

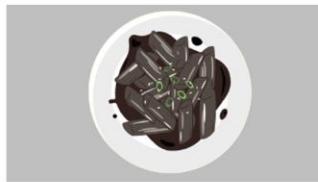
FINAL PROJECT WRAP-UP 리포트

CV-08 떡볶이

I. 프로젝트 개요

프로젝트명:	CueView (실시간 음성 인식 기반 지능형 발표 보조 솔루션)
기간	2025.01.07 ~ 2025.02.11 (5 주)
개요	전통적인 프레젠테이션 환경은 발표자의 음성 정보와 슬라이드의 시각 정보가 유기적으로 결합되지 못해 청중의 인지 부하를 가중시키는 고질적인 한계를 지니고 있습니다. CueView 는 이러한 '정보의 파편화'를 해결하기 위해 멀티모달 AI 와 실시간 스트리밍 기술을 결합한 전략적 솔루션입니다. 본 프로젝트는 발표자의 발화를 실시간으로 분석하여 슬라이드 내 핵심 요소를 자동 강조함으로써, 청중의 몰입감을 극대화하고 정보 전달의 효율성을 혁신적으로 개선하는 데 목적이 있습니다.

1.1 팀 R&R (Roles and Responsibilities)



김해찬

LLM Generation / Front-End

- LLM 프롬프트 설계
- 발표 대본 생성
- 대본 기능 FE 구현

문재영

RAG / Speech AI

- 검색 로직 설계
- STT-검색 파이프라인 구현
- VectorDB 구축

오연서

VLM Analysis / Front-End

- PPT 분석 프롬프트 설계
- STT 기능 개선
- UX 개선 및 FE 개발

정회성

Back-End / Infrastructure

- 서버 API 설계
- 모델 연동
- 배포 및 인프라 구축

1.2 프로젝트 타임라인

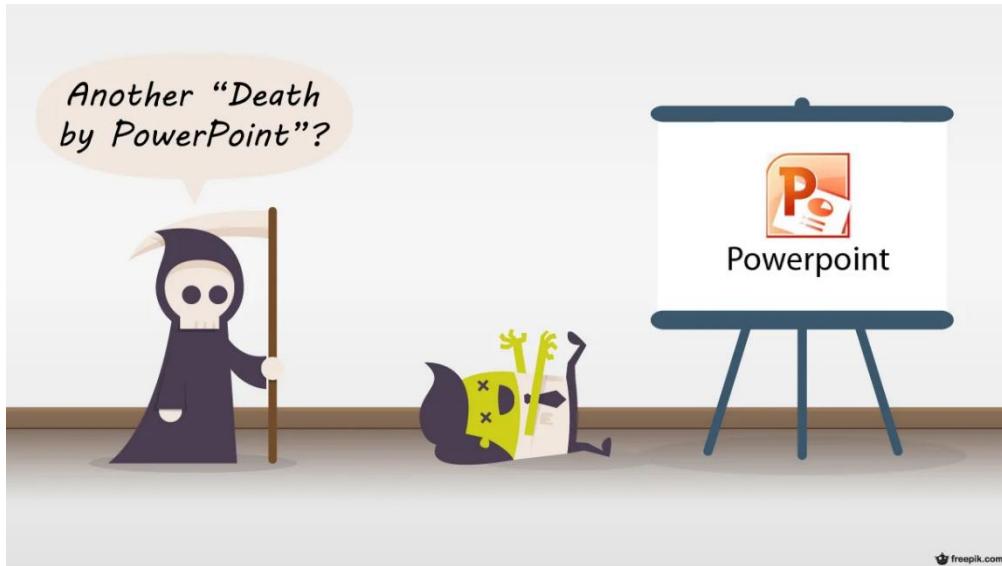


- 1~2 주차 (기획 및 리서치): 서비스 Flow 정의 및 기술 스택 검토
- 3 주차 (MVP 구현): Gemini-3-Pro 기반 요소 추출 엔진 및 기본 Vector 검색 환경 구축
- 4 주차 (고도화): Hybrid Search 도입, STT 이원화 트리거 및 RRF 가중치 투닝, 대본 생성 기능 통합
- 5 주차 (안정화): E2E 통합 테스트, 인프라 리소스 최적화 및 최종 성능 평가

본 프로젝트는 단순한 기능 구현을 넘어, 기존 발표 환경의 인지적 병목 현상을 기술적으로 정의하고 이를 해결하기 위한 정교한 아키텍처를 설계하는 데 집중했습니다.

II. 서비스 소개

2.1 문제 정의: 인지적 멀티미디어 학습 이론(CTML) 관점



기존의 프레젠테이션은 'Death by PowerPoint'로 대변되는 높은 인지 부하 문제를 안고 있습니다. 발표자의 음성과 시각 자료 간의 불일치는 청중이 정보를 매칭하기 위해 뇌 용량을 낭비하게 만드는 외재적 인지 부하(Extraneous Cognitive Load)를 초래합니다.

CueView는 이를 해결하기 위해 다음의 기술적 가설을 설정했습니다.

- **시간적 균질성 원리 적용:** 음성 설명과 시각적 강조를 실시간으로 동기화하여 정보 처리 효율을 극대화한다.
- **주의 분산 방지:** 청중이 시각적 단서를 찾는 수고를 자동화함으로써 학습과 본질적 정보 이해를 위한 본래적 인지 부하(Germane Load)에 집중하게 한다.

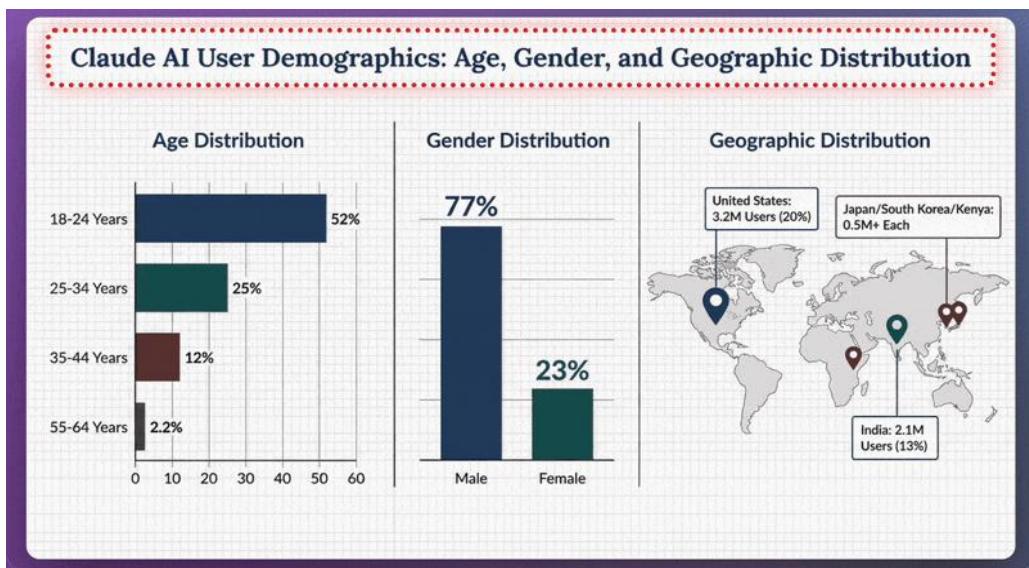
2.2 핵심 솔루션 개요



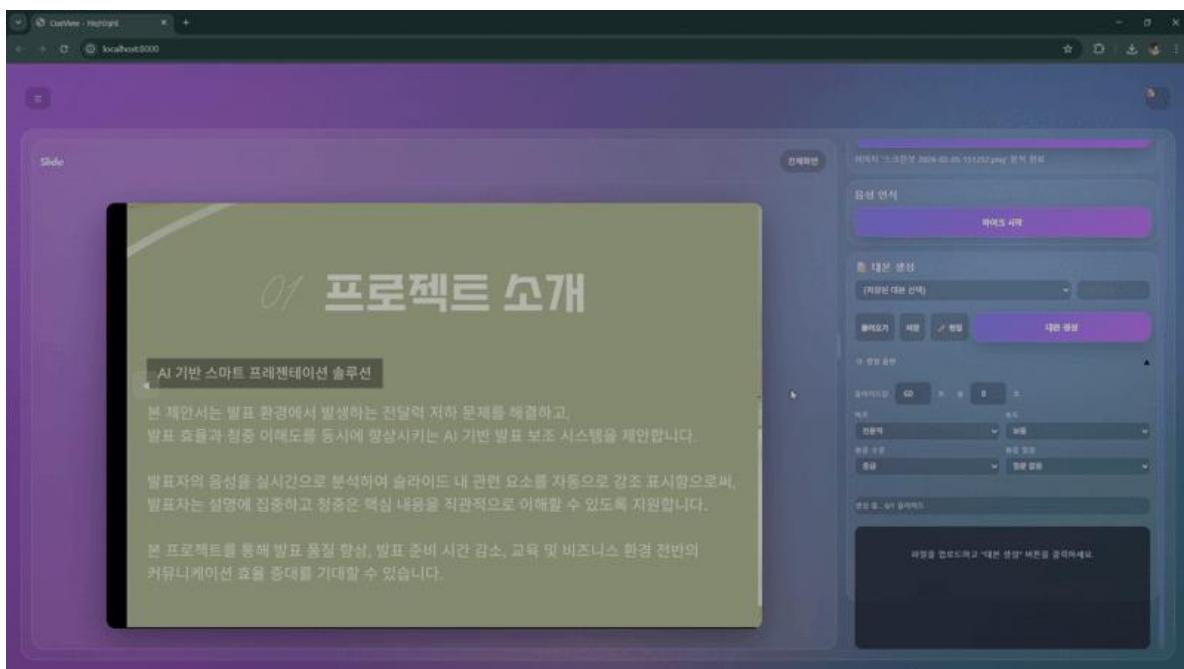
발표자의 음성을 실시간으로 인식하여 슬라이드 내 텍스트, 차트, 이미지 등 의미 단위(Semantic Block)를 자동으로 강조하는 지능형 인터페이스를 제공합니다.

2.3 핵심 기능 상세

- **실시간 지능형 하이라이트:** BGE-M3 임베딩과 BM25 검색을 결합하여 발표자의 의도를 0.5 초 내외의 지연 시간으로 슬라이드 위에 시각화합니다.



- **AI 자동 대본 생성 파이프라인:** Gemini-3-Pro 를 통한 슬라이드 구조화와 HyperCLOVA X(HCX-007)의 생성 능력을 결합합니다. 사용자의 옵션(톤, 청중 수준 등)에 맞춰 HTML 기반의 가독성 높은 대본을 자동 생성하며 즉각적인 편집을 지원합니다.

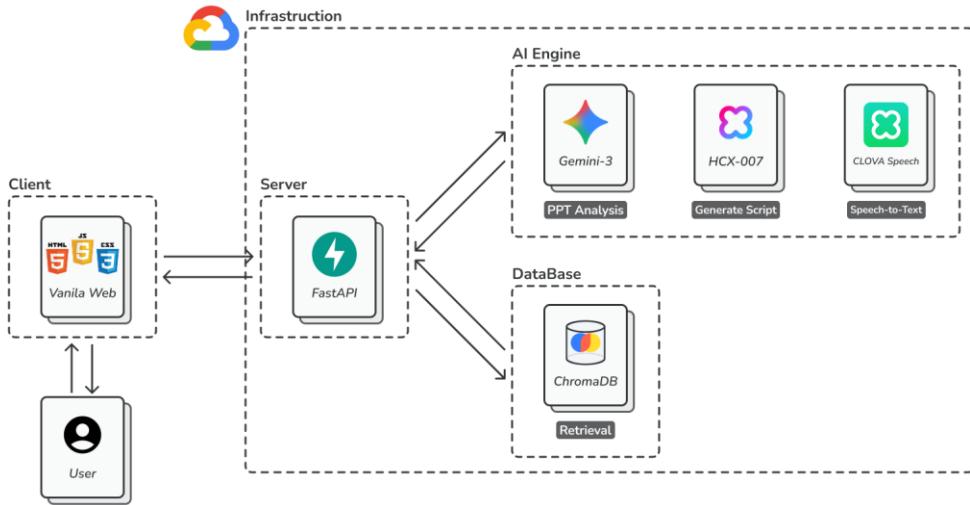


III. 기술 설계 및 아키텍쳐

3.1 시스템 아키텍처 및 데이터 흐름

CueView 는 고성능 AI 엔진과 실시간 백엔드 구조를 유기적으로 결합했습니다.

- **Client:** Vanilla Web 기반 하이라이트 오버레이 및 폴링 기반 대본 상태 관리
- **Server:** FastAPI 를 활용한 비동기 처리 및 WebSocket 통신
- **AI Engine:** Gemini-3-Pro(슬라이드 분석), HCX-007(대본 생성), CLOVA Speech(STT)
- **Database:** ChromaDB(Dense) 및 slide-specific BM25 Index(Sparse)



3.2 전처리 파이프라인 (VLM & Prompt Engineering) + 대본 생성 전략

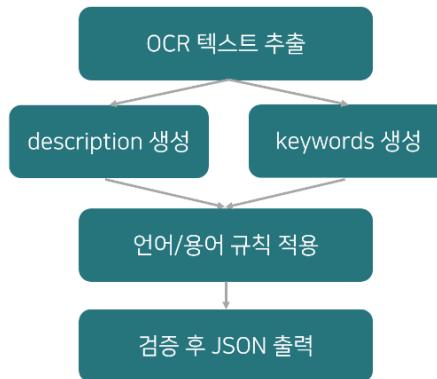
슬라이드 이미지에서 의미 있는 메타데이터를 추출하기 위해 **Gemini-3-Pro**를 활용하여 정교한 전처리를 수행합니다.

- **Semantic Block Grouping:** 텍스트 박스, 이미지, 차트 등을 독립적인 의미 단위로 그룹핑하여 0~1000 정규화 좌표(bbox)를 산출합니다.
- **검색 최적화 메타데이터:** 검색 정확도를 위해 content(OCR), description(발표용 요약), keywords(구체적 명사/숫자 포함)를 추출합니다. 이 과정에서 추상어 대신 숫자와 단위를 포함하도록 강제하여 검색 변별력을 높였습니다.

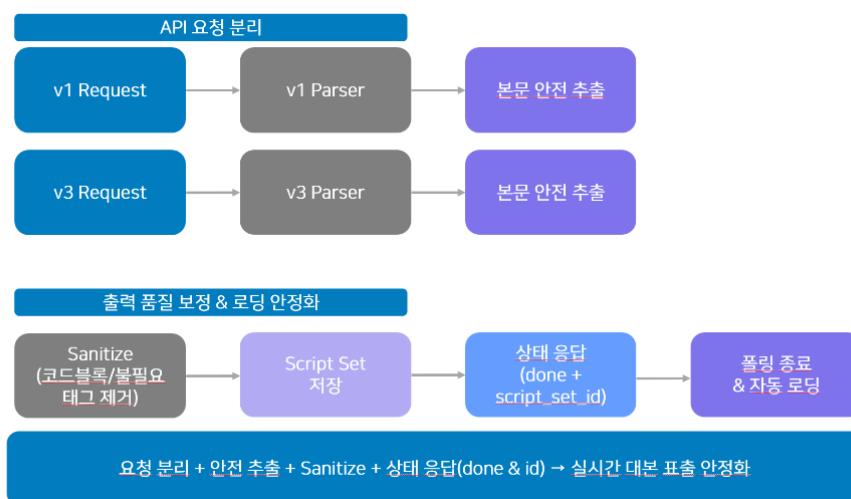
[요소 추출 + 의미 단위 그룹핑]



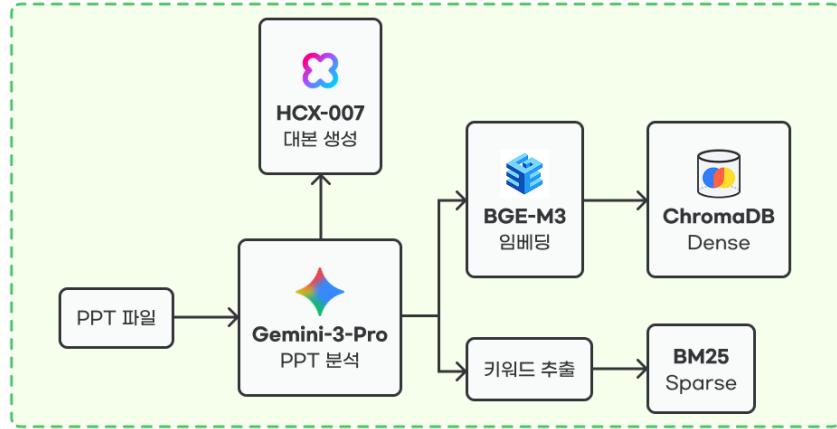
[메타데이터 생성 + 검색 최적화 룰]



- **대본 생성 api 안정화 전략:** 클로바 스튜디오 모델 사용에 있어서 HCX-003 모델과 HCX-007 모델 사용 실험 중 요청 및 응답 규격이 혼재된 구조 때문에 일부 환경에서는 요청이 제대로 수행되지 않은 문제가 있었습니다. 출력 품질과 모델 로딩 안정화를 위해 모델 별 요청 분리를 수행했습니다.



1. PPT 분석 및 대본 생성



3.3 실시간 STT 전략

- **STT 엔진의 전환:**
 - Web Speech API에서 CLOVA Speech로 초기 MVP 단계에서는 브라우저 내장 기능인 Web Speech API를 사용했으나, 다음과 같은 한계가 존재했습니다.
 - 한계: 브라우저별로 인식 방식이 상이하고 정확도가 중간 수준이며, 특히 검색 트리거를 세밀하게 제어(단어 수, 문장 종결 여부 등)하기 어려웠습니다.
 - **CLOVA Speech 도입:** 한국어 인식 정확도를 높였습니다.
- **이원화 검색 트리거:** 0.5 초 묵음 방식의 불안정성을 탈피하여 '2.5 초 누적 + 문장 종결' 트리거를 도입함으로써 발표 호흡에 따른 유연한 검색을 구현했습니다.



3.3 실시간 하이브리드 검색 엔진 (Retrieval)

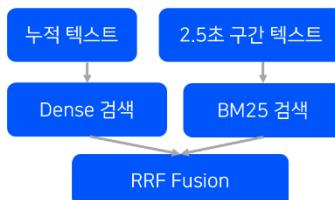
단일 검색 방식의 성능 한계를 극복하기 위해 Dense(문맥)와 Sparse(키워드) 검색 전략을 이원화하여 도입했습니다.

- **하이브리드 쿼리 전략 (Cumulative vs Discrete):**
 - **Dense (BGE-M3):** 문맥 이해를 위해 음성 스트리밍 텍스트를 누적하여 검색합니다.
 - **Sparse (BM25):** 쿼리가 길어질수록 키워드 가중치가 희석되는 문제를 방지하기 위해 해당 구간의 단발성 텍스트만 사용하여 정확도를 유지합니다.
- **BM25 성능 최적화:** 전체 문서를 순회하는 $O(n \times m)$ 복잡도를 해결하기 위해 슬라이드별 독립 인덱스를 구축하여 검색 범위를 국소화하고 속도를 개선했습니다.
- **RRF(Reciprocal Rank Fusion) 기반 가중치 투팅 Logic:**
 - **Decision Logic:** Rank 1과 Rank 2의 RRF 점수 차이가 약 0.0002 인 점을 분석했습니다. 기존 15% 가중치(+0.0025)는 유사도 차이를 압도해버리는 부작용이 있어, 최종적으로 **8% (+0.0013)** 가중치를 적용하여 안정성과 정확도의 균형을 맞췄습니다.

- **가중치 구성:** 첫 발화 시 초기 안정화 가중치(+4.8%)를 부여하고, 이후 현재 요소 8%, 인접 슬라이드 Window 3 범위 내(90/70/50%) 가중치를 차등 적용합니다.

01.

하이브리드 검색

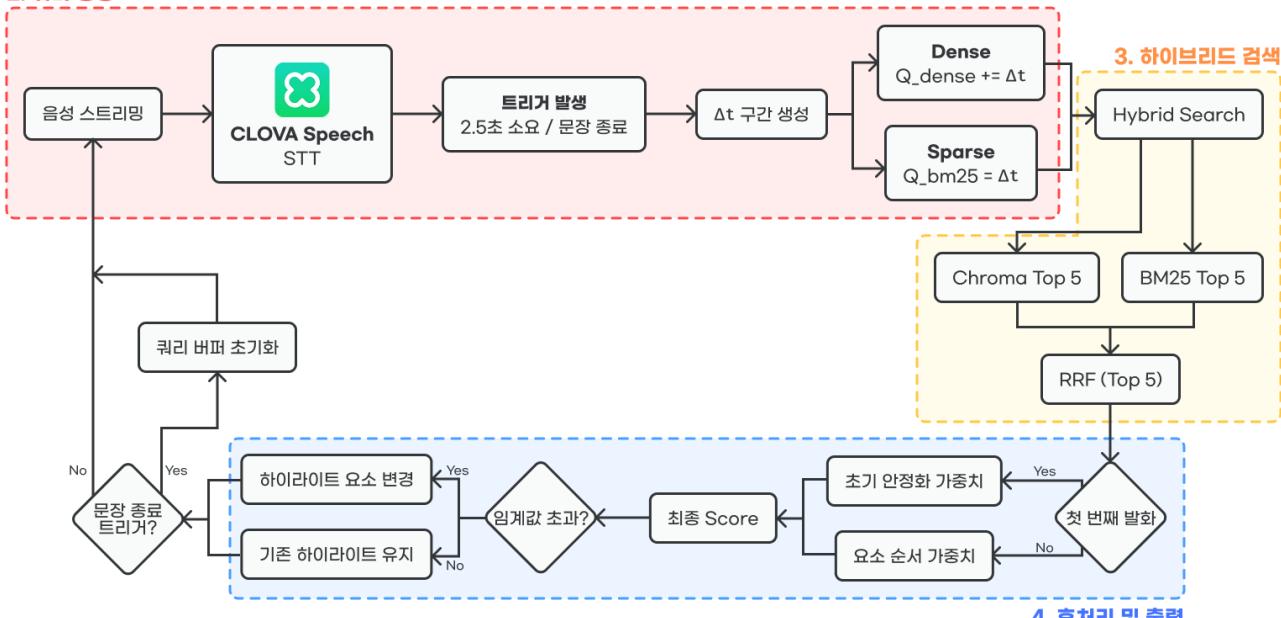


02.

위치 정보 가중치 + 점수 최적화



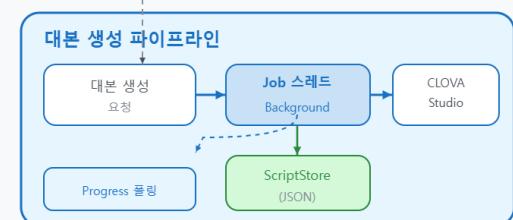
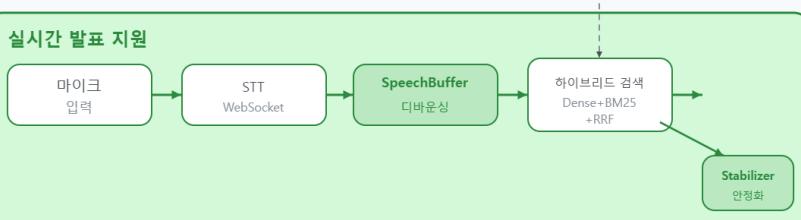
2. 쿼리 생성



3.4 백엔드 구현 핵심 (Infrastructure)

백엔드는 크게 파일 업로드 파이프라인, 실시간 발표 지원 파이프라인, 그리고 대본 생성 파이프라인으로 구성되어 있으며, 전체 흐름은 다음과 같습니다.

전체 시스템 아키텍처



WebSocket + Thread Bridge

Per-Connection 상태 관리 (Lock)
`asyncio.run_coroutine_threadsafe()`
 Thread ↔ Async 안전한 브릿지
 → 다중 사용자 실시간 처리

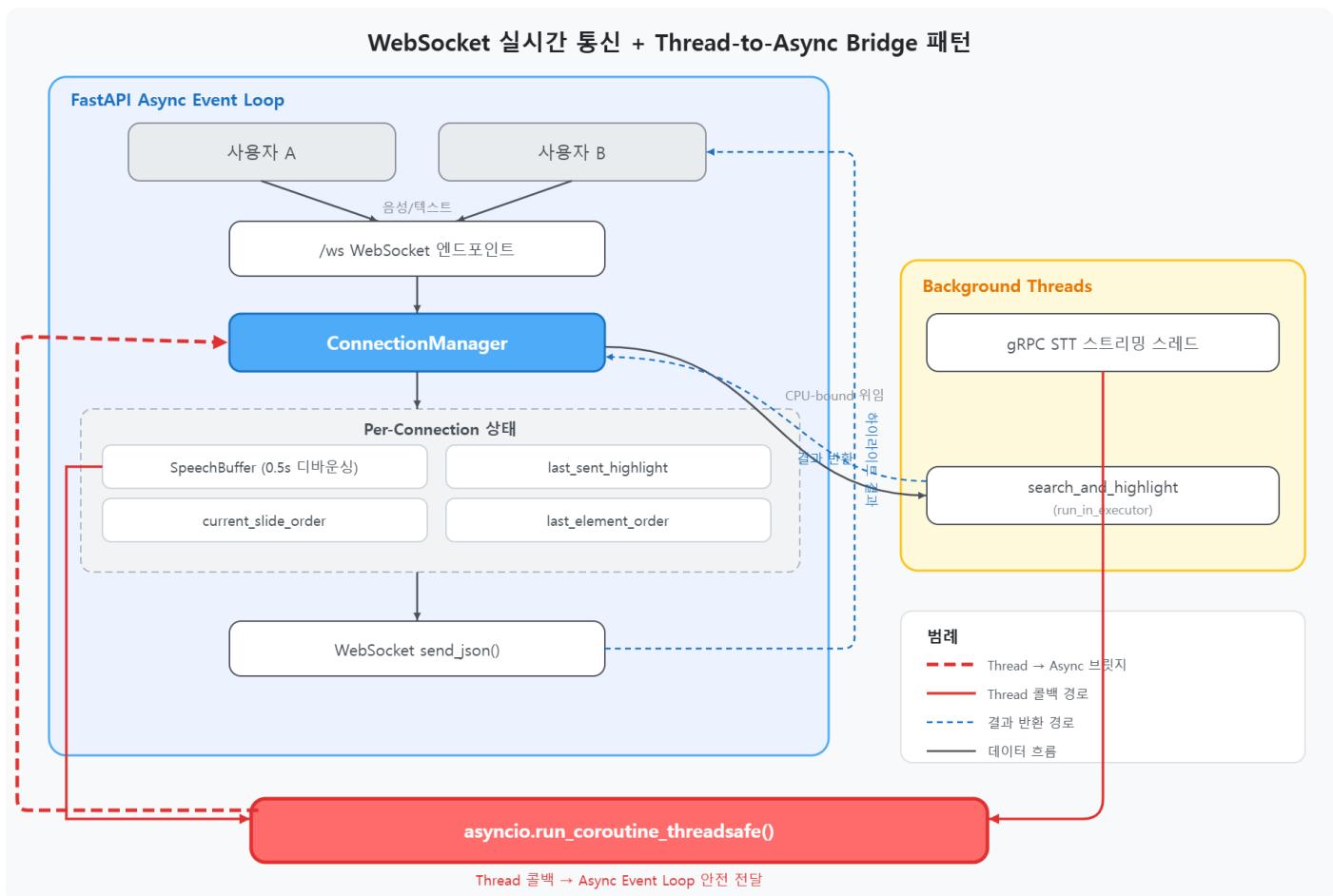
3단계 UX 안정화

SpeechBuffer 디바운싱
 HighlightStabilizer sticky 로직
 중복 전송 제거
 → 화면 깜빡임 방지

백그라운드 Job 시스템

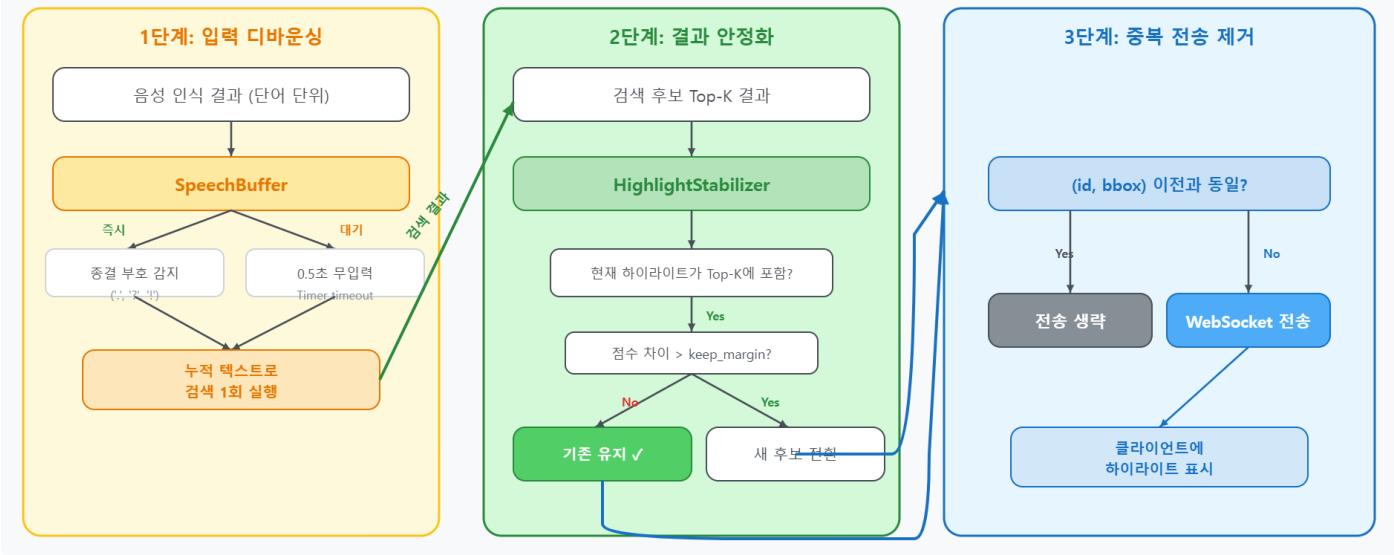
Thread-safe 상태 관리 (Lock)
 queued → running → done 상태 마신
 Progress Polling 기반 주적
 → Non-blocking 장시간 작업

- Thread-to-Async Bridge:** CueView는 사용자의 음성을 실시간으로 입력받아 즉시 하이라이트 결과를 반환해야 하기 때문에, HTTP 풀링 방식이 아닌 WebSocket 기반 양방향 통신으로 설계됐습니다.
 - gRPC 기반의 CLOVA Speech 스레드와 비동기 WebSocket 루프 간의 통신을 위해 `asyncio.run_coroutine_threadsafe()` 패턴을 적용했습니다. 이를 통해 I/O 바운드 작업의 논블로킹 처리를 보장했습니다.
 - ConnectionManager** 를 설계해 세션별 음성 버퍼 및 상태를 관리하며, 연결 종료 시 모든 리소스를 일괄 정리하여 메모리 누수를 방지하는 안정적 구조를 설계했습니다. 하나의 WebSocket 연결마다 음성 버퍼, 현재 슬라이드 위치, 마지막 하이라이트 상태, 토큰 이력 등을 분리해 유지함으로써 다중 사용자 동시 접속 환경에서도 안정적인 처리가 가능하도록 했습니다.



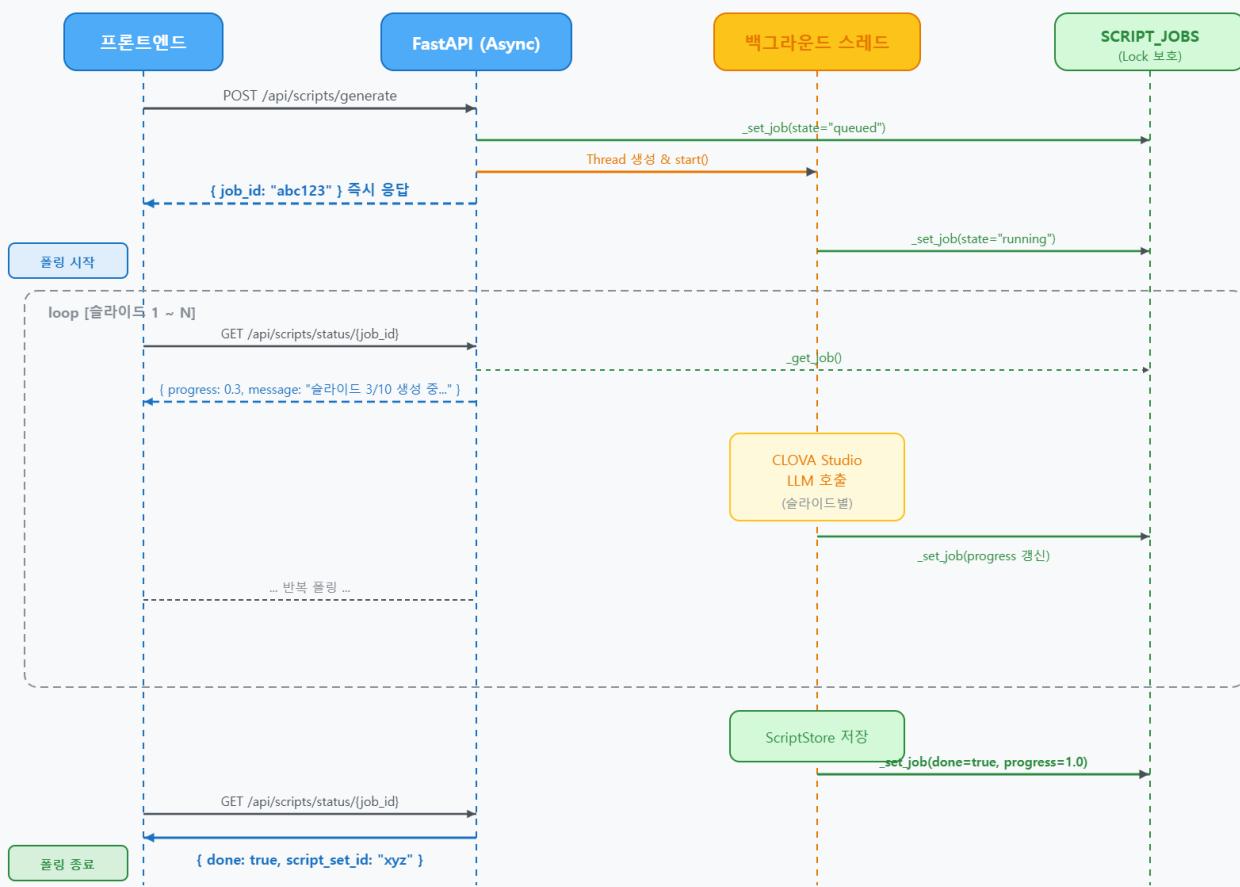
- UX 안정화 3 단계 파이프라인:** 실시간 음성 기반 서비스에서 "검색 결과가 너무 자주 바뀌어 화면이 깜빡이는 문제"는 치명적인 UX 이슈입니다. 이를 백엔드 레벨에서 3 단계 UX 안정화 파이프라인을 적용해 해결했습니다.
 - 첫 번째 단계에서 `SpeechBuffer` 를 구현해 0.5 초 디바운싱을 적용했습니다. 이를 통해 불필요한 검색 호출을 줄이고 입력을 안정화했습니다.
 - 두 번째 단계에서는 검색 결과 상위 후보들의 점수가 미세하게 변하면서 순위가 자주 바뀌는 문제를 해결하기 위해 `HighlightStabilizer` 를 구현했습니다. 현재 하이라이트가 여전히 상위 후보에 포함되어 있다면 이를 유지하는 `sticky` 로직을 적용해 시각적 흔들림을 최소화했습니다.
 - 세 번째 단계에서는 동일한 하이라이트 결과의 중복 전송을 제거해 사용자에게 불필요한 업데이트가 전달되지 않도록 했습니다.

디바운싱 → 안정화 → 중복 제거 (3단계 UX 파이프라인)



- **백그라운드 Job 시스템과 Progress Polling:** 발표 대본 생성은 슬라이드마다 LLM API를 호출해야 하므로 수십 초~수 분이 소요됩니다. 이를 비동기 백그라운드 Job으로 처리하고, 프론트엔드가 폴링으로 진행 상태를 추적하는 구조를 설계했습니다.

백그라운드 Job 시스템 (대본 생성)



IV. 결과 및 회고

4.1 목표 대비 달성을 분석

본 프로젝트는 PDF 문서의 시각적 요소 추출부터 실시간 STT 기반 하이라이트 구현까지, 당초 계획했던 E2E 파이프라인을 완벽히 확보했습니다

- **End-to-End 프로세스 완성:** 문서 업로드 → 이미지화 → VLM 요소 추출 → 실시간 검색/하이라이트 → 대본 생성으로 이어지는 전체 파이프라인 구축했습니다.
- **성능 지표 수립 및 최적화:**
 - 슬라이드 분석 시간 단축 및 대본 생성 성공률(빈 HTML 방지) 극대화하였습니다.
 - STT-하이라이트 간 지연 시간 최소화 및 검색 정확도(Top-1 적중률) 향상하였습니다.
- **검색 알고리즘 고도화:**
 - Hybrid Search(Dense + BM25) 도입을 통해 키워드와 문맥 매칭 정확도를 동시에 확보하였습니다.
 - STT 트리거 최적화(2.5 초 지연 + 문장 종결 로직)를 통해 실시간 발표 상황에서의 안정성 강화하였습니다.
- **시스템 안정성 확보:** Singleton 및 Lifespan 패턴을 적용하여 메모리 점유율을 절감하고, 저사양 클라우드 환경에서도 다중 사용자 세션 격리 및 동시 접속(2 인 이상) 안정성 확보하였습니다.

4.3 팀원별 회고 및 Lesson Learned

[기술적 교훈]

- LLM 운영의 정교화: 프롬프트 엔지니어링을 넘어 응답 규격(v1/v3) 관리, HTML 정규화, 파싱 보강 등 사후 처리 프로세스가 서비스 품질에 결정적임을 확인하였습니다.
- 아키텍처의 유연성 vs 관리: 클린 아키텍처를 통한 모듈화는 유지보수에 유리하나, 실시간 모델 로드가 필요한 환경에서는 리소스 관리 전략과 긴밀히 결합되어야 함을 이해하였습니다.
- 인터페이스 정의의 중요성: 개발 초기 단계에서 좌표 규격(0~1000 정규화) 및 메시지 타입을 사전 정의하는 '약속'이 협업 병목을 줄이는 핵심임을 체득하였습니다.
- UX 중심의 기술 설계: 단순히 검색 정확도를 높이는 것보다, 사용자 체감 품질을 결정하는 UX 안정성과 디버깅이 쉬운 단순한 로직 설계가 실전에서 더 효율적임을 이해하였습니다.

[협업적 교훈]

- 사전 타당성 검토: 주제 선정 단계에서의 기술적 타당성 조사와 구현 범위 합의가 전체 팀의 효율성을 좌우함을 이해하였습니다.
- 공유 프로세스의 체계화: '문제-가설-실험-결론' 구조로 진행 상황을 공유할 때 팀 내 커뮤니케이션 비용이 가장 낮아지고 결과물의 완성도가 높아짐을 체득하였습니다.

4.4 향후 고도화 목표

[Short-term] 기능 강화 및 UX 개선

- **대본-검색 연동 심화:** 대본을 검색 인덱스 근거 텍스트로 활용하여, 발표자의 발화가 슬라이드 원문뿐 아니라 대본과도 매칭되도록 개선.
- **리허설 모드 도입:** 사용자가 직접 하이라이트 트리거 및 좌표를 보정할 수 있는 사전 연습 기능 추가.
- **발표자 프리뷰:** 하이라이트와 연동된 '다음에 말할 포인트' 미리보기 기능을 통해 발표 흐름의 연속성 지원.

[Mid-term] 지능형 서비스 확장

- **추가 설명(Explain) 모드:** "더 설명해줘"와 같은 특정 발화 시, 관련 요소 기반의 추가 설명 텍스트를 실시간 생성하여 사이드 패널에 표시.
- **피드백 루프 구축:** 발표 종료 후 타임스탬프 로그와 사용자 피드백을 결합 분석하여 하이라이트 정확도를 지속적으로 고도화.

[Long-term] 아키텍처 및 성능 최적화

- **분산 처리 아키텍처:** 단일 서버 구조를 탈피하여 임베딩 및 VLM 분석 부하를 분산하는 작업 할당 모델 도입.

- **검색 성능 극대화:** Reranker 도입 및 MRL(Matryoshka Representation Learning) 기반 임베딩 최적화로 검색 속도와 정확도 동시 확보.
- **멀티모달 확장:** 음성뿐만 아니라 발표자의 제스처나 시선을 결합한 다중 모달리티 검색 트리거 구현.

V. 결론

CueView 는 AI 가 발표자의 의도를 실시간으로 해석하고 청중의 시각적 경험을 능동적으로 가이드하는 혁신적인 솔루션으로서, AI 보조 도구의 새로운 표준을 제시했습니다.

단순히 정보를 나열하는 기존의 발표 도구를 넘어, 발표자의 목소리와 슬라이드의 시각적 요소를 유기적으로 연결하는 이 파이프라인을 구축하기 위해 우리 팀은 수많은 기술적 도전과 마주해 왔습니다. VLM 의 요소 추출 정확도를 높이기 위한 반복적인 실험부터, 0.1 초의 지연 시간이라도 줄이기 위한 하이라이트 트리거 최적화, 그리고 제한된 자원 내에서 안정적인 서비스를 제공하기 위한 아키텍처 설계에 이르기까지 모든 과정에 팀원들의 치열한 고민과 노력을 녹여냈습니다.

비록 프로젝트는 현재의 MVP 구현을 통해 그 가능성을 충분히 입증했으나, 실제 현장에서의 완성도를 극대화하기 위해 보완하고 고도화해야 할 과제들 또한 명확히 확인하였습니다. 우리는 이번 프로젝트를 통해 얻은 기술적 자산과 협업의 경험을 발판 삼아, 향후 더욱 정교한 검색 인덱싱과 사용자 중심의 리허설 모드, 그리고 분산 처리 아키텍처를 통한 시스템 확장을 지속적으로 추진해 나갈 것입니다.

CueView 는 앞으로도 발표자와 청중 사이의 간극을 좁히고, 기술이 인간의 전달력을 극대화하는 '진정한 지능형 파트너'로 거듭날 수 있도록 중단 없는 발전을 이어가겠습니다.