



Future Academy 2기

AI Agent 기반 제조공장 예지보전 및 지능형 운영관리 시스템

2025.04.12

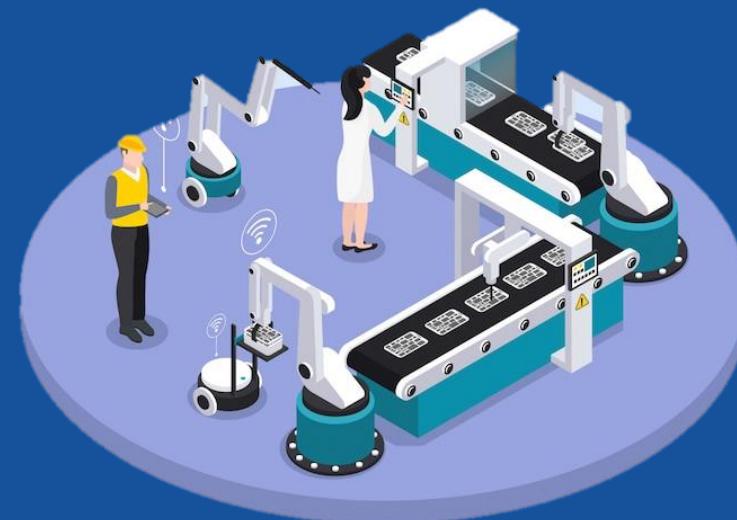
FACMAN조

박찬영 chanreverse@gmail.com

강영혜 grace142@naver.com

황나영 nayeong123@naver.com

목차



Chapter 1

기획 배경

Chapter 2

서비스 구성

Chapter 3

서비스 시연

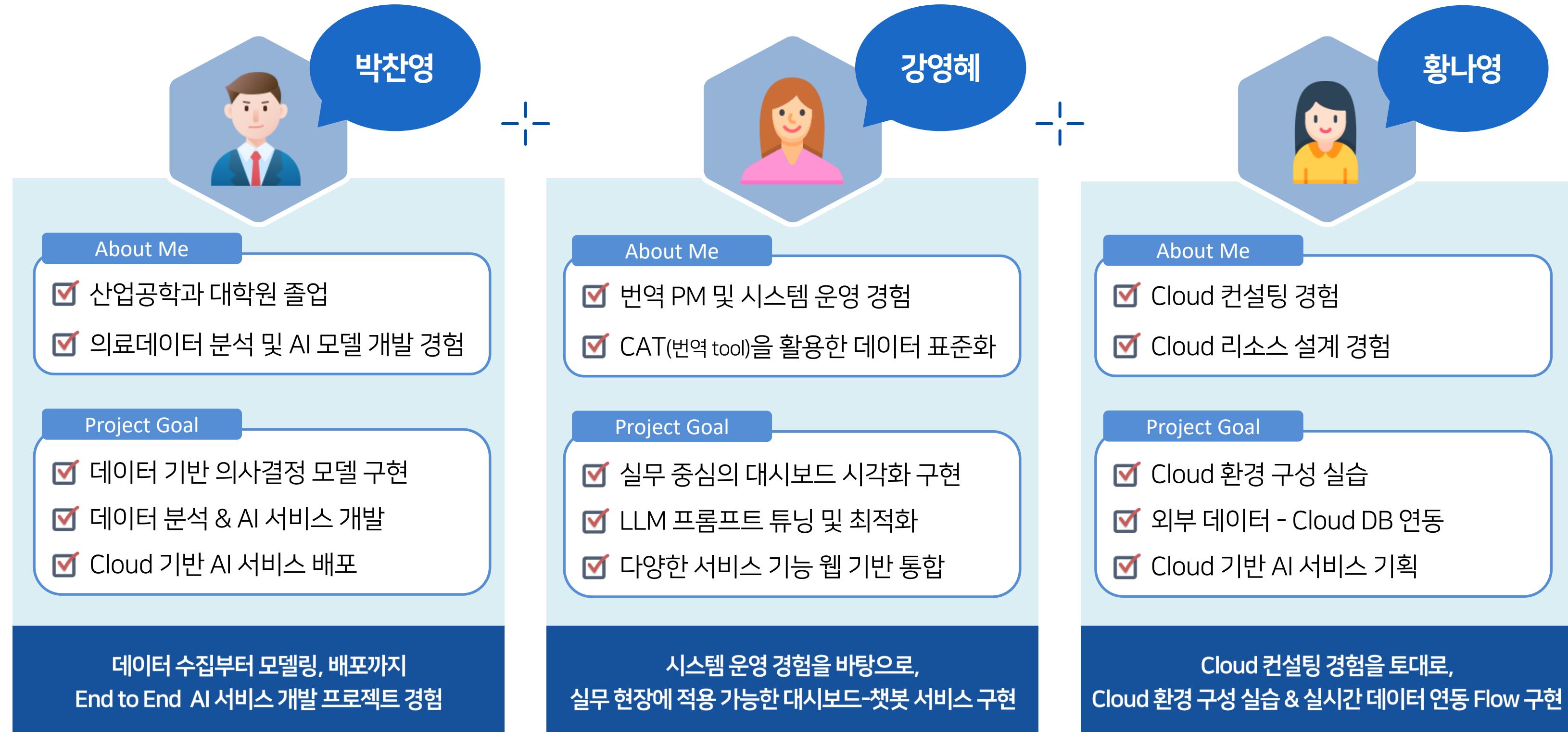
Chapter 4

기대효과

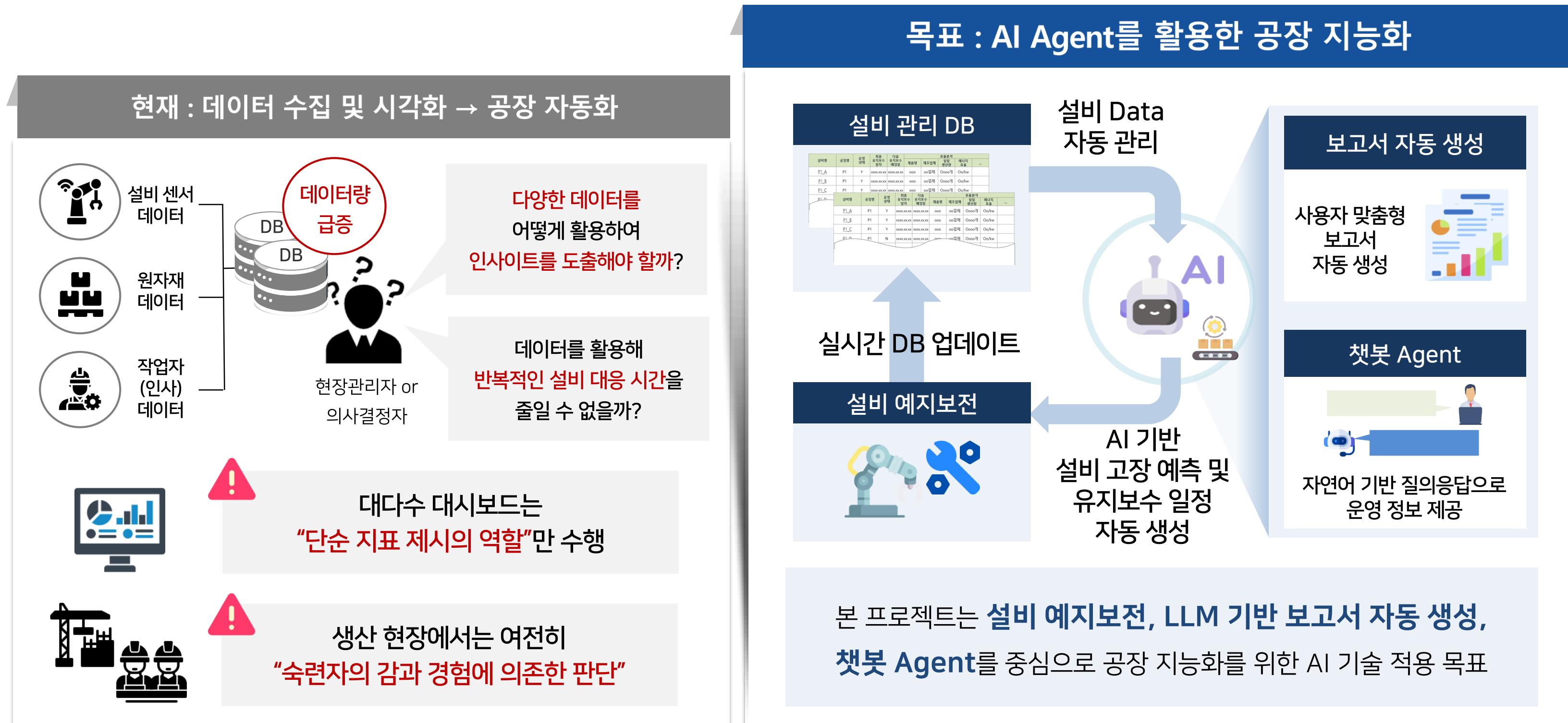
Chapter 5

현장 적용 및 고도화 전략

팀원 각자의 경험과 기술 역량을 바탕으로, 제조 산업 내 AI 서비스 구현 가능성을 탐색하는 프로젝트를 기획함



『2025 스마트공장·자동화산업전』주요 인사이트: 제조공장은 AI Agent 기반 지능화·자율화 초기 단계이며, 핵심 과제는 **데이터 기반의 설비 운영 최적화**, 자연어 인터페이스를 포함한 AI 기반 의사결정 체계의 고도화임



다양한 자동화 도구와 Agent 유형 중, **목표를 설정하고 스스로 학습하며 가장 효율적인 행동을 선택하는 'AI Agent'** 구현을 본 프로젝트의 목표로 설정함

자동화 도구 비교

	AI Agent	AI 어시스턴트	봇
목적	자율적이고 선제적으로 태스크 수행	사용자의 태스크 지원	간단한 태스크 또는 대화 자동화
기능	<ul style="list-style-type: none"> 복잡한 다단계 작업 수행 학습 및 적응 독립적으로 의사 결정을 내릴 수 있음 	<ul style="list-style-type: none"> 요청, 프롬프트에 응답 정보 제공 및 간단한 태스크 수행 작업을 추천할 수 있지만 결정은 사용자가 내림 	<ul style="list-style-type: none"> 사전 정의된 규칙을 따름 제한된 학습 기본적인 상호작용
상호 작용	선제적, 목표 지향적	반응형 사용자 요청에 응답	반응형 트리거, 명령에 응답

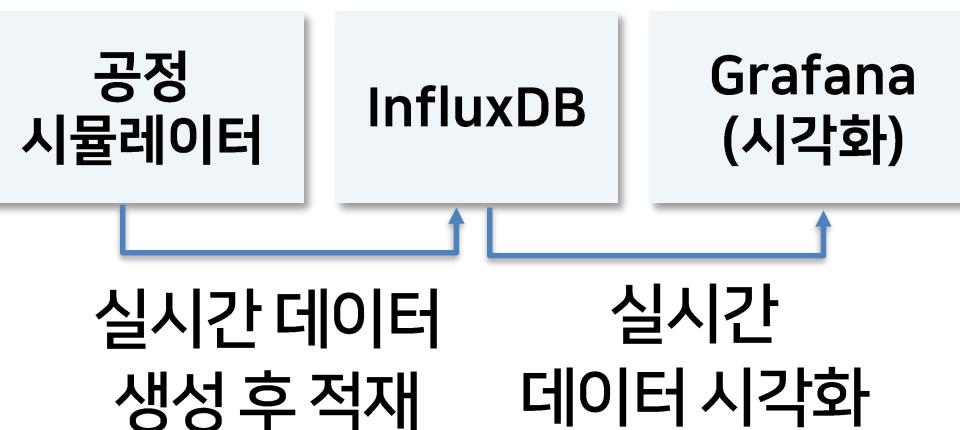
Agent의 종류

단순 반사 Agent 현재 상태만 보고 정해진 규칙에 따라 반응	모델기반 반사 Agent 과거 정보를 저장해 내부 상태 기반으로 반응
 목표 기반 Agent 특정 목표를 설정하고, 목표 달성을 위한 행동 결정	 유ти리티 기반 Agent 가능한 행동 중 가장 높은 효용(Utility) 선택
 학습 Agent 경험을 통해 스스로 학습하고 성능 개선	계층적 Agent 상위 목표-하위 작업 구조로 복잡한 작업 처리

인프라 환경 구성 및 보고서 구현

- 1 AWS EC2 설치 및 환경 구성
 - 공정 시뮬레이터, InfluxDB, Grafana (오픈소스 Tool) 설치

EC2 간 데이터 Flow 설계



- 3 InfluxDB → Flask 서버
WebSocket 실시간 연동

- 4 LLM 기반 보고서 생성 구현
→ InfluxDB 데이터 조회 및
주요 KPI 지표 계산, 내용 요약

공정 데이터 기반 AI Agent 개발

- 1 공정 시뮬레이터 구현
 - 대기행렬이론에 입각한 2x2 공정 구현
 - 도착률, 서비스율, 고장률, 수리시간, 점검시간 파라미터 설정

- 2 데이터 파이프라인 설계
 - 생산이력, 설비 점검 로그 설계
 - 가상 공정 시나리오 설계

- 3 LLM 기반 예지보전 AI Agent 구현 및 평가
 - LangGraph 오픈 프레임워크를 통한 복잡한 로직 구현
 - LLM 모델: gpt-4o

대시보드 구현 및 프롬프트 튜닝

- 1 Grafana 기반 대시보드 시각화 구성
 - 실시간 설비 데이터를 기반으로 KPI 시각화 및 설비별 패널 구성

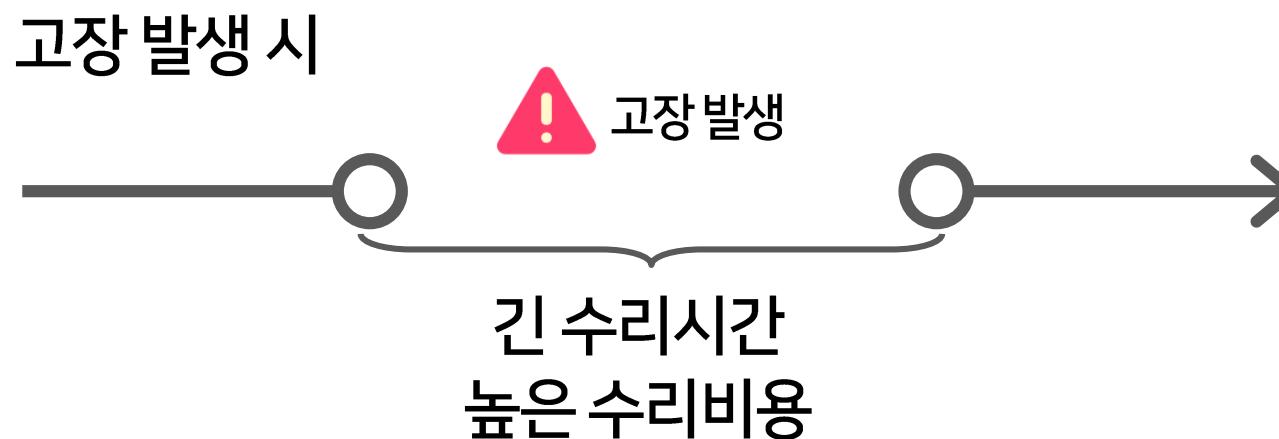
- 2 Flask 기반 웹 대시보드 및 UI 구성
 - 사이드바, 상태 박스 등 레이아웃 구성
 - WebSocket 실시간 연동

- 3 LLM 프롬프트 튜닝 및 응답 최적화
 - System prompt 작성 및 KPI 응답 품질 개선

- 4 챗봇 UI 연동 및 대화 흐름 설계
 - 말풍선, 히스토리, 탭 전환 등
직관적 대화 UI 구현

예지보전의 필요성과 구현 방안을 고민한 결과,
과거 수리 이력과 운영 데이터를 기반으로 **설비 점검 시점을 판단하고 자동으로 작업을 지시하는 서비스를 기획함**

설비 예지보전의 필요성



예지보전을 한다면?



설비 예지보전 방법론

"Q. 어떻게 점검 시점을 정할까?"

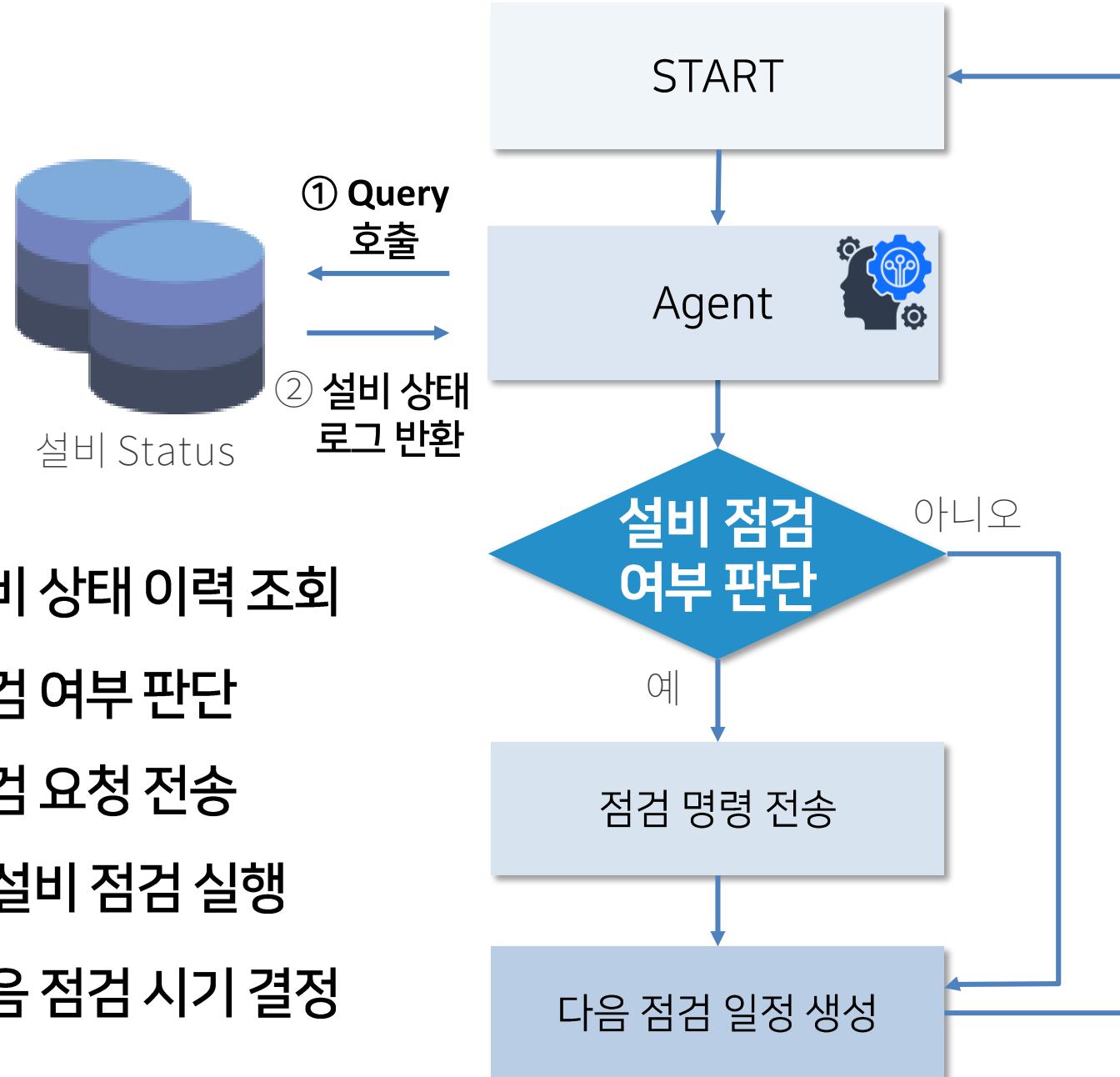
설비 진동을 활용한
고장확률 예측

불량 생산 기반
간접적인 고장 유추

과거 수리이력을 통한
점검 시기 설정

과거 설비 수리 이력과 운영 데이터를 기반으로 AI Agent가 설비 점검 시점을 판단하고 자동으로 작업을 지시하는 구조로 설계함

Agent 구현 방식



예지보전 Agent

```

sim - chanyoungpark@bagchan-yeong-ui-MacBookAir ...ub/FACMAN/sim ... zsh > python3 - 58x16
#####
P1-A
#####
다음 고장 상태 주 히 . P1-A
#####
점 검 여부 판 단 #####
결정 응답 : ````json
{"decision": true, "next_inspection": "2025-04-11T10:47:17.139453+00:00"}}

InfluxNode: {'db_outputs': ['2025-04-11 09:47:17.139453+00:00: 1\n2025-04-11 09:47:17.080870+00:00: 1\n2025-04-11 09:47:17.139453+00:00: 1']}

redis publish: P1-A_maintenance -> maintenance_request
request_maintenance: {'messages': [AIMessage(content='공정 P1-A에 대한 점검 요청을 전송했습니다.', additional_kwargs={}, response_metadata={}, id='48c65692-cd0f-432c-88b2-9b538ffb52ba')]}
final_answer: {'next_inspection': ['````json\n{\n    "next_inspection": "2025-04-11T09:51:00+00:00",\n    "reason": "\n        평균 고장 간격과 평균 수리 시간을 분석한 결과, 고장 간격은\n        약 4분에서 5분 사이로 나타났으며, 수리 시간은 약 1분 내외\n       로 나타났습니다. 따라서, 다음 점검은 마지막 고장 시점인\n        2025-04-11T09:47:17+00:00로부터 약 4분 후인 2025-04-11T09:51:\n        00+00:00에 수행하는 것이 적절하다고 판단됩니다. 이는 고장\n        이 발생하기 전에 점검을 수행하여 공정의 안정성을 유지하기\n        위함입니다.\n    }\n````']}
  
```

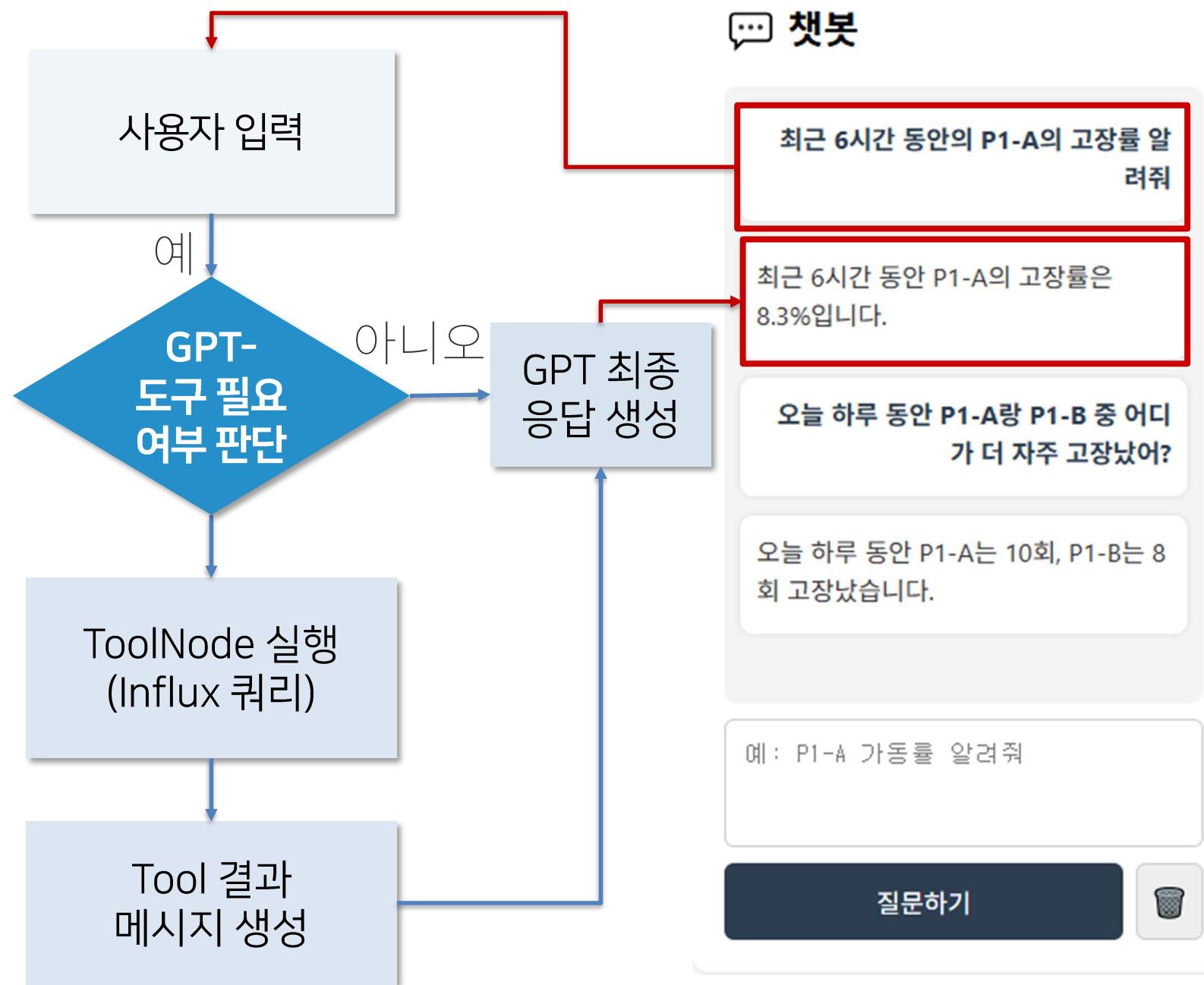
공정 시뮬레이터

```

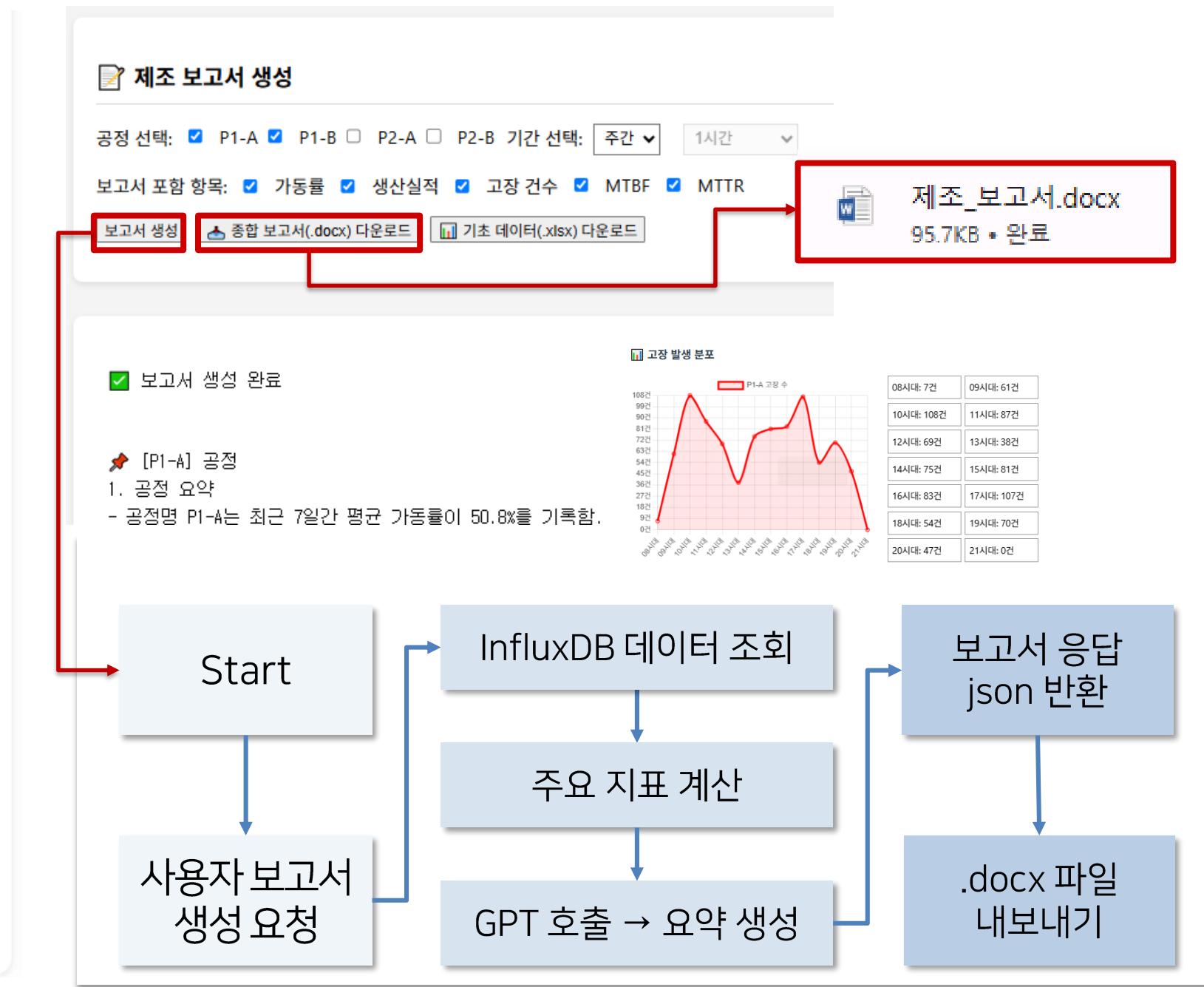
sim - chanyoungpark@bagchan-yeong-ui-MacBookAir ...ub/FACMAN/sim ... zsh > python3 - 58x16
Logging process: process 25041110170A P0 input
Produced: 25041110170A
Received maintenance command: maintenance_request
Logging status: failure False
Logging process: process 25041110021A P2 P2-A int
errrupt
Logging status: maintenance start False
Logging status: repair start False
Logging status: repair finish True
  
```

공정 데이터에 LangGraph 기반 챗봇 Agent를 연동하여 사용자 요청에 실시간 질의 응답을 제공하고, 데이터를 소켓 방식으로 조회 및 주요 지표를 계산함으로써 자동 보고서 생성 기능을 구현함

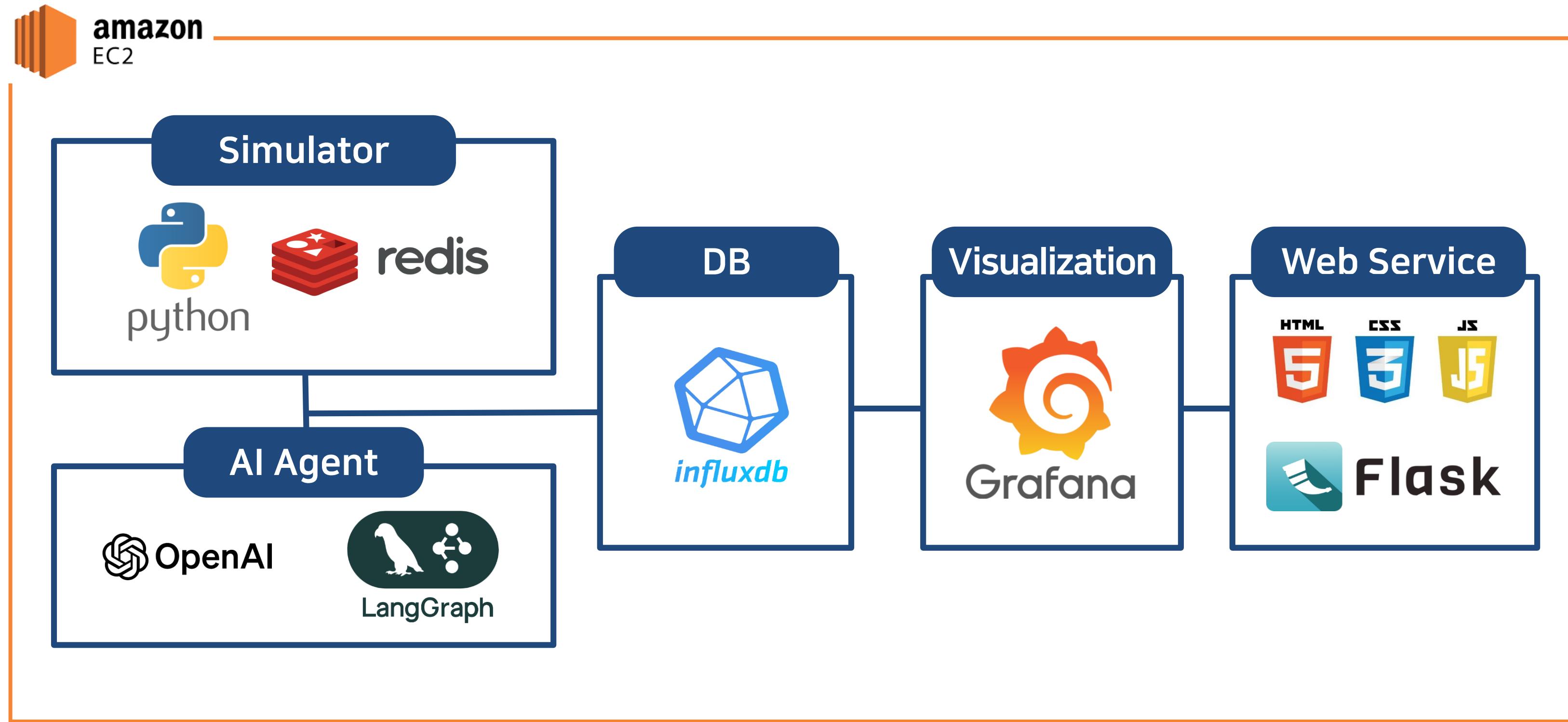
챗봇 Agent 구현 방식



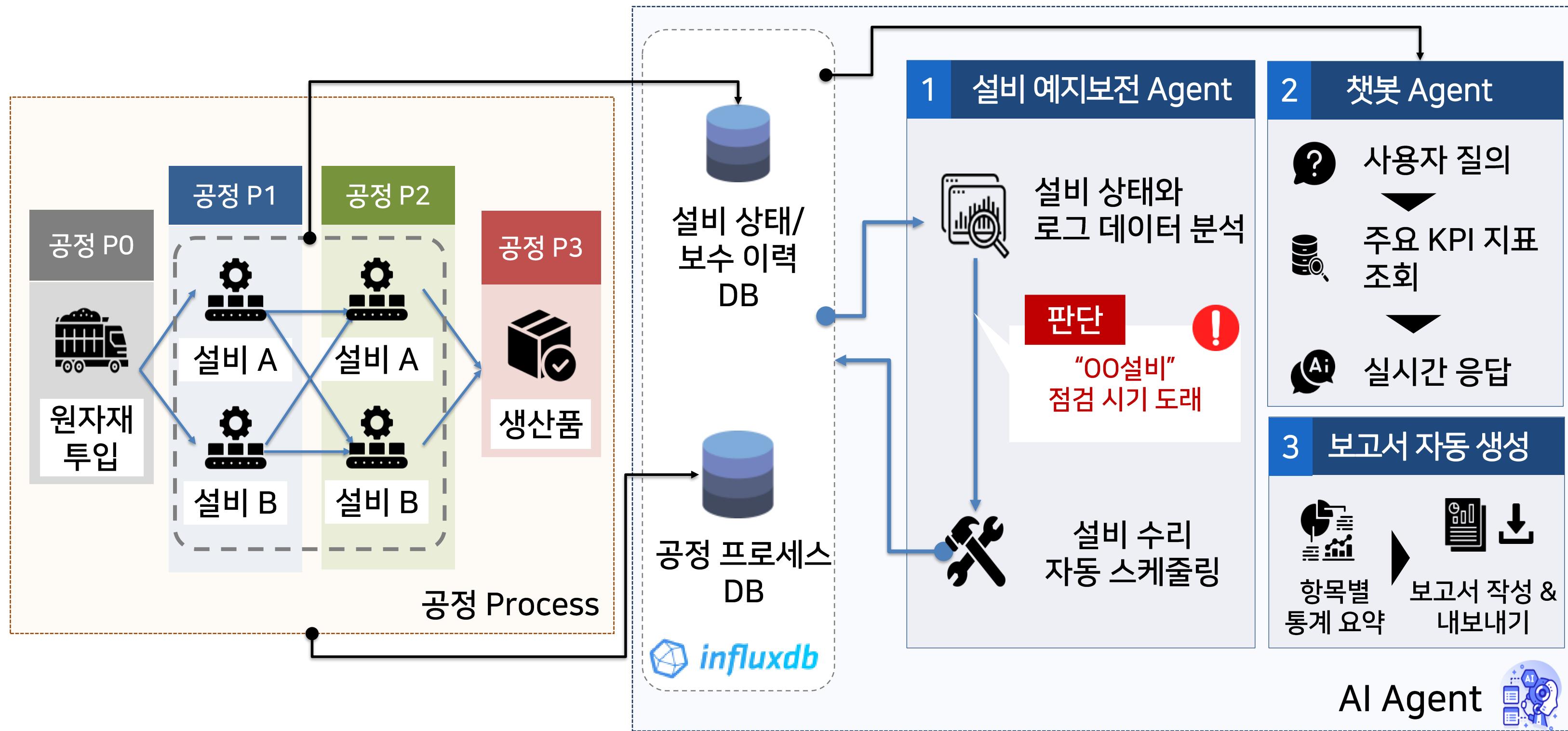
LLM 보고서 생성 구현 방식

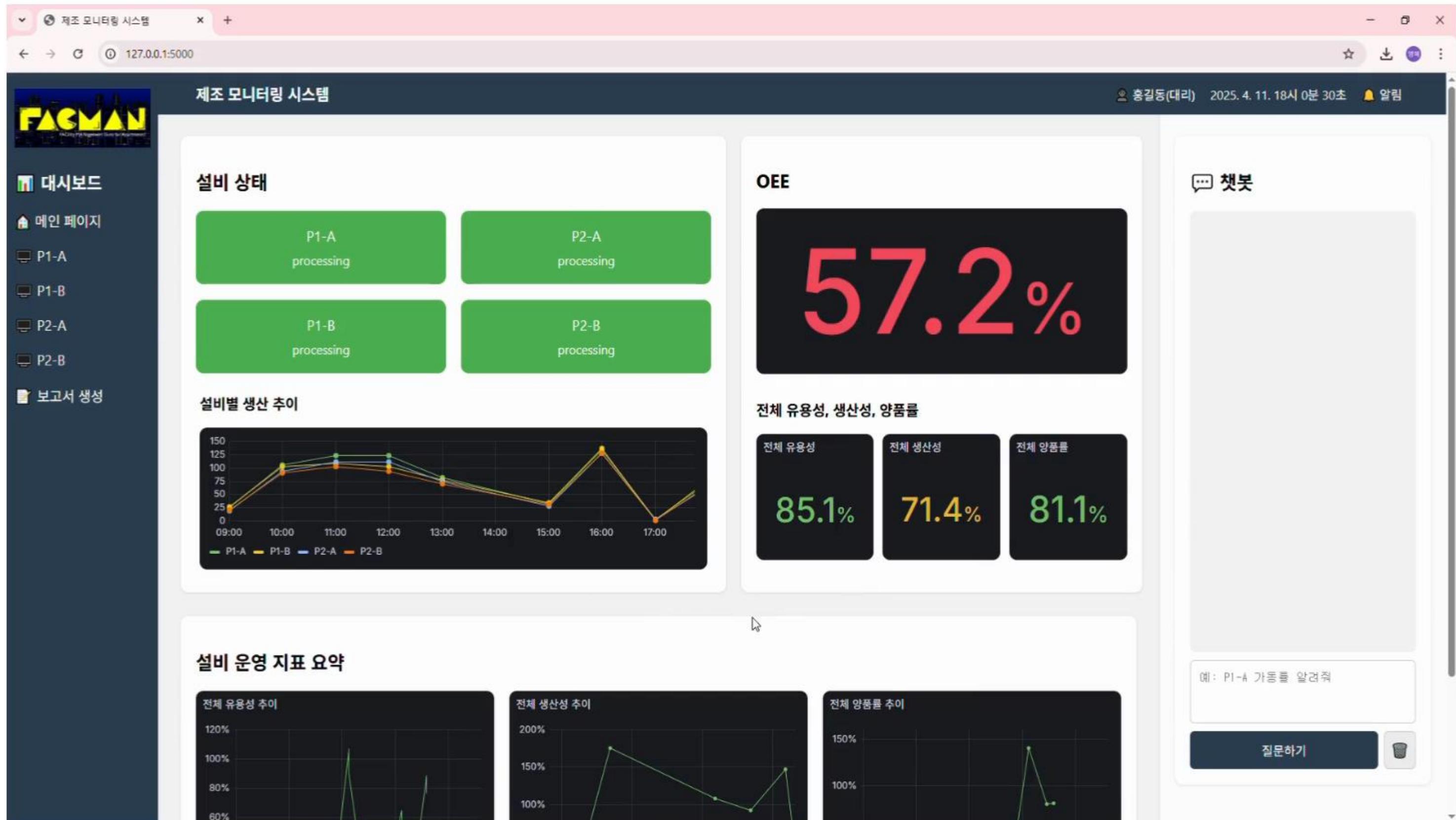


Amazon EC2 기반으로 시뮬레이터 데이터 생성부터 분석·시각화·서비스까지의 구조를 설계하고,
Python, InfluxDB, Grafana, Flask 등 다양한 오픈소스 툴과 AI 프레임워크를 연계하여 구현함

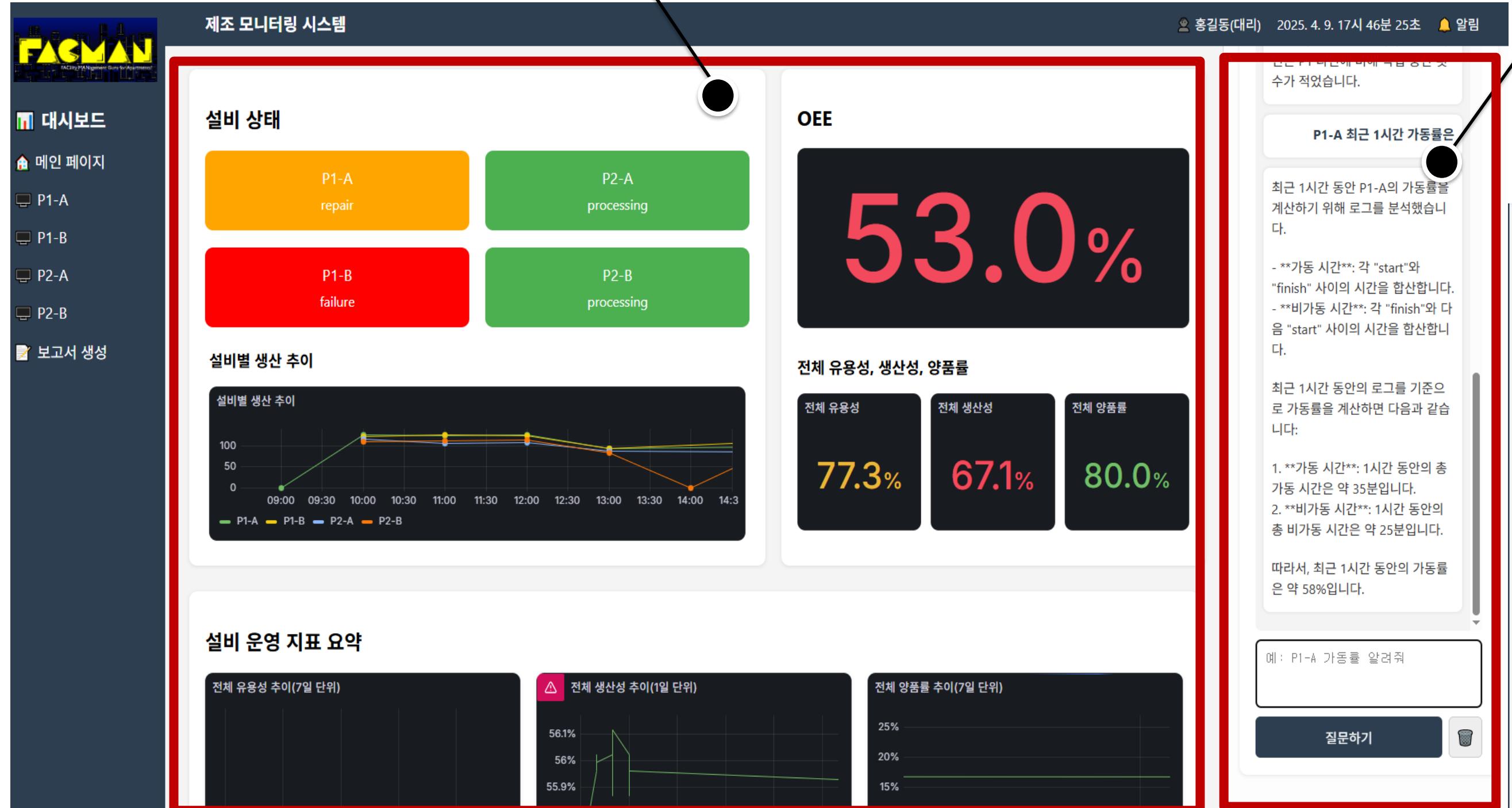


원자재 투입부터 생산 완료까지 전 공정 및 설비 데이터를 기반으로, AI Agent와 자동화 기술을 활용한 지능형 통합 운영관리 플랫폼 구현함

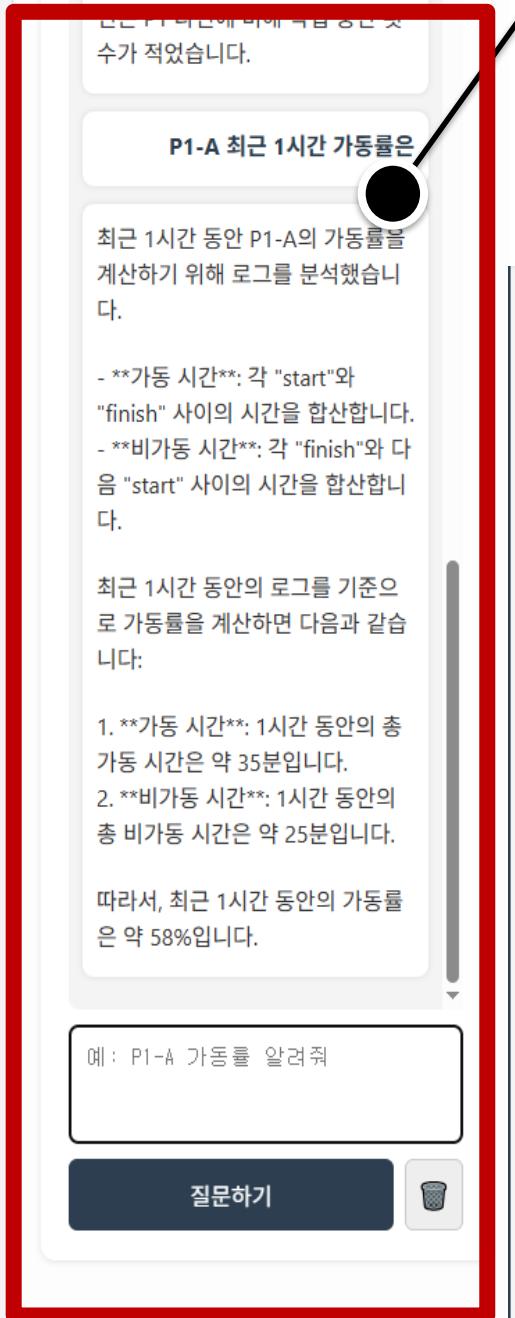




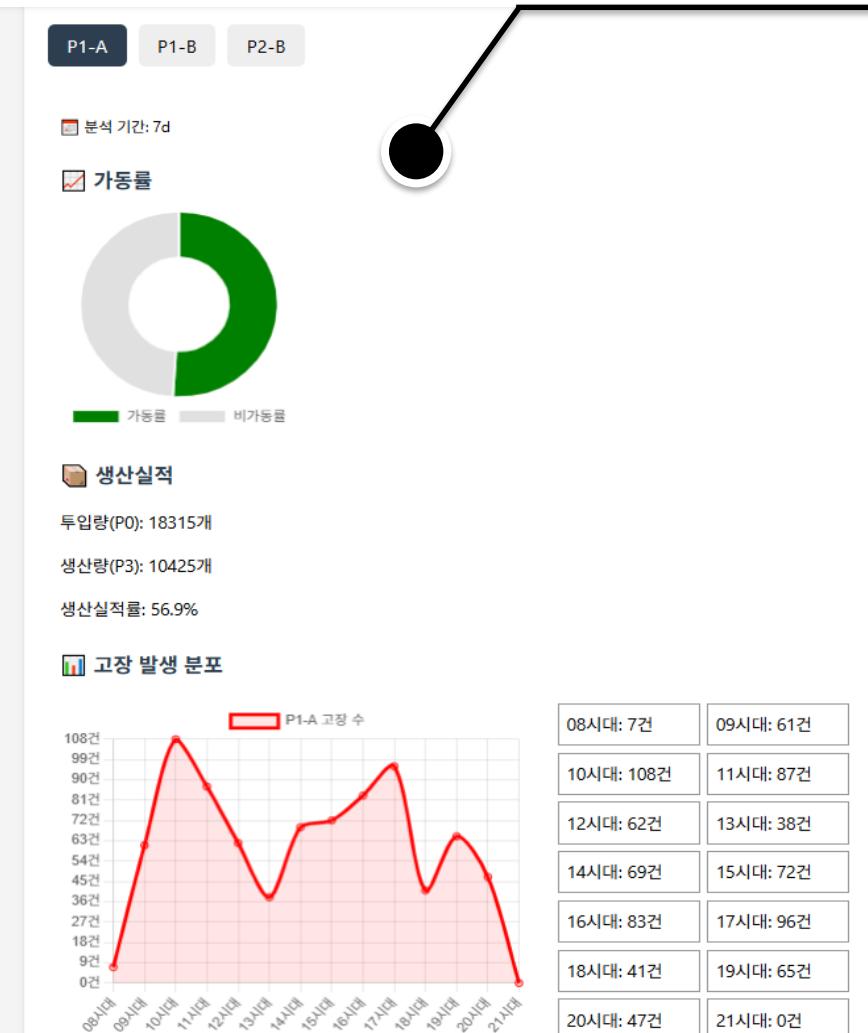
메인 대시보드



챗봇 서비스



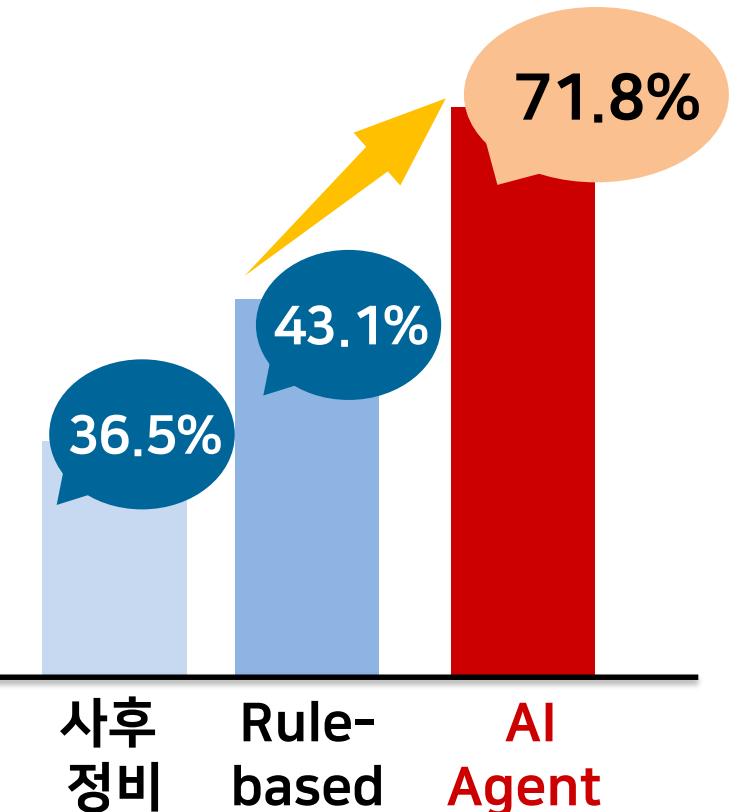
보고서 생성



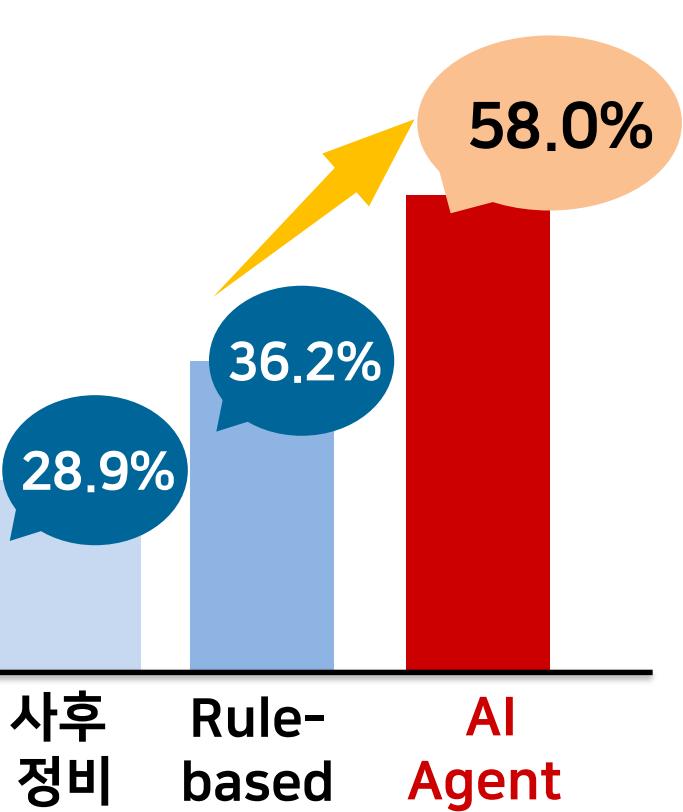
Agent 기능 도입을 통해 설비 운영 효율성 증가, 작업자 현장 대응 부담 감소, 현장 정보 공유 효율 등 주요 지표 전반에서 개선 효과를 확인함

정량적 기대효과

설비 가동률 향상



OEE 개선



- Agent 기반 예지 보전 도입 시, Rule-based 대비 가동률 **28.6%p** 향상

- Agent 기반 예지 보전 도입 시, Rule-based 대비 OEE **21.9%p** 향상

정성적 기대효과



현장 대응 자동화 및 관리자 개입 최소화

- 설비 고장 패턴을 조기 감지하여 실시간 현장 대응 및 조치 가능
- 예지보전으로 관리자의 불필요한 순회 점검, 현장 대응 횟수 감소



관리 업무 자동화 및 보고서 효율 향상

- 운영 KPI, 설비 로그, 이벤트 이력 등을 자동 집계하여 데이터 정합성이 확보된 자동 보고서 생성 가능
- 담당자는 중요 지표 분석과 의사결정 자료 작성에 집중



사용자 친화적 시각화·챗봇으로 정보 전달력 향상

- 직관적인 대시보드 UI로 전문지식 없이도 운영 상황 파악 용이
- 챗봇 기반 질의응답을 통해 데이터 해석 시간을 단축하고 필요한 정보를 즉시 전달

본 프로젝트 결과를 바탕으로, 현장 적용을 전제로 한 기능·구조·운영 측면의 확장 방향을 제시함

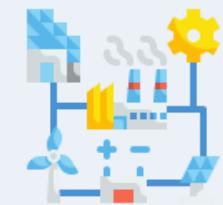
1. IoT 센서 데이터 인공지능 기반 고장 확률 예측

- **센서 기반 고장 예측**
 - 온도, 진동, 전류 등 센서 데이터를 활용해 설비 상태 변화 감지
- **AI 모델 적용**
 - 예측 모델로 고장 징후 조기 판단
- **이벤트 기반 사후보전에서 수치 기반 예지보전으로 전환**



2. 공정-자재-물류 연계형 멀티 Agent 확장

- **운영 기능별 Agent 확장**
 - 생산관리, 자재관리, 물류관리 등 각 영역을 담당하는 전용 Agent로 확대
- **Agent 간 분업과 협업 구조 설계**
 - 생산 일정 조정 Agent, 재고 예측 Agent, 입출고 자동 대응 Agent 등
- **공장 운영 전반의 지능화 기반 마련**
 - 공정 → 자재 → 생산 → 물류까지 연결된 AI 기반 관리 구조 구현



3. 프롬프트 엔지니어링 및 LLM 파인튜닝을 통한 응답 품질 향상

- **도메인 특화 프롬프트 설계**
 - 제조 현장에서 자주 사용하는 질문 유형과 표현 방식에 맞춰 응답 구조 최적화
- **LLM 파인튜닝 적용**
 - 실제 설비 로그와 운영 기록 기반으로 학습시켜, 전문성과 일관성 강화
- **실무 적합도 향상**
 - 반복되는 현장 질의에도 정확하고 일관된 대응 가능



4. 실제 데이터 기반 디지털 트윈 적용 방안

- **설비 시뮬레이션 환경 구축**
 - 실측 데이터를 기반으로 설비의 동작을 가상으로 재현
- **AI 판단 로직 사전 검증**
 - 고장 시나리오 반복 적용을 통해 Agent의 의사결정 정확도 검증
- **운영 테스트 환경 확보**
 - 실제 설비에 영향 없이 전략·시나리오별 테스트 수행 가능



Q&A



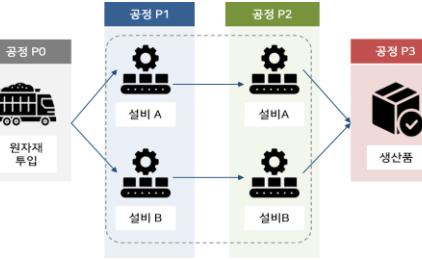
본 프로젝트에서는 2개의 공정, 2개의 라인을 운영중인 공장으로 가정했으며, 공장의 주요 성능 평가 지표에는 OEE의 항목을 활용함

데이터 정의서

<공장 정의>

- **2개의 공정 (P1, P2)** : 평균 작업시간: **10초**, 평균 고장간격: **5분**, 평균 수리시간: **1분**, 평균 점검시간: **15초**
- **2개의 라인 (A, B)**

bucket	measurement	field	세부내용
Status	status_log	available	0,1
		event_type	processing, repair, maintenance, failure
Process	process_log	event_status	processing repair maintenance failure
		process_status	producer(P0) relay(P1-A,P2-A,...) consumer(P3)



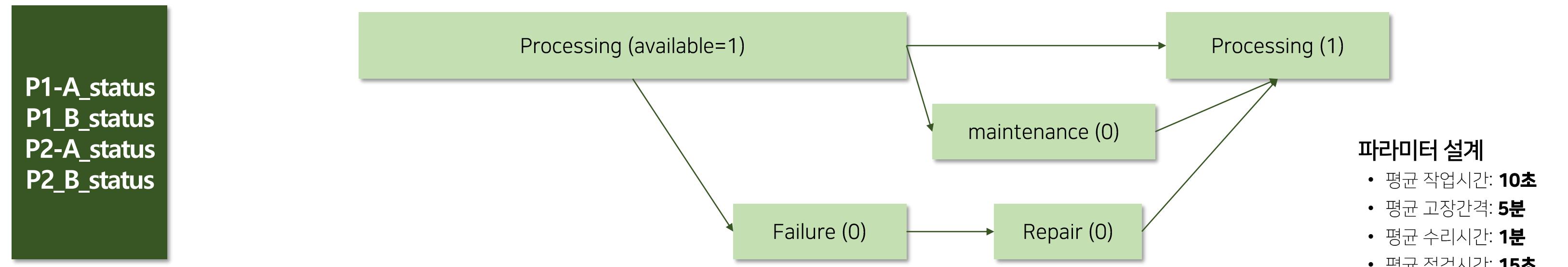
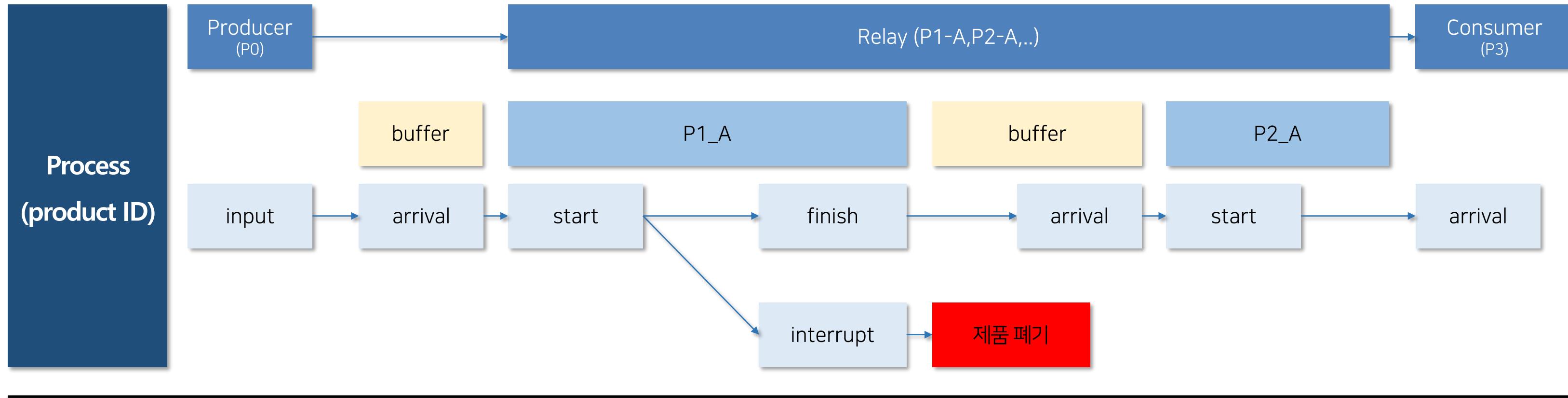
주요 성능 평가 지표

"OEE" (*Overall Equipment Effectiveness, 설비 종합 효율*)
 $= Availability(\text{유용성}) * Productivity(\text{생산성}) * Quality(\text{양품률})$

Availability (유용성)	계획가동시간(9) - 정지 시간 / 계획가동시간(9) $= \frac{\text{계획가동시간(9)}}{\text{계획가동시간(9)} - \text{정지 시간}}$
Productivity (생산성)	총 생산 수량 * 기준 사이클 타임 / 실제 가동시간 $= \frac{\text{Count(process_id=P0)} * 20}{\text{계획가동시간(9)} - \text{정지 시간}}$
Quality (양품률)	생산량 - 불량수량 / 생산량 $= \frac{\text{Count(process_id=P3)}}{\text{Count(process_id=P0)}}$

Appendix 데이터 Flow

제품 단위 공정 흐름과 설비 상태 변화 구조를 기반으로, 생산·버퍼·고장·수리까지 전 과정의 시뮬레이션 데이터 흐름 구조를 정의함



Appendix 서비스 아키텍처 구성도

실시간 공정 데이터를 기반으로 AI 분석 및 판단 시스템을 설계하여, 공장 운영의 자동화와 신속한 의사결정이 가능한 구조로 구성

시스템 구성도



주요 기능

1 설비 예지보전 Agent

- 설비 상태와 예지보전 로그 기반으로 자동 수리 일정 시점 산정
- 공정 간 병목 구간을 분석해 버퍼 조정 및 설비별 가동률 계산
- 생산량 예측 데이터를 활용한 설비별 생산 계획 생성

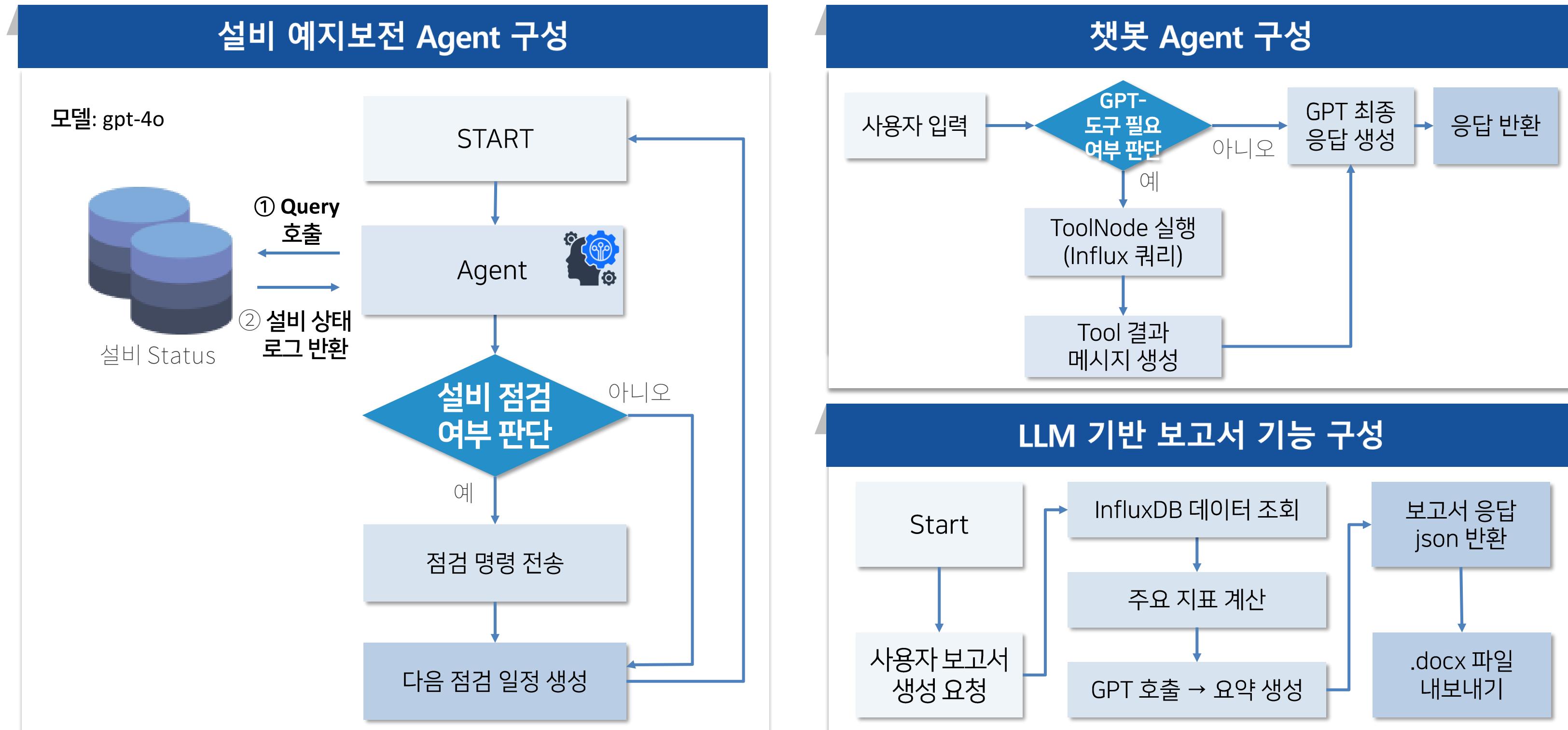
2 챗봇 서비스

- 공정/설비 관련 질문에 대해 자연어 기반 응답 및 연관 데이터 요약
- 기간별 생산 실적, 이벤트 로그 등 주요 지표 질의 응답 처리
- 질의 의도 분석과 자동 응답 구성을 포함한 챗봇 Agent 처리

3 보고서 서비스

- InfluxDB 기반 로그를 자동 수집하여 보고서 자동 생성 및 요약
- 사용자 정의 항목 설정 및 시각화 포함 리포트 구성 기능 제공
- .docx 파일 내보내기 기능으로 수정 보완 가능

InfluxDB 기반의 실시간 로그 데이터를 GPT 모델과 연계하여, 설비 진단, 사용자 요청 대응, 주요 지표 기반 보고서 생성을 자동화하는 모듈형 아키텍처로 구성함



End of Document

