10th Anniversary

GISC 2026

Global ICT Standards Conference 2026

피지컬 AI ICT 융합 표준프레임워크 공청회

이종 다수 로봇의 협업을 위한 ROS2 네이티브 Agentic 런타임 개발 및 실증

2026. 11. 03

구 세 완

LG전자

ICT Standards and Intellectual Property:
Shaping the Physical AI

주최 과학기술정보통신부 지식재산처 주관 국립전파연구원 TTA IITP KEA kista ETRI

Index

01 Introduction

02 Research Trends

03 Development of Edge Agentic AI Systems for Autonomous Robot Control

04 Future Plans

국가연구개발사업으로 부터 지원

[연구과제명] **일상생활 공간에서 자율행동체의 복합작업 성공률 향상을 위한 자율행동체 엣지 AI SW 기술 개발**

- 세부과제: **기밀성 데이터 활용 지능 고도화를 위한 엣지와 클라우드 분산 협업 학습 프레임워크 기술**

[과제고유번호] RS-2024-00346798

[부처명] 과학기술정보통신부

[연구관리전문기관] 정보통신기술진흥센터

[연구사업명] 정보통신 방송 연구개발 사업

[주관기관] 한국전자기술연구원

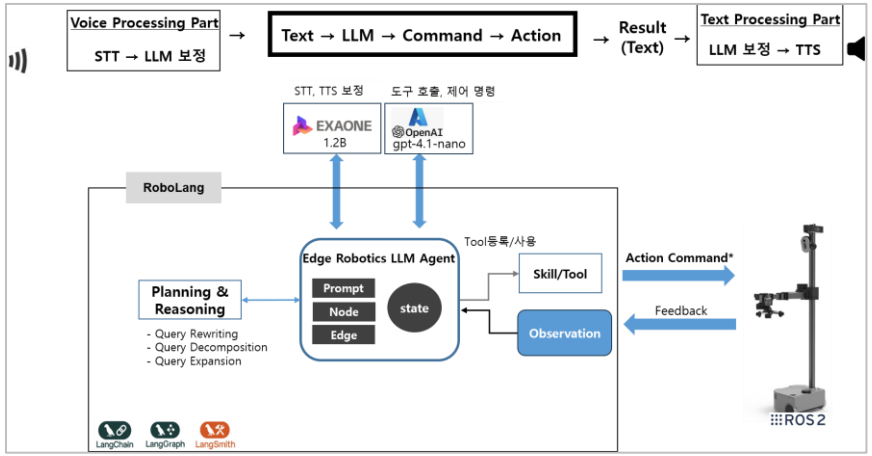
[연구기간] 2024.04.01-2027.12.31

[참여기관] 한국전자기술연구원, 한국전자통신연구원, 이노그리드, 동의대학교, 광주과학기술원, LG전자

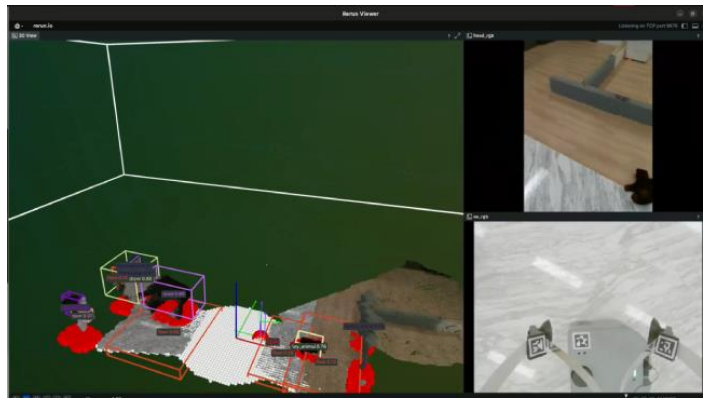
Agentic AI System Related Development (2025~2026)

LLM Driven Control for Robotics (2025)

RoboLang AI Agent

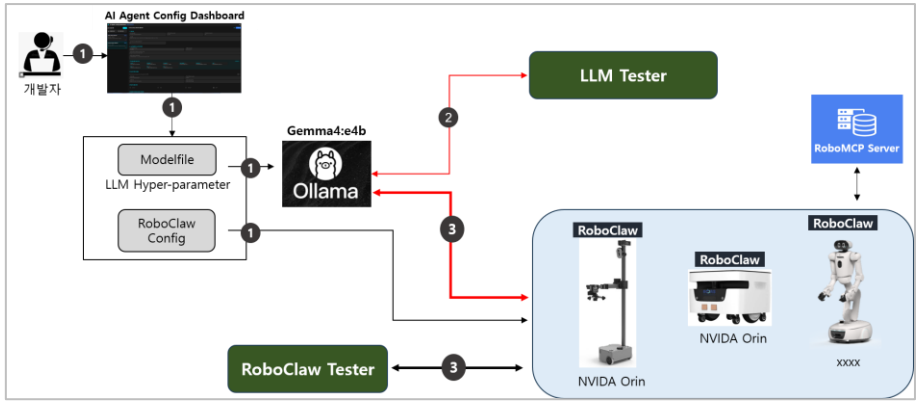


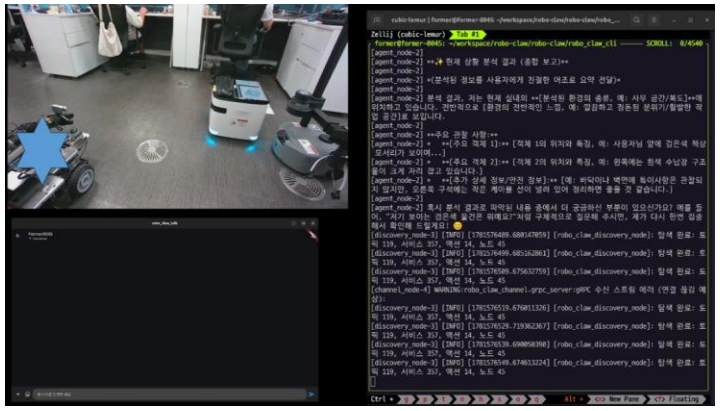




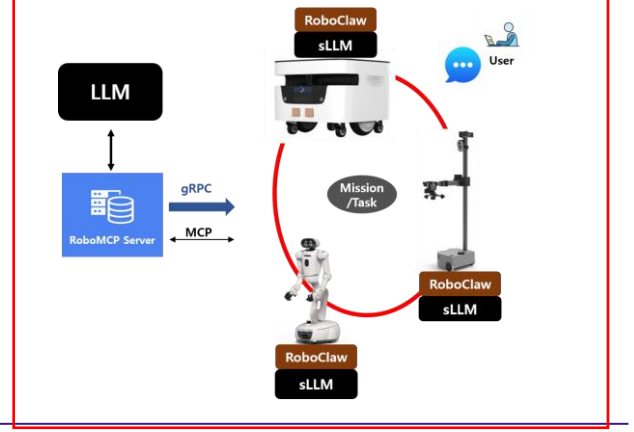
AI Agentic System for Autonomous Robotics (2026)

RoboMCP Agentic AI



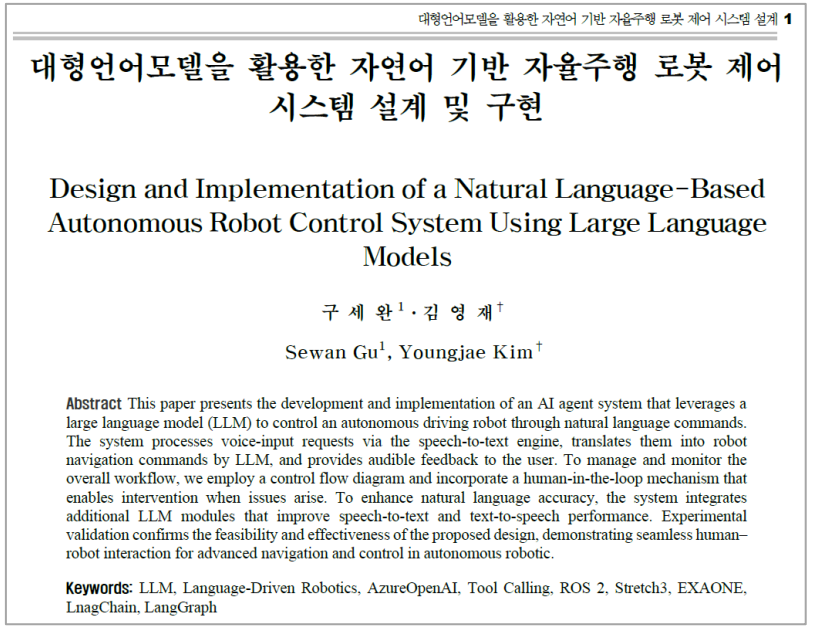


RoboClaw Agentic AI



금일 발표 내용

LLM기반의 자율주행 로봇 제어



대형언어모델을 활용한 자연어 기반 자율주행 로봇 제어 시스템 설계 1

대형언어모델을 활용한 자연어 기반 자율주행 로봇 제어 시스템 설계 및 구현

Design and Implementation of a Natural Language-Based Autonomous Robot Control System Using Large Language Models

구 세 완[1] · 김 영 재[†]

Sewan Gu[1], Youngjae Kim[†]

Abstract This paper presents the development and implementation of an AI agent system that leverages a large language model (LLM) to control an autonomous driving robot through natural language commands. The system processes voice-input requests via the speech-to-text engine, translates them into robot navigation commands by LLM, and provides audible feedback to the user. To manage and monitor the overall workflow, we employ a control flow diagram and incorporate a human-in-the-loop mechanism that enables intervention when issues arise. To enhance natural language accuracy, the system integrates additional LLM modules that improve speech-to-text and text-to-speech performance. Experimental validation confirms the feasibility and effectiveness of the proposed design, demonstrating seamless human–robot interaction for advanced navigation and control in autonomous robotic.

Keywords: LLM, Language-Driven Robotics, AzureOpenAI, Tool Calling, ROS 2, Stretch3, EXAONE, LnagChain, LangGraph

로봇학회논문지, vol21, no2, pp. 131-139

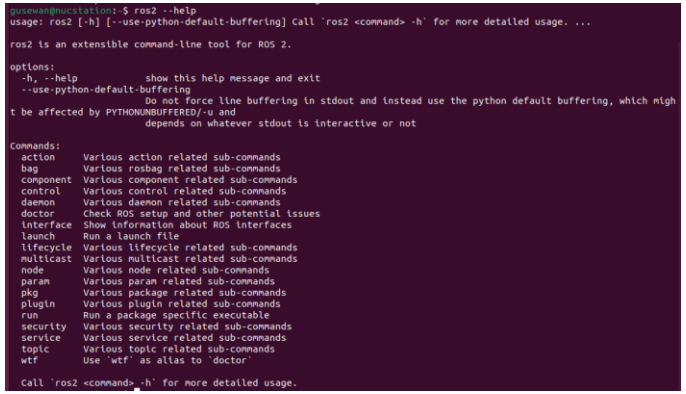
AI Agent

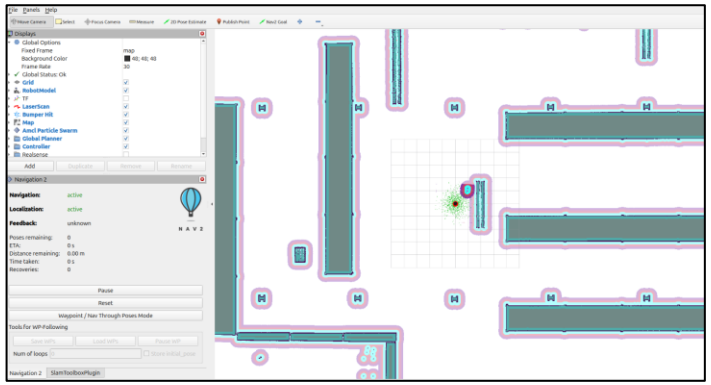
Agentic AI

-생각지 못한
-학습하지 못한 **환경**
-새로운 변수를 갖는

자율주행 로봇 제어

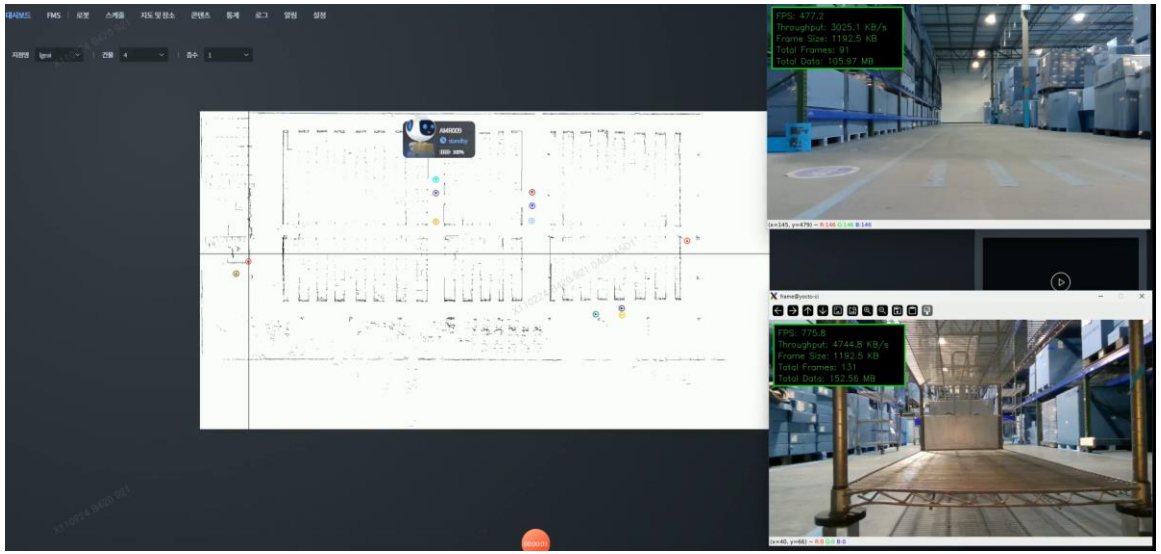
- Command-based Monitoring & Control (Expert Tool)





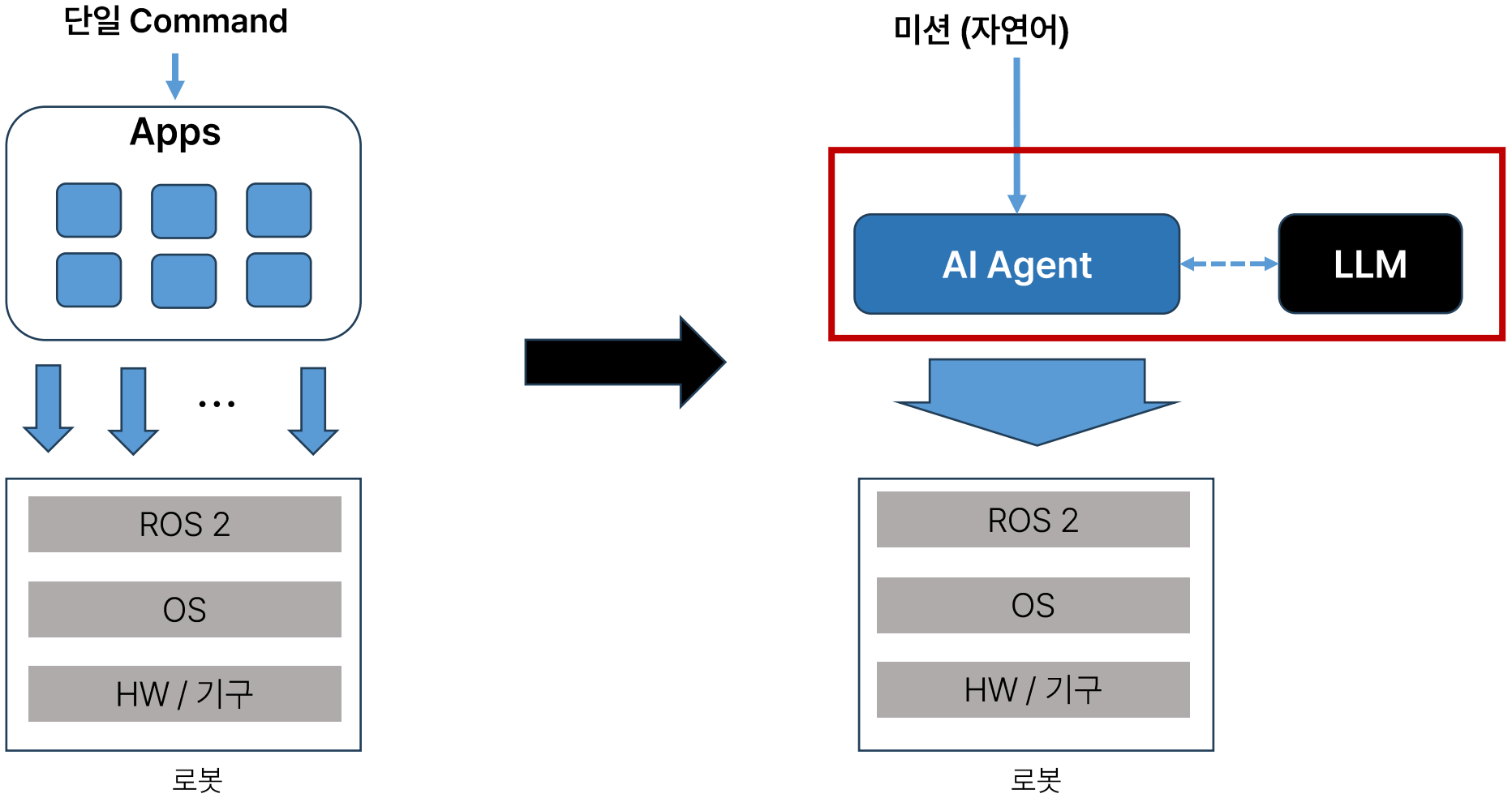


- Rule-based Monitoring & Control Dashboard





로봇 제어 방법 변화


단일 Command
Apps
...
ROS 2
OS
HW / 기구
로봇
미션 (자연어)
AI Agent
LLM
ROS 2
OS
HW / 기구
로봇

LLM 기반 로봇 제어 관련 최신 연구 동향

[1] ChatGPT for Robotics: Design Principles and Model Abilities (**Microsoft**), 2023

: LLM이 로봇 제어 코드를 작성하거나 API를 호출하도록 하는 연구

[2] Do As I Can, Not As I Say : Grounding Language in Robotic Affordances (**Google**), 2022

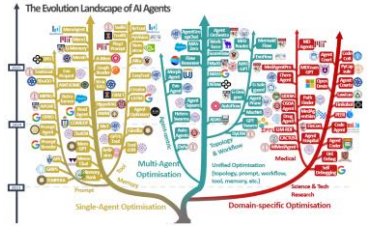
: 로봇의 물리적 제약 가능성(Affordance)과 LLM의 추론(Reasoning)을 결합한 Embodied AI 연구

[3] NavGPT : Explicit Reasoning in Vision-and-Language Navigation with Large Language Models, 2024

[4] ROS-LLM : A Framework for Embodied AI, 2025

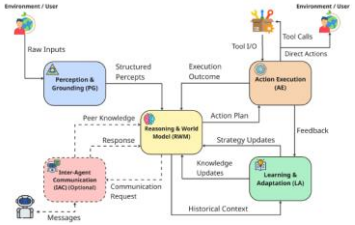
AI Agent 기반 로봇 제어 관련 최신 연구 동향

[1] A Comprehensive Survey of Self-Evolving AI Agents,
A New Paradigm Bridging Foundation Models and Lifelong Agentic Systems, 2025



AI Agent 발전 방향 트리

[2] Agentic Design Patterns: A System-Theoretic Framework, 2025



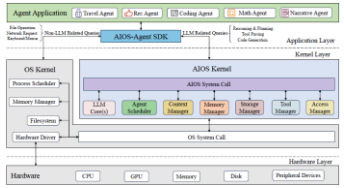
AI Agent Core Subsystem

[3] Agentic LLM-based robotic systems for real-world applications: a review on their agenticness and ethics, 2025



AI Agent의 로봇 적용 논문들
광범위한 소개

[4] AIOS: LLM Agent Operating System, 2025



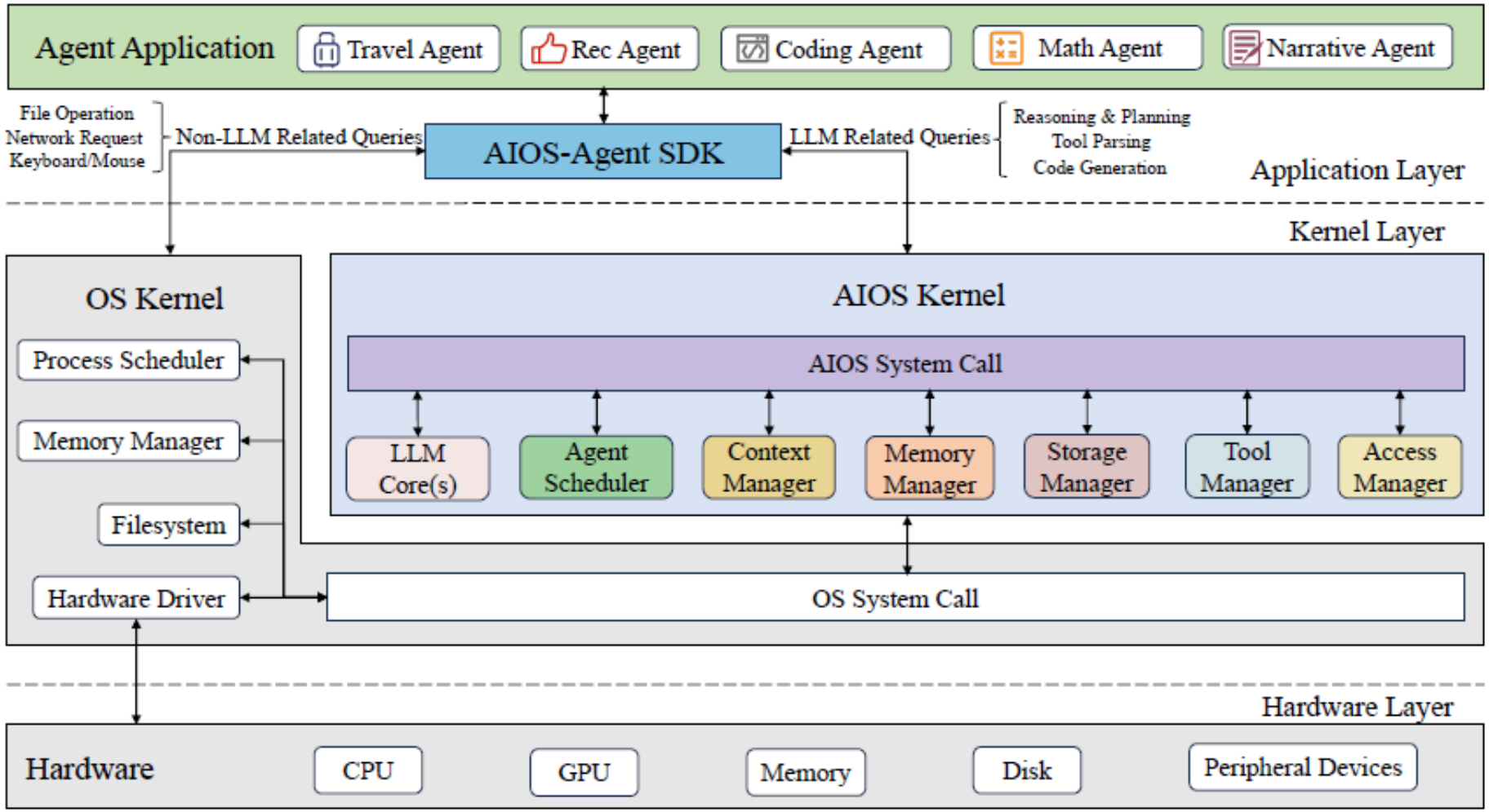
OS Kernel → AIOS Kernel
→ AIOS-Agent SDK
→ Agent Application

[5] Towards Embodied Agentic AI: Review and Classification of LLM- and VLM-Driven Robot Autonomy and Interaction, 2025

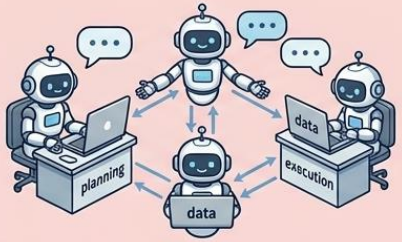
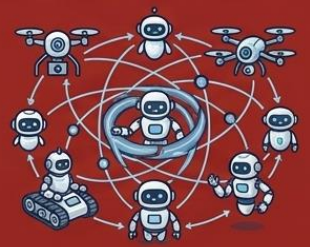
Agentic LLM-based robotic systems for real-world applications: a review on their agenticness and ethics (2025)



AIOS: LLM Agent Operating System, 2025



Agent 발전 방향

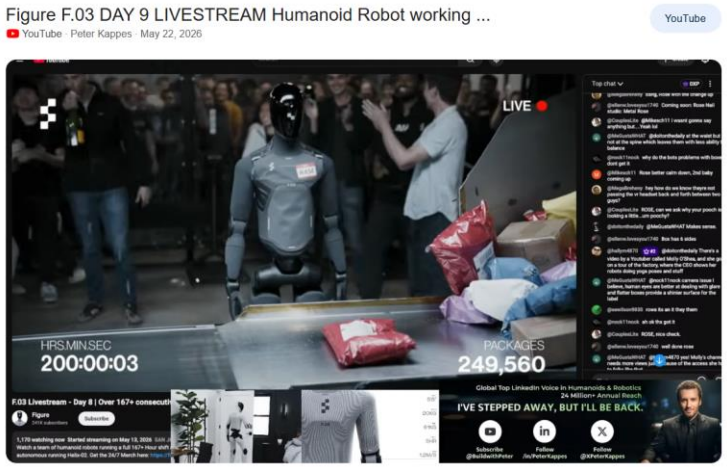
	1단계: 개별적 AI 에이전트	2단계: 에이전트 간 협업 (Co-op AI)	3단계: 자율 협업 에이전틱 시스템 (Agentic AI)
Agentic AI 발전 단계	지능 및 자율성 증가		
	목표를 가진 독립적 수행자	목표 달성을 위한 동료 에이전트와 연합	스스로 팀을 구성하고 문제를 해결하는 군집 지능
			
	사용자가 정의한 단일 목표 수행 도구 사용 및 의사결정 기능 탑재 예시: 자동화 리서치 에이전트	역할 분담 및 정보 공유를 통한 협력 사람의 역할은 전략 및 목표 제시로 축소 예시: 다층적 프로젝트 관리 에이전트 팀	상위 목표만으로 자율적 하위 작업 분할 및 실행 시스템 자가 진단 및 지속적 학습/최적화 예시: 완전 자율 운영 '미래 부서' AI 시스템

*AI 생성 컨텐츠

공개 도구	LangChain/LangGraph MCP (Model Context Protocol)	OpenClaw	Hermes Agent
로봇	- 단일작업 실행 - 멀티로봇 관제 필요	- 복합작업 실행 - 로봇간 협업	- 미션만으로 실행 - 미션이 다른 멀티로봇 적용
개발	RoboLang / RoboMCP	RoboClaw	

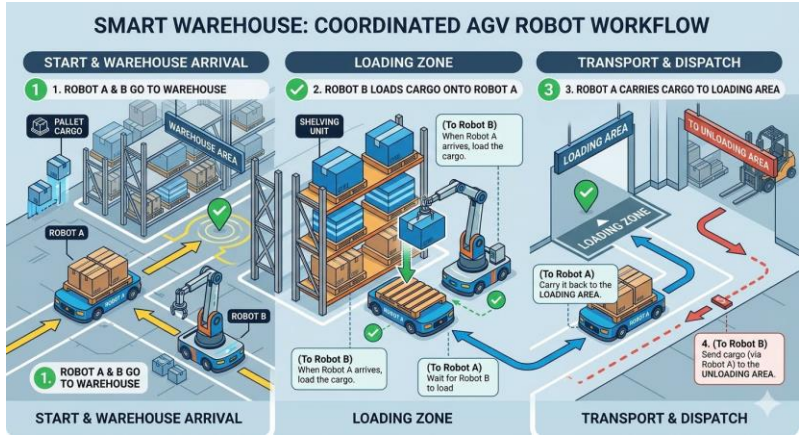
Agent 발전 방향에 따른 시나리오

1단계: 개별적 AI 에이전트	2단계: 에이전트 간 협업 (Co-op AI)	3단계: 자율 협업 에이전틱 시스템 (Agentic AI)
- 창고에 가서 30초 대기하고, 하역장으로 돌아와 - 택배 정리 후 배터리 충전	1. (로봇 A에게) 창고로 이동해서 로봇B가 짐을 싣으면 그걸 싣고 거실로 와 2. (로봇 B에게) 창고에 가서 로봇 A 가 도착하면 짐을 싣어서 하역장으로 보내줘	- 창고에 있는 짐을 하역장으로 옮겨줘









Model to Agent

- Model

```
Output = model (prompt)
```

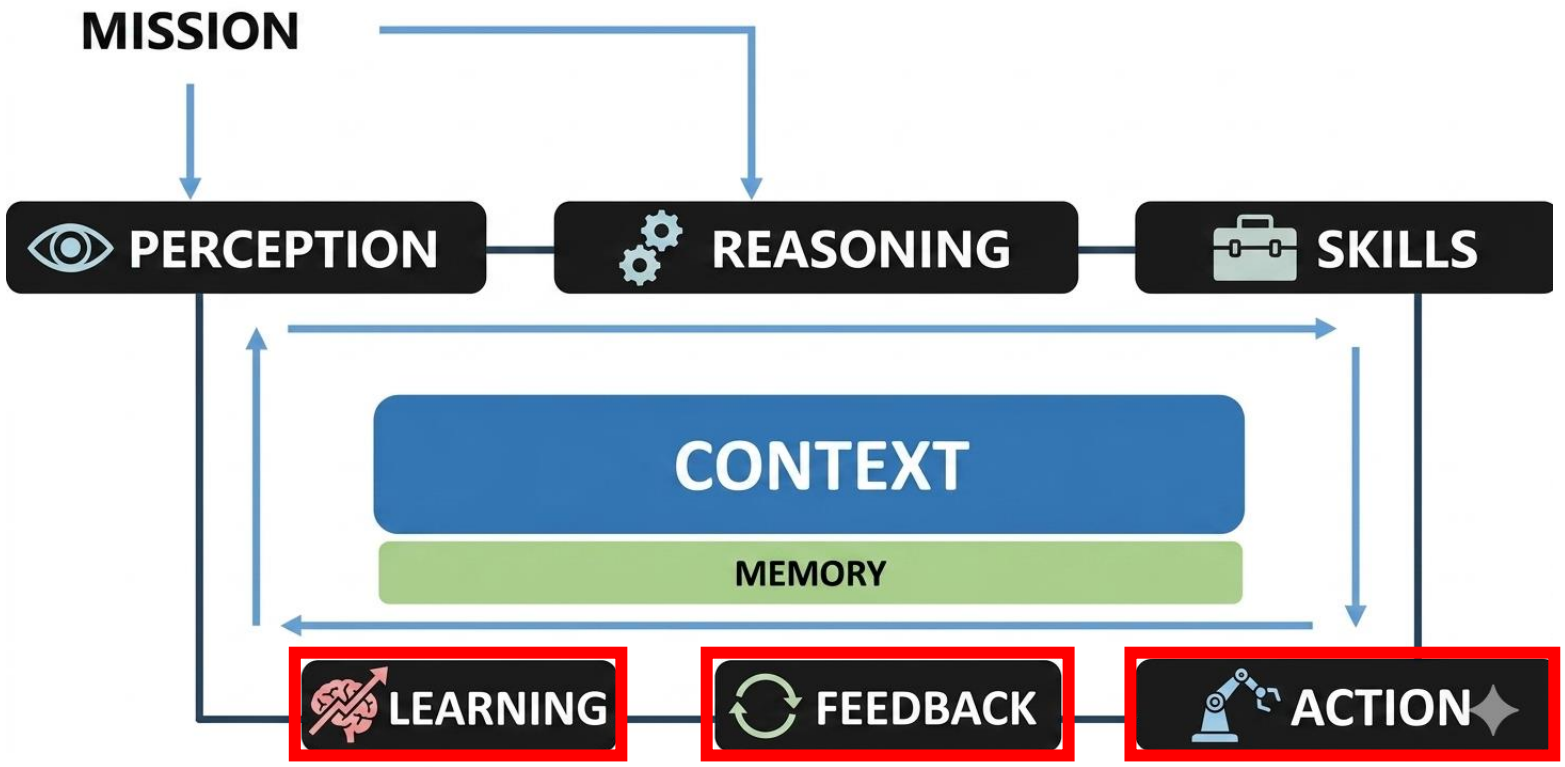
- Agent

```
Memory = prompt
While not is_task_done() :
    response = model (memory)
    memory = update_memory(memory, response)
```



Robotics AI Agent 개요

- Context Management를 LLM기반의 AI Agent가 담당

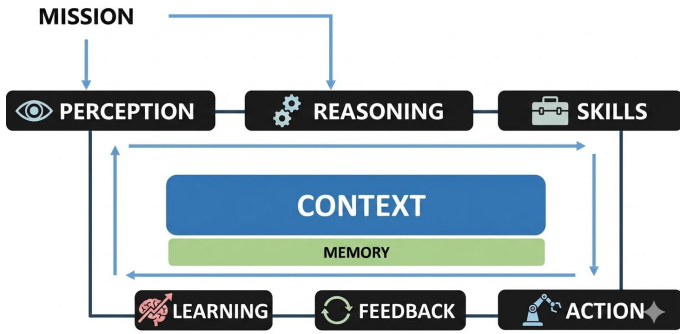


"Physical AI"

자연어 기반의 로봇 제어 흐름

- 구현 시나리오

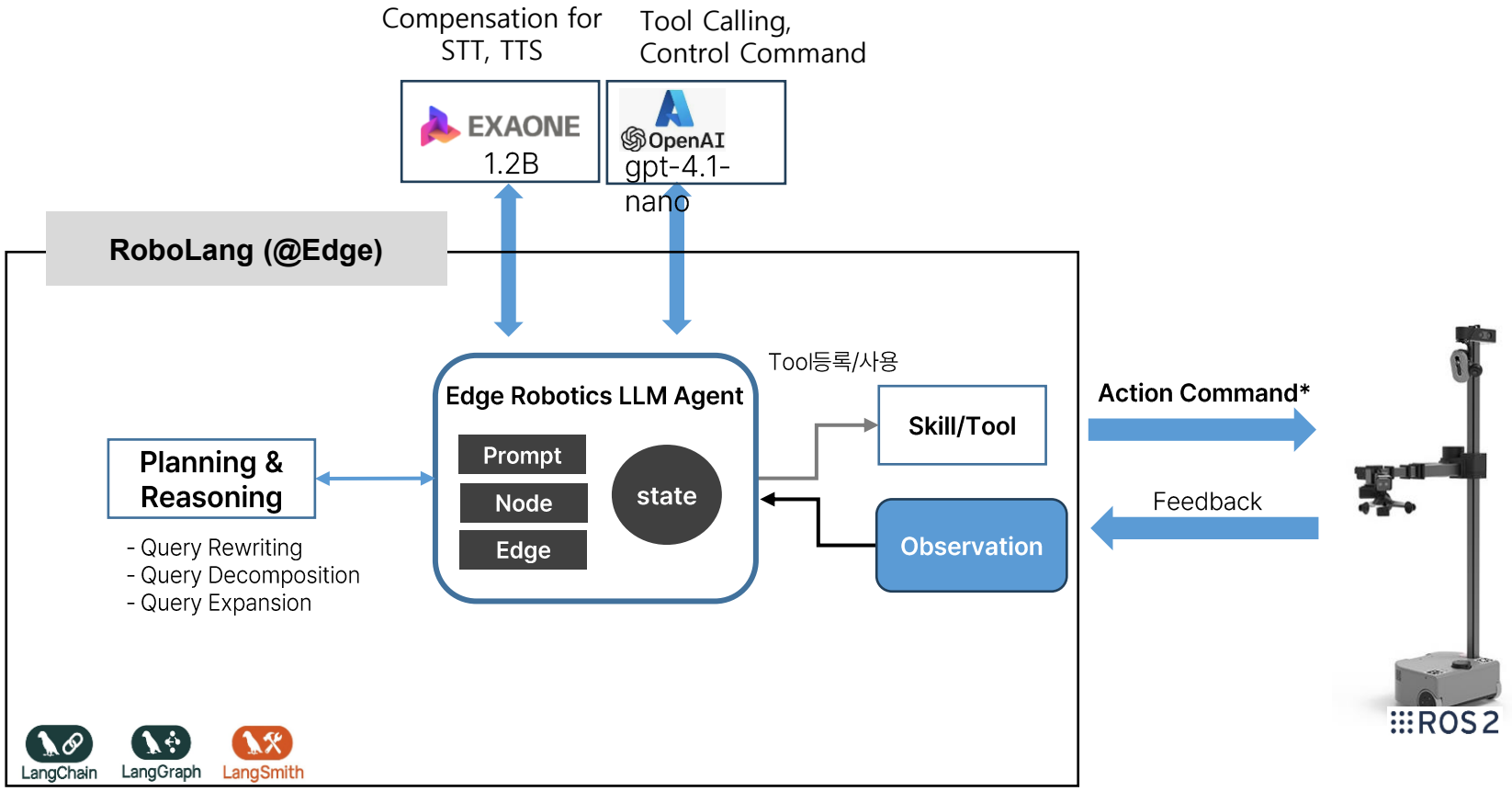




AI Agent 구현 기술

1. LangChain + LangGraph
2. MCP Host/Client, MCP Server
3. RoboClaw (Openclaw in Robot)
4. Continuous Test Piple line for LLM & Agent

1. LangChain / LangGraph 를 활용한 AI Agent 구성



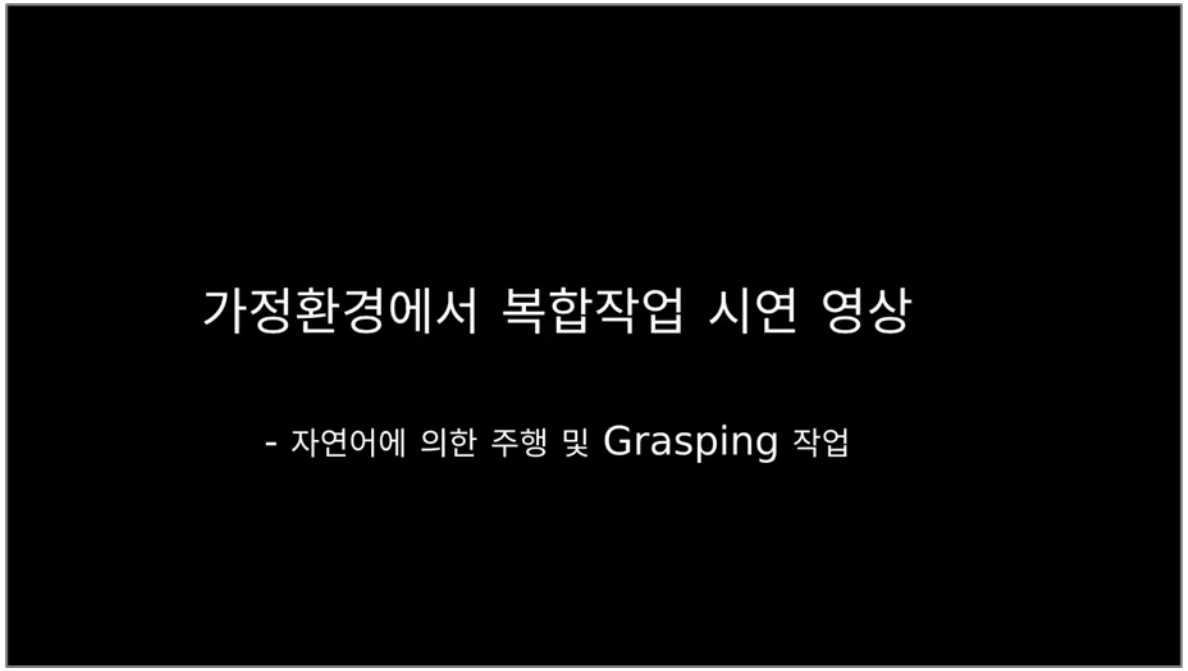
***로봇HW구동에 적합한 Action Generation 전략**
- 형식강제 (Format Enforcing)
- 템플릿 기반 응답 (Templated Response)
- 단계별 추론 (Step-by-step reasoning)

자율행동체의 복합 작업을 위한 그래스핑 모션 기술 실증 동영상

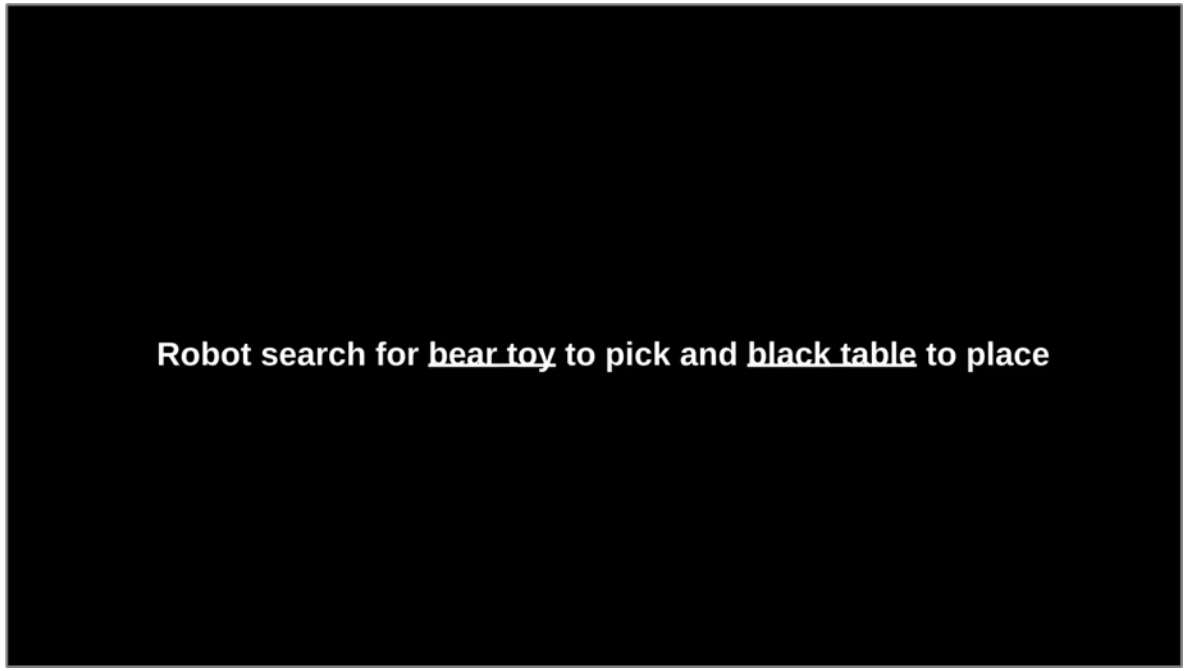
가정환경에서 아래 명령어를 Stretch3 로봇이 실행 함.

- 명령어: **거실로 가서 bear toy 를 black table 에 옮겨 놓은 후, 다시 원점으로 이동해**

View to Robot

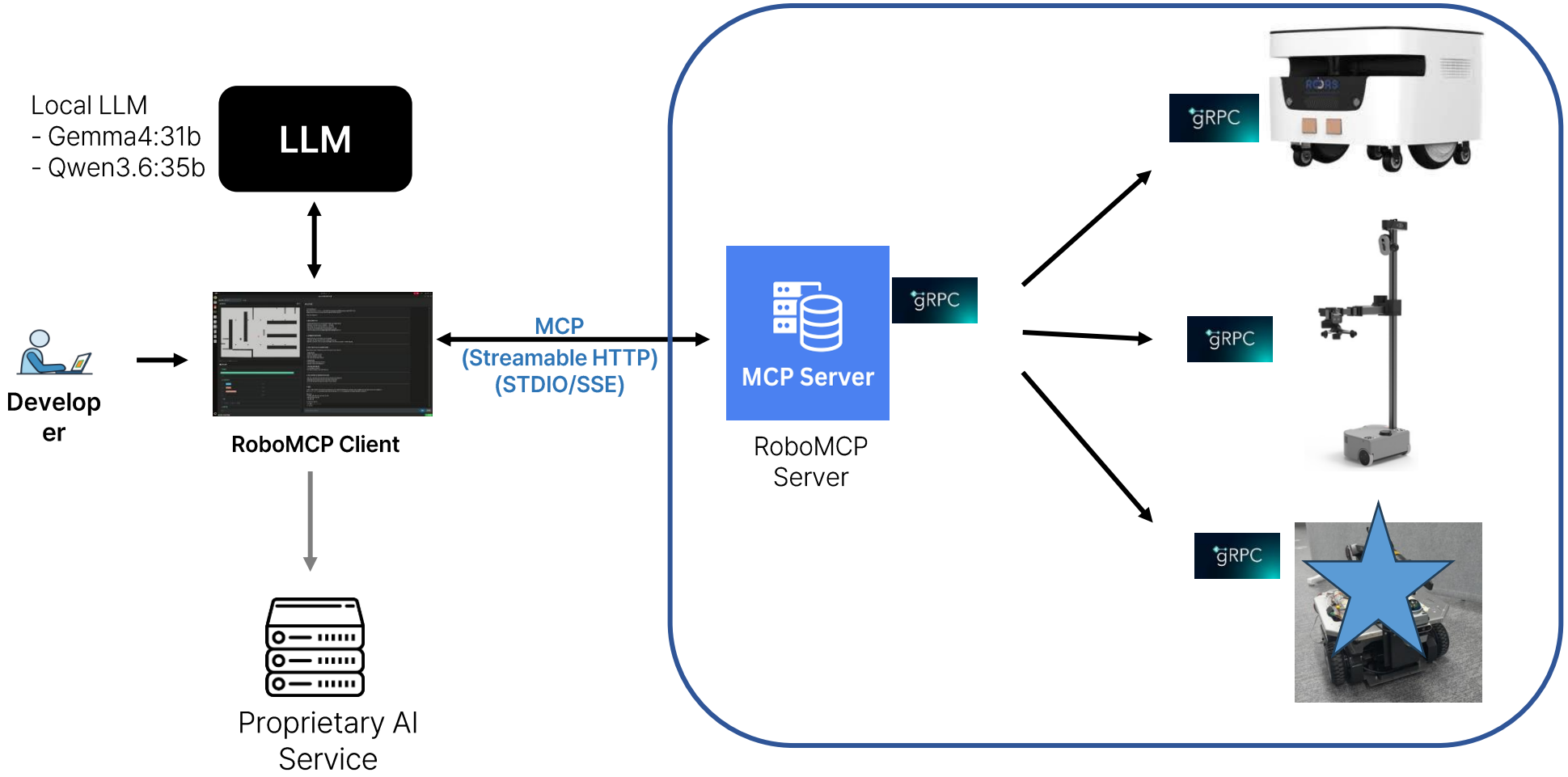


View from Robot Manipulator

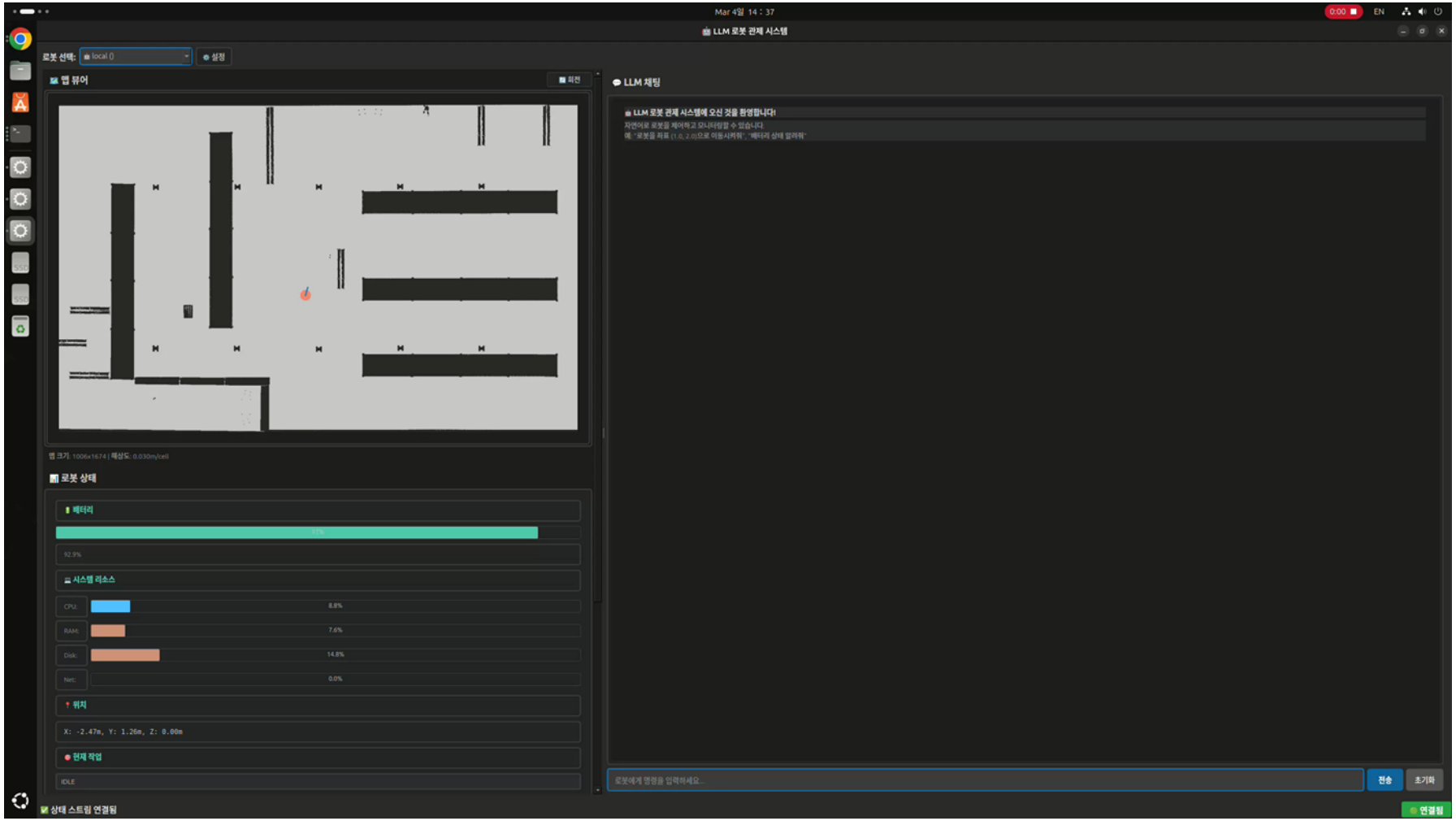


2. RoboMCP를 통한 다수 로봇 제어 실험 구성

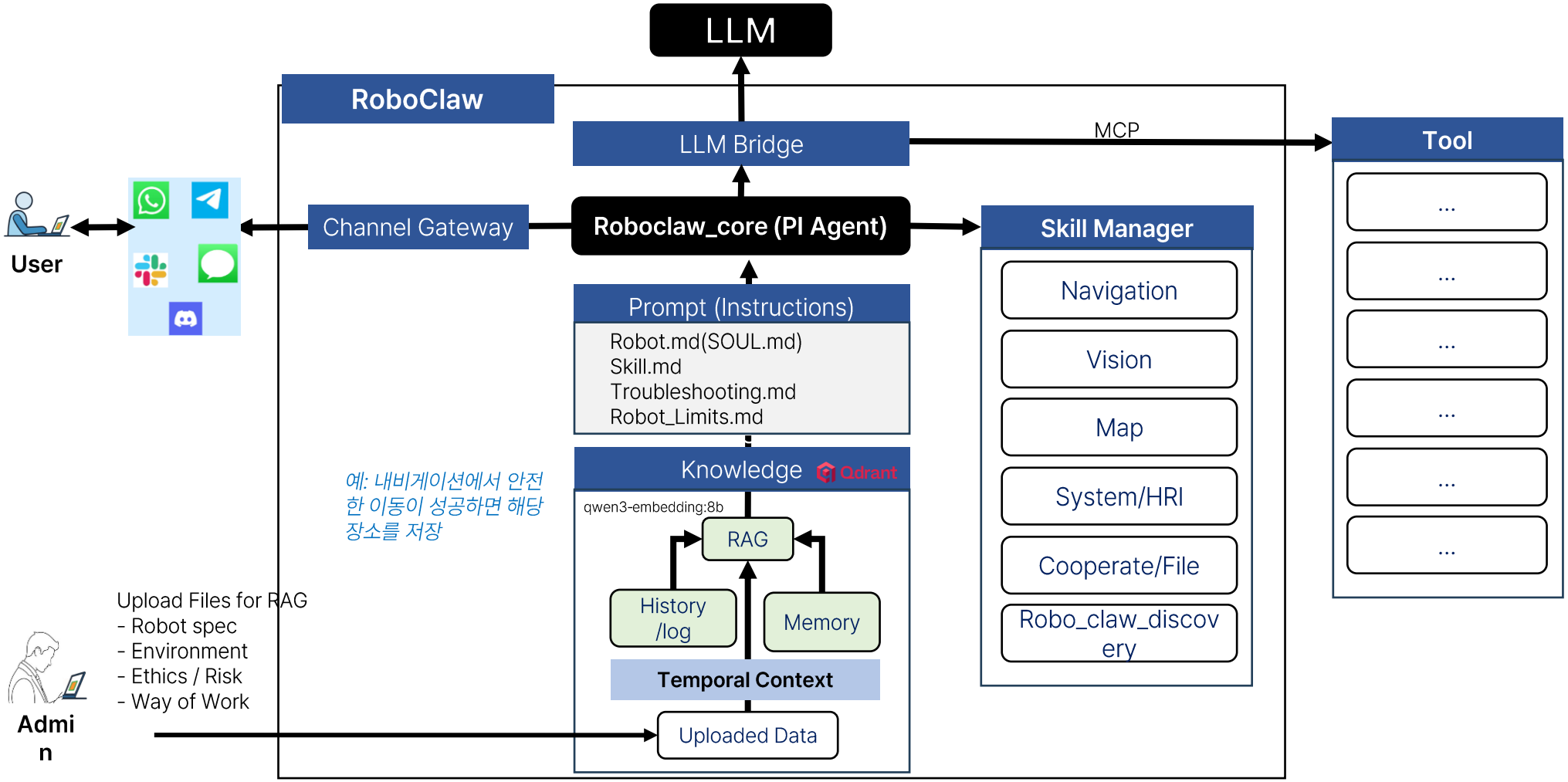
- RoboMCP Server 를 통해서 누구든 로봇 정보 획득 및 로봇 제어



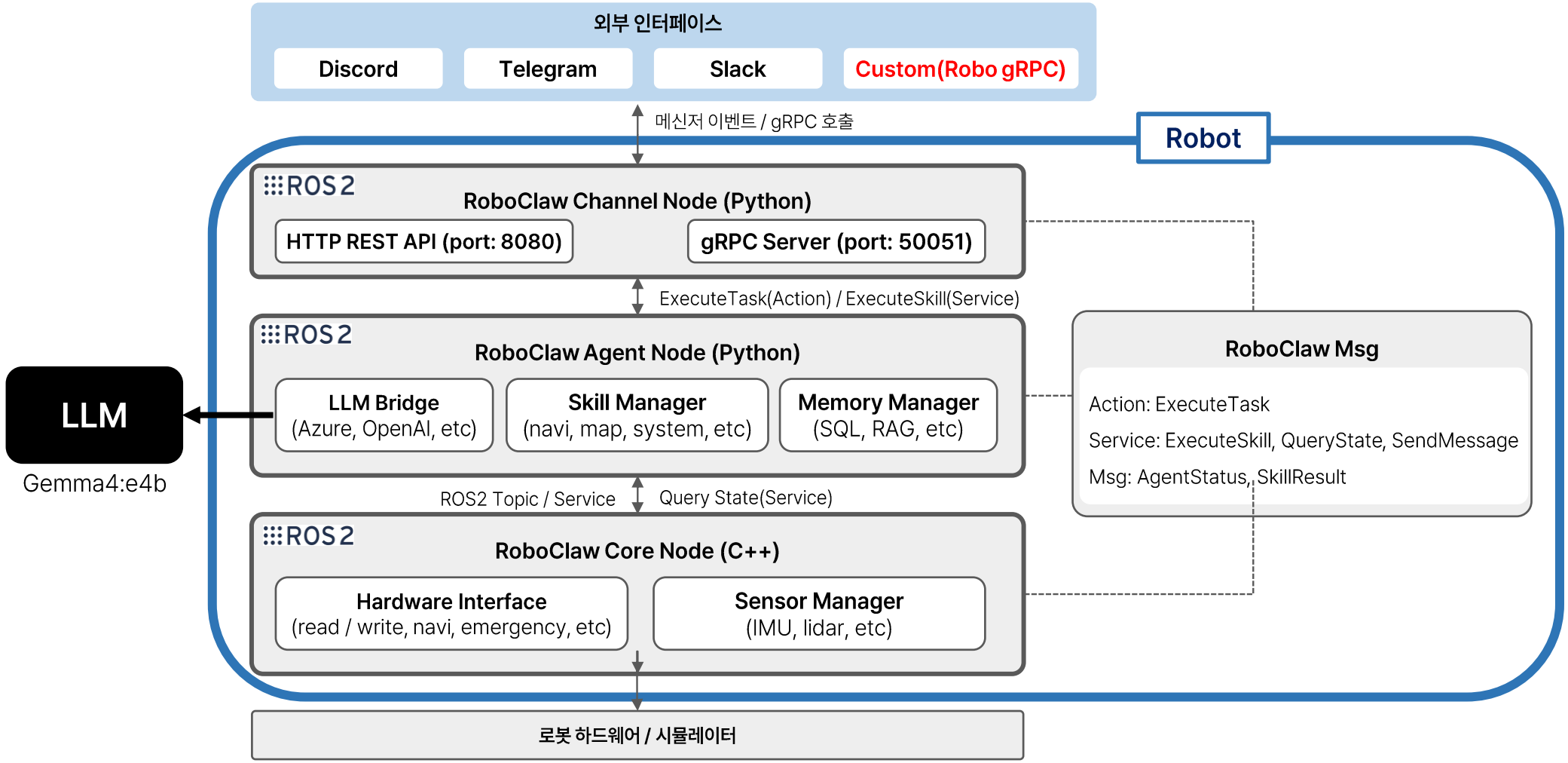
RoboMCP 시연


Mar 4월 14 : 37
LLM 로봇 관제 시스템
로봇 선택:
local ()
설정
맵 뷰어
새로고침
LLM 채팅
LLM 로봇 관제 시스템에 오신 것을 환영합니다!
로봇 상태
배터리
92.9%
시스템 리소스
CPU:
8.8%
RAM:
7.6%
Disk:
14.8%
Net:
0.0%
위치
X: -2.47m, Y: 1.26m, Z: 0.00m
현재 작업
IDLE
로봇에게 명령을 입력하세요...
전송
초기화
상태 스트림 연결됨
연결됨

3. RoboClaw Agent 구조



RoboClaw Agent Platform 구조



RoboClaw 특징

- 에이전트간 협업 가능 구조로 발전

  - LLM (대규모 언어 모델)을 로봇 제어에 통합하고, 단순 명령 수행을 넘어 지능적인 판단과 기억이 가능

  - 에이전트 구현부와 로봇 구현부를 분리하여 설계 -> Skill / Tool 을 Scalable하게 관리

1. 로봇과 연동 : ROS 2 노드로 구현함으로 로봇과 연동 효율화

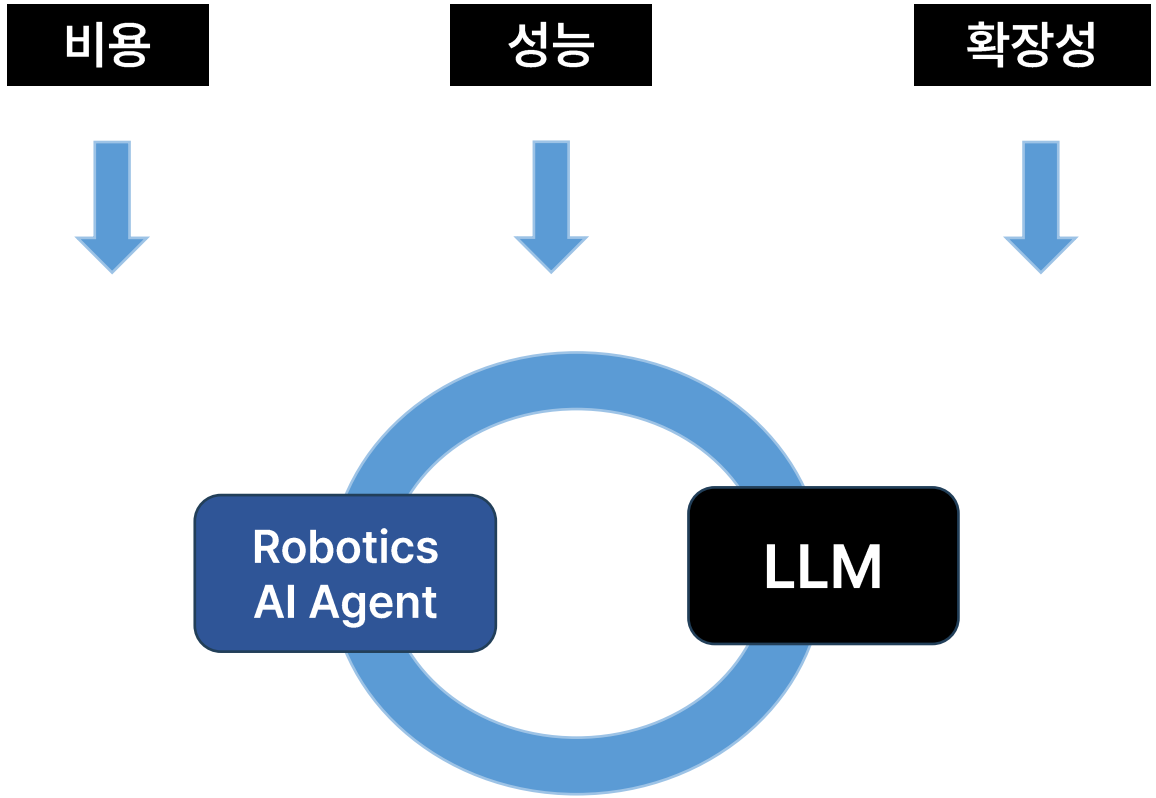
2. Scalability: 내비게이션, 인식, 조작 등 로봇의 기능을 독립적인 Skill / Tool 단위로 구현 관리 및 확장

3. Performance: 성능이 중요한 코어는 C++ 로 구현하여 성능 극대화

4. 다양한 메시지 채널 연동
- **gRPC, HTTP REST API 방식으로 다양한 외부 Custom 메시지 채널 연동 가능**
- **Discord, Slack, Telegram과 직접 연동 가능**

Lesson Learned

- Robotics AI Agent에 적합한 LLM과 Agent Option을 끊임없이 선택, 판단, 튜닝 과정 필요



AI Agent 최적화 시도 : AIAgentOps에 의한 도메인 최적화

- LLMOps -> AIAgentOps 필요
- LLM / Agent 성능 비교 방법 (측정자) 필요

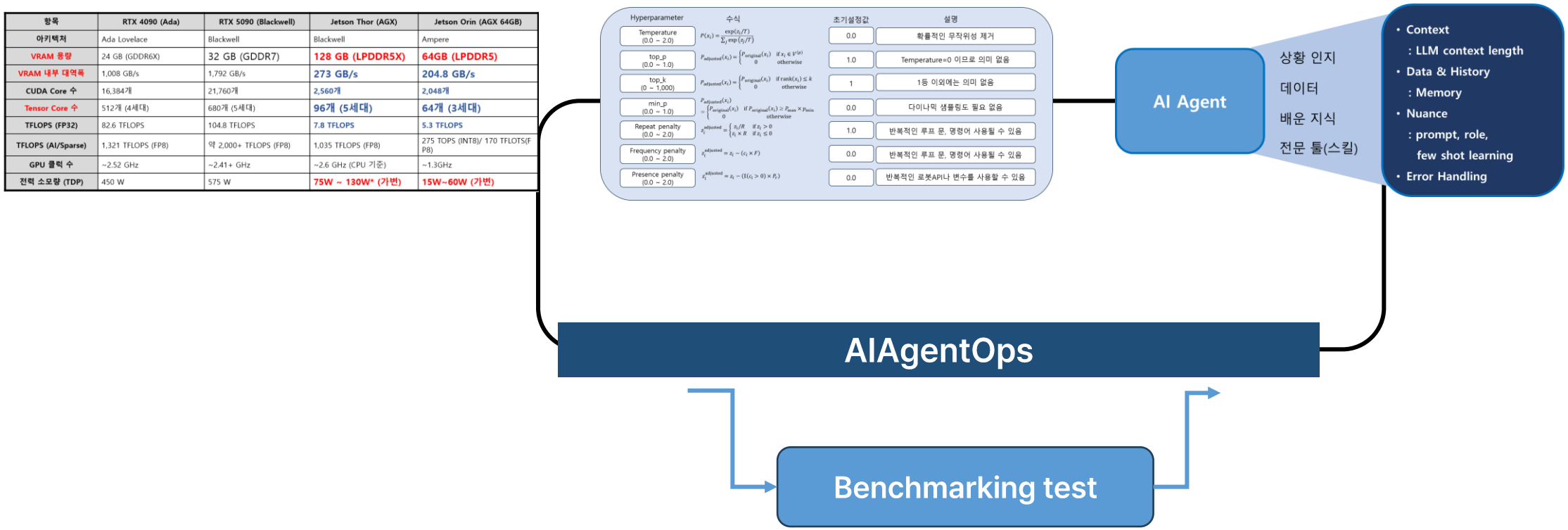
1. 적합한 하드웨어 선정

항목	RTX 4090 (Ada)	RTX 5090 (Blackwell)	Jetson Thor (AGX)	Jetson Orin (AGX 64GB)
아키텍처	Ada Lovelace	Blackwell	Blackwell	Ampere
VRAM 용량	24 GB (GDDR6X)	32 GB (GDDR7)	128 GB (LPDDR5X)	64GB (LPDDR5)
VRAM 내부 대역폭	1,008 GB/s	1,792 GB/s	273 GB/s	204.8 GB/s
CUDA Core 수	16,384개	21,760개	2,560개	2,048개
Tensor Core 수	512개 (4세대)	680개 (5세대)	96개 (5세대)	64개 (3세대)
TFLOPS (FP32)	82.6 TFLOPS	104.8 TFLOPS	7.8 TFLOPS	5.3 TFLOPS
TFLOPS (AI/Sparse)	1,321 TFLOPS (FP8)	약 2,000+ TFLOPS (FP8)	1,035 TFLOPS (FP8)	275 TOPS (INT8)/ 170 TFLOTS(FP8)
GPU 클럭 수	~2.52 GHz	~2.41+ GHz	~2.6 GHz (CPU 기준)	~1.3GHz
전력 소모량 (TDP)	450 W	575 W	75W ~ 130W* (가변)	15W~60W (가변)

2. LLM 튜닝

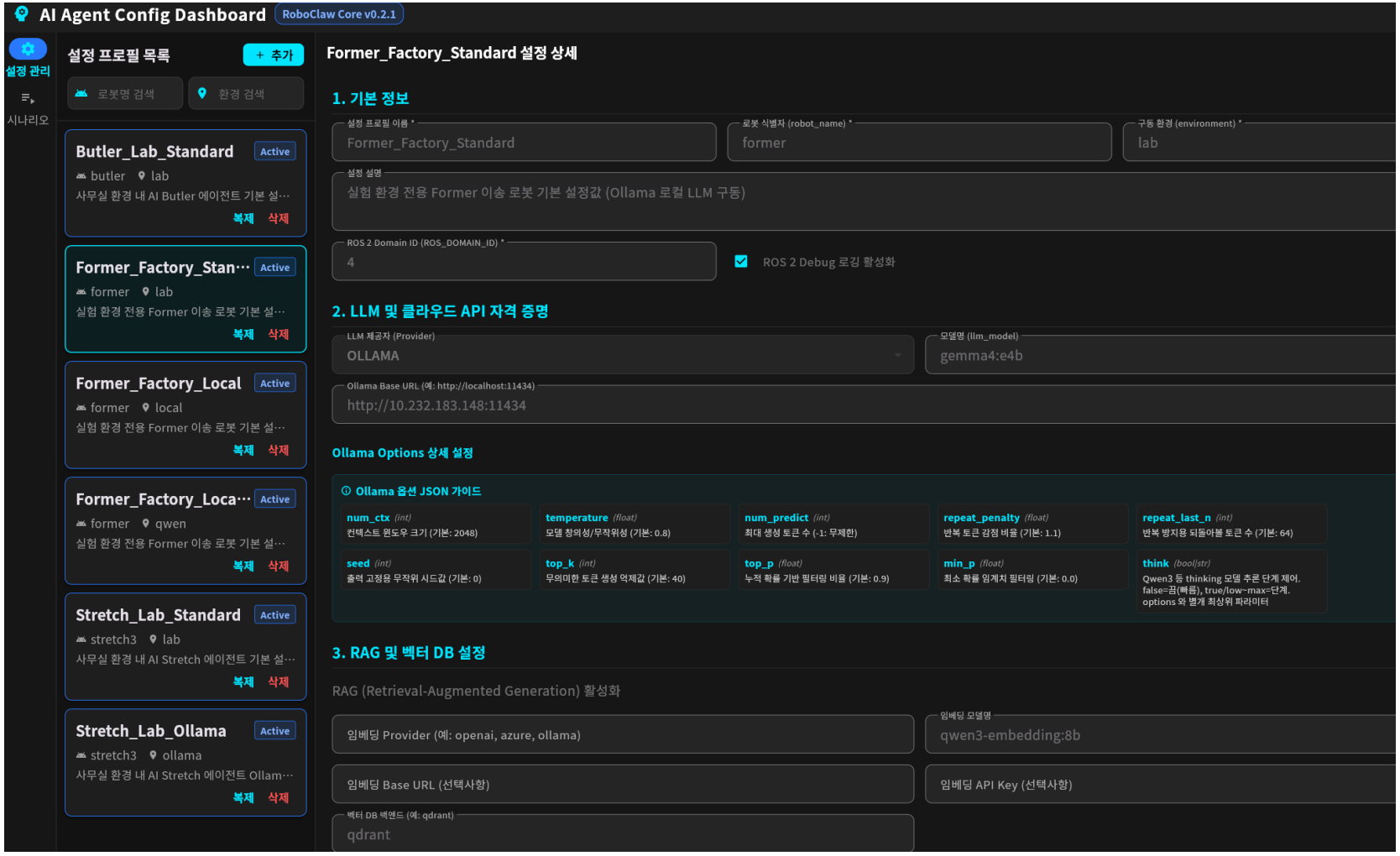
Hyperparameter	수식	초기설정값	설명
Temperature (0.0 ~ 2.0)	$P(x_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$	0.0	확률적인 무작위성 제거
top_p (0.0 ~ 1.0)	$P_{\text{adjusted}}(x_i) = \begin{cases} P_{\text{original}}(x_i) & \text{if } x_i \in V^{(p)} \\ 0 & \text{otherwise} \end{cases}$	1.0	Temperature=0 이므로 의미 없음
top_k (0 ~ 1,000)	$P_{\text{adjusted}}(x_i) = \begin{cases} P_{\text{original}}(x_i) & \text{if } \text{rank}(x_i) \le k \\ 0 & \text{otherwise} \end{cases}$	1	1등 이외에는 의미 없음
min_p (0.0 ~ 1.0)	$P_{\text{adjusted}}(x_i) = \begin{cases} P_{\text{original}}(x_i) & \text{if } P_{\text{original}}(x_i) \ge P_{\max} \times P_{\min} \\ 0 & \text{otherwise} \end{cases}$	0.0	다이나믹 샘플링도 필요 없음
Repeat penalty (0.0 ~ 2.0)	$z_i^{\text{adjusted}} = \begin{cases} z_i / R & \text{if } z_i > 0 \\ z_i \times R & \text{if } z_i \le 0 \end{cases}$	1.0	반복적인 루프 문, 명령어 사용될 수 있음
Frequency penalty (0.0 ~ 2.0)	$z_i^{\text{adjusted}} = z_i - (c_i \times F)$	0.0	반복적인 루프 문, 명령어 사용될 수 있음
Presence penalty (0.0 ~ 2.0)	$z_i^{\text{adjusted}} = z_i - (\mathbb{I}(c_i > 0) \times P_r)$	0.0	반복적인 로봇API나 변수를 사용할 수 있음

3. Agent 튜닝



AI Agent Config Server

: AI Agent Configuration Dashboard for LLM & RoboClaw



AI Agent Config 설정 정보

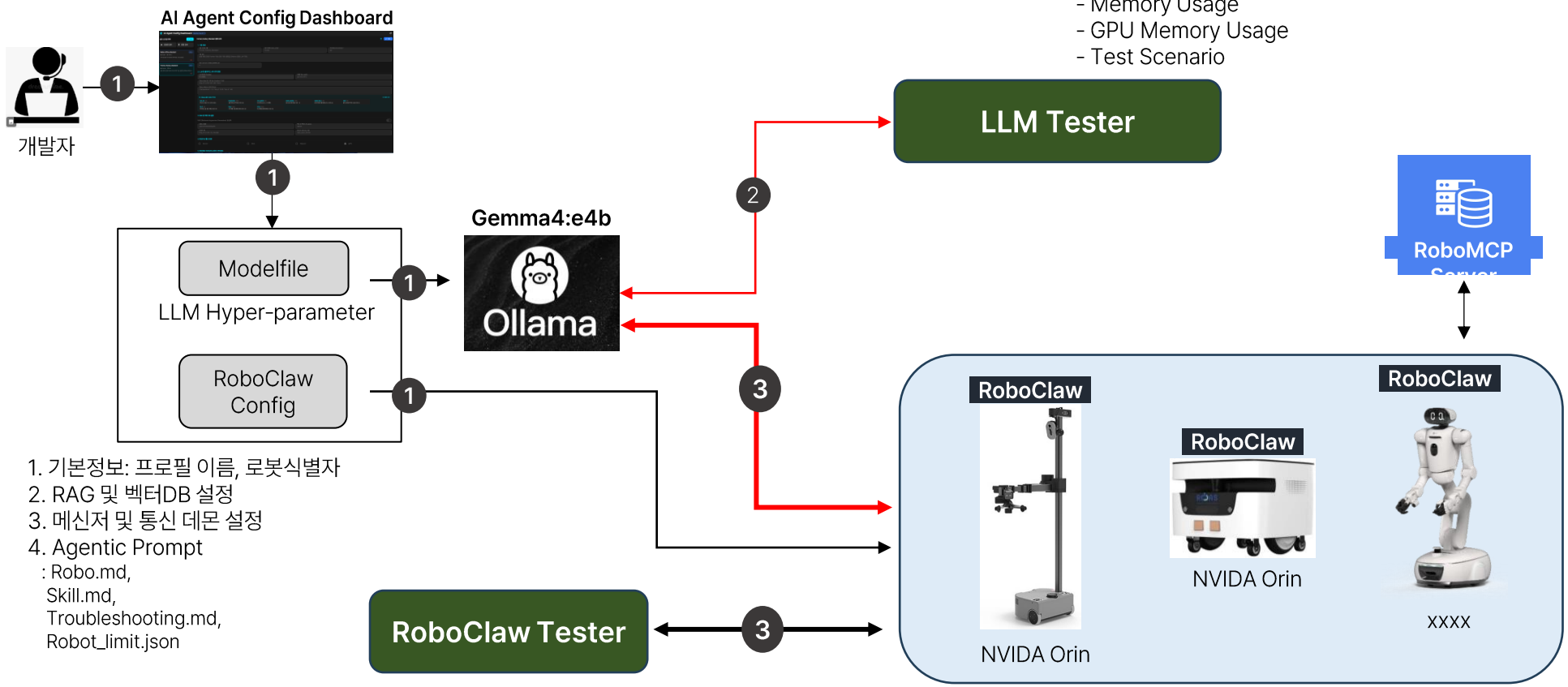
- **로봇 기본 정보**
  - name · robot_name
  - environment · is_active
  - description
- **LLM 기본 설정**
  - llm_provider (multi-cloud)
  - llm_model
  - ros_domain_id
- **LLM 접속 자격증명 (마스킹)**
  - azure_openai_api_key
  - openai_api_key
  - anthropic_api_key
- **Ollama 파라미터**
  - ollama_base_url
  - ollama_options_json (num_ctx · temperature ...)
- **RAG / Qdrant 설정**
  - enable_rag · embedding
  - rag_vector_backend
  - qdrant_url · collection
- **메신저 연동**
  - discord · telegram · slack
  - grpc integration
  - bot_tokens management
- **System Prompt Added**
  - ROBOT.md
  - SKILLS.md
  - Troubleshooting.md
  - ROBOT_LIMITS.md
- **MCP서버 설정**
- **Langsmith 서버 설정**
- **복합명령 처리를 위한 환경 설정**
  - Langsmith Key
  - Project
- **System 1 모델 (Router) 설정**
  - System One Server Key
  - Select Provider : Rule / Laya
  - Timeout for Fallback
  - Confidence Threshold

gorm.Model: ID · CreatedAt · UpdatedAt · DeletedAt (Soft Delete)

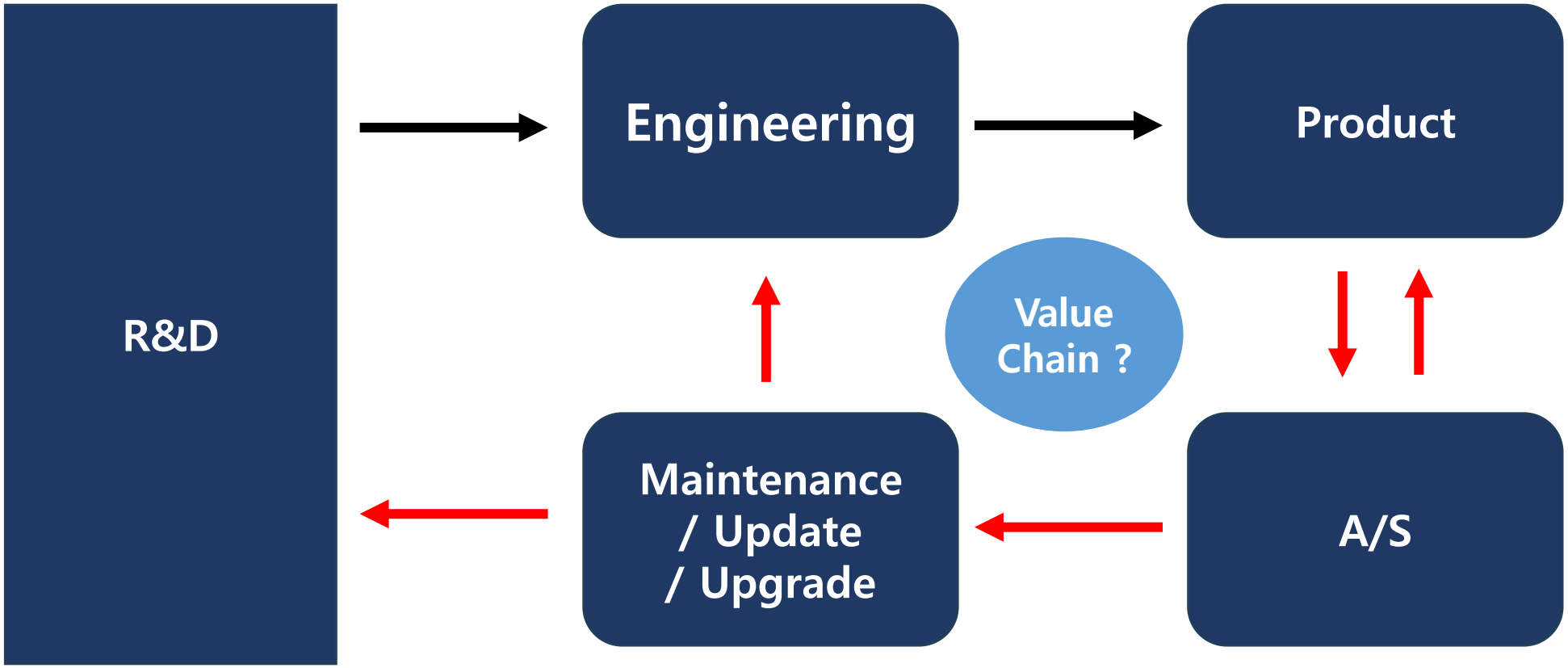
SQLite: WAL · busy_timeout 5000ms · MaxOpenConns: 1 (Write Lock)

AIAgentOps 실험 구성

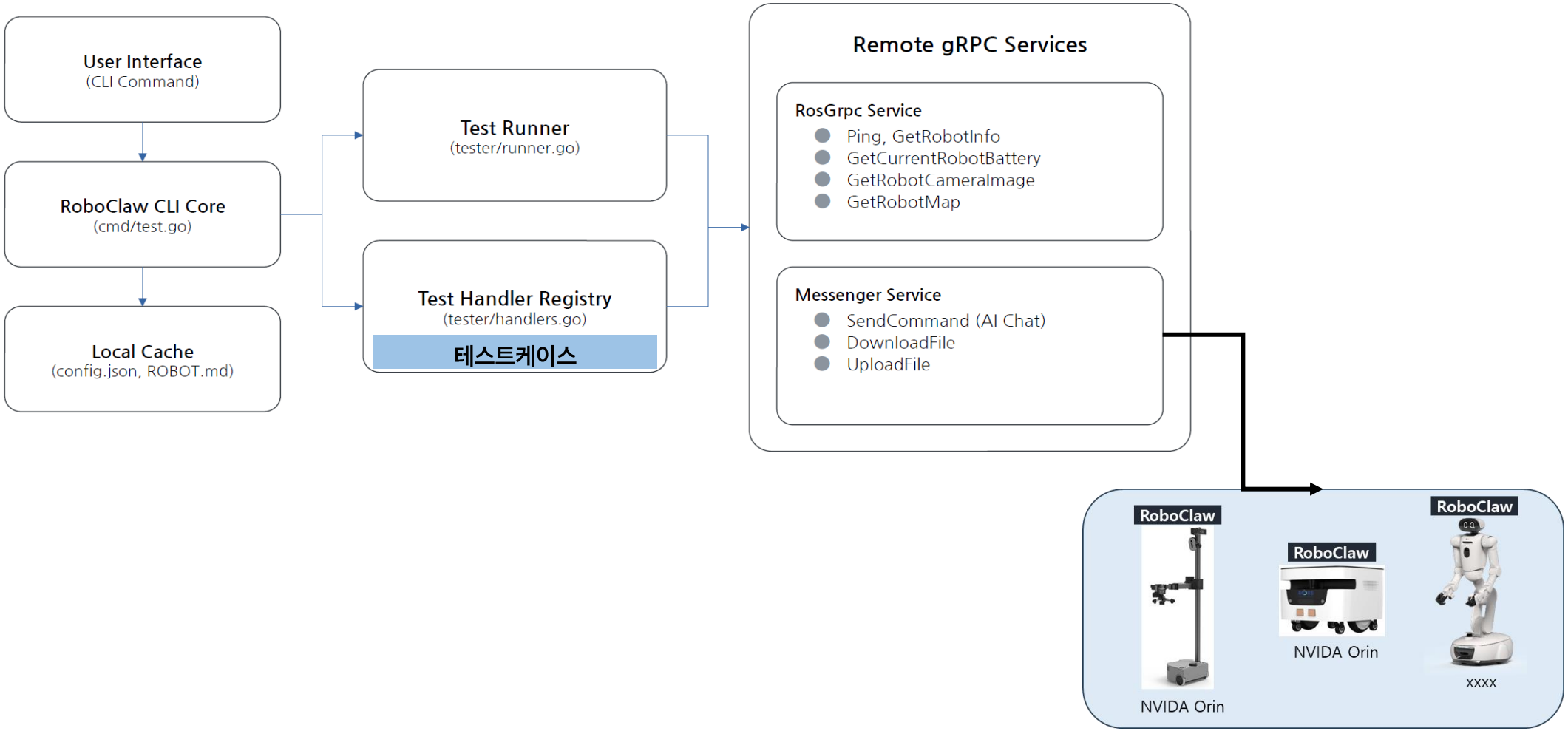
1. AI Agent Config Dashboard 를 통한 설정 (LLM, RoboClaw)
2. LLM Test
3. RoboClaw (Robot) Test



문제는 테스트 !


R&D
Engineering
Product
Value Chain ?
Maintenance / Update / Upgrade
A/S

RoboClaw Tester 구조



RoboClaw Tester 개요

- 로봇 구동(Bringup) 환경 상태를 검증하고 AI Agent 성능을 시나리오 기반으로 자동화 진단
- RoboClaw Config 서버로 부터 최신 설정 프로파일과 활성 테스트 시나리오를 연동하고 동기화

MCP 서버용 RosGrpc 와 대화형 RoboMessenger 채널을 다각도로 활용

네트워크 단절 등 예기치 않은 환경에서도 내장 시나리오 기반으로 기본 검증 프로세스 진행 가능

Robot.md 파일과 LLM에이전트의 답변을 대조하여 페르소나 일치 여부 진단

스텝별 결과를 취합하여 test_report_*.md 보고서 자동 작성

결 론

- Model

```
Output = model (prompt)
```

- Robotics AI Agent

```
Memory = prompt        "robot action"
While not is_task_done() :
    response = model (memory)
    memory = update_memory(memory, response)
```





Harness Engineering

RoboLang	RoboMCP	RoboClaw

AIAgentOps

LLM Tester / RoboClaw Tester

발전 방향 – Self-Evolving AI Agent

- 진화 단계별 진화대상과 엔진 (Hermes Agent)

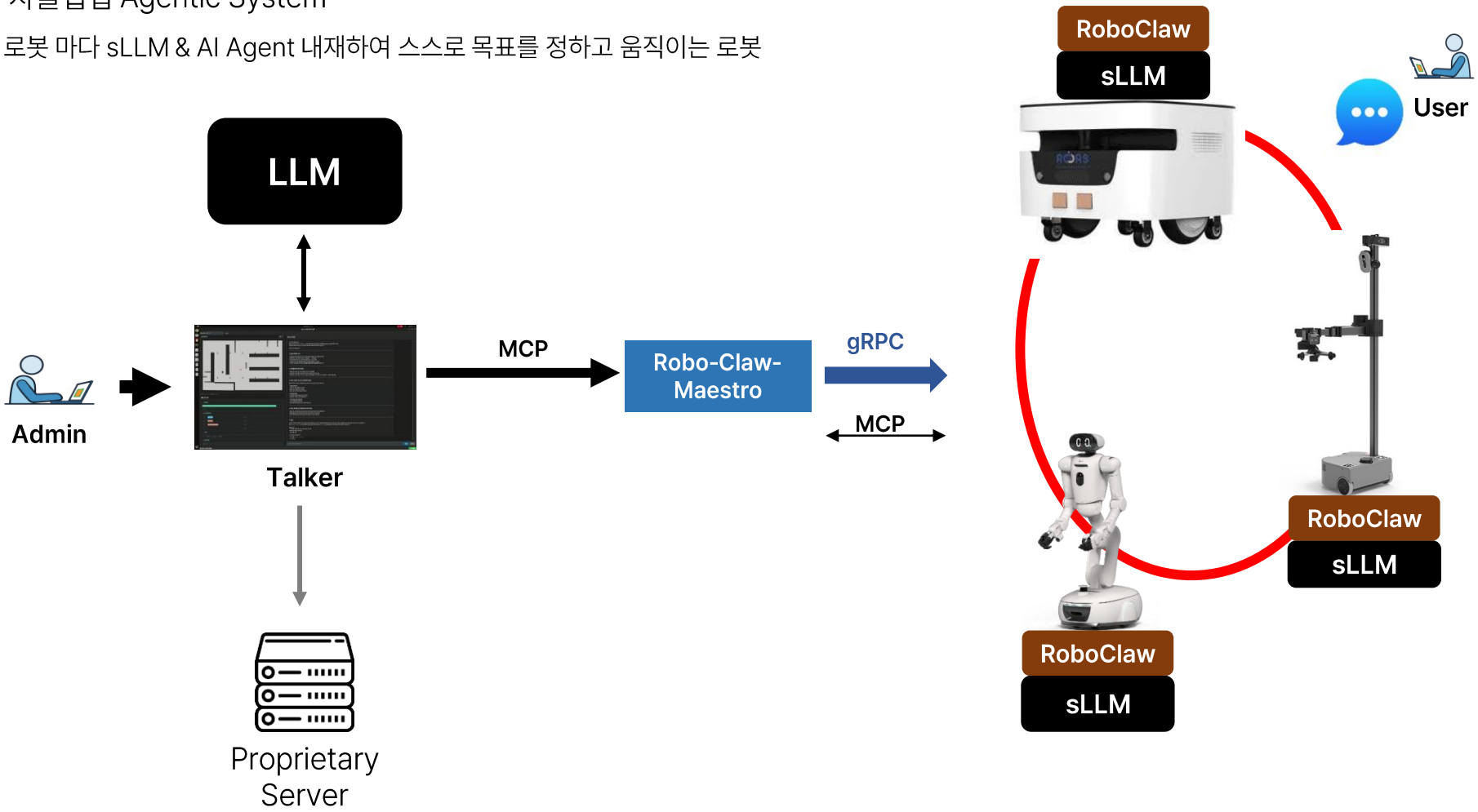
진화 단계	진화 대상	엔진	상태
Phase 1	스킬 파일(SKILL.md)	DSPy + GEPA	구현됨 ✅
Phase 2	도구 설명(Tool Description)	DSPy + GEPA	계획 ⬜
Phase 3	시스템 프롬프트 섹션	DSPy + GEPA	계획 ⬜
Phase 4	도구 구현 코드	Darwinian Evolver	계획 ⬜
Phase 5	지속적 개선 루프	자동화 파이프라인	계획 ⬜

참고: https://discuss.pytorch.kr/t/hermes-agent-self-evolution-hermes-agent-nousresearch/10136

향후 계획

- 자율협업 Agentic System

  - 로봇 마다 sLLM & AI Agent 내재하여 스스로 목표를 정하고 움직이는 로봇



10th Anniversary

GISC 2026

Global ICT Standards Conference 2026

Q&A

ICT Standards and Intellectual Property:
Shaping the Physical AI

주최 과학기술정보통신부 지식재산처 주관 국립전파연구원 TTA IITP KEA kista ETRI