2026 전국민 AI 활용사례 공모전 · 업무 생산성 개선

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 규모**의 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 정합성을 확보했습니다. 이 과정에서 누락분이 발견되어 **부가세 경정청구로 약 20억 환급**을 받아냈습니다.

핵심 메시지 · "AI는 똑똑하게 쓰는 게 아니라 통제해서 쓰는 것이다"

AI 도구 사용 명세서

제출 구분: 개인 응모
출품작명: 통제된 AI 비서: 직장인 1명이 만든 디지털 직원 시스템
작성 목적: 공모 가이드에 따라 사용한 AI 도구 및 활용 방식을 사전 명세함

---

1. 사용 AI 도구 목록

구분	도구명	버전·모델	제공사	역할
클라우드 LLM	Claude Code (Anthropic CLI)	claude-sonnet-4-6	Anthropic	에이전트 시스템 하네스·개발 환경
클라우드 LLM	Claude API (직접 호출)	claude-haiku-4-5-20251001	Anthropic	고정밀 답변 폴백
로컬 LLM	Ollama + qwen2.5:3b	qwen2.5:3b	Alibaba Cloud / Ollama	분류·태깅·대량 배치 추론
로컬 임베딩	nomic-embed-text	v1.5	Nomic AI / Ollama	위키 노트 벡터 인덱싱
메시징 플랫폼	Telegram Bot API	v7.x	Telegram	원격 지시 큐 송수신
지식 관리 도구	Obsidian	최신 안정 버전	Obsidian	개인 도메인 지식 vault

> 위 도구 중 Claude Code, Claude API, Telegram Bot API, Obsidian은 각사가 제공하는 기성 서비스이다.
> 이하 섹션 2에서 기술하는 에이전트 구조·거버넌스·훅·라우팅 정책은 응모자가 직접 설계·구현하였다.

---

2. 응모자 직접 설계 요소

이 출품작의 핵심은 기성 AI 도구를 단순 사용하는 것이 아니라, **조직형 에이전트 시스템과 거버넌스 레이어를 개인이 직접 설계·운영**한 데 있다.

2-1. 전문 에이전트 R&R 분리 (10명 디지털 직원)

단일 AI에 모든 질문을 던지는 방식 대신, 업무 유형별로 독립된 에이전트를 정의하고 역할을 분리하였다. 응모자 본인은 "팀장(orchestrator)" 역할을 수행하며, 아래 10명의 디지털 직원이 각자의 영역을 담당한다.

#	에이전트	담당 역할
1	writer	문서·메일·블로그·보고서 초안 작성
2	designer	시각 자산·차트·슬라이드·서식 디자인
3	developer	코드·매크로·자동화·API 연동
4	scheduler	일정 관리·리마인더·캘린더 정리
5	data	자료 수집·표 정리·중복 제거
6	research	외부 자료 조사·요약·인용 정리
7	review	작성물 검토·논리·맞춤법·일관성 점검
8	curator	지식 vault 정리·태깅·인제스트
9	dispatcher	외부 발송·메일·알림 통보
10	verifier	수치·논리 정합성 최종 검증

각 에이전트는 `description`, 허용 도구, 모델, 전용 점검 체크리스트를 마크다운 파일로 정의하며, 하네스(Claude Code)가 description을 기반으로 자동 라우팅한다. 정의 위치: `.claude/agents/*.md`.

2-2. 결정 불변성(Decision Immutability) 거버넌스

AI가 이전 합의를 임의로 번복하거나 목표 달성을 위해 제약을 우회하는 문제를 방지하기 위해 두 층의 방어 체계를 직접 구현하였다.

(a) decision_guard — PreToolUse 훅

- 위치: `.claude/hooks/decision_guard.py`
- 동작: 파일 쓰기·수정·Bash 명령 실행 직전, 핵심 결정(`memory/decisions/`)과의 충돌 여부를 키워드 매칭으로 검사
- 위반 감지 시: `exit 2` 반환 → Claude Code가 도구 실행을 자동 차단하고 사용자에게 사유 제시
- 우회 방지: 컬럼 파생·별칭 사용·간접 경로 등 모든 우회 패턴을 동일하게 차단
- 로그: `.claude/hooks/logs/decision_guard_YYYY-MM.log` 에 차단 이력 자동 기록

(b) goal_hack_detector — UserPromptSubmit 훅

- 위치: `.claude/hooks/goal_hack_detector.py`

- 동작: 사용자 입력에서 "도메인 키워드 + 수치 최적화 목표" 패턴을 감지하면 관련 결정 파일 환기(차단 아닌 경고·컨텍스트 주입)
- 목적: "성과를 부풀려줘" 류의 spec gaming 프롬프트에 제약 상기

(c) 결정 파일 관리 규칙

- 위치: `memory/decisions/` (git 관리, 이력 보존)
- 상태: `status: 확정` + `immutable: true` 결정은 변경 조건 충족 + 사용자 명시 승인 없이 수정 불가
- 기존 결정 삭제 금지: `status: 폐기`로 마킹 후 새 ID로 supersede

2-3. 메모리 + 위키 이중 저장소 분리

AI가 대화마다 컨텍스트를 초기화하는 문제를 해결하기 위해 두 종류의 장기 저장소를 분리 운영한다.

저장소	경로	저장 대상	판단 기준
memory/	`memory/` (자동 로드)	사람·팀·프로젝트 사실, 의사결정 근거, 반복 피드백	"특정 사람·시점에 묶인 정보?" → Yes
wiki/ (Obsidian vault)	`wiki/`	도메인·기술 지식, 재사용 가능한 개념	"누가 쓰든 동일한 지식?" → Yes

- wiki/ 수정 권한은 curator만 보유 (다른 에이전트는 읽기 전용)
- memory/ 갱신은 하네스가 사용자 승인 후 수행

2-4. 클라우드 / 로컬 LLM 자동 라우팅

비용·보안·정밀도 요구를 동시에 충족하기 위해 작업 특성에 따라 모델을 자동 분기하는 라우터를 직접 구현하였다.

- 위치: `scripts/llm_router.py`
- 분기 기준:

티어	사용 모델	분기 조건
fast	qwen2.5:3b (로컬)	짧은 분류·태깅·yes/no (토큰 300 이하)
quality	qwen2.5:3b (로컬, 확장 컨텍스트)	중간 길이 일반 QA (토큰 300~3000)
cloud	claude-haiku-4-5	고정밀 도메인 키워드 포함, 또는 3000토큰 초과

- 민감 개인정보 처리: 항상 로컬 모델 (클라우드 전송 금지)
- 정밀 답변 필요 작업: 항상 클라우드 (로컬 단독 금지)
- 클라우드 장애 시 로컬 quality로 자동 폴백

2-5. 텔레그램 inbox/outbox 원격 지시 큐

이동 중·퇴근 후에도 AI에게 업무를 위임하기 위해 파일 기반 비동기 큐 시스템을 구성하였다.

- 흐름: 텔레그램 봇 → `.claude/inbox/` (JSON 드롭) → Claude Code 5분 폴링 → 작업 실행 → `.claude/outbox/` 결과 → 봇 회신
- 봇 코드: `bot/telegram_bot.py` (직접 작성)
- 특징: 세션이 끊겨도 큐에 지시가 남아 있어 다음 Claude Code 기동 시 자동 처리

2-6. 자동 리플렉션 및 스킬 자동 포착

반복 작업을 재사용 가능한 스킬로 자동 축적하는 메커니즘을 설계하였다.

- **리플렉션 트리거**: 복잡 작업 완료 직후, 에러 해결 후, 사용자 종료 신호 시
- **스킬 포착 기준**: 도구 5회 이상 또는 멀티 에이전트 호출이 포함된 작업이 3개월 내 반복 가능한 패턴일 때
- **스킬 파일**: `.claude/skills/{이름}/SKILL.md` — 사용 횟수·최근 결과·개선 이력 자동 갱신
- 현재 축적된 범용 스킬 예시: html-email, doc-revision-review, gdrive-project-scan, gmail-monthly-analysis, karpathy-wiki-builder

2-7. 세션 연속성 자동 유지 (SessionStart / Stop 훅)

AI 대화 세션이 매번 초기화되는 한계를 훅으로 극복하였다.

- **SessionStart 훅** (`.claude/hooks/session_start_brief.py`): 세션 시작 시 STATUS.md(진행중 프로젝트), CURRENT.md(직전 스냅샷), 최근 3세션 요약을 자동 주입
- **Stop 훅** (`.claude/hooks/session_end_summarize.py`): 매 턴 종료 시 CURRENT.md 자동 갱신, 일 1회 세션 원본 `.jsonl` 아카이브 백업

---

3. 일반 사용자 대비 차별화 요약

구분	일반적 AI 활용	본 출품작
질의 방식	단발성 프롬프트 → 응답	역할 분리된 10명 디지털 직원에게 팀장이 라우팅
기억	세션 내 한정	memory/(사실) + wiki/(지식) 이중 장기 저장소
일관성 보장	없음	decision_guard 훅이 합의 위반 시 자동 차단
안전성	없음	goal_hack_detector로 목표 해킹 패턴 경보
접근성	PC 앞에서만	텔레그램 원격 큐로 이동 중에도 업무 위임
비용 관리	항상 클라우드	로컬/클라우드 자동 분기로 불필요한 API 비용 절감
노하우 축적	대화에 소멸	스킬 파일로 절차 자동 저장·재사용

구분	일반적 AI 활용	본 출품작
회사 자료 처리	외부 클라우드로 무방비 전송	회사 자료는 강제 로컬 LLM 처리, 외부 전송 0건

3-A. 실제 적용 사례 (정량 임팩트)

사례 1 (메인) — 로컬 LLM 개인 업무비서

- **상황**: 새 안건이 떨어졌을 때 최근 2년치 우리 팀 활동(검토자료·전자결재·전표 등)을 통합 조회·시간순 정리해야 함. 회사 자료라 외부 AI에 보낼 수 없음.
- **방법**: Ollama + qwen2.5:3b + nomic-embed-text 로컬 임베딩으로 본인 PC 안에서 인덱싱·답변. 라우터에서 회사 자료 키워드 감지 시 강제 로컬 분기.
- **결과**: 자료 통합 조회·정리 작업 **하루(약 8시간) → 약 10분, 약 98% 시간 단축**.
- **부가 효과**: 외부 API 전송 0건. 안건 분석에 즉시 들어갈 수 있어 일의 질 자체가 달라짐.

사례 2 (서브) — ISBN API 검증으로 부가세 경정청구 약 20억 환급

- **상황**: 콘텐츠 3~4만 건 규모의 ISBN 부가세 면세 요건 검토. 옛날에는 서지정보시스템 사이트에 들어가 한 건씩 제목·저자로 검색·매칭, 한 달 소요.
- **방법**: orchestrator → searcher → verifier 3단 파이프라인. searcher가 서지정보시스템 API를 호출해 ISBN·메타데이터 일괄 수집, verifier가 체크디짓·서지 매칭·형식 정합성 3단 교차 검증.
- **결과**: 작업 소요 **약 1개월 → 약 5시간, 약 99% 시간 단축**.
- **부가 효과**: 검증 과정에서 누락 면세 적용분 발견, **부가세 경정청구로 약 20억 환급**.

---

4. 응모자 설계 부분 vs 외부 도구 명확 구분

외부 도구(기성 서비스)

- Claude Code CLI, Claude API: Anthropic 제공
- Ollama 런타임: Ollama 제공 / qwen2.5:3b 모델: Alibaba Cloud 제공
- nomic-embed-text: Nomic AI 제공
- Telegram Bot API: Telegram 제공
- Obsidian 앱: Obsidian 제공

응모자 직접 설계·구현

- 10명 디지털 직원 description 및 역할·도구·점검 체크리스트 정의

- `decision_guard.py` — PreToolUse 훅 (결정 불변성 차단 로직)
- `goal_hack_detector.py` — UserPromptSubmit 훅 (목표 해킹 감지 로직)
- `session_start_brief.py` / `session_end_summarize.py` — 세션 연속성 훅
- `llm_router.py` — 클라우드/로컬 자동 분기 라우터
- `local_llm.py` — Ollama API 래퍼 (프로파일 3종, 스트리밍)
- `bot/telegram_bot.py` — 텔레그램 원격 지시 큐 봇
- memory/ + wiki/ 이중 저장소 구조 및 갱신 규칙
- `.claude/skills/` 스킬 자동 포착·갱신 체계
- `memory/decisions/` 결정 파일 관리 규칙 및 git 이력 체계

---

5. 윤리·저작권·개인정보 준수 선언

1. **소속 회사 데이터·고객 정보 미사용**: 본 출품작 시스템 설계 및 출품 자료에는 소속 조직의 내부 데이터, 고객 정보, 미공개 수치를 일체 포함하지 않는다.
2. **개인정보 처리**: 개인정보가 포함될 수 있는 작업은 전량 로컬 LLM(qwen2.5:3b, nomic-embed-text)으로만 처리하며 외부 API로 전송하지 않는다.
3. **외부 콘텐츠 인용**: 외부 자료를 참조한 경우 출처를 명시한다.
4. **AI 생성물 표시**: 출품 문서 작성 과정에서 AI 도구를 활용하였으며, 이를 투명하게 공개한다.
5. **라이선스 준수**: 사용된 오픈소스 모델(qwen2.5:3b, nomic-embed-text) 및 소프트웨어(Ollama, Obsidian)의 라이선스 조건을 준수한다.

작성일: 2026-04-28 / 초안 v1.1 (검수 반영)
심사위원 요청 시 개별 파일(`.claude/agents/` , `.claude/hooks/` , `scripts/`) 제출 가능