Falle eines Vorfalls ist der Radius der Auswirkungen kleiner und ermöglicht eine schnellere Reaktion.
- **Flexibel skalieren** – Einzelne Komponenten oder Dienste können sowohl vertikal als auch horizontal skaliert werden.
- **Ausfallzeiten minimieren** - Kleinere, unabhängige Microservices sind einfacher zu warten und zu aktualisieren als monolithische Anwendungen.



**Mathias Weiss**  
Software Architect  
mathias.weiss@cloud37.io