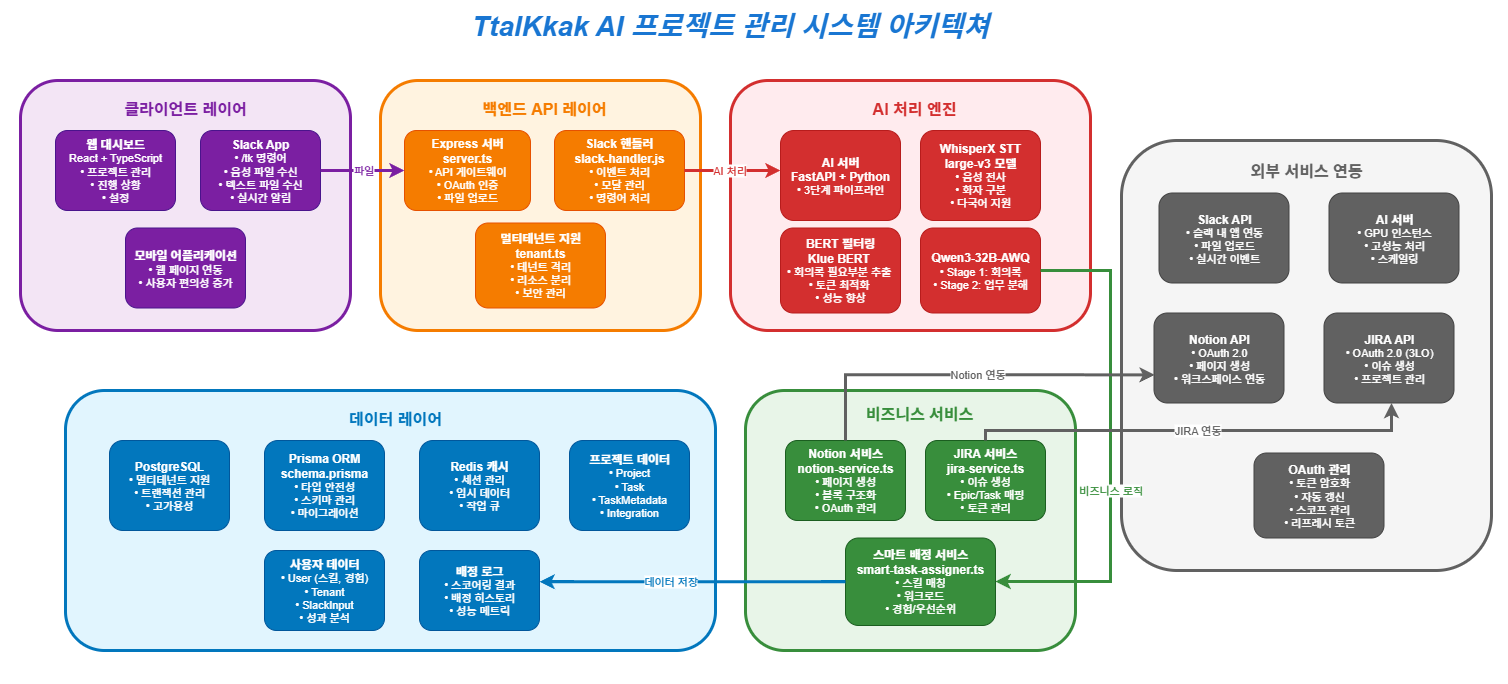


**SK네트웍스 Family AI 과정 12기  
 모델링 및 평가 시스템 아키텍처**



| **산출물 단계** | 모델링 및 평가 |
| --- | --- |
| **평가 산출물** | 시스템 아키텍처 |
| **제출 일자** | 2025.07.18 |
| **깃허브 경로** | <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-3TEAM> |
| **작성 팀원** | 노명구 |

1. **시스템 아키텍쳐**



**1-1. 클라이언트 레이어**

사용자 인터페이스 계층

**Slack Bot (@slack/bolt)**

- /tk 명령어로 프로젝트 세션 시작

- 음성 파일 업로드, 텍스트 파일 업로드 및 검증

- 실시간 진행 상황 알림 및 결과 링크 제공

- Block Kit 기반 인터랙티브 UI

**웹 대시보드 (React + TypeScript)**

- 프로젝트 현황 및 진행률 모니터링

- 팀원별 워크로드 시각화 및 배정 관리

- OAuth 연동 설정 및 시스템 구성

- 성과 분석 및 리포팅 대시보드

**모바일 어플리케이션**

- 웹 페이지 연동

- 사용자 편의성 증가

**1-2. 백엔드 API 레이어**

비즈니스 로직 및 API 처리

**Express 서버 (server.ts)**

- API 게이트웨이 역할 및 요청 라우팅

- OAuth 2.0 인증 시스템 (Slack, Notion, JIRA)

- 파일 업로드 관리 및 보안 검증

- Rate Limiting 및 에러 처리

**Slack 핸들러 (slack-handler.js)**

- Slack 이벤트 실시간 처리

- 모달 UI 관리 및 명령어 라우팅

- 세션 상태 관리 및 워크플로우 제어

**멀티테넌트 미들웨어 (tenant.ts)**

- 테넌트별 데이터 완전 격리

- 리소스 분리 및 보안 정책 적용

- 동적 스케일링 지원

**1-3. AI 처리 엔진**

Runpod 클라우드 기반 AI 파이프라인

**AI 서버 (FastAPI + Python)**

- 3단계 파이프라인 오케스트레이션

- GPU 리소스 효율적 관리

- 비동기 처리 및 상태 추적

**WhisperX STT (large-v3)**

- 고정확도 음성 전사

- 화자 구분 (Speaker Diarization)

- 다국어 지원 및 노이즈 제거

**BERT 필터링 (Klue BERT)**

- 회의록 핵심 부분 추출

- 토큰 최적화로 성능 향상

- 불필요한 내용 필터링

**Qwen3-32B-AWQ LLM**

- Stage 1: 음성 → 프로젝트 회의록

- Stage 2: 회의록 → 업무 분해

- 메타데이터 생성 (스킬, 복잡도, 우선순위)

**1-4. 비즈니스 서비스 레이어**

핵심 비즈니스 로직

**Notion 서비스 (notion-service.ts)**

- AI 콘텐츠를 구조화된 Notion 페이지로 변환

- 계층적 페이지 구조 자동 생성

- 실시간 JIRA 동기화

**JIRA 서비스 (jira-service.ts)**

- Epic/Task 계층 구조 자동 생성

- 이슈 상세 정보 및 담당자 할당

- OAuth 토큰 자동 갱신 관리

**스마트 업무 배정 (smart-task-assigner.ts)**

**스마트 업무 배정 알고리즘:**

- 스킬 매칭 + 워크로드 + 경험

- 실시간 재배정 및 성과 학습

- 설명 가능한 배정 근거 제공

**1-5. 데이터 레이어**

데이터 저장 및 관리

**PostgreSQL**

- 멀티테넌트 스키마 분리

- ACID 트랜잭션 및 고가용성

- Master-Slave 복제 구성

**Prisma ORM (schema.prisma)**

- TypeScript 타입 안전성

- 자동 마이그레이션 및 스키마 관리

- N+1 문제 해결 최적화

**Redis 캐시**

- 세션 및 토큰 관리

- 실시간 작업 큐

- API 응답 캐싱

**핵심 데이터 모델:**

- Project, Task, TaskMetadata

- User, UserSkill, Tenant

- AssignmentLog, PerformanceMetric

**1-6. 외부 서비스 연동**

외부 API 통합

**Slack API**

- 봇 토큰 관리 및 실시간 이벤트

- 파일 업로드 및 인터랙티브 UI

**Notion API**

- OAuth 2.0 워크스페이스 연동

- 페이지/데이터베이스 자동 생성

**JIRA API**

- 3-legged OAuth 및 프로젝트 관리

- 웹훅 기반 실시간 동기화

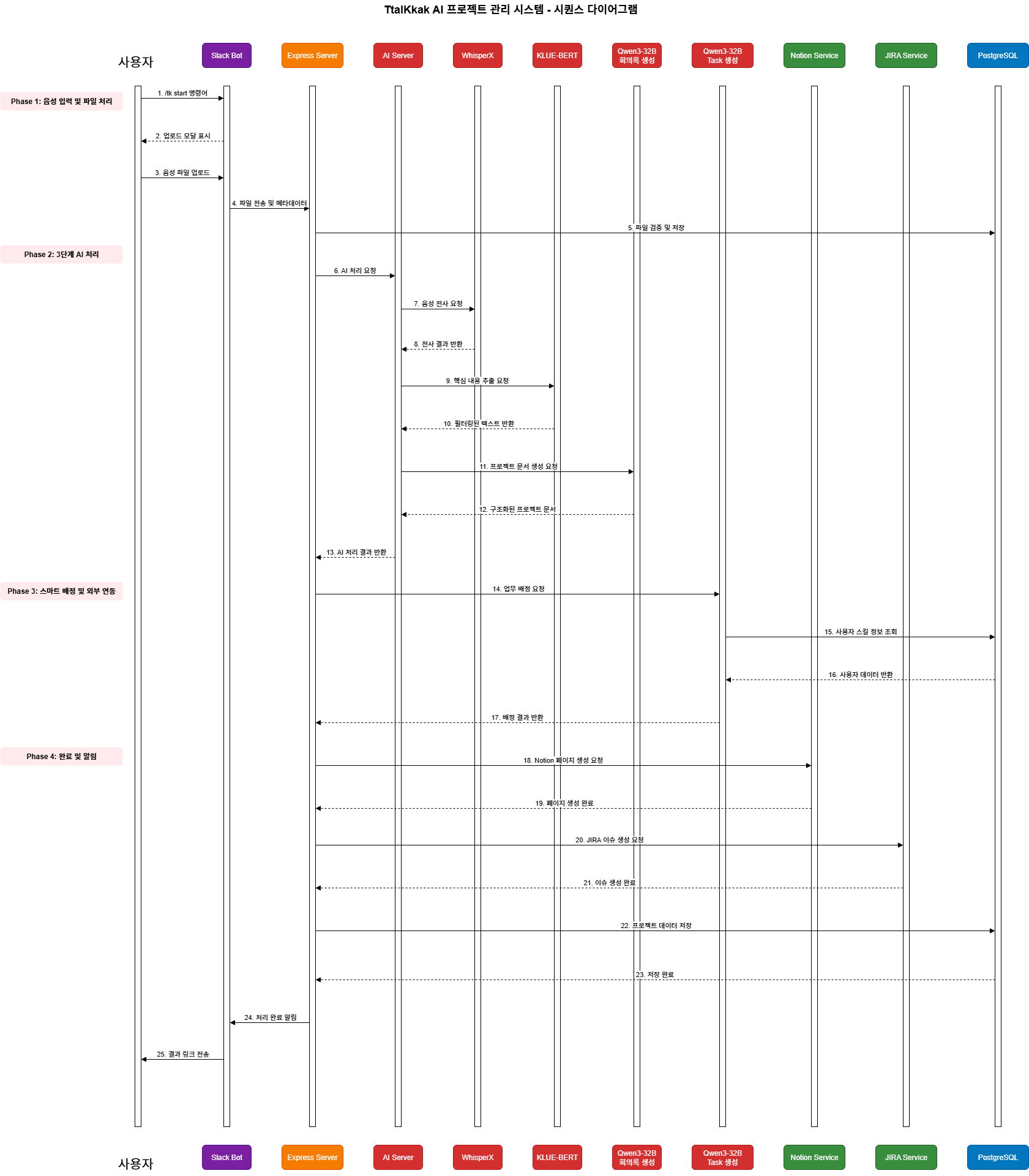
**OAuth 관리자**

- 중앙집중식 토큰 관리

- AES-256 암호화 저장

- 자동 갱신 및 만료 처리

1. **시퀀스 다이어그램**

****

**2-1. 4단계 프로세스 흐름**

**참여자 (Actors)**

1. 사용자 - 시스템을 사용하는 팀원

2. Slack Bot - Slack 인터페이스 처리

3. Express Server - 백엔드 API 서버

4. AI Server - AI 파이프라인 오케스트레이터

5. WhisperX - 음성 전사 엔진

6. KLUE-BERT - 텍스트 필터링 모델

7. Qwen3-32B - 문서 및 Task 생성 LLM

8. Task Assigner - 스마트 업무 배정 시스템

9. Notion Service - Notion 페이지 생성 서비스

10. JIRA Service - JIRA 이슈 생성 서비스

11. PostgreSQL - 데이터베이스

**2-2. 4단계 프로세스 흐름**

**Phase 1: 음성 입력 및 파일 처리 (1-5단계)**

1단계: 사용자 → Slack Bot

- 사용자가 /tk start 명령어 입력

- 새로운 프로젝트 세션 시작 요청

2단계: Slack Bot → 사용자

- Slack Bot이 파일 업로드 모달 UI 표시

- 음성 파일 업로드 인터페이스 제공

3단계: 사용자 → Slack Bot

- 사용자가 회의 음성 파일 업로드

- 파일 메타데이터와 함께 전송

4단계: Slack Bot → Express Server

- 업로드된 파일과 메타데이터 전송

- API 요청으로 백엔드 서버에 전달

5단계: Express Server → PostgreSQL

- 파일 검증 및 임시 저장

- 세션 정보 데이터베이스 기록

**Phase 2: 3단계 AI 처리 (6-13단계)**

6단계: Express Server → AI Server

- AI 처리 파이프라인 시작 요청

- 파일 경로 및 처리 옵션 전송

7-8단계: AI Server ↔ WhisperX

- 7단계: AI Server가 WhisperX에 음성 전사 요청

- 8단계: WhisperX가 전사된 텍스트 반환 (화자 구분, 타임스탬프 포함)

9-10단계: AI Server ↔ KLUE-BERT

- 9단계: AI Server가 KLUE-BERT에 핵심 내용 추출 요청

- 10단계: KLUE-BERT가 필터링된 텍스트 반환 (토큰 최적화 완료)

11-12단계: AI Server ↔ Qwen3-32B

- 11단계: AI Server가 Qwen3-32B에 프로젝트 문서 생성 요청

- 12단계: Qwen3-32B가 구조화된 프로젝트 문서 반환 (회의록, 업무 분해)

13단계: AI Server → Express Server

- 3단계 AI 처리 완료 결과 반환

- 구조화된 프로젝트 데이터 전송

**Phase 3: 스마트 배정 및 외부 연동 (14-21단계)**

14단계: Express Server → Task Assigner

- 스마트 업무 배정 시스템에 배정 요청

- AI가 생성한 업무 목록 전송

15-16단계: Task Assigner ↔ PostgreSQL

- 15단계: Task Assigner가 사용자 스킬 정보 조회

- 16단계: PostgreSQL이 사용자 데이터 반환 (스킬, 경험, 워크로드)

17단계: Task Assigner → Express Server

- 4가지 가중치 알고리즘 적용 결과 반환

- 최적 배정 결과 및 스코어링 근거 제공

18-19단계: Express Server ↔ Notion Service

- 18단계: Express Server가 Notion 페이지 생성 요청

- 19단계: Notion Service가 페이지 생성 완료 응답

20-21단계: Express Server ↔ JIRA Service

- 20단계: Express Server가 JIRA 이슈 생성 요청

- 21단계: JIRA Service가 이슈 생성 완료 응답

**Phase 4: 완료 및 알림 (22-25단계)**

22-23단계: Express Server ↔ PostgreSQL

- 22단계: Express Server가 프로젝트 데이터 영구 저장

- 23단계: PostgreSQL이 저장 완료 응답

24단계: Express Server → Slack Bot

- 전체 처리 완료 알림 전송

- 생성된 리소스 정보 포함

25단계: Slack Bot → 사용자

- 최종 결과 링크 전송

- Notion 페이지 URL, JIRA 이슈 링크 제공