

Modernized Legacy Microservices for Rakuten's Streaming Site Viki

SUMMARY

Rakuten Viki is a global streaming site for Asian primetime shows & movies, subtitled in over 200 languages. The client's existing website had legacy microservices built on multiple technologies and with customized dependencies, making it difficult to maintain code & add new features. Gophers Lab implemented a solution for Viki, rebuilding the microservices in Go. This helped make the code clean and maintainable, allowed Viki to add more features with ease, save resources, and simplify testing.

- ✓ **Redeveloped microservices in Go for clean and maintainable code**
- ✓ **Reduced resource requirements & operational costs for new development**

ABOUT THE CUSTOMER

Rakuten Group, Inc. is the largest e-Commerce company in Japan, and the third largest e-Commerce marketplace company worldwide, with over 1.3bn registered users globally.

Founded in 2010, Rakuten Viki is its premier global entertainment streaming site for people to discover and watch primetime shows and movies, subtitled by its community of fans in more than 200 languages. The site has made Asian entertainment global, with billions of videos viewed and over one bn words translated. Viki reaches over 40mn fans in more than 190 countries.

OTT

DOMAIN

Singapore

COUNTRY

Back-end, Microservices

SERVICES

Media & Entertainment

CUSTOMER CHALLENGES

Viki's existing streaming platform was based on legacy microservices, which was causing some problems for the client:

- The legacy microservices were built on different technologies and had a lot of customized dependencies, making it difficult to maintain the code
- To add any new features to the existing site, Viki required a team of developers that knew all these technologies, along with performing extensive testing

SOLUTION IMPLEMENTED

Viki required a partner who could work on multiple technologies and make the code less complex and more maintainable. Gophers Lab helped Viki achieve its objectives by implementing the following solution:

- Built the microservices from scratch in Go and deployed them independently
- Extensive testing to ensure existing issues were resolved
- Created the API documentation for the client

TECHNOLOGY STACK

BACK-END	  
DATABASE	 PostgreSQL
CLOUD	 
DEVOPS	 docker  kubernetes
TESTING	 testify
TOOLS	 redis  RabbitMQ™

BUSINESS RESULTS

With Gophers Lab's help in migrating the legacy applications that were built in Python and Ruby to Go, Viki was able to achieve the following:

- Clean and maintainable code and microservices that are independent of each other
- Ability to add new features without the need for developers from different technologies; hence, fewer resources are required to run and lower operating costs
- Simplified unit and integration testing