

Hermes Agent 가 제공하는 기업용 AI 에이전트 필수 기능

Hermes Agent 공식 문서 기준 도입 검토 자료

OWASP 위험 분류 10개 · 필수 기능 8가지 항목 · 제품 기본 제공 일곱과 조직 몫 하나

PROJECT CODE

AGENT-REVIEW-2026

DATE

2026-08-25

KEY INSIGHT

4개 파트가 순서대로 이어지는 구조 — 앞 파트에서 세운 검토 기준을 뒤 파트가 그대로 사용

검토 기준

PART 01

제품 구성

PART 02

항목별 확인

PART 03

도입 판단

PART 04



검토 기준 세우기

OWASP 위험 분류 10개
실행 단계 다섯으로 압축
기능 요구 8가지 항목 확정

5장



제품 구성 확인

오픈소스 배포 형태
설치 전제 3가지
코어 · 스킴 · 3개 축

3장



항목별 근거 대조

공식 문서가 밝힌 범위
수치와 설정 기본값
남는 빈칸 표시

11장



도입 판단 조건

제품 몫과 조직 몫 구분
도입 전 확인 3가지 조건
공개 사례의 한계

3장

SECTION 01

기업이 AI 에이전트에 요구하는 8가지 항목

OWASP 위험 분류 10개에서 뽑아낸 도입 검토 기준

챗봇과 에이전트

OWASP 위험 10개 범주

기능 요구 8개

KEY INSIGHT

되돌리기 어려운 동작의 폭이 통제 범위를 결정 — 위임한 동작 목록을 먼저 적어야 검토 기준이 정해집니다



응답형 도구 (챗봇)

사람이 매 단계를 지시하는 방식

- 실행 단위**
요청 하나에 응답 하나로 종료
- 다음 단계 결정**
언제나 사람이 판단하고 지시
- 실행이 남기는 것**
대화창에 표시된 텍스트

되돌리는 비용 — **낮음** 표시된 텍스트만 남는 실행



자율 실행 에이전트

목표만 받고 실행을 이어 가는 방식

- 실행 단위**
목표 하나로 여러 단계를 연속 수행
- 다음 단계 결정**
도구 선택과 중단 시점까지 스스로 판단
- 실행이 남기는 것**
파일 변경 · 외부 호출 · 메시지 발송

되돌리는 비용 — **높음** 시스템에 상태가 남는 실행

도입 심사에서 가장 먼저 적을 것 — 그 에이전트에 실제로 위임한 동작의 목록. 목록의 폭이 이후에 걸어야 할 통제의 범위를 그대로 결정

KEY INSIGHT

4개 부품 중 하나만 빠져도 반복 업무는 사람 손으로 회귀 — 자기 업무에 필요한 만큼만 저울질하는 틀



절차 기억

같은 일을 다시 설명하지
않도록 절차를 저장

빠지면

요청마다 맥락 재설명 —
말기는 이득 소멸



채널

담당자가 자리에 없어도
요청이 들어오는 입구

빠지면

근무 시간에만 요청 수신 —
반복이라는 말이 무색



무인 실행

단계마다 개입 없이
목표까지 실행을 지속

빠지면

사람이 단계마다 확인 —
자동화라 부를 것이 없음



검증

실행이 끝난 뒤 결과가
맞는지 확인하는 절차

빠지면

확인 없는 결과가
다음 업무 입력으로 유입

업무마다 달라지는 비중

하루 한 번 결과만 확인하는 정산 업무 — 검증 비중이 크고 무인 실행 비중은 작음
야간 요청을 아침까지 처리해야 하는 운영 업무 — 채널과 무인 실행이 핵심 요건

KEY INSIGHT

10개 범주가 2가지 질문으로 압축 — 실행을 승인한 주체, 그리고 결과를 확인하는 주체

01

목표 탈취

지시를 조작해 목표를 돌려놓는 위험

02

도구 오남용

연결한 도구를 의도 밖까지 쓰는 위험

03

신원과 권한 남용

위임받은 권한을 넘어 동작하는 위험

04

메모리 오염

저장된 맥락이 오염돼 이후 실행까지 번지는 위험

05

임의 코드 실행

코드나 명령이 검사 없이 실행되는 위험

06

공급망

밖에서 들어온 스킬과 모델이 옮기는 위험

07

에이전트 간 통신

주고받는 메시지를 타고 조작이 퍼지는 위험

08

연쇄 장애

한 요소의 실패가 번져 전체가 멈추는 위험

09

신뢰 착취

에이전트를 믿는다는 사실을 노리는 위험

10

불량 에이전트

통제 밖 에이전트가 조직 안에 남는 위험

질문 하나 — 이 실행을 누가 승인했나

답이 없을 때 커지는 위험: 목표 탈취 · 메모리 오염 · 신뢰 착취

질문 둘 — 그 결과를 누가 확인하나

답이 없을 때 커지는 위험: 도구 오남용 · 임의 코드 실행 · 연쇄 장애

KEY INSIGHT

범주는 위험의 이름, 점검표에 필요한 것은 기능의 이름 — 실행 단계를 거쳐야 검토 항목으로 전환

실행 단계	대표 위험 범주	대응하는 기능 요구
1 누가 실행을 시작하나	목표 탈취 · 신뢰 착취 · 불량 에이전트	채널 통합권한 분리
2 무엇을 반복해 실행하나	메모리 오염	절차 기억무인 실행
3 무엇을 만지나	도구 오남용 · 임의 코드 실행	실행 격리위험 명령 승인
4 어디로 나가나	공급망 · 에이전트 간 통신	데이터 처리 위치 공개
5 누가 확인하나	연쇄 장애 · 신원과 권한 남용	실행 결과 검증뒤에서 다시 확인할 항목

Source: OWASP Top 10 for Agentic Applications 2026 [S10]

한 범주가 여러 항목으로 흩어지고 한 항목이 여러 범주를 함께 받으므로 일대일 대응으로 읽지 않기

KEY INSIGHT

8가지 이름을 끝까지 고정 — 장마다 다른 말로 바꿔 부르면 같은 요구를 두 번 검토하거나 통째로 누락

1

**절차 기억**

수행 절차를 저장해
다시 쓰는 능력

PART 03

2

**채널 통합**

직원이 이미 쓰는
메신저에서 여는 입구

PART 03

3

**무인 실행**

단계마다 사람 개입 없이
목표까지 잇는 실행

PART 03

4

**실행 격리**

도구가 손댈 범위를
컨테이너 안으로 한정

PART 03

5

**데이터 처리 위치 공개**

조직 밖으로 나가는
데이터 경로 명시

PART 03

6

**권한 분리**

사람과 역할마다 다른
사용 범위 부여

PART 03

7

**위험 명령 승인**

되돌리기 어려운 명령
직전에 실행 정지

PART 03

8

**실행 결과 검증**

실행이 끝난 뒤 결과가
맞는지 확인하는 절차

PART 04

SECTION 02

오픈소스 런타임 Hermes Agent의 구성 요소

배포 형태 · 설치 전제 · 코어와 스킬 · 교체 가능한 3개 축

설치 전제 3가지

코어 · 스킬 · 3개 축

8가지 항목 대조표

KEY INSIGHT

전제는 Python 3.11 · Node.js · ripgrep 셋뿐 — CPU · 메모리 · 디스크 수치는 공식 문서에 명시 없음



Python 3.11

에이전트 코어를 실행하는 런타임



Node.js

연동 도구 실행에 필요한 런타임



ripgrep

코드·문서를 빠르게 훑는 검색 도구

✓ 공식 문서가 밝힌 실행 범위

- 5달러짜리 가상 사설 서버(VPS)에서도 실행 가능
- GPU 클러스터 · 서버리스 인프라까지 지원 범위 선언
- 시범 단계와 운영 단계가 같은 코드베이스에서 출발

도입 초기에 인프라 사양을 미리 정하지 않아도 검토 착수 가능

! 공식 문서가 밝히지 않은 값

- 최소 CPU 코어 수 — 명시 없음
- 최소 메모리 용량 — 명시 없음
- 최소 디스크 용량 — 명시 없음

작업량에 맞는 사양은 시험 설치를 거쳐 조직이 직접 측정



상시 구동을 계획한다면 먼저 확인할 조건

WSL(Windows Subsystem for Linux)의 systemd 지원이 불안정 — 재시작이나 유희 종료 뒤 서비스 중단 가능. 실제 배치는 리눅스 서버나 컨테이너 환경 권장

KEY INSIGHT

3개 축이 따로 나뉜 구조 — 모델을 바꿔도 채널 설정과 스킬은 그대로 유지



교체가 실제로 뜻하는 것

모델 제공자를 바꿔도 채널 설정과 스킬은 그대로 남고, 검색 백엔드를 교체해도 코어의 실행 방식은 변화 없음

다만 모델을 바꾸는 순간 프롬프트 캐시 초기화 — 전환 직후 첫 요청은 대화 전체를 다시 읽는 비용 발생

KEY INSIGHT

8번째 실행 결과 검증만 공식 문서 어디에도 없음 — 나머지 일곱은 보안 문서와 기능 문서에 기능으로 명시

번호	항목	제품 기본 제공	근거 출처	자세히 다루는 쪽
1	절차 기억	제공	[S4] 스킬 문서 · [S5] FAQ	14 – 15쪽
2	채널 통합	제공	[S6] 기능 개요 문서	16쪽
3	무인 실행	제공	[S6] 기능 개요 문서	17 – 18쪽
4	실행 격리	제공	[S3] 보안 문서	19 – 20쪽
5	데이터 처리 위치 공개	제공	[S6] 기능 개요 · [S7] 연동 문서	21 – 22쪽
6	권한 분리	제공	[S3] 보안 문서	23쪽
7	위험 명령 승인	제공	[S3] 보안 문서	23쪽
8	실행 결과 검증	제공 없음	근거로 삼을 공식 기술 없음	24쪽 · PART 04

SECTION 03

Hermes Agent 주요 기능과 제약

스킬 · 채널 · 무인 실행 · 도구 격리 · 모델 · 데이터 경로 · 안전장치

스킬 · 채널 · 무인 실행

도구 격리 · 모델 · 데이터 경로

안전장치 8개 계층

KEY INSIGHT

스킬이 늘어도 실행 한 번의 입력 비용은 정비례하지 않음 — 필요한 본문만 그때 펼치는 적재 방식

조건 1



점진적 공개 (progressive disclosure)

- 제목과 요약은 먼저 훑고 필요한 본문만 그때 펼침
- 10개를 쌓은 조직과 백 개를 쌓은 조직의 입력 비용 차이가 스킬 개수에 비례하지 않음
- 절차가 사람 기억 대신 문서에 남아 담당자 교체에도 같은 품질의 실행 유지

이 조건이 답하는 실무 질문
같은 설명을 매번 반복해야 하는가

조건 2



agentskills.io 공개 표준 호환

- 제품 전용 스키마 대신 외부 공개 명세를 따르는 형식
- 사람이 읽는 마크다운 문서 — 별도 뷰어 불필요
- 형상 관리 저장소에 그대로 넣어 변경 이력 관리 가능
- 도구를 바꾸어도 스킬 파일 자체는 잔존

이 조건이 답하는 실무 질문
제품을 바꾸면 지금까지 만든 스킬이 사라지는가

! 공식 문서가 답하지 않는 몫

스킬이 수십 개로 늘었을 때 관련 없는 문서를 얼마나 걸러 내는지, 잘못 고른 스킬로 작업이 어긋나는 비율이 얼마인지에 대한 수치 없음
이름과 요약을 서로 구별되게 쓰고 내용이 겹치는 문서를 합치는 작업은 도입 조직의 몫

KEY INSIGHT

origin hash로 로컬 수정 여부는 대조 가능 — 업데이트가 수정분을 지키는지는 명시가 없어 시험 프로파일 확인 필요

1 실행 중 자동 작성



- 지시나 트리거 없이 남길 만하다고 판단하면 문서화
- 실행과 기록이 같은 시점 — 결정 근거까지 함께 저장
- 축적 속도가 관리자의 문서화 일정과 무관

2 세션 종료 후 자기개선 리뷰



- 백그라운드 검토가 방금 처리한 작업을 되짚어 점검
- 찾아낸 개선점을 다시 스킬 문서로 저장
- 사용자가 화면을 보지 않는 동안에도 절차 기억 증가

프론트매터 version 필드

1.0.0 형식의 버전 표기
사람이 읽는 식별자 역할

origin hash

배포 원본과 현재 파일이 같은지
기계가 판정하는 대조 값

hermes update 명령

갱신 스킬을 모든 프로파일에
자동 동기화 — 배포 스크립트 불필요

! 도입 조직이 채워야 할 몫

업데이트가 해시로 추적되지 않는 변경분을 그대로 두는지 배포본으로 되돌리는지 공식 문서에 명시 없음 — 시험용 프로파일 하나로 먼저 확인
사람이 지켜보지 않는 동안 잘못된 절차도 함께 굳으므로, 자기개선 리뷰가 만든 스킬을 정기 검토 대상에 포함

KEY INSIGHT

3가지 제한이 함께 걸릴 때 무차별 대입 차단 — 코드 경우의 수 1조는 그 자체로 재시도를 막지 못함

코드 길이

8글자

혼동하기 쉬운 글자를
뺀 32자 알파벳

유효 시간 (TTL)

1시간

발급된 코드가
만료되기까지의 시간

요청 제한

10분 1회

사용자당 코드 요청
간격 하한

대기 코드 상한

3개

플랫폼마다 동시에
남을 수 있는 코드 수

잠금 조건

5회 실패

이후 해당 사용자를
1시간 잠금



사내 기준에 대보는 방법

- 사내 규정이 일회용 코드 유효 시간을 30분 이내로 제한한다면 1시간 값은 규정과 상충 — 예외 승인 또는 설정 조정 여부를 먼저 결정
- 별도 기준이 없다면 5가지 값을 사내 표준 초안으로 사용



이 절차가 지키는 범위

- 누가 에이전트에 말을 걸 수 있는지까지만 통제
- 말을 건 사람이 무엇까지 시킬 수 있는지는 권한 분리의 몫
- 짝짓기 인증은 입구를 잠그는 절차, 권한 분리는 들어온 사람이 손댈 범위를 정하는 절차 — 함께 설계

공식 문서가 밝히는 범위는 메신저에서 부를 수 있다는 사실까지 — 메시지가 어느 서버를 지나나는지는 명시 없음(데이터 경로에서 다시 확인)

KEY INSIGHT

무인 실행은 설치만으로 켜지지 않는 기능 — Windows·WSL 환경은 하룻밤 실제 구동 시험 뒤 판단

상시 구동을 위해 조직이 먼저 정할 3가지

- 1 프로세스 관리자**
요청이 들어오는 내내 프로세스가 살아 있도록 관리하는 주체
- 2 재시작 정책**
중단이 생겼을 때 자동으로 복구하는 규칙
- 3 로그 보존 기간**
밤사이 실행을 나중에 되짚을 기록의 보관 기간

무인 실행이 성립하는 조건

밤 11시 요청 수신 → 새벽 2시 작업 종료까지 프로세스 상주 필요
중간에 호스트가 절전에 들어가면 요청은 미처리 상태로 잔류

3가지가 정해지지 않은 채 채널만 붙이면

낮에는 잘 돌던 에이전트가 밤에는 응답하지 않는 상태가 반복 — 원인 추적에 사람 시간이 더 소모

⚠ WSL systemd 제약

- 많은 WSL2 설치에서 systemd 비활성 상태
- 켜 두어도 WSL 재시작이나 Windows 유틸 종료를 넘기지 못하는 사례 존재
- 유틸 종료 조건이 맞을 때만 발생해 재현 곤란
- 원인이 에이전트인지 WSL인지 전원 설정인지 가려내는 데 상당한 시간 소요

대안 구성

상시 구동은 별도 리눅스 호스트나 상시 가동 서버로 분리하고, Windows 쪽은 요청을 넣는 채널 용도로만 사용

도입 판단 전 절차

하룻밤 실제로 구동하고 다음 날 아침 프로세스 생존 여부 확인

KEY INSIGHT

얼마나 썼는지는 확인 가능, 무엇을 제대로 했는지는 확인 불가 — 무인 실행 건수가 늘수록 사람 확인 부담이 함께 증가

 이미 갖춰진 계측 4개 조각

토큰 회계

요청마다 소비한 토큰량 집계

추정 비용

토큰량을 기준으로 산출한 예상 비용

세션 길이

한 세션이 이어진 길이 — 비정상 확대 시 확인 단서

사용 요약

insights 명령으로 확인하는 사용 현황

도입 초기에는 이 4가지만으로도 실제 작업량을 경영진에게 보고 가능

 아직 없는 계측 계층

- 조각을 하나로 모으는 통합 텔레메트리 계층 부재
- 하나의 무인 실행을 처음부터 끝까지 이어 되짚는 화면 없음
- 산출물이 요구대로 나왔는지는 결국 사람이 파일을 직접 열어 확인
- 무인 실행이 하루 수십 건으로 늘면 그 확인 작업 자체가 새로운 업무로 전환

8번째 항목 실행 결과 검증의 부재와 같은 결함

기록이 남는 위치

개발 가이드 정책상 사용자가 켜기 전까지 외부로 나가는 텔레메트리를 새로 추가하지 않으므로, 실행 기록은 기본적으로 조직 안에 잔류
기록이 밖으로 새지 않는 점은 이점 — 다만 그 기록을 결과 검증에 쓰려면 조직이 직접 모으고 대조하는 절차 필요

KEY INSIGHT

web.backend 를 비워 두면 등록된 API 키가 백엔드를 결정 — 설정을 건드리지 않아도 외부 호출 경로는 이미 작동



웹 검색 · 추출 백엔드

바깥 정보를 읽어 오는 경로
선택지를 가장 뚜렷하게 공개



메시징 플랫폼

사람이 에이전트를 부르는 입구
세부 경로는 명시 없음



모델 제공자

판단을 맡길 모델을 받는 곳
3개 경로 공개 (21쪽)



MCP 서버

사내 시스템을 도구로 드러내는
표준 연결 규격

검색 · 추출 백엔드 8종

기본값 Firecrawl · 자체 운영 가능한 SearXNG 포함

Firecrawl

기본값

SearXNG

자체 운영 가능

Brave

DuckDuckGo

Tavily

Exa

Parallel

xAI



설정을 비워 두면 일어나는 일

- web.backend 값을 지정하지 않으면 사용 가능한 API 키를 기준으로 백엔드를 자동 선택
- 다른 업무용 Brave API 키를 같은 환경 변수에 등록해 두면 검색 트래픽이 조직이 예상하지 못한 대상으로 유출

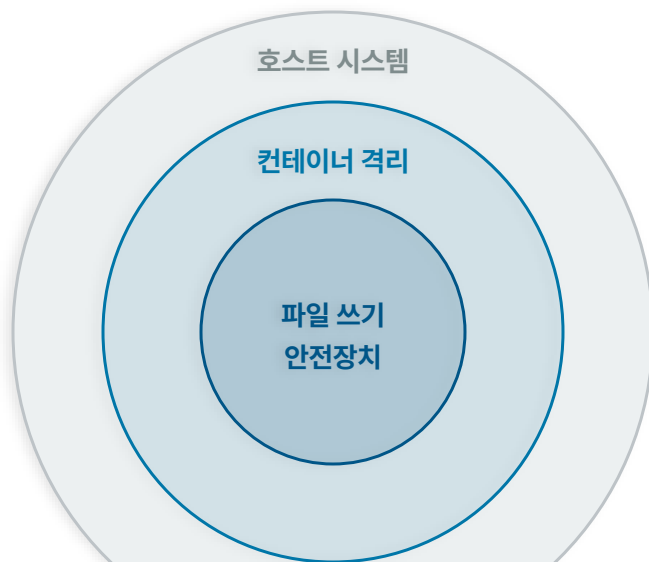


조직이 정해야 할 3가지

- 1 검색·추출 백엔드 선택 — 데이터가 어디로 나가는지의 결정
- 2 web.backend 값 명시 후 등록된 키와 정기 대조
- 3 컨테이너에 마운트할 디렉터리와 열어 줄 네트워크 범위

KEY INSIGHT

2개 계층 모두 설치 직후부터 작동하는 기준선 — 마운트한 디렉터리와 열어 준 네트워크가 실효 범위를 결정



2개 계층이 같은 실행 경로 위에 겹쳐 있어
하나를 넘어선 명령도 나머지에서 재차 차단

8개 계층 중 4번째

컨테이너 격리

- 에이전트가 만든 셸 명령은 호스트 셸 대신 컨테이너 안에서 프로세스로 기동
- 경계가 그어지는 단계는 명령 문자열을 만든 직후
- 모델이 잘못된 명령을 만들어도 호스트의 다른 서비스나 사내망 타 서버로 미확산

8개 계층 중 3번째

파일 쓰기 안전장치

- 컨테이너 안에서도 손대면 안 되는 파일을 지키는 계층
- 스킴 저장 · 설정 변경 · 산출물 기록이 모두 파일 쓰기로 발생
- 오래된 로그를 정리하라는 지시가 경로를 잘못 잡아도 검사 단계에서 차단

조작이 정할 몫 — 컨테이너에 마운트할 디렉터리와 열어 준 네트워크만큼은 에이전트가 그대로 열람 가능하며, 이 구성에 대한 정량 기준은 공식 문서에 없음

KEY INSIGHT

제공자 선택이 곧 데이터 처리 위치 선택 — 프롬프트가 어느 회사 인프라를 거치는지가 선택의 실질

네이티브 Anthropic 연동

중계자 없는 직접 연동

호출 창구 Anthropic API 직접 호출
 계약 성격 Anthropic과 직접 계약, 중계자 없음
 사내에 이미 엔터프라이즈 계약이 있다면 결재 절차가 가장 짧음

OpenRouter

여러 모델을 한 API로 중계

호출 창구 OpenRouter API 경유
 계약 성격 외부 라우팅 계층 한 곳과 계약
 여러 모델을 바꿔 가며 성능을 비교하려는 조직에 적합

Nous Portal

개발사가 직접 운영

호출 창구 Nous Research 운영 게이트웨이
 계약 성격 개발사가 운영하는 게이트웨이 계약
 별도 계약 없이 시범 운영부터 시작하려는 조직에 문턱이 가장 낮음

 모델 전환에 드는 비용

전체 재계산 1회

모델을 바꿀 때마다 발생

- 캐시 키에 모델 식별자가 포함돼 전환 즉시 프롬프트 캐시 초기화
- 전환 직후 첫 메시지는 대화 전체를 다시 읽고 할인 없는 전체 입력 가격 청구
- 대화가 길수록, 전환이 잦을수록 손실 확대

조직이 미리 정할 운영 규칙

상시 전환 방식 또는 업무 단위 모델 고정 방식 중 택일

예산 담당자가 잡아야 할 전제

짧은 단발 질의 용도라면 체감 없음 — 며칠에 걸친 긴 세션에서 모델을 자주 오가면 전환 횟수만큼 전체 재계산 비용이 청구서에 누적

KEY INSIGHT

코어를 사내 서버에 올려도 4개 경로의 목적지는 경로별 설정으로 별도 결정 — 프로세스 위치와 외부 호출은 서로 다른 질문

경로	기본값	전환 가능 여부	근거의 성격
 검색 · 추출 백엔드	Firecrawl API 키를 기준으로 자동 선택	SearXNG 등 7종 대안 가운데 선택해 교체 가능	공식 문서로 확정
 모델 제공자	구성 시 지정 Anthropic / OpenRouter / Nous Portal	3개 경로 중 자유 선택 전환 시 프롬프트 캐시 초기화	공식 문서로 확정
 사용 정보 수집	신규 계층은 선택 동의(opt-in) 게이트가 갖춰지기 전까지 병합 금지	토큰 회계 · 비용 추정 · 세션 길이 조각 존재 — 통합 계층 없음	정책과 현황이 같림
 메시징 게이트웨이	연동한 플랫폼의 서버 경유 채널마다 운영 주체가 다른 별개 사업자	채널 선택에 따라 상이 세부 경로는 공개 문서에 없음	세부 명시 없음
정책과 현황을 하나로 뭉쳐 수집 없음으로 단정하면 감사나 실사에서 근거 제시 불가 — 정책 한 줄과 현황 한 줄로 분리해 기재			

KEY INSIGHT

3개 묶음으로 읽기 — 앞 둘은 실행 주체, 가운데 셋은 실행 범위, 나머지 셋은 드나드는 데이터

실행 주체 통제

계층 1-2

1 사용자 인가

요청을 보낸 사람에게 사용 자격이 있는지 실행 이전에 확인

2 위험 명령 승인

삭제·전송처럼 되돌리기 어려운 명령을 실행 직전에 정지

2개 계층이 함께 서는 방법

야간 무인 실행 중에도 되돌리기 어려운 명령 앞에서는 승인 대기

실행 범위 통제

계층 3 · 4 · 7

3 파일 쓰기 안전장치

쓸 수 있는 파일 범위를 사전 제한

4 컨테이너 격리

실행 환경을 호스트 시스템과 분리

7 세션 간 격리

서로 다른 세션의 실행이 서로에게 개입하지 못하도록 분리

앞서 다룬 2개 계층을 여기서는 8개 계층 안의 위치로 재배치

데이터 · 입력 검사

계층 5 · 6 · 8

5 MCP 자격증명 필터링

호출 시 오가는 API 키·토큰 유출 차단

6 컨텍스트 파일 스캔

읽어 들이는 파일에 숨은 지시문이 섞였는지 검사

8 입력 정제

채널로 들어오는 입력에서 위험한 문자열과 패턴을 앞단에서 제거

외부 연동이 늘수록 검사 대상도 함께 증가

! 공식 문서가 밝히지 않은 부분과 다음 쪽으로 넘기는 몫

세션 간 격리의 분할 단위와 미공유 대상은 명시 없음 — 여러 팀이 한 인스턴스를 공유할 계획이라면 세션 2개를 동시 기동해 직접 확인
8개 계층 어디에도 실행이 끝난 뒤 결과가 맞는지 확인하는 계층 없음 — 다음 쪽에서 그 경계를 확인

KEY INSIGHT

8개 계층은 권한과 범위만 통제 — 잘못 요약한 보고서는 명령 자체가 안전해 승인 게이트에 미검출

✓ 8개 계층이 다루는 물음

- 누가 실행을 시작할 수 있는가
- 무엇을 실행하도록 허락할 것인가
- 실행 범위를 어디까지 가둘 것인가
- 자격증명과 입력을 어디에서 거를 것인가

작동 시점 실행 이전 또는 실행 도중
8개 계층이 다루는 문제는 권한과 범위

! 8개 계층 밖에 있는 물음

- 실행이 끝난 뒤 그 결과가 맞았는가
- 보고서를 잘못 요약하거나 숫자를 잘못 집계한 산출물 — 명령 자체는 안전해 게이트 통과
- 절차가 잘못 연결된 채 스킬 문서에 굳은 경우 에이전트는 멈추지 않고 그럴듯한 결과를 계속 산출

작동 시점 실행 이후 — 해당 계층 없음
명령이 규칙을 지켰는지와 결과가 옳은지는 서로 다른 질문

제품이 맡는 몫

누가 시작할 수 있고 어디까지 손댈 수 있는가

공식 문서도 이 체계를 보안 모델이라고 부를 뿐 결과 품질을 보증하는 장치로 기술하지 않음 — 설계 범위의 경계이지 제품 결함으로 읽을 이유 없음

조직이 채워야 할 몫

실행이 끝난 뒤 결과가 맞는지 확인하는 절차 전부

SECTION 04

Hermes Agent 지원 범위와 결정 사항

설치 전제 · 제품 몫과 조직 몫 · 도입 전 3가지 조건 · 공개 사례의 한계

8가지 항목 점검표

도입 전 3가지 조건

공개 사례의 한계

KEY INSIGHT

직접 구성 열에 값이 적힌 4줄이 이번 검토의 작업 목록 — 실행 범위 조정 · 백엔드 선택 · 승인 담당자 지정 · 검증 절차 전부

항목	기본 제공 (제품 몫)	직접 구성 필요 (도입 몫)	우리 조직 확인
① 절차 기억	제공	—	☐
② 채널 통합	제공	—	☐
③ 무인 실행	제공	실행 범위 조정	☐
④ 실행 격리	제공	—	☐
⑤ 데이터 처리 위치 공개	선택지 공개	검색·추출 백엔드 선택	☐
⑥ 권한 분리	제공	—	☐
⑦ 위험 명령 승인	절차 제공	승인 담당자 지정	☐
⑧ 실행 결과 검증	—	검증 절차 전부	☐

표를 채우기 전 선결 조건 — Python 3.11 · Node.js · ripgrep 을 갖춘 시험 서버를 먼저 확보하고, 맡기려는 반복 업무 한들을 시범 운영

KEY INSIGHT

3가지 조건 모두 기능 유무가 아닌 조직의 값 지정 — 승인 담당자를 정하지 않으면 승인 계층은 알림만 쌓이는 빈 절차

01

무인 실행 범위를 우리 기준으로 좁힐 수 있는가

- 우리 조직의 반복 업무 가운데 사람 확인 없이 맡겨도 되는 범위가 어디까지인가
- 그 범위를 설정으로 실제로 좁힐 수 있는지 시험 환경에서 확인했는가

판단 기준

조회·요약 업무는 넓게
외부 발송·설정 변경이 섞이면
좁히고 승인 단계를 촘촘히

02

위험한 명령의 승인을 누가 어떤 절차로 하는가

- 승인 알림이 누구에게 가는가
- 그 사람이 부재중일 때 대신 판단할 사람을 정해 두었는가
- 판단 기준을 문서로 남겨 두었는가

갈라 두어야 할 2가지 사실

승인 절차가 시스템에 있다는 것
말을 사람을 지정해 두었다는 것
앞은 제품, 뒤는 도입하는 쪽의 몫

03

검색·추출 백엔드와 사용 정보 수집 범위를 확인했는가

- 우리가 고른 백엔드가 어느 서버로 요청을 보내는지 우리 환경에서 확인했는가
- 토큰·비용·세션 계측 조각이 우리 데이터 처리 정책과 충돌하지 않는지 검토했는가

기록 방식

개발 정책 한 줄과 배포판 현황
한 줄로 나누어 기재
하나로 뭉치면 근거 제시 불가

KEY INSIGHT

전사 규모 공개 사례는 미확인 — 우리 조직이 그 규모의 첫 사례가 되는 부담을 감당할지 도입 전 결정

✓ 공개 문서로 확인되는 것

- 저장소 이슈 트래커의 통합 계측 요청 등 개인 개발자와 소규모 팀 단위 논의
- README의 지원 범위 선언 — 5달러 VPS · GPU 클러스터 · 서버리스 인프라

! 공개 자료로 확인되지 않는 것

- 조직 이름이 붙은 전사 규모 도입 사례
- 몇 명 규모 팀에 몇 개월째 운영 중이라는 형태의 공개 기록

공개 사례가 적다는 사실이 뜻하는 것

제품이 나온 지 얼마 되지 않았거나, 도입한 조직이 사례를 밖으로 알리지 않는다는 것 — 어느 쪽이든 근거로 쓸 자료가 적다는 사실은 잔존

종류가 다른 2가지 진술을 따로 세어 두기

지원 범위 선언

제품이 기술적으로 감당하도록 설계한 폭
공식 문서에서 그대로 확인 가능

고객 운영 사례

그 폭 안에서 실제로 오래 버텼다는 증거
공개한 조직이 있을 때만 확인 가능

2개 문장을 접속사로 이어 하나의 결론으로 읽으면 도입 판단의 근거를 실제보다 크게 산정

🔍 마지막 조건 — 8번째 실행 결과 검증

- 1 에이전트가 만든 산출물을 누가 확인하는가
- 2 어떤 기준으로 통과와 반려를 가르는가
- 3 반려한 결과를 어떤 절차로 되돌리는가

3가지 물음에 조직의 이름과 절차로 답할 수 있는지가 도입 여부를 가르는 마지막 조건

THANK YOU

감사합니다

8가지 항목 점검표에 우리 조직 이름을 채우는 일이 첫 단계입니다

검토 기준 8가지 항목

제품 기본 제공 일곱

조직이 채울 몫 하나

MSAP.ai

msap.ai

WEB <https://www.msap.ai/>

TEL 02-6953-5427

MAIL hello@msap.ai