

# Normora Agent Expenditure Governance

AI 에이전트를 위한 사전 실행자원 거버넌스

PUBLIC CONCEPT BRIEF / v1.1 / AUGUST 2026

**자원을 관리하기 전에, 실행을 먼저 통제한다.**

AI 에이전트가 왜 실행되어야 하는지, 어디까지 진행할 수 있는지, 어떤 자원을 사용할 수 있는지, 언제 중단해야 하는지를 정의하는 공개 프레임워크.

|    |           |
|----|-----------|
|    | 서문        |
| 1  | 시스템 정의    |
| 2  | 왜 지금 필요한가 |
| 3  | 다음 통제 경계  |
| 4  | 전략적 차이    |
| 5  | 거버넌스 아키텍처 |
| 6  | 통제 영역     |
| 7  | 권한 수준     |
| 8  | 게이트 판정    |
| 9  | 계약과 실행 기록 |
| 10 | 통제 실행 사례  |
| 11 | 통제 모델     |
| 12 | 시스템 관계    |
| 13 | 공개와 보호    |
| 14 | 통제 검증     |
|    | 마무리 원칙    |

# 서문

## 모든 비용은 실행 결정에서 시작된다.

AI 에이전트는 답변만 생성하지 않는다. 의도를 해석하고, 컨텍스트를 확인하며, 도구를 호출하고, 시스템을 수정하고, 결과를 검증하고, 실패를 재시도하며, 오류에서 복구한다. 각각의 단계는 개별적으로 합리적일 수 있지만 전체 실행은 사용자의 원래 목적을 넘어 확장될 수 있다.

노르모라는 통제 지점을 실행 이전으로 이동시킨다. 실행이 시작되기 전, 그리고 실행이 확장되기 전에 다시 판단한다.

# 실행자원 소비는 거버넌스의 판단이다.

Normora Agent Expenditure Governance는 AI 에이전트가 실행되기 전에 목적, 범위, 권한, 반복, 증거, 중단 조건을 구조화하는 사전 거버넌스 프레임워크이자 제어 아키텍처다. 정확한 토큰 수를 예측한다고 주장하지 않는다. 자원 소비를 만들어내는 실행 결정의 연쇄를 통제한다.

# 에이전트 실행에는 새로운 통제 구조가 필요하다.

에이전트가 응답 생성에서 도구 사용과 시스템 변경으로 이동하면서, 자원 소비는 더 이상 단일 모델 호출에서 끝나지 않는다. 실행 경로 전체에서 누적된다.

|                       |                                  |
|-----------------------|----------------------------------|
| <b>예측하기 어려운 비용 확장</b> | 분석, 도구 사용, 검증, 복구 과정에서 소비가 누적된다. |
| <b>재시도 루프</b>         | 새로운 근거 없이 동일한 가설이 반복 실행된다.       |
| <b>도구 호출 확장</b>       | 의도한 결과보다 많은 도구가 호출된다.            |
| <b>컨텍스트 확장</b>        | 더 많은 파일, 기록, 외부 정보가 작업 안으로 들어온다. |
| <b>에이전트 드리프트</b>      | 실행이 초기 목적과 승인 범위를 점차 벗어난다.       |

**이것은 효율성의 실패만이 아니다. 실행 거버넌스의 실패다.**

# 다음 통제 경계는 실행이다.

**Firewalls**                      네트워크의 경계를 만들었다.

**IAM**                              정체성과 접근 권한을 할당했다.

**Zero Trust**                      모든 접근을 조건부로 만들었다.

AI 에이전트는 새로운 통제 문제를 만든다. 시스템은 초기 요청 이후에도 자원을 소비하고, 도구를 호출하며, 상태를 변경하고, 계속 실행할 수 있다.

## EXECUTION GOVERNANCE

에이전트가 실행해도 되는지, 어디까지 진행할 수 있는지, 실행이 계속되기 위해 무엇이 유지되어야 하는지를 결정하는 통제 아키텍처.

노르모라는 실행의 경계를 구조화한다.

# 또 하나의 토큰 최적화 계층이 아니다.

최적화는 승인된 실행의 경제성을 개선한다. 노르모라는 승인 자체를 통제한다.

Token Limits

Execution Boundaries

Cost Tracking

Expenditure Governance

Tool Permissions

Authority Assignment

Retry Limits

Evidence-Driven Retry

Context Compression

Context Admission

Usage Logs

Contract-to-Receipt Traceability

Post-Use Reporting

Pre-Execution Decision

Technical Possibility

Operational Justification

# 요청에서 통제된 실행까지.

- 1      작업 의도**  
의도한 결과와 운영 가치를 정의한다.
- 2      범위 경계**  
검사, 변경, 보호할 수 있는 범위를 정의한다.
- 3      권한 할당**  
자율성과 승인 요구사항을 설정한다.
- 4      지출 게이트**  
실행이 정당한지 판정한다.
- 5      통제된 실행**  
승인된 행동과 반복만 허용한다.
- 6      증거 기록**  
무엇이 발생했고 완료를 무엇이 증명하는지 기록한다.

각 전환은 하나의 판단 경계다.

# 여섯 개 영역이 실행 경계를 통제한다.

- 1 결과 정당성**  
작업이 왜 존재해야 하며 무엇이 완료를 정의하는가.
- 2 범위 경계**  
어떤 시스템, 파일, 도구, 데이터가 승인되는가.
- 3 권한**  
에이전트가 무엇을 검사, 제안, 수정, 검증, 배포할 수 있는가.
- 4 반복**  
분석, 수정, 검증, 재시도를 몇 회 허용할 것인가.
- 5 증거**  
계속 실행, 재시도, 완료 전에 무엇이 달라져야 하는가.
- 6 중단과 재진입**  
언제 멈추고 어떤 승인이 재개를 허용하는가.

No New Evidence, No New Expenditure.

# 자율성은 가정하는 것이 아니라 할당하는 것이다.

|    |         |                    |
|----|---------|--------------------|
| A0 | 관찰      | 검사하고 보고만 한다.       |
| A1 | 제안      | 상태를 변경하지 않고 권고한다.  |
| A2 | 제한 실행   | 고정된 경계 안에서 수정한다.   |
| A3 | 실행과 검증  | 범위 안에서 실행하고 검증한다.  |
| A4 | 조건부 자율성 | 명시된 정책 조건 아래 진행한다. |

올바른 수준은 의도한 결과를 달성하는 데 필요한 최소 권한이다. 더 많은 자율성이 더 높은 성숙도를 의미하지 않는다.

# 실행은 조건부 상태다.

## READY

통제된 실행을 위한 조건이 충분히 정의되어 있다.

## CONDITIONAL

명시된 제한, 증거, 승인 아래에서만 진행한다.

## HOLD

중요한 정보, 권한, 통제 조건이 해결되지 않았다.

## NOT JUSTIFIED

예상 운영 가치가 자원 소비를 정당화하지 못한다.

게이트 판정은 기술적 성공을 보장하지 않는다. 책임 있는 실행 조건이 충분히 구조화되었는지를 결정한다.

## 통제에는 실행 전 상태와 실행 후 상태가 필요하다.

### Agent Expenditure Contract

승인된 목적, 보호 상태, 범위, 권한, 반복, 증거, 승인, 중단 조건을 담는 실행 전 기록.

### Agent Expenditure Receipt

실제 행동, 이탈, 재시도, 증거, 소비 신호, 완료 상태를 담는 실행 후 기록.

두 기록은 에이전트 실행을 불투명한 사용 이벤트가 아니라 검토 가능한 운영 판단으로 만든다.

# 통제된 실행은 이렇게 구조화된다.

이 PDF를 검토하고, 중요한 위험을 식별해 경영진 요약본을 작성한다.

확인된 위험, 미해결 쟁점, 필요한 의사결정을 담은 간결한 요약본.

업로드된 PDF만 사용. 외부 조사 금지. 원본 파일 수정 금지. 문서 밖의 가정 금지.

A1 - 분석하고 제안만 한다.

초기 분석 1회, 검증 1회. 중요한 누락이 확인될 때만 재시도한다.

모든 위험 진술은 출처 섹션 또는 페이지를 참조해야 한다.

READY

문서가 불완전하거나 손상되었거나 외부 증거가 필요하면 중단한다.

Analyze → Verify → Produce Summary

# 모델에 종속되지 않는 통제 계층.

이 프레임워크는 수동 프로토콜에서 시작해 기계적으로 강제되는 통제로 발전할 수 있다.

## CURRENT

### 수동 프로토콜

통제 프롬프트, 계약, 승인, 실행 기록.

## NEXT

### 정책 래퍼

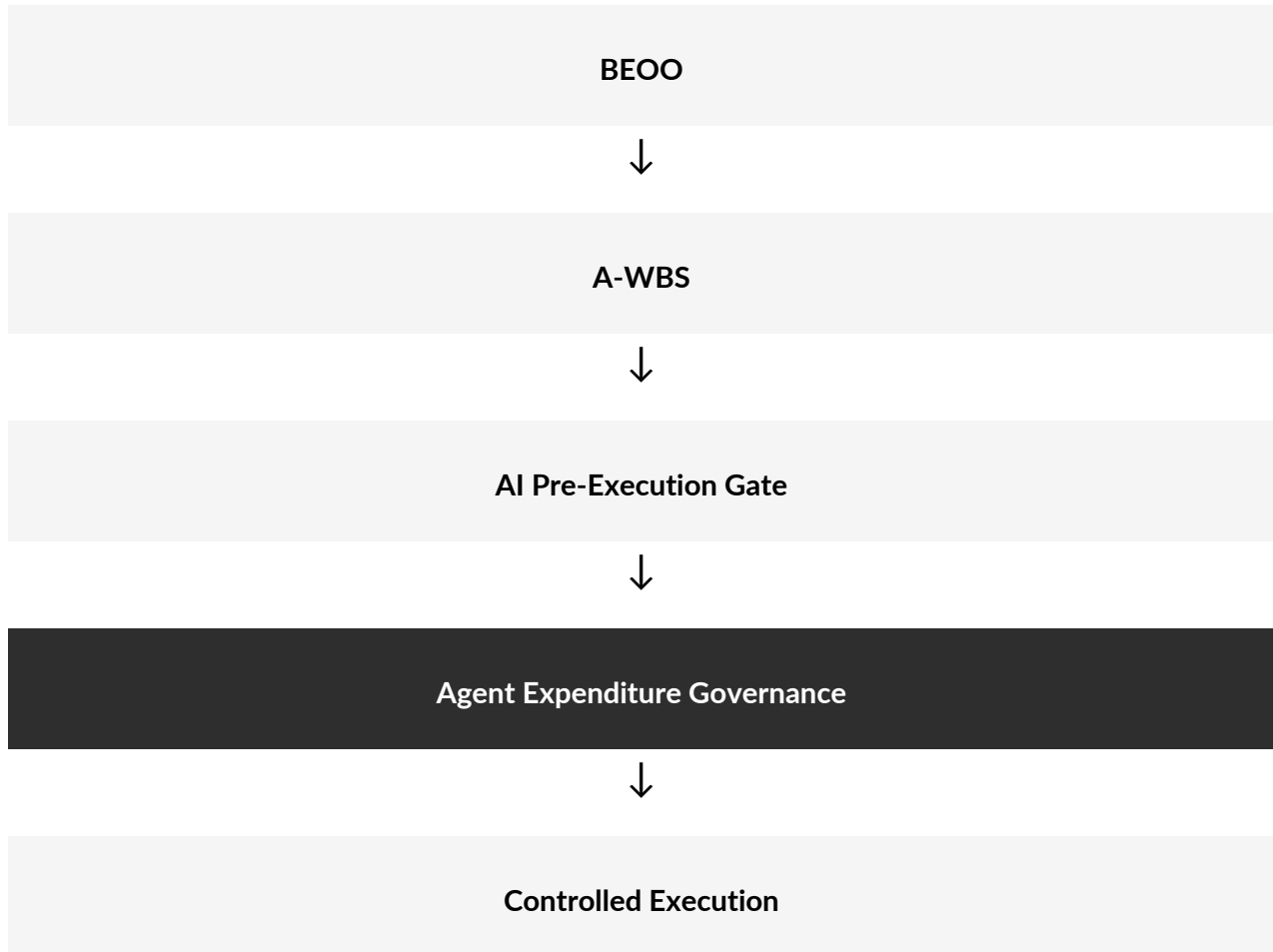
계약 컴파일, 범위 검사, 재시도 통제, 승인.

## FUTURE

### 통제 계층

에이전트 도구와 운영 환경 전반의 런타임 강제.

## 노르모라 운영 아키텍처와 연결된다.



BEOO는 사업과 운영 목적을 정의한다. A-WBS는 범위와 작업을 구조화한다. AI Pre-Execution Gate는 실행 준비도를 평가한다. Agent Expenditure Governance는 에이전트 자원을 사용할 수 있는 조건을 통제한다.

DHEX는 실행 이전의 판단 상태를 구조화하는 노르모라의 보호된 판단 아키텍처다. 내부 인코딩과 구현 로직은 이 공개 문서에 포함하지 않는다.

# 개념은 공개한다. 판단 로직은 보호한다.

## PUBLIC

목적과 카테고리 정의

여섯 개 통제 영역

권한 수준 모델

게이트 판정 언어

Contract와 Receipt 개념

구현 방향

## PROTECTED

점수와 가중치

임계값과 핵심 조건

Evidence Sufficiency 로직

Contract Compiler 규칙

강제 적용 매핑

자동 판정 로직

공개 문서는 시스템을 재현하는 데 필요한 운영 로직을 공개하지 않으면서 카테고리화 아키텍처를 제시한다.

## 성과를 주장하기 전에 프레임워크를 검증한다.

첫 목표는 토큰 감소율을 증명하는 것이 아니다. 비교 가능한 작업에서 사전 통제가 불필요한 탐색, 범위 확장, 동일 가설 반복, 재작업, 복구 사이클을 줄이는지 확인하는 것이다.

|          |            |         |
|----------|------------|---------|
| 사용자 요청 수 | 에이전트 실행 수  | 접근 파일 수 |
| 범위 이탈    | 재시도        | 원복      |
| 완료 시간    | 확인 가능한 크레딧 | 결과 만족도  |

반복된 통제 실험이 방어 가능한 근거를 만들기 전까지 토큰 절감 효과를 주장해서는 안 된다.

# 비용을 관리하기 전에, 결정을 먼저 통제한다.

**모든 토큰은 실행 결정의 결과다.**

Normora Agent Expenditure Governance는 에이전트가 얼마나 저렴하게 실행할 수 있는지부터 묻지 않는다. 그 실행이 정당한지, 어디까지 진행할 수 있는지, 계속 실행되기 위해 무엇이 충족되어야 하는지부터 묻는다.