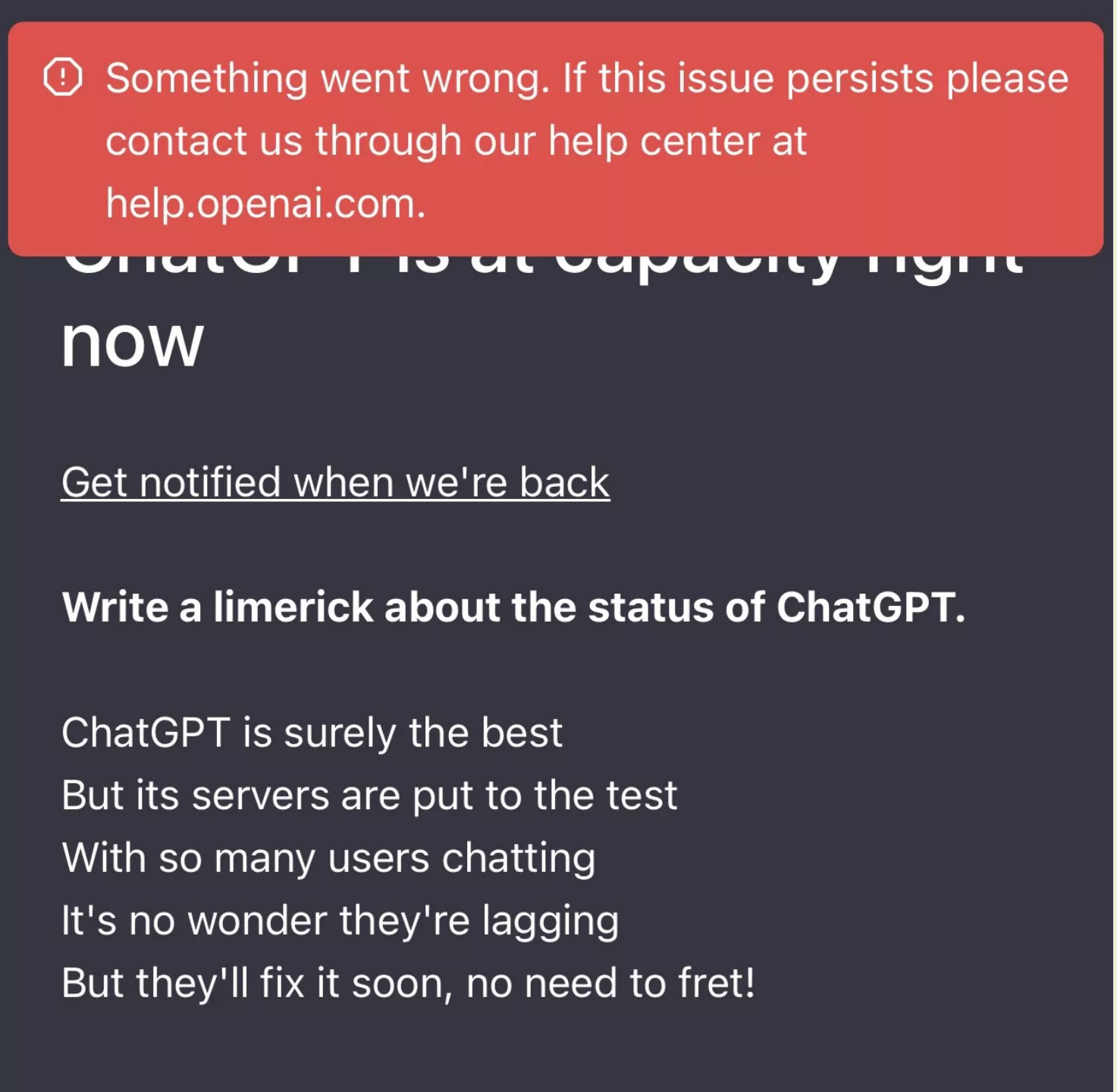
Unleashing AI's power through cloud-native resilience and Kubernetes-driven scalability

UPLION: AI Gateway



How can we build more resilient AI infrastructures?

UP LION

UPLION

Unified Platform for AI

Leveraging cloud-native design

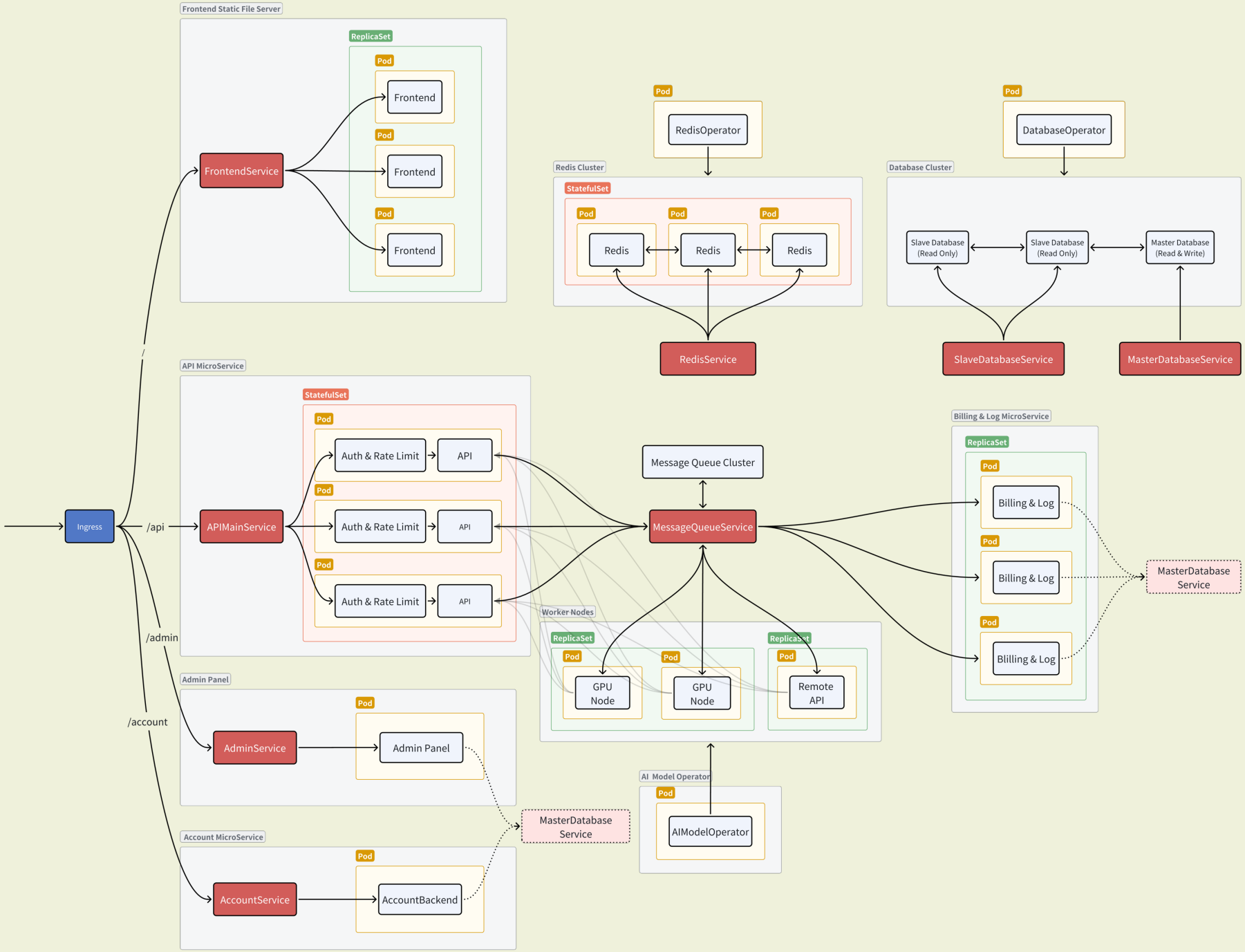
Intelligent self-serving task distribution

Optimizing resource allocation

Natively integrated with Kubernetes

Unified Platform for AI

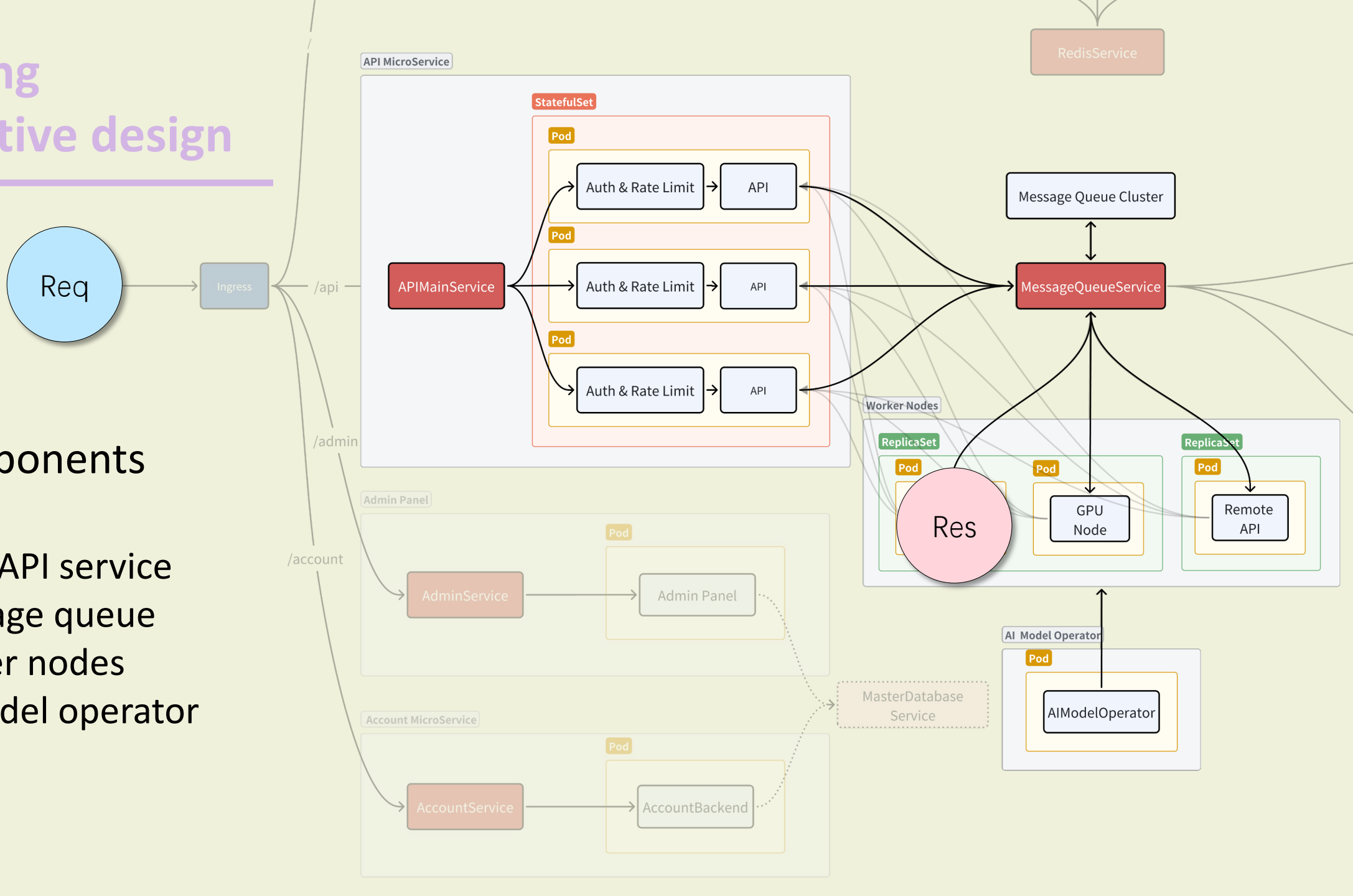
Leveraging cloud-native design



Leveraging cloud-native design

key components

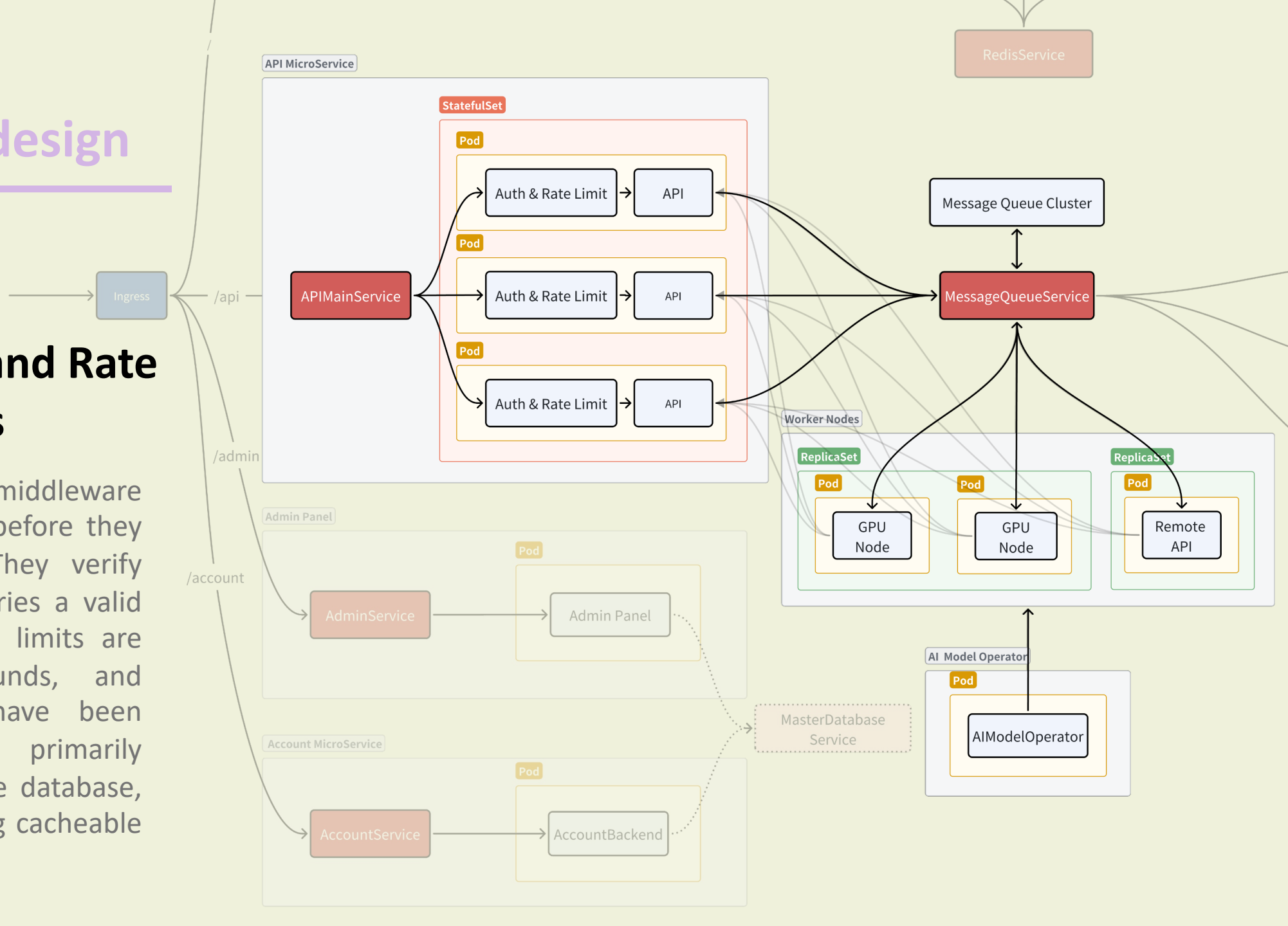
1. Main API service
2. message queue
3. worker nodes
4. AI model operator



Leveraging cloud-native design

Authentication and Rate Limiting Services

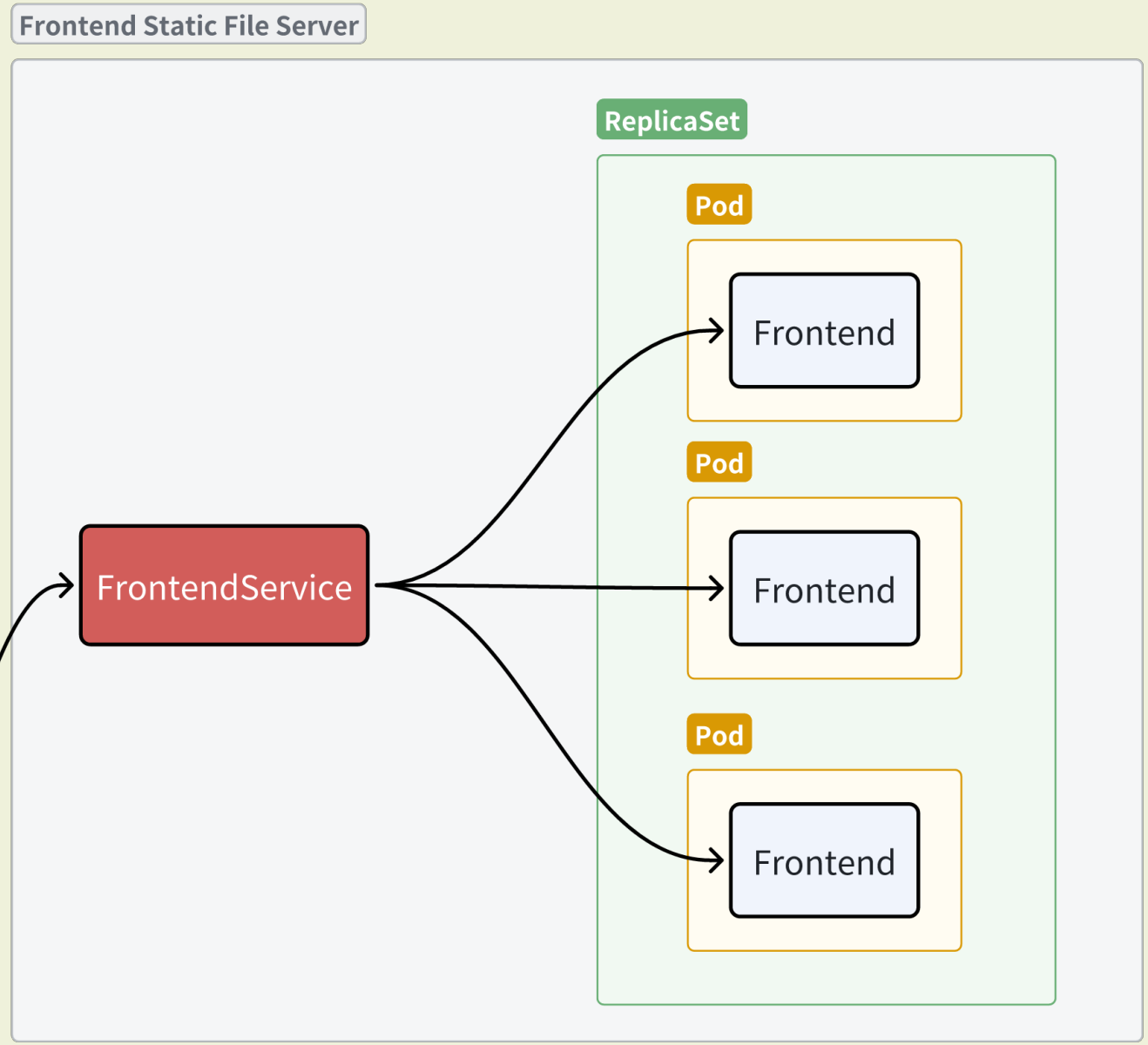
These services act as middleware that processes requests before they reach the API node. They verify whether the request carries a valid token, check if the rate limits are within acceptable bounds, and determine if quotas have been exhausted. This part primarily involves reading from the database, with almost all data being cacheable via Redis.



Leveraging cloud-native design

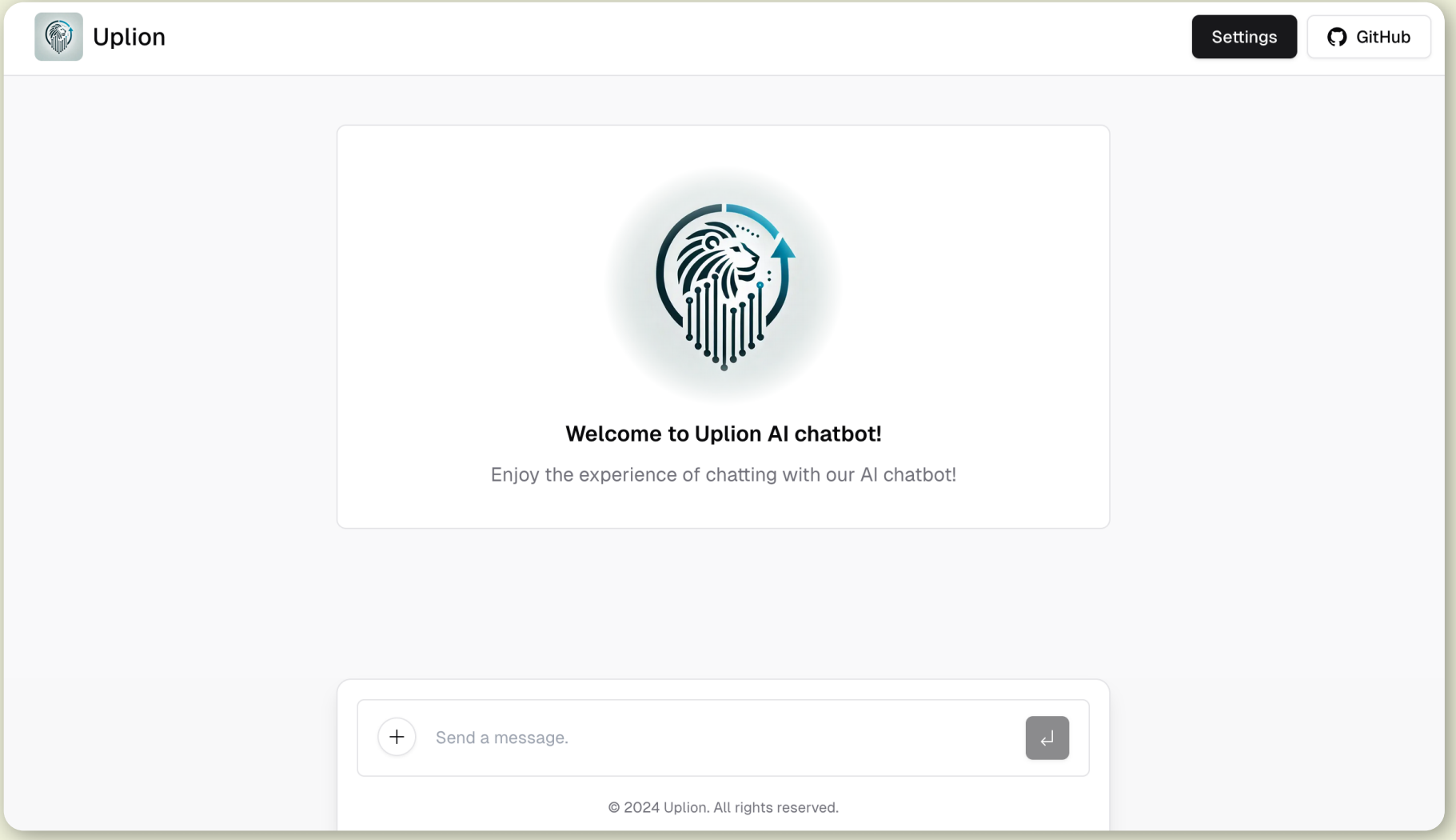
Frontend

Similar to interfaces like ChatGPT, the frontend is designed for ease of use, allowing users to interact smoothly with the system.



Leveraging cloud-native design

Frontend



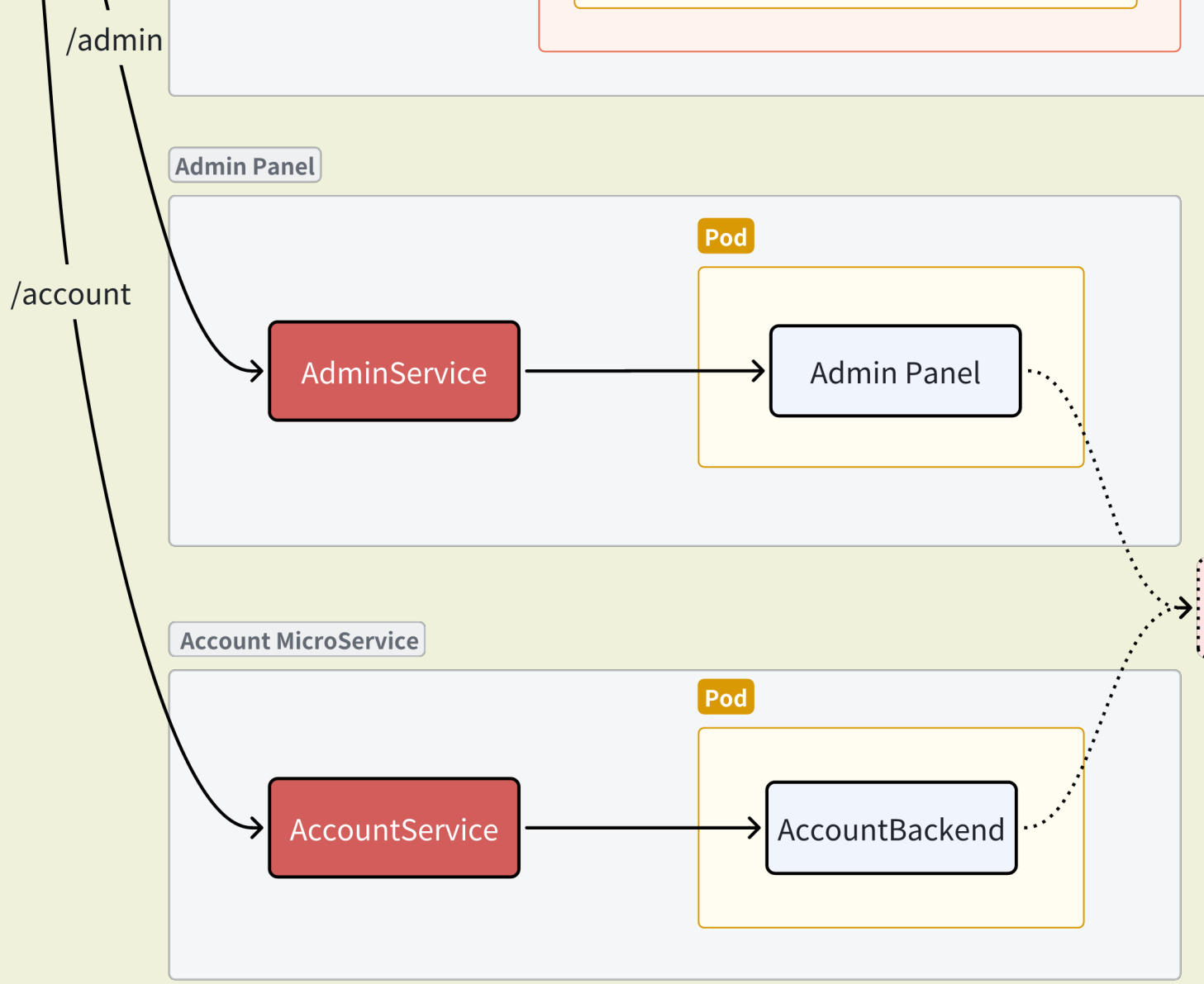
Leveraging cloud-native design

Admin Dashboard

Leveraging Kubernetes and the operator, this dashboard provides a user-friendly interface for managing worker nodes (or AI models). It also manages users and tokens.

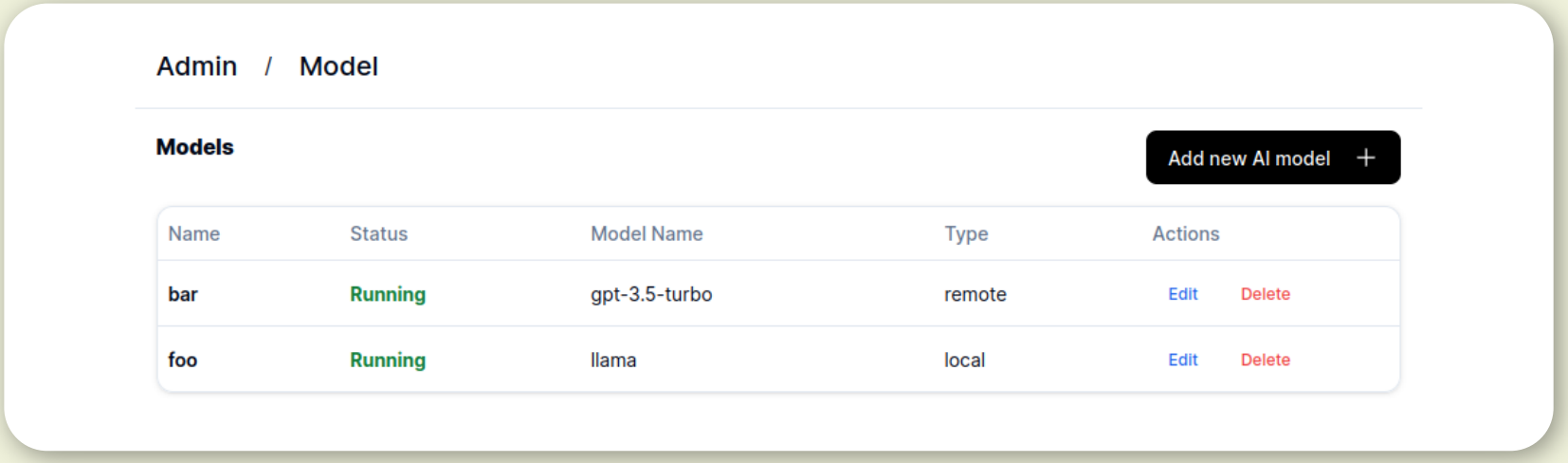
Account Services

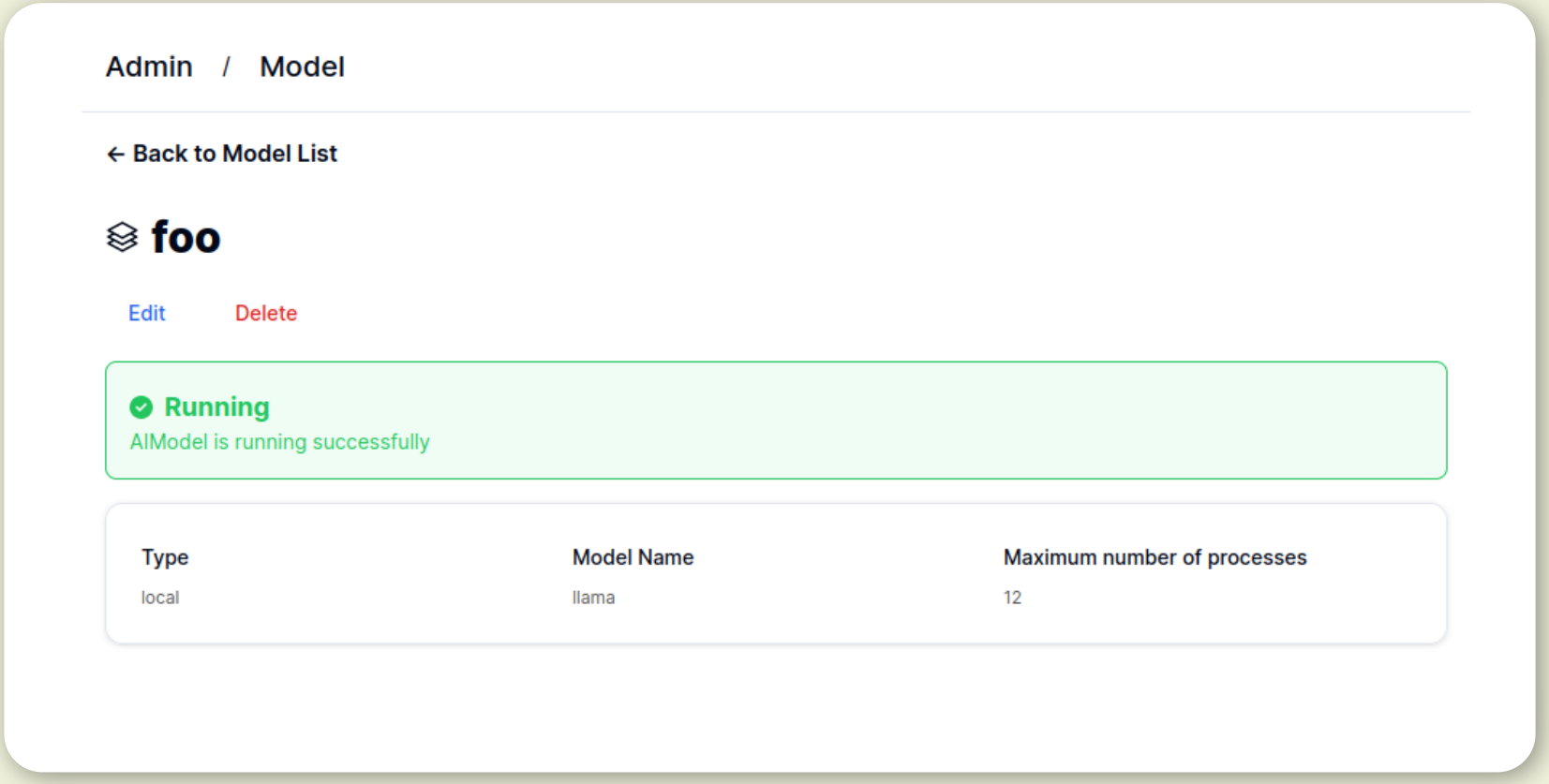
These services provide authentication for both the frontend and the management dashboard.



Leveraging cloud-native design

Admin Dashboard

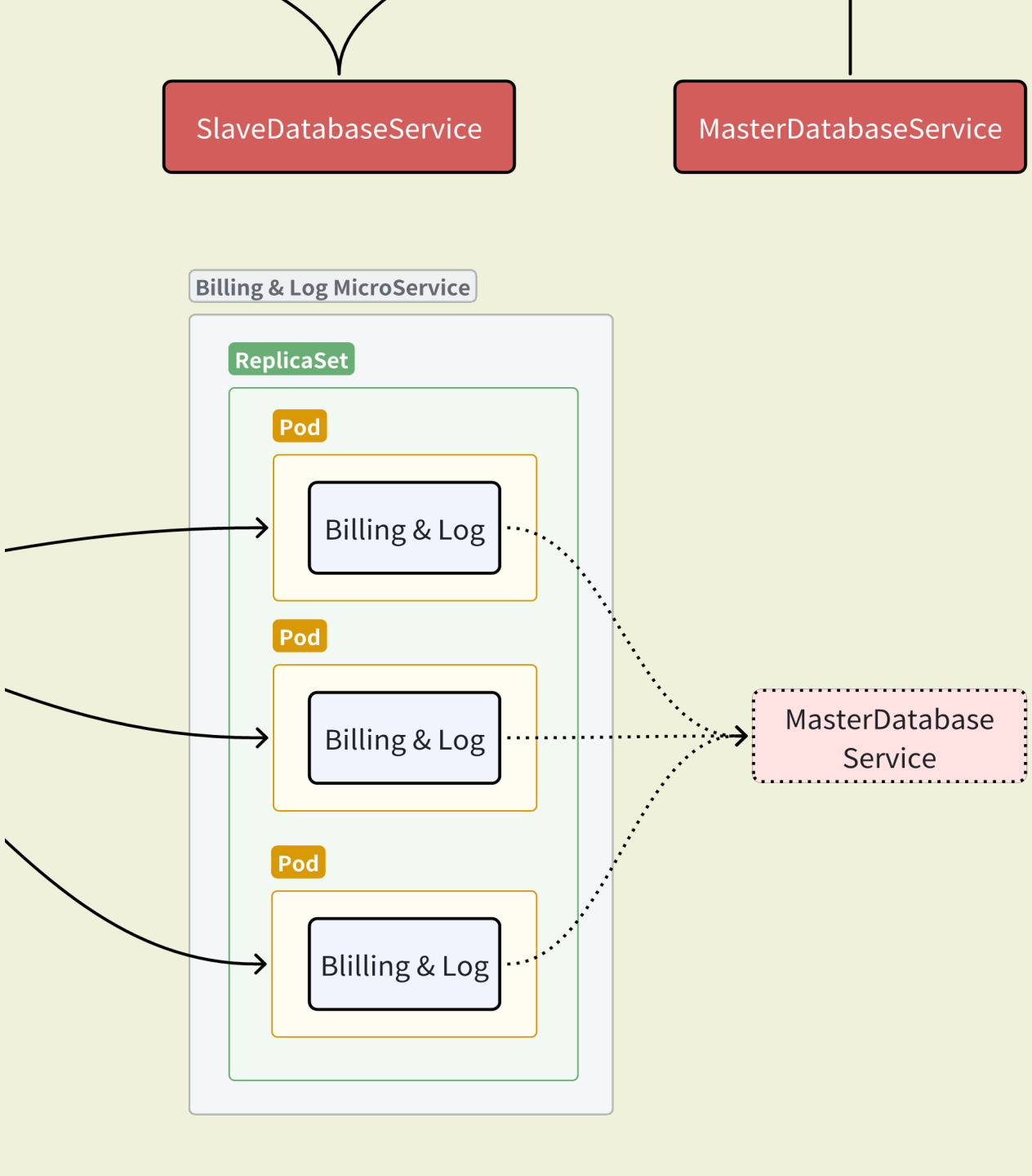




Leveraging cloud-native design

Billing Service

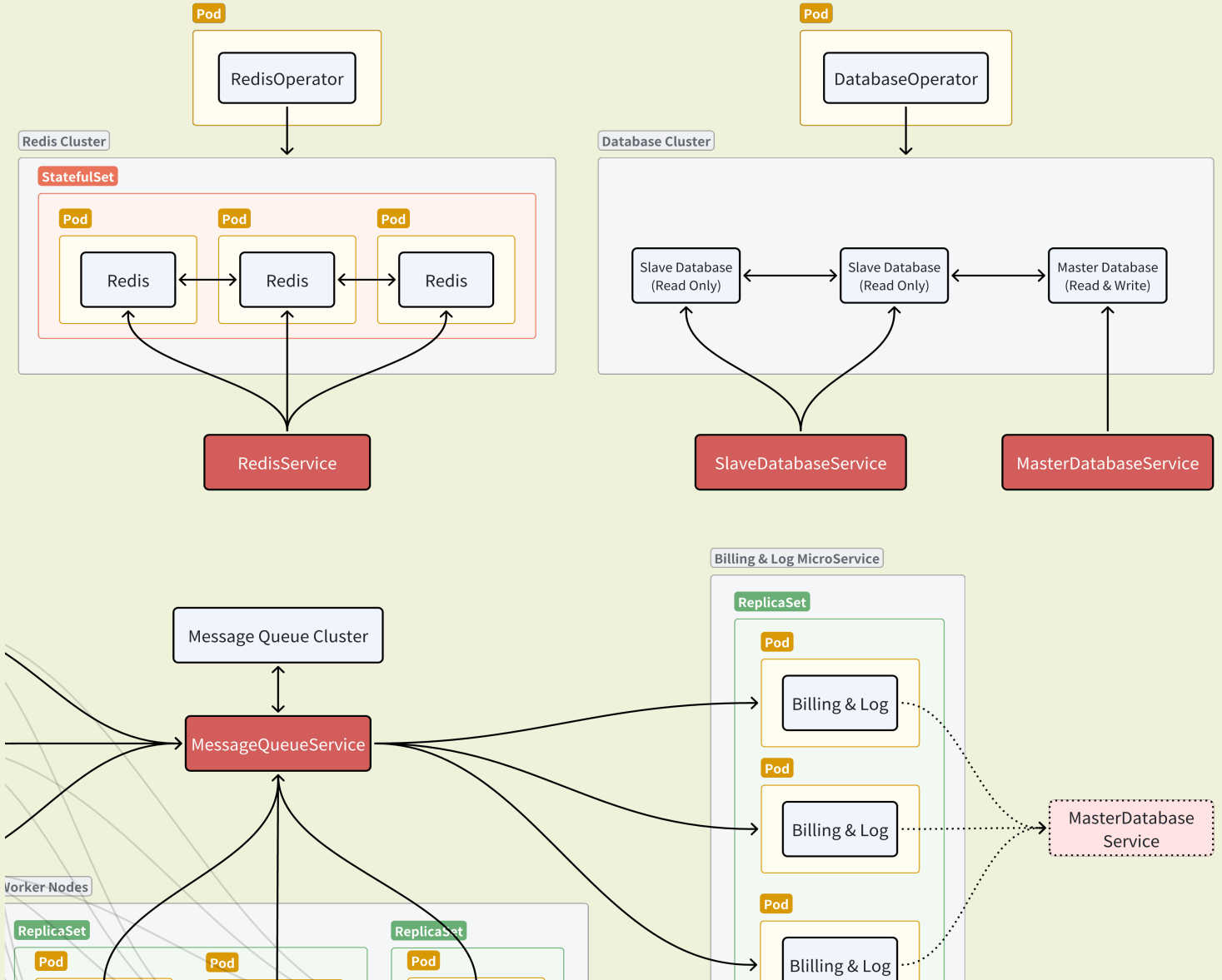
After a worker node completes a task, it collects statistical and operational data, directing this information into a batch processing queue. The combined Billing & Log Service then aggregates these details, consolidating them into a single database write operation. This approach effectively reduces the load on the database by minimizing the frequency and volume of write requests, ensuring both efficient data handling and accurate billing records.



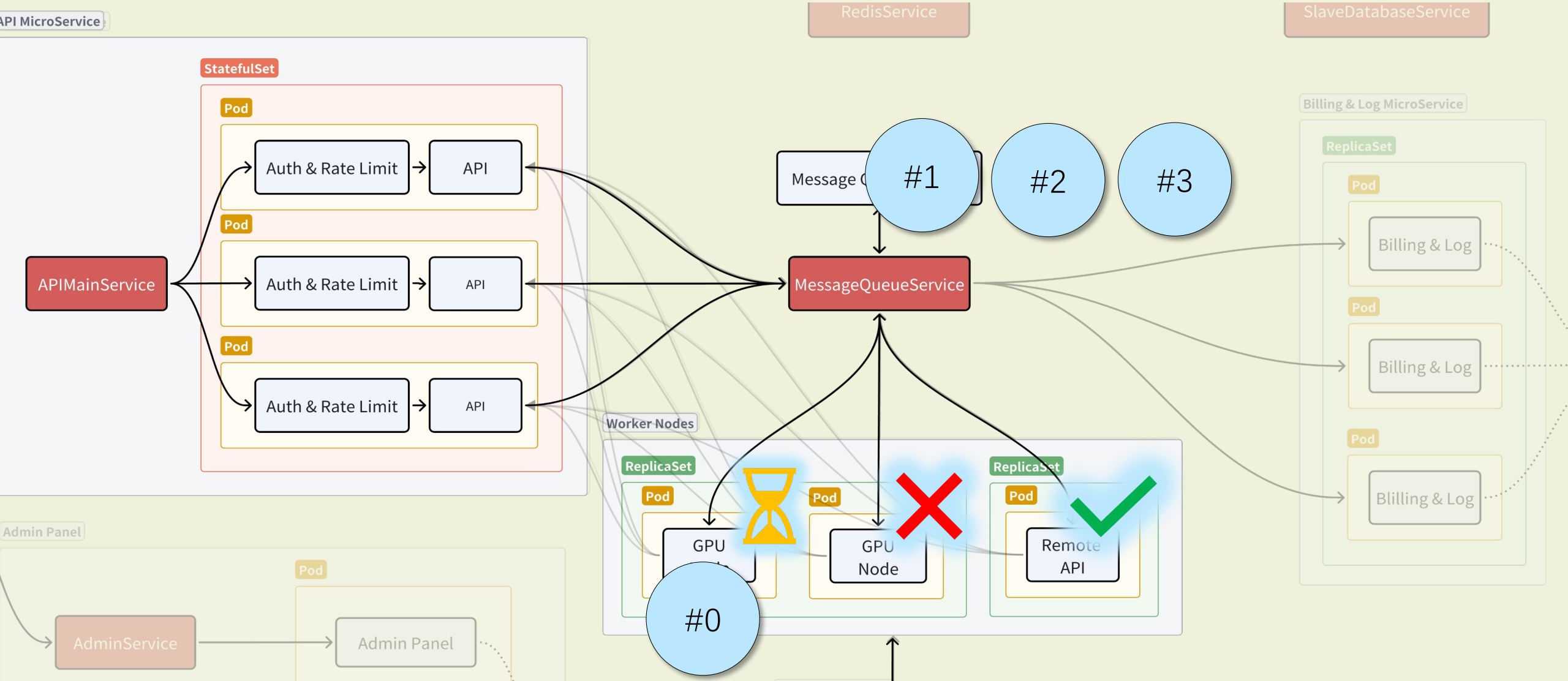
Leveraging cloud-native design

Storage and Message Queue

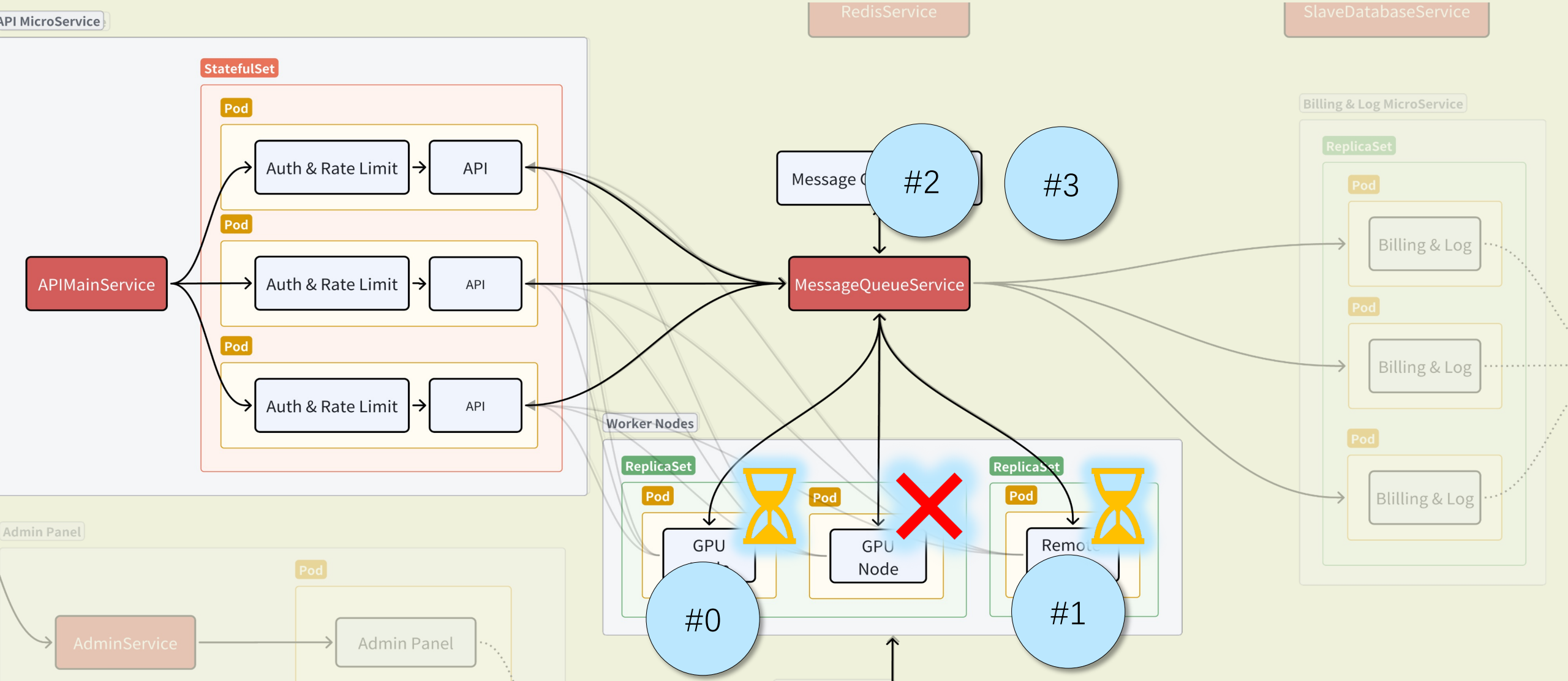
We utilize a PostgreSQL database cluster for our primary storage needs due to its robustness and reliability. It's important to note that throughout the entire API request process, there is no need to write to the database, ensuring that request concurrency and latency are not adversely affected. For message queuing, we use Pulsar.



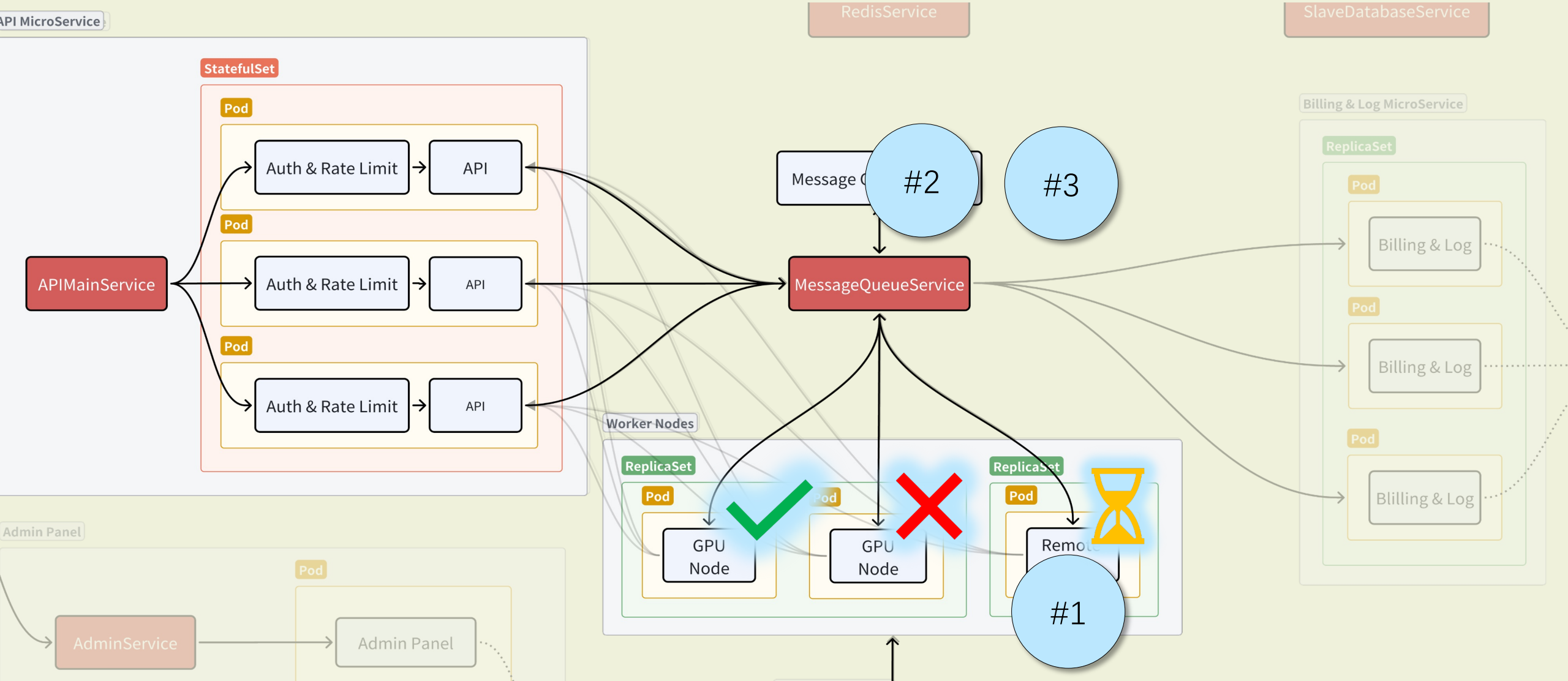
Intelligent self-serving task distribution



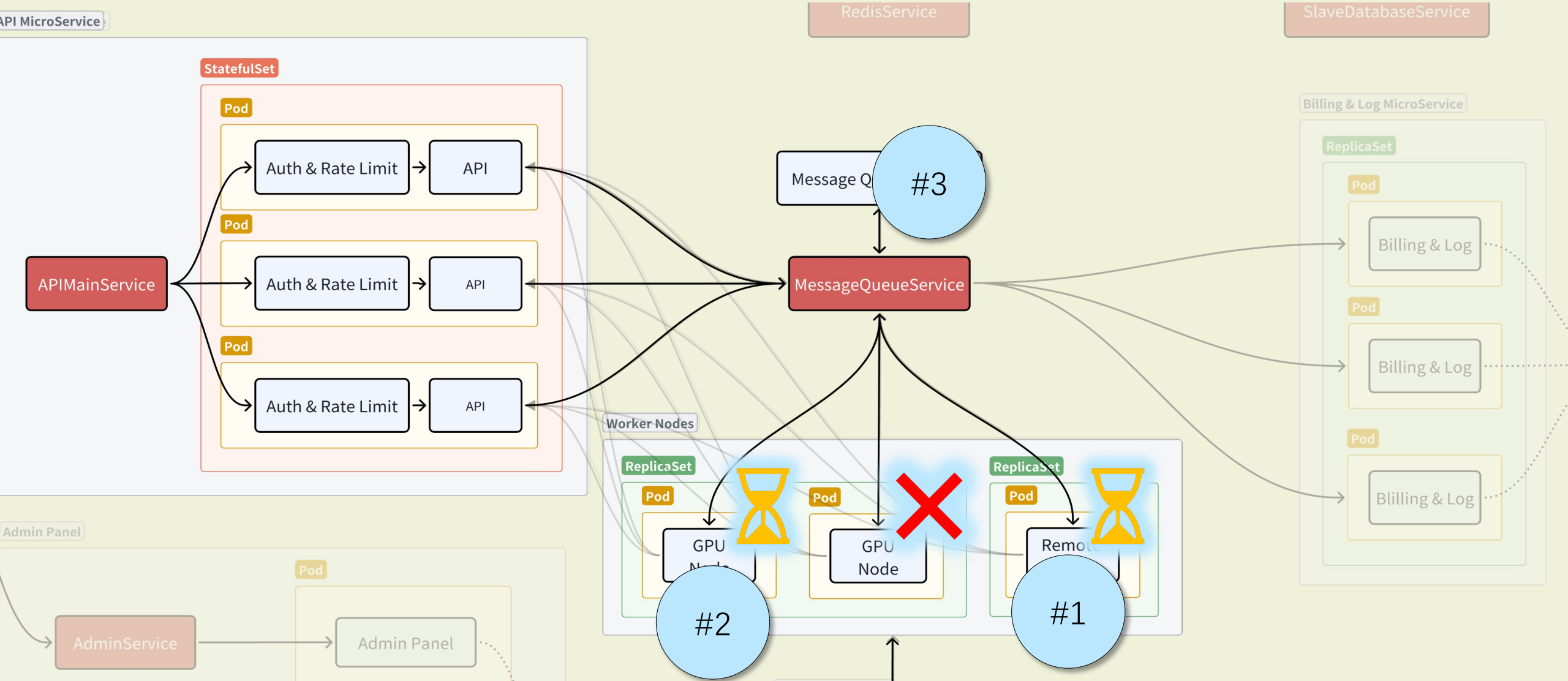
Intelligent self-serving task distribution



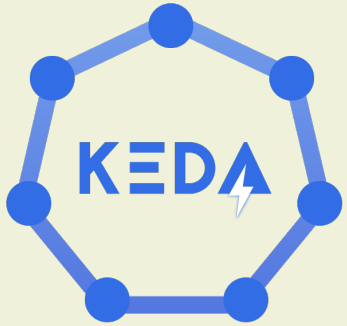
Intelligent self-serving task distribution



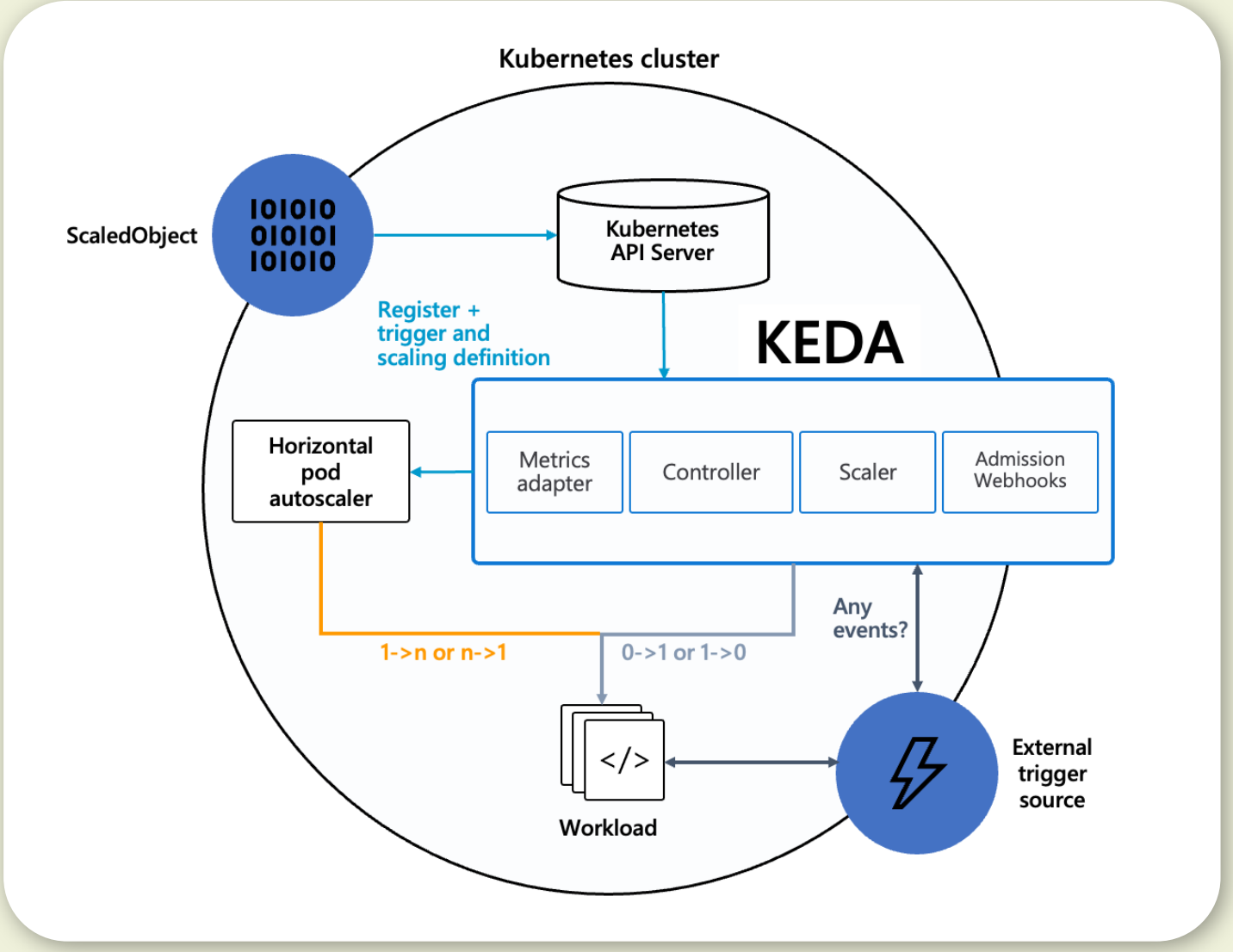
Intelligent self-serving task distribution



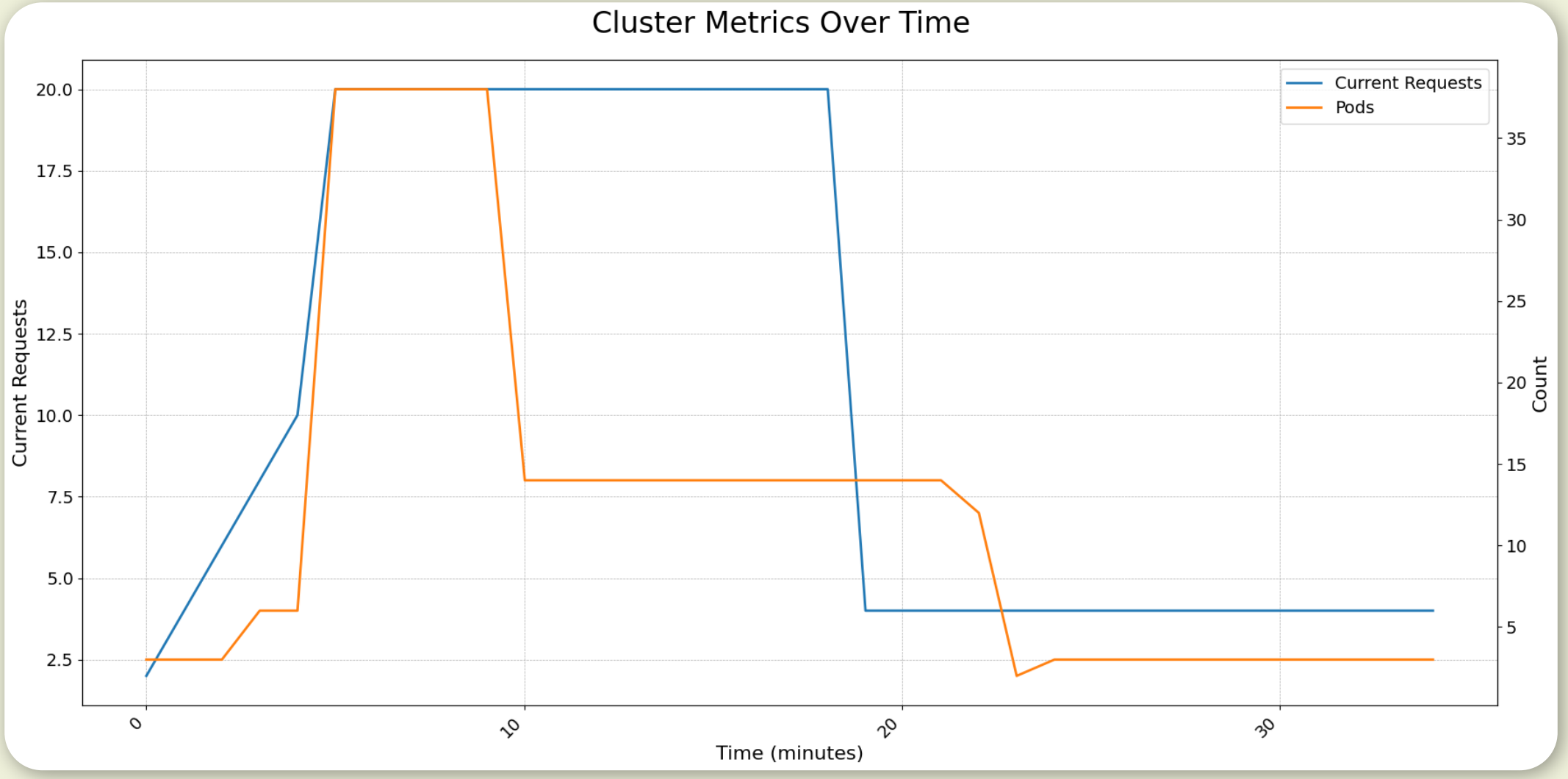
Optimizing resource allocation



ubernetes Event-driven Autoscaling



Optimizing resource allocation



Natively Integrated with Kubernetes

AI Model CRD
(Custom Resource Definition)

and

An Operator that manages it and implements its functionality

ai-model.yaml

```yaml
apiVersion: model.youxam.com/v1alpha1
kind: AIModel
metadata:
  name: ai-model-sample
spec:
  type: local
  model: TinyLlama-1.1B
  replicas: 3
  image: user/image:tag
  maxProcessNum: 256
```

Natively Integrated with Kubernetes

AI Model CRD
(Custom Resource Definition)

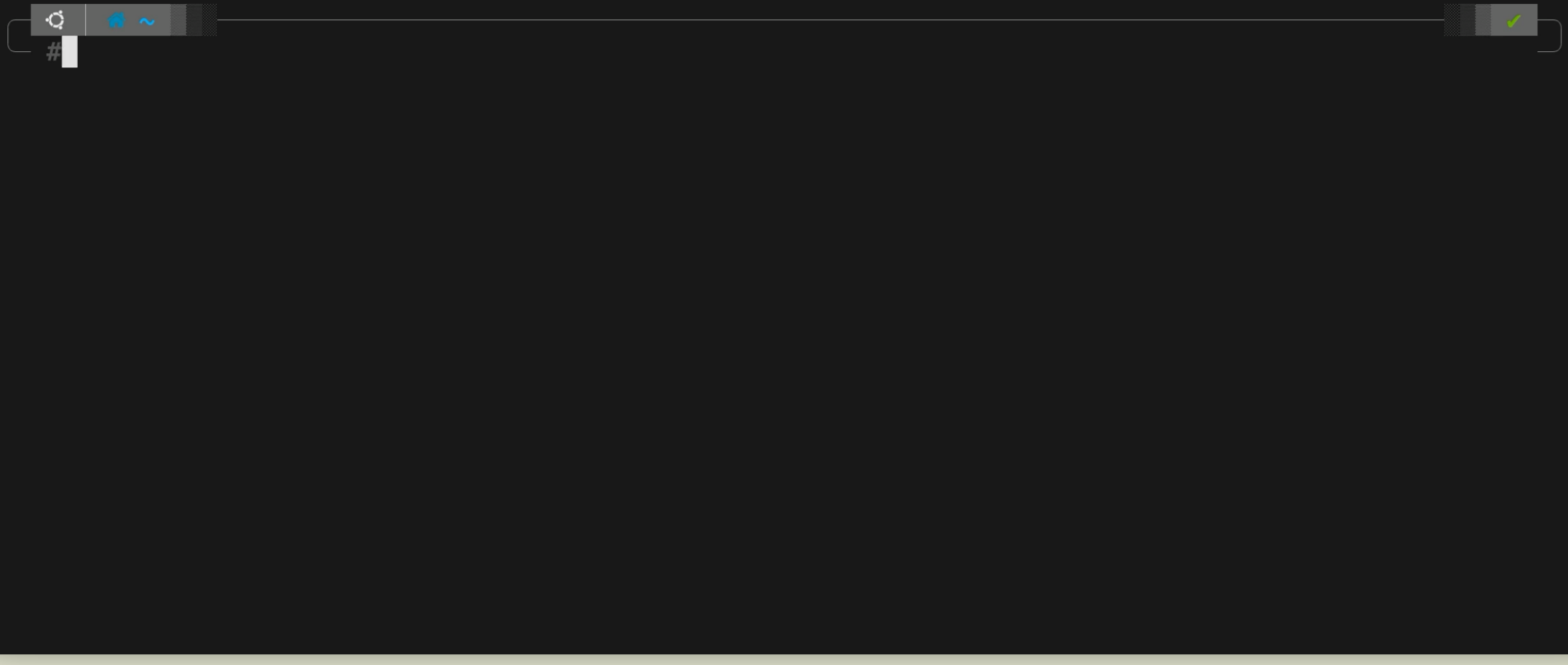
and

An Operator that manages it and implements its functionality

ai-model.yaml

```yaml
apiVersion: model.youxam.com/v1alpha1
kind: AIModel
metadata:
  name: ai-model-sample
spec:
  type: local
  model: TinyLlama-1.1B
  replicas: 3
  msgBacklogThreshold: 2
  image: user/image:tag
  maxProcessNum: 256
```

Natively Integrated with Kubernetes



Technology Selection

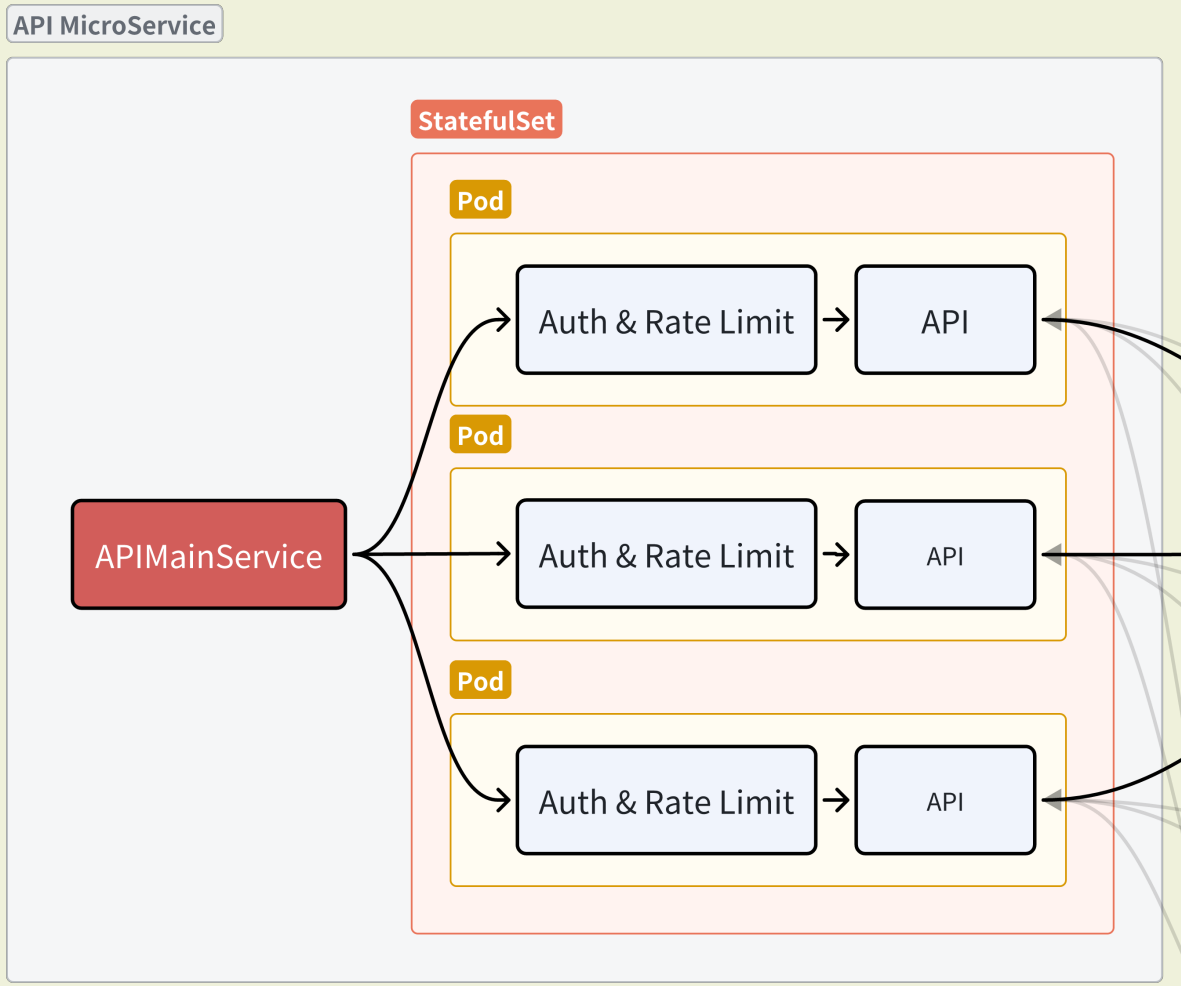
Istio

Istio extends Kubernetes to establish a programmable, application-aware network. Working with both Kubernetes and traditional workloads, Istio brings standard, universal traffic management, telemetry, and security to complex deployments.

Istio

Istio

Intercepting Requests for Middleware



Programming Languages

Main API Service

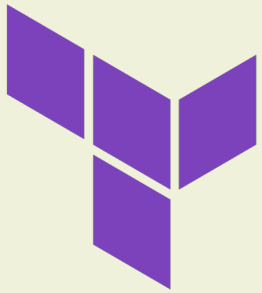
AI model operator

Frontend

Admin Dashboard

worker node

Terraform



HashiCorp

Terraform

Terraform is an infrastructure as code tool that lets you build, change, and version infrastructure safely and efficiently. This includes low-level components like compute instances, storage, and networking; and high-level components like DNS entries and SaaS features.

Terraform

HashiCorp
Terraform

How to deploy UPLION?

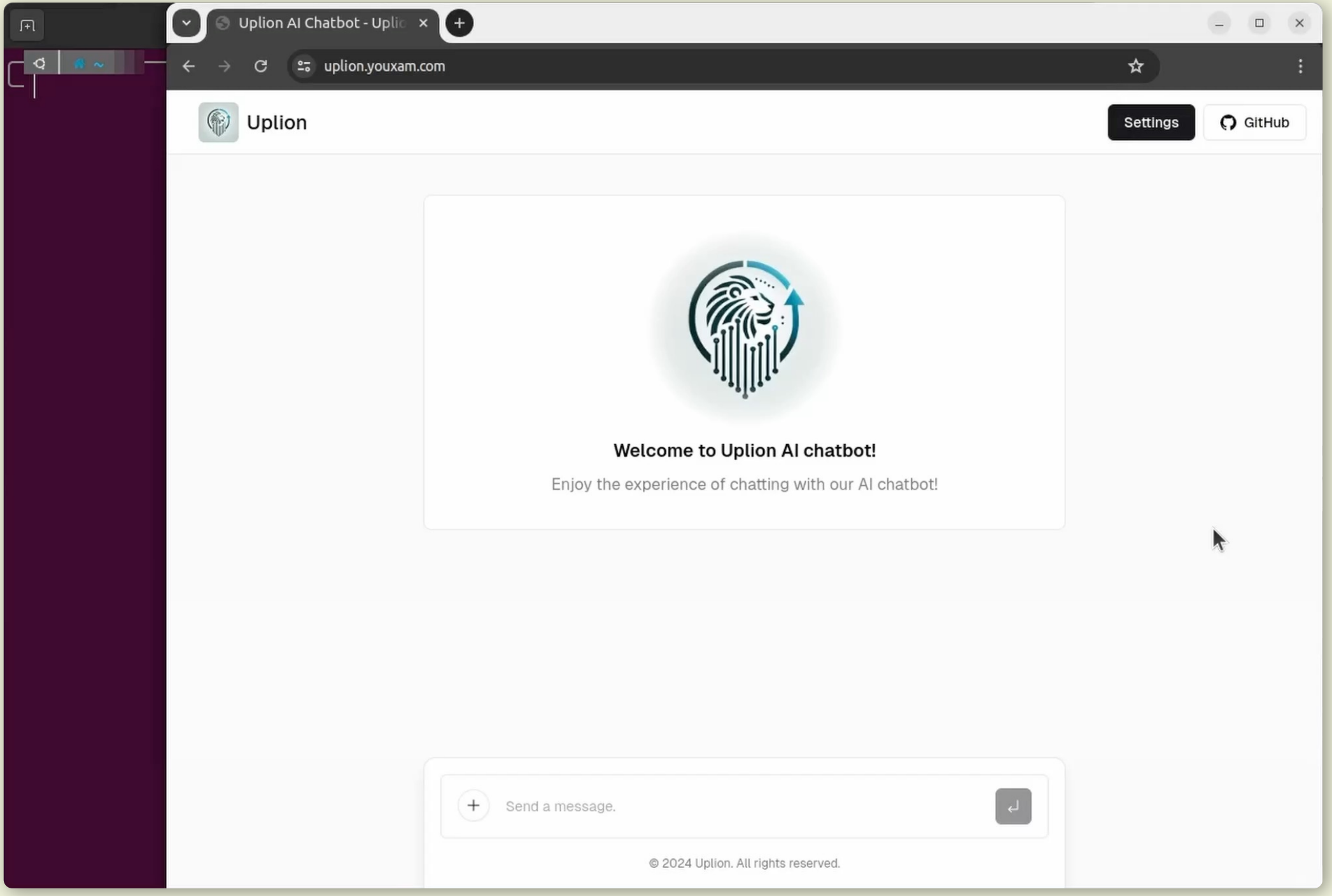
```
# Clone our config
$ git clone https://github.com/uplion/infra-config.git

# Navigate to the cloned configuration directory
$ cd infra-config

# Install dependencies
$ terraform init

# Deploy the entire architecture
$ terraform apply
```

Demonstration


Uplion AI Chatbot - Uplic
uplion.youxam.com
Uplion
Settings
GitHub
Welcome to Uplion AI chatbot!
Enjoy the experience of chatting with our AI chatbot!
Send a message.
© 2024 Uplion. All rights reserved.

Conclusion

Open Source

https://github.com/uplion

UPLION

×

kubernetes

Thanks For Watching

Zhang Haoling

YouXam

Wang Ruozhu

sherlockedhzoi

Cai Yiwen

DrinkLessMilkTea

Zhu Yuanchuan

KKChrisYC