

CLOUD NATIVE APPLICATION REFACTORING

Thanks to refactoring, i.e. rebuilding existing applications, you can fully exploit the potential of Cloud Native.



Problem

Business-critical applications are a fundamental part of a company's digital transformation, and their modernisation is therefore a strategic necessity.

Many business-critical applications are often no longer able to keep pace with constantly changing requirements. Not only does the provision of new functionalities pose a challenge, but also rising maintenance and operating costs or dependencies on legacy systems. Together with the possible loss of central application expertise, this can quickly threaten the existence of the company.

Key features

- Architecture workshops – status analysis & target definition.
- Code reviews.
- Decomposition into microservices – use microservices to map specific business functions, develop, provide and scale independently of other services.
- Customise data management – ensure data sovereignty and avoid data dependency between services.
- API definition and structure – ensure that services can interact seamlessly with each other.
- Security & compliance – ensuring security along the entire software supply chain.

Solution

The transition from a monolithic to a cloud-native architecture is a complex process that requires careful planning and step-by-step implementation. The following strategies and approaches can be helpful:

- Strangler-fig pattern – gradual replacement of specific functions of a monolith with new microservices.
- Decomposition by business function – breaking down a monolith into small, independent parts based on business functions or services.
- Database refactoring – revision of data storage.
- API-first – ensuring that services can interact seamlessly with each other.

Benefits

- Increased flexibility and agility – strengthening your own innovative power and shortening time-to-market and familiarisation times for new employees.
- Cost optimisation and environmental friendliness – cloud resources are used more efficiently, which can lead to cost savings and a lower environmental impact.
- Better security – enables precise security control at the level of individual services, and in the event of an incident, the radius of impact is smaller and allows for a faster response.
- Scale flexibly – individual components or services can be scaled both vertically and horizontally.
- Minimise downtime – smaller, independent microservices are easier to maintain and update than monolithic applications.



Mathias Weiss
Software Architect
mathias.weiss@cloud37.io