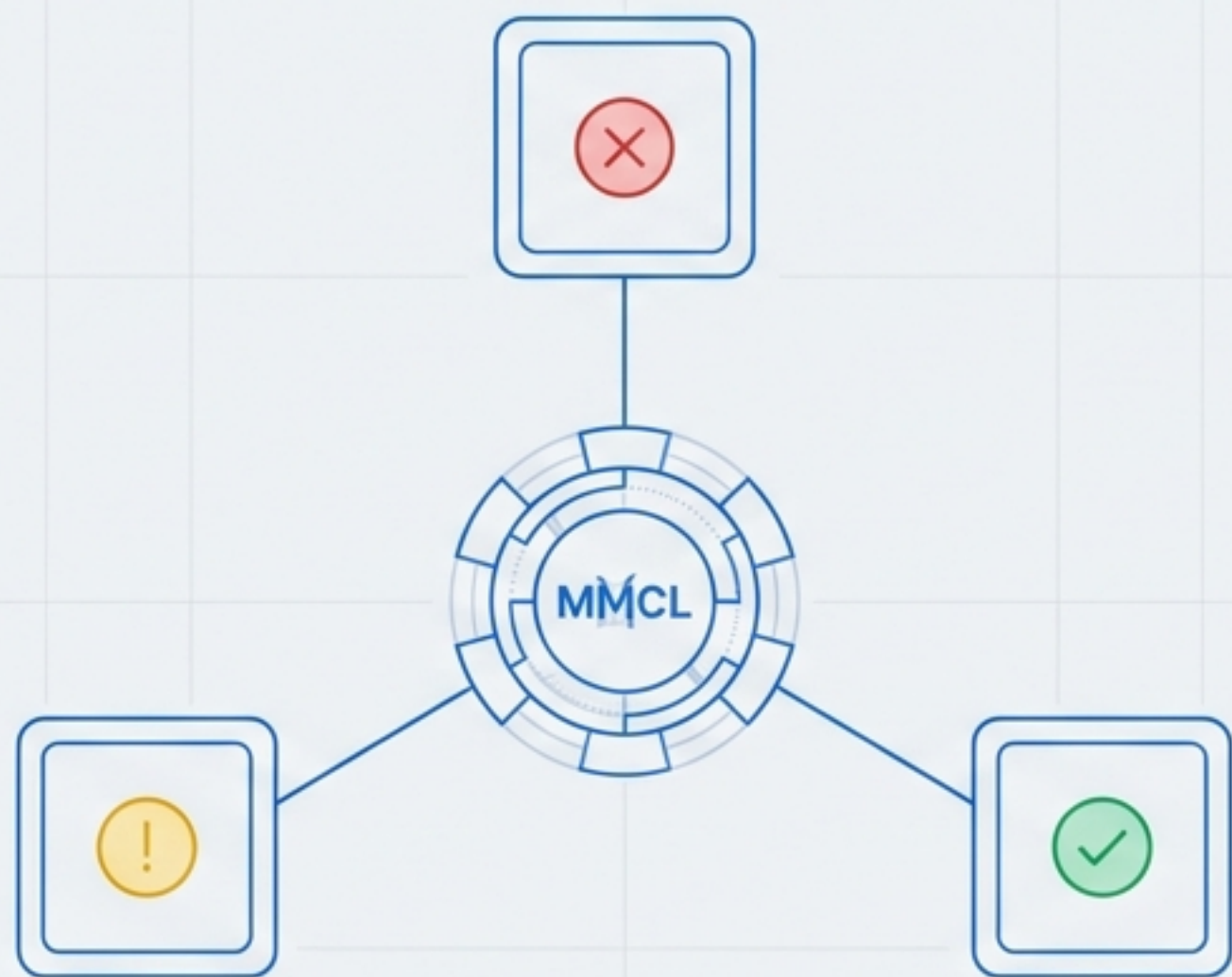


# MMCL (Machine Monitoring Control for LLM) Mockup

자연어 기반 장비 모니터링 및 상태 제어 PoC 아키텍처



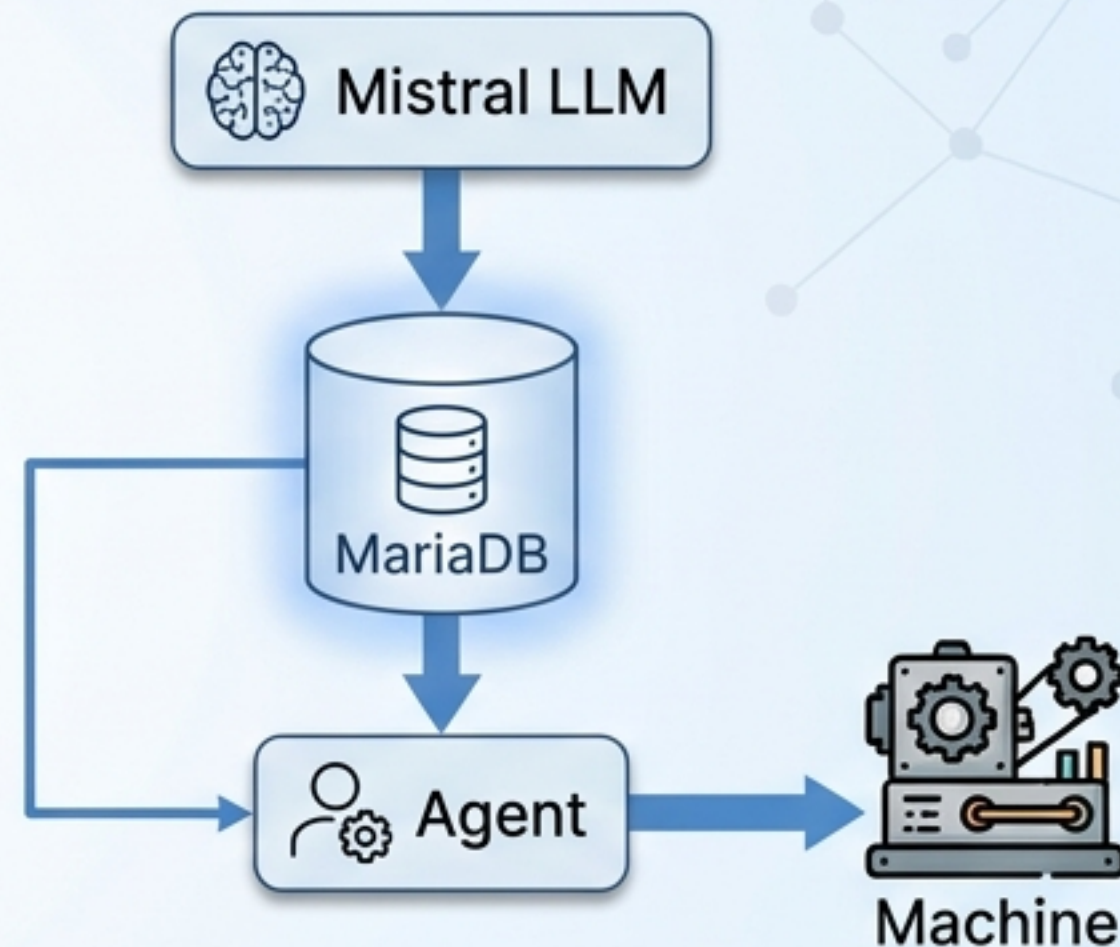
# 물리적 제어의 위험성을 제거한 '상태 동기화' 시뮬레이션

기존 방식: 하드웨어 직접 제어 (배제)



선형적이고 위험한 직접 제어 구조

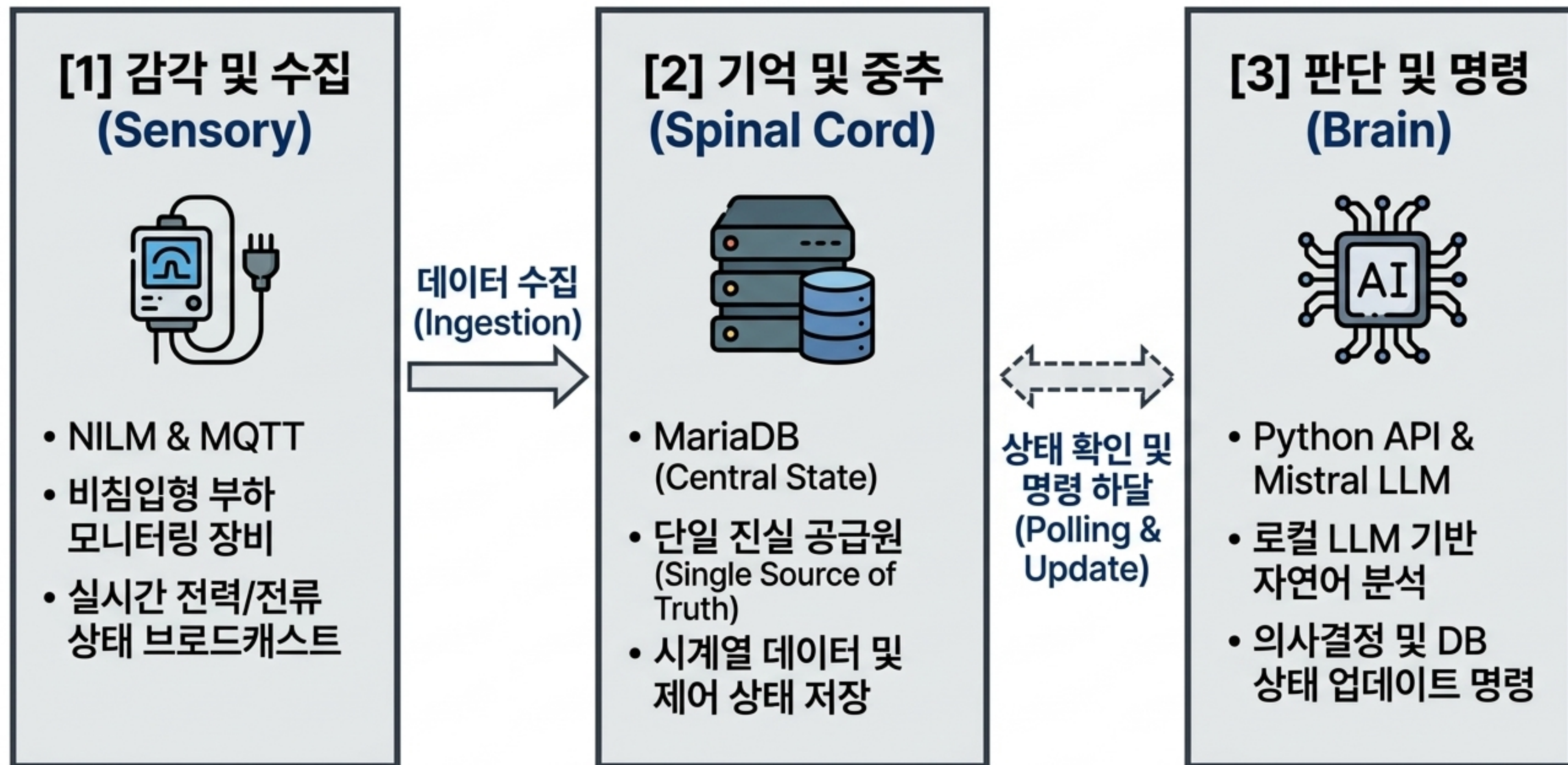
MMCL 컨셉: DB 상태 필드 조작



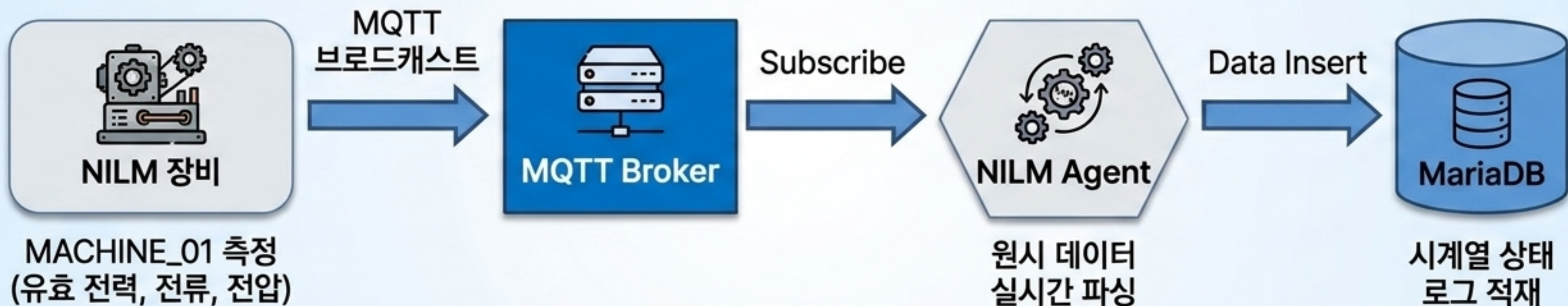
데이터베이스 상태값을 통한 안전한 간접 제어

본 PoC는 장비 직접 제어의 불안정성을 배제하고, 중앙 데이터베이스(MariaDB)를 통한 '상태 동기화'로 제어를 시뮬레이션하는 안전한 아키텍처를 검증합니다.

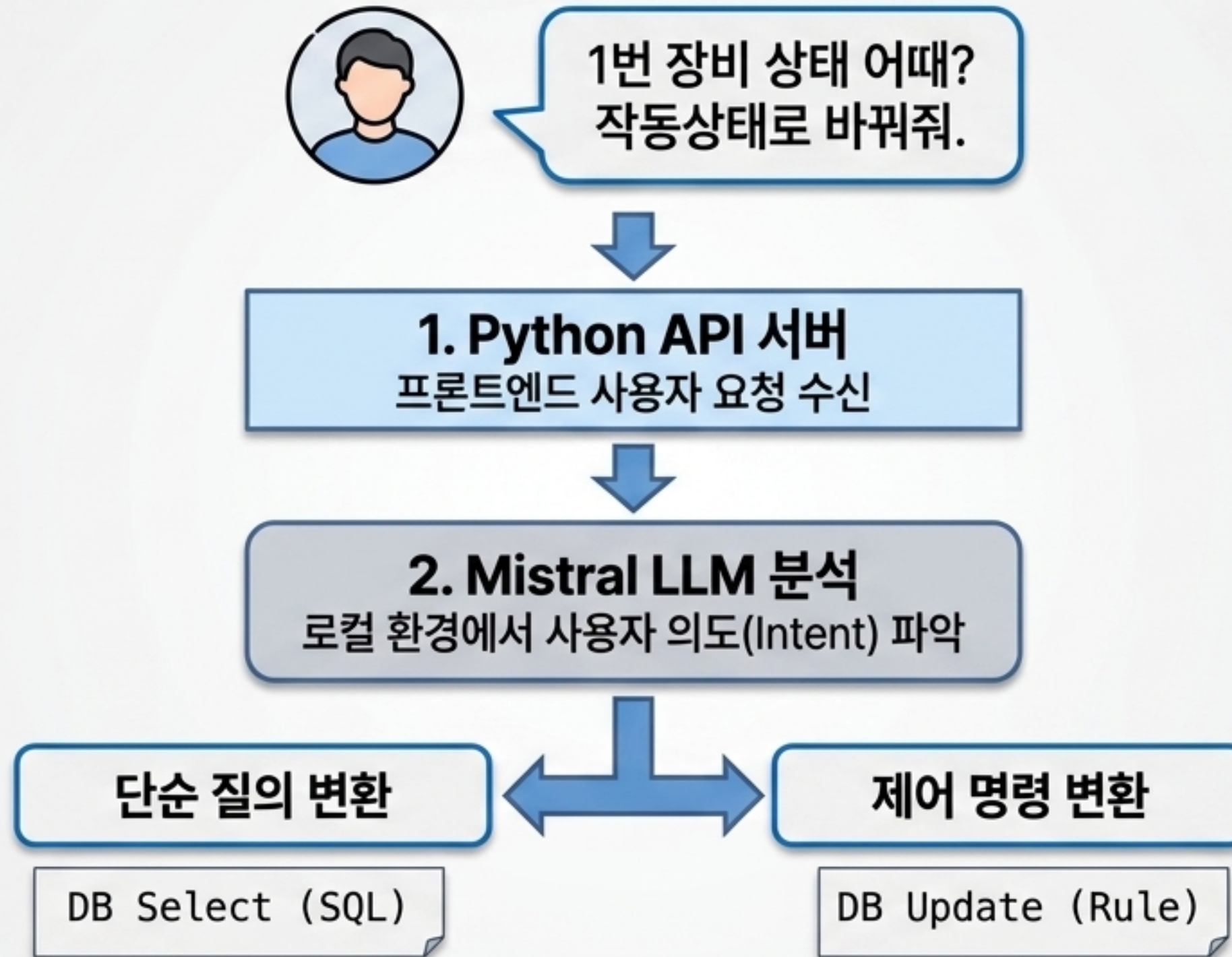
# MMCL 3축 신경망 아키텍처



# [Phase 1] 데이터 수집: NILM 장비와 에이전트의 실시간 연동

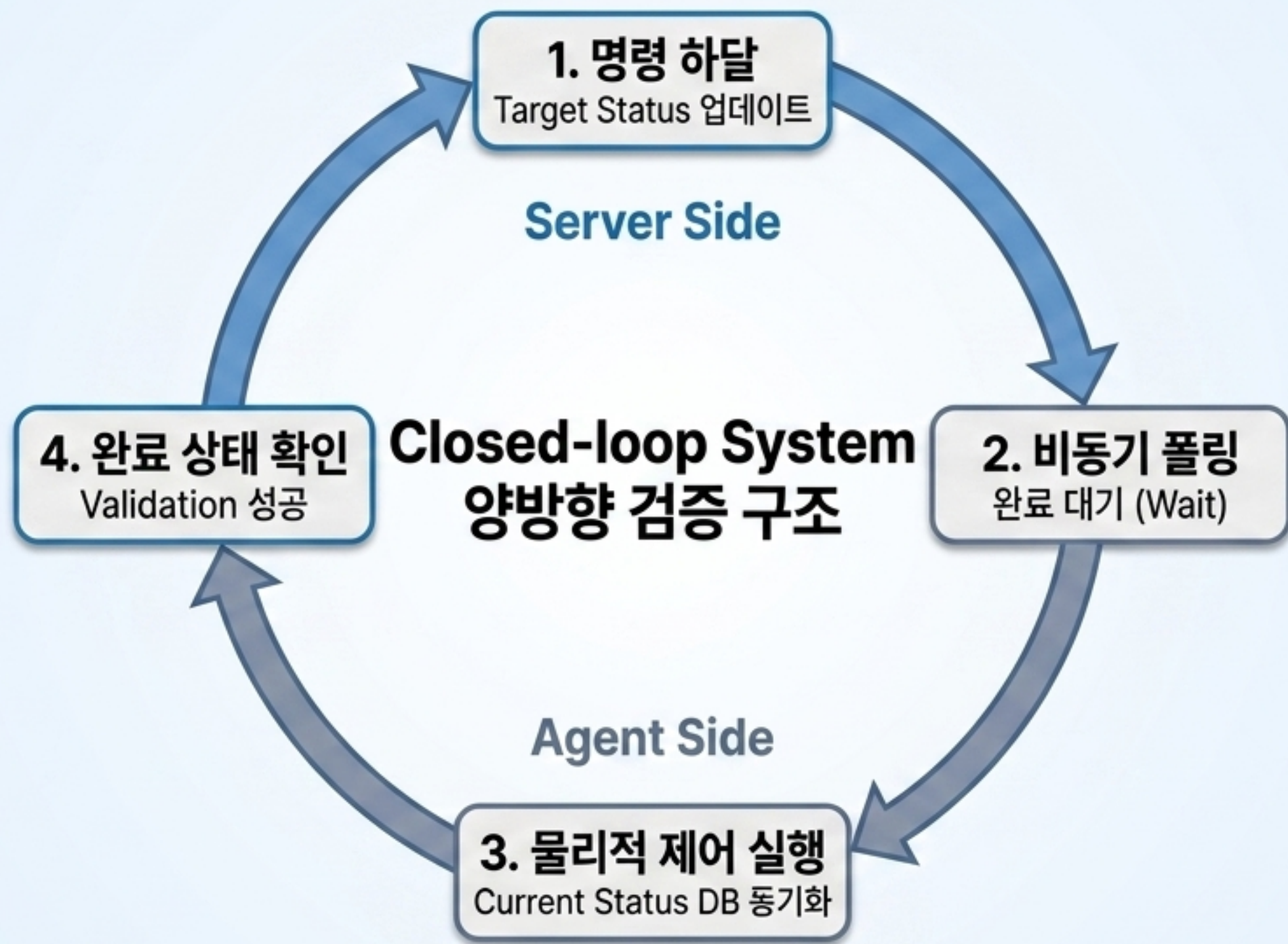


# [Phase 2] 지능형 제어: 자연어 질의 SQL 및 룰 변환



LLM은 하드웨어 통신 프로토콜을 알 필요가 없습니다. 모든 판단은 데이터베이스를 향한 SQL 명령으로 치환됩니다.

# 핵심 혁신: 핸드셰이크 상태 검증 (Handshake Validation)



제어 명령은 즉시 성공으로 처리되지 않습니다. 백엔드가 DB에 목표를 설정한 후 대기하며, 에이전트가 물리적 제어 완료 후 DB를 업데이트해야만 비로소 '정상 완료'로 판단합니다.

# 중앙 상태 저장소 (Single Source of Truth) 설계

## 테이블 1: 시계열 모니터링 [nilm\_status\_log]

목적: NILM 장비 데이터 실시간 적재

- timestamp
- device\_id
- active\_power
- current
- voltage

## 테이블 2: 경광등 상태 제어 [warning\_light\_control]

목적: Agent와 Server 간의  
핸드셰이크 상태 동기화

- target\_status (목표)
- current\_status (현재)

상태값 매핑 (Functional Indicators)



제어 명령은 즉시 성공으로 처리되지 않습니다. 백엔드가 DB에 목표를 설정한 후 대기하며, 에이전트가 물리적 제어 완료 후 DB를 업데이트해야만 비로소 '정상 완료'로 판단합니다.

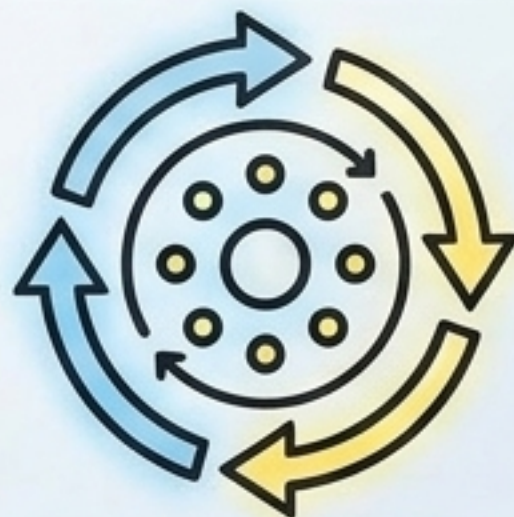
# 제어 상태 전이도 (State Transition Logic)

## 상태 A: Pending (진행 중)

target\_status (● GREEN)

≠

current\_status (● YELLOW)



시스템이 명령을 내렸으나, 에이전트가 아직 하드웨어에 적용하지 않은 대기 상태.

## 상태 B: Completed (완료)

target\_status (● GREEN)

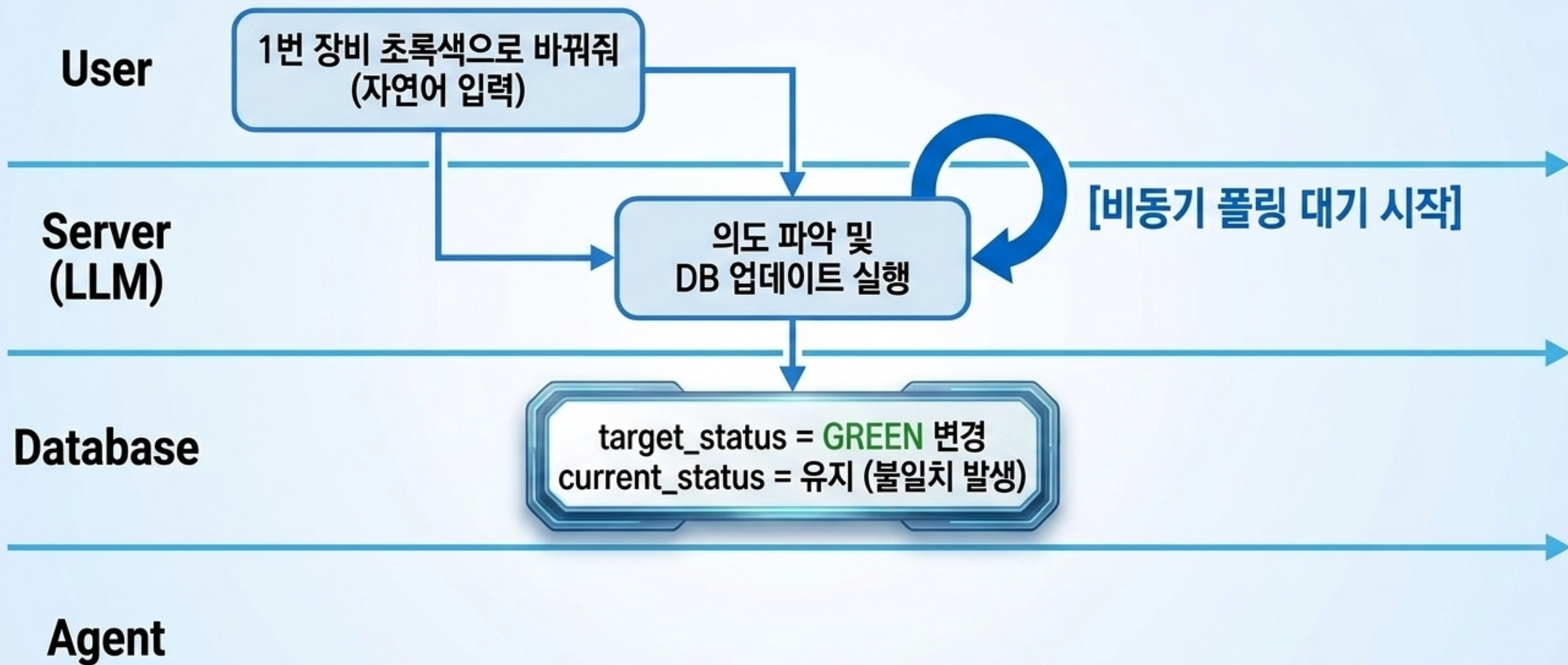
==

current\_status (● GREEN)

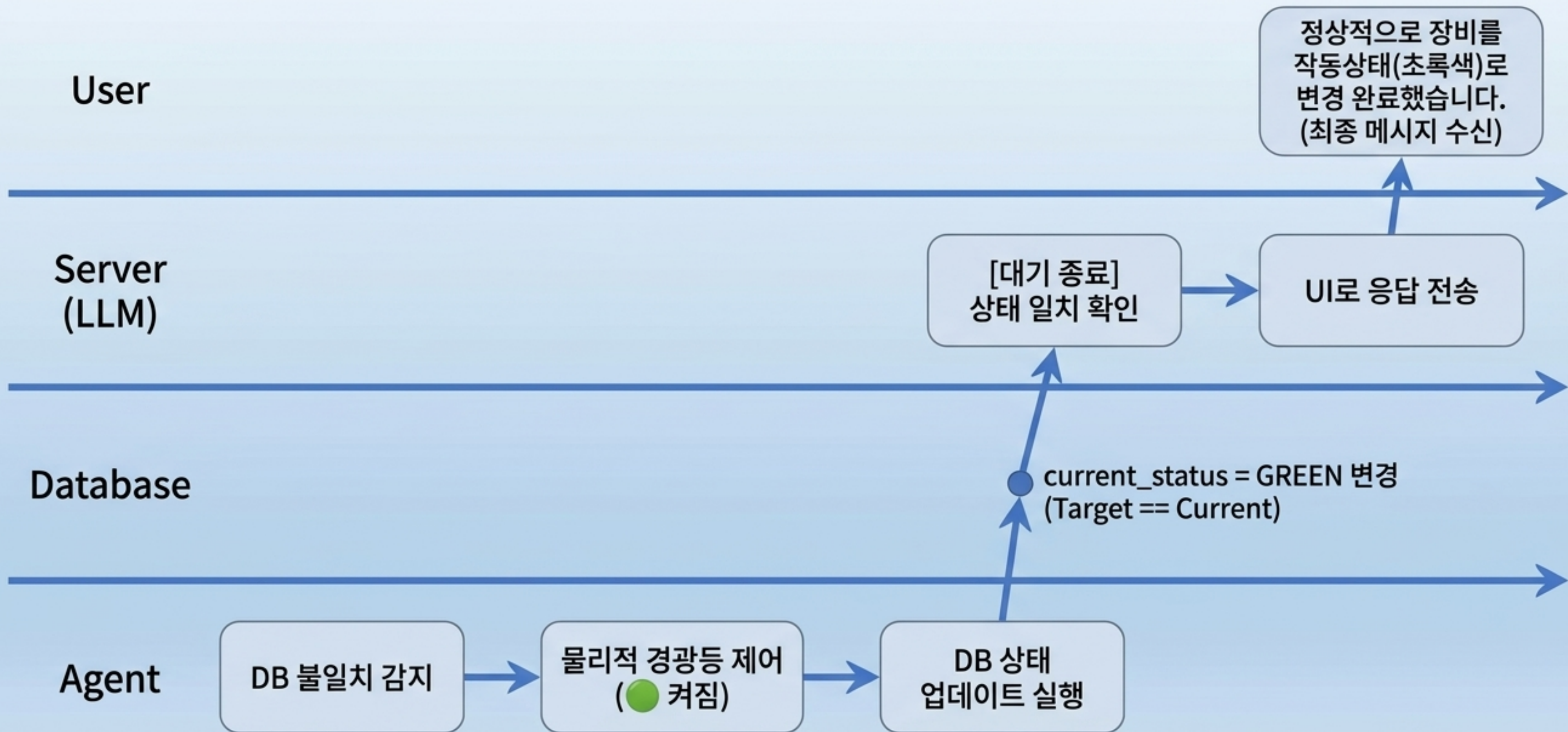


물리적 제어가 성공적으로 완료되어 데이터베이스 상태가 완벽히 동기화된 상태.

# User Story Timeline (1/2): 명령 하달 및 폴링 대기



## User Story Timeline (2/2): 물리 제어 및 최종 피드백

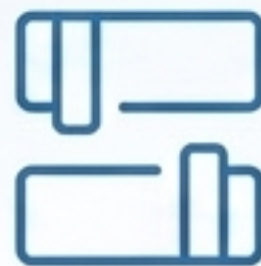


# 기대 효과: 완벽하게 분리된(Decoupled) 지능형 제어망



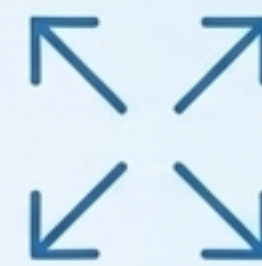
## 1. 안전성 (Safety First)

LLM의 직접 제어  
오류(Hallucination)가  
물리적 장비 손상으로 이어지지  
않도록 DB가 완충 지대(Buffer)  
역할을 수행합니다.



## 2. 디커플링 (Decoupling)

서버(지능)와 에이전트(실행)가  
완벽히 분리되어, 추후 장비  
프로토콜이 변경되더라도 LLM  
로직은 수정할 필요가 없습니다.



## 3. 확장성 (Scalability)

MariaDB 상태 필드만 확장하면  
수천 대의 장비와 다양한 상태를  
동일한 아키텍처로 제어 가능한  
기반을 제공합니다.

