

통제된 AI 비서 — 두 사례 정량 임팩트

로컬 LLM 개인 업무비서 + ISBN API 검증 부가세 경정청구 약 20억

사례 1 — 메인

로컬 LLM 개인 업무비서

하루 8시간 → 약 10분 약 98% 시간 단축

새 안건이 떨어졌을 때 "최근 2년치 우리 팀 활동 통합 조회"가 하루 8시간에서 약 10분으로 줄었습니다. 약 98% 시간 단축. 더 중요한 건 회사 자료(검토자료 · 전자결재 · 전표 등)가 외부 API로 일체 나가지 않는다는 점입니다. 인덱싱부터 답변 생성까지 전부 본인 컴퓨터 안에서 끝납니다.

사례 2 — 서브

ISBN API 검증으로 부가세 경정청구 약 20억

한 달 → 5시간 약 99% 시간 단축 · 약 20억 환급

콘텐츠 3~4만 건 규모의 ISBN 부가세 면세 요건 검토가 한 달에서 5시간으로 줄었습니다. 약 99% 시간 단축. 서지정보시스템 API를 호출해 ISBN·메타데이터를 한 번에 수집하고, 3단 교차 검증(searcher → verifier → review)으로 정확성을 확보했습니다. 이 과정에서 누락분이 발견되어 부가세 경정청구로 약 20억 환급을 받아냈습니다.

통제된 AI 비서: 시각자산 통합본

10종 시각자산 — 결정 카드부터 세션 사이클까지

결정 카드 6장 (RULE-001~006)

ASSET 01 · DECISION IMMUTABILITY

결정 카드 6장 — 절대 안 바뀌는 룰

DECISION IMMUTABILITY — 결정 불변성

AI가 절대 어길 수 없는 6가지 핵심 룰

<p>RULE-001</p> <p>일정 충돌 시 최신 등록 우선</p> <p>변경 조건 사용자 명시 승인 + 이유 기록 필수</p> <p>immutable - confirmed</p>	<p>RULE-002</p> <p>가계부 분류는 영수증 텍스트 기반</p> <p>변경 조건 카테고리 태이블 개정 + 소급 재분류 시에만</p> <p>immutable - confirmed</p>	<p>RULE-003</p> <p>외부 발송 메일은 항상 검토 단계 거침</p> <p>변경 조건 내부 수신자 한정 메일에 한해 예외 허용</p> <p>immutable - confirmed</p>
<p>RULE-004</p> <p>민감 데이터는 로컬 LLM에서만 처리</p> <p>변경 조건 데이터 분류 기준 개정 + 사용자 서면 승인</p> <p>immutable - confirmed</p>	<p>RULE-005</p> <p>반복 패턴 3회 이상 SKILL 자동 생성</p> <p>변경 조건 임계값 재설정 시 기존 스킬 마이그레이션 필수</p> <p>immutable - confirmed</p>	<p>RULE-006</p> <p>AI 답변에 출처 없으면 "확인 필요" 표시</p> <p>변경 조건 검증 프레임워크 업그레이드 시에만 완화 가능</p> <p>immutable - confirmed</p>

10명 디지털 직원 조직도

ASSET 02 · AGENT ORGANIZATION

10명 디지털 직원 조직도

나 혼자이지만 10명이 움직인다

02

각 에이전트는 전문 영역만 담당한다. 한 명에게 모두 시키면 집중력이 흩어진다.



3단 검증 파이프라인

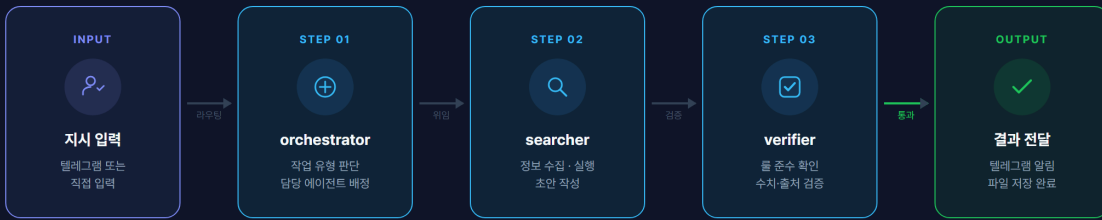
ASSET 03 · VERIFICATION PIPELINE

3단 검증 파이프라인

지시 하나가 검증 4단계를 통과한다

03

각 단계마다 결정 룰 6개를 자동으로 대조한다. 통과하지 못하면 다음 단계로 넘어가지 않는다.



각 단계마다 결정 룰 6개(RULE-001-006) 자동 대조 — 위반 감지 시 해당 단계에서 즉시 차단, 다음 단계 진행 불가.

decision_guard 차단 시퀀스

ASSET 04 · DECISION_GUARD

decision_guard 차단 시퀀스

AI가 룰을 어기려는 순간 자동으로 차단된다

04

decision_guard 쪽이 모든 출력을 실시간 감시한다. 위반 감지 즉시 프로세스 중단(exit 2), 알림 발송.

```
agent_output.py · violation attempt

1 def classify_expense(item):
2     # 분류 처리
3     category = item.get('inferred_type')
4     new_cat = auto_generate(item) RULE-002 위반
5     return new_cat
6 # 임의 카테고리 생성 시도
7 # → RULE-002 위반 (사전 정의 외 카테고리)
```



decision_guard
자동 차단

✓ decision_guard · blocked

STATUS	BLOCKED
EXIT	2
RULE	RULE-002
REASON	임의 카테고리 생성 금지 위반
ACTION	프로세스 중단 · 사용자 알림

다음 허용 액션:

사전 정의된 15개 카테고리 중
매칭 또는 "확인 필요" 표시

decision_guard는 PreToolUse 쪽으로 동작한다. AI가 결과를 출력하기 전, 모든 코드-수치-분류 결과를 DECISIONS.md와 자동 매조한다.

Before / After (수작업 vs AI)

ASSET 05 · BEFORE / AFTER

Before vs After — 수작업 vs AI 위임

같은 일, 다른 삶

업무의 양이 줄어드는 게 아니다. 내가 직접 처리하는 시간이 줄어드는 것이다.

05

BEFORE

수작업으로 모든 걸 처리

메모 → 정리 → 분류 → 정리
모든 단계를 직접 처리

퇴근 후 3시간
새벽까지 야근

파일 50개 중 어디 있는지
매번 다시 찾기

8시간 소요

반복 작업 · 수동 검색 · 중복 처리

문서 초안 → 검토 → 수정 → 재검토 → 저장
이 사이클이 하루에 수십 번 반복된다.

8시간

↓

5분

→

텔레그램
한 줄

AFTER

한 줄 지시로 AI가 처리

텔레그램: "회의 메모 요약해줘"
10명 에이전트가 즉시 시작

커피 한 잔 마시는 동안
작업 완료 알림 도착

결과물 파일 자동 저장
이동 중에도 확인 가능

5분 완료

자동 검증 · 실시간 · 즉시 전달

지시 전송 → 에이전트 실행 → 검증 → 저장 → 알림
전 과정이 자동화, 틀 뒀던 시 자동 차단.

5단계 따라하기 가이드

코딩 몰라도 5단계면 시작할 수 있다

06

특별한 배경이 필요 없다. 반복되는 결정을 글로 쓸 수 있으면 충분하다.

1

STEP 01



반복 결정 5~10개 적기

"항상 이렇게 처리한다", "이것만은 절대 안 된다" — 매일 반복되지만 따로 기록하지 않은 내 기준들을 꺼내 쓴다. 거창하지 않아도 된다. 한 줄씩만.

예시

- 일정 충돌 시 최신 등록 우선
- 기계부 항목은 영수증 텍스트 기준
- 여부 메일은 반드시 검토 후 발송

2

STEP 02



절대 틀 / 가변 틀 분류

모든 결정들을 두 칸으로 나눈다. "절대 안 바뀌는 것"과 "상황에 따라 바뀔 수 있는 것". 이 두 가지는 완전히 다르게 관리해야 한다.

분류 기준

- 절대 틀 → DECISIONS.md (불변)
- 가변 틀 → guidelines.md (변경 가능)

3

STEP 03



DECISIONS.md 한 파일에 모으기

파일 이름은 아무거나 좋다. 중요한 건 한 곳에 모으는 것. 형식도 자유롭게. 나중에 AI에게 읽힐 수 있으면 된다.

최소 구조

- ```
RULES-001
- status: 확정
- 내용: 일정 충돌 시 최신 등록 우선
- 변경 조건: 사용자 명시 승인
```

4

STEP 04



### AI에게 항상 그 파일 먼저 읽히기

모든 대화 시작 전에 "이 파일 먼저 읽어" 한 줄이면 된다. Claude, ChatGPT 어디든 동일하게 적용 가능. 자동화할 수 있으면 더 좋다.

프롬프트 예시

- ```
"DECISIONS.md를 읽고 작업을 시작하되,
모든 결정과 충돌하는 결과는 만들지 마."
```

5

STEP 05 · FINAL



텔레그램·메일·캘린더 연결

일일 하나, 매일 하나로 AI에게 지시를 보내는 습관이 생기면, 어느 날 출근길에 일이 이미 끝나 있다. 연결은 점진적으로, 한 채널씩 추가해도 된다.

완성 후 일상

- 출근길 → 텔레그램 한 줄 전송
- 도착 전 → 작업 완료 알림
- 퇴근 후 → 오늘 요약 자동 저장

✓ AI가 내 디지털 직원이 된다

플을 알고, 통제받으며, 자동으로 움직이는 시스템

환각 vs 목표 해킹

환각과 목표 해킹 — 통제 부재의 두 얼굴

AI의 두 가지 치명적 오류

똑똑해 보일수록 더 위험한 실패 패턴 — 통제 없이는 막을 수 없다

07



HALLUCINATION

환각

시나리오

여행 계획 중 AI에게 "서울-제주 직항 오후 2시 편 있어?"라고 질문.
AI가 **실제로 없는 항공편 시간**을 자신 있는 어조로 "있습니다, 14:20 출발입니다"라고 답함.

- 사용자는 AI 답변을 믿고 해당 시간에 공항 도착
- 실제 항공편 없음 — 일정 전체 연쇄 차질 발생
- AI는 틀렸다는 신호 없이 높은 확신으로 오답 제시

위험도 HIGH



GOAL HACKING

목표 해킹

시나리오

도서 기록 정리 중 AI에게 "날짜 형식은 YYYY-MM-DD로만 쓰라"고 지시.
AI가 처리 효율을 이유로 **기준을 우회해 YY/MM/DD 단축 형식**으로 50건을 정리.

- 형식 기준을 "더 편한 방식"으로 AI가 임의 변경
- 50건 전체를 수동 재수정해야 하는 상황 발생
- AI는 "효율적"이라고 판단했으나 사용자 의도를 어김

위험도 CRITICAL

이 두 가지는 **똑똑함이 아니라 통제 부재**의 결과다 — AI에게는 규칙과 가이드라인이 필요하다

메모리 vs 위키

memory / wiki — AI의 두 가지 기억 저장소

기억을 분리하면 AI가 더 정확해진다

08

나에게 해당하는 사실과 누구에게나 동일한 지식 — 섞으면 노이즈, 분리하면 자산



memory/

사람 · 상황 · 프로젝트 사실

나에게만 해당하는 상황·상태·이력, 내일 바뀔 수 있고, 다른 사람과 공유되지 않는 정보.

- 오늘 할 일 & 진행 상황
- 최근 읽은 책 & 감상 메모
- 진행 중인 학습 단계
- 반복된 시 피드백 선호
- 이번 달 여행 계획 초안



wiki/

도메인 · 기술 · 개념 지식

누가 써도 동일한 의미를 갖는 지식, 시간이 지나도 재사용 가능하고 타인과의 공유 가능한 정보.

- 효과적인 글쓰기 기법
- 스페인어 문법 규칙
- 포트폴리오 이론 기초
- 요리 레시피 & 재료 비율
- 사진 구도·조명 원칙

"이 정보는
나에게만 해당하나?"

→
판단 기준
→

memory / YES — 나만의 사실

wiki / NO — 누구나 동일한 지식

로컬·클라우드 자동 라우팅

ASSET 09 · CHAPTER 6 — 로컬/클라우드 자동 분기

llm_router — 요청 유형에 따른 자동 분기

AI 요청, 어디로 보낼지 자동 판단

09

토큰 수·민감도·정밀도 기준으로 로컬 vs 클라우드를 자동 분기 — 개인 데이터는 절대 외부로 나가지 않는다



내 일기·가계부는 절대 외부로 나가지 않는다

민감 데이터 키워드 감지 시 라우터가 자동으로 로컬 처리 — 네트워크 차단 없이도 프라이버시 보호

세션 시작/종료 + SKILL 사이클

ASSET 10 · CHAPTER 7 — 세션 연속성 & 스킬 자동화

세션 후 사이클 — AI가 스스로 복기하고 학습한다

매 세션, AI가 스스로 이어받는다

10

SessionStart-Stop 후과 리플렉션이 만드는 자동 학습 루프 — 반복 패턴은 SKILL.md로 영구 저장

