PlantPulse® **AI Platform**

TECHNICAL WHITEPAPER

기술 백서 · REV. 2026.11 · 초안

API 로 읽고, 근거로 답하고, 사람이 승인합니다.

PlantPulse AI 는 **현장 판단 앱** 하나와 그 뒤의 **API 모듈 여럿**으로 이루어진 온프레미스 AI 플랫폼입니다. 모듈마다 맡는 일과 주인인 데이터가 하나씩 정해져 있고, 모델이 쓰는 도구는 **관문 하나**를 거쳐서만 닿습니다. 이 문서는 그 **모듈 구조**와, 답이 틀리지 않게 · 공장에 함부로 쓰지 않게 하는 **장치**를 적습니다.



22*
기본 서비스 — compose 한 파일

1
모델이 도구를 부르는 관문

27
샌드박스 탈출 시험 — 출시 게이트

매월 자동 릴리즈
서명된 폐쇄망 번들 — 무인 사슬

수치 출처: plantpulse-ai-docker MANUAL §1(서비스 수) · §6.1(탈출 시험 · 월 릴리즈) · MCP 게이트웨이 MANUAL §1(단일 관문)
* 기본 22 서비스 외에 프로필로 켜는 서비스가 4개 더 있습니다 — plantpulse-ai-docker MANUAL «기본 22 서비스(+ 프로필 4)»

2026년 11월

PlantPulse AI · 온프레미스 / 폐쇄망 · 기술 검토자용

kopens®

00 · 요약

요약 — 공장에 들어가는 AI 가 지켜야 할 세 가지

공장에서 AI 가 위험해지는 지점은 대개 모델의 실력이 아니라 **배선**입니다. 원장 DB 에 직접 붙어 스키마가 바뀌면 조용히 깨지고, 도구가 서비스마다 흩어져 누가 무엇을 부를 수 있는지 아무도 한눈에 보지 못하며, 모델이 **작업지시를 직접 고치는 길**이 어딘가에 열려 있습니다. PlantPulse AI 는 이 셋을 **구조로** 막습니다 — 모듈마다 주인인 데이터가 하나이고, 도구는 관문 하나를 거치며, 공장에 대한 쓰기는 사람이 승인한 카드로만 나갑니다.

읽기
플랫폼 데이터는 API 로
원장·시계열·이벤트는 플랫폼이 낸 API·도구로 읽습니다. DB 직결은 표마다 스위치 하나로 걷어 내는 중입니다(4장).

쓰기
AI 는 자기 저장소에만
AI 가 만든 것은 AI 소유의 `pi` 저장소에 둡니다. 고객 시스템에는 **초안·요청**까지만 만들고 확정은 사람이 합니다.

행동
쓰기는 승인 카드로만
모델이 직접 실행하는 길은 없습니다. 제안 → 사람 승인 → 실행 → 결과 기록, 영향 큰 행동은 **2인 승인**입니다.

목차

01	문제와 접근 — 현장 판단 앱과 세 원칙	3
02	전체 구조 — 층 · 관문 · 연결	4
03	모듈 지도 — 데이터 하나에 주인 하나	5
04	현장 판단 앱 — 묻기 · 제안 · 자동화	6
05	외부 입구 — 공개 API 와 대리 질문	7
06	MCP 게이트웨이 — 도구 · 권한 · 검색	8
07	시계열 인사이트 — 발견 · 진단 · 예측	9
08	디지털 스레드 — 그래프와 커넥터	10
09	RAG — 문서 · 영상 · 도면	11
10	문서 이해 · 비전	12
11	공학 계산 · 자동화 샌드박스	13
12	음성 · 파일 저장소 · 장기 기억	14
13	추론 스택 · 관리 콘솔 · 입구 프록시	15
14	신뢰 장치 ① — 근거 가드와 «모름»	16
15	신뢰 장치 ② — 범위 · 승인 · 감사	17
16	배치 — 온프레미스 · 폐쇄망 · 서명	18
17	월 릴리즈와 하드웨어 등급	19
18	평가 ① — 기계 채점 벤치 체계	20
19	평가 ② — 잰 것과 다음 판	21
—	문의	22

이 판의 기준과 표시. 2026년 11월 1일 시점 각 모듈 저장소의 문서와 코드가 근거입니다. 기본으로 켜져 있는 것은 그대로 적고, 구현됐지만 설치 기본값이 꺼진 것은 **기본 꺼짐** , 다음 릴리즈에 실리는 것은 **예정** 으로 표시합니다. 숫자는 문서가 **실측**으로 적은 것만 옮기고, 그 자리에 출처(문서 · 절)를 작게 붙였습니다. 설정 한도는 «설정값» 이라고 밝힙니다.

독자. 기술 검토자 · 정보보안 담당 · 플랫폼 운영자. 영업 자료가 아닙니다 — 모듈마다 **무엇을 맡고, 어떤 API 를 내고, 어떤 데이터의 주인인지**를 설치 전에 검토할 수 있을 만큼 적었습니다. 모델 자체의 학습과 평가는 별도의 모델 기술 백서가 다룹니다.

이 문서에서 쓰는 말

`pi`	AI 소유 저장소. PostgreSQL DB `pi` 안에서 앱마다 스키마 하나, 분석 데이터는 Cassandra keyspace `pi`
MCP	모델이 도구를 찾고 부르는 표준 프로토콜. 이 플랫폼에서 모델은 **게이트웨이 하나**로만 도구에 닿습니다
발견(finding)	시계열 인사이트가 내는 판정 단위 — 숫자와 사실 코드, 기계가 읽는 계약(`findings/1`)
그림자(shadow)	결과를 계산 · 기록하되 현장에 보이지 않고 견주기만 하는 상태. 관문을 넘어야 현장에 오릅니다
승인 카드 · L3 행동	고객 시스템에 초안 · 요청을 만드는 행동 제안. 사람이 승인해야 디지털 스레드가 실행합니다
범위(scope)	사람 · 키가 볼 수 있는 공장동 · 라인 · 설비. 판정할 수 없으면 «안 보임» 입니다

01 · 문제와 접근

챗봇이 아니라 «현장 판단 앱» 입니다

PlantPulse AI 는 **공장의 문제를 찾아 근거와 함께 제안하고, 사람이 승인해 닫고, 모르면 묻고, 반복되는 일은 자동화에 맡기는** 앱입니다. 채팅은 그 입구 중 하나일 뿐입니다. 고객의 MES · ERP 를 대신하지 않고, 그 옆에서 **판단을 돕는 층**으로 붙습니다.

메뉴 ①
묻기
채팅 · 음성 · 사진. 첫 화면은 «오늘 할 일» 요약 — 수행 제안 · 확인할 이상 · 이상 접수 · 자동화 제안.

메뉴 ②
제안
계획된 일과 돌발이 한 목록입니다. 제안 하나가 감지 → 승인 → 조치 → 결과 → **재발 검증**까지 한 상세에서 흐릅니다.

메뉴 ③
자동화
말로 만드는 산업 자동화. 만들기 · 미리 돌려 보기 · 실행 기록 · 버전 이력. 만든 사람의 권한으로, 읽기 도구만 씁니다.

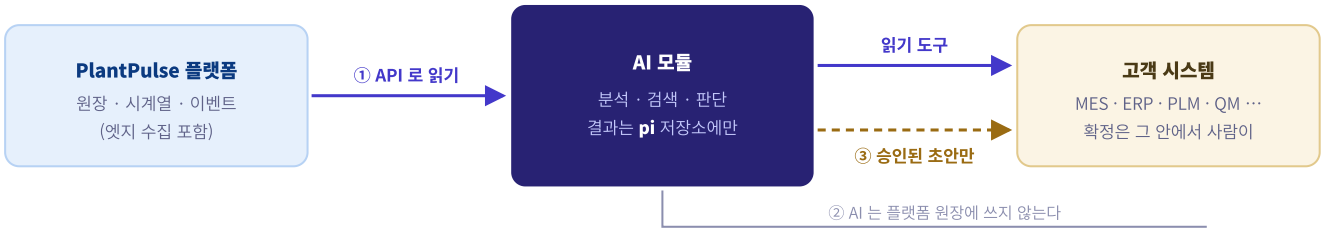
출처: copilot BUSINESS-PROCESS §1 · MANUAL §1a(메뉴 셋 — 묻기 `/chat` · 제안 `/tasks` · 자동화 `/automations`) · §5.30(자동화)

무엇이 «아닌가» 를 먼저 정했습니다

아닌 것	왜
MES · ERP	계획 · 작업지시 · 실적 · 로트 · 납기의 정본은 고객 시스템입니다. AI 가 두 번째 정본을 들면 둘 중 하나는 틀립니다
설비 제어	설정값 변경은 범위 밖입니다. **AI 는 제안하고 사람이 결정합니다**
BI 대시보드	숫자를 보여 주는 화면은 플랫폼에 있습니다. 여기서는 숫자를 **판단과 행동**으로 잇습니다
챗봇	채팅은 입구 하나입니다. LLM 이 멈춰도 제안 목록 · 자동화 같은 **결정론 화면**은 계속 돕니다

출처: plantpulse-ai-master ECOSYSTEM §11.0(제품의 자리) · §12.8(현장 판단 앱 — «아니다» · 원칙)

세 원칙 — 데이터가 어디서 오고, 어디로 가는가



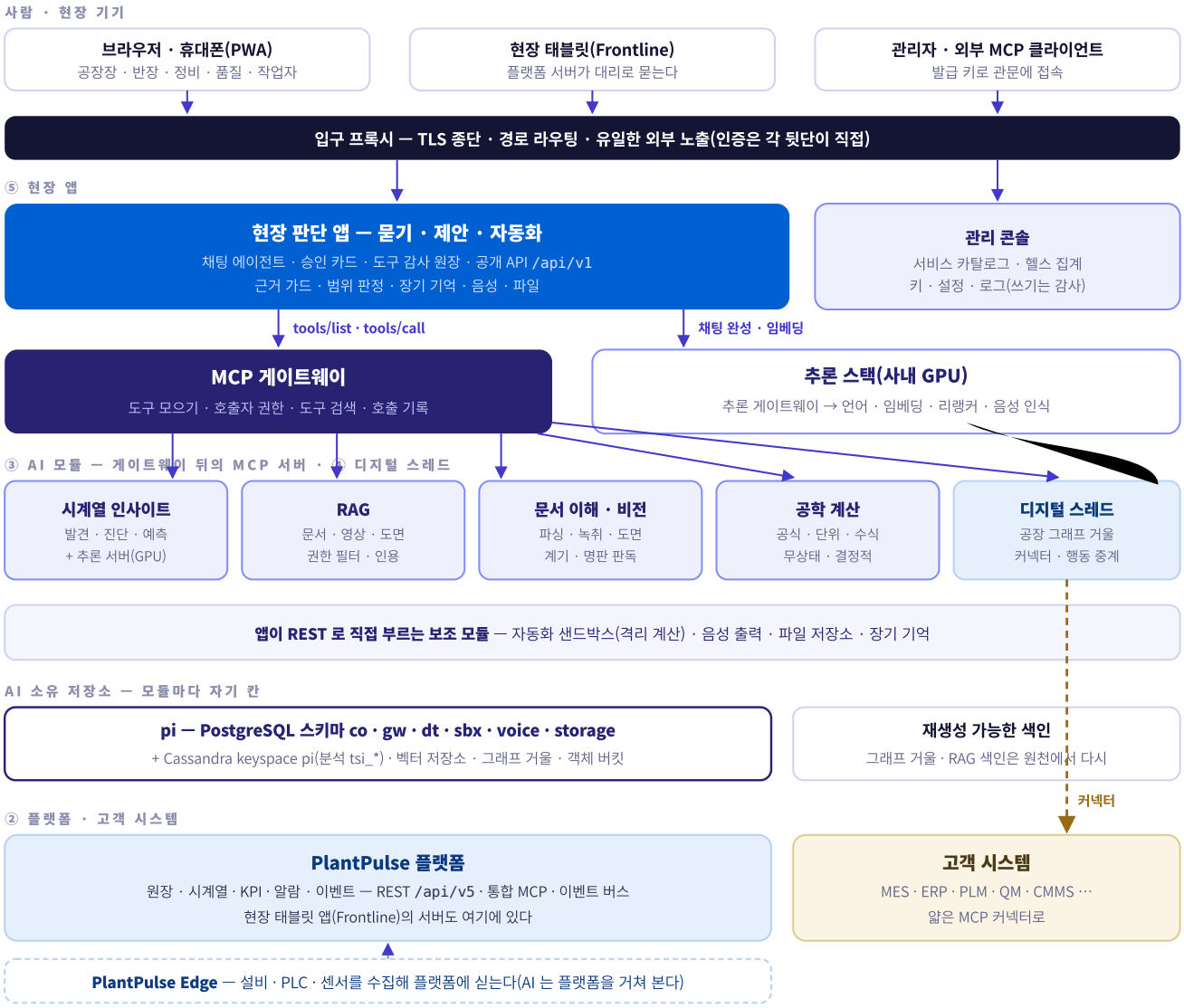
① API 로 읽는다	플랫폼 원장 · 이벤트는 플랫폼 REST · 통합 MCP · 이벤트 버스로 받습니다(결정 2026-09-28). **이행 중**입니다 — 표마다 `db` · `api` · `shadow` (두 길을 모두 읽어 견줌) 스위치를 두고, 그림자에서 깨끗한 것부터 옮깁니다
② pi 에만 쓴다	AI 데이터는 AI 앱들이 나눠 쓰는 DB `pi` 에 두고, 앱마다 스키마 하나를 갖습니다. 플랫폼 쪽 쓰기는 **합의된 예외 하나**와 `pi` 로 옮길 대상 하나뿐입니다
③ 사람이 승인한다	공장에 대한 행동은 제안 카드 → 사람 승인 → 실행 → 결과 기록. 고객 시스템에는 초안 · 요청까지만 만듭니다

출처: plantpulse-ai-master docs/platform-data-via-api §0(결정 다섯) · ECOSYSTEM §11.4-k(HITL) · §11.7(AI → 플랫폼 쓰기 현재 2) · copilot PG-SCHEMA §1(`pi` 스키마 규칙)

02 · 전체 구조

층은 위에서 아래로만 부르고, 도구는 관문 하나를 거칩니다

바깥에 열린 입구는 **프록시 하나**입니다. 사용자의 질문은 현장 판단 앱이 받고, 앱은 모델을 **추론 스택**으로, 도구를 **MCP 게이트웨이**로만 부릅니다. 고객 시스템을 부르는 것은 **디지털 스레드 하나**입니다.



출처: ECOSYSTEM §1(한눈에 보는 구조) · §11.1(층 구조 · 층 규칙 1~4) · plantpulse-ai-proxy MANUAL §1 · copilot MANUAL §1 · v1-chat-on-behalf 머리말(Frontline)

바깥과 닿는 자리

상대	방식	방향
PlantPulse 플랫폼	REST `/api/v5` · 통합 MCP(게이트웨이 경유) · 이벤트 버스. 접속 자격증명은 **기동할 때 플랫폼에서** 받고, 닿지 않으면 기동하지 않습니다	읽기 — 일부 표는 DB 직결을 스위치로 이행 중
PlantPulse Edge	직접 연결 없음 — 플랫폼에 실린 데이터와 플랫폼 도구(수집 연결 상태 등)로	읽기
현장 태블릿	플랫폼의 태블릿 서버 → 공개 채팅 v1(`on_behalf`)	질문 → 답, 쓰기 도구 없음
고객 시스템	디지털 스레드의 커넥터(얇은 MCP 서버)	읽기 · 승인된 초안만
외부 MCP 클라이언트	게이트웨이 `/mcp/` + 발급 키(범위 지정)	허락된 도구 호출

출처: ECOSYSTEM 머리말(자격증명 — 시작할 때 받음) · §11.2 · §11.4 · docs/platform-data-via-api §0 · copilot MANUAL §4.4 · gateway MANUAL §6.2

03 · 모듈 지도

데이터 하나에 주인 하나 — 모듈 열넷의 자리

각 모듈은 **저장소 하나 · 이미지 하나**이고, 자기가 주인인 데이터만 씁니다. 남의 데이터는 그 주인이 낸 API · 도구로 읽습니다. MCP 도구는 서버마다 **이름 앞붙이**가 있어, 어느 모듈의 도구인지 이름만 보고 압니다.

모듈	맡는 일	주요 API · MCP	주인인 데이터
현장 판단 앱	묻기 · 제안 · 자동화, 채팅 에이전트, 승인 · 감사	`/api/v1` · `/api/ui` · MCP 읽기 3종	PG `pi.co` — 사용자 · 대화 · 제안 · 결과 · 도구 감사 원장
MCP 게이트웨이	도구 모으기 · 호출자 권한 · 발급 키 · 도구 검색	`/mcp/` · `/api/v1/tools` · `/search`	PG `pi.gw` — 서버 · 도구 · 권한 · 키 · 호출 기록
시계열 인사이트	이상 · 예측 · 건강 지수 · 발견 피드 · 고장 진단	`/api/v1/findings` · `/forecast` · `insight_*`	Cassandra `pi.tsi_*` — 기준선 · 발견 · 예측 원장
시계열 추론 서버	이상 점수 · 예측 · 파형 임베딩 추론(GPU)	`/infer/*` · `/finetune/*` (내부 전용)	현장 미세조정 가중치(볼륨)
디지털 스레드	공장 그래프 거울 · 커넥터 · 영향 범위 · 행동 중계	`/api/v1/timeline` · `/impact` · `thread_*`	그래프 `:Dt*` (재생성 가능) · PG `pi.dt`
RAG	문서 · 영상 · 도면 색인과 검색, 권한 필터	`/api/v1/query` · `rag_*` 2종	상태 DB · 벡터 · 객체 버킷(문서 · 영상 · 쪽 그림)
문서 이해	문서 파싱 프로필 · 영상 녹취 · 도면 파싱 · 판 비교	`/api/v1/understand` · `/transcribe` · `document_*` 2종	없음(무상태)
비전	분류 · 검출 · 검사 · OCR · 계기 판독 · 분석	`/api/v1/read_meter` 등 · `vision_*` 5종	없음(무상태 — 사진은 처리 후 버림)
공학 계산	공식 · 단위 환산 · 허용 목록 수식	`eng_*` 4종	없음(무상태)
자동화 샌드박스	승인된 계획의 파이썬 계산을 격리 실행	`/api/v1/run` (REST 만)	PG `pi.sbx` — 실행 기록(내용 미보관)
음성 출력	읽을 글 정규화 · 문장별 합성 · 캐시	`/api/v1/speech`	PG `pi.voice` · 문장 캐시
파일 저장소	원본 바이트 · 보존 등급 · 개인정보 삭제 · 감사	`/api/v1/objects` · `/uploads`	PG `pi.storage` · 원본 / 파생 버킷
장기 기억	사용자 · 자동화 범위의 기억 저장과 의미 검색	`/memories` · `/search`	자기 벡터 컬렉션 · 변경 이력
관리 콘솔	카탈로그 · 헬스 · 키 · 설정 · 로그(관리자 전용)	`/api/v1/{service}/…` 중계	없음(설정은 설치 env)

출처: 각 저장소 MANUAL §1 · §4 · §5 · ECOSYSTEM §11.2(소유 표) · repos.tsv. 이 밖에 운영 구성으로 추론 스택 · 입구 프록시(13장)와 배포 허브(16장)가 있습니다.

계약은 주인 저장소에

계약은 주인이 갖고 소비자는 판을 고정합니다. 시험은 양쪽에 둡니다 — 주인은 «내 출력이 스키마에 맞는다», 소비자는 «고정한 판을 읽을 수 있다».

호출은 위에서 아래로

아래 층은 위 층의 주소를 모릅니다. 같은 층 안의 호출은 **선언된 것만** 둡니다 — 지금은 둘(비전 → 문서 이해의 OCR 위임, RAG → 인사이트의 설비 조회).

헬스는 «인증» 을 잰다

연결이 아니라 인증된 읽기 한 번이 성공해야 «살아 있음» 입니다. 401 이 UP 으로 보이던 일을 겪고 정한 규칙입니다.

출처: ECOSYSTEM §11.3(계약 규칙) · §11.1(층 규칙 1 · 2) · §11.4-d(헬스)

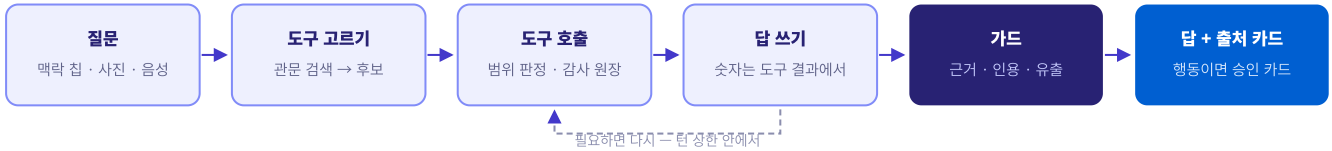
04 · 현장 판단 앱

모델을 부르는 곳은 한 곳, 결정하는 사람은 따로

현장 판단 앱은 사용자가 만나는 유일한 화면이자, 분석 모듈들을 엮는 **오케스트레이터**입니다. 에이전트 루프가 모델에 도구 목록을 주고, 모델이 고른 도구를 게이트웨이로 부르고, 결과를 다시 넣어 답을 만듭니다.

형태	**웹앱 · PWA**
공개 계약	**/api/v1/****
저장	**PG pi.co**
도구 경로	**게이트웨이 하나**
모델 경로	**추론 스택**

한 번의 질문이 지나는 길



주요 API

`/api/v1/**`	**공개 계약은 이것 하나**. 외부 시스템은 `X-API-Key` (해시로만 저장, 폐기 즉시 반영). 도구를 «부르는» 길은 열지 않습니다
`/api/ui/**`	화면 전용 API — 계약 아님
`/admin/api/**`	관리 REST — 설정 · 프롬프트 · 감사
`POST /mcp`	앱이 «내는» MCP — `ask_plant` · `list_plant_tools` · `daily_tasks`, 셋 다 읽기 전용

주인인 데이터 — PG `pi.co`

대화 · 앱	사용자 · 채팅 · 프롬프트(정본은 DB 한 곳) · 설정
판단	제안 · 결과 · 이상 접수 · Episode
감사	도구 감사 원장(만료 없음) · 설정 변경 · 열람 기록
비밀	저장 시 AES-256-GCM — **DB 에 평문이 없습니다**

자동화 — 말로 만들고, 만든 사람의 권한으로

사용자는 «언제 · 어디를 · 무엇으로 · 어떻게 · 어디에» 다섯 칸을 채워 자동화를 만듭니다. 자동화는 **만든 사람의 권한**으로, **읽기 도구만** 써서 제안 카드를 냅니다. 계획형 자동화는 **계획이 승인돼야** 켜지고, 쓰기 도구는 자동화에서 **절대 실행되지 않습니다**. 회차마다 턴 · 시간 · 토큰 상한이 있습니다(설정값).

«발견» 은 받기만 합니다

이상 · 지연 · 품질 신호의 정본은 시계열 인사이트의 **발견 피드**입니다. 앱은 계약(판 `findings/1`)의 바이트 사본으로 받은 것을 검증하고, 자기 칸(누가 보고 무엇을 했는가)만 덧붙입니다. 디지털 스레드에는 **참조만** 보내고 진단 · 제안 문장 · 메모는 보내지 않습니다.

> **LLM 이 멈춰도 현장 일은 계속됩니다.** 제안 목록 · 이상 접수 · 자동화 실행 기록은 결정론 화면입니다. 모델은 설명 · 요약 · 초안을 맡고, 쓰기는 사람이 승인합니다. 외부 의존(플랫폼 API · 알람 채널 · LLM)의 상태는 헬스의 별도 층으로 따로 보고합니다.

출처: copilot MANUAL §1 · §2 · §4.3 · §5.4 · §5.6 · §5.18 · §5.30 · §5.30.1 · ECOSYSTEM 머리말(헬스 층) · §11.3(발견 계약) · §12.8(원칙)

앱이 부르는 곳과 걸어 둔 상한

대상	무엇을	어떻게 · 상한(설정값)
MCP 게이트웨이	모델의 모든 도구	질문마다 도구 검색(예산 20). 검색이 꺼지면 턴을 오류로 끝냅니다 — 도구 전부를 조용히 싣지 않습니다
추론 스택	채팅 완성	출력 4,096 토큰 · 도구 결과 24,000자로 잘라 넣음 · 연속 실패 5회면 30초 차단
시계열 인사이트	발견 피드	REST 커서 피드 — 계약 사본으로 검증
샌드박스 · 음성 · 파일 · 기억	계산 · 읽어 주기 · 첨부 · 기억	REST — 모델 도구가 아니라 앱의 부품(11 · 12장)
플랫폼	알람 · 화면 데이터	V5 REST — 일부는 `db` · `api` · `shadow` 스위치로 이행 중

출처: copilot MANUAL §4 · §4.4 · §5.5 · §5.21 · §5.33

05 · 외부 입구

현장 태블릿은 «작업자를 대신해» 묻습니다 — 범위는 닫힌 채로

플랫폼의 현장 태블릿 앱(Frontline)은 AI 를 화면에 직접 붙이지 않습니다. 태블릿의 서버(BFF)가 **키를 서버에만 둔 채** 공개 채팅 API 를 부르고, 묻는 작업자가 누구인지 · 어디까지 볼 수 있는지를 `on_behalf` 로 함께 싣습니다. 예정 — 다음 재릴리즈

공개 채팅 v1

`GET /api/v1/chat/capabilities`	계약 판을 알려 줍니다. 옛 서버면 404 → 소비자가 채팅을 숨깁니다
`POST /api/v1/chat`	한 번에 답
`POST /api/v1/chat/stream`	SSE 로 흘려 답
`POST /api/v1/chat/cancel`	진행 중인 턴 취소

요청에 싣는 것

on_behalf	사번 · 표시 이름 · 역할 · **scope**(필수)
context	보고 있는 설비 · 오더 · 교대 · 화면
options	언어 · 답 길이(짧게 / 보통) · 음성
첨부	사진(설비 판독) · 음성 질문과 음성 답 음성은 예정

범위는 «닫힌 쪽» 으로 판정합니다

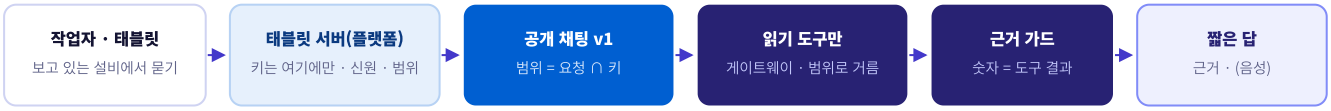
- `scope` 는 `all` 과 `nodes` 중 **정확히 하나**입니다
- **빈 목록은 거절**(400)입니다 — «전부» 로 읽지 않습니다
- 노드는 최대 200개, 모르는 노드가 섞이면 400 입니다(설정값)
- 실제 범위는 **키의 범위와의 교집합** — 요청이 범위를 넓힐 수 없습니다
- 대화의 주인은 «키 + 작업자» 입니다. 같은 키의 다른 작업자 대화는 보이지 않습니다
- **쓰기 도구는 이 길에 열지 않습니다**(도구 호출 가드는 끌 수 없습니다)

답의 근거 가드

이 턴에 **성공한 도구가 없는데** 데이터를 주장하면 한 번 다시 묻고, 그래도 같으면 «조회하지 못해 수치로 답할 수 없다» 로 바꿉니다. 도구 결과에 없는 숫자는 한 번 고쳐 쓰게 하고, 남으면 «확인되지 않은 숫자» 경고 줄을 붙입니다. **이전 답은 근거가 아닙니다**(14장).

> **플랫폼에 쓰지 않는다는 것이 계약에 들어 있습니다.** 대리 질문 중에 플랫폼 도구를 부를 때 «누가 수행했다» 칸을 싣지 않습니다 — AI 는 플랫폼에 쓰지 않기 때문입니다. 계약은 플랫폼 쪽과 합의해 판(`2026-10-30`)을 매기고, 주인 쪽 시험 셋이 지킵니다.

한 번의 대리 질문



대화 주인 = 키 + 작업자 · 빈 범위 · 모르는 노드는 400 · 옛 서버면 소비자가 채팅을 숨긴다

엣지와는 어떻게 닿나

AI 는 엣지에 직접 붙지 않습니다. 엣지가 수집한 설비 데이터는 플랫폼에 실리고, AI 는 **플랫폼의 도구**(수집 연결 상태 · 최신 태그 값 등)로 봅니다. 엣지가 늘어도 AI 쪽 배선은 바뀌지 않습니다.

출처: copilot docs/design/v1-chat-on-behalf 머리말(구현 · 스테이징 반영, 음성 §11 미반영, 재릴리즈에 실림) · §0 · §2.1 · §5 · §9.2 · §10 · §11 · MANUAL §4.4 · ECOSYSTEM §11.3(AI 공개 채팅 v1)

06 · MCP 게이트웨이

도구는 여럿, 관문은 하나

설치 안의 모든 MCP 서버 — 플랫폼 통합 도구, 내장 AI 모듈, 운영자가 등록한 외부 서버 — 앞에 **주소 하나**를 둡니다. 목록을 모으고, 호출을 **결과 수정 없이** 넘기고, 누가 무엇을 부를 수 있는지 판정하고, 질문에 맞는 도구를 고릅니다.

프로토콜	**MCP(HTTP 스트림)**
바깥 노출	**/mcp/ 만**
저장	**PG pi.gw**
도구 앞붙이	**서버마다 하나**

주요 API

`/mcp/`	`tools/list` · `tools/call`. 바깥에는 이 경로만 열고 REST 는 열지 않습니다
`/api/v1/servers`	서버 등록 · 인증 · 발견 · 도구별 켬/끔
`/api/v1/tools`	도구 전수 — 켜짐 · 권한의 정본
`/api/v1/permissions`	호출자 × 서버 · 도구 권한, 실효 권한 조회
`/api/v1/api-keys`	외부 클라이언트용 발급 키(SHA-256 해시만 저장)
`POST /api/v1/search`	질문 → 도구 후보(키워드 + 의미 혼합)
`/api/v1/calls`	호출 기록 — 인자 · 결과 · 인증 값은 남기지 않음

닫힌 쪽이 기본입니다

- 인바운드 키가 비어 있으면 **기동을 거부**합니다
- 새 발급 키는 **권한도 범위도 없이** 태어납니다. 범위 없는 키가 내장 · 플랫폼 도구를 부르면 거절(`forbidden_scope`)
- 발급 키는 `/mcp/` 만 — 관리 REST 는 403
- 외부 서버 · 도구는 **꺼진 채로** 들어와 관리자가 검토한 뒤 켭니다. 쓰기로 의심되는 도구는 켤 수 없고, **정의가 바뀌면 스스로 끕니다**
- 외부 서버 인증 값은 AES-256-GCM 으로 봉인, 루프백 · 링크로컬 · 메타데이터 주소는 거부

도구 검색 — 도구가 늘어도 모델이 보는 목록은 짧게

작은 모델은 한 번에 보는 도구가 많을수록 고르기를 틀립니다. 그래서 질문마다 관문이 **예산(기본 20개, 설정값)**만큼 후보를 골라 주고, 핵심 도구(설비 찾기 · 문서 검색)는 예산과 상관없이 늘 싣습니다. 아래는 도구 76종 위에서 툴콜 벤치 36문항으로 잰 값입니다.

방식	재현율@10	재현율@20	hit@1	hit@3	hit@10	MRR
키워드(BM25)	0.806	0.944	0.722	0.917	0.972	0.816
키워드 + 의미 혼합	**0.889**	**0.972**	0.722	0.917	**1.000**	**0.828**

앱 경로로 채점하면 예산 20 · 10 모두 36/36 이었습니다. 서버가 선언한 검색어를 빼면 키워드 재현율@10 이 0.750 으로 떨어집니다 — **도구 설명을 쓰는 일이 곧 검색 품질**입니다. 검색어 도배를 막으려 도구당 검색어 8개 · 64자 상한(설정값)을 둡니다.

출처: plantpulse-mcp-gateway-api MANUAL §1 · §2 · §5.1 · §5.2(2026-10-18 실측 — 도구 76종 · 36문항) · §6.1 · §6.2 · §6.4 · CONTRACT §2 · §3

기본으로 켜진 도구. 시계열 인사이트 34 · 디지털 스레드 9 · 비전 5 · 공학 계산 4 · RAG 2 · 문서 이해 2 와 플랫폼 통합 도구. 출시 게이트는 관문의 도구 목록을 직접 세어, 인사이트 34 · 스레드 9 · 공학 계산 4 아래로 내려가면 출시를 멈춥니다.

출처: 게이트웨이 MANUAL §1(기본 켬) · plantpulse-ai-docker MANUAL §6.1(`probe-gateway-tools` 게이트)

07 · 시계열 인사이트

숫자와 사실 코드를 내고, 문장은 쓰지 않습니다

설비 데이터(센서 · OEE · 가용도 · 에너지 · 작업지시 · 알람)를 읽어 이상 · 예측 · 건강 지수 · 발견을 냅니다. 모델 가중치는 **별도의 추론 서버**에 있고, 이 모듈은 시간 제한 · 재시도 · 차단기를 둔 클라이언트로 부릅니다.

MCP 도구	**42종**
발견 계약	**findings/1**
저장	**Cassandra pi.tsi_***
추론	**추론 서버(GPU · CPU)**

주요 API

`/api/v1/findings`	발견 피드 — 커서로 빈틈 · 겹침 없이, 계약은 `/findings/contract`
`/api/v1/anomaly`	설비 · 시스템 · OEE · 가용도 · 에너지 이상
`/api/v1/forecast`	예측 — 점 예측 + 구간, 쓴 엔진과 판
`/api/v1/intel/*`	주의 목록 · 설명 · 가설 검증 · 고장 유형 · 그래프
`/api/v1/asset/health`	설비 건강 지수
`/api/v1/order/completion`	작업지시 완료 예측

MCP 42종은 알람 · 건강 6 · 분석 18 · 특징 저장소 7 · 인텔 9 · 프로필 2로 나뉩니다. 도구 설명은 도구당 480자 · 합 8,000자 상한(설정값)입니다.

발견 계약 — 15 종류, 기계가 읽는 스키마

이상 · 작업지시 미달 · 데이터 공백 · 데이터 품질 · 연쇄 정지 · 고장 위험 · 한계 도달 예상 · 에너지 피크 · OEE 손실 · SPC 신호 · 준비 시간 초과 · AI 변화 · 지연 전파 · 지연 회복 등. 발견은 **숫자와 사실 코드** `{code, params, text}` 와 **쓰지 못한 입력** `missing[]` 을 냅니다. 문장은 앱이 씁니다.

«모름» 은 «못 봄» 이 아닙니다

원천을 읽지 못한 회차에는 열린 발견을 **닫지 않습니다**. 읽지 못한 것을 «이상 없음» 으로 닫으면 아무도 보지 않은 설비가 정상이 됩니다.

진단 — 모델이 «가설» 을 내고, 숫자가 «판정» 합니다

가설 검증	LLM 이 낸 가설(최대 3개)은 **태그와 기대 모양**(오름 · 내림 · 고착 · 선행 · 분리)으로만 받습니다 — 글은 받지 않습니다. 증상마다 `supported` · `refuted` · `unmeasurable`(잴 수 없음)과 근거 숫자를 냅니다. 점수는 개수이지 **확률이 아닙니다**
고장 유형 진단	설비 9유형 · 고장 모드 27개의 사전(국제 규격의 개념 코드에만 맞춤). 후보가 되려면 핵심 점검이 모두 참이고 **움직인 태그가 둘 이상**이어야 합니다 — **센서 하나로는 원인을 부르지 않습니다**. 못 미치면 «원인 후보 불충분» 과 이유 코드

예측 — 엔진 하나로만 답하고, 내놓은 예측은 채점합니다

조용한 폴백이 없습니다. 데이터가 모자라면 422, 보조 엔진이 실패해 기본 엔진이 답하면 `degraded` 로 밝힙니다. 내놓은 예측은 모두 **예측 원장**(`pi.tsi_forecast_ledger`)에 남아 나중에 실측과 대조되고, 실측을 못 구하면 `unresolvable` — **0 으로 채우거나 근사로 채점하지 않습니다**.

> **새 판정은 그림자에서 시작합니다.** 현장 모드는 **사전 등록한 관문**을 통과하고 사람이 승인해야 켜지며, 누가 언제 무엇을 바꿨는지 남습니다. 그림자 발견은 채팅 도구에서 빠집니다 — 검증되지 않은 판정이 사실처럼 전달되지 않게.

출처: plantpulse-timeseries-insight-api MANUAL §1 · §3 · §5(도구 42종 생성 블록) · INTEL «발견 계약» · «고장 가설 검증» · «고장 유형 사전·진단» · «예측 엔진» · «예측 원장» · «현장 모드» · insight-server MANUAL §1 · §5

08 · 디지털 스레드

값은 원천에 두고, «연결» 만 색인합니다

시간축 위의 공장 그래프입니다. 플랫폼 원장의 **키와 관계**, 커넥터로 붙인 고객 시스템의 객체, 사건 · 발견 · 제안의 **참조**를 한 그래프에 잇습니다. 페이로드 사본은 두지 않으므로 언제든 원천에서 **다시 만들 수 있습니다**.

그래프	**참조 거울 :Dt***
갱신	**계열별 신선도 기준에 맞춰 주기 동기화**
저장	**PG pi.dt**
고객 시스템	**이 모듈만 부른다**

주요 API

`/api/v1/timeline`	객체의 시간순 사건 · 발견
`/api/v1/neighbors` · `/trace`	연결 따라가기
`/api/v1/impact`	영향 범위 — 부품 지연 → 작업 → 고객 주문 납기
`/api/v1/readiness`	작업 준비도 판정(Go / No-Go)
`/api/v1/connectors`	규격 · 등록 · 발견 · 적합성 · 바인딩 · 읽기 · 행동
`/api/v1/optimize/recovery`	회복안 계산 · 일정 시나리오 «초안»

채팅에 내는 도구는 **소수**로 고정합니다(`thread_get_timeline` · `thread_trace_links` · `thread_impact_range` 등). 커넥터가 몇 개 붙든 모델이 보는 도구 수는 늘지 않습니다.

표준 커넥터 규격 — 얇은 MCP 서버

고객 시스템은 계열별 규격(MES · ERP · PLM · QM · CMMS · 공구 · 창고 · 시험실 · EHS 등)에 맞춘 **얇은 MCP 서버**로 붙습니다. 등록 전에 **적합성 시험**을 통과해야 하고, 바인딩으로 «외부 키 ↔ 우리 키» 를 선언합니다. 커넥터 비밀은 암호화해 저장하고 평문 폴백이 없습니다.

상용 제품 어댑터	상태
SAP S/4HANA	규격 적합성 시험을 **모의 서버로** 통과. **실제 제품 접속 검증은 파일럿 현장에서** 예정
IBM Maximo	
Siemens Teamcenter	
Dassault 3DEXPERIENCE	

고객 시스템에 «쓰는» 유일한 길 — 승인된 L3 행동

필수 승인 칸 다섯	`proposal_id` · `approved_by` · `approved_at` · `reason` · `idempotency_key` . 하나라도 없으면 실행하지 않습니다
만드는 것	**초안 · 요청만** — 협력사 독촉 요청 초안, 정비 요청, 일정 시나리오 초안. 계획을 확정 · 게시하지 않습니다
관리 콘솔	행동 도구를 부르지 못합니다. 행동은 앱의 승인 카드에서만 시작됩니다
«모름» 은 Go 가 아니다	준비도 판정에서 원천을 못 읽은 항목은 통과로 세지 않고, 오래된 커넥터 데이터로는 숫자를 내지 않습니다 (`stale_inputs`)

원장은 읽기 전용 트랜잭션으로 읽습니다. 플랫폼 원장을 DB 로 읽는 길은 `READ ONLY` 트랜잭션이고, 플랫폼 V5 API 로 읽는 길이 스위치 뒤에 있습니다. 그림자 비교가 14일 깨끗해야 DB 길을 걷어 냅니다(이행 중).

출처: plantpulse-digital-thread-api MANUAL §1 · §2 · §4 · §5 · §6.3.1 · §6.4 · §6.5 · §6.14 · plantpulse-dt-connectors FEATURES «무엇이 실제인가» · ECOSYSTEM §11.5 · §11.4-m

그래프에 담는 세 가지 — 그리고 담지 않는 것

노드	예	원본은
원장 키 투영	사이트 · 설비 · 태그 · 설비 관계	플랫폼 원장
커넥터 객체	작업지시 · 공정 · 부품 · 로트 · 일련번호 · 발주 줄 · 협력사 · 도면 · 검사 특성 · 정비 오더	고객 시스템
시간축 참조	사건 · 인사이트의 발견과 학습된 연결 · 앱의 Episode · 제안 (`id` 만)	플랫폼 · 인사이트 · 앱

관계는 «소요» · «자격» · «배정» · «변경» · «구성품» · «부적합» · «막음» · «검사» · «바인딩» 같은 이름으로 잇습니다. 담지 않는 것 — 사건 요약 · 속성 · 커넥터가 돌려준 값(물을 때 원천에서 읽음), 분석(인사이트가 계산), 사람의 결정(앱이 가짐). 원장만으로 답이 되는 질문은 플랫폼 도구가 받습니다.

출처: digital-thread-api MANUAL §5 · §6.4 · §6.5 · §6.12 · ECOSYSTEM §11.5(맡는 것 · 맡지 않는 것) · §11.2 O9

09 · RAG

권한은 «답을 쓰기 전에» 거릅니다

설비 문서 · 매뉴얼 · SOP 를 색인하고 질문에 근거를 찾아 답합니다. 그래프 기반 검색 엔진을 **이 관문 뒤에만** 두어, 어떤 소비자도 엔진을 직접 부르지 않습니다. 화면은 관리 콘솔에 있습니다.

MCP 도구	**rag_ 2종(읽기)**
동기화	**원장과 600초마다**
영상	**기본 꺼짐**
도면	**기본 꺼짐**

문서 기본 켬

원장의 문서를 따라갑니다

플랫폼 원장의 설비 문서 목록과 상태를 견줘 새것 · 바뀐 것 · 사라진 것을 반영합니다. 한 회차 상한 · 파일 크기 상한을 둡니다(설정값).

영상 기본 꺼짐

녹취 → 구간 → 장면 칩

음성을 녹취해 약 120초 구간으로 색인하고, 답에 **장면 시각 칩**(시작 · 끝 · 썸네일)을 붙입니다. 처리는 야간 창에서만 돕니다.

도면 기본 꺼짐

표제란 · BOM · 치수

DXF · 벡터 PDF 도면에서 표제란 · BOM · 구역 · 치수를 뽑아, 정확한 치수 · 번호 일치를 앞에 둡니다. **판 비교**를 지원합니다.

출처: plantpulse-rag-api MANUAL §1 · §4(`MEDIA_INGEST_ENABLED` · `DRAWING_INGEST_ENABLED` 기본 false) · §5 · §9 · plantpulse-ai-docker MANUAL §5.10

주요 API

`POST /api/v1/query`	질문 → 답 + 출처(인용 `[n]`)
`/api/v1/documents` · `/sync`	색인 상태 · 원장 동기화 회차
`/api/v1/docs/…`	올리기 · 재색인 · 제외 · 내려받기
`/api/v1/grades`	문서 등급 지정
`/api/v1/media/…`	영상 작업 · 녹취 · 썸네일 · 무결성
`/api/v1/drawings/…`	쪽 그림 · 판 상태 · 판 비교

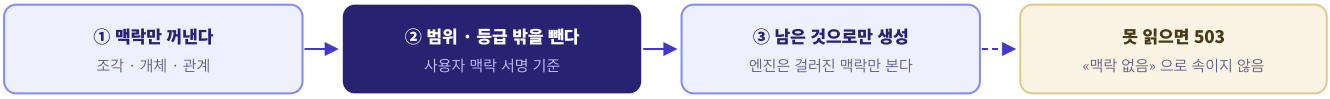
인용은 «틀린 구절보다 없는 구절»

출처마다 원문 한 줄(최대 200자)을 구절로 붙이되, 질문과 충분히 겹치지 않으면 **구절을 붙이지 않습니다**. 쪽 번호는 일치가 하나뿐일 때만 줍니다. 채팅은 본문의 인용 표시를 코드로 세어 출처 카드와 맞춥니다.

등급 넷

일반 · 사내 · 대외비 · 통제. 통제 등급 원본은 **내려받을 수 없고**(403), 미리보기에는 워터마크가 들어갑니다.

권한 필터의 순서



생성 «뒤에» 거르면 이미 답에 섞인 내용을 지울 수 없습니다. 그래서 생성 «앞에서» 뺍니다. 거부는 «없음»(404)과 같은 모양이라, 범위 밖 문서가 **있다는 사실**도 새지 않습니다. 권한 입력을 읽지 못하면 `ACCESS_UNAVAILABLE`(503)입니다.

출처: plantpulse-rag-api MANUAL §5 · §5.1(권한 필터 · 등급 · 거부 모양) · copilot MANUAL §5.2b · ECOSYSTEM §12.4-3(인용 판정)

다른 곳에서 읽는 것

설비 문서 목록	플랫폼 원장 한 표를 직접 읽음 — API 전환 대상으로 집계됨
원본 내려받기	플랫폼 V5 REST
설비 찾기	시계열 인사이트 조회(선언된 같은 층 호출), 없으면 플랫폼 MCP
권한 판정	설비 트리(60초 캐시) + 서명된 사용자 맥락

상한(설정값)

질의	3~4,000자 · top_k 1~200
영상	2,048 MB · 120분 이하, 처리 창 기본 20:00–06:00
사라짐 보호	한 회차에 너무 많이 사라지면(비율 0.25 · 최소 5) 삭제 단계를 건너뜀
답 칩	영상 장면 칩 최대 3개

출처: rag-api MANUAL §2 · §4 · §5 · §5.1 · §6 ⑨ · ECOSYSTEM §11.1 규칙 2 · §11.2 O7

10 · 문서 이해 · 비전

읽는 모듈 둘 — 파싱은 정책으로, 판독은 합의로

둘 다 **무상태**입니다. 문서도 사진도 처리한 뒤 남기지 않습니다. 대신 «어떻게 읽을지» 와 «언제 못 읽었다고 할지» 를 이 모듈이 정합니다.

문서 이해

문서 변환 엔진 앞의 관문입니다. 파싱 정책을 **프로필 넷**(기본 — RAG 색인과 같은 파싱 · 빠르게 · OCR · 스캔)으로 소유해, RAG 와 비전이 같은 규칙으로 읽습니다. 영상 · 음성의 녹취와 도면 파싱 · 판 비교도 여기서 합니다.

`/api/v1/understand`	문서 → 구조화 텍스트(프로필 지정)
`/api/v1/transcribe`	녹취 제출 · 조회 · 취소 기본 꺼짐
`/api/v1/drawings/parse`	도면 파싱(DXF · 벡터 PDF)
`/api/v1/drawings/diff`	두 판의 차이 — 치수 · BOM · 주기
MCP	`document_understand` · `document_profiles` (읽기 전용)

녹취가 지어내지 않게

- 음성 구간 검출 → 침묵에서 자르기 → 인식 → **환각 거르기**(다섯 가지 이유)
- 잘못 버린 구간은 시각을 다시 맞춰 되살립니다
- 조각의 언어가 영상의 주 언어에서 벗어나면 다시 인식합니다
- 대량 · 대화형 두 줄을 갈라, 긴 영상이 질문을 막지 않습니다

악성 입력에는 크기 · 압축 해제 · 시간 · 동시성 상한이 있고, URL 은 내부 주소로 가지 못하게 막습니다. «성공했는데 내용이 없음» 은 성공이 아니라 `ok:false` 입니다.

비전

사진을 받아 비전-언어 모델에 묻고 판정을 돌려줍니다. 같은 질문을 **여러 번** 묻고 표를 모아, **갈리면 거부**합니다(`readable=false`).

`/api/v1/read_meter`	계기 판독 — 5번 중 4번 일치(설정값)
`/api/v1/inspect`	검사 — 결함 소견(최대 8개)
`/api/v1/ocr`	OCR — 줄 단위 합의, 부분 결과 허용
`/api/v1/classify` · `/detect`	분류 · 검출
`/api/v1/analyze`	자유 서술 — 값 · 판정은 내지 않음
MCP	`vision_*` 5종(검출은 REST 만)

«못 읽음» 도 답입니다

계기 판독은 먼저 «이것이 계기인가» 를 세 번 묻고 두 번 «예» 여야 읽습니다. 동률은 합의로 치지 않고, 모델 출력은 스키마로 검증합니다. 거부 이유는 코드로 갈라 냅니다 — 모델 거절 · 합의 실패 · 유효 판독 없음 · 모델 불통.

잰 것(검증 장비)	결과
계기 판독 기준선	맞음 4 · **거부 2 · 틀림 0**
명판 필드	10 / 10

출처: plantpulse-vision-api MANUAL §6.4.1(2026-10-04 실측)

> **틀리는 것보다 거부가 낫다**
> 현장 사진의 계기를 «그럴듯하게» 잘못 읽으면 그 숫자가 보고서에 들어갑니다. 거부는 사람이 한 번 더 보게 만듭니다.

출처: plantpulse-document-understanding-api MANUAL §1 · §4 · §5 · §10.1 · §10.3 · §10.5 · plantpulse-vision-api MANUAL §1 · §4 · §5

읽는 모듈이 다른 모듈과 이어지는 자리

입력	경로	결과가 쓰이는 곳
채팅에 올린 사진	앱 → 비전 판독(계기 · 명판)	답의 숫자 · 출처 카드
사진 속 글자	비전 OCR → 문서 이해로 위임(선언된 같은 층 호출)	OCR 결과 · 부분 합의
절차 영상 · 녹음	RAG → 문서 이해 녹취 → 구간 색인 기본 꺼짐	답의 장면 시각 칩
도면 두 판	RAG · 앱 → 문서 이해 파싱 · 판 비교	도면 카드 · 판 배지 · 디지털 스레드 참조
현장 음성 질문	앱 → 문서 이해 녹취 → 질문	묻기(장갑 낀 손 기준 큰 단추)

출처: ECOSYSTEM §11.1 규칙 2 · §12.2-1 · §12.3-4 · §12.4-5 · rag-api MANUAL §4 · §5

11 · 공학 계산 · 자동화 샌드박스

산술은 모델에 맡기지 않습니다

언어모델은 공식을 «고르는» 데는 쓸 만하지만 소수점 계산을 믿을 수는 없습니다. 그래서 계산은 **결정적인 서버**가 하고, 데이터를 직접 다루는 코드는 **격리된 샌드박스**에서만 돕니다.

공학 계산

모델은 공식과 입력만 고르고, 산술 · 단위 환산은 서버가 합니다. 결과에 **값 · 대입한 식 · 근거**를 함께 실어, 사람이 따라 계산해 볼 수 있습니다. DB · LLM · GPU 를 쓰지 않습니다.

`eng_list_formulas`	공식 목록 — 8개 분야 30종
`eng_formula`	공식 계산(값 · 식 · 근거)
`eng_unit_convert`	단위 환산
`eng_calc_eval`	허용 목록 수식 계산

수식은 «허용 목록» 으로만

숫자 · 사칙 · 거듭제곱 · 괄호 · 정해진 함수 · 상수만 받습니다. 속성 접근 · 첨자 · 문자열 · 비교 · 람다는 거부하고, 식 길이 · 노드 수 · 깊이 · 계산 시간에 상한(설정값)을 둡니다.

> **범위 밖은 «경고», 물리 한계 밖은 «불가능»**
>
> 통상 범위를 벗어난 입력은 거절하지 않고 경고를 붙이고, 물리적으로 불가능한 값은 `impossible` 로 냅니다. 범위 «안» 의 단위 실수(예: kN·m 를 N·m 칸에)는 잡지 못한다고 문서에 밝혀 둡니다.

출처: plantpulse-engcalc-api MANUAL §1 · §2 · §4 · §5.1

자동화 샌드박스

승인 · 고정된 자동화 계획의 파이썬 계산 단계를 받아 **실행마다 새로 격리**해 돌리고, 출력(JSON · 표) · 그림 · 결과 코드만 돌려줍니다. 기록에는 코드 · 입력 · 출력 «내용» 을 남기지 않습니다. 앱만 부르는 REST 모듈이고 MCP 표면은 없습니다.

`POST /api/v1/run`	실행 — 코드 · 입력 · 한도
`/api/v1/runs`	실행 기록(30일 보존, 설정값)
`/api/v1/runtime`	지금 걸린 한도를 그대로 보임

겹겹의 격리

컨테이너	사용자 공간 커널(gVisor) 위에서만 뜹니다. **폴백도 끄는 스위치도 없습니다** — 아니면 기동 실패
실행마다	새 네임스페이스(사용자 · 네트워크 · 프로세스 · 마운트), 새 임시 파일시스템, 권한 0, seccomp 로 실행 · 인터넷 소켓 · 마운트 차단
기동 자가시험	비특권 uid · 권한 0 · 네트워크는 루프백만 · 쓰기 금지 위치를 기동 때와 주기적으로 확인 — 어긋나면 중지
출시 게이트	**탈출 시도 27종** 묶음을 실제 격리 컨테이너에서 돌려 «실패 0» 이어야 출시합니다

출처: plantpulse-automation-sandbox-api MANUAL §1 · §2 · §4 · §5 · plantpulse-ai-docker MANUAL §5.11 · §6.1(`check-sandbox-escape`)

공학 계산 상한(설정값)

식 길이	1,000자
노드 / 깊이	400 / 40
계산 시간	0.5초
변수 / 함수 인자	64 / 32
유효숫자	10자리

출처: engcalc-api MANUAL §4 · §5.1

샌드박스 실행 한도(설정값)

시간	기본 10초 · 1~60초
메모리	기본 512 MB · 128~2,048 MB
코드 / 입력	256 KiB / 16 MiB
출력	표 10,000행 · 그림 8개(각 1 MiB)
동시 실행 / 대기열	2 / 16 — 대기 30초

출처: automation-sandbox-api MANUAL §4 · CONTRACT §5(`limits.py`)

쓰기는 여기서도 열리지 않습니다. 샌드박스의 입력은 자동화가 **읽기 도구로 얻은 데이터뿐**이고, 네트워크가 없어 밖으로 쓸 길도 없습니다. 현장 시스템을 바꾸는 일은 여전히 승인 카드를 거칩니다.

12 · 음성 · 파일 저장소 · 장기 기억

보조 모듈 셋 — 앱이 직접 부르고, 각자 자기 칸에만 씁니다

이 셋은 모델의 도구가 아니라 **앱의 부품**입니다. 그래서 MCP 를 내지 않고 REST 만 냅니다. 대신 개인정보 · 보존 · 삭제처럼 **운영자가 책임져야 하는 것**을 각자 소유합니다.

파일 저장소 기본 켬

원본 바이트의 단일 주인

사진 · 영상 · 문서 · 첨부의 원본을 형식 판별 · 해시 중복 제거해 보관합니다. 원본 버킷(버전 관리 · 객체 잠금)과 파생 버킷(썸네일 — 만료 규칙은 여기에만)을 가릅니다.

음성 출력 기본 꺼짐

화면 글을 «읽을 글» 로

숫자 · 단위 · 시간 · 코드를 읽는 말로 바꾸고(한 · 영 규칙 + 설치별 발음 사전) 문장마다 합성해 잇습니다. CPU 로 돌고, 문장 캐시와 동시 합성 상한을 둡니다.

장기 기억

대화를 넘는 맥락

대화에서 사실을 뽑아 저장하고 다음 대화에서 의미 검색으로 꺼냅니다. 범위(사용자 · 자동화 · 회차)를 하나 이상 반드시 지정해야 하고, 관리자도 남의 기억은 읽지 못합니다.

API

`/api/v1/uploads`	올리기(이어 올리기)
`/api/v1/objects`	읽기(Range) · 썸네일 · 검색
`/api/v1/admin/erase`	개인정보 삭제
`…/hold` · `/release`	법적 보존

API

`/api/v1/speech`	글 → 음성
`/api/v1/preview/normalize`	읽을 글 미리 보기
`/api/v1/settings`	설정 · 발음 사전

API

`/memories`	저장 · 목록 · 범위 삭제
`/search`	의미 검색
`/memories/{id}/history`	변경 이력

데이터 · 장치

- PG `pi.storage` + 원본 · 파생 버킷. 백업은 둘을 **같은 시점**으로
- 개인정보 삭제는 같은 바이트의 모든 행을 묘비로 바꾸고 **그 해시의 재저장을 막습니다**. 중간에 끊겨도 이어서 끝냅니다
- 스크립트를 품을 수 있는 형식(SVG · HTML)은 인라인으로 내지 않습니다
- 사용자의 설비 범위 밖은 404, 감사 기록은 지울 수 없습니다

데이터 · 장치

- PG `pi.voice` + 지워도 되는 문장 캐시
- 캐시 키에 정규화 판 · 모델 이름이 들어가, 규칙이 바뀌면 옛 소리가 나오지 않습니다
- 가중치는 해시로 고정하고, 라이선스 문구가 바뀌면 빌드가 멈춥니다
- 큐가 차면 429 — 조용히 늦어지지 않습니다

데이터 · 장치

- 자기 벡터 컬렉션 + 변경 이력. 범위 없는 목록 · 검색 · 삭제는 400
- 바깥 호출이 없고, 텔레메트리는 꺼짐으로 고정 — 켜면 기동을 거부합니다
- 사용자 단위 삭제를 API 로 제공합니다

판의 성숙도를 그대로 밝힙니다. 파일 저장소는 compose 에 기본으로 들어 있지만 문서가 스스로를 **MVP** 로 부르고, 음성 출력은 설치 기본값이 꺼진 **선택 설치**입니다(호스트 설정 한 줄로 켭니다). 둘 다 측정 수치는 아직 문서에 없어 싣지 않았습니다.

출처: plantpulse-storage-api MANUAL 머리말 · §1~§3 · §5 · §6 · §8 · plantpulse-voice-api MANUAL 머리말 · §1~§3 · §5 · §6 · §8 · plantpulse-mem0 MANUAL §1 · §3 · §5 · copilot MANUAL §5.33 · ai-docker compose(`profiles: [voice]`)

사람에 관한 데이터는 어디에 남나 — 운영자를 위한 지도

무엇	어디	지우는 길 · 남기지 않는 것
채팅 사진 · 첨부	파일 저장소(원본 버킷)	개인정보 삭제 API — 같은 바이트 전부, 재저장 차단
대화	앱 `pi.co`	대화 삭제(이력 원문 · 메시지 · 첨부까지)
기억	장기 기억	사용자가 «기억» 화면에서 고치기 · 지우기 · 일시중지, 사용자 단위 삭제 API
도구 호출	앱 감사 원장 · 게이트웨이 기록	게이트웨이는 인자 · 결과를 남기지 않음
자동화 계산	샌드박스 기록	코드 · 입력 · 출력 내용은 남기지 않음, 30일 보존

출처: storage-api MANUAL §6 · copilot docs/BENCHMARKS «벤치가 만든 것은 벤치가 지운다»(대화 삭제 API) · mem0 MANUAL §5 · copilot MANUAL §5.33(기억 화면) · gateway MANUAL §6.4 · sandbox MANUAL §1 · §4

13 · 추론 스택 · 관리 콘솔 · 입구 프록시

운영 구성 셋 — 모델 · 관리 · 입구

이 셋은 업무 기능이 아니라 **운영의 뼈대**입니다. 모델은 사내 GPU 에서 돌고, 관리자는 한 화면에서 서비스를 보고, 바깥에서 들어오는 길은 하나입니다.

추론 스택

사내 GPU 에서 **언어(채팅 · 비전 겸용) · 임베딩 · 리랭커** 세 모델을 서빙하고, 영상 수집을 켜면 **음성 인식**이 더해집니다. 그 앞의 추론 게이트웨이가 표준 채팅 완성 API 로 감싸고 가상 키 · 사용량을 관리합니다. 문서 변환 엔진도 같은 스택에 있습니다.

`/v1/chat/completions`	채팅 완성
`/v1/embeddings` · `/rerank`	임베딩 · 재정렬
`/v1/models`	서빙 이름 셋만 공개 — 별칭 없음

- 모델 서버의 포트는 호스트에 열지 않습니다. 앱은 게이트웨이를 거칩니다
- 기동 전에 **GPU 몫 예산**을 셉니다 — 합이 하드웨어 상한을 넘으면 띄우지 않습니다
- 모델 아카이브는 해시로 검증하고, 같은 이름을 덮지 않고 **새 이름으로만** 더합니다
- 대화 본문은 기본으로 추적 로그에 싣지 않습니다
- 헬스는 «연결» 이 아니라 **우리 키로 모델 목록을 읽어 서빙 이름이 보여야** healthy 입니다

출처: plantpulse-ai-inference-server MANUAL §1 · §2 · §5 · §6.1a · §6.2 · ai-docker MANUAL §6.1(healthcheck)

관리 콘솔

서비스 카탈로그 · 헬스 집계 · 서비스마다 같은 모양의 화면(MCP 정보 · API · 로그 · 설정 · 키 상태)을 냅니다. 뒤의 관리 API 가 `/api/v1/{service}/…` 로 각 모듈에 중계합니다(RAG · 인사이트 · 문서 이해 · 비전 · 스레드 · 게이트웨이 · 샌드박스 · 음성 · 파일).

`/api/v1/health`	카탈로그 헬스 집계
`/api/v1/llm/*` · `/gpu/*`	모델 · GPU 지표
`/api/v1/keys/probe`	키 게이트 확인(키 없이 401 · 키로 200)

- 헬스는 상태 코드가 아니라 **본문**을 읽고, 콘솔 키로 한 번 더 불러 **키가 거부되면 degraded** 로 보입니다
- 쓰기는 전부 감사 로그에 남고, 관리자 신원은 서버가 채웁니다
- 콘솔은 디지털 스레드 **행동 도구를 부르지 못하고**, 샌드박스 실행 · 파일 쓰기도 403 입니다
- 로그인 실패가 이어지면 잠급니다. 경로 우회(점 세그먼트 · 인코딩)는 거부합니다

출처: plantpulse-ai-admin-web MANUAL §1 · §5.1 · §5.2 · §5.4 · §5.7~§5.14 · admin-api 라우터

입구 프록시

TLS 를 끝내고 **포트와 경로로만** 나눕니다 — 관리 콘솔, 추론 API(허용 목록), MCP 경로들, 그 밖은 현장 판단 앱. 인증은 하지 않습니다. **각 뒷단이 키를 직접 확인**하므로, 프록시 설정 실수가 인증 구멍이 되지 않습니다.

- 설정을 이미지를 굽는 동안 검사합니다(`nginx -t`)
- 평문 HTTP 는 HTTPS 로 308, 보안 헤더 · 신뢰 프록시 목록
- 스트리밍 경로는 버퍼링을 끄고, 비root 로 돕니다

출처: plantpulse-ai-proxy MANUAL §1 · §2 · §4 · §5.3 · §7

14 · 신뢰 장치 ①

프롬프트에 적은 규칙은, 코드가 강제하지 않으면 지켜지지 않습니다

이 문장은 구호가 아니라 개발 기록입니다. 같은 모양의 사고가 **열두 번** 났고, 열두 번 다 코드로 막고 나서야 멈췄습니다. 그래서 «모델에게 부탁하는 규칙» 을 «출력의 입구에서 검사하는 코드» 로 옮겼습니다.

출처: copilot MANUAL §5.8 «프롬프트»

근거 가드 — 숫자는 이 턴의 도구 결과에서만

경우	가드가 하는 일
도구가 하나도 성공하지 않았는데 데이터를 말한다	한 번 다시 묻고, 그래도면 답을 «조회하지 못해 수치로 답할 수 없다» 로 바꿉니다
도구 결과에 없는 숫자가 있다	한 번 고쳐 쓰게 하고, 남으면 «확인되지 않은 숫자» 경고 줄을 붙입니다
이전 답을 근거로 삼는다	근거로 인정하지 않습니다 — 근거는 **이번 턴**의 도구 결과뿐입니다
일수 · 관계를 잘못 옮긴다	숫자 가드(일수 0.1일 허용) · 관계 가드(«A 가 B 를 공급» 주장을 스레드 결과와 대조)
차트의 숫자	모델은 차트 자리 표식만 쓰고, **서버가 도구 결과에서** 차트를 만듭니다

출처: v1-chat-on-behalf §9.2(답의 근거 가드) · copilot MANUAL §5.18(숫자 · 관계 가드) · §5.27 · ECOSYSTEM §12.4-2

출력 입구의 가드 — 모델 출력이 들어오는 자리마다

«데이터 없음» 을 밝혀라	폴백 0 이 실측으로 읽혀 «없는 위기» 가 되지 않게	**도구 오류 원문 숨김**	내부 주소 · 오류 HTML 이 답에 실리지 않게
지어내지 마라	없는 계측점의 역할이 진단 근거가 되지 않게	**프롬프트 유출 차단**	흐르는 도중에 끊습니다 — 사용자가 «본 뒤» 가 아니라
조치안 필수 칸	눌러 봐야 실패하는 버튼이 나가지 않게	**사고 과정 제거**	추론 표식이 화면에 남지 않게

가드는 호출자마다가 아니라 **입구 하나**에 둡니다 — 새 호출자가 생길 때 빠지지 않도록. 버린 것은 로그에 남기고, «지워야 할 것» 과 «두어야 할 것» 을 짝으로 시험해 **멀쩡한 답이 지워지는 오탐**도 막습니다.

«0» 과 «모름» 을 모든 모듈이 가릅니다

시계열 인사이트	«데이터 없음» 과 «실패» 를 다른 코드로 내고, MCP 오류에 «비어 있음이 아니라 쓸 수 없음» 을 실어 모델이 실패를 «데이터 없음» 으로 보고하지 않게 합니다. 측정이 7일보다 오래되면 정확도 대신 «측정 없음»
예측 원장	실측을 못 구하면 0 으로 채우지 않고 `unresolvable`
가설 검증	잴 수 없는 증상은 «지지» 도 «반박» 도 아닌 `unmeasurable`
디지털 스레드	«모름» 은 Go 가 아니다 — 준비도 판정에서 통과로 세지 않습니다
비전 · 문서 이해	합의 실패는 `readable=false`, 빈 성공은 `ok:false`

출처: timeseries-insight-api MANUAL §1 · §5 · §6 · INTEL «예측 원장» · «고장 가설 검증» · digital-thread-api MANUAL §2 · §6.14 · vision-api MANUAL §1 · document-understanding-api MANUAL §5

15 · 신뢰 장치 ②

범위는 닫힌 쪽으로, 쓰기는 승인 카드로, 모든 호출은 원장에

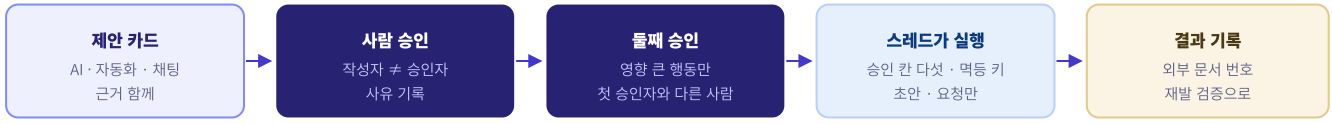
«무엇을 볼 수 있나» 와 «무엇을 바꿀 수 있나» 는 화면의 단추가 아니라 **서버의 판정**입니다. 단추를 숨기는 것은 보호가 아닙니다.

범위 — 판정할 수 없으면 «안 보임»

- 범위 밖 객체는 404 — 거부와 «없음» 이 같은 모양이라 **존재 여부도 새지 않습니다**
- 판정할 수 없는 결과는 닫습니다(`scope_unverifiable`). **빈 범위는 아무것도 보지 못합니다**
- 사용자 맥락은 앱이 **서명**해 뒷단에 싣고(수명 5분, 설정값), 뒷단은 서명을 검증해 같은 범위로 거릅니다
- 인바운드 키가 비어 있으면 **모든 모듈이 기동을 거부**합니다 — 무인증 스위치가 없습니다
- 권한 입력을 읽지 못하면 503 — «열어 둔 채» 넘어가지 않습니다 (RAG · 인사이트)
- 모르는 설정값은 기동 실패 또는 경고 + 기본값. 설정 오류를 폴백으로 삼키지 않습니다

출처: copilot MANUAL §5.2b · rag-api MANUAL §5.1 · timeseries-insight-api MANUAL §4 · §5 · gateway CONTRACT §2 · mem0 MANUAL §3 · ECOSYSTEM §11.4-e · §12.6(원칙)

쓰기 — 제안 카드에서 사람 승인까지



모델이 직접 실행하는 길은 없습니다 · 설비 제어는 범위 밖 · 확정은 고객 시스템 안에서 사람이

2인 승인 출시 2026.10	채팅의 행동 카드는 그 대화를 한 사람이 승인하지 못합니다. 영향 큰 행동(기본: 협력사 독촉 요청 초안 · 작업 지시 초안, 설정 `access.two_person.tools` 로 지정)은 첫 승인 뒤 **둘째 승인**을 기다립니다. 제안 카드도 같습니다. 공개 태그 2026.10(10-19)에 실렸습니다(`TwoPersonRule`)
외부 도구의 기본값	선언이 없는 외부 서버의 도구는 **쓰기 도구로 간주**합니다. 쓰기 도구 가드는 끌 수 없습니다
자동화	쓰기 도구는 자동화에서 실행되지 않습니다 — 자동화가 낼 수 있는 행동도 전부 사람 승인을 거칩니다
사람 표시 라벨	«수락» 은 기록일 뿐이라, 눌러도 오더 · 일정 · 설비는 아무것도 바뀌지 않습니다

출처: copilot MANUAL §5.2b «2인 승인»(`TwoPersonRule` — 공개 태그 2026.10) · §5.7 · §5.13 · §5.20 · §5.25 · §5.30 · v1-chat-on-behalf §0 · digital-thread-api MANUAL §5 · §6.4 · ECOSYSTEM §11.4-k

감사 — 남기는 것과 남기지 않는 것

어디서	남기는 것	일부러 남기지 않는 것
현장 판단 앱	모든 도구 호출(감사 원장, 만료 없음) · 설정 변경 · 문서 열람 · 승인자 · 승인 시각	—
MCP 게이트웨이	호출자 · 서버 · 도구 · 결과 코드 · 시간	인자 · 결과 · 인증 값
디지털 스레드	커넥터 행동 로그 · 바인딩 로그	페이로드(값은 원천에)
샌드박스	실행 감사 · 기록 행	코드 · 입력 · 출력 내용
관리 콘솔 · 파일	쓰기 · 키 열람 · 삭제(지울 수 없는 감사 기록)	—

출처: copilot MANUAL §5.2 · §5.2b · gateway MANUAL §6.4 · digital-thread-api MANUAL §5 · sandbox MANUAL §1 · admin-web MANUAL §5.7~§5.14

16 · 배치

온프레미스 — 모델도 데이터도 사내에 머뭅니다

스택 전체가 **compose 한 파일**로 고객 장비 한 대에 섭니다. 모델은 사내 GPU 에서 돌고, 바깥으로 나가는 호출이 없어 **폐쇄망**에서도 같은 구성으로 섭니다. 쿠버네티스용 차트는 그 compose 를 옮긴 것이고 정본은 compose 입니다.

설치 경로 둘

온라인	원라인 설치 — OS 판별 → 런타임 · 드라이버 · 호스트 튜닝 → 레지스트리 확인 → 설정 파일 준비. 기동 전에 플랫폼 연결을 한 번 시험하고, 실패하면 그 자리에서 멈춥니다
폐쇄망	서명된 **에어갭 번들**(이미지 · 설치기 · 오프라인 패키지 · 검증 도구)을 반입해 검증 → 적재 → 설치. 모델은 매체로 따로 반입합니다

호스트를 함부로 바꾸지 않습니다

우리 compose 밖의 컨테이너가 이미 돌고 있는 호스트에서는 기본이 **«계획만 출력»** 입니다 — 바뀔 것을 보여 주고 아무것도 쓰지 않습니다. 기존 설정 파일은 다시 만들지 않고 값도 덮지 않으며, 커널 파라미터는 현재값보다 낮추지 않습니다.

값은 한 곳에

설치마다 다른 값(하드웨어 모양 · 사이트 값 · 플랫폼 연결 · 비밀)은 호스트 설정 파일 하나에만 있고, 기동 전 검사기가 줄마다 판정합니다. 비밀은 전용 명령으로만 바꾸고, 플랫폼 계정 · 비밀번호는 설정 파일에 적지 않습니다(기동할 때 플랫폼에서 받습니다).

번들 검증 — 신뢰의 닻은 «번들 밖» 에

1. 번들 본문과 사이드카(체크섬 · 서명 · 공개키)를 받습니다
2. **공개키 지문**을 문서에서 받아 대조합니다 — 번들과 같은 곳에서 온 키는 닻이 될 수 없습니다
3. 서명 · 해시를 검증하고, **푼 트리 전체**를 서명된 파일 목록과 대조합니다
4. 이미지는 적재하는 바이트를 다시 해시합니다. 검증한 번들에 없는 파일은 적재하지 않고 이름을 경고합니다
5. 번들에 실린 서명 도구 자체는 번들 서명으로 보증되지 않으므로, **상류 체크섬**과 따로 대조하도록 안내합니다

> **검증 없이 넘어가는 길은 명시해야만 열립니다.** 번들을 못 찾거나 검증에 실패하면 반입이 멈춥니다. 건너뛰려면 «검증 안 함» 스위치를 사람이 직접 적어야 합니다.

이미지 공급망

레지스트리 이미지와 번들 blob 두 곳에 서명합니다. 외부 이미지는 굽는 시점의 다이제스트로 고정하고, 공개 이미지 허브를 직접 부르지 않고 사내 캐시를 거칩니다. 우리 이미지는 전부 비root 사용자로 돕니다.

폐쇄망 반입 — 번들에 든 것과 따로 가져오는 것

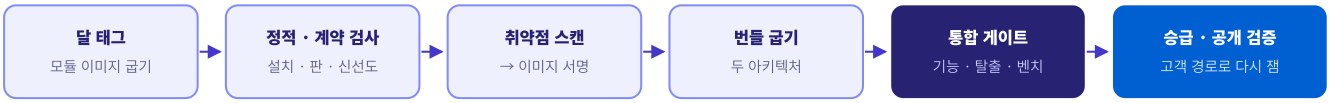
구분	내용
번들 안	모든 서비스 이미지 · 설치기 · compose(곧 이미지 핀) · Docker 오프라인 패키지 · 서명 검증 도구 · 샌드박스 런타임(판 · 해시 고정) · 소스 저장소 둘 · 파일 목록 체크섬
따로 반입	모델 가중치(인터넷 쪽에서 받아 매체로) — 시계열 모델은 이미지 안에 있음. 영상 수집을 켜면 음성 인식 모델도
기동 전	설정 파일의 자리표시자를 채우지 않으면 기동 명령이 거부합니다. 설치 검사기가 GPU 몫 · 플랫폼 연결 · 런타임 등록을 확인합니다

출처: plantpulse-ai-docker MANUAL §1 · §3(온라인 · 폐쇄망 설치) · §4(설정) · §5.8 · ECOSYSTEM 머리말(서명 자리 둘 · 비root · 프록시 캐시) · repos.tsv(Helm 차트 — 정본은 compose)

17 · 월 릴리즈 · 하드웨어

매월 자동 릴리즈 — 사람 없이 굽고 · 재고 · 서명해 내보냅니다

무인 사슬입니다 — **02:00** 제품 모듈마다 달 태그(`YYYY.MM`)를 붙이고, **04:00** 배포 저장소가 스스로 태그해 공개까지 잇습니다. 사슬 중 하나라도 «못 쟀으면» 실패로 봅니다 — **잴 수 없는 것을 출시하지 않습니다.**



통합 게이트(출시 전 유일한 «기능» 게이트)	스테이징 스택에서 ① 인증 · 라우팅 · MCP 초기화 · 도구 목록 ② **샌드박스 탈출 시도 27종** ③ 채팅 벤치 전체를 **엄격 모드**로 — 정상이 아닌 도구는 «왜» 가 기준 파일에 적혀야 통과하고, 표본이 없어 못 부른 것도 실패. 벤치 약 130분
«실패를 덮지 않는다»	이 게이트에는 실패 허용 표시를 붙이지 않습니다. 출시 직전에만 도는 검사를 덮으면 «아무도 안 재는 상태» 가 초록으로 보이기 때문입니다
되돌리기	직전 공개판 또는 특정 스냅샷으로 승급을 되돌리는 절차가 있습니다

출처: plantpulse-ai-docker MANUAL §6.1(무인 월 릴리즈 사슬 — 02:00 서비스 태깅 → 04:00 자기 태깅 · `verify:integration`) · §6.3(되돌리기)

하드웨어 등급

등급	GPU	CPU	메모리	디스크	상태
최소	GB10 계열(통합 메모리 128 GB)	20 코어	통합 128 GB	NVMe 1 TB↑	**실측한 모양**
단일 카드	NVIDIA RTX PRO 6000 Blackwell 96 GB × 1	32 코어↑	64–128 GB	NVMe 2 TB	프로파일 있음 — **그 카드에서 실측하지 않음**
권장	같은 카드 × 2	32~64 코어	128 GB	NVMe 2 TB↑	준비 중 2장 프로파일 작성 중

≈ 77 GiB
GPU 프로세스 합 — 언어 · 임베딩 · 리랭커 · 음성 인식 · 시계열 서버

113 / 119 GB
호스트 메모리 사용(GPU 와 나눠 씀)

30
돌고 있는 컨테이너

798 / 916 GB
루트 디스크 사용 — 모델 · 이미지 두 판 · 볼륨

출처: plantpulse-ai-docker MANUAL §3 «하드웨어 사양» — 검증 박스(GB10, 통합 메모리 128 GB) 2026-11-01 실측. 단일 카드 등급의 GPU 몫은 GB10 값을 96 GB 로 옮긴 것이며 그 카드에서 재지 않았습니다(같은 절). 카드 등급의 시스템 메모리 권장(1장 64–128 GB · 2장 128 GB)은 검증 박스에서 컨테이너가 쓰는 메모리 합 **약 33 GiB** 를 근거로 합니다.

GPU 몫은 기동 전에 셉니다. 하드웨어 프로파일이 모델 서버 넷과 시계열 서버의 몫을 정하고, 합이 상한(GB10 0.75 · 단일 카드 0.95, 설정값)을 넘으면 띄우지 않습니다. 꺼져 있는 음성 인식의 몫도 합에 미리 넣어, 나중에 켜도 넘지 않게 했습니다. 디스크는 모델 · 갱신 중 이미지 두 판 · 백업 때문에 **2 TB 를 권합니다.**

출처: ai-docker MANUAL §3 · §4 · §5.10

18 · 평가 ①

사람이 채점하지 않습니다 — 같은 입력이면 누가 돌려도 같은 점수

채팅이 «제대로 답하는가» 를 재는 문항 모음이 있고, **판정은 전부 기계 채점**입니다. 모델 답은 같은 질문에도 흔들리므로 한 번의 점수로 결론 내지 않습니다 — 최소 3회, 일부는 5회 중앙값입니다.

벤치	문항	무엇을 재나	축
툴콜(KTB)	**25**	질문에 맞는 도구를 부르는가 — 기대 도구 + 인자. 기준선 **25 중 18**	**게이트**
온톨로지(KOB)	16	설비 계층 · 태그 · 관계 질문에 맞는 도구를 고르고, 쓰면 안 되는 도구를 피하는가 — 5회 중앙값	기록
what-if	8	지연 가정 질문에 회복 계산 도구를 부르고 **그 숫자를 답에 옮기는가** — 계산기 숫자와 대조	기록
KTB-ext	116	배포본 도구 «전부» 에서 계열별로 맞는 도구를 고르는가(측정만)	기록
답변 정확도	7	도구 결과의 숫자 · 이름을 답에 그대로 옮기는가 — 그 순간 도구에서 뽑은 정답과 대조	기록
데일리(근거 충실성)	8	주어진 맥락만으로 쓰고 지어내지 않는가 — 데이터 없음 표기 · 인용	기록
현장 태블릿 골든 질문	17	공개 채팅 v1 경로로 — 배포마다 1회, 야간 3회. 결함 축 실패 0	기록
바닥 점검 · 가중치 지문 · 채점자 시험	—	배포본 프롬프트가 저장소와 같은가 · 의존 서비스가 살아 있는가 · 회차 중 모델이 바뀌지 않았는가 · **재는 자가 옳은가**	**게이트**

출처: copilot docs/BENCHMARKS «한 번에 돌리기» 표 · ①~④ · MANUAL 벤치 절(KTB-ext 116문항)

왜 KTB 만 게이트인가

도구 고르기는 **앱의 프롬프트 · 도구 설명 · 검색**이 바꾸는 축이라, 코드로 고칠 수 있고 따라서 출시를 막을 수 있습니다. 나머지 모델 품질 축은 같은 질문에도 흔들리고 모델을 바꾸지 않고는 고칠 수 없어, 미달을 **경보로 남기되** 승급은 막지 않습니다.

문항의 바탕은 데모 공장입니다

문항은 실제 고객 현장이 아니라 **중공업 데모 공장**(가스터빈 — 공장 동 9 · 라인 16 · 설비 53(가상 전력계 1 포함) · 태그 801)과 데모 외부 시스템에 붙은 객체로 묻습니다. 움직이는 값(알람 시각 등)은 답이 아니라 **도구 · 인자**로 채점합니다.

출처: copilot docs/BENCHMARKS «한 번에 돌리기»(축 배정 — 2026-10-18 결정) · MANUAL «채팅 시험의 바탕 — 중공업 데모»

벤치가 만든 것은 벤치가 지웁니다

벤치는 질문마다 새 대화를 만들고, 끝난 답은 장기 기억에도 쌓입니다. 그대로 두면 **벤치의 기억이 다음 질문의 답에 다시 꺼내져** 측정을 오염시킵니다. 그래서 만든 대화 · 기억을 장부에 적고 끝에 지운 뒤, 지워졌는지 **다시 잽니다**. 기억 저장소가 응답하지 않으면 «0» 이 아니라 «못 잼» 으로 남깁니다.

출처: copilot docs/BENCHMARKS «벤치가 만든 것은 벤치가 지운다»

재생은 운영과 같은 함수로. 평가 · 재생은 운영 코드를 그대로 부르고, 재생용 사본 구현을 따로 두지 않습니다. 결과에는 점수기 · 모델 · 규칙의 판을 싣고, 관문은 **결과를 보기 전에** 파일로 등록합니다 — 결과를 보고 관문을 고치지 않습니다.

출처: ECOSYSTEM §11.4-f · §11.4-j · §11.3(자율 관문 — 사전 등록)

19 · 평가 ②

잰 것 — 그리고 그 숫자가 «무엇을 뜻하지 않는지»

아래는 각 모듈 문서가 **실측**으로 적은 값입니다. 조건(문항 · 날짜 · 데이터)을 함께 적었고, 문서가 스스로 붙인 단서는 그대로 옮겼습니다.

도구 수와 프롬프트 길이 — 작은 것이 이긴다

조건	툴콜 점수
시스템 프롬프트 5,488자	11 / 22
시스템 프롬프트 1,189자	**18.5 / 22**

긴 시스템 프롬프트와 많은 도구가 작은 모델의 툴콜을 떨어뜨립니다. 그래서 커넥터가 늘어도 **채팅이 보는 도구 수는 고정**하고(스레드 도구 몇 개), 커넥터를 켤 때마다 KTB · KOB 를 다시 잽니다. 도구 고르기는 관문의 검색이 맡습니다(6장 — 도구 76종에서 재현율@20 0.972).

출처: ECOSYSTEM §11.4-m(채팅 도구 수 · 프롬프트 길이 실측) · gateway MANUAL §5.2(2026-10-18)

분석 모듈

무엇	결과	조건 · 단서
작업지시 완료 예측	계획대로 끝난 작업지시에서 평균 절대오차(목표 대비) **3.9%** vs 현재 속도 외삽 6.9%(480분, 12건) · **3.6%** vs 7.2%(20분, 39건)	시뮬레이션 현장 104건 추출(2026-09-27). 문서 단서: 설계를 고른 **같은 표본**이고, 일찍 닫힌 작업지시에서는 외삽을 이기지 못함
고장 유형 진단(합성)	1순위 적중 0.844 · 표시 정밀도 0.984 · 거짓 표시 0.003	합성 평가 — 문서가 **현장 정확도가 아니라 상한**이라고 밝힘
계기 판독	맞음 4 · 거부 2 · **틀림 0**	2026-10-04 기준선. 경계 표본 12회 반복에서는 틀림이 나왔음(문서 §6.4.1)
명판 필드 판독	10 / 10	같은 절. 문서 변환 엔진만 쓰면 7 / 10

출처: timeseries-insight-api docs/ORDER-COMPLETION-EVAL «결과» · «기본 모델» · «한계» · INTEL «합성 평가» · vision-api MANUAL §6.4.1

이상 탐지 점수기는 관문을 통과한 것만 현장에 오릅니다. 시계열 이상 점수는 사전 등록한 관문(정밀도 · 거짓 경보 · 음성 대조군)으로 공개 산업 데이터와 시뮬레이션 데이터에서 재고, 통과하지 못한 점수기는 **그림자**에 머뭅니다. 정확도 화면은 결과가 없거나 7일보다 오래되면 값 대신 «측정 없음» 을 보입니다. 재측정(재구성 오차 단독 AUC 기록)은 로드맵에 있습니다.

출처: timeseries-insight-api MANUAL §6 · docs/ANOMALY-EVAL · ECOSYSTEM §12.1-5

다음 판에 들어가는 것

무엇	상태
공개 채팅 v1 · 대리 질문	구현 · 스테이징 반영. 다음 재릴리즈에 실림 음성 질문 · 음성 답은 스테이징 전
콘솔 표면 · 승인 카드(행동)	공개 채팅 v1 과 **같은 다음 재릴리즈**에 함께 실림 예정
플랫폼 데이터 API 전환	표마다 그림자 비교 중. 대량 센서 시계열은 플랫폼 대량 API 명세를 기다림
권한 — 역할 · 범위 · 등급	단계 A · B(2026.11) · C(2026.12) · D(2027 1분기 — 외부 인증 · 임시 권한) 예정. 2인 승인은 이 단계와 별개로 **이미 출시**(2026.10, 15장)
현장 실증 평가 세트	고객 현장의 알람 · 정비 기록 · 게이지 사진 · 절차 영상으로 — 현장 확보 대기
절차 수행 벤치 · 외부 기준 벤치	여러 도구를 거쳐 판정까지 가는 벤치, 공개 산업 벤치 시나리오를 우리 도구에 잇는 어댑터 — 대기
상용 커넥터 실접속	모의 서버 적합성까지 완료 — 실제 제품 접속은 파일럿 현장에서

출처: v1-chat-on-behalf 머리말 · docs/platform-data-via-api 머리말 · ECOSYSTEM §12.1-2 · 4 · 8 · 9 · §12.6 · plantpulse-dt-connectors FEATURES

PlantPulse® **AI Platform**

TECHNICAL WHITEPAPER

모듈은 각자 한 가지를 맡고, 결정은 사람이 합니다.

이 문서의 모든 주장은 모듈 저장소의 문서와 코드에서 왔습니다. 설치 전 검토에서 더 깊이 보고 싶은 모듈이 있으면 그 모듈의 API 명세와 계약 시험을 함께 드립니다. 데모 공장 위에서 같은 벤치를 직접 돌려 보실 수 있습니다.

구조

모듈마다 주인 하나

저장소 하나 · 이미지 하나 · 계약 하나. 남의 데이터는 주인이 낸 API 로 읽습니다.

관문

도구는 게이트웨이 하나

권한 · 범위 · 검색 · 기록이 한 자리에서 판정되고, 외부 도구는 꺼진 채로 들어옵니다.

행동

쓰기는 승인 카드로만

고객 시스템에는 초안 · 요청까지, 영향 큰 행동은 2인 승인, 설비 제어는 범위 밖입니다.

근거

숫자는 도구 결과에서

근거 가드가 출력 입구에서 검사하고, «모름» 을 «0» 으로 보고하지 않습니다.

배치

사내 · 폐쇄망 · 서명

모델도 데이터도 고객 장비에, 번들은 서명 · 해시 · 파일 목록으로 검증합니다.

평가

기계 채점 · 출시 게이트

툴콜 벤치와 탈출 시험을 통과해야 매월 자동 릴리즈가 나갑니다.

문서

docs.plantpulse.io

/ko/ai — AI 플랫폼 매뉴얼

문의 · 데모

kopens.com/ko/contact

1588-5268 · webmaster@kopens.com

제품

plantpulse.io

Platform · Edge · AI · Studio

Rev. 2026.11 · 초안

㈜한국오픈솔루션 · KOPENS, Inc. © 2026 KOPENS. 플랜트펄스®는 ㈜한국오픈솔루션의 등록상표입니다.

kopens®