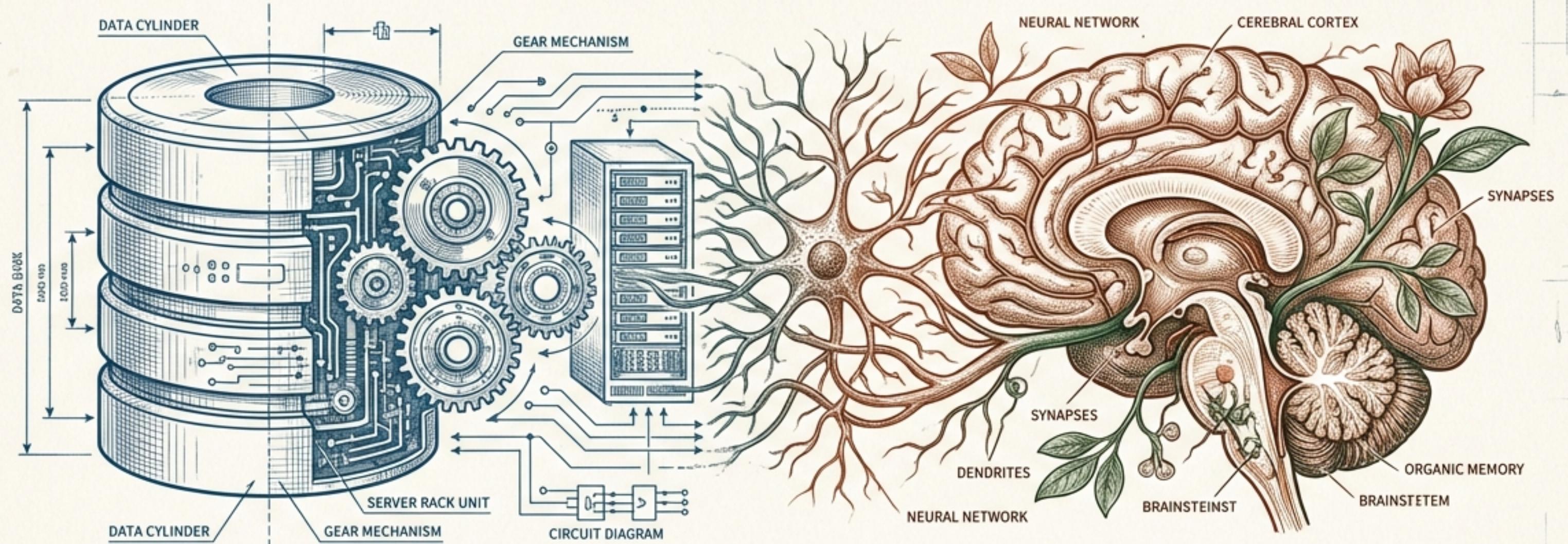


AI 에이전트 시대의 SQL 모먼트



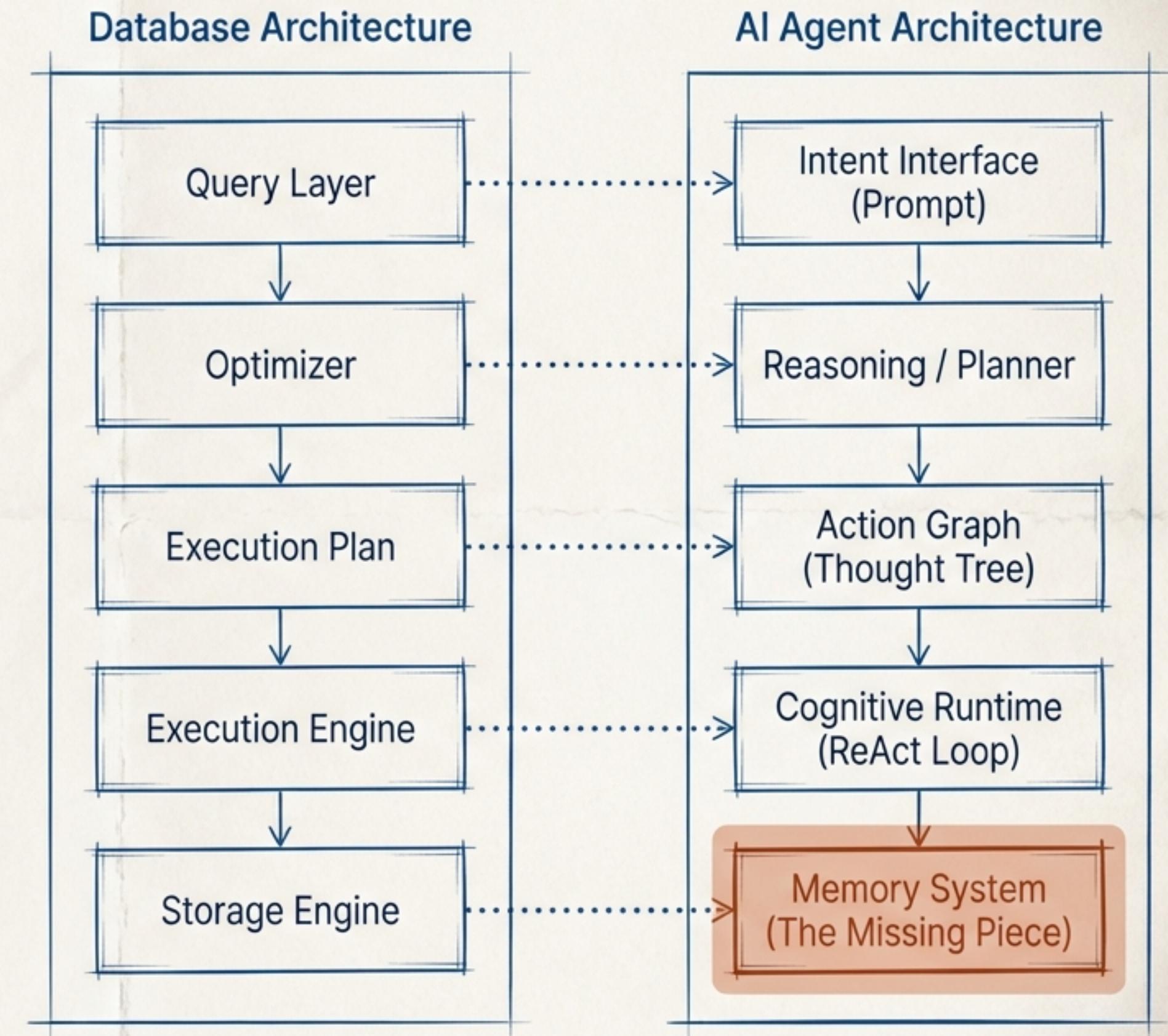
DB 아키텍처의 진화에서 찾은 코딩 에이전트의 '잃어버린 기억(Missing Memory)' 설계 전략

The Gardener's Strategy

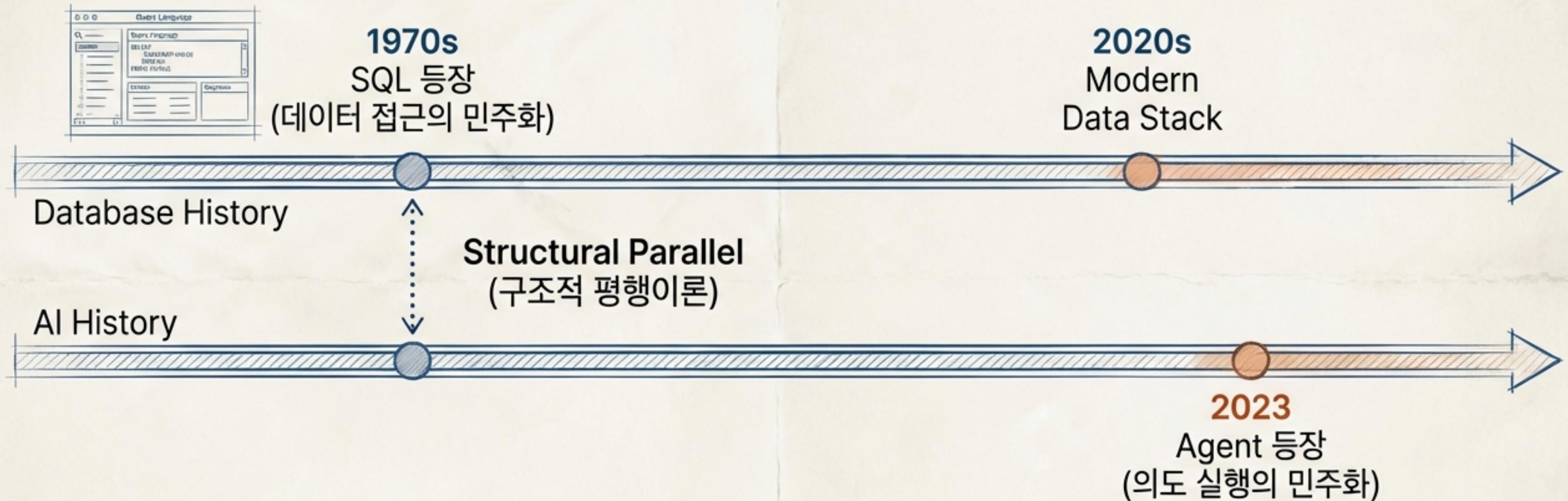
구조적 동형성:
DB는 데이터 실행 엔진이고,
Agent는 의도 실행 엔진이다.

데이터베이스가 쿼리를 처리하는 방식과
AI 에이전트가 프롬프트를 처리하는 방식은
거울처럼 닮아 있습니다.

실행 계획(Execution Plan) 자체가 곧
사용자가 원했던
'애플리케이션(Application)'이 되는
시대입니다.



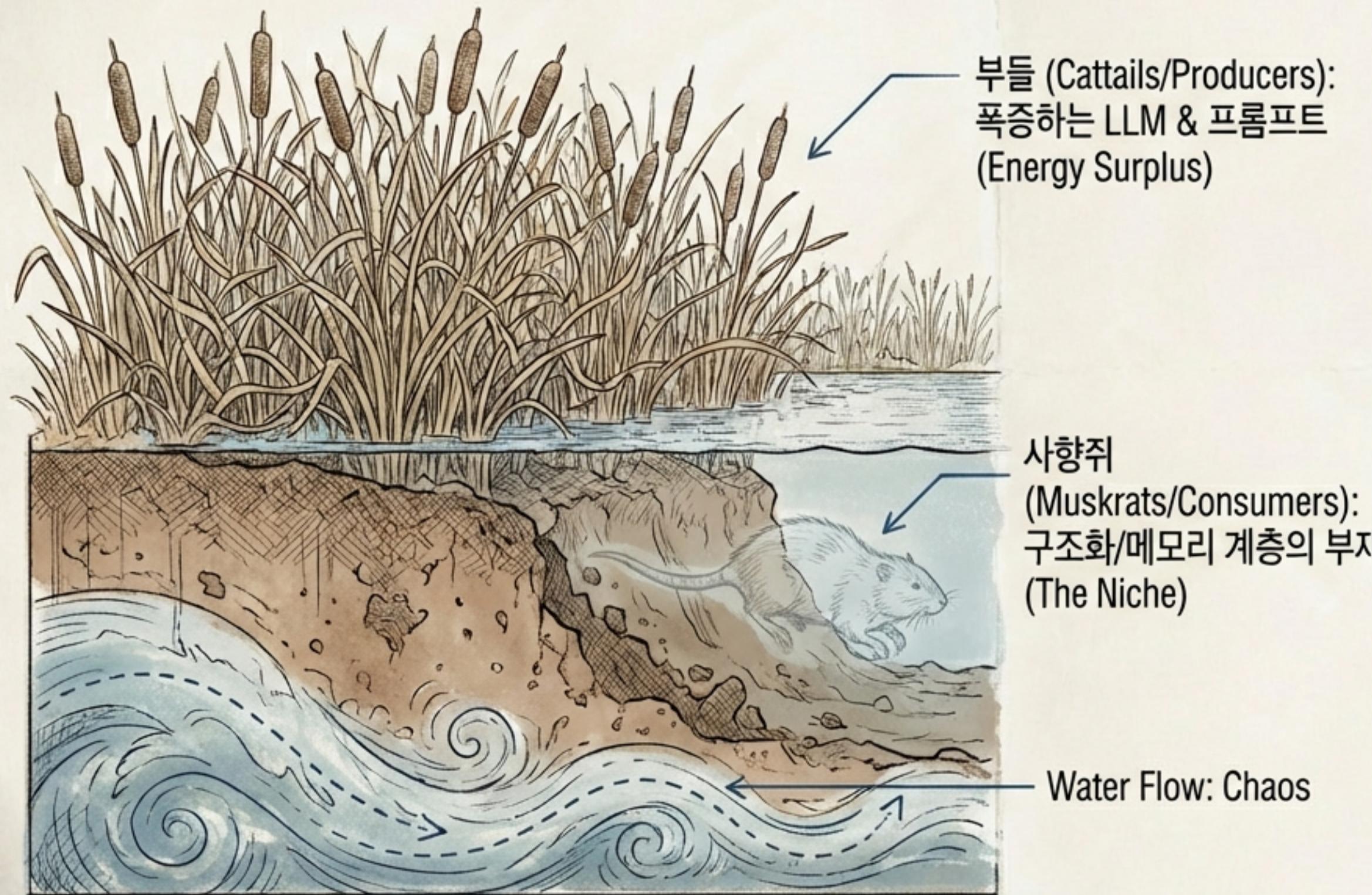
시대적 좌표: 우리는 지금 어디에 있는가?



Current Status: 생성(Generation)은 쉬워졌으나, 일관성(Consistency)과 인덱스(Index)가 부재한 상태.

Analogy: 현재 AI 시장은 강력한 엔진(LLM)은 있지만, 이를 정교하게 제어할 '스티어링 휠'과 '내비게이션(Memory/Structure)'이 없는 초기 자동차 시장과 같습니다.

생태계의 불균형: 과잉된 '생성'과 결핍된 '지속'

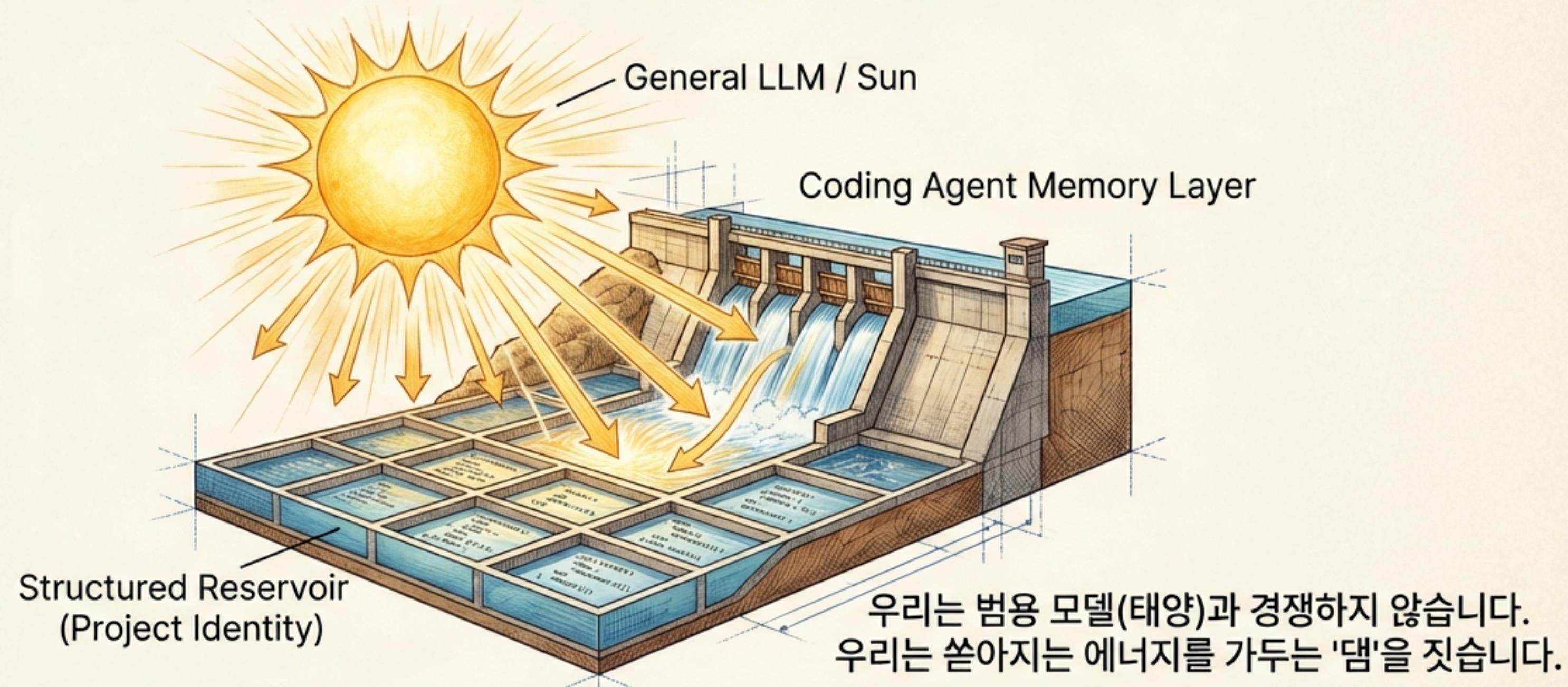


Key Insight: 자연에서는
잉여 에너지가 발생하면
그것을 먹이로 하는 새로운
종(Niche)이 탄생합니다.

사람들은 매번 프롬프트를 다
시 작성하고, AI는 이전 세션
을 기억하지 못합니다.

이것이 바로 '토양
침식(Context Collapse)'이며,
우리의 기회입니다.

니치(Niche) 정의: 코딩 에이전트를 위한 영속적 기억 레이어



NOT

General Chatbot (휘발성 대화)

YES

Coding Agent (프로젝트 단위의 구조적 지속성)

Strategic Goal: 단순 로그 저장이 아닌, 코드베이스의 정체성(Identity)을 보존하는 시스템.

The Pain Point: 맥락의 붕괴 (Context Collapse)

Current State: Session Reset



세션이 끝날 때마다 초기화.
'어제 말한 그 리팩토링 규칙 기억해?'
-> '죄송합니다, 기억하지 못합니다.'

Desired State: Compound Memory



기억이 축적되어 성장하는 구조.
세션을 넘나드는 설계 철학 유지.

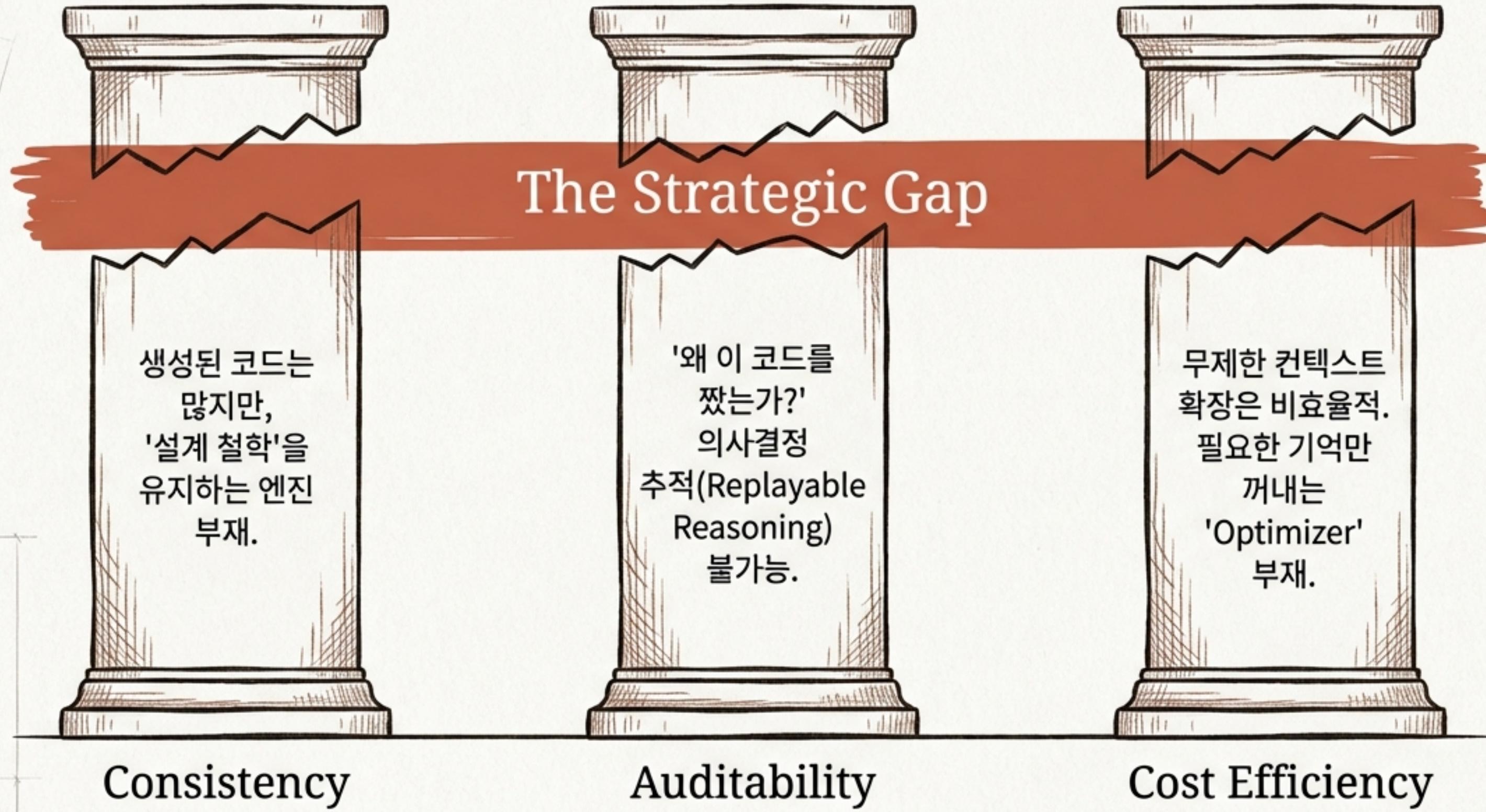
Fact: 최신 모델(GPT-5.3/Opus)도 'Long Context'를 제공할 뿐, '구조적 기억'을 제공하지 않음.

시장 현황 분석: 그들은 무엇을 놓치고 있는가?

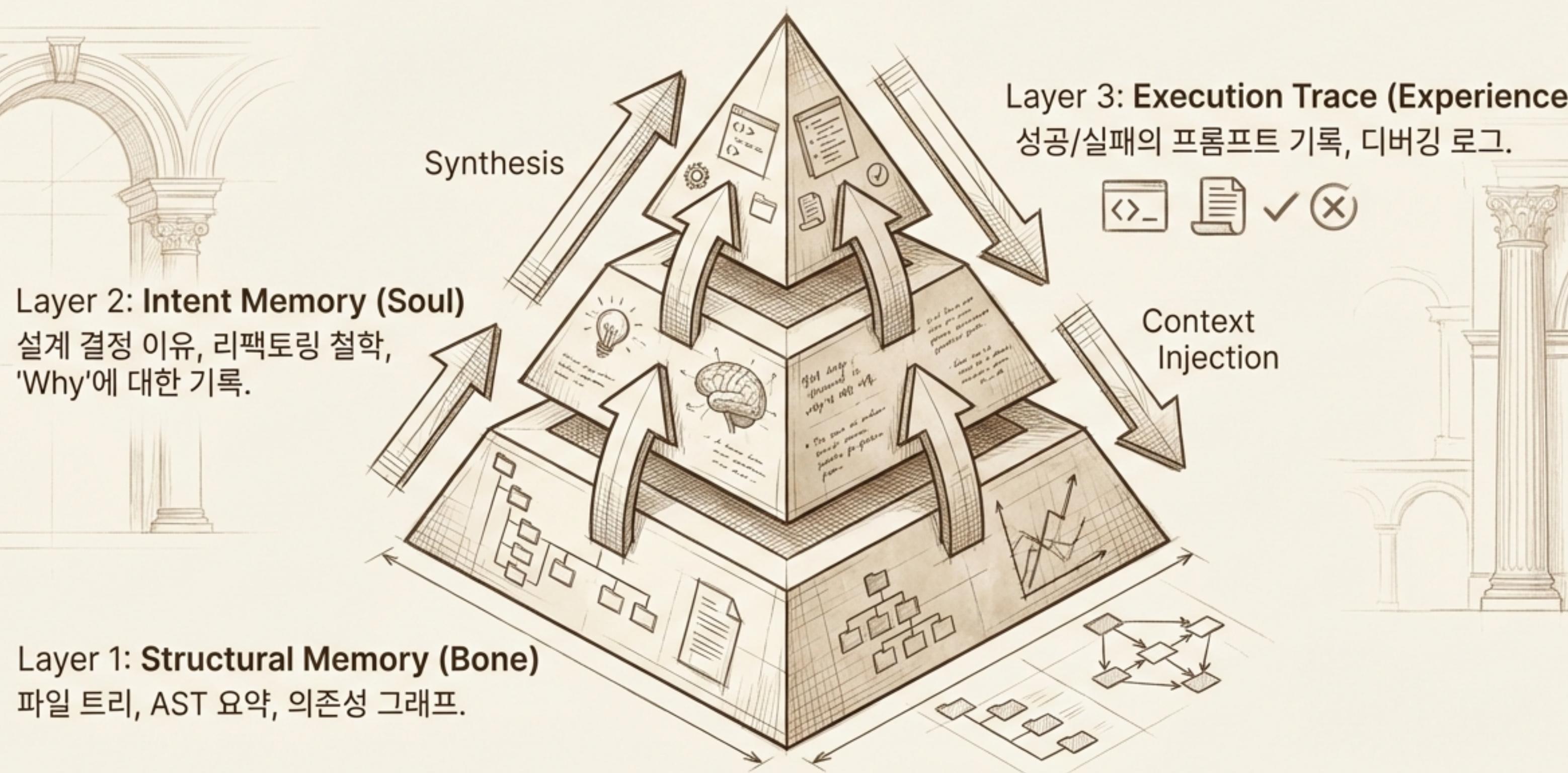
Solution	Key Feature	Critical Limit	Our Analysis
OpenClaw	 로컬 파일(Markdown) 기반	단순 검색/로그 중심, 추론 부족	Log ≠ Memory 
Claude Code	 대용량 컨텍스트(Stuffing)	컨텍스트 희석(Dilution), 비용 증가	Quantity ≠ Quality 
Oh-My-OpenCode	 멀티 에이전트 공유	복잡성 증가, 비용 통제 난해	Complex ≠ Structured 

Synthesis: 기존 솔루션은 '더 많은 텍스트(Quantity)'에 집중하고 있습니다.
우리는 '구조화된 정체성(Quality)'에 집중합니다.

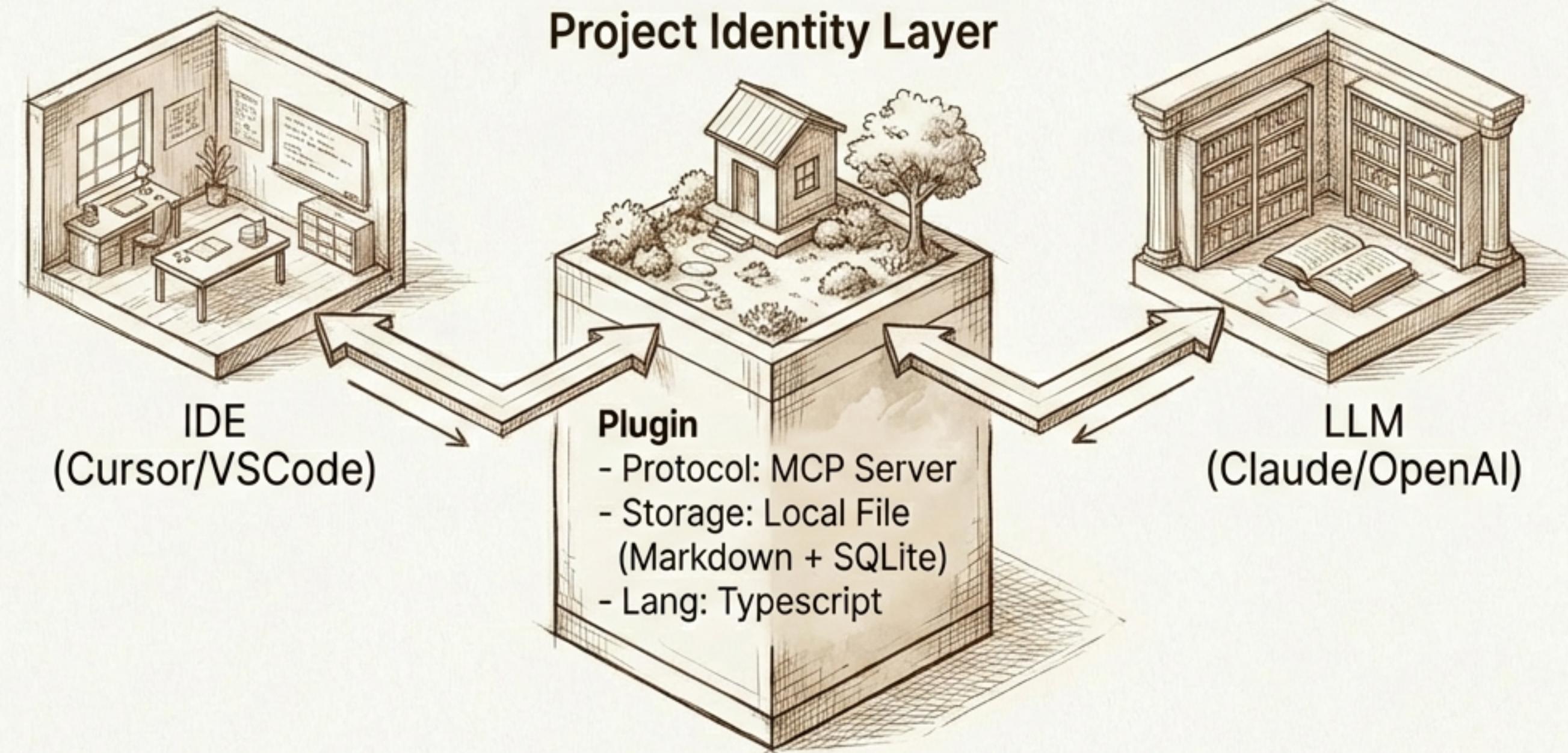
전략적 공백: Cost-based Optimizer & Consistency Engine



제안 아키텍처: 3-Layer Project Memory System

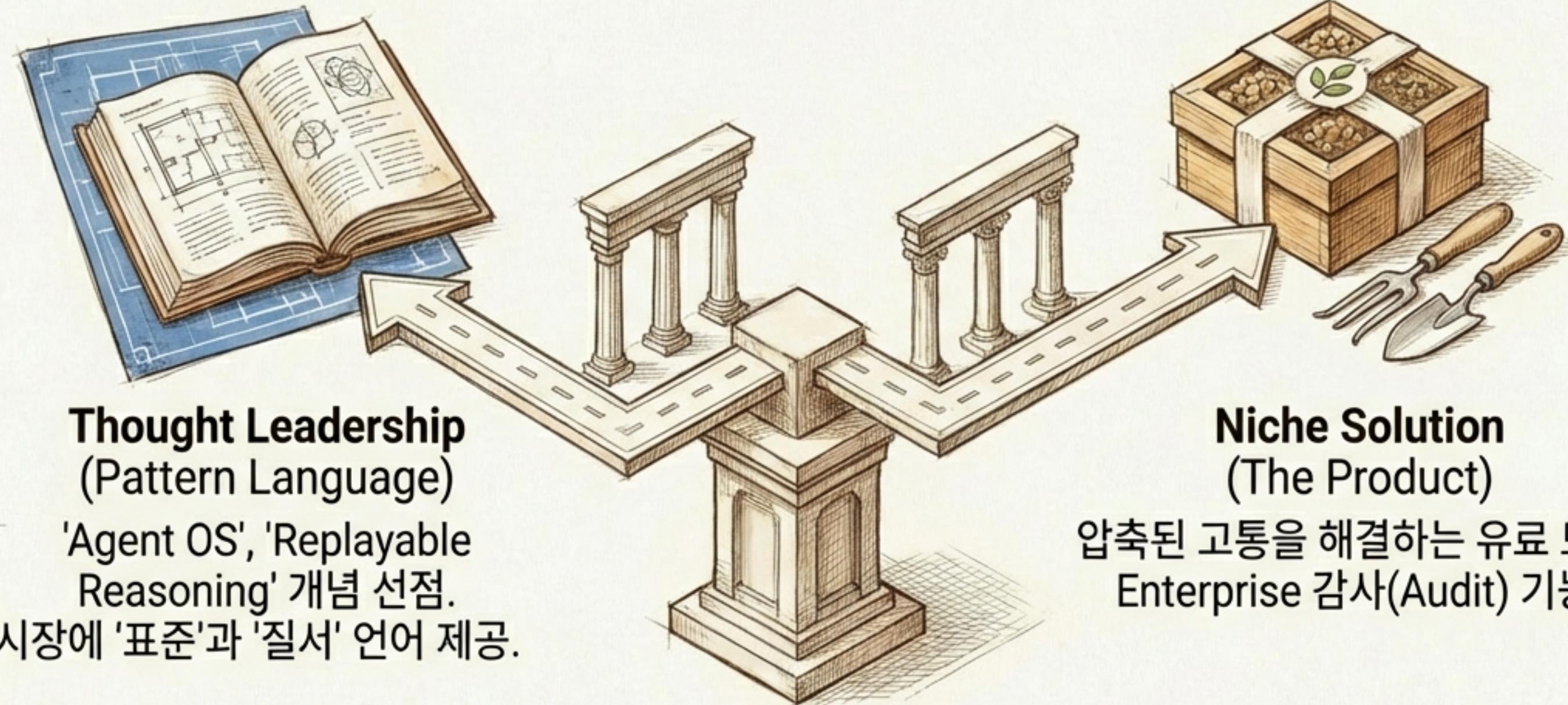


MVP 전략: 작지만 완벽한 정원 (Small Garden Strategy)



Key Feature: 세션 종료 시 자동 요약(Compaction) 및 다음 세션 시 '기억 복원(Restoration)'.

비즈니스 모델: 가드너(Gardener) 전략



Thought Leadership (Pattern Language)

'Agent OS', 'Replayable Reasoning' 개념 선점.

시장에 '표준'과 '질서' 언어 제공.

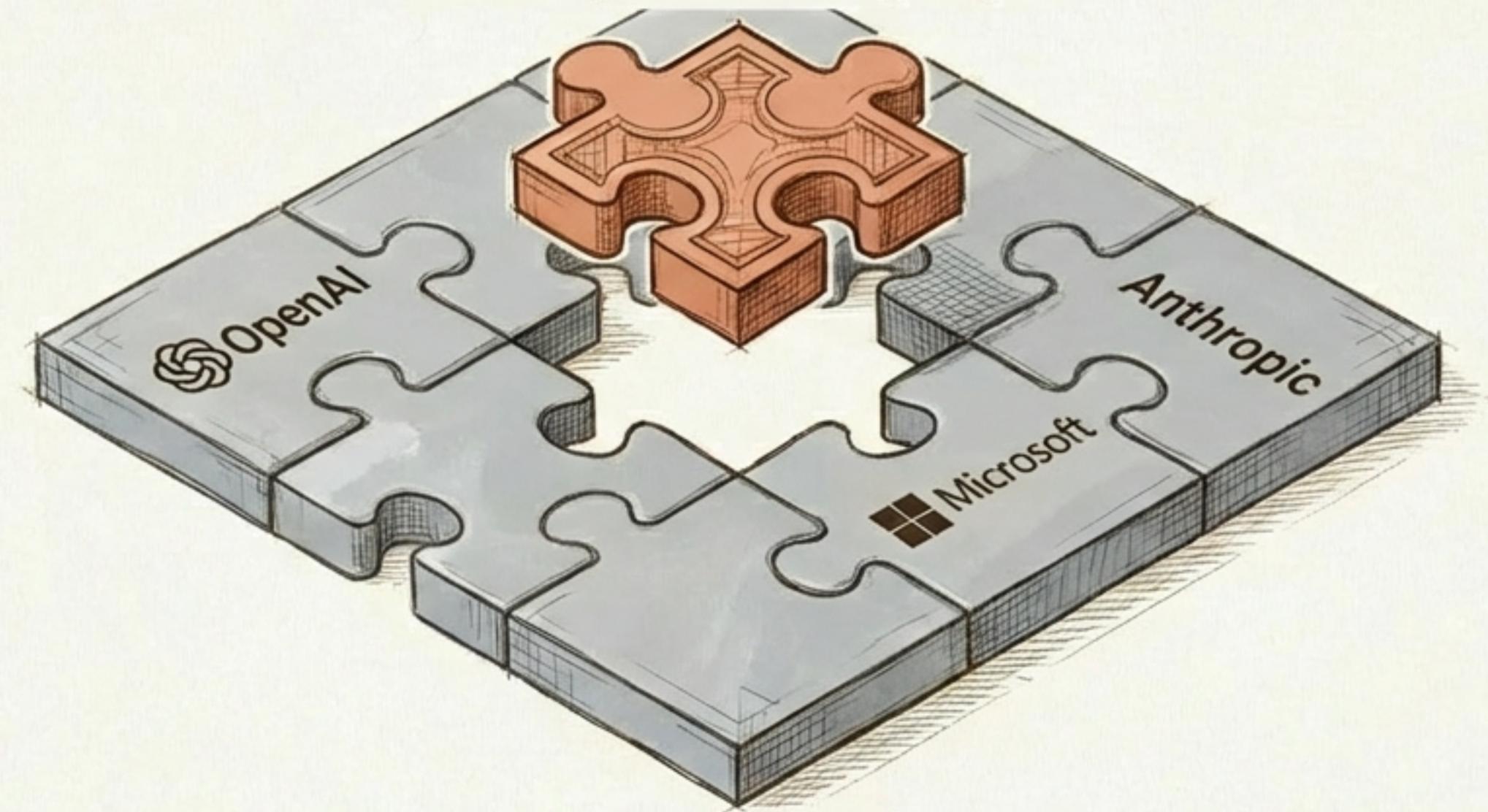
Niche Solution (The Product)

압축된 고통을 해결하는 유료 모듈.
Enterprise 감사(Audit) 기능.

Value: 혼란스러운 생태계에 질서를 제공.

엑시트 전략 (Exit Strategy): From Niche to Standard

Memory UX / Identity Layer



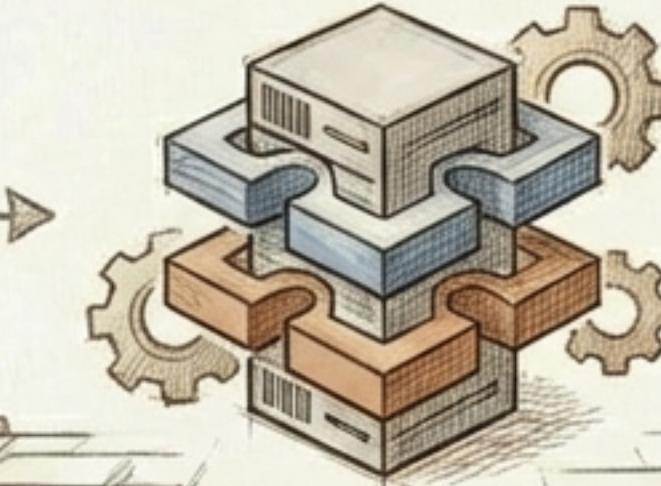
Logic: 대형 플레이어는 인프라 경쟁에 매몰되어 '응용 레이어의 디테일'을 놓치고 있습니다.
우리는 그들이 결국 필요로 할 '플러그 앤 플레이 메모리 모듈'을 완성하여 매각합니다 (Acqui-hire).

실행 로드맵 (90-Day Plan)

Phase 1: Concept & Spec
(Days 1-30)



Phase 2: MVP Development
(Days 31-60)



Phase 3: Validation & Traction
(Days 61-90)



Phase 1: Concept & Spec
(Days 1-30)

Memory Architecture Spec
공개 (GitHub),
Thought Leadership 아티클.

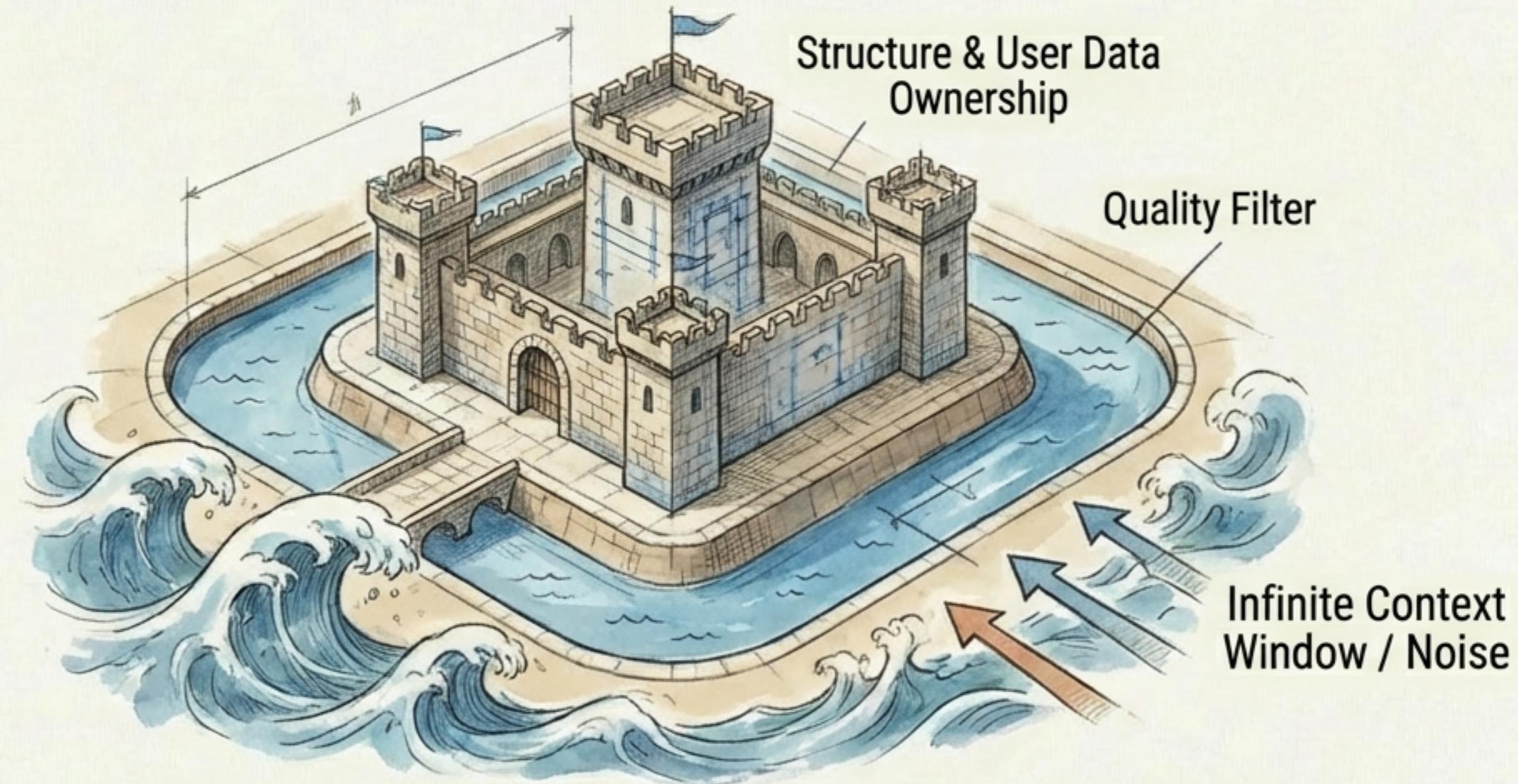
Phase 2: MVP Development
(Days 31-60)

MCP 서버 프로토타입.
Layer 1 (Structural) &
Layer 2 (Intent) 구현.

Phase 3: Validation & Traction
(Days 61-90)

얼리어답터 배포 (Extension).
사용자 피드백 및 '기억 보존'
검증.

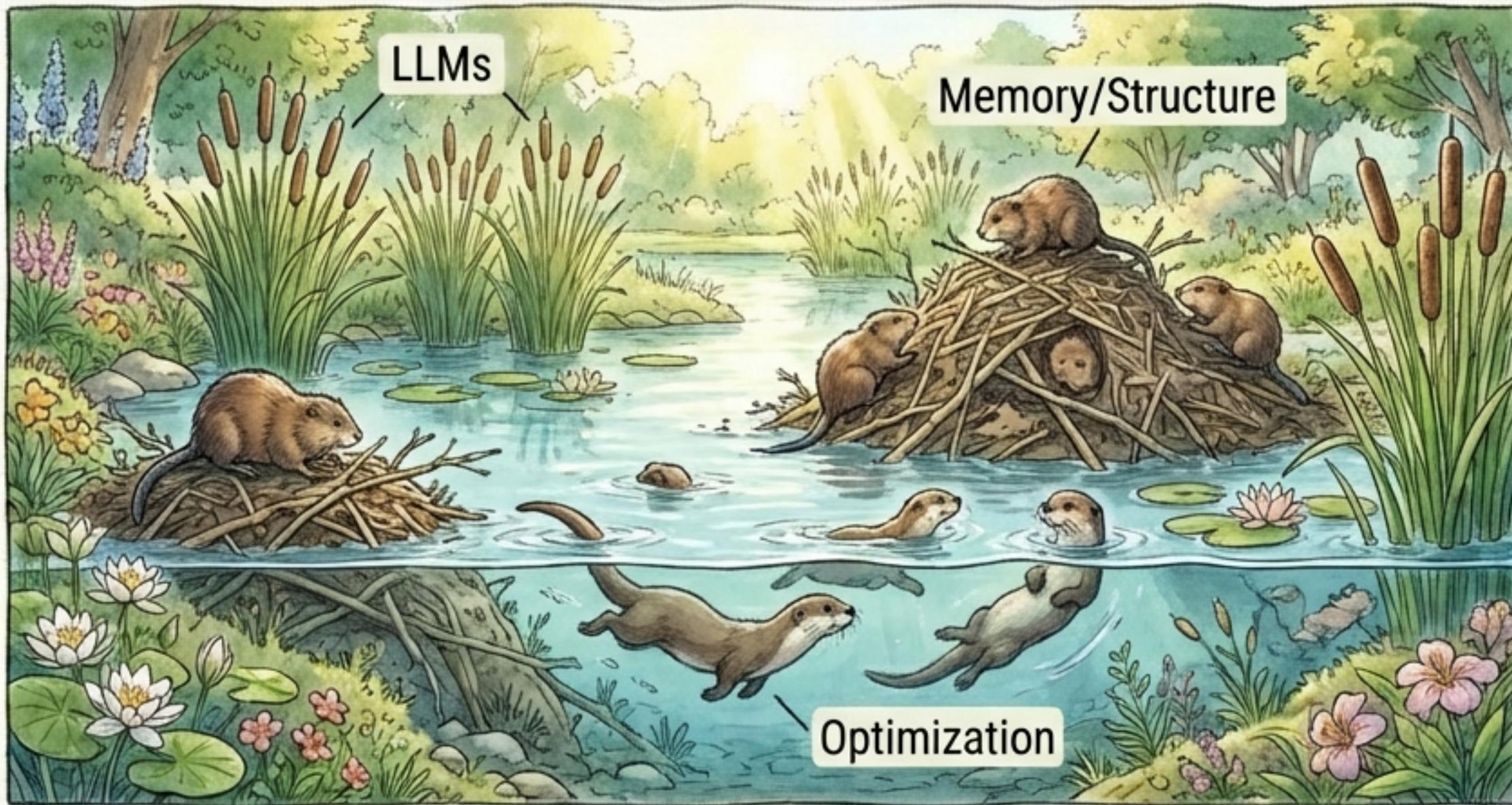
리스크 관리 & 방어 전략 (Moats)



Risk: ‘모델의 Context Window가 무한대가 되면?’

Mitigation: Context가 커질수록 Noise와 비용도 증가합니다. 우리는 ‘양(Quantity)’이 아니라
‘구조(Structure)’를 팝니다. 데이터 주권(Local-first)은 사용자가 소유합니다.

Final Vision: From Chatbot to Co-Founder



우리는 단순히 코드를 짜는 AI가 아니라,
프로젝트의 역사를 함께 써 내려가는 동료를 설계합니다.
흐르는 에너지를 저장하고, 순간의 대화를 영속적인 자산으로 바꾸는 일.
그것이 AI 생태계의 가드너가 할 일입니다.