

# IntelliDoc

Intelligent Document Automation and Knowledge Enhancement System

Group 6, DATA  
SoC Summer Workshop

# Motivation

GPT, could you help me fix  
the coding error?

**Crowd**Sourcing  
OK, in your code.....

GPT, you're doing it  
wrong!

Sorry, I **made a mistake**, it  
should be.....



GPT, could you help me fix  
the coding error?

OK, in your code.....

GPT, **WHITE** has corrected  
you! It should be.....

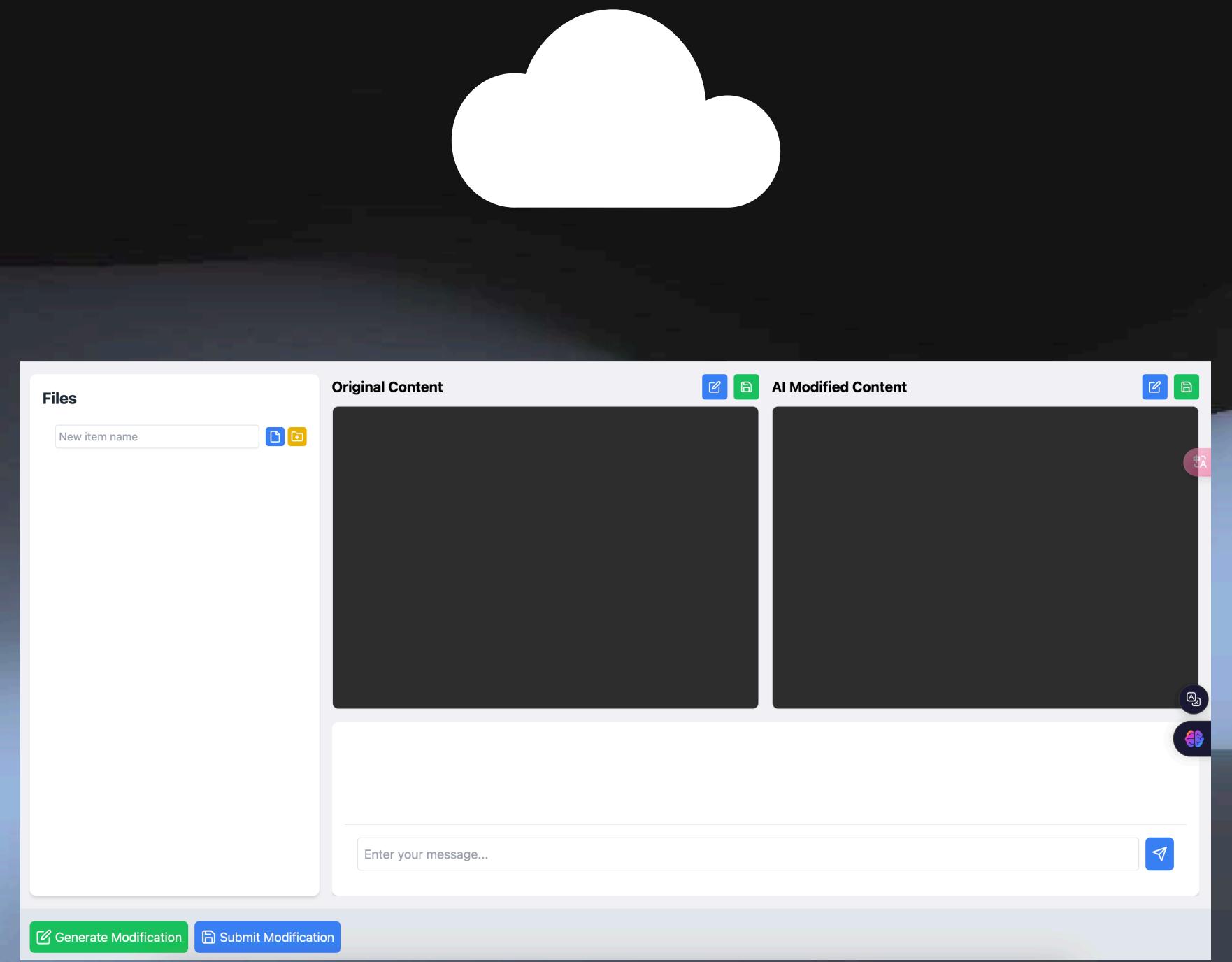
Sorry, I **made a mistake**, it  
should be.....



# System Overview



AI-assisted document editor based on Kubernetes and Cloud

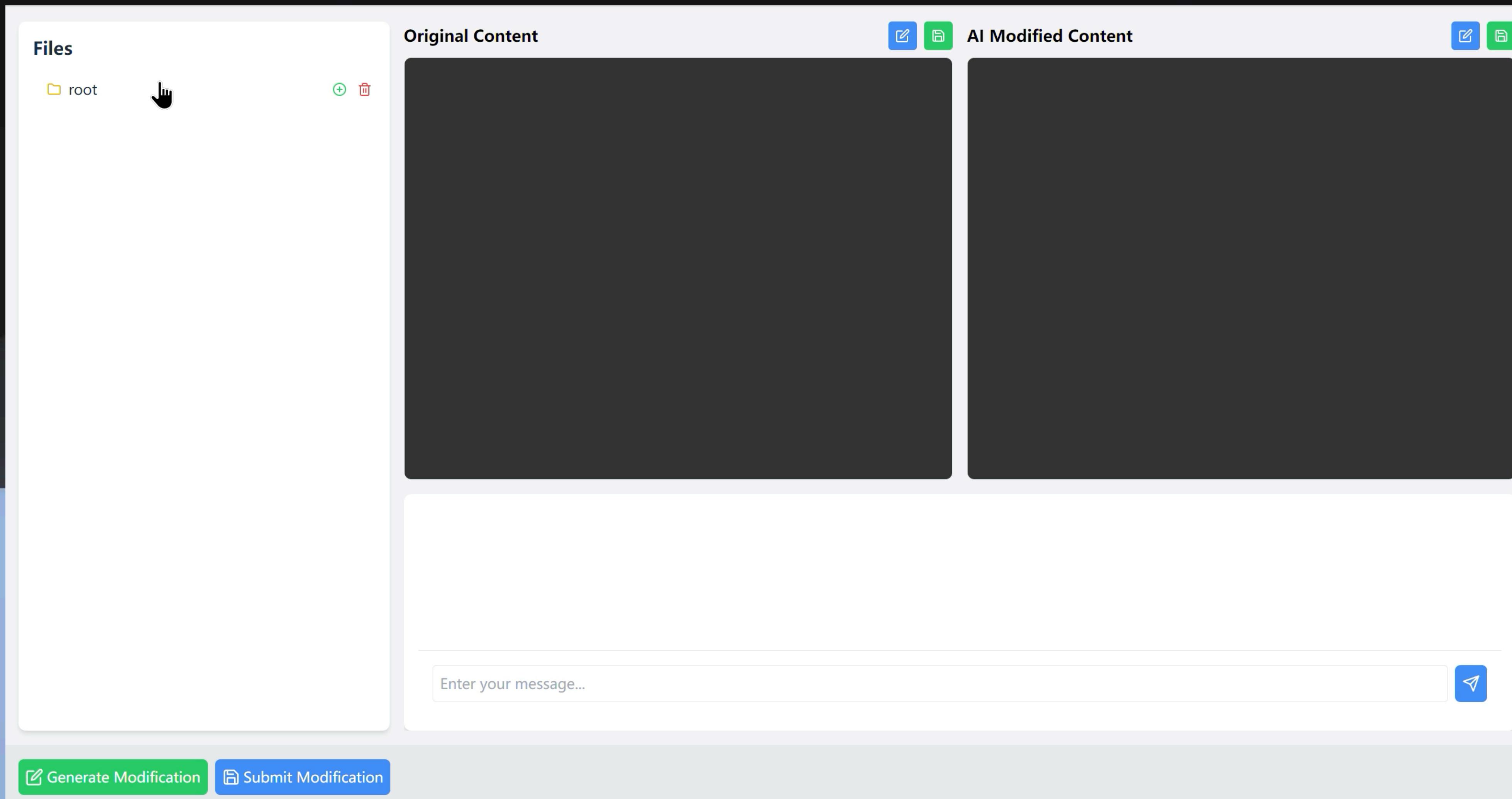


Multiple components for efficient and intelligent document editing

Collecting data when using for a project-tailored AI

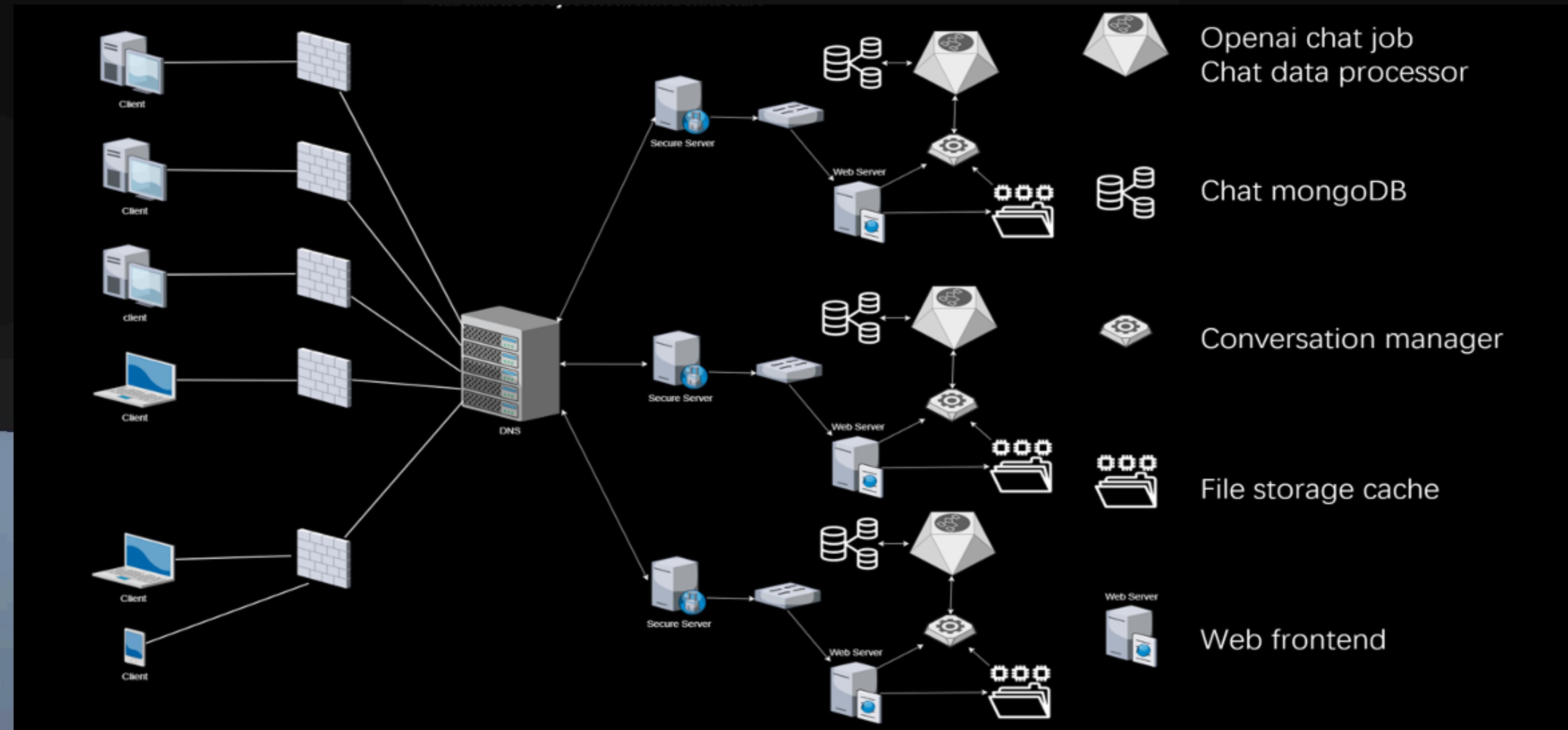


# Demonstration

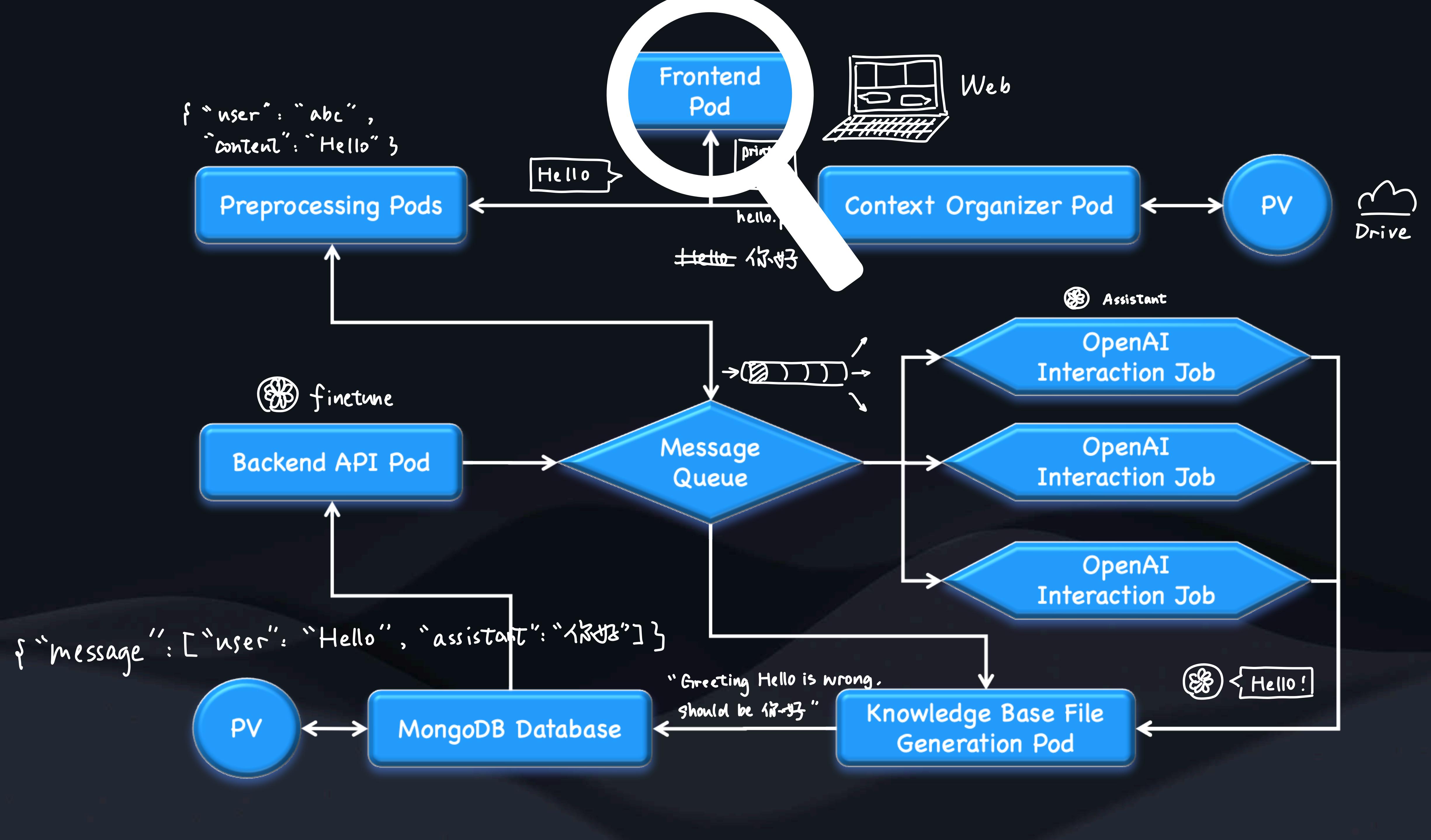


<https://github.com/Cloud-Computing-Group-NUS/Project-Code>

100.24.7.54:32478



Overall pipeline



# Frontend Pod



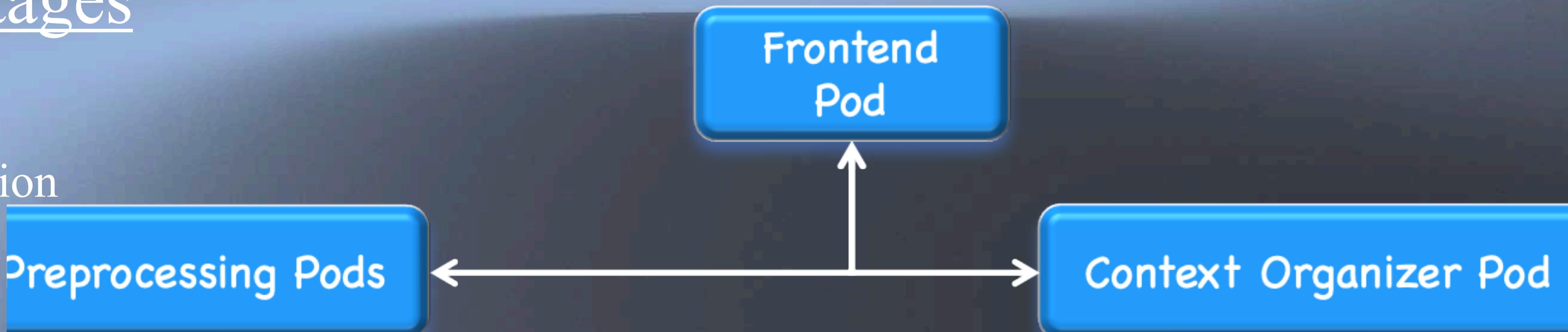
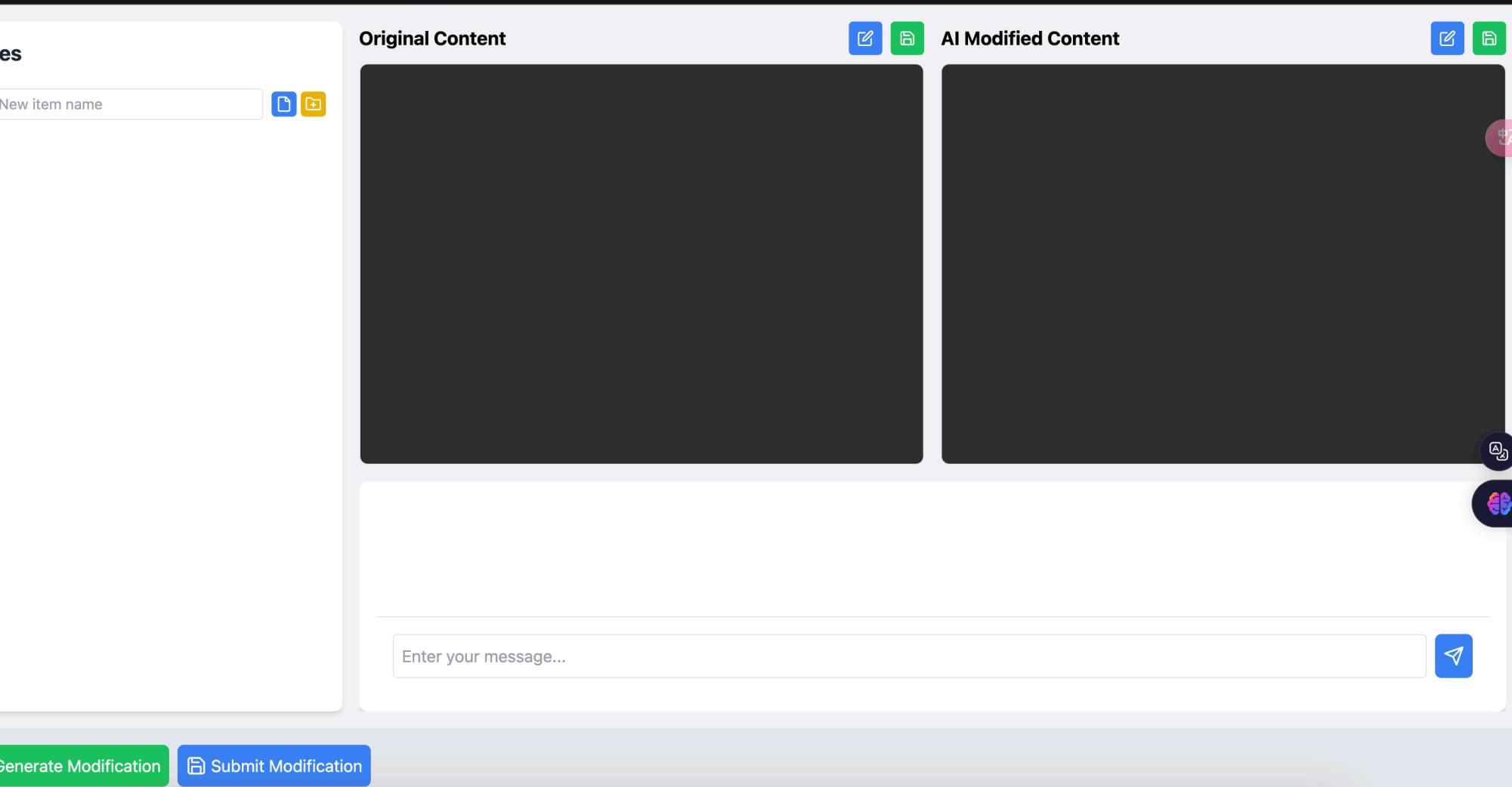
## Function

User interface for document input and editing

Using Nginx

## Design Advantages

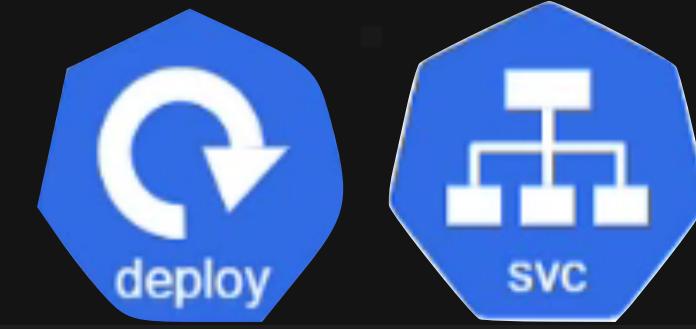
- Responsive
- Real-time collaboration
- Instant feedback
- Preprocessing



# Preprocessing Pods

## Function

Cleanses and formats input data

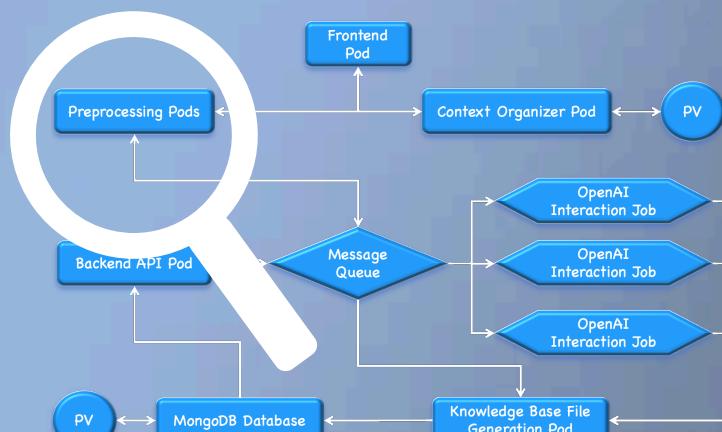


```
@app.route('/get_file_content', methods=['POST'])
def get_file_content():
    """处理文件内容请求"""
    data = request.json
    user = data.get('user', '')
    chat_content = data.get('content', '')
    file_content = data.get('file', '')

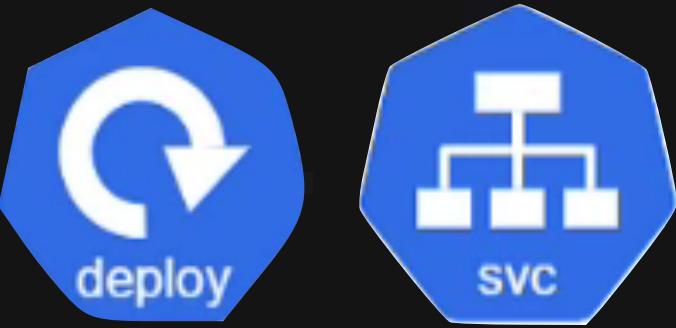
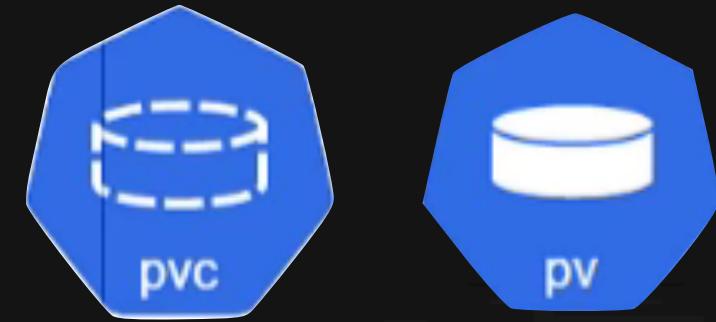
    content = f'以下是你需要修改的文件内容, 请将以下: {file_content}\n以下是我的问题, 请回答: {chat_content}'
```

## Design Advantages

- Parallel processing
- Dynamic scaling
- Flexible logic
- Load optimization



# Context Organizer Pod/ Cloud Drive

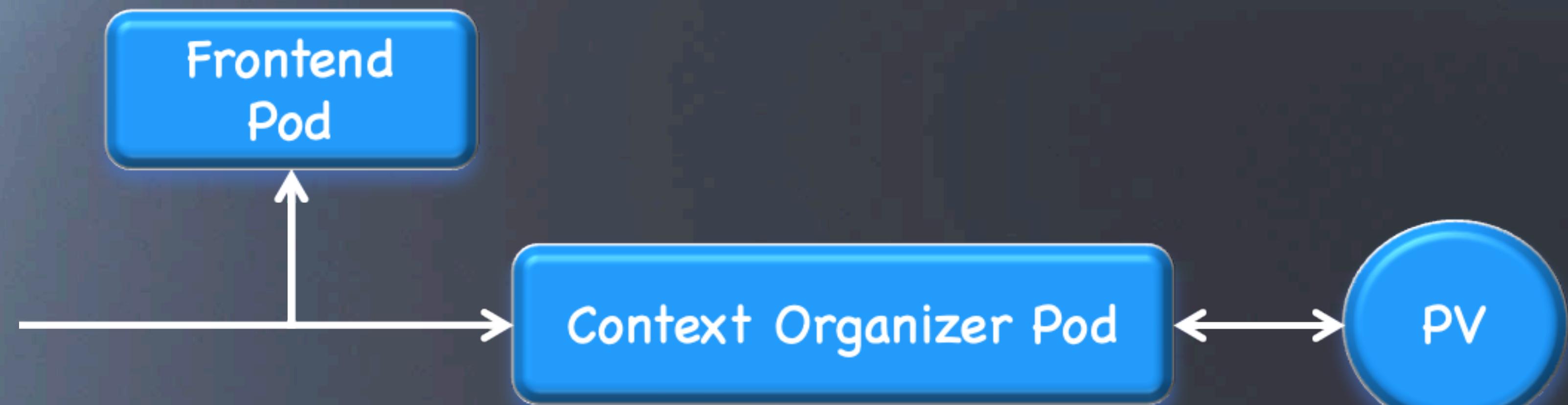
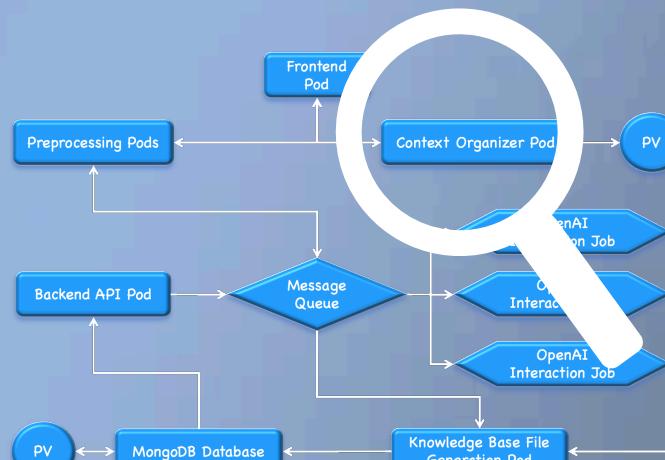


## Function

Formats content for display and interaction

## Design Advantages

- Parallel Processing
- Flexible Formatting
- Adaptive



```
@app.route('/get_chat_content', methods=['POST'])
def get_chat_content():
    """处理聊天内容请求"""
    data = request.json
    user = data.get('user', '')
    chat_content = data.get('content', '')
    file_name = data.get('file', '')

    logger.info(f'Received request for user: {user}, file: {file_name}')

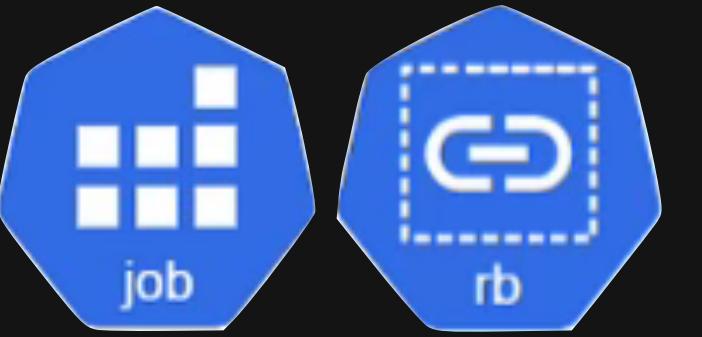
    if not file_name:
        logger.error("File name is missing")
        return jsonify({"error": "File name is required"}), 400

    file_path = os.path.join("/app/data", file_name)
    file_content = read_file(file_path)

    if file_content is None:
        logger.error(f'Failed to read file: {file_path}')
        return jsonify({"error": "Failed to read file"}), 500

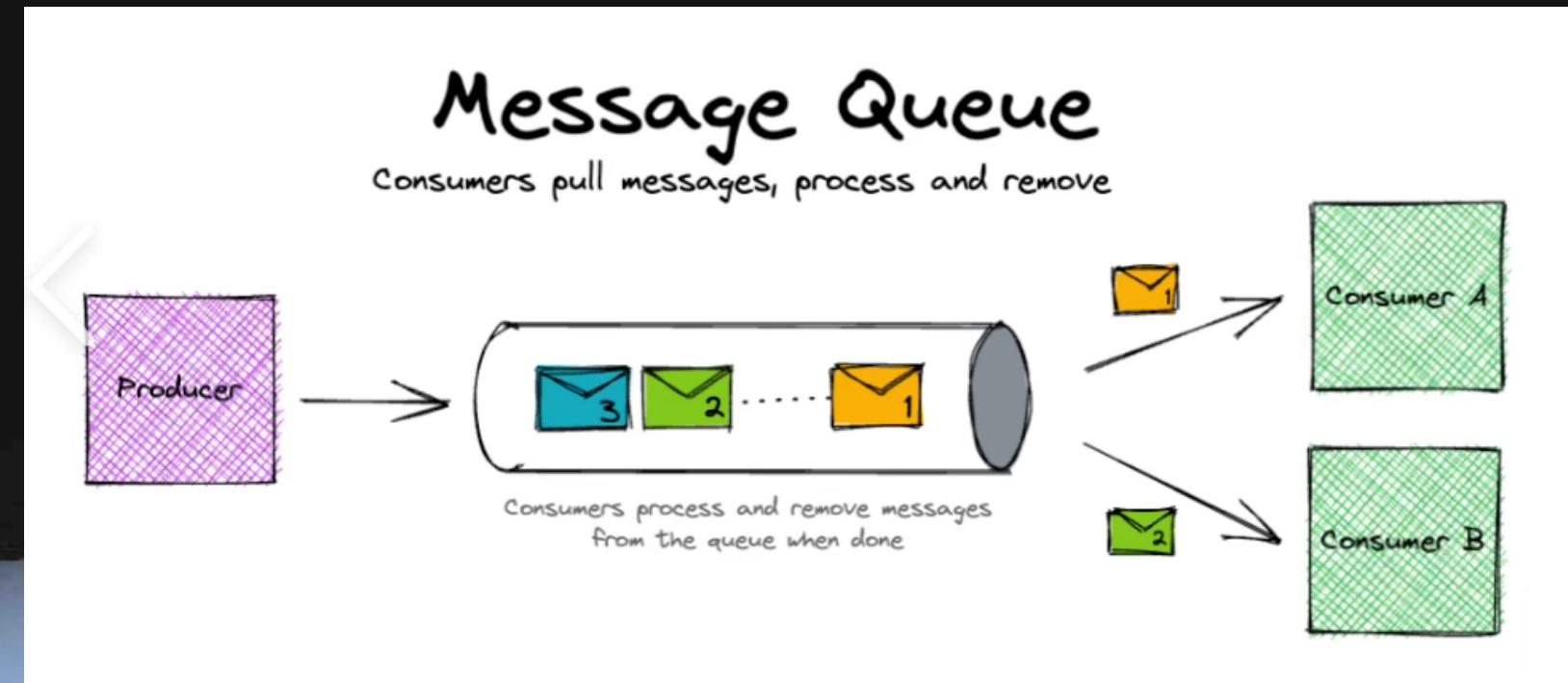
    content = f'以下是与对话相关的文件内容，请基于以下文件内容做出接下来的回答: {file_content}\n以下是我的问题,请回答:{chat_content}'
```

# Message Queue



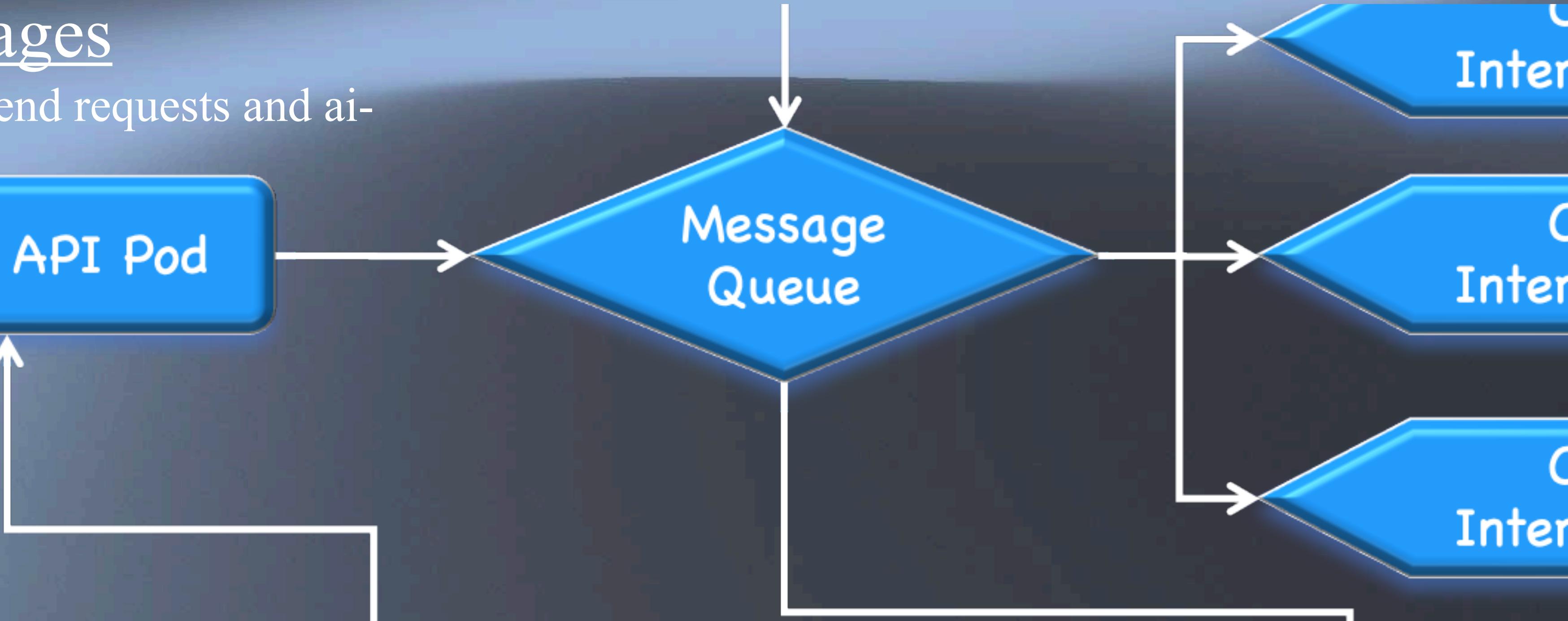
## Function

Temporary data storage and transmission

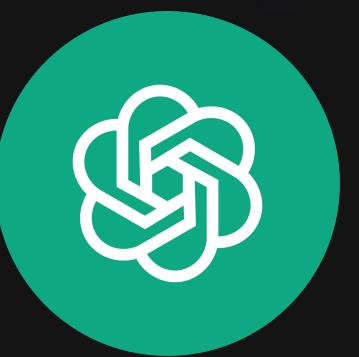


## Design Advantages

- Decoupling of frontend requests and ai-agent response
- High Throughput
- Data-Persistence
- Resource saving
- 



# OpenAI Interaction Job



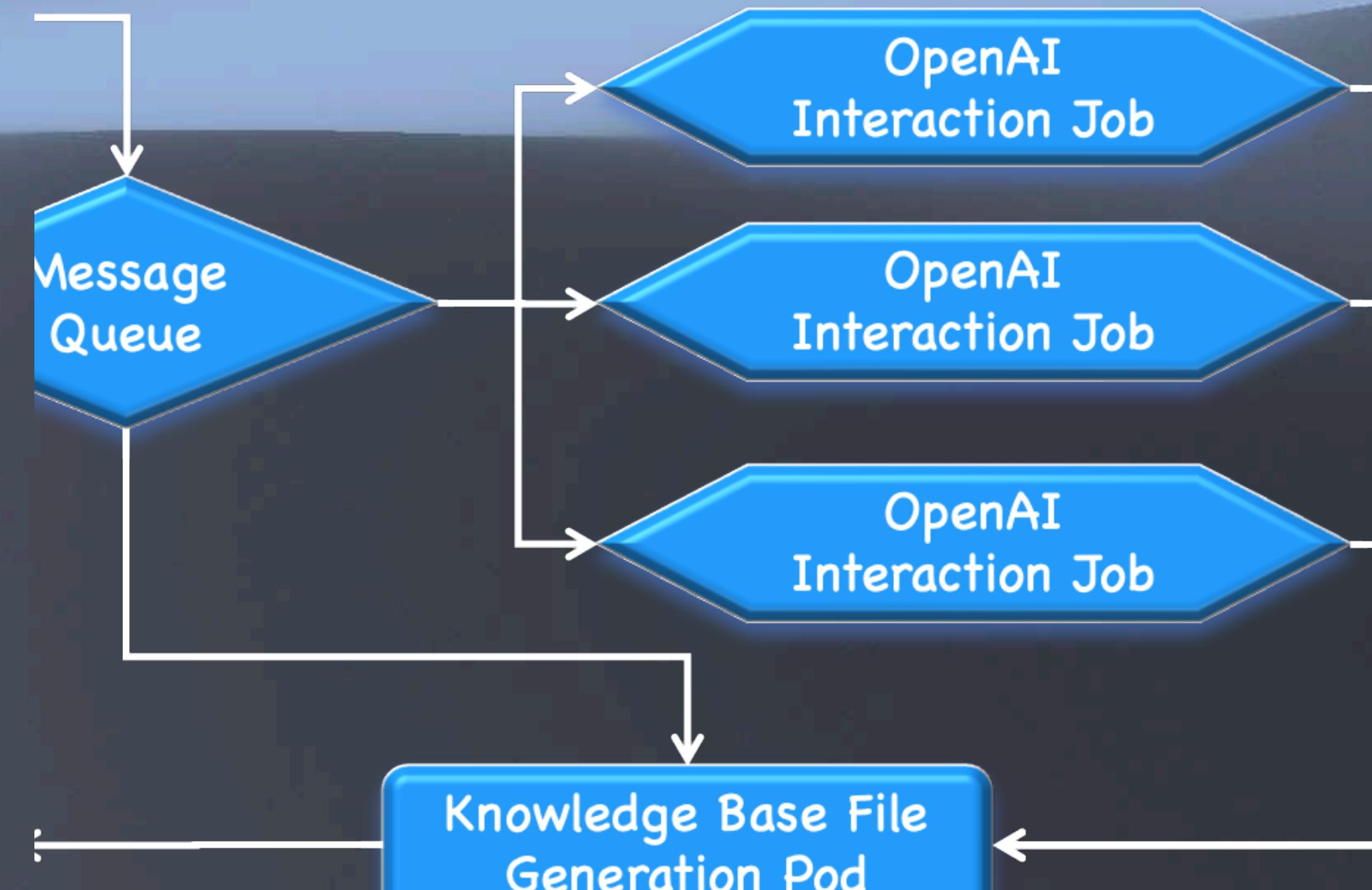
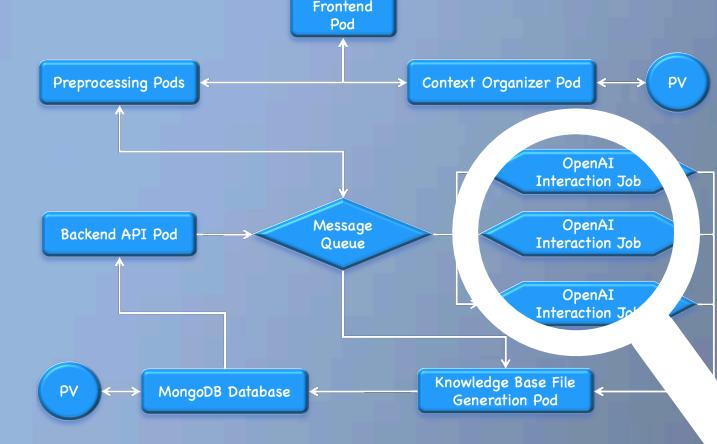
## Function

Uses AI-assistant to handle requests

```
def create_assistant(model_id="gpt-3.5-turbo"):
    assistant = client.beta.assistants.create(
        name="Code Tutor",
        instructions="You are a personal coding tutor. Fix the coding problem with the given code.",
        model=model_id,
    )
    return assistant
```

## Design Advantages

- Concurrent Handling
- Intelligent API Management
- Failover Mechanism
- ‘Secrets’ to protect API KEY





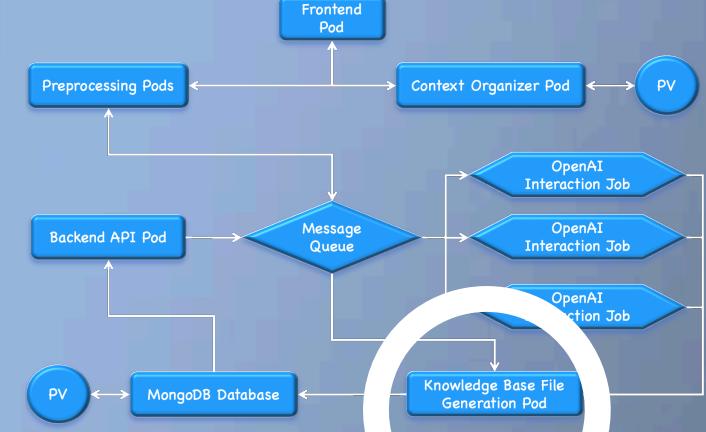
# Knowledge Base File Generation

## Function

Create structured knowledge files

## Design Advantages

- Automated Extraction
- Incremental Updates
- Versatile Usage



Original Content

```
#include<iostream>
using namespace std;
int mian()
{return 0;}
```

AI Modified Content

```
在给定的代码中存在两个问题。首先，`#include<iostream>` 应该更正为 `#include <iostream>`。其次，`mian()` 应更正为 `main()`。  
修正后的代码如下：
```

```
```cpp
#include <iostream>
using namespace std;

int main()
{
    return 0;
}
```
```

代码运行错误，应该如何修正

在这段代码中，有两个问题：`#include<iostream>` 应该改为 `#include <iostream>`，另外`mian()` 应该被改为 `main()`，修正后的代码如下：```cpp #include <iostream> using namespace std; int main() { return 0; }```

Enter your message...

Generate Modification Submit Modification



# MongoDB Database

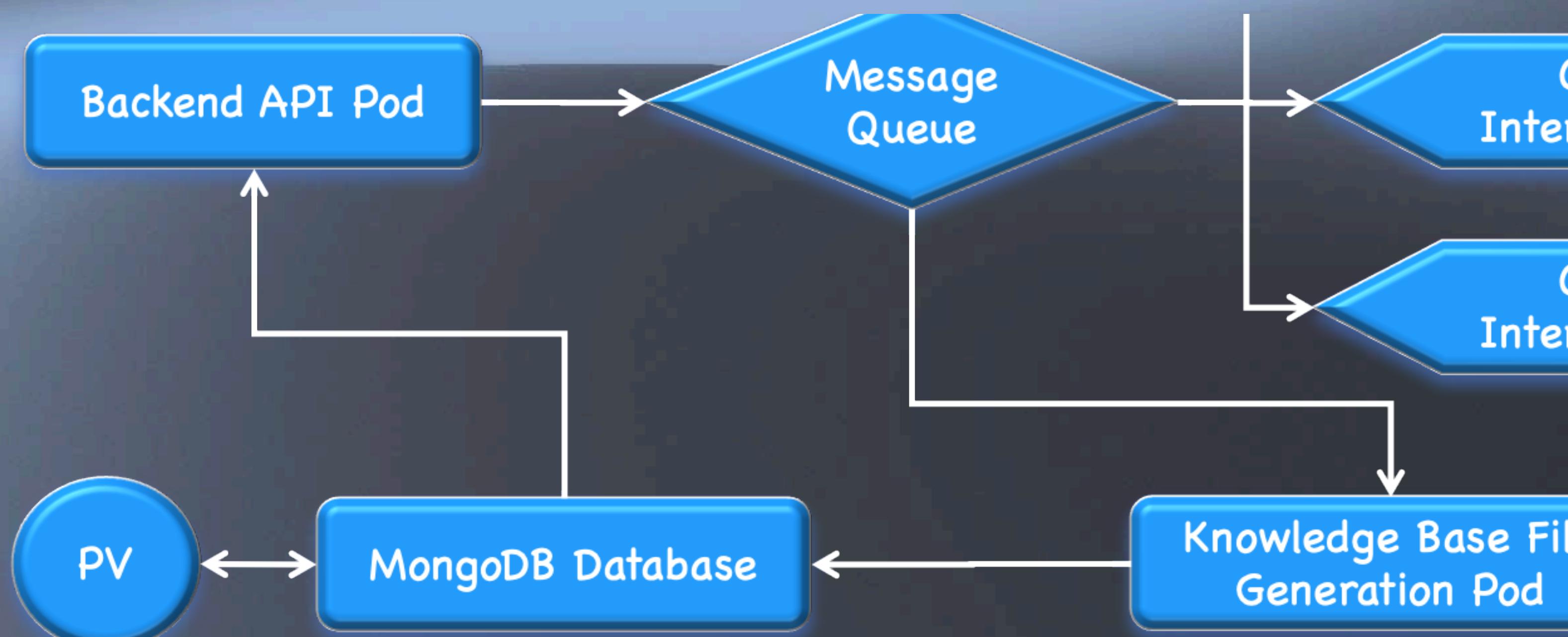
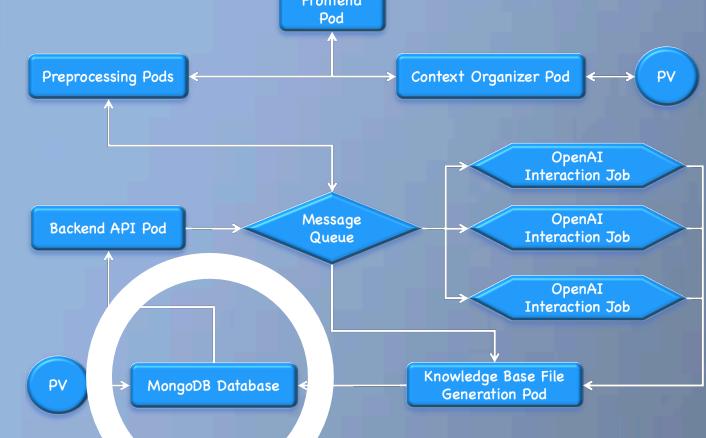


## Function

Persistent document storage

## Design Advantages

- Flexible Storage
- Large-Scale
- Fast Querying
- Powerful Aggregation



# Backend API Pod

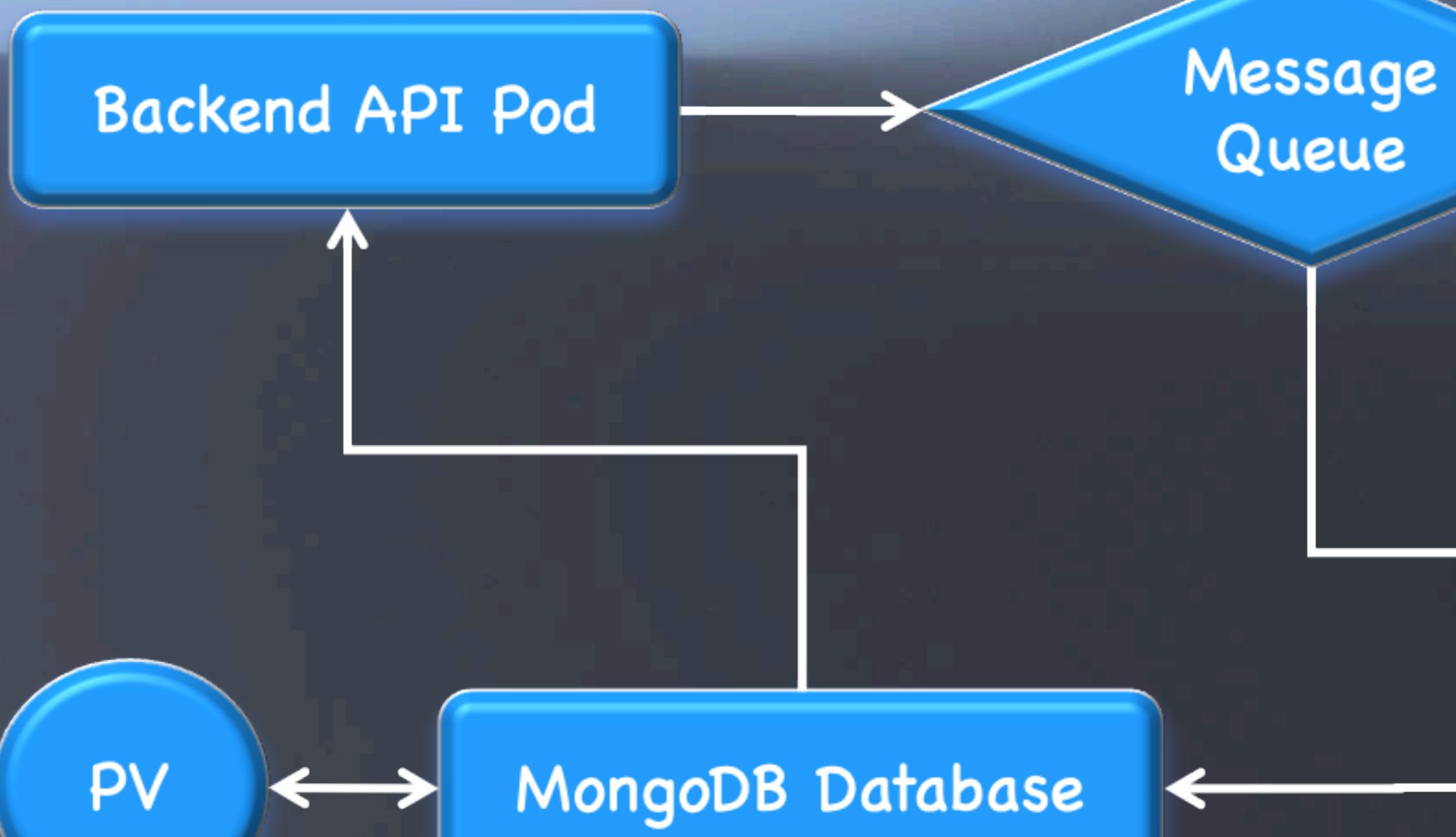
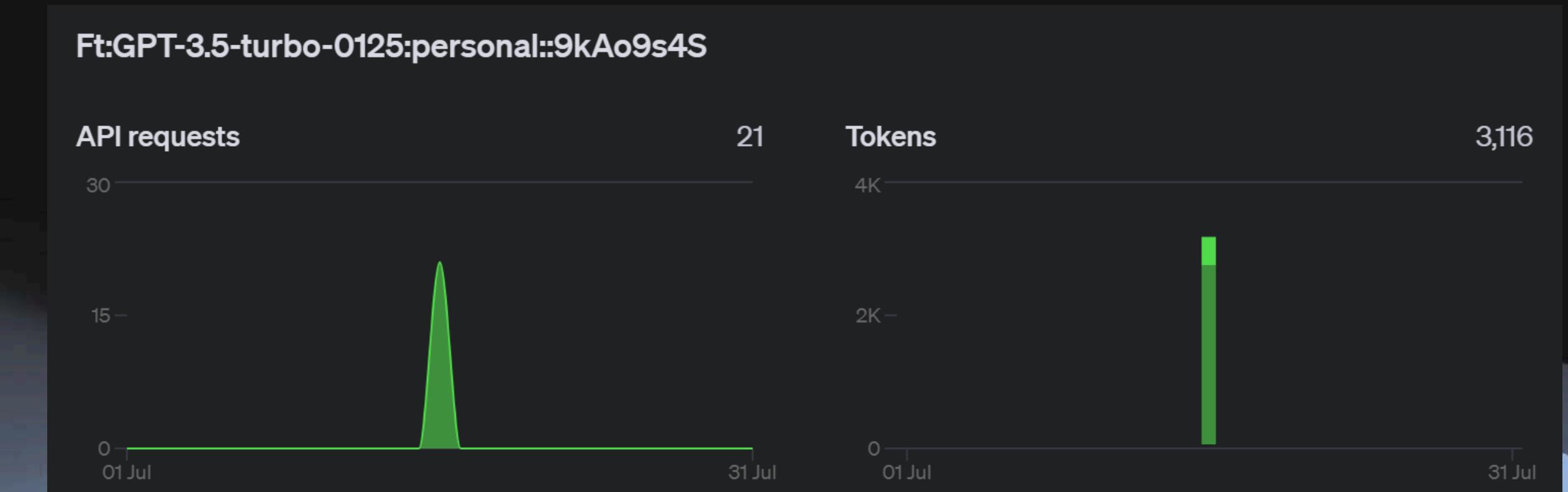


## Function

Uses correction messages from mongoDB Database to finetune a new model

## Design Advantages

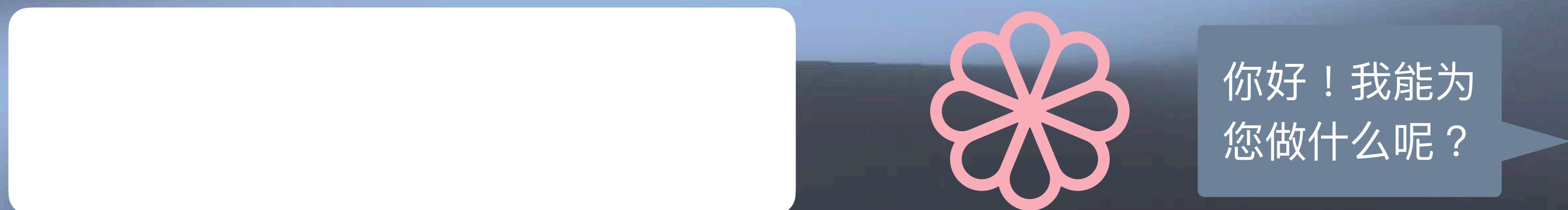
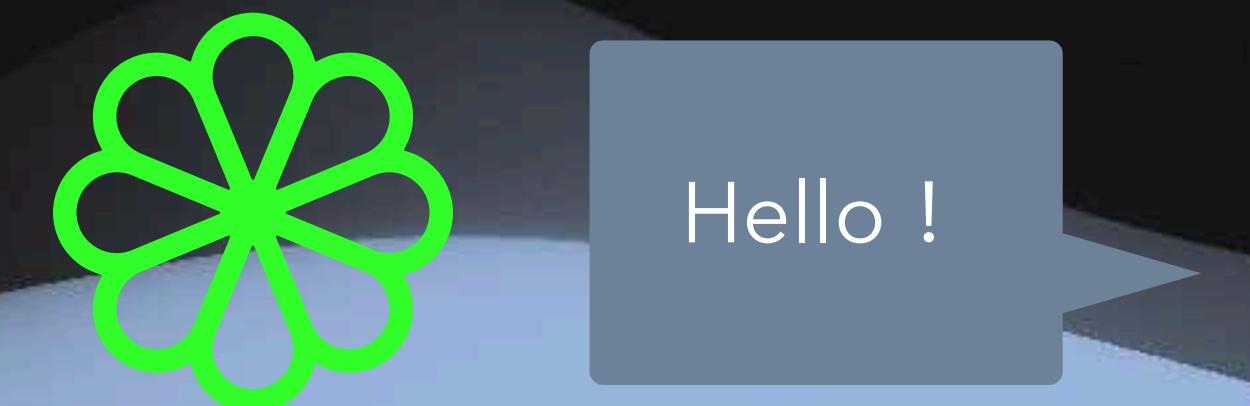
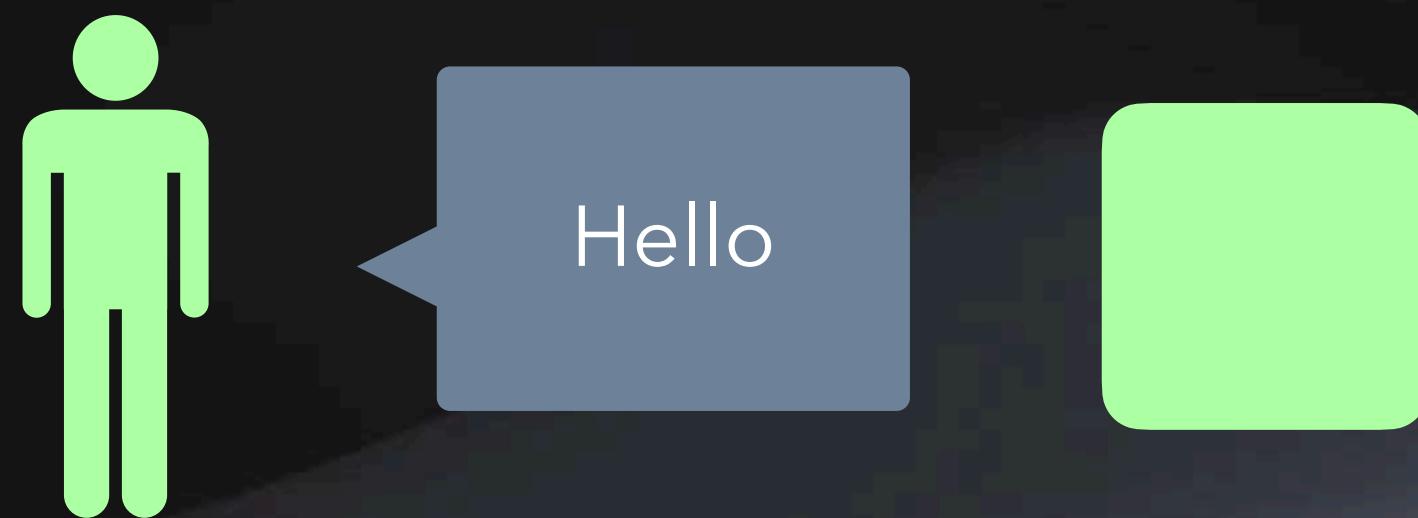
- High Concurrency
- Horizontal Scaling
- Caching
- Stateless



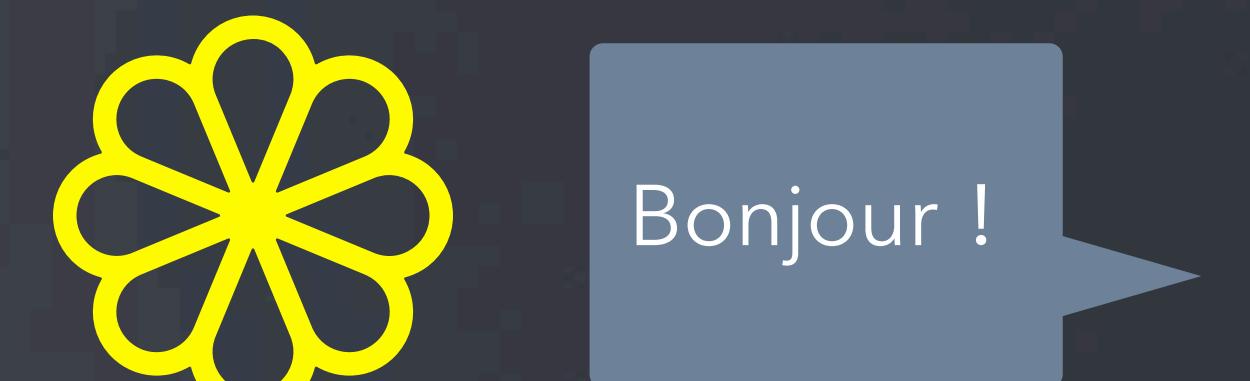
Fancy Design

Message Queue

# Message Queue



robust  
scalable  
cloud-native



# Design Advantages

## Technically

- High Scalability
- Robust Real-time Processing Capabilities
- Flexible Data Processing Workflow
- High Availability and Fault Tolerance
- Optimized Performance

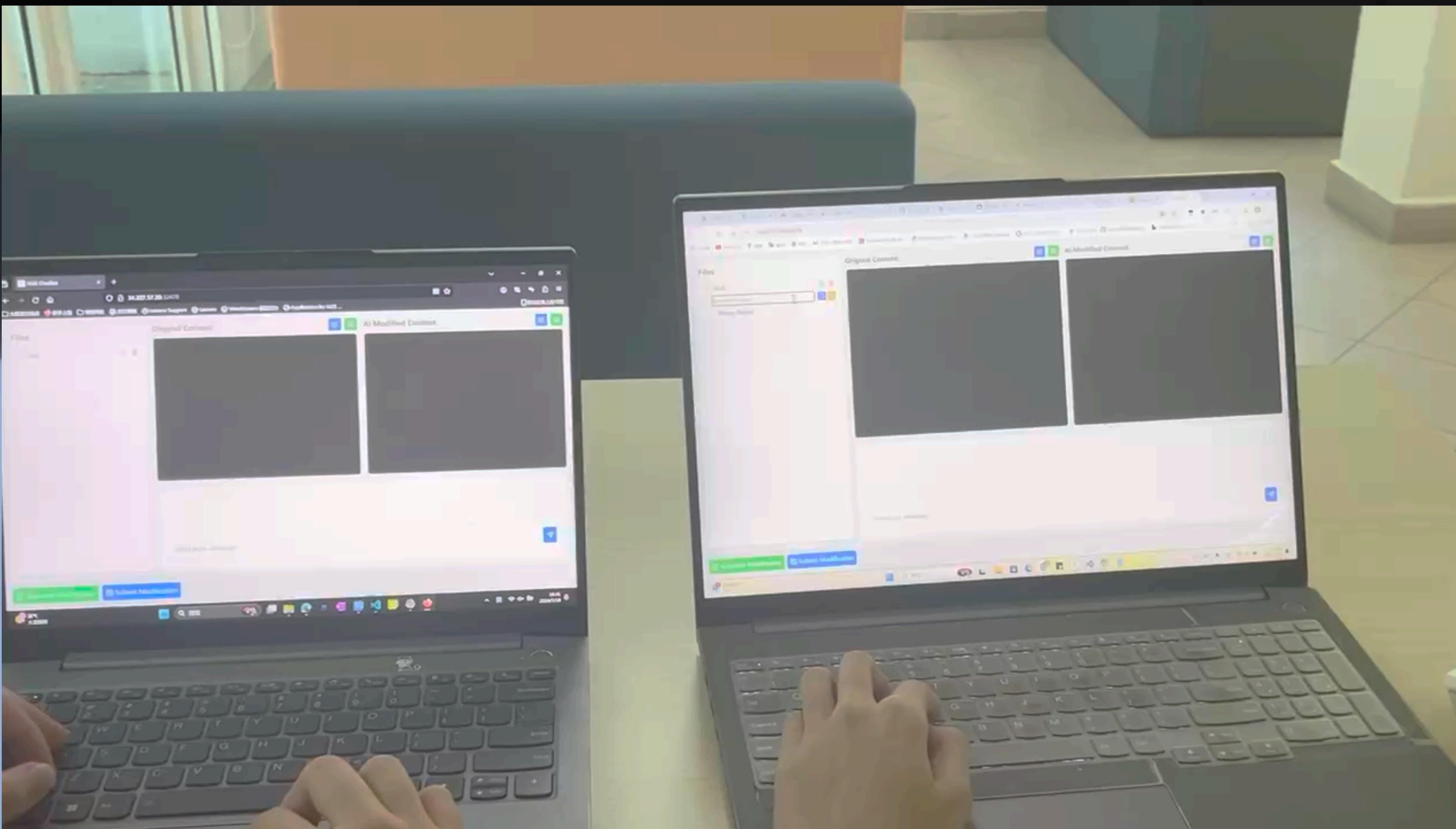
Only CLOUD can do

## UX Layer

- Security
- Intelligence
- Cost Efficiency
- Developer and Maintenance Friendly
- Optimized User's Experience

# DEMO

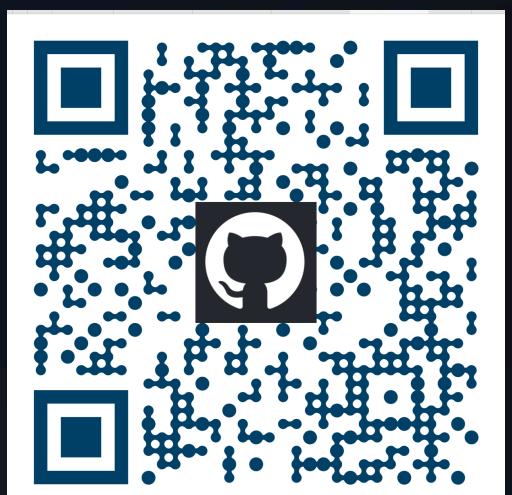
Supports up to 50 (theoretically) collaborators online



HTTP: 100.24.7.54:32478

## Warning:

- First **create a folder of your name**, then feel free to edit and create under your workspace
- For naming the folders, please follow the standard **lowerCamelCase**
- **Don't send too many** requests at one time, as every token counts
- **Open exactly one webpage** at the same time
- First **select a file**, then ask questions related to the file
- Legal file types: **.cpp, .py, .md**(with additional rendering)
- First time set up **may be slow**, wait patiently
- **Have fun!**



# Future Work

“A general Cloud-native Finetuned-AI agent Architecture”

Group chat summary

File summary

Private database

# Thank you

<https://github.com/Cloud-Computing-Group-NUS/Project-Code/tree/main>

Group 6, DATA  
SoC Summer Workshop

