

---

# 로컬 AI 기반 사내 회의 워크스페이스 시스템



202110763 | 컴퓨터공학과 | 신연준

담당교수님: 송해상 교수님

26.03.19

# 목차

<b>01</b>	요구사항 분석	—————	03p ~ 06p
<b>02</b>	시스템 아키텍처	—————	07p ~ 9p
<b>03</b>	기능표	—————	10p ~ 11p
<b>04</b>	ER-다이어그램	—————	12p ~ 17p
<b>05</b>	기술 검증(PoC) : STT 벤치마킹	—————	18p
<b>06</b>	기대효과 및 개발환경	—————	19p ~ 20p
<b>07</b>	향후 개발 계획 및 질의응답	—————	21p

## 기존 회의 시스템

### 플랫폼

Teams, Zoom, Google Meet, Webex 등은 회의 개설, 참여, 화면 공유, 기본 자막 기능을 제공. 그러나 회의 중 생성된 정보가 이후 업무 시스템과 자연스럽게 연결되지는 않는다.

### 기록

회의 내용을 텍스트로 기록하고 요약하는 기능은 제공하지만, 주로 기록 보존 중심으로 활용된다. 실시간 보조나 조직 내부 데이터 추적 관점에서는 한계가 있다.

### 문서화

회의 후 결과는 문서, 메신저, 협업 툴 등으로 다시 정리되고 공유된다. 이 과정에서 회의 데이터가 여러 도구에 분산되어 관리되며, 재검색과 후속 업무 연결이 비효율적이다.

### 기존 회의 시스템의 한계

01

데이터 분산

- 회의 플랫폼, 전사 도구, 문서화 도구가 분리되어 회의 데이터가 여러 서비스에 흩어진다.
- 기록, 요약, 결정사항, 후속 업무를 일관되게 관리하기 어렵다.

02

플랫폼 종속

- 사용하는 회의 플랫폼이 바뀌면 동일한 사용 경험을 유지하기 어렵다.
- 기능 제공 범위와 데이터 접근 방식이 플랫폼 정책에 크게 좌우된다.

03

사후 활용 한계

- 회의 내용은 기록되더라도 이후 검색, 히스토리 연결, 후속 업무로 체계적으로 이어지지 않는다.
- 결과적으로 회의 데이터가 조직 내부 자산으로 충분히 활용되지 못한다.



Teams



zoom



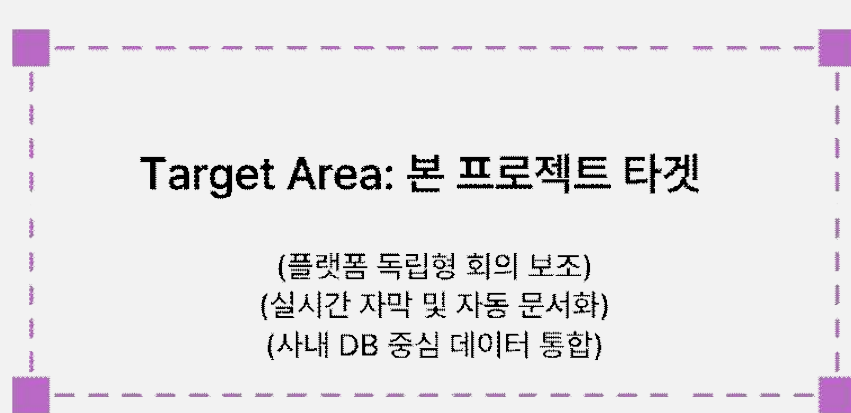
webex

플랫폼 종속

## 기존 서비스의 한계

기존 서비스는 회의 진행과 기록에는 강하지만, 플랫폼 독립성과 회의 데이터의 사내 축적·활용 측면에서는 한계가 있다.

회의 데이터 활용도



플랫폼 독립



Rev AI Transcription

단순 기록

## 핵심 요구사항 도출

- 회의 중에는 플랫폼에 종속되지 않는 실시간 보조 기능이 필요하다.
- 회의 과정에서 생성되는 정보는 자동으로 정리되고 구조화되어야 한다.
- 모든 회의 데이터는 사내 데이터베이스에 통합 저장되어 이후 검색, 관리, 후속 업무에 활용될 수 있어야 한다.

플랫폼 독립형 오버레이

실시간 보조 및 자동 문서화

사내 DB 중심 데이터 통합



플랫폼 독립형 오버레이

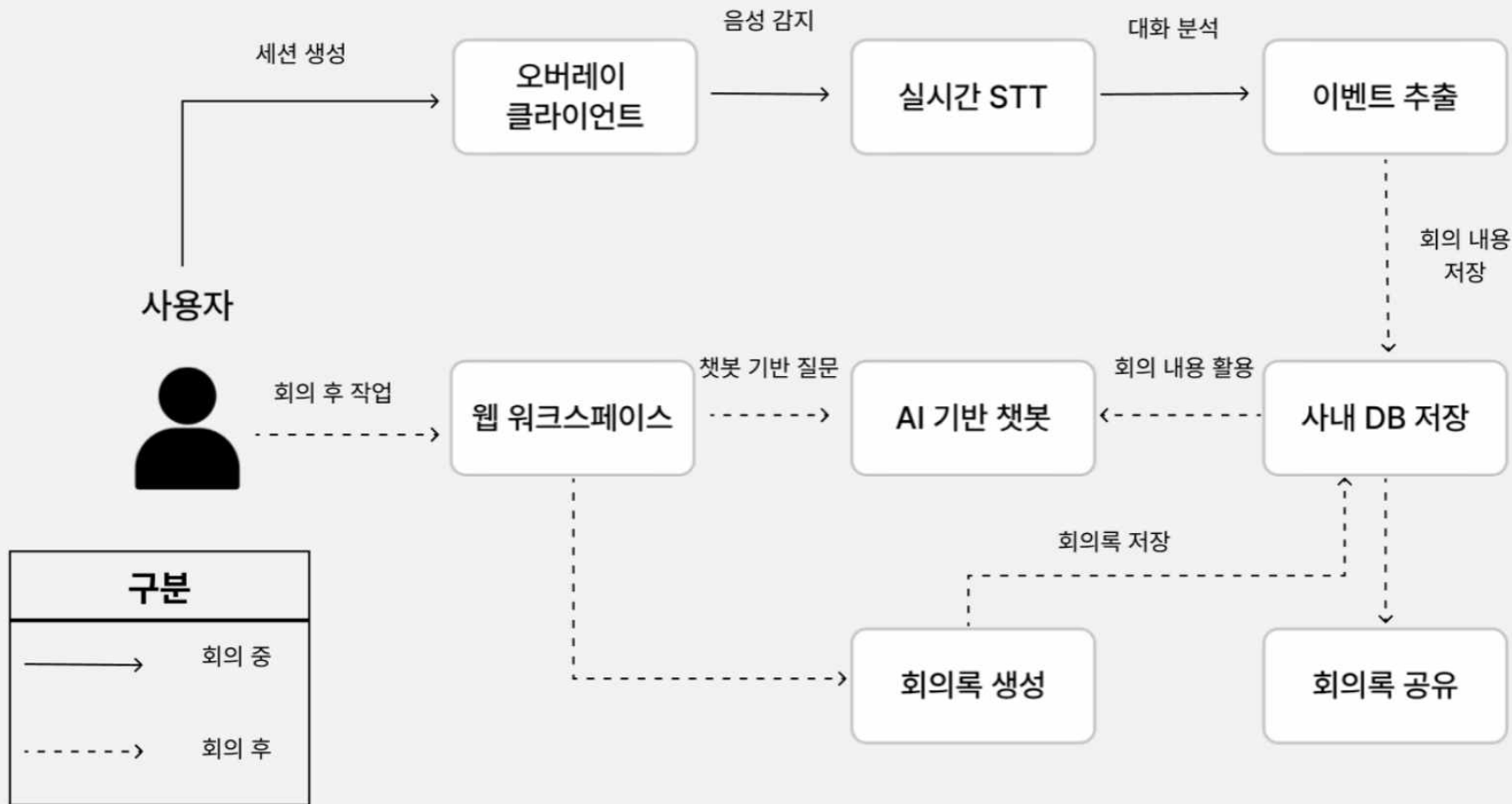


실시간 보조 및 자동 문서화

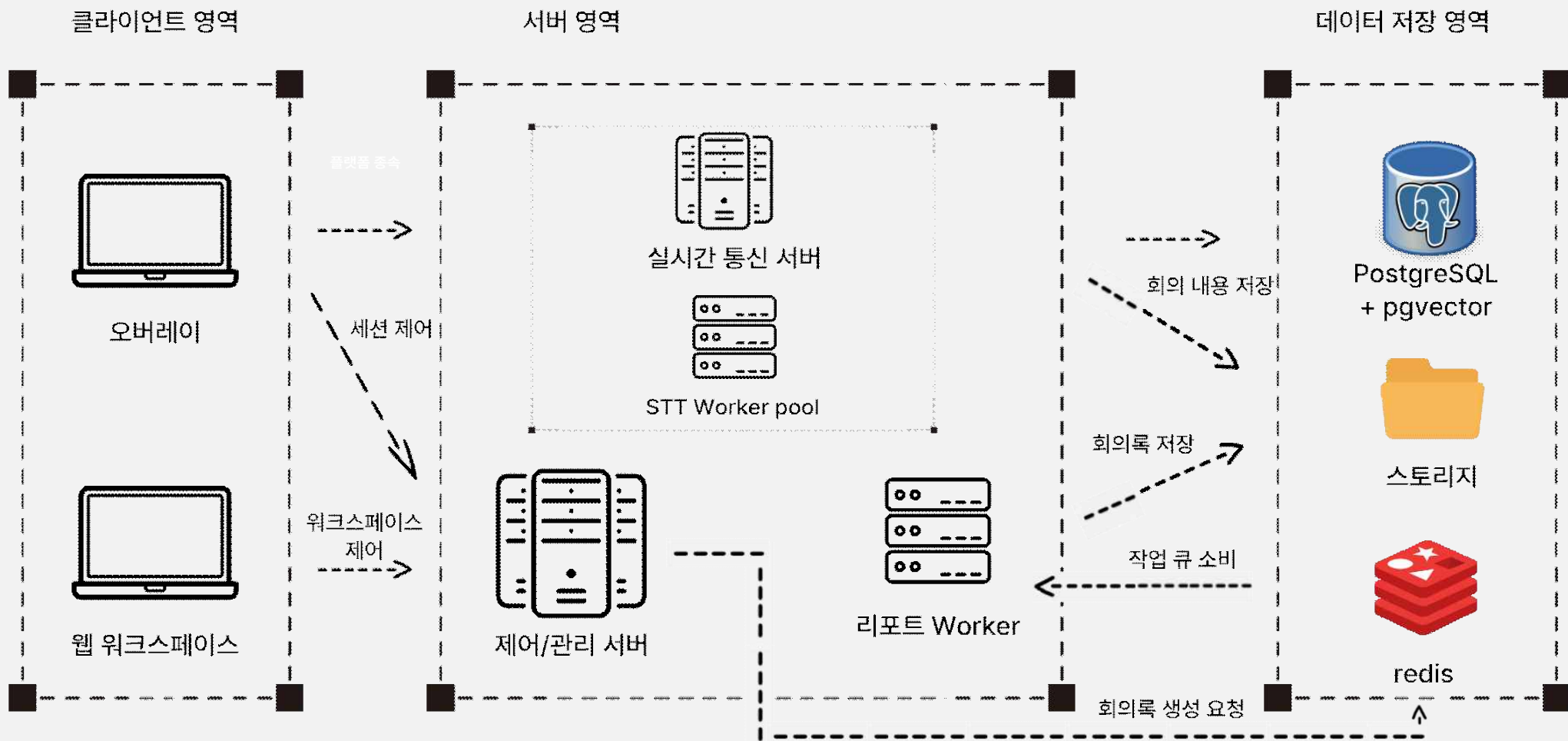


사내 DB 중심 데이터 통합

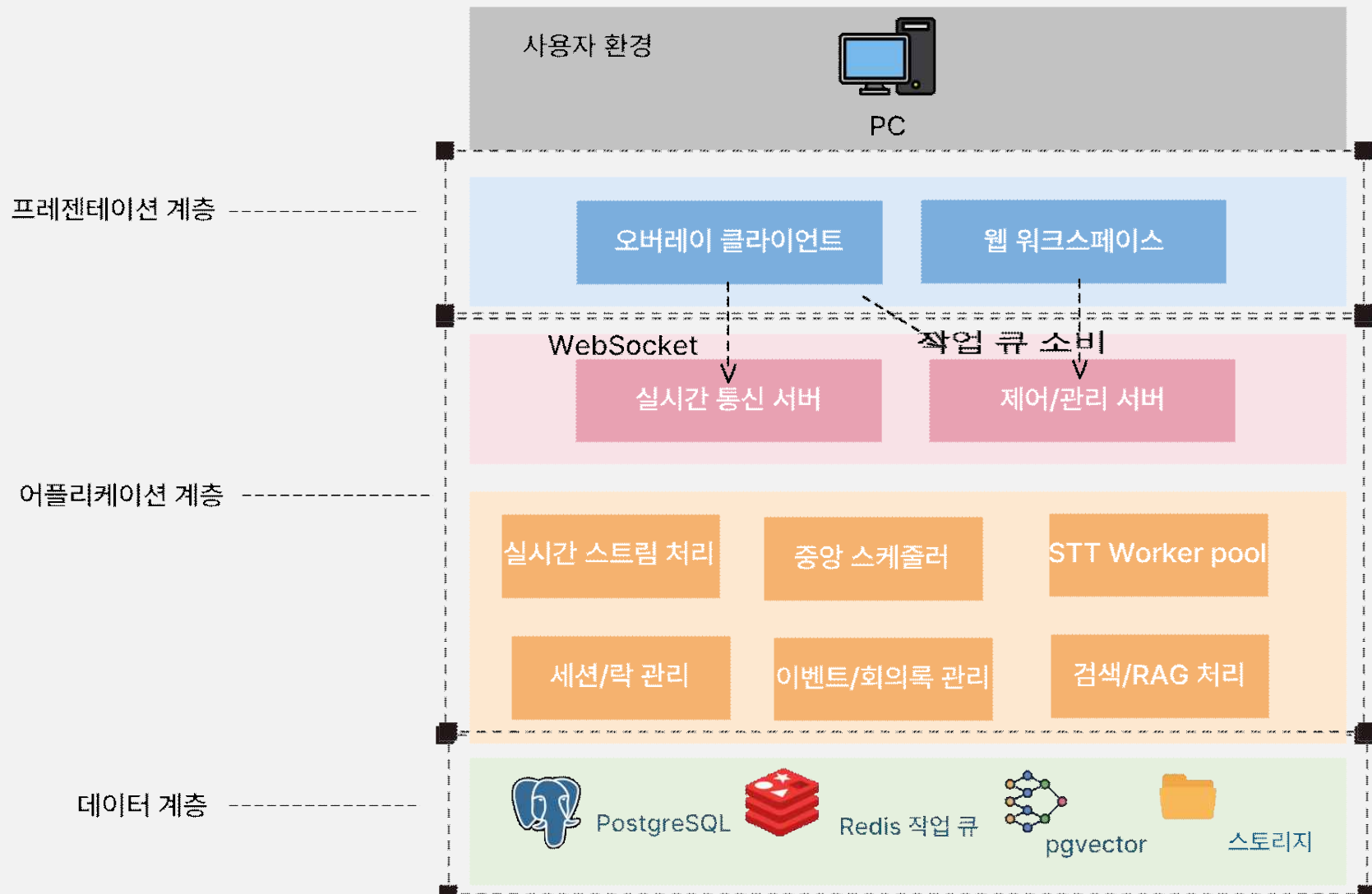
# 시스템 개요



# C/S 아키텍처



# 시스템 구조도



# | 기능표

대분류	단위기능	상세 설명	비고/제약사항
1. 회의 준비	1.1. 새 회의 생성	회의 단위 기록 생성	회의 전
	1.2. 회의 정보 입력	제목·참여자 정보 등록	회의 전
	1.3. 입력 소스 선택	마이크·시스템 오디오 선택	회의 전
	1.4. 회의 시작	실시간 기록 시작	회의 전
2. 회의 진행	2.1. 실시간 자막	발화 내용 실시간 표시	회의 중
	2.2. 핵심 질문 추출	질문 항목 분리 표시	회의 중
	2.3. 결정사항 추출	결정된 내용 정리	회의 중
	2.4. 할 일 추출	후속 업무 항목 정리	회의 중
	2.5. 회의 흐름 확인	현재 주제 및 흐름 요약	회의 중
	2.6. 워크스페이스 연동	회의 기록을 웹 워크스페이스와 연결	회의 중/후
3. 회의 종료/정리	3.1. 회의 종료	실시간 기록 종료 및 정리 전환	회의 종료 시
	3.2. 회의 내용 저장	자막·질문·결정사항·할 일 저장	회의 종료 시
	3.3. 회의록 생성	회의록 초안 자동 생성	회의 종료 후
	3.4. 회의록 수정	중요 내용 보완 및 정리	회의 종료 후
	3.5. 담당자·기한 정리	후속 업무별 담당자·기한 정리	회의 종료 후
	3.6. PDF 회의록 저장	전달·보관용 PDF 생성	회의 종료 후

# | 기능표

4. 기록 활용	4.1. 회의 목록 조회	저장된 회의 목록 확인	사후 활용
	4.2. 회의 상세 조회	자막·요약·회의록 열람	사후 활용
	4.3. 핵심 내용 조회	질문·결정사항·할 일만 확인	사후 활용
	4.4. 관련 회의 검색	주제·참여자 기준 검색	사후 활용
	4.5. 회의 맥락 이어보기	이전 회의 흐름 참고	사후 활용
5. 워크스페이스 활용	5.1. 보고서·회의록 관리	회의 결과 문서 모아보기	워크스페이스
	5.2. AI 챗봇	회의 기록 기반 질의응답	워크스페이스
	5.3. 후속 업무 관리	액션 항목 정리 및 확인	워크스페이스
6. 공유/협업	6.1. 워크스페이스 공유	사내 사용자 대상 공유	사내 공유
	6.2. 공유 기록 열람	공유된 회의 내용 확인	사내 공유
	6.3. 열람 이력 확인	열람 및 공유 이력 확인	사내 공유
7. 데이터 활용	7.1. 회의 기록 통합 저장	회의 데이터 통합 관리	핵심 기능
	7.2. 누적 기록 활용	검색·보고·후속 업무에 재활용	핵심 기능

# ER-다이어그램 (사용자 / 워크스페이스)

## users

PK: id

UNIQUE: login\_id

역할: 사용자 기본 정보

## auth\_sessions

PK: id

FK: user\_id → users.id

UNIQUE: token\_hash

역할: 로그인 세션 관리

## auth\_password\_credentials

PK/FK: user\_id → users.id

역할: 비밀번호 인증 정보

## workspaces

PK: id

UNIQUE: slug

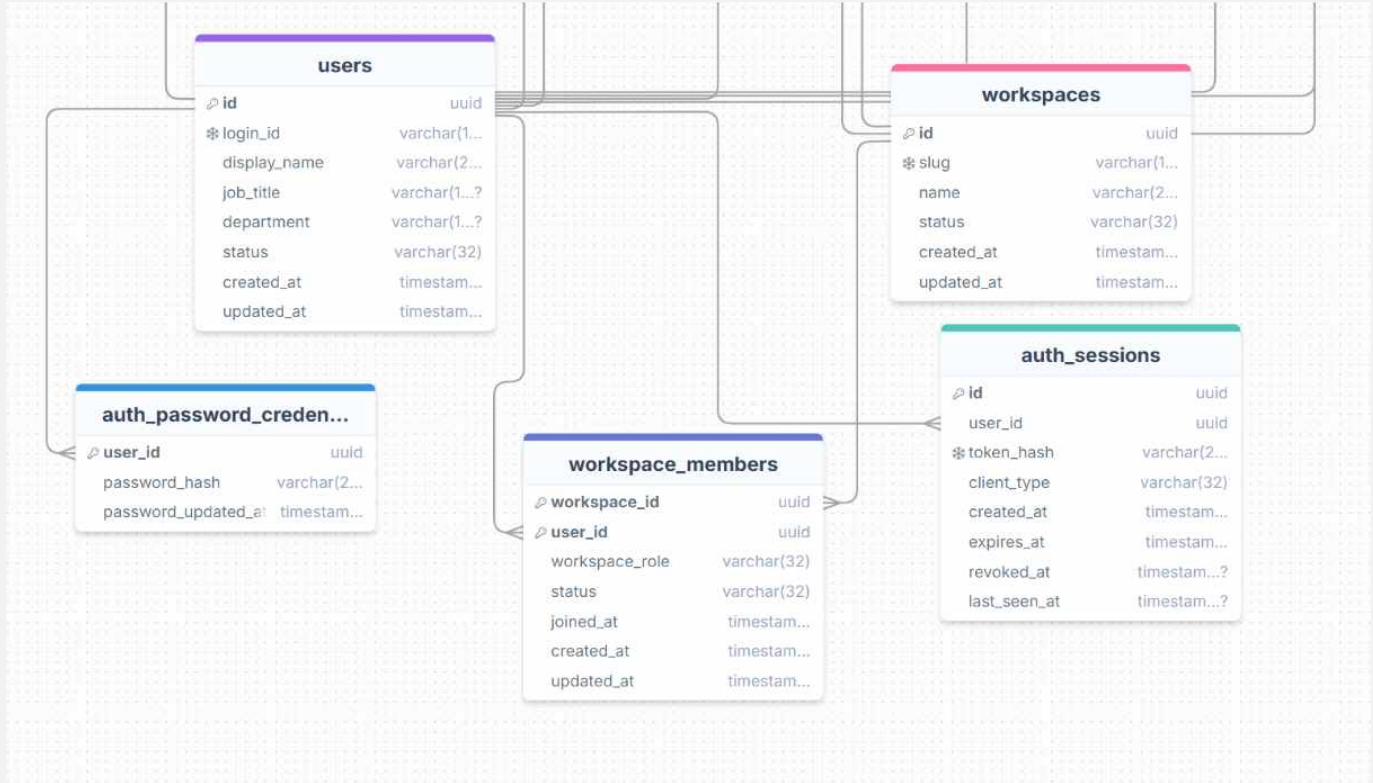
역할: 워크스페이스 기본 정보

## workspaces\_members

PK: (workspace\_id, user\_id) 복합키

FK: workspace\_id → workspaces.id, user\_id → users.id

역할: 워크스페이스-사용자 소속 관계



# ER-다이어그램 (참석자 / 맥락 관리)

## accounts

PK: id

FK: workspace\_id → workspaces.id,

created\_by\_user\_id → users.id

역할: 회사/고객사 단위

## contacts

PK: id

FK: workspace\_id → workspaces.id,

account\_id → accounts.id,

created\_by\_user\_id → users.id

역할: 고객사 내 인물 정보

## context\_threads

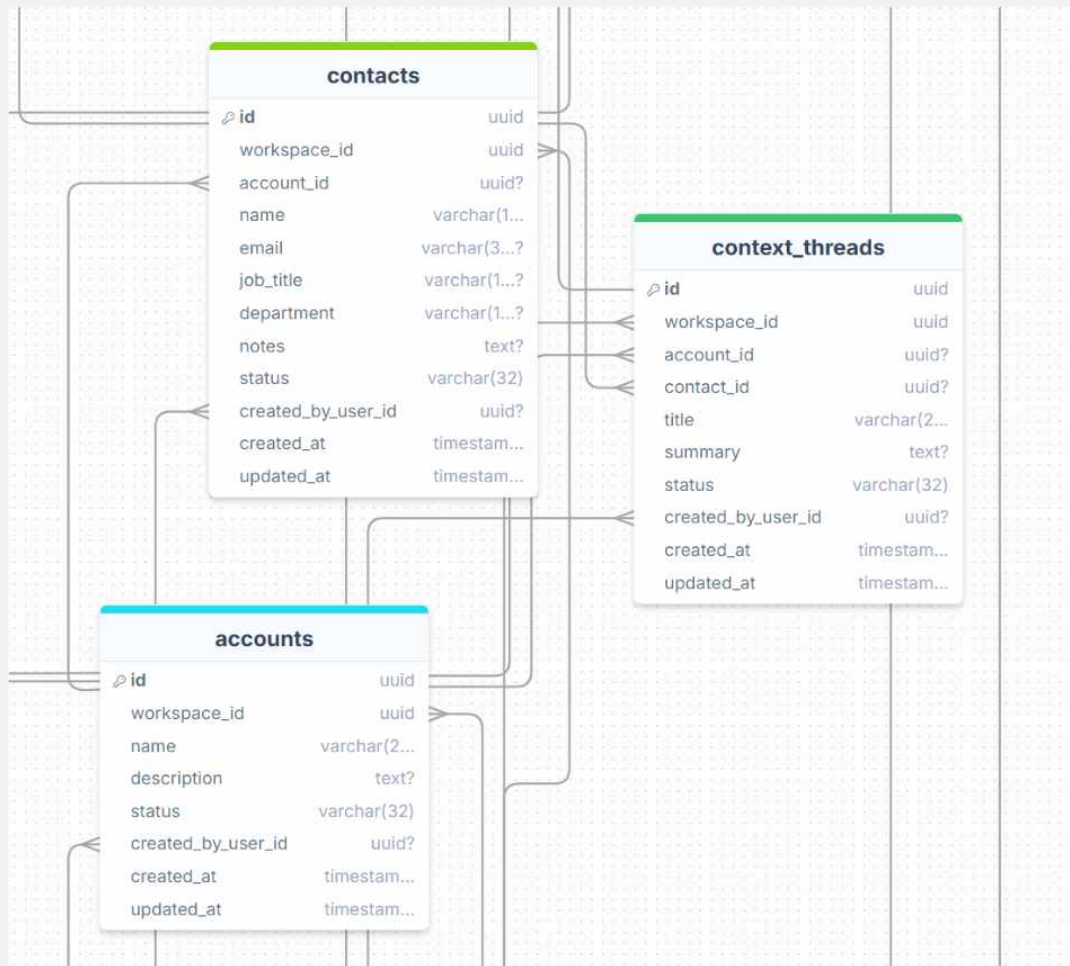
PK: id

FK: workspace\_id → workspaces.id,

account\_id → accounts.id, contact\_id →

contacts.id, created\_by\_user\_id → users.id

역할: 특정 회사/사람과 이어지는 대화 맥락



# ER-다이어그램 (회의 세션 중심 도메인)

## sessions

PK: id

FK: created\_by\_user\_id → users.id,

account\_id → accounts.id,

contact\_id → contacts.id,

context\_thread\_id → context\_threads.id

역할: 회의 1건의 중심 엔터티

## session\_participants

PK: (session\_id, participant\_order) 복합키

FK: session\_id → sessions.id,

contact\_id → contacts.id,

account\_id → accounts.id

역할: 세션 참여자 원본/매핑 정보

## participant\_followups

PK: id

FK: session\_id → sessions.id,

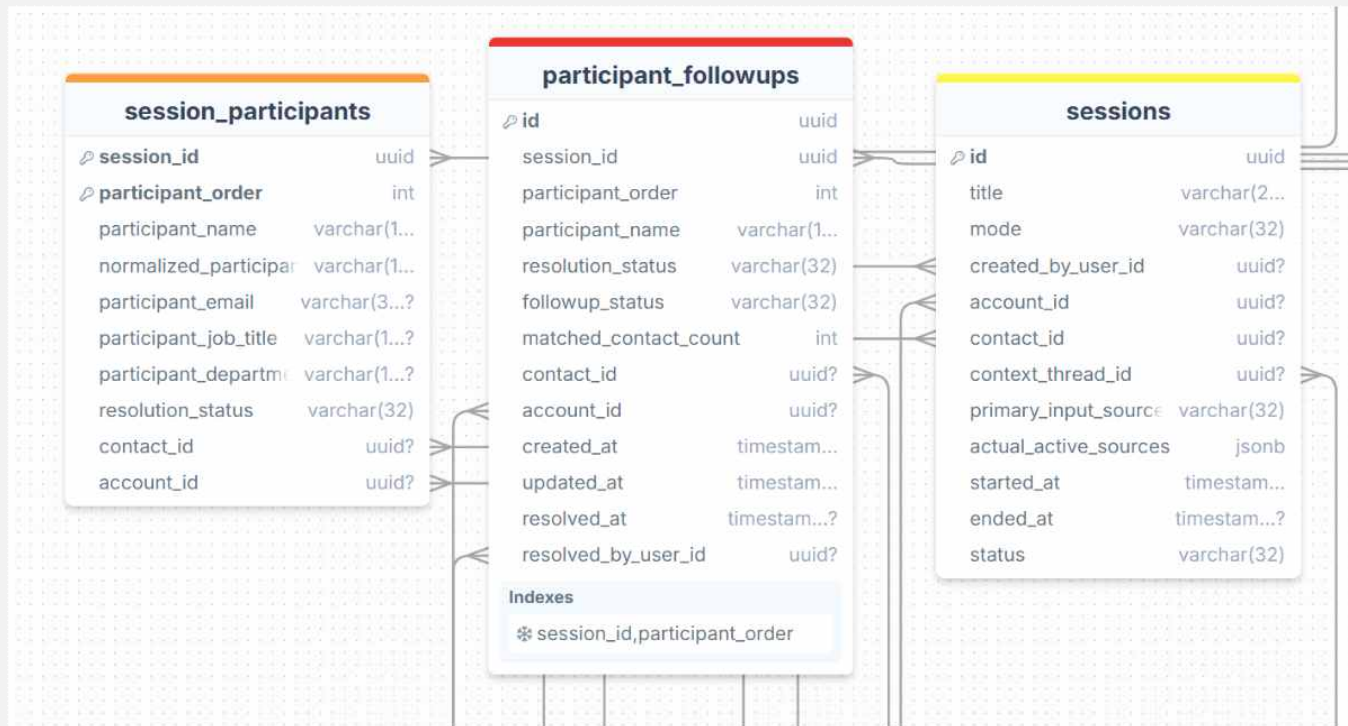
contact\_id → contacts.id,

account\_id → accounts.id,

resolved\_by\_user\_id → users.id

UNIQUE: (session\_id, participant\_order)

역할: 참여자 후속 정리 상태



## ER-다이어그램 (실시간 회의 기록 / 이벤트)

### utterances

PK: id

FK: session\_id → sessions.id

UNIQUE: (session\_id, seq\_num)

역할: 실시간 발화/자막 저장

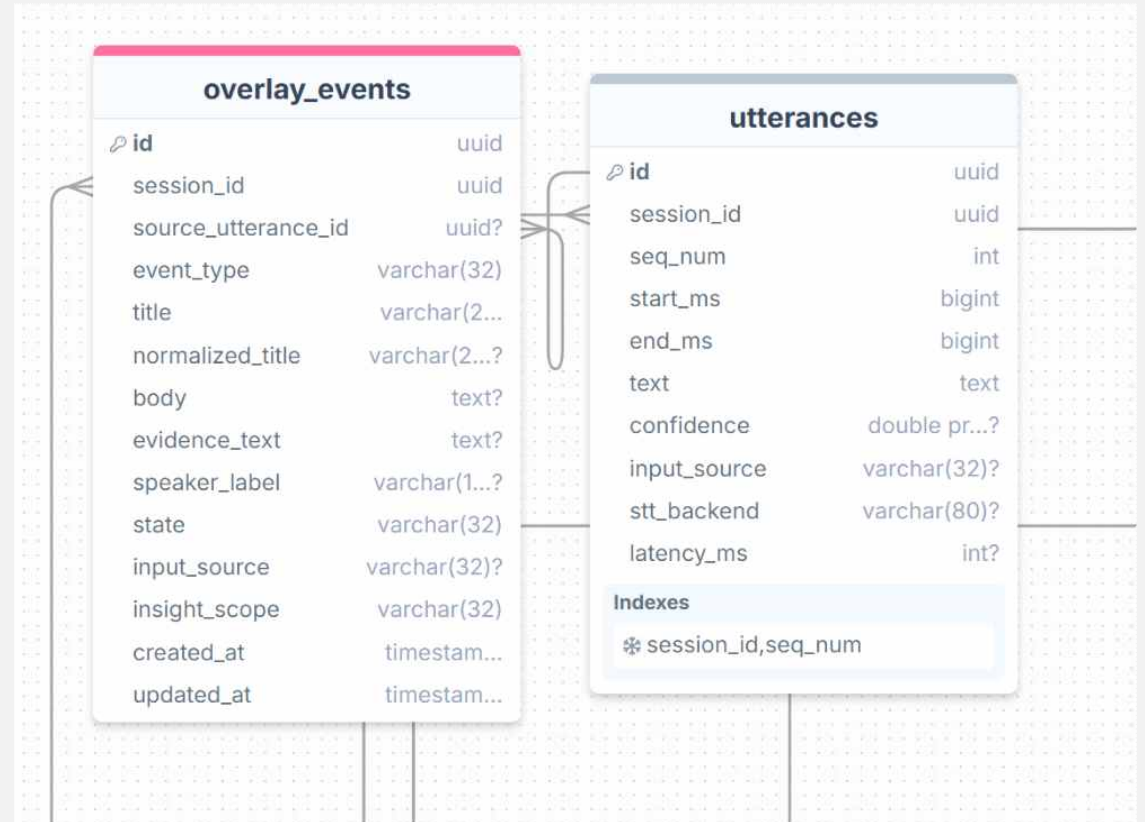
### overlay\_events

PK: id

FK: session\_id → sessions.id,

source\_utterance\_id → utterances.id

역할: 질문, 핵심 이벤트 저장



# ER-다이어그램 (리포트 / 결과물 / 공유)

## reports

PK: id

FK: session\_id → sessions.id,

generated\_by\_user\_id → users.id

UNIQUE: (session\_id, report\_type, version)

역할: 생성된 리포트 메타데이터

## report\_generation\_jobs

PK: id

FK: session\_id → sessions.id,

markdown\_report\_id → reports.id,

pdf\_report\_id → reports.id,

requested\_by\_user\_id → users.id

역할: 리포트 생성 작업 상태 관리

## report\_shares

PK: id

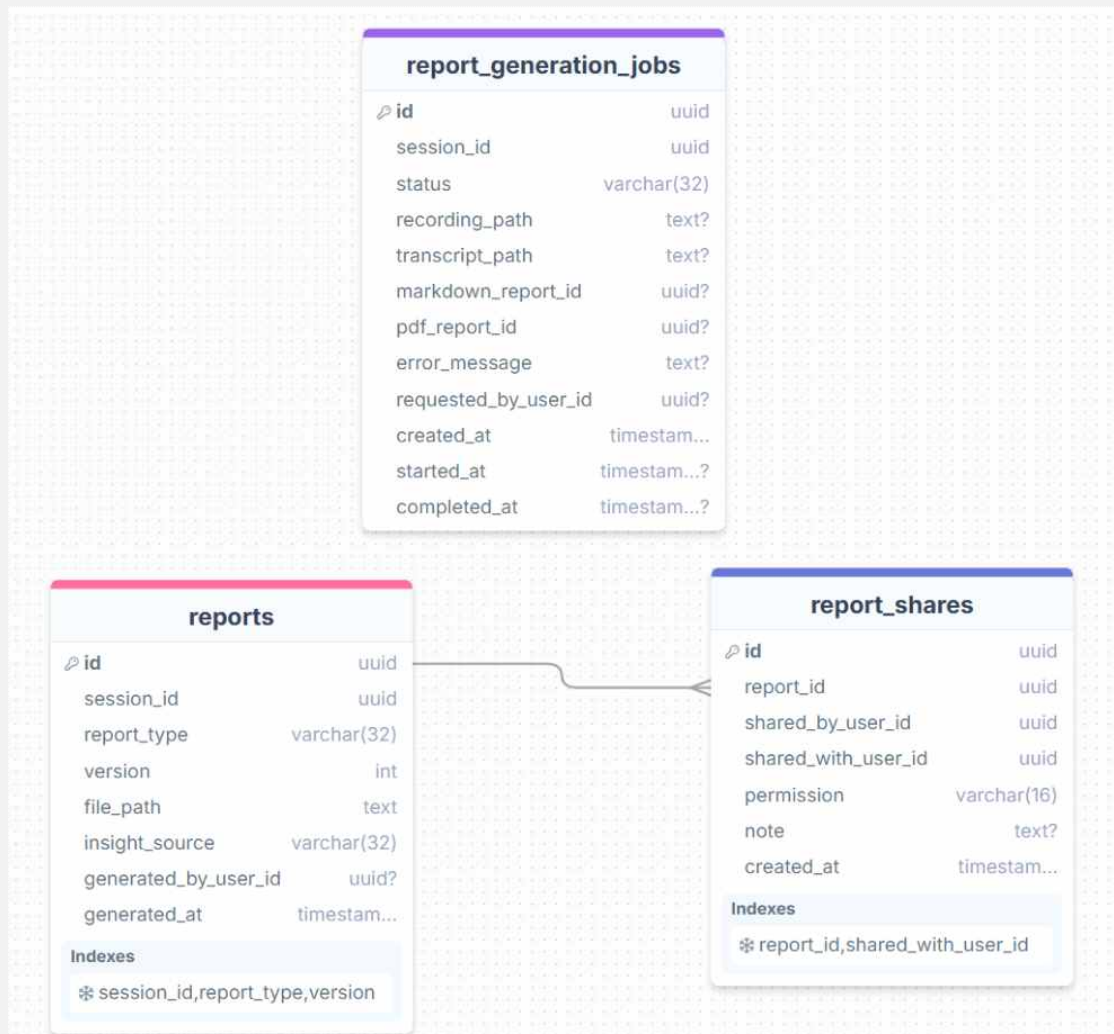
FK: report\_id → reports.id,

shared\_by\_user\_id → users.id,

shared\_with\_user\_id → users.id

UNIQUE: (report\_id, shared\_with\_user\_id)

역할: 리포트 공유 권한 관리



# ER-다이어그램 (검색 및 로그)

## knowledge\_documents

PK: id

FK: workspace\_id → workspaces.id,

session\_id → sessions.id,

report\_id → reports.id,

account\_id → accounts.id,

contact\_id → contacts.id,

context\_thread\_id → context\_threads.id

UNIQUE: (source\_type, source\_id)

검색 포인트: search\_tsv

역할: 검색 가능한 문서 단위

## knowledge\_chunks

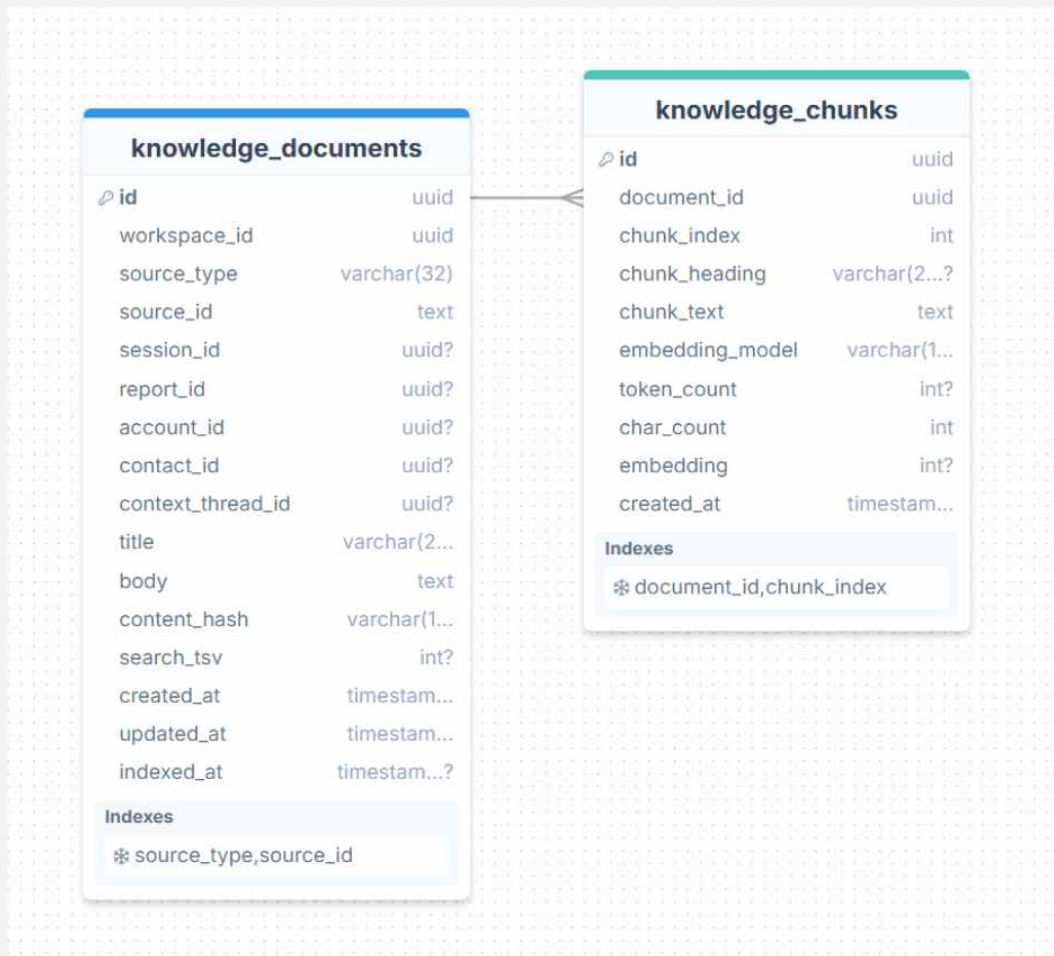
PK: id

FK: document\_id → knowledge\_documents.id

UNIQUE: (document\_id, chunk\_index)

검색 포인트: embedding VECTOR(768)

역할: 문서 청크 + 벡터 임베딩 저장



## 기술 검증(PoC) : 실시간 스트리밍 비교 기준

모델	첫 partial 지연	첫 final 지연	전체 RTF	최종 WER	최종 판단
Moonshine Tiny (ko)	2.3954s	4.3924s	2.6565	0.8571	X
Faster-Whisper (CT2)	-	11.0484s	1.0833	0.4286	O
Whisper Large-v3 Turbo (.pt)	1.2400s	15.0000s	1.9320	0.9714	X
Whisper Large-v3 Turbo	-	1.2000s	3.1003	1.0286	X
Sherpa-ONNX	0.0262s	0.5312s	0.2014	0.9429	O

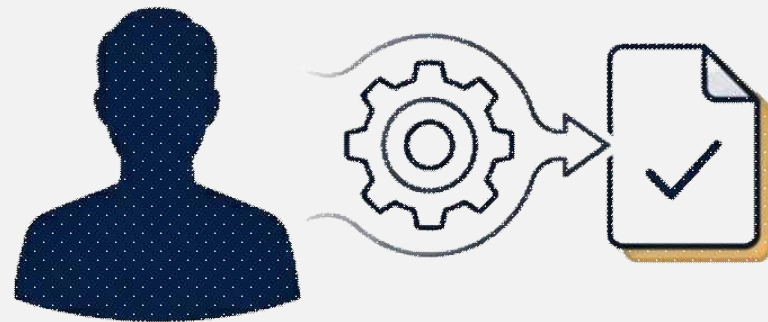
실시간성: Sherpa-ONNX  
 정확도: Faster-Whisper (GPU)



하이브리드 방식

\* WER: 단어 오류율  
 RTF: 처리 시간

## 기대효과

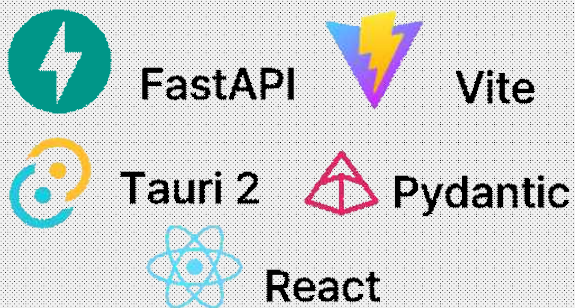


기존: 대화와 메모 분산 + 회의 후 요약

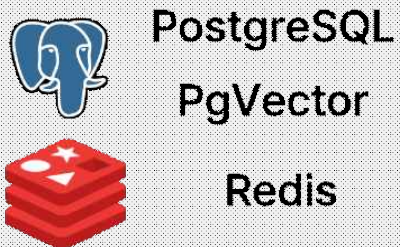
도입: 대화에 100%집중 +  
회의 후 자동 문서화

# 개발 환경

## 프레임 워크



## 저장소 / 큐



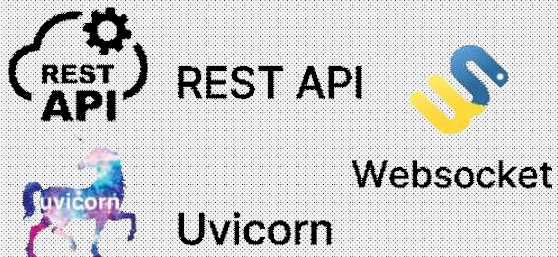
## 사용 언어



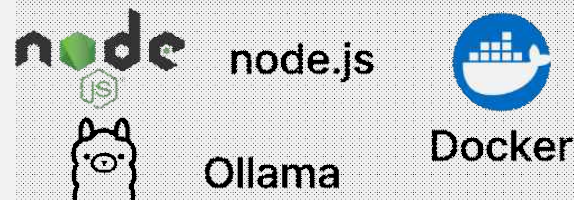
## AI 모델

Faster-Whisper (정밀 STT)  
Sherpa-ONNX (실시간 STT)  
Qwen2.5 3B Instruct (요약/챗봇)  
pyannote (화자 구분)

## 서버 구성



## 개발환경



## 개발 계획 및 질의 응답



감사합니다

202110763 신연준  
컴퓨터 공학과