# CLOUD COMPUTING WITH BIG DATA

By Group 8


**Group members**

秦尧
林志祥
吕哲坤
刘艺奇


**Group Name: SkyNet Innovator**

2024

# Project

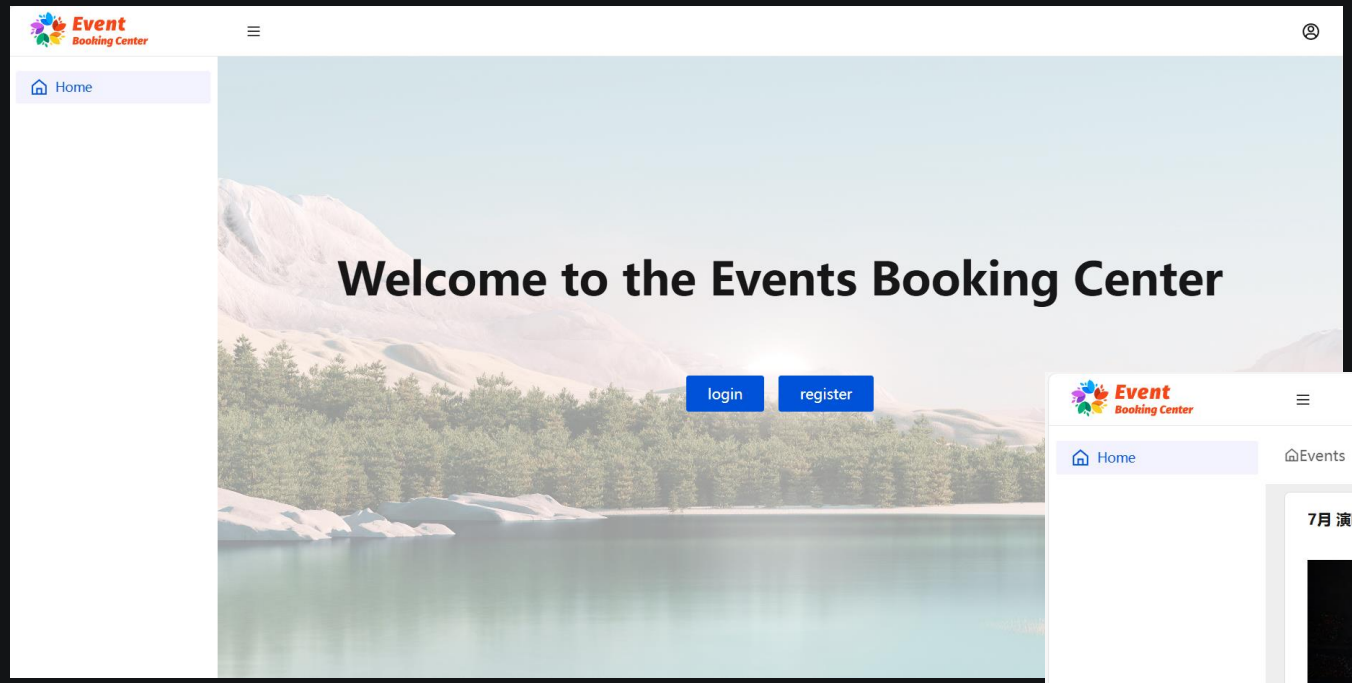
# Introduction

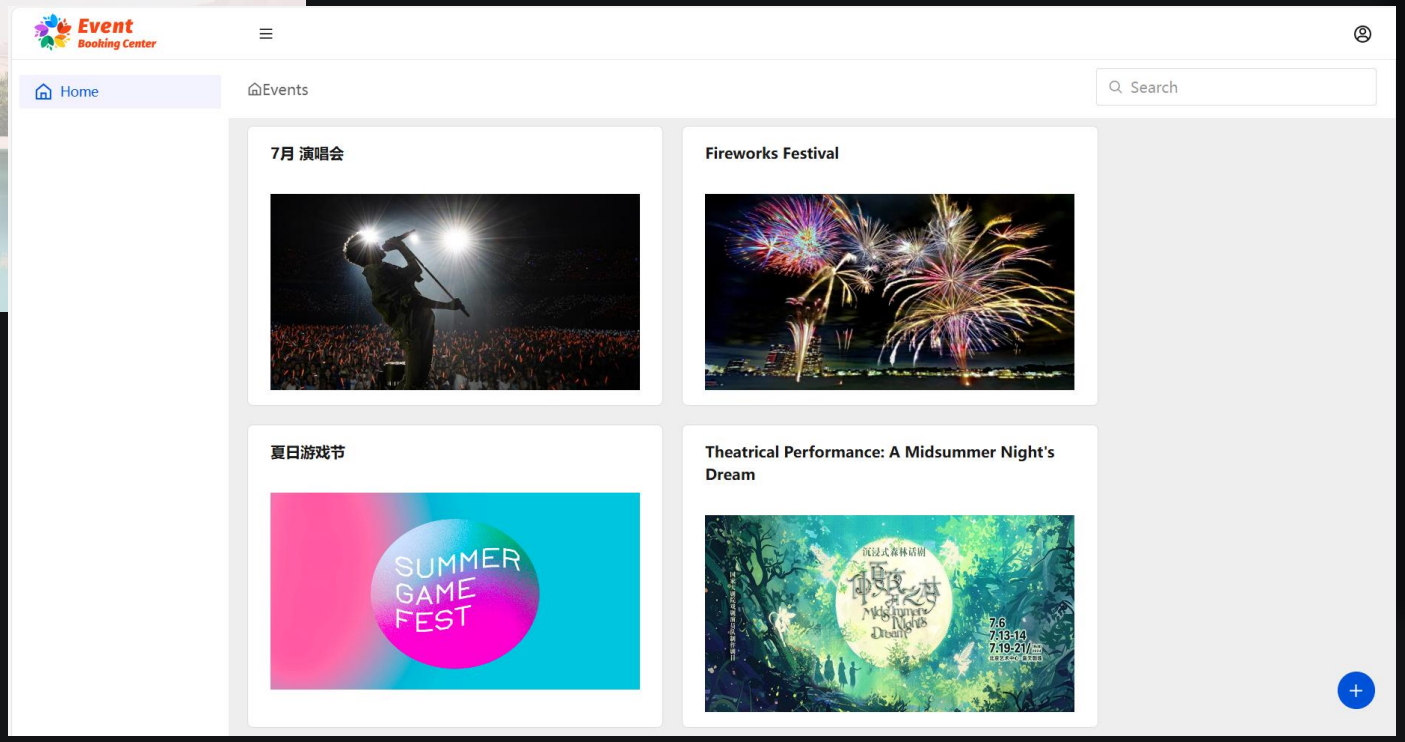
# Event Booking Center

## Everything convenient

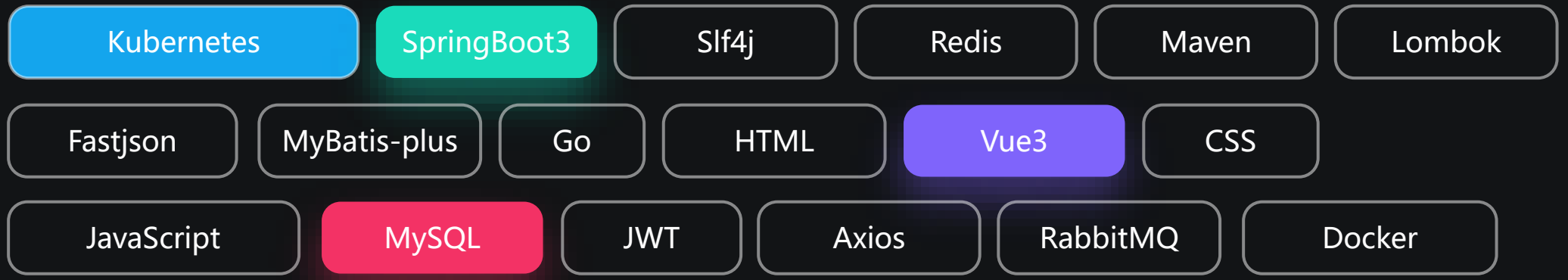
- Our event publishing and booking system allows users to easily manage activities and enables seamless scheduling of events
- Users can browse available slots, select their preferred time, and confirm their booking with just a few clicks.

# Front end display



# Main technology stacks



# Main technology stacks

## Cloud Cluster

- Kubernetes
- Docker

## Front-end Design

- Vue3
- HTML
- Axios
- CSS
- JavaScript

## Back-end Design

- SpringBoot3
- MyBatis-plus
- Maven
- Lombok
- Slf4j
- Fastjson

Beautiful UI design

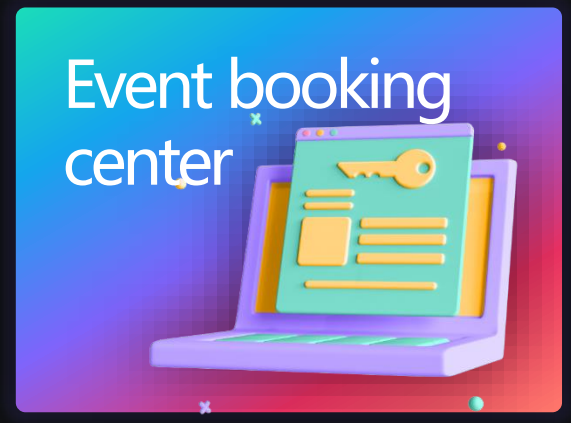
## middleware

- RabbitMQ
- Redis

## Database

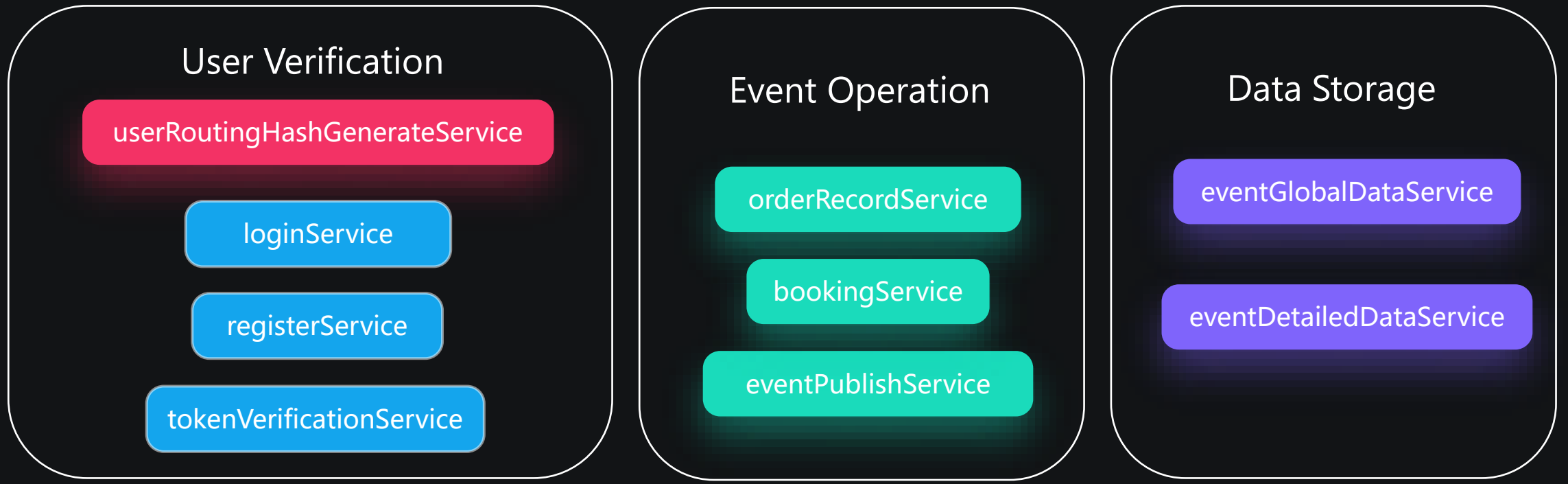
- MySQL

Distributed database cluster

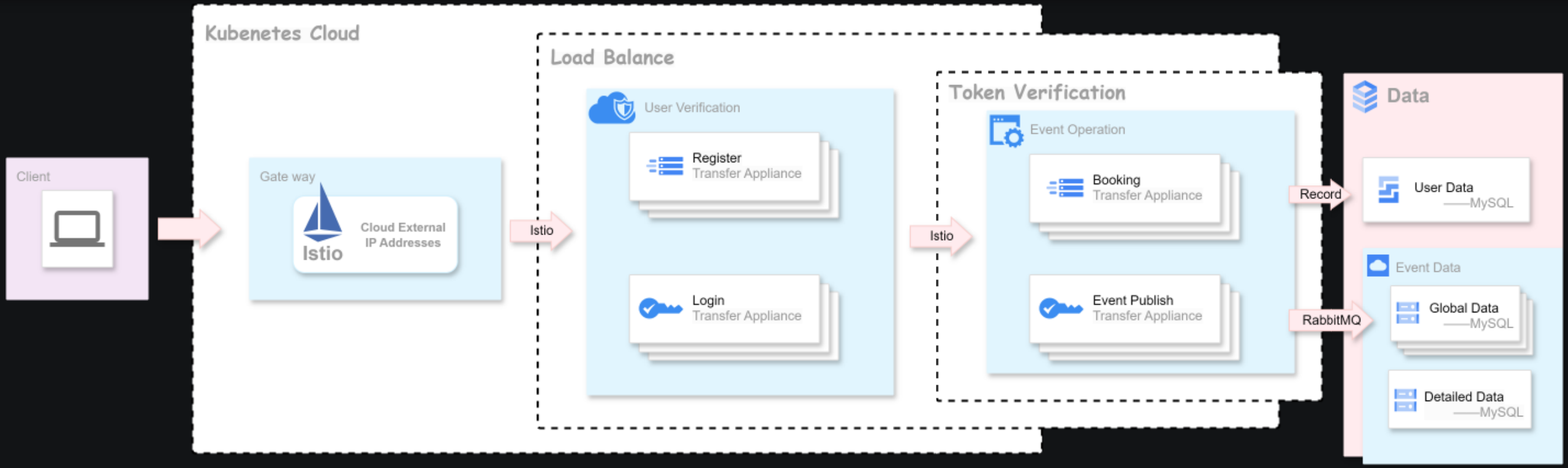


# Microservices Architecture

There are **9** Microservice including:

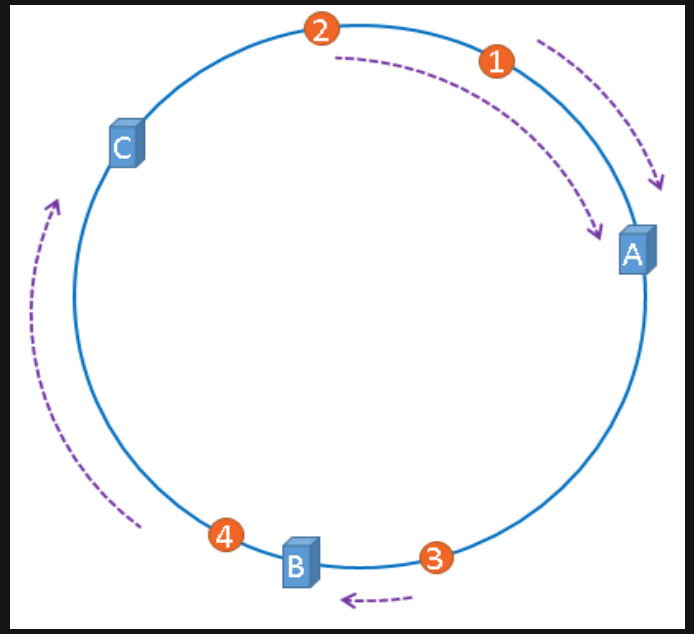


# System sturucture



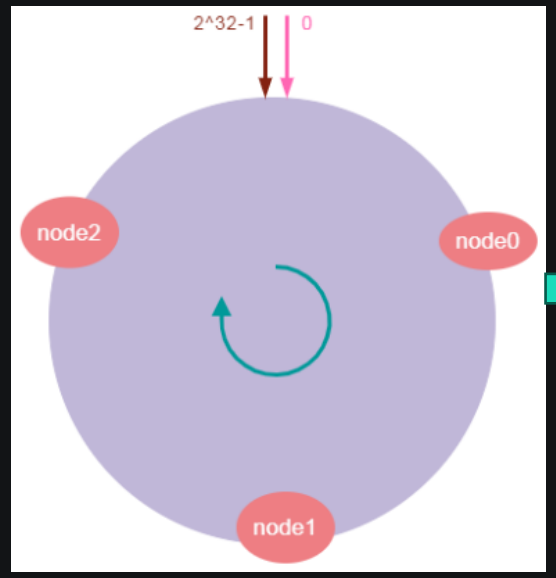
# Consistent Hashing

- We use consistent hashing .
  - Why we considered it?
- For the reason : load balancing problem

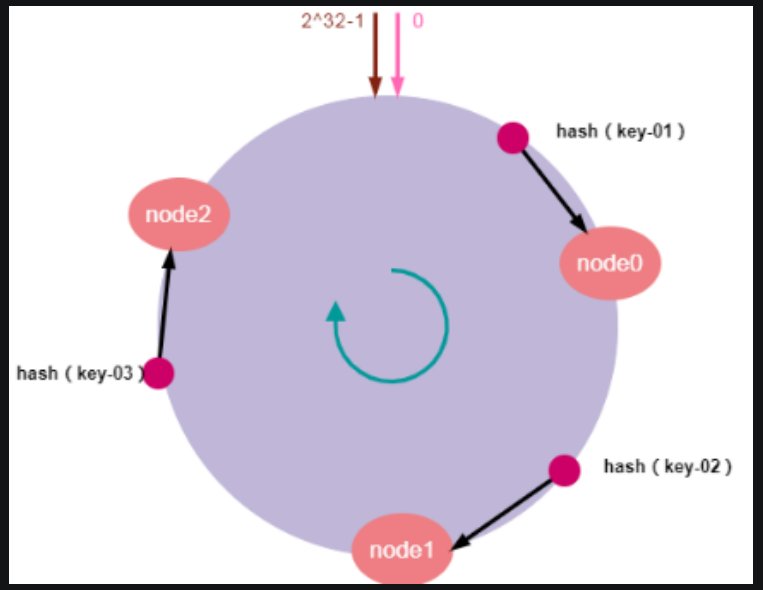


# Consistent Hashing Principle

- **e.g. 3 Deployment.**

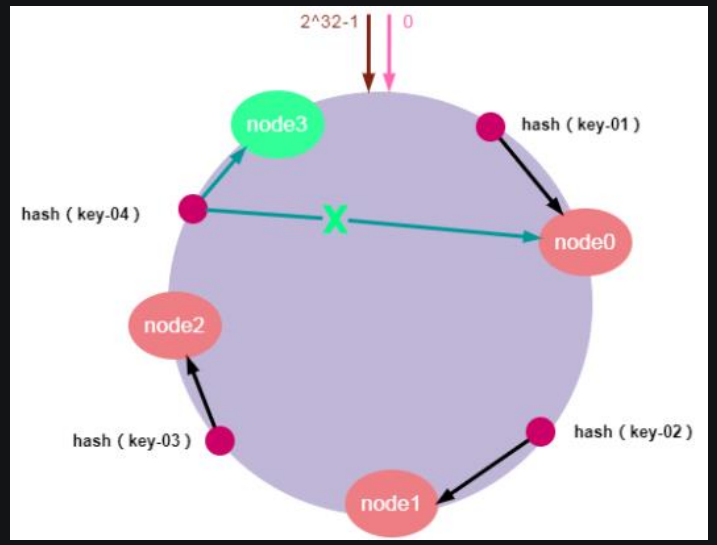



- **entire hash value space into a virtual ring**




- **Next, 3 server deployment to the corresponding positions on the hash ring**

- **Core Feature: Server Scaling**




- **Well-integrated with Kubernetes**

# Service mesh and Istio

Service mesh

- A dedicated infrastructure layer to handle service-to-service communication in a microservices architecture
- Secure, fast, and reliable

- A set of lightweight network proxies
- Sidecar pattern

Istio


One of the most popular service mesh implementations

# Ingress vs. Istio

## Ingress

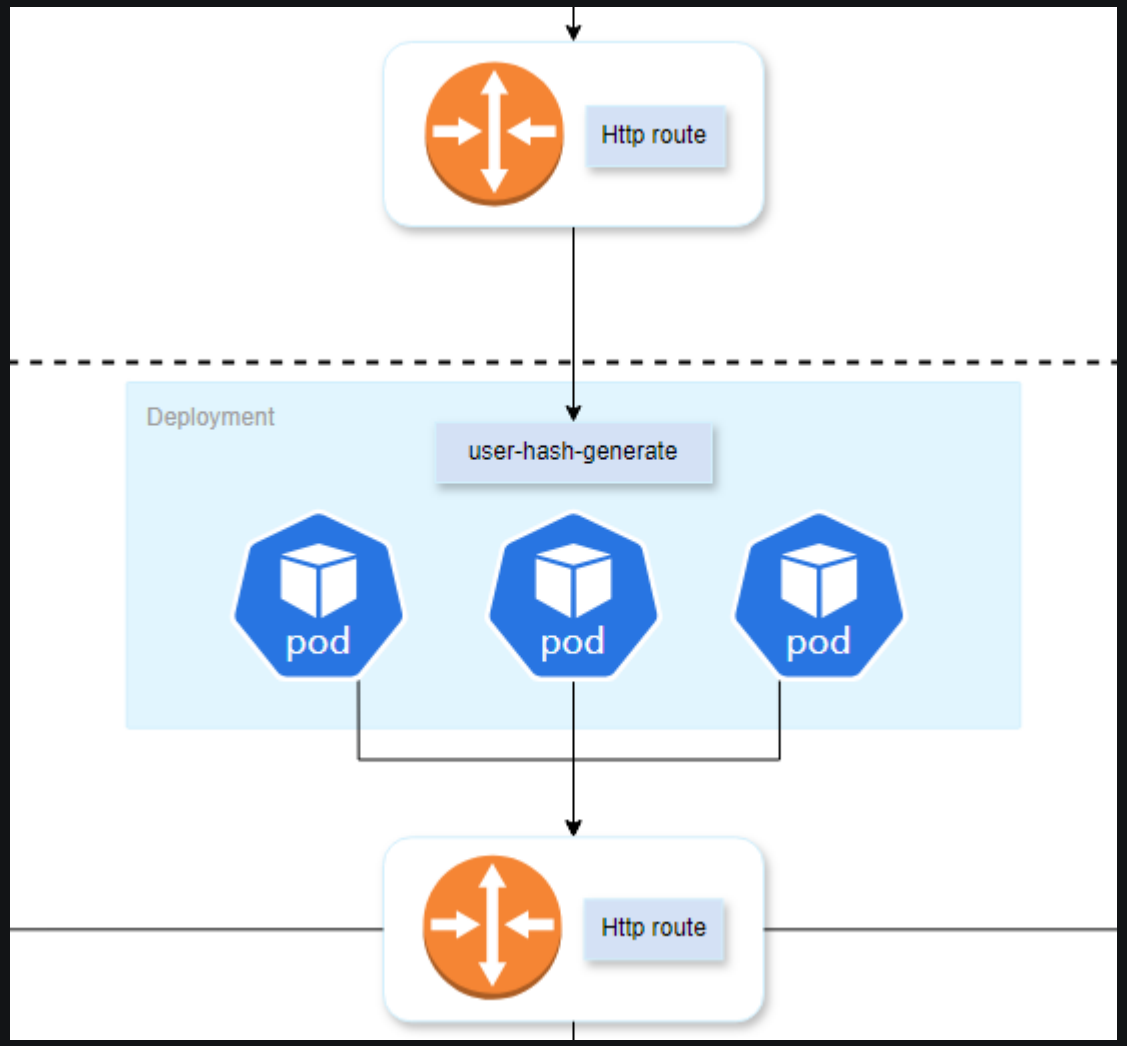
**Purpose:** Focus on routing from external networks
**Implementation:** Requires an Ingress controller (e.g., Nginx)
**Configuration:** Configured through Kubernetes resource objects (Ingress)

## Istio

**Purpose:** A comprehensive service mesh solution
**Implementation:** Injecting a Sidecar proxy (Envoy Proxy) into Pods
**Configuration:** Through a series of CRDs



# Hash generating and routing



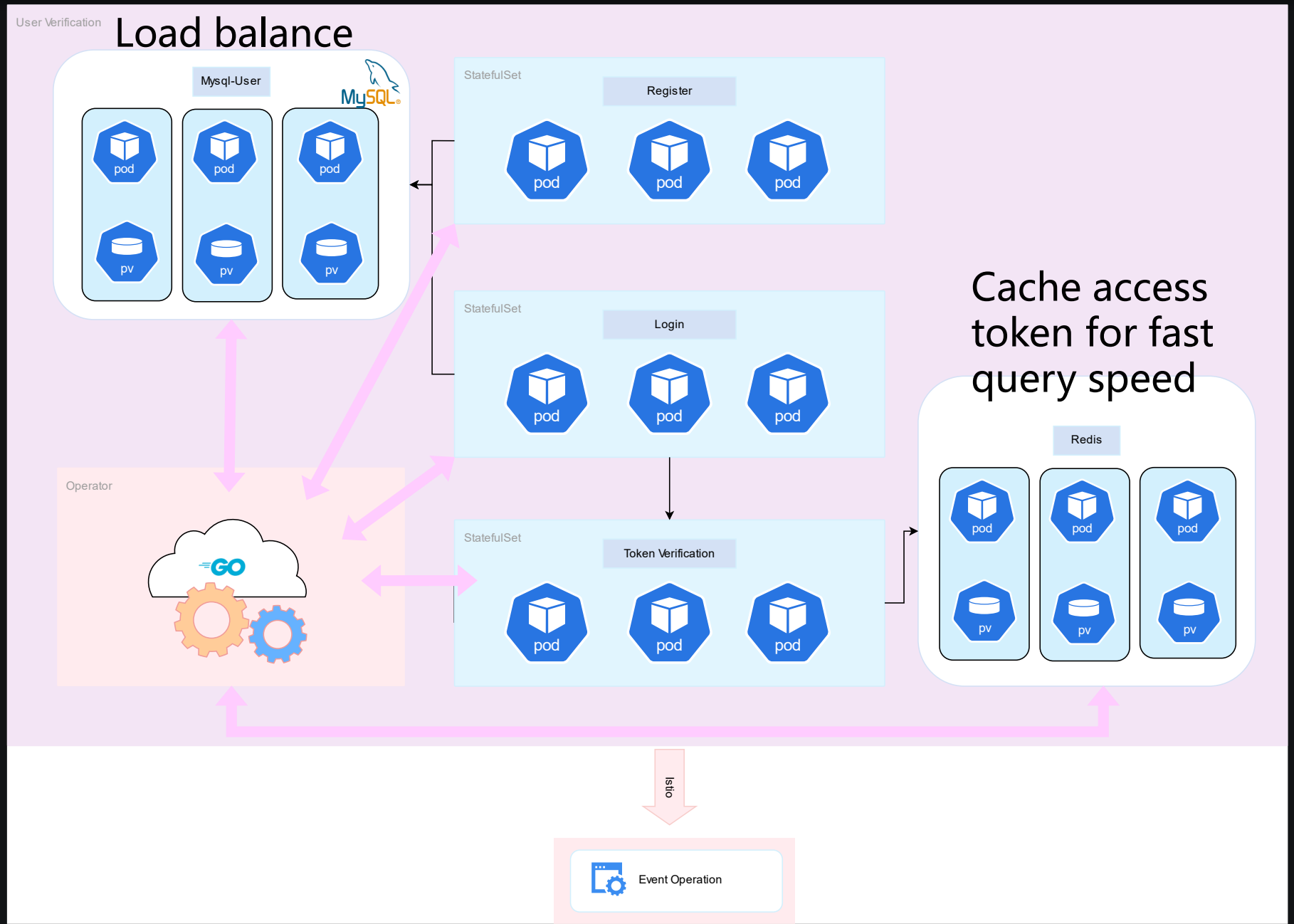

Load balancer

Generate hash based on the email address of users

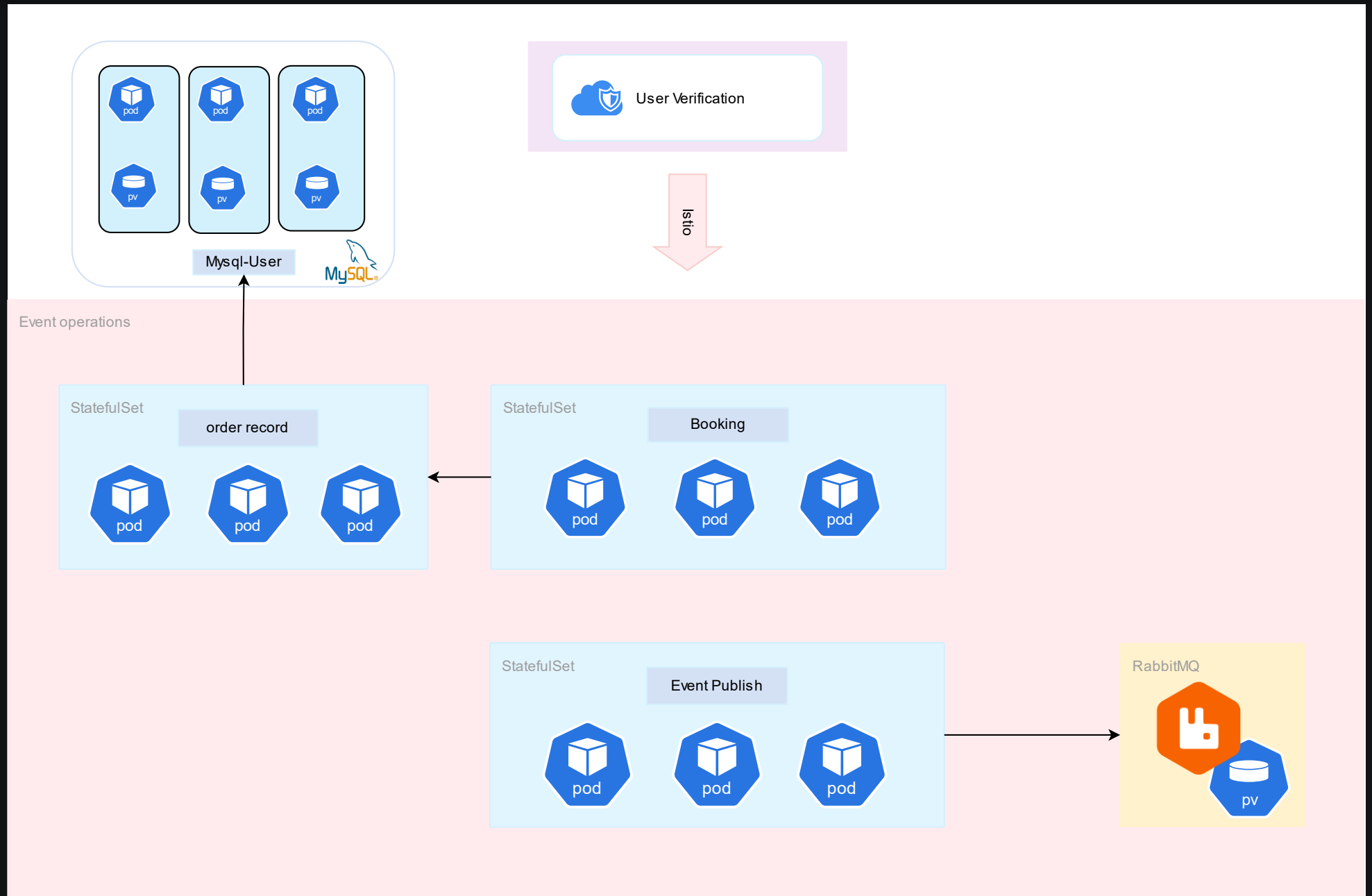
Every email address will be mapped to a hash of length 8

Routing based on hash

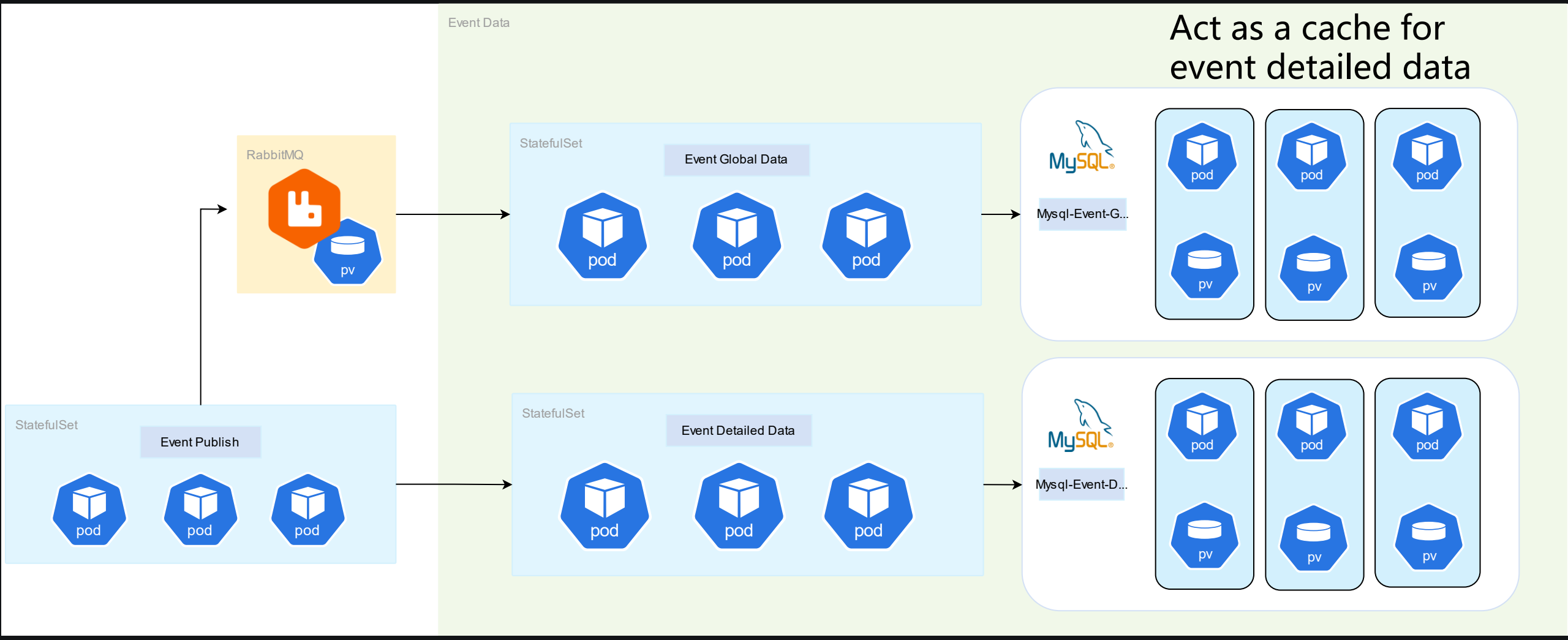
# User Verification



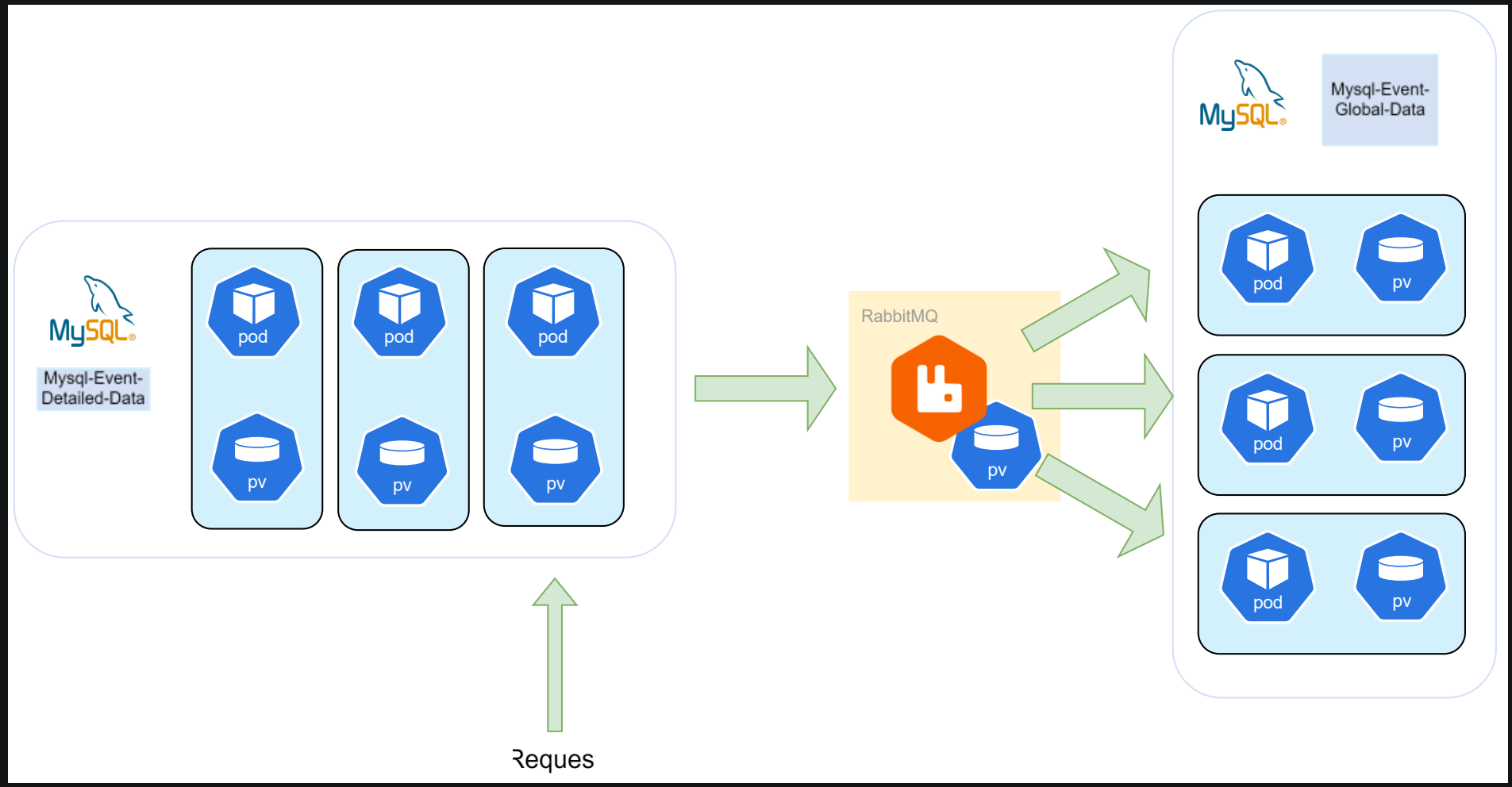
# Event Operations



# Data Storage



# Data Synchronization

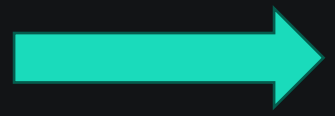


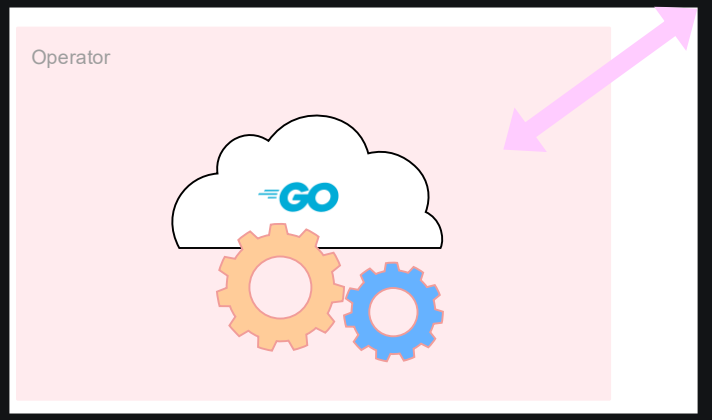
# An Operator Implement by Our Own

**Main purpose:** Enable auto-scaling of our distributed databases

```yaml
apiVersion: "auto-scaler.nus-cloud-project/v1"
kind: AutoScaler
metadata:
  namespace: nus-cloud-project
  name: test-auto-scaler
spec:
  maxTableSize: 10
```

A CRD to config the operator






Watching data in databases, split it if the table size exceeds the limit

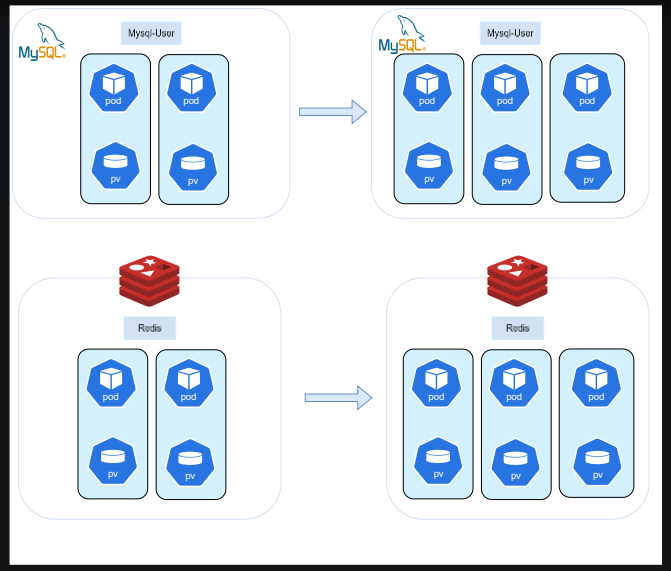
# An Operator Implement by Our Own

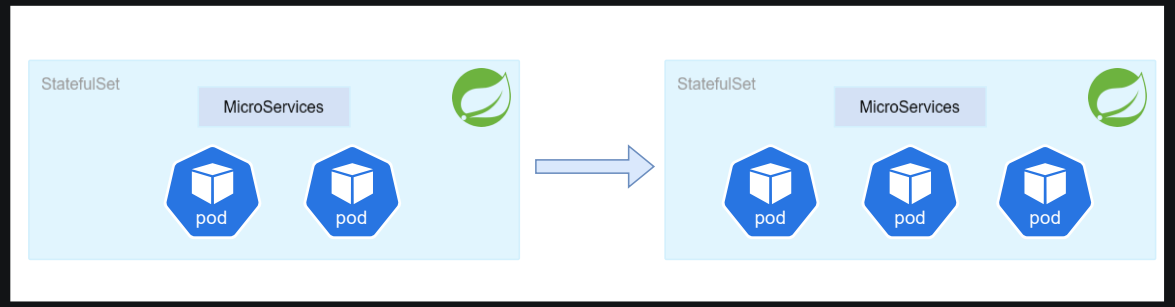
1. Create new databases

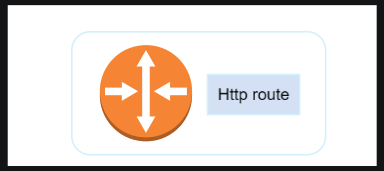
2. Synchronize data to new databases

3. Create new microservices

4. Update routing rules



# Our Workload

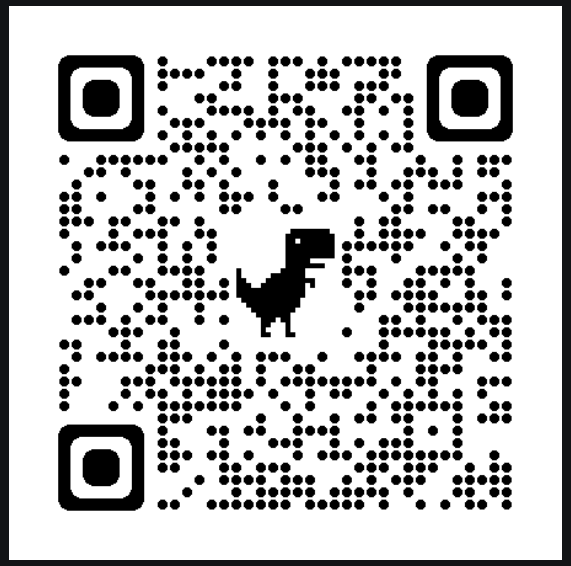
There are:

**9** microservices

**57** Kubernetes YAMLs

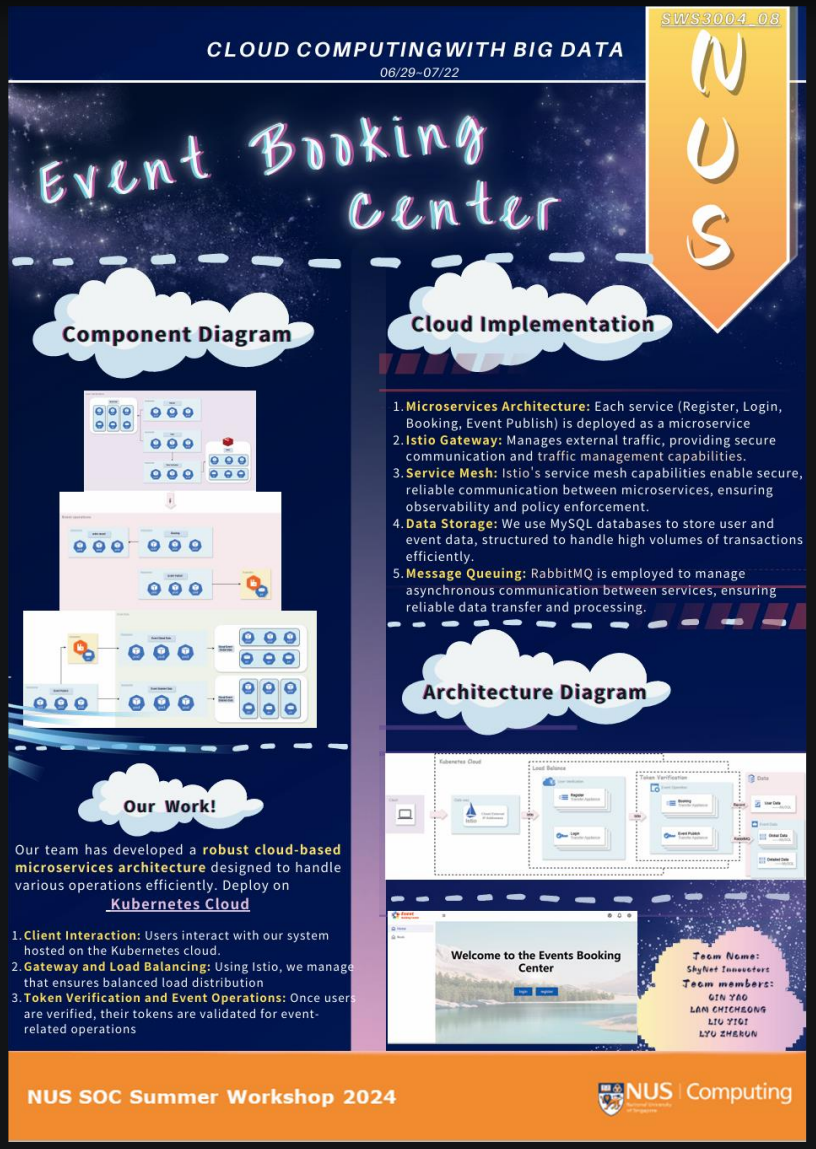
**1** operator implemented by our own
with about **1000** lines of Go code

# Our Code

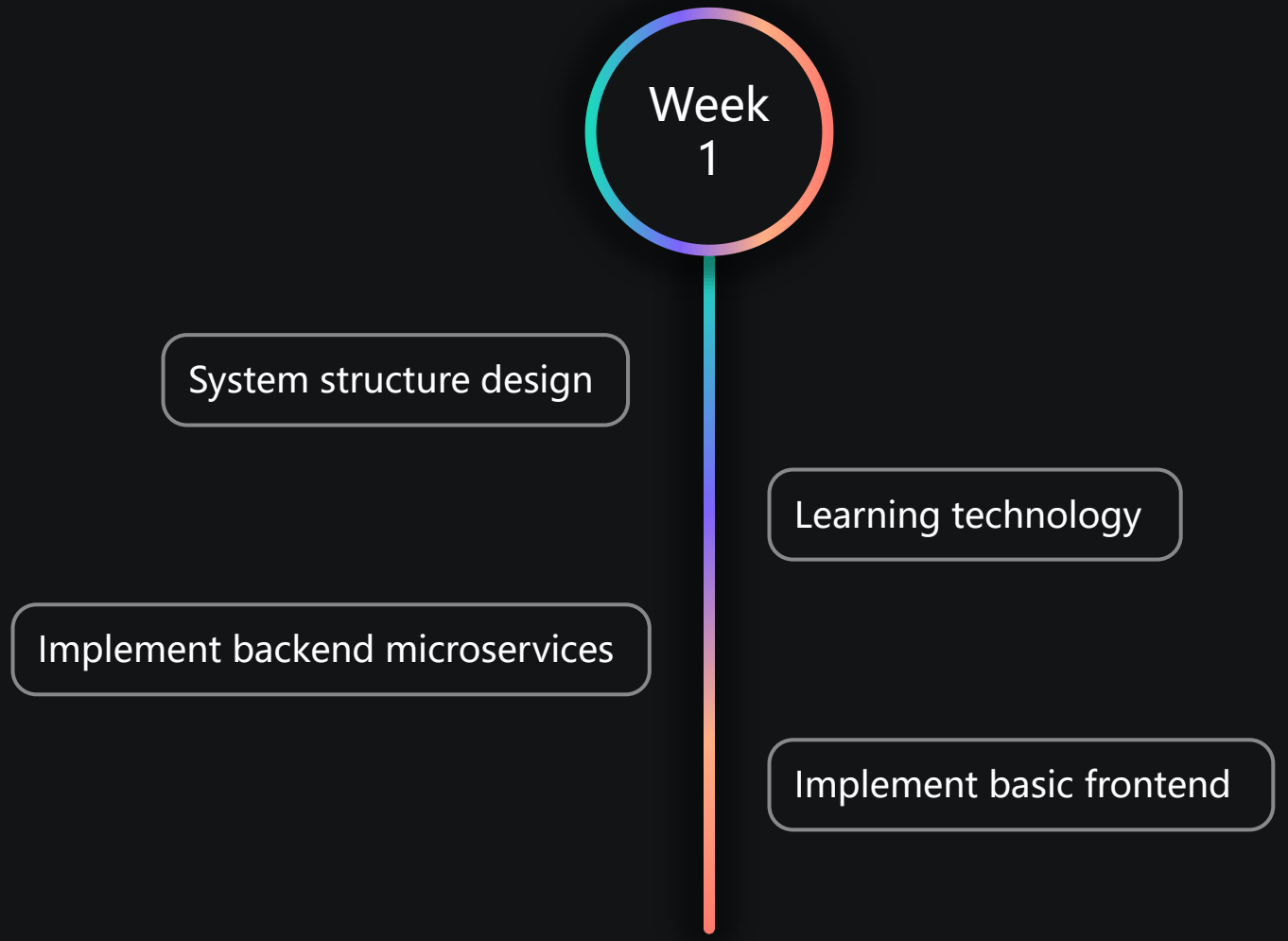


# Demonstration Time

# Poster


CLOUD COMPUTINGWITH BIG DATA
06/29~07/22
SWS3004_08
NUS
Event Booking Center
Component Diagram
Cloud Implementation
1. Microservices Architecture: Each service (Register, Login, Booking, Event Publish) is deployed as a microservice
2. Istio Gateway: Manages external traffic, providing secure communication and traffic management capabilities.
3. Service Mesh: Istio's service mesh capabilities enable secure, reliable communication between microservices, ensuring observability and policy enforcement.
4. Data Storage: We use MySQL databases to store user and event data, structured to handle high volumes of transactions efficiently.
5. Message Queuing: RabbitMQ is employed to manage asynchronous communication between services, ensuring reliable data transfer and processing.
Architecture Diagram
Our Work!
Our team has developed a robust cloud-based microservices architecture designed to handle various operations efficiently. Deploy on Kubernetes Cloud
1. Client Interaction: Users interact with our system hosted on the Kubernetes cloud.
2. Gateway and Load Balancing: Using Istio, we manage that ensures balanced load distribution
3. Token Verification and Event Operations: Once users are verified, their tokens are validated for event-related operations
Welcome to the Events Booking Center
Team Name:
SkyNet Innovators
Team members:
GIN YAO
LAM CHICHEONG
LIU YIGI
LYU ZHERUN
NUS SOC Summer Workshop 2024
NUS | Computing

# Project Timeline



# Preliminary Work

Week 2

Implement system structure on k8s

Implement routing by hash

Implement routing using Istio

Implement data synchronization using RabbitMQ

Week 3

# Middle Presentation

# Main Development

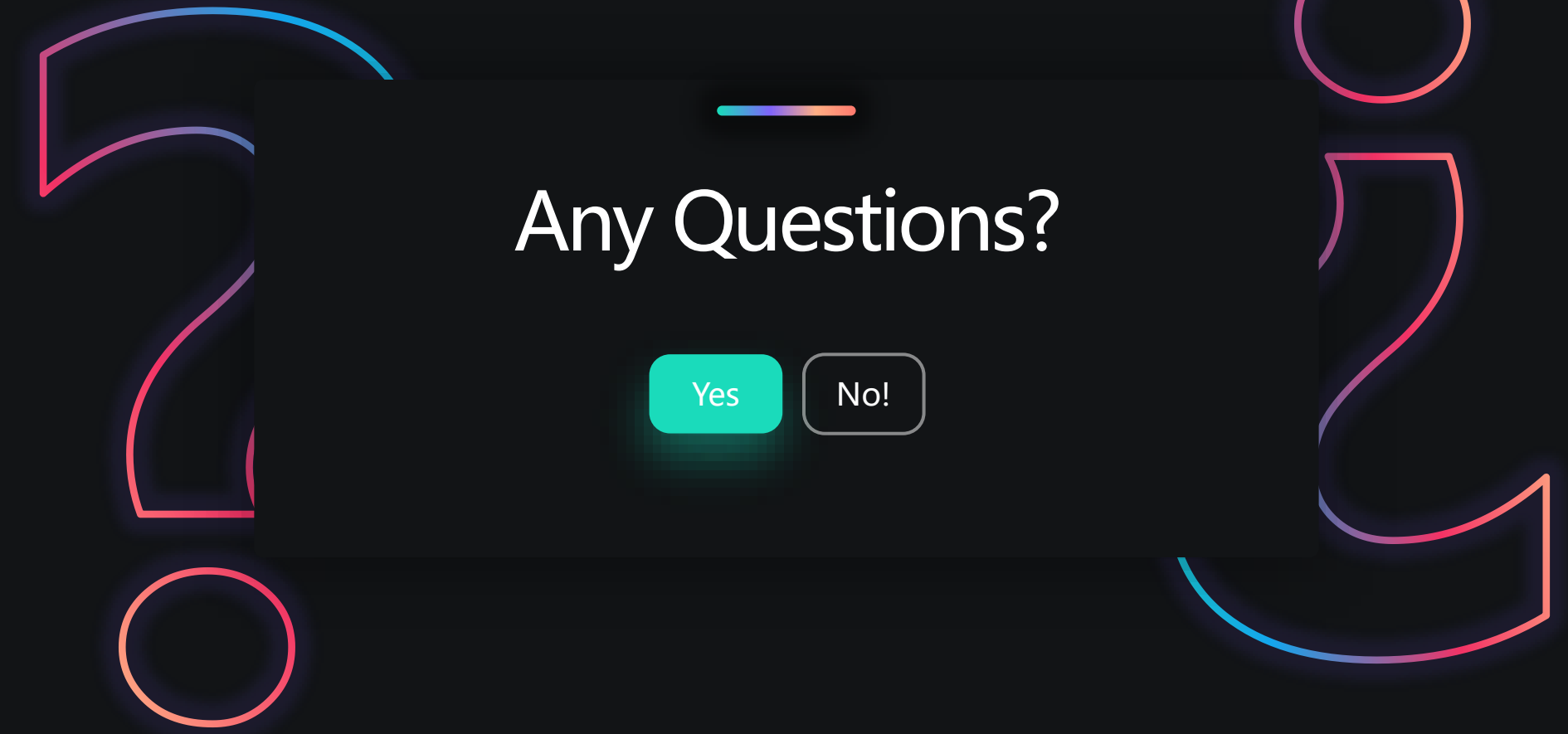
Week 3

Implement auto scaling using operator

Improve frontend

Poster and ppt

# Final Presentation



# Any Questions?

Yes

No!


Thanks