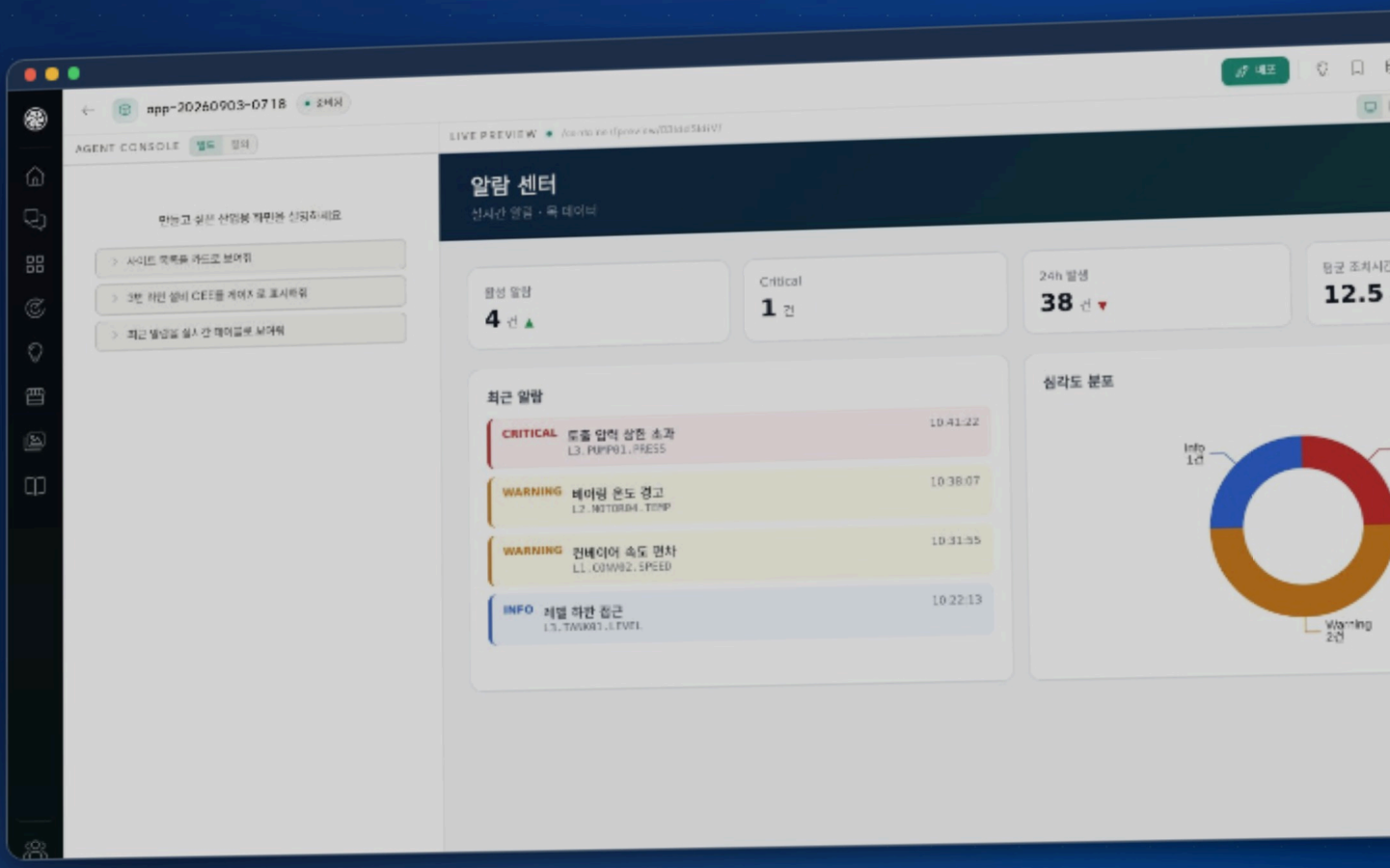
PlantPulse® **Studio**

INDUSTRIAL APP BUILDER

제품 소개 및 도입 제안서

현장이 원하는 화면을, 현장이 직접 만듭니다.

채팅 한 문장으로 산업용 대시보드 · 알람 · 분석 화면을 만들고, 공장의 실데이터에 연결해 즉시 배포하는 **산업용 AI 앱 빌더**입니다.



한 문장
자연어 → 실제 앱

6종
기본 제공 템플릿

3모드
플랜 · 자동 · 수락

2026년 9월

산업용 AI 앱 빌더 · 온프레미스 / 폐쇄망

kopens®

CONTENTS

목차

01	왜 지금인가 — 현장의 문제	3
02	PlantPulse Studio 란	4
03	무엇을 만들 수 있나	5
04	무엇이 만들어지는가 (실제 생성 앱)	6–7
05	특장점 ①②③	8–10
06	특장점 ④⑤⑥	11–12
07	아키텍처 — 토폴로지	13
08	기술 스택	14
09	빌더 엔진 · 데이터 이해 계층	15
10	사용 흐름 (실제 화면)	16–18
11	보안 · 거버넌스	19
12	배포 · 납품 모델	20
13	레퍼런스 · 검증 체계	21
14	도입 절차 및 일정	22

한 장 요약

> 공장에서 “화면 하나 더”가 필요할 때, 그동안은 **요청서 → 외주 → 몇 주**였습니다. PlantPulse Studio 는 그 자리를 **한 문장 → 몇 분**으로 바꿉니다.

한 문장
자연어로 설명하면
실제 React 앱이 만들어집니다

실데이터
플랜트 API 에 자동 연결
이름 → 태그 자동 해석

폐쇄망
온프레미스 · 에어갭 설치
서명 검증된 이미지 번들

핵심 차이
드래그&드롭 대시보드 툴이 아닙니다. AI 에이전트가 실제 소스코드(React·TypeScript)를 작성·수정하고, 타입 검사와 화면 검증을 스스로 돌려 고칩니다. 결과물은 **고객이 소유하는 소스**이며 Git 으로 인도됩니다 — 도구를 떠나도 자산이 남습니다.

이 문서가 다루는 것

제품이 무엇이고 무엇이 다른가
3–12쪽. 현장의 문제, 제품 정의, 실제 생성 앱, 여섯 가지 특장점.

어떻게 만들어졌는가
13–15쪽. 토폴로지, 기술 스택, 빌더 엔진과 데이터 이해 계층.

실제로 어떻게 쓰는가
16–18쪽. 구동 중인 인스턴스에서 캡처한 다섯 단계.

도입하면 어떻게 되는가
19–22쪽. 보안, 납품 방식, 레퍼런스, 도입 일정.

본 문서는 **제품 및 기술 검증 자료**입니다. 산업별 구축 사례와 정량 효과는 담당 영업 채널을 통해 별도 자료로 제공합니다.

01 · 배경

왜 지금인가 — 현장의 문제

설비 데이터는 이미 쌓이고 있습니다. 문제는 **데이터가 아니라 화면**입니다. 현장이 지금 필요로 하는 한 장의 화면이, 늘 몇 주 뒤에 도착합니다.

문제 1

요청서에서 화면까지 몇 주

"3번 라인 압력만 따로 보고 싶다"는 한 줄이 기획서 → 견적 → 개발 → 검수를 거칩니다. 그사이 현장의 관심사는 이미 다른 설비로 옮겨가 있습니다.

문제 2

전달 과정에서 요구가 왜곡된다

현장을 아는 사람은 코드를 못 쓰고, 코드를 쓰는 사람은 현장을 모릅니다. 그 간극을 문서로 메우는 동안 "원했던 것"과 "받은 것"이 벌어집니다.

문제 3

작은 수정도 외주를 거친다

색 하나, 임계치 하나 바꾸는 데도 담당자와 일정이 필요합니다. 결국 화면은 만들어진 그대로 굳고, 아무도 고치지 않게 됩니다.

문제 4

기성 대시보드 툴의 한계

정해진 위젯 조합 밖으로 나가지 못합니다. 공정 고유의 계산·화면·상호작용이 필요한 순간 다시 개발 프로젝트가 됩니다.

그래서 필요한 것

> 현장 담당자가 **자기 말로 설명하면**, 그 자리에서 **실제로 동작하는 화면**이 나오고, 마음에 안 들면 **그 자리에서 고쳐지는** 도구.

다만 산업 현장이라는 조건이 붙습니다. 클라우드 SaaS 로는 들어갈 수 없는 폐쇄망, 기존 플랜트 시스템의 인증·데이터 체계, 그리고 "AI가 만든 것을 믿을 수 있는가"라는 검증 문제까지 함께 풀어야 합니다. PlantPulse Studio 는 이 세 가지를 전제로 설계됐습니다.

02 · 소개

PlantPulse Studio 란

채팅으로 산업용 앱을 만들고, 배포하고, 운영하는 제품입니다. 사용자가 만들고 싶은 화면을 한 문장으로 설명하면 AI 에이전트가 실제 프로젝트 소스를 작성하고, 컨테이너에서 즉시 실행해 보여주고, 승인하면 그대로 배포합니다.



그림 1. 로그인 화면 — 고객사 로고·제품명으로 교체 가능한 화이트라벨 구조입니다. (실제 구동 인스턴스 캡처)

입력
한 문장
"3번 라인 실시간 온도·압력 대시보드를 만들어줘. 임계치 넘으면 빨갛게."

처리
에이전트가 코드를 쓴다
React·TypeScript 프로젝트를 실제로 편집하고, 타입 검사와 화면 검증으로 스스로 고칩니다.

출력
동작하는 앱
라이브 프리뷰로 즉시 확인하고, 버튼 한 번으로 버전을 남기며 배포합니다.

기존 방식과의 차이

	기성 대시보드 툴	PlantPulse Studio
만드는 방식	위젯 배치 · 설정	자연어 → 소스코드 생성
표현의 한계	제공된 위젯 조합 내	코드로 가능한 범위 전부
결과물 소유	툴 안의 설정값	고객 소유 소스 (Git 인도)
수정	설정 변경 또는 외주	대화로 요청 → 즉시 반영
데이터 연결	수동 매핑	이름 → 태그 자동 해석

03 · 활용

무엇을 만들 수 있나

빈 화면에서 시작할 수도, 현장에서 자주 쓰는 형태를 템플릿으로 골라 시작할 수도 있습니다. 어느 쪽이든 만들어지는 것은 **수정 가능한 실제 앱**입니다.

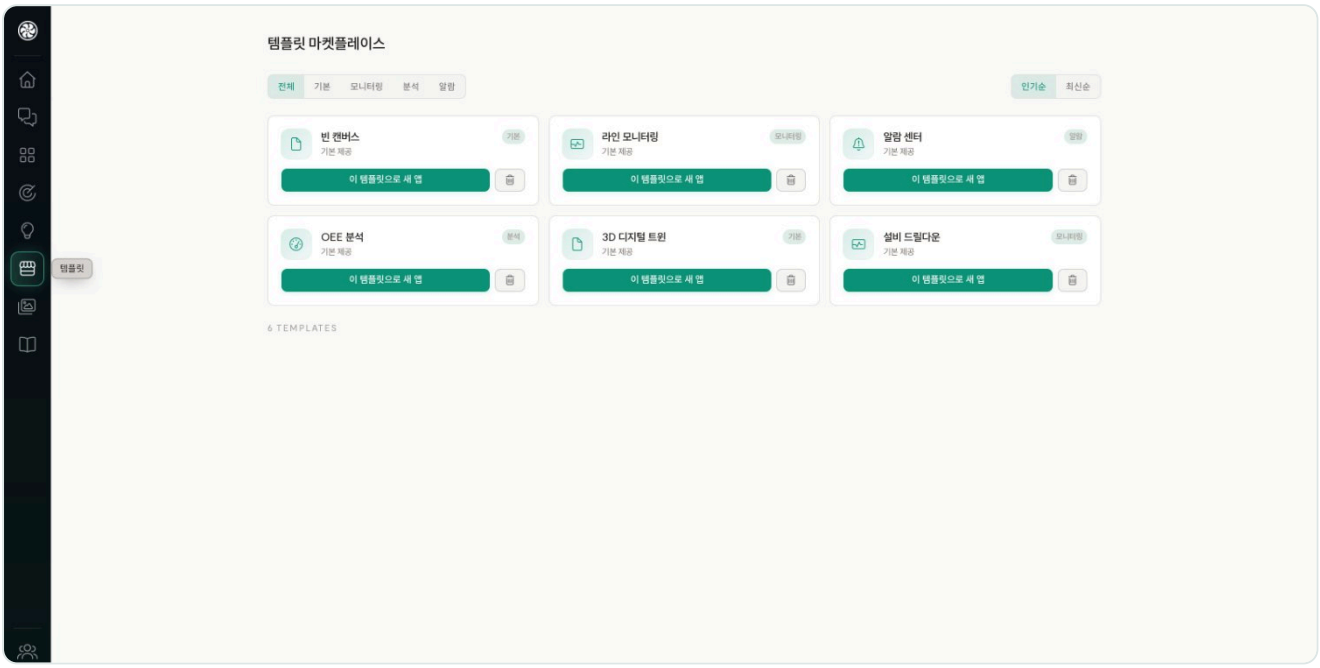


그림 2. 템플릿 마켓플레이스 — 라인 모니터링, 알람 센터, OEE 분석, 3D 디지털 트윈, 설비 드릴다운 등이 기본 제공되며, 사내에서 만든 화면을 사내 템플릿으로 추가할 수 있습니다.

대표 활용 시나리오

실시간 라인 모니터링

라인·설비별 현재 값과 임계 상태를 한 화면에. 교대조 인수인계 화면으로 그대로 사용.

알람 센터

심각도별 필터, 최근 알람 목록, 발생 추이. 조치 시간 KPI 를 함께 표시.

설비 OEE · 생산성 분석

가동률·성능·양품률 조합 지표를 설비 단위로. 기간 비교와 저조 설비 식별.

설비 드릴다운

자산 트리를 따라 공장 → 라인 → 설비 → 태그로 파고드는 탐색 화면.

품질 · 공정 리포트

일·주·월 단위 집계 화면. 필요하면 인쇄용 레이아웃까지 요청 가능.

현장 전용 소도구

점검 체크리스트, 교대 로그, 간이 계산기 — 외주 넣기엔 작은 것들.

> **"우리 공정에만 있는 화면"이 가장 잘 맞습니다.** 기성 툴이 커버하지 못하는 고유 계산식·고유 레이아웃일수록 이 방식의 이점이 커집니다.

03 · 활용

무엇이 만들어지는가 ① — 대시보드형

아래는 모두 **실제로 생성해 캡처한 앱**입니다 — 목업이 아닙니다.

실시간 모니터링형

KPI 타일 + 추이 차트 + 상태 목록. 교대 인수인계 화면으로 가장 많이 쓰이는 형태입니다.

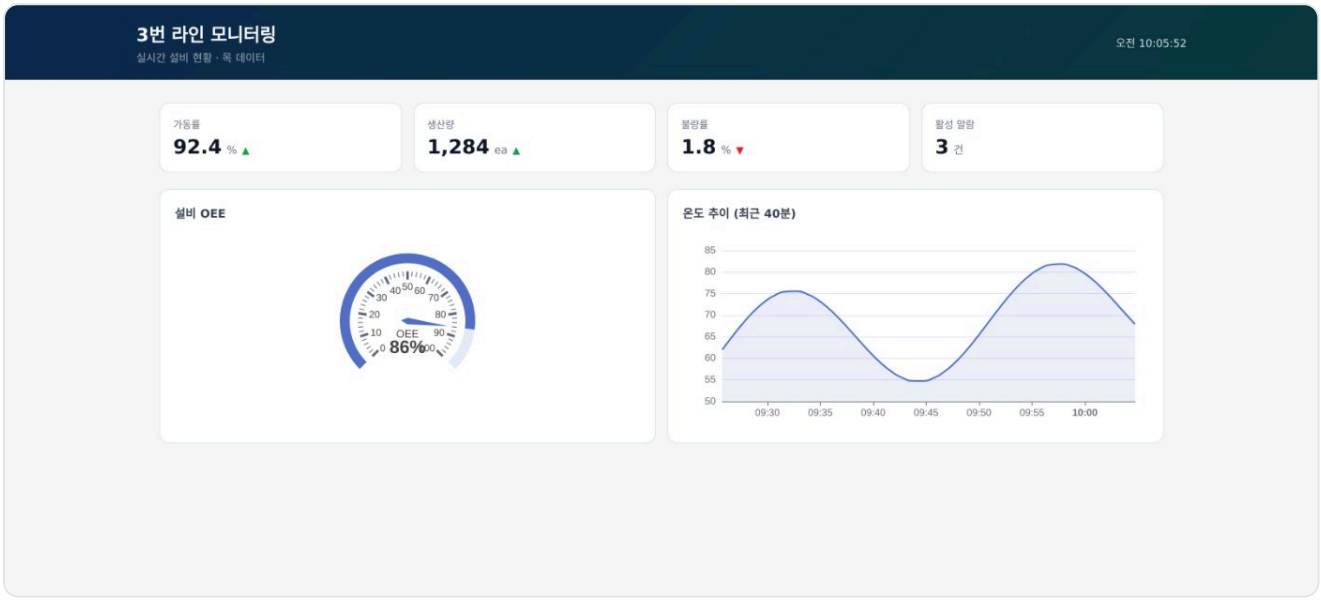


그림 3. 「3번 라인 모니터링」 — 가동률 · 생산량 · 불량률 · 활성 알람을 상단 KPI 로, 설비 OEE 와 온도 추이를 아래 차트로. *데모 데이터 기준입니다.*

분석형

지표를 분해해 보여주는 형태. 구성 요소별 기여도와 기간 추이를 함께 봅니다.

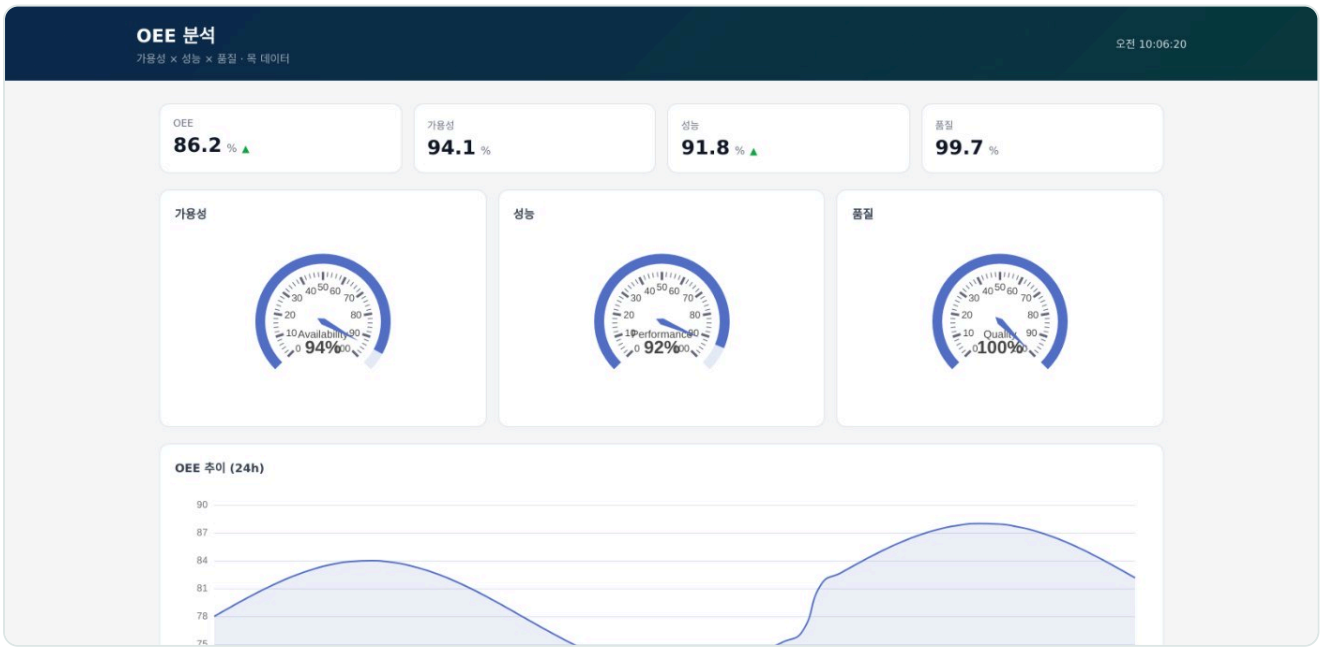


그림 4. 「OEE 분석」 — 종합 OEE 와 가용성 · 성능 · 품질 분해, 24시간 추이. *데모 데이터 기준입니다.*

03 · 활용

무엇이 만들어지는가 ② — 트윈 · 탐색형

디지털 트윈형

라인 배치를 3D 로 그리고 설비 상태를 색으로 표현합니다. 드래그로 회전하고 휠로 확대하며, 설비를 클릭하면 상세로 들어갑니다.

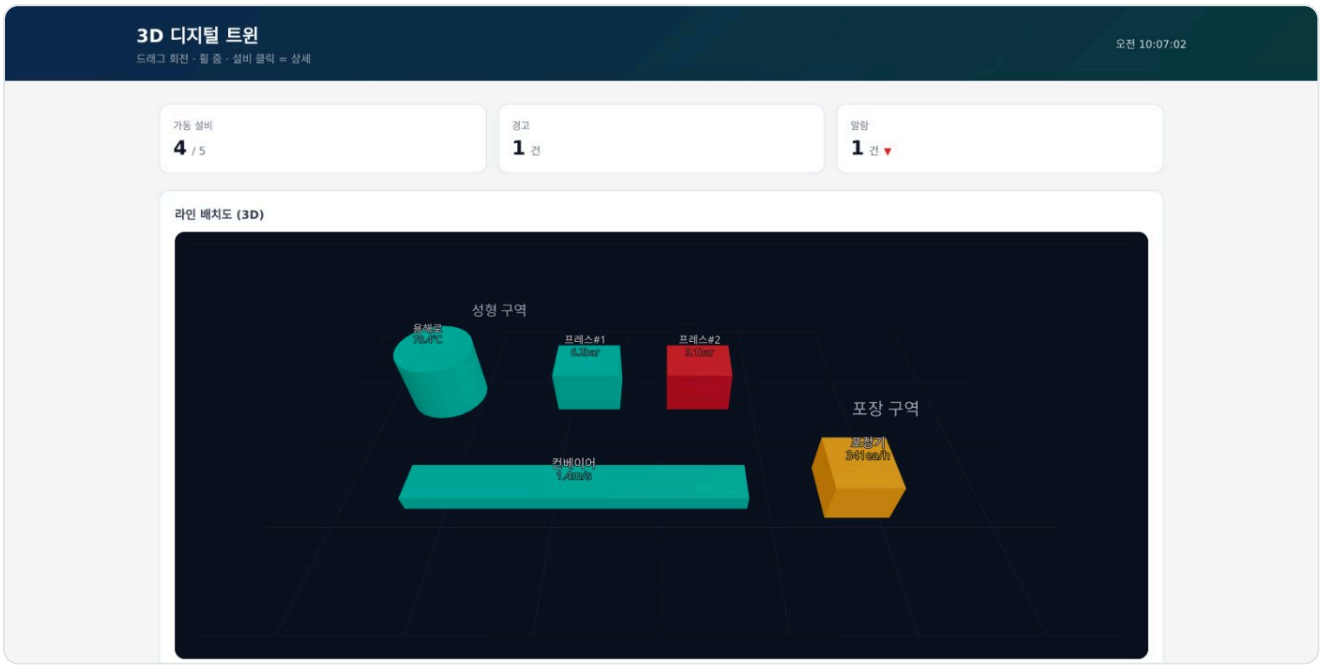


그림 5. 「3D 디지털 트윈」 — 성형·포장 구역의 설비를 3D 배치도로 표현하고 현재 값(온도 · 압력 · 속도 · 생산량)을 각 설비 위에 표시. 이상 설비는 색으로 구분됩니다.

탐색 · 표 형태

자산 트리를 따라 파고들며 태그 값을 표로 확인하는 형태. 목록 · 검색 · 상세로 이어지는 업무 화면의 기본 골격입니다.



그림 6. 「설비 드릴다운」 — 좌측 자산 트리에서 설비를 고르면 상태 · 가동률 · 활성 알람과 태그 목록이 표로 나옵니다.

> **형태가 정해져 있지 않습니다.** 위 넷은 자주 쓰이는 골격일 뿐이고, 실제로는 **말한 대로** 만들어집니다 — 등록·수정·삭제가 있는 관리 화면, 인쇄용 리포트, 키오스크 전용 전광판 형태도 같은 방식으로 나옵니다.

04 · 특장점

① 화면을 “조립”하지 않고 “작성”합니다

PlantPulse Studio 의 에이전트는 위젯을 배치하는 것이 아니라 **실제 프로젝트의 소스 파일을 읽고 쓰고 실행**합니다. 그래서 표현의 상한이 툴이 아니라 코드입니다.

한 턴 안에서 일어나는 일

1. **워크스페이스 준비** — 요청마다 스타터 템플릿을 복사한 독립 프로젝트가 만들어지고, `git init` 됩니다.
2. **도구 사용** — 에이전트가 `read_file` · `write_file` · `list_dir` · `run_command` · `platform_get` · `semantic_search` 를 직접 호출합니다.
3. **즉시 실행** — 세션 전용 컨테이너에서 개발 서버가 떠 있어, 파일이 바뀌는 즉시 라이브 프리뷰에 반영됩니다.
4. **자동 커밋** — 턴이 끝날 때마다 커밋이 남습니다. 언제든 이력을 보고 되돌릴 수 있습니다.

세 가지 작업 모드

플랜

에이전트가 먼저 계획을 제시하고, 승인해야 코드를 건드립니다. 큰 변경에 적합합니다.

자동

계획 없이 바로 실행합니다. 반복 수정이나 작은 조정에 적합합니다.

수정수락

변경 내용을 먼저 검토하고 커밋할지 되돌릴지 고릅니다. 신중해야 할 때 씁니다.

결과물은 고객의 자산입니다

생성된 앱은 툴 내부의 설정값이 아니라 **온전한 소스 프로젝트**입니다. 사내 GitLab/GitHub 로 푸시할 수 있고(앱별 저장소 자동 생성), 이후 사내 개발팀이 이어받아 손대도 됩니다. **도구를 걷어내도 자산이 남습니다** — 이 점이 SaaS 대시보드와 결정적으로 다릅니다.

부가 기능

앱 포크(기존 앱 복제 후 변형) · 앱 문서 자동 생성 · 버전 보관과 롤백 · QR/임베드 공유 · 접근 권한(관리자/빌더/뷰어) · 갤러리 공개.

04 · 특장점

② “3번 라인”이라고 말하면 알아듣습니다

산업용 화면의 진짜 난관은 그리는 것이 아니라 **연결하는 것**입니다. 수만 개의 태그 중 어느 것이 “3번 라인 토출 압력”인지 사람이 찾아 매핑하는 일이 대부분의 시간을 잡아먹습니다.

데이터 이해 계층 (Semantic Layer)

PlantPulse Studio 는 플랜트의 자산 모델·태그·별칭·단위·계층 구조를 정규화해 캐시하고, 자연어에 등장한 **이름을 실제 식별자로 해석**합니다.

사용자가 말한 것	시스템이 해석한 것
“3번 라인”	해당 라인의 자산 ID 집합
“토출 압력”	설비별 압력 태그 ID
“어제부터 지금까지”	설정된 표준 시간대 기준 절대 시간 범위

> **이 해석은 AI 가 추측하지 않습니다.** 이름 → 식별자 변환은 규칙 기반으로 **결정적으로** 수행됩니다. 같은 입력이면 언제나 같은 결과가 나오고, 매핑이 조용히 흔들리지 않습니다.

플랫폼과의 역할 분담

역할	담당
실데이터 · 검색 · 관계 탐색	기존 PlantPulse 플랫폼 (전수 DB 보유)
이름 해석 · 화면 생성	PlantPulse Studio
인증 · 권한	플랫폼 위임 또는 스튜디오 자체 계정(현장용)

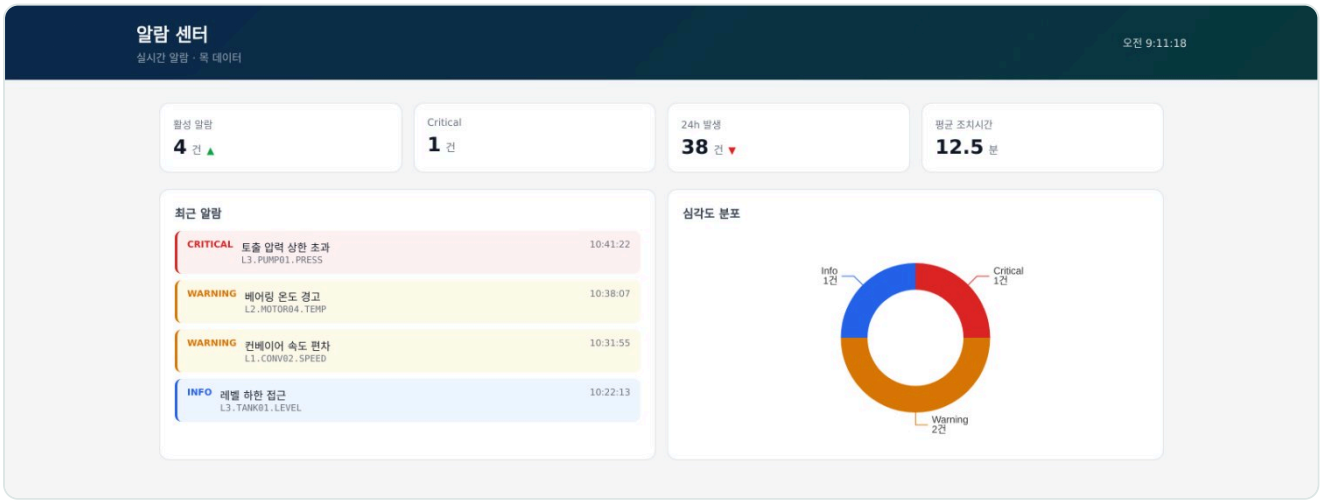
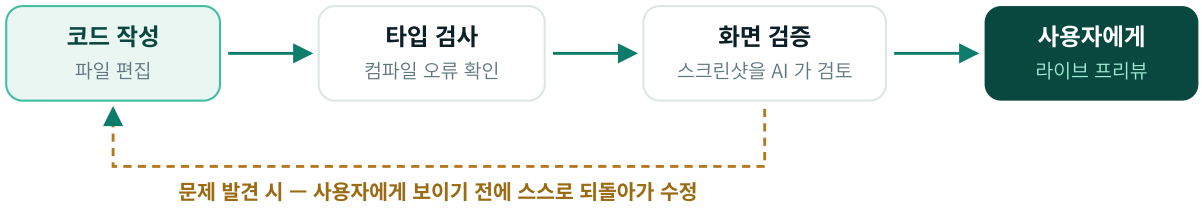


그림 3. 대화 한 번으로 생성된 알람 센터 — 심각도별 목록·분포·조치시간 KPI. 실제 태그 명명 규칙(`L3.PUMP01.PRESS`)이 그대로 화면에 반영됩니다. *본 캡처는 데모 데이터 기준입니다.*

04 · 특장점

③ 스스로 검증하고, 스스로 고칩니다

"AI 가 만든 코드를 믿을 수 있는가"에 대한 답을 제품 안에 넣었습니다. 에이전트는 결과를 내놓기 전에 **두 번 검증**하고, 문제가 있으면 사용자에게 보이기 전에 되돌아가 고칩니다.



1차 · 기계 검증

타입 검사

컴파일 단계에서 잡히는 오류를 먼저 걸러냅니다. 실패하면 에이전트가 그 오류 메시지를 받아 다시 고칩니다.

2차 · 시각 검증

비전 자가검증

턴이 끝날 때 화면을 캡처해 AI 가 직접 눈으로 검토합니다. 레이아웃이 깨지거나 요소가 겹치는 문제를 코드만 봐서는 알 수 없기 때문입니다.

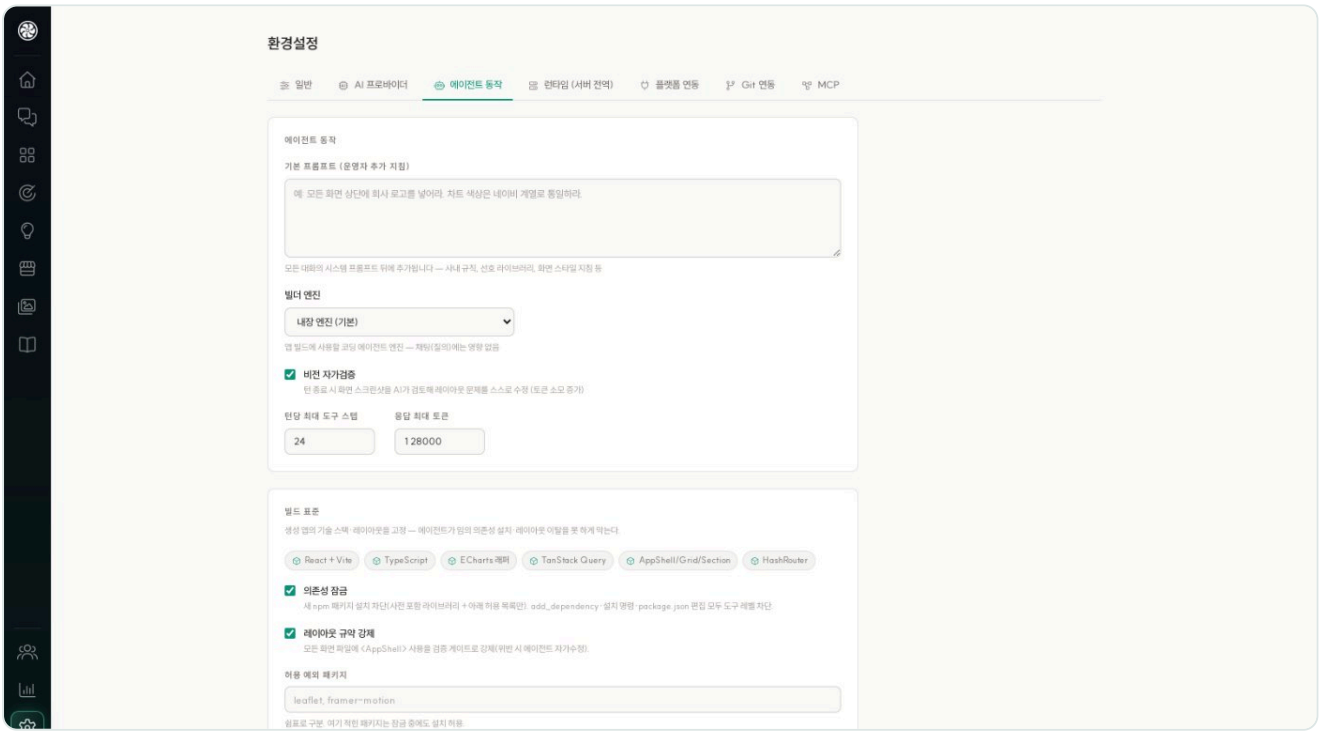


그림 4. 관리자는 자가검증 사용 여부, 빌더 엔진, 턴당 최대 도구 스텝, 사내 공통 지침 (선호 라이브러리·화면 스타일 규칙 등)을 조직 정책으로 설정할 수 있습니다.

05 · 특장점

④ 베테랑의 노하우가 자산으로 쌓입니다

현장에는 문서화되지 않은 규칙이 있습니다 — “이 설비는 기동 30분은 무시한다”, “압력은 항상 온도와 같이 본다”. **현장 스킬** 기능은 이 노하우를 저장해 이후 모든 앱 생성에 에이전트가 활용하게 합니다.

기록

대화 중 나온 규칙을 스킬로 저장합니다. 반복 설명이 필요 없어집니다.

적용

새 앱을 만들 때 에이전트가 관련 스킬을 자동으로 참조합니다.

전파

스킬 팩으로 내보내 다른 공장·법인에 그대로 이식할 수 있습니다.

담당자가 바뀌어도 노하우가 조직에 남습니다. 시간이 지날수록 생성 품질이 올라가는 구조입니다.

⑤ 우리 회사 제품처럼 보입니다

화이트라벨을 기본 기능으로 제공합니다. 고객사 로고와 제품 표시명을 올리면 로그인 화면·사이드바·갤러리에 즉시 반영됩니다.

브랜딩

로고 이미지 업로드와 제품 표시명 변경. 재기동 없이 적용됩니다.

다국어

한국어 · 영어 기본 지원. 해외 법인 공용 배포에 그대로 사용합니다.

소스 인도

생성된 앱을 사내 Git 저장소로 자동 푸시. 앱마다 별도 저장소가 만들어집니다.

테마

라이트 · 다크 모드. 현장 조도 조건에 맞춰 선택합니다.



그림 5. 사용자 관리 — 관리자 · 빌더 · 뷰어 세 가지 역할. 계정은 플랫폼 로그인 위임과 스튜디오 자체 계정(현장 전용)을 함께 지원합니다.

05 · 특장점

⑥ 인터넷이 없는 공장에도 들어갑니다

제조 현장의 상당수는 외부망이 차단돼 있습니다. PlantPulse Studio 는 클라우드 서비스가 아니라 **고객 인프라에 설치되는 제품**이며, 완전 폐쇄망 설치를 정식 절차로 지원합니다.

세 가지 설치 방식

방식	설명	필요 조건
원라인 설치	설치 스크립트 한 줄로 도커 설치 · 환경 구성 · 기동까지 자동 수행	외부망 접속 가능
에어갭 설치	이미지 전체를 파일로 반출입. 반입 시 체크섬과 서명을 자동 검증	완전 폐쇄망
사내 레지스트리	고객사 사내 이미지 레지스트리에 올려 표준 배포 절차로 관리	사내 레지스트리 운영 중

공급망 신뢰

- 배포되는 모든 컨테이너 이미지에 **디지털 서명**이 부착됩니다.
- 폐쇄망 반입 번들은 **체크섬 + 서명**을 반입 전에 자동 검증합니다 — 검증 실패 시 설치가 중단됩니다.
- 릴리즈마다 **이동하지 않는 스냅샷 버전**이 함께 남아, 특정 시점 빌드에 배포를 고정할 수 있습니다.

데이터 주권
공정 데이터는 고객 인프라를 벗어나지 않습니다. Studio 는 기존 플랜트 시스템에 데이터를 위임할 뿐 별도로 저장·전송하지 않습니다. AI 모델 역시 외부 상용 API 와 사내 호스팅 모델(OpenAI 호환 게이트웨이) 중 선택할 수 있어, 프롬프트조차 외부로 나가지 않는 구성이 가능합니다.

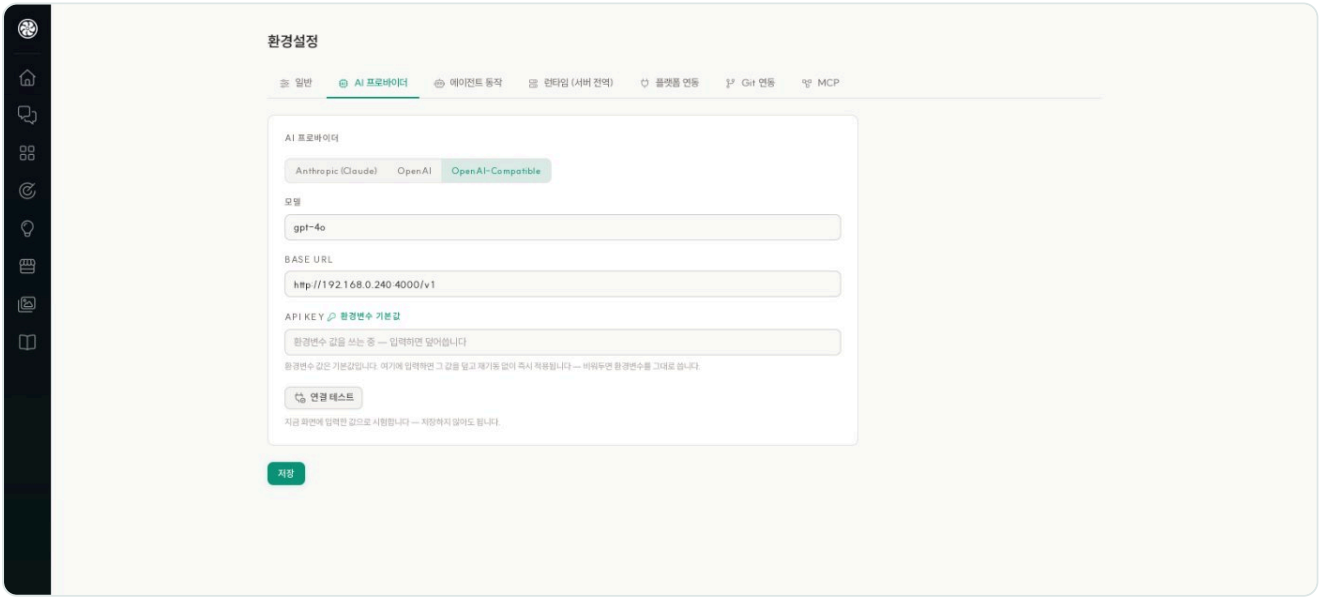


그림 6. AI 프로바이더 설정 — 상용 API 또는 사내 호스팅 모델(OpenAI 호환)을 선택하고, 저장 전에 연결을 시험해 볼 수 있습니다.

06 · 아키텍처

토폴로지 — 무엇이 어디서 도는가

Studio 는 데이터를 소유하지 않는 **얇은 오케스트레이터**입니다. 인증과 실데이터는 기존 플랜트 시스템에 위임하고, Studio 는 "앱을 만들고 실행하는 일"만 담당합니다.

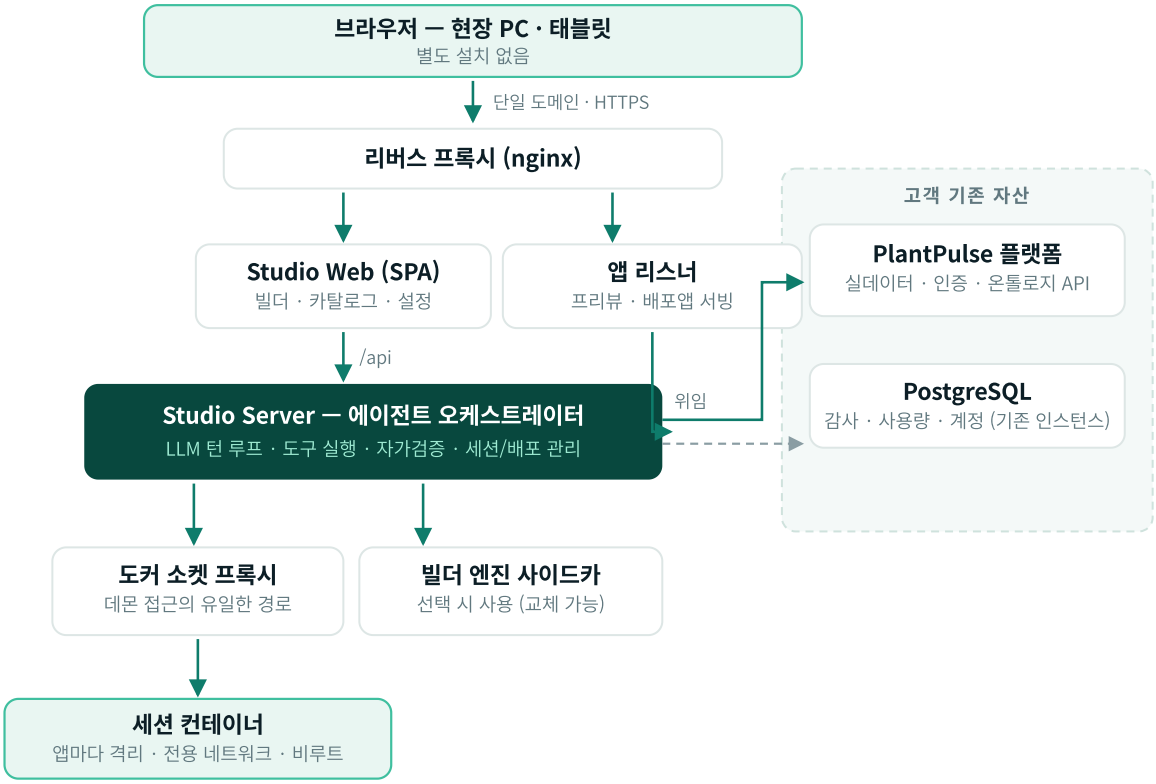


그림 7. 요청 흐름과 배치. 점선은 기존 고객 자산과의 경계입니다 — Studio 는 그 안쪽에 새 데이터 저장소를 만들지 않습니다.

> **운영 스택은 네 개의 컨테이너입니다.** 스튜디오 서버 · 웹 · 빌더 엔진 사이드카 · 도커 소켓 프록시. 모두 헬스체크와 메모리/CPU 한도, 로그 로테이션이 설정된 상태로 제공됩니다.

요청 한 번이 지나가는 길

1. **브라우저 → 리버스 프록시** — 사용자는 하나의 주소만 압니다. 스튜디오 화면과 생성된 앱이 같은 도메인 아래 경로로 나뉘어 서빙되므로, 인증서와 방화벽 규칙을 여러 벌 관리할 필요가 없습니다.
2. **스튜디오 서버가 턴을 실행** — 요청을 받아 AI 프로바이더를 호출하고, 도구 실행 결과를 받아 다음 행동을 정합니다. 진행 상황은 실시간으로 화면에 스트리밍됩니다.
3. **실데이터는 플랫폼에서** — 데이터가 필요하면 기존 플랜트 API 를 호출합니다. Studio 는 그 응답을 중계할 뿐 따로 저장하지 않습니다.
4. **결과는 세션 컨테이너에** — 편집된 파일이 컨테이너의 개발 서버에 즉시 반영되고, 그 화면이 프리뷰로 돌아옵니다.

07 · 아키텍처

기술 스택

특수하거나 폐쇄적인 기술을 쓰지 않았습니다. 고객사 개발팀이 이어받아 유지보수할 수 있는 **표준 스택**으로 구성돼 있습니다.

레이어	구성
프런트엔드	Vite · React 19 · TypeScript · Tailwind CSS — 차트(ECharts) · 다이어그램(Mermaid) · QR 코드
백엔드	Node.js LTS · Express · TypeScript — 입력·환경변수 검증(Zod) · WebSocket · PostgreSQL 드라이버
보안 미들웨어	보안 헤더(helmet) · 레이트리밋(로그인/LLM/전체 API) · CORS 기본 차단
AI 연동	Anthropic 공식 SDK + OpenAI 호환 인터페이스 — 사내 게이트웨이 연결 가능
플랫폼 SDK	OpenAPI 명세에서 **자동 생성**되는 TypeScript 클라이언트. 명세가 바뀌면 CI 가 차이를 감지합니다
데이터	PostgreSQL (기존 플랜트 인스턴스 공용). DB 없이도 파일 저장으로 전 기능 동작
런타임	Docker · Docker Compose. 생성 앱은 세션마다 별도 컨테이너에서 실행
품질 게이트	정적 분석 · 포맷 검사 · 타입 검사 · 자동화 테스트 — 전부 통과해야 병합됩니다
공급망	사설 이미지 레지스트리 · 이미지 디지털 서명 · 폐쇄망 번들 서명 검증

세 가지 설계 원칙

얇은 오케스트레이터

데이터를 소유하지 않습니다. 인증·실데이터는 기존 플랫폼에 위임합니다.

앱 격리

에이전트가 만든 앱은 세션마다 별도 컨테이너·별도 네트워크에서 실행됩니다.

엔진 교체 가능

코딩 엔진을 런타임에 바꿀 수 있습니다. 특정 AI 구현에 묶이지 않습니다.

버전과 릴리즈

제품 버전은 `YYYY.MM` 형식의 **월 단위 캘린더 버전**입니다. 릴리즈마다 이동하지 않는 스냅샷 버전이 함께 남으므로, 고객사는 “이번 달 최신”이 아니라 **검증을 마친 특정 시점**에 배포를 고정할 수 있습니다.

08 · 아키텍처

빌더 엔진과 데이터 이해 계층

코딩 엔진은 교체 가능합니다

AI 분야는 6개월이면 판이 바뀝니다. 특정 구현에 제품을 묶지 않기 위해 코딩 엔진을 **런타임 설정으로 교체**할 수 있게 설계했습니다.

엔진	특징	적합한 경우
내장 엔진	제품에 포함된 자체 도구 사용 루프. 추가 컨테이너가 필요 없습니다	가벼운 설치 · 최소 자원
오픈소스 사이드카	검증된 오픈소스 코딩 에이전트를 사이드카로 실행. 긴 작업의 컨텍스트 관리에 강점	복잡한 앱 · 긴 작업

어느 엔진을 쓰든 상위의 자가검증(타입 검사 · 화면 검증) 로직은 동일하게 적용됩니다. 엔진은 "한 턴을 실행하는 부품"으로 추상화돼 있습니다.

데이터 이해 계층은 어떻게 동작하나

1. **스냅샷 캐시** — 플랫폼의 자산 모델·태그·별칭·계층을 정규화해 캐시합니다. 만료돼도 즉시 끊기지 않고 갱신 중에도 이전 값으로 응답합니다.
2. **결정적 해석** — "3번 라인", "어제"를 식별자와 시간 범위로 바꾸는 일은 **AI 가 아니라 규칙**이 합니다. 같은 입력이면 언제나 같은 결과입니다.
3. **선주입 + 검색** — 턴 시작 시 자산 구조 요약을 미리 넣어주고, 더 필요하면 에이전트가 검색 도구로 직접 조회합니다.
4. **플랫폼 도구 편입** — 건강도 롤업, 알람 진단, 태그 이력, 관계 그래프 등 플랫폼이 제공하는 분석 도구를 에이전트가 그대로 호출합니다.

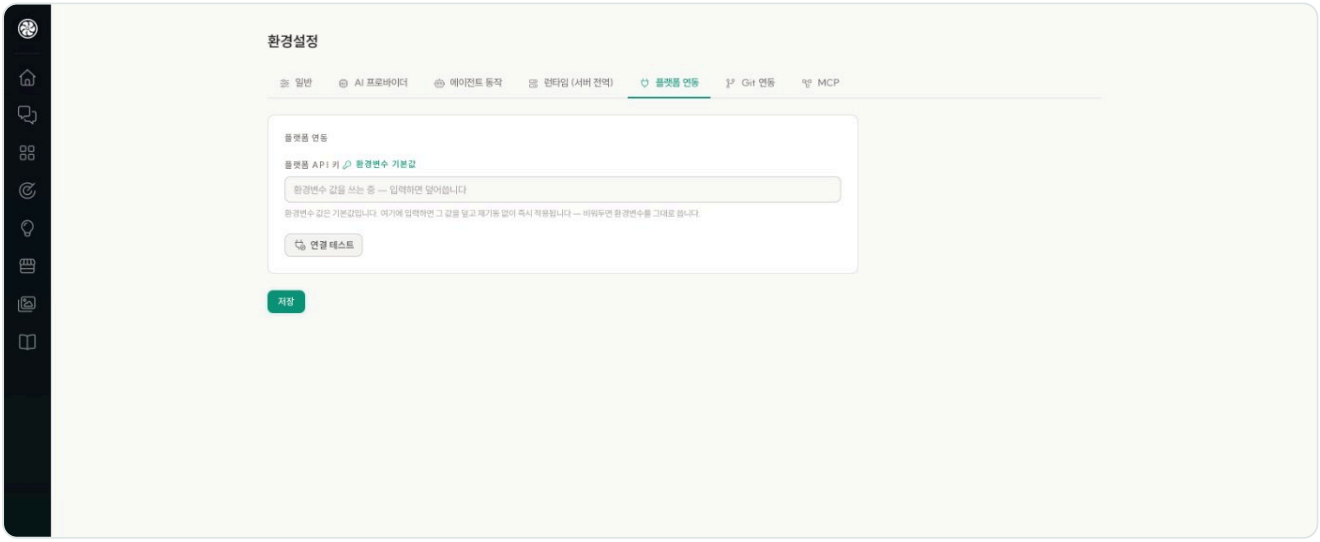


그림 8. 플랫폼 연동 설정 — 환경변수로 주입된 값을 기본으로 쓰되, 화면에서 덮어쓰면 재기동 없이 즉시 반영됩니다. 저장 전 연결 시험을 제공합니다.

09 · 사용 흐름

실제로 어떻게 쓰나 — 다섯 단계

아래 화면은 모두 **실제로 구동 중인 인스턴스에서 캡처**한 것입니다. 설명을 위한 목업이 아닙니다.

1 로그인 → 2 한 문장 입력 → 3 빌드 · 라이브 확인 → 4 배포 → 5 공유 · 운영

1단계 · 로그인

현장 담당자는 별도 설치 없이 브라우저로 접속합니다. 계정은 기존 플랜트 계정을 위임해 쓰거나, 현장 전용 스튜디오 계정을 따로 발급할 수 있습니다.

2단계 · 만들고 싶은 화면을 한 문장으로

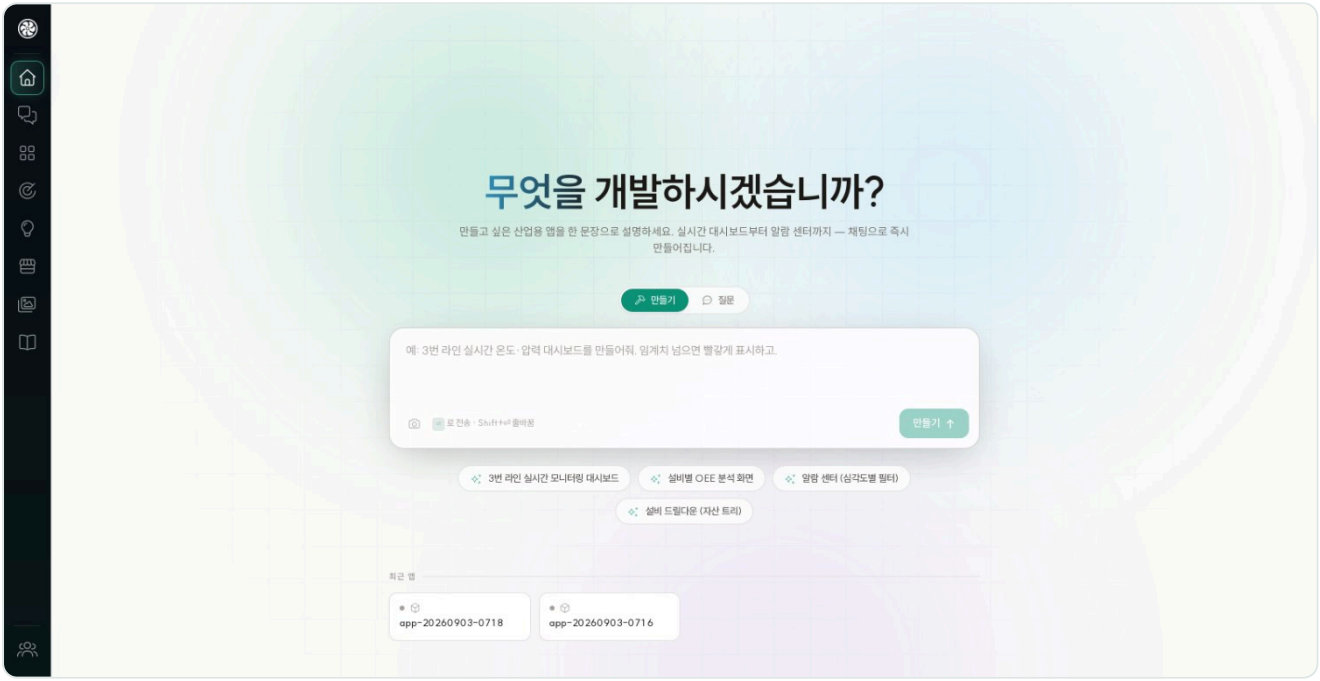


그림 9. 시작 화면. 자주 쓰는 요청은 예시 버튼으로 제공되고, **만들기 / 질문** 두 모드가 있습니다 — 질문 모드는 코드를 건드리지 않고 데이터만 조회합니다.

기존에 만든 앱은 아래 목록에서 이어서 작업합니다. 대화 이력이 함께 복원되므로 며칠 뒤에 돌아와도 “어디까지 이야기했는지”가 남아 있습니다.

09 · 사용 흐름

3단계 · 만들어지는 과정을 눈으로 봅니다

왼쪽에서 에이전트가 무엇을 하고 있는지 한국어로 설명하고, 오른쪽 **라이브 프리뷰**에 결과가 실시간으로 반영됩니다. 완성될 때까지 기다릴 필요가 없습니다.

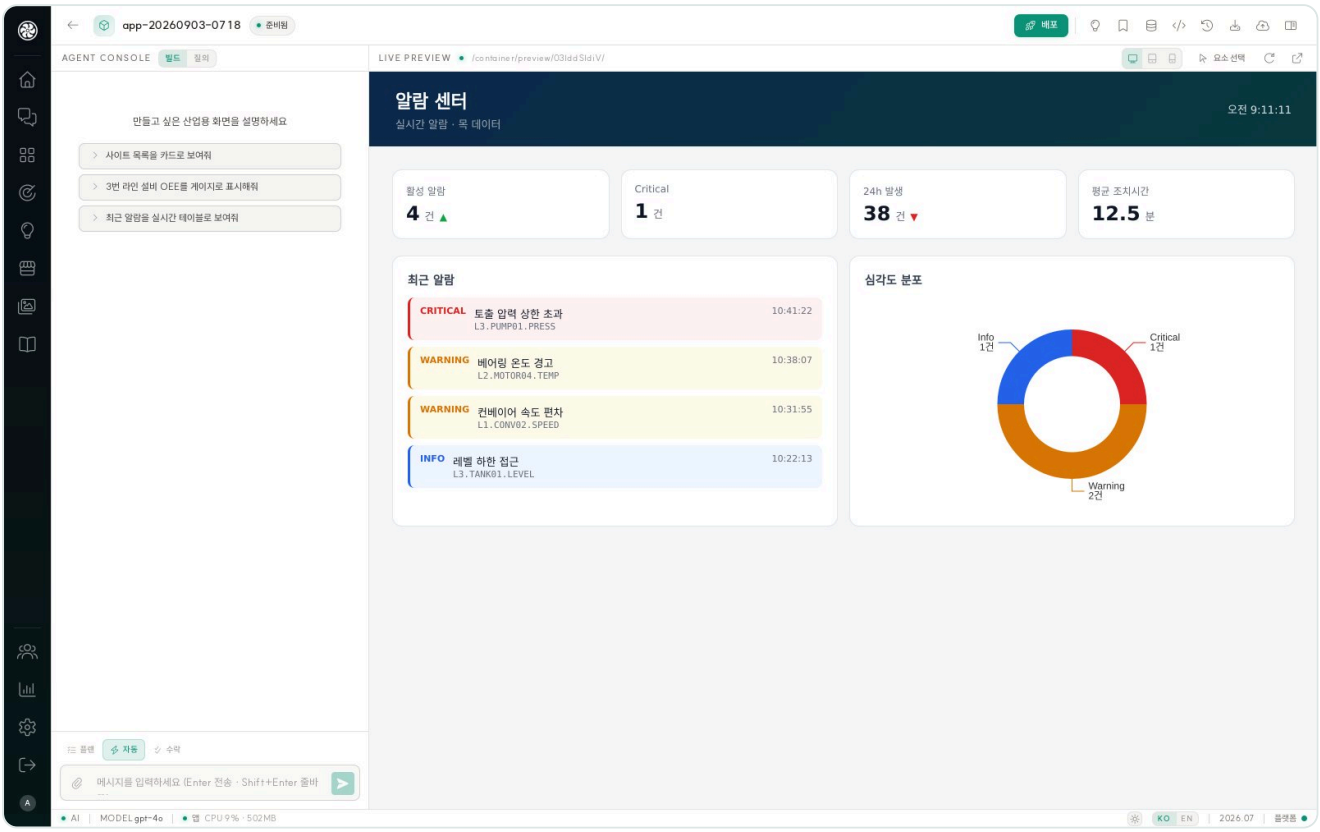


그림 10. 빌더 화면. 좌측 **에이전트 콘솔**, 우측 **라이브 프리뷰**. 상단 우측의 **배포** 버튼으로 현재 상태를 버전으로 남깁니다. 프리뷰는 데스크톱 · 태블릿 · 모바일 폭으로 전환해 확인할 수 있습니다.

화면을 가리켜 고치기
요소 선택

프리뷰에서 고치고 싶은 부분을 직접 클릭해 지목하면, 에이전트가 그 요소를 대상으로 수정합니다. “가운데 위쪽 그래프”라고 설명할 필요가 없습니다.

돌아갈 수 있음
이력과 되돌리기

턴마다 자동 커밋이 남습니다. 마음에 들지 않으면 이전 상태로 되돌리고 다시 시도할 수 있습니다.

에이전트는 자기가 한 일을 설명합니다. 어떤 데이터를 찾아봤는지, 왜 그렇게 만들었는지, 무엇이 부족한지를 한국어로 남깁니다 — 예를 들어 필요한 태그가 플랫폼에 아직 없으면 그 사실을 알리고 대체 방안을 제시합니다. “왜 이렇게 나왔는지 모르겠는” 결과가 나오지 않도록 설계된 부분입니다.

09 · 사용 흐름

4–5단계 · 배포하고, 나눠 쓰고, 지켜봅니다

4단계 · 배포

버튼 한 번으로 현재 상태가 버전으로 고정되어 배포되고, 이전 버전은 보관되어 **롤백**할 수 있습니다. 배포된 앱은 **QR 코드**나 **임베드**로 공유합니다 — 현장 태블릿에서 QR 한 번으로 바로 엽니다.

버전 관리

배포마다 버전이 남고 이전 버전으로 되돌릴 수 있습니다.

접근 권한

앱마다 볼 수 있는 사람을 따로 지정합니다.

공유

QR · 임베드 · 갤러리 공개 중에서 고릅니다.

5단계 · 운영과 가시성

관리자는 누가 무엇을 얼마나 썼는지 한 화면에서 봅니다. AI 사용량은 **사용자별 · 프로젝트별 · 엔진별**로 집계되므로 비용을 예측하고 배분할 수 있습니다.

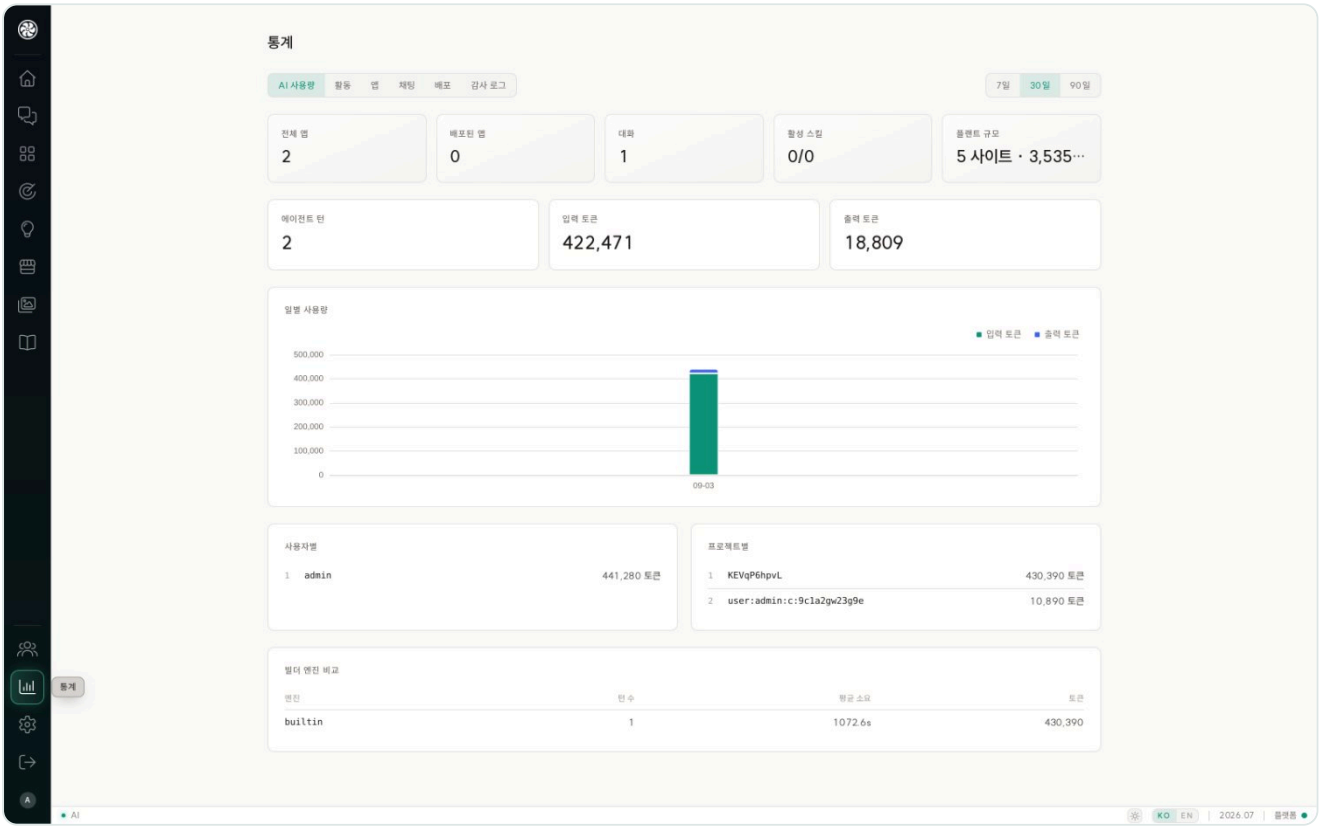


그림 12. 통계 — 앱/배포/대화 현황, 연결된 플랜트 규모, AI 토큰 사용량, 빌더 엔진별 소요 비교, 감사 로그. *수치는 데모 인스턴스 실측값입니다.*

감사 로그

누가 언제 무엇을 했는지 페이징 조회.

사용량 관리

사용자·프로젝트별 집계로 비용 배분.

자동 복구

컨테이너 유실 복구, 유휴 세션 정리.

10 · 보안

보안 · 거버넌스

"AI 가 코드를 쓰고 그 코드를 실행한다"는 구조는 그 자체로 통제가 필요합니다. PlantPulse Studio 는 생성된 앱을 **신뢰하지 않는 코드**로 취급하고, 격리와 감사를 전제로 설계했습니다.

실행 격리

- **앱마다 별도 컨테이너** — 생성된 앱은 세션 전용 컨테이너에서 실행됩니다.
- **비루트 실행 · 자원 상한** — 메모리 · CPU · 프로세스 수 제한이 걸립니다. 한 앱이 폭주해도 다른 세션과 호스트에 영향을 주지 않습니다.
- **세션 간 네트워크 차단** — 세션 컨테이너는 컨테이너 간 통신이 꺼진 전용 네트워크에 배치됩니다. 한 세션의 앱이 다른 세션을 들여다볼 수 없습니다.
- **파일 경로 격리** — 에이전트의 파일 도구는 자기 워크스페이스 밖으로 나갈 수 없습니다.
- **플랫폼 조회 화이트리스트** — 에이전트의 데이터 조회는 허용된 읽기 경로로 제한됩니다.

인프라 접근 통제

스튜디오 서버는 컨테이너 관리 데몬에 **직접 접근하지 않습니다.** 전용 프록시를 통해서만, 그리고 **실제로 필요한 조작만** 허용된 상태로 접근합니다. 불필요한 관리 기능은 아예 노출되지 않으며, 데몬 접근이 한 곳으로 모여 감사 지점이 하나가 됩니다.

계정 · 권한 · 감사

항목	내용
역할	관리자 · 빌더 · 뷰어 3단계. 역할은 데이터베이스에만 존재하며, 환경변수나 설정 파일로 권한을 부여하는 경로는 없습니다
인증	기존 플랜트 계정 위임 로그인 또는 현장 전용 계정. 비밀번호는 표준 해시 알고리즘으로 저장됩니다
세션	토큰 유효기간 제한. 서버 재기동 후에도 세션 유지
키 관리	플랫폼 접근 키는 서버가 보관하고 대행 조회합니다. 생성된 앱의 데이터 요청에는 **로그인한 사용자에게만** 키가 주입됩니다
감사 로그	주요 행위를 기록하고 관리자 화면에서 조회 가능
비밀 관리	환경 설정·워크스페이스·빌드 산출물은 형상관리에서 제외됩니다

애플리케이션 계층 방어

보안 헤더 적용 · 로그인/AI 호출/전체 API 각각에 대한 요청 제한 · 교차 출처 요청 기본 차단 · 외부 호출(웹훅) 대상 검증. 컨테이너 제어 명령은 셸을 거치지 않고 인자로 직접 전달되어 명령 주입 경로를 만들지 않습니다.

> **도입 전 보안 검토를 권장합니다.** 위 통제는 제품이 기본 제공하는 것이며, 고객사 보안 정책(망 분리 수준, 인증 연동 방식, 로그 수집 대상)에 맞춘 구성은 도입 단계에서 함께 확정합니다. 요청 시 상세 보안 아키텍처 문서를 제공합니다.

11 · 납품

배포 · 납품 모델

설치는 한 줄, 또는 파일 반입

구분	절차	소요
일반망	설치 스크립트 1회 실행 → 컨테이너 런타임 설치 · 환경 구성 · 기동까지 자동	수십 분
폐쇄망	이미지 번들 파일 반입 → 체크섬 · 서명 자동 검증 → 로드 → 기동	수십 분 (반입 절차 별도)
업그레이드	새 버전 이미지로 교체 후 재기동. 데이터베이스 스키마는 기동 시 자동 적용	분 단위

운영에 필요한 것

백업

데이터 루트와 데이터베이스 백업 스크립트를 함께 제공합니다. 정기 백업 스케줄 등록 도구가 포함됩니다.

복구

백업본으로부터의 복구 절차가 스크립트화되어 있습니다.

진단

설치 상태·연결·헬스체크를 한 번에 점검하는 진단 도구를 제공합니다.

로그

네 컨테이너 모두 로그 로테이션이 설정되어 디스크를 잠식하지 않습니다.

업그레이드 주기와 버전 고정

월 단위 캘린더 버전(`YYYY.MM`)으로 정기 릴리즈합니다. 동시에 **이동하지 않는 스냅샷 버전**이 릴리즈마다 남습니다. 고객사는 “최신”을 자동으로 따라가는 대신 **검증을 마친 스냅샷에 고정**해 두고, 필요할 때만 계획적으로 올릴 수 있습니다.

시스템 요구사항 (권장)

Linux 서버 1대 · 컨테이너 런타임 · 기존 PlantPulse 플랫폼 접근 경로 · (선택) 사내 AI 게이트웨이 또는 외부 AI API 접근. 세션 컨테이너가 동시에 여러 개 뜨므로 **메모리 여유**가 성능에 가장 크게 작용합니다 — 정확한 사양은 예상 동시 사용자 수를 기준으로 산정해 드립니다.

12 · 레퍼런스

레퍼런스 — 사례와 검증 체계

구축 사례와 함께, 제품의 신뢰도를 **실제로 돌고 있는 검증 장치**로 제시합니다. 아래 검증 항목은 모두 자동화되어 릴리즈마다 반복 실행됩니다.

릴리즈 파이프라인에서 자동 수행되는 검증

검증	내용
코드 품질 게이트	정적 분석 · 포맷 검사 · 타입 검사 · 자동화 테스트. **하나라도 실패하면 병합되지 않습니다**
설치 드릴	깨끗한 환경을 매번 새로 띄워 **고객이 실행할 설치 절차 그대로** 완주시키고 동작을 확인합니다 — “설치되지 않는 패키지”가 배포되는 것을 원천 차단합니다
이미지 내용물 검증	배포 이미지가 실제로 선언된 내용물을 담고 있는지, 헬스체크가 정상 통과하는지 확인합니다
이미지 서명	레지스트리 반영 후 다이제스트에 디지털 서명을 자동 부착합니다
폐쇄망 번들 검증	반입 번들의 체크섬과 서명을 설치 전에 자동 검증합니다
API 명세 동기화	플랫폼 API 명세와 클라이언트 SDK 사이의 차이를 감지해 조용한 불일치를 막습니다
데이터베이스 마이그레이션	기동 시 자동 적용. 다중 인스턴스 동시 기동에도 중복 적용되지 않도록 잠금이 걸려 있습니다

제품 문서

사용자 매뉴얼

현장 사용자를 위한 온라인 매뉴얼을 제공합니다. 제품 화면에서 바로 연결됩니다. — **docs.plantpulse.io**

기술 문서

아키텍처 · 설치 · 배포 · 백업 · 릴리즈 절차 문서를 납품 시 함께 제공합니다.

구축 사례

중공업 · 발전

두산중공업

가스터빈 디지털 트윈 프로젝트

회전기기 설비를 대상으로 한 디지털 트윈 화면 구축.

조선 · 해양

한화오션

설비 대시보드 노코드 제작 프로젝트

현장 담당자가 직접 설비 대시보드를 제작하는 노코드 방식 적용.

상세 구축 범위와 도입 효과는 담당 영업 채널을 통해 개별 안내드립니다.

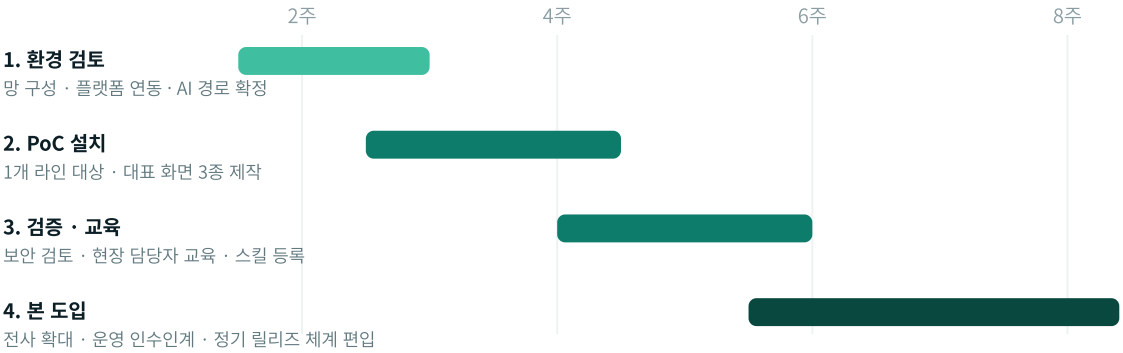
운영 실적

본 제안서의 모든 화면은 **상시 가동 중인 인스턴스**에서 캡처했습니다. 해당 인스턴스는 릴리즈 파이프라인이 매 릴리즈마다 자동 배포하는 환경으로, 제품이 문서상이 아니라 **실제로 설치되고 동작하는 상태**임을 보여줍니다.

13 · 도입

도입 절차 및 일정

PoC 부터 본격 도입까지 **4단계**로 진행합니다. 아래는 표준 일정이며, 고객사 환경과 검토 절차에 따라 조정됩니다.



PoC 에서 확인하시길 권합니다

- 현장 담당자가 **교육 없이** 한 문장으로 원하는 화면을 얻는가
- 우리 공정의 태그 명명 규칙을 **이름만으로** 찾아내는가
- 기존 대시보드로는 못 만들던 화면이 **실제로 만들어지는가**
- 폐쇄망 설치 절차가 고객사 보안 정책 안에서 **완주되는가**

도입 단계에서 함께 정할 것

AI 실행 경로

외부 상용 API 를 쓸지, 사내 게이트웨이/자체 호스팅 모델을 쓸지. 망 정책에 따라 결정합니다.

계정 연동 방식

기존 플랜트 계정 위임과 현장 전용 계정 중 어느 쪽을 주로 쓸지, 역할 배분은 어떻게 할지.

서버 사양

예상 동시 사용자 수를 기준으로 메모리·CPU 를 산정합니다.

백업 · 로그 정책

백업 주기와 보관 기간, 로그 수집 대상을 사내 정책에 맞춥니다.

다음 단계

30분 데모 · PoC 환경 협의

실제 인스턴스에서 “한 문장 → 화면”을 직접 보시는 것이 가장 빠릅니다. 고객사 태그 체계 일부를 미리 알려주시면 **귀사 데이터 이름으로** 시연해 드립니다.

PlantPulse® Studio

산업용 AI 앱 빌더 · 온프레미스 / 폐쇄망 지원

제품 문서 docs.plantpulse.io · 본 문서 Rev. 2026.09

kopens

PlantPulse® Studio

INDUSTRIAL APP BUILDER

도입 문의

산업 AX 전환, 지금 시작하세요.

도입 상담 · 데모 요청 · 견적 문의 — 무엇이든 남겨주세요. 담당 전문가가 영업일 기준 1일 이내에 답변드리고, 원하시면 실제 제품 화면으로 데모 시연도 바로 잡아드립니다.

전화

1588-5268

이메일

webmaster@kopens.com

도입 문의 · 데모 요청

kopens.com/ko/contact

회사 홈페이지

kopens.com

제품 홈페이지

plantpulse.io

제품 문서

docs.plantpulse.io

PlantPulse® 제품군 — 함께 도입하면 현장부터 AI 까지 한 벌로 이어집니다

Edge
산업 프로토콜 엣지 게이트웨이

Platform
엔터프라이즈 IIoT / DataOps 플랫폼

AI
플랜트 운영 AI 코파일럿

Studio
AI 애플리케이션 빌더

㈜한국오픈솔루션 · KOPENS, Inc.

© 2026 KOPENS. All rights reserved. 플랜트펄스®는 ㈜한국오픈솔루션의 등록상표입니다.

본 문서 Rev. 2026.09 · 기재된 수치·화면은 작성 시점 기준이며 사전 통지 없이 변경될 수 있습니다.

kopens®