

LivekitDev 아키텍처

| 작성일: 2026-07-19 · 한글 복구: 2026-09-07

| 초기 아바타 데모의 설계 기록입니다. 이후 Meet 및 Windows 서비스 구성은 관련 가이드를 참고합니다.

컴포넌트 경계

사용자 브라우저

```
|
+-- HTTPS --> Caddy --> 웹 UI (Flutter / Meet)
|
+-- WSS -----> Caddy --> LiveKit SFU :7880
+-- WebRTC -----> LiveKit 미디어 포트
                        ^
                        | 영상·음성 트랙 발행
                    Avatar Bot (Python)
                        ^
                        | PyAV 파일 디코딩
                    avatar_bot/videos/avatar.mp4
                        ^
                        | 오프라인 영상 생성
                    Wav2Lip / SadTalker
                    GFPGAN / CodeFormer
```

Caddy는 HTTPS 및 시그널링 프록시를 담당하고 LiveKit은 참가자 사이의 미디어와 데이터를 중계합니다. Avatar Bot은 미리 만든 영상을 읽어 방에 발행합니다.

데이터 흐름

A. 실시간 미디어

1. 클라이언트가 JWT로 LiveKit 방에 입장합니다.
2. 로컬 카메라와 마이크 트랙을 발행합니다.
3. 아바타를 포함한 원격 참가자의 트랙을 구독합니다.
4. LiveKit SFU가 WebRTC 미디어를 중계합니다.

B. 아바타 제어 (DataChannel)

1. Flutter가 avatar-control 토픽으로 video:off 등의 명령을 보냅니다.
2. 봇이 비디오 또는 오디오 트랙의 음소거 상태를 변경합니다.
3. 봇이 avatar-status 토픽으로 상태를 알립니다.
4. Flutter가 토글 UI 상태를 동기화합니다.

C. 오프라인 콘텐츠 제작

1. videos/의 얼굴 영상과 오디오를 입력으로 준비합니다.
2. Wav2Lip으로 립싱크를 생성하고 필요하면 GFPGAN 또는 CodeFormer로 복원합니다.
3. 결과를 avatar_bot/videos/avatar.mp4에 배치합니다.
4. 봇이 파일 수정 시간 변경을 감지해 영상을 다시 불러옵니다.

SadTalker는 사진과 오디오를 이용한 대체 제작 경로입니다.

인증 모델 (초기 Flutter 데모)

- 개발 키: devkey / secret
- Flutter의 TokenGenerator가 클라이언트에서 HS256 JWT에 서명합니다.
- 봇은 같은 키를 사용해 서버 SDK의 AccessToken을 생성합니다.

이 구조는 클라이언트 번들에 secret이 포함되는 한계가 있습니다. 운영용 Flutter 구성에는 서버 측 토큰 발급 API가 필요합니다. 별도 Meet 앱은 서버의 connection-details API를 사용합니다.

배포 토폴로지

계층	위치 및 역할
공개 도메인	meet.dev.tictechtoeai.com
TLS 종료	Caddy (초기 WSL 구성, 이후 Windows 네이티브 서비스)
SFU	Windows LiveKit 실행 파일
Bot	같은 머신의 Python 프로세스
초기 정적 웹	Flutter build/web를 Caddy로 제공
이후 Meet UI	Next.js 3000 포트를 Caddy로 프록시

확장 방향 (초기 설계 제안)

방향	설명
Flutter 서버 토큰	짧은 수명의 JWT를 발급하는 API
실시간 립싱크	마이크 또는 TTS에서 생성한 프레임을 봇이 발행
여러 아바타	방과 identity별로 다른 영상 또는 에이전트 사용
Reflex 통합	초기 Caddy의 3000/8000 경로 활용 구상
경로 통일	실행기와 설정을 현재 C 드라이브 작업 경로로 정리

위 표는 구현 완료 목록이 아닙니다. 배포 변경 이력은 Meet 연동과 Windows 서비스를 참고합니다.