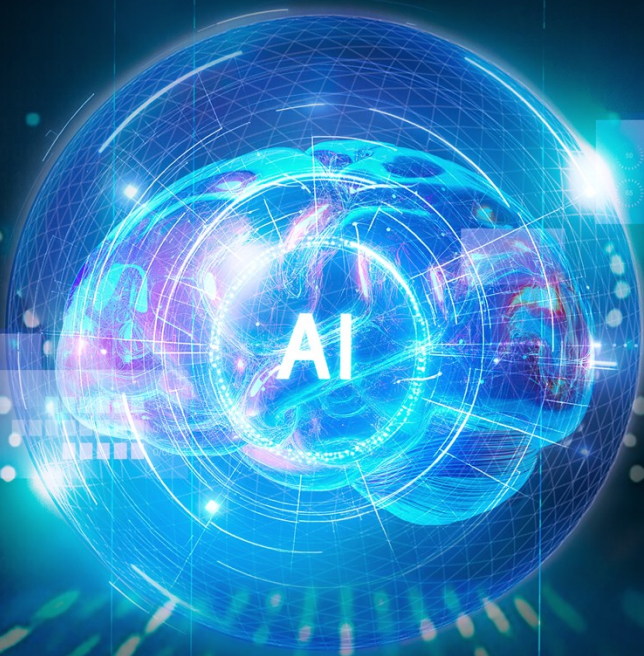


AX 플랫폼 설계/구축

sLLM 확산 시대, 기업의 시스템 개발구축환경은 어떻게 달라지고 있는가

글 | SK AX, CAIO/AI Innovation Lab 조기수 Lab장
SK AX, CAIO/AI Innovation Lab 최철원 전문위원



01. '고성능 모델'에서 '통제 가능한 모델'로

02. sLLM 확산의 배경:
폐쇄망에서의 모델 선택

03. 시스템 구축 환경의 구조적 변화

04. 핵심 과제 (1):
기업 표준의 일관성을 어떻게
지킬 것인가

05. 핵심 과제 (2):
레거시의 암묵적 업무 로직들을
어떻게 빠짐없이 이전할 것인가

06. 기업을 위한 실행 로드맵

07. IT의 구축 및 운영 과정에서
AI가 잘 쓰이는 환경을 만드는 것이 우선

폐쇄망 sLLM 전환

- 국내 금융, 공공, 대기업 대부분은 ChatGPT나 Claude 같은 SOTA 모델을 직접 쓸 수 없는 폐쇄망 환경에 놓여 있다. 이 때문에 시장의 관심은 '얼마나 성능 높은 모델을 쓰는가'에서 '모델을 어떤 환경에서 어떻게 통제하며 쓰는가'로 이동하고 있으며, 그 중심에는 자체 구축한 sLLM이 있다.
- 이는 단순한 대안 선택이 아니라, 코드를 사람이 쓰고 AI가 보조하던 방식에서 AI가 쓰고 사람이 검수하는 방식으로 개발 패러다임 자체가 바뀌는 신호다. 이 리포트는 그 전환 과정에서 실제로 가장 첨예하게 부딪히는 두 가지 과제(기업 표준의 일관성 확보와 레거시 암묵지 이전)를 구체적으로 다룬다.

데이터 오너십과 도메인 특화

- sLLM 도입의 핵심 동인은 비용보다 데이터 오너십 문제다. 에이전트 방식의 개발 워크플로우는 하나의 작업에 수십~수백 회의 추론 호출을 반복하기 때문에, 토큰 과금 구조의 상용 LLM으로는 비용 예측이 불가능해진다. 반면 자체 GPU 기반 sLLM은 초기 투자 이후 한계비용이 낮아 무제한에 가까운 내부 추론을 감당할 수 있다.
- 여기에 도메인 특화 sLLM은 특정 영역에서 범용 대형 모델과 동등한 정확도를 더 낮은 비용으로 구현하며, LoRA 같은 경량 어댑터 기술로 부서별 특화 모델을 동적으로 교체하는 멀티테넌트 서빙까지 가능해졌다. 반복 작업은 내부 sLLM이, 복잡한 추론은 대형 모델이 맡는 계층적 라우팅 구조가 실무의 표준 답안으로 자리잡고 있다.

에이전트 네이티브 개발 환경

- 개발자는 더 이상 코드 한 줄을 요청하지 않는다. 이슈 해결, 테스트 작성, 리팩토링 전체를 에이전트에 위임하는 시대로 넘어가면서, 관심의 초점도 모델 성능에서 컨텍스트 관리, 권한 제어, MCP 연동, 샌드박스, 감사 로그 같은 시스템 엔지니어링 역량으로 이동했다.
- 이 변화는 폐쇄망 환경에서 단품 도구가 아닌 5계층 플랫폼(모델 서빙, 지식, 도구, 오케스트레이션, 거버넌스)의 구축을 요구한다. 특히 코드를 직접 짜지 않고 여러 에이전트를 조율하는 '에이전틱 엔지니어'라는 새로운 직군의 부상, 개발 조직의 역할 재편이 이미 시작됐음을 보여준다.

기업 표준의 기계가독화

- 공개 코드로 학습된 AI 에이전트는 사내 프레임워크의 관례를 모르고 금지된 라이브러리를 자연스럽게 추가한다. 팀마다 다른 스타일의 코드가 생성되며 겉으로는 동작하지만 운영 단계에서 원인 추적이 어려운 코드베이스가 만들어지는 것이 AI 대량 생성 시대의 첫 번째 리스크다.
- 해법은 세 단계로 구조화된다. ① 코딩 표준을 AGENTS.md 같은 기계가독 자산으로 전환하고, ② 골든 레퍼런스와 스캐폴딩을 RAG로 검색 가능하게 하며, ③ 최종적으로는 린터, 정적분석, 아키텍처 적합성 테스트 같은 결정론적 게이트로 확률적 생성의 한계를 보완한다. 이때 게이트의 사용자가 사람에서 에이전트로 바뀐다는 발상의 전환이 핵심이다.

레거시 암묵지 이전

- 가장 가치 있는 비즈니스 로직일수록 문서화가 안 되어 있다는 역설이 차세대 시스템 전환을 어렵게 만드는 근본 원인이다. 담당자가 퇴직하고 문서가 늙아가는 사이, 조직의 기억은 사라진다.
- 이에 대한 실무적 해법은 결정론적 파싱과 LLM 해석의 결합이다. 구문 분석기로 호출 그래프와 테이블 접근 패턴을 정확히 추출하고, 그 위에서 LLM이 비즈니스 의도를 요약한다. 여기에 특성화 테스트와 스트랭글러 피그 패턴을 결합한 병행 검증을 더하면, 신/구 시스템의 출력 차이가 곧 '문서에 없던 요건'의 발견으로 이어지는 안전망이 완성된다.

단계별 실행 로드맵

- 표준의 기계가독화와 지식 자산화라는 기반 없이 코딩 에이전트만 먼저 확산시키면 품질 저하와 일관성 붕괴를 그대로 겪는다는 것이 이 리포트가 강조하는 핵심 경고다. 이 때문에 로드맵은 반드시 순서를 지켜야 한다.
- 1단계(0~6개월)는 폐쇄망 인프라와 표준 문서화, 단일 팀 파일럿에 집중하고, 2단계(6~12개월)는 레거시 리버스 엔지니어링과 RAG, CI 통합으로 지식을 자산화하며, 3단계(12개월~)는 스펙 주도 개발과 모델 교체 가능한 거버넌스로 전사 플랫폼을 정착시킨다. 생산성 지표는 생성량이 아니라 결함률, 재작업률로 측정해야 한다는 점도 실무자가 놓치기 쉬운 포인트다.

격차의 원천: 환경 설계 역량

- 바둑계 AI 도입 사례는 시사하는 바가 크다. 모두에게 같은 훈련 도구가 주어졌지만 상위권과 하위권의 격차는 오히려 벌어졌다.(AI가 주는 답을 이해하고 흡수하는 역량이 저마다 달랐기 때문이다.)
- 기업도 마찬가지다. 오픈 웨이트 모델은 누구나 내려받을 수 있어 기회는 평등해졌지만, 성과는 평등하지 않을 것이다. 진짜 격차는 어떤 모델을 쓰느냐가 아니라, 자사의 표준과 수십 년의 암묵지를 AI가 소비 가능한 자산으로 정비했는지, 그리고 생성, 검증, 피드백의 루프를 조직의 일하는 방식으로 정착시켰는지에서 벌어진다.



01 '고성능 모델'에서 '통제 가능한 모델'로

소프트웨어 개발 과정에 AI를 활용하는 것은 인터넷 공용망 환경에서는 거의 보편적인 환경으로 자리 잡고 있다. 또한 차세대 시스템 구축 프로젝트의 RFP에도 구축 과정에 AI를 활용하는 요건이 포함된 경우도 쉽게 찾을 수 있다. 그런데 국내 금융, 공공, 대기업에서는 우리가 일반적으로 생각하는 ChatGPT나 Claude 같은 SOTA 모델을 직접 활용하는 것은 불가능한 폐쇄망 환경이 대부분이다.

이러한 상황에서 시장의 관심이 이동하고 있다. '얼마나 성능 높은 모델을 쓰는가'가 아니라 '모델을 어떤 환경에서, 어떻게 통제하며 쓰는가'가 새로운 트렌드가 되고 있고, 그 중심에 sLLM(small/specialized LLM)이 있다. 특히 보안 규제가 강한 기업 환경에서는 폐쇄망에 자체 구축한 sLLM이 선택이 아니라 컴플라이언스를 위한 방어적 필수 인프라로 자리 잡고 있다.

이 변화는 기업용 애플리케이션을 '구축하는 방식' 자체를 바꾸고 있는데, 코드를 사람이 작성하고 AI가 보조하던 시대에서, AI가 생성하고 사람이 검수 및 조율하는 시대로 넘어가면서, 개발 표준, 품질 게이트, 지식 관리라는 전통적 SI 역량이 완전히 새로운 형태로 재구성되어야 하는 상황이다.

본 리포트에서는 sLLM 확산이 기업의 시스템 개발 환경을 어떻게 바꾸고 있는지를 조망한 후, 실제 구축 현장에서 가장 첨예하게 부딪히는 두 가지 과제를 다루고 있다. 첫째, AI가 생성하는 코드가 기업의 표준 개발 환경, 코딩 방식, 패턴, 네이밍 룰을 일관성 있게 따르도록 만드는 방법이다. 둘째, 레거시 시스템에 수십 년간 축적된 암묵지를 신규 시스템으로 빠짐없이 이전하는 방법이다. 이 두 과제는 AI 도입의 성패를 가르는 지점이며, 도구의 문제가 아니라 환경 설계의 문제로 봐야 할 것이다.

02 sLLM 확산의 배경: 폐쇄망에서의 모델 선택

2.1 데이터 오너십과 비용 문제

sLLM이 필요한 이유는 비용 문제도 있겠지만 데이터 오너십과 관련된 문제가 더 크다. 데이터 유출을 근본적으로 막으려면 물리적으로 차단된 프라이빗 시스템을 구성하는 것이 기본이다. 클라우드 LLM 또는 상용 LLM 서비스는 토큰 단위 과금 구조를 가지므로, 사용량이 전사로 확산될수록 운영 비용이 기하급수적으로 증가한다. 특히 에이전트 방식의 개발 워크플로우는 하나의 작업을 위해 수십, 수백 회의 추론 호출을 반복하기 때문에, API 과금 모델에서는 비용 예측이 어렵다. 반면 자체 GPU 서버에 구축한 sLLM은 초기 투자 이후 한계비용이 낮아, 무제한에 가까운 내부 추론 작업을 부담 없이 운용할 수 있다. 이것이 폐쇄망 하이브리드 아키텍처의 가장 큰 경쟁 우위가 될 수 있다.

더 많은 내용을 보시려면

파일 다운받기

버튼을 눌러주세요