ChatGPT

SecurePipe: Real-Time LLM Gateway with AI-Powered Security Filtering

Overview

SecurePipe is a modular, real-time gateway service that inspects and processes data sent to large language models (LLMs), ensuring security, compliance, and safe usage of generative AI systems. It provides traffic filtering, sandboxing, AI-based content analysis, and intelligent forwarding to external LLMs such as OpenAI or Anthropic.

This service is designed to simulate a production-grade environment with strong emphasis on:

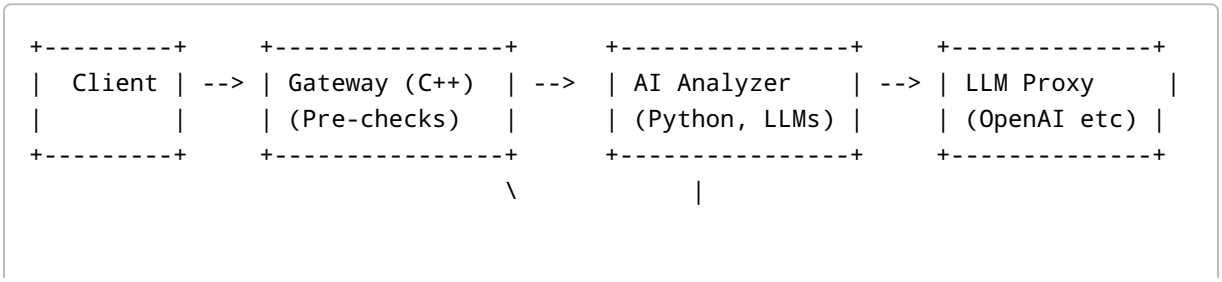
- Security and data integrity
- AI-enhanced policy enforcement
- CI/CD and team-oriented development practices
- Hybrid development using C++, Python, and Docker

SsecurePipe is ideal for organizations looking to deploy or gate access to LLMs in a secure, auditable, and controlled fashion.

Key Features

- **Secure Input Inspection**: Text and file inputs are pre-validated using C++ for speed and safety.
- **AI Content Filtering**: Python-based service classifies content using LLMs and heuristics.
- **File Sandbox Execution**: Suspicious files are executed in Docker sandbox environments and analyzed.
- **Forwarding Gateway**: Approved content is forwarded automatically to external LLMs.
- **Audit Logging**: Full traceability of input, analysis, and output decisions.
- **CI/CD Integration**: GitHub Actions pipeline for testing, building, and security checks.

System Architecture



```
                                    \--> Sandbox Runner (Docker)
                                          (optional for files)
```

Components

Component	Language	Description
Gateway API	FastAPI / C++	Entry point for client requests; validates and routes traffic
Pre-Check Agent	C++	Validates input structure, size, and type quickly
AI Analyzer	Python	Uses LLM APIs or local models to analyze sensitive content
Sandbox Runner	Python + Docker	Executes uploaded files in isolation to detect malicious behavior
Forward Proxy	Python	Sends approved content to external LLMs and returns responses
CI/CD Pipeline	GitHub Actions	Tests, linting, Docker builds, and security scans
Logging & Auditing	SQLite / Redis	Logs each step of the request lifecycle

Technologies Used

Tool	Purpose
C++	High-performance pre-validation of requests
Python	Business logic, AI integration, and sandbox control
FastAPI	API server and WebSocket support
Docker	Containerized microservices and sandboxing
ClamAV	Optional AV scan on uploaded files
OpenAI API / LLaMA.cpp	AI classification engine
GitHub Actions	Continuous Integration and automated testing
Docker Compose	Multi-service orchestration for local and production deployment

API Workflow

1. **Client Submits Request**
2. Via REST or WebSocket

3. Payload includes input type (text or file), user ID, and LLM target
4. **Gateway Pre-Checks**
5. Validate size, structure, format
6. Calculate hash and compare with cache
7. **AI Analysis**
8. Text: sent to classification model (e.g., OpenAI/GPT-4)
9. File: basic metadata inspection
10. **Optional Sandbox**
11. File executed in Docker container with resource limits
12. Observes behavior (file writes, network access)
13. **Verdict Engine**
14. If safe → forward to LLM
15. If flagged → respond with reason
16. **Forwarding & Response**
17. Forwarded to selected LLM (proxy)
18. Return answer to client

Security Considerations

- Input validation and strict file type enforcement
- Isolation via Docker containers for sandboxing
- Encrypted API traffic (HTTPS, JWT optional)
- Access control policies for users
- Full audit logs for post-mortem analysis

Deployment

- Docker Compose for full environment (gateway, AI, sandbox, Redis, database)
- Environment variables for model keys, sandbox limits, etc.
- CI/CD auto-deploys to staging/dev via GitHub Actions

Project Structure (Proposed)

```
securepipe/
├── api/                # FastAPI server + endpoints
├── gateway_cpp/        # C++ validation agent
├── analyzer/           # Python LLM and content classification
├── sandbox_runner/     # Docker-based file executor
├── forwarder/          # LLM proxy clients
├── tests/              # Unit & integration tests
├── .github/workflows/  # CI/CD pipelines
```

```
├── docker-compose.yml
└── README.md
```

Roadmap (Suggested Phases)

1. 🔘 Build REST API for text input
2. 🔘 Implement AI filtering with OpenAI API
3. ○ Add C++ validation service
4. ○ Enable file uploads and AV scan
5. ○ Add sandbox executor for files
6. ○ Integrate full LLM forwarding pipeline
7. ○ Add WebSocket for streaming input
8. ○ Create dashboard + reporting

Summary

SecurePipe is a secure, AI-enhanced, multi-language gateway service designed for filtering and forwarding data to LLMs in real time. It is suitable for demonstration in interviews, internal security projects, or as a foundation for enterprise LLM access governance.

The system balances performance (C++), flexibility (Python), and security (Docker, sandboxing), while maintaining professional engineering standards (CI/CD, modularity, documentation).

Authors & Contributions

- [Your Name Here] – System Design, Backend, Security Architecture

[Include GitHub link or project board if available]
