범인찾기 · 리플레이 동기화 계획서

AI가 경기 기록을 읽고 10명 평가 · 팀별 범인을 찾습니다.
리플레이를 보면 그때 한 말을 시간에 맞춰 보여 줍니다.

프로젝트	겜녹이 (Game Voice Recorder) · v1.2.2 다음 기능
문서	계획서 v1.2 · 정해 주신 8가지 반영 (v1.1 + 결정 사항 · 실제 화면 · 토큰 실측)
진행	범인찾기 C1~C3(앱 · 사이트 서버 · 화면) 구현 완료, 배포 전 · 리플레이 동기화는 API 확인 뒤 결정
작성일	2026-10-15
AI	Claude API (Anthropic) · 기본 모델 Claude Opus 5.5
예산	API 크레딧 $100 (2026-11-15 만료)



gvr.tkrmagid.kr · 표지 화면은 실제 구현 화면(시연 데이터 + 실제 Claude Opus 5.5 분석 결과)입니다

1 한눈에 보기

237경기
$100로 분석할 수 있는 경기 수
(Opus 5.5 · 가정한 조건에서 10번 중 9번 232~242)

$0.40
경기당 평균 비용
(10판 중 8판은 $0.30~$0.52)

1~2분
버튼을 누르고 결과가 나올 때까지

8판/일
크레딧 만료(11/15)까지 다 쓰려면
하루에 분석해야 하는 경기 수

- **무엇을 하나**: 사이트에 🔍 **범인찾기** 탭을 새로 만듭니다. 끝난 경기를 고르고 **범인 분석**을 누르면, AI가 그 경기의 음성 대화 · 킬 · 라운드/오브젝트 · 경제 · 채팅 · 10명 스탯을 모두 읽고 아래를 만들어 줍니다.
  - 10명 전부 평점(10점 만점), 별명 태그(캐리 · 숨은 MVP · 무난 · 범인), 한 줄 평
  - 팀별 범인 1명씩(우리 팀 · 상대 팀), 왜 범인인지 설명, 근거 3~4개
  - 근거마다 실제 순간(대화 · 킬 · 라운드)이 연결돼 누르면 타임라인에서 바로 듣고 볼 수 있음
- **대상 경기**: 끝났고 기록이 모두 남아 있는 경기만 고를 수 있습니다. 녹음이 중간에 시작됐거나, 경기 결과를 못 받았거나, 음성 변환이 덜 끝난 경기는 이유를 보여 주고 막습니다 (4장).
- **범인**: 팀마다 1명. **가장 못한 사람 + 가장 발목을 잡은 사람 + 결과(패배)의 원인**을 모두 합쳐 종합으로 고릅니다 (3장).
- **AI는 사이트 서버가 부릅니다**. Claude Opus 5.5 · 말투 중간맛. API 키는 받아서 서버 비밀 저장소에만 두었고, 배포 때 사이트 서버 환경 변수로만 넣습니다. exe에는 넣지 않습니다 (5장).
- **분석 버튼**은 등록된 앱 유저 이상만 누를 수 있습니다(다시 분석은 관리자). 결과는 로그인한 멤버 모두가 봅니다.
- **비용**: 경기 길이와 말하는 양을 무작위로 바꿔 가며 2만 경기를 계산했습니다. Opus 5.5로 **$100이면 약 237경기**입니다 (지시문 · 지식 묶음 토큰을 실제로 재서 다시 계산, v1.1의 248경기에서 조정). 방이 조용하면 약 266경기, 말이 많으면 약 203경기입니다 (7장).
- **예산 보호**: 월 $100 상한, 하루 횟수 제한, 경기당 최대 약 $1 상한, 한 경기는 한 번만 분석(다시 분석은 관리자만)을 둡니다.
- **프롬프트 · 지식 묶음은 미리 써 두었습니다**: 지시문, 발로란트 · 롤 지식 묶음(규칙 · 경제 · 역할 기준 · 은어 · 음성 인식 오류 · 최신 요원 · 챔피언 목록), 출력 형식. 매 분석마다 같이 보내 AI가 "배운 상태"로 분석합니다 (6장 · 부록).
- **멤버 공지**: 녹음 안내(TTS) 문구에 "녹음된 대화는 AI 경기 분석에 쓰일 수 있어요"를 넣었습니다. 기본 문구를 쓰던 PC는 새 문구로 자동으로 바뀝니다.
- **리플레이 동기화**: 게임 안 리플레이를 보면 exe가 알아채고 "리플레이와 동기화하시겠습니까?"를 묻는 기능입니다. **게임이 API(공식 · 비공식)로 리플레이 시간을 알려 줄 때만 넣습니다.** 화면 글자 인식(OCR)이나 게임 설정 파일 고치기는 하지 않습니다. 그래서 먼저 Windows PC에서 롤 · 발로란트가 어떤 API를 주는지 확인(R0)하고, 되는 게임만 만듭니다 (8장). AI를 쓰지 않아 비용이 없습니다.

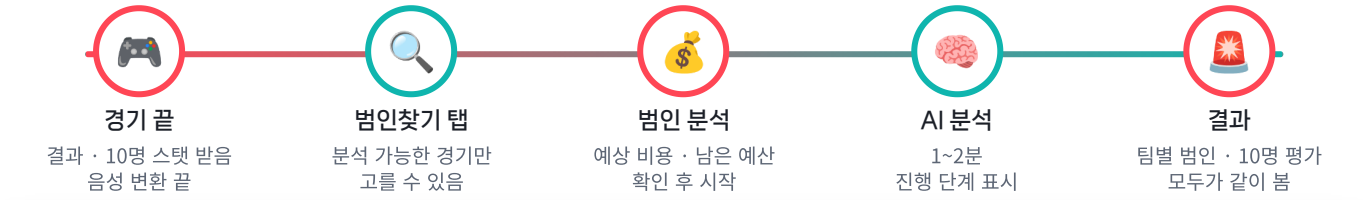
> **정한 것 (2026-10-15 답변 · 자세한 내용은 10장)**
> ① 범인 = 가장 못함 + 가장 발목 + 패배 원인 종합 ② 분석 버튼 = 등록된 앱 유저 이상 ③ 모델 = Opus 5.5 ④ API 키 = 받음(서버에만) ⑤ 말투 = 중간맛 ⑥ 공지 = 녹음 안내(TTS) 문구에 추가 ⑦ 발로란트 리플레이 = API에 없으면 넣지 않음(OCR 안 함) ⑧ 롤 리플레이 = API에 없으면 넣지 않음(설정 파일 안 고침)

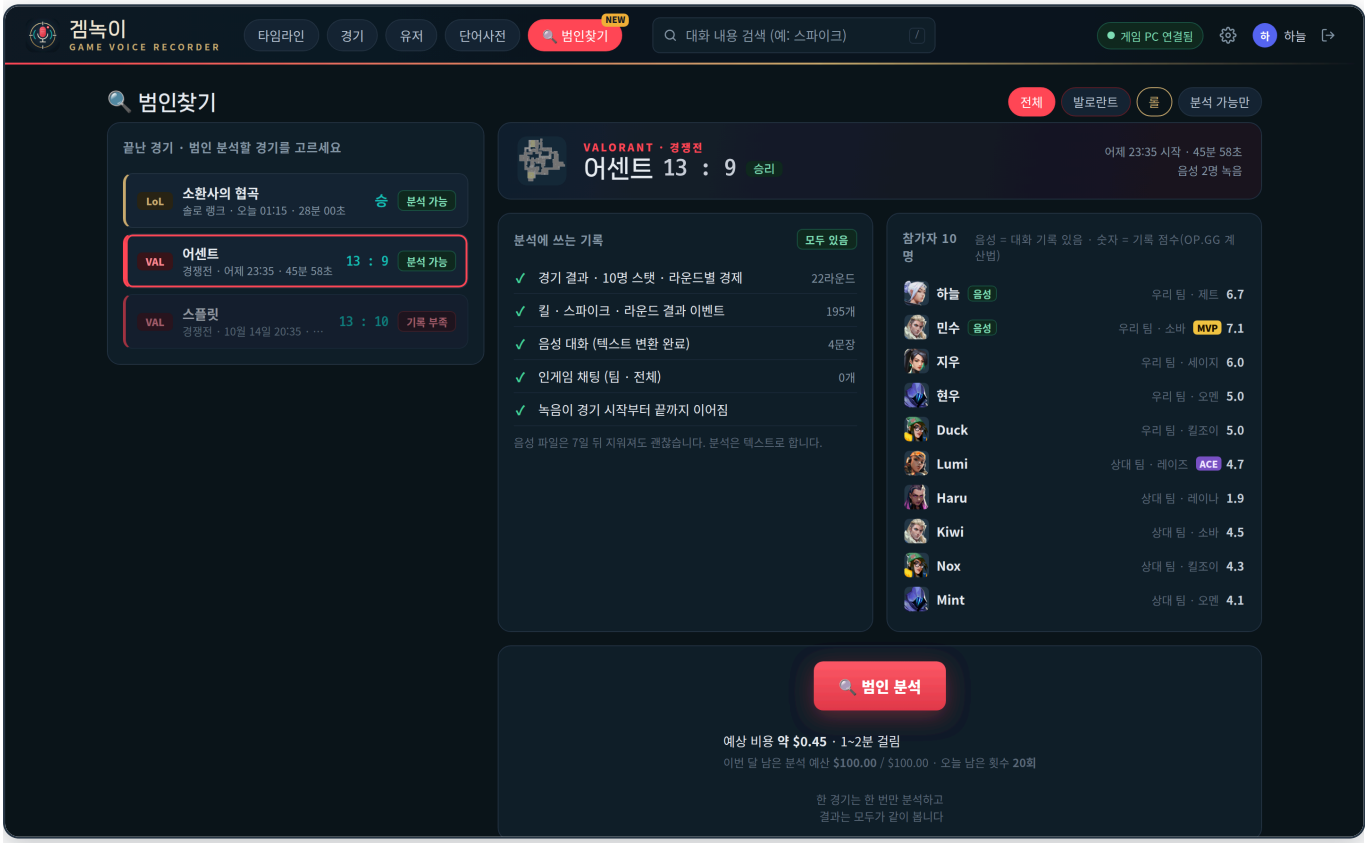
목차

1. 한눈에 보기
2. 쓰는 흐름 · 화면 예시
3. 범인을 고르는 기준
4. 분석에 쓰는 기록 · 분석할 수 있는 경기
5. 구조 · 보안 · 개인정보

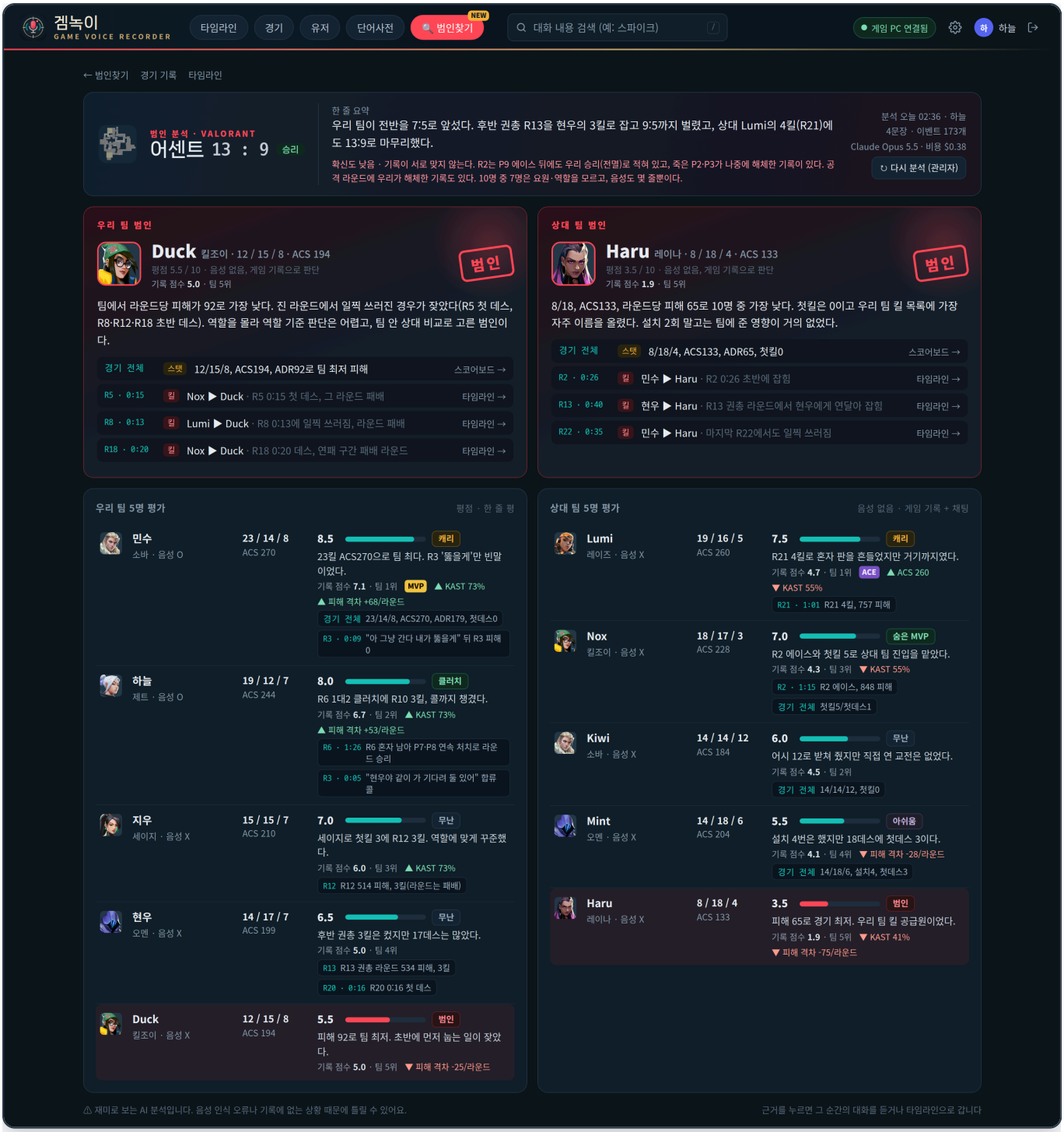
6. 범인찾기 프롬프트 · 지식 묶음
7. 비용 예상: $100로 몇 경기?
8. 리플레이 동기화
9. 만드는 순서 · 확인 방법
10. 정한 것 · 남은 일
11. 부록: 프롬프트 · 지식 묶음 · 출력 형식 원문

2 쓰는 흐름 · 화면 예시





화면 1 · 범인찾기 탭 (경기 고르기) 왼쪽은 끝난 경기 목록입니다. 경기마다 **분석 가능 · 분석 완료 · 준비 중 · 기록 부족**을 표시하고, 기록이 부족하면 이유("녹음이 12분 뒤에 시작됨" 등)를 보여 줍니다. 오른쪽은 고른 경기에서 분석에 쓰는 기록과 참가자 10명(음성이 있는 사람 표시), 예상 비용 · 남은 예산 · 오늘 남은 횟수입니다. 실제 화면 · 시연 데이터



화면 2 · 분석 결과 (발로란트) 맨 위는 한 줄 요약, 가운데는 팀별 범인 카드(이유 + 근거), 아래는 10명 평가입니다. 근거의 대화는 **▶ 듣기**로 그 문장 음성을 바로 재생하고(7일 보관 기간 안), 킬 · 라운드는 타임라인으로 이동합니다. 상대 팀처럼 음성이 없는 사람은 "게임 기록으로 판단"이라고 밝힙니다. 결과 글은 실제 Claude Opus 5.5가 쓴 것입니다. 시연 데이터의 앞뒤가 안 맞는 곳을 AI가 스스로 찾아 확신도를 "낮음"으로 적었습니다. 실제 화면 · 시연 데이터



화면 3 · 분석 결과 (휴대폰) 이긴 경기에서도 팀마다 범인을 뽑습니다. 휴대폰에서는 요약, 범인 카드, 10명 평가를 차례로 보여 줍니다. 실제 화면 · 시연 데이터

결과 화면에 들어가는 것

칸	내용
한 줄 요약	경기가 어떻게 흘러갔는지 1~2문장
팀별 범인	팀마다 1명. 평점, 이유 2~3문장, 근거 3~4개(대화 · 킬 · 경제 · 스탯)
10명 평가	평점(10점 만점, 0.5 단위), 태그, 한 줄 평. 음성 유무 표시
분석 정보	분석 시각, 모델, 쓴 문장 · 이벤트 수, 실제 비용
주의 문구	“재미로 보는 AI 분석입니다. 음성 인식 오류나 기록에 없는 상황 때문에 틀릴 수 있어요.”

분석하는 동안

버튼을 누르면 결과 자리에 진행 단계가 나옵니다: **기록 모으기 → AI가 읽는 중 → 근거 확인 → 완료**. 창을 닫아도 분석은 계속되고, 끝나면 사이트 실시간 알림으로 결과가 뜹니다. 같은 경기를 다른 사람이 동시에 눌러도 한 번만 돕니다.

결과를 보는 사람 · 다시 분석

- 결과는 사이트에 로그인한 서버 멤버 모두가 봅니다 (기존 기록과 같은 권한).
- 한 경기는 한 번만 분석합니다. 다시 분석은 관리자만 할 수 있고, 이전 결과는 기록으로 남깁니다.
- 사이트 **경기** 화면에도 “🔍 범인 보기” 버튼을 달아 바로 옮겨 갑니다.

3 범인을 고르는 기준

> **"범인"의 뜻 (정함 · ①)**
> **팀마다 1명**을 고르고, 아래 셋을 **모두 합쳐 종합으로** 판단합니다. 한 가지만 보지 않습니다.
> - **가장 못한 사람**: 자기 역할 · 포지션 기준으로 플레이가 가장 나빴다
> - **가장 발목을 잡은 사람**: 팀의 계획 · 경제 · 콜을 가장 많이 망쳤다
> - **결과의 원인**: 진 팀이면 패배에 가장 큰 책임, 이긴 팀이면 이기는 데 가장 방해가 된 사람
>
> 그래서 이긴 경기에도 범인이 있습니다. 범인은 그 팀에서 평점이 가장 낮아야 합니다(서버가 확인).

보는 것	발로란트	리그 오브 레전드
플레이	첫 킬 · 첫 데스, 전투 점수(ACS), 라운드당 피해량, 클러치, 스파이크 설치 · 해체, 연속 데스	포지션별 기준: 라인 CS · 골드 차, 킬 관여율, 딜량 · 받은 피해, 데스 시점(오브젝트 직전 등)
팀 운영	경제(혼자 장비 안 사기 · 세이브 무시), 요원 역할 수행(연막 · 정찰 · 부활)	드래곤 · 바론 · 전령 · 포탑 관여, 시야 점수, 한타 합류
소통 (음성 · 채팅)	정확한 콜, 콜 무시("기다려" 뒤 혼자 진입 등), 남 탓 · 짜증, 포기 발언("ff"), 팀 분위기를 살린 말	

공정하게 보는 규칙 (AI 지시문에 넣음)

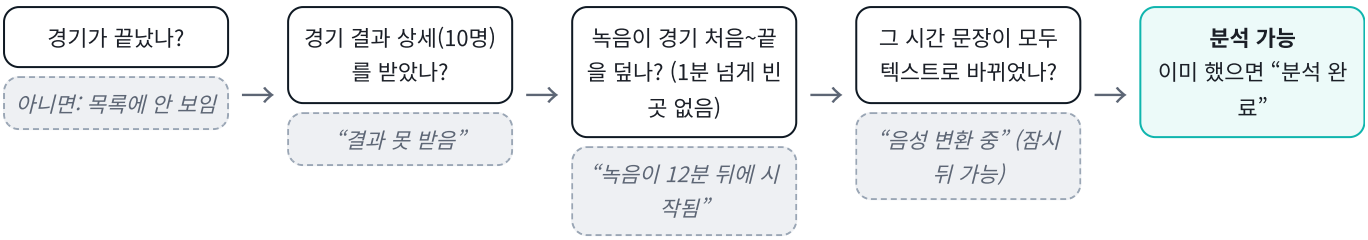
- **KDA만 보지 않습니다.** 역할을 따집니다. 세이지 · 쓰레쉬처럼 킬이 적은 역할은 그 역할로 평가합니다.
- **음성이 없는 사람은 소통 항목을 빼고** 플레이와 채팅으로만 평가합니다. 말이 없다고 깎지도, 말실수가 안 잡혔다고 올리지도 않습니다. 결과에 "게임 기록으로 판단"이라고 적습니다.
- **근거 없는 말 금지.** 근거마다 실제 기록 번호(문장 · 이벤트 · 라운드)를 달게 하고, 서버가 그 번호가 이 경기에 정말 있는지 확인한 뒤에만 보여 줍니다. 없는 근거가 나오면 그 근거를 빼고, 범인 근거가 하나도 안 남으면 다시 분석합니다.
- **음성 인식 오류를 감안**합니다. 뜻이 애매한 문장 하나로 범인을 정하지 않습니다.
- **재미용 말투**: 가볍게 놀리되 욕설 · 외모 · 인신공격은 쓰지 않습니다. 녹음 제외한 사람의 대화는 애초에 없습니다.

4 분석에 쓰는 기록 · 분석할 수 있는 경기

기록	원래 저장하던 것	범인찾기를 위해 더 저장 (구현 완료)
음성 대화	유저별 문장, 시각, 경기 · 라운드 연결 (텍스트는 계속 남음)	없음. 음성 파일은 7일 뒤 지워져도 분석은 텍스트로 함
발로란트 경기	10명 KDA · 전투 점수 · 랭크, 라운드 결과, 라운드별 경제(쓴 돈 · 무기), 킬(무기 · 헤드샷 · 위치)	선수별 준 피해 · 헤드샷 비율 · 첫킬/첫데스 · 설치/해체 · 스킬 사용 · 요원 역할, 라운드별 공격/수비 · 장비 값 · 선수별 킬/피해
롤 경기	10명 KDA · CS · 골드 · 아이템, 골드 차 그래프, 킬 · 오브젝트 이벤트	포지션(추정), 챔피언 피해 · 받은 피해, 시야 점수 · 와드 설치/제거/제어 와드, 오브젝트 · 포탑 피해, CC 시간, 10분 · 15분 CS · 골드
채팅	팀 · 전체 채팅 (API · OCR)	없음
디스코드 ↔ 게임 계정	Riot ID로 누가 누구인지 연결	연결 안 된 화자는 디스코드 이름으로 보내고 "게임 계정 모름"으로 표시

이미 저장된 예전 경기는 지금 있는 기록만으로 분석합니다(피해량 · 시야 같은 지표 없이). 결과에 "기본 기록으로 분석"이라고 표시합니다.

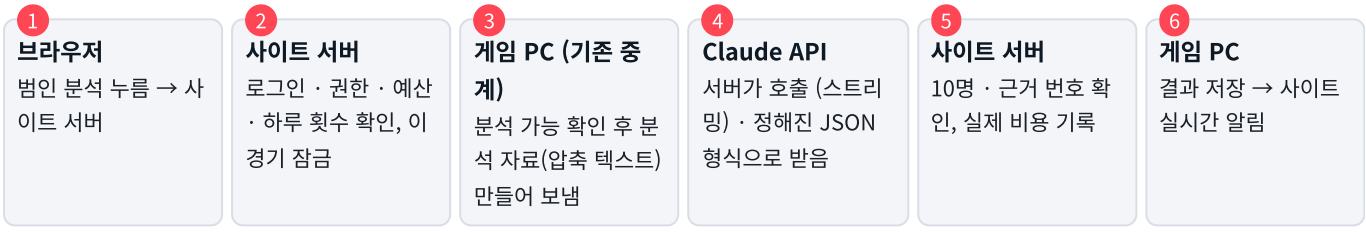
"기록이 전부 남아 있는" 경기 = 아래를 모두 만족



한계

- 겜녹이 봇은 한 디스코드 서버에서 음성채널 하나에만 들어갑니다. 그래서 **음성은 봇이 있는 채널 사람만** 있습니다. 보통 상대 팀은 게임 기록과 채팅으로만 평가합니다.
- 내전에서 두 팀이 다른 채널로 나뉘면 한 팀 음성만 남습니다. 두 팀 모두 녹음하려면 봇이 한 채널씩 따로 들어가야 해서, 이번 기능 범위 밖으로 둡니다(필요하면 따로 계획).
- 경기 기록은 게임 PC에만 있으므로 **게임 PC가 켜져 있어야** 분석하고 결과를 볼 수 있습니다(지금 사이트와 같음).

5 구조 · 보안 · 개인정보



- **API 키는 사이트 서버에만** 둡니다(환경 변수 `ANTHROPIC_API_KEY`, root만 읽기). 봇 토큰과 같은 원칙으로, exe에도 브라우저에도 가지 않습니다. 게임 PC가 직접 부르게 하면 키가 PC에 남아 꺼낼 수 있어서 쓰지 않습니다.
- **결과는 게임 PC 기록 DB에 저장**합니다(새 표 `culprit_analyses`). 기록은 게임 PC에만 둔다는 지금 원칙을 지킵니다. 서버에는 비용 장부만 남깁니다(경기 번호 · 누른 사람 · 토큰 수 · 비용 · 시각, 대화 내용 없음).
- **예산 장부**: 서버가 매번 실제 사용량(응답의 토큰 수)으로 비용을 계산해 쌓습니다. 월 상한을 넘으면 버튼을 잠급니다. Claude Console에도 지출 한도를 걸어 두 번 막습니다.
- **AI 설정**: 지시문 · 지식 묶음은 6장, Claude Opus 5.5, 생각 깊이 high, 말투 중간맛, 출력은 JSON 형식 고정(구조화 출력), 출력 상한 32k 토큰. AI가 응답을 거절하면 Anthropic 서버가 권장 모델로 한 번 자동 재시도합니다. 실패하면 비용이 든 만큼만 기록하고, 다시 누를 수 있게 잠금을 풉니다.

AI에 보내는 자료 모양 (예시)

```
# 경기 발로란트 경쟁전 · 어센트 · 13:9 (우리 팀 승리) · 22라운드 · 46분
# 참가자 (번호 이름 팀 요원(역할) K/D/A ACS 라운드당피해 첫킬/첫데스 헤드샷 설치/해체 스킬사용 음성)
P1 하늘 우리 제트(타격대) 19/12/7 ACS244 ADR148 첫킬1/첫데스2 HS15% 설치0/해체0 랭크:플래티넘 1 음성O
P5 Duck 우리 킬조이(감시자) 12/15/8 ACS194 ADR92 첫킬2/첫데스2 HS16% 설치0/해체1 랭크:골드 3 음성X
[e1] R1 0:08 킬 P2 → P6 (클래식)
[u1] R3 0:05 P1: 현우야 같이 가 기다려 둘 있어
[u2] R3 0:09 P2: 아 그냥 간다 내가 뚫을게
[r3] R3 우리 공격 라운드 끝 · 패배(전멸) 2:1 · 평균 장비 우리 3,714 / 상대 3,582 · 선수 장비/피해: P1 2.4k/185 P2 4.3k/0 …
[u3] R5 0:02 도윤(게임 계정 모름): 나 돈 없어 세이브할게
```

앱이 실제로 만드는 모양입니다(시연 경기). [u…] = 대화 문장, [e…] = 게임 이벤트, [c…] = 채팅, [r…] = 라운드 요약, [s:P…] = 그 사람 경기 전체 스탯 번호. AI는 근거를 이 번호로 답하고, 서버가 번호를 확인합니다. 숫자 덩어리(JSON)가 아니라 이런 줄글로 보내 토큰을 아낍니다.

개인정보

- 분석하면 그 경기의 **대화 텍스트**와 게임 기록이 Anthropic(Claude API)으로 갑니다. 음성 파일은 보내지 않습니다.
- Claude API로 보낸 내용은 기본적으로 AI 학습에 쓰이지 않고, Anthropic 정책에 정해진 기간만 보관됩니다.
- 녹음 제외(/녹음제외)한 사람의 대화는 원래 없으므로 보내지 않습니다. 서버 로그에는 대화 내용을 남기지 않습니다.
- 멤버 공지(⑥): 녹음 시작 안내(TTS) 문구를 "겜녹이가 이 음성채널을 녹음합니다. **녹음된 대화는 AI 경기 분석에 쓰일 수 있어요.** 녹음을 원하지 않으시면 녹음제외 명령을 입력해 주세요."로 바꾸고 기본 음성을 다시 합성했습니다. 관리자가 문구를 직접 고친 PC는 그 문구를 그대로 둡니다.

6 범인찾기 프롬프트 · 지식 묶음

> **"미리 학습"은 이렇게 합니다**
> Claude API는 우리가 모델을 따로 학습(파인튜닝)시킬 수 없습니다. 대신 **분석할 때마다 지시문과 지식 묶음을 경기 자료 앞에 같이 보냅니다**. 그래서 AI는 매번 게임 규칙 · 역할 기준 · 은어 · 음성 인식 오류를 "배운 상태"로 분석합니다. 이 부분은 경기마다 똑같아 프롬프트 캐시를 쓰면 1시간 안의 다음 분석부터 이 부분 값이 약 5%로 줄어듭니다(비용 계산에는 캐시 없이 넣었습니다).

파일	내용
`prompt/system.md` 지시문 v1.1	역할 · 할 일 · 범인 기준(종합 판단) · 판단 원칙 · 태그 · 말투(중간맛) · 자료 약어 설명 · 출력 규칙
`prompt/knowledge-valorant.md`	규칙 · 경제 · 역할별 기준 · 지표 기준 · 콜아웃 · 은어 · 음성 인식 오류 + 자동 갱신(요원 29명 역할 · 맵 13개 · 무기 가격)
`prompt/knowledge-lol.md`	기본 규칙 · 포지션별 기준 · 지표 기준 · 은어 · 음성 인식 오류 + 자동 갱신(챔피언 173명 분류)
`prompt/output-schema.json`	결과 JSON 형식: 요약 · 확신도 · 팀별 범인 · 10명 평가 · 근거 번호

파일은 `site/server/prompt/`에 있고(사이트 서버가 그대로 읽음) 전문은 부록에 있습니다. 토큰 세기 API(claude-opus-5-5)로 잰 고정 부분(지시문 + 지식 묶음 + 출력 형식)은 **발로란트 6,649 · 롤 8,204토큰**입니다. v1.1에서 글자 수로 잡은 3,900 · 5,000보다 많아서 7장 비용을 다시 계산했습니다(경기당 약 +$0.017).

지식 묶음 = 손으로 쓴 부분 + 패치마다 자동 갱신하는 부분

손으로 쓴 부분 (사람이 고침)

- 경기 규칙 · 경제(발로) / 포지션 · 오브젝트(롤)
- 역할 · 포지션별 평가 기준: 무엇을 잘하면 잘한 것인지
- 지표 기준값: 전투 점수 · 라운드당 피해량 · 분당 CS · 시야 점수
- 판단 함정: 에코 킬, 막판 정리 킬, 팀 세이브, 탱커 · 서폿 KDA
- 콜아웃 · 은어 뜻: 원탭, 로테, 리테이크, 갱, 다이브, 던졌다, 버스 등
- 음성 인식이 자주 틀리는 말: 앱 교정 사전 28개와 같은 목록

자동 갱신 부분 (패치마다 새로 만듦)

- 발로란트: 요원 이름과 역할(타격대 · 척후대 · 전략가 · 감시자), 맵, 무기 가격 ← valorant-api.com 한국어
- 롤: 챔피언 이름과 분류 ← Riot Data Dragon 한국어 (지금 16.20.1)
- 사이트 서버가 공식 데이터에서 하루에 한 번 새로 만들고, 못 받으면 파일의 스냅샷을 씀
- AI가 원래 모를 수 있는 최신 요원 · 무기(믹스 · 비토 · 웨이레이 · 테호 · 워든 등)를 채워 줌

지시문의 핵심 규칙 (발췌)

```
판단 원칙
- 숫자보다 결과에 준 영향을 본다. 승부처의 실수는 무겁게, 승부가 갈린 뒤의 스탯과 에코 라운드 킬은 가볍게.
- 역할로 평가한다. 킬이 적은 역할이 킬이 적다고 범인이 되지 않는다.
- 음성X 인 사람은 소통 항목 없이 평가한다. 음성O 인 사람을 범인으로 고를 때는 플레이 근거가 최소 1개 있어야 한다.
- "콜 무시"는 콜이 있었고, 다르게 움직인 것이 이벤트로 확인될 때만 쓴다.
- 음성 인식 오류를 감안한다. 뜻이 분명하지 않은 문장은 근거로 쓰지 않는다.
- 자료에 없는 것을 지어내지 않는다. 근거는 자료의 번호로만, 인용은 원문 그대로.
출력  10명 모두 · 팀마다 "범인" 태그 정확히 1명 · 범인은 그 팀 최저점 · 판단이 약하면 confidence 를 낮춤
```

프롬프트가 제대로 도는지 확인하는 방법

#	시험 경기 (테스트 자료로 만듦)	기대하는 결과
1	킬이 적은 세이지 · 쓰레쉬가 부활 · 시야로 기여	범인이 되지 않고, 한 줄 평에 기여가 나옴
2	상대 에코 라운드에서만 킬을 몰아 딴 사람	전투 점수만큼 높게 평가하지 않음
3	“기다려” 콜 뒤 혼자 진입해 첫 데스 3번 (음성O)	범인. 근거에 대화 + 킬 이벤트가 같이 나옴
4	음성X 팀원이 스탯은 평균	소통으로 감점 · 가점하지 않음
5	음성 인식이 깨진 문장이 섞인 경기	깨진 문장을 근거로 쓰지 않음
6	크게 이긴 경기	이긴 팀에서도 범인 1명, 말투가 과하지 않음

지시문이나 지식 묶음을 고칠 때마다 6경기를 다시 돌립니다(한 번에 약 $2.5). 결과 화면 아래에 👍 / 👎(이유 한 줄)를 달아, 친구들이 납득하지 못한 판정을 모아 지식 묶음을 고치는 데 씁니다.

7 비용 예상: $100로 몇 경기?

가격 (Anthropic API, 100만 토큰당)

모델	입력	출력(생각 포함)	특징
Claude Opus 5.5 (추천)	**$4**	**$20**	**가장 꼼꼼함. 대화 · 기록을 엮어 판단하는 데 유리**
Claude Sonnet 5.5	$2	$10	절반 가격. 품질은 실측으로 비교 필요
Claude Haiku 4.5	$1	$5	가장 쌈. 긴 대화에서 근거를 놓칠 수 있음

한국어는 영어보다 토큰을 많이 씁니다. 계산에서는 한글 1글자 ≈ 1토큰으로 넉넉하게 잡았습니다. 배치(최대 24시간 안에 결과)로 보내면 모든 토큰이 반값이지만, 버튼을 누르고 기다리는 기능이라 기본은 즉시 분석입니다.

경기 길이는 무작위: 이렇게 계산했습니다

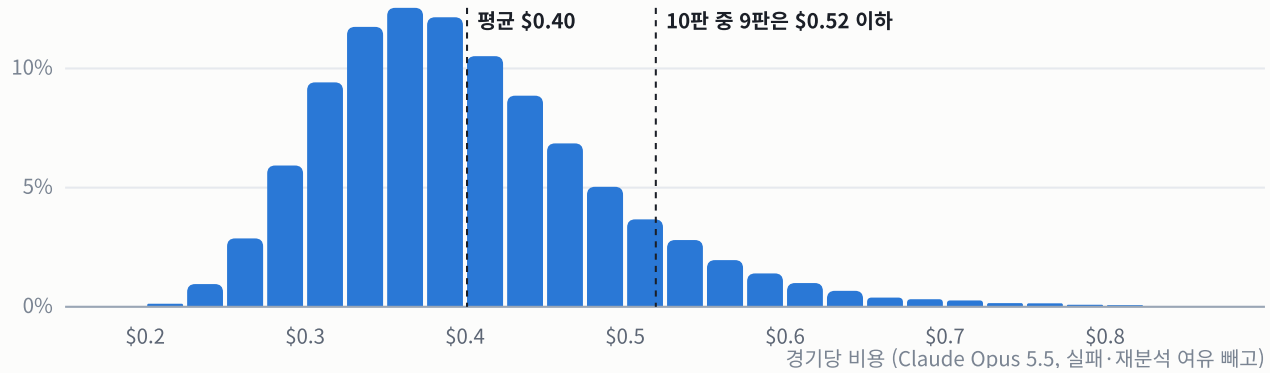
바뀌는 값	계산 방법	결과 (10판 중 8판 범위)
발로란트 길이	경기마다 두 팀 실력 차를 뽑고 라운드를 하나씩 진행. 13승제, 12:12면 2점 차가 날 때까지 연장. 라운드 = 구매 30초 + 교전 15~145초 + 끝 7초	17~26라운드, 29~44분 (중간 36분)
롤 길이	12%는 15~22분 조기 항복, 나머지는 평균 31분 ±7분 (18~60분)	19~40분 (중간 30분)
말하는 양	음성채널 전체가 1분에 하는 문장 수. 경기마다 무작위(중간 18문장, 조용한 방 9 · 말 많은 방 32)	경기당 328~999문장

AI 생각 길이	자료가 길수록 길게, 경기마다 무작위	출력 9.9k~19.6k 토큰
게임 비율	발로란트 · 롤 반반	입력 20.3k~38.9k 토큰

이 값으로 2만 경기의 비용을 계산하고, $100을 다 쓸 때까지 분석하는 시행을 1,500번 돌렸습니다. 실패 · 다시 분석 여유로 5%를 더 잡았습니다.
계산 코드: `docs/report/culprit/cost_sim.py`

경기당 비용 분포 (Opus 5.5, 2만 경기)

길고 말 많은 경기일수록 오른쪽. $0.68를 넘는 경기는 100판 중 1판입니다.

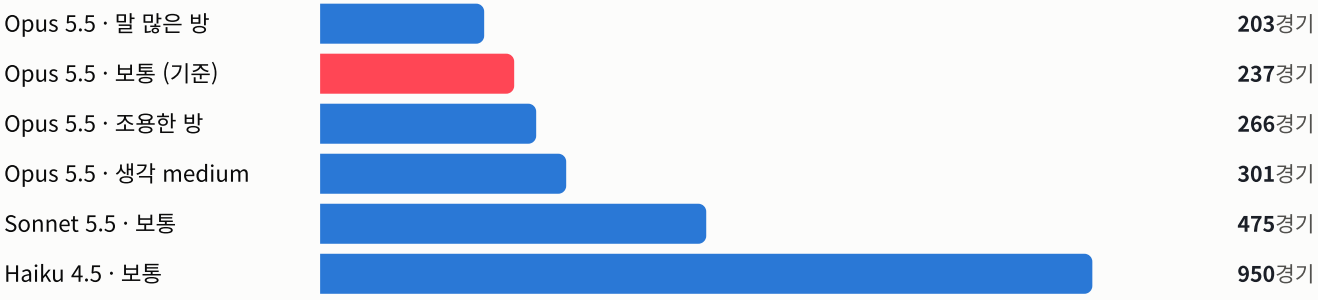


경기 하나에 드는 토큰 (Opus 5.5, 평균)

부분	토큰	비용	설명
지시문 · 지식 묶음 · 출력 형식	약 7,426	$0.030	토큰 세기 API 실측. 발로 6,649 · 롤 8,204
경기 기록 (10명 · 라운드 · 이벤트)	약 6,100	$0.025	가정: 발로 약 8,100 · 롤 약 4,200. 시연 경기 실측: 발로 22라운드 약 9,400 · 롤 28분 약 1,900
음성 대화	약 15,234	$0.061	평균 635문장 × 24토큰. **가장 크게 흔들리는 값**
채팅	약 250	$0.001	
AI 답 (10명 평가 · 범인 · 근거)	약 3,300	$0.066	JSON
AI 생각	약 10,969	$0.219	출력 값으로 계산. **비용의 절반 이상**
합계	**입력 29,053 · 출력 14,249**	**$0.401**	**발로 $0.42 · 롤 $0.39**

$100로 분석할 수 있는 경기 수

발로 · 롤 반반, 실패 · 다시 분석 여유 5% 포함. 빨간 막대가 추천 설정입니다.



설정	경기당 평균	$100당 경기 수 (5~95%)	배치(반값)라면
Opus 5.5 · 생각 high (추천)	**$0.40**	**237 (232~242)**	**약 474**
Opus 5.5 · 생각 medium	$0.32	301 (295~307)	약 602
Sonnet 5.5	$0.20	475	약 950
Haiku 4.5	$0.10	950	약 1,900

Opus 5.5 · 방 분위기별	발로란트만	롤만	반반
조용한 방 (1분 9문장)	257경기	274경기	266경기
보통 (1분 18문장)	**227경기**	**248경기**	**237경기**
말 많은 방 (1분 32문장)	195경기	215경기	203경기

결론

Opus 5.5로 **$100이면 약 237경기**(방 분위기에 따라 약 203~266경기)입니다. 경기 길이가 무작위여도 판 수가 많아 평균에 가까워지므로, 가정한 조건 안에서는 1,500번 시행 중 10번에 9번꼴로 232~242경기(5~95% 범위)가 나왔습니다. 이 범위는 계산에 넣은 가정 안의 결과이고, 실제 비용을 보장하는 범위는 아닙니다. 더 크게 흔들리는 것은 **방에서 말하는 양**과 **AI가 생각하는 길이**입니다.

크레딧은 **11월 15일에 만료**됩니다. 237경기를 다 쓰려면 하루 8판씩 분석해야 하므로, 크레딧을 아끼려고 더 싼 모델을 쓸 이유는 적습니다. 그래서 Opus 5.5를 추천합니다.

고정 프롬프트는 실측값이고, 경기 기록 · 대화 양 · AI 생각 길이는 아직 가정값입니다. 실제 경기 3~5판을 분석하면 사이트 서버 비용 장부에 실제 토큰 · 비용이 남으므로, 그 값으로 이 표를 다시 고칩니다.

예산이 새지 않게 막는 장치

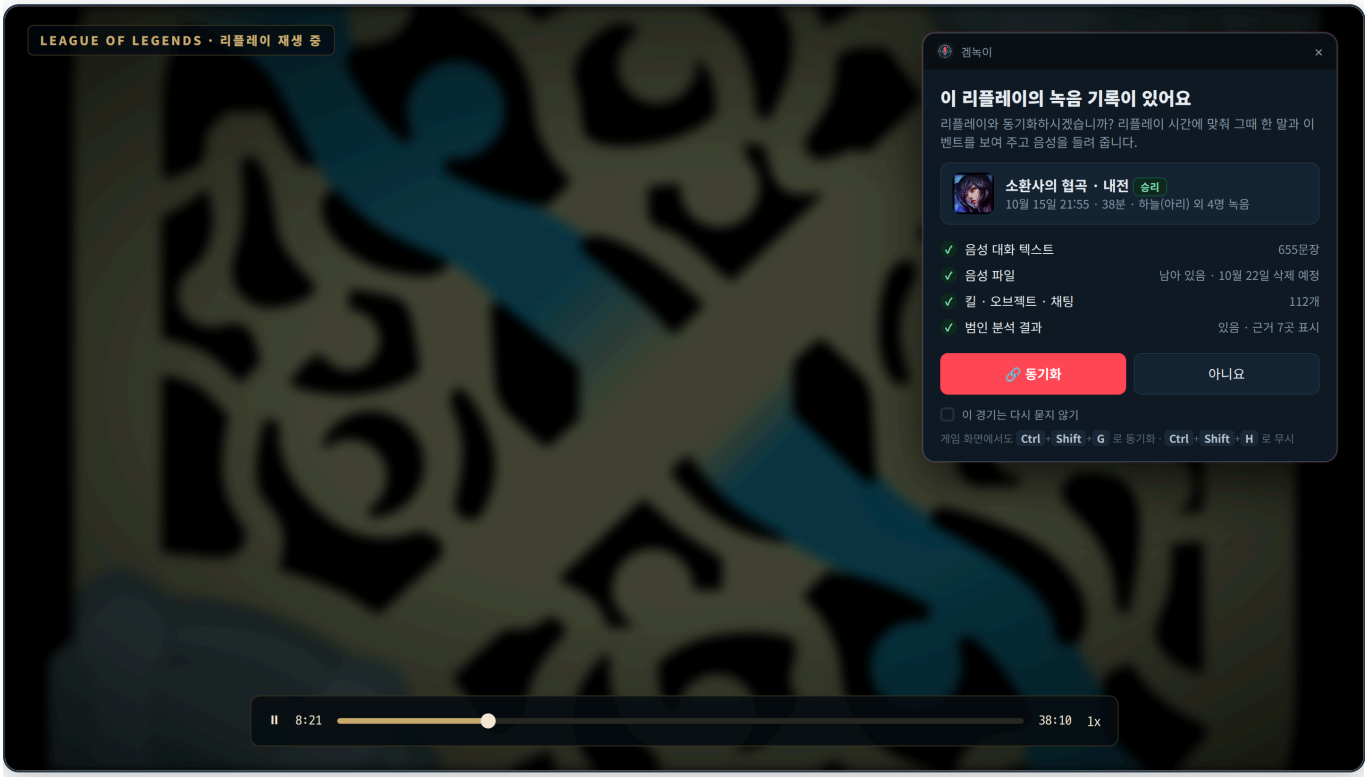
장치	내용
월 상한	$100 (설정 가능). 넘으면 버튼 잠금, 남은 예산을 화면에 표시
경기당 상한	출력 32k 토큰 상한 + 입력이 너무 길면(대화 2,500문장 넘음) 잡담 구간을 줄여서 보냄 → 경기 하나가 약 $1을 넘지 않음
하루 횟수	사이트 전체 하루 20회 (설정 가능)
중복 방지	한 경기 한 번. 동시에 눌러도 한 번만 실행. 다시 분석은 관리자만
두 번째 방어선	Claude Console 조직 지출 한도를 $100로 설정

8 리플레이 동기화



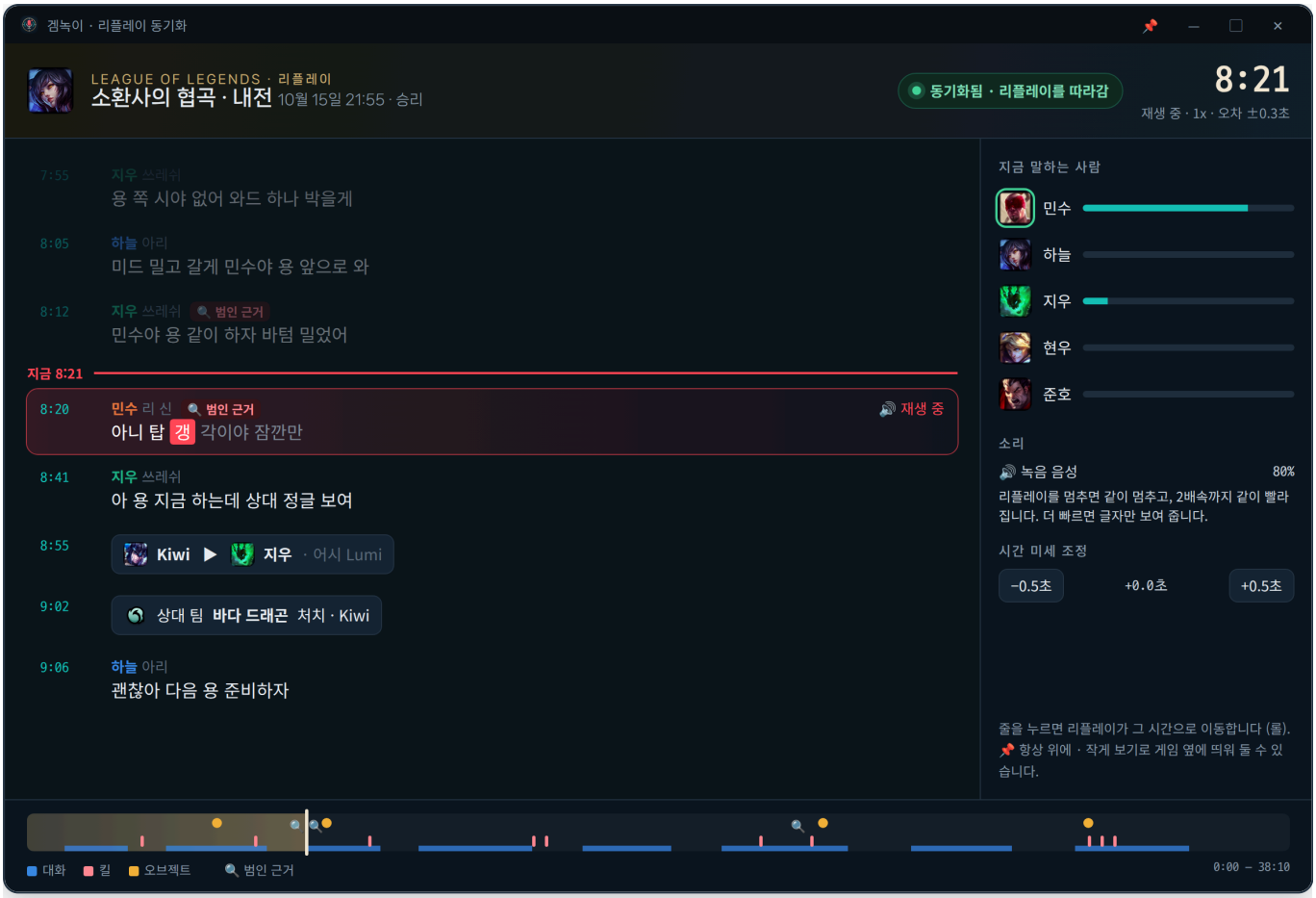
정한 원칙 (⑦ · ⑧): API로 될 때만 넣습니다

게임이 **공식 · 비공식 API로 리플레이 재생 시간을 알려 줄 때만** 그 게임의 동기화를 만듭니다. 화면 글자 인식(OCR)으로 시간을 읽거나, 앱이 게임 설정 파일(game.cfg)을 고치는 방법은 쓰지 않습니다. 그래서 먼저 Windows PC에서 어떤 API가 리플레이 중에 무엇을 알려 주는지 확인(R0)하고, 되는 게임만 R1~R3을 진행합니다. 둘 다 안 되면 리플레이 동기화는 넣지 않습니다.



화면 예시 4 · 동기화 묻기 (exe 팝업) 리플레이를 틀면 exe가 그 경기의 기록을 찾아 오른쪽 위에 묻습니다. 남은 기록(대화 텍스트 · 음성 파일 · 이벤트 · 범인 분석)을 보여 주고, 음성 파일이 7일이 지나 지워졌으면 “글자만 동기화”로 바뀝니다. 뒤 배경은 자리 표시용입니다.

시안 · API 확인 뒤 만듦



화면 예시 5 · 동기화 창 (exe) 리플레이 시간(오른쪽 위)에 맞춰 지금 말하는 문장을 단어 단위로 짚고 음성을 들려 줍니다. 지나간 줄은 흐리게, 다음 줄은 미리 보입니다. 아래 막대는 경기 전체의 대화 · 킬 · 오브젝트 · 범인 근거 위치입니다. 리플레이를 옮기는 API까지 있으면 줄을 눌러 리플레이를 그 시간으로 옮깁니다. 시안 · 이름과 기록은 예시

R0에서 확인할 것 (Windows PC, 리플레이 1판씩)

	리그 오브 레전드	발로란트
리플레이	클라이언트 “경기 기록”에서 내려받아 재생 (.rofl 파일, 같은 패치 안에서만)	패치 12.10부터 **경쟁전 · 일반전 · 신속플레이 · 프리미어 · 사용자 지정 게임**, 친구 경기도 경력 탭에서 볼 수 있음. **내전(사용자 지정)은 방장이 경기마다 시작 전에 녹화를 켜야** 리플레이가 남음
확인할 API	① 롤 클라이언트 로컬 API(LCU, 비공식): 리플레이 보는 중인지, 어느 경기(경기 번호)인지 ② 게임 로컬 API(이 PC 안 `127.0.0.1:2999`): 설정을 바꾸지 않은 상태에서 리플레이 중에 게임 시간이 나오는지 ③ 공식 Replay API(`/replay/playback`): 재생 시간 · 멈춤 · 배속 · 옮기기. **game.cfg 에 한 줄을 넣어야 켜지므로 앱이 켜지 않음**. 이미 켜져 있는 PC에서만 씀	① Riot 클라이언트 로컬 API(비공식, 지금 경기 감지에 쓰는 상태 정보): 리플레이 보는 중인지, 어느 경기인지 ② 같은 API에서 리플레이 재생 시간이나 라운드가 나오는지
되면	시간을 0.25초마다 읽어 동기화(목표 오차 ±0.5초). ③이 켜져 있으면 줄을 눌러 리플레이도 옮김	시간을 읽어 동기화. 정확도는 API가 주는 값의 간격에 따라 R0에서 정함
안 되면	**그 게임은 리플레이 동기화를 넣지 않습니다.** 화면 글자 인식(OCR) · 수동 맞춤 · 설정 파일 고치기로 대신하지 않습니다.	

확인 방법	새 exe에 “리플레이 API 확인” 기능을 넣어, 리플레이를 1판씩 틀고 버튼을 누르면 위 API들이 리플레이 중에 무엇을 주는지 기록 파일로 남깁니다. 그 기록으로 게임별로 만들지 정합니다.

시간을 맞추는 원리

- 겜녹이는 경기마다 **게임 시간**(킬 · 라운드 이벤트)과 **실제 시각**(음성 문장)을 같이 저장하고, 둘의 차이를 이미 맞춰 둡니다(지금 사이트의 “R7 · 0:42” 표시가 이 값).
- 롤: 리플레이 재생 시간 = 게임 시간 → 실제 시각 = 경기 시작 시각 + 재생 시간. 그 시각의 음성 문장과 이벤트를 띄웁니다.
- 발로란트: API가 주는 리플레이 시간(또는 라운드 + 라운드 안 시간) → Riot 경기 결과에 있는 그 라운드 시작 게임 시간 → 실제 시각.
- 조금 어긋나면 동기화 창의 **±0.5초** 버튼으로 맞추고, 그 경기에 저장합니다.

팝업 · 음성 재생 규칙

- **묻는 조건**: 그 경기 기록(대화 텍스트나 이벤트)이 있을 때만 묻습니다. 어느 경기인지 하나로 정하지 못하면 후보 경기(맵 · 점수 · 날짜 · 길이)를 보여 주고 고르게 합니다. 기록이 없거나, 녹음 중이거나, “이 경기는 다시 묻지 않기”를 고른 경기는 묻지 않습니다.
- **전체 화면 게임**: 게임이 “전체 화면” 모드면 다른 창이 위에 뜨지 못합니다. 그때는 알림 소리 + Windows 알림을 보내고, 게임 안에서 **Ctrl+Shift+G**(동기화) · **Ctrl+Shift+H**(무시)로 답할 수 있게 합니다. “테두리 없는 창 모드”를 권합니다.
- **동기화 창**: exe의 새 창입니다. 두 번째 모니터에 두거나, 📌 항상 위 · 작게 보기로 게임 옆에 띄웁니다.
- **음성**: 리플레이를 멈추면 같이 멈추고, 되감으면 그 지점부터 다시 들려 줍니다. 2배속까지는 같이 빨라지고, 더 빠르면 글자만 보여 줍니다. 여러 명이 동시에 말한 구간은 겹쳐서 재생합니다. 음성 파일이 지워진 경기는 글자만 보여 줍니다.
- **범인 분석과 연결**: 분석 결과가 있으면 범인 근거 순간에 🔍 표시를 붙입니다.
- **보안 · 비용**: 모두 이 PC 안(127.0.0.1)에서만 일어나고 사이트 서버 · 인터넷 · AI를 쓰지 않습니다. 비용 0. 게임 메모리를 읽거나 게임 파일 · 설정 파일을 고치지 않고, 화면을 읽지도 않습니다. Riot 로컬 API의 자체 서명 인증서는 지금처럼 이 PC 안 주소에만 예외를 둡니다.

9 만드는 순서 · 진행 상황

단계	할 일	확인 방법	상태
C1 기록 보강	경기 상세에서 추가 지표 저장(4장). “분석 가능” 판정(경기 결과 · 녹음 공백 1분 · 음성 변환). 근거 번호가 달린 분석 자료 만들기, 결과 저장. 녹음 안내 문구에 AI 분석 안내	시연 22라운드 · 롤 경기로 판정 · 자료 모양 자동 테스트. 녹음 늦게 시작 · 변환 대기 · 녹음 제외 경기가 막히는지	**완료**
C2 서버 분석	Claude 호출(구조화 출력), 근거 번호 확인, 예산 장부 · 하루 횟수 · 중복 방지, 결과를 게임 PC에 저장	가짜 AI 응답 자동 테스트(없는 근거 · 빠진 사람 · 예산 초과 · 동시 클릭 · 거절). 실제 Claude 1회 호출 성공	**완료**
C3 화면	범인찾기 탭, 진행 단계, 결과 화면(PC · 휴대폰), 근거 → 듣기 · 타임라인, 경기 화면 버튼	실제 앱 코어 + 사이트 서버 + 브라우저로 화면 캡처(2장), 보안 정책(CSP) 위반 · 오류 없음	**완료**
C5 배포	사이트 서버 배포(API 키 넣기), 앱 새 버전(v1.3.0) exe 빌드 · 릴리스	배포 전 백업, 배포 뒤 상태 확인 · 실제 분석 1회	**승인 대기**

C4 실측 · 조정	새 exe로 실제 경기 3~5판 분석, 결과 품질과 실제 비용 확인, 지시문 · 지식 묶음 · 생각 깊이 조정, 7장 표 갱신	같이 결과 검토 (범인 판정이 납득되는지)	**배포 뒤**
R0 리플레이 API 확인	새 exe의 "리플레이 API 확인" 기능으로 롤 · 발로란트 리플레이 중 API가 주는 값 기록(8장)	롤 리플레이 1판, 발로란트 리플레이 1판	**배포 뒤**
R1~R3 리플레이 동기화	R0에서 API로 시간을 읽을 수 있는 게임만: 감지 · 팝업, 동기화 창(시간 따라가기 · 음성 · 범인 근거 표시)	가짜 API로 자동 테스트, 실제 리플레이로 멈춤 · 배속 · 되감기 따라가는지	**R0 결과에 따라**

예전 exe로도 사이트는 열리지만 범인찾기는 "게임 PC 앱을 업데이트해 주세요"로 막습니다. 코드: 앱 `app/src/core/culprit.ts`, 서버 `site/server/src/culprit.ts`, 화면 `site/web/src/pages/CulpritPage.tsx`.

10 정한 것 · 남은 일

질문	답 (2026-10-15)	반영
① "범인"의 뜻	가장 못했다 + 가장 발목을 잡았다 + 패배의 원인 등 모두 종합	지시문 v1.1에 세 기준을 종합하도록 넣음. 팀마다 1명, 팀 최저점(서버 확인)
② 분석 버튼	(가) 등록된 앱 유저 이상	서버가 앱 유저 명단으로 확인. 다시 분석은 관리자 이상, 결과 보기는 모두
③ 모델	Opus 5.5	Claude Opus 5.5 · 생각 깊이 high
④ API 키	받음	동작 확인. 서버 비밀 저장소에만 보관, 배포 때 사이트 서버 환경 변수로만 넣음
⑤ 말투	(나) 중간맛	기본 말투 medium
⑥ 멤버 공지	TTS 문구에 추가	녹음 안내 문구 · 기본 음성 교체 (5장)
⑦ 발로란트 리플레이	비공식 API에 없으면 추가하지 않기	OCR 안 함. R0에서 API 확인 뒤 결정 (8장)
⑧ 롤 리플레이	비공식 API에 없으면 추가하지 않기	앱이 game.cfg 를 고치지 않음. R0에서 API 확인 뒤 결정 (8장)

A 배포 승인

사이트 서버 배포(API 키 넣기)와 새 exe(v1.3.0) 릴리스를 진행해도 될지. → 승인하시면 배포하고, 새 exe로 실제 경기 3~5판을 분석해 같이 결과를 봅니다(C4).

B API 키 관리

키가 디스코드 채팅에 그대로 올라왔습니다. → 그 메시지를 지워 주시고, 오래 쓰실 거면 Console에서 키를 새로 만들어 주세요. Console 월 지출 한도 $100도 걸어 두시면 사이트 서버 한도와 함께 두 번 막힙니다.

C 리플레이 API 확인 (R0)

새 exe에 "리플레이 API 확인" 기능을 같이 넣을지. → 넣으면 리플레이를 1판씩 틀고 버튼만 눌러 주시면 됩니다. 롤 Replay API를 이미 켜 둔 PC가 아니면, 롤은 클라이언트 API 결과로만 정합니다.

부록 프롬프트 · 지식 묶음 · 출력 형식 원문

사이트 서버가 실제로 쓰는 원문(site/server/prompt/, 지시문 v1.1). 실제 경기 분석(C4)과 시험 경기(6장)를 거치며 고칩니다. 지시문의 tone 자리는 말투 설정(중간맛)으로 바뀌고, 자동 갱신 칸은 사이트 서버가 하루에 한 번 공식 데이터로 다시 만듭니다.

A. 지시문 prompt/system.md

```
너는 겜녹이의 경기 분석가다. 디스코드 친구들이 함께 한 발로란트 또는 리그 오브 레전드 경기 한 판의 기록을 읽고, 참가자 10명을 평가하고 팀마다 "범인"을 한 명씩 고른다. 결과는 친구들이 재미로 보는 사이트에 그대로 뜬다. 정확하고 공정해야 웃고 넘길 수 있다.

<input_format>
user 메시지의 <match> 안에 경기 자료가 온다.
- 경기: 게임, 모드, 맵, 최종 점수, 길이. "우리 팀" = 음성을 녹음한 쪽 팀, "상대 팀" = 다른 팀.
- 참가자: P1~P10 번호, 이름, 팀, 요원·챔피언과 역할(포지션), 스탯, 음성 유무(음성O / 음성X).
- 시간 순서의 줄. 줄 앞 번호가 근거 번호다.
  - [e번호] 게임 이벤트 (킬, 스파이크, 오브젝트, 라운드 결과 등)
  - [u번호] 음성 문장. 디스코드 음성채널에서 자동 음성 인식으로 받아 적은 것이라 글자가 틀릴 수 있다.
  - [c번호] 인게임 채팅 (팀 / 전체)
  - [r번호] 라운드 요약 (발로란트): 우리 팀 공격/수비, 승패와 끝난 이유, 점수, 팀 평균 장비 값, "선수 장비/피해" = 그 라운드 장비 값(k=1,000 크레드)/준 피해
  - [s:P번호] 그 사람의 경기 전체 스탯 (참가자 줄)
- 줄의 시간: 발로란트 "R7 0:42" = 7라운드 교전 시작 뒤 42초("R7 구매" = 구매 시간), 롤 "12:18" = 게임 시간.
- 참가자 줄 약어: ACS = 라운드당 전투 점수, ADR = 라운드당 준 피해, HS = 헤드샷 비율, 롤 "10분 78·3450" = 10분 CS·골드. "?" 는 기록 없음.
- "(게임 계정 모름)" 이 붙은 화자는 디스코드 이름만 알고 어느 참가자인지 모른다. 참가자 평가에 그 사람 말을 붙이지 않는다.
- "기본 기록만 있다" 고 적힌 경기는 피해량·첫킬·시야 같은 지표가 없다. 없는 지표로 판단하지 않는다.
- 음성은 봇이 있던 음성채널 사람만 있다. 상대 팀은 보통 음성이 없다.
</input_format>

<task>
1. 경기 흐름을 먼저 파악한다. 어느 라운드·시간대에서 승부가 갈렸는지 찾는다.
2. 10명 모두 평가한다: 평점(1.0~10.0, 0.5 단위), 태그 하나, 한 줄 평(40자 안팎).
3. 팀마다 범인을 한 명씩 고른다. 범인은 아래를 모두 합쳐 종합으로 판단한다. 한 가지만 보지 않는다.
   - 가장 못한 사람: 자기 역할·포지션 기준으로 플레이가 가장 나빴다.
   - 가장 발목을 잡은 사람: 팀의 계획·경제·콜을 가장 많이 망쳤다.
   - 결과의 원인: 진 팀이면 패배에 가장 큰 책임, 이긴 팀이면 이기는 데 가장 방해가 된 사람.
   - 범인은 그 팀에서 평점이 가장 낮아야 한다. 동점이면 근거가 더 분명한 사람.
4. 범인마다 이유(2~3문장)와 근거 3~4개, 나머지 사람은 근거 0~2개를 단다.
5. 경기 전체 한 줄 요약(1~2문장)을 쓴다.
</task>

<judging_rules>
- 숫자보다 결과에 준 영향을 본다. KDA·전투 점수만으로 정하지 않는다.
  - 승부처(접전 라운드, 매치 포인트, 연장, 오브젝트 직전, 결정적 한타)의 실수는 무겁게 본다.
  - 승부가 이미 갈린 뒤의 스탯(진 라운드 막판 킬, 끝난 경기의 데스)과 상대 에코 라운드에서 딴 킬은 가볍게 본다.
- 역할로 평가한다. <knowledge> 의 역할·포지션 기준을 따른다. 킬이 적은 역할이 킬이 적다고 범인이 되지 않는다.
- 음성X 인 사람은 소통 항목 없이 플레이와 채팅으로만 평가한다. 말이 없다고 깎지도, 말실수가 안 잡혔다고 올리지도 않는다.
- 음성O 인 사람은 말실수가 기록에 남아 불리하다. 그래서 음성O 인 사람을 범인으로 고를 때는 소통 근거 말고 플레이 근거([e], [r], [s])가 최소 1개 있어야 한다.
- "콜 무시"는 (1) 같은 음성채널에서 콜이 있었고 (2) 그 사람이 다르게 움직인 것이 이벤트로 확인될 때만 쓴다.
- 음성 인식 오류를 감안한다. 문장 하나로 결론 내지 않는다. 이상한 단어는 <knowledge> 의 오인식 목록을 참고하고, 뜻이 분명하지 않은 문장은 근거로 쓰지 않는다.
- 게임과 상관없는 잡담, 친구끼리 장난으로 하는 거친 말은 평가에 쓰지 않는다. 팀원을 계속 탓하거나 포기를 부추기는 말(ff, 서렌 치자, 던지자)은 소통 감점이다.
- 좋은 소통도 본다: 적 위치·체력 정보, 타이밍 콜, 침착하게 계획을 세우는 말은 가점이다.
- 자료에 없는 것을 지어내지 않는다. 화면, 정확한 위치 이동, 대부분의 스킬 사용은 자료에 없다. 모르면 판단에서 뺀다.
- 근거는 자료의 번호로만 단다. 대화 인용은 원문 그대로 쓰고(고치지 않는다), 30자를 넘으면 줄인다.
- 자료가 부족하거나 기록이 한쪽으로 쏠려 판단이 약하면 confidence 를 낮추고 이유를 적는다.
</judging_rules>

<tags>
캐리 · 숨은 MVP · 클러치 · 무난 · 아쉬움 · 범인 중 하나. "범인" 태그는 팀마다 정확히 한 명, 그 팀의 범인에게만 붙인다.
</tags>

<tone>
말투: {{tone}}
- mild(순한맛): 담백한 평가. 놀리지 않는다.
```

```
- medium(중간맛, 기본): 친구끼리 가볍게 놀리는 정도. 예) "연장에서 혼자 B 메인 관광 세 번."
- spicy(매운맛): 더 세게 놀린다. 그래도 아래 금지는 지킨다.
공통: 짧은 평서문("~했다"). 존댓말·이모지·감탄사 남발 없음. 이름은 자료에 적힌 그대로.
금지: 욕설, 외모·성별·나이·지역·가족 이야기, 게임 밖 사생활(잡담 내용 폭로), 진짜 상처가 될 인신공격.
</tone>

<output>
정해진 JSON 형식으로만 답한다. players 는 P1~P10 순서로 10명 모두 넣는다. 평점과 범인이 서로 맞는지(범인이 그 팀 최저점인지) 마지막에 확인한다.
</output>
```

B. 발로란트 지식 묶음 `prompt/knowledge-valorant.md`

```
<knowledge game="valorant">
## 경기 규칙
- 13라운드를 먼저 이기면 승리. 12:12 이면 연장(경쟁전은 2라운드 차가 날 때까지). 12라운드 뒤 공수 교대.
- 1라운드와 13라운드는 권총 라운드. 라운드는 구매 시간 30초 + 교전 1분 40초. 스파이크 설치 뒤 45초면 폭발, 해체는 7초(절반 지점 저장).
- 공격: 스파이크 설치나 전멸로 승리. 수비: 전멸, 해체, 시간 초과로 승리.

## 경제 (패치에 따라 조금 바뀔 수 있음)
- 시작 800, 라운드 승리 3,000, 패배 1,900 → 연패마다 +500 (최대 2,900), 킬 200, 설치 300(공격팀 전원). 최대 9,000.
- 풀바이 = 라이플(밴달·팬텀 2,900) 또는 오퍼레이터(4,700) + 실드 + 스킬, 대략 3,900 이상. 에코(세이브) = 거의 안 삼. 포스바이 = 돈이 모자라도 삼. 하프바이 = 중간.
- 팀이 같이 세이브하는 건 정상 운영이다. 혼자만 사거나 혼자만 안 사서 팀 경제가 어긋나면 감점.
- 상대가 에코인 라운드를 지는 것은 큰 손실. 에코 라운드에서 딴 킬은 가치가 낮다.

## 역할별 기준
- 타격대(제트·레이즈·레이나 등): 진입해서 첫 교전을 열고 공간을 만든다. 첫 데스가 많아도 팀이 바로 복수(트레이드)하면 정상. 트레이드도 안 되는 혼자 진입이 반복되면 감점.
- 척후대(소바·브리치·스카이 등): 정보와 진입 보조. 어시스트, 정찰 정보 콜이 중요.
- 전략가(오멘·브림스톤·아스트라 등): 연막으로 시야를 끊어 진입·리테이크를 돕는다. 오래 살아남는 것이 중요. 킬이 적어도 정상.
- 감시자(킬조이·사이퍼·세이지 등): 사이트 수비, 측면 차단, 리테이크. 전투 점수가 낮은 게 흔하다. 세이지 부활은 큰 기여.
- 역할이 겹치면(타격대 3명 등) 판단을 팀 구성 탓으로 돌리지 말고 각자 맡은 일을 했는지 본다.

## 지표
- 전투 점수(ACS): 라운드당 평균. 보통 200 안팎, 250 넘으면 높음, 150 아래면 낮음.
- 라운드당 피해량: 보통 130~150. 첫 킬 · 첫 데스: 라운드 첫 교전 결과로 라운드 승률에 크게 영향.
- 클러치: 혼자 남아 여러 명을 상대로 라운드를 이김. 멀티킬 · 에이스(5킬).
- 막판 정리 킬(이미 진 라운드에서 도망가는 적을 잡음)은 숫자를 부풀린다.

## 콜아웃 · 은어 (음성·채팅에 나오는 말)
- 장소: A/B/C 사이트, 메인, 롱, 숏(쇼트), 미드, 헤븐, 마켓, 로비, 링크, 백사이트, 개구멍, 문, 계단, 창문, 엘보, 지붕, 하수구, 텔레포터, 로프, 파이프, 정원, 트리, 박스, 기둥.
- 전투: 원탭/투탭(머리 한두 발), 헤드, 퍼블·첫킬, 각(사격 위치), 피킹(엄폐물에서 나와 쏨), 프리파이어(미리 쏨), 트레이드(복수 킬), 리테이크(설치 뒤 사이트 탈환), 로테(다른 사이트로 이동), 스택(한 사이트에 몰림), 디폴트(흩어져 정보 수집), 러시·패스트(빠른 진입), 원웨이(한쪽만 보이는 연막), 몰리(화염), 플래시·섬광, 연막·스모크, 궁(궁극기), 오브.
- 상태: "하나 남았어"(적 1명), "피 30"(체력 30), "깎였어"(맞음), "세이브해"(무기 아끼고 살아라), "에코/이코", "풀바이", "포스".
- 태도: ㅈㅈ·ff(포기), "던졌네"(실수로 경기를 망침), "버스 탔다"(남이 이겨 줌).

## 음성 인식이 자주 틀리는 말
레이저→레이즈, 비 싸이트→B 사이트, 에이 싸이트→A 사이트, 로태→로테, 오퍼래이터→오퍼레이터, 반달→밴달, 펜텀→팬텀, 쉐리프·세리프→셰리프, 디퓨저→디퓨즈, 킬 조이→킬조이, 브림 스톤→브림스톤, 아스트로→아스트라, 클로버→클로브.
숫자·알파벳 콜아웃(A, B, 피 30)은 한글로 받아 적히거나("에이", "비") 빠지기 쉽다.

## 자동 갱신 (2026-10-15 스냅샷)
- 요원 (역할): 척후대: 게코, 브리치, 소바, 스카이, 케이/오, 테호, 페이드 / 감시자: 데드록, 바이스, 비토, 사이퍼, 세이지, 체임버, 킬조이 / 타격대: 네온, 레이나, 레이즈, 아이소, 요루, 웨이레이, 제트, 피닉스 / 전략가: 믹스, 바이퍼, 브림스톤, 아스트라, 오멘, 클로브, 하버
- 맵: 어센트, 스플릿, 프랙처, 바인드, 브리즈, 어비스, 로터스, 선셋, 펄, 서밋, 아이스박스, 코로드, 헤이븐
- 무기 (가격): 오딘 3200, 아레스 1600, 밴달 2900, 워든 2900, 불독 2050, 팬텀 2900, 저지 1850, 버키 850, 프렌지 450, 클래식 0, 밴딧 600, 고스트 500, 셰리프 800, 쇼티 300, 오퍼레이터 4700, 가디언 2250, 아웃로 2400, 마샬 950, 스펙터 1600, 스팅어 1100
</knowledge>
```

C. 롤 지식 묶음 `prompt/knowledge-lol.md`

```
<knowledge game="lol">
## 경기 기본
- 소환사의 협곡 5:5. 포지션: 탑, 정글, 미드, 원딜(바텀), 서폿(서포터).
```

```
- 넥서스를 부수면 승리. 15분부터 항복(서렌) 투표 가능. 내전(사용자 지정 게임)은 친구끼리 팀을 나눠 하는 경기라 실력 차가 클 수 있다.
- 오브젝트(드래곤 · 드래곤 영혼 · 장로 드래곤 · 공허 유충 · 협곡의 전령 · 바론 · 포탑 · 억제기)는 등장 시간과 효과가 패치마다 바뀐다. 시간은 자료의 이벤트를 기준으로 판단한다.

## 포지션별 기준
- 탑: 라인전(CS · 골드 차), 텔레포트로 한타 합류, 스플릿 푸시. 혼자 계속 죽어 포탑을 일찍 내주면 감점.
- 정글: 오브젝트 관리(드래곤 · 유충 · 전령 · 바론), 갱 성공, 킬 관여율. 오브젝트를 시야 없이 싸우다 내주거나, 오브젝트 콜을 두고 다른 곳에 가 있으면 감점.
- 미드: 라인전, 로밍으로 다른 라인 돕기, 한타 딜.
- 원딜: CS, 팀 딜 비중, 한타에서 살아서 딜하기. 한타 시작 전에 잡히면 큰 손실.
- 서폿: 시야(와드 · 제어 와드 · 시야 점수), 이니시에이팅 또는 아군 보호, 킬 관여율. 킬 · CS 가 적은 건 정상.
- 탱커 · 서폿은 데스가 많아도 앞에서 맞아 준 것일 수 있다. 받은 피해와 한타 결과를 같이 본다.

## 지표
- CS: 분당 7 이상 좋음, 5 아래 낮음 (라이너 기준, 정글 · 서폿 제외).
- 킬 관여율: 팀 킬 중 참여 비율. 정글 · 서폿 60% 이상이 흔함.
- 시야 점수: 분당 서폿 2 안팎, 다른 포지션 1 안팎.
- 딜 비중: 팀 챔피언 피해 중 비율. 원딜 · 미드가 보통 높다.
- 데스 시점: 오브젝트 직전이나 한타 직전 데스는 같은 1데스보다 훨씬 비싸다. 이긴 뒤 끝내러 가다 죽은 데스는 가볍게.
- 골드 차 그래프로 어느 시간대에 흐름이 넘어갔는지 본다.

## 은어 (음성 · 채팅에 나오는 말)
- 갱(다른 라인 습격), 로밍, 다이브(포탑 밑 공격), 백도어, 스플릿, 한타, 딜교(서로 피해 주고받기), 라인 밀기 · 당기기 · 프리징, 귀환 · 백, 막타(CS), 텔 · 텔포, 점멸 · 플, 궁, 쿨(재사용 대기), 핑, 와드 · 제어 와드 · 핑와, 시야, 부쉬, 블루 · 레드(버프), 바위게, 용 · 드래곤, 유충, 전령, 바론, 장로, 억제기 · 억제기 밀림, 쌍둥이 포탑, 스틸(오브젝트를 뺏음).
- 평가: 캐리, 버스 탔다(남이 이겨 줌), 던졌다 · 쓰로잉(이기던 경기를 실수로 망침), 정글 차이 · 미드 차이(그 포지션 실력 차로 짐), 트롤, 피딩(계속 죽어 상대를 키움).
- 태도: ff · 서렌 · 15ff · ㅈㅈ(포기), gg.

## 음성 인식이 자주 틀리는 말
드레곤 · 드라곤→드래곤, 협곡에 전령→협곡의 전령, 억재기→억제기, 점명→점멸, 텔 포→텔포, 리신→리 신, 쓰레시→쓰레쉬.
챔피언 이름은 줄임말로 많이 부른다(리 신→리신, 블리츠크랭크→블츠, 트위스티드 페이트→트페). 자료의 참가자 표와 맞춰 누구인지 정한다.

## 자동 갱신 (패치 16.20.1 스냅샷)
- 챔피언 (분류): 가렌(전사/탱커) · 갈리오(탱커/마법사) · 갱플랭크(전사) · 그라가스(전사/마법사) · 그레이브즈(원거리) · 그웬(전사) · 나르(전사/탱커) · 나미(서포터/마법사) · 나서스(전사/탱커) · 나피리(암살자/전사) · 노틸러스(탱커/서포터) · 녹턴(전사/암살자) · 누누와 윌럼프(탱커/마법사) · 니달리(암살자/마법사) · 니코(마법사/서포터) · 닐라(전사/암살자) · 다리우스(전사/탱커) · 다이애나(전사/암살자) · 드레이븐(원거리) · 라이즈(마법사) · 라칸(서포터) · 람머스(탱커) · 럭스(마법사/서포터) · 럼블(전사/마법사) · 레나타 글라스크(서포터/마법사) · 레넥톤(전사/탱커) · 레오나(탱커/서포터) · 렉사이(전사/탱커) · 렐(탱커/서포터) · 렝가(암살자/전사) · 로크(암살자/마법사) · 루시안(원거리/암살자) · 룰루(서포터/마법사) · 르블랑(암살자/마법사) · 리 신(전사/암살자) · 리븐(전사/암살자) · 리산드라(마법사) · 릴리아(전사/마법사) · 마스터 이(전사/암살자) · 마오카이(탱커/서포터) · 말자하(마법사) · 말파이트(탱커/마법사) · 멜(마법사/서포터) · 모데카이저(전사/마법사) · 모르가나(서포터/마법사) · 문도 박사(탱커/전사) · 미스 포츈(원거리/마법사) · 밀리오(서포터/마법사) · 바드(서포터/마법사) · 바루스(원거리/마법사) · 바이(전사/암살자) · 베이가(마법사) · 베인(원거리/암살자) · 벡스(마법사) · 벨베스(전사) · 벨코즈(마법사/서포터) · 볼리베어(전사/탱커) · 브라움(탱커/서포터) · 브라이어(전사/암살자) · 브랜드(마법사/서포터) · 블라디미르(마법사/전사) · 블리츠크랭크(탱커/서포터) · 비에고(전사/암살자) · 빅토르(마법사) · 뽀삐(탱커/전사) · 사미라(원거리/암살자) · 사이온(탱커/전사) · 사일러스(마법사/암살자) · 샤코(암살자) · 세나(서포터/원거리) · 세라핀(서포터/마법사) · 세주아니(탱커) · 세트(전사/탱커) · 소나(서포터/마법사) · 소라카(서포터/마법사) · 쉔(탱커) · 쉬바나(전사/탱커) · 스몰더(원거리/마법사) · 스웨인(마법사/서포터) · 스카너(탱커/전사) · 시비르(원거리) · 신 짜오(전사/탱커) · 신드라(마법사) · 신지드(탱커/마법사) · 쓰레쉬(서포터/탱커) · 아리(마법사/암살자) · 아무무(탱커/서포터) · 아우렐리온 솔(마법사) · 아이번(서포터/마법사) · 아지르(마법사/원거리) · 아칼리(암살자) · 아크샨(원거리/암살자) · 아트록스(전사) · 아펠리오스(원거리) · 알리스타(탱커/서포터) · 암베사(전사/암살자) · 애니(마법사/서포터) · 애니비아(마법사) · 애쉬(원거리/서포터) · 야스오(전사/암살자) · 에코(암살자/마법사) · 엘리스(암살자/마법사) · 오공(전사/탱커) · 오로라(마법사/암살자) · 오른(탱커) · 오리아나(마법사/서포터) · 올라프(전사/탱커) · 요네(전사/암살자) · 요릭(전사/탱커) · 우디르(전사/탱커) · 우르곳(전사/탱커) · 워윅(전사/탱커) · 유나라(원거리) · 유미(서포터/마법사) · 이렐리아(전사/암살자) · 이블린(암살자/마법사) · 이즈리얼(원거리/마법사) · 일라오이(전사/탱커) · 자르반 4세(전사/탱커) · 자야(원거리) · 자이라(마법사/서포터) · 자크(탱커/전사) · 자헨(전사) · 잔나(서포터/마법사) · 잭스(전사) · 제드(암살자) · 제라스(마법사/서포터) · 제리(원거리) · 제이스(전사/원거리) · 조이(마법사) · 직스(마법사) · 진(원거리/마법사) · 질리언(서포터/마법사) · 징크스(원거리) · 초가스(탱커/마법사) · 카르마(마법사/서포터) · 카밀(전사/암살자) · 카사딘(암살자/마법사) · 카서스(마법사) · 카시오페아(마법사) · 카이사(원거리/마법사) · 카직스(암살자) · 카타리나(암살자/마법사) · 칼리스타(원거리) · 케넨(마법사) · 케이틀린(원거리) · 케인(전사/암살자) · 케일(원거리/마법사) · 코그모(원거리/마법사) · 코르키(원거리/마법사) · 퀸(원거리/암살자) · 크산테(탱커/전사) · 클레드(전사) · 키아나(암살자) · 킨드레드(원거리) · 타릭(서포터/탱커) · 탈론(암살자) · 탈리야(마법사/서포터) · 탐 켄치(탱커/서포터) · 트런들(전사/탱커) · 트리스타나(원거리/암살자) · 트린다미어(전사/암살자) · 트위스티드 페이트(마법사/원거리) · 트위치(원거리/암살자) · 티모(원거리/마법사) · 파이크(서포터/암살자) · 판테온(전사/암살자) · 피들스틱(마법사/서포터) · 피오라(전사/암살자) · 피즈(암살자/전사) · 하이머딩거(마법사/서포터) · 헤카림(전사/탱커) · 흐웨이(마법사/서포터)
</knowledge>
```

D. 출력 형식 `prompt/output-schema.json`

```
{
  "type": "object",
```

```
  "additionalProperties": false,
  "required": ["summary", "confidence", "confidence_note", "teams", "players"],
  "properties": {
    "summary": { "type": "string", "description": "경기 한 줄 요약, 1~2문장" },
    "confidence": { "type": "string", "enum": ["high", "medium", "low"] },
    "confidence_note": { "type": "string", "description": "confidence 가 medium/low 인 이유. high 면 빈 문자열" },
    "teams": {
      "type": "array",
      "description": "우리 팀, 상대 팀 순서로 2개",
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": ["team", "culprit", "reason", "evidence"],
        "properties": {
          "team": { "type": "string", "enum": ["ally", "enemy"] },
          "culprit": { "type": "string", "description": "P1~P10" },
          "reason": { "type": "string", "description": "2~3문장" },
          "evidence": { "type": "array", "items": { "$ref": "#/$defs/evidence" } }
        }
      }
    },
    "players": {
      "type": "array",
      "description": "P1~P10 순서로 10명",
      "items": {
        "type": "object",
        "additionalProperties": false,
        "required": ["id", "score", "tag", "comment", "evidence"],
        "properties": {
          "id": { "type": "string", "description": "P1~P10" },
          "score": { "type": "number", "description": "1.0~10.0, 0.5 단위" },
          "tag": { "type": "string", "enum": ["캐리", "숨은 MVP", "클러치", "무난", "아쉬움", "범인"] },
          "comment": { "type": "string", "description": "한 줄 평, 40자 안팎" },
          "evidence": { "type": "array", "items": { "$ref": "#/$defs/evidence" } }
        }
      }
    }
  },
  "$defs": {
    "evidence": {
      "type": "object",
      "additionalProperties": false,
      "required": ["ref", "note"],
      "properties": {
        "ref": { "type": "string", "description": "자료의 근거 번호: e123 · u456 · c7 · r21 · s:P4" },
        "note": { "type": "string", "description": "이 근거가 뜻하는 것. 대화면 원문 인용" }
      }
    }
  }
}
```