

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

---

UPLION: AI Gateway

Something went wrong. If this issue persists please contact us through our help center at [help.openai.com](https://help.openai.com).

~~ChatGPT is at capacity right now~~

now

[Get notified when we're back](#)

**Write a limerick about the status of ChatGPT.**

ChatGPT is surely the best  
But its servers are put to the test  
With so many users chatting  
It's no wonder they're lagging  
But they'll fix it soon, no need to fret!

**How can we build  
more resilient AI  
infrastructures?**



U P L I O N



UP

L

I

O

N

Unified Platform for AI

Leveraging cloud-native design

Intelligent self-serving task distribution

Optimizing resource allocation

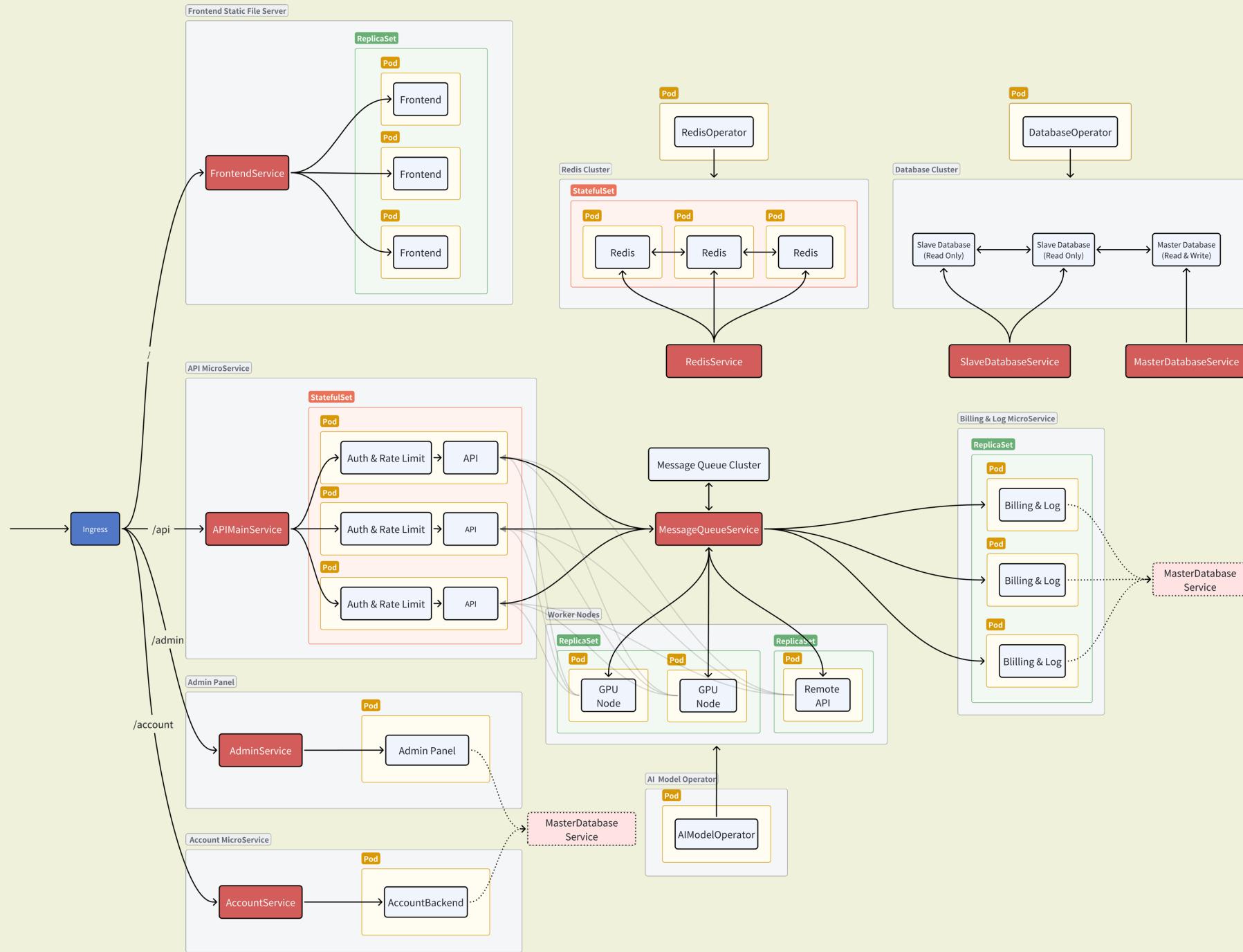
Natively integrated with Kubernetes

# Unified Platform for AI

---



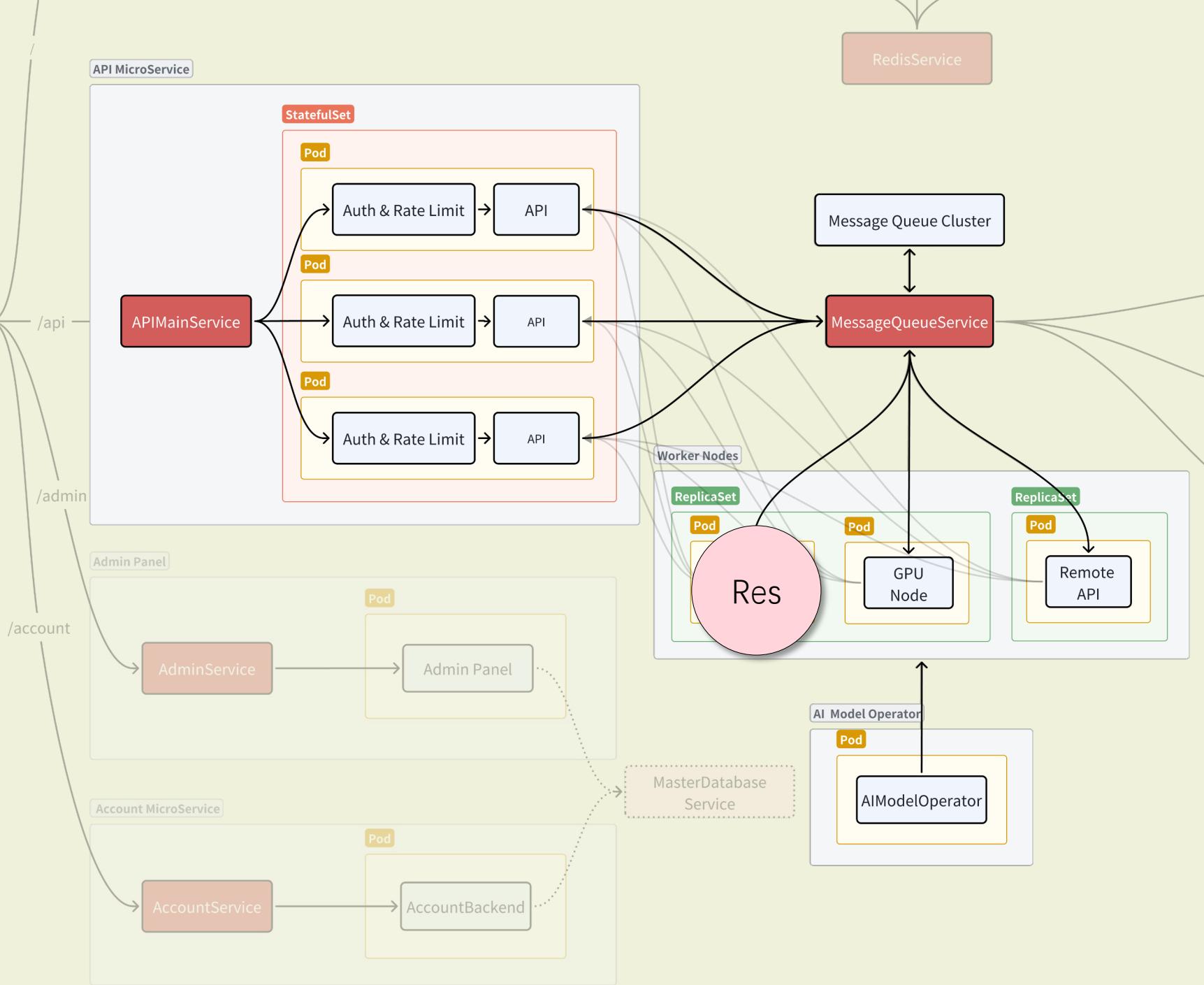
# Leveraging cloud-native design



# Leveraging cloud-native design

## key components

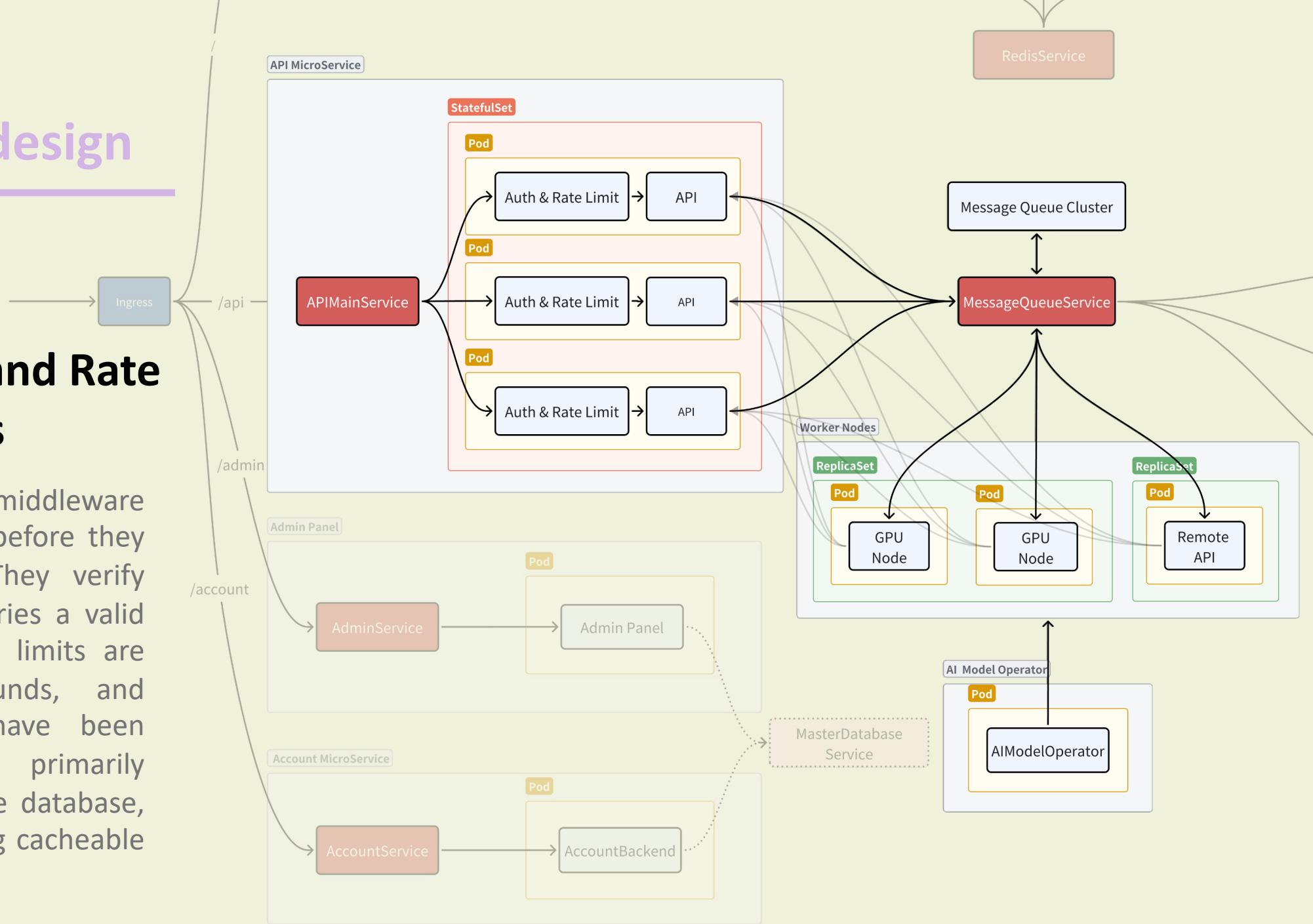
1. Main API service
2. message queue
3. worker nodes
4. AI model operator



# 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.



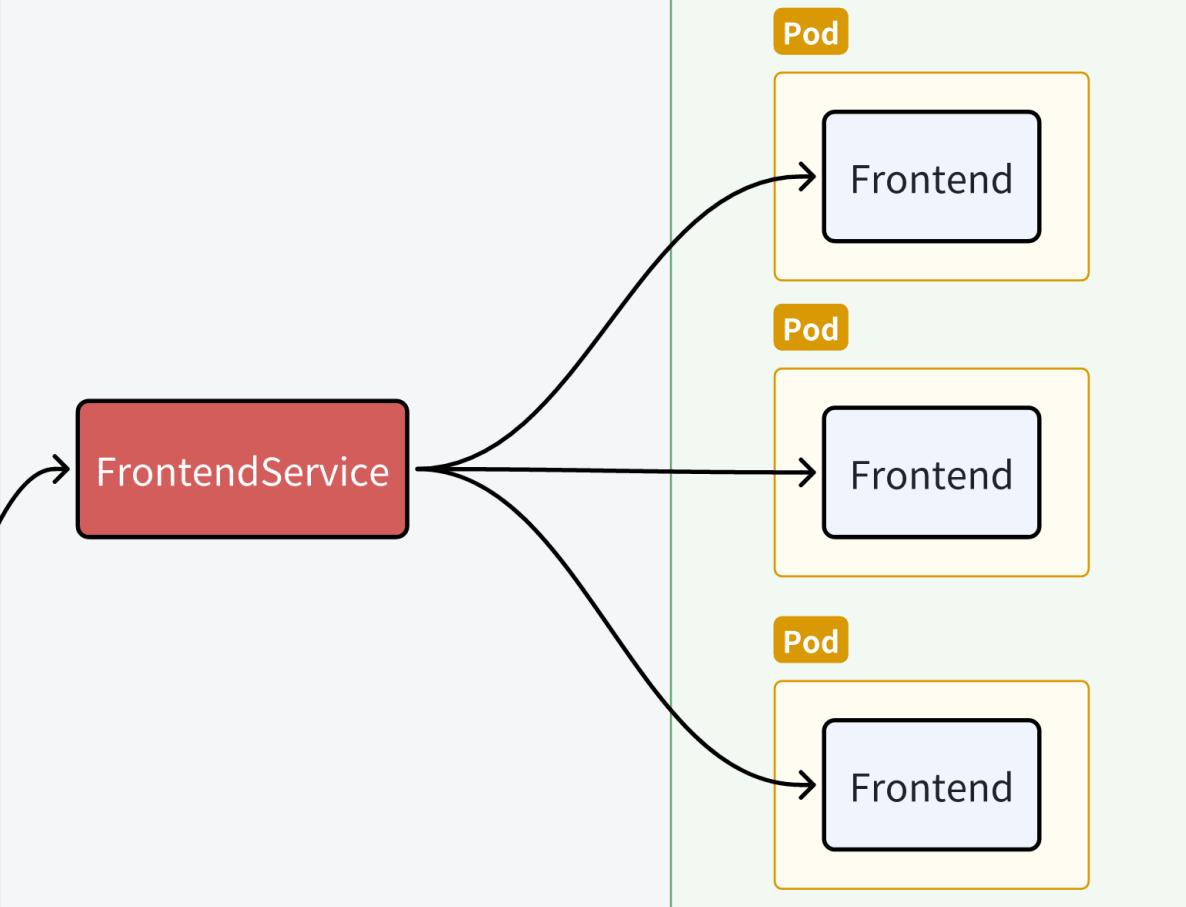
# Leveraging cloud-native design

## Frontend

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

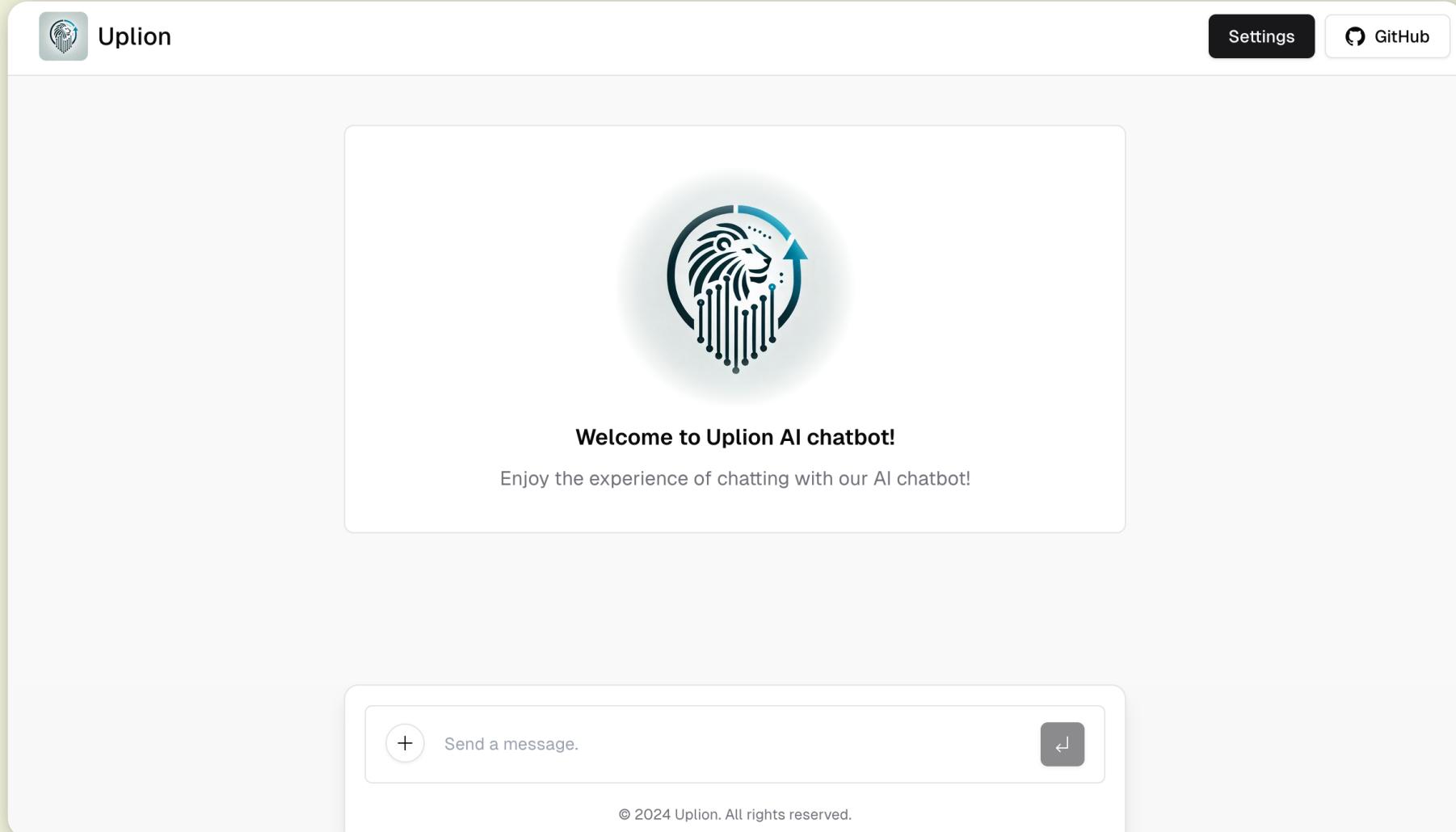
Frontend Static File Server

ReplicaSet



# Leveraging cloud-native design

## Frontend



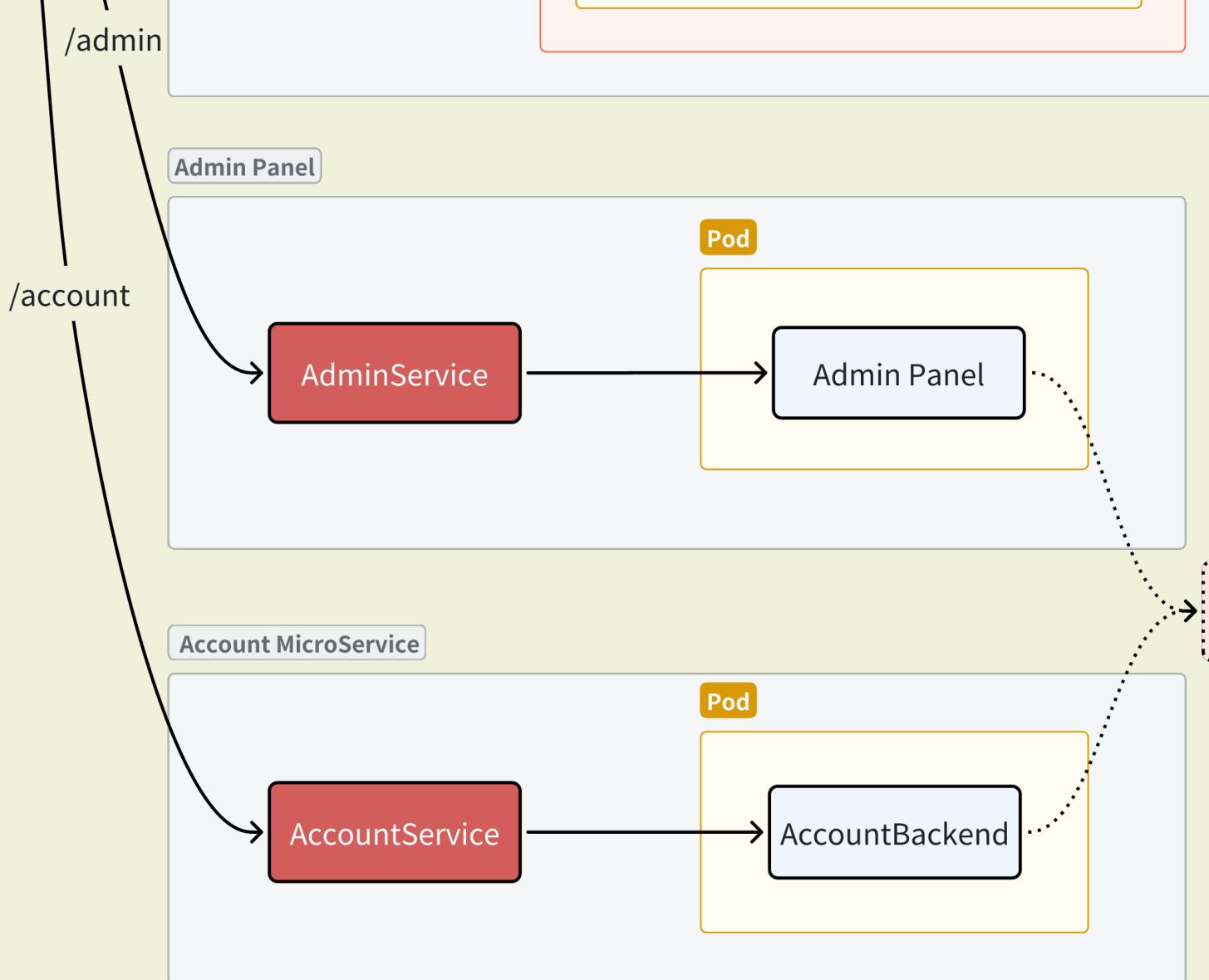
# Leveraging cloud-native design

## 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

Admin / Model

**Models**

Add new AI model +

Name	Status	Model Name	Type	Actions
bar	Running	gpt-3.5-turbo	remote	Edit Delete
foo	Running	llama	local	Edit Delete

Admin / Model

← Back to Model List

 **foo**

[Edit](#) [Delete](#)

 **Running**

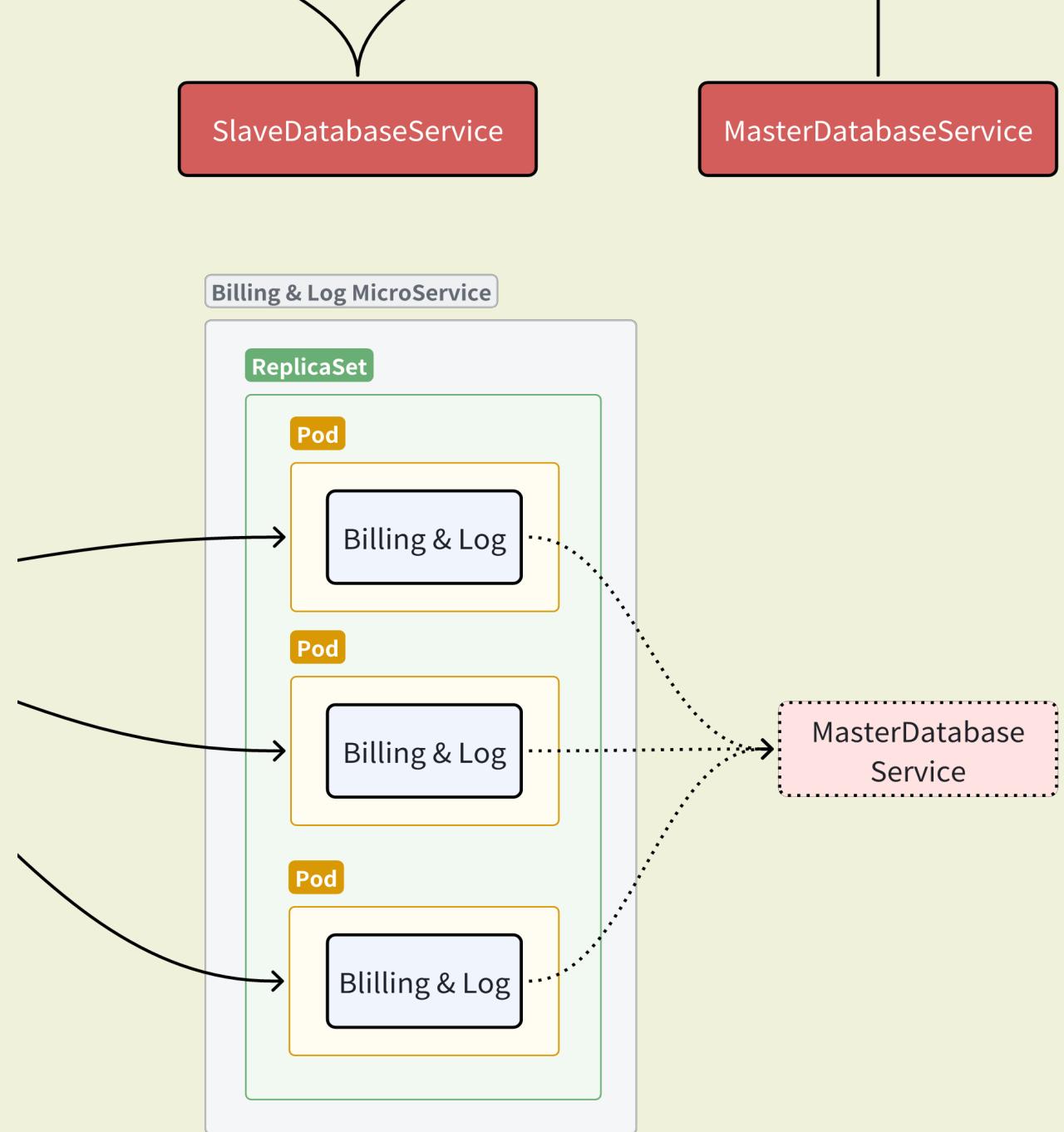
AIModel is running successfully

Type	Model Name	Maximum number of processes
local	llama	12

# Leveraging cloud-native design

## Billing Service

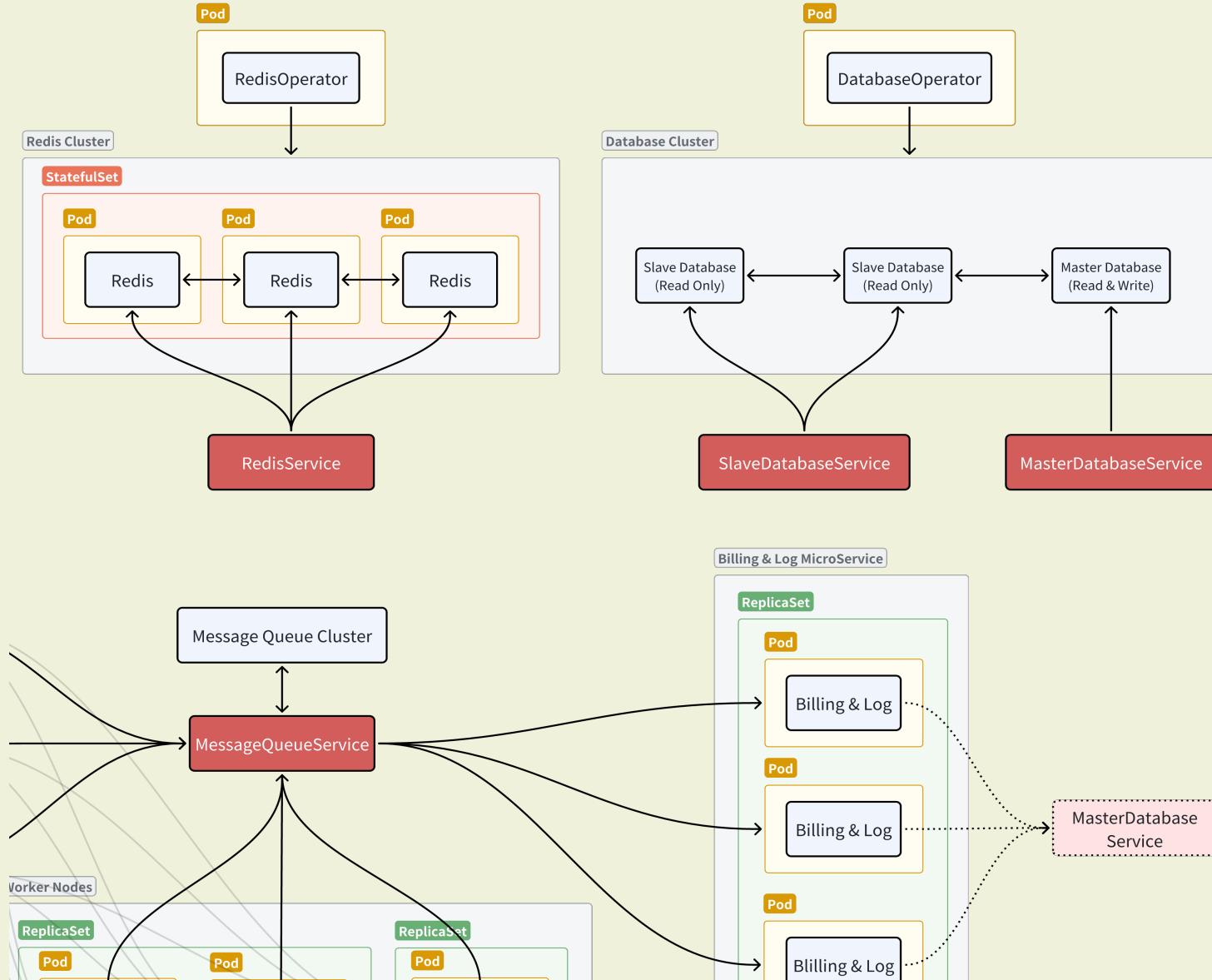
After a worker node completes a task, it collects statistical and operational data, directing this information into a batch processing queue. The combined Billing & Log Service then aggregates these details, consolidating them into a single database write operation. This approach effectively reduces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



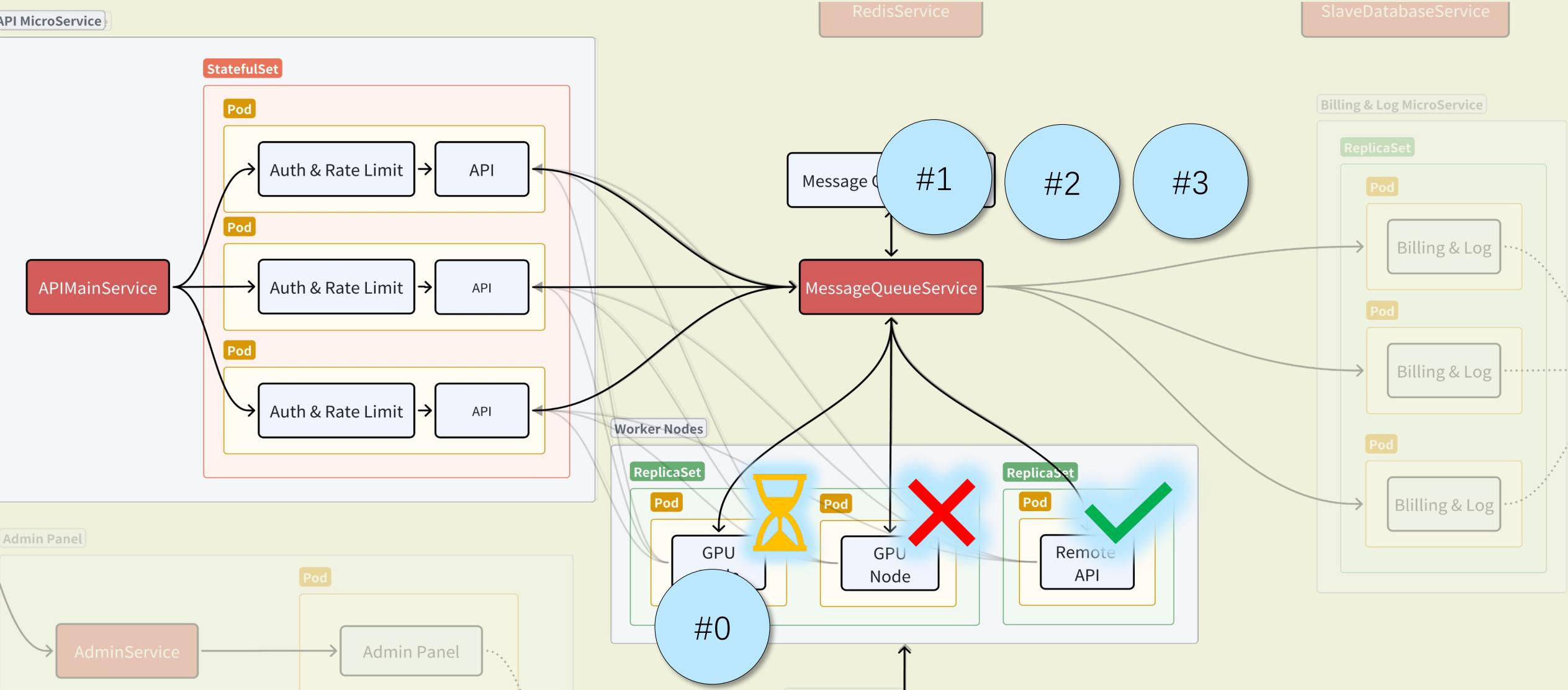
# Leveraging cloud-native design

## Storage and Message Queue

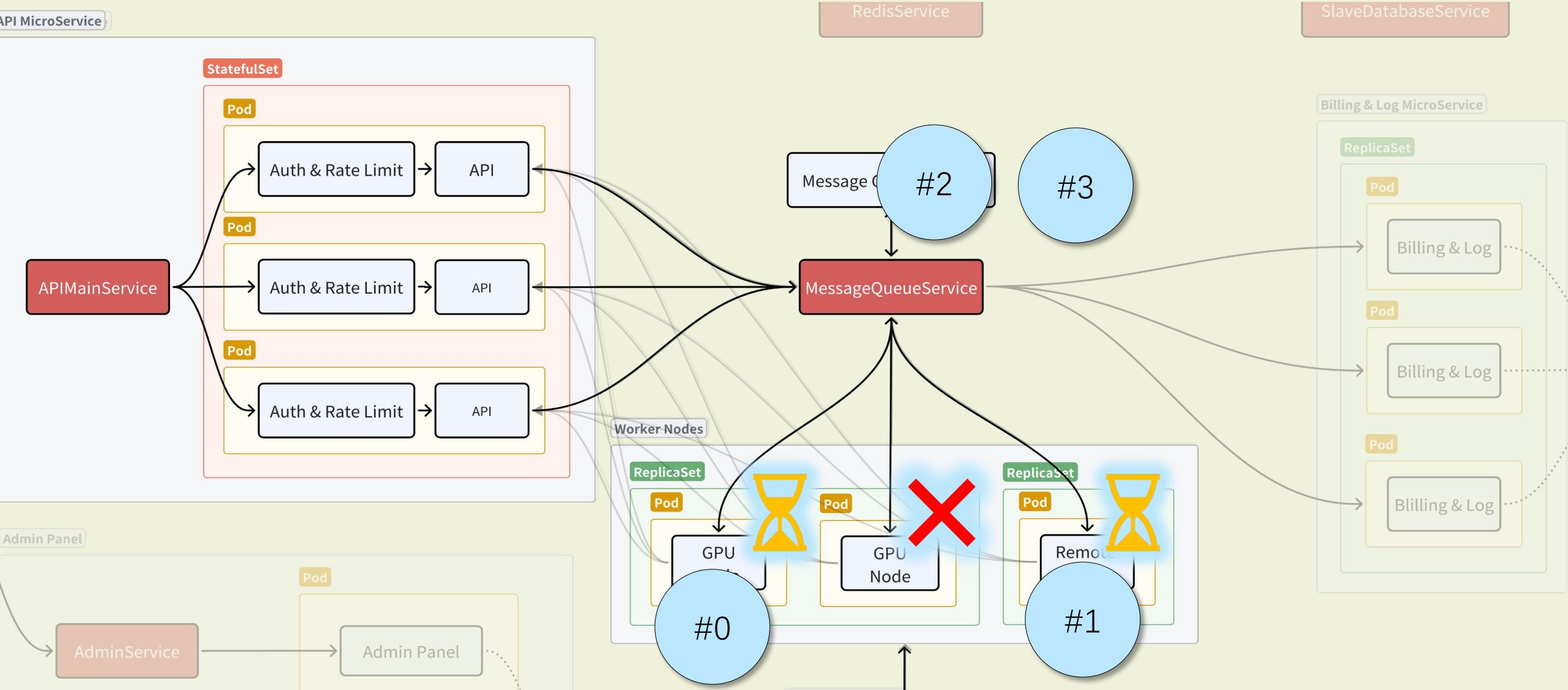
We utilize a PostgreSQL database cluster for our primary storage needs due to its robustness and reliability. It's important to note that throughout the entire API request process, there is no need to write to the database, ensuring that request concurrency and latency are not adversely affected. For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



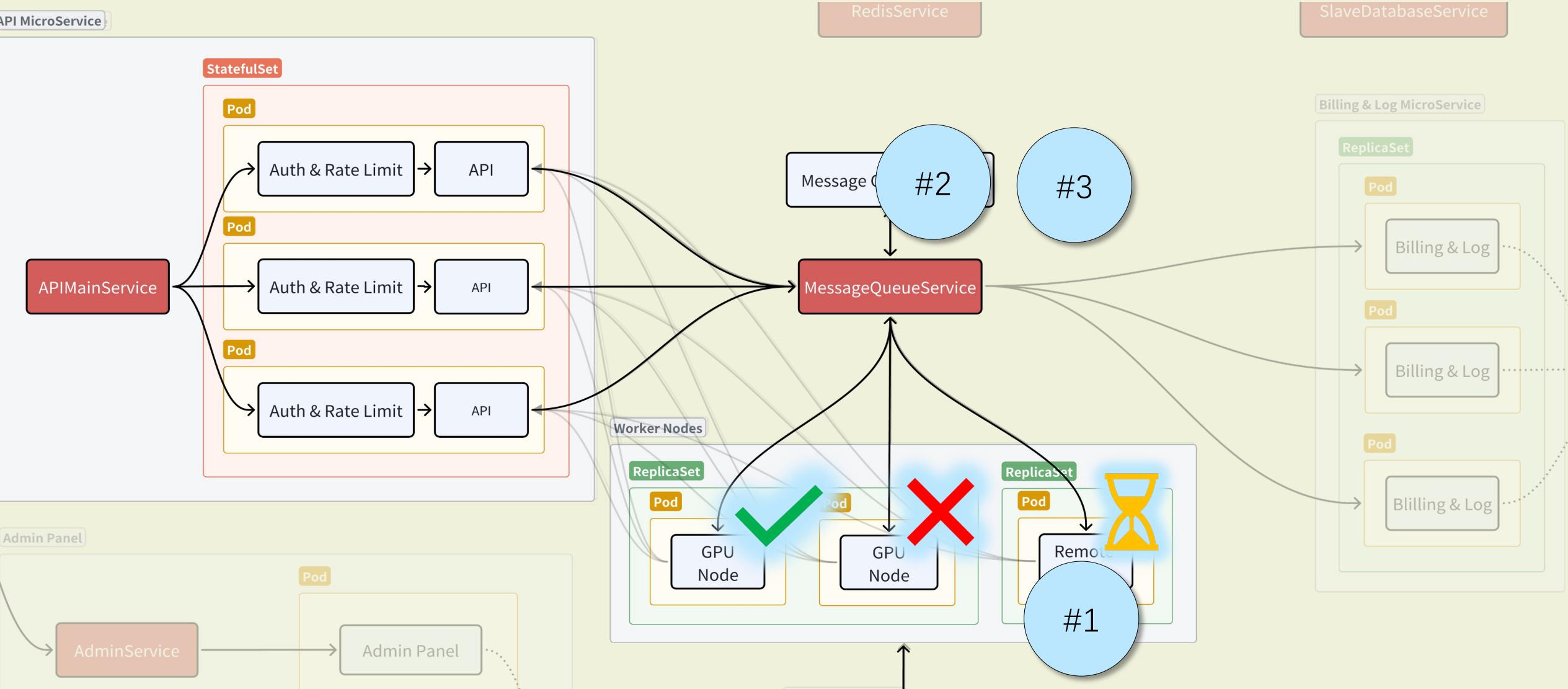
# Intelligent self-serving task distribution



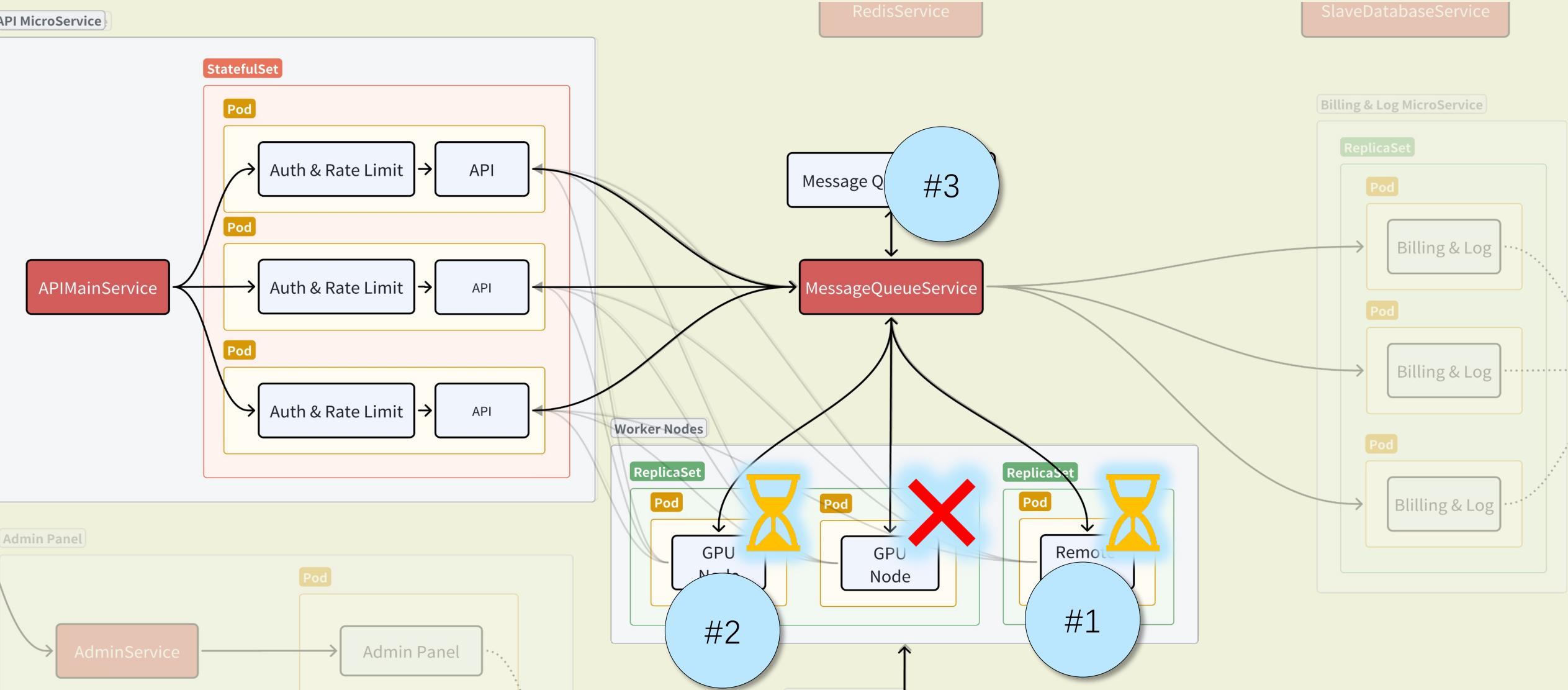
# Intelligent self-serving task distribution



# Intelligent self-serving task distribution



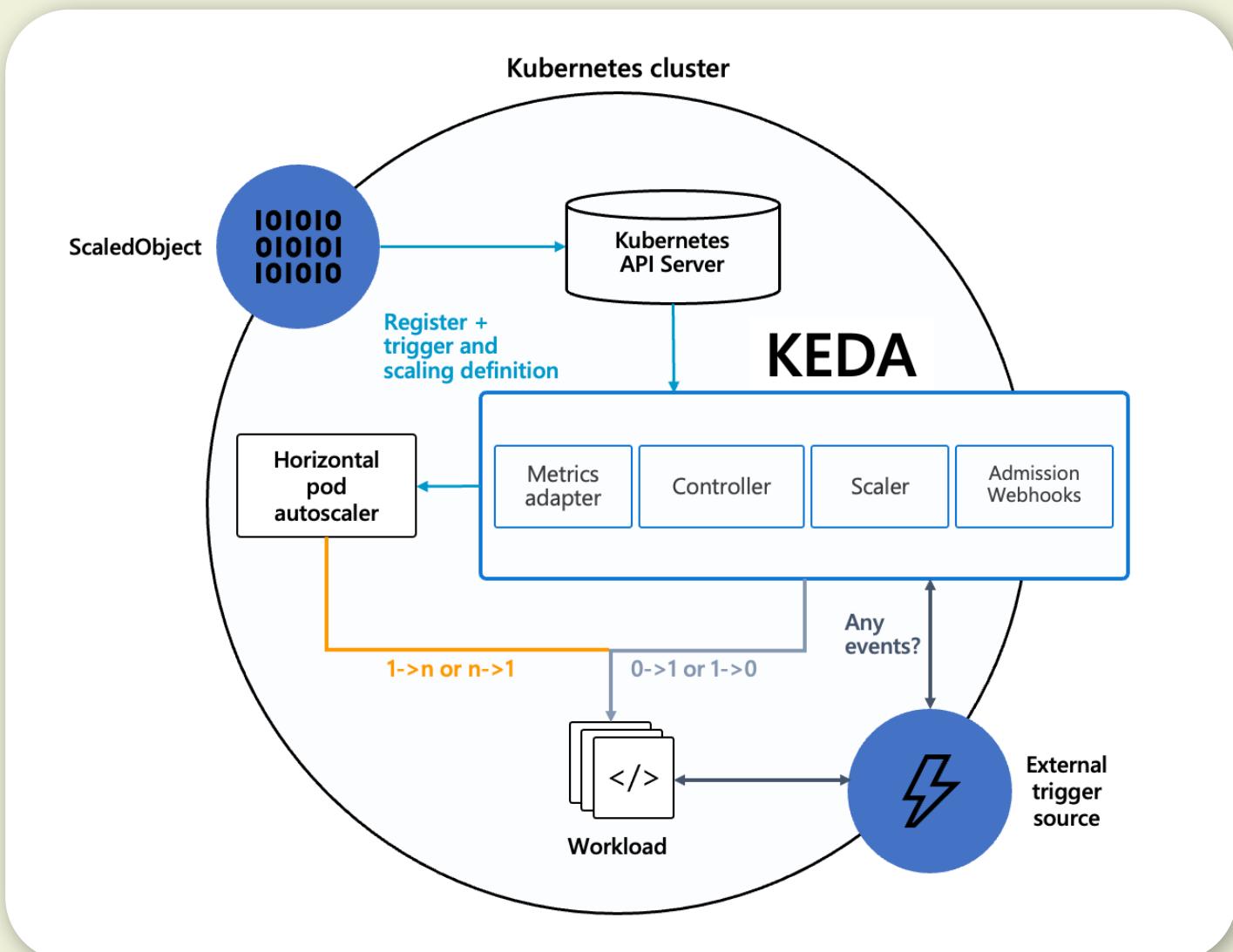
# Intelligent self-serving task distribution



# Optimizing resource allocation

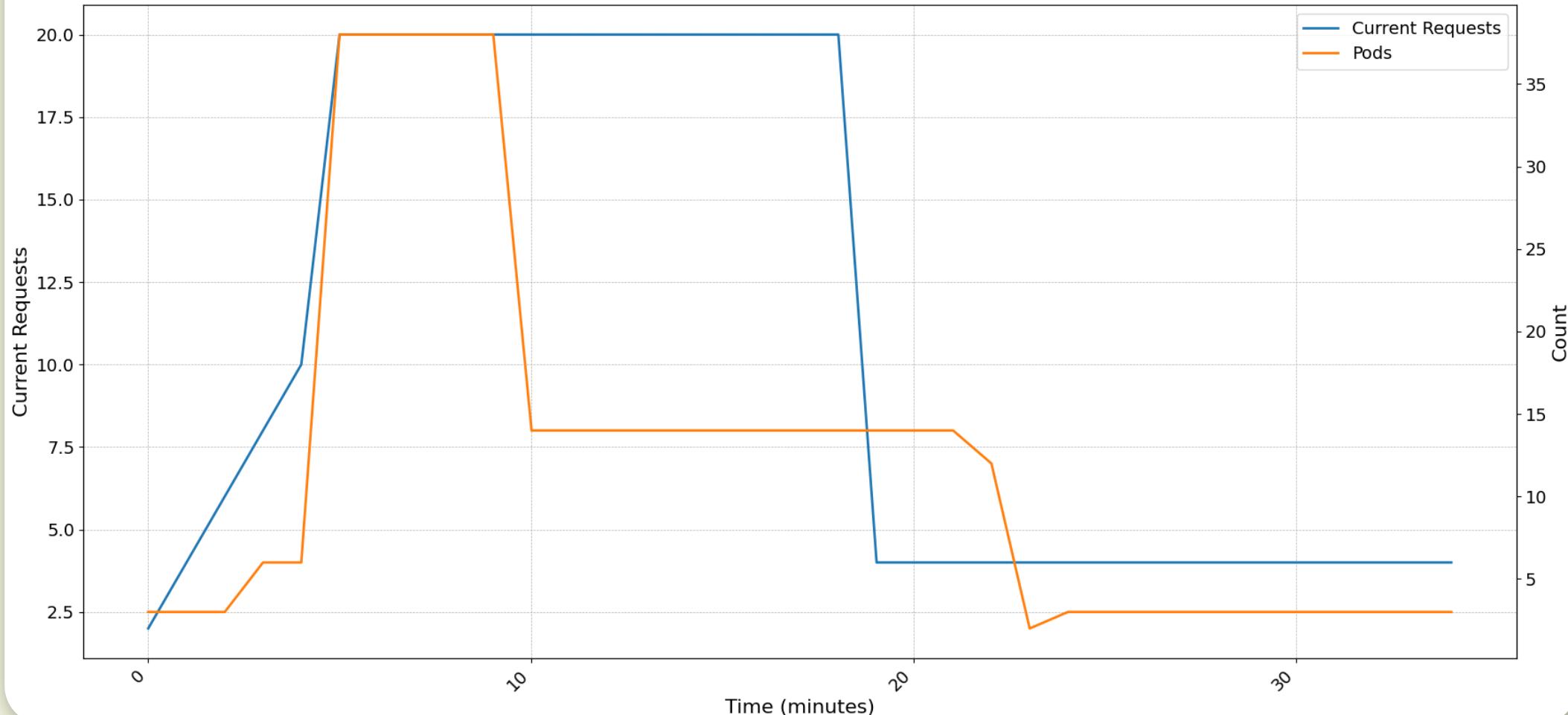


Kubernetes Event-driven Autoscaling



# Optimizing resource allocation

Cluster Metrics Over Time



# Natively Integrated with Kubernetes

---

AI Model CRD  
(Custom Resource Definition)

and

An Operator that manages it  
and implements its  
functionality



The image shows a dark-themed terminal window with a file named "ai-model.yaml" open. The file contains YAML code defining an AI Model Custom Resource Definition. The code includes fields for apiVersion, kind, metadata (name), spec (type, model, replicas, image, maxProcessNum), and three ellipsis dots at the top.

```
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
```

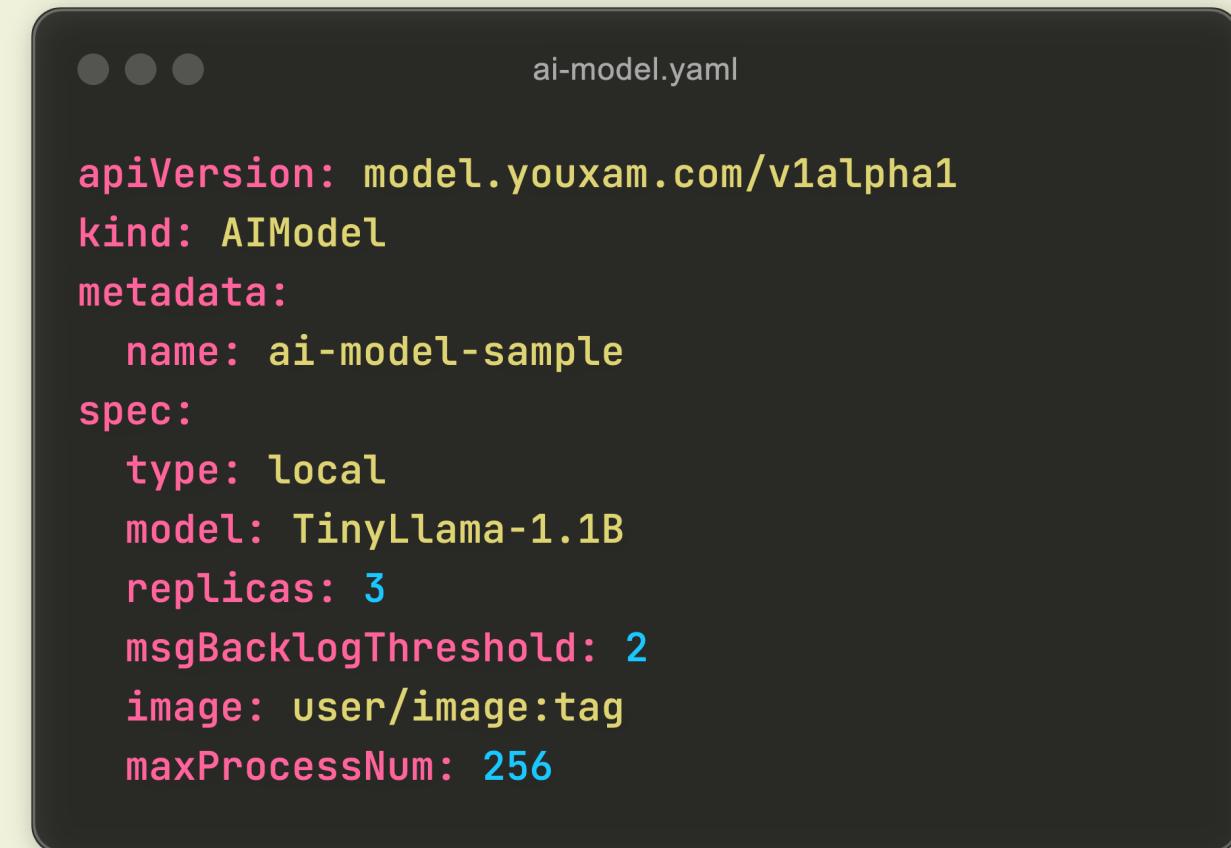
# Natively Integrated with Kubernetes

---

AI Model CRD  
(Custom Resource Definition)

and

An Operator that manages it  
and implements its  
functionality

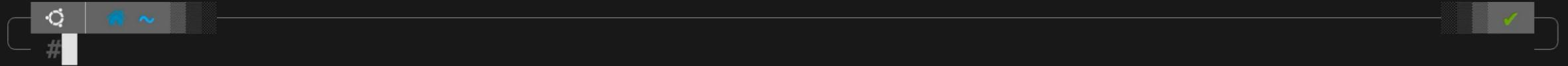


The image shows a dark-themed code editor window with a black background. At the top right, there is a small icon of three dots and the text "ai-model.yaml". The main content area contains a YAML configuration for an AI Model:

```
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

---

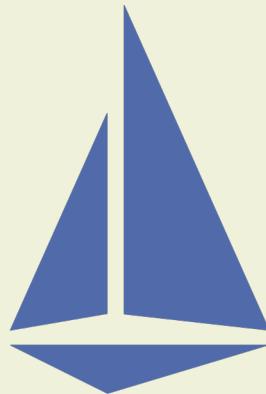


---

# Technology Selection

# Istio

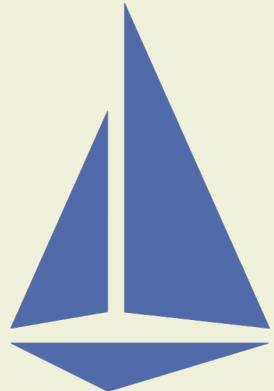
---



# Istio

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.

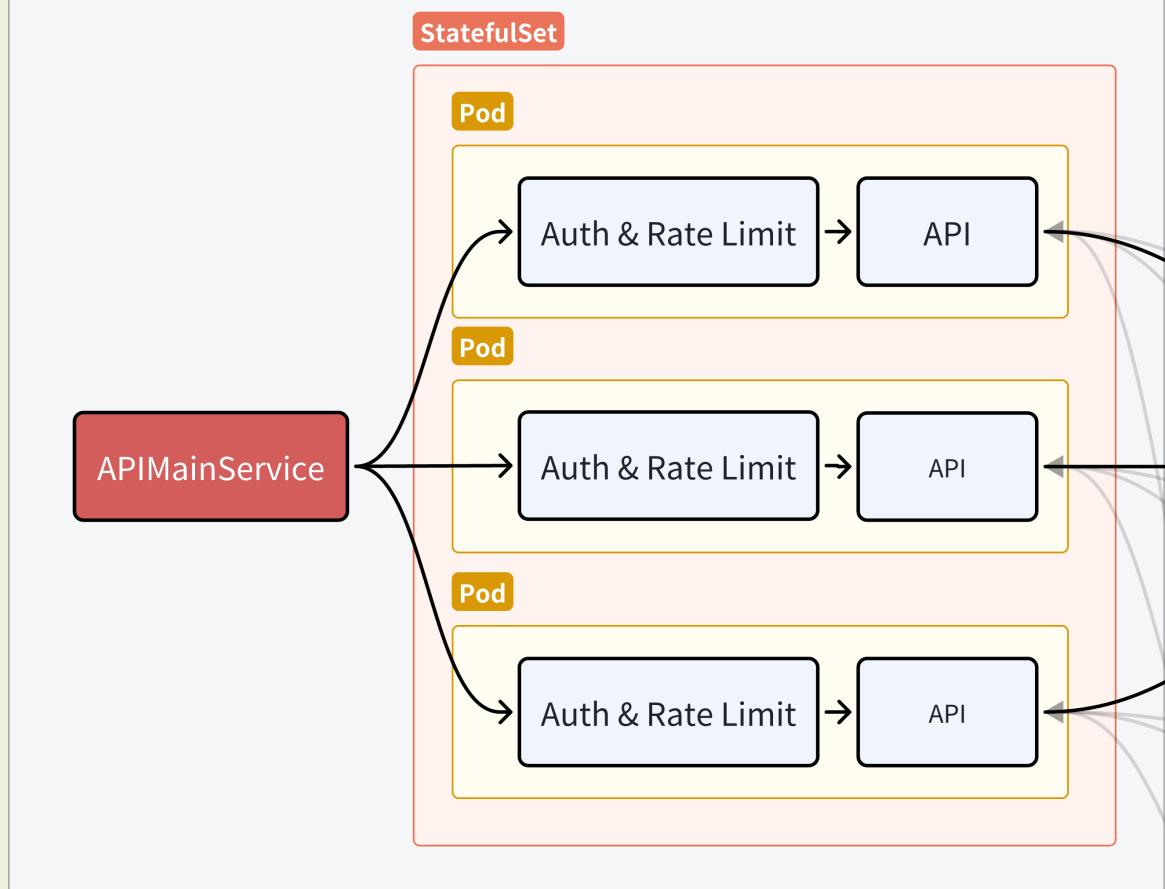
# Istio



# Istio

## Intercepting Requests for Middleware

API MicroService



# Programming Languages

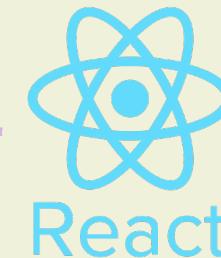
---

Main API Service



AI model operator

Frontend



/ **NEXT.JS**

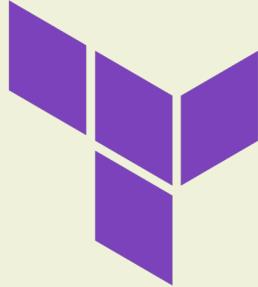
Admin Dashboard

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 - Uplio

uplion.youxam.com

Uploion

Settings GitHub



Welcome to Uploion AI chatbot!

Enjoy the experience of chatting with our AI chatbot!

+ Send a message.

© 2024 Uploion. All rights reserved.

A screenshot of a web browser window showing the Uploion AI Chatbot interface. The page title is "Uplion AI Chatbot - Uploion" and the URL is "uplion.youxam.com". The main content area features a central logo of a lion's head with a circular arrow above it, followed by the text "Welcome to Uploion AI chatbot!" and "Enjoy the experience of chatting with our AI chatbot!". At the bottom, there is a message input field with a placeholder "Send a message." and a send button icon. The browser has a dark theme, and the Uploion logo is located in the top left corner of the main content area.

---

# Conclusion

# Open Source

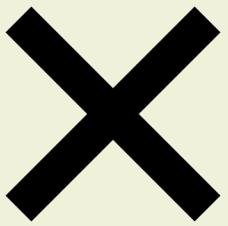
---



<https://github.com/uplion>



**UPLION**



**kubernetes**

Thanks  
For  
Watching

Zhang Haoling



Wang Ruozhu



Cai Yiwen



Zhu Yuanchuan

