LogMate

복잡한 로그 관리 환경을 단순화하다.

졸업프로젝트
담당교수 : 유준범
강찬욱, 양승원, 정주연

Team Logmate

강찬욱

프로젝트 총괄
Agent 개발
Streaming 서버 개발

양승원

Frontend 개발
AI 이상탐지 모델 개발
AI 서버 구축

정주연

API 서버 개발
UI/UX 디자인

Contents

Overview

“기존 로그 모니터링은 너무 복잡하고 어렵습니다.”

LogMate는 **로그 수집부터 분석, 시각화, 그리고 이상 탐지**까지
한 번에 처리할 수 있는 **통합 오픈소스 로그 모니터링 시스템**입니다.
복잡한 설정 없이 바로 시작할 수 있고,
실시간으로 설정을 바꿔가며 로그를 모니터링할 수 있습니다.

Problem research

1 복잡한 모니터링 시스템 구축 과정

고난이도 인프라 설정과 초기 부담

2 단순한 로그 수집에도 과도한 비용 발생

단순 모니터링에도 높은 과금 및 러닝커브

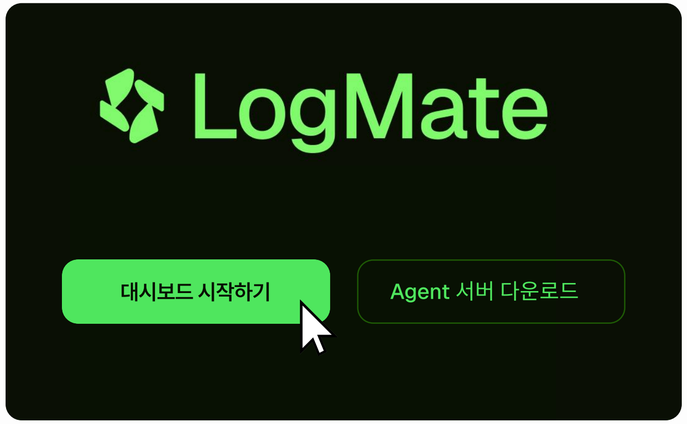
3 AI 분석 접근성 부족

전반적인 AI 로그 분석 활용 어려움

4 오픈소스 솔루션의 파편화

다양한 솔루션이 있지만 파편화되어 존재함

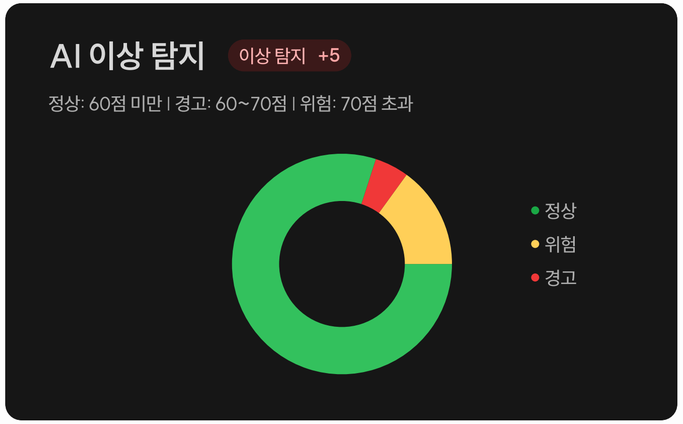
결국,로그 관리의 기본조차 접근하기 어려운 환경입니다.
LogMate는 이 네 가지 문제를 통합적으로 해결하기 위해 만들어졌습니다.



1

로그 모니터링을 더 쉽게, 더 간단하게

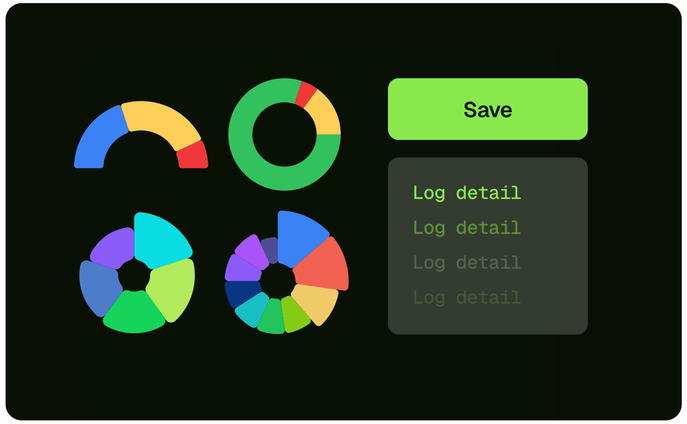
복잡한 설정이나 고비용 인프라 없이,
UI 기반의 직관적인 설정 환경을 통해
누구나 손쉽게 설치하고 바로 활용할 수 있는
오픈소스 로그 모니터링 서비스를 목표로 합니다.



2

로그 모니터링을 더 스마트하게

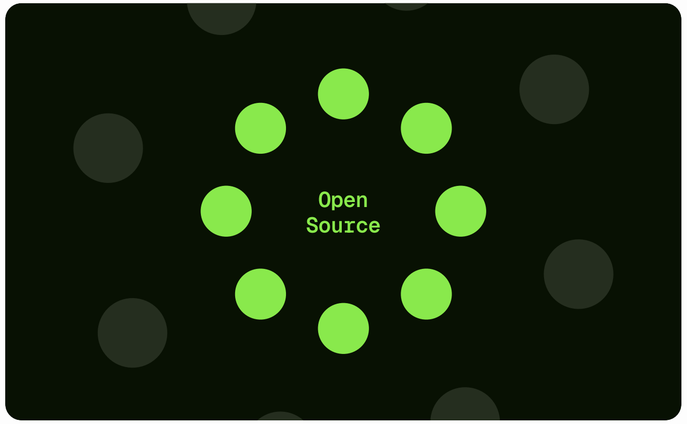
단순 모니터링을 넘어 AI가 로그 속
비정상 패턴을 자동 탐지하고,
위험 수준을 점수화하여
신속한 대응과 분석 지원을 목표로 합니다.



3

올인원 플랫폼

로그 수집, 저장, 분석, 시각화, AI 기반 탐지까지
하나의 파이프라인으로 구성된
SaaS형 통합 플랫폼을 구축해
운영 부담을 최소화를 목표로 합니다.



4

함께 성장하는 오픈소스 생태계

모듈화된 구조와 오픈소스 공개를 통해
누구나 기능을 확장하고 기여할 수 있는
지속 가능한 커뮤니티 중심 생태계 조성을 목표로 합니다.

목표 달성 기준-Functionality

소분류	평가내용	성공기준	성공 여부
핵심 비즈니스 기능 구현	로그 수집, 필터링, 시각화, 이상 탐지, webhook 알림 제공 여부	모든 기능이 통합되어 정상 작동하고 시연 가능	성공
로그 파일 지원 범위	다양한 로그 포맷 수집 가능 여부	최소 4종 이상의 기본 로그 포멧 처리 지원 Spring Boot Application Log Json Log PM2 Log Tomcat Access Log 등	2종 이상의 포멧 처리 지원 - Spring Boot Application, Tomcat Access
다중 로그 파일 처리	여러 로그 파일 동시 수집 가능 여부	최소 5개 로그 파일 병렬 처리 가능	성공
AI 이상탐지	정상과 비정상 분류 가능 여부	임의 로그 1000개로 시뮬레이션	성공

목표 달성 기준-Security

소분류	평가내용	성공기준	성공 여부
데이터 전송 보안	로그 전송 및 설정 전송 시 암호화/사용자 인증 적용 여부	HTTPS 암호화, 인증 토큰 적용	성공
API 보안	관리자/사용자 인증 및 권한 통제	JWT 기반 인증/인가 적용	성공

목표 달성 기준-Usability

소분류	평가내용	성공기준	성공 여부
UI 사용 편의성	초보자 사용 가능 여부	사용자 5명 중 4명 이상이 도움 없이 즉시 사용 가능	평가 예정
설치 편의성	에이전트 설치 난이도	간단한 커멘드로 설치	성공
OS 호환성	에이전트 실행 가능 환경	Linux, Windows, MacOS 에서 정상 작동	성공
반응속도	UI 및 API 전반의 응답 시간	모든 주요 요청 응답 시간 3초 이하	성공
실시간성	실시간 로그 수집 UI 반응 오차	수집 시점 대비 UI 반영 오차 5초 이하 유지	성공

목표 달성 기준-Performance

소분류	평가내용	성공기준	성공 여부
트래픽 처리	동시 연결된 에이전트의 로그 송신 요청 처리 가능 여부	30개 이상의 에이전트로부터 지연 없이 수신 가능	5개까지 확인
수집 처리량	초당 로그 수집량	초당 2,000 라인 이상 처리 가능	200라인까지 확인
리소스 사용량	에이전트 서버 자원 소비	2코어 CPU, 2GB RAM 컴퓨터 기준 CPU 사용률 5% 이하, 메모리 사용률 200MB 이하	성공
누적 로그 안정성	장기간 수집 시 안정	하루 500MB 이상 로그 누적 후 무장애 운영	평가 예정

시스템 컴포넌트

Agent

Java 기반 경량 로그 수집기

-로컬 로그 파일 실시간 수집 및 병합
-파싱, 필터링, 전송 기능 수행
-압축 전송, 배치 전송 기능 제공
-동적 설설정 재설정 지원

Streaming Server

Spring WebFlux 기반 로그 중계 및 실시간 처리 서버

-Agent 로부터 실시간 로그 수신
-수신 로그를 Kafka, OpenSearch 에 저장
-AI Server 와 연동하여 이상탐지 결과 송신
-WebSocket을 통한 실시간 로그 스트리밍

API Server

Spring Boot 기반 핵심 비즈니스 로직 REST API 서버

-사용자 인증 관리
-팀, 대시보드, Webhook 관리 기능 제공
-Agent 설정 관리 및 Pulling API 제공
-MySQL과 연동된 핵심 비즈니스 데이터 관리

AI Server

FastAPI 기반 로그 이상탐지 서버

-Isolation Forest 기반 이상 탐지 및 위험 점수화
-로그의 핵심 Feature 추출 및 정규화
-비정상 패턴 감지 후 Streaming 서버에 결과 전송

Frontend

React + TypeScript 기반 대시보드 UI

-실시간 WebSocket 로그 뷰어 및 검색 UI 제공
-Agent 설정 규칙 시각화 및 관리 UI 제공
-Webhook 설정, 팀/대시보드 관리 UI 제공
-로그인/회원가입 UI 제공

시스템 인프라

MySQL

사용자 및 시스템 데이터 관리 데이터베이스

-API 서버의 관계형 데이터 영속화
-사용자 정보, 팀/대시보드, 로그 수집 설정 등
비즈니스 도메인 데이터 관리

OpenSearch

로그 저장 및 검색 엔진

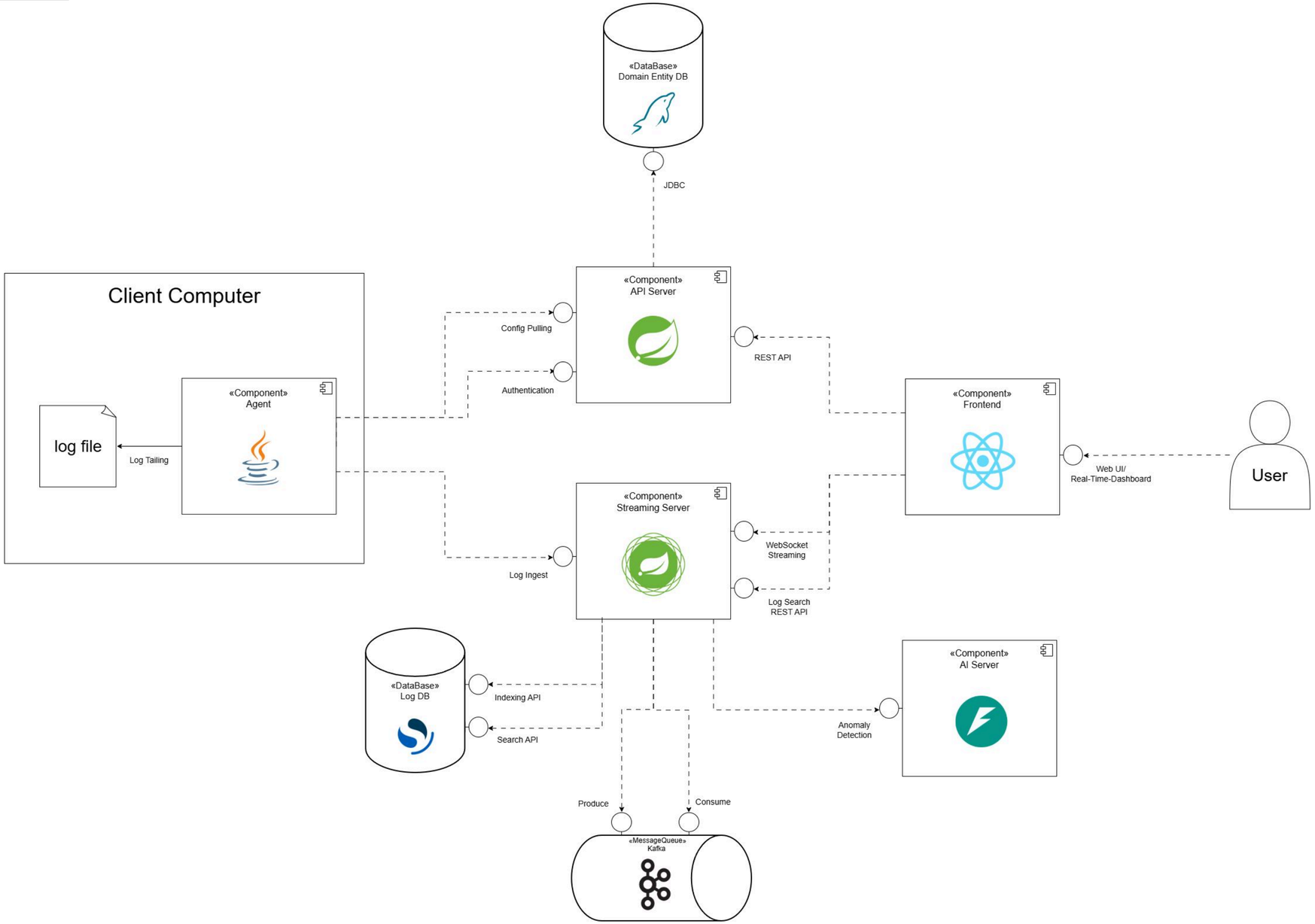
-Streaming Server에서 전달받은 로그를 인덱싱
-시간 기반 검색과 필터링 검색 지원

Kafka

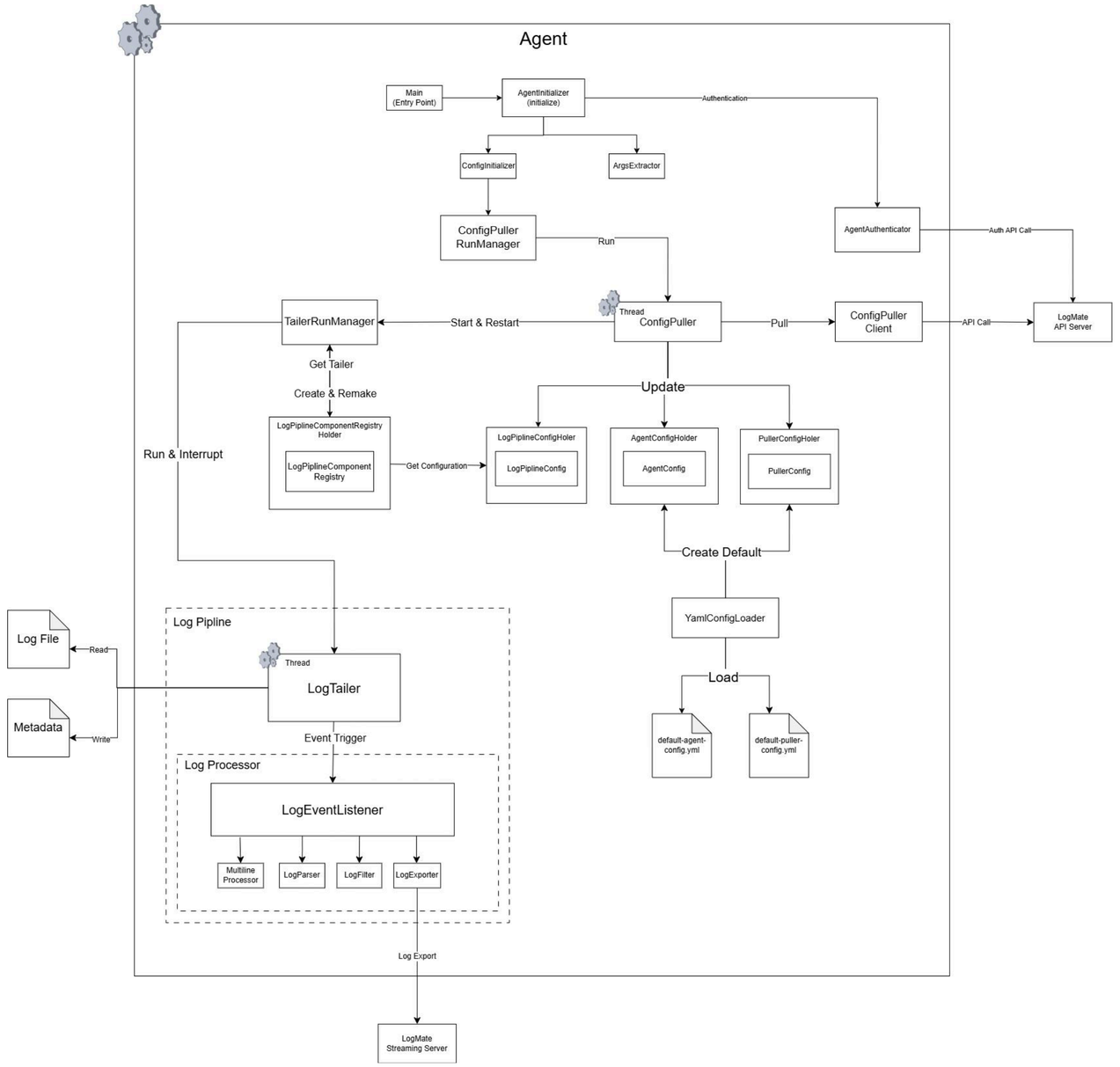
실시간 로그 수집과 복원을 위한 메시징 큐

-실시간 수집된 로그의 버퍼링
-Streaming 서버 장애 시 DLQ 를 통한 복구 지원

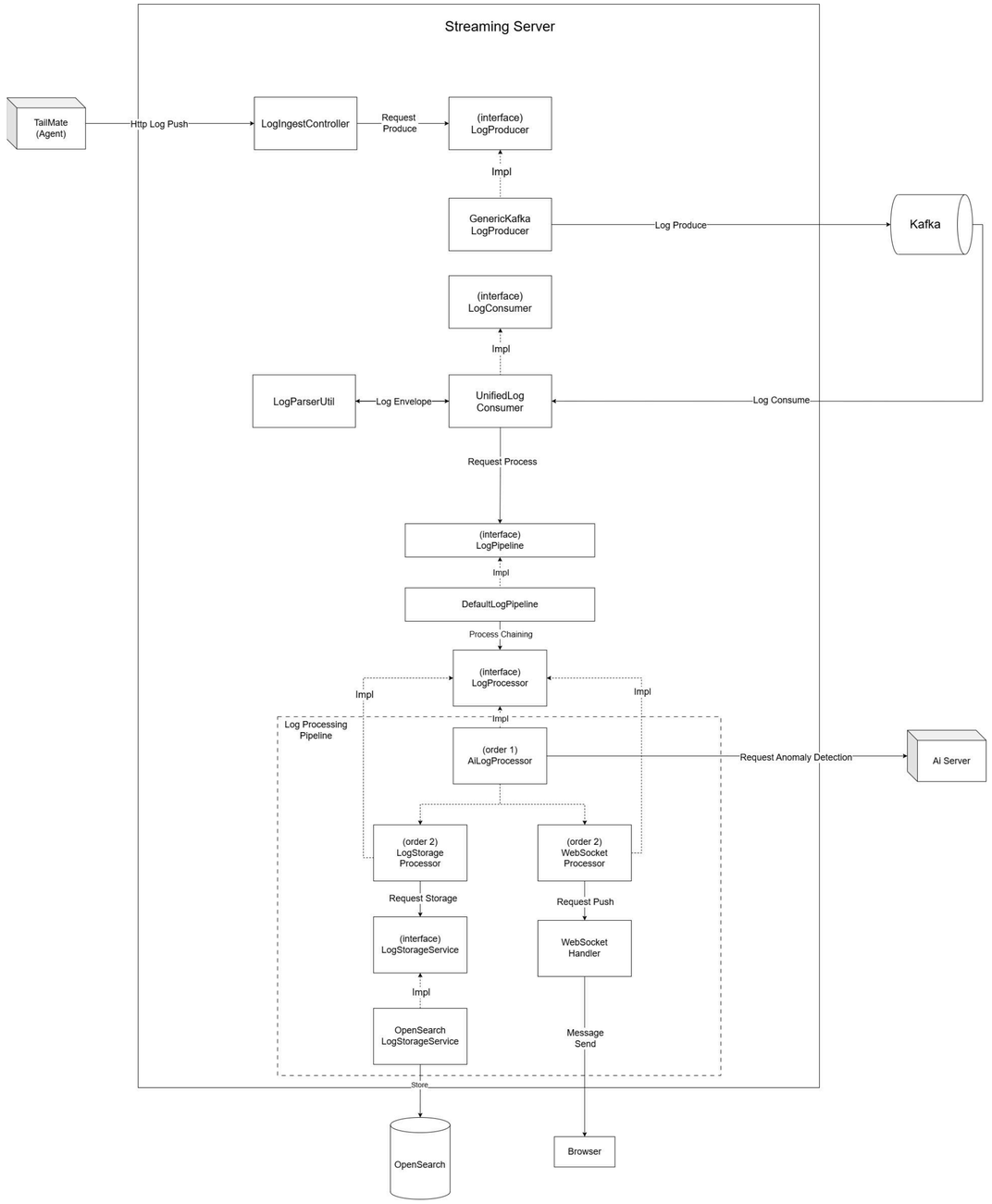
시스템 구성-Component Design



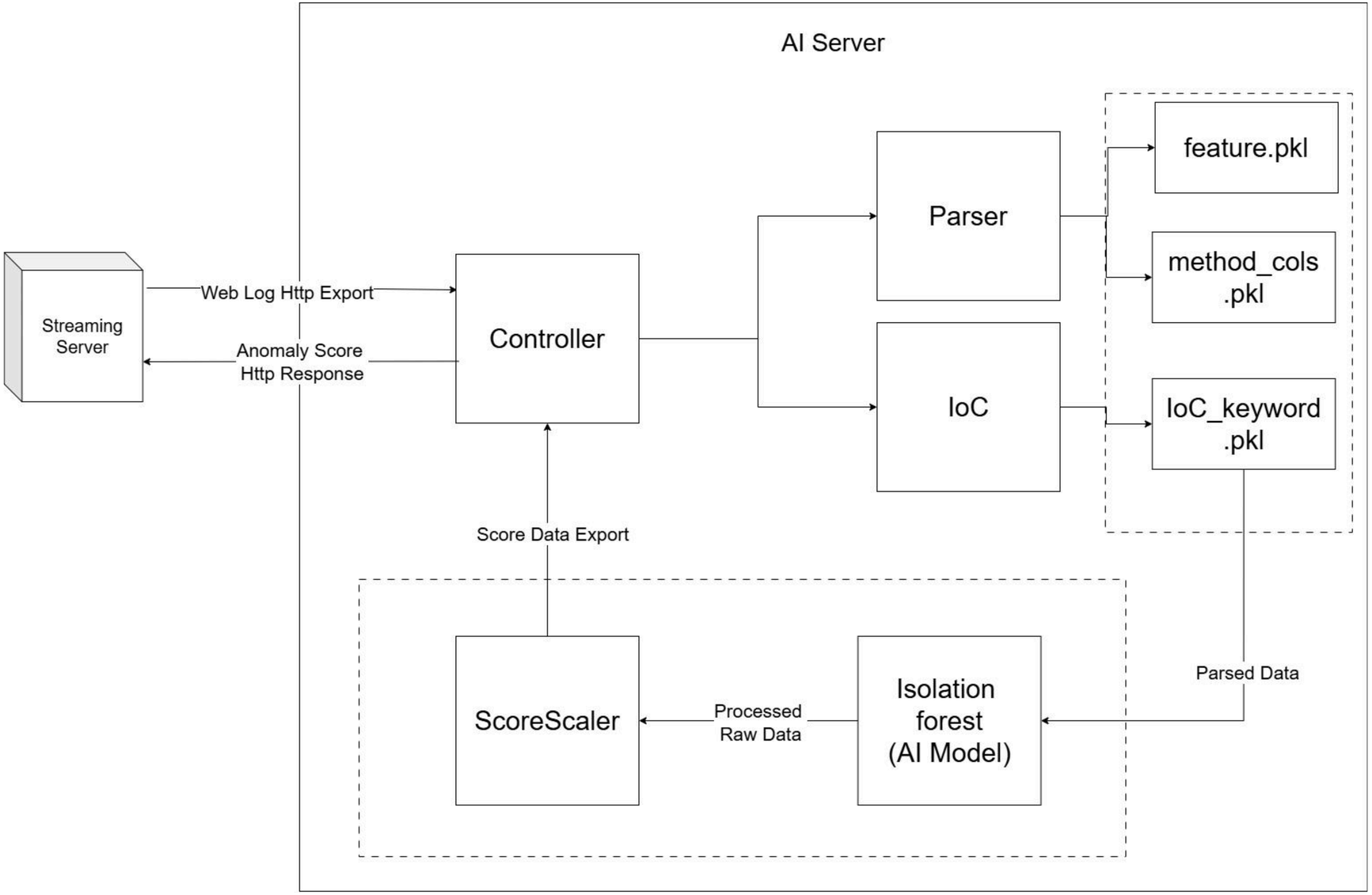
Agent-Component Design



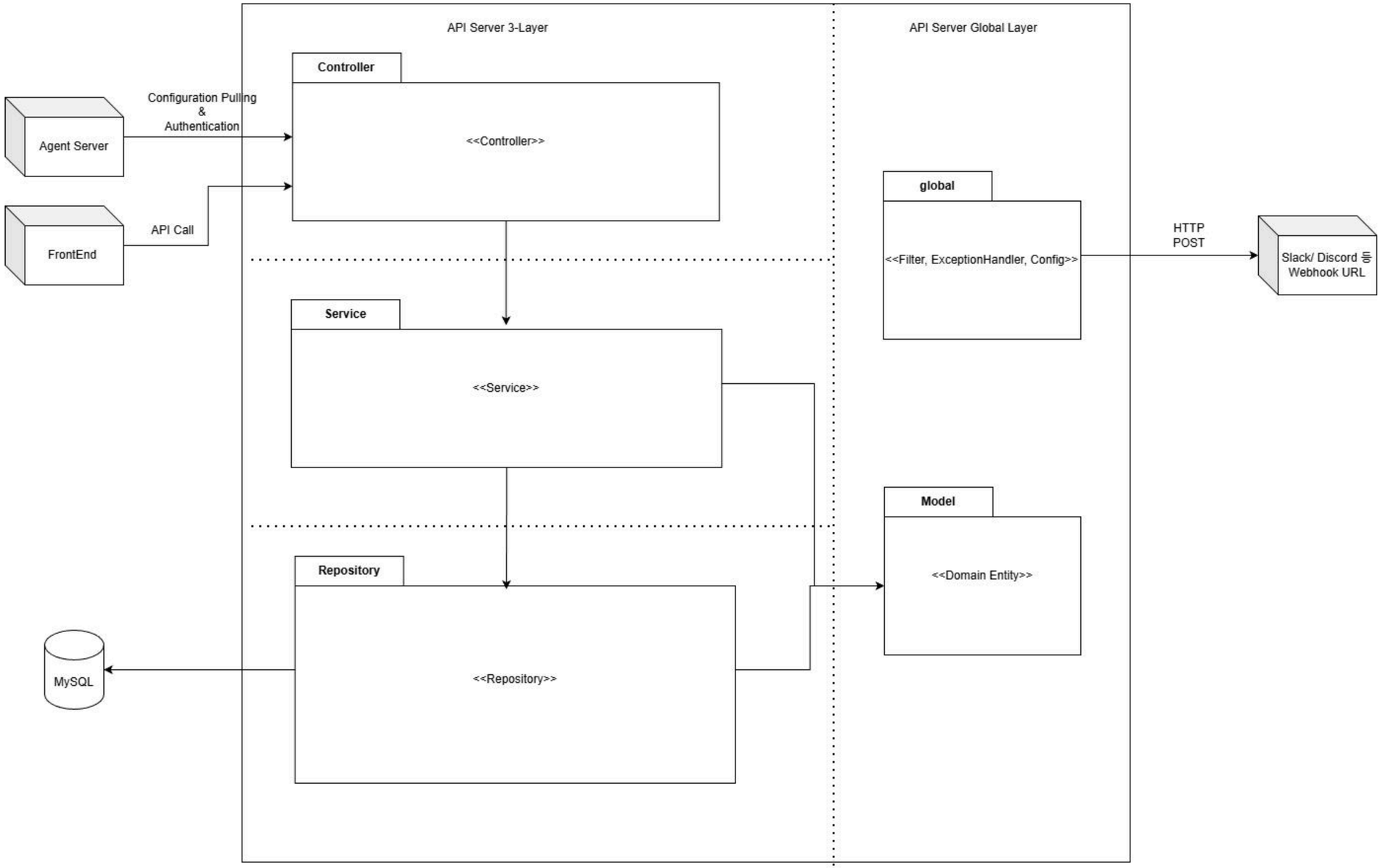
Streaming-Component Design



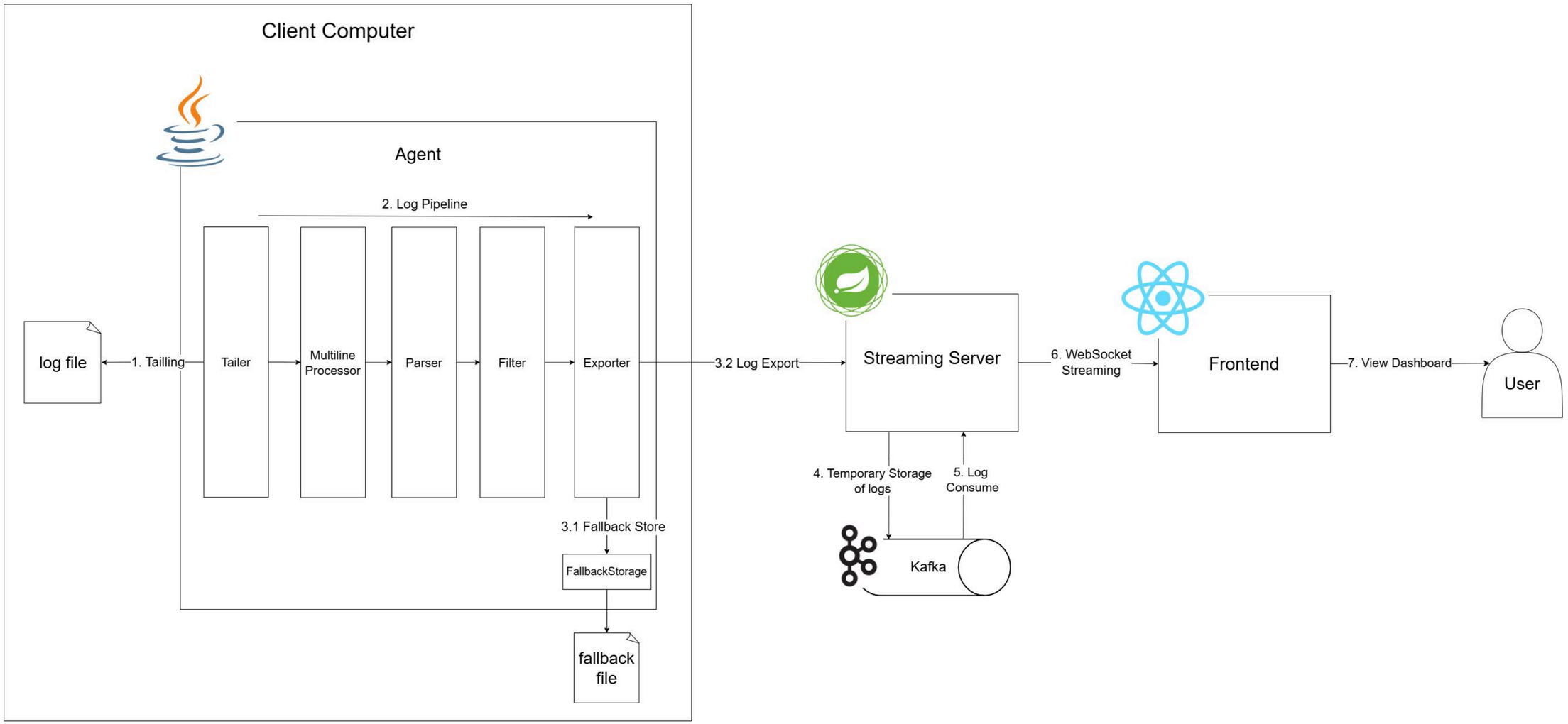
AI-Component Design



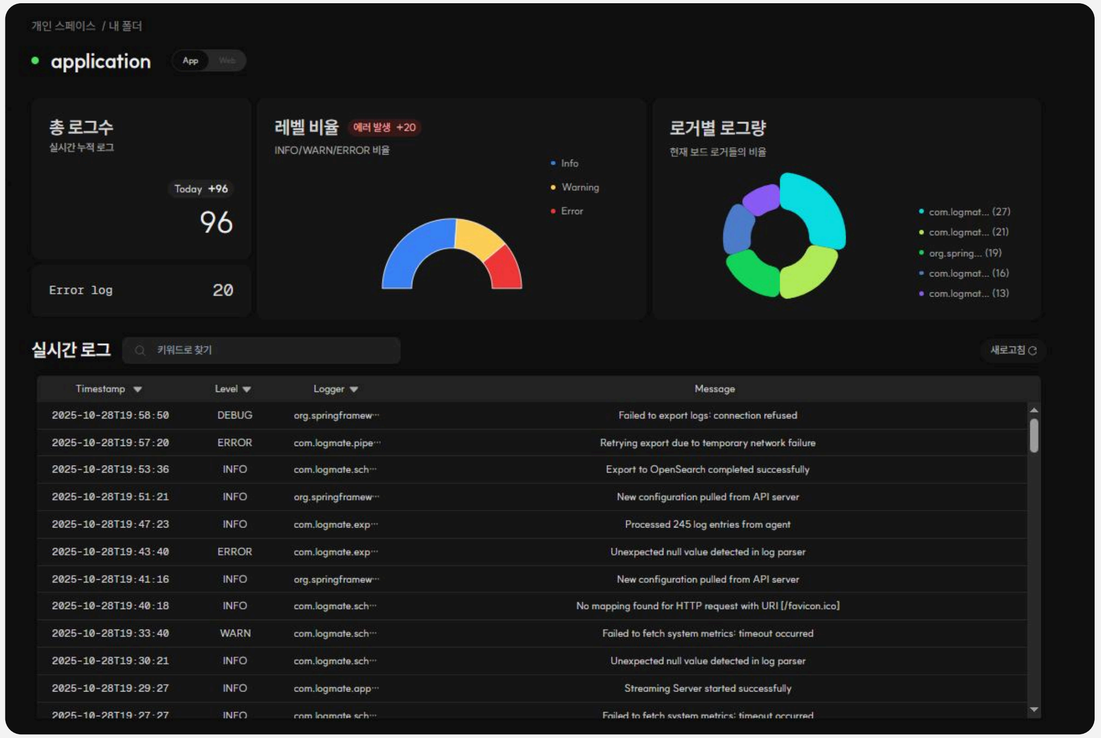
API-Component Design



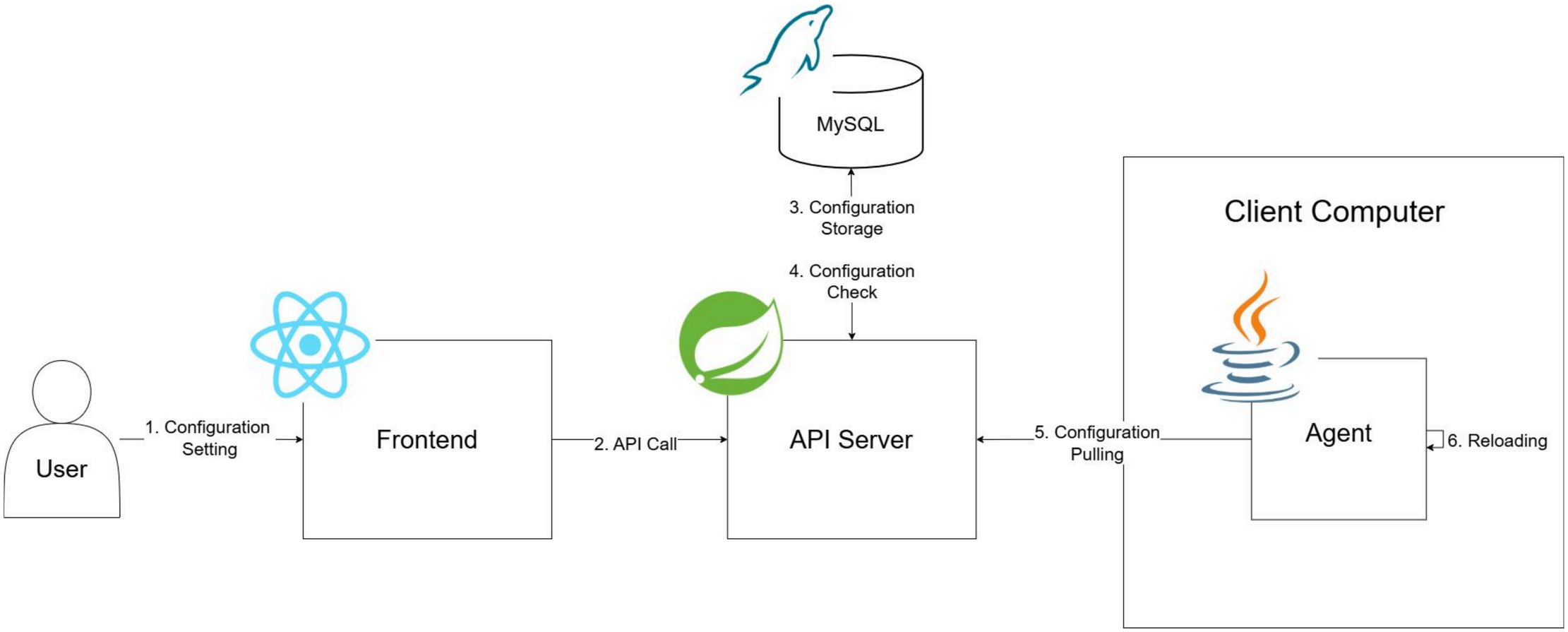
핵심 기능 - 실시간 로그 수집



핵심 기능 - 실시간 로그 수집

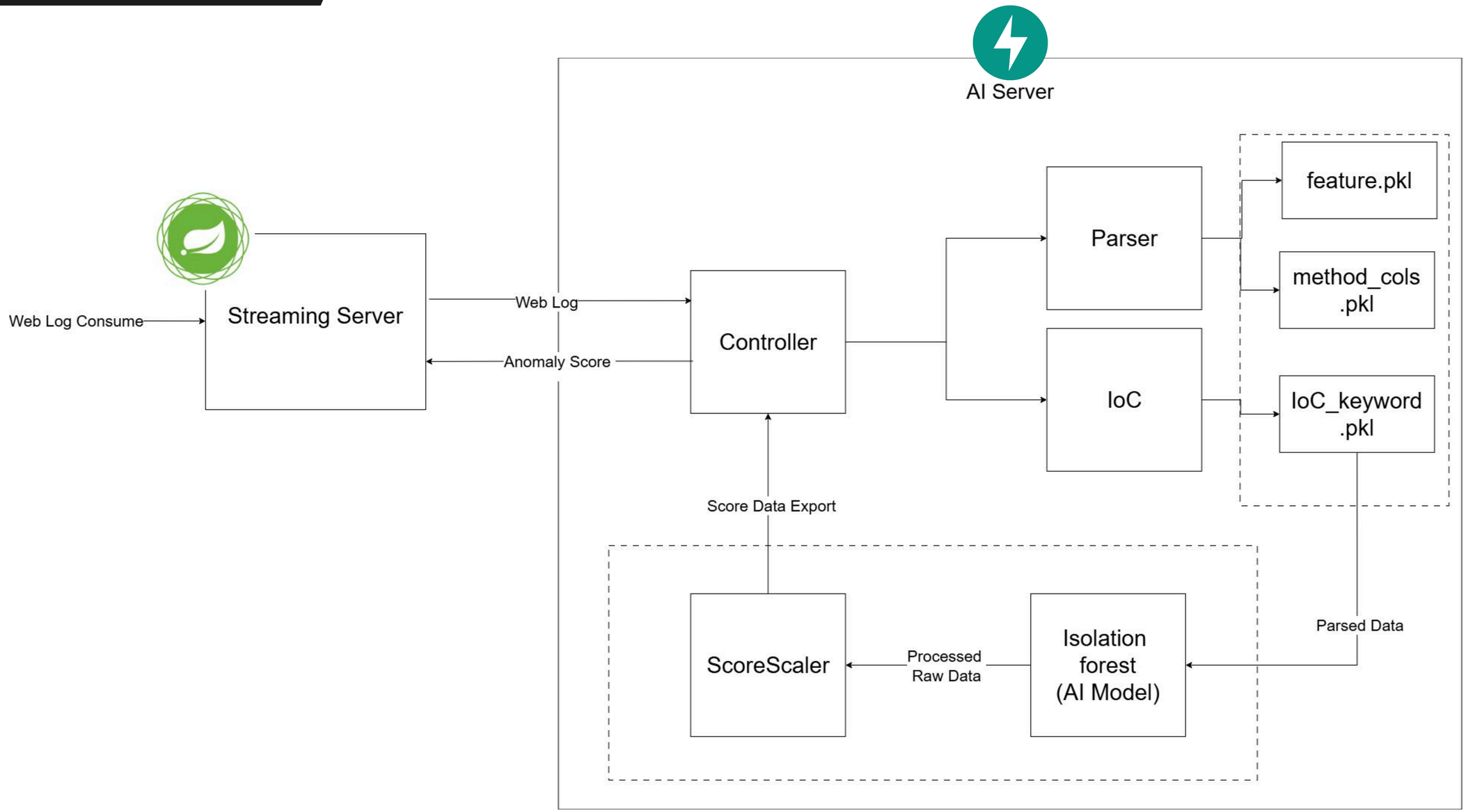


핵심 기능 - Agent 동적 재설정

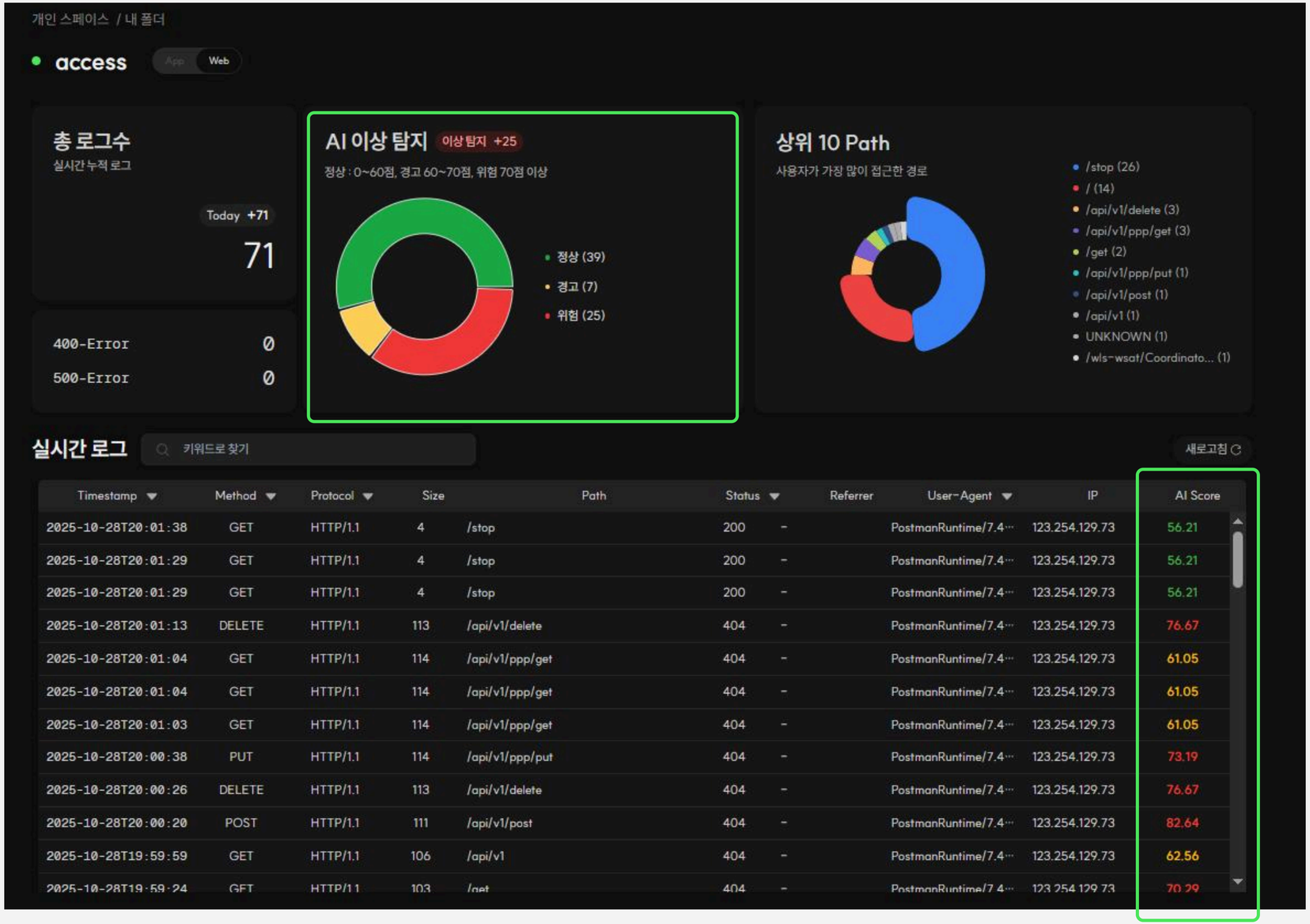


핵심 기능 - Agent 동적 재설정

핵심 기능 - Isolation Forest 기반 로그 이상탐지



핵심 기능 - Isolation Forest 기반 로그 이상탐지



Timestamp	Method	Protocol	Size	Path	Status	Referrer	User-Agent	IP	AI Score
2025-10-28T20:01:38	GET	HTTP/1.1	4	/stop	200	-	PostmanRuntime/7.4…	123.254.129.73	56.21
2025-10-28T20:01:29	GET	HTTP/1.1	4	/stop	200	-	PostmanRuntime/7.4…	123.254.129.73	56.21
2025-10-28T20:01:29	GET	HTTP/1.1	4	/stop	200	-	PostmanRuntime/7.4…	123.254.129.73	56.21
2025-10-28T20:01:13	DELETE	HTTP/1.1	113	/api/v1/delete	404	-	PostmanRuntime/7.4…	123.254.129.73	76.67
2025-10-28T20:01:04	GET	HTTP/1.1	114	/api/v1/ppp/get	404	-	PostmanRuntime/7.4…	123.254.129.73	61.05
2025-10-28T20:01:04	GET	HTTP/1.1	114	/api/v1/ppp/get	404	-	PostmanRuntime/7.4…	123.254.129.73	61.05
2025-10-28T20:01:03	GET	HTTP/1.1	114	/api/v1/ppp/get	404	-	PostmanRuntime/7.4…	123.254.129.73	61.05
2025-10-28T20:00:38	PUT	HTTP/1.1	114	/api/v1/ppp/put	404	-	PostmanRuntime/7.4…	123.254.129.73	73.19
2025-10-28T20:00:26	DELETE	HTTP/1.1	113	/api/v1/delete	404	-	PostmanRuntime/7.4…	123.254.129.73	76.67
2025-10-28T20:00:20	POST	HTTP/1.1	111	/api/v1/post	404	-	PostmanRuntime/7.4…	123.254.129.73	82.64
2025-10-28T19:59:59	GET	HTTP/1.1	106	/api/v1	404	-	PostmanRuntime/7.4…	123.254.129.73	62.56
2025-10-28T19:59:24	GET	HTTP/1.1	103	/get	404	-	PostmanRuntime/7.4…	123.254.129.73	70.29

기능 시연

시스템 테스트 결과

Step	Action	Sender	Receiver	Expected Result	Success
1	인증 요청	Agent	API	200 OK + Access Token 발급	**성공**
2	Config Pull	Agent	API	eTag 비교 후 변경된 설정 반환	**성공**
3	Log Push	Agent	Streaming	JSON 로그 수신	**성공**
4	Kafka Publish	Streaming	Kafka	Topic 정상 저장	**성공**
5	Kafka Consume	Kafka	Streaming	Topic 메시지 정상 소비	**성공**
6	Log Streaming	Streaming	Front	실시간 로그 데이터 전달	**성공**
7	Log Search	Front	Streaming	검색 요청 → 필터링된 로그 반환	**성공**
8	WS Broadcast	Streaming	Front	실시간 WebSocket 반영	**성공**

시스템 테스트 결과

Step	Action	Sender	Receiver	Expected Result	Success
9	Dashboard Update	Front	API	Dashboard CRUD 동작	**성공**
10	대시보드 시작하기	Front	API	초기 대시보드 로딩	**성공**
11	회원가입	Front	API	신규 사용자 등록	**성공**
12	로그인	Front	API	JWT 토큰 발급	**성공**
13	로그아웃	Front	API	토큰 무효화/세션 종료	**성공**
14	내 정보 확인	Front	API	사용자 정보 반환	**성공**
15	내 정보 수정	Front	API	사용자 프로필 업데이트	**성공**
16	개인 스페이스 폴더 추가	Front	API	개인 폴더 생성	**성공**
17	팀 스페이스 팀 추가	Front	API	팀 공간 생성	**성공**
18	팀 스페이스 팀원 추가	Front	API	팀원 등록	**성공**
19	팀 스페이스 팀원 조회	Front	API	팀원 목록 조회	**성공**
20	팀 설정 변경	Front	API	팀 설정 업데이트	**성공**
21	팀 삭제	Front	API	팀 공간 삭제	**성공**

시스템 테스트 결과

Step	Action	Sender	Receiver	Expected Result	Success
20	팀 설정 변경	Front	API	팀 설정 업데이트	**성공**
21	팀 삭제	Front	API	팀 공간 삭제	**성공**
22	개인 스페이스 대시보드 추가	Front	API	대시보드 생성	**성공**
23	대시보드 응답 확인	Front	API	생성된 대시보드 데이터 반환	**성공**
24	대시보드 설정 변경	Front	API	설정 업데이트	**성공**
25	대시보드 삭제	Front	API	대시보드 삭제	**성공**
26	대시보드 조회	Front	API	대시보드 리스트/세부 정보 조회	**성공**

요구사항 달성도

1

Functional Requirements

- 94개 중 92개 만족
- FR-1.3.2, FR-1.4.3 미구현
(로그파일 접근 제어, 중복 전송 방지)

2

Non-Functional Requirements

- 25개 중 22개 만족
- Availabilty 중 1개, Security 중 1개 불만족
(WS 연속 스트림 유지, 민감 정보 마스킹)
- Maintainability 중 1개 불만족
(운영 지표 수집)

요구사항 달성도

1

System Test 결과 분석

사용자가 실제로 시스템을 사용하는
“전체흐름” 은 모두 정상 동작

2

FR, NFR 관점의 결론

필수 기능은 완성. 선택 기능 보완 필요
NFR 의 경우 성능 측면에서 더 정교한 테스트 필요
보안성, 확장성, 유지보수성 등에서 보완 필요

추후 계획

쉽고 간단하지만, 기능은 결코 가볍지 않은
로그 모니터링 플랫폼을 향한 앞으로의 여정입니다.

1 > 2 > 3

테스트 안정성 확보 및 요구사항 충족

부족한 테스트 목록 확인 후 검증하고,
다양한 환경에서의 안정성 확보를 목표
FR, NFR 에 대한 모든 사항 충족 및 검증

AI 고도화 (LLM 활용 분석)

로그 내용을 자연어로 요약/설명하고,
탐지된 이상 로그에 대한 원인 분석 및
대응 방안 생성

사용자 맞춤 대시보드

대시보드 커스터마이징 기능 제공

Agent 파이프라인 확장

Parser, Filter, Exporter 의 사용자 커스텀 기능
추가 및 파싱 가능한 로그 포맷 다양화

보안/신뢰성 강화

보안 취약점 분석 및 mTLS·RBAC 등 보안
표준기술 적용을 통한 신뢰성 확보

종합 모니터링 플랫폼

로그, 매트릭을 통합한
종합 모니터링 플랫폼으로 확장

Thank You

Team LogMate