HyperTeams

6회차 · 50분

도구와 MCP

도구를 붙인다는 것은 권한을 주는 일입니다

표준이 없던 시절

도구 3개 × 에이전트 3개 = 9번 만들어야 했습니다

표준 전	
3 × 3	**9**
5 × 4	**20**
10 × 5	**50**

MCP	
3 + 3	**6**
5 + 4	**9**
10 + 5	**15**

제공하는 것

세 가지

01

도구 (Tools)

에이전트가 부르는 동작. 실무에서 대부분 이것

02

리소스 (Resources)

에이전트가 읽는 자료

03

프롬프트 (Prompts)

미리 준비된 지시 템플릿

도구 설명은 사람이 아니라 모델에게 쓰는 문서입니다. 모델이 그걸 읽고 언제 쓸지 판단합니다.

도구가 많아지면

비용보다 정확도가 더 아픕니다

서버 4개만 붙여도 도구는 80개가 넘습니다.

- 도구 정의가 전부 컨텍스트에 들어갑니다
- 비슷한 도구가 여럿이면 잘못 고릅니다
- 사람도 공구 200개 작업대에서 더 자주 틀리게 집습니다

~12,000

도구 82개의 정의 토큰

아직 아무 일도 안 했는데

붙이기 전

네 가지를 확인하세요

01

닿는 범위는 어디까지인가

발급하는 토큰의 권한이 곧 그 답

02

되돌릴 수 없는 동작이 있나

삭제 · 발송 · 결제

03

사람 승인을 끼울 수 있나

없으면 그 도구는 붙이지 않는 게 맞습니다

04

자격증명을 어디에 두나

대화창·코드·저장소에 두지 말 것

각 도구는 안전한데 조합이 위험할 수 있습니다 — 파일읽기 + 메일발송 = 사내 문서 외부 유출 경로.

AI가 읽는 데이터가 곧 명령이 될 수 있습니다

지시가 들어오는 통로와 데이터가 들어오는 통로가 분리되어 있지 않습니다 — 그래서 완전히 막는 방법이 아직 없습니다

사고의 세 조건

하나만 끊어도 사고가 안 납니다

01

심긴 지시가 들어옴

웹 · 문서 · 메일 · 남의 MCP 응답. 외부 자료를 읽는 게 업무라 못 끊습니다

02

민감한 것에 닿음

읽기와 발신을 같은 워크스페이스에 두지 않습니다

03

밖으로 나가는 길

메일 · 웹훅 · 게시 · 결제에 사람 승인을 겁니다

실무의 초점은 2번과 3번입니다. 필터는 우회됩니다 — 필터에 기대어 권한을 넓히면 안 됩니다.

실습 · 30분

지금 켜져 있는 도구를 훑습니다

1
전부 적기
내장 도구까지. “마지막으로 쓴 때”를 꼭 채울 것

2
세 갈래로
읽기 / 되돌릴 수 있는 쓰기 / 되돌릴 수 없음

3
읽기를 다시
사내 기밀 · 개인정보를 읽는 것을 따로

4
조합 찾기
[민감한 읽기] + [밖으로 나가기] = 유출 경로

판정 기준은 “지울 수 있는가”가 아니라 “누군가 이미 봤는가”입니다 — 슬랙 게시는 지워도 본 사람은 봤습니다.

정리

오늘 가져가실 것

1. MCP는 AI와 도구 사이의 USB — 연결 방식만 표준화했습니다
2. 도구 설명은 모델에게 쓰는 문서입니다
3. 안 쓰는 도구를 끄는 것은 절약이자 정확도 개선
4. 승인은 도구가 아니라 "되돌릴 수 없는 동작"에 겁니다
5. 읽은 것이 명령이 될 수 있습니다 — 막는 대신 닿는 범위와 나가는 길을 끊습니다