FROM MAINFRAME BLACKBOX TO OPEN CLOUD ARCHITECTURE

IT systems that have grown over decades and form the backbone of a company eventually become a risk: the underlying technology is barely scalable, the code is no longer maintainable, and the system evolves into a blackbox.

This was exactly our starting situation: outdated infrastructure and software were slowing down innovation, making changes expensive and leading to increasingly frequent errors. The central question was: How can we prepare for the future while maintaining the trust of our customers?

CHALLENGE

The company faced the task of modernising its IT infrastructure without jeopardising business continuity.

This meant gradually transitioning the COBOL codebase and existing software architecture — which limited scalability and made new developments expensive and error-prone — into modern, cloud-based software development.

APPROACH

Proven methods combined with modern technology to overcome key obstacles.

We combined proven methods with cutting-edge technology — not just to solve technical problems, but to master key challenges such as availability, scalability, time-to-market, and knowledge building around existing legacy stacks, paving the way for future innovations.

IMPLEMENTATION

1 MODERN CODEBASE

By switching from COBOL and mainframe technology to a modern Java-based codebase, the system's maintainability was significantly improved, creating a future-proof foundation for further development. The migration was carried out step by step, ensuring ongoing operations at all times.

2 MICROSERVICE ARCHITECTURE

The transition to a domain-driven microservice architecture enabled better scaling of components and their independent evolution. This allows new features to be implemented faster and changes to be rolled out with lower risk, significantly increasing the agility and maintainability of the overall system.

3 MIGRATION WITH OPENSHIFT

The migration to OpenShift created the foundation for a microservice architecture by providing a containerised, orchestrated environment for operating individual services. OpenShift enables automated and standardised deployments, ensures efficient resource utilisation, and increases the resilience of the entire platform.

4 GITOPS GITHUB ACTIONS

Using GitHub Actions, we implemented GitOps principles and automated build, test, and deployment processes from the repository. This ensured that changes to infrastructure and applications were versioned and traceably documented, guaranteeing secure and consistent delivery of microservices. This shortened development cycles while simultaneously reducing potential sources of error.

5 KAFKA INTEGRATION

The introduction of Kafka enabled asynchronous processing of requests. This significantly improved both the fault tolerance and performance of the overall system. Kafka also served as a buffer for batch jobs that were processed in a time-controlled and managed manner.

RESULTS

HIGHER SCALABILITY

The microservice architecture strengthened the system at critical points and prepared it for future requirements.

RELIABILITY

Kafka increased availability and data integrity.

IMPROVED MAINTAINABILITY

The codebase modernisation led to shorter development cycles and lower operating costs.

INCREASED EFFICIENCY

The established CI/CD processes with integrated quality assurance led to a noticeable increase in productivity and faster feature delivery.

SUSTAINABILITY

The migration from host systems to the cloud achieved sustainable cost reductions.

The IT landscape transformation follows the vision of continuously optimising the customer experience with faster, more reliable services and continuous feature releases. The project is a success story that demonstrates how a strategic digitalisation project with technical excellence can simultaneously increase availability and innovation while reducing costs.

MORE INFO

www.posedio.com
sales@posedio.com
+43 660 915 44 19

POSEDIO
Do it RIGHT.