Ⅱ 주요 동향(2) : ICT

1 에이전트 AI, 단일 서비스를 넘어 '에이전트 플랫폼' 경쟁으로 확산

에이전트 AI의 업무 영역 확산과 상위 실행 플랫폼의 부상

○ '말하는 AI'에서 '일하는 AI'로, 에이전트 AI 도입에 따른 업무 방식 재편

- 생성형 AI가 단순 질의응답을 넘어 복합·다단계 업무를 자율 수행하는 '에이전트' 형태로 진화하면서, 업무 현장에서 AI 활용이 빠르게 확산되고 업무 방식 전반의 재편이 본격화
  · (코딩 영역) Claude Code, Codex 등 AI 코딩 에이전트가 실제 개발 현장에 투입되면서, 엔지니어의 역할이 '코드 작성자'에서 'AI 감독·검증자'로 이동
  · (사무 영역) Anthropic의 Claude Cowork가 영수증 제출·경비 처리 등 다단계 사무 업무를 자율 수행하고, '26년 2월 법률·금융 특화 기능까지 추가되며 전문직 루틴으로 자동화 범위 확대
  · (전문직 영역) Harvey 등 전문직 특화 AI가 계약 검토와 법률 리서치 같은 고난도 업무까지 지원하면서, 법률 분야에서도 AI의 실무 적용 범위가 빠르게 확대되는 단계에 진입
  · (업무 환경) 직원이 매일 사용하는 브라우저와 업무용 앱 자체에 에이전트가 내장되면서, AI가 직접 화면을 조작하는 방식으로 일상 업무 접점도 재편
- 이처럼 에이전트 AI는 코딩·사무·전문직·브라우저 업무 전반으로 확산되며, 개념 검증 단계를 넘어 실제 업무에 투입되는 서비스 단계에 진입

○ 에이전트 경쟁, '개별 품질'과 함께 '플랫폼 운영 역량'까지 확장

- 에이전트 수가 늘어날수록 공통 실행 환경과 연결 규격의 부재로 파편화가 심화되며, 조직 전체로의 확산이 오히려 어려워지는 '난립의 역설'이 부상
- 이에 따라 주요 사업자들이 개별 에이전트를 통합 운영하는 상위 실행 플랫폼(에이전트 플랫폼)을 제시하면서, 시장 경쟁의 초점도 단일 에이전트의 성능과 함께 플랫폼의 운영 역량으로 확대
- 에이전트 플랫폼은 계정 관리·보안 체계와 연동되는 공통 운영 기반으로 자리 잡으며, 조직 단위 확산을 뒷받침하는 핵심 인프라로 부상

에이전트 AI의 작동 원리와 기술 구조

‘답하고 끝나는’ 챗봇에서 ‘과업을 완수하는’ 에이전트로

- 기존 챗봇이 ‘출장 영수증 제출 방법을 알려주는 AI’로서 사용자가 직접 앱을 실행해야 했다면, 에이전트는 ‘출장 영수증을 제출해주는 AI’로서 목표만 주면 여러 앱·도구를 스스로 조작해 결과까지 도달
- 기존 챗봇은 사용자 질문을 받아 LLM이 답변을 생성한 뒤 종료되는 1회성 응답 구조로, 작업 실행 권한이 없고 세션 간 맥락 유지도 제한적이어서 주로 정보 제공 역할에 국한
- 반면 에이전트는 ①사용자의 목표를 해석하고 작업 단계를 분해한 뒤, ②도구를 호출·실행하고, ③결과를 스스로 검증하고, ④맥락을 유지해 작업을 이어 수행하는 루프를 자율적으로 수행
- 결국 에이전트 AI의 차별점은 챗봇보다 더 정교한 답변을 만드는 능력이 아니라, 계획·실행· 검증을 반복하는 구조를 바탕으로 실제 결과를 끝까지 완수할 수 있다는 데 존재

에이전트의 4단계 자율 루프, 성패는 ‘멈출 줄 아는 능력’

- 에이전트는 ‘목표 해석→도구 호출→검증→상태 유지’의 4단계 자율 루프로 작동하며, 각 단계를 반복하면서 과업을 스스로 수행
  - (① 목표 해석) 사용자의 지시를 구조화된 과업으로 바꾸는 단계로, 예를 들어 ‘출장 영수증 제출’ 요청을 ‘사진 금액 추출→항목 분류→정책 확인→시스템 제출’의 4개 하위 단계로 분해
  - (② 도구 호출) 웹 서비스, 앱, 데이터베이스, 파일 등 외부 자원을 활용해 각 하위 단계를 수행하는 단계로, 동일 도구를 반복 호출하거나 여러 도구를 조합해 실제 작업을 처리
  - (③ 결과 검증) 수행 결과를 스스로 점검해 실패나 미완료를 감지하고 경로를 수정하는 단계로, 완료 여부를 판단해 보고하거나 추가 조치를 이어가는 피드백 구조로 작동
  - (④ 상태 유지) 이전 맥락과 중간 결과를 기록으로 보존해 긴 작업을 여러 단위에 걸쳐 이어 수행하는 단계로, 어제 시작한 프로젝트를 오늘 이어받아 진행하는 장기 업무 구조를 실현
- 이러한 루프가 실제 업무에서 작동하려면 에이전트가 언제 계속 진행하고 언제 사용자 확인이 필요한지를 스스로 구분할 수 있어야 하며, 이 판단이 부족할 시 자율성 저하와 오작동이 발생

● 4대 구성요소의 설계가 에이전트 성능을 좌우

- 앞서 살펴본 자율 실행 루프가 작동하려면 단일 모델(LLM)이 아닌 추론, 연결, 실행, 기억의 4가지 기능 요소를 유기적으로 설계할 필요
  · (추론) 사용자 의도 해석과 작업 계획 수립을 담당하는 '두뇌'로, 모델(LLM)이 핵심이나 같은 모델이라도 주어지는 지침이나 역할 정의에 따라 계획 품질이 크게 좌우
  · (연결) 툴·인터페이스는 외부 API, 업무 앱, 데이터베이스 등과 에이전트를 잇는 '손과 발'로, 연결 범위가 좁으면 실행 자체가 불가능하며 연결이 불안정하면 과업 중단 위험 발생
  · (실행) 실행 루프는 계획·실행·검증·수정을 자동으로 되돌리는 '지휘자'로, 중간 오류나 환경 변화가 발생해도 과업을 끝까지 완수하도록 전체 흐름을 조정하는 핵심 메커니즘
  · (기억) 메모리는 세션 내외의 맥락과 작업 이력을 보존하는 '작업 노트'로, 기억이 부실하면 반복 작업마다 맥락이 초기화되고 장기 업무와 개인화가 불가능
- 결국 에이전트의 성능은 모델 자체만으로 결정되지 않으며, 판단·기억·실행·검증을 묶어 실제 과업 완수로 연결하는 내부 구성요소 전체의 설계 완성도가 핵심 경쟁력으로 부상

에이전트 AI가 최근 본격화된 구조적 배경

● 범용 모델의 상품화로 수익 회수 한계 노출, 주요 사업자는 응용 계층으로 이동

- 범용 AI 모델 시장은 소수 선도 기업 중심의 과점 구조와 급속한 성능 평준화가 겹치며, 모델 판매만으로는 수익을 안정적으로 회수하기 어려운 단계에 진입
  · '25년 말 기준 전 세계 엔터프라이즈 AI 시장은 Anthropic(40%), OpenAI(27%), Google (21%) 등이 약 90%를 점유하는 뚜렷한 과점 구조 형성(Menlo Ventures, '25.12)
  · 선도 기업 간 성능 격차도 빠르게 축소되며, 경쟁사가 단기간 내 성능을 따라잡는 현 환경에서 모델 자체만으로는 차별화가 점차 어려워지는 추세(Mary Meeker, '25.05)
  · 반면 AI 인프라에 투입되는 자본은 천문학적 규모로, 약 5조 달러 규모의 글로벌 AI 인프라 투자에서 10% 수익률을 확보하려면 연 6,500억 달러 수준의 AI 매출이 필요(JPMorgan, '26.04)
- 결국 주요 사업자는 모델 공급을 넘어, 더 높은 수익과 고객 접점을 확보할 수 있는 상위 응용 및 플랫폼 계층으로 이동해야 하는 압박에 직면

● 실행 인프라 성숙으로 행동하는 AI로 급속하게 전환

- 에이전트가 실제 업무를 하려면 이전 작업을 기억하고, 안전한 작업 공간에서 움직이며, 외부 시스템과 연결될 수 있어야 하는데, 최근 이를 뒷받침하는 인프라가 빠르게 성숙
  · (작업 기억) 에이전트가 긴 업무를 하다가 중간에 멈춰도, 이전 맥락과 진행 상황을 다시 불러와 이어서 처리할 수 있는 기능이 주요 플랫폼의 기본 요소로 정착
  · (안전한 작업 공간) 코드 실행, 파일 접근, 네트워크 사용이 가능한 격리된 작업 환경이 보편화되면서, 에이전트가 실제 시스템을 다루더라도 더 안전하게 실행할 수 있는 기반 마련
  · (외부 도구 연결) 에이전트가 데이터베이스, 업무용 소프트웨어, 웹 서비스 등 외부 도구와 자연스럽게 연결되는 기능이 확산, 여러 시스템을 오가는 다단계 작업의 실용적 자동화 기반 형성
- 이를 통해 에이전트는 단순히 답변을 생성하는 도구를 넘어, 실제 업무를 수행할 수 있는 실행형 시스템으로 발전

● 연결 표준의 공용화, 에이전트 확산의 기반 형성

- '25년 12월 Anthropic의 연결 표준 MCP가 Linux Foundation 산하 Agentic AI Foundation에 기부되며, 특정 기업의 사유 규격에서 업계 공용 표준으로 전환되는 계기 마련
- 이에 따라 외부 도구 및 다른 에이전트와의 연결 방식이 공용 자산화되면서, 시장 진입 장벽은 낮아지고 생태계 확산의 기반은 한층 넓어지고 있음
- 또한 에이전트 간 협업, 상거래, 결제, 사용자 인터페이스 연동 등 용도별 규격도 함께 등장하면서, 연결 기반이 단일 표준을 넘어 다층 구조로 확장
- 그 결과 기업 고객은 특정 사업자에 종속되지 않고 여러 플랫폼을 병행 활용할 수 있는 선택지 확보

에이전트 플랫폼의 부상 배경과 핵심 기능

● 에이전트 AI 플랫폼의 필요성 급부상

- 에이전트 수가 늘어날수록 실행 환경과 연결 방식, 권한 관리 기준이 서로 달라져 개별 에이전트만으로는 전체 업무 흐름을 안정적으로 조율하기 어려운 한계 발생
- 하나의 에이전트가 특정 과업을 수행하는 것과 여러 에이전트가 함께 작동하는 환경을 안정적으로 유지하는 것은 다른 문제라는 점이 점차 분명해지는 흐름

- 이 때문에 플랫폼은 여러 에이전트와 도구, 데이터, 규칙을 공통 기준 아래 연결·조율·관리하는 조정 계층으로서의 필요성이 확대
- 즉, 최근에는 개별 에이전트의 능력과 함께, 이를 일관된 구조 안에서 안정적으로 작동시키는 설계 역량이 플랫폼의 핵심 가치로 부각

○ 에이전트 AI 플랫폼의 핵심 기능

- (공통 실행 환경) 다양한 에이전트와 서비스가 동일한 기준과 방식으로 작동할 수 있도록 공통 실행 기반을 제공, 이를 통해 개별 서비스 간 호환성을 높이고 개발·운영 환경의 일관성 확보
- (연결 및 오케스트레이션) 여러 에이전트와 외부 도구, 업무 시스템 간 연계 지원 및 협업 흐름 조율, 단일 에이전트의 응답을 넘어 복수의 기능과 자원을 통합해 과업 수행하도록 지원
- (관리 및 통제) 데이터 접근 권한, 보안정책, 운영 규칙 등 플랫폼 전반의 통제 체계 관리, 이를 통해 에이전트의 자율성을 보장하고 정책과 거버넌스 기준안에서 안전하게 운영하도록 지원
- (운영 최적화) 배포, 모니터링, 성능 점검, 장애 대응, 지속적 개선 기능을 포함, 플랫폼 운영 과정에서 실행 데이터를 바탕으로 안정성과 효율성을 높이고 서비스 품질을 지속적으로 고도화

에이전트 플랫폼 생태계 전략: 개방형 vs 자사 서비스 중심의 플랫폼

○ 플랫폼 진화의 두 방향, 생태계 확장형과 자사 서비스 중심의 플랫폼 병존

- 최근 에이전트 AI 플랫폼은 단일한 형태로 수렴하기보다, 외부 연결과 생태계 확장을 중시하는 개방형과 통합된 사용자 경험과 운영 완성도를 중시하는 자사 서비스 중심 플랫폼으로 나뉘는 흐름
- 개방형 플랫폼은 더 많은 개발자, 도구, 서비스, 파트너를 끌어들이는 데 강점이 있고, 자사 서비스 중심의 플랫폼은 하나의 일관된 경험 안에서 빠르게 성능과 안정성을 제공하는 데 강점
- 결국 플랫폼 경쟁의 축도 단순한 모델 성능보다, 누가 더 넓은 생태계를 만들 것인가와 누가 더 매끄러운 사용자 경험을 제공할 것인가의 두 방향으로 분화되는 국면
- 향후 플랫폼 경쟁은 기술 우위 자체보다, 개방성과 통합성 가운데 어떤 전략이 더 많은 사용자와 개발자, 서비스 사업자를 끌어들이느냐에 따라 갈릴 가능성 상존

1. 개방형 플랫폼: OpenClaw와 NemoClaw

- Openclaw, 확장 중심의 개방형 생태계 지향
  - Openclaw는 고가의 전용 하드웨어 없이도 일반 PC나 소규모 가상 서버에 구동 가능하도록 최적화, 막대한 라이선스 비용 없이 최첨단 기술을 내재화할 수 있어 벤더 종속성 탈피 가능
  - WhatsApp, Slack, Discord 등 직원이 이미 쓰는 메신저와 연동되는 구조라, 회사가 별도 도구를 도입하지 않아도 개인 단위에서 바로 쓸 수 있는 낮은 진입 장벽 확보
  - 또한 사용자가 필요한 기능과 연결 대상을 직접 추가할 수 있어, 하나의 완성형 서비스라기보다 여러 기능을 붙여가며 확장하는 개방형 생태계
  - OpenClaw의 핵심 강점인 자기 학습은 에이전트가 자신의 기억·성격·기능 파일을 매번 스스로 고쳐 쓰는 방식으로 작동하나, 바로 이 자기 수정 통로가 공격자에게도 열려 있다는 점이 취약점으로 부상
- NemoClaw, 개방형 구조에 보안과 통제를 더한 방식
  - NemoClaw는 Nvidia가 '26.3 GTC 2026에서 공개한 기업용 에이전트 보안 플랫폼으로, OpenClaw의 개방형 구조 위에 샌드박스 격리와 내부 처리, 정책 적용 기능을 적용한 형태
  - 사용자의 모호한 의도를 명확한 실행 논리로 분해하는 인지 계층, 기업 내부 시스템에 직접 접근하여 실행하는 실행 계층, 모든 행동이 안전 규정에 부합하는지 감시하는 자가 검증 계층으로 구분
  - NemoClaw의 핵심은 에이전트 실행 자체를 격리 공간에 제한하는 보안 런타임 OpenShell과, 민감 데이터를 외부로 이전하지 않고 회사 내부에서 처리하는 오픈 모델 Nemotron의 결합
  - OpenShell은 각 에이전트를 독립된 샌드박스에 격리해 파일, 네트워크, 도구 사용 범위를 사전에 정의된 정책안으로 제한하는 방식으로 작동
  - '프라이버시 라우터' 기능을 통해 민감한 내용은 회사 내부 GPU에서만 처리하고, 일반 질문만 외부 모델로 보내는 방식을 지원해, 클라우드를 활용하면서도 정보 유출을 최소화하는 선택적 운영 가능
- 개방형 플랫폼이 주목받는 이유
  - 개방형 플랫폼은 사용자가 데이터 경로, 연결 방식, 모델 선택을 직접 정할 수 있어 통제권이 크다는 점에서 관심을 끄는 흐름

- 외부 개발자와 기업이 기능을 계속 추가할 수 있어, 플랫폼 혼자 키우기보다 생태계 전체가 함께 성장하는 구조를 만들기 쉬움
- 특히 Openclaw는 연결과 확장에, NemoClaw는 여기에 보안과 보호 기능을 더하는 방향에 강점이 있어 서로 다른 수요를 흡수하는 모습
- 결국, 개방형 플랫폼은 유연하게 넓히는 힘이 강점이고, 그 과정에서 생기는 보안·운영 부담을 어떻게 줄이느냐가 다음 과제로 남는 구조

2. 자사 서비스 중심의 플랫폼: Claude, OpenAI, Google

Claude, 관리형 운영 완성도를 앞세운 플랫폼

- Claude Managed Agents는 미리 구성된 하네스와 관리형 인프라를 함께 제공함으로써, 사용자가 에이전트 루프와 실행 환경을 직접 설계하지 않아도 되는 구조
- 파일 읽기, 명령 실행, 웹 탐색, 코드 실행 같은 기능을 보안이 적용된 관리형 환경 안에서 바로 쓸 수 있어 장시간 작업과 비동기 작업에 강점
- 또한, 빠르게 진화하는 모델 역량에 맞춰 운영 구조를 계속 단순화·재설계할 수 있도록 설계된 점에서, 한 번 구축하면 고정되는 것이 아니라 '지속 갱신 가능한' 플랫폼 지향
- 앤트로픽은 인간 통제, 정렬, 상호작용 보안, 투명성, 프라이버시를 포함한 신뢰 프레임워크를 제시하며, 관리형 에이전트 운영의 기준을 구조화

OpenAI, 실행 중심 통합 플랫폼

- '26년 2월 공개한 통합 관리 플랫폼 Frontier가 기업 내 모든 에이전트를 총괄하는 상위 지능 계층으로 기능하며, 개별 에이전트가 개별 업무를 수행하는 수준을 넘어 기업 전체의 에이전트 운영을 조율
- 프런티어 모델부터 에이전트 개발·운영 도구까지 전 계층을 자체 보유해 모델과 실행 체계가 일체화된 수직 통합 구조를 형성하며, 모델 성능 개선이 플랫폼 전체 품질로 즉시 연동
- 기존 업무 시스템을 새로 바꾸지 않고도, 현재 사용하는 애플리케이션과 연결해 단계적으로 도입할 수 있는 구조를 제공
- ChatGPT·AI 브라우저·기존 업무 앱 등 어느 접점에서든 동일한 에이전트에 접근 가능하여, 약 9억 명에 달하는 사용자 기반에서 축적된 친숙성이 기업 도입 시 직원의 초기 학습 부담을 줄이는 요소로 작용

- Google, 연결과 통합을 결합한 생태계
  - A2A(에이전트 간 통신 프로토콜) 등 개방 표준을 선제적으로 제안·채택하며 AI 에이전트와 외부 시스템·타사 에이전트 간 연결 표준 자체를 업계 공용 자산으로 확장하는 전략 전개
  - 상거래·결제·사용자 인터페이스 등 용도별 연결 규격도 잇따라 제시해, 에이전트가 업무 도구뿐 아니라 결제·화면 등 실제 업무 전반과 연결될 수 있는 기반 다층화
  - 진입 장벽을 낮춘 개발 도구를 폭넓게 배포해 외부 개발자와 파트너의 참여를 유도하며, 이들이 만든 에이전트가 다시 플랫폼의 가치를 높이는 생태계 확장 효과 추구
  - 결국, Google은 개방형 생태계 요소도 일부 품고 있지만, 실제 사용 방식은 단일 보안 환경 안에서 에이전트를 만들고 운영하는 하이브리드 플랫폼 전략으로 진화

에이전트 AI 플랫폼 경쟁의 본질: OS와 슈퍼앱의 결합

- 백엔드의 플랫폼, 여러 에이전트를 실행하는 차세대 AI OS로 진화
  - 에이전트 AI 플랫폼은 여러 에이전트를 공통 실행 환경 위에서 작동시키고, 도구 연결과 권한 부여, 운영 점검을 함께 관리한다는 점에서 차세대 AI 운영체제에 가까운 역할을 수행
  - 과거 운영체제가 여러 애플리케이션을 하나의 시스템 위에서 실행·관리했다면, 에이전트 플랫폼은 여러 에이전트를 하나의 구조 안에서 연결하고 움직이게 하는 AI OS형 기반으로 진화
  - 이 과정에서 실행 안정성, 연결 조율, 중앙 통제, 운영 가시성 같은 기능이 플랫폼의 기본 역량으로 부상하며, 플랫폼 자체의 완성도가 더 중요한 경쟁 요소로 부각
  - 결국 백엔드 차원의 경쟁은 더 많은 에이전트를 보유하는 문제를 넘어, 누가 더 안정적이고 일관된 실행 기반을 제공하느냐로 수렴
- 프론트엔드의 플랫폼, 다양한 기능을 묶는 슈퍼앱형 사용자 경험으로 진화
  - 사용자 관점에서 에이전트 플랫폼은 검색, 문서 작성, 일정 관리, 업무 자동화 같은 다양한 기능이 하나의 흐름 안에서 이어지는 슈퍼앱형 경험으로 발전하는 방향
  - 이는 개별 앱을 번갈아 여는 방식보다, 하나의 플랫폼 안에서 여러 기능이 자연스럽게 연결되고 사용자가 AI를 통해 이를 한 번에 실행하는 구조를 의미

- 플랫폼이 강해질수록 사용자는 개별 서비스보다 플랫폼 자체를 기본 진입 지점으로 사용하게 되고, 그 안에서 다양한 기능과 에이전트가 결합된 통합 경험을 소비하게 되는 흐름
- 반면 프론트엔드 차원의 경쟁은 기능의 양보다, 여러 기능을 하나의 매끄러운 사용 경험으로 통합할 수 있는지에 달려 있음

출처 : McKinsey 외(2026.4.)
https://www.mckinsey.com/capabilities/mckinsey-technology/our-insights/building-the-foundations-for-agentic-ai-at-scale
https://www.mckinsey.com/capabilities/tech-and-ai/our-insights/tech-forward/state-of-ai-trust-in-2026-shifting-to-the-agentic-era
https://www.deloitte.com/us/en/what-we-do/capabilities/applied-artificial-intelligence/articles/agentic-ai-insights.html
https://themiilk.com/articles/a8e11a610
https://openai.com/index/next-phase-of-enterprise-ai/
https://developers.openai.com/blog/openai-for-developers-2025
https://docs.cloud.google.com/agent-builder/overview?hl=ko
https://www.anthropic.com/research/trustworthy-agents
https://www.anthropic.com/engineering/managed-agents
https://learn.microsoft.com/en-us/azure/foundry/agents/overview
https://menlovc.com/perspective/2025-the-state-of-generative-ai-in-the-enterprise/
https://www.wakeupnews.co.kr