# Overview:

A renowned Asian Bank with a massive footprint of legacy data centers plan to transition entirely to cloud in next 5-10 years. Driven by mobile first strategy with an opportunity to rethink customer banking experience.

They decided to introduce a social payment application which allows the users to send money to their friends and family without asking for their phone number or the bank account, all for free. The customers avail bank-grade security on the app and the key highlight is 95% of the transactions terminate within 600 ms.

It also allows merchants to collect payments, with the critical differentiation to traditional methods like credit cards, it is transferred in real-time. The integration of analytics tools means there are value add services, like enabling merchants to analyze sales trends, giving them the data needed to grow their business.

# Technical Design Details

They first decided to go cloud native with a microservices architecture. A critical feature of the architecture is that each microservice is entirely self-contained and operates independently of the others. Each has its own instance copies and each makes use of its own Cloud resources, such as databases, Keys etc.

The app consists of a dozen or so microservices each performing a specific function, such as a user and business profile service, payments and reporting. There is also an API for integrating with e-commerce stores and POS units. At the times of festivities, they scale out some specific microservices that are needed such as payments but not enrolment. At the same time, the database layer also gets scaled by obtaining direct replicas of the MySQL databases.

Each microservice runs in its own subnet and communicates with GCP services like databases and caches via service principles and credentials stored in KMS. They also use VPC service endpoints, so that microservices are making calls to local endpoints within the same vnet. This is a design feature critical to security on the public Cloud.

They use a network security group with a base rule that denies all incoming and outgoing traffic, and then add specific rules that allow only the traffic needed for their services. Importantly they also automate the deployment of these rules to eliminate the risk of human error.

**Analytics:** they also use Databricks as the data platform which would stitch together all the data interactions and perform all the data engineering and data science workloads. They use a combination of ingesting data, such as transaction classifications, as well as keeping some in place such as click stream data which is kept in storage and viewed remotely.

# Problem Statement

1. Company is facing heavy challenges in managing the microservice environment. They want to move away from maintenance and spare human resources for better-aligned work. GKE, Managed DB
2. Somebody made a wrong code change years ago and they are unable to identify the same. The audit team is behind them to track the actions. VCS, Limited access
3. End users often have conflicts with app uptime and issue with the service team. While the service team says that the application was up and running most of the time, users say they faced lots of errors while working. Error SLI and SLO Dashboard at Prom and Grafana
4. During one large event customers complaint for errors and service team logged into each server and couldn’t find any error at all. We need to ensure such issues must not occur again and must be tracked. Centralize logging with STackDriver in GCP
5. While doing manual upgrades, lots of mistakes are being made by Service engineers. Management wants to reduce the same. Managed Service and use automation with ansible
6. Provisioning systems and infrastructures are usually taking lots of time and mostly it contains hundreds of errors. Terraform/Automation
7. Company is facing issues with proper communication during the incident and problem handling. While service engineer says, they have limited updates to send always, management still want to get better updates and to right owners. Even outside world (end-users) should get a proper update. Need to work on a better solution. CICD Pipelines
8. Need to improve SLA – SLO Dashboards

# ASK:

1. Depict the right architecture as per your understanding with all the components
2. Find solutions to the above problems and present the same.
3. Presentation time – 20 minutes, QnA – 10 min