

## Case: 06 Solution and Operations Reliability

A customer had this interesting business requirement...

- The back-office system needs to support frequent updates
- The back-office system needs to be available - especially between 06:00 CEST and 18:00 CEST
- A failure in one part of the back-office system shouldn't bring down the entire system
- Customer wants to re-architect the system. Does not want to bring down the entire system when doing an update.

We mapped that to technical requirements like this...

Microservices!

- Break apart the back-office system into independent services
- Create a standard way for teams to publish logs and metrics for their services
- Create a standard way for services to be rolled out

This was a natural fit for microservices. They knew that when they told development groups that they would be developing their own microservices, that they needed standards for reliability and scalability, and they wanted common ways to monitor the applications.

And this is how we implemented that technical requirement.

Google Kubernetes Engine

- Microservices deployed into a shared cluster
- Surging Rolling Deployments with GKE's deployment resource

Stackdriver

- Custom Metrics - a wrapper library around the Cloud Monitoring client libraries to:
- Expose "common" metrics
- Expose custom metrics

Solution was to use Cloud Monitoring. Exposing the metrics could be done through dashboards. Exposed metrics through prometheus standard, scraped from APIs, and sent to Cloud Monitoring where it could be exposed through dashboards.

They used custom metrics in Cloud Monitoring, so they were able to monitor and scale their microservices based on those metrics.