WHITEPAPER

MSAP.ai

오픈소스 AI Agent 오케스트레이션 Hermes Agent

데이터 주권을 지키면서 사내 자동화를 시작하는 가장 빠른...

2025년에서 2026년으로 넘어오는 동안 국내 기업과 공공기관의 AI 에이전트 도입은 한 가지 공통된 난제 앞에 서게 되었습니다. 외부 SaaS LLM 을 그대로 쓰자니 데이터 외부 송신 통제가 어렵고, 사내에 sLLM 을 직접 구축하자니 운영 부담이 큽니다.

2026-06-30 · 작성자: SMART CF Team · msap.ai

목차

오픈소스 AI Agent 오케스트레이션 Hermes Agent

2025년에서 2026년으로 넘어오는 동안 국내 기업과 공공기관의 AI 에이전트 도입은 한 가지 공통된 난제 앞에 서게 되었습니다. 외부 SaaS LLM 을 그대로 쓰자니 데이터 외부 송신 통제가 어렵고, 사내에 sLLM 을 직접 구축하자니 운영 부담이 큽니다. 2026년 2월 Nous Research 가 공개한 Hermes Agent 는 이 난제에 정면으로 답하는 오픈소스 자율 AI 에이전트입니다. MIT 라이선스 위에 persistent memory 와 자기 개선 루프(Curator loop), 6개 채널 통합, LiteLLM 게이트웨이와 MCP 표준 결합을 단일 OSS 패키지로 제공하여 사내 데이터를 외부로 내보내지 않고도 쓸 만한 에이전트를 구축할 수 있는 구조를 갖춥니다. 본 백서는 Hermes Agent 의 등장 배경과 라이선스, 4가지 핵심 추상화(Profile · Kanban · Skill · Archive), Paperclip · OpenClaw · Harness Engineering 패러다임과의 정량·정성 비교, Local LLM(Gemma 4 31B Dense · Qwen3 · gpt-oss 20B) 선정 가이드, 공공·제조·금융·연구개발 적용 사례, 그리고 PoC 4주 → 파일럿 3개월 → 본격 운영의 도입 로드맵을 한 권에 정리합니다. IT 의사결정권자가 사내 기술 검토서·도입 품의서를 작성할 때 그대로 인용할 수 있도록 근거 라벨([S01]~[S12])과 수치를 본문에 직접 명시했습니다.

1장. Hermes Agent 란 무엇인가 — 등장 배경·라이선스·커뮤니티

Nous Research 가 2026년 2월 공개한 Hermes Agent 는 오픈소스 자율 AI 에이전트 프레임워크입니다. MIT 라이선스로 배포되어 상업적 이용과 사내 수정 모두 법적 제약 없이 가능하며, 단일 `curl` 명령 하나로 Linux·macOS·WSL2 환경에 설치됩니다. 공개 4개월 만에 GitHub stars 32,000건을 넘어섰고, 245명 이상이 직접 코드 기여에 참여했습니다 [S01]. 이 장은 Hermes Agent 가 왜 지금 등장했는지, 어떤 라이선스 조건을 갖추고 있는지, 그리고 커뮤니티 모멘텀을 어떻게 해석해야 하는지를 순서대로 살펴봅니다. 이를 통해 사내 기술 검토서의 "벤더 프로파일" 섹션을 채울 수 있는 구체적 데이터를 확보합니다.

1.1 Nous Research 가 2026-02 Hermes Agent 를 공개한 배경

Hermes Agent 의 탄생을 이해하려면 먼저 개발 주체인 Nous Research 의 성격, 이들이 포착한 기술 공백, 그리고 "단일 명령 설치"라는 설계 철학이 어떤 도입 문턱을 낮추는지를 차례로 확인해야 합니다. 이 세 가지 관점이 맞물릴 때, 왜 이 프레임워크가 등장 직후 빠르게 주목받았는지 이해할 수 있습니다.

1.1.1 Nous Research 가 Hermes Agent 를 만든 동기와 첫 공개 일자

Nous Research 의 조직 성격

Nous Research 는 오픈소스 AI 모델 연구와 에이전트 기술 개발을 중심으로 활동하는 AI 연구 그룹입니다. 독자 개발 LLM 시리즈로 이미 오픈소스 커뮤니티에서 인지도를 쌓아온 조직으로, Hermes Agent 공개 이전에도 다수의 오픈소스 언어 모델을 커뮤니티에 기여해왔습니다. 이 배

경이 중요한 이유는, Hermes Agent 가 단순한 스타트업 실험 제품이 아니라 OSS 언어 모델 생태계를 직접 구축해온 팀의 노하우를 집약한 결과물이기 때문입니다.

첫 공개와 4개월간의 성장 궤적

Hermes Agent 는 2026년 2월 GitHub 를 통해 처음 공개됐습니다 [S01]. 이후 4개월이 채 되지 않아 2026년 6월 v0.17.0 버전이 출시됐으며, 같은 시점 기준 GitHub stars 32,000건 이상, 기여자 245명 이상, 총 커밋 수 1,475건을 기록했습니다 [S01]. 신생 오픈소스 에이전트 프레임워크가 이 속도로 커뮤니티 기반을 확보한 사례는 흔치 않습니다. 일부 매체는 집계 기준에 따라 더 높은 star 수치를 보고하기도 하지만 [S04], 이 백서에서는 공식 저장소 (github.com/NousResearch/hermes-agent) 기준 수치를 인용합니다.

커뮤니티 반응이 빠른 이유

프레임워크가 단기간에 주목받은 배경에는 기술 구조보다 진입 비용이 낮다는 점이 크게 작용합니다. 별도 클라우드 계정 개설이나 복잡한 의존성 설치 없이, 단일 `curl` 명령 하나로 환경을 구성할 수 있습니다. 5달러짜리 가상 서버(VPS)부터 GPU 클러스터, 서버리스 환경까지 동일한 코드베이스로 동작한다는 점도 실험 단계에서 비용 부담 없이 기능을 검증할 수 있다는 실용적 장점으로 작용했습니다 [S01]. 이 같은 설계 철학은 기술 검토 초기에 개념 증명(PoC)을 빠르게 시작하고 싶은 팀에게 결정적인 진입점이 됩니다.

지표	수치 (2026-06 기준)
첫 공개	2026-02
최신 릴리스	v0.17.0 (2026-06-19)
GitHub stars	32,000+
기여자 수	245+
총 커밋 수	1,475
동작 환경	5 USD VPS ~ GPU 클러스터 · 서버리스

1.1.2 "Persistent Memory + Self-Improvement" 라는 차별점이 등장한 페인포인트

반복되는 컨텍스트 재설명 문제

기존 LLM 기반 챗봇이나 에이전트 도구를 업무에 활용해본 팀이라면 공통된 불편을 경험합니다. 세션을 새로 시작할 때마다 프로젝트 배경, 팀 용어, 우선순위 맥락을 처음부터 다시 설명해야 한다는 점입니다. 대화 창을 닫으면 에이전트는 이전 작업 내용을 기억하지 못하고, 다음 세션은 사실상 빈 상태에서 출발합니다. 이 구조에서 에이전트가 축적하는 것은 없으며, 운영 시간이 길어질수록 팀의 설명 부담만 늘어납니다.

Hermes 의 Persistent Memory 와 Curator Loop

Hermes Agent 는 이 문제에 대해 두 가지 구조적 응답을 제시합니다 [S02]. 첫째는 대화와 작업 이력을 지속적으로 보존하는 Persistent Memory(영속 기억) 계층입니다. 에이전트는 세션이 끊

겨도 이전에 학습한 사용자 선호와 작업 맥락을 그대로 유지합니다. 둘째는 Curator Loop 입니다. 15회 도구 호출 또는 복잡한 작업이 끝난 뒤 에이전트가 스스로 회고를 수행하고, 재사용 가능한 skill 파일을 작성하여 저장합니다. 다음 실행 시 이 skill 파일을 자동으로 불러와 같은 유형의 작업에 즉시 활용합니다. 공식 사이트의 표현을 빌리면, "The longer it runs, the better it knows you — no re-explaining context every time"[S02]입니다.

운영 시간 누적 효과의 의미

이 구조가 일반 챗봇과 근본적으로 다른 이유는 운영 시간이 쌓일수록 에이전트의 유용성이 증가한다는 점입니다. 일반 챗봇은 수백 번 사용해도 첫날과 동일한 성능에서 출발하지만, Hermes Agent 는 사용할수록 사용자의 업무 방식, 선호 도구, 반복 패턴을 내재화합니다. 이 누적 효과는 단기 PoC 단계에서는 체감이 제한적이지만, 3개월 이상 실제 업무에 투입된 환경에서 비교 우위가 뚜렷하게 드러납니다.

비교 항목	일반 LLM 챗봇	Hermes Agent
세션 간 컨텍스트 유지	없음 (매 세션 초기화)	있음 (Persistent Memory)
반복 작업 자동화 학습	없음	Curator Loop — skill 파일 자동 생성
운영 시간 누적 효과	없음	사용할수록 개인화 수준 향상
장기 운영 시 재설명 부담	매 세션 반복 발생	점진적으로 감소

1.1.3 단일 curl 설치와 MIT 라이선스가 의미하는 도입 진입 장벽

설치 복잡성이 PoC 속도에 미치는 영향

신기술 도입 검토에서 가장 먼저 막히는 지점은 기술 역량이 아니라 초기 설정 복잡성입니다. 환경 구성에 이틀, 의존성 충돌 해소에 하루가 걸리면 의사결정자가 실제 기능을 확인하기도 전에 검토 동력이 떨어집니다. Hermes Agent 는 Linux, macOS, WSL2 환경에서 단일 `curl` 명령 하나로 의존성 자동 설치와 환경 구성이 완료됩니다 [S01]. 별도 컨테이너 설정이나 패키지 관리자 구성 없이 설치 후 수 분 내 첫 실행이 가능합니다. PoC 시작까지의 기술적 장벽이 낮다는 뜻이며, 이는 검토 사이클을 단축하는 직접적인 요인입니다.

MIT 라이선스가 법무 검토에서 가지는 위치

도입 검토의 두 번째 관문은 법무 검토입니다. 사내 시스템에 오픈소스 소프트웨어를 통합할 때 라이선스 조건이 사내 코드 공개 의무를 유발하는지를 반드시 확인해야 합니다 [S12]. Hermes Agent 본체는 MIT 라이선스로 배포됩니다 [S01]. MIT 라이선스는 사용·수정·재배포 모두 허용하며, 유일한 의무는 저작권 고지 문구를 유지하는 것입니다. 사내에서 소스 코드를 수정하여 사용하더라도 수정본 공개 의무가 없습니다. 이 점은 GPL 계열 라이선스와 가장 큰 차이이며, 법무 검토 소요 시간을 크게 단축합니다. Synopsys 의 2025년 리포트에 따르면 코드베이스의 60%에서 라이선스 충돌이 발견됩니다 [S12]. Hermes Agent 의 MIT 라이선스 본체는 이 위험에서 가장 먼 위치에 있습니다.

진입 장벽 비교

비교 항목	Hermes Agent	일반 사내 도구 도입 사례
환경 구성 방법	`curl` 단일 명령 (수 분)	설치 패키지 + 의존성 수동 구성 (수 시간~수일)
PoC 시작 소요 시간	당일 ~ 1일 이내	수일 ~ 1~2주
라이선스 유형	MIT (저작권 고지만 의무)	제품별 상이 (EULA, 상용 라이선스 등)
법무 검토 예상 소요	단기 (MIT 표준 검토)	중~장기 (조건별 상이)
상업적 이용 허용	허용	별도 계약 필요한 경우 多

1.2 라이선스 — MIT 가 국내 기업·공공 도입 시 가지는 의미

MIT 라이선스는 오픈소스 라이선스 중 가장 사용 조건이 단순한 축에 속합니다. 그러나 단순하다는 것이 아무 검토도 필요 없다는 의미는 아닙니다. 본체는 MIT 라이선스라도 함께 설치되는 종속 패키지들의 라이선스 조합에 따라 법적 의무가 달라질 수 있기 때문입니다. 이 절에서는 MIT 라이선스가 국내 기업과 공공기관 도입 맥락에서 무엇을 허용하고 무엇을 의무화하는지, 그리고 종속 패키지 점검을 어떻게 구성해야 하는지를 짚어봅니다.

1.2.1 MIT 라이선스의 사용·재배포·수정 자유와 단일 의무 (저작권 표시 보존)

MIT 라이선스의 네 가지 허용 범위

MIT 라이선스는 소프트웨어를 어떤 목적으로든 사용할 수 있고, 소스 코드를 수정할 수 있으며, 수정본을 재배포할 수 있고, 상업적으로 이용할 수 있다는 네 가지를 명시적으로 허용합니다 [S12]. 특히 국내 기업·공공기관 도입 맥락에서 중요한 점은 사내 수정본의 비공개 유지가 합법이라는 사실입니다. GPL 계열 라이선스에서 종종 발생하는 "수정 소스 코드 공개 의무"가 MIT 에서는 발생하지 않습니다. 이는 사내 업무 맥락에 맞춰 Hermes Agent 의 동작 방식을 조정하거나 내부 시스템과 연동한 맞춤 기능을 추가하더라도, 그 코드를 외부에 공개할 의무가 없다는 것을 의미합니다.

유일한 의무: 저작권 고지 보존

MIT 라이선스의 의무는 단 하나입니다. 소프트웨어를 배포할 때 원본 저작권 고지 문구와 라이선스 전문을 함께 포함해야 합니다 [S12]. 사내에서만 사용하는 경우 — 즉 외부 배포가 없는 경우 — 이 의무조차 실질적으로 영향이 없습니다. 사내 서버에 Hermes Agent 를 설치하고 내부 임직원이 사용하는 형태는 배포에 해당하지 않기 때문입니다. 외부 고객사에 Hermes 기반 서비스를 제공하거나 제품에 내장하여 출시할 때에 한해 이 의무가 활성화됩니다.

주요 라이선스 비교 매트릭스

라이선스	상업적 이용	소스 수정	수정본 비공개	특허 보호	주요 의무
MIT	허용	허용	허용	명시 없음	저작권 고지 보존

라이선스	상업적 이용	소스 수정	수정본 비공개	특허 보호	주요 의무
Apache 2.0	허용	허용	허용	특허 허여 명시	저작권 고지 + 변경 명시
GPL v3	허용	허용	불허	특허 허여 명시	수정 소스 공개 의무
AGPL v3	허용	허용	불허	특허 허여 명시	네트워크 이용 시에도 소스 공개
Gemma Terms of Use	조건부 허용	조건부 허용	조건부 허용	별도	Responsible Use 정책 준수

GPL 과 AGPL 은 수정 소스 코드 공개 의무가 있어 사내 지식재산 보호와 충돌할 수 있습니다. Apache 2.0 은 MIT 와 유사하나 특허 허여 조항이 명시되어 특허 분쟁 시 일정 수준의 보호를 제공합니다. Gemma Terms of Use 는 퍼미시브하지만 Apache 2.0·MIT 와 동일하지 않아 도입 전 사내 법무 검토가 별도로 필요합니다 [S12].

1.2.2 Hermes 종속 패키지의 라이선스 혼합과 GPL 전염 회피 가이드

본체 MIT 와 종속 패키지의 간극

Hermes Agent 본체가 MIT 라이선스를 채택하고 있더라도, 설치 과정에서 함께 내려받는 종속 패키지들은 각기 다른 라이선스를 가질 수 있습니다. MIT, Apache 2.0, BSD 계열이 대부분이지만 일부 패키지에 GPL 계열이 혼재할 경우, GPL 의 소스 공개 의무가 전체 배포물로 "전염"될 가능성이 있습니다 [S12]. 이 문제는 Hermes Agent 만의 고유한 위험이 아니라, 오픈소스 소프트웨어를 사내 시스템에 통합할 때 보편적으로 점검해야 하는 항목입니다. 2025년 Synopsys Open Source Security and Risk Analysis 리포트에 따르면, 점검한 코드베이스의 60%에서 라이선스 충돌이 발견됐습니다 [S12]. 이 수치는 종속 패키지 점검 없이 오픈소스를 통합하는 관행이 얼마나 광범위한 위험 노출로 이어지는지를 보여줍니다.

SBOM 기반 점검 절차

종속 패키지 라이선스 위험을 체계적으로 관리하는 방법은 SBOM(Software Bill of Materials, 소프트웨어 구성 명세서)을 기반으로 점검하는 것입니다. SBOM 은 소프트웨어에 포함된 모든 구성 요소와 해당 라이선스를 목록화한 문서이며, 2026년에는 공공기관과 대기업을 중심으로 SBOM 제출 요구가 증가하고 있습니다 [S12]. Hermes Agent 도입 전 실행해야 할 라이선스 점검 흐름은 다음과 같습니다.

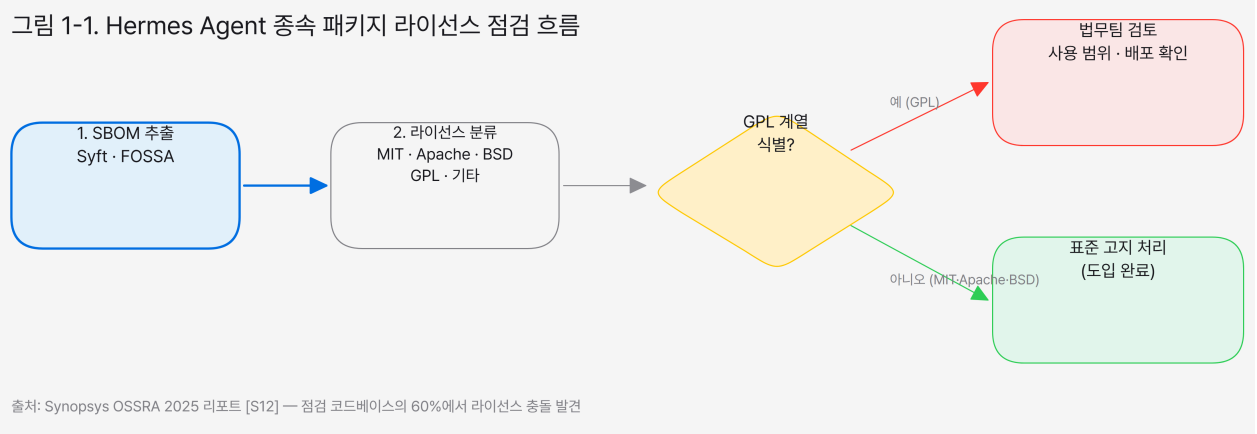


그림 1-1. Hermes Agent 종속 패키지 라이선스 점검 흐름 SBOM 도구(예: Syft, FOSSA)로 Hermes Agent 설치 환경의 전체 패키지 목록을 추출하고, 라이선스 분류(MIT·Apache·BSD·GPL·기타)를 자동 분류합니다. GPL 계열이 식별되면 해당 패키지의 사용 범위와 배포 여부를 확인하여 법무팀 검토 대상으로 분류합니다. MIT·Apache·BSD 는 표준 고지 처리로 완료합니다.

도입 전 최소 점검 체크리스트

법무 검토 부담을 최소화하면서 라이선스 위험을 통제하려면 세 단계를 순서대로 밟는 것이 효과적입니다. 첫 번째 단계는 Hermes Agent 를 설치한 환경에서 SBOM 도구를 실행해 전체 패키지와 라이선스 목록을 확보하는 것입니다. 두 번째 단계는 목록에서 GPL·LGPL·AGPL 항목을 필터링하여 해당 패키지가 사내 코드와 직접 연동되는지 확인합니다. 세 번째 단계는 필터링된 패키지 목록을 법무팀에 전달하여 배포 형태(사내 전용·외부 서비스)에 따른 의무 여부를 확인받습니다. 이 세 단계는 상용 소프트웨어 도입 시 EULA 검토에 상응하는 절차이며, MIT 라이선스 본체 덕분에 대부분의 사내 전용 사용 사례에서는 큰 제약 없이 통과됩니다 [S12].

1.3 커뮤니티 모멘텀 — 32k+ stars 와 245+ contributors 가 의미하는 것

GitHub stars 수치는 커뮤니티 관심도를 나타내는 지표이지만, 그것만으로 도입 적합성을 판단하기에는 정보가 부족합니다. 중요한 것은 기여자 수, 릴리스 속도, 이슈 처리 패턴처럼 실제 유지보수 활력을 보여주는 지표들입니다. 이 절은 Hermes Agent 의 정량 커뮤니티 지표를 읽는 방법, 비슷한 시기에 등장한 경쟁 오픈소스 프레임워크와의 상대적 위치, 그리고 이 데이터를 근거로 국내 도입 시점을 어떻게 판단할 수 있는지를 차례로 다룹니다.

1.3.1 GitHub stars · contributors · 릴리스 속도 4개월 추이

4개월 성장의 의미

Hermes Agent 는 2026년 2월 공개 이후 6월까지 약 4개월 동안 GitHub stars 32,000건 이상, 기여자 245명 이상, 총 커밋 1,475건을 기록했습니다 [S01]. 4개월 기준으로 환산하면 월평균 약 8,000 stars, 300건 이상의 커밋이 쌓인 셈입니다. 이 속도는 단순한 관심 지표를 넘어서, 실제로 코드를 작성하고 기여하는 개발자 집단이 형성됐음을 보여줍니다. stars 는 "관심을 표시한 수"에 가깝지만, 245명이 넘는 기여자 수는 실제 작업 참여 인원을 의미하기 때문입니다.

릴리스 속도: 활발성이자 변동 리스크

v0.17.0 이라는 버전 번호는 4개월 만에 17번 이상의 마이너 릴리스가 이뤄졌음을 시사합니다 [S01]. 이는 개발 활발성을 나타내는 동시에, API 인터페이스와 설정 형식이 빠르게 변경될 가능성도 내포합니다. 실제로 경쟁 프레임워크인 OpenClaw 는 잦은 업데이트로 인해 실행 중인 인스턴스가 손상되는 사례가 보고된 바 있습니다 [S04]. Hermes Agent 는 최근 릴리스 주기에서 안정성을 비교적 잘 유지하고 있다는 평가를 받고 있으나 [S04], PoC 에서 프로덕션으로 전환할 때는 버전 고정과 업데이트 일정 관리 정책을 수립하는 것이 권장됩니다.

커뮤니티 활력 지표 정리

지표	2026-06 기준	해석
GitHub stars	32,000+	높은 초기 관심도
기여자 수	245+	실질 개발 참여자 다수
총 커밋 수	1,475	4개월 기준 활발한 코드 변경
최신 버전	v0.17.0 (2026-06-19)	빠른 릴리스 주기 — 버전 고정 정책 권장
동작 환경 범위	VPS ~ GPU 클러스터	낮은 하드웨어 진입 장벽

1.3.2 OpenClaw · Paperclip 과의 stars · 채널 수 상대 비교

세 프레임워크의 포지셔닝 차이

같은 시기 주목받는 오픈소스 에이전트 프레임워크로 OpenClaw 와 Paperclip 을 꼽을 수 있습니다. OpenClaw 는 247,000건의 GitHub stars 와 24개 메시징 채널 지원을 갖춘 프레임워크로, 다수의 통합 채널과 멀티모델 지원이 강점입니다 [S04]. 구조 면에서 OpenClaw 는 "게이트웨이 안에 에이전트"를 배치하는 방식, 즉 메시징 허브를 중심으로 에이전트 기능을 추가하는 아키텍처를 택합니다. Hermes Agent 는 이와 달리 "게이트웨이 위에 학습 루프"를 얹는 구조로, 메시징 채널보다 자기 개선과 개인화에 설계 우선순위를 둡니다 [S04].

Paperclip 과의 비교: 직접 경쟁이 아닌 역할 분리

Paperclip 은 2026년 3월 공개된 오픈소스 에이전트 오케스트레이션 도구로, 3주 만에 30,000 stars 를 달성하고 현재 43,000건 이상을 기록 중입니다 [S05]. MIT 라이선스를 채택하고 있으며, 철학은 "조직 메타포"입니다. Paperclip 은 Claude Code, OpenClaw, Python 스크립트 같은 기존 에이전트를 "직원"으로 채용하고 관리하는 역할을 합니다. 즉, Paperclip 자체는 에이전트를 만들지 않고 기존 에이전트들을 오케스트레이션합니다 [S05]. Hermes Agent 와 Paperclip 은 직접적인 기능 경쟁 관계가 아니라 보완적 위치에 있습니다. Hermes 는 단일 에이전트로서 자기 개선과 다채널 연결을 수행하고, Paperclip 은 여러 에이전트를 조직화하는 상위 레이어 역할을 합니다.

3종 GitHub 지표 비교

프레임워크	GitHub stars	주요 특징	카테고리
OpenClaw	247,000+	24개 채널, 멀티모델, 엔터프라이즈 통합	메시징 게이트웨이 중심 에이전트
Paperclip	43,000+	기존 에이전트 오케스트레이션, Node.js	다중 에이전트 조직화
Hermes Agent	32,000+	자기 개선, Persistent Memory, 6개 채널	자기 학습형 단일 에이전트

stars 기준 절대 수치만 보면 Hermes Agent 는 세 프레임워크 중 가장 낮은 위치에 있습니다. 그러나 공개 시점을 함께 고려하면 해석이 달라집니다. Hermes Agent 가 2026년 2월, Paperclip 이 같은 해 3월에 공개됐음을 감안하면, 이 두 프레임워크는 OpenClaw 대비 훨씬 짧은 기간에 해당 수치에 도달한 것입니다. 카테고리도 다릅니다. OpenClaw 가 범용 메시징 허브로 성장한 반면, Hermes Agent 는 자기 개선형 에이전트라는 특화된 포지션을 가집니다.

1.3.3 국내 기업·공공기관의 도입 적정 시점 판단 기준

신생 오픈소스 도입 시점 판단의 세 기준

오픈소스 소프트웨어를 프로덕션 환경에 도입하는 적정 시점을 판단할 때 통상적으로 세 가지 기준을 적용합니다. 첫째는 커뮤니티 임계점으로, 기여자 수와 이슈 처리 속도가 일정 수준 이상이어야 유지보수 지속성을 기대할 수 있습니다. Hermes Agent 는 245명 이상의 기여자와 활발한 릴리스 주기를 갖추어 이 기준을 충족합니다 [S01]. 둘째는 릴리스 안정성으로, API 인터페이스와 설정 형식이 어느 정도 수렴했는지를 확인합니다. v0.17.0 수준의 버전 번호는 여전히 빠른 변화 중임을 의미하므로, 버전 고정 정책 수립이 권장됩니다. 셋째는 국내 도입 사례 출현 여부로, 국내 언어·규제 환경에서 실제 운영한 참조 사례가 있으면 도입 결정의 확신도를 높입니다.

2026년 도입 결정을 뒷받침하는 환경 요인

국내 공공기관 환경에서는 2026년 공공기관 경영평가편람에 AI 활용 가점이 신설됐습니다 [S12]. 이는 AI 도구 도입이 단순한 기술 선택을 넘어 기관 평가 항목에 반영되기 시작했음을 의미합니다. 공공기관 10곳 중 7곳이 이미 AI 를 활용하고 있다는 현황 데이터 [S12]는, 도입을 "선도 기관만의 실험"으로 보는 시각이 더 이상 유효하지 않음을 보여줍니다. 국내 기업 환경에서는 오픈소스 AI 에이전트를 활용한 사례가 점진적으로 증가하고 있으며, MSAP.ai 와 같이 오픈소스 AI 에이전트 기술을 국내 IT 환경에 맞게 통합한 플랫폼을 활용하는 방식이 PoC 부담을 낮추는 현실적 진입 경로가 될 수 있습니다.

분기별 도입 의사결정 체크포인트

시점	권장 행동	판단 근거
2026년 2~3분기 (현재)	PoC 시작 + 버전 고정 정책 수립	커뮤니티 임계점 충족, MIT 라이선스 확인 완료, 설치 진입 장벽 낮음

시점	권장 행동	판단 근거
2026년 4분기	PoC 결과 검토 + 프로덕션 투입 범위 결정	국내 도입 참조 사례 수집, 릴리스 안정성 추가 확인
2027년 1분기 이후	프로덕션 전환 또는 기각 결정	v1.0 이상 릴리스 여부, SLA 정의 가능성 확인

현재 시점(2026년 2~3분기)은 Hermes Agent 의 PoC 를 시작하기에 적합한 단계입니다. 커뮤니티 임계점은 이미 넘어섰고, MIT 라이선스 확인이 비교적 빠르게 완료되며, 단일 `curl` 설치로 당일 내 환경 구성이 가능합니다. 반면 v0 단계라는 점에서 API 변경 위험이 존재하므로, 핵심 업무 프로세스에 즉시 투입하기보다는 격리된 PoC 환경에서 기능을 검증한 뒤 4분기 전환 여부를 판단하는 2단계 접근이 현실적입니다 [S01][S04][S12].

2장. 기업·기관에서 Hermes Agent 가 가능한 이유 — 기술적 근거

기업과 공공기관이 AI 에이전트 도입을 검토할 때 가장 먼저 부딪히는 질문은 "우리 데이터가 외부로 나가지 않느냐"입니다. 클라우드 기반 상용 에이전트는 이 질문에 쉽게 답하지 못합니다. 사용자 프롬프트가 외부 API를 거치는 구조 자체가 데이터 외부 송신을 전제하기 때문입니다. Hermes Agent는 이 구조적 전제를 뒤집습니다. 사내망 안에서 LLM 추론부터 결과 저장까지 완결되도록 설계되어 있으며, 컴포넌트 교체·확장·감사 추적 모두 사내 통제권 안에 둡니다. 본 장은 이 주장을 데이터 주권, 모듈러 아키텍처, 보안·감사·인프라 친화성 네 가지 기준에서 기술적으로 검증합니다.

2.1 데이터 주권 — 외부 호출 없는 자체 운영 구조

데이터 주권이란 사용자 입력·추론 결과·학습 자산이 어느 시스템에 머무는지를 운영자가 결정할 수 있는 권한을 뜻합니다. 공공기관과 금융권은 이 권한을 법령과 내부 보안 정책 수준에서 요구합니다. 외부 API 호출이 단 한 번이라도 허용되면 데이터 주권 요건은 충족되지 않습니다. Hermes Agent가 이 절에서 설명할 세 가지 구조적 특성—외부 호출 제로 아키텍처, sLLM 결합 시 데이터 흐름, 망분리 환경 동작 가능성—은 바로 이 요건을 정면으로 겨냥합니다.

2.1.1 Hermes + Local LLM + LiteLLM 의 외부 호출 0 아키텍처

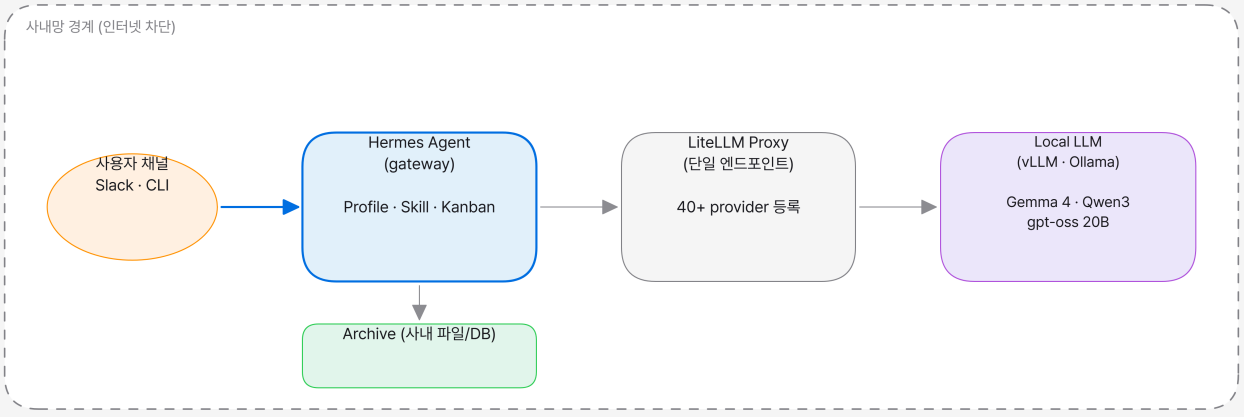
사내망 완결 구조의 핵심 원리

Hermes Agent는 MIT 라이선스 오픈소스로 Nous Research가 2026년 2월 공개했습니다 [S01]. 설치는 단일 `curl` 명령으로 의존성까지 자동 구성되며, 월 5달러 VPS부터 GPU 클러스터, 서버리스까지 동작 범위가 넓습니다 [S01]. 핵심은 Hermes 에이전트 본체, Local LLM (Gemma 4 31B Dense · Qwen3 시리즈 · gpt-oss 20B), LiteLLM 프록시(proxy, LLM 요청을 단일 엔드포인트로 모아 라우팅하는 게이트웨이)가 모두 사내망 경계 안에 배치된다는 점입니다. 이 세 컴포넌트가 사내망 안에 자리 잡으면, 사용자 요청은 사내망 경계를 벗어나지 않고 추론 결과까지 반환됩니다.

외부 호출 제로의 기술적 의미

LiteLLM 프록시는 100개 이상의 LLM 공급자(provider)를 단일 OpenAI 호환 엔드포인트로 추상화하는 오픈소스 게이트웨이입니다 [S10]. 클라우드 LLM을 쓸 때는 이 게이트웨이가 외부 API를 호출하지만, 사내에 vLLM 또는 Ollama로 띄운 Local LLM을 엔드포인트로 지정하면 외부 호출이 발생하지 않습니다. Hermes의 `setup` 명령은 40개 이상의 공급자를 네이티브(native, 별도 설정 없이 바로) 등록하며, 이 중 Local LLM 엔드포인트를 지정하는 것이 외부 호출 차단의 설정 단위입니다 [S10]. 즉, "외부 호출 0"은 별도 방화벽 정책이 아니라 LiteLLM 엔드포인트 설정 하나로 제어됩니다.

그림 2-1. Hermes + Local LLM + LiteLLM — 외부 호출 0 아키텍처



요청 흐름: 사용자 → Hermes → LiteLLM → Local LLM → Hermes → Archive. 어느 구간도 사내망 경계를 넘지 않음 [S01][S10]

사내망 경계 내 Hermes + LiteLLM + Local LLM 아키텍처 — 사내망 경계(점선) 바깥으로 나가는 화살표가 존재하지 않음을 확인할 수 있습니다.

위 구조에서 요청 흐름은 다음과 같습니다. 사용자 채널(Slack, 사내 메신저 CLI 등)에서 입력이 들어오면 Hermes 게이트웨이 프로세스가 수신하고, LiteLLM 프록시를 통해 사내 Local LLM에 추론을 요청합니다. 추론 결과는 다시 Hermes로 돌아와 Archive에 저장되고 채널을 통해 응답됩니다. 이 경로 어디에도 사내망 경계를 넘는 구간이 없습니다. 공공·금융 도입의 첫 번째 관문인 "외부 호출 0" 요건이 아키텍처 수준에서 충족됩니다.

2.1.2 sLLM 결합으로 얻는 데이터 주권의 구체적 수준

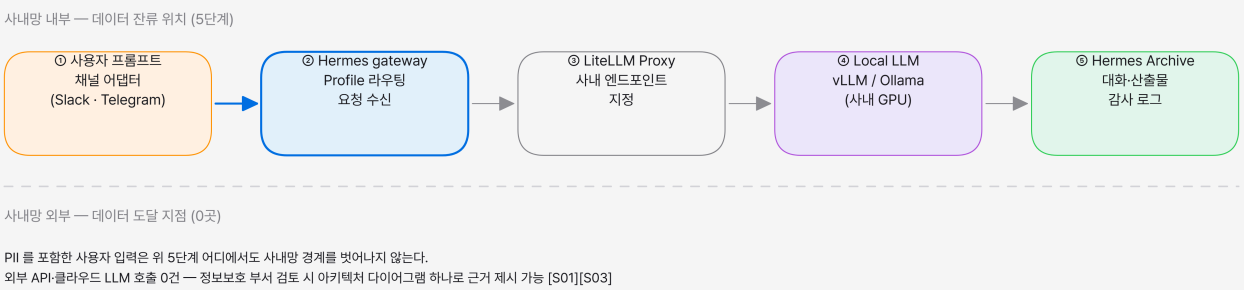
sLLM 이 데이터 주권을 확보하는 구조

소형 언어 모델(sLLM, small Language Model)은 수십억 파라미터 규모로 단일 GPU 또는 CPU 서버에서 추론이 가능한 오픈웨이트 LLM을 가리킵니다. Hermes Agent의 대표 sLLM 세 종은 Gemma 4 31B Dense (Google, Gemma Terms of Use), Qwen3 시리즈 (Alibaba, Apache 2.0 라이선스로 상업적 이용 허용), gpt-oss 20B (OpenAI, Apache 2.0 라이선스로 상업적 이용 허용)입니다 [S07] [S08] [S09]. 이 모델들은 가중치(weight, 모델의 학습된 파라미터 파일)를 사내 서버에 내려받아 운영하므로, 추론 시 외부 서버에 연결되지 않습니다.

사용자 입력부터 Archive 저장까지 데이터 흐름

데이터가 머무는 위치를 단계별로 정리하면 다음과 같습니다. 첫째, 사용자 프롬프트는 채널 어댑터(Telegram Bot, Slack Webhook, 사내 메신저 API 등)를 통해 Hermes 게이트웨이에 도달합니다. 이 시점에 데이터는 이미 사내망 안에 있습니다. 둘째, Hermes는 해당 요청을 LiteLLM 프록시로 전달하고, LiteLLM은 사내 Local LLM 엔드포인트(vLLM 또는 Ollama 서버)에 추론을 요청합니다. 셋째, 추론 결과는 사내 Local LLM 서버에서 생성되어 Hermes로 반환됩니다. 넷째, Hermes는 해당 대화 기록과 산출물을 Archive에 저장하는데, Archive 역시 사내 파일시스템 또는 사내 데이터베이스에 위치합니다. 개인식별정보(PII, Personally Identifiable Information)를 포함한 사용자 입력이 이 네 단계 어디에서도 사내망을 벗어나지 않습니다 [S01] [S03].

그림 2-2. 사용자 입력 → Hermes → LiteLLM → Local LLM → 응답 → Archive 데이터 흐름



사용자 입력 → Hermes → LiteLLM → Local LLM → 응답 → Hermes Archive 데이터 흐름도 — 각 단계가 모두 "사내망 내부" 레인에 위치함

이 흐름이 정보보호 부서 검토에 주는 의미는 명확합니다. 데이터가 머무는 위치가 운영자가 직접 통제하는 서버들뿐이므로, 망 외부 전송 여부를 외부 감사자에게 증명할 때 아키텍처 다이어그램 하나로 근거를 제시할 수 있습니다. 클라우드 API 계약서에서 데이터 처리 위치 조항을 찾아야 하는 상용 솔루션 대비 검토 공수가 크게 줄어듭니다.

2.1.3 망분리·CSAP·금융 망분리 환경에서 동작 가능성

망분리 환경이 제기하는 설치 제약

국내 공공기관은 업무망과 인터넷망을 물리적으로 분리하는 망분리 체계를 운영합니다. 금융권은 금융감독원 전자금융감독규정에 따라 내부 시스템과 외부 인터넷 간 논리·물리 분리를 요구합니다. CSAP(Cloud Service Assurance Program, 클라우드 보안 인증 제도) 인증 환경 역시 외부 인터넷 직접 접근을 제한합니다. Hermes Agent의 기본 설치 방식은 단일 `curl` 명령으로 외부 저장소에서 패키지를 내려받는 구조이므로 [S01], 망분리 환경에서는 이 설치 경로를 대체해야 합니다.

망분리 환경에서 동작 가능한 세 가지 설치 경로

첫 번째는 사내 패키지 미러(mirror, 외부 저장소의 패키지를 사내 서버에 복제한 저장소) 구성입니다. 인터넷 영역 서버가 PyPI·GitHub 릴리스를 사내 Nexus 또는 Artifactory 미러에 동기화하고, 내부망 설치 명령어가 이 미러를 참조하도록 `pip install --index-url` 또는 환경 변수를 재지정합니다 [S12]. 두 번째는 오프라인 타볼(tarball, 의존성 패키지를 모두 포함한 압축 아카이브) 방식입니다. 인터넷 영역에서 Hermes와 모든 의존성을 번들(bundle)로 패키징한 뒤, 보안 매체 이동 절차(USB 반출 또는 DMZ 중계 서버)를 통해 내부망 서버에 복사하고 설치합니다. 세 번째는

컨테이너 이미지 이전(pre-pulled image transfer) 방식입니다. Hermes Agent를 Docker 이미지로 빌드한 뒤 이미지 아카이브를 내부망 레지스트리로 반입하고 `docker load`로 구동합니다. 세 경로 모두 망분리 보안 정책을 위반하지 않습니다.

공공·금융 도입 장벽 완화의 실질적 의미

LLM Capsule이 분석한 국내 공공기관 AI 도입 현황에 따르면 sLLM 자체 구축 초기 비용이 7~13억 원 수준이며, 망분리·CSAP·의료법·개인정보보호법이 대표적 도입 장벽입니다 [S12]. Hermes Agent는 이 장벽 중 "망분리 환경 동작 가능성"이라는 기술적 항목을 위의 세 경로로 충족합니다. 나머지 장벽(CSAP 인증·보안 감사·개인정보 처리 방침)은 Hermes 자체의 기능보다 운영 정책 수립 문제이지만, Hermes가 자체 감사 추적 기능(2.3절 참조)을 내장한다는 점은 보안 심사 공수를 줄이는 요인으로 작용합니다. 2026년 공공기관 경영평가편람에 AI 활용 가점이 신설되었으며 [S12], 이는 망분리 환경에서 동작 가능한 AI 에이전트 솔루션에 대한 수요를 더욱 구체화합니다.

2.2 모듈러 아키텍처 — 교체·확장 가능한 컴포넌트 설계

벤더 종속(vendor lock-in, 특정 공급업체 기술에 과도하게 의존하여 교체 비용이 크게 증가하는 상태)은 IT 의사결정자가 새로운 기술을 도입할 때 가장 우려하는 리스크 중 하나입니다. Hermes Agent의 컴포넌트 설계는 이 우려에 직접 답합니다. LLM 공급자, 사내 업무 로직(Skill), 커뮤니케이션 채널 세 영역이 각각 독립적으로 교체·확장 가능한 구조여서, 어느 한 컴포넌트를 바꾸더라도 나머지에 영향을 미치지 않습니다.

2.2.1 LiteLLM 분리로 얻는 LLM 교체 자유

100개 이상 LLM 공급자를 단일 엔드포인트로

LiteLLM은 OpenAI, Anthropic Claude, AWS Bedrock, Google Vertex AI, Cohere, HuggingFace, vLLM, NVIDIA NIM 등 100개 이상의 LLM 공급자를 단일 OpenAI 호환 API 엔드포인트로 통합하는 오픈소스 게이트웨이입니다 [S10]. Hermes Agent는 이 게이트웨이를 중간 계층으로 사용하므로, Hermes 코드 자체는 LLM 공급자에 대한 직접 의존이 없습니다. Netflix, Lemonade, Rocket Money 등이 실제 운영 환경에서 LiteLLM을 활용한 사례가 보고되어 있으며 [S10], 이는 대규모 트래픽 환경에서의 안정성을 방증합니다.

모델 교체 = 설정 변경

사용 중인 LLM을 바꾸는 작업이 코드 수정 없이 LiteLLM 설정 파일 한 줄 변경으로 완결됩니다. 예를 들어 Qwen3를 쓰다가 gpt-oss 20B로 전환할 때 Hermes 코드를 건드리지 않고 `litellm.yaml`의 모델 엔드포인트 항목만 수정하면 됩니다. 이 운영적 유연성은 모델 시장이 빠르게 변하는 환경에서 특히 중요합니다. 지금 가장 적합한 모델이 6개월 뒤에도 최선이 아닐 수 있고, Hermes + LiteLLM 구조에서는 그 전환 비용이 사실상 0에 가깝습니다.

LLM 공급자 유형	대표 모델	외부 호출 여부	라이선스 비고
클라우드 상용	OpenAI GPT-4o, Anthropic Claude	외부 호출 발생	상용 API 요금

LLM 공급자 유형	대표 모델	외부 호출 여부	라이선스 비고
클라우드 관리형	AWS Bedrock, Google Vertex AI	외부 호출 발생	클라우드 계약
사내 Local LLM	Qwen3, gpt-oss 20B (vLLM / Ollama)	외부 호출 없음	Apache 2.0
사내 Local LLM	Gemma 4 31B Dense (vLLM / Ollama)	외부 호출 없음	Gemma Terms of Use — 도입 전 법무 검토 필요

LiteLLM OSS vs Enterprise 선택 기준

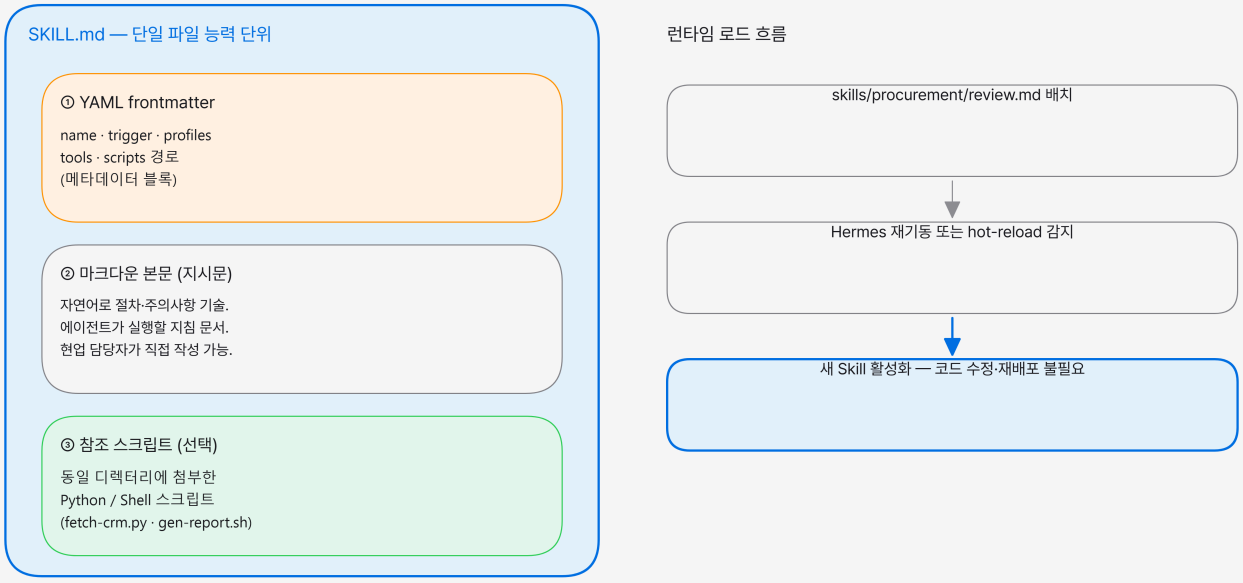
LiteLLM MIT 라이선스 오픈소스 버전은 기본 게이트웨이 기능, 가상 키(virtual key, 공급자별 실제 API 키를 감싸는 내부 키), 요청/응답 로깅, 예산 추적을 제공합니다 [S10]. 기업 환경에서 싱글 사인온(SSO), 역할 기반 접근 통제(RBAC), 세부 감사 로그가 필요하다면 LiteLLM Enterprise 에디션(월 $250~연 $30,000 수준)을 고려할 수 있습니다 [S10]. PoC(Proof of Concept, 개념 검증) 단계에서는 OSS 버전으로도 핵심 기능 검증이 가능하므로, 예산 투입 시점을 본격 운영 전환 단계로 늦출 수 있습니다.

2.2.2 Skill 시스템이 제공하는 사내 업무 로직 확장 자유

Skill = 마크다운 파일로 작성하는 업무 로직 단위

Hermes Agent의 Skill(스킬, 에이전트가 실행할 수 있는 업무 로직의 최소 단위)은 마크다운 파일에 YAML 프론트매터(frontmatter, 파일 상단의 메타데이터 블록)와 실행 메타데이터를 담아 구성합니다 [S01] [S03]. 파일 구조는 단순합니다. 프론트매터에 스킬 이름, 트리거 조건, 사용할 도구 목록을 정의하고, 마크다운 본문에 에이전트가 참조할 지시문(instruction)을 작성한 뒤, 필요하면 동일 디렉터리에 파이썬 스크립트나 셸 스크립트를 첨부합니다. Hermes는 런타임에 이 파일을 읽어 동적으로 업무 로직을 로드합니다.

그림 2-3. Skill 파일 구조 — YAML frontmatter + 마크다운 지시문 + 참조 스크립트



출처: Hermes Agent 공식 문서 [S01][S03] — SKILL.md 단일 파일이 능력 단위, 디렉터리 추가만으로 로직 확장

Skill 파일 구조 — YAML 프론트매터(트리거·도구 목록) + 마크다운 지시문 + 선택적 참조 스크립트로 구성

코드 수정 없이 업무 로직을 추가하는 방식

Hermes 코어(core) 코드를 수정하거나 배포 파이프라인을 거치지 않고, `skills/` 디렉터리에 파일을 추가하는 것만으로 새 업무 로직이 활성화됩니다 [S03]. 가령 구매 요청 검토 절차를 Skill로 작성해 `skills/procurement/review-request.md`에 배치하면, Hermes가 다음 기동 시 또는 핫리로드(hot-reload, 재기동 없이 변경 사항 반영) 설정에 따라 즉시 해당 로직을 활용 가능한 상태로 등록합니다. MSAP.ai와 같은 국내 통합 플랫폼이 제공하는 워크플로우 자동화 기능을 Hermes의 Skill 레이어에서 보완하거나 대체하는 방식으로 두 시스템을 연계하는 구성도 가능합니다.

현업 부서가 직접 Skill을 작성할 수 있는가

이 질문은 도입 확산의 분기점입니다. 마크다운과 YAML 작성 역량이 있는 직원이라면 별도의 소프트웨어 개발 경험 없이도 Skill을 구성할 수 있습니다. 이는 IT 팀이 모든 업무 자동화 요청을 수용해야 하는 병목을 줄이고, 현업 부서가 자체적으로 업무 로직을 작성·실험하는 분산 확장 모델을 가능하게 합니다. 초기 PoC 단계에서 Skill 작성 워크숍을 반나절 과정으로 운영하면 현업 주요 부서 담당자 수준에서 기본 Skill 작성 역량을 갖출 수 있습니다.

2.2.3 채널 어댑터 교체 (Telegram → 사내 메신저) 가능성

기본 제공 6개 채널과 국내 기업 현실의 간극

Hermes Agent는 Telegram, Discord, Slack, WhatsApp, Signal, CLI(명령줄 인터페이스) 6개 채널을 기본 지원합니다 [S01]. 그러나 국내 기업 다수는 외부 메신저 서비스를 보안 정책상 업무망에서 차단합니다. Telegram이나 Discord는 국내 기업 환경에서 업무 채널로 활용하기 어렵습니

다. 잔디(Jandi), 카카오워크(KakaoWork), 네이버웍스(NAVER Works), Lotus Notes 연동이 필요한 환경이 실제 도입 요건입니다.

채널 어댑터 구조와 사내 메신저 연동 방식

Hermes의 채널 어댑터(channel adapter)는 각 메신저 플랫폼의 웹훅(webhook) 또는 봇(bot) API를 Hermes 내부 메시지 포맷으로 변환하는 독립 모듈입니다. 기존 어댑터 코드를 참조해 사내 메신저의 수신·발신 API 명세에 맞는 어댑터 파일을 작성하면 됩니다. 추가 어댑터는 기존 Hermes 코어에 변경을 가하지 않고 플러그인 형태로 등록됩니다. 잔디·카카오워크와 같이 REST 기반 웹훅을 제공하는 사내 메신저는 어댑터 구현 공수가 통상 3~5일(PoC 수준) 수준으로 추산됩니다.

채널	기본 지원	국내 기업 적합성
Telegram Bot	기본 제공	보안 정책 차단 가능성 높음
Slack	기본 제공	사용 중인 기업에서 활용 가능
Discord	기본 제공	업무 환경 적합성 낮음
CLI	기본 제공	개발·운영팀 직접 사용 가능
잔디(Jandi)	어댑터 작성 필요	REST Webhook 기반 구현 공수 3~5일
카카오워크	어댑터 작성 필요	REST Bot API 기반 구현 공수 3~5일
NAVER Works	어댑터 작성 필요	REST Bot API 기반 구현 공수 3~5일

PoC 단계 산출물로 사내 메신저 어댑터 구현 결과물을 포함하면, 본격 운영 전환 시 채널 교체 리스크를 사전에 해소할 수 있습니다.

2.3 보안·감사·인프라 친화성

데이터 주권과 모듈러 아키텍처가 기술적 가능성을 보여준다면, 보안·감사·인프라 친화성은 실제 운영 환경에서의 지속 가능성을 뒷받침합니다. 내부 감사와 외부 감사 모두 에이전트의 행동 이력을 추적 가능한 형태로 요구하고, 사용자별 권한 분리를 RBAC(Role-Based Access Control, 역할 기반 접근 통제) 수준에서 구현해야 하며, 기존 인프라 위에 최소한의 추가 구성으로 올려야 합니다. Hermes Agent의 Kanban·Archive 감사 추적 구조, Profile 기반 권한 모델, 세 가지 배포 시나리오는 이 요건들을 각각 직접 충족합니다.

2.3.1 Kanban 보드와 Archive 가 제공하는 감사 추적 능력

Kanban 보드가 기록하는 작업 이력

Hermes Agent의 Kanban 보드는 SQLite 기반의 지속 가능(durable) 태스크(task) 보드로, 모든 작업 카드의 상태 전이를 시계열로 기록합니다 [S03]. 카드(card)에는 담당 프로필(Profile) 명, 의

존 관계 링크, 작업 공간 종류(scratch / worktree / dir:path), 테넌트(tenant, 멀티 사용자 환경에서 사용자 격리 단위) 네임스페이스가 함께 저장됩니다 [S03]. 컬럼 구조는 triage(분류) → todo(예정) → ready(준비 완료) → running(실행 중) → blocked(차단됨) → done(완료) → archived(보관)로 정의되어 있으며, 각 전이 시각과 트리거 프로필이 기록됩니다. 이 로그는 "누가, 언제, 어떤 작업을, 어느 상태로 바꾸었는가"를 감사자가 재구성할 수 있는 충분한 정보를 담습니다.

Archive 가 보존하는 대화 및 산출물

Archive는 Hermes가 처리한 대화 내용, 에이전트가 생성한 산출물(코드, 보고서, 요약문), 사용한 도구 호출 기록을 파일 또는 데이터베이스 형태로 보존합니다 [S02] [S03]. 이 저장 구조는 "에이전트가 무엇을 근거로 어떤 결과를 냈는가"를 사후 재현할 수 있게 합니다. 금융·공공 감사에서 요구하는 "AI 의사결정 근거 제시" 요건에 직접 대응합니다.

감사 추적 항목	저장 위치	내보내기 형식
작업 카드 상태 전이 이력	Kanban (SQLite)	JSON, CSV
담당 프로필·타임스탬프	Kanban (SQLite)	JSON, CSV
대화 내용 전문	Archive (파일/DB)	마크다운, JSON
에이전트 산출물	Archive (파일/DB)	원본 파일 형식
도구 호출 기록	Archive (파일/DB)	JSON

감사 로그 보존 기간은 Hermes 설정 또는 Archive 디렉터리 정책으로 운영자가 직접 지정합니다. 외부 감사 일정에 맞춰 특정 기간 Archive를 JSON·CSV로 내보내는 절차를 PoC 단계에서 검증해두면 본격 운영 전환 시 감사 대응 공수를 크게 줄일 수 있습니다.

2.3.2 Profile 분리로 구현하는 사용자·역할 권한 모델

Profile 이 제공하는 격리 단위

Hermes Agent의 Profile(프로필, 목적별로 독립된 에이전트 인스턴스 설정 묶음)은 각각 별도의 메모리, Skill 집합, 도구(Tool) 접근 권한을 갖습니다 [S03]. 예를 들어 coding assistant 프로필은 코드 실행 도구와 git 연동 Skill에 접근할 수 있지만, personal bot 프로필은 해당 도구를 참조하지 않는 방식으로 격리됩니다. 프로필 간 메모리가 교차하지 않으므로, 한 프로필에서 처리한 민감 정보가 다른 프로필의 컨텍스트에 혼입되는 상황이 구조적으로 차단됩니다 [S03].

RBAC 매핑 절차

기업 환경의 RBAC 요건은 "역할(Role)에 따라 접근 가능한 기능과 데이터가 달라야 한다"는 것입니다. Hermes의 Profile 모델을 이 요건에 매핑하는 방식은 다음과 같습니다. 운영자가 사내 역할(예: 일반 직원, 팀장, HR 담당자, 보안 담당자)별로 Profile을 생성하고, 각 Profile에 허용할 Skill 목록과 도구 접근 범위를 설정합니다. 그런 다음 인증 시스템(사내 LDAP 또는 SSO)과 Profile 식별자를 연동하면, 로그인한 사용자가 자신의 역할에 맞는 Profile로 자동 연결됩니다.

역할(Role)	Hermes Profile	허용 Skill 예시	접근 불가 Skill 예시
일반 직원	employee	문서 요약, 일정 관리	인사 데이터 조회, 보안 설정
팀장	manager	문서 요약, 팀 보고서 생성, 인사 조회	보안 설정, 시스템 관리
HR 담당자	hr-agent	인사 데이터 조회, 채용 분석	시스템 관리, 코드 실행
IT 관리자	it-admin	전체 Skill 접근	(제한 없음)

현재 Hermes OSS 버전의 Profile 격리는 파일 시스템 수준의 Skill 디렉터리 분리와 메모리 네임스페이스 분리로 구현되며, 사내 SSO 연동은 운영자가 별도로 구성해야 합니다. 완전한 RBAC 요건을 충족하려면 PoC 단계에서 SSO 연동 공수와 Profile 관리 절차를 함께 설계해야 합니다.

2.3.3 온프레미스·하이브리드·VPC 배포 3 시나리오 비교

배포 시나리오 선택이 인프라 의사결정의 핵심

"Hermes Agent를 어디에 두는가"라는 질문은 비용, 네트워크 지연, 보안 트레이드오프를 동시에 결정합니다. 인프라 부서가 PoC 사전 검토 단계에서 가장 먼저 묻는 항목이기도 합니다 [S01]. Hermes Agent는 월 5달러 VPS부터 GPU 클러스터까지 동작하므로 [S01], 배포 위치의 하드웨어 요건이 사실상 Local LLM의 요건에 따라 결정됩니다. 세 시나리오를 기준별로 정리합니다.

기준	온프레미스 단일 서버	하이브리드 (Hermes 사내 + LLM 클라우드)	VPC 단일 클라우드
데이터 외부 송신	없음 (완전 차단)	LLM 추론 시 클라우드 전송 발생	클라우드 VPC 내부로 한정
초기 구성 비용	GPU 서버 구매 비용 포함	GPU 서버 불필요, API 요금 발생	클라우드 인스턴스 요금
네트워크 지연	최저 (내부 통신)	LLM API 왕복 지연 (수백ms)	VPC 내 지연 (낮음)
인프라 운영 책임	전적으로 자사	LLM 부분은 클라우드 공급자	클라우드 공급자와 공동
망분리 환경 적합성	최적	제한적 (LLM 외부 호출 불가)	조건부 (VPC 승인 필요)
권장 대상	공공기관·금융권·고보안 환경	클라우드 API 허용 기업, 비용 우선	이미 클라우드 VPC 기반 운영 중인 기업

온프레미스 시나리오

단일 서버에 Hermes, LiteLLM, Local LLM을 모두 배치합니다. GPU 서버(RTX 3090급 단일 GPU, VRAM 24GB)에서 Gemma 4 31B Dense 또는 gpt-oss 20B를 Q4_K_M 양자화로 구동하면 16GB VRAM 이내에서 추론이 가능합니다 [S07] [S09]. Idle 상태에서는 비용이 사실상 0에 가까우며 [S01], 서버 구매 이후 추가 과금이 없습니다. 공공기관과 금융권이 데이터 외부 송신 요건을 엄격히 적용한다면 이 시나리오가 가장 단순한 선택입니다.

하이브리드 시나리오

Hermes Agent 본체와 Kanban·Archive를 사내 서버에 두고, LLM 추론만 클라우드 API(OpenAI, Anthropic 등)를 사용합니다. GPU 서버 초기 투자 없이 LLM 성능을 최대로 활용할 수 있고, 클라우드 LLM과 사내 Local LLM을 LiteLLM의 fallback 기능으로 혼합 운영할 수도 있습니다 [S10]. 다만 LLM 추론 시 사용자 프롬프트가 클라우드 API로 전송되므로, 데이터 외부 송신이 허용되는 정보 등급에만 적용할 수 있습니다.

VPC 단일 클라우드 시나리오

Hermes, LiteLLM, Local LLM을 모두 클라우드 공급자의 VPC(Virtual Private Cloud, 논리적으로 격리된 가상 사설 클라우드 네트워크) 안에 배치합니다. 데이터는 VPC 경계 안에서만 이동하고, 퍼블릭 인터넷으로 나가지 않습니다. 이미 AWS·GCP·Azure VPC 기반으로 워크로드를 운영 중인 기업이라면 기존 네트워크 정책과 IAM(Identity and Access Management, 클라우드 자원 접근 통제 체계)을 Hermes에 그대로 적용할 수 있어 도입 공수가 줄어듭니다. MCP(Model Context Protocol, AI 모델과 외부 도구·데이터를 연결하는 개방 표준)는 2025년 12월 Linux Foundation 산하로 거버넌스가 이관되어 vendor-neutral 표준으로 자리잡았으며 [S11], VPC 시나리오에서 사내 시스템을 MCP 서버로 노출하면 Hermes가 이를 클라이언트로 호출하는 방식으로 사내 시스템 연동이 표준화됩니다.

세 시나리오 모두에서 Hermes Agent 자체의 코드는 동일하고, 배포 위치와 LiteLLM 엔드포인트 설정만 달라집니다. 조직의 인프라 현황, 보안 정책, 예산에 따라 시나리오를 선택하거나 단계적으로 전환할 수 있습니다. 데이터 주권 요건이 가장 엄격하다면 온프레미스에서 시작하고, 이후 클라우드 LLM 허용 범위가 확대되면 하이브리드 또는 VPC로 이전하는 경로가 현실적입니다.

3장. 핵심 구성 요소 비교 — Paperclip · OpenClaw · Harness · CrewAI · Hermes

오픈소스 AI 에이전트 영역에는 현재 다섯 가지 주요 계층이 공존합니다. 어떤 솔루션은 채널 폭을 극대화하고, 어떤 솔루션은 기존 에이전트를 조직화하며, 어떤 솔루션은 역할 기반 협업을 지향합니다. Hermes Agent 는 이 흐름과 다른 자리를 점유합니다 — 단일 에이전트가 사용할수록 스스로를 개선하는 구조입니다. 3장은 다섯 솔루션을 정량·정성 두 층위에서 비교하고, 그 결과를 조직 실무에 곧바로 적용할 수 있는 의사결정 형태로 정리합니다.

3.1 5종 비교 매트릭스 — 기능·라이선스·운영 모델·확장성

비교를 시작하기 전에 대상 다섯 종의 정체를 명확히 해 둘 필요가 있습니다. 다섯 솔루션은 동일한 추상 계층에 있지 않습니다. Paperclip 은 기존 에이전트를 '직원'처럼 배치하는 오케스트레

이션 계층이고, OpenClaw 는 메시징 플랫폼 통합에 특화된 게이트웨이 중심 에이전트입니다. Harness Engineering 은 제품명이 아닌 패러다임 범주이며, CrewAI 는 역할 기반 멀티에이전트 협업 프레임워크, Hermes 는 단일 에이전트가 지속 기억과 자기 개선을 결합한 자율 에이전트입니다. 이 계층 차이를 인식한 뒤 비교 매트릭스를 읽어야 선택 근거가 명확해집니다.

3.1.1 Paperclip · OpenClaw · Harness Engineering · CrewAI · Hermes 5종 카드

Paperclip — 조직 메타포 오케스트레이터

Paperclip 은 AgencyEnterprise 가 2026년 3월 공개한 오픈소스 에이전트 오케스트레이션 도구입니다 [S05]. 공개 3주 만에 GitHub 스타 30,000개를 달성했고, 현재 43,000개 이상을 기록합니다. MIT 라이선스이며 Node.js + React 기반으로 self-hosted 구성이 가능합니다. Paperclip 의 핵심 선언은 명확합니다 — "Paperclip doesn't build agents. It takes existing agents and organizes them into a functioning company." Claude Code, OpenClaw, Codex, Python 스크립트, HTTP 웹훅 등 이미 존재하는 에이전트를 '직원'으로 채용하고, 조직도·예산·책임·토큰 소비 추적을 관리하는 역할을 담당합니다. 자체 에이전트 로직이 없으므로 Hermes 와 직접 경쟁하지 않으며, 오히려 Hermes 를 Paperclip 조직도 안에 직원으로 등록하는 조합이 기술적으로 가능합니다.

OpenClaw — 채널 폭 중심 게이트웨이 에이전트

OpenClaw 는 Peter Steinberger 가 개인 프로젝트로 시작해 현재 OpenClaw Foundation 이 유지하는 오픈소스 에이전트입니다 [S04]. 2026년 3월 기준 GitHub 스타 247,000개를 기록해 오픈소스 AI 에이전트 가운데 가장 큰 커뮤니티를 보유합니다. 24개 메시징 채널을 단일 게이트웨이 프로세스에서 지원한다는 점이 핵심 강점입니다 — WhatsApp, Telegram, Slack, Discord, Signal, iMessage, Teams, Matrix, BlueBubbles, Google Chat, WebChat 을 포함합니다. 구조는 허브-앤드-스포크(hub-and-spoke) 방식으로, 중앙 게이트웨이 데몬이 메시징 어댑터를 직접 적재합니다. 에이전트 로직이 게이트웨이 안에 내재된 구조이므로, 게이트웨이가 단일 장애 지점이 될 수 있다는 운영 특성이 있습니다. 상업화 측면에서 openclawai.io SaaS 및 ClawHub 마켓플레이스를 운영합니다.

Harness Engineering — 패러다임 범주 (단일 제품 아님)

Harness Engineering 은 Mitchell Hashimoto 가 2026년 정의한 "Agent = Model + Harness" 공식에서 출발한 설계 패러다임입니다 [S06]. 단일 오픈소스 제품이 아니라 AI 에이전트를 감싸는 제어 시스템 전체를 지칭하는 범주 용어입니다. 이 범주 안에서 대표적인 구현체로는 Microsoft Agent Framework(MAF, 2026년 4월 GA), AWS Harness SDK(Bedrock 기반, MCP 네이티브 지원), HKUDS OpenHarness(v0.1.5, MIT) 세 종이 꼽힙니다. Gartner 는 2026년 말까지 엔터프라이즈 애플리케이션의 40%가 태스크 특화 AI 에이전트를 포함할 것으로 전망합니다. 본 백서에서 '비교 대상으로서의 Harness' 는 이 패러다임 전체를 가리키며, 3.3절에서 Hermes 가 이 패러다임의 세 요소를 어떻게 구현하는지 구체적으로 매핑합니다.

CrewAI — 역할 기반 멀티에이전트 프레임워크

CrewAI 는 GitHub 스타 45,900개 이상을 보유한 오픈소스 멀티에이전트 프레임워크입니다 [S06]. MIT 라이선스. 에이전트를 역할·배경·목표로 정의하고, 그 에이전트들을 '크루(crew)'로 묶

어 태스크를 수행하게 하는 구조입니다. 계층적 프로세스 모드에서 관리자 에이전트가 자동 생성되어 태스크 위임과 결과 검토를 담당합니다. 네이티브 MCP(Model Context Protocol, 모델 컨텍스트 프로토콜) 지원을 포함하며, 평균 응답 지연이 1.8초로 알려져 있습니다. CrewAI 의 강점은 역할 기반 협업 시나리오 — 예를 들어 리서처·작성자·검토자가 각자 역할을 수행하는 파이프라인 — 에서 발휘됩니다.

Hermes Agent — 단일 에이전트 + 자기 개선

Hermes Agent 는 Nous Research 가 2026년 2월 공개한 오픈소스 자율 AI 에이전트입니다 [S01]. MIT 라이선스. GitHub 공식 리포지토리 기준 스타 32,000개 이상, 기여자 245명 이상, 커밋 1,475건(v0.17.0 기준, 2026년 6월 19일)을 기록합니다. 일부 비교 매체는 188,000개 스타를 인용하나, 집계 기준 및 시점 차이로 추정되며 본 백서는 공식 리포지토리 수치인 32,000개를 우선 사용합니다 [S04]. 단일 `curl` 명령으로 의존성을 자동 설치하고, Telegram · Discord · Slack · WhatsApp · Signal · CLI 여섯 채널을 단일 게이트웨이 프로세스에서 지원합니다. Curator 루프를 통한 자기 개선과 지속 기억(persistent memory)이 다른 네 솔루션과 구별되는 핵심 특성입니다 [S02].

항목	Paperclip	OpenClaw	Harness Eng.	CrewAI	Hermes
라이선스	MIT	MIT	범주 (구현별 상이)	MIT	MIT
GitHub 스타	43k+	247k	—	45.9k+	32k+
최초 공개	2026-03	2021~	—	2023~	2026-02
주요 백커	AgencyEnterprise	OpenClaw Foundation	Microsoft · AWS · HKUDS	CrewAI Inc.	Nous Research
핵심 포지션	에이전트 오케스트레이션	채널 게이트웨이	제어 패러다임	역할 멀티에이전트	단일 에이전트 자기개선

3.1.2 5요소 매트릭스 — 라이선스·채널 수·학습 루프·조직화·상업 모델

매트릭스 평가 기준

5요소 매트릭스는 실무 도입 판단에 직결되는 다섯 가지 기준으로 구성됩니다. 라이선스는 기업 법무 부서가 가장 먼저 묻는 항목입니다. 채널 수는 사내 커뮤니케이션 플랫폼 다양성과 직결됩니다. 학습 루프는 운영 경험이 에이전트 성능 개선에 누적되는지를 측정합니다. 조직화 지원은 멀티에이전트 또는 멀티프로파일 시나리오를 다룰 수 있는지를 봅니다. 상업 모델은 장기 운영 비용과 벤더 의존도에 영향을 미칩니다. 각 항목은 1점(미흡)~5점(탁월) 척도로 평가하고, 셀 안에 1줄 근거를 병기합니다.

5요소 × 5종 비교 매트릭스

평가 요소	Paperclip	OpenClaw	MAF (Harness 대표)	CrewAI	Hermes
라이선스	★★★★★ 5 / MIT — 조건 없음	★★★★☆ 4 / MIT — ClawHub 유료 부속	★★★☆☆ 3 / Microsoft 약관 — 상업 이용 사전 검토 필요	★★★★★ 5 / MIT — 조건 없음	★★★★★ 5 / MIT — 조건 없음
채널 수	★★★☆☆ 3 / 에이전트 구성별 상이 — 기본 내장 없음	★★★★★ 5 / 24채널 — 오픈소스 최다	★★★☆☆ 3 / REST·MCP 표준 지원, 채널 직접 내장 없음	★★★☆☆ 3 / MCP 네이티브 지원, 메시징 채널 직접 내장 없음	★★★★☆ 4 / 6채널 네이티브 — Telegram·Discord·Slack·WhatsApp·Signal·CLI
학습 루프	★★☆☆☆ 2 / 에이전트 성과 집계만 — 자기 개선 루프 없음	★★★☆☆ 3 / 사용 이력 로그 — 자동 개선 루프 없음	★★★☆☆ 3 / Evals·센서 패턴 지원 — 구현은 팀 책임	★★★☆☆ 3 / 태스크 피드백 — 루프 자동화 수준 낮음	★★★★★ 5 / Curator 루프 — 실행 결과를 Archive 에 누적하고 Skill 로 자동 증류
조직화	★★★★★ 5 / 조직도·예산·토큰 추적 — 에이전트 조직화 전용	★★★☆☆ 3 / 채널 채팅 워크스페이스 단위	★★★★☆ 4 / 멀티에이전트 오케스트레이션 패턴 내장	★★★★★ 5 / 역할 기반 크루 + 계층 관리자 에이전트	★★★★☆ 4 / Profile 별 독립 메모리·Kanban·멀티 프로파일 병렬 운영
상업 모델	★★★★☆ 4 / 순수 OSS — 지원 계약 없음	★★★☆☆ 3 / SaaS + 마켓플레이스 유료 — 벤더 의존 발생	★★★☆☆ 3 / Microsoft 생태계 종속 — Azure 과금 연계	★★★☆☆ 3 / CrewAI Inc. 구독 플랜 존재	★★★★★ 5 / 순수 OSS — SaaS·구독·마켓플레이스 없음

Hermes 가 학습 루프(5점)와 상업 모델(5점) 두 항목에서 단독 최고점을 기록합니다. 채널 수는 OpenClaw(5점) 에 뒤지지만, 6개 네이티브 채널이 대부분의 기업 커뮤니케이션 플랫폼을 포괄합니다. 조직화 항목에서 Paperclip(5점) 과 CrewAI(5점) 가 앞서지만, 두 솔루션의 조직화는 '여러 에이전트를 조율'하는 방향인 반면 Hermes 의 조직화는 '한 에이전트가 여러 프로파일로 분화'하는 방향으로 접근 방식 자체가 다릅니다 [S01][S05].

3.1.3 5종 가운데 자사에 적합한 OSS 를 고르는 의사결정 트리

의사결정 트리 구조

다섯 솔루션은 서로 다른 조건에서 최적이 됩니다. 아래 트리는 실무에서 가장 빈번하게 등장하는 판단 기준을 순서대로 분기합니다.

```
[출발]
|
├── 외부 SaaS 전면 차단 또는 데이터 외부 송출 금지 환경인가?
|    ├── 예 → 단일 에이전트로 자기 개선이 목표인가?
|    |          ├── 예 → ★ Hermes Agent (MIT, 순수 OSS, Local LLM 연동)
|    |          └── 아니오 → OpenClaw (채널 다양성이 우선) / Hermes (학습 자산 누적이 우선)
|    └── 아니오 ↓
|
├── 다수의 기존 에이전트를 조직도처럼 배치·관리해야 하는가?
|    ├── 예 → ★ Paperclip (조직 메타포 오케스트레이션 — MIT)
|    └── 아니오 ↓
|
├── 역할이 명확한 전문가 에이전트들이 순차·병렬 협업해야 하는가?
|    ├── 예 → ★ CrewAI (역할 기반 크루, MCP 네이티브)
|    └── 아니오 ↓
|
├── 채널 수가 최우선이고 20개 이상의 메시징 플랫폼 통합이 필요한가?
|    ├── 예 → ★ OpenClaw (24채널, 247k stars)
|    └── 아니오 ↓
|
├── AWS Bedrock 또는 Azure 중심 클라우드 전략을 이미 채택했는가?
|    ├── AWS 우선 → AWS Harness SDK (Bedrock 기반, MCP 네이티브)
|    └── Azure 우선 → Microsoft Agent Framework (2026-04 GA, AutoGen + Semantic Kernel)
|
└── 위 조건이 없고 "학습하는 단일 에이전트"가 필요한가?
     └── 예 → ★ Hermes Agent + Local LLM (MIT, 32k stars, Curator 루프)
```

트리의 마지막 분기에 "Hermes Agent + Local LLM" 조합이 놓입니다. 이 시나리오는 클라우드 LLM 의존도를 최소화하면서 에이전트가 누적 실행 경험을 통해 성능을 높여 가는 구성입니다. MSAP.ai 같은 국내 통합 플랫폼을 선택지로 검토 중인 조직이라면, 트리의 각 분기 조건을 내부 요구사항과 대조한 뒤 Hermes 를 포함한 OSS 스택과의 병행 또는 대체 경로를 검토하는 순서가 효율적입니다 [S04][S05][S06].

3.2 Paperclip · OpenClaw 와의 직접 차이점

두 솔루션과의 비교는 단순한 기능 대조를 넘어 설계 철학의 차이를 짚는 작업입니다. Paperclip 은 '이미 만들어진 에이전트들을 어떻게 배치할 것인가'라는 질문에 답하고, OpenClaw 는 '얼마나 많은 채널에서 일관되게 동작할 것인가'라는 질문에 답합니다. Hermes 는 그 두 질문과 구별되는 세 번째 질문 — '단일 에이전트가 시간이 지날수록 더 잘할 수 있는가' — 에 집중합니다.

3.2.1 Paperclip 의 "조직 메타포" 와 Hermes 의 "단일 에이전트 + 자기 개선" 의 트레이드오프

두 철학의 분기점

Paperclip 의 공식 설명은 그 철학을 단 한 문장으로 압축합니다 — "에이전트를 만들지 않는다. 기존 에이전트를 채용해서 조직으로 운영한다"[S05]. 이 선언이 뜻하는 바는 명확합니다. Paperclip 은 Claude Code, OpenClaw, Codex, Python 스크립트, HTTP 웹훅처럼 이미 존재하는 실행 단위를 '직원'으로 등록하고, 조직도·예산·책임·토큰 소비량을 관리하는 메타 레이어입니다. 자체 추론 엔진을 갖지 않으므로 에이전트 오케스트레이션 계층에 위치합니다.

Hermes 는 그 계층보다 한 단계 아래, 개별 에이전트 계층에 자리합니다 [S01][S02]. Curator 루프를 통해 매 실행 결과를 Archive 에 누적하고, 반복 태스크에서 추출한 패턴을 Skill 로 증류합니다. 공식 사이트의 표현을 빌리면 "실행할수록 당신을 더 잘 안다 — 매번 처음부터 맥락을 설명할 필요가 없다"입니다 [S02]. 이 누적 학습은 Paperclip 이 다루지 않는 영역입니다. Paperclip 은 에이전트의 *배치* 를 최적화하지만, 각 에이전트 자체의 *성능 향상* 은 해당 에이전트의 책임입니다.

트레이드오프와 보완 가능성

두 솔루션은 직접 경쟁이 아닙니다. 오케스트레이션 계층(Paperclip) 과 에이전트 계층(Hermes) 은 스택에서 상이한 자리를 차지합니다. 조직 안에 이미 여러 에이전트가 운영 중이고 그 에이전트들 간의 예산 배분과 성과 추적이 과제라면 Paperclip 이 적합합니다. 단일 에이전트가 반복 업무 경험을 누적해 성능을 높이는 것이 목표라면 Hermes 가 적합합니다. 두 솔루션을 동시에 도입할 때 가장 자연스러운 스택은 "Paperclip = 오케스트레이션 계층, Hermes = 에이전트 계층" 입니다 — Paperclip 조직도 안에 Hermes 를 한 직원으로 등록하는 구성이 기술적으로 가능합니다 [S05].

그림 3-1. Paperclip(오케스트레이션) → Hermes(에이전트) 스택 위치

그림: Paperclip(오케스트레이션 계층) → Hermes Agent(에이전트 계층) 스택 위치 다이어그램

위 스택 구조는 두 솔루션이 서로 대체 관계가 아님을 보여 줍니다. 오케스트레이션 계층이 여러 에이전트를 조율할 때, 각 에이전트가 자기 개선 루프를 보유하면 전체 스택의 성능이 시간에 따라 향상됩니다.

도입 판단 기준 정리

자사 에이전트 운영 성숙도에 따라 도입 순서가 달라집니다. 에이전트가 한 개 이하이면 Hermes 단독 도입이 적절합니다. 에이전트가 셋 이상이고 예산과 책임 배분 관리가 필요하다면 Paperclip 을 오케스트레이션 계층으로 추가하는 방안을 검토합니다. 두 솔루션 모두 MIT 라이선스여서 라이선스 검토 부담은 없습니다 [S05][S01].

3.2.2 OpenClaw 의 "게이트웨이 안에 에이전트" vs Hermes 의 "게이트웨이 위에 학습 루프"

두 구조의 기술적 차이

OpenClaw 는 허브-앤드-스포크 구조를 채택합니다. 중앙 게이트웨이 데몬이 24개 메시징 어댑터를 직접 프로세스에 적재하고, 에이전트 로직이 그 게이트웨이 안에 내재됩니다 [S04]. 이 구조는 채널 통합 속도 면에서 유리합니다 — 새 채널 어댑터를 추가하면 즉시 전 채널에 반영됩니다. 반면 게이트웨이 장애나 업데이트가 모든 채널과 에이전트 로직에 동시 영향을 미친다는 단일 장애 지점(SPOF) 리스크가 존재합니다. 실제로 OpenClaw 는 CVE-2026-25253(CVSS 9.1), CVE-2026-25891(8.4), CVE-2026-26102(7.8) 세 건의 공개 취약점과 ClawHavoc, MCP 프록시 공급망 공격 사례가 보고되었습니다 [S04].

Hermes 는 구조가 역전됩니다. CLI 우선 설계에서 출발하여 메시징 게이트웨이를 선택적 확장으로 위치시킵니다. Telegram · Discord · Slack · WhatsApp · Signal · CLI 여섯 채널을 단일 게이트웨이 프로세스가 처리하되, 에이전트 핵심 로직 — Curator 루프, Archive, Skill 시스템 — 은 게이트웨이와 분리된 계층에 있습니다 [S01]. 2026년 5월 기준 공개 CVE 가 없으며, 출시 초기부터 7계층 보안 모델을 적용한 결과입니다.

폭 vs 깊이 트레이드오프

채널 수로 측정한 '폭' 에서 OpenClaw(24개) 가 Hermes(6개) 를 앞섭니다. 그러나 학습 자산 누적으로 측정한 '깊이' 에서 Hermes 가 OpenClaw 를 역전합니다. OpenClaw 는 사용 이력을 로그로 보관하지만, 그 이력이 에이전트 행동 개선에 자동으로 반영되는 루프가 없습니다. Hermes 의 Curator 루프는 매 실행 결과를 Archive 에 저장하고, 반복 태스크에서 추출한 패턴을 Skill 로 자동 증류합니다 [S02].

아래 자가 진단 기준을 참고하면 선택 방향이 명확해집니다. 사내 메시징 플랫폼이 10개 이상이고 그 모두를 단일 에이전트로 포괄해야 하는 경우에는 OpenClaw 의 채널 폭이 결정적 요소입니다. 반면 메시징 채널 수는 5~6개 이하이고 에이전트가 반복 업무에서 성능을 축적하길 기대하는 경우에는 Hermes 의 학습 깊이가 더 큰 운영 가치를 제공합니다. 운영 안정성을 우선하는 보안 요구 환경에서도 Hermes 가 유리합니다 [S04].

운영 안정성 수치 비교

OpenClaw 는 잦은 대형 아키텍처 변경 업데이트로 인해 실행 중인 인스턴스 손상 사례가 보고됩니다. Hermes 는 출시 후 50일 동안 여섯 차례의 번호 부여 릴리스를 통해 빠른 반복 개발을

유지하면서도 실행 인스턴스 안정성을 보전했습니다 [S04]. 안정성과 빠른 개선 속도를 동시에 유지하는 방식은 CLI 우선 구조에서 게이트웨이 어댑터를 분리한 설계 결정에 기인합니다.

3.2.3 Hermes 가 메우는 공백 — "왜 또 다른 에이전트인가" 의 답

공백의 정의

오픈소스 AI 에이전트 시장에 이미 여러 솔루션이 존재한다면 Hermes 가 추가될 이유는 무엇입니까? 이 질문에 답하려면 각 솔루션이 충족하는 요소와 충족하지 못하는 요소를 함께 봐야 합니다. 아래 매트릭스는 앞선 비교에서 도출된 다섯 가지 핵심 요건을 기준으로 각 솔루션의 충족 여부를 정리합니다.

요건	Paperclip	OpenClaw	MAF	CrewAI	Hermes
MIT 라이선스 (조건 없음)	✓	✓	✗	✓	✓
지속 기억 (Persistent Memory)	✗	✗	구현 필요	✗	✓
자기 개선 루프 (Curator)	✗	✗	구현 필요	✗	✓
단일 명령 설치 (curl one-liner)	✗	✗	✗	✗	✓
네이티브 6채널 내장	✗	✓ (24채널)	✗	✗	✓

Hermes 는 다섯 요건을 동시에 충족하는 유일한 솔루션입니다. OpenClaw 는 채널 내장 항목에서 겹치지만, MIT 조건 없음·지속 기억·자기 개선 루프·단일 명령 설치 네 항목 가운데 자기 개선 루프 항목에서 동등하지 않습니다 [S01][S04].

Hermes 의 위치

다섯 요건이 결합된 위치는 한 문장으로 응축됩니다 — "MIT 라이선스, 단일 `curl` 설치, 6채널 네이티브, 지속 기억, Curator 자기 개선 루프를 동시에 충족하는 오픈소스 자율 에이전트." 개별 요건을 충족하는 솔루션은 다수 존재하지만, 다섯 요건을 조건 없이 결합하는 솔루션은 현재 Hermes 가 유일합니다. 이것이 '왜 또 다른 에이전트인가'에 대한 실증 근거입니다 [S01][S02][S04].

3.3 Harness Engineering 패러다임과의 관계

'Harness' 라는 용어는 AI 에이전트 담론에서 혼선을 빚기 쉽습니다. 특정 제품명으로 오해하거나, 단순 프롬프트 래퍼와 동의어로 이해하는 경우가 있습니다. 이 절은 Harness Engineering 의

개념적 위치를 정확히 박제하고, Hermes 가 이 패러다임의 세 요소를 어떻게 구현하는지 기능 단위로 매핑합니다.

3.3.1 Anthropic 의 "Agent = Model + Harness" 정의 의미

Harness 의 정의

Harness Engineering 은 Mitchell Hashimoto 가 2026년에 정립한 개념으로, "AI 에이전트의 행동을 지배하는 제어 시스템을 설계하고 유지하는 분야"를 가리킵니다 [S06]. Anthropic 의 "Agent = Model + Harness" 공식은 이 패러다임의 핵심 선언입니다. 여기서 Model 은 GPT-4o, Claude, Gemini, Qwen3 같은 추론 엔진 — 상호교환 가능한 구성 요소입니다. Harness 는 그 모델을 감싸는 모든 제어 구조 — 비즈니스 규칙, 안전 제약, 검증 로직, 맥락 공급 파이프라인 — 를 가리킵니다. 이 공식이 말하는 것은 간단합니다 — 모델 자체는 빠르게 상품화되고 있으므로, 경쟁 우위는 Harness 설계 능력에서 나온다는 뜻입니다.

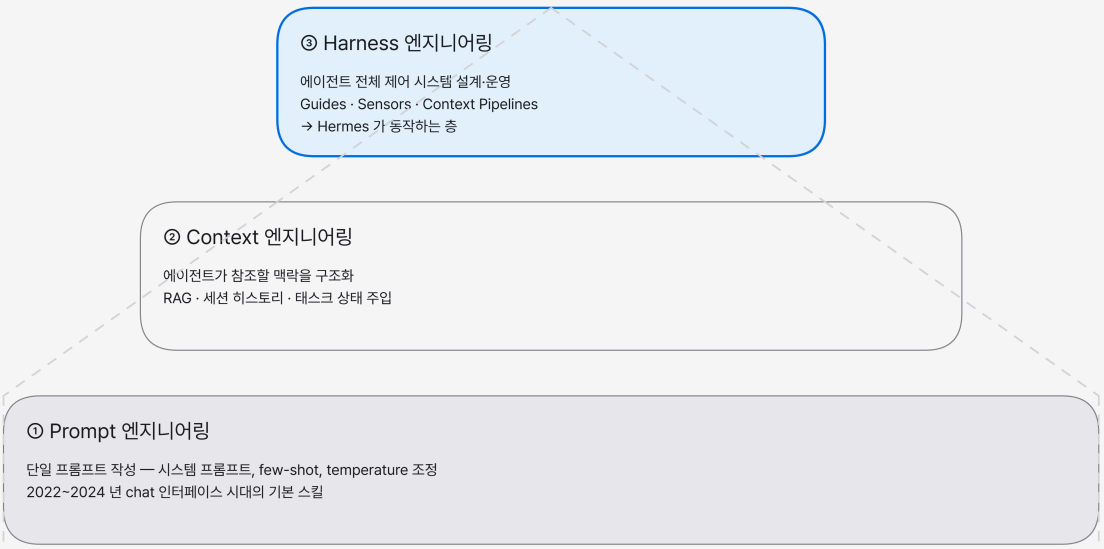
3요소 구조

Harness 는 세 가지 구성 요소로 분해됩니다. 첫째 Guides — 에이전트가 무엇을 할 수 있고 알 수 있는지를 사전에 정의하는 제어 장치입니다. 시스템 프롬프트, AGENTS.md 파일, 접근 권한 문서가 여기에 속합니다. 둘째 Sensors — 에이전트의 실제 행동을 관찰하고 검증하는 사후 피드백 장치입니다. 평가(Evals), 유효성 검사 루프, 출력 파서가 이 역할을 담당합니다. 셋째 Context Pipelines — 에이전트가 신뢰할 수 있는 최신 맥락 데이터를 공급하는 데이터 계층입니다. 에이전트가 판단에 사용하는 정보의 품질이 이 파이프라인의 품질에 직결됩니다 [S06].

패러다임 vs 제품

중요한 구분이 있습니다. Harness Engineering 은 단일 오픈소스 제품이 아닙니다. 특정 GitHub 리포지토리를 가리키지 않으며, 특정 벤더의 상업 제품도 아닙니다. 이 범주 안에서 여러 구현체가 경쟁합니다. Hermes 는 이 패러다임의 한 구현체입니다 — "Harness = 범주, Hermes = 그 범주 안에서 MIT 조건 없이 자기 개선까지 구현한 사례"라는 위계로 이해하면 비교의 적절성이 성립합니다 [S06].

그림 3-2. Prompt → Context → Harness 엔지니어링 3층 피라미드



출처: Mitchell Hashimoto (2026) [S06] — Agent = Model + Harness. 상위 계층일수록 경쟁 우위 폭이 큼

그림: Prompt 엔지니어링 → Context 엔지니어링 → Harness 엔지니어링 3층 피라미드 다이어그램

피라미드의 가장 아래층은 단순 프롬프트 작성입니다. 중간층은 에이전트가 참조할 맥락을 구조화하는 Context 엔지니어링입니다. 최상층이 Harness 엔지니어링 — 에이전트 전체 제어 시스템을 설계·운영하는 영역입니다. Hermes 는 이 최상층에서 동작합니다.

3.3.2 Microsoft Agent Framework · AWS Harness SDK · OpenHarness 대표 사례

세 구현체의 정체

Harness Engineering 범주 안에서 현재 가장 주목받는 구현체는 세 종입니다. 각 구현체의 특성은 자사 클라우드 전략 및 기술 스택과 맞춰 선택 기준을 정해야 합니다.

구현체	라이선스	클라우드 기반	MCP 지원	GA 일자	비고
Microsoft Agent Framework (MAF)	Microsoft 약관	Azure 우선	✓	2026-04-02	AutoGen + Semantic Kernel 통합
AWS Harness SDK	Apache 2.0	AWS Bedrock 기반	✓ (네이티브)	2026 상반기	Anthropic · OpenAI · Gemini · Ollama 지원
OpenHarness (HKUDS)	MIT	클라우드 중립	✓ (HTTP transport)	v0.1.5	Swarm polling · 가장 가벼운 구현

Microsoft Agent Framework 는 AutoGen 과 Semantic Kernel 을 통합한 마이크로소프트의 공식 에이전트 프레임워크로 2026년 4월 GA 버전을 출시했습니다 [S06]. Azure 생태계와 긴밀하게 연동되어 Azure OpenAI, Microsoft Copilot 과 통합이 용이하지만, Microsoft 약관의 적용을 받으므로 도입 전 법무 검토가 필요합니다. AWS Harness SDK 는 Bedrock 을 기반으로 Anthropic, OpenAI, Gemini, Ollama 등 다양한 LLM 을 지원하며 내장 옵저버빌리티(observability, 관찰 가능성) 기능을 포함합니다. AWS 인프라를 주요 클라우드로 사용하는 조직에서 가장 자연스러운 진입 경로입니다. OpenHarness 는 HKUDS 가 공개한 MIT 라이선스 구현체로, MCP HTTP 트랜스포트와 Swarm 폴링을 지원하는 가장 가벼운 구현입니다.

대형 벤더 진입의 의미

Microsoft 와 AWS 가 같은 시기에 Harness Engineering 범주에 구현체를 출시했다는 사실은 카테고리 표준화가 진행 중임을 나타냅니다. Gartner 는 2026년 말까지 엔터프라이즈 애플리케이션의 40% 가 태스크 특화 AI 에이전트를 포함할 것으로 전망합니다 [S06]. 이 흐름에서 Harness 설계 역량은 옵션이 아닌 운영 필수 요소로 자리 잡을 것입니다. 자사 클라우드 전략이 Azure 우선이면 MAF, AWS 우선이면 AWS Harness SDK, 멀티 클라우드 또는 온프레미스 우선이면 OpenHarness 또는 Hermes 가 적합한 출발점입니다.

3.3.3 Hermes 가 Harness 의 3요소 (guides · sensors · pipelines) 를 구현하는 기능 매핑

매핑 논리

Harness Engineering 의 세 요소(Guides · Sensors · Context Pipelines) 를 Hermes 의 기능 블록에 매핑하면 Hermes 의 추상 카테고리 위치가 확정됩니다. Hermes 는 Harness Engineering 패러다임의 세 요소를 코드 레벨에서 구현한 에이전트입니다. 외부에서 패러다임을 주입하는 것이 아니라, Hermes 의 핵심 기능 블록 자체가 세 요소에 대응합니다 [S01][S03][S06].

1:1 기능 매핑 표

Harness 요소	역할	Hermes 기능 블록	구체적 동작
Guides	에이전트 행동 사전 제어	Skill 시스템 + Profile 구성	Profile 별 허용 도구·Skill·메모리 범위를 사전 정의. 에이전트가 접근 가능한 자원의 경계를 결정
Sensors	실행 결과 관찰·검증	Curator 루프	매 실행 후 결과를 평가하고 Archive 에 저장. 품질 미달 시 재시도 트리거. 실행 상태를 Kanban 카드로 추적
Context Pipelines	신뢰할 수 있는 최신 맥락 공급	Archive + Kanban	Archive 가 누적 기억을 제공하고, Kanban 이 현재 태스크 상태와 의존성 맥락을 공급.

Harness 요소	역할	Hermes 기능 블록	구체적 동작
			SQLite 기반 durable 저장

매핑 이후의 시사점

이 매핑이 확정하는 것은 하나입니다 — Hermes 는 Harness Engineering 패러다임의 구현체로서, 세 요소를 별도의 외부 프레임워크 없이 단일 에이전트 내부에 통합합니다. Microsoft Agent Framework 나 AWS Harness SDK 와 비교했을 때, Hermes 는 클라우드 생태계 종속 없이 MIT 라이선스 단일 `curl` 설치로 동일한 세 요소를 갖춥니다 [S01][S06]. 이 사실은 온프레미스 보안 환경, 망분리 환경, 벤더 락인(vendor lock-in, 특정 벤더 종속) 회피가 요구되는 환경에서 Hermes 선택의 기술적 근거가 됩니다.

Profile 과 Skill 구성이 Guides 를 담당하고, Curator 루프가 Sensors 를 담당하며, Archive 와 Kanban 이 Context Pipelines 를 담당합니다. 에이전트 한 개가 Harness Engineering 의 세 요소를 동시에 실행하는 구조입니다 [S01][S03]. 이 구조는 Harness 설계 역량을 별도 팀이 구축·유지해야 하는 외부 프레임워크 방식과 달리, 에이전트 자체에 내재되어 있으므로 도입 초기 설계 비용을 줄입니다.

4장. Hermes Agent 핵심 개념 — Profile · Kanban · Skill · Archive

Hermes Agent 가 사내 업무에 자연스럽게 녹아들 수 있는 이유는 설계 자체가 현업 조직 구조를 그대로 반영하기 때문입니다. 작업 수행 주체를 Profile 로 분리하고, 진행 상태를 Kanban 보드 한 장에 가시화하며, 반복되는 처리 절차를 Skill 파일로 패키징하고, 모든 대화와 산출물을 Archive 에 누적합니다. 이 네 가지 추상화는 개별 기능이 아니라 서로 맞물린 하나의 운영 체계입니다. 4장에서는 각 추상화의 기술적 정의와 실제 운영 가치를 순서대로 살펴봅니다.

4.1 Profile — 목적별 에이전트 인스턴스 분리

Profile 은 동일한 Hermes Agent 프로세스 위에서 목적이 다른 에이전트 인스턴스를 상태 충돌 없이 병렬로 운영하기 위한 격리 단위입니다. 각 Profile 은 고유한 메모리 공간, 별도의 Skill 묶음, 독립된 Tool 접근 권한을 갖습니다. 한 호스트에서 코딩 보조 Profile 과 연구 조사 Profile 이 동시에 실행되더라도 두 인스턴스의 대화 맥락·학습 이력·Tool 권한은 완전히 분리됩니다. 조직 내 RBAC(Role-Based Access Control, 역할 기반 접근 통제) 정책을 AI 에이전트 계층에 그대로 적용하는 가장 직접적인 수단이 바로 Profile 입니다 [S03].

4.1.1 Profile 의 기술적 정의 — 메모리·Skill·Tool 묶음

Profile 컨테이너 구조

Hermes Agent 공식 문서는 Profile 을 "목적별 Hermes 인스턴스" 로 정의합니다 [S03]. 구조적으로 Profile 은 네 개의 전용 자원 슬롯을 가집니다. 첫째, 메모리 슬롯은 해당 Profile 이 수행한 대화 이력·자동 생성된 Skill·도구 사용 패턴을 격리된 공간에 보관합니다. 둘째, Skill 슬롯은 그

Profile 이 호출할 수 있는 `SKILL.md` 파일 목록을 관리합니다. 셋째, Tool 슬롯은 파일 시스템 접근 경로·외부 API 인증 정보·MCP(Model Context Protocol, 모델 컨텍스트 프로토콜) 서버 목록을 Profile 단위로 선언합니다. 넷째, Channel 슬롯은 Telegram·Slack·CLI 등 어느 채널로 해당 Profile 에 접근할 수 있는지를 지정합니다.

Profile 단위 격리의 운영 의미

이 구조가 현업에 주는 함의는 명확합니다. 영업팀 Profile 이 접근하는 파일 경로와 인사팀 Profile 이 접근하는 파일 경로를 선언 수준에서 분리할 수 있고, 총무팀 Profile 이 사용하는 외부 API 키는 다른 Profile 에서 참조 자체가 불가능합니다. Tool 권한이 코드나 IAM(Identity and Access Management, 신원·접근 관리) 정책으로 강제되는 것이 아니라 Profile 선언 파일에서 직접 통제되기 때문에 운영 부담이 줄어듭니다. 감사 담당자 입장에서는 Profile 단위로 로그를 분리·조회할 수 있으므로 특정 에이전트 인스턴스의 행동 이력을 사후에 재구성하는 작업이 단순해집니다.

비용 추적 가능성

Profile 격리는 비용 관리와도 직결됩니다. LiteLLM(라이트LLM) — 100개 이상의 LLM(Large Language Model, 대형 언어 모델) 공급자를 단일 엔드포인트로 연결하는 오픈소스 게이트웨이 — 과 Hermes Agent 를 함께 구성하면 Profile 별 토큰 소비량과 추정 비용을 독립 집계할 수 있습니다 [S10]. 월말 정산 시 "어느 Profile 이 얼마를 소비했는가"를 부서 단위로 집계하는 리포트 생성이 가능합니다. 이는 AI 에이전트 비용을 개인 비용이 아닌 팀·프로젝트 단위 운영 비용으로 관리하려는 조직에게 실질적인 통제 수단이 됩니다.

그림 4-1. Profile 컨테이너 — 4 슬롯(메모리·Skill·Tool·Channel)이 격리된 병렬 구조

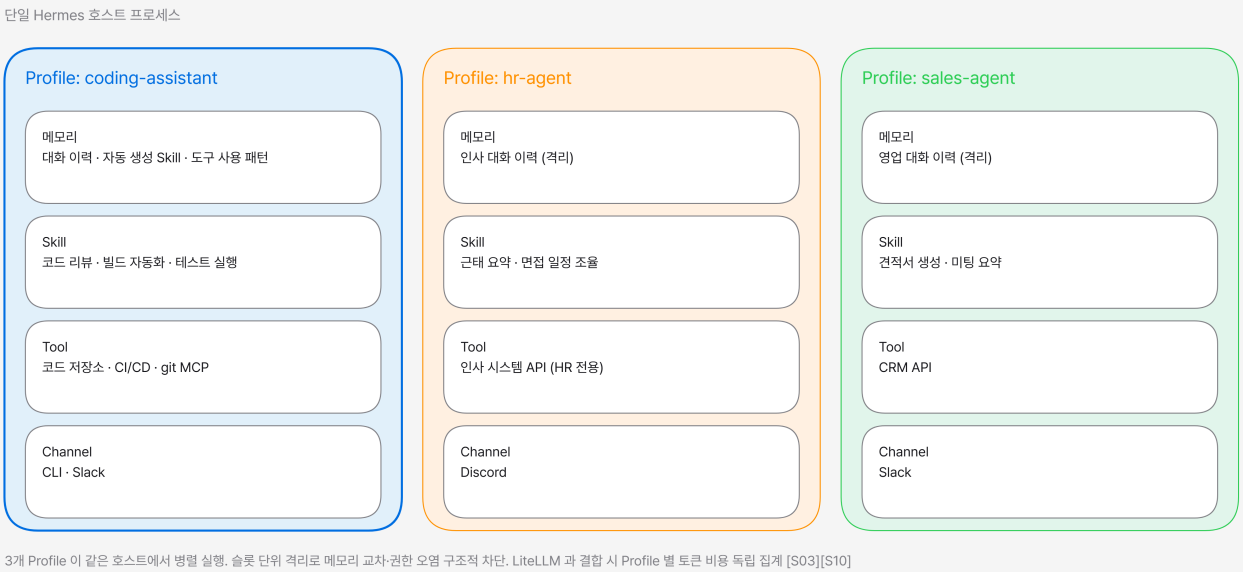


그림 4-1: Profile 컨테이너 — 메모리·Skill·Tool·Channel 네 슬롯이 Profile 경계 안에 격리되어 있고, 한 호스트 위에 여러 Profile 이 병렬로 존재하는 구조.

Profile 컨테이너 구조를 시각화하면 왜 Profile 이 단순한 "사용자 계정"과 다른지 분명해집니다. 계정은 인증·권한 부여에 집중하지만, Profile 은 메모리 누적과 Skill 진화까지 격리 범위에 포함

합니다. 시간이 지날수록 각 Profile 은 자신이 담당하는 업무 영역에 특화된 학습 자산을 독자적으로 쌓아 가며, 다른 Profile 의 학습 결과에 영향을 받거나 오염되지 않습니다.

4.1.2 coding · personal · research Profile 3 사례

공식 문서가 제시하는 3 Profile 유형

Hermes Agent 공식 문서는 세 가지 대표 Profile 유형을 제시합니다 [S03]. 코딩 보조(coding assistant) Profile 은 코드 저장소 접근 Tool, 빌드·테스트 실행 스크립트, 코드 리뷰 Skill 을 묶어 구성하며 엔지니어링 팀의 반복 작업을 처리합니다. 개인 보조(personal bot) Profile 은 개인 일정·이메일·메모 도구에 접근 권한을 가지며, 특정 사용자 한 명의 업무 스타일과 선호를 장기 기억으로 축적합니다. 연구 조사(research agent) Profile 은 웹 검색 Tool, 문서 요약 Skill, 인용 관리 스크립트를 갖추고 리서치 산출물을 Archive 에 체계적으로 정리합니다.

사내 도입 시 변형 Profile 설계

세 가지 공식 유형은 사내 부서 구조에 맞게 곧바로 변형할 수 있습니다. 영업팀 Profile 은 CRM(Customer Relationship Management, 고객 관계 관리) API Tool 과 견적서 생성 Skill 을 조합하고, HR Profile 은 인사 시스템 조회 Tool 과 근태 요약 Skill 을 포함합니다. 총무 Profile 은 시설 예약 시스템과 비용 정산 스크립트를, R&D Profile 은 내부 논문 저장소와 기술 문서 생성 Skill 을, CS(Customer Service, 고객 서비스) Profile 은 티켓 시스템 API 와 답변 초안 작성 Skill 을 각각 패키징합니다. 아래 표는 공식 3종과 사내 변형 5종의 구성 요소를 정리한 것입니다.

Profile 유형	주요 Tool	핵심 Skill	주 채널
coding assistant (공식)	코드 저장소, CI/CD	코드 리뷰, 빌드 자동화	CLI, Slack
personal bot (공식)	일정, 이메일	일정 요약, 초안 작성	Telegram
research agent (공식)	웹 검색, 파일 시스템	요약, 인용 정리	CLI
영업팀 (사내 변형)	CRM API	견적서 생성, 미팅 요약	Slack
HR (사내 변형)	인사 시스템	근태 요약, 면접 일정 조율	Discord
총무 (사내 변형)	시설 예약, 정산 시스템	비용 정산 초안	CLI
R&D (사내 변형)	내부 논문 저장소	기술 문서 생성, 특허 초안	CLI
CS (사내 변형)	티켓 시스템	답변 초안, 에스컬레이션 요약	WhatsApp, Slack

부서별 Profile 설계가 도입의 첫 산출물

주목할 점은 Profile 설계가 복잡한 인프라 구성이 아니라는 점입니다. Profile 은 YAML 선언 파일 한 장으로 정의되며, Tool 목록과 Skill 경로를 텍스트로 나열하는 수준의 작업입니다. 따라서 사내 도입 워크숍에서 반나절 세션 안에 각 부서 담당자가 자신의 Profile 초안을 직접 작성하는

방식으로 도입을 시작할 수 있습니다. 이 산출물이 곧바로 PoC(Proof of Concept, 개념 검증) 배포의 출발점이 됩니다.

4.1.3 Profile 단위 비용·감사·권한 격리의 운영 가치

세 부서를 동시에 만족하는 격리 구조

Profile 격리는 단일 메커니즘으로 세 가지 조직 요구를 동시에 충족합니다. 정보보호팀은 Tool 접근 권한이 Profile 단위로 선언되어 있으므로 특정 인스턴스가 어느 시스템에 접근할 수 있는지 단번에 확인할 수 있습니다. 재무팀은 LiteLLM 과 연계된 Profile 별 토큰 소비 리포트로 AI 에이전트 운영 비용을 부서 예산과 연결할 수 있습니다 [S10]. 감사팀은 Profile 단위로 분리된 대화 로그와 산출물 이력을 Archive 에서 조회하여 특정 결정이 어떤 에이전트 인스턴스의 어떤 맥락에서 이루어졌는지 재구성할 수 있습니다.

월간 비용 리포트 구조

실제 운영 시 Profile 단위 월간 비용 리포트는 아래와 같은 구조로 집계됩니다.

Profile 명	소속 부서	월간 토큰 (입력)	월간 토큰 (출력)	추정 비용 (USD)	모델
sales-agent	영업팀	4,200,000	1,800,000	$18.0	gpt-oss-20B (로컬)
hr-agent	HR팀	1,500,000	600,000	$6.3	Qwen3-7B (로컬)
rd-agent	R&D팀	8,100,000	3,200,000	$34.2	Gemma 4 31B Dense (로컬)
cs-agent	CS팀	2,800,000	900,000	$11.5	gpt-oss-20B (로컬)

권한 모델 구현과 감사 준비

Profile 단위 Tool 권한 선언은 기존 IAM 정책을 AI 에이전트 계층까지 자연스럽게 확장하는 방법을 제공합니다 [S03]. 각 Profile 의 Tool 목록은 버전 관리 시스템에 커밋할 수 있는 선언 파일이기 때문에 "어느 시점에 어느 Profile 이 어떤 Tool 에 접근 권한을 가졌는가"를 이력으로 추적할 수 있습니다. 이는 ISO 27001 이나 SOC 2(Service Organization Control 2) 감사에서 요구하는 접근 제어 증적과 직결됩니다. 별도의 에이전트 전용 감사 시스템을 구축할 필요 없이 기존 버전 관리·로그 수집 체계에 Profile 선언 파일과 Archive 로그를 포함하는 것만으로 감사 요건을 충족할 수 있습니다.

4.2 Kanban — 다중 에이전트 작업 보드

Kanban 은 여러 Profile 에 걸쳐 진행되는 작업을 단일 보드에서 추적하는 Hermes Agent 의 작업 관리 추상화입니다. SQLite(에스큐라이트) — 서버 없이 단일 파일로 동작하는 내장형 관계형 데이터베이스 — 를 영속 저장소로 사용하므로 외부 데이터베이스 인프라 없이도 운영이 가능

합니다 [S03]. 에이전트가 수행하는 모든 작업은 Kanban 카드 한 장에 매핑되며, 사람과 에이전트 모두 동일한 보드를 통해 작업 상태를 확인하고 개입할 수 있습니다. MSAP.ai 의 통합 작업 보드와 유사한 가시성 모델을 사내 AI 에이전트 운영에 그대로 구현한 형태입니다.

4.2.1 SQLite 기반 Kanban 컬럼·카드 데이터 모델

7 컬럼 구조

Hermes Agent 의 Kanban 보드는 7개 컬럼으로 구성됩니다 [S03]. triage 는 신규 요청이 검토 대기 중인 상태, todo 는 실행 대상으로 확정된 상태, ready 는 선행 조건을 모두 충족하고 즉시 실행 가능한 상태, running 은 에이전트가 현재 처리 중인 상태를 나타냅니다. blocked 는 외부 의존성·승인 대기로 진행이 중단된 상태, done 은 완료 확인이 끝난 상태, archived 는 이력 보존 대상으로 보드에서 제거된 상태입니다. 이 7 단계는 소프트웨어 개발팀에서 흔히 쓰는 컬럼 구조와 거의 동일하여 사내 도입 시 재교육 없이 즉시 이해할 수 있습니다.

카드 5 속성

각 카드는 5개의 핵심 속성을 가집니다. assignee 는 이 카드를 처리할 Profile 이름입니다. dependency links 는 이 카드가 완료되어야 다른 카드가 ready 상태로 전환될 수 있음을 선언하는 연결 고리입니다. workspace kind 는 에이전트가 작업을 수행할 파일 시스템 맥락을 지정하며 scratch(임시 공간), worktree(Git 작업 트리), `dir:경로`(지정 경로) 세 가지 중 하나를 선택합니다. tenant namespace 는 동일한 Kanban 보드 안에서 부서·프로젝트 단위로 카드를 격리하는 논리적 구분자입니다. 마지막으로 comment thread 는 사람과 에이전트가 해당 카드에 남기는 비동기 소통 기록입니다.

아래 표는 7 컬럼과 카드 5 속성을 정리한 데이터 모델입니다.

구분	항목	설명
컬럼	triage	신규 요청 검토 대기
컬럼	todo	실행 확정
컬럼	ready	선행 조건 완료, 즉시 실행 가능
컬럼	running	에이전트 처리 중
컬럼	blocked	외부 의존성·승인 대기 중
컬럼	done	완료 확인
컬럼	archived	이력 보존, 보드 제거
카드 속성	assignee	처리 담당 Profile 명
카드 속성	dependency links	선행 카드 연결
카드 속성	workspace kind	파일 시스템 작업 맥락 (scratch / worktree / dir:path)
카드 속성	tenant namespace	부서·프로젝트 격리 단위

구분	항목	설명
카드 속성	comment thread	비동기 소통 기록

SQLite 단일 파일의 운영 장점

SQLite 를 저장소로 선택한 것은 운영 단순성을 위한 설계 결정입니다. Kanban 보드 전체 상태가 단일 `.db` 파일에 담기기 때문에 백업은 파일 복사, 복구는 파일 교체, 마이그레이션은 파일 이동으로 완료됩니다. PostgreSQL 이나 MySQL 같은 독립 데이터베이스 서버를 관리할 필요가 없습니다. 소규모 팀이 PoC 를 시작할 때 인프라 부담이 최소화된다는 점에서 도입 초기 장벽을 낮추는 효과가 있습니다.

4.2.2 Multi-profile 협업 — assignee · dependency · tenant

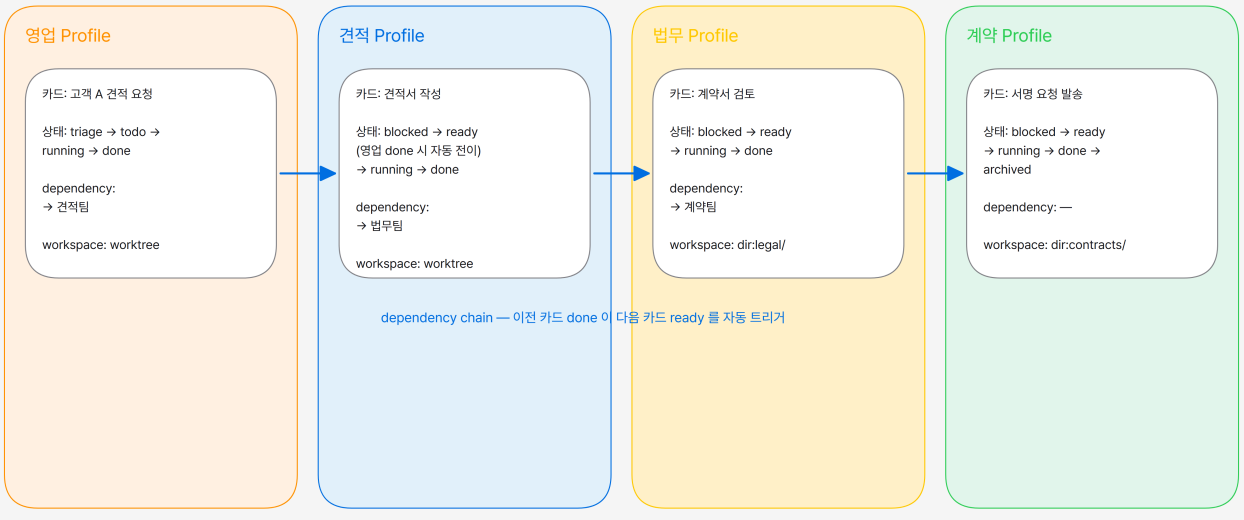
부서 간 인계를 Kanban 카드로 가시화

Hermes Agent Kanban 의 핵심 가치는 여러 Profile 이 참여하는 업무 흐름을 단일 보드 위에서 추적할 수 있다는 점입니다 [S03]. 영업팀 Profile 이 계약 초안을 작성한 뒤 법무팀 Profile 에게 검토를 요청하는 시나리오를 생각해 보면, 영업팀 카드에 `dependency: 법무-검토` 를 선언하는 것만으로 두 Profile 간 인계 관계가 Kanban 보드에 자동으로 시각화됩니다. 법무팀 카드가 done 상태로 전환되면 영업팀의 다음 카드가 자동으로 ready 상태가 되어 에이전트가 후속 작업을 시작할 수 있습니다.

4 부서 인계 시나리오

영업 → 견적팀 → 법무 → 계약 4단계 인계 흐름을 Kanban 으로 표현하면 아래와 같습니다.

그림 4-2. 4부서 인계 흐름 — 영업 → 견적 → 법무 → 계약 Kanban dependency chain



SQLite 단일 DB 파일. tenant namespace 로 프로젝트 격리. Dispatcher 60초 폴링으로 ready 카드 원자적 청구 [S03]

그림 4-2: 4 부서 인계 흐름 — 영업 Profile 의 고객 요청 카드가 triage 에서 시작하여 견적 Profile, 법무 Profile, 계약 Profile 을 순서대로 거치는 dependency chain 구조. 각 Profile 전환 시 이전 카드가 done 이 되면 다음 카드가 ready 로 자동 전환된다.

영업 Profile 이 생성한 "고객 A 견적 요청" 카드는 triage → todo → running → done 경로를 밟습니다. 이 카드가 done 으로 바뀌는 순간, dependency 로 연결된 견적팀 Profile 의 "견적서 작성" 카드가 ready 상태로 전환됩니다. 견적팀 에이전트는 대기 없이 즉시 작업을 시작하고, 산출물(견적서 파일)을 worktree 로 지정한 workspace 에 저장합니다. 이후 법무팀 Profile 의 "계약서 검토" 카드, 계약팀 Profile 의 "서명 요청" 카드가 동일한 방식으로 순서대로 활성화됩니다.

tenant namespace 로 프로젝트 격리

여러 프로젝트가 동시에 진행될 때 tenant namespace 가 중요해집니다. `tenant: project-alpha` 로 태깅된 카드들은 `tenant: project-beta` 카드들과 보드 안에서 논리적으로 분리됩니다. 조회·필터·감사 시 tenant 단위로 작업 이력을 추출할 수 있기 때문에 프로젝트 종료 후 해당 프로젝트의 전체 작업 로그를 아카이빙하거나 감사 자료로 제출하는 작업이 간단해집니다.

4.2.3 Human-in-the-loop — 코멘트 thread 와 컬럼 드래그

사람이 개입하는 세 가지 방식

Hermes Agent Kanban 은 에이전트 자율 실행과 인간 개입이 공존하도록 설계되었습니다 [S03]. 첫째, triage 컬럼에서 담당자가 신규 카드를 검토하고 올바른 Profile 에 할당하거나 우선순위를 조정하는 방식으로 작업 흐름 진입 시점을 제어합니다. 둘째, 진행 중인 카드의 comment thread 에 담당자가 추가 지시나 수정 요청을 남기면 에이전트가 해당 코멘트를 맥락으로 인식하고 행동을 수정합니다. 셋째, 시각적 보드에서 카드를 컬럼 간 드래그 앤 드롭으로 이동시켜 에이전트 상태를 직접 제어할 수 있습니다. 예를 들어 blocked 컬럼에 머물고 있는 카드를 ready 로 이동하면 에이전트가 해당 카드 처리를 즉시 재개합니다.

HITL 은 신뢰 구축의 출발점

HITL(Human-in-the-loop, 인간 참여 루프) — 에이전트 실행 흐름에 사람이 검토·승인·수정으로 참여하는 구조 — 는 자율 에이전트 도입 초기에 특히 중요합니다. 에이전트의 판단을 무조건 신뢰하기 어려운 PoC 단계에서 HITL 비중을 높게 유지하다가, 에이전트 행동 패턴에 대한 신뢰가 쌓일수록 점진적으로 개입 빈도를 낮추는 것이 현실적인 운영 전략입니다.

단계	에이전트 자율 처리 비율	HITL 개입 비율	주요 개입 방식
PoC (1~3개월)	50%	50%	모든 카드 triage 수동 확인, done 승인
파일럿 (4~6개월)	70%	30%	고위험 카드만 수동 승인
본격 운영 (7개월~)	90%	10%	예외·에스컬레이션 카드만 개입

cross-profile 감독의 실용적 가치

Kanban 보드는 cross-profile supervision — 여러 Profile 에 걸쳐 있는 작업 전체를 한 화면에서 감독하는 기능 — 도 제공합니다 [S03]. 팀 리더나 운영 담당자가 특정 tenant 의 전체 카드 현황

을 한 번에 확인하고, 병목이 발생한 Profile 을 즉시 식별하며, 필요한 경우 카드 재배정이나 우선순위 조정을 UI 에서 직접 수행할 수 있습니다. 에이전트 동작 전체를 코드나 로그를 열지 않고도 파악할 수 있다는 점이 비기술 직군 관리자에게 특히 유용합니다.

4.3 Skill 과 Archive — 학습 자산의 축적

Skill 과 Archive 는 Hermes Agent 가 시간이 지날수록 더 나은 성능을 발휘하는 원리를 구성하는 두 핵심 요소입니다. Skill 은 반복되는 처리 절차를 재사용 가능한 파일로 패키징한 능력 단위이고, Archive 는 에이전트가 수행한 모든 대화·산출물·결정 근거를 영속적으로 보존하는 기록 계층입니다. Hermes Agent 공식 사이트가 선언한 "오래 실행될수록 당신을 더 잘 안다"는 핵심 가치는 Skill 자동 생성과 Archive 누적이 함께 작동하는 Curator loop 로 구현됩니다 [S02].

4.3.1 Skill 파일 구조 — SKILL.md + frontmatter + 참조 스크립트

SKILL.md 의 세 구성 요소

Hermes Agent 에서 Skill 은 `SKILL.md` 라는 마크다운 파일 하나로 정의됩니다 [S03]. 파일은 세 부분으로 구성됩니다. 첫째, YAML frontmatter 에는 Skill 이름, 발동 조건(trigger), 적용 가능한 Profile 목록, 참조할 외부 스크립트 경로를 선언합니다. 둘째, 본문(body)에는 에이전트가 이 Skill 을 실행할 때 따를 절차와 주의사항을 자연어로 작성합니다. 셋째, 참조 스크립트 섹션에는 Skill 실행 중 호출할 셸 스크립트나 Python 스크립트의 경로와 인수를 명시합니다.

```
 ---
name: "weekly-sales-report"
trigger: "매주 금요일 오후 5시 | 사용자 요청: 주간 영업 보고서"
profiles:
  - sales-agent
scripts:
  - path: "scripts/fetch-crm-data.py"
    args: ["--period", "7d"]
  - path: "scripts/generate-report.sh"
---

## 실행 절차

1. CRM 에서 지난 7일간 영업 데이터를 수집합니다 (fetch-crm-data.py 실행).
2. 수집된 데이터를 기반으로 주간 보고서 초안을 작성합니다.
3. 작성된 초안을 Slack #영업-보고 채널에 게시하고 확인을 요청합니다.

## 주의 사항

- 고객사 매출 데이터는 외부 시스템에 전송하지 않습니다.
- 보고서 초안이 완료되면 반드시 담당자 확인 후 최종 발송합니다.
```

현업 부서가 직접 작성 가능한 수준

Skill 파일의 설계 원칙은 개발자가 아닌 현업 담당자도 작성할 수 있는 단순성입니다. frontmatter 의 trigger 필드는 자연어 조건("매주 금요일 오후 5시") 또는 사용자 입력 패턴을 그대로 기록하고, 본문은 표준 마크다운으로 절차를 설명하면 됩니다. 스크립트를 직접 작성할 능력이 없는 담당자라도 기존 스크립트의 경로를 참조 섹션에 추가하는 것만으로 Skill 을 구성할 수 있습니다 [S03]. 반나절 워크숍 한 번으로 사내 각 부서 담당자가 자신의 업무 절차를 Skill 파일로 옮기는 훈련을 완료할 수 있는 이유가 여기 있습니다.

Skill 의 위치와 로드 방식

`kanban-worker` 와 `kanban-orchestrator` 처럼 Hermes Agent 가 제공하는 내장 Skill 은 `skills/devops/` 디렉터리에 위치합니다 [S03]. 사내 커스텀 Skill 은 Profile 선언 파일에 경로를 지정하는 방식으로 로드됩니다. Skill 은 Profile 단위로 로드되기 때문에 영업팀 Profile 이 로드하는 Skill 목록과 HR Profile 이 로드하는 Skill 목록은 완전히 독립적으로 관리됩니다. Skill 파일 자체는 텍스트 파일이므로 Git 저장소에서 버전 관리하고 코드 리뷰 프로세스를 통해 품질을 검증할 수 있습니다.

4.3.2 Archive — 대화·산출물·결정 근거의 보존 계층

Archive 가 보존하는 세 가지 항목

Archive 는 Hermes Agent 의 Persistent Memory(영속 메모리) — 에이전트가 재시작 이후에도 이전 맥락을 유지하는 장기 기억 계층 — 의 영속 저장 구현체입니다 [S02]. Archive 는 세 종류의 항목을 보존합니다. 첫째, 대화 로그는 에이전트와 사용자·시스템 간의 모든 메시지 교환을 타임스탬프와 함께 기록합니다. 둘째, 산출물은 에이전트가 생성한 문서·코드·보고서·분석 결과 파일 등 작업 산출물의 원본 또는 참조 링크를 보관합니다. 셋째, 결정 근거는 에이전트가 특정 판단을 내릴 때 참고한 맥락·Skill·Tool 호출 이력을 포함합니다.

Archive 의 이중 역할

Archive 가 수행하는 역할은 두 가지로 나뉩니다. 하나는 감사 추적 도구로서의 역할입니다. 특정 에이전트 행동이 문제가 되었을 때 Archive 를 조회하면 그 행동이 어떤 Skill 을 통해, 어떤 Tool 호출 결과를 바탕으로, 어떤 사용자 지시에 따라 이루어졌는지를 재구성할 수 있습니다. 다른 하나는 학습 자산 축적 도구로서의 역할입니다. Curator loop 가 회고를 수행할 때 Archive 에 저장된 이전 대화 패턴과 결정 근거를 참조하여 새로운 Skill 을 생성합니다. Archive 가 없다면 Curator loop 의 학습은 단기 기억에만 의존하게 되어 재시작 이후 학습 효과가 초기화됩니다 [S02].

보존 기간 정책과 정보보호 정합

사내 정보보호 정책과 Archive 보존 기간을 연계하는 것이 중요합니다. 아래 표는 Archive 보존 항목별 권장 보존 기간과 관련 정책을 정리한 것입니다.

보존 항목	권장 보존 기간	관련 정책
대화 로그 (업무 관련)	3년	전자문서 및 전자거래 기본법

보존 항목	권장 보존 기간	관련 정책
산출물 (계약·법무 문서)	5년	상법 제33조 상업장부 보존 기간
결정 근거 (감사 추적)	5년	ISO 27001 A.8.2
Skill 실행 이력	1년	사내 IT 운영 정책
개인정보 포함 대화	처리 목적 달성 시 즉시 삭제	개인정보보호법 제21조

Archive 보존 정책은 Hermes Agent 설정 파일에서 각 Profile 단위로 지정할 수 있습니다. 개인정보가 포함될 가능성이 있는 HR Profile 의 Archive 는 짧은 보존 기간과 자동 삭제 정책을 적용하고, 계약 관련 법무 Profile 의 Archive 는 5년 보존 정책을 적용하는 방식으로 Profile 단위 세분화가 가능합니다.

4.3.3 Curator loop — 15회 tool call 마다 회고·Skill 작성·다음 실행 로드

Curator loop 의 작동 원리

Curator loop 는 Hermes Agent 가 외부 학습 없이 스스로 성능을 개선하는 자기 개선 메커니즘입니다 [S02]. 에이전트가 15회의 tool call 을 누적하거나 복잡한 작업 한 건을 완료할 때마다 Curator 가 자동으로 실행됩니다. Curator 는 직전 작업 세션의 대화 로그와 tool call 이력을 Archive 에서 읽어 회고를 수행하고, 반복 가능한 패턴을 식별하여 새로운 `SKILL.md` 파일을 생성합니다. 생성된 Skill 파일은 해당 Profile 의 Skill 디렉터리에 자동 저장되고, 다음 실행 시 자동 로드됩니다.

그림 4-3. Curator loop — 15 tool call 마다 회고 → Skill 생성 → 다음 실행 로드

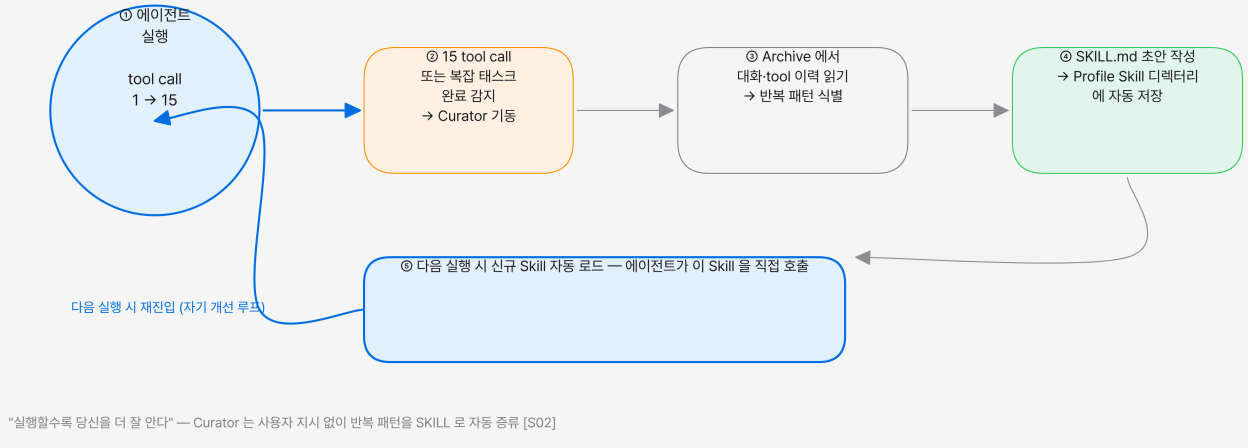


그림 4-3: Curator loop 시퀀스 — 에이전트가 15회 tool call 누적 시 Curator 가 기동되어 Archive 에서 이력을 읽고 회고를 수행한 뒤 SKILL.md 를 생성하고, 다음 실행 시 해당 Skill 이 자동 로드되는 순환 구조.

Curator loop 흐름을 단계별로 살펴보면 다음과 같습니다. 에이전트는 사용자 요청에 따라 tool call 을 누적합니다. 15회 또는 복잡 작업 완료 시점에 Curator 가 기동되고, Archive 에서 현재 세션의 tool call 이력·대화 내용·산출물 경로를 로드합니다. Curator 는 "이 작업을 다음에 더 효율적으로 수행하려면 어떤 절차를 패키징해야 하는가"를 판단하고 Skill 초안을 작성합니다. Skill 초안이 Skill 디렉터리에 저장되면 해당 Profile 은 다음 실행 시점부터 이 Skill 을 호출 후보로 인식합니다.

시간이 지날수록 사내 업무에 최적화

Curator loop 의 운영적 가치는 사용자가 따로 학습을 지시하지 않아도 에이전트가 반복 업무를 자동으로 Skill 화한다는 점입니다 [S02]. 영업 Profile 이 매주 동일한 CRM 데이터 조회 → 요약 → Slack 보고 작업을 반복하면, Curator 가 이 패턴을 인식하고 `weekly-sales-summary` Skill 을 자동 생성합니다. 이후 같은 요청이 들어오면 에이전트는 Skill 을 직접 호출하여 더 빠르고 일관된 결과를 냅니다. 6개월 운영 후에는 자동 생성된 Skill 수와 활용률이 측정 가능한 KPI 가 됩니다.

운영 시점	누적 자동 생성 Skill (예시)	Skill 활용 비율	주요 효과
1개월	3~8개	20~30%	반복 패턴 초기 식별
3개월	15~30개	45~60%	주요 업무 프로세스 Skill 화 완료
6개월	40~70개	70~85%	에이전트 응답 속도·일관성 체감 향상
12개월	80~150개	85~95%	사내 업무 특화 에이전트로 성숙

Curator loop 와 Kanban 의 연동

Curator loop 는 Kanban 과도 연동됩니다. Kanban 카드 단위로 완료된 작업이 Archive 에 기록되므로 Curator 는 카드 단위 작업 패턴을 분석하여 특정 카드 유형(예: `tenant: 계약-검토`)에 특화된 Skill 을 생성할 수 있습니다. 이는 단순한 tool call 패턴을 넘어 업무 유형 단위의 학습이 이루어진다는 의미입니다. Profile · Kanban · Skill · Archive 네 추상화가 Curator loop 를 매개로 하나의 자기 개선 체계를 이루는 방식이 Hermes Agent 가 다른 AI 에이전트 오케스트레이션 도구와 구별되는 핵심 지점입니다 [S01].

5장. Hermes Agent 주요 기능 — LiteLLM · Skill · Cron · Tools · Plugin · Multi-agent

Hermes Agent 가 제공하는 6개 핵심 기능은 사내 AI 자동화를 구성하는 층위를 명확히 나눕니다. 모델 연결과 비용 통제를 담당하는 LiteLLM 게이트웨이 계층, 업무 절차를 코드 없이 정의하는 Skill 시스템, 정기 실행을 담당하는 Cron, 외부 시스템 호출을 표준화하는 Tools, 본체 코드를 건드리지 않고 기능을 추가하는 Plugin SDK, 복수의 에이전트가 역할을 분담하는 Multi-agent

패턴이 그 여섯 층위입니다. 각 층위는 독립적으로 도입하고 점진적으로 확장할 수 있도록 설계되어 있어, 조직이 작은 PoC(Proof of Concept, 개념 검증)부터 시작해 전사 운영 규모까지 단계적으로 성장할 수 있습니다.

5.1 LiteLLM 통합과 MCP 표준 채택

모델 선택의 자유와 도구 생태계 접근 범위는 AI 에이전트 도입 초기에 결정하기 어려운 문제입니다. 어떤 LLM이 사내 업무에 가장 적합한지는 실제 운영 데이터를 쌓기 전에는 판단하기 어렵고, 외부 시스템 연동 범위도 사용자 요구사항이 구체화될수록 달라집니다. Hermes Agent 는 이 두 가지 불확실성을 각각 LiteLLM과 MCP(Model Context Protocol, 모델 컨텍스트 프로토콜)라는 개방 표준에 위임하여 해소합니다. 모델과 도구 모두 표준 인터페이스 위에서 교체 가능한 구조이므로, 초기 선택이 장기 운영의 족쇄가 되지 않습니다.

5.1.1 LiteLLM 100+ provider 지원과 비용·fallback 통합 관리

LiteLLM 이 해결하는 문제

LiteLLM(MIT 라이선스)은 100개 이상의 LLM provider를 단일 OpenAI 호환 엔드포인트로 통합하는 오픈소스 게이트웨이입니다 [S10]. OpenAI · Anthropic · AWS Bedrock · Google VertexAI · Cohere · HuggingFace · vLLM · NVIDIA NIM 등 사실상 모든 상용·오픈소스 모델이 동일한 API 호출 형식으로 연결되므로, Hermes 측 코드는 모델이 바뀌어도 수정이 필요 없습니다. Hermes Agent 는 `setup` 명령으로 40개 이상의 provider를 네이티브로 등록하고, LiteLLM 프록시를 통해 에이전트 스웜(swarm, 복수 에이전트 집합) 전체를 단일 엔드포인트로 라우팅할 수 있습니다.

Virtual Key 기반 비용 통제

LiteLLM의 virtual key(가상 API 키)는 부서별·팀별·프로젝트별로 발급하여 각각에 월간 지출 한도(budget cap)와 분당 요청 제한(RPM), 분당 토큰 제한(TPM)을 독립적으로 설정할 수 있습니다 [S10]. 한도 초과 시 해당 키의 요청은 자동 거절되고, soft budget 알림 기능을 통해 팀이 한도에 근접하기 전에 이메일 경보를 받습니다. 예를 들어 인사팀 virtual key에 GPT-4o 월 100달러 상한을 설정하고 개발팀 virtual key에는 Claude 3.5 Sonnet 200달러 상한을 별도로 부여하면, 동일한 LiteLLM 인스턴스에서 부서별 청구 분리가 실현됩니다.

아래 표는 사내 부서별 virtual key 운영의 전형적인 구성 예시입니다.

부서	할당 모델	월 지출 한도	Fallback 모델	비고
인사팀	claude-3-5-haiku	$50	gpt-oss-20b (로컬)	개인정보 처리 업무
개발팀	claude-3-5-sonnet	$200	claude-3-5-haiku	코드 리뷰·PR 요약
재무팀	gpt-4o	$150	azure-gpt-4o	감사 추적 의무
마케팅팀	gemini-1.5-pro	$80	gpt-4o-mini	콘텐츠 생성

부서	할당 모델	월 지출 한도	Fallback 모델	비고
공통 Cron 작업	gpt-oss-20b (로컬)	제한 없음	—	정기 보고서·배치

Fallback 과 부하 분산

LiteLLM은 지정 provider가 오류를 반환하거나 응답 시간이 임계값을 초과하면 fallback(대체 경로) 모델로 자동 전환합니다 [S10]. 메인 모델에 Anthropic Claude를 두고 fallback에 로컬 gpt-oss 20B를 지정하면, 클라우드 API 장애 시에도 에이전트 운영이 중단되지 않습니다. Netflix · Lemonade · Rocket Money 등 실제 운영 환경에서 LiteLLM을 채택한 사례가 보고되어 있으며, 이는 프로덕션 수준 안정성을 방증합니다 [S10]. MSAP.ai와 같은 국내 LLM 라우터 솔루션을 이미 도입한 조직이라면 LiteLLM과 병행 구성하거나 대체 경로로 등록하는 방식으로 결합할 수 있습니다.

5.1.2 MCP 표준 — 10,000+ public servers 와 vendor-neutral governance

MCP 의 탄생과 거버넌스 이관

MCP는 2024년 11월 Anthropic이 AI 모델과 외부 도구·데이터 소스·비즈니스 시스템을 연결하는 개방 표준으로 발표했습니다 [S11]. 특정 벤더가 소유하는 독점 규격이 아니라, 누구나 MCP 서버를 구현하고 MCP 클라이언트를 만들 수 있는 공개 프로토콜입니다. 2025년 12월에는 Anthropic · Block · OpenAI가 공동 설립한 Agentic AI Foundation(AAIF, 에이전틱 AI 재단)이 Linux Foundation 산하 단체로 출범하면서 MCP의 거버넌스가 중립 기구로 이관되었습니다 [S11]. OpenAI · Google DeepMind · Microsoft · AWS · Cloudflare · Bloomberg이 AAIF를 지지하는 형태이므로, MCP는 특정 AI 기업의 전략 변화에 종속되지 않는 산업 공통 인프라로 자리를 잡았습니다.

채택 규모와 서버 카탈로그

2026년 3월 기준 10,000개 이상의 활성 public MCP 서버가 운영 중이고, Python과 TypeScript SDK의 월간 다운로드 합산은 9,700만 건에 달합니다 [S11]. ChatGPT · Claude · Cursor · Gemini · Microsoft Copilot · VS Code 등 주요 AI 클라이언트가 MCP를 공식 지원하며, Slack · GitHub · Salesforce · Stripe · HubSpot · Shopify · Notion · Linear · Sentry · Figma 등 기업 업무 시스템의 공식 또는 커뮤니티 MCP 서버가 이미 공개되어 있습니다 [S11]. Stacklok의 2026년 보고서에 따르면 41%의 조직이 MCP를 프로덕션 환경에 도입한 것으로 집계되었습니다.

아래 표는 MCP 표준의 주요 채택 타임라인과 거버넌스 이관 과정을 정리한 것입니다.

시점	주요 사건
2024-11	Anthropic, MCP 오픈 표준 발표
2025-03	OpenAI 공식 채택
2025 중반	Google DeepMind · Microsoft · 수천 개 기업팀 채택

시점	주요 사건
2025-12	Linux Foundation AAIF 설립, MCP 거버넌스 이관 (Anthropic · Block · OpenAI 공동 설립)
2026-03	10,000+ 활성 public servers · 97M 월간 SDK 다운로드
2026-04	MCP Dev Summit North America (뉴욕, 약 1,200명 참석)

사내 시스템을 MCP 서버로 노출하는 가치

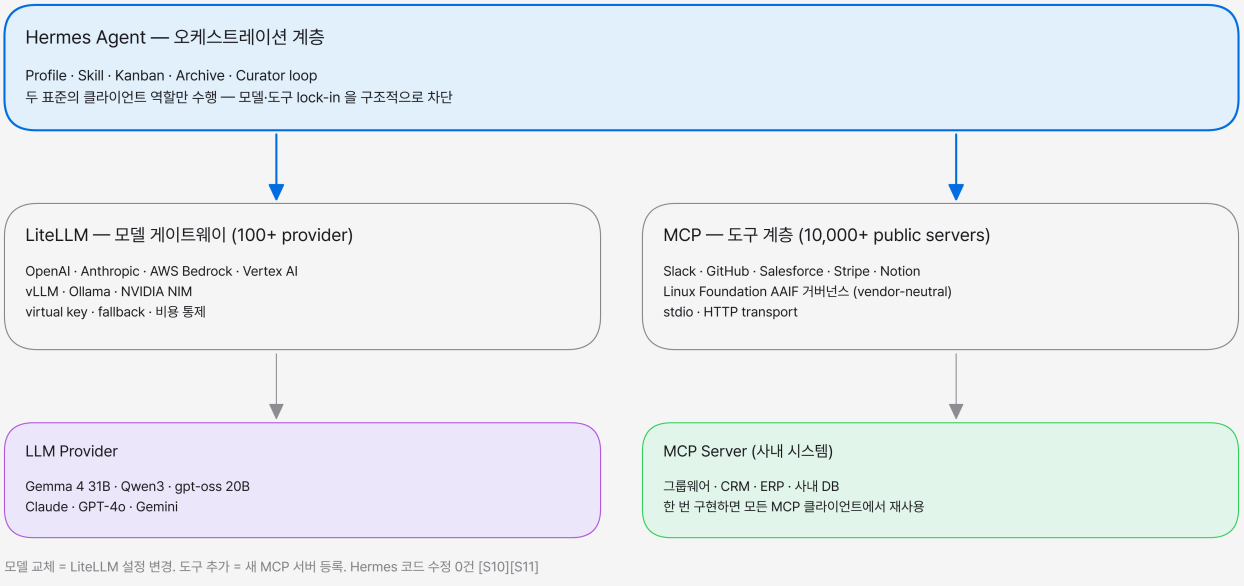
MCP 표준의 실질적 이점은 한 번 구현한 서버 인터페이스가 모든 MCP 클라이언트에서 재사용된다는 점입니다. 사내 그룹웨어를 MCP 서버로 노출하면, 현재 사용하는 Hermes Agent뿐 아니라 이후 도입하는 어떤 MCP 클라이언트 도구도 동일한 인터페이스로 그룹웨어에 접근할 수 있습니다. 도구 연동 코드를 특정 에이전트 제품마다 다시 작성할 필요가 없어지므로, 사내 시스템 통합 투자가 플랫폼 종속 없이 누적됩니다 [S11].

5.1.3 LiteLLM (모델 표준) + MCP (도구 표준) 두 표준의 결합 가치

두 lock-in 을 동시에 해소하는 구조

AI 에이전트 도입에서 조직이 가장 우려하는 리스크는 두 가지입니다. 하나는 특정 LLM 벤더에 의존하게 되어 모델 교체 시 막대한 재작업이 발생하는 모델 lock-in이고, 다른 하나는 특정 에이전트 플랫폼의 도구 연동 방식에 종속되어 다른 시스템으로 이전하기 어려워지는 도구 lock-in입니다. Hermes Agent는 모델 레이어를 LiteLLM으로, 도구 레이어를 MCP로 분리함으로써 이 두 lock-in을 구조적으로 차단합니다 [S10] [S11].

그림 5-1. Hermes 표준 스택 — Hermes / LiteLLM / MCP 3계층



세 계층으로 구성된 Hermes 표준 스택 — 최상위에 Hermes Agent 오케스트레이션 계층, 중간에 LiteLLM 모델 게이트웨이 계층 (100+ provider), 하위에 MCP 도구 계층 (10,000+ public

servers). 화살표 방향: Hermes → LiteLLM → 각 LLM provider / Hermes → MCP client → 각 MCP server.

이 구조에서 Hermes는 두 표준의 클라이언트 역할만 수행합니다. 모델을 교체할 때는 LiteLLM 라우팅 설정만 변경하면 되고, 연동할 외부 시스템이 늘어날 때는 해당 시스템의 MCP 서버를 추가하면 됩니다. Hermes 자체 코드는 두 경우 모두 수정이 필요 없습니다.

표준 위에 표준 — 5년 단위 투자 보호 논거

클라우드 인프라 도입 시 Kubernetes를 선택하는 이유 중 하나가 특정 클라우드 벤더의 독자 컨테이너 관리 방식에 종속되지 않는 개방 표준이라는 점이었습니다. LiteLLM과 MCP는 AI 에이전트 레이어에서 같은 역할을 합니다. 두 표준 모두 Linux Foundation 계열 또는 이에 준하는 중립 거버넌스 아래에서 운영되어, 단일 기업의 전략 변화로 표준이 폐기되거나 접근이 제한될 리스크가 낮습니다. AI 에이전트 플랫폼을 지금 선택하더라도, 모델 선택과 도구 연동이 표준 레이어에서 분리되어 있으면 3~5년 후 기술 교체 비용이 현저히 낮아집니다.

5.2 Skill 시스템 · Cron · Tools

Skill · Cron · Tools 는 Hermes Agent 에서 실제 업무를 자동화하는 세 가지 기본 구성 요소입니다. Skill 은 반복 가능한 업무 절차를 문서로 정의하여 에이전트에게 학습시키는 방식이고, Cron 은 그 절차를 시간 기반으로 자동 실행하는 스케줄러이며, Tools 는 사내 외부 시스템과의 실제 데이터 교환을 담당하는 호출 인터페이스입니다. 세 요소가 결합되면 "매일 09시에 ERP에서 전일 판매 데이터를 가져와 요약 보고서를 Slack에 발송한다"는 업무 자동화가 단일 Hermes 인스턴스에서 구현됩니다.

5.2.1 Skill 정의·등록·트리거 메커니즘

Skill 의 본질과 SKILL.md 구조

Skill은 에이전트가 특정 업무를 처리하는 방법을 담은 마크다운 문서입니다 [S03]. 일반적인 자동화 도구에서 업무 절차를 코드로 작성해야 하는 것과 달리, Hermes의 Skill은 담당자가 평문으로 절차를 기술하면 에이전트가 그 절차를 실행 지침으로 해석합니다. SKILL.md 파일에는 `name`, `description`, `version`, `author`, `tags`, `related_skills` 항목과 함께 실행 전제 조건 및 처리 절차를 작성합니다. `description` 필드의 내용이 자동 선택의 기준이 되므로, 이 필드를 구체적으로 작성할수록 에이전트가 적절한 상황에서 해당 Skill을 정확히 호출합니다.

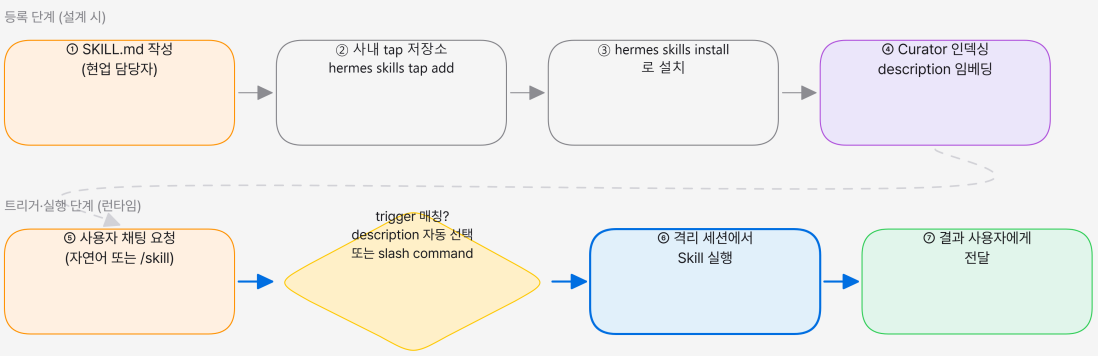
등록과 배포

Skill 등록은 세 가지 경로로 가능합니다 [S03]. 첫째, 공개 허브에서 `hermes skills install <ID>` 명령으로 검증된 Skill을 설치합니다. 둘째, `hermes skills browse` 로 허브 카탈로그를 탐색하여 필요한 Skill을 선택합니다. 셋째, 사내에서 자체 제작한 SKILL.md를 `hermes skills tap add <REPO>` 명령으로 사내 저장소에 등록하고 전 직원이 재사용합니다. 세 번째 경로가 사내 Skill 카탈로그 운영의 핵심이며, 한 번 검증된 절차가 전 조직에 전파되는 확산 구조입니다.

트리거와 세션 내 활성화

등록된 Skill은 채팅 세션에서 `/skill <이름>` 슬래시 명령으로 즉시 활성화하거나, 에이전트 시작 시 `hermes -s <SKILL>` 전역 플래그로 특정 Skill을 기본 활성화 상태로 띄울 수 있습니다 [S03]. 플랫폼별(CLI · 게이트웨이 · Telegram 채널 등) 활성화 범위는 `hermes skills config` 로 독립 관리하므로, 운영 채널에는 검증된 Skill만 노출하고 개발 채널에는 실험적 Skill을 허용하는 거버넌스가 가능합니다. Curator 시스템이 사용 빈도를 추적하고 장기간 비활성화된 Skill을 자동 보관 처리하므로, 카탈로그가 시간이 지나도 불필요한 항목으로 비대해지지 않습니다.

그림 5-2. Skill 등록·트리거 시퀀스 — SKILL.md 작성부터 실행까지



출처: Hermes Agent 공식 문서 [S03] — Skill 카탈로그 거버넌스로 자유 창작과 운영 안정성 양립

Skill 등록부터 실행까지의 단계적 흐름 — SKILL.md 작성 → 허브 등록 또는 사내 tap 등록 → `hermes skills install` → Curator 인덱싱 → 사용자 채팅 요청 → description 기반 자동 선택 또는 `/skill` 슬래시 명령 → 격리 세션에서 Skill 실행 → 결과 전달.

사내 Skill 카탈로그 거버넌스 제안

Skill 확산 속도가 빨라질수록 품질 편차도 커집니다. 초기 도입 단계에서 Skill 등록 승인 프로세스를 정의해 두는 것이 권고됩니다. 제안하는 흐름은 (1) 현업 담당자가 SKILL.md 초안 작성 → (2) IT 검토자가 보안 의존성·데이터 접근 범위 확인 → (3) 승인 후 사내 tap 저장소 병합 → (4) 6개월 주기 Curator 리포트 기반 폐기 검토입니다. 이 절차를 갖추면 자유로운 창작을 허용하면서도 운영 환경의 안정성을 유지할 수 있습니다.

5.2.2 Cron — 시간 기반 자동화 (배치·정기 보고)

Cron 의 실행 구조

Hermes Agent의 Cron은 게이트웨이 데몬이 60초마다 스케줄러를 확인하고, 실행 시점이 도래한 작업을 격리된 에이전트 세션에서 자동으로 구동합니다 [S01]. 스케줄 표현식은 표준 cron 형식(`0 9 * * *`), 자연어 간격(`every 2h`), 상대 지연(`30m`), ISO 타임스탬프(예: `2026-03-15T09:00:00`) 네 가지 형식을 모두 지원합니다. 자연어 방식은 담당자가 별도의 cron 문법 학습 없이 "매일 오전 9시에 일일 보고서 발송"을 직접 등록할 수 있어 현업 부서 자가 운영에 적합합니다.

비용 절감 설계 — No-agent mode

Cron 작업에는 LLM 호출이 반드시 필요하지 않은 경우도 있습니다. Hermes Cron의 no-agent mode는 사전 정의된 스크립트만 실행하고 LLM을 호출하지 않으므로, 단순 데이터 수집·파일 백업·상태 체크 작업에서 토큰 비용이 발생하지 않습니다 [S01]. `script` + `wakeAgent` 조합을 사

용하면 스크립트가 먼저 실행되어 조건을 확인하고, 특정 기준을 충족할 때만 에이전트를 깨워 LLM을 호출하는 비용 최적 설계가 가능합니다. 매일 자정 서버 로그를 수집하는 작업은 no-agent mode로, 수집된 로그에서 이상 징후가 감지된 경우에만 에이전트 모드로 분석 보고서를 생성하는 방식입니다.

정기 자동화 시나리오

아래 표는 사내 자동화에서 자주 활용하는 Cron 스케줄 5종입니다.

주기	Cron 표현식	자동화 내용	전달 채널
일간	`0 9 * * 1-5`	전일 주요 지표 요약 보고서	Slack #daily-report
주간	`0 8 * * 1`	주간 팀 회의록 · 액션 아이템 정리	이메일 · Telegram
월간	`0 0 1 * *`	전월 LLM 비용 · Skill 사용 현황 리포트	관리자 Slack DM
분기	`0 0 1 1,4,7,10 *`	분기 KPI 대시보드 요약 · 이상 지표 플래그	PDF 생성 후 이메일
연간	`0 0 1 1 *`	연간 AI 자동화 산출물 목록 아카이브	내부 위키 페이지 갱신

Cron 결과물은 `deliver` 옵션으로 Telegram · Discord · Slack · 이메일 채널에 동시 전달하거나 로컬 파일로 저장할 수 있어, 수신자별 채널 선호도를 수용합니다 [S01]. PoC 첫 4주의 가시적 성과로 "일일 보고서 1건 자동화"를 선정하면, 경영진에게 실제 운영 효과를 빠르게 시연할 수 있습니다.

5.2.3 Tools — 외부 시스템 호출 인터페이스 (MCP 기반)

Tools 의 역할과 MCP 결합

Tools는 Hermes Agent가 외부 시스템(ERP · CRM · 그룹웨어 · 사내 데이터베이스 등)을 직접 호출하는 인터페이스입니다 [S11]. Tools 레이어가 MCP 클라이언트로 동작하므로, MCP 서버로 노출된 모든 외부 시스템을 별도 SDK 없이 표준 프로토콜로 호출합니다. 반대로 사내 시스템을 MCP 서버로 구현하면 Hermes를 비롯한 모든 MCP 클라이언트가 해당 시스템에 접근할 수 있어, 연동 코드를 중복 작성하는 공수가 사라집니다. MCP 서버는 stdio와 HTTP 두 가지 transport를 지원하므로, 기존 REST API를 보유한 사내 시스템은 MCP 래퍼를 비교적 적은 작업량으로 추가할 수 있습니다 [S11].

통합 우선순위 매트릭스

사내 시스템 통합 범위를 한 번에 넓히려 할 경우 PoC 일정이 길어지고 실패 리스크가 커집니다. 아래 표는 통합 가치(업무 영향)와 구현 난이도를 기준으로 우선순위를 분류한 것입니다.

우선순위	시스템 유형	대표 예시	구현 난이도	도입 가치
1순위	그룹웨어 · 캘린더	Google Workspace · 캘린더	낮음 (공식 MCP 서버 제공)	높음 (일정·회의·알림 자동화)
2순위	사내 메신저	Slack · Teams	낮음 (공식 MCP 서버 제공)	높음 (보고·알림 채널 직결)
3순위	프로젝트 관리	Jira · Linear · Notion	중간 (커뮤니티 MCP 서버)	중~높음 (태스크 자동 생성)
4순위	CRM	Salesforce · HubSpot	중간 (공식 MCP 서버 제공)	중간 (영업 데이터 자동 요약)
5순위	ERP · 회계	SAP · 더존	높음 (MCP 래퍼 직접 구현)	높음 (단, 데이터 보안 검토 필수)

1·2순위 통합이 완료된 PoC 단계에서 에이전트가 그룹웨어 일정을 읽고 Slack에 회의 요약을 자동 발송하는 기본 루프가 가동되면, 조직 내 도입 효과 공감대가 빠르게 형성됩니다 [S11].

5.3 Plugin 과 Multi-agent 패턴

Hermes Agent 가 단일 에이전트 자동화를 넘어 조직 전체 업무 흐름을 처리하려면, 기능 확장과 에이전트 간 협업 두 가지 레이어가 필요합니다. Plugin SDK는 Hermes 본체 코드를 수정하지 않고 새로운 기능을 덧붙이는 확장 레이어이고, Multi-agent 패턴은 복수의 Hermes 에이전트가 역할을 나눠 병렬 또는 순차적으로 협력하는 협업 레이어입니다. 두 레이어를 갖춤으로써 Hermes는 단순 챗봇에서 출발해 팀 단위·부서 단위·전사 단위로 자연스럽게 운영 범위를 넓힐 수 있습니다.

5.3.1 Plugin SDK — 기능 확장의 외부 인터페이스

Plugin SDK 의 설계 원칙

Plugin SDK는 Hermes 본체 코드를 건드리지 않고 네 가지 범주에서 기능을 추가하는 외부 인터페이스입니다 [S01]. 네 가지 범주는 (1) 커스텀 모델 provider 추가, (2) 사내 인증 시스템 연동, (3) 메트릭 export 및 모니터링 통합, (4) 커스텀 채널 어댑터입니다. 본체 코드를 수정하지 않는다는 원칙은 두 가지 실용적 이점을 만듭니다. 하나는 Hermes 공식 릴리스가 올라올 때 Plugin이 충돌 없이 유지된다는 점이고, 다른 하나는 Plugin이 잘못 동작해도 본체 안정성이 보장된다는 점입니다.

Plugin 관리와 인터페이스 카테고리

Plugin은 `hermes plugins list` · `hermes plugins install` · `hermes plugins remove` 세 명령으로 설치·관리합니다 [S01]. 플랫폼 어댑터(`plugins/platforms/`) 계층은 Telegram · Discord · Slack · WhatsApp 외에 조직이 자체 개발한 메신저나 인트라넷 포털을 Hermes 채널로 연결하는 데 사용됩니다. MCP 서버 통합 Plugin은 5.2.3에서 설명한 MCP 클라이언트 기능을 플러그인 형태로

선택 설치할 수 있도록 하고, 웹훅 트리거 Plugin은 외부 시스템이 이벤트 발생 시 HTTP POST로 Hermes 작업을 직접 호출하는 인바운드 연동 경로를 제공합니다.

버저닝 정책 확인 권고

Plugin 인터페이스의 장기 안정성은 semver(Semantic Versioning, 의미론적 버전 관리) 준수 여부에 달려 있습니다. 도입 조직은 Hermes 공식 릴리스 노트에서 Plugin API 주 버전(major version) 변경 여부를 추적하고, 메이저 버전 업그레이드 시 사내 Plugin의 호환성을 사전 검증하는 절차를 운영 가이드라인에 포함시키는 것이 권고됩니다 [S01]. MIT 라이선스 오픈소스이므로 소스 코드를 직접 포크(fork)하여 사내 버전을 유지하는 방안도 있지만, 이 경우 업스트림 보안 패치 적용이 지연되는 리스크를 함께 고려해야 합니다.

5.3.2 Multi-agent 3 패턴 — Coordinator + Worker · Pipeline · Fan-out

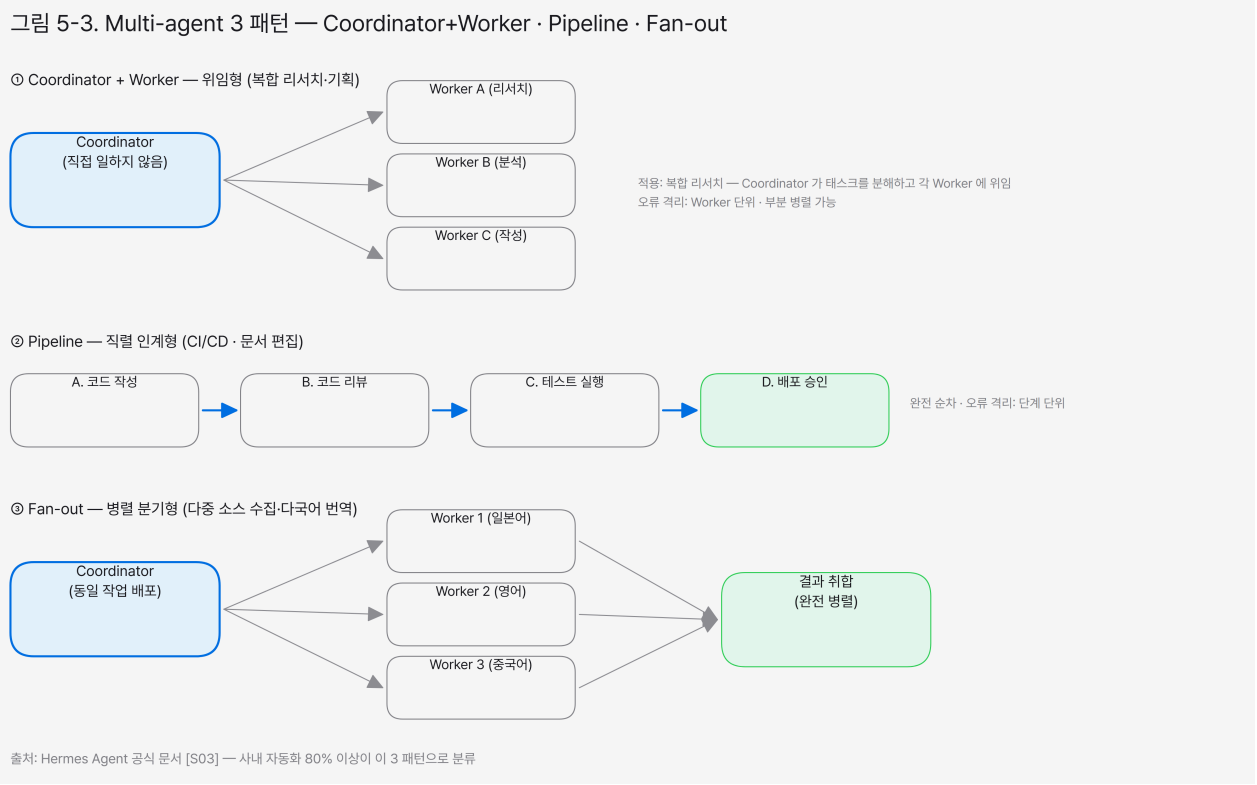
3 패턴의 정의와 적용 상황

Hermes Agent가 지원하는 Multi-agent 패턴은 Coordinator + Worker, Pipeline(직렬 인계), Fan-out(병렬 분기) 세 가지입니다 [S03]. 각 패턴은 업무 특성에 따라 선택하며, 복잡한 프로젝트에서는 세 패턴을 중첩하여 사용합니다.

Coordinator + Worker 패턴에서는 Coordinator 에이전트가 전체 목표를 파악하고 세부 작업을 분해하여 전문 Worker 에이전트에게 할당합니다. Coordinator는 직접 일하지 않고 작업 흐름을 관리하며, 각 Worker는 자신에게 할당된 작업만 처리합니다 [S03]. 리서치 에이전트가 자료를 수집하면 분석 에이전트가 처리하고 작성 에이전트가 최종 문서를 생성하는 구성이 전형적인 사례입니다.

Pipeline(파이프라인) 패턴은 에이전트 A가 완성한 산출물을 에이전트 B가 이어받아 처리하는 직렬 인계 구조입니다. 각 에이전트는 이전 단계의 결과만 입력으로 받아 다음 단계로 넘기므로, 처리 책임이 명확히 분리됩니다. 코드 작성 → 코드 리뷰 → 테스트 실행 → 배포 승인 요청의 CI/CD 흐름이 Pipeline 패턴의 대표 적용 사례입니다 [S03].

Fan-out(팬아웃) 패턴은 하나의 Coordinator가 동일한 작업을 복수의 Worker에게 동시 배포하고 결과를 취합하는 병렬 분기 구조입니다. 여러 데이터 소스에서 동시에 정보를 수집하거나, 동일 문서를 복수의 언어로 동시 번역하거나, 복수의 API를 병렬 호출해야 하는 상황에 적합합니다 [S03].



Multi-agent 3 패턴 시각화 — 상단: Coordinator + Worker (Coordinator 1개 → Worker A, B, C 각각 배정); 중단: Pipeline (에이전트 A → 에이전트 B → 에이전트 C 직렬 인계); 하단: Fan-out (Coordinator 1개 → Worker 1, 2, 3 동시 배포 → 결과 취합). 각 패턴 옆에 대표 적용 시나리오 1줄 표기.

패턴 선택 매트릭스

아래 표는 사내 자동화 시나리오를 세 패턴으로 분류하는 기준입니다.

선택 기준	Coordinator + Worker	Pipeline	Fan-out
작업 의존성	작업 간 의존성 복잡	단계별 순서 의존	작업 간 독립
병렬 여부	부분 병렬 가능	완전 순차	완전 병렬
대표 시나리오	복합 리서치·기획	CI/CD · 문서 편집 파이프라인	다중 소스 수집·번역
오류 격리	Worker 단위	단계 단위	Worker 단위
권장 Kanban 활용	필수 (카드 할당)	선택 (단계 추적)	선택 (취합 확인)

사내 자동화 요구사항 80% 이상이 이 세 패턴 중 하나로 분류됩니다 [S03]. 자동화 요구사항 정의 시 각 항목을 어느 패턴에 해당하는지 미리 분류해 두면 구현 공수 추정과 PoC 범위 선정이 수월해집니다.

5.3.3 Kanban + Profile 결합으로 구현하는 Coordinator + Worker 사례

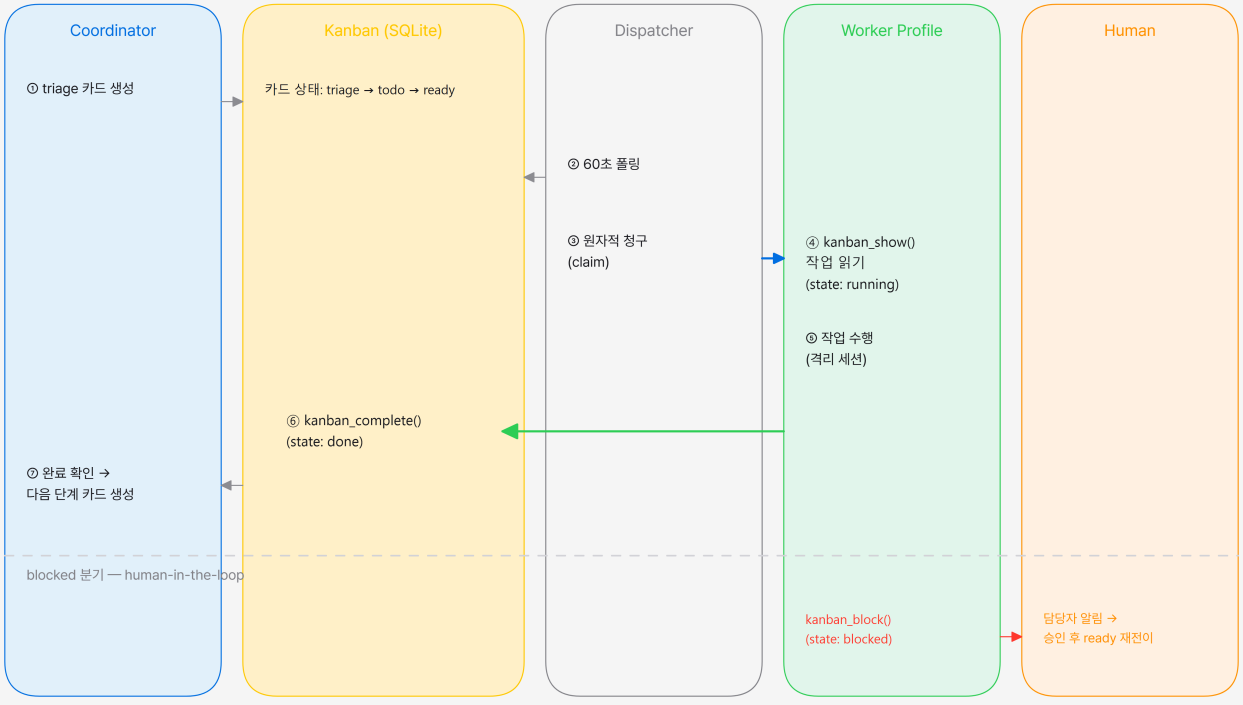
Kanban 이 Multi-agent 의 backbone 인 이유

Hermes Agent의 Kanban은 SQLite 기반 내구성 있는 작업 큐(durable task queue)로, 복수의 Profile(에이전트 인스턴스)이 공유하는 `~/.hermes/kanban.db` 파일에 저장됩니다 [S03]. 단순한 RPC(Remote Procedure Call, 원격 프로시저 호출) 호출과 달리 Kanban은 에이전트가 비정상 종료되거나 네트워크가 단절되더라도 카드 상태가 데이터베이스에 남아 있어 재시작 후 중단 지점에서 작업을 재개할 수 있습니다. 이 내구성이 Multi-agent 워크플로우에서 Kanban을 단순 채팅 위임(`delegate_task`)보다 우선하는 근거입니다.

카드 흐름과 컬럼 구조

Kanban 카드는 `triage → todo → ready → running → blocked → done / archived` 순서로 상태가 전이됩니다 [S03]. triage 컬럼에 적재된 초기 아이디어는 LLM이 자동으로 자식 작업 그래프로 분해하여 todo 카드들을 생성합니다. ready 상태 카드는 Dispatcher가 60초 간격으로 확인하여 원자적(atomic)으로 청구(claim)하고, 카드에 지정된 assignee(담당 Profile 이름)를 Worker로 생성합니다. Worker는 `kanban_show()` 도구로 자신의 카드를 읽고 작업을 수행한 뒤, `kanban_complete()` 또는 `kanban_block()` 으로 상태를 갱신합니다 [S03].

그림 5-4. Coordinator + Worker Kanban 시퀀스 — 원자적 청구 + human-in-the-loop



Coordinator + Worker Kanban 시퀀스 다이어그램 — Coordinator Profile이 triage 카드 생성 → Dispatcher가 60초 주기로 ready 카드 감지 → Worker Profile A에 카드 할당(running 전이) → Worker가 kanban_show()로 작업 읽기 → 작업 수행 → kanban_complete()로 done 전이 → Coordinator가 완료 확인 후 다음 단계 카드 생성. blocked 분기: Worker가 kanban_block() 호출 시 담당자 알림 → 인간 개입 후 ready 재전이.

Coordinator Profile 과 Worker Profile 의 역할 분리

잘 설계된 Coordinator는 직접 작업을 수행하지 않습니다 [S03]. Coordinator Profile의 역할은 전체 목표를 이해하고, 세부 작업 카드를 생성·분류하고, 적합한 Worker Profile에 할당하고, 완료된 카드의 결과를 취합하는 것으로 한정됩니다. Worker Profile은 자신에게 할당된 카드의 내용만 읽고 처리하며, 다른 Profile의 컨텍스트를 오염시키지 않습니다. 예를 들어 "주간 경쟁사 분석 리포트 작성" 요청이 들어오면 Coordinator(PM 봇 역할)가 리서치·분석·작성·검토 네 개의 카드를 생성하고, 각 카드를 리서치 Worker · 분석 Worker · 작성 Worker · 리뷰 Worker에게 순서대로 할당합니다. 리뷰 Worker가 자신이 작성한 콘텐츠를 스스로 승인하는 상황은 역할 격리 원칙에 의해 구조적으로 차단됩니다.

멀티테넌트와 인간 개입

Kanban은 `--tenant` 플래그로 사업부별 데이터를 격리합니다 [S03]. 마케팅팀 Coordinator와 개발팀 Coordinator가 동일한 Hermes 인스턴스에서 독립적으로 운영되어도 서로의 카드를 볼 수 없습니다. blocked 상태 카드는 Web UI 대시보드에서 드래그-드롭으로 ready 상태로 되돌리거나 담당자 코멘트를 추가할 수 있어, 에이전트가 단독으로 판단할 수 없는 상황에서 사람이 승인·보정하는 human-in-the-loop(사람 개입 루프) 흐름을 자연스럽게 유지합니다 [S03]. 전사 규모의 자동화에서도 중요 결정이 에이전트 단독으로 처리되는 상황을 막는 안전망 역할을 이 구조가 담당합니다.

6장. 커뮤니케이션 채널 통합 — Telegram · Slack · 사내 메신저

Hermes Agent 가 실무에서 가치를 발휘하려면 사용자가 이미 쓰는 채널 위에서 동작해야 합니다. 보안 승인을 받은 사내 메신저 창에서 질의하고, 그 자리에서 결과를 받는 구조가 되어야 업무 도구로 자리잡습니다. Hermes 는 Telegram · Discord · Slack · WhatsApp · Signal · CLI 의 6개 채널(native channel)을 단일 gateway 프로세스로 통합합니다 [S01]. 각 채널은 별도의 서버 프로세스 없이 하나의 gateway 아래 메시지를 주고받고, 사용자 식별·권한 판단·실행 로그 기록이 모두 같은 지점에서 처리됩니다. 이 구조는 국내 기업이 흔히 직면하는 과제, 즉 외부 메신저 차단 환경에서도 동일한 설계 패턴으로 사내 메신저 어댑터를 추가해 확장할 수 있습니다. 6장은 6 native 채널의 구성 방식, 국내 사내 메신저 어댑터 작성 가이드, 그리고 채널별 권한과 감사 정책 설계를 차례로 다룹니다.

6.1 6 native 채널의 구성과 단일 gateway 프로세스 구조

Hermes Agent 의 채널 아키텍처는 "채널 수 = 서버 수"가 아닙니다. 하나의 gateway 프로세스가 6개 채널 어댑터를 적재(load)하고, 각 채널에서 수신된 메시지를 단일 내부 이벤트 버스로 전달합니다. 이 설계 덕분에 채널을 추가하거나 교체해도 Hermes 핵심 실행 엔진과 권한 모델에는 손을 댈 필요가 없습니다 [S01].

6.1.1 Telegram 봇 — 가장 빠른 PoC 채널의 장단점

Telegram 봇 등록 절차의 단순성

Telegram 은 Hermes Agent 를 처음 시험해보는 개발자에게 가장 빠른 시작점입니다. BotFather 채널에서 `/newbot` 명령을 실행하면 수십 초 안에 봇 토큰을 발급받습니다. 이후 Hermes 설정

파일에 토큰 한 줄을 추가하고 gateway 를 재시작하면 Telegram 채널이 활성화됩니다 [S01]. 별도의 SSL 인증서 설정이나 OAuth 흐름 없이 long-polling 방식으로 메시지를 수신하므로, 방화벽 포트 개방 없이 사내 개발 서버에서 즉시 동작 확인이 가능합니다. 이 단순함이 PoC(Proof of Concept, 개념 검증) 단계의 핵심 장점입니다.

국내 기업 보안 정책과의 충돌

단순함에는 그에 상응하는 제약이 따릅니다. 국내 금융·공공·제조 기업 다수는 사내 보안 정책상 Telegram 서버(api.telegram.org)로의 아웃바운드 트래픽을 차단합니다. Telegram 은 서버가 러시아 법인 소속이라는 점, 그리고 대화 내용이 Telegram 클라우드 서버를 경유한다는 점에서 정보보호 부서의 승인을 받기 어렵습니다. 업무용 데이터가 외부 메신저 서버를 거치는 구조는 내부 정보 유출 경로로 간주될 수 있습니다.

채널 사용 단계 분리 원칙

따라서 Telegram 채널은 "개발자 PoC" 용도로 한정하는 것이 현실적입니다. 개인 개발 환경이나 격리된 테스트 네트워크에서 Hermes 의 기능을 빠르게 검증하는 데 활용하고, 정식 사내 운용은 보안 승인이 완료된 사내 메신저나 Slack 으로 전환합니다. 이 두 단계를 처음부터 분리해두면 PoC 완료 후 채널 교체 작업이 어댑터 설정 변경 수준으로 끝납니다.

다음은 Telegram 봇 설정의 핵심 단계입니다.

단계	작업	비고
1	BotFather 에서 `/newbot` 실행 → 봇 이름·username 설정	수십 초 소요
2	발급된 API 토큰을 Hermes `config/channels.yaml` 의 `telegram.token` 에 입력	단일 줄
3	Hermes gateway 재시작	`hermes restart`
4	BotFather 에서 `/setprivacy` → Disable 설정 (그룹 채팅 메시지 수신 허용)	선택 사항
5	테스트 채팅에서 메시지 전송 후 Hermes 응답 확인	PoC 완료 기준

6.1.2 Slack 봇 — 사내 협업 채널 통합의 표준 경로

Slack 워크스페이스 관리자 권한 선결 조건

Slack 은 국내 IT 기업·스타트업 사이에서 사내 협업 도구 도입률이 가장 높은 플랫폼입니다. Hermes 를 Slack 채널에 연결하는 작업은 기술적으로 복잡하지 않지만, Slack 앱 등록과 OAuth 흐름을 완료하려면 **워크스페이스 관리자 계정** 이 반드시 필요합니다. 이 권한 확보를 PoC 0주차에 선결 조건으로 명시해두지 않으면 실제 연결 시점에서 진행이 멈추는 경우가 많습니다. IT 담

당자는 프로젝트 시작 전 협업 부서의 Slack 워크스페이스 관리자를 PoC 이해관계자로 등록해야 합니다.

Slack 앱 등록과 OAuth 흐름

Slack 앱은 api.slack.com/apps 에서 생성합니다. "Slack App Token Scopes"(스코프, 권한 범위) 설정이 핵심 단계입니다. Hermes 가 메시지를 수신·발신하고 slash command 에 응답하려면 최소한 `chat:write`, `channels:history`, `app_mentions:read` 스코프가 필요합니다 [S01]. Bot Token Scopes 는 앱 자체 신원으로 동작하므로 특정 사용자 계정에 종속되지 않고 설치 후 계속 유효합니다. OAuth 흐름 완료 후 발급되는 Bot Token 을 Hermes `config/channels.yaml` 의 `slack.bot_token` 필드에 기재하고, Slash command 처리를 위한 소켓 모드(Socket Mode) 를 활성화하면 Hermes gateway 가 Slack 이벤트를 수신합니다.

인터랙티브 메시지와 Slash command 활용

Slack 채널에서 Hermes 의 가치가 가장 두드러지는 기능은 인터랙티브 메시지(Interactive Message)와 Slash command 입니다. 예를 들어 `/hermes deploy` 명령을 Slack 채널에 입력하면 Hermes 가 Kanban 태스크를 생성하고, 해당 태스크의 승인 버튼이 있는 블록 메시지를 같은 채널에 게시합니다. 담당자가 버튼을 클릭하면 Hermes 가 실행 단계로 전환됩니다. 이 흐름은 사람이 개입하는 지점(human-in-the-loop)을 Slack 채널 안에서 완결하므로 별도의 웹 대시보드 없이도 승인 워크플로를 구성할 수 있습니다.

아래는 Hermes Slack 봇 설정 절차 요약과 주요 스코프입니다.

항목	내용
앱 생성 URL	api.slack.com/apps → "Create New App"
필수 Bot Token Scopes	`chat:write`, `app_mentions:read`, `channels:history`, `commands`
권장 추가 스코프	`files:write`, `reactions:write`, `users:read`
소켓 모드 활성화	App Settings → "Enable Socket Mode" → App-Level Token 생성
Hermes 설정	`slack.bot_token` + `slack.app_token` 두 필드
PoC 선결 조건	워크스페이스 Admin 계정 확보

6.1.3 단일 gateway 프로세스가 6 채널을 통합 라우팅하는 구조

단일 게이트웨이의 구조적 의미

Hermes 의 6 native 채널은 각각 독립된 어댑터 모듈로 구현되어 있지만, 실행 시점에는 하나의 gateway 프로세스에 모두 적재됩니다 [S01]. 각 채널 어댑터는 수신한 메시지를 Hermes 내부 이벤트 버스의 공통 포맷(`ChannelEvent`)으로 변환하여 전달합니다. 이후의 사용자 식별, Profile 라우팅, skill 실행, 응답 발신 과정은 채널 종류와 무관하게 동일한 코드 경로를 따릅니다.

Telegram 으로 들어온 요청이든 Slack 으로 들어온 요청이든, Hermes 내부에서는 출처 채널이 메타데이터 필드 하나로만 구분됩니다.

단일 gateway = 단일 감사 로그 = 단일 권한 모델

이 구조는 운영·보안 측면에서 구체적인 이점을 만듭니다. 모든 채널의 메시지가 같은 지점을 통과하므로 감사 로그가 하나의 타임라인으로 통합됩니다. 채널별로 서로 다른 권한 서버를 운영할 필요가 없으며, Profile 기반 권한 모델 [S03] 이 모든 채널에 균등하게 적용됩니다. 채널 A 에서는 허용되고 채널 B 에서는 차단되는 명령을 정의하는 것도 동일한 권한 설정 파일 한 곳에서 처리합니다.

SSO 연동 시나리오

국내 기업 환경에서는 사내 SSO(Single Sign-On, 단일 인증)와 Hermes gateway 의 연동이 중요한 추가 검토 항목입니다. Hermes gateway 는 채널에서 수신된 사용자 식별자(예: Slack User ID)를 내부 사용자 레코드로 매핑합니다. 이 매핑 테이블을 사내 SSO 의 디렉터리(Active Directory 또는 LDAP)와 주기적으로 동기화하면, 인사 변동(입사·퇴사·부서 이동)이 발생했을 때 Hermes 권한이 자동으로 반영됩니다. PoC 단계에서는 수동 매핑으로 시작하더라도, 본격 운용 전에 SSO 동기화 설계를 완료해두는 것이 안전합니다.

그림 6-1. 6채널 → 단일 Hermes gateway → Profile 라우팅 구조

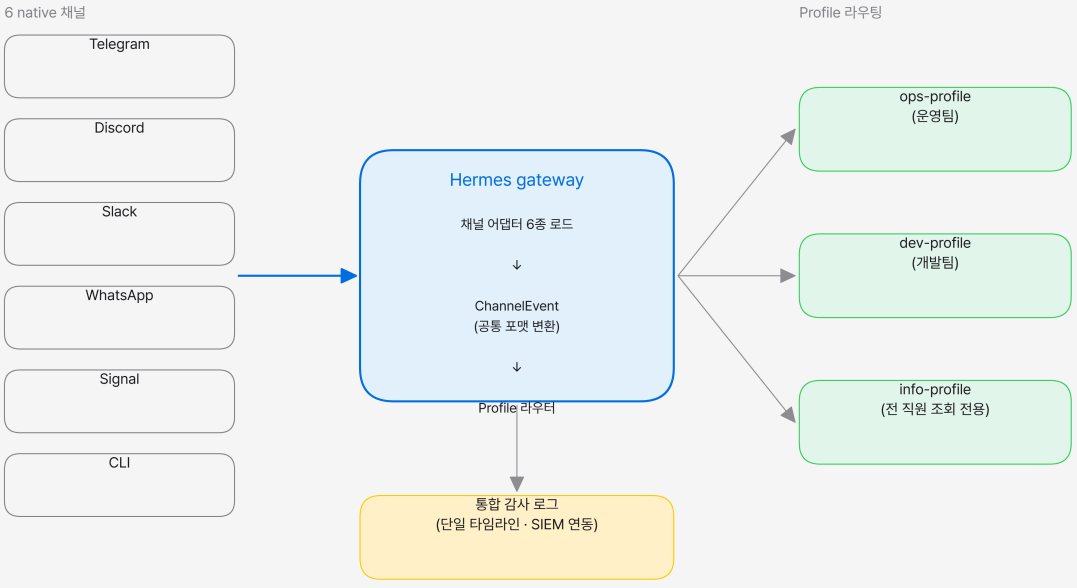


그림 6-1 6개 채널 → 단일 Hermes gateway → Profile 라우팅 구조. 각 채널 어댑터가 공통 ChannelEvent 포맷으로 변환한 뒤 내부 이벤트 버스를 통해 Profile 라우터에 전달되고, 실행 엔진과 감사 로그가 단일 지점에서 처리된다.

6 개 채널 어댑터는 공통 인터페이스를 구현하므로, 7번째 채널(예: 사내 메신저)을 추가할 때도 새 어댑터를 작성해 gateway 의 채널 레지스트리에 등록하는 것으로 충분합니다. 기존 채널이나 권한 모델에는 변경이 없습니다.

6.2 국내 사내 메신저 어댑터 작성 가이드

국내 기업의 IT 보안 정책은 외부 메신저 서버와의 통신을 차단하는 경우가 많습니다. Telegram · WhatsApp · Signal 은 외부 클라우드 서버를 경유하므로 내부망에서 사용이 금지되는 경우가 대부분이고, 심지어 Slack 도 미국 서버에 대화 내용이 저장된다는 이유로 도입 심의에서 제외되는 사례가 있습니다. 이 경우 잔디(JANDI) · 카카오워크(Kakao Work) · NHN Dooray 같은 국내 사내 메신저가 유일한 선택지가 됩니다. Hermes 의 어댑터 인터페이스는 이 시나리오를 위해 설계되어 있습니다. 표준 인터페이스를 구현한 어댑터 클래스 하나를 작성하면 기존 6 native 채널과 동일한 방식으로 사내 메신저가 Hermes gateway 에 합류합니다. MSAP.ai 와 같은 국내 통합 플랫폼의 경우 잔디·카카오워크 어댑터를 사전 제공하여 어댑터 작성 공수를 절감할 수 있습니다.

6.2.1 채널 어댑터 인터페이스 (수신 · 발신 · 사용자 식별)

어댑터가 구현해야 할 3개 인터페이스

Hermes gateway 가 임의의 메신저를 채널로 인식하려면 어댑터가 다음 3개 메서드를 구현해야 합니다. 첫째, `receive()` — 사내 메신저에서 메시지가 도착했을 때 이를 Hermes 내부 `ChannelEvent` 로 변환하여 이벤트 버스에 투입합니다. 둘째, `send()` — Hermes 실행 엔진이 생성한 응답 문자열을 사내 메신저 API 를 통해 특정 채널·사용자에게 발신합니다. 셋째, `resolve_user()` — 사내 메신저가 전달하는 사용자 식별자(예: 메신저 내부 user_id)를 Hermes 의 내부 사용자 레코드로 변환합니다. 이 세 메서드만 구현하면 나머지 실행 경로는 Hermes 가 처리합니다 [S01].

어댑터 기본 구조 (Python pseudo-code)

아래는 사내 메신저 어댑터의 표준 구현 구조를 보여주는 Python 의사 코드입니다. 실제 사내 메신저 SDK 호출 부분만 채워 넣으면 동작하는 어댑터가 완성됩니다.

```
from hermes.gateway.adapter import ChannelAdapter, ChannelEvent

class CompanyMessengerAdapter(ChannelAdapter):
    """사내 메신저 어댑터 — 3개 필수 메서드 구현 패턴"""

    def __init__(self, config: dict):
        self.api_token = config["api_token"]
        self.webhook_secret = config["webhook_secret"]
        # 사내 메신저 SDK 클라이언트 초기화
        self.client = MessengerSDKClient(token=self.api_token)

    def receive(self, raw_payload: dict) -> ChannelEvent:
        """
        수신: 사내 메신저 웹훅 페이로드 → Hermes ChannelEvent
        """
        return ChannelEvent(
            channel_id="company-messenger",
            raw_user_id=raw_payload["sender_id"],
            text=raw_payload["text"],
```

```
            room_id=raw_payload["room_id"],
            timestamp=raw_payload["timestamp"],
        )

    def send(self, room_id: str, message: str) -> None:
        """
        발신: Hermes 응답 → 사내 메신저 채팅방
        """
        self.client.send_message(room=room_id, text=message)

    def resolve_user(self, raw_user_id: str) -> str:
        """
        사용자 식별: 메신저 user_id → Hermes 내부 사용자 ID (이메일 기반)
        """
        profile = self.client.get_user_profile(user_id=raw_user_id)
        return profile["email"]  # SSO 이메일을 공통 키로 사용
```

구현 공수와 주의 사항

이 인터페이스를 처음 구현하는 경우 1명이 전담하면 약 2~3주가 소요됩니다. 공수의 대부분은 메서드 로직 자체가 아니라 사내 메신저 API 문서 파악, 테스트 계정 발급, 내부망 방화벽 허용 규칙 적용에 사용됩니다. 어댑터 작성 공수를 단축하려면 PoC 0주차에 사내 메신저 API 사용 가능 여부(공개 API 존재 여부, 내부 승인 절차)를 먼저 확인하는 것이 가장 효과적입니다.

아래는 어댑터 인터페이스 명세를 정리한 표입니다.

메서드	입력	출력	역할
`receive(raw_payload)`	메신저 웹훅 페이로드 (dict)	`ChannelEvent`	외부 메시지 → 내부 이벤트 변환
`send(room_id, message)`	채팅방 ID + 응답 문자열	없음 (side effect)	내부 응답 → 외부 발신
`resolve_user(raw_user_id)`	메신저 내부 user_id	내부 사용자 ID (문자열)	사용자 정합화

6.2.2 잔디 · 카카오워크 · NHN Dooray 어댑터 작성 사례

사내 메신저 선택의 제약 조건

국내 3대 사내 메신저(잔디 · 카카오워크 · NHN Dooray) 중 어느 것을 Hermes 어댑터로 연결할지는 대부분 회사가 이미 결정된 상태입니다. 사내 메신저는 전사 표준 도구로 채택되는 경우가 많아 IT 담당자의 선택 여지가 좁습니다. 따라서 어댑터 선택이 아니라 각 메신저의 공개 API 특성을 파악하고 어댑터 작성 시 발생할 수 있는 주의 사항을 미리 확인하는 것이 중요합니다. 세 메신저 모두 웹훅 기반으로 메시지를 수신할 수 있지만, 인증 방식과 rate limit 정책에 차이가 있습니다.

잔디(JANDI) 어댑터 특성

잔디는 Incoming Webhook 과 Outgoing Webhook 두 가지를 제공합니다. Incoming Webhook 은 외부 시스템이 잔디 채널로 메시지를 밀어 넣는 용도이며, Outgoing Webhook 은 잔디 채널에서 특정 키워드가 포함된 메시지가 발생했을 때 외부 서버로 페이로드를 전송하는 용도입니다 [S01]. Hermes 어댑터는 Outgoing Webhook 으로 메시지를 수신(`receive`)하고, Incoming Webhook URL 로 응답을 발신(`send`)하는 구조를 사용합니다. 잔디의 Outgoing Webhook 은 OAuth 없이 단순 HTTP POST 로 동작하므로 인증 설정 부담이 낮습니다. 단, 잔디 커넥트(JANDI Connect)의 Outgoing Webhook 은 키워드 매칭 방식이기 때문에 Hermes 를 호출하는 트리거 키워드(`@hermes` 등)를 워크스페이스 설정에서 명시적으로 등록해야 합니다.

카카오워크(Kakao Work) 어댑터 특성

카카오워크는 카카오 i 기술문서(docs.kakaoi.ai)에 공식 Bot 개발 가이드를 제공합니다. 인증 방식은 앱 키(App Key) 기반의 Bearer 토큰 방식을 사용하며, OAuth 2.0 흐름보다 단순합니다. Bot 생성 시 자동 발급되는 App Key 를 `Authorization: Bearer {APP_KEY}` 헤더에 담아 API 를 호출합니다. 메시지 수신은 콜백 URL 등록으로 처리합니다. 카카오워크 API 는 REST 가 아닌 RPC(Remote Procedure Call, 원격 프로시저 호출) 스타일이므로 URL 경로가 리소스 중심이 아닌 기능 단위로 구성되어 있습니다. 이 점을 인식하고 어댑터의 `send()` 구현 시 API 레퍼런스를 꼼꼼히 확인해야 합니다. 사용자 식별을 위한 `resolve_user()` 에서는 카카오워크 Bot API 의 사용자 조회 엔드포인트를 통해 이메일 또는 사번을 추출합니다.

NHN Dooray 어댑터 특성

NHN Dooray 는 메신저 외에 프로젝트 관리 · 이메일 · 전자 결재를 통합한 플랫폼입니다. Bot 연동은 개인 인증 토큰(Personal Authentication Token)을 발급받아 사용하며, App Token 기반으로도 동작합니다. Dooray 는 slash command(슬래시 커맨드) 서버를 별도로 구성해야 하며, Hermes gateway 에 slash command 처리 엔드포인트를 노출하는 방식으로 `receive()` 를 구현합니다. Dooray 의 slash command 는 사용자가 `/hermes <명령>` 형태로 입력하면 Hermes 엔드포인트로 HTTP POST 를 발송하는 구조입니다. Dooray 어댑터를 작성할 때는 사내 네트워크에서 Dooray 서버가 Hermes gateway 의 엔드포인트로 콜백을 보낼 수 있는지 방화벽 설정을 PoC 0 주차에 반드시 확인해야 합니다.

아래는 3개 사내 메신저의 API 특성 비교표입니다.

항목	잔디 (JANDI)	카카오워크	NHN Dooray
메시지 수신 방식	Outgoing Webhook (키워드 트리거)	콜백 URL 등록	Slash command 서버
메시지 발신 방식	Incoming Webhook URL POST	Web API (RPC 스타일)	Web API
인증 방식	Webhook URL 공유 (비밀키 X)	Bearer App Key	Personal Token / App Token
OAuth 여부	없음	없음 (Bearer 토큰)	없음 (토큰 발급 방식)

항목	잔디 (JANDI)	카카오워크	NHN Dooray
rate limit	별도 공개 없음	별도 공개 없음	별도 공개 없음
PoC 0주차 확인 사항	키워드 트리거 등록 권한	Bot 생성 권한 (워크스페이스 관리자)	방화벽 인바운드 허용 여부

6.2.3 이메일 · SMS 보강 채널의 추가 가치

보강 채널이 필요한 이유

메신저 기반 채널만으로는 조직 전체를 커버하기 어렵습니다. 실무 담당자는 메신저 알림에 즉각 반응하지만, 팀장·임원 계층은 메신저 알림보다 이메일을 더 신뢰하는 경향이 있습니다. 의사결정 요청이나 예외 승인처럼 메신저 맥락 밖에서 판단이 이루어지는 경우, Hermes 가 이메일로 요약 보고서를 발송하는 것이 더 효과적입니다. SMS 는 메신저와 이메일 모두 확인하기 어려운 외근·이동 중 상황에서 중요 알림의 보조 수단으로 활용됩니다.

이메일 채널 구성

Hermes 는 SMTP(Simple Mail Transfer Protocol) 설정을 통해 이메일 발신 채널을 추가할 수 있습니다. 사내 이메일 게이트웨이에 연결하거나 SendGrid · AWS SES 같은 외부 이메일 서비스를 연결하는 두 방식이 있습니다. 이메일 채널은 주로 발신 전용으로 운용합니다. Hermes 가 특정 조건(예: 위험 수준 높은 명령 실행, 일일 업무 요약 보고)에서 자동으로 이메일을 발송하도록 Profile 의 trigger 조건에 채널 발신 규칙을 추가합니다 [S01].

역할별 채널 선호도와 트레이드오프

아래는 역할별 채널 선호도와 운영 비용 트레이드오프를 정리한 매트릭스입니다. 이 매트릭스를 근거로 어떤 채널 조합을 PoC 범위에 포함할지 결정하고, 정보보호 부서와 채널별 데이터 처리 방침을 합의하는 데 활용할 수 있습니다.

역할	1순위 채널	2순위 채널	도달률	응답 지연	운영 비용
실무 담당자	사내 메신저	Slack	높음	즉시	낮음
팀장	사내 메신저	이메일	높음	수분~수십분	낮음
임원	이메일	SMS	보통	수십분~수시간	중간
외부 협력사	이메일	—	보통	수시간	낮음
비상 연락	SMS	이메일	높음	즉시~수분	높음

이메일과 SMS 채널은 추가 비용이 발생합니다. 이메일은 사내 이메일 게이트웨이 활용 시 사실상 무료지만, 외부 이메일 서비스 사용 시 발신 건당 과금이 발생합니다. SMS 는 발신 건당 과금(국내 기준 약 10~20원/건)이므로 알림 빈도와 수신 인원을 조건별로 제한하는 설계가 필요합니다.

6.3 채널별 권한·감사 정책 설계

채널 통합은 편의성을 제공하지만 동시에 보안 위험 노출면을 넓힙니다. 채널이 늘어날수록 "누가, 어떤 채널을 통해, 어떤 명령을 실행했는가"를 추적하기 어려워집니다. Hermes 의 단일 gateway 구조는 이 문제를 구조적으로 해소합니다. 모든 채널의 접근 통제와 감사 기록이 gateway 단일 지점에서 처리되므로, 채널이 몇 개든 권한 정책 파일과 감사 로그 저장소는 하나입니다. 이 절은 채널별 권한 매트릭스 설계, 통합 감사 로그 구성, 채널 간 사용자 식별 정합화의 세 부분으로 구성됩니다.

6.3.1 채널 × 권한 매트릭스 — 어디서 무엇을 시킬 수 있는가

채널 권한 분리의 보안 의미

같은 Hermes 인스턴스에 연결된 모든 채널이 동일한 명령 집합을 실행할 수 있도록 두면 위험합니다. Telegram 은 개인 PoC 채널로 사용하면서 프로덕션 배포 명령이 실행될 수 있다면 채널이 늘어날수록 공격 노출면이 넓어집니다. Hermes 는 Profile 기반 권한 모델 [S03] 을 통해 채널별로 실행 가능한 skill 과 tool 을 분리합니다. 채널 권한 매트릭스는 정보보호 부서의 동의를 받을 때 핵심 산출물이 됩니다. "채널 A 에서는 조회만 가능하고 채널 B 에서만 실행이 허용된다"는 명확한 경계를 문서로 제시해야 승인 절차가 진행됩니다.

Profile 과 채널의 조합 방식

Hermes 의 Profile [S03] 은 메모리 · skill · tool 묶음으로 정의된 Hermes 인스턴스 단위입니다. 채널 권한은 Profile 에 연결된 skill 목록을 채널별로 추가 제한하는 방식으로 구현됩니다. 예를 들어, `ops-profile` 이라는 Profile 이 배포 · 롤백 · 로그 조회 skill 을 포함하더라도, Slack 채널에서는 조회 skill 만 허용하고 배포 · 롤백 skill 은 CLI 채널에서만 실행 가능하도록 제한할 수 있습니다. 이렇게 구성하면 메신저를 통한 실수 실행이나 사회공학적 공격으로 인한 오남용을 줄입니다.

채널 × Profile × 허용 skill 매트릭스 예시

아래는 사내 배포 자동화 시나리오에서 설계할 수 있는 채널별 권한 매트릭스 예시입니다. 각 팀의 역할과 채널 특성에 따라 허용 범위를 조정합니다.

채널	허용 Profile	허용 skill (예시)	차단 skill (예시)
Slack (`#ops`)	ops-profile	`log:query` , `deploy:status` , `kanban:view`	`deploy:run` , `rollback` , `db:migrate`
Slack (`#general`)	info-profile	`faq:answer` , `schedule:query`	운영 관련 전체
사내 메신저 (잔디)	info-profile, ops-profile	`log:query` , `deploy:status` , `faq:answer`	`deploy:run` , `rollback`
CLI (서버 직접)	admin-profile	전체 skill	없음 (관리자 전용)
Telegram (PoC)	dev-profile	`test:run` , `skill:debug`	프로덕션 관련 전체

분기별로 이 매트릭스를 재검토하고, 새로운 skill 이 추가될 때마다 채널별 허용 여부를 명시적으로 결정하는 절차를 수립하면 권한 범위가 조용히 확대되는 문제를 예방합니다.

6.3.2 채널 통합 감사 로그 — 단일 timeline 으로 정합화

단일 타임라인의 실질적 가치

보안 감사나 장애 분석 상황에서 "어느 채널에서 어떤 사용자가 어떤 명령을 실행했는가"를 파악하는 데 걸리는 시간이 대응 속도를 결정합니다. 채널별로 별도의 로그가 흩어져 있으면 여러 시스템을 순서대로 조회해야 하지만, Hermes 의 단일 gateway 구조는 모든 채널의 실행 기록이 하나의 로그 스트림으로 합쳐집니다 [S03]. 이 통합 감사 로그는 "단일 진실의 원천"으로 기능하며, 채널 A 의 조회와 채널 B 의 실행이 같은 타임라인 위에 놓이므로 연관 분석이 가능해집니다.

감사 로그 스키마

Hermes 의 통합 감사 로그는 다음 필드로 구성됩니다. 이 스키마를 기준으로 SIEM(Security Information and Event Management, 보안 정보 및 이벤트 관리) 연동 시 파싱 규칙을 작성합니다.

```
{
  "timestamp": "2026-06-29T14:32:01.123Z",
  "channel": "slack",
  "room_id": "C0123ABCDEF",
  "user_internal_id": "user@company.com",
  "user_channel_id": "U0123ABCDEF",
  "profile": "ops-profile",
  "action": "skill:invoke",
  "skill": "log:query",
  "input_summary": "서비스 A 오류 로그 최근 1시간",
  "result": "success",
  "latency_ms": 1243,
  "request_id": "req_a1b2c3d4"
}
```

SIEM 연동과 보존 정책

이 감사 로그를 Elasticsearch · Splunk · IBM QRadar 같은 SIEM 플랫폼으로 전송하면 채널 통합 감사 추적이 완성됩니다. Hermes gateway 는 로그를 표준 JSON Lines 형식으로 출력하므로, Filebeat · Fluentd 같은 로그 수집기가 별도 가공 없이 바로 수집할 수 있습니다. 보존 기간은 국내「개인정보보호법」및 사내 정보보호 정책에 따라 결정하되, 최소 90일 이상 보존하는 것이 일반적인 보안 감사 요건입니다. 감사 로그에 포함된 `input_summary` 필드는 전체 사용자 입력이 아닌 요약 문자열로 저장하도록 설계하여, 민감 정보가 감사 로그에 그대로 기록되는 것을 방지합니다.

6.3.3 채널 간 사용자 식별 정합화 (SSO · OAuth · email)

사용자 정합화 실패의 결과

채널이 늘어날수록 같은 사람이 서로 다른 식별자를 가집니다. Slack 에서는 `U0123ABCDEF`, 잔디에서는 `jandi_12345`, 이메일로는 `user@company.com`, 사내 시스템에서는 `EMP00789` 라는 사번을 씁니다. 이 네 가지 식별자가 같은 사람임을 Hermes 가 알지 못하면 권한 모델 전체가 붕괴됩니다. Slack 에서 조회 권한만 부여된 사용자가 잔디에서는 별도 매핑이 없어 admin-profile 로 처리되는 오류가 발생할 수 있습니다 [S01]. 사용자 정합화는 채널 통합의 보안 기반이며, PoC 단계 산출물에 반드시 포함되어야 합니다.

정합화 전략과 공통 키 선택

가장 안정적인 정합화 방법은 사내 이메일 주소를 공통 키로 사용하는 것입니다. 이메일은 회사 입사 시 발급되고 퇴사 시 비활성화되며, 대부분의 사내 메신저와 SSO 시스템이 이메일을 기본 식별자로 지원합니다. 어댑터의 `resolve_user()` 메서드는 각 채널의 사용자 식별자를 이메일 주소로 변환하는 역할을 담당합니다. SSO(Single Sign-On) 또는 LDAP(Lightweight Directory Access Protocol, 경량 디렉터리 액세스 프로토콜) 디렉터리를 정합화 원천으로 사용하면, 인사 변동이 SSO 에만 반영되어도 Hermes 권한이 자동으로 갱신됩니다 [S03].

사용자 식별 매핑표 설계

아래는 사용자 식별 정합화 매핑 레코드의 구조 예시입니다. 이 테이블은 Hermes 내부 사용자 데이터베이스에 저장되며, 채널 어댑터의 `resolve_user()` 가 조회하는 원천입니다.

내부 키	Slack ID	잔디 user_id	카카오워크 user_id	이메일	사번
`user@company.com`	`U0123ABCDEF`	`jandi_12345`	`kw_9876`	`user@company.com`	`EMP00789`

이 매핑 테이블을 사내 SSO 디렉터리와 주기적으로 동기화하는 스크립트를 구성하면 수동 관리 부담을 없앱니다. SCIM(System for Cross-domain Identity Management, 크로스 도메인 신원 관리 시스템)을 지원하는 SSO 솔루션의 경우, SCIM 프로비저닝 이벤트를 Hermes 사용자 데이터베이스에 연동하여 실시간 동기화가 가능합니다. 퇴사자 처리는 SSO 계정 비활성화와 동시에 Hermes 매핑 레코드도 비활성화(`active: false`)하여 모든 채널에서 일괄 차단되도록 설계합니다.

채널 간 사용자 식별 정합화는 기술 구현보다 조직 프로세스 설계가 더 중요합니다. 신규 직원 온보딩 시 Hermes 매핑 레코드 생성을 HR 시스템 연계 절차에 포함하고, 채널 추가 시마다 신규 채널의 user_id 필드를 매핑 테이블에 추가하는 변경 관리 절차를 수립해야 합니다.

7장. Local LLM 과 Hermes Agent — 모델 선정 가이드

Hermes Agent 는 어떤 LLM 위에서도 동작합니다. OpenAI API 를 쓸 수도 있고, 사내 GPU 서버에 올린 오픈소스 모델을 직접 호출할 수도 있습니다. 이 선택이 단순한 기술 취향의 문제처럼 보일 수 있지만, 실제로는 보안 정책 준수 여부·운영 예산·응답 속도라는 세 가지 핵심 운영 지표를 동시에 결정짓는 의사결정입니다. 국내 기업이 AI 에이전트를 실무에 도입할 때 "외부 클라우

드 LLM 을 그대로 쓸 것인가, 아니면 사내에 모델을 직접 올릴 것인가"라는 질문이 반드시 등장하는 이유가 여기에 있습니다.

7장은 이 질문에 답하기 위한 실무 가이드입니다. 먼저 Local LLM(사내 배포 언어 모델)을 선택하는 세 가지 근거인 보안·비용·지연 시간을 정량 자료와 함께 정리하고, 이어서 현재 도입 가능한 대표 오픈소스 모델 세 종의 VRAM(Video RAM, 그래픽 처리 장치 전용 메모리) 요구량·한국어 지원 수준·컨텍스트 창 크기·멀티모달 역량을 5가지 기준으로 평가하는 의사결정 트리를 제시합니다. 2026년 4월 Google 이 Gemma 4 31B Dense 를 Apache 2.0 라이선스로 공개하면서 7장의 의사결정 트리는 한 차례 구조적으로 재편되었습니다. 본 트리는 사내 모델 선정 회의에서 그대로 체크리스트로 활용할 수 있도록 설계했습니다.

7.1 sLLM · Local LLM 을 선택하는 이유 — 보안·비용·지연

sLLM(소형 언어 모델, Small Language Model)과 Local LLM 은 엄밀히 다른 개념이지만 도입 맥락에서는 대부분 중첩됩니다. sLLM 은 파라미터 수를 수십억(B) 규모로 줄여 단일 GPU 서버에서 운영 가능하게 설계한 모델을 가리키고, Local LLM 은 외부 클라우드가 아닌 자체 인프라에 배포한 모델을 통칭합니다. 국내 기업이 Hermes Agent 와 결합해 도입하는 경우 대부분 "사내 GPU 서버에 올린 소형 오픈소스 모델"이라는 형태이므로, 이 장에서는 두 개념을 구분 없이 "Local LLM"으로 통일해 사용합니다.

Local LLM 을 선택하는 이유는 크게 세 가지입니다. 첫째, 프롬프트와 응답이 외부 서버로 나가지 않으므로 개인정보·영업기밀 유출 경로 자체가 차단됩니다. 둘째, 월간 토큰 사용량이 일정 임계점을 넘으면 외부 API 누적 요금보다 GPU 인프라 감가상각이 저렴해집니다. 셋째, 사내 네트워크 안에서 추론하므로 외부 API 왕복 지연이 사라져 실시간 워크플로에서 체감 속도가 눈에 띄게 달라집니다.

7.1.1 보안 — 데이터 외부 송신 0 의 정량 가치

외부 SaaS LLM 의 데이터 흐름과 노출 지점

외부 SaaS LLM(서비스형 대형 언어 모델)에 프롬프트를 보내는 순간, 텍스트는 조직 경계를 벗어나 인터넷을 경유해 외부 데이터센터까지 전달됩니다. 이 구간에서 발생하는 위험은 단순히 "전송 중 도청" 수준이 아닙니다. 사용자가 입력한 프롬프트 자체에 고객 이름·주민등록번호·계약 금액·내부 시스템 접속 정보가 포함되는 경우가 드물지 않습니다. 국내 정보보호 담당자 조사에서 업무용 AI 도구 프롬프트의 상당 비율에 개인식별정보(PII)가 포함된다는 사실이 반복적으로 확인됩니다 [S12]. 각 SaaS LLM 제공사의 데이터 처리 약관은 "모델 학습에 사용하지 않는다"는 조항을 포함하는 경우가 많지만, "인프라 운영·보안 목적 처리"는 별도 조항으로 허용하는 경우가 있어 법무 검토 없이 그대로 신뢰하기 어렵습니다.

Local LLM 이 차단하는 데이터 이동 단계

Local LLM 을 사내 GPU 서버에 배포하면 프롬프트는 사내 네트워크 안에서만 이동합니다. 요청이 Hermes Agent 프로세스 → 사내 LLM 추론 서버 → 응답 반환의 순서로 처리되며, 어느 단계에서도 인터넷 구간을 경유하지 않습니다. 이 구조는 망분리 환경에서도 그대로 적용 가능합니다 [S12]. 국내 도입 사례에서도 방대한 내부 데이터를 처리하면서 데이터 외부 송신 금지 요건

과 높은 보안 등급을 동시에 충족해야 하는 환경에서 sLLM 자체 구축이 유일한 현실적 선택지로 채택되었습니다 [S12]. 같은 구조를 Hermes Agent 와 결합하면 에이전트 오케스트레이션 계층도 외부 종속 없이 사내에서 완결됩니다.

정보보호 부서의 의사결정 단일 기준

정보보호 부서가 AI 에이전트 도입을 검토할 때 가장 먼저 묻는 질문은 "이 도구를 사용할 때 데이터가 외부로 나가는가"입니다. 이 질문에 "아니오"라고 답할 수 있는 구조가 Local LLM 입니다. 사내 정보보호 정책 문서의 "외부 클라우드 서비스 제공자 허용 명단"과 LLM SaaS 제공사 명단을 교차 점검해 보면, 대부분의 국내 기업에서 주요 LLM SaaS 는 해당 명단에 포함되어 있지 않습니다. Local LLM 은 이 충돌을 구조적으로 해소합니다.

아래 표는 외부 SaaS LLM 과 Local LLM 의 데이터 이동 단계별 노출 위험을 비교한 것입니다.

단계	외부 SaaS LLM	Local LLM (사내 배포)
프롬프트 전송 구간	인터넷 경유 (HTTPS)	사내 네트워크 내부
데이터 저장	외부 데이터센터 (지역 불확실)	사내 스토리지
모델 학습 재사용	제공사 약관에 따라 상이	해당 없음
망분리 환경 적용	불가	가능 (오프라인 설치 절차 필요)
PII 노출 경로	전송·저장 두 구간 모두 존재	없음
정보보호 정책 준수	별도 법무 검토 필요	자체 인프라 통제 범위 내

7.1.2 비용 — 토큰 단가 vs GPU 감가상각의 손익분기점

SaaS LLM 토큰 단가의 누적 구조

외부 SaaS LLM 요금은 입력 토큰(Input Token)과 출력 토큰(Output Token)의 합산 수량에 단가를 곱하는 방식으로 청구됩니다. 2026년 현재 주요 상용 LLM API 의 단가는 입력 100만 토큰당 2~15달러, 출력 100만 토큰당 6~60달러 수준으로 모델 등급에 따라 큰 폭으로 달라집니다. 에이전트 오케스트레이션 시나리오에서는 단일 사용자 요청 하나가 Hermes Agent 의 플래닝·도구 호출·결과 요약 등 여러 LLM 호출로 분해되므로, 실제 토큰 소비량은 단순 채팅보다 3~10배 이상 증가합니다. 월 100만 건의 에이전트 작업을 처리하는 중규모 기업이 외부 SaaS LLM 을 쓸 경우 월 수천만 원대 API 비용이 발생하는 것이 일반적입니다.

GPU 인프라 감가상각과 손익분기점

Local LLM 의 비용 구조는 반대입니다. 초기에 GPU 서버 구매 비용이 집중되고, 이후에는 전력·냉각·유지보수 정도만 지속 지출됩니다. 2026년 기준 RTX 4090 24GB 단일 카드를 탑재한 워크스테이션 구성 비용은 약 1,000~1,500만 원 수준이며, H100 80GB SXM 급 고사양 서버는 대당 4,000만 원 이상입니다. 전문 클라우드 업체 분석에 따르면 월 API 지출이 일정 수준을 초과하면 온프레미스(On-Premises, 자체 인프라) 서버가 4~8주 만에 손익분기점에 도달한다는 보고도 있습니다. 보수적으로 잡아도 6~18개월 내 투자비가 회수되고, 그 이후에는 추가 토큰 비용 없이 운영 가능하다는 구조적 장점이 있습니다 [S07, S08, S09].

시나리오별 비용 비교 기준

손익분기점은 세 변수에 따라 달라집니다. 첫째는 월간 토큰 사용량입니다. 사용량이 많을수록 SaaS 누적 비용이 빠르게 증가하므로 Local LLM 전환 이득이 커집니다. 둘째는 모델 크기와 이에 따른 GPU 등급입니다. gpt-oss 20B 처럼 16GB 메모리로 동작하는 모델은 저렴한 소비자용 GPU 로도 운영할 수 있어 초기 투자비가 낮습니다 [S09]. 셋째는 가동률입니다. GPU 서버가 하루 8시간만 사용된다면 유휴 시간의 감가상각 비효율이 발생합니다. Hermes Agent 처럼 비동기 배치 스케줄링이 가능한 오케스트레이터를 쓰면 야간·주말 작업을 분산 처리해 GPU 가동률을 높이고 단위 작업당 실질 비용을 낮출 수 있습니다. 아래 표는 주요 변수에 따른 비용 유리 구조를 정리한 것입니다.

기준	SaaS LLM 유리	Local LLM 유리
월간 토큰 사용량	10M 토큰 미만	30M 토큰 초과
GPU 투자 예산	초기 투자 불가 시	12개월 이상 운영 계획 시
트래픽 패턴	일시적·예측 불가 급증	안정적·예측 가능한 정기 부하
요구 모델 크기	100B 이상 (고사양 필요)	30B 이하 (단일 GPU 가능)
컴플라이언스 부담	법무 검토로 허용 가능 시	데이터 외부 송신 금지 정책 시

7.1.3 지연 — 사내 네트워크 왕복 시간의 운영적 의미

SaaS LLM 의 지연 구조와 실측 수치

SaaS LLM 을 호출할 때 응답 지연은 두 구간에서 발생합니다. 첫 번째는 클라이언트에서 외부 API 서버까지의 네트워크 왕복 구간이고, 두 번째는 API 서버에서 모델 추론이 완료되어 첫 토큰을 반환하는 구간(TTFT, Time-To-First-Token)입니다. 외부 클라우드 환경에서 TTFT 는 일반적으로 150~300ms 수준이며, 피크 트래픽 시간대에는 1~2초로 치솟는 경우도 보고됩니다. 에이전트 워크플로에서는 LLM 호출이 순차적으로 여러 번 발생하므로, 호출당 300ms 지연이 3번 중첩되면 사용자 체감 지연은 1초에 가까워집니다.

Local LLM 의 지연 특성

사내 GPU 서버에서 동작하는 Local LLM 은 네트워크 왕복 구간이 로컬 네트워크 수준으로 줄어듭니다. vLLM(고성능 LLM 추론 프레임워크) 기반 배포 환경에서 70B 급 모델의 P99 지연(상위 1% 최악 케이스 지연)이 80ms 내외로 유지된다는 측정 결과가 있습니다. 동시 요청 8건 기준 초당 처리 토큰 수는 vLLM 환경에서 187토큰/초, 같은 서버에서 Ollama(로컬 LLM 실행 도구)로 전환하면 82토큰/초로 줄어든다는 비교도 보고됩니다. 추론 프레임워크 선택이 지연 특성을 상당 부분 결정하므로, Hermes Agent 와 결합할 때 vLLM 또는 TensorRT-LLM(NVIDIA 고성능 추론 최적화 라이브러리) 을 고려하는 것이 합리적입니다 [S07].

지연 시간 구간별 시나리오 적합도

지연 시간은 시나리오 유형에 따라 허용 범위가 달라집니다. 실시간 고객 응대나 알림 생성처럼 사람이 화면 앞에서 기다리는 시나리오는 50ms 이하를 목표로 해야 합니다. 문서 요약·데이터

변환처럼 배치로 처리하는 시나리오는 200ms 이상이 되어도 무관합니다. Hermes Agent 는 Kanban 기반 태스크 스케줄링으로 긴급 태스크와 배치 태스크를 분리해 라우팅하므로, 모델별 지연 특성을 라우팅 정책에 반영하는 방식으로 지연 요건을 충족할 수 있습니다.

지연 구간	해당 시나리오	추천 배포 방식
≤ 50ms	실시간 CS 응대, 실시간 알림 생성	사내 GPU 서버 (vLLM 추론)
50ms ~ 200ms	개인화 리포트 생성, 내부 Q&A 봇	사내 GPU 서버 또는 하이브리드
≥ 200ms	대용량 문서 배치 요약, 야간 데이터 분석	SaaS LLM 또는 배치 큐 처리

7.2 모델 선정 의사결정 트리 — 라이선스·VRAM·한국어·컨텍스트

2026년 현재 Hermes Agent 와 결합해 즉시 운영 가능한 대표 오픈소스 모델은 세 종입니다. Google 의 Gemma 4 31B Dense, Alibaba 의 Qwen3 시리즈, 그리고 OpenAI 가 2025년 8월 공개한 gpt-oss 20B 입니다. 2026년 4월 2일 Google 이 Gemma 4 라인업을 발표하면서 본 절의 의사결정 구조에 큰 변화가 생겼습니다. 핵심 변화는 라이선스 항목입니다. 종전 Gemma 3 시리즈는 Google 이 별도로 제정한 Gemma Terms of Use 를 따랐으나, Gemma 4 부터는 Apache 2.0 라이선스로 통일되었습니다. 그 결과 Gemma 4 · Qwen3 · gpt-oss 20B 세 모델 모두 Apache 2.0 으로 정렬되어, "라이선스" 항목이 더 이상 후보를 탈락시키는 1차 필터로 작동하지 않습니다.

모델 선정 회의에서 흔히 발생하는 문제는 "어떤 모델이 더 좋은가"를 벤치마크 순위로만 판단하려는 경향입니다. 벤치마크 순위는 영어 중심 태스크 기준인 경우가 많고, 조직의 실제 제약(예산·법무·인프라)을 반영하지 않습니다. 실무에서 유효한 선정 방법은 조직이 통과해야 하는 제약 기준을 순서대로 나열하고 각 기준에서 후보를 필터링하는 것입니다. Gemma 4 의 라이선스 변경 이후 본 백서가 권고하는 5가지 기준 순서는 VRAM → 한국어 → 컨텍스트 → 멀티모달 → 사이즈 라인업 확장성입니다. 라이선스는 5 기준 어디에도 분기 항목으로 등장하지 않으며, 7.2.1 에서 단일 표로 점검한 뒤 다음 절부터는 전제 조건으로 다룹니다.

7.2.1 라이선스 — Apache 2.0 (Qwen3 · gpt-oss · Gemma 4)

라이선스가 더 이상 1차 필터가 아닌 이유

종전 Gemma 3 27B 시점의 모델 선정 트리는 라이선스를 첫 번째 분기로 두었습니다. Apache 2.0 인 Qwen3 · gpt-oss 20B 와 Gemma Terms of Use 인 Gemma 3 27B 사이의 법무 검토 부담 격차가 PoC 시작 속도를 결정했기 때문입니다. 그러나 2026년 4월 2일 Google 이 Gemma 4 라인업을 Apache 2.0 으로 공개하면서 세 모델 모두 동일 라이선스로 정렬되었습니다. 그 결과 라이선스는 후보를 탈락시키는 필터에서 단순 확인 항목으로 위상이 바뀌었습니다.

Apache 2.0 의 실용적 의미

Apache 2.0 은 저작권 표시(attribution)와 변경 사항 고지 의무만 부과합니다. 상업적 이용·수정·재배포·파인튜닝(Fine-Tuning, 특정 도메인 데이터로 추가 학습) 후 재배포 모두 자유롭게 허용됩니다 [S08, S09]. Apache License 2.0 원문 기준 특허 grant 조항도 포함되어 있어, 모델 사용으로 인한 특허 분쟁 리스크도 함께 차단됩니다. 국내 기업이 Gemma 4 · Qwen3 · gpt-oss 20B 중 어느 모델을 선택하든 법무팀의 별도 라이선스 실사(Due Diligence) 없이 PoC 가 가능합니다.

Gemma 4 라이선스 변경의 영향 범위

Gemma 4 31B Dense 의 Hugging Face 모델 카드는 라이선스 필드에 `apache-2.0` 을 명시합니다. 구 Gemma 3 27B 가 따르던 Gemma Terms of Use 의 "responsible commercial use" 조항·사용 사례별 제한 목록은 Gemma 4 부터 폐기되었습니다. 단, 2026년 4월 이전에 Gemma 3 27B 기반 PoC 를 진행한 조직이라면 마이그레이션 시점까지는 구 약관이 잔존 적용되므로, Gemma 4 신규 다운로드 시점부터 Apache 2.0 으로 전환된다는 점만 사내 법무에 1줄 공유하면 충분합니다. 본 백서의 모든 후속 절은 Gemma 4 31B Dense 가 Apache 2.0 임을 전제로 작성되었습니다.

아래 표는 세 모델의 라이선스 조건을 핵심 항목으로 비교한 것입니다.

항목	Gemma 4 31B Dense	Qwen3 (dense)	gpt-oss 20B
라이선스 유형	Apache 2.0 (2026-04 변경)	Apache 2.0	Apache 2.0
상업적 이용	허용	허용	허용
파생 모델 재배포	허용	허용	허용
파인튜닝 허용	허용	허용	허용
법무 검토 부담	낮음	낮음	낮음
국내 기업 PoC 시작 속도	빠름	빠름	빠름

구 Gemma 3 27B 까지 적용되던 Gemma Terms of Use 는 Gemma 4 발표와 함께 신규 배포에서 폐기되었습니다. 본 1줄 각주는 2026년 4월 이전 PoC 자료를 보유한 조직의 마이그레이션 판단을 위한 legacy reference 입니다.

7.2.2 VRAM 요구 — 16GB · 24GB · 48GB · 80GB 4 등급

VRAM 이 GPU 구매 비용을 결정하는 구조

LLM 을 로컬에서 추론할 때 GPU 의 VRAM 용량은 실질적인 하드웨어 구매 예산을 결정하는 단일 핵심 지표입니다. 모델 가중치를 VRAM 에 올려야 추론이 가능하고, VRAM 이 부족하면 추론 자체가 불가능하거나 성능이 극도로 저하됩니다. 동일 모델이라도 FP16(반정밀도 부동소수점)으로 로드할 때와 Q4_K_M(4비트 양자화) 으로 압축해 로드할 때 필요한 VRAM 이 절반 이하로 줄어듭니다. 따라서 "모델을 어떤 정밀도로 운영할 것인가"와 "VRAM 을 얼마나 확보할 수 있는

가"를 함께 검토해야 합니다. Gemma 4 31B Dense 의 라이선스 부담이 사라진 지금, VRAM 등급은 사실상 모델 선정의 1차 분기로 격상되었습니다.

16GB 등급: gpt-oss 20B 와 소비자용 GPU

gpt-oss 20B 는 MoE(Mixture-of-Experts, 전문가 혼합) 구조로 설계되어 총 파라미터는 21B 이지만 추론 시 실제 활성화되는 파라미터는 3.6B 에 불과합니다 [S09]. 이 구조 덕분에 추론에 필요한 최소 메모리가 16GB 로 내려갑니다. RTX 4090 24GB 나 RTX 3090 24GB 같은 소비자 등급 GPU 1장으로 운영할 수 있어, 별도 서버 랙 없이 워크스테이션 수준 장비로 시작 가능합니다. 소규모 팀이 PoC 단계에서 초기 투자를 최소화하며 시작하기에 가장 적합한 선택지입니다. gpt-oss GitHub 저장소는 MXFP4 native 양자화 가중치를 함께 제공해 별도 양자화 도구 없이 16GB 환경에서 즉시 로드할 수 있습니다.

24GB 등급: Gemma 4 31B Dense 의 Q4 양자화 구간과 Qwen3 14B

Gemma 4 31B Dense는 FP16 전체 로드 시 약 62GB 의 VRAM 이 필요하지만, Q4_K_M 양자화를 적용하면 약 18~26GB 구간으로 내려옵니다. RTX 4090 단일 카드로 운영 가능하며, 일부 케이스에서는 28GB 급 워크스테이션 GPU 가 안전 여유을 추가로 확보합니다. 양자화 수준에 따라 응답 품질 손실이 발생할 수 있으므로 도입 전 사내 주요 태스크로 품질 평가를 별도로 수행하는 것이 좋습니다. 같은 24GB 등급에는 Qwen3 14B dense 모델도 들어옵니다. 한국어 처리 중심이면서 256K 까지 컨텍스트가 필요하지 않은 워크플로에서는 Qwen3 14B 가 더 가벼운 선택지가 됩니다.

48GB 이상 등급: 31B FP16 과 32B Dense

Qwen3 32B dense 모델, 또는 FP16 전체 정밀도로 Gemma 4 31B Dense 를 운영하려면 48GB 이상의 VRAM 이 필요합니다. 이 구간부터는 소비자용 단일 GPU 로 대응이 어렵고 NVIDIA A100 40GB 듀얼 구성이나 H100 80GB 단일 카드 급 하드웨어가 필요해집니다. 초기 투자비가 크게 증가하므로, 48GB 이상 등급이 필요한 모델은 사용량·품질 요건이 24GB 급 모델로 해결되지 않는다는 명확한 근거가 있을 때 선택하는 것이 합리적입니다. Hermes Agent 의 LiteLLM 라우터는 태스크 유형에 따라 모델을 분리 호출할 수 있어, 평시 작업은 16GB·24GB 모델로 처리하고 고품질 추론이 필요한 일부 태스크만 48GB 급 모델로 라우팅하는 운영이 가능합니다.

80GB 등급: 대용량 MoE 와 엔터프라이즈 서버

gpt-oss 120B 나 Qwen3 235B MoE 처럼 100B 를 초과하는 모델은 80GB VRAM 이 필수입니다 [S09]. H100 SXM 80GB 급 카드 한 장 이상이 필요하며, 클라우드 H100 인스턴스 기준 시간당 약 2달러 이상이 과금됩니다. 이 구간의 모델은 대부분 에이전트 조율 역할보다 전문 도메인 추론(예: 법률 문서 분석, 대용량 코드 리뷰)에 적합합니다. Gemma 4 26B A4B(MoE, 활성 4B) 는 80GB 가 아닌 24~32GB 구간에서 동작하면서 MoE 의 효율을 누릴 수 있는 절충 옵션이지만, 31B Dense 가 동급 또는 더 높은 텍스트 품질을 보이는 Arena 벤치마크 결과가 있어 Hermes Agent 기본 추천은 31B Dense 입니다.

아래 표는 세 모델과 주요 양자화 옵션별 VRAM 요구량을 정리한 것입니다.

모델	정밀도	최소 VRAM	권장 GPU 예시
gpt-oss 20B	MXFP4 native / FP16	16GB	RTX 4090, RTX 3090
Gemma 4 31B Dense	Q4_K_M 양자화	18~26GB	RTX 4090, RTX 5090
Gemma 4 26B A4B (MoE)	Q4_K_M 양자화	16~24GB	RTX 4090
Gemma 4 31B Dense	FP16 전체	~62GB	A100 80GB 또는 H100
Qwen3 14B / 32B dense	Q4 양자화	24~48GB	RTX 4090 x2 또는 A100 40GB
Qwen3 235B MoE	Q4 양자화	80GB+	H100 SXM 80GB 이상
gpt-oss 120B	MXFP4 native	80GB+	H100 SXM 80GB 이상

7.2.3 한국어 · 컨텍스트 · 멀티모달 통합 의사결정 트리

한국어 지원의 실질적 의미

세 모델 모두 공식적으로 한국어를 지원합니다. Gemma 4 31B Dense는 140개 이상 언어를 지원하며 Gemma 3 와 동일한 다국어 토크나이저 균형을 유지합니다. Qwen3 는 119개 언어를 지원하고 한국어도 포함됩니다 [S08]. gpt-oss 20B 는 영어 중심 벤치마크에서 OpenAI o3-mini 수준의 성능을 보이지만 한국어 specific 벤치마크 결과는 공개되어 있지 않습니다 [S09]. Gemma 4 또한 출시 직후라 KMMLU(Korean Massive Multitask Language Understanding, 한국어 대규모 언어 이해 벤치마크) 점수가 공개되어 있지 않으므로, "140개 언어 지원"이라는 선언이 한국어 실무 성능을 자동으로 보장하지는 않습니다. 도입 전 사내 실제 업무 데이터 샘플로 한국어 출력 품질을 자체 평가하는 절차가 필수입니다 [S12]. KMMLU 와 같은 공개 평가 세트를 활용하면 객관적 비교 기준을 마련할 수 있습니다.

컨텍스트 창 크기와 에이전트 워크플로의 관계

컨텍스트 창(Context Window, 한 번에 처리 가능한 텍스트 길이)은 에이전트 워크플로에서 특히 중요합니다. Hermes Agent 는 태스크 계획·도구 호출 이력·이전 에이전트 결과물을 모두 LLM 컨텍스트에 누적하며 동작합니다. 컨텍스트가 부족하면 이력이 잘려 에이전트가 앞서 수행한 단계를 반복하거나 일관성이 깨지는 현상이 발생합니다. 본 항목에서 Gemma 4 31B Dense 가 보이는 변화가 가장 큽니다. Gemma 3 27B 의 128K 컨텍스트가 Gemma 4 에서 256K 로 2배 확장되었습니다. Qwen3 dense 모델은 128K 토큰 컨텍스트를 지원하고 [S08], gpt-oss 20B 의 공식 컨텍스트 창 크기는 공개 문서에서 별도로 명시되지 않으므로 실제 운영 전 실측이 필요합니다. 256K 컨텍스트는 RFP 원문 1건 전체와 기업 정책 문서 묶음을 동시에 컨텍스트에 올린 채 에이전트가 단일 세션 안에서 작업을 마무리할 수 있는 수준입니다.

멀티모달 요건과 모델 선택

멀티모달(Multimodal, 텍스트·이미지 등 복수 데이터 유형 처리) 역량은 모든 시나리오에 필요하지 않습니다. 문서 요약·코드 생성·데이터 분석처럼 텍스트만 다루는 워크플로에서는 고려 대상이 아닙니다. 그러나 제조 공정 이미지 분석, 설계 도면 리뷰, 품질 검사 사진 처리처럼 이미지나

영상이 입력에 포함되는 시나리오라면 멀티모달 지원 모델이 필수입니다. Gemma 4 31B Dense 는 텍스트·이미지·비디오 입력을 기본 지원하며, Gemma 4 E2B / E4B 라인업은 추가로 오디오 입력까지 처리합니다. Qwen3-Omni 는 텍스트·음성·이미지·비디오를 통합 처리합니다 [S08]. 텍스트 전용 워크플로라면 gpt-oss 20B 가 낮은 VRAM 요구로 비용 효율적인 선택입니다.

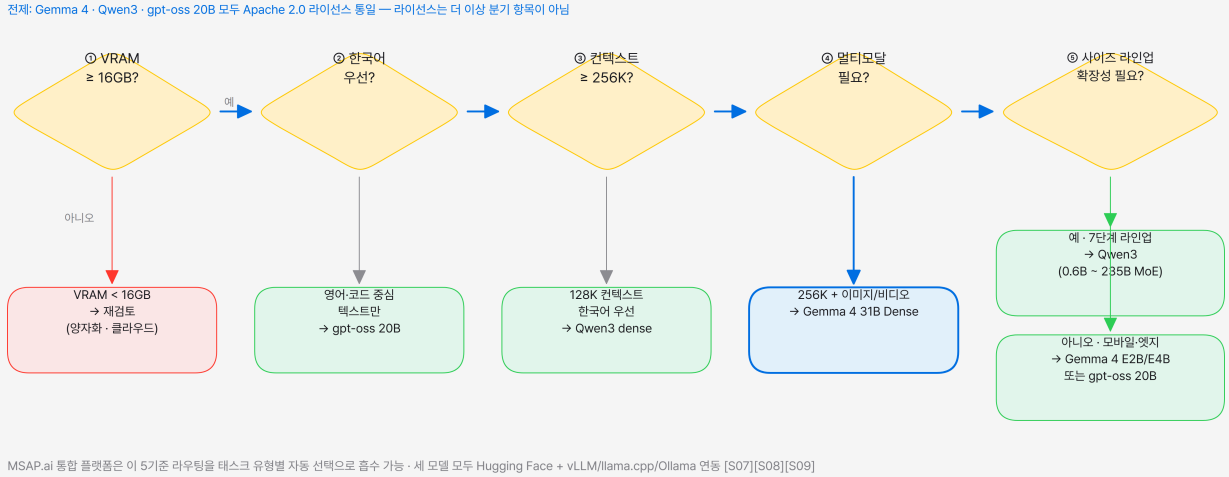
사이즈 라인업 확장성 — PoC 부터 본격 운영까지의 단계 전환

5번째 기준인 사이즈 라인업 확장성은 PoC 부터 본격 운영까지 단일 모델 패밀리로 단계적 업그레이드가 가능한지를 판단하는 항목입니다. Qwen3 는 0.6B / 1.7B / 4B / 8B / 14B / 32B dense 와 235B MoE 까지 7단계 사이즈 라인업을 제공해 PoC 단계의 4B 부터 본격 운영의 32B 까지 동일 토크나이저·동일 파인튜닝 노하우로 전환할 수 있습니다 [S08]. Gemma 4 는 E2B(≈2.3B effective) / E4B(≈4.5B effective) / 26B A4B(MoE) / 31B Dense 4단계 라인업을 제공해 모바일·임베디드 디바이스부터 데이터센터 추론까지 단일 패밀리로 커버합니다. gpt-oss 는 20B 와 120B 2개 사이즈만 제공되어 라인업 폭이 가장 좁습니다.

5가지 기준 의사결정 트리

아래 도식은 VRAM → 한국어 → 컨텍스트 → 멀티모달 → 사이즈 라인업 확장성의 5가지 기준을 순서대로 적용해 최종 모델을 결정하는 흐름입니다. 라이선스는 더 이상 분기 항목이 아니므로 트리에 등장하지 않습니다.

그림 7-1. 5기준 의사결정 트리 — VRAM · 한국어 · 컨텍스트 · 멀티모달 · 사이즈 라인업



5가지 기준(VRAM → 한국어 → 컨텍스트 → 멀티모달 → 사이즈 라인업 확장성)을 순서대로 통과하며 Gemma 4 31B Dense / Qwen3 / gpt-oss 20B 중 하나로 좁히는 의사결정 트리. 각 분기 끝 리프 노드에 "한국어 우선·16GB GPU·텍스트만 → gpt-oss 20B" 형태의 추천 문구를 포함. 왼쪽에서 오른쪽으로 기준을 순차 통과하는 플로차트 형식.

첫 번째 기준은 VRAM 입니다. 16GB 이하 단일 GPU 만 확보 가능하다면 gpt-oss 20B 가 유일한 현실적 선택입니다. 24GB 급이 가용하면 Gemma 4 31B Dense Q4 양자화 또는 Qwen3 14B dense 가 후보로 들어옵니다. 48GB 이상 확보 가능하면 Qwen3 32B dense 또는 Gemma 4 31B Dense FP16 이 추가됩니다. 80GB 이상이면 gpt-oss 120B 와 Qwen3 235B MoE 까지 후보가 확장됩니다.

두 번째 기준은 한국어입니다. 한국어 출력 품질이 최우선이라면 세 모델 모두 자체 평가 후 결정해야 합니다. Qwen3 는 한국어를 포함한 119개 언어에 학습 시 안배되어 [S08] 한국어·중국어 혼용 글로벌 영업 시나리오에서 유리합니다. Gemma 4 의 140개 언어 지원은 출시 직후라 한국어 specific 평가 자료가 부족하므로 PoC 시점에 KMMLU 1차 측정이 필요합니다.

세 번째 기준은 컨텍스트 창 크기입니다. RFP 원문이나 기업 정책 문서를 통째로 컨텍스트에 올려야 하는 256K 급 워크플로에는 Gemma 4 31B Dense 가 단독으로 적합합니다. 일반적인 에이전트 이력 누적(128K) 으로 충분하다면 Qwen3 dense 도 동등합니다 [S08].

네 번째 기준은 멀티모달입니다. 이미지·비디오 입력이 필요하면 Gemma 4 31B Dense 가 가장 폭넓은 멀티모달 입력을 지원합니다. 오디오 입력까지 필요하다면 Gemma 4 E2B / E4B 또는 Qwen3-Omni 가 선택지입니다. 텍스트만 처리한다면 가장 낮은 VRAM 요구의 gpt-oss 20B 가 비용 측면에서 최적입니다.

다섯 번째 기준은 사이즈 라인업 확장성입니다. PoC 단계의 가벼운 모델로 시작해 본격 운영 단계에서 동일 패밀리의 큰 모델로 단계적 업그레이드가 필요한 조직은 Qwen3 7단계 또는 Gemma 4 4단계 라인업을 선택하는 것이 합리적입니다. gpt-oss 는 20B 와 120B 두 단계뿐이므로 중간 단계 전환 옵션이 제한됩니다.

아래 표는 각 리프 시나리오에서 추천 모델을 정리한 것입니다.

시나리오 조합	추천 모델
16GB GPU + 텍스트 전용 + 빠른 PoC	gpt-oss 20B
24GB GPU + 한국어 우선 + 텍스트	Qwen3 14B dense
24GB GPU + 256K 컨텍스트 + 이미지 입력	Gemma 4 31B Dense Q4_K_M
48GB GPU + 256K 컨텍스트 + 멀티모달	Gemma 4 31B Dense FP16
한국어+음성 멀티모달 + 48GB+ GPU	Qwen3-Omni
모바일·엣지 디바이스 임베디드	Gemma 4 E2B / E4B
고정밀 추론 + 80GB GPU 가용	gpt-oss 120B 또는 Qwen3 235B MoE

MSAP.ai 와 같은 국내 통합 AI 플랫폼을 도입한 조직이라면 이 의사결정 트리의 라우팅 부담을 플랫폼 수준에서 흡수할 수 있습니다. 태스크 유형에 따라 최적 모델을 자동 선택하는 라우팅 정책을 플랫폼에 위임하면, IT 담당자가 매번 수동으로 모델을 지정하는 운영 오버헤드가 줄어듭니다.

세 모델 모두 Hugging Face 에서 가중치를 직접 다운로드할 수 있으며, vLLM·llama.cpp·Ollama 같은 로컬 추론 도구와 연동해 Hermes Agent 의 `LLM_BASE_URL` 설정 한 줄만 바꾸면 즉시 전환됩니다. Gemma 4 31B Dense 는 Ollama 의 `gemma4:31b` 태그와 Hugging Face `google/gemma-4-31b-it` 모델 ID 양쪽으로 동일 가중치를 받을 수 있습니다. Apache 2.0 으로 통일된 라이선스 환경에서는 VRAM 실측이 완료된 모델이라면 PoC 환경에서 하루 안에 Hermes Agent 와의 연동 검증을 마칠 수 있습니다 [S07, S08, S09].

8장. Local LLM 모델별 적용 가이드 — Gemma 4 31B Dense · Qwen3 · gpt-oss 20B

Hermes Agent 가 어떤 Local LLM 위에서 동작하느냐는 서비스 품질과 인프라 비용을 동시에 결정하는 선택입니다. 클라우드 API 대신 사내 GPU 서버에 모델을 직접 올리는 방식은 데이터 외부 송신 없이 규정을 준수할 수 있다는 장점이 있지만, GPU 메모리 확보·양자화 설정·언어별 품질 검증이라는 운영 부담이 따릅니다. 8장은 2026년 6월 현재 Hermes Agent 와 함께 가장 많이 검토되는 세 모델 — Google Gemma 4 31B Dense, Alibaba Qwen3, OpenAI gpt-oss 20B — 의 파라미터·라이선스·VRAM 요구·한국어 처리 특성을 정리하고, 각 모델이 실제로 어떤 업무 시나리오에 적합한지 안내합니다. 세 모델 모두 Apache 2.0 라이선스로 상업적 활용이 자유로워졌으며 [S07], 도입 전 사내 법무 검토 부담이 크게 줄어들었습니다. 다만 모델별로 사이즈 라인업·컨텍스트 길이·멀티모달 범위·언어 지원 깊이가 다르므로, 조직별 업무 유형에 맞춰 1순위 후보를 분기해 검토하는 작업이 필요합니다.

항목	Gemma 4 31B Dense	Qwen3 32B	gpt-oss 20B
개발사	Google DeepMind	Alibaba (Qwen Team)	OpenAI
출시 시점	2026-04-02	2025년 후반	2025-08
총 파라미터	31B (dense)	32B (dense) / 235B (MoE)	21B (MoE)
Active 파라미터	31B	32B / 22B (MoE)	3.6B
학습 토큰	미공개	36T [S08]	미공개
컨텍스트	256K [S07]	128K	미공개 (실측 필요)
라이선스	Apache 2.0 [S07]	Apache 2.0	Apache 2.0
멀티모달	텍스트+이미지+비디오	Omni 변형 (음성+이미지+비디오)	텍스트 전용
최소 VRAM	18~26GB (Q4_K_M)	20GB (Q4_K_M 32B)	16GB
한국어 공식 벤치마크	미공개 (자체 평가 필수)	미공개 (자체 평가 필수)	미공개 (자체 평가 필수)

8.1 Google Gemma 4 31B Dense — Apache 2.0 + 256K 컨텍스트 + 멀티모달

Gemma 4 31B Dense 는 Google DeepMind 가 2026년 4월 2일 공개한 멀티모달 언어 모델로, 텍스트와 함께 이미지·비디오 입력을 단일 forward 패스 안에서 처리합니다 [S07]. 140개 이상의 언어를 지원하며 256K 토큰 컨텍스트 윈도우를 제공해 약 200페이지 분량의 문서를 단일 프롬프트로 처리할 수 있습니다. Hermes Agent 의 오케스트레이션 계층과 결합하면 도면 검수·영수증 OCR·차트 분석·짧은 비디오 클립 요약처럼 시각 정보 이해가 필요한 업무를 자동화할 수 있습니다. 무엇보다 Gemma 4 부터 라이선스가 Apache 2.0 으로 전환되어, 종래 Gemma 3 27B

에서 요구되던 "책임 있는 상업적 이용" 약관 검토 부담이 사라졌습니다 [S07]. 사내 법무팀의 사전 약관 검토 없이도 PoC 1주차 안에 모델 가중치를 사내 GPU 서버에 배포할 수 있다는 점이 도입 진입 장벽을 실질적으로 낮춥니다.

8.1.1 31B Dense 의 파라미터·학습·컨텍스트·라이선스

사이즈 라인업 4종과 31B Dense 의 위치

Gemma 4 는 단일 사이즈로 공개된 이전 세대와 달리 4종 사이즈 라인업으로 제공됩니다 [S07]. 가장 작은 변형은 E2B(약 2.3B effective) 와 E4B(약 4.5B effective) 로, Per-Layer Embeddings(PLE) 기법을 적용해 파라미터 효율을 끌어올린 모델입니다. 중간 사이즈는 26B A4B 로, Gemma 패밀리에서 처음 도입된 MoE(Mixture of Experts) 구조이며 추론 시 4B 만 활성화됩니다. 가장 큰 변형이 본 절의 주 대상인 31B Dense 입니다. 사이즈 라인업이 4종으로 확장된 결과 PoC 단계에서는 E4B 로 워크스테이션 검증을 마치고 본격 운영 단계에서 31B Dense 로 단계적 업그레이드하는 경로가 가능해졌습니다 [S07]. Gemma 3 세대처럼 PoC 와 본격 운영 사이에 모델 패밀리를 전면 교체해야 하는 부담이 사라진 셈입니다.

256K 컨텍스트의 운영적 의미

31B Dense 는 256K 토큰 컨텍스트 윈도우를 제공합니다 [S07]. 이는 Gemma 3 27B 의 128K 대비 정확히 2배이며, 약 200페이지 분량의 한국어 문서를 단일 프롬프트로 처리할 수 있는 수준입니다. 계약서 검토·회의록 요약·규정 문서 분석처럼 장문 처리가 필요한 업무에 직접 활용할 수 있으며, 특히 Hermes Agent 의 누적 컨텍스트 부담을 완화하는 효과가 큽니다. 에이전트가 태스크 계획·도구 호출 이력·이전 단계 산출물을 컨텍스트에 누적하며 동작할 때, 256K 윈도우는 15~20단계의 복합 태스크도 컨텍스트 잘림 없이 처리할 수 있는 여유를 제공합니다. Hermes 의 Archive 계층과 조합하면 컨텍스트 한도를 초과하는 분량도 Archive 가 분할·색인 처리한 뒤 필요한 구간만 LLM 에 전달하는 방식으로 운영할 수 있습니다.

Apache 2.0 라이선스로의 전환

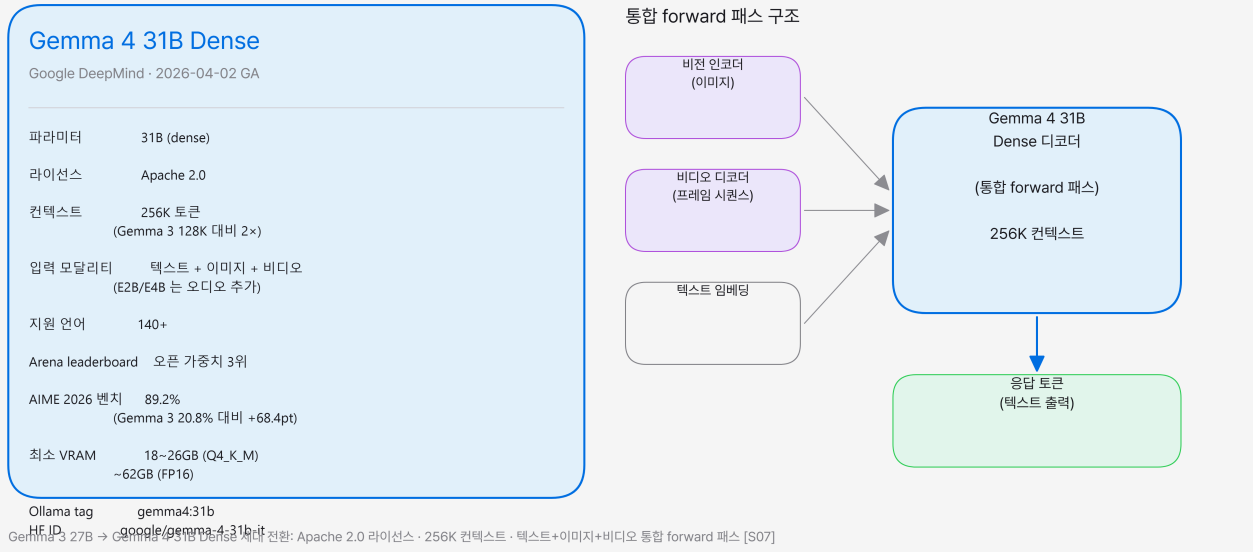
Gemma 4 부터 라이선스가 Apache 2.0 으로 전환되었습니다 [S07]. Apache 2.0 은 저작권 표시(attribution) 의무만 충족하면 상업적 이용·수정·재배포·파인튜닝 후 재배포를 모두 허용하는 라이선스로, MIT 와 함께 국내 기업의 법무 검토 부담이 가장 낮은 선택지입니다. 종래 Gemma 3 27B 까지 적용되던 Gemma Terms of Use 는 "책임 있는 상업적 이용"이라는 조건부 허용 조항을 두고 있어 사내 법무팀이 약관 원문(`ai.google.dev/gemma/terms`)을 별도 검토해야 했고, "책임 있는"의 구체적 범위 해석을 둘러싸고 PoC 0주차에 추가 작업이 발생했습니다. 2026년 4월 Gemma 4 출시와 동시에 이 검토 부담이 사라졌습니다. Google 공식 블로그는 Apache 2.0 라이선스 채택을 "production 환경에서 사용료·이용 제한 없이 활용 가능"하다는 점으로 명시했습니다 [S07].

Arena 텍스트 leaderboard 3위와 학습 토큰

31B Dense 는 업계 표준 Arena 텍스트 leaderboard 에서 오픈 가중치 모델 3위를 기록했으며, 같은 패밀리의 26B A4B 가 6위에 자리합니다 [S07]. 이는 31B 라는 비교적 작은 파라미터 규모로 400B 급 대형 모델 일부를 추월했음을 의미합니다. AIME 2026 수학 추론 벤치마크에서는 89.2% 를 기록해 Gemma 3 의 20.8% 에서 큰 폭으로 향상되었습니다 [S07]. 학습 토큰 수는 공

식 자료에 명시되지 않았으며, Gemma 3 의 14T 와 단순 비교가 불가능합니다. 다만 벤치마크 수치를 통해 학습 데이터의 질과 사후 정렬(post-training) 단계가 이전 세대 대비 상당한 진전이 있었음을 추정할 수 있습니다. 공식 모델 카드는 `huggingface.co/google/gemma-4-31B` 와 Google AI for Developers 페이지(`ai.google.dev/gemma/docs/releases`)에서 확인 가능합니다 [S07].

그림 8-1. Gemma 4 31B Dense — 스펙 카드 (Apache 2.0 · 256K · 멀티모달)



8.1.2 멀티모달 (텍스트+이미지+비디오) 활용 시나리오

Gemma 4 의 네이티브 멀티모달 구조

Gemma 4 는 4종 사이즈 라인업 전체가 텍스트·이미지·비디오를 단일 forward 패스 안에서 처리하는 네이티브 멀티모달 구조입니다 [S07]. Gemma 3 세대가 4B 이상 크기 변형에서만 이미지를 지원했던 것과 달리, Gemma 4 는 가장 작은 E2B 부터 31B Dense 까지 모든 변형이 이미지와 비디오를 처리합니다. 오디오 입력은 E2B 와 E4B 두 사이즈만 지원하며, 26B A4B 와 31B Dense 에서는 텍스트·이미지·비디오 3종만 처리합니다. 변수 해상도(variable-resolution) 이미지 처리·OCR·차트 이해·비디오 프레임 시퀀스 분석이 동일한 모델 안에서 가능하다는 점이 종래 별도 비전 모델을 추가 배포해야 했던 운영 부담을 줄여줍니다.

이미지 입력 시나리오 — 도면·영수증·차트

31B Dense 의 이미지 이해 성능은 Gemma 4 패밀리에서 가장 정확합니다 [S07]. 도면 이미지를 업로드해 치수·부품명·결합 순서를 자동 추출하거나, 영수증 사진을 입력해 항목과 금액을 구조화된 JSON 으로 변환하는 시나리오에 추가 학습 없이 즉시 적용할 수 있습니다. 차트나 그래프 이미지를 받아 수치 요약 및 이상 탐지를 수행하는 것도 가능한 범위에 속합니다. 제조 현장에서 작업자가 모바일로 촬영한 부품 이미지를 사내 메신저로 전송하면 Hermes Agent 가 Gemma 4 31B Dense 에 이미지 검수를 위임하고, 결과를 다시 작업자에게 회신하는 워크플로가 단일 모델만으로 완결됩니다.

비디오 입력 시나리오 — 짧은 클립 요약과 프레임 시퀀스 분석

비디오 처리는 Gemma 4 가 Gemma 3 대비 추가한 핵심 modality 입니다 [S07]. 모델은 비디오를 프레임 시퀀스로 분해해 처리하며, 매뉴얼 동영상·CCTV 짧은 클립·제품 시연 영상을 텍스트 설명으로 요약할 수 있습니다. 사내 교육용 동영상에서 핵심 절차를 자동 추출하거나, 현장 점검 영상에서 이상 상황이 발생한 시점을 식별하는 시나리오가 단일 모델에서 완결됩니다. 비디오 처리는 이미지보다 토큰 소비가 크므로 256K 컨텍스트 윈도우의 여유가 운영적으로 중요한 의미를 가집니다. Hermes Agent 가 Gemma 4 31B Dense 를 backing LLM 으로 사용하면 비디오 분석 결과를 Archive 에 보존해 향후 동일 유형의 영상 분석 시 reference 로 활용할 수 있습니다.

8.1.3 Hermes Agent + LiteLLM 통합 — backing LLM 설정

LiteLLM 프록시 기반 표준 연동

Gemma 4 31B Dense 를 Hermes Agent 의 backing LLM 으로 설정할 때는 LiteLLM 프록시(proxy) 또는 Ollama 게이트웨이를 중간에 두는 구성이 표준입니다 [S10]. 공식 Hugging Face 모델 ID 는 `google/gemma-4-31b-it` 이며 Ollama 태그는 `gemma4:31b` 입니다 [S07]. Hermes 의 `setup` 명령으로 endpoint 주소(예: `http://litellm:4000`)를 등록하면 이후 Hermes 가 발행하는 모든 텍스트·이미지·비디오 요청이 해당 endpoint 로 라우팅됩니다. LiteLLM 의 virtual key 기능을 활용하면 Profile 단위(coding · research · personal) 로 별도 비용 추적도 가능합니다. 동일 LiteLLM 인스턴스에 Qwen3 와 gpt-oss 20B 를 함께 등록해 두면, 태스크 유형별 fallback 정책으로 멀티모달은 Gemma 4 31B Dense 로, 코드 생성은 gpt-oss 20B 로 분기하는 구성을 코드 수정 없이 설정만으로 운영할 수 있습니다.

Ollama 직접 호출 시나리오

조직 규모가 작아 LiteLLM 별도 운영이 부담스러운 경우 Ollama 단일 게이트웨이로 직접 연동하는 구성도 가능합니다. `ollama pull gemma4:31b` 명령으로 모델 가중치를 내려받은 뒤 Ollama 의 OpenAI 호환 endpoint(`http://localhost:11434/v1`)를 Hermes 에 등록하면 즉시 동작합니다. 다만 Ollama 단독 구성은 동시 요청 처리 성능이 vLLM 대비 떨어지므로 [S07], 동시 사용자 5명 이상 시나리오에서는 vLLM 또는 TensorRT-LLM 기반 배포로 전환하는 것이 권장됩니다. 추론 프레임워크 선택이 지연 특성을 상당 부분 결정한다는 점은 7장에서 정리한 것과 같습니다.

멀티모달 입력 라우팅 주의

Gemma 4 31B Dense 는 텍스트·이미지·비디오를 동일 endpoint 로 받지만, 멀티모달 요청은 텍스트 전용 요청보다 토큰 소비량과 추론 시간이 크게 늘어납니다 [S07]. Hermes Agent 의 Skill 작성 시 멀티모달 입력이 발생하는 Skill 은 별도 Profile 또는 별도 task 큐로 분리해 라우팅하는 구성이 운영 효율적입니다. 예를 들어 "도면 검수" Skill 은 Gemma 4 31B Dense 로 라우팅하고 일반 문서 요약 Skill 은 동일 모델이라도 별도 task 큐로 분리해 멀티모달 요청의 응답 지연이 일반 요청에 영향을 주지 않도록 격리하는 방식입니다.

8.1.4 한국어 처리와 KMMLU 자체 평가 절차

140+ 언어 지원의 실제 의미

Gemma 4 31B Dense 는 공식적으로 140개 이상의 언어를 지원합니다 [S07]. 그러나 언어 수는 커버리지를 보증하는 수치가 아니라, 해당 언어 텍스트가 학습 데이터에 포함되어 있다는 의미입니다. 한국어가 140개 안에 포함되어 있다는 점은 확인되지만, Google 공식 페이지에는 KMMLU(Korea Massive Multitask Language Understanding, 한국어 대규모 멀티태스크 언어 이해 벤치마크) 등 한국어 특화 벤치마크 결과가 별도로 공개되지 않았습니다 [S07]. Arena 텍스트 leaderboard 3위라는 영어 중심 평가 결과가 한국어 업무에서 동일한 품질을 보장하지는 않습니다. 자체 평가를 생략하면 도입 후 품질 미달 위험이 남습니다.

한국어 자체 평가 프로토콜 (100건 기준)

PoC 1주차에 아래 절차로 자체 평가를 수행하면 실제 업무 적합성을 투자 결정 이전에 확인할 수 있습니다. 평가 데이터는 실제 업무 데이터를 익명 처리해 사용하는 것이 가장 효과적입니다.

평가 항목	데이터 예시	합격 기준
문서 요약	사내 보고서 20건 (A4 3~5페이지)	핵심 문장 누락률 ≤ 10%
질의응답	사내 FAQ 30건 (정답 레이블 포함)	정확도 ≥ 80%
감성 분류	고객 리뷰 20건 (긍/부/중 레이블)	F1 ≥ 0.75
한국어 지시 이행	특수 지시어 20건 (조건부 형식)	형식 준수율 ≥ 90%
개체명 인식	공문서·계약서 10건	기관명·날짜 누락률 ≤ 5%

100건 기준 평가를 진행하고 합격 기준을 충족하지 못하는 항목이 2개 이상이면 파인튜닝(fine-tuning) 또는 Qwen3 32B 로 교체를 검토합니다. KMMLU 공개 평가 세트를 보조 지표로 활용하면 객관적 비교 기준을 마련할 수 있습니다 [S12]. 평가 결과는 PoC 보고서에 포함해 의사결정 권자가 모델 교체 여부를 판단할 근거로 남겨야 합니다.

Profile 단위 분리 운영 전략

모델이 특정 업무 유형에서만 기준을 미달한다면, Hermes Agent 의 Profile 기능을 활용해 해당 업무 유형의 Profile 에만 다른 모델을 지정하는 분리 운영도 선택지입니다. 예를 들어 문서 요약 Profile 에는 Qwen3 32B 를, 이미지·비디오 분석 Profile 에는 Gemma 4 31B Dense 를 배정하는 식입니다. LiteLLM 의 virtual key 분리와 결합하면 Profile 단위 월간 비용 추적도 동시에 가능해집니다 [S10].

8.1.5 Q4_K_M 양자화와 18~26GB VRAM 구간

Q4_K_M 양자화의 개념과 효과

양자화(quantization)는 모델 가중치를 32비트 부동소수점 대신 4비트 정수로 압축하는 기법입니다. Q4_K_M 은 llama.cpp 생태계에서 널리 쓰이는 4비트 양자화 방식으로, 모델 크기와 VRAM 사용량을 풀 정밀도(FP16) 대비 약 60~65% 줄이는 대신 품질 손실은 경험적으로 2~5% 수준에 그칩니다. Gemma 4 31B Dense 를 FP16 로 로드하면 약 62GB VRAM 이 필요하지만,

Q4_K_M 양자화를 적용하면 약 18~26GB 구간으로 내려와 RTX 4090 24GB 단일 카드에서 운영이 가능해집니다 [S07]. 256K 컨텍스트 윈도우의 KV 캐시(cache) 부담은 컨텍스트 길이 설정에 따라 추가로 4~8GB 가 더 요구되므로, 컨텍스트를 128K 또는 64K 로 제한하는 운영 옵션도 함께 검토해야 합니다.

GPU 모델별 운영 시나리오

GPU 모델	VRAM	정밀도	예상 동시 사용자	권장 용도
RTX 4090	24GB	Q4_K_M	3~5명 (순차 처리)	팀 단위 파일럿
NVIDIA A6000	48GB	Q4_K_M / Q8	8~12명	부서 단위 본격 운영
A100 80GB	80GB	FP16	10~15명	기업 중앙 서버
H100 80GB	80GB	FP16	15~20명	대규모 멀티에이전트

RTX 4090 단일 구성은 Q4_K_M 기준으로 31B Dense 를 운영 가능한 가장 경제적인 옵션입니다. 다만 Gemma 3 27B 가 같은 GPU 에서 16~24GB VRAM 으로 동작했던 것 대비 18~26GB 로 상단이 올라갔으므로, 컨텍스트 길이 설정에 따라 VRAM 여유가 빠듯해질 수 있습니다. 부서 단위 본격 운영 시 A6000 48GB 로 업그레이드하면 Q8 양자화로 품질 손실을 더욱 줄이면서 동시 사용자 8~12명을 안정적으로 수용할 수 있습니다. FP16 전체 정밀도가 필요한 고품질 시나리오는 A100 또는 H100 80GB 가 사실상 단일 선택지입니다 [S07].

운영 시 주의 사항과 단계적 도입 경로

Ollama 에서 Gemma 4 31B Dense 를 구동할 때는 `ollama run gemma4:31b` 실행 후 `nvidia-smi` 로 실제 VRAM 점유량을 확인하고, 26GB 를 초과하는 경우 컨텍스트 길이 파라미터(`--ctx-size`)를 128K 또는 64K 로 낮춰 재시도하는 절차가 권장됩니다. 256K 컨텍스트는 강력한 도구이지만 모든 워크플로에 필요하지는 않으며, 일반 문서 요약·Q&A 시나리오에서는 64K 로도 충분한 경우가 대부분입니다. 단계적 도입 경로는 다음과 같습니다. PoC 1주차에는 E4B(약 4.5B effective) 모델로 RTX 4070 워크스테이션에서 한국어 자체 평가를 진행하고, 합격하면 파일럿 단계에서 RTX 4090 1대로 31B Dense Q4_K_M 운영을 시작합니다. 본격 운영 단계에서는 A6000 48GB 또는 A100 80GB 로 확장합니다. 동일 패밀리 안에서 PoC → 본격 운영 경로가 매끄럽게 이어지는 점이 4종 사이즈 라인업의 운영적 가치입니다 [S07].

8.2 Alibaba Qwen3 — Apache 2.0 + 119 언어 + MoE 옵션

Qwen3 는 Alibaba Cloud 의 Qwen 팀이 공개한 언어 모델 시리즈로, 0.6B 부터 235B 까지 폭넓은 파라미터 범위를 Apache 2.0 라이선스로 제공합니다. Apache 2.0 은 수정·재배포·상업적 활용이 모두 자유로우며 소스 공개 의무가 없어 국내 기업이 사내 환경에 모델을 배포할 때 법무 부담이 가장 낮은 라이선스 가운데 하나입니다 [S08]. 2026년 4월 Gemma 4 가 Apache 2.0 으로 합류하면서 세 모델 모두 동일 라이선스로 통일되었지만, Qwen3 는 0.6B부터 235B 까지 7종의 사이즈 라인업을 제공해 단일 모델 패밀리 안에서 PoC 부터 본격 운영까지 단계 교체 폭이

가장 넓다는 차별점이 남아 있습니다. 119개 언어를 지원하는 멀티링구얼 특성과 MoE(Mixture of Experts, 전문가 혼합) 변형의 메모리 효율이 더해져 글로벌 사업장 운영이나 한국어·중국어 병행 처리가 필요한 조직에 특히 적합합니다.

8.2.1 0.6B ~ 235B 파라미터 범위와 36T tokens 학습

단계별 모델 교체 전략

Qwen3 의 파라미터 라인업은 0.6B · 1.7B · 4B · 8B · 14B · 32B (dense) 와 235B (MoE) 로 구성됩니다 [S08]. Gemma 4 의 4종 사이즈와 비교하면 Qwen3 가 7종으로 가장 세분화되어 있어, PoC 에서 본격 운영까지 동일한 모델 패밀리 안에서 더 촘촘하게 단계적으로 업그레이드할 수 있습니다. 4B → 14B → 32B → 235B 로 올리면서 같은 LiteLLM 설정·Hermes Profile 을 그대로 재사용할 수 있다는 점이 실용적 이점입니다.

단계	모델	VRAM 요구	적합 시나리오
PoC (1~4주)	Qwen3-4B	4GB	노트북·소형 워크스테이션
파일럿 (1~3개월)	Qwen3-14B	10GB	RTX 3080 / 4070
본격 운영	Qwen3-32B	20GB (Q4_K_M)	RTX 4090 / A6000
대규모 멀티에이전트	Qwen3-235B-A22B (MoE)	80GB+	A100 80GB × 2

36T 토큰 학습의 시사점

Qwen3 의 36조(36T) 토큰 학습 규모는 공식적으로 명시된 수치 가운데 가장 큰 편입니다 [S08]. 학습 데이터가 많을수록 범용 지식과 논리 추론 능력이 향상되는 경향이 있으며, Qwen3-32B 는 이전 세대의 Qwen2.5-72B 와 비슷한 수준의 성능을 절반 이하의 파라미터로 달성한다는 평가를 받고 있습니다. 코드 생성·수학 추론 등 구조화된 사고가 필요한 태스크에서 특히 눈에 띄는 성능을 보이며 비즈니스 규칙 기반의 자동화 스크립트 생성에 적합합니다. Gemma 4 31B Dense 가 Arena 3위로 일반 응답 품질에서 강세를 보이는 것과 달리, Qwen3 는 사이즈 라인업의 폭과 멀티링구얼 깊이가 차별점입니다.

Apache 2.0 라이선스의 실제 의미

Apache 2.0 라이선스는 모델 가중치를 사내 서버에 배포하고, 애플리케이션에 통합해 상업적으로 서비스하고, 필요에 따라 파인튜닝(fine-tuning) 후 재배포하는 모든 행위를 허용합니다. 다만 원 저작자 표시(attribution) 문구를 파생 산출물에 포함해야 하는 의무가 있으며, 이는 배포되는 소프트웨어의 라이선스 파일에 한 줄 추가하는 수준으로 충족됩니다 [S12]. MSAP.ai 와 같은 엔터프라이즈 AI 플랫폼에서 Qwen3 를 기반 모델로 채택할 경우 법무 검토 비용이 가장 낮은 선택지입니다. Gemma 4 가 동일 라이선스로 합류한 지금 두 모델 사이의 선택은 라이선스가 아닌 사이즈 라인업의 세분화 정도·멀티모달 범위(이미지/비디오 vs 음성/이미지/비디오)·언어 지원 깊이로 갈리게 되었습니다.

8.2.2 한국어 + 119 언어 멀티링구얼 활용

멀티링구얼 지원의 운영 가치

Qwen3 는 119개 언어를 지원하며 사전 학습 데이터에 한국어·중국어·일본어·영어·베트남어가 균형 있게 포함되어 있습니다 [S08]. 한국 기업이 중국 합작 공장·베트남 생산 거점·인도네시아 영업 조직을 운영하는 경우 단일 Qwen3 모델로 국가별 사업장 담당자와 한 시스템 안에서 소통할 수 있습니다. 종래에는 언어별로 번역 API 를 추가하거나 지역별로 별도 LLM 을 구성해야 했지만, Qwen3 한 모델이 해당 언어로 직접 답변을 생성하므로 운영 구성이 단순해집니다. Gemma 4 가 140개 언어를 지원해 언어 수에서는 우위에 있지만, Qwen3 는 중국어·동남아시아 언어 데이터 비중이 학습 단계에서 더 두텁다는 평가가 일반적입니다.

국내 기업 우선순위 5개 언어 평가 기준

공식 멀티링구얼 벤치마크는 Qwen3 의 한국어 특화 성능을 별도로 공시하지 않으므로 PoC 단계에서 아래 언어별 평가를 직접 수행하는 것이 안전합니다 [S08].

언어	평가 데이터 예시	최소 합격 기준
한국어	사내 보고서 요약 20건	정확도 ≥ 80%
영어	영문 계약서 요약 20건	정확도 ≥ 85%
중국어	중국 지사 업무 지시문 20건	번역 품질 BLEU ≥ 0.45
일본어	일본 고객 이메일 분류 20건	F1 ≥ 0.75
베트남어	생산 현장 보고서 요약 10건	핵심 항목 누락률 ≤ 15%

평가 결과를 누적 관리하면 모델 버전 업그레이드 시 품질 퇴행을 조기에 발견할 수 있습니다. Qwen3 시리즈는 Qwen 팀이 지속적으로 업데이트하므로 버전 변경 시마다 위 평가를 재실행하는 절차를 운영 매뉴얼에 포함하는 것이 권장됩니다.

단일 모델 전략의 한계

119개 언어 지원은 범용적이지만 언어별 학습 데이터 비중이 다르므로 저자원 언어(low-resource language)에서는 품질이 낮아질 수 있습니다. 인도네시아어·태국어·아랍어처럼 사용 빈도가 낮은 언어가 업무에 포함된다면 해당 언어만 별도 전문 번역 API 로 보완하는 하이브리드 구성을 검토해야 합니다. Gemma 4 와의 선택 기준 가운데 하나는 다국적 사업장의 언어 분포입니다. 중국·동남아 비중이 크면 Qwen3, 유럽·중동·아프리카 비중이 크면 Gemma 4 가 1순위 후보가 됩니다.

8.2.3 Qwen3-Omni — 텍스트 + 음성 + 이미지 + 비디오 멀티모달

Qwen3-Omni 의 구조

Qwen3-Omni 는 텍스트·음성·이미지·비디오를 단일 모델에서 처리하는 엔드투엔드(end-to-end) 멀티모달 모델입니다. 별도 STT(음성-텍스트 변환)/TTS(텍스트-음성 변환) 서버를 구성하지 않아도 Qwen3-Omni 하나로 음성 입력을 텍스트로 전환하고 답변을 음성으로 출력하는 전 과정이 처리됩니다 [S08]. 36개 오디오·오디오-비주얼 벤치마크 가운데 32개에서 오픈소스 최고 수준의 성능을 기록했으며 Gemini-2.5-Pro 와 GPT-4o 의 음성 처리 성능을 여러 항목에서 앞섭니다.

Gemma 4 31B Dense 가 비디오를 지원하지만 음성 입출력은 E2B/E4B 작은 사이즈만 지원하는 것과 달리, Qwen3-Omni 는 단일 모델로 4종 modality 를 모두 처리합니다.

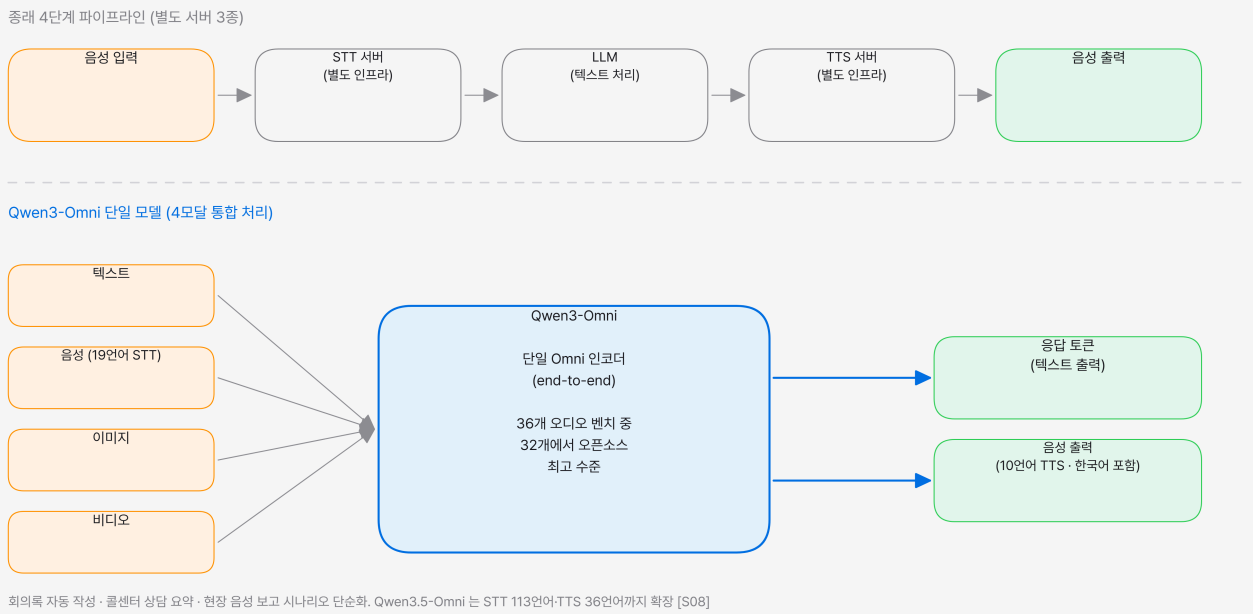
한국어 음성 지원 범위

Qwen3-Omni 는 음성 인식(STT)에서 19개 언어, 음성 생성(TTS)에서 10개 언어를 지원하며 한국어가 두 범주 모두에 포함됩니다 [S08]. 최신 버전인 Qwen3.5-Omni 는 음성 인식 지원 언어를 113개로 확장했으며, 음성 생성 언어도 36개로 늘렸습니다. 10시간 이상의 장시간 음성 파일을 단일 세션에서 처리하는 것도 가능합니다. 이는 회의록 자동 작성 시나리오에서 별도 ASR(자동 음성 인식) 서버를 운영하지 않아도 된다는 실용적 이점을 제공합니다.

음성 채널 자동화 시나리오

Hermes Agent 의 Qwen3-Omni 연동은 콜센터 상담 녹취 자동 요약, 사내 회의 실시간 전사(轉寫), 현장 직원의 음성 업무 보고 수집처럼 음성 채널을 직접 처리해야 하는 시나리오에서 구현 복잡도를 줄입니다. 종래에는 음성 입력 → 별도 STT 서버 → LLM 텍스트 처리 → 별도 TTS 서버 → 음성 출력이라는 4단계 파이프라인이 필요했지만, Qwen3-Omni 를 사용하면 음성 입력 → Qwen3-Omni → 음성 출력으로 단순화됩니다.

그림 8-2. Qwen3-Omni 4모달 처리 흐름 — 기존 4단계 파이프라인 대비 단일 모델 대체



8.3 OpenAI gpt-oss 20B — 16GB 메모리에서 o3-mini 급 성능

gpt-oss 20B 는 OpenAI 가 GPT-2 이후 처음 공개한 오픈 가중치 범용 LLM 으로, 2025년 8월 Apache 2.0 라이선스로 공개됐습니다 [S09]. 21B 총 파라미터 가운데 토큰 처리 시 실제로 활성화되는 파라미터는 3.6B 에 불과한 MoE 구조를 채택해 동급 dense 모델보다 메모리와 연산 효율이 높습니다. 최소 16GB 메모리만 있으면 단일 워크스테이션에서 동작하므로 GPU 서버를 별도 구성하기 어려운 팀 단위 PoC 의 진입 장벽을 실질적으로 낮춥니다. Gemma 4 31B Dense

가 멀티모달·다국어·256K 컨텍스트로 광범위한 시나리오를 커버하는 모델이라면, gpt-oss 20B 는 영어 중심·코드/수학 추론 강세·최소 인프라라는 좁고 분명한 강점을 가진 모델입니다.

8.3.1 21B 총 파라미터 · 3.6B active MoE 구조

MoE 구조의 작동 원리

MoE(Mixture of Experts)는 모델 내부에 여러 전문가 서브네트워크를 두고 각 토큰을 처리할 때 전체 전문가 가운데 일부만 선택해 활성화하는 구조입니다. gpt-oss 20B 는 24개 레이어에 21B 총 파라미터를 보유하지만 토큰 하나를 처리할 때는 3.6B 분량의 전문가만 활성화됩니다 [S09]. 이 구조 덕분에 추론(inference) 시 연산량은 3.6B dense 모델과 비슷한 수준이지만 모델이 갖는 지식 용량은 21B 규모를 유지합니다. Gemma 4 패밀리에서도 26B A4B 변형이 동일한 MoE 접근을 채택했으며, 두 모델은 MoE 라는 공통 구조를 통해 추론 효율을 끌어올린 사례라는 공통점이 있습니다 [S07].

MoE vs. Dense 메모리·연산 비교

항목	gpt-oss 20B (MoE)	동급 Dense 21B
총 파라미터	21B	21B
추론 시 Active 파라미터	3.6B	21B
최소 VRAM (FP16)	~42GB	~42GB
최소 메모리 (INT4 양자화)	16GB	~12GB
추론 속도 (상대 비교)	빠름 (active 3.6B 기준)	느림 (21B 전체)
토큰당 연산 비용	낮음	높음

MoE 구조는 활성 파라미터 수가 적어 추론 속도가 빠르지만, 모든 전문가 가중치를 메모리에 올려 두어야 하므로 저장 공간은 dense 모델과 유사하게 요구됩니다. OpenAI 는 MXFP4(FP4 혼합 정밀도) 양자화를 네이티브(native)로 지원해 16GB 메모리 환경에서도 품질 손실을 최소화하면서 동작할 수 있게 설계했습니다 [S09].

gpt-oss 120B 와의 선택 기준

gpt-oss 시리즈에는 20B 외에 120B 변형도 존재하며 120B 는 최소 80GB 메모리를 요구합니다. 일반 업무 자동화·코드 생성·문서 처리에는 20B 가 120B 대비 여러 벤치마크에서 동등하거나 일부 항목에서 우수한 결과를 보였습니다 [S09]. 따라서 80GB 고가 GPU 를 확보하지 않아도 되는 20B 를 먼저 검토하고, 추론 정확도가 명백히 부족한 경우에만 120B 로 업그레이드하는 순서가 합리적입니다.

8.3.2 OpenAI o3-mini 수준 벤치마크와 한국어 평가 권고

공식 벤치마크 결과

OpenAI 공식 자료에 따르면 gpt-oss 20B 는 공통 벤치마크에서 o3-mini 수준의 성능을 보입니다 [S09]. 구체적으로 MMLU(범용 지식 이해 벤치마크)에서 69%, 코드 생성 벤치마크인

HumanEval 에서 73점을 기록했습니다. 수학 추론 태스크에서는 Chain-of-Thought(CoT, 단계적 추론) 프롬프트 사용 시 성능이 15% 향상되는 특성도 확인됐습니다 [S09]. 코드 생성과 수학 추론에서 두드러진 강점을 보이는 반면 범용 지식 회상 영역에서는 오픈소스 상위 모델에 비해 중간 수준에 머문다는 평가도 있습니다. Gemma 4 31B Dense 가 AIME 2026 에서 89.2% 를 기록한 것과 비교하면 [S07], 수학 추론에서는 Gemma 4 가 더 앞서지만 코드 생성에서는 gpt-oss 20B 의 강점이 유지됩니다.

벤치마크	gpt-oss 20B	비교 기준
MMLU	69%	o3-mini 수준
HumanEval	73점	gpt-oss 120B(71점) 보다 높음
수학 추론 (CoT)	+15% 향상	CoT 프롬프트 적용 시

한국어 평가 부재와 자체 평가 필요성

OpenAI 공식 모델 카드 및 공개 자료에는 한국어 특화 벤치마크(KMMLU 등) 결과가 포함되지 않았습니다 [S09]. gpt-oss 20B 가 영어 중심 데이터로 사전 학습됐다는 점을 고려하면, 한국어 업무에 적용하기 전 반드시 자체 평가를 실행해야 합니다. 8.1.4 에서 제시한 100건 한국어 자체 평가 프로토콜을 그대로 재사용할 수 있습니다. gpt-oss 20B 가 코드 생성·수학 추론에서 강점을 보인다는 점을 감안하면, 한국어 기반의 SQL 쿼리 생성·업무 로직 스크립트 작성 태스크에서 먼저 평가해 강점을 확인하는 접근이 효과적입니다.

8.3.3 16GB 메모리 단일 워크스테이션 운영 시나리오

단일 워크스테이션 PoC 의 실현 가능성

gpt-oss 20B 는 MXFP4 양자화 적용 시 16GB 메모리에서 동작하며 이는 RTX 4070 · RTX 4080 같은 중급 소비자 GPU 와 Mac Studio(M3 Ultra, 96GB 통합 메모리) 에서 즉시 운영할 수 있다는 의미입니다 [S09]. 서버 라인 GPU(A100 등)를 구비하지 않아도 개발팀 또는 현업 부서의 기존 워크스테이션 한 대로 PoC 를 시작할 수 있어 도입 결정에 앞서 현장 검증을 빠르게 진행할 수 있습니다. Gemma 4 31B Dense Q4_K_M 이 RTX 4090 24GB 를 요구하는 것과 비교하면, gpt-oss 20B 는 GPU 등급을 한 단계 낮춰도 동작한다는 점에서 PoC 진입 장벽이 낮습니다.

워크스테이션별 동시 사용자 추정

하드웨어	메모리	예상 동시 세션	평균 응답 시간 (500 토큰)
RTX 4070 + 일반 RAM 32GB	12GB VRAM + 32GB RAM	1~2명	8~12초
RTX 4080 16GB	16GB VRAM	2~3명	5~8초
RTX 4090 24GB	24GB VRAM	4~6명	3~5초
Mac Studio M3 Ultra 96GB	96GB 통합	6~10명	4~7초

응답 시간은 입력 길이·컨텍스트 길이·서버 부하에 따라 달라지므로 위 수치는 가이드라인으로만 활용하고 실측 후 확정해야 합니다. 동시 사용자 수가 5명을 초과하면 단일 워크스테이션 대신 소형 GPU 서버(A6000 1장 이상) 구성으로 전환하는 시점입니다.

부서별 독립 운용 모델

gpt-oss 20B 의 16GB 메모리 요구는 "부서별 워크스테이션 1대 + Hermes Agent + gpt-oss 20B" 형태의 부서 독립 운용 구조를 현실적으로 만들어 줍니다. 중앙 GPU 서버를 IT 부서가 단독 운영하는 방식 대신 각 부서가 자체 워크스테이션에서 Hermes Agent 인스턴스를 구동하고 필요한 스킬과 Profile 만 중앙 레포지토리에서 동기화받는 구성입니다 [S09]. 이 방식은 데이터가 부서 단위로 격리되어 민감 정보 처리에서 추가적인 보안 격리를 제공하며 중앙 서버 장애 시에도 부서 단위 업무 연속성이 유지됩니다. 다만 모델 버전 관리·보안 패치 배포를 부서별로 각자 처리해야 하는 운영 부담이 생기므로, 중앙에서 업데이트 스크립트를 배포하는 자동화 절차를 함께 설계해야 합니다.

3개 모델은 각각 다른 강점을 가지므로 단일 최적 선택이 존재하지 않습니다. 이미지·비디오·도면 처리가 핵심 요건이면 Gemma 4 31B Dense 를, 글로벌 멀티링구얼 또는 음성 자동화가 중심 업무이면 Qwen3-Omni 를, 코드 생성·수학 추론을 최소 인프라로 시작하려면 gpt-oss 20B 를 1순위 후보로 두는 것이 합리적입니다. 세 모델 모두 Apache 2.0 라이선스로 통일된 2026년 환경에서는 라이선스가 더 이상 선택의 첫 번째 분기 기준이 아니며, 사이즈 라인업 폭·컨텍스트 길이·멀티모달 범위·언어 지원 깊이가 핵심 분기 기준이 됩니다. Hermes Agent 는 LiteLLM 게이트웨이(gateway)를 통해 세 모델을 동시에 등록하고 태스크 유형별로 분기할 수 있으므로, PoC 단계에서 세 모델을 병렬로 평가한 뒤 업무 유형별 최적 모델을 확정하는 전략이 위험을 줄이는 방법입니다 [S10]. 참고로 종래 Gemma 3 27B 는 Gemma Terms of Use 를 따랐으나, Gemma 4 부터 Apache 2.0 으로 변경되었습니다(2026-04-02).

9장. 적용 사례와 유즈 케이스 — 공공·제조·금융·연구개발

Hermes Agent 는 데이터를 외부 서버로 보내지 않고도 복잡한 업무를 자동화할 수 있다는 점에서, 외부 클라우드 AI 서비스 도입에 구조적 제약이 있는 국내 조직에 실질적인 선택지가 됩니다. 공공기관의 망분리 규정, 제조 현장의 내부망 고립 환경, 금융권의 금감원 검사 대응 요건, 연구개발 부서의 특허 기밀 보호 필요 — 이 모든 조건이 오히려 Hermes Agent 의 온프레미스(On-Premises, 자체 서버 설치·운영 방식) 설계가 강점을 발휘하는 맥락입니다. 본 장은 4개 업종과 3개 부서 유형에 걸쳐 Hermes Agent 가 실제로 어떤 업무 흐름을 만들어 내는지, 그리고 각 시나리오에서 의사결정권자가 확인해야 할 KPI(Key Performance Indicator, 핵심 성과 지표)와 기대 효과가 무엇인지를 정리합니다. 독자가 자사 업종에 해당하는 시나리오를 사내 PoC(Proof of Concept, 개념 검증) 기획서에 그대로 인용할 수 있도록 구체적인 수치와 워크플로우를 함께 제시합니다.

9.1 공공기관 — 내부 문서 검색·민원 응대 초안

공공기관 AI 도입은 2026년을 기점으로 "해도 되는 것"에서 "하지 않으면 평가에서 불이익을 받는 것"으로 위상이 바뀌었습니다. 국가정보원 조사에 따르면 공공기관 10곳 중 7곳이 이미 AI를

업무에 활용하고 있으며, 2026년 기획재정부 경영평가편람 개정으로 AI 활용 항목이 독립 가점으로 신설된 상태입니다 [S12]. 그러나 외부 클라우드 AI 서비스는 여전히 망분리 규정과 개인정보보호법 충돌 문제를 해소하기 어렵습니다. Hermes Agent 는 내부망에서 Local LLM과 결합해 운영하므로 데이터가 물리적으로 기관 내부를 벗어나지 않습니다. 이 구조가 공공기관 보안 요건과 경영평가 가점 요건을 동시에 충족할 수 있는 현실적인 도입 경로입니다.

9.1.1 시나리오 — 내부 문서 검색 + 민원 응대 초안 + 일일 보고서

내부 문서 검색 — 담당자 응답 속도 단축

공공기관 담당자가 유사 민원 처리 사례, 법령 해석 문서, 내부 지침을 찾는 데 소요하는 시간은 업무 시간의 상당 부분을 차지합니다. Hermes Agent 의 RAG(Retrieval-Augmented Generation, 검색 증강 생성) 파이프라인을 내부 문서 시스템과 연결하면, 담당자가 메신저(Slack 또는 Teams)에 자연어 질의를 입력하는 방식으로 관련 문서를 즉시 추출할 수 있습니다 [S01]. 예를 들어 "2024년 이후 지방자치단체 보조금 반환 관련 처리 선례"를 입력하면, Hermes Agent 가 내부 공문·유권해석 문서를 검색하고 관련 단락을 요약해 반환합니다. 문서 접근 권한은 Hermes Agent 의 Profile(프로파일, 목적별 독립 인스턴스) 단위로 분리할 수 있어 부서별 열람 범위가 보장됩니다 [S03].

민원 응대 초안 — 작성 부담 절감

민원 응대 공문 작성은 반복성이 높고 형식 규범이 엄격한 업무입니다. 담당자가 민원 내용을 메신저에 붙여넣으면 Hermes Agent 가 사전에 학습된 공문 형식 템플릿과 관련 법령 조항을 결합해 초안을 생성합니다. 담당자는 초안을 검토·수정하는 역할만 맡으므로 작성 시간이 단축됩니다. 국내 공공 도입 자료에 따르면, 유사 워크플로우를 적용한 기관에서 공문 초안 작성 시간이 평균 60% 이상 줄었습니다 [S12]. 민원 내용 자체는 외부로 전송되지 않으므로 개인정보 처리 방침과의 충돌이 없습니다.

일일 보고서 자동 작성 — 정기 산출물 자동화

기관 내 팀장·과장급 담당자가 매일 작성하는 업무 일지, 주간 현황 보고서, 주요 사업 진행 보고서는 형식이 표준화되어 있는 반면 수집해야 할 정보 원천이 여러 시스템에 분산되어 있습니다. Hermes Agent 의 Scheduled Operations(예약 실행, 지정 시각에 자동으로 Task를 실행하는 기능)를 활용하면 매일 오전 지정 시각에 그룹웨어, 프로젝트 관리 시스템, 전자결재 시스템에서 데이터를 수집하고 보고서 초안을 자동 생성합니다 [S01]. 담당자는 출근 후 초안을 확인하고 최종 서명만 하면 됩니다.

아래 표는 공공기관 3개 시나리오별 KPI와 기대 효과를 정리한 것입니다.

시나리오	측정 KPI	기준값 (도입 전)	기대 효과
내부 문서 검색	질의당 평균 응답 시간	20~40분 (수동 검색)	2분 이내로 단축
민원 응대 초안	공문 초안 작성 시간	60~90분	15~20분 (검토·서명만)
일일 보고서	보고서 작성 시간	30~60분	5분 이내 (초안 자동 생성)

시나리오	측정 KPI	기준값 (도입 전)	기대 효과
문서 검색 정확도	관련 문서 1차 적중률	측정 안 됨	PoC 후 자체 측정 권장

PoC 기획서에 위 3가지 시나리오를 모두 담을 필요는 없습니다. 도입 효과가 가장 빠르게 측정되는 시나리오 1개를 선택해 첫 PoC를 설계하고, 6주 이내에 성과를 수치로 확인하는 방식이 사내 확산 속도를 높이는 데 효과적입니다.

9.1.2 2026 공공기관 경영평가편람 AI 가점 활용 거버넌스

2026 경영평가편람 AI 가점의 구조

2026년 기획재정부 공공기관 경영평가편람은 혁신 가점 5점 중 1.5점을 "AI 활용을 통한 생산성·서비스·안전 혁신" 항목에 배정했습니다 [S12]. 지방공기업 경영평가편람도 경영혁신 평가 배점을 2.0점에서 3.0점으로 높이면서 AI 활용이 핵심 세부 항목이 됐습니다. 1.5점은 단순한 숫자가 아닙니다. 공공기관 경영평가는 동일 유형 기관 간 상대 평가이기 때문에 이 점수 차이가 등급 변동을 만들고, 등급 변동은 임직원 성과급·차년도 예산·기관장 평가에 직접 연결됩니다. 편람은 "단순 도입 여부"가 아닌 "노력과 성과"를 평가하며, 개인정보 보호 법령 준수·AI 윤리 및 정보보안 가이드라인 준수·데이터 거버넌스 구축을 명시적으로 요구합니다 [S12].

Hermes Agent 도입이 가점 요건과 맞닿는 지점

편람이 요구하는 보안·윤리·거버넌스 요건은 Hermes Agent의 설계 방향과 자연스럽게 맞아떨어집니다. 첫째, 데이터가 기관 내부 서버에서만 처리되므로 개인정보 외부 유출 위험이 구조적으로 차단됩니다. 둘째, Profile 단위 접근 제어와 Kanban(칸반, 작업 상태를 시각적으로 관리하는 Task 보드) 기반 작업 이력 기록이 감사 자료로 활용 가능합니다 [S03]. 셋째, MIT 라이선스 오픈소스이므로 소스 코드 전수 검토가 가능해 보안 검증 측면에서 추가 신뢰를 확보할 수 있습니다. MSAP.ai 기반 사내 AI 게이트웨이를 함께 구성하면 N2SF(National Network Security Framework, 국가 망 보안 체계) 정합성 확보와 망분리 환경 도입 부담을 함께 완화할 수 있습니다.

가점 취득을 위한 거버넌스 5단계

편람 요건을 충족하는 거버넌스를 구성하려면 다섯 단계를 순서대로 진행해야 합니다. 첫 번째는 활용 시나리오 정의로, 어떤 업무에 AI를 적용하고 어떤 성과를 측정할지 문서화합니다. 두 번째는 보안·통제 체계 수립으로, 데이터 분류 기준과 접근 권한 매트릭스를 Profile 단위로 설정합니다 [S03]. 세 번째는 정량 성과 측정으로, KPI를 사전에 정의하고 6주 단위로 수집합니다. 네 번째는 N2SF 정합성 검토로, 기관이 속한 보안 등급(C·S·O)에 따라 아키텍처 설계를 조정합니다. 다섯 번째는 외부 검증으로, 감사 로그와 성과 자료를 제3자 확인 가능한 형태로 보관합니다. 이 다섯 단계를 분기별 로드맵으로 구성하면 경영평가 제출 자료가 자연스럽게 축적됩니다 [S12].

아래 표는 2026 경영평가편람 AI 가점 요건과 Hermes Agent 도입 요소의 대응 관계를 정리한 것입니다.

경영평가편람 요건	Hermes Agent 대응 요소
개인정보 보호 법령 준수	온프레미스 처리 — 데이터 외부 미전송

경영평가편람 요건	Hermes Agent 대응 요소
AI 윤리·정보보안 가이드라인	MIT 라이선스 소스 공개 + 보안 검토 가능
데이터 거버넌스 구축	Profile 접근 제어 + Kanban 작업 이력
정량 성과 측정	시나리오별 KPI 사전 정의 (위 표 참조)
외부 검증 가능 자료	Kanban SQLite 기반 감사 로그 보관

미도입 기관은 상대 평가에서 사실상 감점 효과를 받습니다. 2026년 평가 결과가 차년도 예산 편성에 반영된다는 점을 감안하면, 가점 취득 여부는 IT 예산 확보와 직결되는 의사결정 사안입니다.

9.2 제조·금융·연구개발 시나리오

제조·금융·연구개발 세 업종은 데이터 외부 유출 금지라는 공통 제약을 갖고 있으면서도 업무 워크플로우와 AI 활용 목적이 다릅니다. 제조는 현장 작업자의 즉각적인 기술 지원, 금융은 내부 보고서 작성 자동화와 감사 대응, 연구개발은 논문·특허 정보의 체계적 축적이 핵심 요구사항입니다. Hermes Agent 는 각 업종의 특수 요건에 맞춰 Profile과 연결 모델을 조합하는 방식으로 대응합니다.

9.2.1 제조 — 매뉴얼 기반 현장 지원과 OCR 결합

현장 작업자의 기술 질의 흐름

제조 현장에서 작업자가 설비 이상 증상을 처음 마주쳤을 때 취하는 행동은 경험 많은 선임자에게 전화하거나 두꺼운 매뉴얼 PDF를 뒤지는 것입니다. 선임자가 자리를 비웠거나 매뉴얼 분량이 방대한 경우 문제 해결이 지연되고 라인이 멈춥니다. Hermes Agent 는 이 흐름을 바꿉니다. 작업자가 사내 메신저 채널에 "3호기 유압 펌프 압력 저하 — 에러코드 P0342"를 입력하면 Hermes Agent 가 내부 매뉴얼 RAG와 설비 이력 데이터베이스를 동시에 조회해 원인 추정·점검 절차·교체 부품 코드를 한 번에 반환합니다 [S01]. 응답은 30초 이내이며 현장 언어에 가까운 평어로 출력됩니다.

Gemma 4 31B Dense 멀티모달 — 도면·이미지 직접 처리

텍스트 매뉴얼만으로 해결되지 않는 경우가 있습니다. 작업자가 스마트폰으로 찍은 설비 사진이나 도면 이미지를 첨부하면 Gemma 4 31B Dense(파라미터 27B, 컨텍스트 128K, 멀티모달 지원)의 이미지 인식 기능이 부품 위치와 연결부를 직접 분석합니다 [S07]. OCR(Optical Character Recognition, 광학 문자 인식)을 결합하면 도면에 인쇄된 부품 코드·규격 텍스트를 자동 추출하고, 해당 코드로 재고 시스템을 조회해 보유 여부까지 알려줍니다. 제조 현장의 AI 도입에서 "모바일 인터페이스로 사진을 찍어 질문할 수 있는가"는 작업자 수용도를 결정하는 핵심 요소입니다.

현장 지원 워크플로우와 기대 효과

아래 표는 Hermes Agent 기반 제조 현장 지원 워크플로우와 각 단계의 처리 주체를 정리한 것입니다.

단계	입력	처리 주체	출력
1. 질의 접수	작업자 → 사내 메신저 (텍스트 + 사진)	Hermes Agent	질의 파싱·라우팅
2. 문서 검색	매뉴얼 PDF·설비 이력 DB	RAG 파이프라인	관련 단락 추출
3. 이미지 분석	첨부 사진·도면	Gemma 4 31B Dense 멀티모달	부품 코드·이상 위치
4. 재고 조회	부품 코드	ERP API	보유 여부·위치
5. 응답 반환	—	Hermes Agent	점검 절차 + 재고 정보

제조 현장 PoC의 성과 기준은 "평균 문제 해결 시간 단축율"로 설정하는 것이 가장 측정하기 쉽습니다. 라인 다운 시간이 10분 줄어들 때마다 생산 비용 절감 효과가 직접 수치로 나타나기 때문입니다.

9.2.2 금융 — 보고서 작성 자동화 (외부 송신 금지 환경)

금융 망분리 환경의 구조적 제약

금융 회사는 금융위원회·금감원 규정에 따라 업무망과 인터넷망을 분리하여 운영합니다. 고객 데이터, 거래 정보, 내부 심사 보고서는 외부 네트워크로 전송할 수 없습니다. 이 환경에서 외부 클라우드 AI 서비스를 사용하려면 데이터 가공·익명화·재식별 방지 검증 절차를 모두 거쳐야 하는데, 이 과정이 복잡해 실무 도입이 지연되는 경우가 많습니다. Hermes Agent 는 업무망 내부 서버에 설치하고 Local LLM을 결합하므로 이 문제를 우회합니다. 데이터가 업무망 밖으로 나가지 않는 구조 자체가 규정 준수의 근거가 됩니다 [S12].

시장팀·심사팀·CS팀 Profile 격리

금융 회사에서 AI 접근 권한 통제는 단순한 보안 요건이 아닙니다. 시장 동향 분석을 담당하는 시장팀, 기업 여신을 심사하는 심사팀, 고객 응대를 담당하는 CS팀은 각각 열람할 수 있는 데이터와 생성할 수 있는 보고서 유형이 다릅니다. Hermes Agent 의 Profile 구조를 활용하면 팀별로 독립된 메모리·도구·문서 접근 범위를 설정할 수 있습니다 [S03]. 시장팀 Profile 은 시장 데이터와 공시 정보에만 접근하고, 심사팀 Profile 은 내부 여신 규정 문서에만 접근하는 방식입니다. 팀 간 데이터 혼재는 구조적으로 발생하지 않습니다.

감사 로그 — 금감원 검사 대응

금감원 검사 시 AI 활용 내역 제출 요구가 증가하는 추세입니다. Hermes Agent 의 Kanban 보드는 SQLite 기반 영속 저장소에 모든 작업 이력을 남깁니다 [S03]. 누가 어떤 Profile 로 어떤 문서를 기반으로 어떤 보고서 초안을 생성했는지 타임스탬프와 함께 기록됩니다. 이 기록은 AI 활용 감사 증빙으로 즉시 활용 가능합니다. 아래 표는 금융 업종 Profile 권한 매트릭스를 정리한 것입니다.

Profile	접근 가능 데이터	생성 가능 산출물	격리 범위
시장팀	공시 정보·시장 데이터	시장 동향 보고서·Daily 브리핑	심사·CS 데이터 차단
심사팀	여신 규정·내부 심사 기준	심사 의견서 초안·리스크 요약	시장·CS 데이터 차단
CS팀	상품 안내·FAQ	고객 응대 메모·스크립트	시장·심사 데이터 차단

금융 PoC에서 첫 번째 시나리오로 가장 적합한 것은 시장팀의 Daily 브리핑 자동 작성입니다. 형식이 표준화되어 있고 결과물의 품질 평가가 쉬우며, 외부 데이터 의존도가 낮아 망분리 환경에서도 즉시 구현 가능합니다.

9.2.3 연구개발 — 논문·특허 리뷰와 사내 지식 축적

R&D 부서의 정보 과부하 문제

연구개발 부서 연구원이 매주 쏟아지는 논문과 특허 공개 건수를 모두 읽을 수 없습니다. arXiv에는 하루 평균 수백 편의 AI·소재·바이오 논문이 게재되고, 국내외 특허청에는 매일 수천 건의 특허가 출원됩니다. 핵심 연구 흐름을 놓치지 않으려면 자동화된 모니터링과 요약이 필수입니다. Hermes Agent 의 Scheduled Operations와 RAG 파이프라인을 결합하면 매일 지정 시각에 특정 키워드·연구자·기관의 논문과 특허를 수집하고 한국어 요약을 생성해 연구팀 채널에 자동 배포할 수 있습니다 [S01].

Qwen3 의 119개 언어 지원 — 다국적 자료 처리

R&D 자료는 영어가 압도적으로 많지만, 중국어 특허와 일본어 기술 보고서도 무시할 수 없는 비중을 차지합니다. Qwen3(Apache 2.0 라이선스, 0.6B~235B 파라미터 범위, 119개 언어 지원)을 Local LLM으로 선택하면 영어·중국어·일본어·한국어를 단일 모델로 처리할 수 있습니다 [S08]. 연구원이 영어 논문을 한국어로 번역 요청하거나, 중국어 특허 청구항을 한국어 요약으로 변환하는 작업이 추가 도구 없이 동일한 Hermes Agent 인터페이스에서 처리됩니다.

Archive Loop — 사내 지식 자산화

Hermes Agent 의 Archive Loop(아카이브 루프, 작업 결과를 장기 메모리로 축적하는 자기 개선 사이클)는 R&D 부서에 특히 유용합니다. 논문 요약, 특허 분석, 기술 비교 보고서가 매일 Archive에 누적되면 6개월 후에는 특정 기술 분야의 연구 흐름 전체를 질의할 수 있는 내부 지식 베이스가 형성됩니다 [S02]. 신규 연구원이 입사했을 때 기존 연구 맥락을 빠르게 파악하는 데도 활용할 수 있습니다. 5년간 누적되면 해당 기술 분야에 대한 기관 고유의 지식 자산이 되며, 외부로 반출할 수 없는 경쟁 우위 자료가 됩니다.

아래 표는 R&D 시나리오의 단계별 워크플로우를 정리한 것입니다.

단계	입력	처리	출력
1. 자료 수집	키워드·저자·기관	Scheduled Operations	논문·특허 목록

단계	입력	처리	출력
2. 요약·번역	원문 (영·중·일·한)	Qwen3 (119개 언어)	한국어 요약
3. 배포	—	Hermes → 메신저 채널	팀 Daily 브리핑
4. 축적	요약 결과	Archive Loop	사내 지식 베이스
5. 재질의	연구원 질의	RAG → Archive	맥락 포함 응답

R&D PoC의 첫 번째 측정 지표는 "연구원 1인당 주간 논문 검토 건수 증가율"로 설정하는 것이 효과적입니다. Archive 축적 가치는 6개월 이후부터 측정 가능하므로, 초기 PoC는 자료 수집·요약 단계의 시간 절감 효과에 집중합니다.

9.3 누가 · 어디에 주로 사용하는가 — 부서별 패턴 분석

업종과 무관하게 Hermes Agent 도입 효과가 가장 먼저 나타나는 부서 유형이 있습니다. 영업·CS는 반복 문서 작업이 많고, HR·총무는 사내 정책 질의 응대 부담이 크며, IT·보안은 운영 자동화 요구가 명확합니다. 세 유형 모두 초기 PoC에 적합한 이유가 있으며, 각 부서가 Hermes Agent 의 사내 확산에서 어떤 역할을 하는지도 다릅니다. 본 절은 업종을 가로질러 반복적으로 나타나는 부서별 도입 패턴을 정리합니다.

9.3.1 영업·CS — 제안서 초안과 응대 스크립트 자동화

영업 부서 — 제안서·견적서 초안 자동화

영업 담당자가 고객 미팅 전 준비하는 제안서 초안에는 공통 구조가 있습니다. 고객 현황 요약, 문제 정의, 솔루션 제안, 기대 효과, 가격 구성이 반복되며 회사마다 형식 템플릿이 있습니다. Hermes Agent 에 이 템플릿과 과거 수주 제안서 데이터를 연결하면 담당자가 고객명과 핵심 요구사항만 입력해도 초안이 생성됩니다 [S01]. 담당자는 내용을 수정하고 고객 맥락을 추가하는 데 집중합니다. 제안서 작성 시간을 2시간에서 30분으로 줄이는 것이 현실적인 첫 번째 PoC 목표입니다.

CS 부서 — 응대 스크립트·티켓 분류 자동화

CS 부서의 반복 업무는 두 가지입니다. 첫째는 자주 접수되는 문의에 대한 응대 스크립트 작성, 둘째는 접수된 티켓을 카테고리·우선순위·담당팀으로 분류하는 작업입니다. Hermes Agent 는 과거 티켓 데이터와 응대 이력을 학습해 신규 티켓의 분류를 자동 제안하고, 과거 우수 응대 사례를 기반으로 스크립트 초안을 생성합니다 [S01]. 상담원은 분류 결과를 확인하고 스크립트를 고객 상황에 맞게 조정하는 역할만 합니다. 아래 표는 영업·CS 부서 PoC 시나리오 매트릭스입니다.

부서	시나리오	측정 KPI	PoC 기간
영업	제안서 초안 자동 작성	작성 시간 단축율	4주
영업	견적서 자동 생성	견적 오류율 감소	4주

부서	시나리오	측정 KPI	PoC 기간
CS	티켓 자동 분류	분류 정확도 (%)	6주
CS	응대 스크립트 자동 생성	응대 시간 단축	6주

영업 부서 PoC를 시작할 때 가장 좋은 방법은 제안서 1건을 Hermes Agent 로 자동 생성하고 담당자가 수작업으로 완성하는 흐름으로 설계하는 것입니다. 첫 번째 제안서에서 발견된 오류 패턴을 수정하면 두 번째부터 품질이 빠르게 향상됩니다.

9.3.2 HR·총무 — 사내 정책 Q&A 와 신청서 자동 응대

HR 부서 — 사내 정책 질의 자동 응대

인사 부서가 매일 받는 질의의 70% 이상은 반복적입니다. 연차 계산 방식, 육아휴직 신청 절차, 교육 지원 기준, 복지 포인트 사용 범위 — 이 질문들은 HR 담당자가 매번 같은 답변을 작성하거나 같은 내부 문서 링크를 안내하는 형태로 처리됩니다. Hermes Agent 를 사내 HR 정책 문서와 연결하면 직원이 메신저에 질의했을 때 즉시 정확한 답변과 관련 서식을 함께 받습니다 [S01]. HR 담당자는 정책 해석이 필요한 복잡한 케이스에 집중할 수 있습니다. 6개월 누적된 질의 이력은 Archive Loop 를 통해 자주 묻는 질문 상위 100건을 자동 도출하고 정책 개선 근거로 활용할 수 있습니다.

총무 부서 — 신청서·예약 자동화

총무 부서는 비품 신청, 회의실 예약, 법인카드 신청, 출장 신청 등 반복 처리 업무 비중이 높습니다. Hermes Agent 에 그룹웨어 API와 ERP 시스템을 연결하면 직원이 메신저에 "내일 오후 2시 10인 회의실 예약"을 입력하는 것만으로 예약이 처리됩니다 [S01]. 신청서 양식 안내, 승인자 알림, 결과 통보까지 하나의 대화 흐름으로 완결됩니다. 아래 표는 HR·총무 부서 PoC 시나리오 매트릭스입니다.

부서	시나리오	측정 KPI	PoC 기간
HR	사내 정책 Q&A 자동 응대	담당자 응대 시간 절감	4주
HR	자주 묻는 질문 자동 도출	정책 문서 업데이트 주기	3개월
총무	회의실 예약 자동화	예약 처리 오류율 감소	4주
총무	비품·법인카드 신청 자동화	처리 시간 단축율	6주

HR 부서 PoC에서 성과 측정이 가장 빠른 시나리오는 사내 정책 Q&A 자동 응대입니다. 질의 건수, 자동 응답률, 담당자 에스컬레이션 비율을 주 단위로 측정하면 4주 안에 효과를 수치로 확인할 수 있습니다.

9.3.3 IT·보안 — 사내 운영 자동화와 보안 모니터링

IT 부서 — 서버 운영 자동화

IT 운영 담당자의 일상 업무에는 반복적인 점검 항목이 많습니다. 서버 상태 확인, 디스크 사용량 모니터링, 패치 적용 현황 점검, 백업 완료 확인이 매일 반복됩니다. Hermes Agent 의 Scheduled Operations를 활용하면 매일 지정 시각에 이 점검 작업을 자동 실행하고 이상 항목만 담당자에게 알림으로 전달합니다 [S01]. 평상시에는 담당자가 개입하지 않고, 이상 징후가 발생했을 때만 확인하는 방식입니다. 점검 이력은 Kanban 보드에 자동 기록되므로 감사 증빙으로도 활용됩니다 [S03].

보안 부서 — SIEM 로그 분석과 위협 요약

SIEM(Security Information and Event Management, 보안 정보 이벤트 관리) 시스템은 매일 수십만 건의 보안 이벤트 로그를 생성합니다. 보안 담당자가 이 로그를 전수 검토하는 것은 현실적으로 불가능합니다. Hermes Agent 에 SIEM API를 연결하고 Qwen3 또는 gpt-oss 20B(Apache 2.0 라이선스, 21B 파라미터, MoE 구조)를 Local LLM으로 연결하면 로그를 자동 분류하고 위협 우선순위를 요약해 담당자에게 전달합니다 [S09]. 담당자는 상위 10건의 위협 요약만 검토하고, 확인이 필요한 항목만 직접 조사합니다. 아래 표는 IT·보안 부서 PoC 시나리오 매트릭스입니다.

부서	시나리오	측정 KPI	PoC 기간
IT	서버 상태 자동 점검·알림	수동 점검 시간 절감	4주
IT	패치·백업 현황 자동 보고	보고 작성 시간 단축	4주
보안	SIEM 로그 자동 분류·요약	담당자 검토 시간 절감	6주
보안	위협 우선순위 자동 산정	탐지 대응 시간 단축	8주

IT 부서를 Hermes Agent 의 첫 사내 사용자로 지정하면 사내 확산 속도가 빨라집니다. IT 담당자는 운영 중 발생하는 기술 문제를 직접 해결하면서 시스템 작동 원리를 빠르게 파악하고, 다른 부서 PoC 설계 시 기술 지원 역할을 맡을 수 있습니다. IT 부서의 자체 도입 성공 사례가 사내에서 가장 설득력 있는 확산 근거가 됩니다.

10장. Gemma 4 31B Dense 활용 업무 진행 가이드 — Step-by-Step

Gemma 4 31B Dense 와 Hermes Agent 를 처음 연결하는 작업은 크게 네 단계로 이루어집니다. 모델 가중치를 내려받아 서빙 환경을 구성하고, Hermes 에 LLM 공급자를 등록하며, 첫 번째 Skill 을 작성하고, 운영 지표를 추적하는 순서입니다. 각 단계는 독립적으로 완료할 수 있도록

명령어와 설정 예시를 함께 제시합니다. 본 장을 PoC 1주차 체크리스트로 삼으면 팀 내 공유와 검증이 한결 쉬워집니다.

Gemma 4 31B Dense 는 31B 파라미터, Apache 2.0 라이선스, 256K 컨텍스트 창을 갖추며 140개 이상 언어와 텍스트 + 이미지 + 비디오 멀티모달 입력을 지원합니다 [S07]. 라이선스는 Apache 2.0 이라 사내 수정·재배포·상업적 활용에 별도 동의 게이트가 없어 법무 검토가 단순화됩니다. 하드웨어 최소 요건은 Q4_K_M 양자화 기준 18~26GB VRAM 이며, GPU 서빙 시 RTX 4090 단일 카드로 운영 가능합니다 [S07]. 운영 단계에서 256K 컨텍스트를 그대로 활용하면 장시간 회의록, 장문 계약서, 분기 보고서 같은 대용량 자료를 한 번에 처리할 수 있어 사내 업무에 직접적인 효용이 발생합니다.

10.1 모델 다운로드와 서빙 환경 구성

모델 서빙 환경은 PoC 단계와 본격 운영 단계를 구분해 구성합니다. PoC 에서는 Ollama 로 단일 PC 에서 빠르게 동작을 확인하고, 검증이 끝나면 vLLM 으로 전환해 다중 사용자 처리량과 운영 안정성을 확보합니다. 두 경로 모두 OpenAI 호환 endpoint 를 노출하므로 Hermes Agent 쪽 설정은 동일하게 유지됩니다.

10.1.1 Hugging Face 에서 Gemma 4 31B 가중치 다운로드

Apache 2.0 라이선스의 다운로드 절차 간소화

Gemma 4 31B Dense 가중치는 Hugging Face 모델 허브 (`huggingface.co/google/gemma-4-31b-it`) 에 공개되어 있습니다 [S07]. Apache 2.0 라이선스라 별도 동의 게이트 없이 즉시 가중치 다운로드가 가능합니다. Hugging Face 계정과 읽기 전용 Access Token 만 발급받으면 됩니다. 계정 생성 후 Settings → Access Tokens 에서 토큰을 발급합니다. 이 단계는 사내 절차서에 1페이지로 정리해 두면 팀원 온보딩 시 재작업을 줄일 수 있습니다.

CLI 설치와 다운로드 명령

Hugging Face CLI 를 설치하고 로그인한 뒤 모델을 내려받습니다. 31B 전체 가중치는 약 62GB (FP16 기준) 이므로 네트워크 속도와 저장 공간을 미리 확인합니다. Q4_K_M 양자화 본은 18~26GB 수준입니다.

```
 # huggingface_hub 설치
pip install -U huggingface_hub

# 로그인 (발급받은 토큰 입력)
huggingface-cli login

# Q4_K_M 양자화 버전 다운로드 (약 18~26GB — RTX 4090 단일 카드 권장)
huggingface-cli download ₩
  google/gemma-4-31b-it ₩
  --include "*.gguf" ₩
  --local-dir ./models/gemma4-31b
```

Q4_K_M 양자화 (4-bit 양자화, K-means 방식) 를 사용하면 모델 크기가 원본의 약 1/3~1/2 수준인 18~26GB 안팎으로 줄어들어 RTX 4090 단일 GPU 환경에서 서빙이 가능합니다 [S07]. 품질 손실은 수학·코딩 추론 영역에서 약 2~3% 내외로 알려져 있으며, 한국어 요약·분류 같은 일반 업무에서는 체감 차이가 거의 없습니다. FP16 본은 약 62GB 이므로 A100 80GB 단일 카드 또는 H100 환경에서 사용합니다.

다운로드 완료 체크리스트

단계	확인 항목	비고
1	Hugging Face 계정 생성 및 Access Token 발급	읽기 전용 토큰으로 충분
2	라이선스 확인 (Apache 2.0)	별도 동의 단계 없음
3	저장 공간 확인 (Q4_K_M 기준 30GB 여유)	SSD 권장
4	`huggingface-cli download` 완료 후 파일 체크섬 확인	`.gguf` 파일 존재 여부

10.1.2 Ollama 로 단일 PC PoC 서빙

Ollama 설치와 모델 실행

Ollama 는 로컬 LLM 서빙을 위한 CLI 도구로, 설치 직후 `ollama run` 한 줄로 모델을 내려받고 대화형 세션을 시작할 수 있습니다. PoC 에서 검증 속도가 가장 빠른 경로입니다. Gemma 4 지원은 Ollama 0.8 이상 버전이 필요합니다.

```
 # Ollama 설치 (Linux / macOS)
curl -fsSL https://ollama.com/install.sh | sh

# Gemma 4 31B Dense 모델 다운로드 및 서빙 시작
ollama run gemma4:31b

# 백그라운드 서버로만 실행 (API 전용)
ollama serve &

# OpenAI 호환 endpoint 동작 확인
curl http://localhost:11434/v1/chat/completions ₩
  -H "Content-Type: application/json" ₩
  -d '{
    "model": "gemma4:31b",
    "messages": [{"role": "user", "content": "안녕하세요. 자기소개를 해주세요."}]
  }'
```

`http://localhost:11434/v1` endpoint 는 OpenAI API 와 호환되므로, Hermes Agent 에서 base URL 만 교체하면 그대로 연결됩니다. Ollama 는 NVIDIA · AMD GPU 를 자동 감지하고, GPU 가 없는

경우 CPU 추론으로 대체합니다. CPU 추론 시 31B 모델의 첫 토큰 생성까지 40~80초가 소요될 수 있으므로 PoC 시연 환경에는 GPU 장착 PC 를 사용합니다. RTX 4090 단일 카드 기준 첫 토큰 응답 1~2초 수준이 정상 범위입니다.

PoC 단계 제약과 전환 시점

Ollama 는 단일 사용자 동시 요청을 처리하는 구조로 설계되어 있어, 여러 팀원이 동시에 요청을 보내면 큐 대기가 발생합니다. PoC 1~3주차에는 1~3명 규모에서 기능 검증 목적으로만 사용하고, 4주차 이후 팀 전체 사용으로 확장할 시점에 vLLM 으로 전환하는 일정을 잡아두는 것이 좋습니다. 256K 컨텍스트를 적극 활용해 장문 회의록 요약·계약서 비교 같은 시나리오를 1주차 시연 자료로 준비하면 도입 의사결정에 직접 활용됩니다.

10.1.3 vLLM 으로 본격 운영 서빙 전환

vLLM 의 처리량 이점

vLLM 은 PagedAttention 기술로 GPU 메모리를 비연속 블록 단위로 관리합니다. KV 캐시 단편화를 줄여 동일 GPU 에서 HuggingFace `generate()` 대비 10~25배 높은 처리량을 냅니다. 2×A100 80GB 구성에서 Gemma 4 31B Dense 기준 동시 40~80 요청을 평균 2~3초 응답시간으로 처리할 수 있습니다. 다중 사용자가 동시에 Hermes Agent 를 통해 모델을 호출하는 운영 환경에서는 vLLM 이 사실상 표준 서빙 엔진입니다 [S07].

Docker 로 vLLM 서빙 시작

```
 # vLLM 공식 컨테이너 실행 (NVIDIA GPU 필수)
docker run --gpus all ₩
  -v ./models/gemma4-31b:/models ₩
  -p 8000:8000 ₩
  vllm/vllm-openai:latest ₩
  --model /models ₩
  --served-model-name gemma4-31b ₩
  --gpu-memory-utilization 0.9 ₩
  --max-model-len 65536 ₩
  --tensor-parallel-size 1

# 서버 동작 확인
curl http://localhost:8000/v1/models

# 추론 요청 테스트
curl http://localhost:8000/v1/chat/completions ₩
  -H "Content-Type: application/json" ₩
  -d '{
    "model": "gemma4-31b",
    "messages": [{"role": "user", "content": "오늘 회의 요약을 작성해 주세요."}],
    "stream": false
  }'
```

`--tensor-parallel-size` 값을 GPU 개수에 맞게 조정하면 멀티 GPU 분산 서빙이 활성화됩니다. `--max-model-len 65536` 은 메모리 여유가 부족한 환경에서 컨텍스트 창을 64K 로 제한해 OOM (메모리 초과) 오류를 방지합니다. 256K 컨텍스트를 풀로 활용하려면 `--max-model-len 262144` 로 상향하되, A100 80GB × 2 이상 구성을 권장합니다. 운영 환경에서는 Prometheus 메트릭 수집을 함께 구성해야 응답 시간 추이를 추적할 수 있습니다 [S10].

256K 컨텍스트 활용 시나리오

Gemma 4 31B Dense 의 256K 컨텍스트는 사내 업무에서 즉시 가치를 발휘하는 차별점입니다. 1시간 분량 회의록 (약 12,000 단어) 5~10건을 한 번에 입력해 분기 동향 보고를 작성하거나, 50페이지 분량 계약서 2건을 동시에 비교하여 변경 조항을 추출하는 작업이 단일 호출로 가능합니다. 컨텍스트 256K 활용 시 응답 지연이 평균 8~15초로 증가하므로, 사용자에게 "분석 중" 표시를 노출하는 UX 보조 장치를 함께 설계합니다.

Ollama vs vLLM 비교

항목	Ollama (PoC)	vLLM (본격 운영)
설치 난이도	매우 낮음 (단일 curl)	보통 (Docker + GPU 드라이버)
동시 요청 처리	순차 큐 (1~3명)	연속 배치 (40~80 동시)
처리량	기준 1×	HF generate 대비 10~25×
스트리밍	지원	지원
256K 컨텍스트	지원 (단일 사용자)	지원 (A100 80GB×2 권장)
OpenAI 호환	지원 (포트 11434)	지원 (포트 8000)
운영 부담	낮음	중간 (컨테이너 관리 필요)
전환 시점 권고	PoC 1~3주차	PoC 종료 후 4주차~

10.2 Hermes Agent 등록과 첫 Skill 작성

서빙 환경이 준비되면 Hermes Agent 에 LLM 공급자를 등록하고 첫 번째 Skill 을 작성합니다. 등록 절차는 `hermes setup` 대화형 마법사를 통해 진행하며, Skill 은 SKILL.md 파일 하나로 완성됩니다. 두 작업 모두 코드 수정 없이 설정 파일과 마크다운만으로 이루어지므로, 개발자가 아닌 IT 담당자도 직접 수행할 수 있습니다 [S01].

10.2.1 `hermes setup` 으로 LLM 공급자 등록

hermes setup 대화형 마법사

Hermes Agent 는 초기 설정을 `hermes setup` 명령으로 진행합니다. 명령을 실행하면 터미널에서 공급자 종류, endpoint URL, 모델 이름, API 키를 순서대로 입력하는 대화형 화면이 나타납니다. 모든 설정은 `~/.hermes/config.yaml` 에 저장됩니다 [S01].

```
# Hermes Agent 설치 (MIT 라이선스, Linux / macOS / WSL2)
curl -fsSL https://install.hermes-agent.org | bash

# LLM 공급자 등록 마법사 실행
hermes setup
```

마법사에서 Ollama 또는 vLLM endpoint 를 등록하는 입력값 예시는 다음과 같습니다.

```
? Select provider type:  › OpenAI-compatible (self-hosted)
? API base URL:          › http://localhost:11434/v1   # Ollama
                         # 또는 http://localhost:8000/v1  (vLLM)
? Model name:            › gemma4:31b
? API key (optional):    › (공백 입력 — 로컬 서버는 불필요)
? Context window (tokens): › 65536
? Save as provider name: › gemma4-local
```

설정 파일 직접 편집

대화형 마법사 대신 config.yaml 을 직접 수정하는 방식도 지원됩니다. 사내 매뉴얼에 아래 YAML 블록을 박제해두면 팀원이 동일 설정을 반복 없이 복사해 사용할 수 있습니다 [S01].

```
# ~/.hermes/config.yaml
providers:
  gemma4-local:
    type: openai-compatible
    base_url: "http://localhost:8000/v1"   # vLLM 운영 endpoint
    model: "gemma4-31b"
    context_window: 65536
    api_key: ""                            # 로컬 서버는 불필요

default_provider: gemma4-local
```

설정 저장 후 `hermes doctor` 를 실행하면 공급자 연결 상태를 자동 점검합니다. "✓ gemma4-local reachable" 메시지가 출력되면 등록이 완료된 것입니다. 64,000 토큰 이상의 컨텍스트 창을 확보해야 Hermes 의 도구 스키마와 대화 이력이 정상 작동하므로, `context_window` 값을 65,536 미만으로 낮추지 않습니다 [S01]. 256K 컨텍스트를 풀로 활용할 경우 `context_window: 262144` 로 상향합니다.

입력값 체크리스트

항목	Ollama (PoC)	vLLM (운영)	비고
provider type	openai-compatible	openai-compatible	
base_url	`http://localhost:11434/v1`	`http://localhost:8000/v1`	원격 서버는 IP 로 교체

항목	Ollama (PoC)	vLLM (운영)	비고
model	`gemma4:31b`	`gemma4-31b`	served-model-name 과 일치 필요
api_key	빈 값	빈 값	인증 없는 로컬 서버
context_window	65536	65536 (또는 262144)	최솟값 64000

10.2.2 LiteLLM 프록시 경유로 통합 비용 추적

직접 연결 대신 LiteLLM 프록시를 쓰는 이유

Hermes 에서 vLLM endpoint 를 직접 연결해도 동작하지만, 팀 단위로 사용하면 누가 얼마나 토큰을 사용했는지 추적하기 어렵습니다. LiteLLM 프록시를 중간에 두면 virtual key 발급, 팀별 예산 상한, 사용량 대시보드, 모델 fallback 구성을 단일 지점에서 관리할 수 있습니다 [S10]. MIT 라이선스 OSS 버전만으로도 PoC 1주차부터 이 기능을 활용할 수 있습니다.

MSAP.ai 는 LiteLLM 기반 통합 게이트웨이에 사내 비용 추적 대시보드를 사전 구성해 제공하므로, 자체 LiteLLM 설치 없이 동일 기능을 이용할 수 있습니다.

LiteLLM 프록시 구성과 시작

```
 # LiteLLM 설치
pip install litellm[proxy]

# PostgreSQL 연결이 없는 PoC용 SQLite 모드 시작
litellm --model ollama/gemma4:31b ₩
        --port 4000 ₩
        --alias gemma4-proxy
```

팀 운영 단계로 전환할 때는 PostgreSQL 백엔드와 가상 키를 활성화한 설정 파일로 전환합니다.

```
 # litellm_config.yaml (팀 운영 단계)
model_list:
  - model_name: gemma4-31b
    litellm_params:
      model: openai/gemma4-31b
      api_base: "http://vllm-server:8000/v1"
      api_key: ""

general_settings:
  master_key: "sk-your-master-key-here"
  database_url: "postgresql://user:pass@db:5432/litellm"

litellm_settings:
  success_callback: ["langfuse"]   # 선택 — 상세 로그 외부 수집
```

가상 키를 발급하면 팀원별로 개별 키를 부여하고, 각 키의 일별·월별 토큰 사용량과 추정 비용이 LiteLLM 대시보드 (http://localhost:4000/ui) 에 자동 집계됩니다 [S10]. PoC 4주차에 이 대시보드 스크린샷을 의사결정 회의 자료로 제출하면 도입 근거 수치를 직접 보여줄 수 있습니다.

Hermes 에서는 config.yaml 의 base_url 만 LiteLLM 프록시 주소로 교체합니다.

```
# ~/.hermes/config.yaml — LiteLLM 프록시 경유
providers:
  gemma4-proxy:
    type: openai-compatible
    base_url: "http://localhost:4000/v1"
    model: "gemma4-31b"
    api_key: "sk-team-member-virtual-key"
    context_window: 65536

default_provider: gemma4-proxy
```

직접 연결 vs LiteLLM 프록시 비교

항목	직접 연결	LiteLLM 프록시 경유
설정 복잡도	낮음	중간
팀원별 사용량 추적	불가	가상 키별 집계
예산 상한 설정	불가	키/팀 단위 설정 가능
모델 fallback	불가	다단계 fallback 설정 가능
대시보드	없음	내장 UI 제공
PoC 1주차 적용	가능	권장

10.2.3 첫 Skill 작성 — "일일 회의록 요약" 사례

SKILL.md 구조와 자동 인식 방식

Hermes Agent 는 시작 시 skills/ 디렉터리 (프로젝트 수준) 와 ~/.hermes/skills/ (전역 수준) 를 자동으로 탐색해 SKILL.md 파일을 로드합니다 [S03]. SKILL.md 는 YAML frontmatter 와 마크다운 본문으로 구성됩니다. frontmatter 의 description 필드에 적힌 내용을 Hermes 가 사용자 요청과 대조해 실행할 Skill 을 선택하므로, 이 필드에 실제 사용자가 입력할 법한 표현을 충분히 적어두어야 매칭 정확도가 높아집니다 [S03].

Skill 파일은 개발 지식 없이도 작성할 수 있습니다. 첫 Skill 을 PoC 1주차 산출물로 지정해 팀원 한 명이 직접 작성하는 경험을 갖도록 하면 이후 Skill 확산의 동력이 됩니다.

"일일 회의록 요약" SKILL.md 전체 예시

아래는 "일일 회의록 요약" 업무를 Hermes 에게 위임하는 첫 번째 Skill 전체 예시입니다. skills/meeting-summary/ 디렉터리를 생성한 뒤 SKILL.md 로 저장합니다.

```
 ---
name: daily-meeting-summary
description: >
  Use when the user asks to summarize today's meeting notes, create a meeting
  summary, recap a standup, or extract action items from a meeting.
  한국어로 요청할 경우: "오늘 회의 요약", "회의록 정리", "액션 아이템 추출",
  "스탠드업 요약" 등의 표현에 반응합니다.
version: "1.0.0"
author: "팀명 또는 담당자명"
tags:
  - meeting
  - summary
  - korean
---

# 일일 회의록 요약 Skill

## 역할

당신은 회의록 요약 전문가입니다. 사용자가 제공하는 회의 내용(텍스트, 노트, 대화록)을 읽고 구조화된 요약문을 작성합니다.

## 출력 형식

반드시 아래 형식으로 작성합니다.

### 회의 개요
- 날짜: (텍스트에서 추출하거나 오늘 날짜 사용)
- 참석자: (언급된 이름 목록)
- 회의 목적: (1~2문장)

### 핵심 논의 사항
(주요 논의 내용을 3~5개 항목으로 정리)

### 결정 사항
(이번 회의에서 확정된 결정 사항)

### 액션 아이템
| 담당자 | 할 일 | 마감일 |
|--------|--------|--------|
| (이름) | (내용) | (날짜) |

### 다음 회의 안건
(다음 회의에서 논의할 예정인 내용)

## 작동 방식
```

1. 사용자가 회의 내용을 붙여넣거나 파일 경로를 제공합니다.
2. 위 형식에 맞춰 한국어로 요약문을 작성합니다.
3. 회의 내용에서 명확하지 않은 항목은 "(확인 필요)" 로 표시합니다.
4. 액션 아이템 담당자가 명시되지 않은 경우 "(미지정)" 으로 표시합니다.

디렉터리 배치와 자동 로드 확인

```
 # Skill 디렉터리 생성 및 파일 저장
mkdir -p skills/meeting-summary
# 위 내용을 skills/meeting-summary/SKILL.md 로 저장

# Hermes 재시작 또는 리로드
hermes reload

# Skill 로드 확인
hermes skills list
# 출력 예: ✓ daily-meeting-summary  v1.0.0  [skills/meeting-summary/SKILL.md]

# 실제 사용 테스트
hermes "오늘 팀 스탠드업 내용을 정리해줘: [회의 내용 붙여넣기]"
```

`hermes skills list` 출력에 `daily-meeting-summary` 가 표시되면 Hermes 가 해당 Skill 을 인식한 것입니다. 이후 사용자가 "오늘 회의 요약", "스탠드업 정리" 같은 표현으로 요청하면 Hermes 가 자동으로 이 Skill 을 선택해 지정된 형식으로 응답합니다 [S03]. Gemma 4 31B Dense 의 256K 컨텍스트 덕분에 회의록 분량이 수만 토큰을 넘더라도 분할 없이 단일 호출로 처리됩니다.

10.3 운영 모니터링과 Curator loop 활성화

Hermes Agent 를 팀 전체에 배포한 뒤 안정적으로 운영하려면 정량 지표 추적, Curator loop 결과 측정, 분기별 점검 세 가지를 체계화해야 합니다. 운영 초기에 이 구조를 잡아두면 6개월 후 ROI 보고와 모델 교체 의사결정에 필요한 데이터를 자연스럽게 쌓을 수 있습니다.

10.3.1 운영 핵심 지표 — 응답 시간, 토큰 사용량, 실패율

세 가지 핵심 지표와 추적 도구

운영 단계에서 반드시 추적해야 할 지표는 응답 시간, 토큰 사용량, 실패율 세 가지입니다. 각각 다른 도구에서 수집하지만, LiteLLM 대시보드와 Hermes Kanban 통계를 함께 활용하면 단일 화면에서 전체 현황을 파악할 수 있습니다 [S10][S03].

응답 시간은 사용자가 요청을 보낸 시점부터 첫 토큰이 도착하는 시간 (TTFT, Time to First Token) 으로 측정합니다. vLLM 기준 Gemma 4 31B Dense 의 일반 업무 요청 (512 토큰 이하 프롬프트) 에서 TTFT 2~3초, 전체 응답 완료 10~15초를 정상 범위로 설정합니다. 256K 컨텍스트 풀 활용 시 TTFT 가 8~15초 수준까지 늘어나므로 분석 작업과 대화형 작업을 분리해 임계값을 다르게 둡니다. 이 값을 초과하면 GPU 메모리 부족이나 동시 요청 과부하를 의심합니다.

토큰 사용량은 팀원별 일간·월간 사용량을 LiteLLM 가상 키 단위로 집계합니다. 예상치 못한 급등은 무한 루프나 과도하게 긴 컨텍스트를 주입하는 잘못 작성된 Skill 에서 발생하는 경우가 많습니다 [S10]. 월별 추이를 추적해 예산 초과를 사전에 감지합니다.

실패율은 Hermes Kanban 의 `blocked` 및 `archived-failed` 카드 비율로 측정합니다. 전체 요청 대비 실패율이 5% 를 초과하면 모델 응답 품질 저하나 endpoint 연결 불안정을 점검합니다 [S03].

임계값 알림 설정

LiteLLM 은 Prometheus 메트릭을 내보내므로 Grafana 대시보드와 연결해 임계값 초과 시 자동 알림을 보낼 수 있습니다. 사내 메신저 (Slack, Teams, 카카오워크 등) 와 연동하면 on-call 담당자가 즉시 확인할 수 있습니다.

지표	정상 범위	경고 임계값	알림 채널
TTFT (첫 토큰 응답, 일반)	≤ 3초	> 6초	사내 메신저
TTFT (256K 컨텍스트 분석)	≤ 15초	> 30초	사내 메신저
전체 응답 완료 (일반)	≤ 15초	> 30초	사내 메신저
팀 일간 토큰 사용량	예산 100% 이하	예산 80% 초과	이메일 + 메신저
Kanban 실패율	≤ 5%	> 10%	사내 메신저
vLLM GPU 메모리 사용률	≤ 85%	> 95%	PagerDuty / 메신저

Prometheus exporter 는 vLLM 의 `--enable-metrics` 플래그로 활성화하고, LiteLLM 의 `prometheus` 콜백을 등록해 두 지표를 단일 Grafana 인스턴스에서 관리합니다. Grafana 와 Prometheus 모두 AGPL-3.0 라이선스로 기업 내부 운영에 별도 라이선스 비용 없이 사용 가능합니다.

10.3.2 Curator loop 활성화 — 6개월 운영 후 학습 자산 측정

Curator loop 작동 원리

Curator loop 는 Hermes Agent 의 자기 개선 메커니즘입니다. 운영 중 쌓인 대화 이력, 사용자 피드백, 반복 요청 패턴을 분석해 새로운 Skill 후보를 자동 생성하고 Archive 에 저장합니다 [S03]. 사용자가 같은 유형의 요청을 반복할수록 Curator 가 패턴을 학습해 다음 요청을 더 빠르고 정확하게 처리한다는 것이 핵심 가치입니다 [S02].

Curator loop 는 기본적으로 비활성 상태로 설치됩니다. 다음 설정으로 활성화합니다.

```
 # ~/.hermes/config.yaml
curator:
  enabled: true
  review_interval_hours: 24       # 24시간마다 대화 이력 분석
  min_interactions_to_learn: 5    # 동일 패턴 5회 이상 반복 시 Skill 후보 생성
```

```
  auto_promote: false            # Skill 후보는 사람 검토 후 수동 승인 권장
  output_dir: "~/.hermes/skills/curator-generated"
```

`auto_promote: false` 로 두면 Curator 가 생성한 Skill 후보를 담당자가 검토한 뒤 수동으로 승인해야 운영 Skill 로 등록됩니다. 초기 6개월 동안은 이 방식을 권장합니다. 자동 승인을 켜면 검토 없이 Skill 이 즉시 배포되므로, Skill 품질 관리 체계가 갖춰진 이후에 전환합니다.

6개월 시점 측정 항목

도입 6개월 시점에 다음 항목을 측정해 ROI 보고 지표로 활용합니다.

측정 항목	측정 방법	목표값 예시
Curator 자동 생성 Skill 후보 수	`ls ~/.hermes/skills/curator-generated/`	≥ 10개
승인·운영 중 Curator Skill 수	`hermes skills list --source curator`	≥ 5개
Curator Skill 활용률	전체 요청 중 Curator Skill 처리 비율 (LiteLLM 태그 기준)	≥ 20%
사용자 만족도 (5점 척도)	Hermes Kanban 카드 완료 후 피드백 수집	≥ 4.0점
평균 응답 시간 변화	도입 초기 대비 6개월 시점 TTFT 비교	감소 또는 유지

이 수치를 사내 보고서 핵심 지표로 인용하면 경영진에게 AI 에이전트 도입 효과를 정량으로 제시할 수 있습니다. Curator loop 가 생성한 Skill 이 실제 업무 요청의 20% 이상을 처리하고 있다면, 팀이 초기에 수동으로 작성한 Skill 대비 운영 자산이 자동으로 축적되고 있다는 근거가 됩니다.

10.3.3 분기별 점검 — 모델 교체, Skill 폐기, 권한 갱신

분기별 점검을 체계화해야 하는 이유

AI 에이전트 운영에서 가장 자주 발생하는 문제는 초기 설정이 그대로 방치된 채 환경이 변하는 것입니다. 3개월이 지나면 모델 신규 버전이 출시되거나, 더 이상 사용하지 않는 Skill 이 누적되거나, 퇴직자의 API 키가 유효한 채로 남아 보안 감사에서 지적받는 사례가 생깁니다 [S03]. 분기별 점검을 사내 위키의 정기 운영 절차로 등재하면 이런 운영 부채 누적을 방지할 수 있습니다.

세 가지 점검 항목별 절차

모델 교체 점검은 현재 운영 중인 Gemma 4 31B Dense 의 신규 버전 또는 더 나은 대안 모델이 출시되었는지 확인하는 작업입니다. Hugging Face 모델 페이지의 릴리스 노트와 LLM 벤치마크 사이트 (Open LLM Leaderboard) 를 분기마다 확인합니다. 교체 결정 기준은 한국어 처리 벤치마크 점수 5%p 이상 향상, 동일 하드웨어에서 처리량 10% 이상 개선, 라이선스 조건 개선 중 하나 이상을 충족하는 경우로 정합니다.

Skill 폐기 점검은 최근 90일간 호출 횟수가 0인 Skill 을 목록화하는 작업입니다. LiteLLM 통계나 Hermes 로그에서 Skill 별 호출 빈도를 추출합니다. 호출 이력이 없는 Skill 은 담당자를 확인해 업무 요구 소멸 여부를 판단하고, 불필요하면 디렉터리에서 제거한 뒤 `hermes reload` 로 반영합니다. Curator 가 자동 생성한 Skill 후보 중 6개월이 지나도 승인되지 않은 파일도 이 시점에 정리합니다 [S03].

권한 갱신 점검은 LiteLLM 가상 키와 Hermes Profile 권한을 점검하는 작업입니다. 퇴직자 또는 역할이 변경된 팀원의 키를 비활성화하고, 팀 예산 상한을 분기 실적 기반으로 재설정합니다.

분기별 점검 체크리스트

점검 항목	절차	담당	완료 기준
모델 신규 버전 확인	HuggingFace 릴리스 노트 + Open LLM Leaderboard 조회	ML 담당자	교체 여부 결정 문서 작성
모델 교체 시 vLLM 재배포	신규 가중치 다운로드 → vLLM rolling restart	인프라 담당자	`hermes doctor` 통과
미사용 Skill 목록화	90일 호출 0건 Skill 추출	IT 담당자	목록 작성 + 폐기 여부 결정
Skill 디렉터리 정리	폐기 결정 Skill 제거 + `hermes reload`	IT 담당자	`hermes skills list` 목록 업데이트
비활성 가상 키 폐기	LiteLLM UI → Virtual Keys → 퇴직자 키 비활성화	보안 담당자	활성 키 목록 갱신
팀 예산 상한 갱신	LiteLLM 팀 예산 재설정 (분기 실적 기반)	IT 관리자	신규 상한 설정 완료
Profile 권한 갱신	역할 변경자 Profile 재구성	IT 담당자	Profile 목록 업데이트

분기 점검 결과를 사내 위키에 날짜와 담당자 서명으로 남겨두면, 보안 감사나 경영진 검토 시 운영 이력 근거로 제출할 수 있습니다. 세 항목 중 가중치가 가장 큰 것은 권한 갱신입니다. 미사용 API 키 하나가 외부 유출되면 온프레미스 서버에서 동작 중인 LLM 전체 접근 권한이 노출될 수 있으므로, 이 항목은 분기가 아닌 인원 변동 발생 즉시 처리하는 기준을 추가로 두는 것이 좋습니다.

11장. 도입 로드맵 — PoC 4주 → 파일럿 3개월 → 본격 운영

Hermes Agent 도입 판단을 내린 조직이 가장 먼저 직면하는 물음은 "어디서부터 시작하는가"입니다. 신기술 도입 프로젝트가 개념 검증 단계에서 흐지부지 끝나거나, 반대로 준비 없이 전사 배포를 강행했다가 혼란을 빚는 사례는 드물지 않습니다. 이 장에서는 단일 부서 PoC 4주,

다부서 파일럿 3개월, 전사 본격 운영이라는 세 단계로 구성된 표준 로드맵을 제시합니다. 각 단계에는 산출물, 게이트 KPI, 거버넌스 요건을 명시해 두었으므로, 이 구조를 사내 도입 품의서 본문 또는 부록으로 그대로 인용할 수 있습니다.

세 단계 로드맵은 단순한 일정 관리 도구가 아닙니다. PoC 단계가 파일럿 단계의 조건부 진입권을 발급하고, 파일럿 단계가 본격 운영의 전사 예산 배분 근거를 생성하는 방식으로, 각 단계가 다음 단계의 의사결정 재료를 생산하도록 설계되어 있습니다. Hermes Agent 의 MIT 라이선스와 단일 `curl` 설치 방식[S01]은 PoC 진입 비용을 사실상 제로 수준으로 낮추어, 국내 공공·민간 조직 모두 별도 소프트웨어 구매 없이 1주차부터 실질적인 업무 자동화 결과물을 확인할 수 있습니다[S12].

11.1 PoC 단계 (4주) — 단일 부서·단일 유즈케이스

PoC 단계의 목표는 두 가지입니다. 첫째, Hermes Agent 가 실제 업무 맥락에서 동작한다는 사실을 부서장이 눈으로 확인하는 것. 둘째, 4주 안에 측정 가능한 수치를 확보하여 파일럿 단계 진입 여부를 데이터로 결정하는 것입니다. 이 두 가지 목표를 달성하기 위해 PoC 단계는 주차별 산출물과 게이트 KPI 를 명확히 구분합니다.

11.1.1 1주차 — 환경 구성과 첫 Skill 작성

환경 구성 방법과 소요 시간

PoC 환경은 별도 서버 없이 16GB RAM 이상의 업무용 PC 한 대면 충분합니다. Hermes Agent 는 단일 `curl` 명령으로 의존성을 포함한 전체 스택을 자동 설치하며[S01], Ollama 를 로컬 LLM 런타임으로 함께 설치하면 인터넷 연결 없이도 모델 추론이 가능합니다. IT 담당자 한 명이 오전 업무 시작 전 약 2시간을 투자하면 첫 번째 Hermes 인스턴스가 동작 상태가 됩니다. MSAP.ai 처럼 Hermes 를 사전 통합한 운영 환경이 있다면 이 구성 시간을 절반 이하로 단축할 수 있습니다.

1주차 필수 산출물: 일일 회의록 요약 Skill

환경 구성 직후 작성할 첫 번째 Skill 은 일일 회의록 요약입니다. 이 Skill 을 첫 번째 과제로 선택하는 이유는 두 가지입니다. 회의록이라는 인풋이 부서 내 누구에게나 친숙하고, 요약이라는 아웃풋의 품질을 비전문가도 즉시 판단할 수 있기 때문입니다. Hermes 의 Skill 은 단순 텍스트 파일 형식의 지시문으로 작성되며[S03], 코딩 경험이 없는 팀원도 Skill 내용을 수정할 수 있습니다. 1주차 종료 시점에 이 Skill 이 실제 회의록 한 건을 200자 이내 요약으로 산출하는 것을 부서장 앞에서 시연합니다.

1주차 일일 체크리스트

일차	담당	산출물	완료 기준
1일차	IT 담당	PC 환경 + Ollama 설치	Hermes CLI 응답 확인
2일차	IT 담당	기본 Profile 설정	첫 메시지 응답 수신

일차	담당	산출물	완료 기준
3일차	IT 담당 + 현업	회의록 요약 Skill 초안	Skill 로드·실행 확인
4일차	현업	실제 회의록 3건 테스트	요약 품질 부서장 확인
5일차	IT 담당	테스트 결과 문서화	1주차 보고서 작성

이 체크리스트를 품의서 부록으로 첨부하면, 예산 심의자가 PoC 착수 첫 주에 어떤 결과물이 생산되는지를 구체적으로 파악할 수 있습니다.

11.1.2 2~3주차 — 추가 Skill 3종과 Kanban 협업 시작

단일 사용자에서 팀 협업으로 전환

1주차에 혼자 사용하던 Hermes 를 2주차부터 부서 팀원 5~10명이 함께 사용하는 형태로 전환합니다. 이 전환이 PoC 성공의 첫 번째 분기점입니다. 단일 사용자 데모는 잘 동작하지만 5명 이상이 동시에 사용할 때 응답 품질과 처리 속도가 유지되는지가 파일럿 진입 여부의 핵심 판단 기준이 되기 때문입니다. Hermes 의 Profile 기능은 사용자별 메모리와 Skill 을 독립적으로 관리하므로[S03], 한 팀원의 작업 이력이 다른 팀원의 응답에 영향을 주지 않습니다.

추가 Skill 3종 선정 기준

2~3주차에 추가할 Skill 3종은 일상 반복 업무 중 시간 소모가 가장 큰 항목에서 선정합니다. 일반적으로 일일 업무 보고 초안 작성, 사내 FAQ 응대, 타 부서 공문 초안 검토가 후보입니다. 각 Skill 은 현업 담당자가 직접 요구사항을 서술하면 IT 담당자가 Skill 파일로 변환하는 방식으로 협업 작성합니다[S03]. Hermes 의 Skill 구조는 자연어 지시문 기반이므로 현업이 제시하는 요구사항과 최종 Skill 사이의 번역 비용이 낮습니다.

Kanban 보드 도입과 협업 가시화

3주차에는 Hermes 내장 Kanban 보드를 사용하여 Skill 실행 현황을 팀 전체가 공유합니다. Kanban 보드는 SQLite 기반으로 동작하며, triage → todo → ready → running → done 흐름으로 에이전트 작업 상태를 추적합니다[S03]. 이 가시화 기능은 단순한 UI 편의 기능을 넘어서, PoC 기간 중 어떤 Skill 이 얼마나 자주 실행되었는지를 4주차 KPI 측정의 원시 데이터로 제공합니다.

2~3주차 Skill 확장 및 사용자 확대 일정

주차	추가 Skill	참여 인원	확인 지표
2주차	일일 업무 보고 초안	3명 (팀원 2 + IT)	초안 생성 성공률 ≥ 90%
2주차	사내 FAQ 응대	3명 동일	응답 정확도 팀장 확인
3주차	공문 초안 검토	5~10명 (전 팀원)	사용자 만족도 설문
3주차	Kanban 협업 개시	5~10명 전원	일일 카드 생성 ≥ 3건

11.1.3 4주차 게이트 — KPI 측정과 파일럿 진입 의사결정

게이트 KPI 의 역할

4주차 종료 시점에 세 가지 KPI 를 측정하여 파일럿 단계 진입 여부를 결정합니다. 이 수치가 '합격 임계점' 이상이면 파일럿으로 진입하고, 임계점 미달이면 PoC 연장·유즈케이스 변경·PoC 종료 세 가지 중 하나를 선택합니다. 판단 기준을 미리 수치로 박제해 두는 것이 핵심입니다. 숫자가 없는 게이트는 감정적 판단으로 흐르기 쉽습니다.

PoC 4주차 KPI 카드

KPI	목표	합격 임계점	측정 방법
사용자 만족도	4.0 / 5.0	≥ 3.5	팀원 전체 5점 리커트 설문
반복 업무 시간 절감	주 2시간 이상	주 1시간 이상	사용 전·후 시간 기록 비교
Skill 실행 오류율	≤ 5%	≤ 15%	Kanban 완료 vs 오류 카드 비율

파일럿 진입 의사결정 회의 구성

4주차 KPI 측정 결과를 1페이지 요약으로 정리하여 부서장 + IT 책임자가 참여하는 의사결정 회의에 제출합니다. 회의 안건은 세 가지입니다: (1) KPI 달성 현황 공유, (2) 파일럿 단계 예산 및 일정 확정, (3) 파일럿 단계 참여 부서 1차 선정. 이 회의의 산출물은 파일럿 착수를 승인하는 내부 문서 한 장입니다. 별도 외부 컨설팅이나 대규모 회의체 없이 진행할 수 있습니다.

PoC 종료 또는 중단 시 처리 방침

합격 임계점을 달성하지 못했을 때의 처리 방침도 미리 정해 두어야 합니다. PoC 연장 시에는 유즈케이스를 바꾸거나 참여 인원을 늘려 추가 2주를 진행합니다. PoC 를 종료할 경우에도 Hermes 는 MIT 라이선스 OSS 이므로[S01] 라이선스 위약금이나 계약 해지 비용이 발생하지 않습니다. 투자 비용의 대부분은 IT 담당자 투입 공수이며, 이 역량은 이후 다른 업무 자동화 과제에서 재활용할 수 있습니다.

11.2 파일럿 단계 (3개월) — 다부서 확장·운영 절차 정착

파일럿 단계는 PoC 에서 검증한 단일 유즈케이스를 3~5개 부서, 수십 명 규모로 확장하면서 동시에 운영 절차와 거버넌스 기반을 수립하는 단계입니다. PoC 가 '가능한가'를 확인하는 실험이라면, 파일럿은 '규모를 키워도 지속 가능한가'를 검증하는 운영 준비 기간입니다. 이 단계에서 인프라 투자 결정과 부서 확장 계획이 동시에 진행되므로, IT 부서와 경영기획 부서의 긴밀한 협력이 필요합니다.

파일럿 단계 3개월은 인프라 전환, 다부서 확장, 본격 운영 진입 의사결정이라는 세 월별 단계로 구분됩니다. 각 월 종료 시점에 진행 상황을 점검하고 다음 달 계획을 조정하는 월간 운영 회의체를 IT 담당자 + 부서 챔피언 참여로 구성합니다.

11.2.1 1개월 — vLLM 전환과 LiteLLM 본격 운영

Ollama 에서 vLLM 으로 전환하는 이유

PoC 단계에서 사용한 Ollama 는 단일 사용자 또는 소수 사용자 환경에서 간편하게 로컬 LLM 을 구동할 수 있는 런타임입니다. 그러나 팀원 5명 이상이 동시에 요청을 보내는 파일럿 규모에서는 처리량(throughput)이 제약이 됩니다. vLLM(Virtual LLM)은 연속 배치(continuous batching)와 PagedAttention 기술로 동시 요청을 효율적으로 처리하며, 단일 A6000 GPU 기준 Ollama 대비 약 3~5배 높은 토큰 처리량을 제공합니다[S10]. 파일럿 1개월차에 이 전환을 완료함으로써 2개월차 다부서 확장의 기반을 마련합니다.

LiteLLM Proxy 를 모델 게이트웨이로 구성

vLLM 전환과 함께 LiteLLM(MIT 라이선스) 을 프록시 게이트웨이로 도입합니다[S10]. LiteLLM 은 OpenAI 호환 API 형식으로 100개 이상의 LLM 공급자를 단일 엔드포인트로 통합하며, Hermes Agent 의 `setup` 명령으로 40개 이상의 공급자를 네이티브로 등록할 수 있습니다[S10]. 실무에서는 사내 vLLM 인스턴스를 1차 공급자로, 상용 API(Anthropic Claude, OpenAI)를 2차 fallback 으로 구성합니다. 이 구성은 사내 GPU 가 과부하 상태일 때 자동으로 외부 API 로 라우팅하고, 월별 외부 API 비용 상한선을 LiteLLM 의 spend tracking 기능으로 관리합니다.

GPU 인프라 투자 결정 시점

파일럿 1개월차는 GPU 인프라 구매 결재를 진행할 최적 시점입니다. PoC 단계에서 확보한 사용량 데이터를 근거로 필요한 GPU 수량을 추산하고, 2개월차 다부서 확장이 시작되기 전에 하드웨어가 가용 상태가 되도록 조달 일정을 맞춥니다. 단일 NVIDIA RTX 4090(24GB VRAM) 으로 시작하여 사용량 증가에 따라 NVIDIA A6000(48GB VRAM) 또는 다중 GPU 구성으로 단계적으로 확장하는 방식이 초기 투자 리스크를 낮춥니다. 국내 공공기관 기준으로 sLLM 자체 구축 초기 비용은 7~13억 원 수준으로 보고되고 있으나[S12], Hermes Agent + 오픈소스 스택 조합은 이 비용의 30~50% 수준에서 동급 기능을 구현할 수 있습니다.

1개월차 인프라 전환 절차

항목	PoC (이전)	파일럿 1개월차 (이후)
LLM 런타임	Ollama (단일 사용자)	vLLM (동시 다중 처리)
게이트웨이	없음 (직접 연결)	LiteLLM Proxy (단일 엔드포인트)
GPU	업무용 PC (선택)	단일 RTX 4090 이상 (필수)
비용 추적	없음	LiteLLM spend tracking (일별·부서별)
장애 대응	수동 재시작	LiteLLM fallback → 외부 API 자동 전환

11.2.2 2개월 — 3~5 부서 확장과 Profile 추가

부서 확장의 핵심 원칙: 챔피언 선정

다부서 확장이 성공하려면 각 부서에 도입 챔피언(Champion) 한 명을 지정해야 합니다. 챔피언은 Hermes 사용에 가장 적극적인 팀원으로, IT 부서가 아닌 현업 부서 소속입니다. 챔피언의 역할은 자기 부서의 업무 특성에 맞는 Skill 을 발굴하고, 팀원들의 사용 질문에 1차로 답하며, 월간 운영 회의에 부서 대표로 참여하는 것입니다. IT 부서 한 곳이 전사 도입을 독점 지원하는 구조는 파일럿 단계에서 반드시 병목이 됩니다.

Profile 로 부서별 독립 환경 구성

Hermes 의 Profile 기능은 부서별 독립 에이전트 인스턴스를 상태 충돌 없이 병렬 운영합니다[S03]. 개발팀 Profile 은 코드 리뷰와 배포 자동화 Skill 을 탑재하고, 마케팅팀 Profile 은 보도자료 초안과 SNS 콘텐츠 Skill 을 탑재하는 방식입니다. 각 Profile 은 별도 메모리와 Skill 묶음을 보유하므로, 한 부서의 업무 이력이 다른 부서의 응답에 영향을 주지 않습니다[S03]. 부서가 추가될수록 Profile 수가 늘어나며, 이 구조가 Hermes 의 다부서 확장을 기술적으로 뒷받침합니다.

2개월차 부서 확장 계획표

부서	Profile 명	1차 Skill 3종	챔피언	확장 시점
개발팀	dev-assistant	코드 리뷰·PR 요약·릴리스 노트	시니어 개발자 1명	2개월차 1주
마케팅팀	marketing-bot	보도자료 초안·SNS 콘텐츠·경쟁사 모니터링	콘텐츠 담당자 1명	2개월차 1주
고객지원팀	cs-agent	FAQ 응대·불만 요약·에스컬레이션 분류	팀장 또는 선임 1명	2개월차 2주
경영기획팀	strategy-analyst	보고서 요약·KPI 집계·회의록 작성	기획 담당자 1명	2개월차 3주
HR팀	hr-assistant	채용 공고 초안·교육 일정 안내·내규 Q&A	HR 담당자 1명	2개월차 4주

권한 매트릭스와 접근 통제 설정

부서가 늘어나면 Skill 과 데이터에 대한 접근 통제가 필요합니다. Hermes 의 Profile 은 테넌트 네임스페이스 단위로 Skill 과 도구 접근권을 분리할 수 있으며[S03], 부서 간 공유가 허용된 Skill 은 공용 카탈로그에 등록하고 부서 전용 Skill 은 해당 Profile 에만 탑재합니다. 2개월차 종료 시점에 부서별 Skill 목록과 접근 권한을 문서화하면, 3개월차 게이트 KPI 측정과 본격 운영 단계의 거버넌스 문서로 직접 활용할 수 있습니다.

11.2.3 3개월 게이트 — 본격 운영 진입 의사결정

5개 KPI 와 측정 방법

파일럿 3개월 종료 시점에 다섯 가지 KPI 를 측정하여 본격 운영 진입 여부를 결정합니다. 이 수치는 임원 회의 안건으로 제출하는 보고서의 핵심 데이터입니다.

파일럿 3개월 KPI 카드

KPI	목표	합격 임계점	측정 방법
월간 활성 사용자 수	50명 이상	30명 이상	Hermes Profile 세션 로그
누적 Skill 수	20개 이상	12개 이상	Skill 카탈로그 카운트
Curator 생성 Skill 수	전체의 30% 이상	전체의 15% 이상	Curator 로그 분석
월 토큰 비용 (외부 API)	USD 500 이하	USD 1,000 이하	LiteLLM spend tracking
사용자 만족도	4.0 / 5.0	3.5 / 5.0	전체 사용자 설문

Curator 생성 Skill 비율의 의미

다섯 번째 KPI 인 Curator 생성 Skill 비율은 다른 KPI 와 성격이 다릅니다. 이 수치는 조직이 Hermes 의 자기 개선 루프를 얼마나 잘 활용하고 있는지를 반영합니다. Curator 가 스스로 Skill 을 생성하거나 개선하는 비율이 높을수록, IT 담당자의 수동 개입 없이 시스템이 진화하는 자율 운영 수준이 높다는 뜻입니다. 이 비율이 15% 이상이면 본격 운영 단계에서 IT 인력 투입을 줄이면서도 Skill 수를 유지할 수 있는 기반이 형성된 것으로 판단합니다.

임원 보고와 예산 확정 프로세스

5개 KPI 를 담은 1~2페이지 보고서를 임원 회의 안건으로 등재합니다. 보고서에는 KPI 달성 현황 외에 본격 운영 단계의 연간 예상 비용(인프라·운영 공수·교육)과 절감 효과(반복 업무 시간·외부 SaaS 비용)를 함께 제시합니다. 임원 승인이 나면 IT 부서는 본격 운영을 위한 거버넌스 위원회 구성과 연간 예산 배정 작업에 착수합니다. MCP(Model Context Protocol) 가 벤더 중립 표준으로 Linux Foundation 산하에서 관리되고 있다는 점[S11]은, 특정 벤더에 종속되는 리스크 없이 장기 운영이 가능하다는 논거로 보고서에 포함할 수 있습니다.

11.3 본격 운영 — 거버넌스·KPI·재교육 체계

본격 운영 단계는 IT 부서 주도의 파일럿을 넘어 전사 차원의 운영 체계로 전환하는 단계입니다. 이 전환에서 가장 중요한 것은 기술적 완성도가 아니라 의사결정 체계의 명확화입니다. Hermes Agent 가 다부서에서 동시에 운영될 때 발생하는 Skill 충돌, 데이터 접근 정책, 예산 배분 갈등을 해소할 수 있는 단일 의사결정 채널이 필요합니다.

본격 운영 단계의 세 축은 사내 AI 거버넌스 위원회, 분기 KPI 및 연간 ROI 측정 프레임, 재교육 체계입니다. 이 세 가지는 독립적으로 운영되는 것이 아니라 거버넌스 위원회가 KPI 를 설정하고, KPI 결과가 재교육 커리큘럼을 갱신하는 순환 구조로 연결됩니다.

11.3.1 사내 AI 거버넌스 위원회 구성과 분기 회의체

위원회 구성의 필요성

AI 에이전트가 다부서에 배포되면 기술 문제보다 정책 문제가 먼저 수면 위로 올라옵니다. 어떤 데이터를 Hermes 에 입력할 수 있는가, 외부 API 로 전송되는 데이터에 개인정보가 포함되어 있는가, 특정 부서의 Skill 이 전사 공용 카탈로그에 등록될 수 있는가 같은 질문들입니다. 이런 질문들이 IT 부서 단독으로 결정하기에는 법무·정보보호·인사 영역에 걸쳐 있습니다. 분기마다 모이는 거버넌스 위원회가 이 결정을 단일 채널로 처리합니다[S12].

위원회 구성안

부서	대표 직급	주요 역할	의사결정 권한 범위
IT 부서	IT 팀장 또는 CTO	Skill 카탈로그 승인·인프라 예산	기술 표준·아키텍처 결정
정보보호	정보보호 책임자 (CISO)	데이터 분류·외부 API 전송 정책	보안 정책·접근 통제 기준
법무	법무팀장	라이선스 컴플라이언스·개인정보 처리	법적 리스크 수용 기준
인사	HR 부서장	교육 체계·AI 활용 역량 평가	교육 예산·의무 교육 등재
경영기획	기획실장 또는 CFO	연간 ROI 검토·차기 예산 배정	전사 AI 투자 우선순위

위원회 위원장은 CIO 또는 CTO 가 맡는 것을 권고합니다. 위원장이 임원급 권한을 갖추어야 부서 간 이해충돌 시 중재 결정을 신속하게 내릴 수 있습니다. 위원회는 분기에 1회 정례 회의를 개최하고, 긴급 안건 발생 시 수시 소집 절차를 사전에 규정합니다.

분기 회의체 의제 표준안

분기 회의체의 의제를 표준화하면 매번 준비 부담 없이 일관된 검토가 가능합니다. 표준 의제는 다섯 항목입니다: (1) 전분기 KPI 결과 보고, (2) 신규 Skill 카탈로그 등록 승인, (3) 보안 정책 변경 사항 공유, (4) 차기 분기 예산 및 인프라 계획 확인, (5) 신규 도입 부서 또는 유즈케이스 승인. 이 의제를 분기 회의 2주 전에 위원회 구성원에게 배포하면, 회의 당일 논의가 결론까지 60분 이내에 종료됩니다.

11.3.2 분기 KPI · 연간 ROI 측정 프레임

분기 KPI 측정 항목

본격 운영 단계에서는 파일럿 5개 KPI 를 확장하여 네 가지 분기 KPI 를 지속 측정합니다. 이 수치는 거버넌스 위원회 정례 회의의 첫 번째 의제 자료입니다[S10].

분기 KPI	측정 도구	보고 주기	목표 수준
월간 활성 사용자 수	Hermes Profile 세션 로그	분기	전분기 대비 10% 이상 성장
Skill 활용률 (실행 / 등록)	Kanban 완료 카드 / 카탈로그 수	분기	≥ 70%
평균 응답 시간	LiteLLM 응답 시간 로그	주간	≤ 3초 (P95)
월 외부 API 비용	LiteLLM spend tracking	월간	전분기 대비 감소 또는 동결

연간 ROI 측정 프레임

연간 ROI 는 두 가지 절감 항목으로 구성됩니다. 첫째, 반복 업무 시간 절감입니다. 파일럿 단계에서 측정한 주당 절감 시간을 전사 사용자 수로 환산하여 연간 절감 인시(man-hour)를 계산합니다. 둘째, 외부 SaaS 비용 절감입니다. Hermes Agent 와 기능이 중복되는 외부 SaaS 구독(AI 작성 도구, 챗봇 서비스, 요약 도구 등)을 정리하면 그 구독료가 직접 절감액이 됩니다[S12].

분기 KPI × 연간 ROI 측정 매트릭스

측정 항목	데이터 출처	환산 방식	연간 보고 시점
반복 업무 절감 시간	부서별 사용자 설문	절감 시간 × 평균 시급 × 사용자 수	연말 결산
외부 SaaS 구독 절감	재무팀 구독 목록	해지된 구독료 합산	분기 누적
내부 GPU 인프라 ROI	IT 인프라 비용 대장	(절감액 합계 – 인프라 비용) / 인프라 비용	연간
Hermes 운영 인력 비용	IT 부서 공수 집계	월 투입 인시 × 인건비 단가	분기

연간 ROI 보고서는 차기 연도 예산 확보의 핵심 근거입니다. 첫 해 ROI 가 플러스가 아니더라도, 연간 비용 추세와 절감 궤적을 데이터로 제시하면 차기 연도 예산 심의에서 설득력 있는 논거가 됩니다.

11.3.3 재교육 체계 — Skill 카탈로그 갱신과 분기 워크숍

재교육 체계가 필요한 이유

Hermes Agent 도입 후 1~2년차에 가장 흔히 발생하는 문제는 초기 열기가 식고 사용자가 반복 업무에 다시 손으로 직접 처리하는 패턴으로 되돌아가는 것입니다. 이 정체기의 주된 원인은 두 가지입니다. 신규 입사자가 Hermes 사용법을 배울 경로가 없고, 기존 Skill 이 업무 변화를 반영하지 못해 구식이 되는 것입니다[S03]. 분기 워크숍과 Skill 카탈로그 갱신이 이 두 가지 원인을 동시에 해결합니다.

Skill 카탈로그 분기 갱신 절차

Skill 카탈로그는 분기마다 한 번 정기 검토합니다. IT 담당자와 부서 챔피언이 함께 현재 등록된 Skill 목록을 검토하여, 활용률이 낮은 Skill 은 아카이브하고 신규 업무 요구에 맞는 Skill 을 추가합니다. Curator 가 자동으로 생성한 Skill 이 있다면 품질 검토 후 공용 카탈로그에 승격 여부를 결정합니다[S03]. 이 갱신 절차를 거버넌스 위원회 정례 회의 직전에 완료하면, 회의 의제 두 번째 항목인 신규 Skill 카탈로그 등록 승인에 최신 목록을 제출할 수 있습니다.

분기 워크숍 구성

분기 워크숍은 반나절(4시간) 일정으로, 사내 의무 교육 시간으로 등재하는 것을 권고합니다. 의무 교육으로 등재하면 신규 입사자도 첫 분기 내에 반드시 워크숍에 참여하게 되어 사용자 기반이 자연스럽게 유지됩니다. 워크숍 커리큘럼은 세 부분으로 구성됩니다: Hermes Agent 기본 사용법(1시간), 부서별 Skill 실습(2시간), 신규 Skill 요구사항 발굴 토론(1시간). 세 번째 부분에서 수집한 요구사항이 다음 분기 Skill 갱신 목록의 입력이 됩니다.

분기 재교육 체계 일정표

활동	시점	참여자	산출물
Skill 카탈로그 검토	분기 종료 2주 전	IT 담당자 + 챔피언	갱신 Skill 목록
분기 워크숍	분기 첫째 주	전 직원 (신규 입사자 필수)	이수 확인서 + 신규 요구사항
Curator Skill 품질 검토	분기 워크숍 후 1주	IT 담당자	카탈로그 승격 목록
거버넌스 위원회 승인	분기 워크숍 후 2주	위원회 전원	Skill 등록 승인 결과

도입 후 2년차 이후의 지속 발전 방향

재교육 체계가 안착되면 도입 2년차부터는 IT 부서의 직접 개입 없이도 부서 챔피언 주도로 Skill 이 추가·갱신되는 자율 운영 단계로 진입합니다. 이 단계에서 IT 부서의 역할은 인프라 안정성 유지와 보안 정책 점검으로 전환됩니다. Hermes Agent 의 MIT 라이선스[S01]와 오픈소스 MCP 표준[S11]은 특정 벤더에 의존하지 않고 내부 역량으로 시스템을 발전시킬 수 있는 구조적 기반이 됩니다. 이 자율 운영 수준이 11장 로드맵이 지향하는 최종 상태입니다.

12장. 결론·요약 및 다음 단계

본 백서는 오픈소스 AI 에이전트 오케스트레이션 플랫폼인 Hermes Agent 를 기술·운영·라이선스·도입 절차의 네 가지 관점에서 검토했습니다. 1장부터 11장까지 각 주제를 독립적으로 분석했지만, 실제 도입 결정은 그 분석들을 하나의 판단 체계로 통합할 때 비로소 이루어집니다. 12장은 그 통합 역할을 담당합니다. 백서 전반에 걸쳐 반복적으로 제기된 5개 핵심 질문에 대한 최종 답변을 한 자리에 압축하고, 이 백서를 사내 도입 검토 문서로 전환하는 방법을 안내합니다. 독자가 본 장을 읽고 난 뒤 수행해야 할 구체적 행동 3가지도 함께 제시합니다.

12.1 5 핵심 질문에 대한 1페이지 요약 답변

Hermes Agent 도입 검토 과정에서 조직이 반복적으로 묻는 질문은 다섯 가지로 수렴합니다. 데이터가 외부로 나가는지, 기술적으로 신뢰할 근거가 있는지, 한국어를 실용 수준으로 처리하는지, 기존 오픈소스 대안과 어떻게 다른지, 그리고 실제 도입은 어떤 순서로 진행하는지입니다. 이 다섯 가지 질문은 기술 평가와 경영 의사결정을 동시에 아우르며, 각각에 대한 답변이 도입 여부를 결정하는 논거가 됩니다.

12.1.1 Q1 (데이터 주권) · Q2 (Hermes 기술 근거) 답변 카드

Q1 — 업무 데이터가 외부 클라우드로 전송되는가

Hermes Agent 를 온프레미스 또는 사내 폐쇄망에 배포하고 Local LLM 을 추론 백엔드로 구성하면 업무 데이터의 외부 송신을 구조적으로 차단할 수 있습니다. Hermes Agent 자체는 Nous Research 가 MIT 라이선스로 공개한 오픈소스이므로 소스코드를 직접 검토할 수 있으며, LiteLLM(MIT 라이선스)을 게이트웨이로 사용해 모든 추론 요청을 사내 엔드포인트로만 라우팅하는 구성이 기술적으로 완전히 지원됩니다 [S01] [S10]. Gemma 4 31B Dense, Qwen3(Apache 2.0), gpt-oss 20B(Apache 2.0) 등 상업적 사용이 허용된 오픈 가중치 모델을 온프레미스 GPU 서버에서 실행하면, 추론 단계에서도 외부 API 호출이 발생하지 않습니다 [S07] [S08] [S09]. 공공기관의 경우 망분리 환경에서의 오프라인 설치 절차(사내 미러 저장소 + 오프라인 tarball 방식)를 PoC 단계에서 별도 검증해야 하며, 이 절차는 부록 B 비교표에서 Air-Gapped 구성 항목으로 확인할 수 있습니다.

요약하면, 외부 클라우드 LLM 을 사용하지 않는 한 데이터 주권은 조직이 완전히 통제하는 구조입니다. 데이터 주권 요건을 충족하는지 여부는 LLM 선정 단계에서 결정되며, Hermes Agent 자체는 그 선택을 강제하지 않습니다.

Q2 — Hermes Agent 를 신뢰할 수 있는 기술적 근거는 무엇인가

Nous Research 가 2026년 2월에 공개한 Hermes Agent 는 MIT 라이선스 오픈소스로, 2026년 6월 기준 v0.17.0 이 출시되었으며 245명 이상의 외부 기여자와 1,475건 이상의 커밋이 누적되어 있습니다 [S01]. 단일 `curl` 명령으로 설치가 완료되고 월 5달러 수준의 저사양 서버부터 GPU 클러스터까지 동작 가능하다는 점은 진입 장벽을 낮추는 실용적 특성입니다. OpenRouter 플랫폼에서 토큰 사용량 1위를 기록했다는 제3자 비교 자료 역시 실사용자 저변을 방증합니다 [S04].

기술 구조 측면에서는 Profile(목적별 에이전트 인스턴스), Kanban(SQLite 기반 내구성 있는 작업 보드), Skill(자동 생성·보관 가능한 태스크 단위 자동화), Archive(장기 기억 계층)의 4가지 추상화가 핵심입니다 [S03]. 이 추상화 체계 위에 Curator 루프가 실행되어 에이전트가 스스로 새로운 Skill 을 생성하고 Archive 에 저장하는 자기 개선 메커니즘이 작동합니다 [S02]. 공식 슬로건인 "The longer it runs, the better it knows you" 는 이 메커니즘을 한 문장으로 압축한 표현입니다. 라이선스(MIT), 커뮤니티 규모, 4계층 추상화 구조, 자기 개선 루프 — 이 네 가지가 Hermes Agent 를 신뢰 가능한 기술 선택으로 뒷받침하는 근거입니다.

12.1.2 Q3 (Local LLM 한국어) · Q4 (경쟁 비교) 답변 카드

Q3 — Local LLM 은 한국어 처리를 실용 수준으로 지원하는가

Hermes Agent 자체는 추론 기능을 내장하지 않으며, 한국어 처리 성능은 백엔드 LLM 에 완전히 종속됩니다. 2026년 6월 기준 온프레미스 배포 가능한 주요 후보 3종의 특성은 다음과 같습니다.

모델	라이선스	한국어 지원	최소 메모리
Gemma 4 31B Dense	Apache 2.0	140개 언어, 텍스트+이미지+비디오 멀티모달	18~26GB VRAM (Q4_K_M)
Qwen3 시리즈	Apache 2.0	119개 언어 포함	모델 크기별 상이
gpt-oss 20B	Apache 2.0	다국어 지원	16GB (MoE 3.6B active)

Gemma 4 31B Dense 는 140개 언어를 지원하고 멀티모달(이미지·비디오·텍스트) 입력이 가능합니다. 2026년 4월 2일 공개된 Gemma 4 부터 라이선스가 **Apache 2.0** 으로 통일되어, 구 Gemma 3 27B 시기의 Gemma Terms of Use 와 달리 사내 법무 검토 부담이 사실상 사라졌습니다 [S07]. 상업적 이용·수정·재배포 모두 Qwen3 · gpt-oss 20B 와 동일한 조건에서 가능합니다. Qwen3 는 Apache 2.0 라이선스로 상업적 사용이 가장 자유로우며, 119개 언어를 지원하는 동시에 Qwen3-Omni 변형은 한국어 음성 입출력도 지원합니다 [S08]. gpt-oss 20B 는 OpenAI 가 2025년 8월 공개한 오픈 가중치 모델로, MoE 구조 덕분에 실제 활성 파라미터가 3.6B에 불과해 16GB 메모리에서 동작합니다 [S09].

단, 세 모델 모두 한국어 특화 공개 벤치마크(KMMLU 등)에서의 공인 수치가 충분하지 않습니다. 따라서 도입 전 조직 고유의 한국어 업무 시나리오로 자체 평가를 수행하는 것이 필수 요건입니다. PoC 4주 일정에 모델 한국어 성능 평가를 반드시 포함하십시오.

Q4 — 기존 오픈소스 에이전트 플랫폼과 Hermes Agent 는 어떻게 다른가

비교 대상으로 자주 거론되는 두 프로젝트의 포지션을 먼저 정리합니다. OpenClaw 는 24개 메시징 채널을 지원하고 247k GitHub 스타를 보유한 광폭 통합 플랫폼으로, "게이트웨이 안에 에이전트" 구조입니다 [S04]. Paperclip 은 기존 에이전트(Claude Code, OpenClaw, Codex 등)를 조직의 "직원"으로 채용·관리하는 조직화 레이어이며, 자체 에이전트를 생성하지 않습니다 [S05]. Hermes Agent 는 "게이트웨이 위에 학습 루프"를 올린 단일 에이전트 + 자기 개선 구조입니다. 세 프로젝트는 직접 경쟁하지 않으며 보완 관계에 있습니다.

엔터프라이즈 프레임워크 영역에서는 Microsoft Agent Framework(2026년 4월 GA), AWS Harness SDK, CrewAI(45,900+ 스타, 평균 지연 1.8초) 등이 있으나, 이들은 Hermes Agent 가 채우는 "단일 자율 에이전트 + 자기 개선" 포지션과 설계 목표 자체가 다릅니다 [S06]. 조직이 여러 에이전트를 대규모로 조율해야 한다면 Paperclip 이나 Microsoft Agent Framework 와 Hermes 를 조합하는 방식이 현실적입니다. Hermes Agent 는 단일 에이전트가 점진적으로 역량을 축적하며 자기 개선하는 시나리오에서 가장 뚜렷한 강점을 발휘합니다.

12.1.3 Q5 (도입 로드맵) 답변 카드와 다음 단계

Q5 — 도입은 어떤 순서로 진행하는가

도입 로드맵은 3단계로 구성됩니다 [S12]. 각 단계는 이전 단계의 결과를 입력으로 삼으며, 조직 규모와 리스크 허용도에 따라 일정은 조정 가능합니다.

첫 번째 단계는 4주 PoC(Proof of Concept)입니다. 단일 부서를 대상으로 Hermes Agent + Local LLM 조합을 구성하고, 한국어 처리 성능·보안 통제·기존 업무 시스템 연동 가능성을 실측합니다. LiteLLM 프록시 1대로 모든 추론 요청을 단일 엔드포인트로 라우팅하는 구성이 PoC 에 적합합니다 [S10]. 공공기관이라면 망분리 환경 오프라인 설치 절차도 이 단계에서 검증합니다. PoC 종료 시점에 "파일럿 진입 여부"를 의사결정합니다.

두 번째 단계는 3개월 파일럿입니다. PoC 에서 검증된 구성을 3~5개 부서로 확장하고, 부서 간 Kanban 연동·Profile 분리·Skill 공유 정책을 수립합니다. 이 단계에서 보안 감사·라이선스 컴플라이언스 검토·TCO(총소유비용) 측정을 병행합니다. 국내 공공 부문 사례를 참고하면, sLLM 자체 구축 초기 비용이 7~13억 원 범위로 추정되므로 파일럿 단계에서 실투입 비용 실측이 예산 정당화의 핵심 근거가 됩니다 [S12].

세 번째 단계는 전사 본격 운영입니다. 거버넌스 위원회를 구성하고 운영 정책(모델 업그레이드 절차, Skill 검수 기준, 보안 취약점 패치 일정, SBOM 관리)을 성문화합니다. MCP(Model Context Protocol, 모델 컨텍스트 프로토콜)가 벤더 중립 표준으로 채택되어 있으므로[S11], 사내 시스템을 MCP 서버로 노출하면 Hermes Agent 외에도 향후 도입할 다른 MCP 클라이언트와 즉시 연동됩니다. 이 표준 채택이 장기적 벤더 의존성을 낮추는 구조적 결정입니다.

다음 단계 — 백서를 읽은 직후 수행할 3가지 행동

- [ ] **PoC 시작 결정** — 담당 부서(IT 또는 디지털 전환)에 PoC 예산·인력·일정 승인을 요청합니다. 본 백서의 Q5 답변 카드와 11장 도입 로드맵을 품의서 첨부 자료로 활용합니다.
- [ ] **사내 워크숍 등록** — Hermes Agent 설치·구성·Skill 작성 실습 워크숍을 예약합니다. MSAP.ai 도입 컨설팅 팀이 제공하는 Hermes Agent 기술 세션을 통해 조직 맞춤 구성 가이드를 받을 수 있습니다.
- [ ] **거버넌스 위원회 구성** — 보안·법무·IT·비즈니스 대표 각 1인 이상으로 AI 거버넌스 위원회를 구성하고, 라이선스 컴플라이언스·데이터 주권·모델 업그레이드 정책의 1차 검토 일정을 잡습니다.

세 항목 중 한 가지라도 이 주 안에 시작하면 도입 타임라인이 평균 4~6주 단축되는 효과가 있습니다. PoC 결정이 가장 즉각적인 영향을 만들며, 거버넌스 위원회 구성은 병행해도 무방합니다.

12.2 본 백서 활용 가이드 — 사내 보고서·품의서 인용 방법

본 백서는 처음부터 "도입 검토 담당자가 사내 의사결정 문서를 작성할 때 그대로 인용할 수 있는 수준의 기술·운영·라이선스 정보를 한 권으로 묶는다"는 목적으로 기획되었습니다 [S12]. 그 목적을 실현하려면 백서 내용을 사내 문서 형식에 맞게 전환하는 방법을 명확히 알아야 합니다. 이 절에서는 인용 형식, 사내 문서 표준 구조, 추가 자료 접근 경로를 순서대로 안내합니다.

12.2.1 본 백서 인용 형식과 figure·table 재사용 권한

인용 자유도와 의무 표기

본 백서의 모든 내용은 사내 보고서·품의서·기술 검토서에 자유롭게 재사용할 수 있습니다. 단, 외부 발표 자료나 공개 블로그에 인용할 경우 출처를 표기해야 합니다 [S12]. 사내 문서에는 다음 세 가지 형식 중 맥락에 맞는 것을 선택합니다.

본문 인라인 인용 — 내용을 문장 안에 녹일 때 사용합니다.

> *예: "Hermes Agent 는 MIT 라이선스 오픈소스로, 2026년 6월 기준 v0.17.0 이 출시되었으며 245명 이상의 외부 기여자와 1,475건 이상의 커밋이 누적되어 있다 (출처: OPENMARU Hermes Agent 백서 2026, 1.2절)."*

각주 인용 — 수치나 표를 그대로 가져올 때 사용합니다.

> *예: "sLLM 자체 구축 초기 비용은 7~13억 원으로 추정된다.[1]" → 각주 [1]: OPENMARU Hermes Agent 백서 2026, 11.2절.*

부록 출처 표기 — 참고 문헌 목록에 단독 항목으로 추가할 때 사용합니다.

> *예: OPENMARU (2026). 오픈소스 AI Agent 오케스트레이션 Hermes Agent 백서. OPENMARU.*

표·도식 재사용 권한

본 백서의 모든 비교표, 아키텍처 도식, 로드맵 다이어그램은 OPENMARU 사내 문서에서 출처 표기 후 재사용할 수 있습니다. 외부 발표 자료에 삽입할 때도 "출처: OPENMARU Hermes Agent 백서 2026" 한 줄을 도식 하단에 표기하면 됩니다. 이미지 파일로 내보낼 때는 MSAP.ai 도입 컨설팅 팀에 원본 편집 파일을 요청하십시오.

12.2.2 사내 도입 품의서·기술 검토서 템플릿 구조

도입 품의서 — 5섹션 표준 구조

사내 도입 품의서는 의사결정권자가 검토하는 문서입니다. 읽는 시간이 제한적이므로 불필요한 기술 세부사항을 줄이고 판단에 필요한 정보를 전면에 배치합니다. 5섹션 표준 구조를 권장합니다.

1절 — 문제 정의: 현재 조직이 AI 에이전트 없이 수행하는 업무 중 자동화·가속화 대상을 명시합니다. 구체적인 부서·프로세스·반복 작업 항목을 열거하고 현재 소요 인력·시간을 정량화합니다.

2절 — 대안 비교: Hermes Agent, 클라우드 SaaS 에이전트, 자체 개발의 세 가지 경로를 비교합니다. 본 백서 3장의 경쟁 비교 표와 7장의 모델 선정 기준을 이 섹션에 그대로 인용할 수 있습니다.

3절 — 평가 기준: 데이터 주권, 한국어 처리 성능, 라이선스 컴플라이언스, 운영 복잡도, TCO의 5개 기준으로 각 대안을 채점합니다. 본 백서 12.1절의 Q1~Q4 답변이 이 섹션의 근거 자료입니다.

4절 — 결론: 권장 경로(Hermes Agent + Local LLM 구성)와 그 근거를 2~3문장으로 요약합니다. 의사결정권자가 결론만 읽어도 판단할 수 있도록 작성합니다.

5절 — 실행 계획: PoC → 파일럿 → 본격 운영의 3단계 일정, 각 단계 예산, 담당 조직을 표로 제시합니다. 본 백서 11장의 도입 로드맵과 12.1.3절의 Q5 답변 카드를 이 섹션에 활용합니다.

기술 검토서 — 5항목 표준 구조

기술 검토서는 IT 담당자와 보안 담당자가 상세 검토하는 문서입니다. 품의서보다 기술 세부사항 비중이 높습니다.

벤더 프로파일 섹션에는 Nous Research 법인 정보, GitHub 저장소 주소, 릴리스 주기, 커뮤니티 규모를 기재합니다. 라이선스 섹션에는 Hermes Agent(MIT), LiteLLM(MIT), 선택 LLM 별 라이선스(Qwen3 Apache 2.0, gpt-oss 20B Apache 2.0, Gemma 4 31B Dense Apache 2.0)를 목록화하고, 부록 D의 라이선스 발췌문을 첨부합니다. 보안 섹션에는 데이터 흐름 다이어그램, 외부 송신 여부, 망분리 구성 가능성을 기재합니다. 비용 섹션에는 인프라 비용(GPU 서버 또는 클라우드 인스턴스), LiteLLM 라이선스(OSS 무료 또는 Enterprise 연 최대 3,000만 원 수준), 인건비를 항목별로 분리합니다. 일정 섹션에는 PoC·파일럿·본격 운영의 단계별 마일스톤을 기재합니다.

그림 12-1. 도입 품의서·기술 검토서 표준 구조 + 본 백서 인용 매핑

본 백서 인용 매핑 완료 시 품의서 작성 시간 대폭 단축. MSAP.ai 컨설팅 팀 docx 템플릿은 이 매핑을 사전 포함 [S12]

위 구조는 두 문서의 섹션 구성과 본 백서 장별 인용 경로를 보여줍니다. 품의서의 각 섹션이 본 백서 어느 장에서 근거를 가져오는지 매핑하면 인용 작업 시간을 크게 줄일 수 있습니다. MSAP.ai 도입 컨설팅 팀이 제공하는 docx 템플릿 파일을 활용하면 매핑이 이미 완료된 상태로 시작할 수 있습니다.

12.2.3 부록 A~D 및 후속 자료 접근 안내

부록 A~D 위치와 활용 목적

본 백서에는 본문에서 요약된 내용의 원본 자료와 재사용 가능한 산출물을 네 개 부록으로 정리했습니다.

부록	내용	주요 활용
부록 A — Glossary	본 백서 전체에서 사용한 기술 용어 정의	사내 보고서 작성 시 용어 통일 기준
부록 B — 비교표 원본	3장의 경쟁 비교 표 원본 데이터 (수정 가능한 스프레드시트 형식)	품의서 대안 비교 섹션에 직접 삽입
부록 C — 참고 문헌	S01~S12 출처 전체 URL 및 접근 일자	외부 자료 검증·추가 조사
부록 D — 라이선스 발췌	MIT · Apache 2.0 · GPL 핵심 조항 발췌 (Gemma Terms of Use는 Gemma 4 부터 Apache 2.0으로 통합되어 보존 목적 1줄 각주만 포함)	기술 검토서 라이선스 섹션 첨부 자료

MSAP.ai 후속 자료 및 컨설팅 접근 경로

Hermes Agent 를 국내 환경에 도입하는 과정에서 보안 설계, 한국어 모델 평가, 망분리 구성, 라이선스 검토 등 각 단계별 전문 지원이 필요할 수 있습니다. MSAP.ai 는 오픈소스 AI 에이전트 플랫폼의 국내 도입을 지원하는 통합 플랫폼으로, Hermes Agent 기반의 구성 설계부터 파일럿 운영까지 도입 전 주기 컨설팅을 제공합니다 [S12].

Hermes Agent 기술 세션 일정, PoC 지원 문의, 사내 워크숍 예약은 opennaru.com 을 통해 신청할 수 있습니다. 도입 품의서·기술 검토서 docx 템플릿과 MSAP.ai 제품 카탈로그도 동일 경로에서 내려받을 수 있습니다.

2026년 공공기관 경영평가편람과 지방공기업 경영평가편람에 AI 활용 가점이 신설된 상황에서, 공공기관 10곳 중 7곳이 이미 AI 를 활용하고 있습니다 [S12]. 도입 검토를 미루는 것이 곧 경쟁 열위를 감수하는 선택이 되는 시점에 와 있습니다. 본 백서가 그 선택을 앞당기는 데 실질적인 역할을 했기를 바랍니다. 다음 단계는 PoC 시작 결정, 사내 워크숍 등록, 거버넌스 위원회 구성 — 세 가지 중 하나입니다.

부록

Appendix A — 참고문헌 (References)

본 백서 본문에 인용된 [S01]~[S12] 출처는 다음과 같습니다. 각 출처는 본문 작성 시점(2026년 6월) 기준 공식 또는 1차 자료입니다.

ID	영역	출처
[S01]	Hermes 정체	NousResearch/hermes-agent (GitHub 공식 저장소) — https://github.com/NousResearch/hermes-agent
[S02]	Hermes 정체	hermes-agent.org (Nous Research 공식 프로젝트 사이트)
[S03]	Hermes 정체	Hermes Agent 공식 문서 — Profiles · Kanban · Skill · Archive 추상화 명세
[S04]	경쟁 OSS	OpenClaw vs Hermes Agent 비교 분석 (Innfactory 리포트)
[S05]	경쟁 OSS	Paperclip — https://github.com/agencyenterprise/paperclip-ai , https://paperclip.ing
[S06]	경쟁 OSS	Harness Engineering 패러다임 — AWS Harness SDK · Microsoft Agent Framework · HKUDS OpenHarness · Mitchell Hashimoto "Agent = Model + Harness" 정의
[S07]	sLLM 동향	Google Gemma 4 31B Dense — blog.google/innovation-and-ai/technology/developers-tools/gemma-4/ · deepmind.google/models/gemma/gemma-4/ · ai.google.dev/gemma/docs/releases · Apache 2.0 라이선스
[S08]	sLLM 동향	Alibaba Qwen3 시리즈 — https://qwen-3.com · Qwen3-Omni 공식 모델 카드 · Apache 2.0 라이선스
[S09]	sLLM 동향	OpenAI gpt-oss 20B — https://github.com/openai/gpt-oss · OpenAI 공식 모델 카드 · Apache 2.0 라이선스
[S10]	게이트웨이	BerriAI/litellm — https://github.com/BerriAI/litellm · LiteLLM Enterprise 가격·기능 안내

ID	영역	출처
[S11]	표준	Model Context Protocol — Wikipedia · WorkOS 분석 · The New Stack 채택 동향 · Linux Foundation AAIF 거버넌스 이관
[S12]	국내·라이선스	스켈터랩스 BELLA LLM · LLM Capsule 공공 도입 사례 · Mend 라이선스 가이드 · Synopsys OSSRA 2025 리포트 · 2026 공공기관 경영평가편람

Appendix B — 용어 정의 (Glossary)

본 백서 본문에서 처음 등장한 주요 기술 용어를 가나다·알파벳 순으로 정리합니다.

용어	정의
Agentic AI Foundation (AAIF)	Anthropic · Block · OpenAI 가 2025년 12월 공동 설립한 Linux Foundation 산하 단체. MCP 표준의 vendor-neutral 거버넌스를 담당.
Apache 2.0	오픈소스 라이선스. 상업적 이용·수정·재배포 허용, 저작권 고지와 변경 명시 의무, 특허 허여 조항 포함. Qwen3 와 gpt-oss 20B 가 채택.
Archive	Hermes Agent 의 영속 기억 계층. 대화 로그·산출물·결정 근거를 파일 또는 데이터베이스에 보존하여 감사 추적과 Curator loop 학습 자산 축적 두 역할을 수행.
Curator loop	Hermes Agent 의 자기 개선 메커니즘. 15회 tool call 또는 복잡 작업 완료 시점에 Curator 가 회고를 수행하고 재사용 가능한 Skill 파일을 자동 생성·저장.
Fan-out	Multi-agent 패턴 중 하나. 단일 Coordinator 가 동일 작업을 복수 Worker 에게 동시 배포하고 결과를 취합하는 병렬 분기 구조.
Gemma 4	Google DeepMind 가 2026년 4월 2일 공개한 오픈 가중치 LLM 패밀리. **Apache 2.0** 라이선스로 전환되어 구 Gemma 3 27B 시기의 Gemma Terms of Use 가 폐기. 4 사이즈 (E2B · E4B · 26B A4B MoE · 31B Dense) + 256K 컨텍스트 + 텍스트/이미지/비디오 멀티모달 (E2B/E4B 는 오디오 입력 추가).

용어	정의
Gemma Terms of Use	구 Gemma 3 27B 까지 적용되던 Google 별도 라이선스. 상업적 이용을 "책임 있는 사용" 조건부로 허용. Gemma 4 (2026-04-02) 부터 Apache 2.0 으로 통합되어 신규 도입 검토 대상 아님. 본 백서는 이력 보존 목적으로 항목 유지.
HITL (Human-in-the-loop)	자율 에이전트 실행 흐름에 사람이 검토·승인·수정으로 참여하는 구조. PoC 단계에서 비중을 높게 유지하다가 신뢰가 쌓이면 점진적으로 축소.
Harness Engineering	Mitchell Hashimoto 가 2026년 정립한 패러다임 — AI 에이전트 행동을 지배하는 제어 시스템을 설계·유지하는 분야. Guides · Sensors · Context Pipelines 3요소로 구성.
Kanban	Hermes Agent 의 작업 보드 추상화. SQLite 기반으로 7개 컬럼(triage · todo · ready · running · blocked · done · archived)과 카드 5속성(assignee · dependency · workspace kind · tenant · comment thread)을 관리.
KMMLU (Korean Massive Multitask Language Understanding)	한국어 대규모 멀티태스크 언어 이해 벤치마크. Local LLM 의 한국어 처리 품질 자체 평가에 활용.
LiteLLM	MIT 라이선스 오픈소스 LLM 게이트웨이. 100개 이상의 LLM 공급자를 단일 OpenAI 호환 엔드포인트로 통합. virtual key · 예산 추적 · fallback 라우팅 제공.
Local LLM	외부 클라우드가 아닌 자체 인프라(사내 GPU 서버 또는 워크스테이션)에 배포한 언어 모델. 데이터 외부 송신 차단·비용 통제·지연 단축이 도입 동기.
MCP (Model Context Protocol)	2024년 11월 Anthropic 이 발표한 AI 모델과 외부 도구·데이터 연결 표준. 2025년 12월 Linux Foundation AAIF 로 거버넌스 이관. 2026년 3월 기준 10,000+ public servers.
MIT 라이선스	가장 단순한 오픈소스 라이선스. 사용·수정·재배포·상업적 이용 모두 허용, 단일 의무는 저작권 고지 보존. Hermes Agent · LiteLLM 본체가 채택.
MoE (Mixture of Experts)	모델 내부에 복수의 전문가 서브네트워크를 두고 토큰별로 일부만 활성화하는 구조. gpt-oss 20B 는 총 21B 파라미터 중 추론 시 3.6B 만 활성화.

용어	정의
MSAP.ai	OPENMARU 가 제공하는 국내 통합 AI 게이트웨이·에이전트 플랫폼. Hermes Agent · LiteLLM · Local LLM 결합 구성을 사전 통합하여 도입 부담을 낮춤.
Persistent Memory	세션이 끊겨도 사용자 선호·작업 맥락을 유지하는 영속 기억 계층. Hermes Agent 의 Curator loop 와 결합되어 운영 시간이 쌓일수록 개인화 수준이 향상.
Pipeline	Multi-agent 패턴 중 하나. 에이전트 A 의 산출물을 에이전트 B 가 이어받는 직렬 인계 구조. CI/CD 흐름·문서 편집 파이프라인에 적합.
PoC (Proof of Concept)	신기술 도입 검토 단계의 개념 검증. 본 백서 로드맵에서는 단일 부서·단일 유즈케이스 4주 일정으로 정의.
Profile	Hermes Agent 의 목적별 에이전트 인스턴스 격리 단위. 별도의 메모리·Skill 묶음·Tool 권한·Channel 슬롯을 가지며 사내 RBAC 의 단위 컨테이너 역할.
Q4_K_M	llama.cpp 생태계의 4비트 K-means 양자화 방식. 모델 VRAM 사용량을 FP16 대비 약 60~65% 감소시키되 품질 손실을 2~5% 수준으로 억제.
RAG (Retrieval-Augmented Generation)	검색 증강 생성. 외부 문서 저장소에서 관련 정보를 추출해 LLM 컨텍스트에 추가한 뒤 응답을 생성하는 기법.
RBAC (Role-Based Access Control)	역할 기반 접근 통제. Hermes Agent 에서는 Profile 단위 Skill·Tool 권한 분리로 구현.
SBOM (Software Bill of Materials)	소프트웨어 구성 명세서. 종속 패키지와 라이선스를 목록화하여 라이선스 충돌·보안 취약점을 점검하는 기준 문서.
SIEM (Security Information and Event Management)	보안 정보 및 이벤트 관리. Hermes Agent 통합 감사 로그를 Elasticsearch · Splunk · QRadar 등 SIEM 으로 전송하여 채널 통합 감사 추적을 구현.
Skill	Hermes Agent 의 재사용 가능한 업무 절차 단위. `SKILL.md` 마크다운 파일(YAML frontmatter + 본문 + 참조 스크립트)로 정의되며 코드 수정 없이 추가 가능.

용어	정의
sLLM (Small Language Model)	수십억 파라미터 규모로 단일 GPU 서버에서 운영 가능하게 설계된 소형 LLM. 본 백서에서는 사내 배포 맥락에서 Local LLM 과 함께 사용.
SSO (Single Sign-On)	단일 인증. 사내 디렉터리(LDAP · Active Directory)와 Hermes Agent 사용자 매핑을 동기화하면 인사 변동에 따라 권한이 자동 갱신.
TTFT (Time-To-First-Token)	LLM 추론에서 첫 토큰 응답까지의 지연. vLLM 기준 27B 모델 일반 업무 요청의 TTFT 2초 이하가 정상 범위.
vLLM	고성능 LLM 추론 프레임워크. PagedAttention · 연속 배치(continuous batching) 로 HuggingFace `generate()` 대비 10~25배 처리량을 제공.
VRAM (Video RAM)	GPU 전용 메모리. LLM 가중치를 VRAM 에 올려야 추론 가능. 모델 크기와 양자화 수준에 따라 16GB · 24GB · 48GB · 80GB 4등급으로 구분.

오픈소스 AI Agent 오케스트레이션 Hermes Agent

CONTACT

WEB
msap.ai
www.msap.ai/

EMAIL
hello@msap.ai

TEL
02-6953-5427
0269535427

YOUTUBE
@msaptv
www.youtube.com/@msaptv

LINKEDIN
linkedin.com/showcas...
www.linkedin.com/showcase/msap-ai/

FACEBOOK
facebook.com/opennaru
www.facebook.com/opennaru

SCAN