

WHITEPAPER

MSAP.ai

저장하는 순간 지식이 된다 — 회사 문서가 AI의 근거가 되는 위키, MindWiki

배경과 도입 시 고려사항을 한 번에

생성형 AI 답변의 품질은 모델 성능보다 근거로 쓰는 회사 데이터가 정합니다. 그런데 회사 문서는 개인 PC와 공유 폴더, 업무 시스템에 흩어져 있고, RAG 구축과 그래프 관계 설정은 개발자 몫으로 남아 있어 근거 있는 답변으로 이어지지 못합니다(1장).

2026-11-27 · 작성자: SMART CF Team · msap.ai

목차

저장하는 순간 지식이 된다 — 회사 문서가 AI의 근거가 되는 위키, MindWiki

의사결정자를 위한 핵심 요약 — 회사 문서가 AI 근거가 되는 설치형 위키

1장: 회사 문서가 AI 답변의 근거 데이터가 되어야 하는 이유 — 흩어진 문서와 외부 시스템에 갇힌 데이터의 한계

1.1 문서 작업이 대부분인 회사 업무와 AI 답변 품질을 가르는 업무 데이터

1.2 AI와 일할수록 흩어지는 마크다운 파일과 외부 시스템에 갇힌 데이터

1.3 RAG 구축과 그래프 관계 설정이 개발자 몫으로 남는 기존 방식의 한계

1.4 근거 있는 AI 답변을 위한 지식 위키의 다섯 가지 조건

2장: 사람과 AI가 함께 쓰는 설치형 지식 위키 MindWiki — 익숙한 협업 노트 사용법과 달라진 세 가지

2.1 익숙한 사용법은 유지하고 설치 위치와 AI 모델, 지식이 되는 방식을 바꾼 설계

2.2 MindWiki 네 가지 동작 — 문서 작성, 근거 검색, 에이전트 위임, 사내 시스템 연결

2.3 MindWiki 전체 구성 — 위키 본체와 사내 AI 모델, MSAP.ai 실행 기반

2.4 MindWiki가 맞는 조직과 업무 — 근거 문서가 쌓이는 답변 업무

3장: 저장하는 순간 만들어지는 키워드·벡터·그래프 색인과 근거를 단 하이브리드 검색

3.1 원본 두 가지와 자동으로 다시 만드는 세 가지 색인 — 본문과 행 속성의 투영

3.2 한국어 형태소 BM25 키워드 색인과 1024차원 HNSW 벡터 색인

3.3 PostgreSQL 안의 Apache AGE 그래프와 저장 후 색인 반영 시간

3.4 근거를 단 답변까지의 하이브리드 검색 파이프라인 — 키워드 50건과 벡터 50건의 RRF 합산

3.5 사내 평가 세트 기준 검색 품질 — MRR 0.815와 재현율@10 0.810

4장: 업무 속성으로 만드는 지식 그래프 — 개발자 없이 현업이 정하는 관계 질의

4.1 문서 더미에서 LLM으로 그래프를 뽑는 방식의 비용과 품질 문제

4.2 담당자가 정한 업무 속성이 그대로 그래프 엣지가 되는 데이터베이스 설계

4.3 비어 있는 속성을 AI가 제안하고 사람이 수락하는 그래프 확장

4.4 관계를 묻는 질문에 답하는 그래프 질의 — 사내 평가 기준 0에서 0.750

5장: AAS 에이전트가 위키를 읽고 쓰는 구조와 사람 승인으로 이어지는 지식 순환

5.1 AAS 로그인 한 번으로 연결되는 에이전트와 사용자 권한 그대로의 접근

5.2 읽기 19개와 쓰기 17개 MCP 도구, 지우기 도구를 두지 않은 설계

5.3 자동화 규칙이 AAS 카드로 일을 넘기고 사람이 승인하는 처리 흐름

5.4 모든 페이지의 AI 버튼과 팀별 AAS 프로필 선택

5.5 위키에 쌓일수록 에이전트 답이 좋아지는 지식 순환 구조

6장: 데이터 주권을 지키는 MSAP.ai 설치형 구성 — 회사 밖으로 나가지 않는 데이터와 AI 모델

6.1 데이터 주권이 공공기관과 기업 AI 도입의 선행 조건이 된 배경

6.2 MSAP.ai 위 설치형과 인터넷 없이 동작하는 화면과 사내 CogentAI 모델

6.3 가드레일과 암호화 속성 — 대량 개인정보 추출 차단과 칼럼 단위 AES-256-GCM

6.4 페이지 트리 6단계 권한과 행 접근 규칙, API 토큰 범위의 권한 3층

6.5 PostgreSQL 이중화와 백업 보관 주기 — DB 매일 7일, WAL 2일, 첨부 매일 7일

7장: 플러그인 SDK와 HWP 변환으로 사내 시스템과 한글 문서를 위키에 연결하는 구조
7.1 4개 언어 플러그인 SDK로 연결하는 사내 게시판과 메신저, 결재 시스템
7.2 한글(HWP/HWPX)과 DOCX 첨부의 원본 보기와 하위 페이지 변환, 자동 색인
7.3 회의 녹음에서 화자 분리 회의록과 칸반 할 일 카드까지 이어지는 흐름
7.4 기본값이 꺼진 외부 연결 선택 기능 — 외부 검색과 지도, 링크 임베드
8장: 회사 세트로 구성하는 팀스페이스와 공용 데이터베이스 체계 — 영업에서 기술지원까지 이어지는 고객 정보
8.1 7개 팀스페이스와 공용 데이터베이스를 한 번에 설치하는 회사 세트 — 자동화 117개와 관계 속성 325개
8.2 공유 규칙 네 단계 — 회사 전체 공개부터 「권한 없음」 칸 표시까지
8.3 영업 수주에서 구축, 기술지원으로 이어지는 고객 정보 흐름
8.4 「데이터베이스는 하나, 보기는 여럿」 원칙과 세트 변경을 반영하는 운영
9장: 공공기관 민원 답변 자동화 사례 — 법령 근거 초안과 담당자 승인으로 바뀐 처리 흐름
9.1 근무일 7일 답변 기한과 법령 근거가 요구되는 공공기관 민원 업무의 과제
9.2 게시판 수집부터 게시판 등록까지 네 단계 민원 처리 흐름
9.3 시험 환경 기준 수치 — 처리 상태 7단계와 자동화 24개, 초안 한 건 약 2.5분
9.4 민원 사례를 다른 답변 업무로 넓히는 적용 조건
10장: 기존 노트·위키 도구 네 범주와 MindWiki 비교 — 설치 위치와 AI 모델, 지식이 되는 방식의 차이
10.1 비교 기준 열 가지 — 설치 위치부터 한국 업무 문서 처리까지
10.2 SaaS형 협업 노트 도구와 MindWiki — 공급사 클라우드와 회사 안 설치의 차이
10.3 엔터프라이즈 협업 위키와 MindWiki — 스페이스·페이지 트리 중심 상용 위키와 AI 근거 위키의 차이
10.4 로컬 파일형 마크다운 노트 도구와 MindWiki — 개인 파일 묶음과 조직 데이터베이스의 차이
10.5 오픈소스 셀프호스팅 위키와 MindWiki — 전문 검색 중심 위키와 저장 즉시 지식이 되는 위키의 차이
11장: MindWiki 도입 기대효과와 다섯 가지 가치 — 데이터 주권부터 조직 단위 즉시 도입까지
11.1 답변 업무와 검색 정확도, 기술지원, 선제 대응 영역의 기대효과
11.2 회의와 문서 자산, AI 답변 품질, 보안 영역의 기대효과와 측정 지표 후보
11.3 MindWiki 다섯 가지 가치 중 데이터 주권과 문서가 곧 AI 자산이 되는 구조
11.4 현업이 설계하는 지식 그래프와 코딩 없는 업무 자동화, 조직 단위 즉시 도입
12장: 세 단계 설치와 데이터 정돈 원칙으로 시작하는 MindWiki 파일럿 — 첫 업무 선정 기준
12.1 세 단계 설치와 회사 세트 맞추기 — 팀 고르기, 확인, 결과 약 5초
12.2 기존 문서 이관 — 사람이 옮기는 부분과 관리 API·관리자 스킬로 옮기는 부분
12.3 데이터 정돈 원칙 — 질문 유형별 답 방법을 정리한 FAQ와 하나의 데이터베이스
12.4 민원 답변과 기술지원, 회의록 중 첫 업무 하나로 정하는 파일럿 범위
부록 — 참고문헌과 본문 용어 풀이
참고문헌
본문에 나온 약어와 전문 용어 풀이

저장하는 순간 지식이 된다 — 회사 문서가 AI의 근거가 되는 위키, MindWiki

의사결정자를 위한 핵심 요약 — 회사 문서가 AI 근거가 되는 설치형 위키

생성형 AI 답변의 품질은 모델 성능보다 근거로 쓰는 회사 데이터가 정합니다. 그런데 회사 문서는 개인 PC와 공유 폴더, 업무 시스템에 흩어져 있고, RAG 구축과 그래프 관계 설정은 개발자 몫으로 남아 있어 근거 있는 답변으로 이어지지 못합니다(1장). MindWiki는 익숙한 협업 노트처럼 쓰면서도 저장하는 순간 키워드·벡터·그래프 색인이 함께 만들어지는 설치형 지식 위키이며, MSAP.ai 위 회사 안에서 사내 AI 모델과 함께 동작합니다(2장).

사내 평가 세트 기준으로 하이브리드 검색의 MRR은 0.815였고, 행 속성을 묻는 질문은 0에서 1.000으로, 여러 기록을 건너는 관계 질문은 0에서 0.750으로 올랐습니다(3장·4장). 이 구조는 데이터 주권, 문서가 곧 AI 자산이 되는 구조, 현업이 설계하는 지식 그래프, 코딩 없는 업무 자동화, 조직 단위 즉시 도입이라는 다섯 가지 가치로 이어집니다(11장). 첫 단계로 민원 답변·기술지원·회의록 가운데 우리 조직의 업무 하나를 골라 MindWiki와 AAS 에이전트로 파일럿 범위를 정하기를 권합니다(12장).

1장: 회사 문서가 AI 답변의 근거 데이터가 되어야 하는 이유 — 흩어진 문서와 외부 시스템에 갇힌 데이터의 한계

생성형 AI를 업무에 쓰려는 조직이 가장 먼저 부딪히는 문제는 모델 성능보다 답변의 근거입니다. 같은 LLM(Large Language Model, 대규모 언어 모델)을 쓰더라도 질문에 답할 때 참고할 회사 문서가 어디에 어떤 형태로 쌓여 있는지에 따라 답변이 상식 수준에 머물기도 하고 조직의 실제 업무 기준을 따르기도 합니다. 그런데 대부분의 조직에서 문서는 개인 PC의 파일, 부서별 공유 폴더, CRM(Customer Relationship Management, 고객 관계 관리)·게시판 같은 업무 시스템에 나뉘어 있어 AI가 한 번에 찾아 읽을 수 있는 상태가 아닙니다. 1장은 이 상태가 왜 근거 있는 AI 답변을 막는지 네 갈래로 짚고, OPENMARU MindWiki가 출발점으로 삼은 지식 위키의 다섯 가지 조건을 정리합니다. 여러분의 조직에서 문서가 지금 어디에 쌓이고 있는지 떠올리며 읽으면 2장 이후의 설명을 판단 기준과 함께 따라갈 수 있습니다.

1.1 문서 작업이 대부분인 회사 업무와 AI 답변 품질을 가르는 업무 데이터

회사 업무의 상당 부분은 보고서·회의록·제안서·기술지원 기록처럼 문서를 쓰고 고치는 일로 이루어지며, AI 답변 품질을 결정하는 것도 결국 그 문서에 담긴 업무 데이터입니다. 모델이 아무리 커도 조직 내부의 계약 조건이나 처리 기준은 학습 데이터에 들어 있지 않기 때문입니다. 따라서 AI 도입을 검토할 때는 모델 선택과 함께 업무 데이터를 AI가 읽을 수 있는 형태로 모으는 방법을 같은 무게로 따져야 합니다.

1.1.1 모델 지식과 공개 정보만으로는 상식 수준에 머무는 AI 답변

모델 지식과 공개 정보의 한계 AI 답변에 쓰이는 정보는 모델이 학습한 지식, 인터넷에 공개된 정보, 회사가 보유한 업무 데이터의 세 층으로 나눌 수 있습니다. 모델은 학습 시점 이후에 바뀐 제도나 제품 정보를 알지 못하고, 공개 정보는 검색으로 보강할 수 있지만 경쟁사를 포함한 모든 조직이 똑같이 얻는 정보입니다. 고객 문의에 대한 답변 초안을 요청했을 때 두 층만으로는 어느 회사에나 통하는 일반적인 안내문이 나옵니다. 조직마다 다른 답, 근거를 따라가 확인할 수 있는 답을 AI가 내려면 세 번째 층인 업무 데이터가 있어야 합니다.

업무 데이터가 만드는 조직별 답변 업무 데이터에는 고객사별 도입 이력, 과거 장애 처리 기록, 부서가 정한 처리 기준처럼 그 조직만 가진 사실이 담겨 있습니다. AI가 이 데이터를 검색해 답변에 근거로 붙이면 담당자는 답변 문장마다 원문을 열어 확인할 수 있고, 틀린 부분이 있으면 원문을 고쳐 다음 답변을 바로잡을 수 있습니다. 그래서 근거 있는 AI 답변을 원한다면 업무 데이터를 한곳에 모으고 검색 가능한 상태로 유지하는 저장소가 모델보다 먼저 준비되어야 합니다.

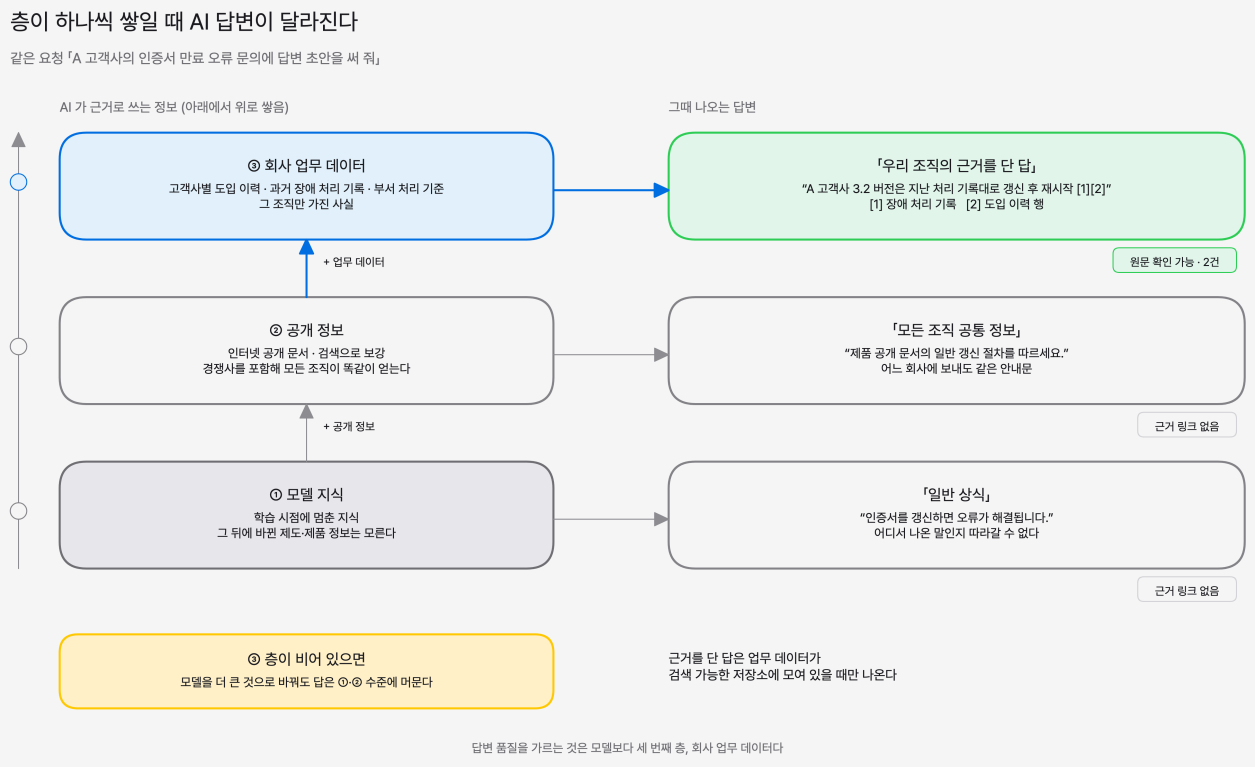

그림. 데이터 층별로 달라지는 AI 답변

1.1.2 평소 문서 작업이 별도 가공 없이 AI 근거로 쓰여야 하는 까닭

가공 단계가 남기는 비용과 시차 업무 데이터를 AI 근거로 쓰는 일반적인 방법은 문서를 따로 내려받아 정리하고, 텍스트를 잘라 임베딩한 뒤 벡터 데이터베이스에 넣는 RAG(Retrieval-Augmented Generation) 구축 프로젝트를 두는 것입니다. 이 방식에서는 직원이 문서를 고칠 때마다 가공 작업을 다시 돌려야 하므로 원문과 AI가 참고하는 사본 사이에 시차가 생깁니다. 가공 파이프라인을 만드는 비용과 운영하는 인력이 별도로 들고, 담당 개발자가 바뀌면 어떤 문서가 언제 반영됐는지 추적하기 어려워집니다.

작성과 색인이 한 번에 이루어지는 구조 반대로 직원이 문서를 저장하는 순간 키워드·벡터·그래프 색인이 함께 갱신되면 가공 단계 자체가 사라집니다. 그러면 AI는 사본을 거치지 않고 원문을 직접 참고하게 되고, 문서를 고친 내용은 몇 초 안에 검색 결과에 반영됩니다. 별도 RAG 구축 프로젝트를 둘지 판단할 때는 구축 비용뿐 아니라 원문과 사본을 계속 맞추는 운영 부담까지 함께 계산해야 합니다. MindWiki는 이 두 번째 구조, 즉 평소의 문서 작업이 그대로 AI 근거가 되는 구조를 기본으로 삼습니다.

1.2 AI와 일할수록 흩어지는 마크다운 파일과 외부 시스템에 갇힌 데이터

AI 도구를 업무에 많이 쓸수록 문서는 오히려 더 많이 흩어집니다. AI가 만들어 내는 요약·초안·분석 결과가 마크다운(Markdown) 파일로 개인 PC와 여러 폴더에 쌓이고, 고객 정보나 문의 기록은 CRM·게시판 같은 외부 시스템에 그대로 남기 때문입니다. 그 결과 문서가 늘어나는 속도에 비해 AI가 근거로 쓸 수 있는 데이터는 좀처럼 늘지 않습니다.

1.2.1 AI 작업 산출물로 늘어나는 흩어진 마크다운 파일과 정리 비용

마크다운 파일 분산이 만드는 정리 비용 마크다운은 AI 도구가 결과를 내보낼 때 가장 많이 쓰는 형식이어서, AI와 함께 일하는 직원일수록 하루에도 여러 개의 마크다운 파일을 만듭니다. 하지만 이 파일들은 작성자의 PC나

개인 폴더에 놓이는 경우가 많아 같은 주제를 다룬 파일이 여러 곳에 중복되고, 어느 파일이 최신판인지 작성자 본인 말고는 알기 어렵습니다. 팀이 함께 쓰려면 누군가 파일을 모아 이름을 정리하고 분류하는 작업을 따로 해야 하며, 그 작업은 대개 우선순위에서 밀려 미뤄집니다.

데이터베이스 안에 쌓여야 가능한 검색·필터·정규화 문서를 한곳에 모으는 것만으로는 충분하지 않습니다. 문서가 데이터베이스 안의 행으로 저장되고 작성자·고객사·상태 같은 속성을 가져야 속성으로 필터링하고, 표기가 다른 값을 하나로 맞추고, 조건을 걸어 검색할 수 있습니다. 예를 들어 「지난 분기에 특정 고객사와 관련해 작성된 회의록」을 찾으려면 문서마다 날짜와 고객사 속성이 붙어 있어야 합니다. 문서 저장 위치를 하나로 모을지 판단할 때는 폴더 통합에서 그치지 않고 문서를 속성을 가진 데이터베이스 행으로 다룰 수 있는지를 함께 살펴야 합니다.

1.2.2 외부 시스템에 남은 데이터에 닿지 않는 키워드 검색과 의미 검색

외부 시스템에 고립된 업무 데이터 CRM의 영업 기록, 게시판의 민원 문의, 결재 시스템의 승인 이력은 조직에 가장 중요한 업무 데이터이지만 각 시스템의 데이터베이스 안에 머물러 있습니다. 이 데이터는 위키의 키워드 검색이나 의미 기반 벡터 검색이 닿지 않으며, 다른 문서와 공통 속성으로 이어 관계를 따라가는 질의도 할 수 없습니다. 이 상태에서 시스템마다 AI 기능을 따로 붙이면 각 시스템 안에서만 답하는 AI가 여러 개 생기고, 시스템을 넘나드는 질문에는 어느 AI도 답하지 못합니다.

데이터를 위키 안으로 모으는 연결 수단 이 문제를 푸는 방향은 AI를 시스템마다 붙이는 대신 데이터를 하나의 지식 저장소로 끌어오는 것입니다. 게시판 글이나 CRM 레코드가 위키의 데이터베이스 행으로 들어오면 저장하는 순간 색인이 만들어지고, 그때부터 기존 문서와 같은 방식으로 검색되고 관계로 이어집니다. 다만 시스템마다 데이터 구조와 인증 방식이 다르므로 이를 연결하는 수단, 즉 플러그인이 필요합니다. MindWiki가 Go·Node.js·Python·Java 네 가지 언어로 플러그인 SDK를 제공하는 이유가 여기에 있습니다(7장).

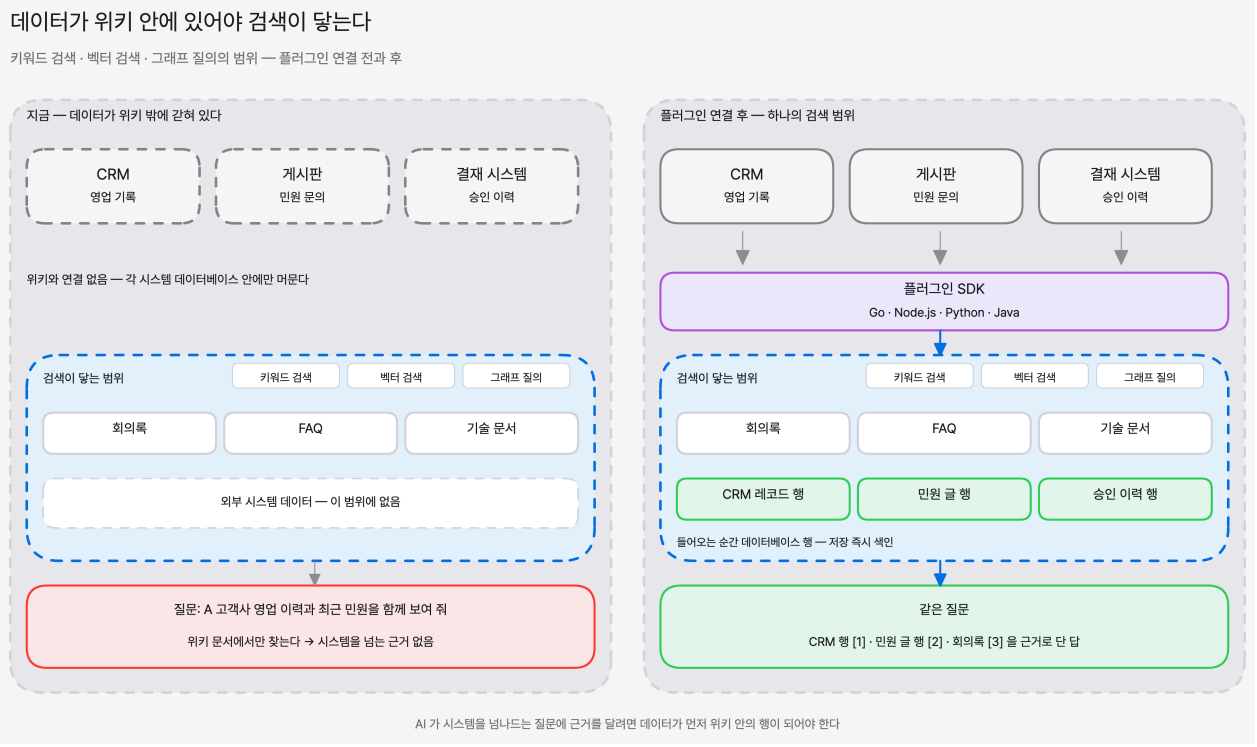

그림. 플러그인 연결 전후의 검색 범위

1.3 RAG 구축과 그래프 관계 설정이 개발자 몫으로 남는 기존 방식의 한계

데이터를 한곳에 모으더라도 그 데이터를 AI가 잘 찾도록 만드는 작업이 개발자에게만 맡겨지면 현업은 여전히 지식 저장소를 직접 다룰 수 없습니다. 기존 RAG 구축 방식은 문서를 잘라 벡터로 바꾸는 단계에 집중하므로 문서 사이의 관계를 묻는 질문에 약하고, 관계를 보강하려고 그래프 데이터베이스를 더하면 그 설정까지 개발 조직이 맡게 됩니다. 결과적으로 지식의 구조를 가장 잘 아는 현업과 그 구조를 시스템에 넣는 개발자가 분리되는 문제가 남습니다.

1.3.1 흩어진 정보를 공통 속성으로 이어야 풀리는 질문에 약한 일반 RAG

일반 RAG의 두 가지 약점 Microsoft Research는 2024년 2월 GraphRAG를 소개하면서 일반 RAG가 약한 질문을 두 종류로 정리했습니다. 하나는 여러 문서에 흩어진 정보를 공통 속성으로 이어야 답이 나오는 질문이고, 다른 하나는 큰 문서 묶음 전체를 이해해야 답할 수 있는 질문입니다[S5]. 일반 RAG는 질문과 의미가 비슷한 문서 조각을 골라 모델에 넘기는 방식이므로, 질문에 직접 등장하지 않는 연결 고리를 따라가야 하는 경우에는 필요한 조각을 찾지 못합니다. 이에 대한 대안으로 비공개 데이터에서 지식 그래프를 만들어 검색에 함께 쓰는 접근이 제안되었습니다[S6].

관계 질문이 필요한 업무의 예 「A 고객사에 도입된 제품과 같은 설치 환경을 쓰는 다른 고객사에서 최근 접수된 기술지원 건은 무엇인가」 같은 질문이 대표적인 관계 질문입니다. 이 질문에 답하려면 고객사, 도입 제품, 설치 환경, 기술지원 건을 속성으로 이어 따라가야 하며, 문장의 의미 유사도만으로는 연결 고리를 찾을 수 없습니다. 영업·구축·기술지원처럼 고객과 제품을 중심으로 정보가 이어지는 업무라면 벡터 검색 하나로는 부족하고 관계를 다루는 그래프 질의가 함께 있어야 합니다.

1.3.2 현업이 직접 채우지 못하는 RAG 데이터베이스와 그래프 DB

개발 조직에 묶이는 구축과 유지 기존 방식에서 RAG용 데이터베이스는 개발자가 수집·변환 코드를 작성해야 채워지고, 그래프 데이터베이스의 노드와 관계를 정의하는 일도 개발자가 맡습니다. 현업 담당자는 어떤 정보가 어떤 관계로 이어져야 하는지 가장 잘 알지만 그 구조를 직접 바꿀 방법이 없으므로, 새 속성 하나를 더하려 해도 개발 요청을 내고 기다려야 합니다. LLM으로 문서에서 관계를 자동 추출하는 방식도 모델 호출 비용이 크고 업무상 의미를 모르는 채로 관계를 뽑기 때문에, 결과를 다시 사람이 검토해야 하는 부담이 생깁니다.

현업 주도로 운영 가능한 지식 구조 이 구조에서는 지식 저장소가 업무 변화를 따라가지 못하고, 개발 인력이 줄면 저장소 갱신도 함께 멈춥니다. 반면 현업이 데이터베이스의 속성을 직접 정하고 그 속성이 그대로 그래프의 관계가 되면, 업무 담당자가 속성을 고치는 것만으로 지식 구조가 바뀝니다. 도구를 고를 때는 검색 성능과 함께 지식 구조를 누가 바꿀 수 있는지, 즉 현업 주도 운영이 가능한지를 확인해야 합니다. 업무 속성이 지식 그래프가 되는 방식은 4장에서 구체적으로 다룹니다.

1.4 근거 있는 AI 답변을 위한 지식 위키의 다섯 가지 조건

앞에서 살펴본 문제를 거꾸로 읽으면 근거 있는 AI 답변을 위한 지식 위키가 갖춰야 할 조건이 나옵니다. MindWiki는 기존 위키 도구를 시험하면서 회사 업무에 그대로 쓸 만한 설치형 위키를 찾지 못한 데서 출발했고, 그 과정에서 정리한 조건은 다음 다섯 가지입니다. 각 조건은 이후 장에서 MindWiki가 그 조건을 어떻게 구현하는지 확인하는 평가 기준이기도 합니다.

조건	점검 질문	관련 장
회사 안 설치	문서·색인·계정·로그가 모두 회사 인프라 안에 있는가	2장·6장
사내 AI 모델 사용	답변 생성·임베딩·재순위 모델이 외부로 데이터를 보내지 않는가	6장
저장 즉시 색인	문서를 저장하는 순간 키워드·벡터·그래프 색인이 함께 갱신되는가	3장
현업이 정하는 관계	개발자 없이 업무 속성으로 관계를 정의하고 질의할 수 있는가	4장
외부 데이터 연결	사내 시스템과 한글(HWP) 문서를 위키 안으로 끌어올 수 있는가	7장

1.4.1 회사 안 설치와 사내 AI 모델 사용이라는 조건

공급사 클라우드가 남기는 데이터 위치 문제 SaaS형 협업 도구는 문서를 공급사 클라우드에 저장하고, AI 기능도 공급사가 계약한 외부 모델로 처리하는 경우가 많습니다. 이 구조에서는 문서 원문뿐 아니라 검색 색인, 사용자 계정 정보, 접근 로그, AI에 보낸 질문과 답변이 어디에 저장되고 얼마 동안 남는지를 조직이 직접 통제하기 어렵습니다. 공공기관과 규제 산업에서는 이 점이 도입 여부를 좌우하며, 국가정보원 국가인공지능안보센터는 2025년 12월 공공기관에 「AI 보안 가이드북」을 배포했습니다[S7]. 개인정보보호위원회도 2025년 8월 「생성형 AI 개발·활용을 위한 개인정보 처리 안내서」를 공개해 생성형 AI 활용 방식별 개인정보 처리 기준을 제시했습니다[S8].

설치형 위키와 사내 모델이 지키는 데이터 주권 따라서 근거 있는 AI 답변을 위한 지식 위키의 첫 두 조건은 위키를 회사 인프라 안에 설치하고, 답변 생성에 쓰는 AI 모델도 회사 안에서 운영하는 것입니다. 이 구성에서는 문서·색인·로그·AI 처리 경로가 모두 회사 안에 머물러 데이터 주권을 조직이 직접 지킬 수 있습니다. MindWiki는 쿠버네티스 플랫폼인 MSAP.ai 위에 설치되고, CogentAI 사내 모델로 답변 생성·임베딩·재순위·가드레일을 처리합니다(6장). 설치형 도입을 검토할 때는 문서 원문과 함께 계정·로그·AI 처리 경로까지 회사 안에 두는지를 기준으로 삼아야 합니다.

1.4.2 저장과 동시에 검색되는 지식과 외부 데이터 연결이라는 조건

저장·속성·연결이 각 조건에 대응하는 방식 나머지 세 조건은 지식이 만들어지는 방식에 관한 것입니다. 저장 즉시 색인은 문서를 저장하는 순간 키워드·벡터·그래프 색인이 함께 만들어져 별도 가공 없이 평소 문서가 AI 근거가 되는 상태를 말합니다(3장). 현업이 정하는 관계는 업무 담당자가 정한 속성이 그대로 그래프의 관계가 되어 개발자 없이도 관계 질문에 답할 수 있는 상태이고(4장), 외부 데이터 연결은 플러그인 SDK와 HWP 변환으로 사내 시스템과 한글 문서를 위키 안으로 끌어와 외부 시스템에 고립된 데이터까지 근거로 쓰는 상태입니다(7장).

다섯 조건이 함께 충족되어야 하는 이유 다섯 조건은 하나씩 떼어 놓으면 효과가 크게 줄어듭니다. 예를 들어 저장 즉시 색인이 되더라도 외부 모델로 처리하면 데이터 주권 요건을 만족하지 못하고, 회사 안에 설치하더라도 외부 시스템의 데이터를 끌어오지 못하면 AI가 답할 수 있는 범위가 위키 문서에 한정됩니다. MindWiki는 다섯 조건을 하나의 제품 안에서 함께 충족하도록 설계되었고, AAS 에이전트가 이 위키를 읽고 쓰며 사람의 승인을

거쳐 지식을 다시 쌓는 구조로 이어집니다(5장). 2장부터는 이 다섯 조건을 기준으로 MindWiki의 구성과 동작을 차례로 살펴봅니다.

2장: 사람과 AI가 함께 쓰는 설치형 지식 위키 MindWiki — 익숙한 협업 노트 사용법과 달라진 세 가지

MindWiki는 블록 편집, 문서 데이터베이스, 공동 편집처럼 협업 노트 도구에서 익숙한 사용법을 그대로 제공하면서 저장된 문서를 곧바로 AI의 검색 근거로 바꾸는 설치형 지식 위키입니다. 회사 안 설치, 사내 AI 모델, 저장 즉시 색인, 현업이 정하는 관계, 외부 데이터 연결이라는 1장의 다섯 가지 조건을 한 제품 안에서 함께 충족하도록 설계했습니다. 생성형 AI를 업무에 쓰려는 조직은 모델을 고르기 전에 근거 데이터를 어디에 어떻게 모을지부터 정해야 하고, 그 결정이 늦어질수록 여러 도구와 파일에 따로 쌓이는 문서도 늘어납니다. 따라서 MindWiki 도입은 새 화면을 처음부터 익히는 도구 교체와 성격이 다르며, 사용 경험은 유지한 채 데이터가 놓이는 위치와 AI가 근거를 얻는 구조를 바꾸는 선택입니다. 이 장에서는 MindWiki의 정의와 달라진 세 가지, 제품이 하는 네 가지 동작, 전체 구성, 먼저 효과가 나는 조직과 업무를 차례로 설명합니다.

2.1 익숙한 사용법은 유지하고 설치 위치와 AI 모델, 지식이 되는 방식을 바꾼 설계

MindWiki 설계는 사용자가 매일 보는 편집 화면은 협업 노트 도구의 익숙한 방식을 따르고, 사용자가 보지 않는 설치 위치와 AI 모델, 문서가 지식으로 바뀌는 과정은 새로 만든 구조입니다. 현업 담당자에게는 전환 교육 부담이 작고, IT 조직에게는 데이터 위치와 AI 처리 경로를 회사가 직접 통제하는 구조가 생깁니다. 다만 두 쪽의 변화 크기가 다르므로 도입을 검토할 때는 화면 비교보다 아래 세 가지 변화가 조직의 요구와 맞는지를 먼저 확인하는 편이 정확합니다.

2.1.1 블록 편집과 문서 데이터베이스, 공동 편집으로 이어지는 사용 경험

MindWiki의 편집 화면은 SaaS형 협업 노트 도구를 써 본 사용자가 별도 안내 없이 바로 쓸 수 있도록 같은 사용 방식을 따릅니다. 문단, 표, 이미지, PDF, 칼럼, 동기화 블록을 블록 단위로 끌어 옮기며 문서를 만들고, 여러 사람이 같은 페이지를 실시간으로 함께 고칩니다. 페이지 안에 문서 데이터베이스를 만들면 각 행이 하나의 문서가 되고, 행마다 글·선택·사람·날짜·URL 같은 속성을 둡니다. 같은 데이터베이스를 표, 칸반, 목록 등 여러 보기로 열 수 있고 수식·관계·롤업 속성으로 행끼리 계산하고 연결합니다. 여기에 공휴일을 빼고 처리 기한을 세는 근무일 함수처럼 한국 공공 업무에 필요한 기능을 더했습니다.

문서 데이터베이스 개념의 의미 MindWiki가 협업 노트 도구에서 가져온 핵심 개념은 화면 모양보다 「모든 문서는 데이터베이스의 행이다」라는 원칙입니다. AI와 함께 일하면 마크다운 파일이 개인 PC와 공유 폴더 곳곳에 쌓이는데, 파일 묶음은 속성으로 거르거나 항목을 정규화하기 어렵습니다. 반면 문서가 데이터베이스 행으로 쌓이면 담당자·상태·제품·고객사 같은 속성으로 찾고 거를 수 있고, 그 속성이 그대로 검색 색인과 지식 그래프의 재료가 됩니다(3장, 4장). 사용자가 새로 익혀야 할 것은 화면 조작보다 업무에 맞는 속성을 정하는 원칙이며, 교육 범위도 그 원칙에 맞춰 좁힐 수 있습니다.

2.1.2 공급사 클라우드와 외부 AI 모델 대신 회사 안 설치와 사내 모델

SaaS형 협업 노트 도구와 엔터프라이즈 협업 위키는 대개 공급사 클라우드에서 동작하고, AI 기능은 공급사가 계약한 외부 모델이나 사용량에 따라 비용이 붙는 부가기능으로 제공합니다. 엔터프라이즈 협업 위키 범주에서는 자체 설치판을 줄이고 클라우드판 중심으로 옮겨 가는 흐름을 공개 안내에서 볼 수 있고, 클라우드판에서 데이터 저장 지역을 지정해도 계정 정보와 일부 로그는 지정 대상에서 빠지는 경우가 있습니다. 즉 문서 본문을 국내 지역에 두더라도 AI가 질문을 처리하는 경로와 운영 기록까지 회사 안에 있는지는 별도로 따져야 합니다.

MindWiki는 이 두 조건을 설치 단계에서 해결하도록 회사의 쿠버네티스 플랫폼인 MSAP.ai 위에 설치하고, 인터넷 연결 없이 동작하며, 사내에 함께 설치하는 CogentAI 모델로 AI 기능을 처리합니다.

세 번째 변화는 문서가 지식이 되는 방식입니다. 일반적인 문서 도구는 문서를 저장하고 전문 검색을 제공하는 데서 멈추므로, 문서를 AI 근거로 쓰려면 별도 RAG(Retrieval-Augmented Generation) 구축 프로젝트가 필요합니다. 반면 MindWiki는 저장하는 순간 키워드 색인, 벡터 색인, 그래프 색인을 함께 갱신하므로 문서 작성이 곧 AI 근거 축적이 됩니다. 아래 표는 세 가지 변화를 익숙한 범주의 일반적인 방식과 나란히 놓은 것입니다.

구분	익숙한 협업 노트·위키 범주의 일반적인 방식	MindWiki
설치 위치	공급사 클라우드 중심, 지역 지정은 데이터 일부에 적용	MSAP.ai 위 회사 안 설치, 인터넷 없이 동작
AI 모델	공급사가 계약한 외부 모델, 사용량 기반 부가기능	사내 CogentAI 모델(LLM·임베딩·재순위·가드레일·음성)
문서가 지식이 되는 방식	저장과 전문 검색 중심, AI 근거는 별도 구축	저장하는 순간 키워드·벡터·그래프 색인 자동 갱신

2.2 MindWiki 네 가지 동작 — 문서 작성, 근거 검색, 에이전트 위임, 사내 시스템 연결

MindWiki가 하는 일은 사람이 쓰고, AI가 근거를 달아 찾고, 에이전트에게 일을 맡기고, 사내 시스템을 위키에 잇는 네 가지 동작으로 요약됩니다. 네 동작은 모두 같은 PostgreSQL 저장소의 같은 문서와 행을 대상으로 하므로, 한 동작의 결과가 곧바로 다른 동작의 입력이 됩니다. 예를 들어 플러그인이 가져온 게시판 글은 저장과 동시에 검색 대상이 되고, 에이전트는 그 글을 근거로 답변 초안을 써서 다시 위키 행에 남깁니다.

2.2.1 사람이 쓰고 AI가 근거를 달아 찾는 문서 작성과 근거 검색

문서 작성과 근거 검색이 한 저장소에서 일어난다는 점이 MindWiki 검색 품질의 출발점입니다. 사용자가 페이지 본문을 고치거나 데이터베이스 행의 속성을 바꾸면 그 내용이 수 초 안에 키워드·벡터 색인에, 1초 안에 그래프 색인에 반영됩니다(설계값, 3장). 담당자가 방금 정리한 FAQ 항목이나 기술지원 해결 기록이 별도 로딩 작업 없이 다음 질문의 근거가 됩니다. 질문을 받으면 MindWiki는 키워드 검색과 의미 검색 결과를 순위 기준으로 합치는 RRF(Reciprocal Rank Fusion, 역순위 융합) 방식으로 후보를 고르고, 재순위 모델로 다시 정렬한 뒤 출처 번호를 단 답을 만듭니다[S1].

MindWiki는 검색 범위를 질문한 사람의 권한 안으로 한정합니다. 권한 밖 문서는 순위를 매기기 전에 데이터베이스 질의 단계에서 제외하므로, 볼 수 없는 문서가 답변 근거로 섞여 들어가지 않습니다. 근거 문서가 하나도 없는 질문에는 언어 모델을 부르지 않아 근거 없는 답과 불필요한 모델 호출을 함께 줄입니다. 사내 평가 기준 검색 품질은 MRR 0.815, 재현율@10 0.810이며, 같은 수치가 조직의 문서 규모와 정돈 상태에 따라 달라진다는 점도 함께 고려해야 합니다.

사용자가 이 검색을 만나는 입구는 두 가지입니다. 화면 어디서나 Ctrl/Cmd+K로 여는 빠른 검색은 키워드와 의미 검색을 합치고 재순위를 거친 결과를 보여 주며, 자주 찾는 페이지를 먼저 띄웁니다. 「AI에게 묻기」는 같은 검색 결과 가운데 질문한 사람이 볼 수 있는 문서만 근거로 답을 쓰고 각 문장을 출처 블록으로 이어 줍니다. 답

을 읽은 뒤 이어서 물을 수 있고 답이 쓸모 있었는지 평가를 남길 수 있어, 근거가 부족했던 질문을 찾아 문서를 보강할 때 실마리가 됩니다.

페이지 AI 버튼으로 하는 작성 보조 모든 페이지에는 AI 버튼이 있어 길게 쓰기와 짧게 쓰기, 키워드로 본문 채우기, 목차 만들기와 목차별 채우기, 문장 다시 쓰기, 번역, 읽어 주기와 받아쓰기를 제공합니다. 이 기능도 사내 CogentAI 모델로 처리하므로 작성 중인 초안이 회사 밖 모델로 전송되지 않습니다.

2.2.2 위키 밖 업무를 위키 안으로 잇는 에이전트 위임과 사내 시스템 연결

에이전트 위임은 MindWiki와 AAS(AI Agent Studio) 에이전트를 MCP(Model Context Protocol)로 연결해 처리합니다. MCP는 LLM 애플리케이션이 외부 데이터와 도구를 표준 방식으로 쓰도록 정한 공개 프로토콜이며, 사용자가 데이터 접근과 도구 실행에 명시적으로 동의하고 통제해야 한다는 보안 원칙을 담고 있습니다 [S4]. MindWiki는 이 원칙에 맞춰 에이전트가 로그인한 사람의 권한 안에서만 읽고 쓰게 하고, 지우기 도구는 등록하지 않으며, 에이전트의 쓰기는 편집 제안으로 남겨 사람이 승인한 뒤에 반영합니다. 데이터베이스 행의 조건이나 버튼이 AAS 에이전트에게 일을 넘기고 결과가 다시 행에 기록되는 흐름은 5장에서 구체적으로 다룹니다.

사내 시스템 연결은 플러그인 SDK가 맡습니다. 플러그인 SDK는 Go, Node.js, Python, Java 네 가지 언어를 지원하며, 사내 게시판 글 가져오기, 사내 메신저 알림, 외부 자료 스크랩, 결재 시스템 승인 호출처럼 고객사마다 다른 시스템을 같은 방식으로 연결합니다. 한글(HWP·HWPX)과 DOCX 첨부는 원본 보기와 함께 하위 페이지 본문으로 변환되어 자동 색인되고, 회의 녹음은 화자를 나눈 받아쓰기와 회의록으로 정리됩니다(7장). 따라서 외부 시스템에 남아 있던 데이터도 위키 안으로 들어오는 순간 나머지 세 동작의 대상이 됩니다.

2.3 MindWiki 전체 구성 — 위키 본체와 사내 AI 모델, MSAP.ai 실행 기반

MindWiki 설치 구성은 위키 본체, 사내 AI 모델 묶음, 두 계층을 실행하는 MSAP.ai 플랫폼의 세 층으로 나뉩니다. 위키 본체는 API 서버, 공동 편집을 처리하는 동기화 서버, 검색 색인 서버, 첨부 문서 변환 서버, 데이터 저장소로 이뤄집니다. 모든 구성 요소가 회사의 쿠버네티스 클러스터 안에서 컨테이너로 동작하고 화면도 외부 CDN이나 웹 글꼴 없이 표시되므로, 인터넷이 차단된 망에서도 같은 구성으로 설치할 수 있습니다.

2.3.1 LLM과 임베딩, 재순위, 가드레일, 음성 모델로 이뤄진 사내 AI 모델 묶음

MindWiki가 쓰는 AI 모델은 모두 CogentAI 모델로 회사 안에서 동작하며, 역할에 따라 다섯 종류로 나뉩니다.

- LLM: 대화, 요약, 문서 작성, 답변 초안 생성
- 임베딩 모델: 문서 블록과 질문을 1024차원 벡터로 바꿔 의미 검색에 사용
- 재순위 모델: 키워드·벡터 검색으로 모은 후보를 질문과의 관련도로 다시 정렬
- 가드레일 모델: 질문을 먼저 검사해 대량 개인정보 추출 같은 위험한 요청 차단
- 음성 모델: 회의 녹음 받아쓰기와 페이지 읽어 주기에 쓰는 음성 인식·합성

가드레일은 업무 환경에 맞춰 조정했습니다. 예를 들어 「회원 정보를 전부 뽑아 달라」는 질문이나 식별 가능한 개인정보가 한꺼번에 많이 담긴 입력은 막고, 명함 한 장을 입력하는 수준은 허용합니다. 운영 시스템 이름에 민감한 단어가 들어 있어 일반 가드레일이 오탐하던 질문을 정상 업무 질문으로 판단하도록 추가 학습했습니다. 모델 주소는 설정값으로 지정하는 구조이므로, 필요한 GPU와 운영 자원은 조직의 문서 규모와 동시 사용자 수에 따라 도입 협의에서 정합니다.

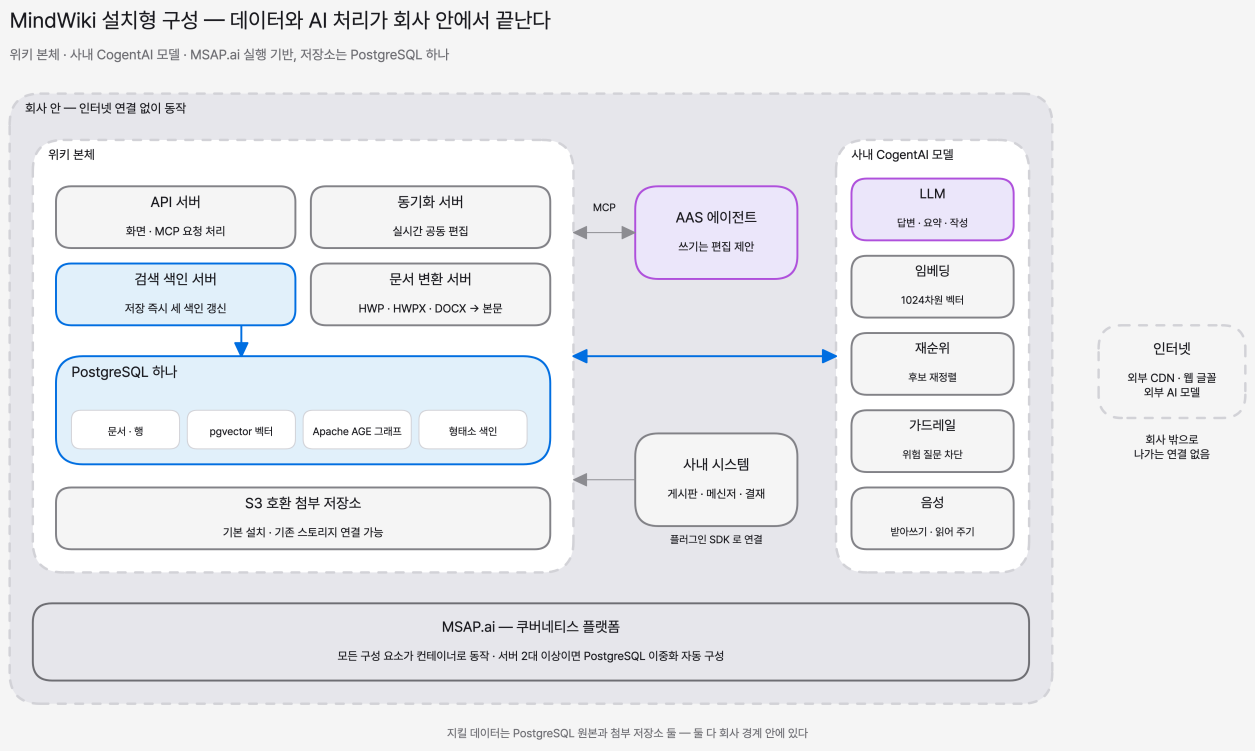

그림. MindWiki 설치형 구성도

2.3.2 PostgreSQL 하나에 문서와 벡터, 그래프를 담는 저장 구조

MindWiki는 문서 본문, 데이터베이스 행, 벡터 색인, 그래프, 한국어 형태소 색인을 모두 PostgreSQL 하나에 담습니다. 벡터 검색은 PostgreSQL에 벡터 유사도 검색을 더하는 오픈소스 확장 pgvector의 HNSW 인덱스를 쓰며, HNSW는 다층 그래프 구조로 속도 대비 재현율이 좋은 대신 인덱스 생성에 시간과 메모리가 더 듭니다[S2]. 그래프는 기존 관계형 데이터베이스 안에서 Cypher와 비슷한 그래프 질의를 쓰게 해 주는 Apache AGE 확장으로 저장합니다[S3]. 이 구성은 문서 데이터베이스, 벡터 데이터베이스, 그래프 데이터베이스를 따로 운영할 때 생기는 동기화 작업과 데이터 불일치를 처음부터 줄입니다.

운영 측면에서는 관리 대상 데이터베이스가 하나라는 점이 가장 큰 차이입니다. 백업과 복구, 이중화, 접근 권한을 한 체계로 관리하고, 서버를 두 대 이상 쓰면 설치기가 PostgreSQL 이중화를 자동으로 구성합니다. 첨부 파일은 S3 호환 저장소에 두는데, 기본 설치 때 함께 깔리며 필요하면 고객이 이미 운영하는 스토리지를 S3 방식으로 연결해 쓸 수 있습니다. 세 색인은 원본 문서와 행에서 언제든 다시 만들 수 있는 파생 데이터이므로, 복구 계획은 PostgreSQL 원본과 첨부 저장소 두 가지를 중심으로 세우면 됩니다.

2.4 MindWiki가 맞는 조직과 업무 — 근거 문서가 쌓이는 답변 업무

MindWiki 효과가 먼저 나타나는 업무는 질문이 반복해서 들어오고, 답을 낼 때마다 근거 문서를 찾아 인용해야 하며, 처리 결과가 다시 다음 답의 근거로 쌓이는 업무입니다. 조직 유형으로는 외부 클라우드와 외부 AI 모델 사용에 제약이 있는 공공기관과 규제 산업이 먼저 해당합니다. 여러분의 조직에 두 조건이 함께 있다면 MindWiki 도입 효과를 작은 파일럿으로 빠르게 확인할 수 있습니다.

2.4.1 민원 답변과 기술지원, 회의록처럼 근거 문서가 쌓이는 업무

파일럿 후보로 가장 먼저 꼽을 업무는 민원 답변, 기술지원, 회의록 세 가지입니다. 세 업무 모두 근거가 되는 문서가 이미 조직 안에 있지만 여러 시스템과 파일로 흩어져 있고, 담당자가 그 문서를 찾아 읽고 정리하는 데 처리 시간 대부분을 씁니다.

- 민원 답변: 질문 유형별 FAQ와 법령 조문을 데이터베이스로 정리해 두면 에이전트가 이를 근거로 답변 초안을 만들고 담당자는 검토와 승인에 집중합니다(9장).
- 기술지원: 고객사, 도입 제품, 버전, 설치 환경을 속성으로 관리하면 문의가 들어올 때 고객 환경과 과거 해결 기록이 함께 표시됩니다(8장).
- 회의록: 녹음을 받아쓰기와 회의록, 할 일 카드로 정리하면 회의 결정 사항이 다음 업무의 검색 근거로 남습니다(7장).

세 업무 가운데 하나를 고르면 근거 데이터의 범위와 성공 기준을 좁게 정할 수 있어 효과를 측정하기 쉽습니다. 다만 근거 문서가 정돈되지 않은 상태에서는 검색과 답변 품질도 그만큼 낮아지므로, 파일럿 첫 단계에서 FAQ나 해결 기록을 질문 유형별로 정리하는 작업을 함께 계획해야 합니다(12장).

2.4.2 외부 클라우드 사용이 어려운 공공기관과 규제 산업의 요구

공공기관과 규제 산업에서는 생성형 AI 활용 기준이 구체적인 지침으로 나오고 있습니다. 국가정보원 국가인공지능안보센터는 2025년 12월 기존 생성형 AI 보안 가이드라인을 개정한 「AI 보안 가이드북」을 중앙행정기관, 지자체, 공공기관에 배포했습니다[S7]. 개인정보보호위원회는 2025년 8월 6일 「생성형 AI 개발·활용을 위한 개인정보 처리 안내서」를 공개해 생성형 AI의 개발과 활용 단계별 개인정보 처리 기준을 제시했습니다[S8]. 이런 지침 아래에서는 업무 문서가 어느 서버에 저장되는지뿐 아니라 질문과 답변을 처리하는 AI 모델이 어디에서 동작하는지도 점검 대상이 됩니다.

MindWiki는 문서 저장, 색인, AI 처리, 에이전트 실행을 모두 MSAP.ai 위 회사 안에서 처리하므로 데이터 주권 요구를 설치 구조로 충족합니다. 민감한 속성은 칼럼 단위로 암호화해 검색 색인과 그래프에서 제외하는 암호화 속성, 페이지 트리 권한과 행 접근 규칙, 질문 단계의 가드레일을 함께 제공합니다(6장). 따라서 외부 SaaS 도입 심사에서 번번이 멈추던 조직도 기존 망 분리 정책과 보안 점검 절차 안에서 AI 지식 위키를 검토할 수 있습니다.

3장: 저장하는 순간 만들어지는 키워드·벡터·그래프 색인과 근거를 단 하이브리드 검색

MindWiki의 검색은 세 가지 색인으로 동작하며, 키워드 색인은 이름을, 벡터 색인은 뜻을, 그래프 색인은 관계를 찾습니다. 일반적인 사내 RAG(Retrieval-Augmented Generation) 구축에서는 문서를 따로 내보내고, 청킹과 임베딩 작업을 거쳐 벡터 DB에 로드하는 파이프라인을 개발 조직이 만들어 관리해야 하므로, 문서가 바뀔 때마다 색인이 원본보다 뒤처지고, 담당자가 없으면 색인이 방치되기 쉽습니다. MindWiki는 사용자가 문서나 데이터베이스 행을 저장하는 순간 세 색인을 함께 다시 만들기 때문에 별도 구축 없이도 AI가 검색하는 근거가 생깁니다. 아래 다섯 절은 원본과 색인의 관계, 색인별 기술 구성, 반영 시간, 근거를 단 답변까지의 검색 파이프라인, 사내 평가 세트 기준 품질 수치를 차례로 설명합니다.

3.1 원본 두 가지와 자동으로 다시 만드는 세 가지 색인 — 본문과 행 속성의 투영

MindWiki가 원본으로 보관하는 데이터는 페이지 본문과 데이터베이스 행 속성 두 가지뿐입니다. 키워드 색인, 벡터 색인, 지식 그래프는 모두 이 두 원본에서 계산해 낸 결과물이므로, 색인을 따로 편집하거나 백업 대상으로 관리할 필요가 없습니다. 운영자가 지켜야 할 데이터는 원본이고, 색인은 원본을 다른 방향에서 비춘 투영(projection)입니다.

3.1.1 문단과 표, 링크, 멘션 본문과 행 속성이라는 두 원본

첫 번째 원본인 페이지 본문은 문단·표·링크·멘션 같은 블록으로 이루어지며, MindWiki는 블록 하나를 색인과 검색 결과의 기본 단위로 씁니다. 두 번째 원본인 데이터베이스 행 속성은 업무 데이터를 담는 칸으로, 기술지원 케이스 데이터베이스라면 고객사, 제품, 버전, 담당자, 처리 상태가 각 행의 속성이 되고, 속성 형식은 글·선택·사람·URL 등으로 나뉩니다. MindWiki는 행 속성도 행마다 한 블록으로 묶어 색인합니다. 따라서 「A 고객사의 3.2 버전 케이스」처럼 본문에는 적혀 있지 않고 속성 칸에만 들어 있는 정보도 본문과 같은 방식으로 찾을 수 있습니다. 문서 위키와 업무 데이터베이스를 따로 운영하던 조직에서는 표 형태의 업무 데이터가 AI 검색에서 빠지기 쉬운데, MindWiki는 두 원본을 같은 색인 체계에 넣어 이 누락을 막습니다.

3.1.2 원본에서 다시 만드는 투영으로서의 키워드·벡터·그래프 색인

세 색인이 원본의 투영이라는 설계는 운영 부담을 크게 줄입니다. 색인이 손상되거나 임베딩 모델을 바꿔야 할 때 원본에서 색인을 처음부터 다시 만들면 되므로, 운영자는 색인 자체를 복구하는 절차를 따로 준비할 필요가 없습니다. 원본과 색인이 어긋나는 경우가 생겨도 기준은 언제나 원본입니다. 반면 별도 벡터 DB에 문서를 복제해 두는 구성에서는 원본 시스템과 벡터 DB가 서로 다른 저장소에 있어, 동기화 작업이 실패하면 어느 쪽이 맞는지 사람이 가려야 합니다.

원본에서 색인을 다시 만드는 일이 평소 부담이 되지 않는 이유는 처리 단위가 블록이기 때문입니다. 편집이 멈추면 MindWiki는 문서를 블록 단위로 펼쳐 저장하고, 그중 바뀐 블록만 골라 키워드·벡터·그래프 색인에 다시 반영합니다. 수백 쪽짜리 규정집에서 조항 하나를 고쳐도 문서 전체를 다시 임베딩하지 않으므로, 문서가 하루에도 여러 번 바뀌는 위키에서도 색인 비용은 전체 문서량이 아니라 실제 편집량에 맞춰 작게 유지됩니다. 색인 전체가 어긋났을 때만 원본에서 처음부터 다시 만드는 명령을 쓰면 됩니다.

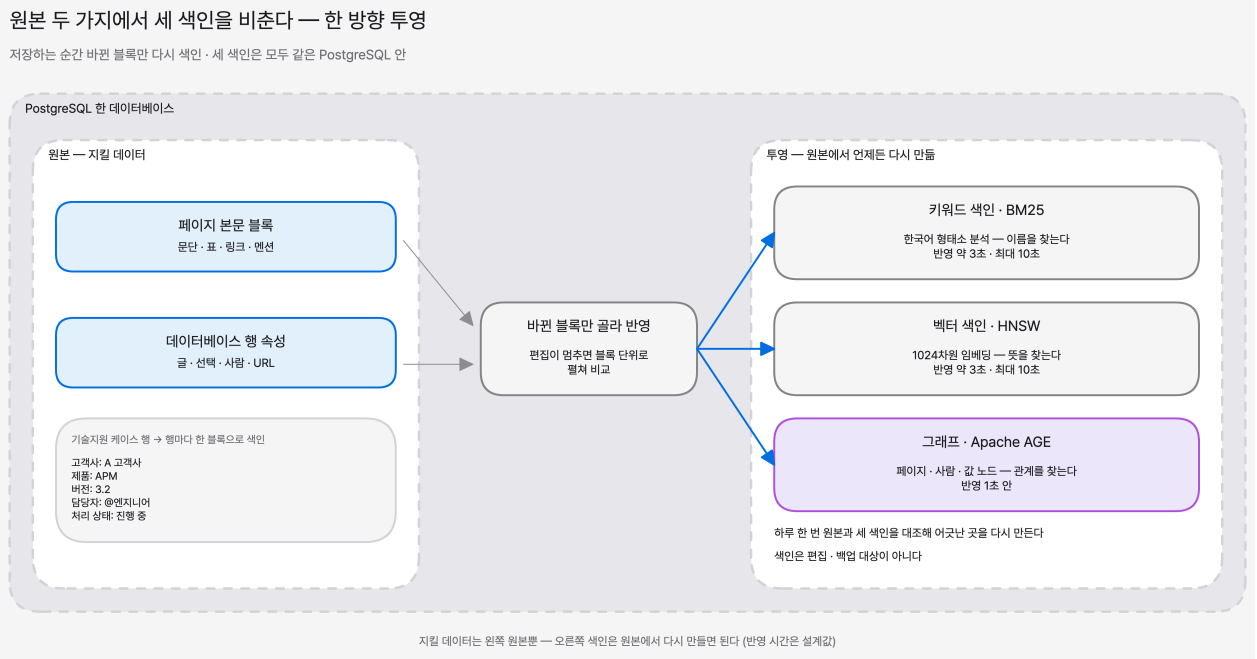

그림. 원본 두 가지에서 만드는 세 색인

3.2 한국어 형태소 BM25 키워드 색인과 1024차원 HNSW 벡터 색인

키워드 색인과 벡터 색인은 서로 다른 질문에 강합니다. 제품명·오류 코드·고객사 이름처럼 정확한 단어를 찾을 때는 키워드 색인이 빠르고 확실하며, 표현은 달라도 뜻이 같은 문서를 찾을 때는 벡터 색인이 필요합니다. MindWiki는 이 차이를 살려 두 색인을 모두 PostgreSQL 안에 만들고 검색할 때 함께 씁니다.

3.2.1 이름을 찾는 BM25 키워드 색인과 한국어 형태소 분석

BM25는 검색어가 문서에 얼마나 자주, 얼마나 드물게 나타나는지로 관련도를 매기는 키워드 순위 방식입니다. 다만 한국어는 명사 뒤에 조사와 어미가 붙기 때문에 띄어쓰기 단위로 자르면 같은 단어가 여러 형태로 나뉘어 서로 다른 단어처럼 보입니다. 예를 들어 「인증서가 만료되기 전에」라는 문장과 「인증서를 갱신하는 절차」라는 문장은 어절 단위로 보면 공통 단어가 없습니다. 형태소 분석을 거치면 두 문장 모두에서 「인증서」라는 명사를 뽑아내므로, 「인증서 만료」로 검색했을 때 두 문서가 함께 후보에 오릅니다. MindWiki는 이 한국어 형태소 분석을 키워드 색인에 기본으로 적용합니다. 한국어 업무 문서를 다루는 여러분의 조직에서는 이 분석 덕분에 검색어를 조사까지 맞춰 입력하지 않아도 원하는 문서를 찾을 수 있습니다.

키워드 색인은 형태소 분석과 별개로, 글자가 정확히 맞아야 하는 질문에서 벡터 색인이 놓치는 자리를 메웁니다. 임베딩은 문장 전체의 뜻을 벡터 하나로 압축하므로 「A-310 모델의 설치 조건」을 물었을 때 번호만 다르고 나머지 문장이 거의 같은 A-301이나 A-320 설명서가 더 가깝게 잡힐 수 있습니다. 「제17조 2항」과 「제17조 3항」처럼 같은 장 안에서 문체가 비슷한 조항도 뜻만으로는 순위가 뒤섞이기 쉽습니다. BM25는 「A-310」이라는 문자열 자체를 검색어로 다루므로 그 문자열이 든 문서를 정확히 위로 올립니다. 제품 코드·조항 번호·계약 번호로 묻는 일이 많은 구매·법무·기술지원 업무에서 키워드 색인을 벡터 색인과 나란히 두는 이유가 여기에 있습니다.

3.2.2 뜻을 찾는 블록 단위 청킹과 1024차원 HNSW 벡터 색인

벡터 색인은 블록 하나하나를 1024차원 임베딩으로 바꿔 저장합니다. 문서를 일정한 글자 수로 자르는 방식은 표나 목록 중간을 끊어 의미를 흐트러뜨리기 쉽지만, MindWiki는 사용자가 작성한 블록 경계를 그대로 청킹 단위로 쓰기 때문에 문단과 표가 온전한 의미 단위로 임베딩됩니다. 임베딩은 사내에서 동작하는 CogentAI 임베딩 모델이 만들고, 저장과 검색은 PostgreSQL 확장인 pgvector가 맡습니다. pgvector는 HNSW와 IVFFlat 두 가지 인덱스를 제공하며, HNSW는 다층 그래프 구조로 속도 대비 재현율이 더 좋은 대신 인덱스 생성 시간과 메모리를 더 씁니다[S2]. 반면 IVFFlat은 데이터를 목록으로 나누는 방식이라 생성은 빠르지만 검색 품질이 데이터 분포에 더 민감합니다[S2]. MindWiki는 문서가 계속 추가되는 위키 환경에서 검색 품질을 일정하게 유지하기 위해 HNSW를 씁니다. 자원 계획을 세울 때는 블록 수가 늘어나는 만큼 데이터베이스 메모리를 함께 늘리는 방향으로 잡아야 하며, 필요한 규모는 문서량과 환경에 따라 달라집니다.

3.3 PostgreSQL 안의 Apache AGE 그래프와 저장 후 색인 반영 시간

세 번째 색인인 지식 그래프는 페이지와 사람, 속성 값 사이의 관계를 담습니다. 키워드와 벡터는 문서 하나의 내용을 기준으로 찾지만, 그래프는 「같은 제품을 쓰는 다른 고객사」처럼 여러 문서에 흩어진 정보를 이어서 찾습니다. 이 그래프도 별도 그래프 DB 없이 같은 PostgreSQL 안에서 동작합니다.

3.3.1 관계를 찾는 페이지·사람·값 노드와 링크·멘션·속성 엣지

MindWiki 그래프의 노드는 페이지, 사람, 속성 값 세 종류입니다. 엣지는 페이지 사이의 링크, 본문에서 사람을 부르는 멘션, 그리고 행이 어떤 속성 값을 가졌는지를 나타내는 속성 연결로 만들어집니다. 예를 들어 기술지원 케이스 행의 「제품」 속성에 특정 제품이 들어 있으면 케이스 페이지와 제품 값 노드 사이에 엣지가 생깁니다. 이 그래프는 Apache AGE로 구현합니다. Apache AGE는 기존 관계형 데이터베이스 안에서 그래프 질의를 쓸 수 있게 하는 PostgreSQL 확장이며, Cypher와 비슷한 질의 언어를 제공하고 필요하면 SQL로도 그래프를 조회합니다[S3]. 그래프가 같은 데이터베이스 안에 있으므로 IT 담당자는 그래프 전용 DB를 새로 설치하고 백업·접근 통제·이중화를 따로 구성할 필요가 없습니다. 업무 속성이 어떻게 그래프 엣지가 되는지는 4장에서 자세히 설명합니다.

3.3.2 키워드·벡터 약 3초와 그래프 1초 반영, 하루 한 번 대조

저장한 내용이 검색에 나타나기까지 걸리는 시간은 설계값 기준으로 키워드와 벡터 색인이 약 3초, 최대 10초이고, 그래프는 계산량이 적어 1초 안에 반영됩니다. 벡터 색인이 키워드보다 오래 걸릴 수 있는 이유는 임베딩 모델 호출이 끼어 있기 때문이며, 모델 서버가 바쁘면 최대치에 가까워집니다. 네트워크 장애나 모델 서버 재시작처럼 일시적인 문제로 일부 블록의 색인이 누락될 수 있으므로, MindWiki는 하루 한 번 원본과 세 색인을 대조해 어긋난 부분을 다시 만듭니다. 회의록을 저장하고 곧바로 AI에게 그 내용을 물어도 대부분 몇 초 안에 근거로 잡히며, 드물게 빠진 블록도 다음 대조에서 바로잡힙니다.

3.4 근거를 단 답변까지의 하이브리드 검색 파이프라인 — 키워드 50건과 벡터 50건의 RRF 합산

하이브리드 검색은 키워드 검색과 벡터 검색의 결과를 하나의 순위로 합치는 방식입니다. 두 방식이 매기는 점수는 범위와 의미가 달라 그대로 더할 수 없기 때문에, 합치는 규칙을 정하는 일이 검색 품질을 좌우합니다. MindWiki는 이 문제를 풀기 위해 공개 검색 엔진들이 널리 쓰는 RRF를 합산 규칙으로 쓰고, 그 뒤에 재순위 모델을 한 번 더 거칩니다.

3.4.1 가드레일 검사부터 재순위 30건과 인용 답변까지 다섯 단계

MindWiki가 질문 하나를 처리하는 순서는 다섯 단계입니다.

1. 가드레일 검사: 사내 가드레일 모델이 대량 개인정보 추출 같은 위험한 질문을 먼저 걸러 냅니다.
2. 후보 수집: 키워드 색인에서 50건, 벡터 색인에서 50건의 후보 블록을 가져옵니다.
3. RRF 합산: 두 후보 목록을 순위 기준으로 합쳐 하나의 목록으로 만듭니다.
4. 재순위: 재순위 모델이 질문과 후보를 함께 읽고 상위 30건을 다시 고릅니다.
5. 인용 답변: LLM이 상위 근거로 답을 쓰고, 문장마다 근거 블록 번호를 답니다.

RRF(Reciprocal Rank Fusion, 역순위 융합)는 각 검색 방식이 매긴 순위의 역수 `1/(순위 + k)` 를 문서마다 더해 최종 순위를 정합니다[S1]. 점수 대신 순위를 쓰기 때문에 BM25 점수와 벡터 유사도처럼 범위가 다른 값을 억지로 맞출 필요가 없으며, k는 60 같은 작은 값에서 결과가 가장 좋았습니다[S1]. 같은 RRF 방식은 공개 검색 엔진의 하이브리드 검색 기능으로도 제공됩니다[S9]. 의미 기반 재순위를 RRF 합산 다음에 적용하는 순서도 공개 검색 서비스의 설계와 같습니다[S1]. pgvector도 전문 검색과 함께 쓸 때 RRF나 교차 인코더(cross-encoder) 재순위로 결과를 합치는 방법을 안내합니다[S2].

다섯 번째 인용 답변 단계에서는 재순위로 고른 근거 둘레의 관계 정보도 함께 씁니다. 찾은 페이지와 링크로 이어진 페이지, 위아래로 이어진 상위·하위 페이지, 같은 사람이나 같은 속성 값을 가진 이웃 페이지를 근거 후보에 더하고, 질문에 특정 페이지 이름이 나오면 그 페이지를 출발점으로 저장된 그래프 질문을 하나 실행해 관계 근거를 보탭니다. 답이 만들어진 뒤에는 가드레일 모델이 출력도 한 번 더 검사하며, 대화 기록은 정해진 보관 기간 동안만 저장하고 그 안에 들어간 개인 번호는 가려 둡니다. 질문이 들어오는 입구와 답이 나가는 출구 양쪽에 검사가 있는 셈입니다.

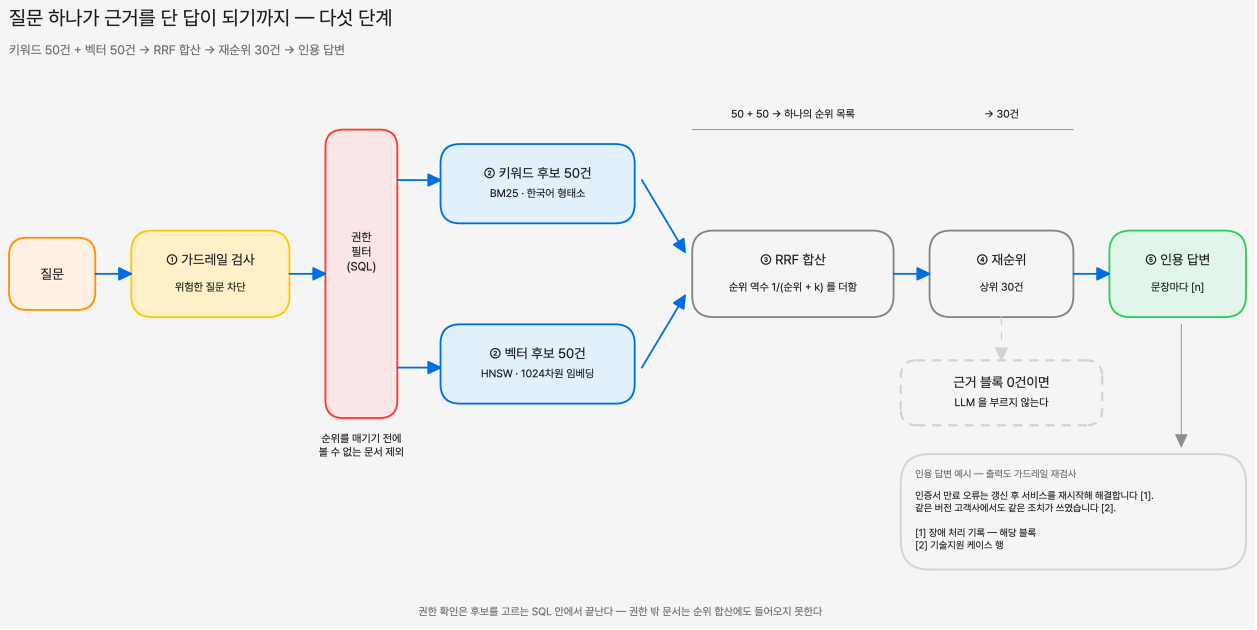

그림. 하이브리드 검색 다섯 단계

3.4.2 순위 전에 SQL에서 거르는 권한과 근거 없는 질문의 모델 호출 생략

권한 확인은 키워드와 벡터 두 경로 모두에서 SQL 질의 안에서 이뤄집니다. 즉 사용자가 볼 수 없는 페이지와 행은 후보 50건을 고르는 단계에서 이미 빠지므로, 순위 합산이나 재순위 단계에 권한 밖 문서가 끼어들 수 없습니

다. 반면 검색 결과를 먼저 뽑은 뒤 애플리케이션에서 권한을 거르는 구성은 걸러진 만큼 후보가 줄어 답 품질이 흔들리고, 필터 코드에 결함이 생기면 권한 밖 문서가 LLM 입력에 섞일 위험이 있습니다. MindWiki는 검색 결과에 근거로 쓸 블록이 없으면 LLM을 호출하지도 않으며, 이 규칙 덕분에 위키에 없는 내용을 물었을 때 모델이 그럴듯한 답을 지어내는 일을 막고, 쓸모없는 모델 호출로 GPU 자원을 쓰는 일도 줄어듭니다. 의사결정권자 입장에서는 보안 위험과 운영 자원 소모를 검색 구조 단계에서 함께 줄인다는 의미입니다.

3.5 사내 평가 세트 기준 검색 품질 — MRR 0.815와 재현율@10 0.810

검색 품질은 사내 평가 세트로 확인했으며, 아래 수치는 모두 그 조건을 전제로 합니다. 평가 세트는 질문마다 정답 문서를 정해 둔 묶음이므로, 실제 도입 조직의 문서 구성과 질문 유형에 따라 값은 달라집니다. 다만 같은 평가 세트 안에서 검색 방식을 바꿨을 때의 차이는 각 색인이 무엇을 더하는지 판단하는 근거가 됩니다.

항목	값	조건
MRR(하이브리드 검색 + 재순위)	0.815	사내 평가 세트 기준
재현율@10(키워드 + 벡터)	0.810	사내 평가 세트 기준, 키워드만 쓸 때 0.763
행 속성을 묻는 질문	0 → 1.000	사내 평가 세트 기준, 행 속성 색인 전후
키워드 검색 응답 시간 p50	10ms	사내 평가 기준, 블록 약 40만 개

3.5.1 키워드만 쓸 때 0.763과 키워드·벡터를 함께 쓸 때 0.810의 재현율 차이

재현율@10은 찾아야 할 정답 문서 가운데 상위 10건 안에 들어온 비율입니다. 사내 평가 세트 기준으로 키워드 검색만 쓰면 0.763이고, 키워드와 벡터를 RRF로 합치면 0.810으로 올라갑니다. 키워드 검색이 놓치던 정답 가운데 일부를 벡터 검색이 표현이 다른 문서에서 찾아낸 결과입니다. 재순위 모델까지 더한 MRR(Mean Reciprocal Rank, 평균 역순위)은 0.815입니다. MRR은 첫 번째 정답이 나온 순위의 역수를 평균한 값이라, 0.815는 대부분의 질문에서 정답이 첫 번째나 두 번째 결과로 나온다는 뜻입니다. 따라서 AI 답변이 상위 근거 몇 건만 읽어도 맞는 문서를 근거로 삼을 가능성이 높습니다.

3.5.2 행 속성 질문 0에서 1.000, 블록 약 40만 개 기준 키워드 검색 p50 10ms

가장 큰 차이는 데이터베이스 행 속성을 묻는 질문에서 나타났습니다. 행 속성을 색인하기 전에는 「담당자가 누구인 케이스」처럼 속성 칸에만 있는 정보를 묻는 질문에 사내 평가 세트 기준으로 정답을 하나도 찾지 못했고, 행마다 속성을 한 블록으로 색인한 뒤에는 1.000이 되었습니다. 이 결과는 업무 데이터를 표로 관리하는 조직일수록 행 속성 색인의 효과가 크다는 점을 보여 줍니다. 응답 속도는 블록 약 40만 개 규모에서 키워드 검색 중간값(p50)이 10ms였습니다. 다만 벡터 검색과 재순위, LLM 답변 생성까지 포함한 전체 응답 시간은 모델 서버 자원과 동시 사용자 수에 따라 달라지므로, 도입 규모에 맞춰 GPU와 데이터베이스 자원을 함께 계획해야 합니다.

4장: 업무 속성으로 만드는 지식 그래프 — 개발자 없이 현업이 정하는 관계 질의

MindWiki의 지식 그래프는 업무 담당자가 데이터베이스를 설계할 때 정한 속성을 그대로 노드와 엣지로 씁니다. 「이 케이스와 같은 제품을 도입한 고객사의 다른 케이스」처럼 여러 기록을 건너야 답이 나오는 질문은 키워드 검색과 벡터 검색만으로는 찾기 어렵고, 이를 풀려고 문서에서 LLM으로 그래프를 추출하면 비용과 품질 문제가 함께 생깁니다. 그래서 MindWiki는 그래프를 따로 구축하는 대신 현업이 이미 관리하는 업무 속성을 연결선으로 삼고, 비어 있는 속성은 AI가 값을 제안하면 사람이 수락하는 방식으로 채웁니다. 아래에서는 추출형 그래프와 속성형 그래프의 차이, 현업이 맡는 설계 범위, 사내 평가 기준의 관계 질문 성능과 아직 남은 사용성 과제를 차례로 설명합니다.

4.1 문서 더미에서 LLM으로 그래프를 뽑는 방식의 비용과 품질 문제

지식 그래프는 개체를 노드로, 개체 사이의 관계를 엣지로 표현한 데이터 구조이며, 흩어진 기록을 공통 속성으로 이어야 답이 나오는 질문에 쓰입니다. 일반적인 RAG(Retrieval-Augmented Generation)는 질문과 비슷한 문단을 찾아 모델에 넘기므로, 답이 한 문단 안에 쓰여 있지 않고 여러 문서에 나뉘어 있으면 약해집니다. 이 약점을 보완하려고 LLM으로 비공개 문서에서 지식 그래프를 추출하는 GraphRAG 방식이 2024년 Microsoft Research에서 공개되었습니다[S5]. 다만 이 방식은 문서가 많아질수록 비용이 커지고, 업무의 뜻을 모르는 모델이 관계를 정하므로 결과 품질도 일정하지 않습니다.

4.1.1 문서가 많을수록 커지는 LLM 추출 비용

추출형 그래프는 문서를 일정한 크기로 나눈 뒤 조각마다 LLM을 호출해 개체와 관계를 추출하고, 그 결과를 합쳐 그래프를 만듭니다[S5]. 이 구조에서는 모델 호출 횟수가 문서 양에 비례해 늘어나고, 문서가 수정되면 바뀐 조각을 다시 추출해 그래프에 반영해야 합니다. 회사 문서가 수만 건을 넘으면 그래프를 처음 만드는 작업만으로도 사내 GPU 자원을 오래 차지하게 되고, 그동안 같은 모델을 쓰는 다른 업무의 응답이 늦어질 수 있습니다. 소요 시간과 자원은 문서 규모, 모델 환경에 따라 달라지지만, 문서가 늘어날 때마다 추출 비용도 함께 늘어나는 구조는 같습니다.

두 방식의 차이를 정리하면 다음과 같습니다.

비교 항목	LLM 추출형 그래프	MindWiki 속성형 그래프
노드·엣지를 정하는 주체	모델이 문장에서 추출	업무 담당자가 데이터베이스 속성으로 정의
문서 추가·수정 시 처리	바뀐 조각마다 LLM을 다시 호출	저장할 때 속성 값으로 엣지를 갱신(약 1초)
관계의 의미	문장에 나온 표현에 따라 달라짐	속성 이름이 곧 관계 이름이라 일정함
그래프 생성에 드는 모델 호출	문서 양에 비례	그래프 생성에는 호출하지 않음(값 제안에만 사용)
잘 맞는 질문	문서 묶음 전체의 주제 요약	업무 기록 사이의 관계를 묻는 질문

4.1.2 업무 뜻을 모른 채 뽑은 노드와 엣지의 품질 한계

LLM은 문장의 명사를 노드로, 동사를 엣지로 옮기는 데 익숙하지만, 그 명사가 업무에서 어떤 역할을 하는지는 문서 밖의 맥락이 있어야 합니다. 기술지원 보고서에 「A사 운영 서버에서 APM 에이전트를 재시작했다」라는 문장이 있으면, 추출형 그래프는 A사·운영 서버·APM 에이전트를 노드로 만들고 「재시작」을 엣지로 남길 가능성이 높습니다. 하지만 담당자가 실제로 묻는 질문은 「A사가 도입한 제품에서 끝나지 않은 다른 케이스」이고, 이 질문에 필요한 「도입」 관계는 보고서 문장에 직접 쓰여 있지 않으면 그래프에 생기지 않습니다. 같은 제품이 문서마다 약칭·영문명·버전 표기로 다르게 쓰이면 노드가 여러 개로 나뉘어, 관계를 따라가도 기록이 이어지지 않습니다. 업무 맥락 없이 추출한 관계는 양이 많아도 질문에 쓸 수 있는 관계가 적고, 이 한계는 모델을 바꿔도 사라지지 않습니다.

4.2 담당자가 정한 업무 속성이 그대로 그래프 엣지가 되는 데이터베이스 설계

MindWiki 데이터베이스의 행은 문서 한 장이면서 표의 한 줄이고, 행마다 작성자·부서·카테고리·제품·버전·고객사·설치 환경 같은 속성을 가집니다. 이 속성은 담당자가 업무를 관리하려고 어차피 채우는 값이므로, 그래프를 만들기 위해 따로 입력할 내용이 없습니다. MindWiki는 관계 속성과 사람 속성, 선택 값을 PostgreSQL 안의 Apache AGE 그래프에 페이지·사람·값 노드와 속성 엣지로 옮깁니다. Apache AGE는 기존 관계형 데이터베이스 안에서 그래프 질의를 쓰게 해 주는 확장이므로[S3], 별도 그래프 데이터베이스를 설치하거나 문서 저장소와 동기화할 필요가 없습니다.

4.2.1 고객사와 제품, 버전, 설치 환경 속성이 관계가 되는 예

기술지원 업무를 예로 들면, 그래프의 노드는 고객사·프로젝트·도입 제품·설치 환경·기술지원 케이스·엔지니어 여섯 종류이고 엣지는 각 데이터베이스의 관계 속성에서 나옵니다. 고객사 행의 프로젝트 속성이 고객사와 프로젝트를 잇고, 도입 제품 속성이 고객사와 제품을 잇습니다. 프로젝트 행의 설치 환경 속성은 프로젝트를 설치 환경에 잇고, 기술지원 케이스 행의 제품·담당자 속성은 케이스를 제품과 엔지니어에 각각 잇습니다. 이렇게 만들어진 그래프에서 「고객사 → 프로젝트 → 설치 환경」이나 「도입 제품 → 기술지원 케이스 → 엔지니어」 같은 경로를 따라가면, 문서 여러 장에 흩어진 사실이 하나의 답으로 모입니다.

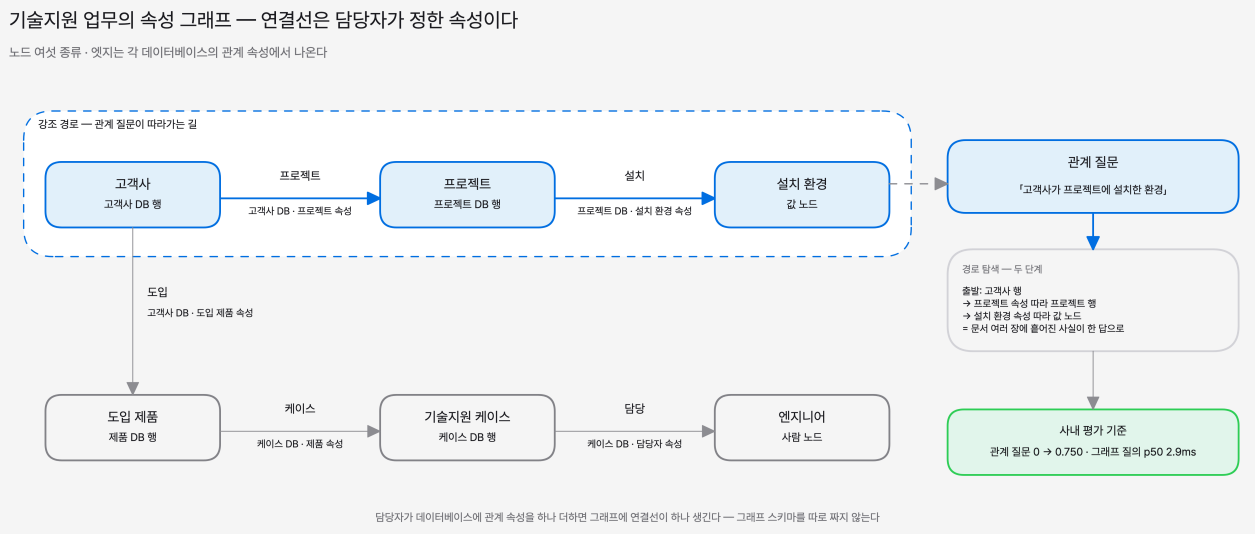

그림. 기술지원 업무의 속성 그래프

4.2.2 데이터베이스 설계 단계에서 정하는 관계 속성

현업 담당자는 데이터베이스를 만들 때 화면에서 속성을 하나씩 추가하고, 속성마다 글·선택·사람·날짜·URL·관계·롤업·수식 같은 종류를 고릅니다. 이 가운데 다른 데이터베이스의 행을 가리키는 관계 속성이 그래프의 엣지가 되고, 사람 속성은 사람 노드로, 선택 값은 값 노드로 이어집니다. 담당자가 정하는 것은 그래프 스키마나 Cypher 질의가 아니고, 「이 업무를 관리하려면 어떤 칸이 필요한가」라는 업무 설계입니다. 개발 조직은 데이터베이스 구조를 검토하고 권한을 설정하는 정도로 관여하며, 노드·엣지 정의나 추출 파이프라인을 따로 만들 필요가 없습니다. 기술지원·영업·구축처럼 자주 쓰는 업무는 회사 세트가 관계 속성을 미리 갖춘 데이터베이스를 제공하므로, 처음부터 설계하지 않고 필요한 속성만 더해 쓸 수 있습니다.

4.3 비어 있는 속성을 AI가 제안하고 사람이 수락하는 그래프 확장

속성형 그래프의 품질은 속성 값이 얼마나 빠짐없이 채워지는지에 달려 있고, 바쁜 담당자가 모든 칸을 직접 채우기는 어렵습니다. MindWiki는 행의 본문을 읽고 비어 있는 속성에 들어갈 값을 AI가 제안하게 하고, 담당자가 그 값을 확인해 수락하거나 거절하게 합니다. 이 방식에서 AI는 정해진 칸 안의 값만 제안하므로 결과를 검토하기 쉽고, 수락된 값이 쌓일수록 그래프의 연결도 함께 늘어납니다.

4.3.1 정해진 틀을 채우는 일에서 가장 정확한 AI

LLM은 자유롭게 글을 쓰게 할 때보다 정해진 틀 안의 값을 찾게 할 때 더 정확합니다. 자유 작성은 정답의 범위가 넓어 사실과 다른 내용이 섞여도 알아차리기 어렵지만, 속성 채우기는 후보가 선택 목록이나 기존 행으로 좁혀져 있고 근거가 되는 본문이 같은 행에 있기 때문입니다. 예를 들어 100줄짜리 장애 보고서에서 「사용된 제품」을 찾게 하면, 모델은 제품 데이터베이스에 등록된 이름 가운데 본문에 나온 것을 고르기만 하면 됩니다. 담당자는 제안된 값과 본문을 나란히 보고 맞는지만 판단하므로, 직접 입력할 때보다 검토 시간이 짧습니다.

4.3.2 기술지원 케이스의 원인 분류와 해결 방법 값 제안

기술지원 케이스 행을 열면 비어 있는 원인 분류·첫 응답 일시·해결 일시·해결 방법 속성 아래에 「AI가 채울 값」이 표시되고, 그 옆에 수락 버튼과 거절 버튼이 붙습니다. 해결 방법 칸에는 「에이전트 설치 서버와 APM 수집 서버의 시각 동기화 상태 확인 및 조정」처럼 본문에서 찾은 조치가 제안되고, 담당자가 수락하면 그 값이 속성으로 저장됩니다. 저장된 값은 키워드·벡터 색인과 그래프에 함께 반영되므로, 같은 원인 분류를 가진 다른 케이스나 같은 해결 방법이 쓰인 제품을 곧바로 관계 질문으로 찾을 수 있습니다. 반면 거절한 값은 저장되지 않으므로, 잘못된 제안이 그래프의 연결로 굳지 않습니다. 담당자의 검토 부담은 제안 하나에 클릭 한 번이고, 케이스가 쌓일수록 원인과 해결 방법 사이의 관계가 촘촘해집니다.

속성 값 제안과 태그 추천은 담당자가 버튼을 누를 때만 작동합니다. 행을 열 때마다 모델이 저절로 돌지 않으므로, 아무도 보지 않는 제안에 모델 자원을 쓰지 않고 사람이 부르지 않은 값이 화면을 채우지도 않습니다. 제안을 받아들여야만 저장된다는 원칙과 함께, AI가 언제 개입하는지를 사람이 정한다는 원칙이 속성 단위까지 적용되는 셈입니다.

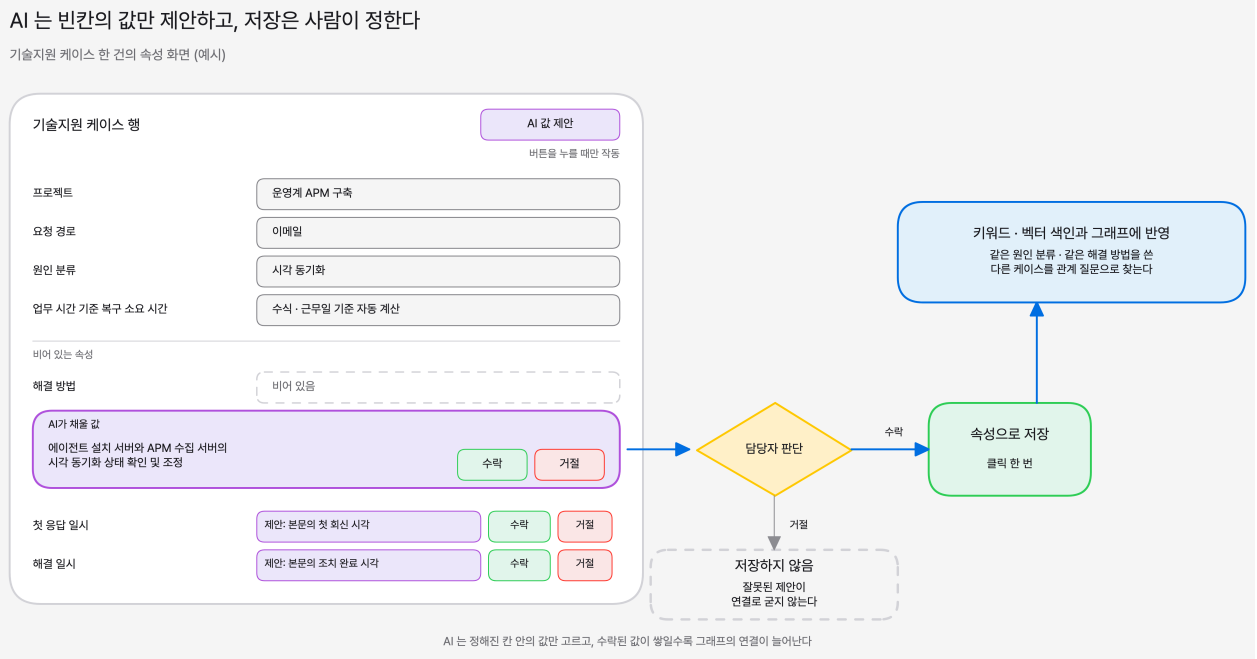

그림. 빈 속성에 대한 AI 값 제안과 수락

4.4 관계를 묻는 질문에 답하는 그래프 질의 — 사내 평가 기준 0에서 0.750

MindWiki의 그래프 질의는 관계를 묻는 질문에 쓰이고, 일반 질문은 3장의 하이브리드 검색이 맡습니다. 사내 평가 기준으로 관계를 묻는 질문은 키워드·벡터 검색만 쓸 때 정답을 하나도 찾지 못했고(0), 그래프 질의를 더한 뒤에는 0.750까지 올라갔습니다. 일반 질문에서는 그래프를 더해도 결과가 나아지지 않았으므로, MindWiki는 질문 유형에 따라 그래프 질의를 선택해 씁니다. 그래프 질의의 응답 시간은 사내 평가 기준 p50 2.9ms로, 검색 단계에 더해도 사용자가 느끼는 지연은 거의 없습니다.

4.4.1 같은 제품을 도입한 곳의 다른 케이스를 찾는 관계 질문 예

기술지원 데이터베이스에서 그래프 질의로 답하는 질문은 다음과 같은 형태입니다.

- 이 케이스와 같은 제품을 도입한 곳의 다른 케이스
- 고객사가 프로젝트에 설치한 환경
- 제품 도입처에서 아직 끝나지 않은 케이스
- 유지보수가 90일 안에 끝나는 고객사의 도입 제품

네 질문 모두 답이 문서 한 장에 쓰여 있지 않고, 케이스·고객사·프로젝트·제품 행을 두세 단계 건너야 나옵니다. 키워드 검색은 질문에 나온 낱말이 들어간 문서를, 벡터 검색은 질문과 뜻이 비슷한 문단을 찾으므로, 「같은 제품을 도입한 곳」이라는 조건을 행 사이의 연결로 풀어내지 못합니다. 첫 번째 질문은 현재 케이스의 제품 속성에서 출발해 그 제품을 도입한 고객사들로 이동한 뒤, 다시 그 고객사들의 케이스로 내려가는 경로 탐색이어야 답이 나옵니다. 따라서 관계 질문이 많은 업무일수록 속성형 그래프를 갖춘 효과가 크게 나타납니다.

4.4.2 Cypher 없이 고르는 미리 만든 질의 패턴과 남은 사용성 과제

Apache AGE는 Cypher와 비슷한 그래프 질의 언어를 지원하고 필요하면 SQL 로도 그래프를 조회할 수 있지만[S3], 현업 담당자에게 질의 언어를 배우게 하는 것은 현실적이지 않습니다. 그래서 MindWiki는 자주 쓰는

관계 질문을 질의 패턴으로 미리 만들어 두고, 사용자가 화면에서 패턴을 고른 뒤 출발 행만 지정하게 합니다. 회사 세트의 기술지원 데이터베이스에는 위 네 가지를 포함한 그래프 질문 일곱 개가 들어 있고, AAS 에이전트도 같은 패턴을 불러 답의 근거를 모읍니다. 다만 패턴에 없는 새로운 관계 질문은 관리자가 패턴을 추가해야 쓸 수 있고, 경로를 고르고 결과를 그래프 보기로 읽는 화면은 더 쉬워져야 하는 개선 과제로 남아 있습니다. 여러분의 조직이 파일럿을 계획한다면, 담당자가 실제로 자주 받는 관계 질문 서너 개를 먼저 적고 그 질문에 필요한 관계 속성이 데이터베이스에 있는지부터 확인하는 편이 효과를 빨리 확인하는 방법입니다.

그래프 질의 화면은 질의문을 쓰는 대신 도식을 보며 출발 노드, 따라갈 관계, 거를 조건을 차례로 고르는 방식입니다. 이렇게 만든 질문은 이름을 붙여 저장해 두고 팀스페이스마다 다시 불러 쓸 수 있으므로, 관계 질문 하나를 잘 만들어 두면 같은 팀 구성원 모두가 같은 기준으로 답을 얻습니다.

5장: AAS 에이전트가 위키를 읽고 쓰는 구조와 사람 승인으로 이어지는 지식 순환

MindWiki는 AAS(AI Agent Studio)와 함께 동작하도록 설계한 지식 위키입니다. AAS는 업무별 AI 에이전트를 만들고 운영하는 오픈마루의 에이전트 플랫폼이고, 두 제품은 MCP(Model Context Protocol, LLM 애플리케이션과 외부 데이터·도구를 잇는 공개 규약)로 연결됩니다[S4]. 에이전트가 회사 문서를 읽고 쓰기 시작하면 의사결정권자가 가장 먼저 묻는 것은 에이전트가 권한 밖의 문서를 열거나 중요한 페이지를 지우지 않느냐는 점입니다. MindWiki는 연결 방식, 도구 범위, 처리 흐름, 저장 단계마다 사람이 통제하는 장치를 두었습니다. 이 장에서는 에이전트가 위키 위에서 일하는 방식과 그 결과가 다시 다음 답의 근거로 쌓이는 순환 구조를 설명합니다.

5.1 AAS 로그인 한 번으로 연결되는 에이전트와 사용자 권한 그대로의 접근

에이전트를 위키에 연결할 때 운영 부담이 가장 큰 일은 연결 키를 발급하고 나눠 주고 만료 전에 바꾸는 작업입니다. MindWiki와 AAS는 같은 사내 SSO(Single Sign-On, 한 번 로그인으로 여러 시스템을 쓰는 인증 방식)를 쓰므로 이 작업을 사용자 로그인 하나로 대신합니다.

5.1.1 키 입력 없는 연결 등록과 토큰 자동 갱신

사용자가 AAS에 로그인하면 그 사용자의 MindWiki 연결이 키 입력 없이 자동으로 등록됩니다. 연결에 쓰는 토큰도 만료 전에 시스템이 스스로 갱신하므로, IT 담당자가 사용자마다 MCP 토큰을 발급해 전달하고 만료 일정을 따로 관리할 필요가 없습니다. 키를 메신저나 메일로 주고받다가 유출하는 위험도 함께 줄어듭니다. 다만 외부 시스템을 자동으로 연동하는 용도라면 사람 계정과 별도로 API 토큰을 발급해 쓰고, 그 토큰의 범위와 사용 기록은 6장에서 설명하는 권한 체계가 관리합니다.

5.1.2 사용자 권한 안에서만 읽고 쓰는 에이전트

AAS 에이전트는 별도의 슈퍼 계정으로 위키에 접근하지 않고, 에이전트를 부른 사용자의 권한을 그대로 이어받습니다. 그 사용자가 열 수 없는 팀스페이스나 페이지는 에이전트도 검색 결과에서 보지 못하고, 그 사용자가 편집할 수 없는 문서는 에이전트도 고치지 못합니다. 에이전트를 도입해도 권한 설계를 새로 만들 필요가 없고, 기존 페이지 트리 권한과 행 접근 규칙이 에이전트에게 그대로 적용됩니다. 이 구조는 MCP 명세가 요구하는 원칙, 즉 사용자가 모든 데이터 접근과 작업에 명시적으로 동의하고 통제권을 가져야 한다는 원칙과도 맞물립니다[S4].

5.2 읽기 19개와 쓰기 17개 MCP 도구, 지우기 도구를 두지 않은 설계

MCP 서버는 자료·프롬프트·도구 세 가지를 에이전트에게 제공하며[S4], MindWiki가 내주는 도구는 읽기 19개와 쓰기 17개입니다. 페이지나 행을 지우는 도구는 처음부터 등록하지 않았습니다. 에이전트가 할 수 있는 일의 범위를 도구 목록으로 정해 둔 것입니다.

5.2.1 편집 제안으로 남는 에이전트의 쓰기와 MCP 사용자 동의 원칙

MCP 명세는 도구 호출을 임의 코드 실행과 같은 수준으로 신중히 다루라고 정하고, 호스트가 도구를 실행하기 전에 사용자 동의를 받도록 요구합니다[S4]. MindWiki는 이 원칙을 쓰기 도구의 동작 방식으로 옮겼습니다. 에

이전트가 쓰기 도구로 문서를 고치면 원문이 곧바로 바뀌지 않고 편집 제안으로 남으며, 사람이 그 제안을 받아들일 때 비로소 본문에 반영됩니다. 그 결과 도구 범주별로 에이전트가 할 수 있는 일은 다음과 같이 나뉩니다.

도구 범주	MCP에 등록한 도구	에이전트 호출 결과
읽기	19개(검색, 페이지 읽기, 데이터베이스 질의 등)	사용자 권한 안의 결과만 반환
쓰기	17개(페이지 작성, 행 값 채우기 등)	편집 제안으로 남고 사람이 수락할 때 반영
지우기·관리	등록하지 않음	MCP로는 호출할 수 없고 REST 관리 API로만 처리

5.2.2 관리 작업을 MCP에서 떼어 낸 REST 관리 API와 관리자 스킬

위키의 모든 API를 MCP 도구로 공개하면 도구 수가 수백 개로 늘어나고, LLM이 잘못된 판단으로 팀스페이스나 데이터베이스를 통째로 지울 가능성도 생깁니다. 그래서 MindWiki는 글을 쓰고 찾는 일은 MCP로, 팀 생성·계정 정리·스페이스 삭제 같은 관리 작업은 별도의 REST 관리 API로 나눴습니다. 관리 API를 에이전트에게 맡겨야 하는 경우, 예를 들어 부서 자료를 한꺼번에 옮기는 마이그레이션에는 관리자용 스킬을 따로 제공하고 그 사용 범위를 관리자가 정합니다. 이렇게 하면 일반 업무용 에이전트는 구조상 삭제 권한에 닿지 않으며, 사고 예방을 프롬프트 지시에 기대지 않고 도구 경계로 보장할 수 있습니다.

도구 경계와 함께 호출량에도 상한을 둡니다. MindWiki는 MCP 호출 수와 쓰기 횟수에 한도를 정하고, API 서버를 여러 대로 늘려 운영할 때도 이 한도를 서버마다 따로 세지 않고 모든 서버가 함께 셉니다. 에이전트가 같은 작업을 되풀이하는 오류에 빠지더라도 서버 대수만큼 한도가 늘어나는 빈틈이 없으므로, 위키 전체의 응답 속도가 에이전트 한 개 때문에 떨어지는 일을 막을 수 있습니다. 자동화가 부르는 AAS 에이전트는 그 에이전트의 주인이 동의한 팀스페이스에서만 움직이고, 실제로 값을 쓰는 순간에 그 동의를 한 번 더 확인합니다.

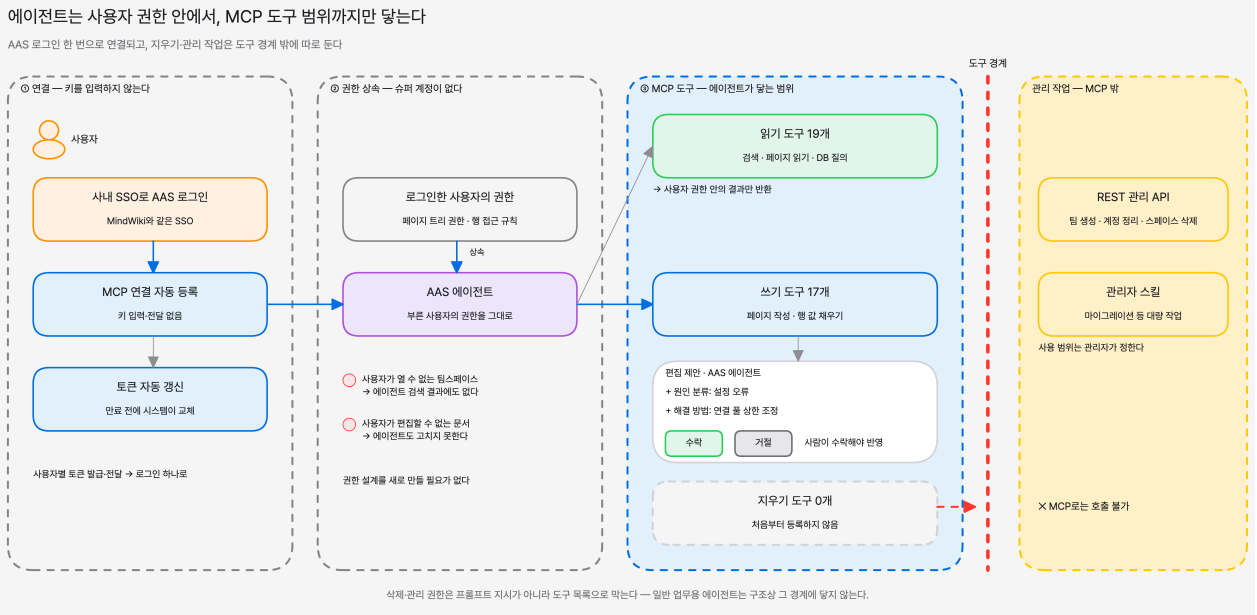

그림. 에이전트 연결 흐름과 MCP 도구 경계

5.3 자동화 규칙이 AAS 카드로 일을 넘기고 사람이 승인하는 처리 흐름

현업은 채팅 창에서 에이전트와 대화하며 일하기보다 업무 화면의 버튼을 누르고 결과를 확인하는 방식을 더 자주 씁니다. MindWiki의 자동화 규칙은 이 방식에 맞춰 데이터베이스 행의 조건이나 버튼을 AAS 에이전트의 작업과 바로 잇습니다.

5.3.1 조건과 버튼에서 AAS 카드, 행 결과 기록까지의 자동화 단계

자동화는 다섯 단계로 진행합니다. 먼저 행의 상태가 바뀌거나 담당자가 버튼을 누르면 자동화 규칙이 실행되고, 규칙은 AAS에 작업 카드를 만들어 일을 넘깁니다. 이때 MindWiki는 그 행과 연결된 FAQ, 관련 문서, 법령 조문 같은 근거를 하이브리드 검색과 그래프 질의로 먼저 모아 카드에 담습니다. 키워드 검색과 벡터 검색 결과는 순위 기반 융합인 RRF(Reciprocal Rank Fusion)로 합치고[S1], 관계를 따라가는 질의는 PostgreSQL 안의 Apache AGE 그래프로 처리합니다[S3]. 에이전트는 카드에 담긴 근거로 답변 초안이나 분류 값을 만들어 해당 행에 기록하고, 마지막으로 담당자가 결과를 승인합니다. 근거를 카드에 미리 모아 두면 에이전트가 검색 도구를 여러 번 되풀이해 부를 필요가 없으므로 모델 호출 횟수를 줄였고 처리 시간도 짧아졌습니다.

5.3.2 행에 보이는 진행 상태와 담당자 승인 단계

에이전트가 일하는 동안 해당 행에는 「AI 작업 중」 표시가 나타나므로, 담당자는 어떤 건이 처리 중이고 어떤 건이 검토를 기다리는지 목록 화면에서 바로 확인합니다. 에이전트가 쓴 결과는 승인 전 상태로 행에 남고, 담당자가 내용을 확인해 승인하거나 고쳐 쓴 뒤에야 다음 업무 단계로 넘어갑니다. 최종 결정과 그 책임은 언제나 사람에게 있고, 에이전트는 근거를 모으고 초안을 만드는 데까지 일합니다. 이 승인 단계를 두면 에이전트의 판단이 틀린 경우에도 고객이나 민원인에게 나가기 전에 걸러 낼 수 있어, 공공기관과 기업이 자동화를 받아들이는 부담이 줄어듭니다.

5.4 모든 페이지의 AI 버튼과 팀별 AAS 프로필 선택

자동화 규칙이 반복 업무를 맡는다면, 모든 페이지에 붙은 AI 버튼은 사람이 문서를 쓰는 과정을 돕습니다. 두 기능 모두 같은 AAS 에이전트를 부르므로, 문서 작성 보조에도 회사 위키의 근거가 그대로 쓰입니다.

5.4.1 길게 쓰기와 목차별 채우기, 번역, 읽어 주기 기능

AI 버튼이 제공하는 문서 작성 보조 기능은 다음과 같습니다.

- 선택한 문단을 길게 쓰기와 짧게 쓰기
- 키워드로 내용 채우기와 문장 다시 쓰기
- 목차 잡기와 목차별 본문 채우기
- 외부 검색 결과로 채우기(외부 접속을 켠 설치에서만 동작)
- 번역, 읽어 주기, 받아쓰기

이 가운데 목차별 채우기는 에이전트에게 정해진 틀을 주고 그 안을 채우게 하는 방식이라, 자유롭게 쓰게 할 때보다 결과가 고르고 검토도 쉽습니다. 에이전트는 내용을 채울 때 위키 안의 관련 문서를 근거로 삼으므로, 제안서나 기술 문서의 초안이 조직의 실제 용어와 사례를 따라갑니다.

다듬기와 번역은 결과를 원문 위에 덮어쓰지 않고 새 페이지로 저장하며, 표·목록·제목 같은 원문 서식을 그대로 살립니다. 원문과 번역본이 나란히 남으므로 검토자는 두 페이지를 비교해 확인할 수 있고, 번역본도 저장하는

순간 색인에 들어가 다른 언어로 묻는 질문의 근거가 됩니다. 해외 법인이나 외국인 직원과 같은 규정을 함께 써야 하는 조직이라면 원문 하나를 기준으로 언어별 페이지를 관리하는 방식이 됩니다.

5.4.2 팀과 업무마다 다른 말투와 스킬의 AAS 프로필

AAS 프로필은 에이전트의 말투, 역할, 사용할 스킬을 묶은 설정이며, MindWiki에서는 팀이나 업무마다 어떤 프로필을 쓸지 고를 수 있습니다. 예를 들어 민원 답변 팀은 공문 말투와 법령 조회 스킬을 갖춘 프로필을 팀 전체가 공유하고, 기술지원 팀은 장애 분석 스킬을 갖춘 다른 프로필을 씁니다. 팀 공용 프로필을 지정하면 사용자가 여럿이어도 같은 스킬과 기준으로 일하므로 답변 품질이 사람마다 흔들리지 않고, 개인이 자신의 작업에 맞는 프로필을 따로 붙여 쓸 수도 있습니다. 한 부서에서 검증한 프로필을 다른 부서로 넓힐 때 에이전트를 새로 만들지 않고 프로필과 스킬만 옮기면 됩니다.

5.5 위키에 쌓일수록 에이전트 답이 좋아지는 지식 순환 구조

에이전트 답변의 품질은 검색할 수 있는 근거의 양과 질이 정합니다. MindWiki와 AAS를 함께 쓰면 에이전트가 만든 결과가 다시 위키에 저장되어 다음 답의 근거가 되므로, 쓸수록 근거가 늘어나는 순환이 생깁니다.

5.5.1 위키에서 에이전트 근거, 결과 저장, 다음 답의 근거로 이어지는 순환

순환은 네 단계로 이어집니다. 위키에 저장한 문서와 행은 저장하는 순간 키워드·벡터·그래프 색인에 들어가고, 에이전트는 이 색인에서 근거를 찾아 답변을 만듭니다[S2]. 에이전트의 결과는 해당 페이지나 행에 기록되고, 승인을 거친 결과는 다시 색인되어 다음 질문의 근거가 됩니다. 예를 들어 기술지원 케이스 하나에 원인 분류와 해결 방법이 기록되면, 같은 제품과 버전을 쓰는 다른 고객사에서 비슷한 문의가 왔을 때 에이전트가 그 기록을 근거로 바로 찾아 씁니다. 일반 RAG는 흩어진 정보를 공통 속성으로 이어야 답이 나오는 질문에 약하다는 한계가 알려져 있는데[S5], 업무 속성을 갖춘 행으로 결과를 쌓는 방식은 이 한계를 데이터가 늘수록 줄여 줍니다. 즉 여러분의 조직이 MindWiki에 업무 기록을 남길수록 에이전트가 쓸 수 있는 근거가 넓어지고, 같은 질문에 대한 답도 점점 구체적으로 바뀝니다.

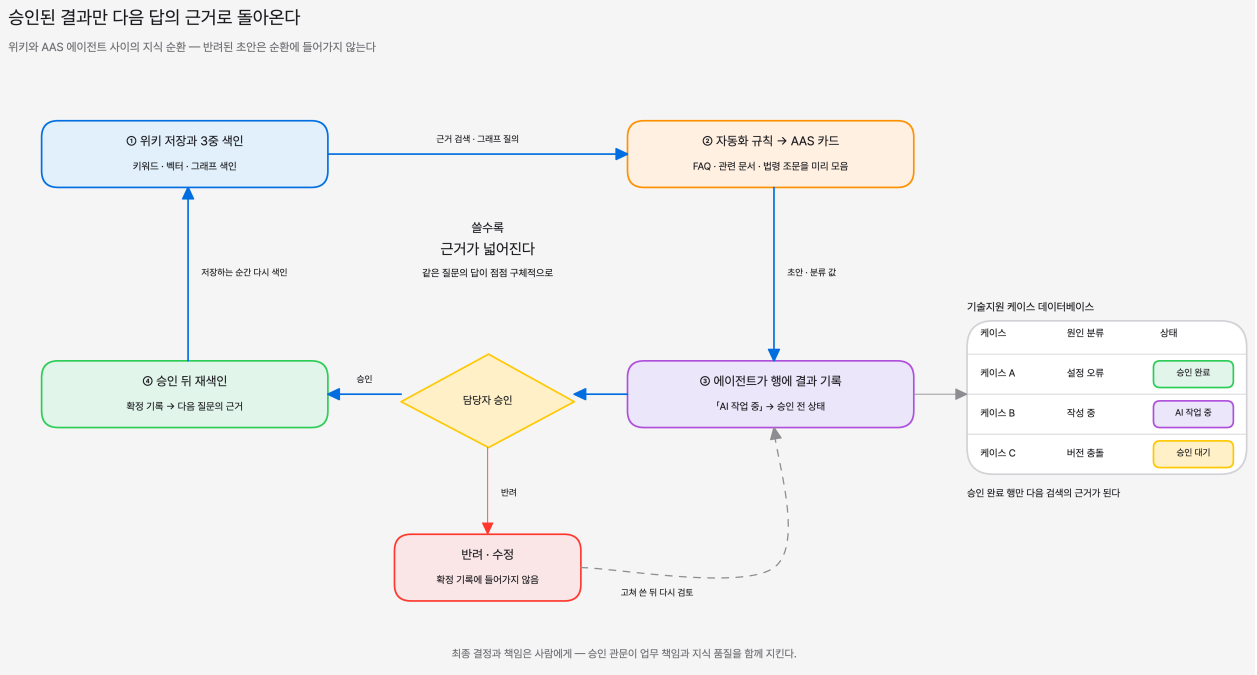

그림. 승인 관문을 거치는 지식 순환

5.5.2 승인된 결과만 지식으로 남기는 품질 관리

지식 순환에는 잘못된 답이 근거로 굳어 다음 답을 다시 오염시킬 위험도 있습니다. 그래서 MindWiki는 에이전트의 결과를 바로 지식으로 삼지 않고, 담당자의 승인을 거친 결과만 확정 기록으로 남깁니다. 반려된 초안이나 사람이 고치기 전의 원래 초안은 확정 기록에 들어가지 않으므로, 틀린 분류나 근거 없는 문장이 다음 검색 결과의 상위에 올라오지 않습니다. 또한 에이전트의 쓰기가 편집 제안으로 남기 때문에 에이전트가 고친 내용과 사람이 받아들인 내용을 제안 단위로 구분해 검토할 수 있습니다. 승인 단계는 업무 책임을 사람에게 두는 장치이자 위키에 쌓이는 지식의 품질을 지키는 장치로 함께 작동합니다.

승인 단계가 새로 들어오는 결과를 거른다면, 이미 쌓인 지식은 위키 점검 기능이 살핍니다. 위키 점검은 어디에서도 링크되지 않은 고아 페이지, 오랫동안 고치지 않은 페이지, 제목이 같은 페이지, 서로 어긋나는 내용, 출처 없이 적힌 주장, 본문에 남은 비밀값을 찾아 목록으로 보여 주며, 매일 또는 매주 정기 실행으로 걸어 둘 수 있습니다. 근거가 낡거나 서로 충돌하면 검색이 그 문서를 그대로 근거로 올리므로, 점검 목록을 담당자가 정리하는 주기를 운영 절차에 넣어 두기를 권합니다.

외부 자료를 위키로 들여올 때도 같은 원칙이 적용됩니다. 「위키에 반영」 기능은 자료를 요약해 위키 문서로 옮기면서 원본에 없는 숫자와 이름이 끼어들었는지 걸러 내고, 문장마다 원본과 맞는지 판정합니다. 모델이 요약 과정에서 만들어 낸 수치가 근거 문서인 척 위키에 쌓이는 일을 들여오는 단계에서 막는 장치입니다.

6장: 데이터 주권을 지키는 MSAP.ai 설치형 구성 — 회사 밖으로 나가지 않는 데이터와 AI 모델

MindWiki는 회사가 운영하는 쿠버네티스 플랫폼인 MSAP.ai 위에 설치되고, 문서·데이터베이스 행·첨부 파일과 이를 읽는 AI 모델이 모두 회사 안에서 동작합니다. 생성형 AI를 업무에 붙이려는 조직은 기능을 따지기 전에 보안 심사에서 데이터가 어디로 가는지를 설명해야 하며, 외부 클라우드형 도구를 쓰면 이 질문에 답하려고 계약서와 데이터 상주 지역 설정을 따로 확인해야 합니다. 이 장은 데이터 주권(데이터가 저장되고 처리되는 위치와 접근 통제를 조직이 스스로 결정하는 권리)을 MindWiki가 어떤 구성 요소로 지키는지, 각 요소가 어떤 위험을 막는지를 위험과 대응의 짝으로 정리했습니다. 여러분의 조직이 보안 검토 문서에 적어야 할 항목을 아래 표의 순서대로 확인할 수 있습니다.

막으려는 위험	MindWiki의 대응	담당 구성 요소
업무 문서가 외부 AI 서비스로 전송됨	사내 CogentAI 모델만 기본 경로로 사용	사내 모델·설정값으로 지정하는 모델 주소
폐쇄망에서 화면이 뜨지 않거나 외부로 요청이 나감	외부 CDN·글꼴 없이 화면 구성, 외부 기능은 기본 꺼짐	MSAP.ai 위 설치형 배포
AI에게 개인정보를 대량으로 뽑아 달라는 질문	대량 추출 질문 차단, 운영 용어 오탐 방지 학습	가드레일 모델
데이터베이스가 유출될 때 민감 정보 노출	칼럼 단위 AES-256-GCM 저장, 색인·그래프 제외	암호화 속성
권한 없는 사람이나 외부 연동이 문서를 읽음	페이지 트리 권한, 행 접근 규칙, 토큰 범위	권한 3층과 SSO 동기화
장애나 실수로 데이터와 키를 잃음	PostgreSQL 이중화, 정해진 주기의 백업	설치기와 백업 작업

6.1 데이터 주권이 공공기관과 기업 AI 도입의 선행 조건이 된 배경

국내 공공 부문은 2025년에 생성형 AI 활용 기준을 두 갈래로 내놓았는데, 하나는 국가정보원이 정한 보안 점검 기준이고 다른 하나는 개인정보보호위원회가 정한 개인정보 처리 기준입니다. 두 기준이 함께 묻는 것은 업무 데이터가 어느 모델로 들어가고 그 결과가 어디에 남는지입니다. 그래서 AI 도입 검토는 모델 성능 비교보다 데이터 경로 설명 단계에서 먼저 멈추는 일이 많아졌습니다.

6.1.1 국가정보원 AI 보안 가이드북이 정리한 위협 유형과 대책, 점검 항목

국가정보원 국가인공지능안보센터는 2023년 6월에 낸 「생성형 AI 활용 보안 가이드라인」을 고쳐 「AI 보안 가이드북」으로 묶고, 이를 중앙행정기관·지방자치단체·공공기관에 배포했습니다[S7]. 가이드북은 AI를 쓸 때 생기는 보안 위협과 그에 맞서는 대책, 기관이 스스로 확인할 점검표를 한 권에 담았으며 배포 사실은 2025년 12월에 보도되었습니다[S7]. 공공기관이 AI 도구를 도입할 때는 이 점검표를 기준으로 데이터 유출 경로와 접근 통제를 설명해야 합니다.

설치형 구성은 점검 항목에 답하는 근거를 기관 내부의 사실로 바꿔 줍니다. 데이터와 모델이 기관 안의 MSAP.ai 위에서만 동작하면 외부 서비스 약관이나 해외 처리 위치를 확인하는 항목 대신, 기관이 직접 운영하는 서버와 권한 설정을 근거로 답할 수 있기 때문입니다. 세부 점검 항목은 원문 가이드북의 조항을 기준으로 도입 협의에서 하나씩 맞춰 봅니다.

6.1.2 개인정보보호위원회 생성형 AI 개인정보 처리 안내서의 개발 방식 구분

개인정보보호위원회는 2025년 8월 6일 「생성형 AI 개발·활용을 위한 개인정보 처리 안내서」를 공개했고, 이 안내서는 생성형 AI를 만드는 방식을 서비스형 LLM 활용, 공개 모델 활용, 자체 개발의 세 가지로 나눕니다[S8]. 방식마다 개인정보가 조직 밖으로 나가는지가 다르므로, 도입하는 조직은 자기 방식이 어디에 해당하는지부터 정해야 개인정보 처리 근거를 쓸 수 있습니다.

MindWiki는 외부 서비스형 LLM에 문서를 보내지 않고 사내에서 운영하는 CogentAI 모델을 기본으로 쓰므로, 프롬프트와 검색 근거로 쓰인 문서가 외부 사업자에게 전달되는 경로가 기본 구성에 없습니다. 반면 외부 모델을 골라 연결하면 그 경로가 새로 생기므로, 어떤 모델을 쓸지는 개인정보 처리 방식과 함께 결정해야 합니다.

6.2 MSAP.ai 위 설치형과 인터넷 없이 동작하는 화면과 사내 CogentAI 모델

MindWiki는 API 서버, 동기화 서버, 문서 변환 서버로 이루어지고 모든 문서와 행 데이터는 PostgreSQL 한 곳에 저장됩니다. PostgreSQL 안에서 pgvector가 벡터 검색을, Apache AGE가 그래프 질의를 맡으므로 별도의 벡터 데이터베이스나 그래프 데이터베이스를 외부에 둘 필요가 없습니다. 첨부 파일은 설치할 때 함께 배포되는 S3 호환 저장소에 두고, 필요하면 고객이 이미 운영하는 S3 호환 스토리지에 연결합니다. 결과적으로 데이터가 머무는 저장소는 모두 MSAP.ai 클러스터 안이나 고객 스토리지 안에 있습니다.

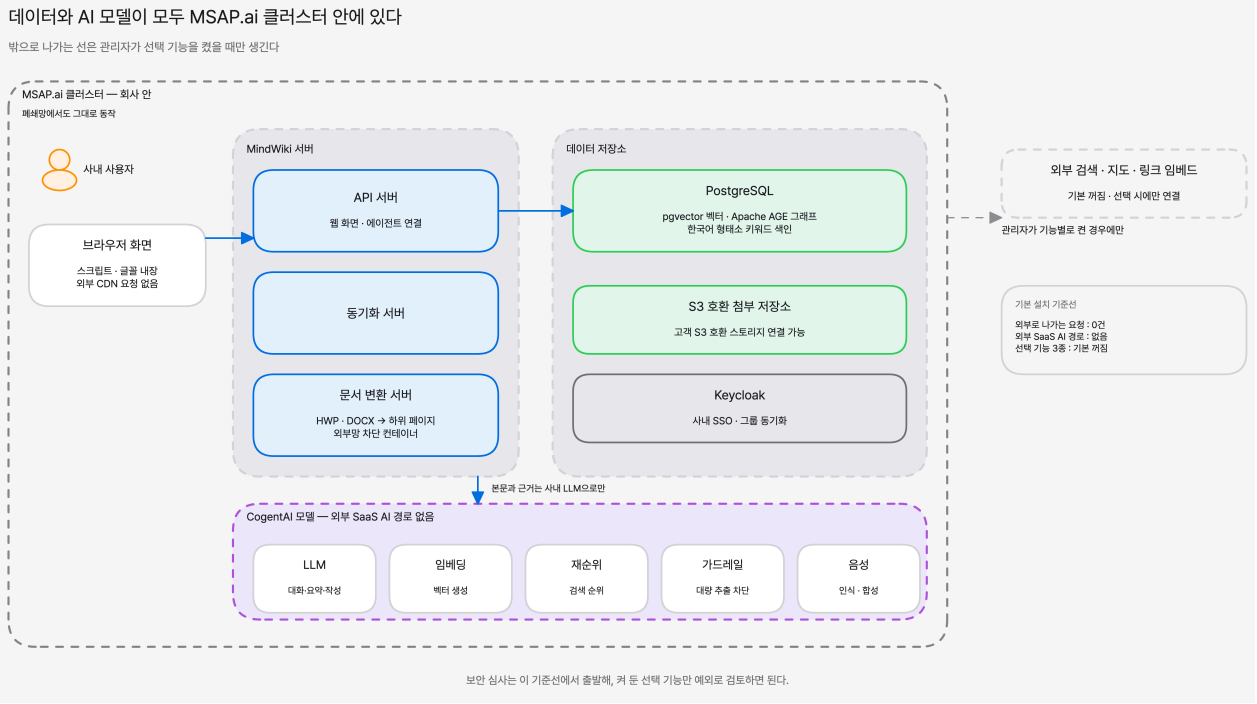

그림. MSAP.ai 클러스터 안의 데이터와 모델

6.2.1 외부 CDN과 글꼴 없이 뜨는 화면과 선택 시에만 켜는 외부 접속

웹 화면을 그리는 데 필요한 스크립트·스타일·글꼴은 모두 MindWiki 컨테이너 이미지 안에 들어 있어서, 브라우저가 외부 CDN(Content Delivery Network)에 파일을 요청하지 않습니다. 그 덕분에 인터넷과 분리된 폐쇄망에서도 화면이 그대로 뜨고, 사용자가 문서를 여는 순간 외부 서버에 접속 기록이 남는 일도 없습니다.

외부 웹 검색, 지도, 외부 링크 임베드처럼 인터넷이 있어야 하는 기능은 기본으로 꺼져 있으며 관리자가 필요한 기능만 골라 켭니다. 보안 담당자는 켜 둔 외부 기능의 목록만 점검하면 이 위키가 외부와 주고받는 접속 범위를 모두 확인할 수 있습니다.

문서 변환 서버도 같은 원칙을 따릅니다. 한글(HWP/HWPX)·DOCX 파일을 하위 페이지로 바꾸는 변환 컨테이너는 외부망 접속이 막힌 상태로 동작하므로, 변환하는 동안 문서 내용이 밖으로 나갈 경로가 없습니다. 외부 기능을 하나도 켜지 않은 기본 설치에서는 MindWiki가 바깥으로 보내는 요청이 0건이며, 보안 심사에서는 이 기본 상태를 기준선으로 두고 켜 둔 선택 기능만 예외로 검토하면 됩니다.

6.2.2 사내 CogentAI 모델과 설정값으로 바꾸는 모델 주소

MindWiki가 쓰는 AI 모델은 오픈마루의 CogentAI 모델이며, 대화·요약·작성을 맡는 LLM과 함께 임베딩 모델, 재순위 모델, 가드레일 모델, 음성 인식·합성 모델로 구성됩니다. 이 모델들은 MSAP.ai 위에서 위키와 함께 운영되고 외부 SaaS AI는 기본 경로에 들어 있지 않습니다. 예를 들어 사용자가 AI 버튼으로 회의록 요약을 요청하면 본문과 검색 근거는 클러스터 안의 LLM으로만 전달됩니다.

모델 주소는 소스 코드를 고치지 않고 설정값으로 지정하므로, 조직이 이미 운영하는 사내 모델 서버로 바꿔 연결할 수 있습니다. 어떤 모델까지 교체할 수 있는지는 모델 종류와 운영 환경에 따라 달라지므로 도입 협의에서 함께 정합니다.

다만 임베딩 모델은 다른 모델과 교체 방식이 다릅니다. 임베딩 모델을 바꾸면 기존 벡터와 새 벡터가 서로 다른 공간에 놓이므로 모든 블록을 다시 임베딩해 벡터 색인을 새로 만들어야 하고, 벡터 차원 수까지 달라지면 색인 구성도 함께 바뀝니다. MindWiki는 원본에서 색인을 다시 만드는 구조라 절차 자체는 단순하지만(3.1.2), 문서가 많을수록 다시 만드는 시간이 길어집니다. 그래서 임베딩 모델을 바꿀 때는 새 색인을 따로 만들어 같은 질문으로 검색 결과를 비교한 뒤 전환하는 순서를 권하며, 처음 모델을 고를 때도 자기 조직 문서로 충분히 평가해 두는 편이 안전합니다.

6.3 가드레일과 암호화 속성 — 대량 개인정보 추출 차단과 칼럼 단위 AES-256-GCM

MindWiki는 개인정보를 두 겹으로 보호합니다. 첫 번째 겹인 가드레일은 AI에게 들어오는 질문을 검사해 개인정보를 대량으로 꺼내려는 요청을 막고, 두 번째 겹인 암호화 속성은 데이터베이스에 저장되는 값 자체를 암호화해 저장소가 유출되더라도 민감 정보를 읽을 수 없게 합니다. 하지만 두 장치 모두 편의를 일부 포기하는 선택이므로, 어떤 속성을 암호화할지는 업무 담당자와 함께 정해야 합니다.

6.3.1 대량 개인정보 추출 질문 차단과 운영 용어 오탐 방지 학습

가드레일은 검색과 답변 생성보다 먼저 실행되어, 「회원 정보를 전부 뽑아 달라」처럼 개인정보를 한꺼번에 꺼내려는 질문을 그 단계에서 차단합니다. 한 입력에 식별 가능한 개인정보가 기준 개수를 넘게 들어 있으면 처리를 멈추며, 명함 한 장을 정리하는 것처럼 업무에 필요한 소량 입력은 통과하도록 기준을 맞췄습니다.

범용 가드레일은 운영 업무 질문을 잘못 막는 경우가 있습니다. 「마약류 관리 시스템의 CPU 사용률」을 묻는 질문은 시스템 이름에 든 낱말 때문에 범용 모델이 위험 질문으로 판정하기 쉽습니다. 그래서 CogentAI 가드레일

은 관리 시스템 이름과 운영 용어를 정상 질문으로 판단하도록 추가 학습했고, 막아야 할 질문과 통과시켜야 할 업무 질문의 경계를 업무 데이터 기준으로 다시 맞췄습니다.

6.3.2 검색 색인과 그래프에서 빠지는 암호화 속성과 열람 비밀번호

데이터베이스 속성 하나를 암호화 속성으로 지정하면 그 칼럼의 값은 AES-256-GCM으로 암호화되어 저장되며, 이메일·휴대전화 번호·서버 접속 비밀번호가 대표적인 적용 대상입니다. 값을 보려면 사용자별 열람 비밀번호를 한 번 더 입력해야 하므로, 페이지 열람 권한만으로는 민감 정보가 화면에 나타나지 않습니다.

암호화 속성은 3장에서 설명한 키워드·벡터 색인과 지식 그래프에도 들어가지 않으므로 AI 검색으로는 이 값을 찾을 수 없고, 가드레일을 우회한 질문이 들어오더라도 근거 데이터에 해당 값이 없습니다. 이 방식은 검색 편의를 일부 줄이는 대신, 저장소 유출과 AI를 통한 추출이 함께 일어나도 민감 정보가 그대로 나가지 않게 합니다.

6.4 페이지 트리 6단계 권한과 행 접근 규칙, API 토큰 범위의 권한 3층

MindWiki의 접근 통제는 세 층으로 나뉩니다. 첫 층은 페이지 트리 권한으로 팀스페이스와 페이지마다 6단계 권한 가운데 하나를 사람이나 그룹에 주고, 둘째 층은 데이터베이스의 행 접근 규칙으로 담당자 속성에 이름이 든 사람만 그 행을 읽고 고치게 하며, 셋째 층은 외부 연동이 쓰는 토큰의 범위입니다. 검색 결과에도 같은 권한이 적용되므로, 사용자나 에이전트가 볼 수 없는 문서는 AI 답변의 근거에도 들어가지 않습니다.

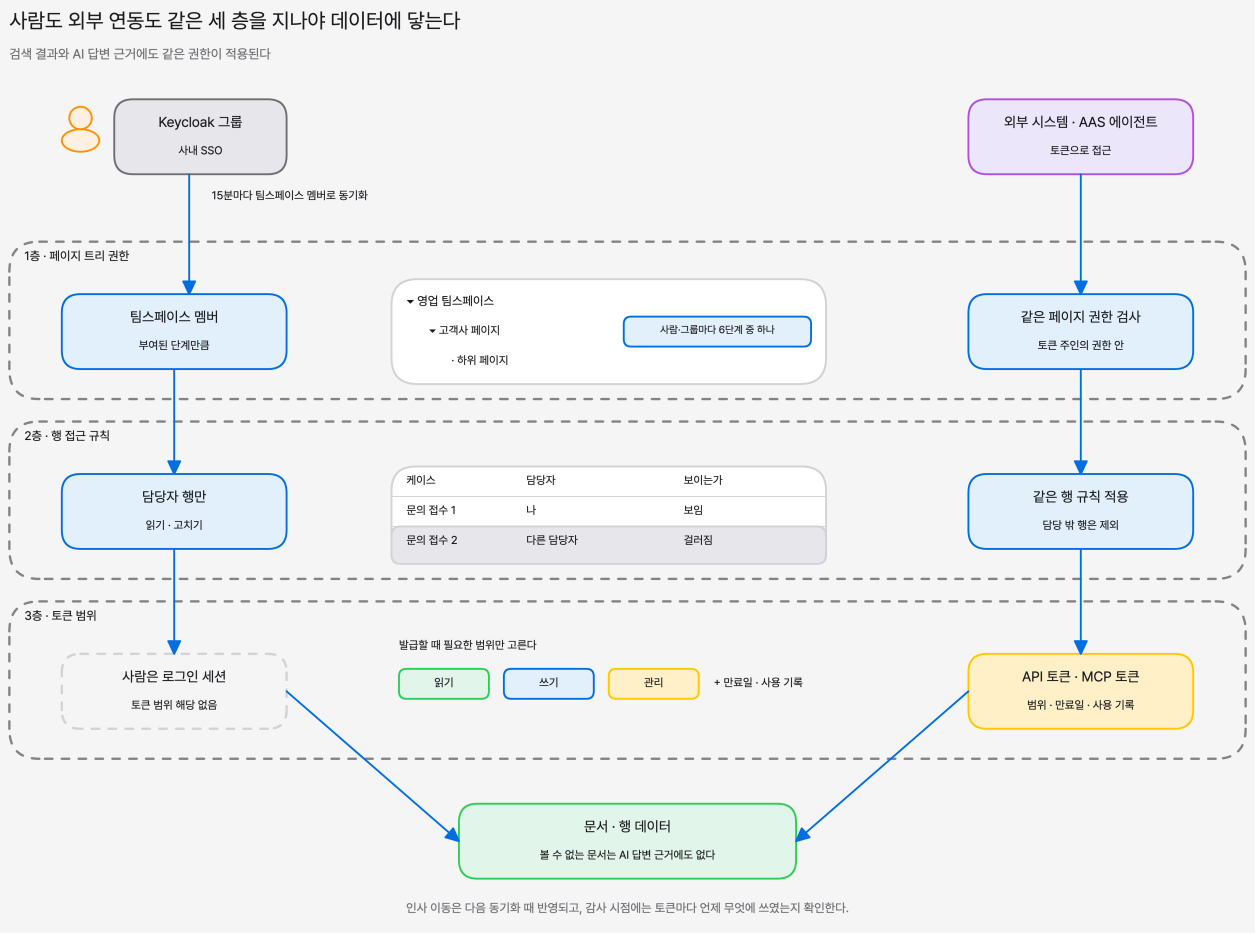

그림. 사람과 외부 연동이 지나는 권한 3층

6.4.1 SSO 그룹을 15분마다 팀스페이스 멤버로 맞추는 계정 연동

MindWiki는 오픈소스 SSO(Single Sign-On, 한 번 로그인으로 여러 시스템을 쓰는 인증 방식) 서버인 Keycloak과 함께 설치됩니다. Keycloak의 그룹 정보는 15분마다 MindWiki 팀스페이스 멤버로 동기화되므로, 인사 이동으로 그룹이 바뀌면 다음 동기화 때 팀스페이스 권한도 함께 바뀝니다. 관리자가 위키 안에서 사람마다 권한을 다시 고칠 필요가 없고, 퇴사자 계정을 SSO에서 막으면 위키 접근도 함께 끊깁니다.

6.4.2 읽기와 쓰기, 관리로 나눈 API 토큰 범위와 사용 기록

외부 시스템이 MindWiki에 접근하는 방법은 MCP 토큰과 API 토큰 두 가지입니다. API 토큰은 읽기·쓰기·관리 가운데 필요한 범위만 골라 발급하고, 토큰마다 만료일을 두며 사용 기록을 남깁니다. 사내 게시판의 글을 위키로 가져오는 연동에는 쓰기 범위만, 위키 내용으로 보고서를 만드는 연동에는 읽기 범위만 주는 방식입니다. 그래서 감사 시점에는 어떤 토큰이 언제 무엇에 쓰였는지를 사용 기록으로 확인할 수 있습니다.

6.5 PostgreSQL 이중화와 백업 보관 주기 — DB 매일 7일, WAL 2일, 첨부 매일 7일

데이터를 회사 안에 둔다는 것은 가용성과 복구 책임도 회사가 진다는 뜻이며, MindWiki는 이 책임을 설치기와 정해진 백업 작업으로 덜어 줍니다. 모든 문서·행·색인이 PostgreSQL에 모여 있으므로 PostgreSQL의 이중화와 백업만 챙기면 위키 전체의 복구 범위가 정해집니다.

6.5.1 2대 이상에서 설치기가 맡는 PostgreSQL 이중화

서버 한 대에 설치하면 PostgreSQL도 한 벌만 구성되지만, 두 대 이상에 설치하면 설치기가 PostgreSQL을 이중화 구성으로 배포하므로 한 대에 장애가 나도 다른 한 대가 서비스를 이어 갑니다. 운영 담당자가 복제 설정을 직접 작성하지 않아도 서버 대수를 정하는 것만으로 가용성 수준이 정해집니다. 사용자 수와 문서 규모에 맞는 서버 대수는 환경에 따라 달라지므로 설치 전에 함께 산정합니다.

6.5.2 암호화 키까지 포함하는 백업 대상과 보관 주기

백업은 다음 세 가지를 따로 받습니다.

- PostgreSQL 데이터베이스: 매일 한 번 받아 7일 보관
- PostgreSQL WAL(Write-Ahead Log, 변경 내역을 먼저 기록하는 로그): 따로 받아 2일 보관
- 첨부 파일: 매일 한 번 받아 7일 보관

위키의 암호화 키도 백업 대상에 들어갑니다. 암호화 속성의 값은 이 키가 있어야 복호화되므로, 데이터베이스를 복원하더라도 키를 잃으면 암호화 속성은 읽을 수 없는 값으로 남습니다. 따라서 키 백업은 데이터베이스 백업과 다른 저장 위치와 접근 권한으로 관리해야 하며, 이 원칙은 운영 절차서에 처음부터 적어 두는 편이 안전합니다.

7장: 플러그인 SDK와 HWP 변환으로 사내 시스템과 한글 문서를 위키에 연결하는 구조

MindWiki의 색인과 검색은 위키 안에 저장된 데이터에만 작동하므로, 사내 게시판·메신저·결재 시스템에 남아 있는 데이터와 한글(HWP) 파일 속 본문은 그대로는 AI의 근거가 되지 못합니다. 키워드 검색도, 의미가 비슷한 문서를 찾는 벡터 검색도, 속성으로 관계를 잇는 그래프 질의도 위키 바깥의 데이터에는 닿지 않기 때문입니다. MindWiki는 데이터를 위키 안으로 가져오는 수단을 세 가지로 갖췄습니다. Go·Node.js·Python·Java 4개 언어의 플러그인 SDK, HWP/HWPX·DOCX 첨부의 자동 변환과 색인, 그리고 회의 녹음을 회의록과 할 일 카드로 바꾸는 흐름입니다. 여러분의 조직이 연결 비용을 따질 때 필요한 것은 각 수단이 어떤 데이터를 어떤 형태로 위키에 넣는지, 그리고 어디까지가 기본 설치의 범위인지입니다.

7.1 4개 언어 플러그인 SDK로 연결하는 사내 게시판과 메신저, 결재 시스템

고객사의 사내 시스템은 대부분 특정 업체 제품을 바탕으로 따로 구축한 것이어서, 널리 쓰이는 메신저 서비스처럼 공개된 연동 API를 기대하기 어렵습니다. 따라서 MindWiki는 연동 대상마다 기능을 새로 만드는 방식 대신, 고객사 시스템에 맞춘 연결 모듈을 표준 규격으로 작성하게 하는 플러그인 SDK를 제공합니다.

7.1.1 고객사마다 다른 시스템을 표준 방식으로 연결하는 플러그인 구조

플러그인 SDK 4개 언어와 세 가지 연결 유형

플러그인 SDK는 Go·Node.js·Python·Java 네 언어로 제공되므로, 고객사 개발 조직이 이미 쓰는 언어로 연결 모듈을 작성할 수 있습니다. 플러그인이 하는 일은 데이터가 흐르는 방향에 따라 가져오기, 알림, 호출의 세 유형으로 나뉩니다. 예를 들어 사내 민원 게시판의 새 글을 주기적으로 가져와 데이터베이스 행으로 쌓으면, 그 순간부터 그 글은 다른 위키 문서와 똑같이 검색과 AI 처리의 대상이 됩니다. 공공기관 민원 답변 자동화도 이 가져오기 플러그인으로 게시판 글을 위키에 넣는 데서 시작합니다(9장).

연결 유형	데이터 방향	연결 대상 예	위키에서 생기는 결과
가져오기	외부 시스템에서 위키로	사내 게시판 새 글, 외부 자료 스크랩	데이터베이스 행이나 페이지로 저장되어 색인 대상에 포함
알림	위키에서 외부 시스템으로	고객사가 자체 구축한 사내 메신저	자동화 조건이 맞을 때 담당자에게 메시지 전달
호출	위키 버튼에서 외부 시스템으로	결재 시스템의 승인 처리	사용자가 누른 버튼이 외부 시스템의 기능을 실행

AI 에이전트 연결과 업무 시스템 연결의 역할 분담

MindWiki는 외부 연결 방식을 용도에 따라 둘로 나눴습니다. AAS 에이전트가 위키를 읽고 쓰는 연결 방식은 MCP(Model Context Protocol)이며, MCP는 LLM 애플리케이션과 외부 데이터·도구를 표준 방식으로 잇는 공개 프로토콜입니다[S4]. 반면 게시판이나 결재 시스템처럼 사람이 쓰던 업무 시스템은 LLM을 거칠 이유가

없으므로 플러그인 SDK로 직접 연결합니다. 이렇게 나누면 업무 시스템 연동이 모델 동작에 영향을 받지 않고, 연동 개발 범위도 연결 대상 시스템 수에 비례합니다.

7.1.2 위키 버튼이 결재 서버 승인까지 호출하는 플러그인 API

위키 화면의 승인 버튼과 결재 서버 기능의 연결

MindWiki의 버튼은 AI를 호출할 수도 있고 플러그인이 제공하는 API를 호출할 수도 있습니다. 결재 연동 플러그인이 「승인」 함수를 제공하고 위키의 승인 버튼을 그 함수에 연결해 두면, 사용자가 위키에서 버튼을 누를 때 플러그인이 실제 결재 서버의 승인 처리까지 실행합니다. 담당자는 위키 화면 하나에서 근거 문서를 확인하고 결재까지 마칠 수 있고, 자동화 규칙이 이 버튼을 다음 단계로 이어 외부 시스템을 움직이게 만들 수도 있습니다.

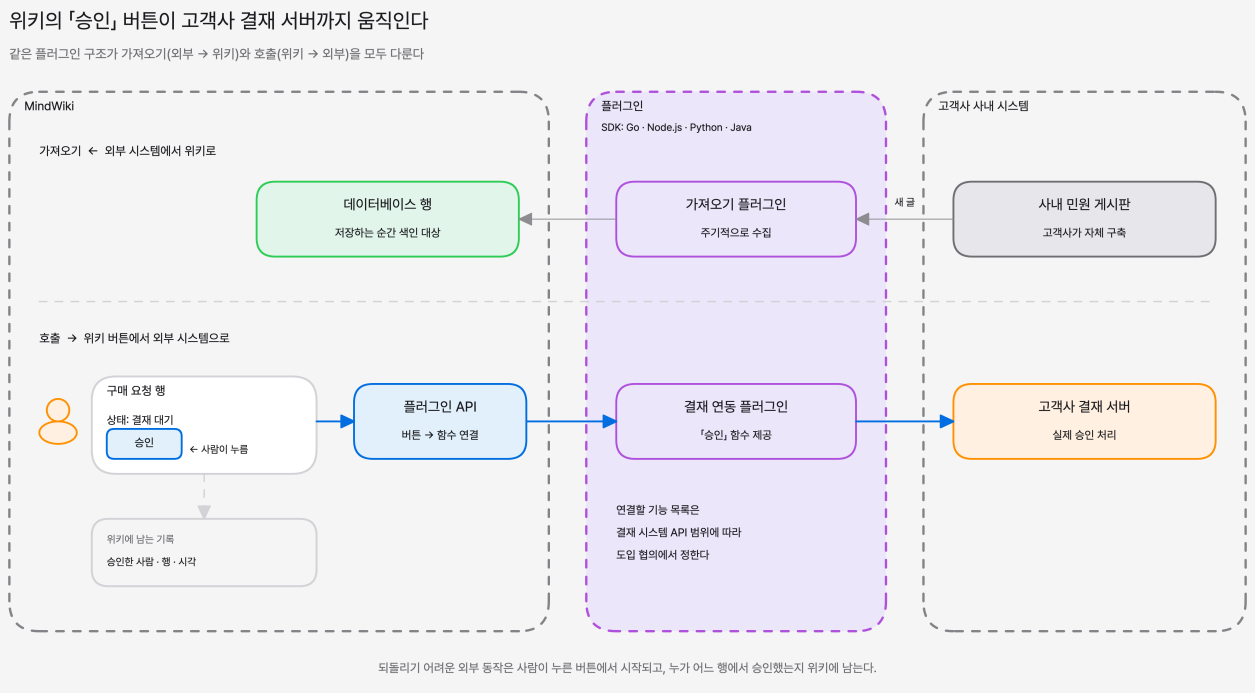

그림. 승인 버튼에서 결재 서버까지의 호출

사람이 시작하는 외부 시스템 동작과 동의 원칙

결재처럼 되돌리기 어려운 동작은 버튼을 누르는 사람의 조작으로 시작되므로, 위키 쪽에서 누가 어떤 행에서 승인했는지가 함께 남습니다. 이 구조는 MCP 명세가 정한 보안 원칙과도 방향이 같습니다. 해당 명세는 사용자가 모든 데이터 접근과 작업에 명시적으로 동의해야 하고, 도구는 임의 코드 실행에 해당하므로 실행 전에 동의를 받아야 한다고 정합니다[S4]. 다만 결재 연동의 세부 동작은 고객사 결재 시스템이 제공하는 API 범위에 따라 달라지므로, 도입 협의 단계에서 연결할 기능 목록을 먼저 정하는 편이 좋습니다.

7.2 한글(HWP/HWPX)과 DOCX 첨부의 원본 보기와 하위 페이지 변환, 자동 색인

국내 기관과 기업의 제안요청서, 보고서, 규정집은 상당수가 한글(HWP/HWPX)이나 DOCX 파일로 남아 있고, 첨부 파일 상태로는 검색도 AI 답변의 근거도 되지 못합니다. MindWiki는 이런 파일을 올리면 원본 보기, 하위 페이지 본문 변환, 자동 색인을 한 번에 처리해 기존 문서 자산을 바로 검색 대상에 넣습니다.

7.2.1 수만 건 문서를 사람이 마크다운으로 바꾸던 작업의 제거

기존 문서를 AI 검색 대상으로 만들던 수작업 단계

지금까지 문서를 AI가 검색하게 하려면 파일을 마크다운으로 바꾸고, 별도 저장소에 올려 임베딩을 만들고, 검색이 제대로 되는지 따로 확인하는 과정을 거쳐야 했습니다. 문서가 열 건이면 감당할 수 있지만 1만 건, 2만 건이 되면 이 작업만으로 이관 일정이 정해지고, 만들어진 임베딩이 검색 품질에 어떤 영향을 주는지도 눈으로 확인하기 어렵습니다. 문서 이관 비용은 AI 도입을 미루게 만드는 가장 현실적인 이유가 되어 왔습니다.

HWP 첨부를 올렸을 때 MindWiki가 처리하는 단계

MindWiki에 한글 파일을 올리면 페이지에 「하위 페이지 변환 중」 표시가 나타나고, 변환이 끝나면 두 가지 결과가 함께 남습니다. 첫째는 원본 파일을 그대로 열어 보는 원본 보기이고, 둘째는 파일 본문을 위키 문서로 옮긴 하위 페이지입니다. 사람은 원본 서식을 확인할 때 원본 보기를 쓰고, 검색과 AI는 변환된 하위 페이지를 읽습니다. 사람이 해야 하는 일은 파일을 올리는 것뿐이며, 문서가 쌓이는 만큼 검색 가능한 지식도 함께 늘어납니다.

7.2.2 변환된 하위 페이지가 키워드·벡터 색인에 바로 들어가는 흐름

변환된 하위 페이지와 일반 위키 문서의 같은 색인 경로

변환된 하위 페이지는 사람이 직접 쓴 위키 문서와 똑같이 저장 시점에 키워드 색인과 벡터 색인에 들어갑니다. 키워드 색인은 PostgreSQL 위에서 BM25 방식에 한국어 형태소 분석을 더해 만들고, 벡터 색인은 pgvector 확장의 HNSW 인덱스로 만듭니다. pgvector는 PostgreSQL에 벡터 유사도 검색을 더하는 오픈소스 확장이고, HNSW는 다층 그래프 구조로 검색 속도 대비 재현율이 좋은 인덱스입니다[S2]. 두 검색 결과는 순위를 기준으로 하나로 합치는 RRF(Reciprocal Rank Fusion, 역순위 융합) 방식으로 묶이므로[S1], HWP에서 온 문단도 다른 문서와 같은 기준으로 순위가 매겨집니다.

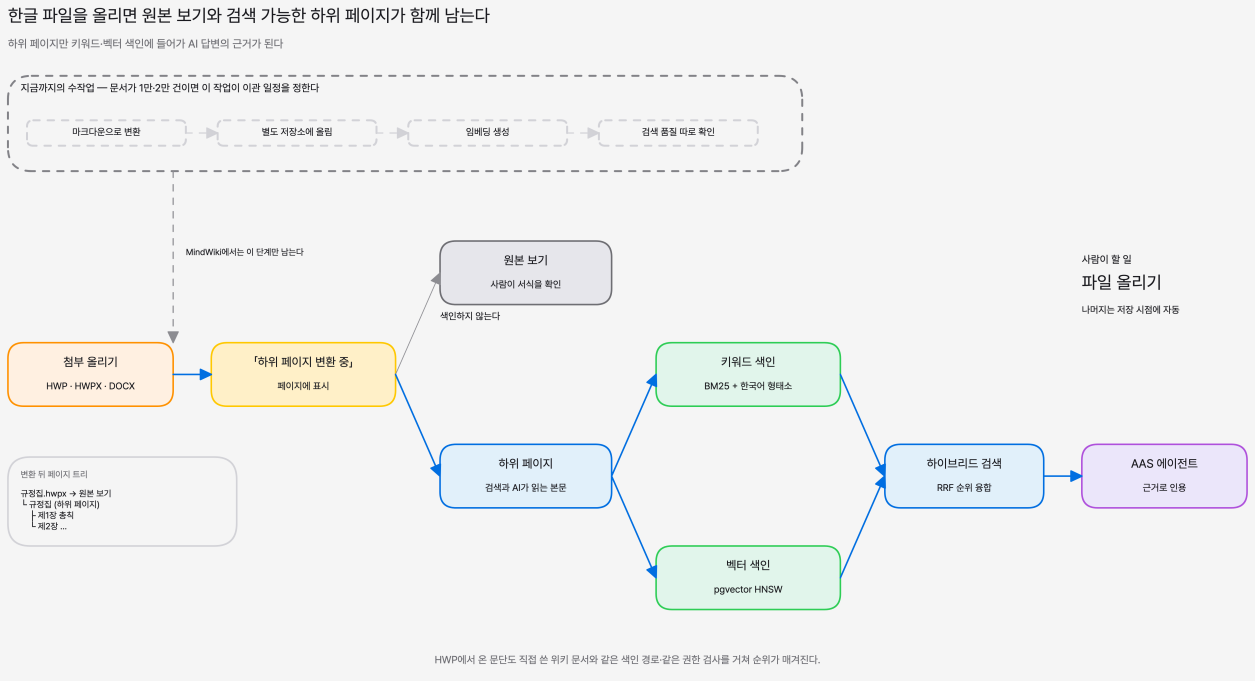

그림. 한글 첨부의 변환과 자동 색인 경로

검색 범위를 정할 때 확인할 권한과 첨부 정책

변환된 하위 페이지도 위키 페이지이므로 검색할 때 다른 페이지와 같은 권한 검사를 거치고, 볼 권한이 없는 사람의 답변 근거로는 쓰이지 않습니다. 다만 대량 이관에서는 첨부 원본과 변환 페이지가 함께 저장되므로, 이관 전에 첨부 저장 용량 계획을 먼저 세워야 합니다. 이관 순서와 저장 계획은 도입 첫 단계에서 함께 정합니다(12장).

7.3 회의 녹음에서 화자 분리 회의록과 칸반 할 일 카드까지 이어지는 흐름

회의는 결정과 할 일이 가장 많이 생기는 업무이지만, 녹음을 받아 적고 회의록을 쓰고 할 일을 칸반에 다시 등록하는 작업은 대부분 사람이 반복해 왔습니다. MindWiki는 이 과정을 녹음, 받아쓰기, 회의록 정리, 칸반 카드 생성의 한 흐름으로 묶어 회의 결과가 곧바로 업무 데이터로 남게 합니다.

7.3.1 녹음 버튼과 녹음 파일 업로드, 화자 분리 받아쓰기

회의 페이지의 녹음 버튼과 파일 업로드 두 가지 입력

회의 페이지에는 녹음 버튼이 있고, 버튼의 이름과 이어지는 동작은 조직의 회의 방식에 맞게 설정할 수 있습니다. 회의실에서 바로 녹음하지 못했다면 따로 녹음한 파일을 올려도 같은 처리를 거칩니다. 받아쓰기는 사내에 설치된 CogentAI 음성 인식 모델이 처리하며, 결과에는 발언자별로 구분한 화자 분리 표시가 붙습니다. 녹음 원본도 페이지에 함께 남으므로, 받아쓴 문장이 의심스러우면 해당 부분을 다시 들어 확인할 수 있습니다.

회의 음성을 사내 모델로 처리하는 이유

회의 녹음에는 고객사 이름, 계약 조건, 직원 개인 정보가 섞이기 쉬워서 외부 음성 인식 서비스로 보내는 순간 데이터 처리 범위가 회사 밖으로 넓어집니다. 개인정보보호위원회가 2025년 8월 6일 「생성형 AI 개발·활용을 위한 개인정보 처리 안내서」를 공개한 것도 이런 처리 경로를 기관이 직접 관리해야 한다는 요구가 커졌기 때문입니다[S8]. MindWiki는 음성 인식까지 사내 모델로 처리하므로 회의 음성과 받아쓴 내용이 회사 안에만 남습니다.

7.3.2 목적과 결정, 할 일로 정리한 회의록과 칸반 카드 생성

받아쓴 내용을 목적·결정·할 일로 나눈 회의록

받아쓴 내용을 회의 페이지에 넣고 회의록 작성을 실행하면, AI가 미리 설정한 작성 지침에 따라 회의 목적, 결정 사항, 할 일을 나눠 정리합니다. 회의록 서식이 항목별로 정해져 있으므로 AI가 자유롭게 문장을 지어내는 대신 정해진 칸을 채우게 되고, 참석자는 결정 사항이 맞는지만 확인하면 됩니다. 예를 들어 회의 지침에 「결정 사항에는 담당자와 기한을 함께 적는다」를 넣어 두면, 이후의 모든 회의록이 같은 형식으로 쌓여 검색과 비교가 쉬워집니다.

회의록의 할 일이 팀 칸반 카드로 이어지는 구조

회의록의 할 일 항목은 버튼 하나로 팀의 칸반 데이터베이스에 카드로 만들어집니다. 회의가 끝난 뒤 누군가 할 일을 다시 옮겨 적다가 빠뜨리는 일이 줄고, 카드마다 출처 회의록이 연결되어 왜 생긴 일인지를 나중에도 추적할 수 있습니다. 회의록과 칸반 카드는 모두 위키 데이터이므로 3장의 색인 경로를 그대로 거쳐, 「지난달 회의에서 이 고객사 관련으로 정한 일」 같은 질문에도 근거를 달아 답할 수 있습니다.

7.4 기본값이 꺼진 외부 연결 선택 기능 — 외부 검색과 지도, 링크 임베드

MindWiki의 기본 설치는 인터넷 연결 없이 동작하도록 만들어졌지만, 조직에 따라서는 외부 웹 검색이나 지도처럼 외부 접속이 필요한 기능을 원하기도 합니다. MindWiki는 이런 기능을 기본값이 꺼진 선택 기능으로 분리해, 필요한 조직만 설치할 때 켜도록 했습니다.

7.4.1 설치 변수로 켜는 외부 검색과 지도 기능

설치 변수로 켜는 선택 기능 세 가지

외부 접속이 필요한 기능은 설치 변수로 하나씩 켜는 방식이며, 현재 제공하는 선택 기능은 다음과 같습니다.

- 외부 검색 — 오픈소스 메타 검색 엔진 SearXNG를 함께 설치해 AI가 위키에 없는 공개 자료를 찾게 하는 기능
- 지도 — 문서 안에 지도를 넣어 위치 정보를 보여 주는 기능
- 링크 임베드 — 외부 웹 페이지 링크를 미리 보기 형태로 문서에 넣는 기능

각 기능은 독립된 변수로 켜고 끄므로, 예를 들어 지도는 쓰되 외부 검색은 허용하지 않는 조합도 가능합니다. 보안 담당자는 이 구조 덕분에 기능 단위로 외부 접속 허용 여부를 판단할 수 있습니다.

7.4.2 인터넷 없이 동작하는 기본 설치와 선택 기능의 경계

기본 설치와 선택 기능을 나누는 외부 접속 기준

기본 설치는 화면 글꼴과 스크립트까지 외부 CDN 없이 불러오고, AI 모델도 사내 CogentAI 모델을 쓰므로 외부로 나가는 연결이 없습니다. 반면 선택 기능을 켜면 그 기능이 쓰는 외부 주소로만 연결이 생기며, 켜지 않은 기능은 외부와 통신하지 않습니다. 6장에서 설명한 폐쇄망 원칙은 기본 설치의 범위로 유지되고, 외부 접속은 조직이 기능별로 명시적으로 허용한 경우에만 생깁니다. 6.1절에서 살펴본 「AI 보안 가이드북」처럼 공공기관의 AI 보안 점검이 강화되는 상황에서는[S7], 이런 구분이 보안 심사 자료를 준비하는 데 직접 쓰입니다. 다만 선택 기능의 허용 범위는 기관의 보안 정책에 따라 달라지므로, 설치 전에 켤 기능 목록을 보안 담당 부서와 함께 정하는 것이 바람직합니다.

8장: 회사 세트로 구성하는 팀스페이스와 공용 데이터베이스 체계 — 영업에서 기술지원까지 이어지는 고객 정보

지식 위키를 조직에 도입할 때 설치보다 오래 걸리는 일은 비어 있는 위키를 팀별 업무 데이터로 채우는 일입니다. 팀마다 어떤 데이터베이스를 두고, 어떤 속성으로 서로 잇고, 누가 무엇을 볼 수 있는지를 정하지 않으면 문서는 다시 팀별로 흩어지고 AI는 근거를 찾지 못합니다. MindWiki는 이 부담을 덜기 위해 회사 업무 전체를 하나의 데이터베이스 체계로 미리 설계한 「회사 세트」를 함께 제공합니다. 이 장에서는 회사 세트의 구성, 공유 규칙 네 단계, 영업에서 기술지원으로 이어지는 고객 정보 흐름, 그리고 운영 중에 세트를 관리하는 원칙을 차례로 설명합니다.

8.1 7개 팀스페이스와 공용 데이터베이스를 한 번에 설치하는 회사 세트 — 자동화 117개와 관계 속성 325개

회사 세트는 한 회사의 부서 구조와 부서 사이의 업무 흐름을 그대로 본떠 미리 설계한 팀스페이스 묶음입니다. 관리자가 설치 화면에서 쓸 팀을 고르면 MindWiki가 팀스페이스, 사용자 그룹, 공용 데이터베이스, 공유 권한을 한 번에 만듭니다. 세트 전체에는 자동화 117개와 관계 속성 325개가 들어 있으며, MindWiki는 이 관계 속성을 4장에서 설명한 지식 그래프의 연결선으로 씁니다.

8.1.1 영업부터 경영지원까지 7개 팀스페이스 구성

팀스페이스 7개의 기본 구성

회사 세트는 영업, 구축, 기술지원, 연구, 마케팅, 인사, 경영지원의 7개 팀스페이스로 구성합니다. 각 팀스페이스에는 그 팀이 날마다 다루는 업무 단위가 데이터베이스로 들어 있습니다. 예를 들어 영업 팀스페이스에는 고객사·담당자·프로젝트·제품·영업 기회·계약·청구 데이터베이스가 있고, 구축 팀스페이스에는 구축 프로젝트와 설치 환경 기록이, 기술지원 팀스페이스에는 문의 접수와 처리 이력을 담는 케이스 데이터베이스가 있습니다. 덕분에 각 팀은 도입 첫날부터 자기 업무를 어디에 적어야 하는지 따로 정하지 않고 바로 입력을 시작할 수 있습니다.

팀 사이를 잇는 관계 속성

7개 팀스페이스는 관계 속성으로 서로 이어져 있으며, 구축 프로젝트의 「고객사」 칸은 영업 팀의 고객사 데이터베이스에서 값을 고르고, 기술지원 케이스의 「도입 제품」 칸은 구축 기록에서 값을 가져옵니다. 이렇게 연결된 속성은 저장하는 순간 그래프 색인에 반영되고, MindWiki는 이 그래프를 PostgreSQL 안의 Apache AGE로 조회합니다[S3]. 여러분의 조직에 영업과 기술지원 사이에 구축 단계가 따로 없다면 해당 팀스페이스를 빼고 설치해도 나머지 팀 사이의 관계 속성은 그대로 남습니다.

8.1.2 예시 행 77개와 그래프 질문 7개가 함께 들어가는 시작 데이터

예시 행 77개로 보여 주는 입력 방식

회사 세트에는 데이터베이스 구조와 함께 실제 업무를 본뜬 예시 행 77개가 들어 있습니다. 가상의 고객사 한 곳이 영업 기회를 거쳐 계약하고, 구축 프로젝트가 만들어지고, 기술지원 문의가 접수되는 과정이 행으로 이어져 있습니다. 현업 담당자는 설명서를 읽기 전에 예시 행을 열어 어떤 칸에 무엇을 적는지 바로 확인할 수 있습니다.

새 도구가 조직에 정착하는 데 걸리는 시간은 대부분 「무엇을 어디에 적는가」를 합의하는 데 쓰이므로, 예시 데이터가 그 합의 시간을 줄여 줍니다.

그래프 질문 7개로 확인하는 연결 상태

예시 행과 함께 들어 있는 그래프 질문 7개는 「이 고객사에 설치한 제품과 버전은 무엇인가」, 「이 엔지니어가 맡은 기술지원 케이스는 어느 프로젝트에 속하는가」처럼 여러 팀의 데이터를 이어야 답이 나오는 질문으로 구성되어 있습니다. 일반 RAG는 흩어진 정보를 공통 속성으로 이어야 하는 질문에 약하다고 알려져 있습니다[S5]. 회사 세트는 관계 속성이 이미 이어져 있으므로, 설치 직후 이 질문을 실행해 보면 데이터 체계가 제대로 연결됐는지 곧바로 검증할 수 있습니다.

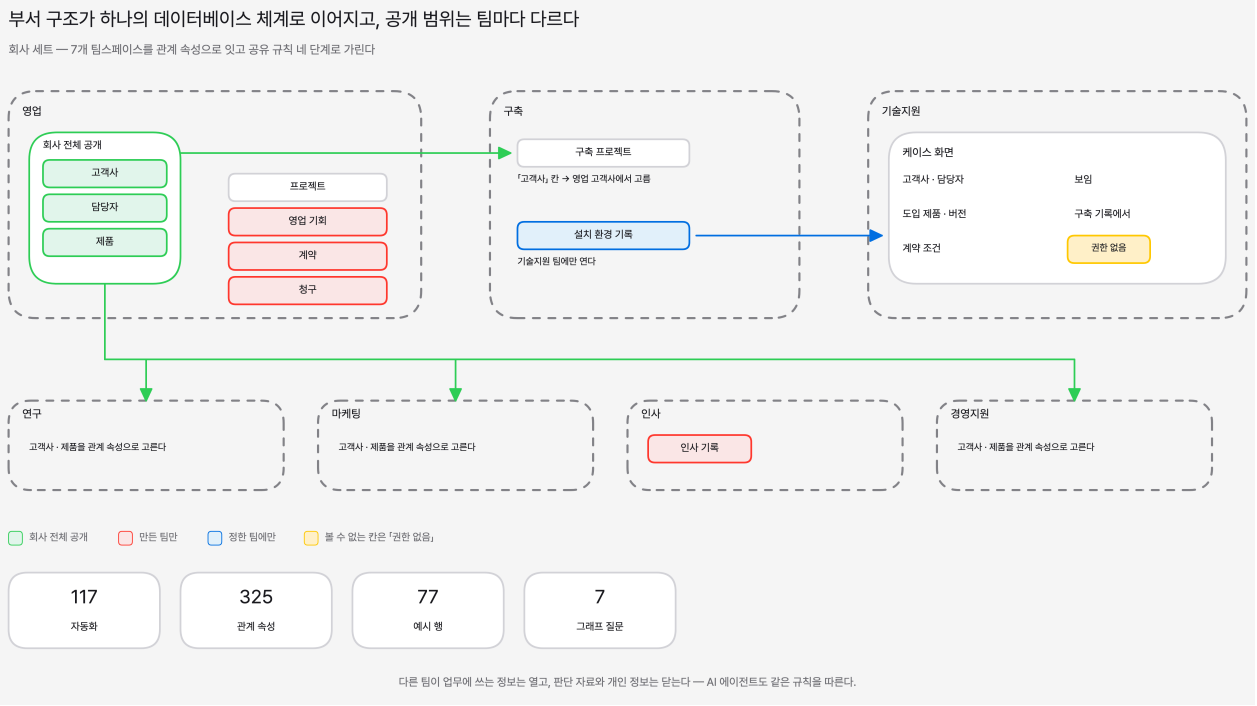

그림. 회사 세트 7개 팀스페이스와 공유 규칙

8.2 공유 규칙 네 단계 — 회사 전체 공개부터 「권한 없음」 칸 표시까지

데이터베이스를 팀 사이에 잇는 순간 함께 정해야 하는 것은 어느 팀이 어떤 데이터와 칸을 볼 수 있는지를 가르는 공개 범위입니다. 고객사 이름은 모든 팀이 알아야 하지만 계약 금액이나 인사 기록은 일부 사람만 봐야 합니다. MindWiki 회사 세트는 이 구분을 공유 규칙 네 단계로 미리 정해 두었고, AI 에이전트도 사용자와 같은 규칙을 따릅니다.

8.2.1 회사 전체가 보는 고객사 정보와 만든 팀만 보는 계약·청구·인사 정보

공유 규칙 네 단계의 기본값

아래 표는 회사 세트가 데이터베이스마다 적용하는 기본 공유 규칙입니다. 관리자는 설치를 마친 뒤에도 데이터베이스 단위로 공유 단계를 바꿀 수 있으며, 바꾼 규칙은 AI 에이전트의 조회 범위에도 같이 적용됩니다.

공유 단계	적용 데이터 예	다른 팀에서 보이는 모습
회사 전체 공개	고객사·담당자·제품	모든 팀이 행을 읽고 관계 속성으로 고를 수 있음
만든 팀만	영업 기회·계약·청구·인사	다른 팀에는 데이터베이스가 보이지 않음
정한 팀에만	구축 기록을 기술지원 팀에만 여는 경우	지정한 팀만 읽기 또는 쓰기 가능
볼 수 없는 칸	연결된 행 가운데 권한 밖의 속성	칸 위치는 남고 값 대신 「권한 없음」 표시

고객사 정보는 공개하고 계약 정보는 닫는 원칙

이 기본값은 실제 업무 흐름에서 각 팀이 어떤 정보를 서로 필요로 하는지를 따져 정한 것입니다. 기술지원 팀은 문의한 회사가 실제 고객사인지, 어떤 프로젝트로 어떤 제품을 도입했는지 확인해야 하므로 고객사와 제품 정보는 회사 전체에 엽니다. 영업 기회·계약·청구는 영업과 경영지원의 판단 자료이고, 인사 정보는 개인정보이므로 만든 팀만 봅니다. 즉 다른 팀이 업무에 쓰는 정보는 열고, 판단과 개인 정보는 닫는다는 한 가지 기준으로 네 단계가 정해져 있습니다.

8.2.2 정한 팀에만 여는 데이터와 볼 수 없는 칸의 처리

정한 팀에만 여는 구축 기록

구축 기록처럼 특정 팀끼리만 이어 쓰는 데이터는 「정한 팀에만」 단계로 엽니다. 구축 팀이 적은 설치 환경과 접속 정보는 기술지원 팀이 유지보수 단계에서 반드시 봐야 하지만, 마케팅이나 인사 팀이 볼 이유는 없습니다. 서버 비밀번호 같은 값은 6장에서 설명한 암호화 속성에 담아 색인과 그래프에서 제외하므로, 공개 범위를 넓혀도 민감한 값이 검색 결과에 섞이지 않습니다.

칸 단위로 가려지는 「권한 없음」 표시

관계 속성으로 다른 팀의 행을 불러오면 그 행의 모든 칸이 같은 권한을 갖지는 않습니다. 기술지원 케이스에서 고객사 행을 열면 회사명과 담당자는 보이지만 계약 조건 칸은 값 대신 「권한 없음」으로 표시됩니다. 칸을 지우지 않고 남겨 두기 때문에 사용자는 정보가 없는 것인지 권한이 없는 것인지를 구분할 수 있습니다. AAS 에이전트가 MindWiki를 조회할 때도 요청한 사용자의 권한이 그대로 적용되는데, 이는 사용자가 모든 데이터 접근을 통제해야 한다는 MCP의 보안 원칙과도 맞습니다[S4].

8.3 영업 수주에서 구축, 기술지원으로 이어지는 고객 정보 흐름

회사 세트의 가치는 한 팀이 입력한 정보가 다음 팀의 업무 화면으로 다시 입력하지 않고 넘어가는 과정에서 가장 잘 드러납니다. 영업이 수주한 고객 정보가 구축 팀의 프로젝트로 이어지고, 구축 팀이 채운 설치 환경이 기술지원 팀의 문의 화면에 나타나는 흐름이 대표 사례입니다. 이 흐름에는 별도 연동 개발이 필요 없고, 회사 세트에 미리 정의된 관계 속성과 자동화만 쓰입니다.

8.3.1 구축팀이 채운 제품과 버전, 설치 환경이 기술지원 문의에 바로 보이는 구조

수주에서 설치 기록까지의 정보 전달

영업 팀이 계약을 마치면 구축 팀은 영업 데이터베이스의 고객사·프로젝트·제품을 골라 구축 프로젝트를 만듭니다. 구축 팀은 설치하면서 제품과 버전, 설치한 서버, 함께 쓰는 데이터베이스와 WAS의 종류·버전, 방문 장소를 프로젝트 행에 채우고, 유지보수 단계에 들어가면 기술지원 팀이 같은 행을 이어받아 씁니다. 따라서 고객사 정보는 영업 단계에서 한 번만 입력되고, 그 뒤 구축과 기술지원 팀은 같은 행을 참조해 일합니다.

버전을 다시 묻지 않는 기술지원 문의

기술지원 문의가 접수되면 케이스 화면에 해당 고객사의 도입 제품과 버전, 설치 환경이 바로 나타납니다. 설치 정보가 정리되어 있지 않은 조직에서는 담당 엔지니어가 고객에게 제품 버전을 다시 묻거나 현장을 다시 방문해야 하고, 고객은 직접 설치해 준 회사가 왜 그것을 모르는지 의문을 갖게 됩니다. 회사 세트는 이 정보를 처음 설치할 때 빠짐없이 적도록 칸을 미리 정해 두었으므로, 문의 접수만으로 고객 환경을 파악할 수 있습니다. AAS 에이전트가 케이스에 답변 초안을 쓸 때도 같은 행에 적힌 제품 버전과 설치 환경을 근거로 삼아 답을 만듭니다.

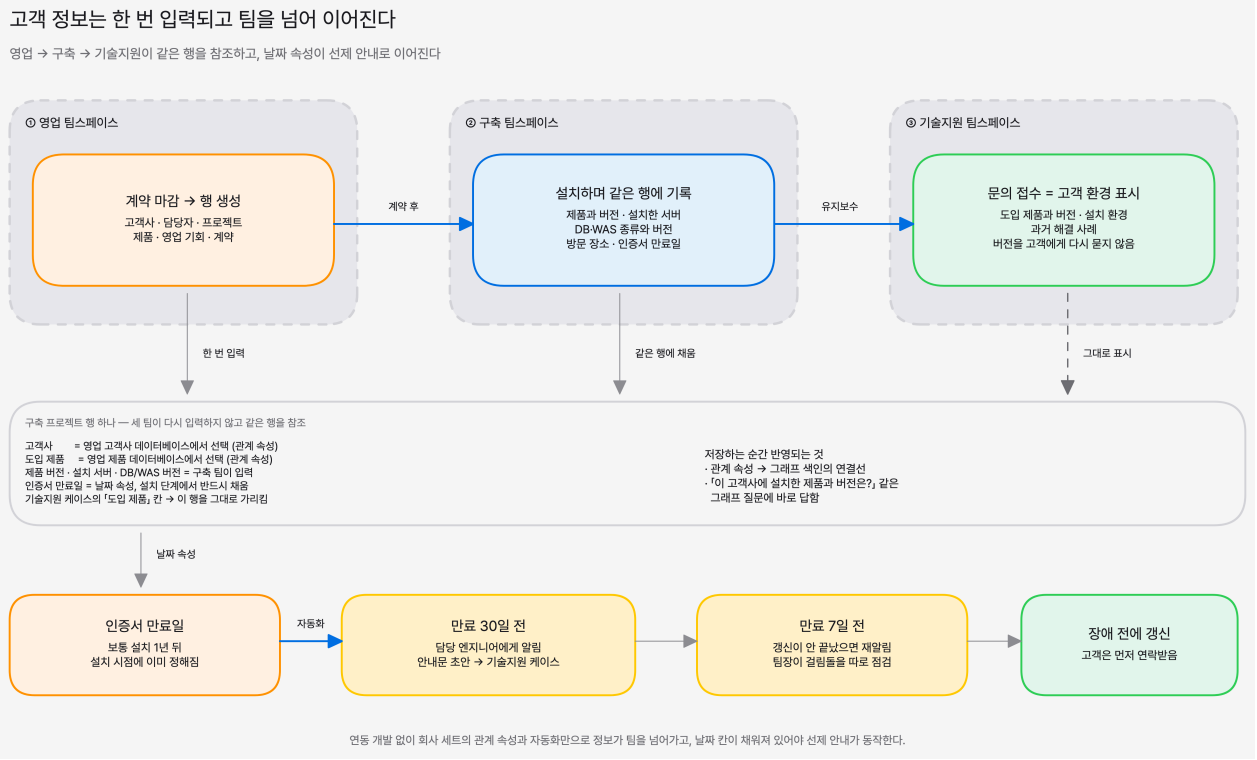

그림. 영업에서 기술지원까지 이어지는 고객 정보

8.3.2 인증서 만료 30일 전 고객 선제 안내

날짜 속성으로 관리하는 인증서 만료일

설치할 때 함께 넣는 인증서는 보통 1년 뒤 만료되며, 그 날짜는 설치 시점에 이미 정해져 있습니다. 그런데도 만료일을 아무도 관리하지 않으면 장애가 난 뒤에야 원인을 알게 되고, 담당자가 바뀐 경우에는 설치 이력을 아는 사람조차 남아 있지 않습니다. 회사 세트는 구축 프로젝트에 인증서 만료일을 날짜 속성으로 두고, 이 값을 설치 단계에서 반드시 채우도록 했습니다.

만료 30일 전에 동작하는 안내 자동화

인증서 만료일 30일 전이 되면 자동화가 담당 엔지니어에게 알림을 보내고, 고객에게 보낼 안내문 초안을 기술지원 케이스로 만듭니다. 안내문에는 만료 예정일, 조치하지 않으면 생길 수 있는 장애, 간단한 갱신 절차가 들어

갑니다. 고객은 장애를 겪기 전에 먼저 연락을 받으며, 이 경험이 서비스를 받고 있다는 신뢰로 이어집니다. 다만 이 자동화는 날짜 칸이 채워져 있어야 동작하므로, 입력 항목을 줄이기보다 고객에게 필요한 칸을 남겨 두는 운영 원칙이 함께 필요합니다.

알림은 한 번으로 끝나지 않습니다. 만료 30일 전 첫 알림 뒤에도 갱신이 끝나지 않았다면 만료 7일 전에 한 번 더 알림이 갑니다. 첫 알림을 받은 담당자가 휴가 중이거나 고객 쪽 일정이 밀려 조치가 늦어지는 경우에도, 만료 직전에 다시 확인할 기회가 남도록 두 번의 시점을 둔 것입니다. 두 번째 알림까지 간 케이스는 갱신 절차나 고객 연락 경로에 걸림돌이 있었다는 신호이므로, 팀장이 따로 모아 원인을 살펴보는 기준으로도 쓸 수 있습니다.

8.4 「데이터베이스는 하나, 보기는 여럿」 원칙과 세트 변경을 반영하는 운영

회사 세트를 오래 쓰려면 업무 데이터를 어떻게 나누고 모을지에 대한 설계 원칙과 세트 갱신 방법을 함께 정해야 합니다. MindWiki는 같은 업무를 하나의 데이터베이스에 모으고 사람마다 보기만 달리하는 원칙을 기본으로 삼습니다. 세트 정의가 바뀌었을 때는 현재 상태와 비교해 빠진 부분만 더하는 「세트와 맞추기」 기능을 제공합니다.

8.4.1 팀원별 데이터베이스 분리가 AI 질의를 늘리는 문제

팀원별 데이터베이스가 늘리는 AI 질의 횟수

칸반 보드를 팀원마다 따로 만들면 사람이 보기에는 편하지만, AI 에게는 질의 횟수가 늘어나는 구조가 됩니다. 예를 들어 「구축 팀은 지금 얼마나 바쁜가」라는 질문에 답하려면 에이전트는 먼저 팀원 명단을 찾고, 팀원 수만큼 데이터베이스를 따로 조회한 뒤 결과를 합쳐야 합니다. 질의 수가 팀원 수에 비례해 늘어나는 만큼 응답 시간과 모델 호출 비용도 함께 늘어납니다. 흩어진 데이터를 잇는 질문일수록 일반 검색 방식이 약해진다는 점도 이 문제를 키웁니다[S5].

하나의 데이터베이스에 사용자별 보기를 두는 설계

회사 세트는 팀의 칸반을 하나의 데이터베이스로 두고, 팀원별 화면은 필터를 건 보기로 만듭니다. 이렇게 하면 같은 질문에 대한 AI 질의는 한 번으로 끝나고, 사람은 자기 보기에서 자기 일만 볼 수 있습니다. 키워드·벡터·그래프 색인도 하나의 데이터베이스를 기준으로 생성하므로, PostgreSQL 전문 검색과 pgvector를 함께 쓰는 하이브리드 검색도 한 범위 안에서 순위를 매깁니다[S2]. 즉 「데이터베이스는 하나, 보기는 여럿」 원칙은 사람의 편의와 AI의 검색 효율을 함께 지키는 데이터 설계 규칙입니다.

8.4.2 빠진 속성과 보기, 자동화만 더하는 세트와 맞추기

세 단계로 끝나는 회사 세트 설치

회사 세트 설치는 쓸 팀을 고르고, 만들어질 팀스페이스와 데이터베이스 목록을 확인한 뒤, 결과를 받는 세 단계로 진행합니다. 시험 환경 기준으로 결과 화면이 나오기까지 약 5초가 걸렸으며, 규모·환경에 따라 달라집니다. 팀스페이스 멤버는 6.4.1에서 설명한 Keycloak 그룹 동기화로 15분마다 맞춰지므로, 인사 이동이 있어도 관리자가 권한을 손으로 고칠 필요가 없습니다.

현재 상태를 유지하며 빠진 부분만 더하는 갱신

운영하다 보면 세트 정의에 새 속성이나 자동화가 추가되고, 현업은 그 사이에 자기 데이터베이스에 칸을 더하거나 보기를 바꿉니다. 「세트와 맞추기」는 기본 세트와 현재 상태를 비교해 빠진 속성·보기·자동화만 더하고, 사용

자가 만든 칸과 이미 쌓인 행은 그대로 둡니다. 새 세트에 「유지보수 종료일」 속성과 그 날짜를 알리는 자동화가 추가됐다면, 맞추기를 실행하면 기존 구축 프로젝트 데이터베이스에 그 두 가지만 생깁니다. 따라서 여러분의 조직은 세트를 처음부터 다시 설치하지 않고도 개선된 업무 체계를 계속 받아 쓸 수 있습니다.

9장: 공공기관 민원 답변 자동화 사례 — 법령 근거 초안과 담당자 승인으로 바뀐 처리 흐름

앞의 여러 장에서 설명한 색인·속성 그래프·AAS 에이전트·플러그인·업무 앱 구성은 하나의 업무 안에서 함께 움직일 때 비로소 효과가 드러납니다. 이 장에서는 주택 청약 민원에 답하는 공공기관 업무를 MindWiki와 AAS 에이전트로 구성한 사례를 따라가면서, 민원이 들어와서 답이 게시되기까지 처리 흐름과 걸리는 시간이 어떻게 바뀌는지 살펴봅니다. 여기에 나오는 수치는 모두 시험 환경에서 시연용으로 구성한 결과이며, 실제 운영 성과를 뜻하지 않습니다.

9.1 근무일 7일 답변 기한과 법령 근거가 요구되는 공공기관 민원 업무의 과제

9.1.1 틀린 답이 소송과 보도로 이어지는 민원 답변의 위험

민원 답변은 빠르게 써야 하면서 동시에 틀리면 안 되는 업무입니다. 이 사례의 기관은 민원을 접수하면 공휴일과 주말을 뺀 근무일 기준 7일 안에 답해야 하고, 답변이 잘못 나가면 손해를 봤다는 소송이 걸리거나 언론 보도로 번질 수 있습니다. 담당자는 질문 하나에 답하기 위해 비슷한 과거 답변을 찾고, 관련 법령 조문을 확인하고, 답변 문구를 다듬는 일을 매번 반복해야 합니다.

시민이 올리는 글의 형태도 담당자의 부담을 키웁니다. 예를 들어 상속받은 주택이 있는데 생애 최초 특별공급에 신청할 수 있는지를 묻는 글은 몇 줄로 끝나기도 하지만, 가족 관계와 재산 내역, 연락처까지 길게 적어 올리는 경우가 많습니다. 담당자는 긴 글에서 핵심 질문을 먼저 골라내야 하고, 글 안에 섞인 전화번호·이메일 같은 개인정보도 업무 시스템에서 따로 관리해야 합니다.

이 업무에서 자동화가 맡아야 할 몫은 답변 속도를 올리는 데서 끝나지 않습니다. 근거가 되는 법령과 기존 답변 방법을 함께 제시해 담당자가 정확성을 빠르게 확인할 수 있게 하고, 기한 계산과 개인정보 처리도 사람의 기억에 맡기지 않도록 업무 흐름 안에 넣어야 합니다. 여러분의 조직에도 정확성과 기한을 함께 지켜야 하는 답변 업무가 있다면, 이 사례의 구조를 그대로 견주어 볼 수 있습니다.

9.1.2 질문 유형별 답변 방법을 정리한 FAQ와 법령 조문 데이터베이스

이 사례에서는 AI 답변의 근거를 MindWiki 안의 데이터베이스 두 개에 담았습니다. 첫째는 청약 민원 FAQ 데이터베이스로, 반복해서 들어오는 질문을 유형별로 묶고 유형마다 「이런 질문에는 이렇게 답한다」는 답변 방법과 참고할 법령, 답변 예시를 행 하나에 담았습니다. 둘째는 법령 조문 데이터베이스로, 관련 법령을 조문 단위의 행으로 나눠 관리하므로 답변 초안이 어느 조문을 근거로 삼았는지 행 단위로 연결할 수 있습니다.

두 데이터베이스는 3장에서 설명한 방식대로 저장하는 순간 키워드·벡터·그래프 색인이 함께 만들어지므로, 별도의 검색 시스템을 구축하지 않아도 에이전트가 바로 찾아 쓸 수 있습니다. FAQ 행이 법령 조문 행을 관계 속성으로 가리키고 있어서, 에이전트는 질문 유형을 판단한 뒤 그 유형에 연결된 조문까지 한 번에 따라갈 수 있습니다. 준비해야 할 근거 데이터는 담당자들이 이미 머릿속이나 개인 파일로 갖고 있던 답변 요령을 표 형태로 옮겨 적은 것이므로, 새로운 형식의 자료를 따로 만들 필요는 크지 않습니다.

9.2 게시판 수집부터 게시판 등록까지 네 단계 민원 처리 흐름

MindWiki는 민원 한 건을 네 단계로 나눠 처리합니다. 아래 표는 단계마다 누가 무엇을 하는지와 그 결과로 남는 것을 정리한 것입니다.

단계	맡는 주체	하는 일	남는 결과
① 가져오기	플러그인	5분마다 민원 게시판의 새 글을 가져와 개인정보를 가린다	「민원」 데이터베이스의 새 행
② 초안 작성	자동화 + AAS 에이전트	FAQ와 법령 조문을 근거로 요지·유형·법령·초안을 쓴다	행 속성에 채워진 답변 초안
③ 검토·승인	담당자	근무일 함수로 계산된 기한 안에 초안을 고치고 승인한다	승인 상태와 확정 답변
④ 등록	자동화 + 플러그인	승인된 답변을 원래 게시판에 올린다	게시판 답변과 처리 이력

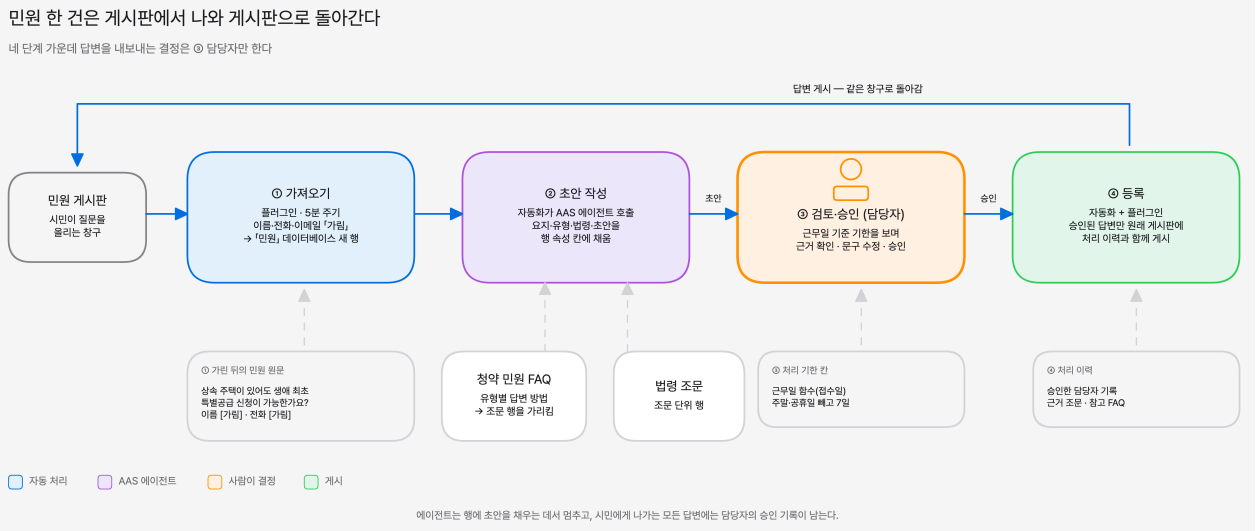

그림. 게시판으로 돌아오는 민원 처리 네 단계

9.2.1 5분 주기 게시판 가져오기와 개인정보 가림 처리

첫 단계는 7장에서 설명한 플러그인 SDK로 만든 게시판 연결 플러그인이 맡습니다. 플러그인은 5분마다 민원 게시판의 새 글을 확인하고, 글을 「민원」 데이터베이스의 행 하나로 만들면서 본문에 들어 있는 이름·전화번호·이메일 같은 개인정보를 「가림」 표시로 바꿉니다. 담당자와 에이전트가 처음 보는 민원 원문에는 질문 내용만 남고, 개인을 알아볼 수 있는 정보는 AI의 처리 대상에서 빠집니다.

이 순서는 공공기관의 생성형 AI 활용 지침과도 맞물립니다. 개인정보보호위원회는 2025년 8월 생성형 AI 개발·활용 단계별 개인정보 처리 안내서를 공개했고[S8], 국가정보원은 2025년 12월 공공기관용 AI 보안 가이드북을 배포했습니다[S7]. 이 사례에서는 개인정보를 가린 뒤에만 AI가 민원을 읽고, 모델도 6장에서 설명한 대로 회사 안에 설치된 CogentAI 모델을 쓰므로 민원 본문이 기관 밖의 AI 서비스로 나가지 않습니다. 다만 어떤 항

목까지 가릴지는 기관의 개인정보 처리 방침에 따라 정해야 하므로, 도입할 때 가림 규칙을 담당 부서와 먼저 맞추는 것이 좋습니다.

9.2.2 AAS 에이전트의 법령 근거 초안과 근무일 함수 기한의 담당자 승인

두 번째 단계에서는 새 민원 행이 생기면 자동화가 AAS 에이전트에게 초안 작성을 맡깁니다. 이때 자동화는 민원 원문과 함께 관련 FAQ 행과 법령 조문 행을 미리 모아 에이전트에게 넘기므로, 에이전트가 근거를 찾으려고 검색을 여러 번 되풀이할 필요가 줄고 모델 호출 횟수도 줄어듭니다. 에이전트는 긴 민원을 몇 줄의 요지로 줄이고, 미리 정해 둔 답변 유형 가운데 하나를 고른 뒤, 관련 법령을 확인하고 시민에게 보낼 답변 초안을 행의 속성 칸에 채웁니다.

세 번째 단계는 담당자의 몫입니다. 처리 기한은 MindWiki의 근무일 함수가 접수일로부터 공휴일과 주말을 빼고 계산하므로, 담당자는 기한이 가까운 민원부터 보기에서 골라 검토할 수 있습니다. 담당자는 초안의 근거 조문과 참고 FAQ를 확인하고, 필요한 문구를 고친 뒤 승인 버튼을 누릅니다. 에이전트는 답변을 직접 게시하지 않고 행에 초안을 채우는 데서 멈추며, 5장에서 설명한 대로 위키에 쓰는 내용도 편집 제안 형태로 남습니다.

이 책임 경계는 MCP 명세가 정한 보안 원칙과 같은 방향입니다. MCP는 사용자가 모든 데이터 접근과 작업에 명시적으로 동의하고 통제권을 가져야 하며, 도구를 실행하기 전에 사용자 동의를 받아야 한다고 정합니다[S4]. 마지막 네 번째 단계에서 자동화와 플러그인이 승인된 답변만 원래 게시판에 등록하므로, 시민에게 나가는 모든 답변에는 담당자의 승인 기록이 함께 남습니다.

9.3 시험 환경 기준 수치 — 처리 상태 7단계와 자동화 24개, 초안 한 건 약 2.5분

시험 환경에서 구성한 민원 업무 앱은 접수부터 게시판 등록까지 처리 상태를 7단계로 나누고, 상태가 바뀔 때마다 다음 일을 시작하는 자동화 24개로 움직입니다. 민원 한 건의 답변 초안이 만들어지기까지는 시험 환경 기준 약 2.5분이 걸렸습니다. 이 시간은 근거 데이터의 양, 검색과 모델 서버의 부하, 동시에 들어온 민원 수에 따라 달라지며, 검색이 늦어지는 조건에서는 몇 분 더 걸리기도 합니다. 도입을 판단할 때는 초안 작성 시간 자체보다, 담당자가 빈 화면에서 근거를 찾던 시간이 근거가 달린 초안을 검토하는 시간으로 바뀐다는 점을 함께 보는 것이 좋습니다.

9.3.1 민원 요지와 답변 유형, 관련 법령, 확신도를 담은 초안 구성

에이전트는 담당자가 검토할 초안을 행 하나에 다섯 가지 정보로 정리합니다. 민원 요지는 긴 원문을 몇 줄로 줄인 것이고, 답변 유형은 미리 정해 둔 분류 가운데 에이전트가 고른 값이며, 관련 법령은 법령 조문 데이터베이스의 행으로 연결됩니다. 여기에 시민에게 보낼 답변 초안과, 에이전트가 이 답을 얼마나 확신하는지 나타내는 확신도가 함께 붙습니다.

이 구성 덕분에 담당자는 초안을 처음부터 다시 읽지 않고 확인할 순서를 정할 수 있습니다. 예를 들어 확신도가 높고 참고 FAQ가 분명한 초안은 근거 조문만 확인하고 승인하고, 확신도가 낮거나 답변 유형이 애매한 초안은 원문을 다시 읽으며 직접 고치는 식입니다. 초안의 모든 근거가 위키 안의 행으로 이어져 있으므로, 나중에 시민이 답변에 이의를 제기하더라도 어떤 조문과 어떤 FAQ를 근거로 답했는지 바로 찾아볼 수 있습니다.

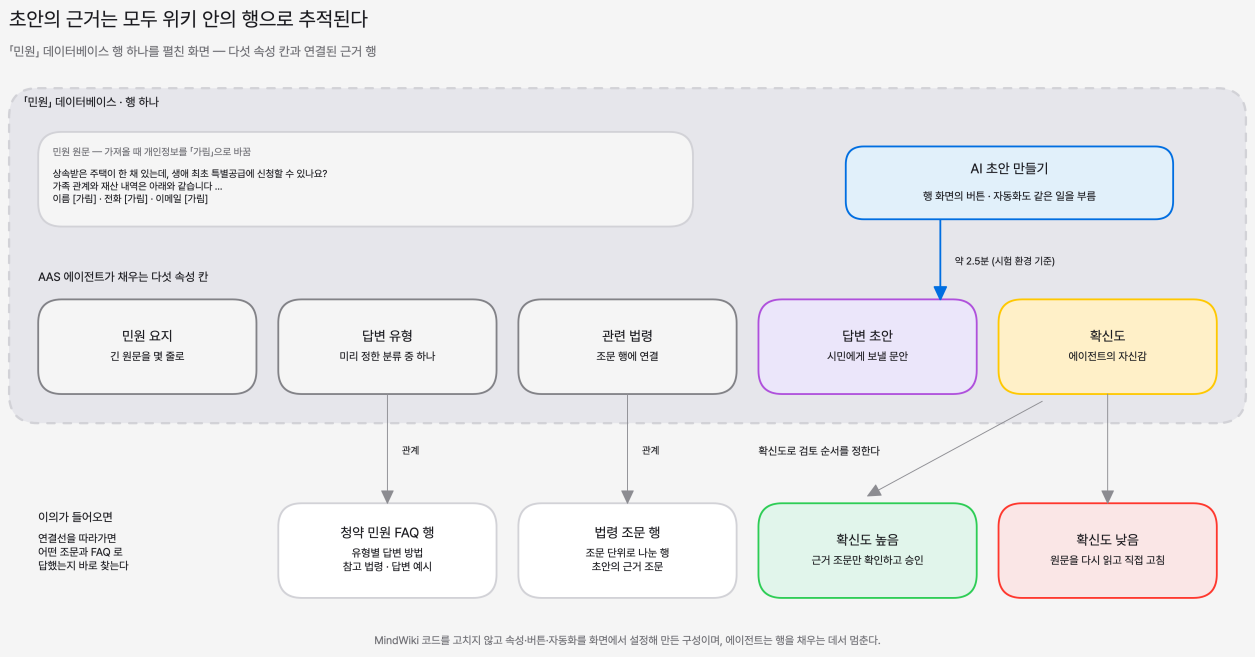

그림. 「민원」 행 하나에 담긴 AI 초안 구성

9.3.2 MindWiki 코드 수정 없이 속성과 버튼, 자동화로 만든 민원 업무 앱

이 민원 업무 앱을 만들면서 MindWiki 제품 코드는 한 줄도 고치지 않았습니다. 「민원」 데이터베이스의 속성, 상태 칸, 「AI 초안 만들기」 버튼, 근무일 함수로 계산하는 기한 칸, 상태 변화에 반응하는 자동화 24개는 모두 화면에서 설정했습니다. 게시판과 주고받는 부분만 플러그인 SDK로 만든 플러그인이 맡았고, 플러그인도 MindWiki 본체와 분리되어 있어 제품을 업그레이드할 때 기관별 수정분을 다시 옮길 필요가 없습니다.

이 점은 업무 앱 구축 비용을 판단할 때 중요합니다. 민원 처리 시스템을 새로 만들 때는 보통 화면 개발, 상태 관리, 알림, 검색 연동을 모두 개발 과제로 잡습니다. 반면 이 사례에서는 현업 담당자가 익숙한 표 형태의 데이터베이스 위에서 속성과 버튼을 정하고, IT 담당자는 게시판 연결 플러그인과 AAS 에이전트의 업무 지침만 준비했습니다. 업무 규칙이 바뀌면 개발 일정을 기다리지 않고 속성이나 자동화를 화면에서 고쳐 바로 반영할 수 있습니다.

9.4 민원 사례를 다른 답변 업무로 넓히는 적용 조건

9.4.1 기술지원 문의 답변과 사내 규정 질의 응답으로의 확장

민원 사례의 구조는 「질문이 들어온다 → 정리된 근거로 초안을 쓴다 → 사람이 승인한다 → 원래 창구로 돌려보낸다」로 요약됩니다. 근거 문서가 있고 답변 책임이 사람에게 있는 답변 업무라면 대부분 이 구조를 그대로 가져다 쓸 수 있습니다. 대표적인 후보는 기술지원 문의 답변과 사내 규정 질의 응답입니다.

기술지원 문의에서는 8장의 회사 세트가 고객사별 제품·버전·설치 환경을 이미 데이터베이스로 갖고 있으므로, 에이전트가 문의 내용과 고객 환경, 과거 해결 사례를 함께 근거로 삼아 초안을 쓸 수 있습니다. 사내 규정 질의는 인사·총무·보안 규정을 조문 단위의 행으로 정리하고, 자주 묻는 질문을 FAQ로 묶으면 민원 사례와 거의 같은 구성이 됩니다. 다만 두 업무 모두 답변을 받는 사람이 다르므로, 플러그인이 연결할 창구와 담당자 승인 단계를 어디에 둘지는 업무마다 따로 정해야 합니다.

회사 세트의 기술지원 팀스페이스에는 이 확장에 쓰는 AI 기능도 이미 들어 있습니다. 케이스 행에서 답변 초안을 만들고, 해결된 케이스의 해결 요약을 쓰고, 문의의 업무 유형을 제안하며, 비슷한 문제를 다룬 지식 문서를 추천합니다. 민원 사례처럼 속성과 버튼, 자동화를 새로 설계하지 않아도 기술지원 문의 답변부터 바로 시험해 볼 수 있다는 뜻입니다.

파일럿 후보를 고를 때는 질문이 반복되는 정도, 근거 문서가 이미 정리되어 있는 정도, 답변이 틀렸을 때의 영향을 함께 봅니다. 질문이 자주 반복되고 근거 문서가 비교적 정리되어 있으며, 승인 단계를 두면 위험을 관리할 수 있는 업무가 첫 파일럿으로 알맞습니다.

9.4.2 정돈된 근거 데이터가 답변 품질을 정하는 조건

이 사례가 보여 주는 가장 중요한 조건은 근거 데이터의 정돈 상태입니다. 검색이 아무리 정교해도, 자료를 아무데나 늘어놓으면 에이전트는 무엇을 먼저 봐야 할지 알 수 없고 답변 품질도 그 이상 올라가지 않습니다. 정돈되지 않은 데이터는 근거가 되지 못한다는 원칙은 MindWiki를 어떤 업무에 쓰든 똑같이 지켜야 합니다.

민원 사례에서는 FAQ에 「이런 질문에는 이 조문을 먼저 찾고 이렇게 답한다」는 우선순위와 예시를 함께 적어 두었습니다. 에이전트는 이 지침을 받아 가장 먼저 확인해야 할 근거를 알게 되고, 검색 결과가 많더라도 담당자가 기대하는 방향의 초안을 씁니다. 따라서 파일럿을 시작하기 전에는 다음 네 가지를 먼저 점검하는 것이 좋습니다.

- 반복되는 질문 유형과 유형별 답변 방법을 FAQ 행으로 정리했는가
- 답변의 근거가 되는 법령·규정·매뉴얼을 조문이나 항목 단위의 행으로 나눴는가
- FAQ와 근거 행을 관계 속성으로 이어 두었는가
- 초안을 승인하고 고칠 담당자와 처리 기한 규칙을 정했는가

이 네 가지가 갖춰지면 에이전트가 쓰는 초안의 품질은 데이터를 정돈한 만큼 올라가고, 담당자가 고친 답변이 다시 FAQ로 쌓이면서 다음 초안의 근거도 함께 늘어납니다. 여러분의 조직에서 민원 답변·기술지원·회의록 가운데 하나를 파일럿으로 고른다면, 그 업무의 근거 데이터가 이 점검표를 얼마나 채우는지부터 확인하는 것이 출발점입니다.

10장: 기존 노트·위키 도구 네 범주와 MindWiki 비교 — 설치 위치와 AI 모델, 지식이 되는 방식의 차이

사내 문서 도구를 새로 고르거나 바꾸려는 조직은 대개 네 범주의 제품을 함께 검토합니다. 공급사 클라우드에서 쓰는 SaaS형 협업 노트 도구, 팀별 스페이스에 문서를 쌓는 엔터프라이즈 협업 위키, 개인 PC의 마크다운 파일로 쓰는 로컬 파일형 마크다운 노트 도구, 사내 서버에 직접 설치하는 오픈소스 셀프호스팅 위키가 그 네 범주입니다. 네 범주 모두 문서를 쓰고 나누는 용도로는 오랫동안 검증된 도구이지만, 회사 문서를 AI 답변의 근거로 쓰려는 요구가 생기면서 평가 기준이 바뀌었습니다. 이 장에서는 열 가지 비교 기준을 정하고 네 범주의 강점과 MindWiki와의 차이를 같은 기준으로 정리해, 여러분의 조직이 검토 중인 대안과 MindWiki를 나란히 놓고 판단할 근거를 제시합니다.

10.1 비교 기준 열 가지 — 설치 위치부터 한국 업무 문서 처리까지

문서 도구 비교표는 흔히 편집 기능 목록으로 채워지는데, AI 근거 저장소로 쓸 도구를 고를 때는 기준을 달리 잡아야 정확한 비교가 가능합니다. 비교 기준은 AI 근거 관점의 네 가지와 운영·보안·조직 도입 관점의 여섯 가지로 나눴습니다.

10.1.1 설치 위치와 AI 모델, 지식이 되는 방식, 관계 질문 기준

설치 위치와 AI 모델의 데이터 주권 기준 설치 위치는 문서 원본과 색인, 검색 기록이 물리적으로 어느 서버에 저장되는지를 뜻하고, AI 모델 기준은 질문과 문서 조각을 받아 답을 만드는 모델이 누구의 인프라에서 실행되는지를 뜻합니다. 두 기준은 따로 보이지만 데이터 주권이라는 하나의 질문으로 묶이는데, 문서를 사내에 두더라도 답을 만드는 모델이 외부 서비스라면 질문할 때마다 관련 문서 조각이 회사 밖으로 전송되기 때문입니다. 국가정보원은 2025년 12월 중앙행정기관과 공공기관에 「AI 보안 가이드북」을 배포했고[S7], 개인정보보호위원회는 2025년 8월 생성형 AI의 개발·활용 단계별 개인정보 처리 기준을 담은 안내서를 공개했습니다[S8]. 따라서 공공기관과 규제 산업의 조직은 편집 기능을 비교하기 전에 이 두 기준부터 확인해야 합니다.

지식화 방식과 관계 질문 처리 기준 지식이 되는 방식은 문서를 저장한 뒤 AI가 검색할 수 있는 상태가 되기까지 무엇이 필요한지를 봅니다. 어떤 제품은 문서를 저장만 하고 검색용 색인을 별도 프로젝트로 구축하게 하며, 어떤 제품은 저장하는 순간 키워드·벡터·그래프 색인을 함께 갱신합니다. 관계 질문 기준은 「이 고객사에 설치된 제품 가운데 지난 분기에 기술지원 케이스가 열린 제품은 무엇인가」처럼 여러 문서와 행에 흩어진 사실을 이어야 답이 나오는 질문을 처리할 수 있는지를 봅니다. 일반적인 RAG(Retrieval-Augmented Generation)는 흩어진 정보를 공통 속성으로 이어야 답이 나오는 질문에 약하다는 연구 결과가 있으므로[S5], 이 기준은 기능 목록보다 실제 업무 질문으로 시험해 보는 편이 정확합니다.

10.1.2 데이터베이스 기능부터 한국 업무 문서까지 나머지 여섯 기준

업무 속성과 에이전트 협업 기준 데이터베이스 기능은 문서 안에 상태·담당자·기한 같은 업무 속성을 두고 보기·필터·수식·관계로 다룰 수 있는지를 봅니다. 업무 데이터가 속성으로 정리돼 있어야 AI가 「답변 기한이 지난 민원」 같은 조건을 정확히 걸러 낼 수 있습니다. 에이전트 협업 기준은 AI가 도구 안의 비서로만 동작하는지, 외부 AI 에이전트가 표준 프로토콜로 문서를 읽고 쓰면서 사람이 결과를 승인하는 구조까지 갖췄는지를 구분합니다. MCP(Model Context Protocol) 명세는 호스트가 도구를 실행하기 전에 사용자 동의를 받도록 정하고 있으므로[S4], 사람의 승인 단계가 설계에 들어 있는지가 이 기준의 핵심입니다.

연결·보안·조직 도입·한글 문서 기준 외부 연결 기준은 게시판·결재·메신저 같은 사내 시스템의 데이터를 위키 안으로 끌어올 수 있는지를 보고, 민감 정보 기준은 주민등록번호나 계약 금액 같은 값을 칼럼 단위로 암호화하고 검색 색인에서 뺄 수 있는지를 봅니다. 조직 도입 기준은 팀스페이스·그룹·권한을 회사 단위로 한 번에 구성할 수 있는지를 따지고, 한국 업무 문서 기준은 한글(HWP/HWPX) 파일 변환과 한국어 형태소 검색을 기본으로 지원하는지를 따집니다. 한국의 공공기관과 기업에는 여전히 한글 파일로 오가는 공문과 보고서가 많기 때문에, 한국 업무 문서 기준을 빼고 비교하면 실제 문서의 상당수가 AI 근거에서 빠집니다.

열 가지 기준으로 네 범주와 MindWiki를 나란히 놓으면 아래 표와 같습니다. 범주 열은 해당 범주 제품들의 일반적인 특성을 적은 것이므로 개별 제품과 판에 따라 다를 수 있고, 각 범주의 강점과 차이는 10.2절부터 차례로 설명합니다.

기준	SaaS형 협업 노트 도구	엔터프라이즈 협업 위키	로컬 파일형 마크다운 노트 도구	오픈소스 셀프호스팅 위키	MindWiki
설치 위치	공급사 클라우드	클라우드 중심(자체 설치판 축소 추세)	개인 PC	사내 서버	MSAP.ai 위 사내 설치, 인터넷 없이 동작
AI 모델	공급사가 계약한 외부 모델	공급사 클라우드의 AI 부가기능	플러그인별 외부 API	없음 또는 외부 연동	사내 CogentAI 모델(LLM·임베딩·재순위·가드레일·음성)
지식이 되는 방식	저장 중심, 내장 AI 검색	문서 저장·전문 검색 중심	파일 단위, RAG 별도 구축	전문 검색 중심	저장하는 순간 키워드·벡터·그래프 색인 갱신
관계 질문	보기·필터·관계 속성	페이지 링크·이슈 연결	문서 간 링크 그래프	페이지 링크	업무 속성 기반 그래프 질의
데이터베이스 기능	보기·수식·관계 속성이 성숙	문서 중심에서 출발해 확장 중	플러그인에 따라 다름	제품에 따라 없거나 단순	수식·관계·롤업·근무일 함수, 행 권한
에이전트 협업	도구 안 AI 비서	공급사 AI 부가기능 안의 비서	개인용 플러그인	제품별 외부 연동	AAS 에이전트가 MCP로 읽고 쓰며 사람이 승인
외부 연결	공개 API·연동 앱	마켓플레이스 앱·같은 공급사 도구	플러그인	API	플러그인 SDK(Go·Node.js·Python·Java)
민감 정보	페이지 권한	스페이스·페이지 권한, 저장 지역 지정	파일 접근 권한	스페이스·페이지 권한	암호화 속성(AES-256-GCM)·색인 제외, 행 접근 규칙

기준	SaaS형 협업 노트 도구	엔터프라이즈 협업 위키	로컬 파일형 마크다운 노트 도구	오픈소스 셀프호스팅 위키	MindWiki
조직 도입	템플릿 복제	스페이스 단위 구성, 성숙한 관리 기능	개인 중심	스페이스 수동 구성	회사 세트 일괄 설치
한국 업무 문서	도입 검토 시 제품별 확인	도입 검토 시 제품별 확인	플러그인별 확인	도입 검토 시 제품별 확인	HWP/HWPX·DOCX 변환·색인, 한국어 형태소 검색

표 10-1. 문서 도구 네 범주와 MindWiki의 기준별 비교(범주의 일반 특성 기준)



그림. 설치 위치와 지식화 방식으로 본 범주 위치도

10.2 SaaS형 협업 노트 도구와 MindWiki — 공급사 클라우드와 회사 안 설치의 차이

SaaS형 협업 노트 도구는 블록 단위 편집기에 문서 데이터베이스를 결합해 문서 작성·업무 관리를 한 화면에서 처리하게 만든 범주로, 설치와 운영 부담 없이 바로 쓸 수 있다는 점 때문에 많은 팀이 처음 고르는 도구입니다. MindWiki도 같은 블록 편집과 데이터베이스 사용법을 따르므로, 두 제품을 가르는 기준은 편집 경험보다 문서와 AI가 동작하는 위치입니다.

10.2.1 공급사 클라우드에서 동작하는 문서 데이터베이스와 내장 AI 비서

블록 편집과 문서 데이터베이스의 성숙도 SaaS형 협업 노트 도구의 강점은 블록 편집, 문서 데이터베이스, 템플릿 공유가 하나의 사용 흐름으로 매끄럽게 이어진다는 점입니다. 표·보드·캘린더 같은 여러 보기를 한 데이터베이스 위에서 바꿔 가며 쓸 수 있고, 관계 속성·수식으로 프로젝트와 담당자를 엮는 기능도 오랜 사용자 요구를 거쳐 다듬어졌습니다. 공개 API·연동 앱이 많아 메신저·일정 도구에 쉽게 이어지며, 도구 안의 AI 비서가 요약이나

초안 작성을 도와줍니다. 데이터 위치에 대한 제약이 없고 사용자 수가 적은 팀이라면 이 범주로도 충분히 업무를 정리할 수 있습니다.

공급사 인프라에서 동작하는 데이터와 모델 다만 이 범주의 문서와 색인, AI 기능은 모두 공급사 클라우드에서 동작하고, AI 비서가 쓰는 모델도 공급사가 계약한 외부 모델입니다. 질문과 문서 조각이 어느 국가의 어떤 모델 서버로 전송되는지는 공급사의 약관과 설정이 정하며, 이용 조직이 직접 통제할 수 있는 범위는 그 안에서 제공되는 선택지로 한정됩니다. 반면 MindWiki는 MSAP.ai 위에 설치하고 인터넷 연결 없이 동작하며, 답변 생성과 임베딩, 재순위, 가드레일까지 사내 CogentAI 모델이 처리합니다. 데이터를 국내 또는 사내에 두어야 하는 요구가 있다면 이 차이가 범주 선택을 결정합니다.

10.2.2 저장 중심 구조와 저장 즉시 지식이 되는 구조의 차이

저장 즉시 갱신되는 세 가지 색인 SaaS형 협업 노트 도구는 문서를 저장하고 내장 AI 검색으로 찾는 구조이며, 색인을 어떻게 만들고 어떤 모델로 검색하는지는 공급사 내부 구현에 속합니다. 이와 달리 MindWiki는 문서 본문과 데이터베이스 행 속성을 저장하는 순간 PostgreSQL 전문 검색 색인과 pgvector 벡터 색인, Apache AGE 그래프를 함께 갱신하는데, 키워드와 벡터 색인은 수 초 안에 반영되고 그래프는 그보다 빠르게 반영됩니다(3.3.2 참조). pgvector는 PostgreSQL 전문 검색과 함께 쓰면 하이브리드 검색이 되는 오픈소스 확장이므로[S2], 검색 근거가 같은 데이터베이스 안에서 권한 규칙과 함께 관리됩니다.

관계 질문과 사내 시스템 연결의 차이 SaaS형 도구는 사용자가 설계한 보기·필터·관계 속성으로 관계를 표현하는 데 강하고, MindWiki는 현업이 정한 업무 속성을 그래프의 연결선으로 삼아 자연어 질문을 그래프 질의로 바꿔 답합니다. 사내 게시판이나 결재 시스템의 데이터를 SaaS형 도구로 가져오려면 그 데이터를 공급사 클라우드로 보내야 하지만, MindWiki는 플러그인 SDK로 회사 안에서 데이터를 끌어옵니다. 블록 편집에 익숙한 사용자는 사용법을 그대로 유지하므로, 전환 이득은 데이터 위치와 지식화 방식에서 나온다고 보는 편이 정확합니다.

10.3 엔터프라이즈 협업 위키와 MindWiki — 스페이스·페이지 트리 중심 상용 위키와 AI 근거 위키의 차이

엔터프라이즈 협업 위키는 기업이 팀별 스페이스와 페이지 트리에 문서를 쌓고 이슈 트래커·개발 도구와 함께 쓰는 상용 기업용 위키 범주로, 대규모 조직에서 가장 오래 쓰여 온 문서 기반입니다. 최근 이 범주가 클라우드 중심으로 옮겨 가면서 설치 위치와 AI 기능의 위치가 도입 판단의 새 변수가 되었습니다.

10.3.1 클라우드판과 데이터센터 설치판, 이슈 트래커 연동과 마켓플레이스 앱 생태계

스페이스 구조와 개발 도구 연동의 강점 엔터프라이즈 협업 위키의 강점은 스페이스와 페이지 트리로 문서를 계층적으로 관리하는 구조, 스페이스·페이지 단위의 권한 관리, 대규모 조직을 위한 관리 기능이 오랜 운영 경험으로 다듬어졌다는 점입니다. 특히 같은 공급사의 이슈 트래커와 코드 저장소에 긴밀하게 연결되어, 요구사항 문서에서 개발 이슈와 배포 기록까지 따라가는 흐름이 개발 조직에 큰 이점을 줍니다. 마켓플레이스의 서드파티 앱으로 다이어그램·승인·보고 기능을 넓힐 수 있어 조직마다 필요한 기능을 골라 붙일 수 있습니다. 이미 이 범주를 깊게 쓰고 있는 개발 조직이라면 전환보다 공존을 먼저 검토하는 편이 합리적입니다.

클라우드 중심 전환과 저장 지역 지정의 범위 한편 이 범주는 클라우드판 중심으로 재편되고 있으며, 대표 제품 가운데에는 자체 설치판의 신규 판매를 끝내고 지원 종료 일정을 공지한 사례가 있습니다. 사내 설치를 유지해야 하는 조직은 선택지가 줄어들고, 클라우드판으로 옮길 때는 저장 지역 지정 기능의 적용 범위를 따져 봐야 합니다. 클라우드판은 한국을 포함한 지역 지정을 지원하지만 사용자 계정 정보, 감사·운영 로그, 일시 캐시 같은 일

부 데이터는 지정 대상에서 빠지는 것으로 공개 안내에 적혀 있습니다. 국내 지역을 지정했다는 사실만으로 데이터 주권을 확보했다고 보기 어렵고, 계정과 로그, AI 처리 경로까지 회사 안에 두는지가 기준이 되어야 합니다.

10.3.2 유료 AI 부가기능에 기대는 검색과 저장 즉시 만들어지는 사내 모델 색인의 차이

AI 부가기능과 확장 앱의 데이터 경로 엔터프라이즈 협업 위키의 AI 검색과 요약은 대개 공급사 클라우드에서 동작하는 부가기능으로 제공되며, 요금제별 월 사용량 한도를 두는 구조입니다. 질문과 문서 내용은 공급사 클라우드의 모델로 전송되고, 마켓플레이스 앱 가운데 외부 서비스로 데이터를 보내는 앱이 있는지도 별도로 점검해야 합니다. 데이터베이스형 속성에 대해서는 공개 사용자 포럼에서 관계·수식 필드가 아직 제한적이라는 의견이 이어지고 있으나, 릴리스에 따라 기능이 빠르게 바뀌므로 최신 판을 기준으로 확인하는 것이 공정합니다. 한국어 형태소 검색과 한글 파일 본문 색인 역시 도입 검토 단계에서 실제 문서로 시연을 받아 판단할 항목입니다.

사내 모델 색인과 속성 그래프의 차이 반면 MindWiki는 AI 기능이 부가기능으로 따로 붙지 않고, 저장하는 순간 사내 CogentAI 모델의 임베딩으로 벡터 색인을 만들고 업무 속성으로 그래프를 갱신합니다. 관계를 묻는 질문은 업무 속성 그래프로 답하는데, 사내 평가 기준으로 관계 질문 정답률이 그래프 질의 도입 전 0에서 도입 후 0.750으로 올랐습니다. AAS 에이전트가 MCP로 위키를 읽고 쓸 때 쓰기는 편집 제안으로 남고 사람이 승인해야 반영되므로, 사용자 동의를 앞세우는 MCP의 보안 원칙과 같은 방향의 설계입니다[S4]. 이슈 트래커 연계가 업무의 중심인 조직이라면 기존 위키를 유지하면서 AI 근거가 필요한 업무부터 MindWiki로 옮기는 단계적 방식도 고려할 수 있습니다.

10.4 로컬 파일형 마크다운 노트 도구와 MindWiki — 개인 파일 묶음과 조직 데이터베이스의 차이

로컬 파일형 마크다운 노트 도구는 개인 PC의 폴더에 마크다운 파일을 쌓고 문서 사이의 링크로 생각을 엮는 범주로, 개인 지식 관리와 AI 작업 메모 정리에 널리 쓰입니다. 이 범주를 조직의 AI 근거 저장소로 넓히려 할 때 문제가 되는 부분은 공유, 권한, 검색 색인입니다.

10.4.1 개인 PC의 마크다운 파일 묶음과 문서 간 링크 그래프

개인 파일 소유와 플러그인 생태계의 강점 로컬 파일형 마크다운 노트 도구의 가장 큰 강점은 모든 문서가 사용자의 PC에 일반 텍스트 파일로 남는다는 점입니다. 특정 공급사의 서비스가 바뀌어도 파일을 그대로 다른 편집기로 열 수 있고, 인터넷 연결 없이도 빠르게 동작합니다. 역링크와 링크 그래프 보기로 문서 사이의 연결을 시각적으로 탐색할 수 있으며, 커뮤니티 플러그인이 많아 개인의 작업 방식에 맞춰 기능을 넓히기 쉽습니다. 개인이 생각을 정리하고 연구 메모를 쌓는 용도로는 지금도 매우 효율적인 도구입니다.

조직 공유와 AI 연결의 한계 다만 조직 단위로 쓰려면 파일 공유를 동기화 서비스나 Git 저장소에 맡겨야 하고, 권한은 파일 접근 권한 수준에 머뭅니다. AI 기능 역시 플러그인마다 각자의 외부 API를 호출하는 방식이 많아, 어떤 문서 조각이 어느 모델로 전송되는지를 조직이 한곳에서 통제하기 어렵습니다. 개인 메모는 이 범주로 유지하고, 조직이 함께 써야 할 지식은 권한과 모델을 한곳에서 관리하는 저장소로 모으는 구성이 현실적입니다.

10.4.2 문서 링크 그래프와 업무 속성 그래프의 차이

링크 그래프와 속성 그래프의 표현력 차이 문서 간 링크 그래프는 「문서 A가 문서 B를 언급한다」는 사실을 보여주지만, 그 연결이 담당 관계인지 설치 관계인지 계약 관계인지는 표현하지 않습니다. 반면 MindWiki의 그래프는 현업이 정한 업무 속성, 예를 들어 고객사·도입 제품·설치 환경·담당 엔지니어 같은 관계 속성이 그대로 연결선이 되므로, 연결마다 의미가 붙어 있습니다. 이 그래프는 PostgreSQL 안에서 그래프 질의를 쓰게 해 주는

Apache AGE 위에 저장되고[S3], 「이 엔지니어가 맡은 고객사 가운데 같은 설치 환경을 쓰는 곳」 같은 질문을 미리 준비한 질의 패턴으로 답합니다. 링크 그래프는 탐색을 돕고, 속성 그래프는 관계를 묻는 질문에 답을 만듭니다.

별도 RAG 구축과 저장 즉시 색인의 차이 로컬 마크다운 파일을 AI 근거로 쓰려면 문서를 조각내고 임베딩을 만들어 벡터 데이터베이스에 넣는 RAG 파이프라인을 따로 구축해야 하고, 파일이 바뀔 때마다 색인을 다시 맞추는 작업도 개발자의 몫이 됩니다. MindWiki는 저장하는 순간 키워드·벡터·그래프 색인을 갱신하고, 검색할 때 페이지 트리 권한과 행 접근 규칙을 순위를 매기기 전에 적용합니다. 이 순서 덕분에 AI가 질문자에게 보이지 않아야 할 문서를 근거로 끌어오는 일을 구조적으로 막을 수 있습니다.

10.5 오픈소스 셀프호스팅 위키와 MindWiki — 전문 검색 중심 위키와 저장 즉시 지식이 되는 위키의 차이

오픈소스 셀프호스팅 위키는 공개된 소스 코드를 받아 사내 서버에 직접 설치하는 위키 범주로, 문서를 회사 안에 둔다는 점에서 MindWiki와 출발점이 같고, 차이는 저장한 문서가 AI 근거로 바뀌는 방식과 업무 데이터를 다루는 깊이에서 생깁니다.

10.5.1 사내 서버 설치와 전문 검색 중심의 문서 위키

사내 설치와 공개 코드의 강점 오픈소스 셀프호스팅 위키의 강점은 문서 원본을 사내 서버에 두면서 소스 코드를 직접 검토할 수 있다는 점입니다. 보안 담당자가 코드와 통신 경로를 확인할 수 있고, 많은 제품이 마크다운 편집기와 전문 검색, SSO(Single Sign-On, 통합 인증) 연동을 기본으로 갖춰 문서 공유 용도로는 가볍고 안정적으로 운영됩니다. 저장 위치의 데이터 주권이라는 측면에서 이 범주는 MindWiki와 같은 장점을 공유합니다.

전문 검색만으로 부족한 의미 검색 하지만 이 범주의 검색은 대부분 키워드가 일치하는 문서를 찾는 전문 검색에 머물러, 같은 뜻을 다른 말로 쓴 문서를 놓치기 쉽습니다. 키워드 검색 점수와 벡터 유사도 점수는 범위가 서로 달라 직접 더할 수 없으므로, 하이브리드 검색은 두 검색이 매긴 순위를 RRF(Reciprocal Rank Fusion, 역순위 융합)로 합치는 방식을 씁니다[S1]. MindWiki는 키워드 후보와 벡터 후보를 RRF로 합친 뒤 재순위 모델로 다시 정렬하며, 사내 평가 기준으로 검색 MRR 0.815를 기록했습니다. 같은 사내 설치형이라도 의미 검색과 재순위가 기본으로 들어 있는지가 AI 답변 품질을 좌우합니다.

10.5.2 데이터베이스와 사내 모델, 회사 세트가 더해지는 차이

데이터베이스 속성과 사내 모델의 기본 제공 오픈소스 셀프호스팅 위키는 문서 공유에 집중한 제품이 많아, 제품에 따라 데이터베이스형 속성이 없거나 단순한 경우가 있고 AI 기능은 외부 모델이나 별도로 구축한 모델에 연결해야 합니다. 반면 MindWiki는 수식·관계·롤업·근무일 함수를 갖춘 데이터베이스와 행 접근 규칙, 칼럼 단위로 값을 암호화하고 색인에서 빼는 암호화 속성을 기본으로 제공합니다. 답변 생성, 임베딩, 재순위, 가드레일, 음성 인식까지 사내 CogentAI 모델이 처리하므로, 오픈소스 위키에 AI를 붙일 때 필요한 모델 서빙과 검색 파이프라인 개발을 따로 할 필요가 없습니다.

회사 세트로 줄어드는 조직 구성 작업 조직 도입 측면에서는 오픈소스 위키가 스페이스와 권한을 관리자가 하나씩 수동으로 구성해야 하는 반면, MindWiki는 팀스페이스·공용 데이터베이스·그룹·권한을 묶은 회사 세트를 한 번에 설치합니다. 다만 MindWiki는 쿠버네티스 플랫폼인 MSAP.ai 위에 사내 모델과 함께 설치하는 구조라 데이터 주권을 얻는 대신 서버와 모델의 운영 책임을 조직이 담당합니다. 그래서 문서 공유만 필요하고 AI 근거 요구가 없는 조직에는 가벼운 오픈소스 위키가 더 적은 운영 부담으로 목적을 달성할 수 있습니다. 결국 네 범주와

MindWiki 가운데 무엇을 고를지는 열 가지 기준 가운데 어느 것이 조직에 필수인지로 정해집니다. 여러분의 조직에서 데이터 주권, 저장 즉시 지식화, 관계 질문, 한국 업무 문서가 모두 필수 기준이라면, 이 네 기준을 한 제품에서 함께 충족하는 선택지는 MindWiki입니다.

11장: MindWiki 도입 기대효과와 다섯 가지 가치 — 데이터 주권부터 조직 단위 즉시 도입까지

MindWiki 도입 효과는 답변 업무부터 보안·통제까지 여덟 개 업무 영역에서 나타나며, 영역마다 효과를 확인할 측정 지표 후보가 정해져 있습니다. 생성형 AI를 업무에 쓰려는 조직은 모델을 고르는 단계보다 근거 데이터를 어디에 어떤 형태로 모을지를 정하는 단계에서 먼저 막히고, 공공 부문에서는 국가정보원과 개인정보보호위원회가 생성형 AI의 보안·개인정보 처리 기준을 잇달아 내놓으면서 도입 심사에서 데이터 처리 위치를 묻기 시작했습니다[S7][S8]. 기대효과도 기능 목록보다 어느 업무가 얼마나 달라지고 그 변화를 무엇으로 잴지를 기준으로 따져야 합니다. 이 장에서는 여덟 영역의 기대효과와 측정 후보를 먼저 정리하고, 그 효과를 만드는 MindWiki의 다섯 가지 가치를 이어서 설명합니다.

11.1 답변 업무와 검색 정확도, 기술지원, 선제 대응 영역의 기대효과

여덟 영역 가운데 답변 업무, 검색 정확도, 기술지원, 선제 대응은 고객이나 시민을 직접 상대하는 업무라서 처리 시간과 응답 품질의 변화가 파일럿 첫 달부터 드러납니다. 아래 표는 여덟 영역 전체의 기대효과와 측정 지표 후보를 한 번에 보여 주며, 표의 지표는 도입 조직이 직접 재야 할 후보이고 확정된 성과 수치는 아닙니다.

영역	기대효과	측정 지표 후보
답변 업무	민원·문의 답변 초안을 근거와 함께 작성하고 담당자는 검토·승인에 집중	초안 작성 시간, 담당자 수정 분량, 기한 내 처리율
검색 정확도	이름·뜻·관계 세 경로로 원하는 문서를 찾음	MRR, 재현율@10, 관계 질문 정답률
기술지원	문의와 함께 고객 환경(제품·버전·설치 환경)이 바로 보임	첫 응답 시간, 재질문 건수
선제 대응	인증서 만료·유지보수 종료를 날짜 속성으로 관리해 미리 안내	사전 안내 건수, 만료로 생긴 장애 건수
회의·협업	녹음에서 회의록, 할 일 카드까지 반복 작업 없이 이어짐	회의록 작성 시간
문서 자산화	한글(HWP/HWPX)·DOCX 첨부가 올리는 즉시 검색 대상	변환·색인 문서 수, 근거 인용 횟수
AI 답변 품질	위키에 근거가 쌓일수록 에이전트 답이 나아지는 지식 순환	근거 인용률, 승인률
보안·통제	문서와 AI 처리가 회사 밖으로 나가지 않음	외부 전송 0건, 가드레일 차단 기록

표 11-1. MindWiki 기대효과 8개 영역과 측정 지표 후보(실제 효과는 문서 규모와 운영 환경에 따라 달라집니다)

11.1.1 민원·문의 답변 초안 자동 작성과 담당자 검토 중심의 업무 전환

답변 업무에서 가장 큰 변화는 담당자의 일이 초안 작성에서 검토와 승인으로 옮겨 가는 것입니다. 민원이 게시판에 올라오면 플러그인이 글을 가져와 개인정보를 가린 뒤 데이터베이스 행으로 만들고, 자동화 규칙이 AAS 에이전트에게 초안 작성을 맡깁니다. 에이전트는 FAQ와 법령 조문 데이터베이스를 근거로 민원 요지, 답변 유형, 관련 조문, 답변 초안을 행에 채우며, 시험 환경 기준으로 초안 한 건에 약 2.5분이 걸렸습니다. 처리 기한은 근무일 함수가 공휴일을 빼고 계산하므로 담당자가 기한을 따로 셀 필요가 없습니다.

다만 초안 작성 시간이 그대로 업무 시간 절감으로 이어지지는 않습니다. 담당자가 근거 조문을 확인하고 문구를 고치는 시간이 남고, 그 시간의 길이는 FAQ와 조문 데이터의 정돈 수준이 좌우합니다. 파일럿에서는 초안 작성 시간과 함께 담당자의 수정 분량과 기한 내 처리율을 측정 후보로 두는 편이 정확합니다. 근거가 붙은 초안은 담당자가 어느 조문을 보고 판단했는지를 기록으로 남기므로, 답변 내용에 대한 이의 제기나 감사에 대응할 자료도 함께 쌓입니다.

11.1.2 이름과 뜻, 관계로 찾는 검색과 고객 환경이 바로 보이는 기술지원

MindWiki는 문서를 저장할 때 키워드·벡터·그래프 색인을 함께 만들기 때문에 같은 질문을 이름으로도, 뜻으로도, 관계로도 찾습니다. 사내 평가 기준으로 하이브리드 검색과 재순위를 거친 MRR은 0.815, 키워드와 벡터를 합친 재현율@10은 0.810이었고, 그래프 질의를 더한 뒤 관계를 묻는 질문의 정답률은 0에서 0.750으로 올랐습니다(3.5절·4.4절 참조). 키워드 순위와 벡터 순위를 점수 대신 순위로 합치는 RRF(Reciprocal Rank Fusion, 역순위 융합)는 공개 검색 서비스 문서에도 정리된 표준 기법입니다[S1]. 도입 조직은 같은 평가 방식을 자기 문서로 반복해 우리 데이터에서 정확도가 어느 정도인지를 먼저 확인할 수 있습니다.

기술지원에서는 문의가 들어오는 즉시 고객사의 도입 제품, 버전, 설치 환경이 같은 화면에 나타납니다. 구축팀이 프로젝트 행에 채운 속성이 기술지원 케이스와 관계 속성으로 이어져 있기 때문입니다. 엔지니어가 고객에게 버전과 환경을 다시 묻는 일이 줄어드는 것도 이 연결 덕분이며, 첫 응답 시간과 재질문 건수가 그 효과를 확인할 측정 후보입니다. 같은 구조로 인증서 만료일과 유지보수 종료일을 날짜 속성으로 관리하면 회사 세트의 자동화가 만료 30일 전에 고객 안내를 시작합니다. 즉 장애가 난 뒤 대응하던 일을 일정에 맞춰 미리 처리하는 일로 바꿀 수 있고, 사전 안내 건수와 만료로 생긴 장애 건수를 비교 지표로 삼을 수 있습니다.

11.2 회의와 문서 자산, AI 답변 품질, 보안 영역의 기대효과와 측정 지표 후보

회의·협업, 문서 자산화, AI 답변 품질, 보안·통제의 네 영역은 하루 단위 처리 시간보다 몇 달에 걸친 지식 축적으로 효과가 나타납니다. 파일럿을 시작하는 주에 기준값을 기록해 두어야 나중에 변화를 비교할 수 있으며, 아래 지표는 도입 첫 주부터 함께 측정한다는 전제로 고른 후보입니다.

11.2.1 회의록 작성 시간과 변환·색인 문서 수

회의 영역에서는 녹음 버튼이나 녹음 파일 업로드 한 번으로 화자 분리 받아쓰기, 목적·결정 사항·할 일을 담은 회의록, 할 일별 칸반 카드가 차례로 만들어집니다. 그동안 회의가 끝난 뒤 메모를 다시 정리하고 할 일을 따로 등록하던 반복 작업이 없어지므로, 회의 한 건당 회의록 작성 시간이 가장 직접적인 측정 후보입니다. 회의록도 저장하는 순간 색인되기 때문에 지난 회의에서 무엇을 결정했는지를 나중에 질문으로 찾을 수 있습니다.

문서 자산 영역의 변화는 한글(HWP/HWPX)과 DOCX 첨부에서 두드러집니다. 파일을 올리면 원본 보기와 함께 본문이 하위 페이지로 변환되고 자동으로 색인되므로, 수만 건의 문서를 사람이 마크다운으로 바꾸고 임베딩하던 사전 작업이 필요 없습니다. 예를 들어 과거 제안요청서와 보고서를 첨부로 올리기만 해도 올린 직후부터

AI 답변의 근거가 될 수 있습니다. 변환·색인된 문서 수와 그 문서가 답변 근거로 인용된 횟수를 함께 보면 문서가 자산으로 바뀌는 속도를 알 수 있습니다.

11.2.2 근거 인용률과 승인률, 외부 전송 0건과 차단 기록

AI 답변 품질은 답에 근거가 붙은 비율과 담당자가 초안을 그대로 또는 조금 고쳐 승인한 비율로 잴 수 있습니다. MindWiki의 검색은 근거 문서를 찾지 못하면 모델을 호출하지 않고 근거가 없다고 답하며, 근거를 찾으면 인용 번호를 단 답을 돌려주므로 두 지표를 시스템 기록에서 바로 뽑을 수 있습니다. 에이전트가 만든 결과물은 승인을 거쳐 다시 위키에 저장되므로, 승인률이 오를수록 다음 답의 근거도 늘어나는 지식 순환이 생깁니다.

보안·통제 영역의 지표는 외부 전송 0건과 가드레일 차단 기록입니다. 문서, 색인, AI 모델이 모두 회사 쿠버네티스 플랫폼인 MSAP.ai가 실행과 이중화를 담당하는 사내 설치 환경에서 동작하므로 업무 문서를 외부 AI 서비스로 보내는 경로가 기본 구성에 없고, 이 사실은 네트워크 기록으로 확인할 수 있습니다. 가드레일이 대량 개인정보 추출 질문을 몇 번 막았는지, 암호화 속성 열람을 누구에게 몇 번 허용했는지도 운영 기록으로 남습니다. 즉 보안 영역의 지표는 효과 측정값인 동시에 보안 심사 때 그대로 제출할 수 있는 근거 자료입니다.

11.3 MindWiki 다섯 가지 가치 중 데이터 주권과 문서가 곧 AI 자산이 되는 구조

MindWiki의 다섯 가지 가치는 여덟 영역의 효과가 왜 생기는지를 설명하는 원리이며, 의사결정권자가 도입 근거를 내부에 보고할 때 그대로 쓸 수 있는 요약입니다. 다섯 가지는 다음과 같습니다.

- 데이터 주권: 문서와 AI 모델이 모두 회사 안에 있음
- 문서가 곧 AI 자산: 별도 RAG 구축 없이 평소 문서 작업이 근거를 쌓음
- 현업이 설계하는 지식 그래프: 업무 담당자가 정한 속성이 곧 관계
- 코딩 없는 업무 자동화: 속성·버튼·자동화·에이전트 조합으로 업무 앱 구성
- 조직 단위 즉시 도입: 회사 세트로 팀 구조·공용 데이터·권한을 한 번에 설치

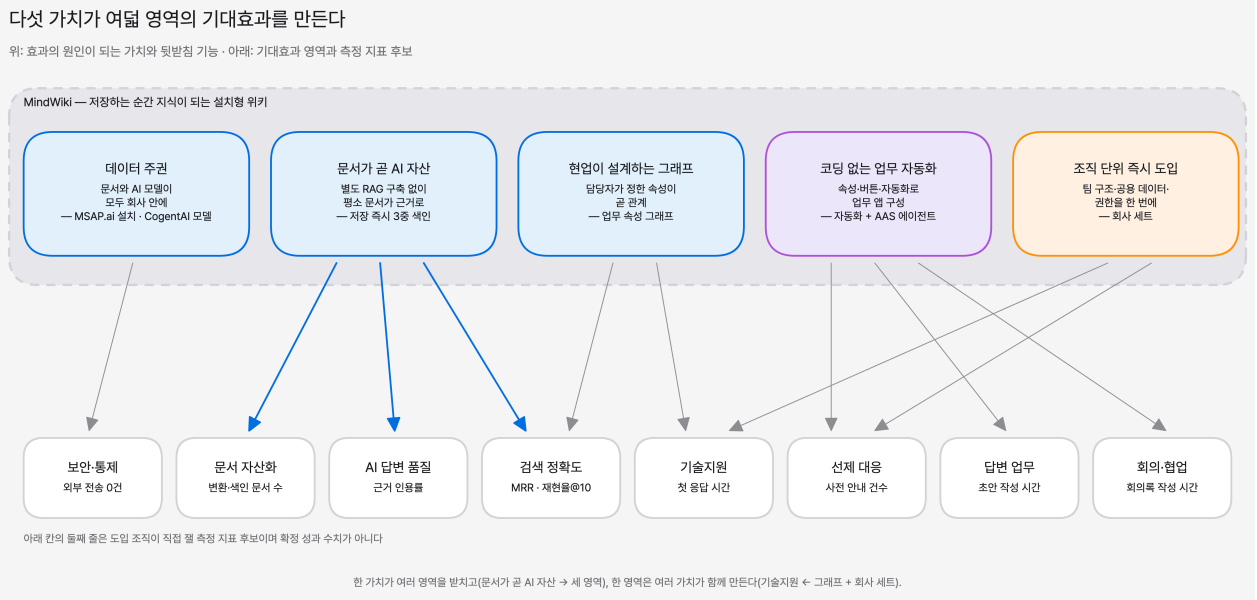

그림. 다섯 가치와 여덟 기대효과 영역의 연결

11.3.1 문서와 AI 모델이 모두 회사 안에 있는 데이터 주권

데이터 주권은 업무 문서와 그 문서를 읽는 AI 모델이 모두 회사가 관리하는 인프라 안에 있다는 뜻입니다. MindWiki는 MSAP.ai 위에 설치되어 인터넷 연결 없이 동작하고, 화면도 외부 CDN이나 웹 글꼴 없이 표시됩니다. 대화와 요약에 쓰는 LLM, 임베딩, 재순위, 가드레일, 음성 인식·합성 모델도 사내 CogentAI 모델을 쓰기 때문에 질문과 문서가 외부 모델 공급사로 넘어가는 경로가 없습니다.

이 가치가 지금 중요한 이유는 국내 기관이 생성형 AI 활용 기준을 구체적으로 내놓기 시작했다는 데 있습니다. 6.1절에서 살펴본 국가정보원의 AI 보안 가이드북과 개인정보보호위원회의 개인정보 처리 안내서가 그 예입니다[S7][S8]. 외부 클라우드 AI를 쓰는 도구는 도입할 때마다 데이터가 어디로 나가는지를 따로 설명해야 하지만, 사내 설치형은 그 설명 대상 자체가 줄어듭니다. 게다가 칼럼 단위 AES-256-GCM으로 저장하는 암호화 속성은 검색 색인과 그래프에서도 빠지므로, 민감 정보가 AI 검색 결과로 노출될 가능성도 구조적으로 막습니다.

11.3.2 별도 RAG 구축 없이 평소 문서 작업으로 쌓이는 AI 근거

문서가 곧 AI 자산이 된다는 가치는 별도 RAG(Retrieval-Augmented Generation) 구축 프로젝트가 필요 없다는 데서 나옵니다. 일반적인 RAG 도입은 문서 수집·형식 변환·청킹·임베딩·로드로 이어지는 파이프라인을 개발자가 따로 만들어야 하고, 원본 문서가 바뀔 때마다 다시 맞춰야 합니다. 반면 MindWiki는 본문과 행 속성을 원본으로 두고, 저장하는 순간 PostgreSQL 안에서 키워드 색인, pgvector 벡터 색인, Apache AGE 그래프를 함께 갱신합니다[S2][S3]. 따라서 평소의 문서 작업이 그대로 AI 근거를 쌓는 일이 됩니다.

이 구조는 투자 판단의 단위도 바꿉니다. 구축 프로젝트형 RAG는 착수 전에 대상 문서 범위와 개발 기간을 정해야 하지만, MindWiki 에서는 한 팀이 한 업무의 문서를 쓰기 시작한 날부터 근거가 쌓이므로 작은 범위로 시작해 효과를 보고 넓힐 수 있습니다. 일반 RAG가 흩어진 정보를 공통 속성으로 이어야 답이 나오는 질문에 약하다는 점은 공개 연구에서도 지적되었는데[S5], MindWiki는 그 약점을 다음 절의 업무 속성 그래프로 보완합니다.

11.4 현업이 설계하는 지식 그래프와 코딩 없는 업무 자동화, 조직 단위 즉시 도입

나머지 세 가치는 지식 체계를 누가 만들고 얼마나 빨리 조직 전체로 넓히는지와 관련됩니다. 세 가치 모두 개발자 일정에 묶이지 않고 업무 담당자가 화면 설정으로 운영한다는 공통점이 있으며, 그래서 도입 뒤 확산 속도를 결정하는 요인이 됩니다.

11.4.1 개발자 없이 업무 담당자가 정하는 지식 그래프

문서 더미에서 LLM으로 노드와 관계를 추출해 지식 그래프를 만드는 방식은 문서가 많을수록 모델 호출 비용이 커지고, 업무의 뜻을 모르는 채 관계를 뽑기 때문에 품질을 보장하기 어렵습니다. 반면 MindWiki는 업무 담당자가 데이터베이스를 설계하며 정한 속성, 예를 들어 고객사·도입 제품·버전·설치 환경·담당 엔지니어를 그대로 그래프의 연결로 씁니다. 업무에 실제로 필요한 속성만 관계가 되므로 그래프가 처음부터 업무 질문에 맞는 모양을 갖춥니다.

그래서 「이 케이스와 같은 제품을 도입한 고객사의 다른 케이스」처럼 관계를 묻는 질문에도 답할 수 있으며, 사내 평가 기준 그래프 질의 p50은 2.9ms였습니다. AI가 본문을 읽고 비어 있는 속성 값을 제안하면 사람이 받아들이는 방식으로 그래프를 채우므로, 데이터가 쌓일수록 관계도 촘촘해집니다. 즉 그래프의 설계 권한과 품질 책임을 업무를 가장 잘 아는 현업이 가진다는 것이 이 가치의 핵심입니다.

11.4.2 코딩 없는 업무 앱과 회사 세트의 조직 단위 도입

코딩 없는 업무 자동화는 속성, 버튼, 자동화 규칙, AAS 에이전트를 조합해 업무 앱을 만드는 방식입니다. 공공기관 민원 답변 업무 앱은 MindWiki 코드를 한 줄도 고치지 않고 화면 설정만으로 만들었고, 게시판 연결처럼 고객사마다 다른 시스템은 Go·Node.js·Python·Java를 지원하는 플러그인 SDK로 붙였습니다. 따라서 새 업무를 자동화할 때마다 개발 요청서를 쓰고 일정을 기다리던 과정이 짧아집니다.

조직 단위 즉시 도입은 8장에서 설명한 회사 세트가 담당합니다. 일곱 개 팀스페이스와 공용 데이터베이스, 공유 규칙을 세 단계 설치로 한 번에 들이고, 세트가 바뀌면 맞추기 기능으로 기존 데이터를 유지한 채 구조를 갱신합니다. 여러분의 조직이 한 팀의 파일럿에서 효과를 확인했다면, 같은 데이터베이스 체계를 다른 팀으로 넓히는 데 별도 설계 기간이 거의 들지 않습니다.

12장: 세 단계 설치와 데이터 정돈 원칙으로 시작하는 MindWiki 파일럿 — 첫 업무 선정 기준

11장까지 MindWiki가 무엇을 하고 어떤 효과를 내는지 설명했다면, 이제 남은 질문은 어디서부터 손을 대야 효과가 빨리 보이느냐입니다. 이 장은 그 순서를 설치, 기존 문서 이관, 데이터 정돈, 첫 업무 선정의 네 단계로 정리합니다. 설치 자체는 세 단계로 끝나지만, 첨부 파일 정책과 데이터 정돈 원칙을 먼저 정하지 않으면 AI가 쓸 근거가 쌓이지 않습니다. 이 장을 다 읽으면 여러분의 조직이 민원 답변, 기술지원, 회의록 가운데 어떤 업무 하나로 파일럿 범위를 정할지 판단할 수 있습니다.

12.1 세 단계 설치와 회사 세트 맞추기 — 팀 고르기, 확인, 결과 약 5초

MindWiki 설치는 회사 세트에서 쓸 팀을 고르고, 설치될 팀스페이스와 공용 데이터베이스를 확인한 뒤, 결과를 받는 세 단계로 이루어집니다(8.4.2 참조). 시험 환경 기준으로 마지막 단계의 결과는 약 5초 만에 나오지만, 설치 버튼을 누르기 전에 인프라와 계정 체계를 준비해 두어야 이 속도가 실제 도입 일정으로 이어집니다.

12.1.1 MSAP.ai 위 설치와 SSO 그룹 권한 매핑

MindWiki는 회사의 쿠버네티스 플랫폼인 MSAP.ai 위에 설치하며, 인터넷 연결 없이 동작합니다. 사내 SSO인 Keycloak을 함께 설치하고, Keycloak 그룹을 15분마다 팀스페이스 멤버로 맞추므로(6.4.1 참조), IT 담당자가 처음 할 일은 영업 그룹은 영업 팀스페이스에, 기술지원 그룹은 기술지원 팀스페이스에 붙이는 권한 매핑입니다. 사용자가 처음 접속하면 계정 정보가 들어오고, 관리자가 그 사람이 맞는지 한 번 확인하면 이후에는 그룹 권한대로 쓰게 됩니다.

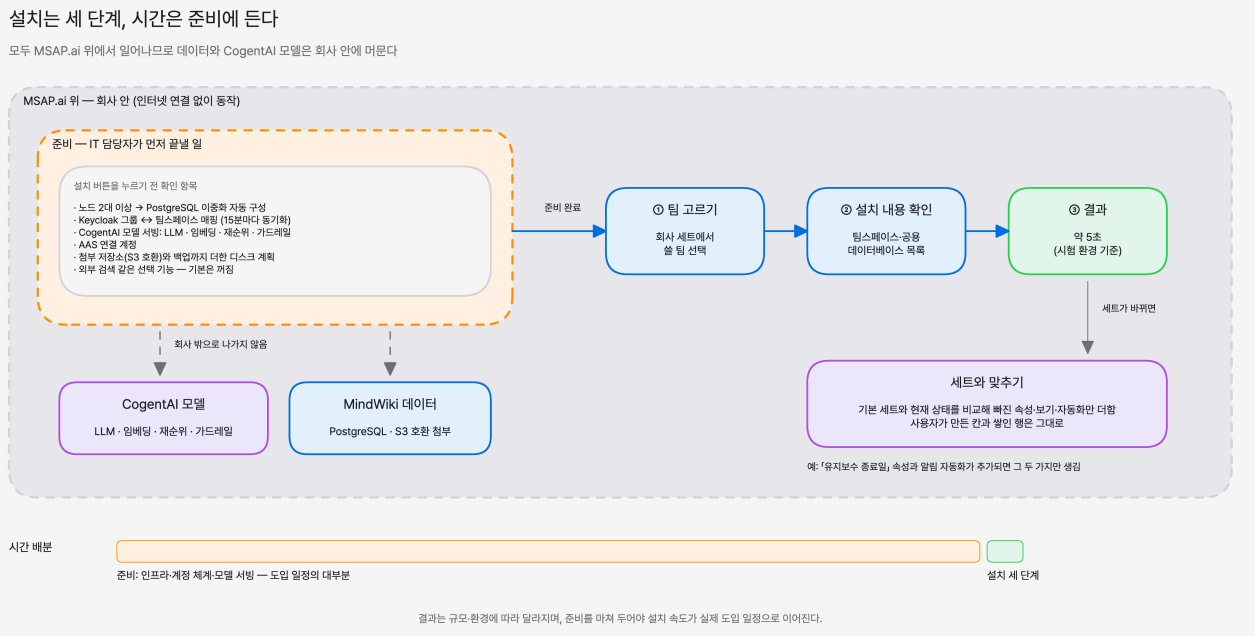

그림. 준비와 세 단계 설치, 세트와 맞추기

회사 세트가 나중에 바뀌어도 처음부터 다시 설치할 필요 없이 「세트와 맞추기」를 실행하면 이미 쓰고 있는 데이터는 그대로 두고 빠진 속성, 보기, 자동화만 더합니다. 설치 전에 IT 담당자가 확인할 항목은 다음과 같습니다.

- MSAP.ai 노드 수: 2대 이상이면 설치기가 PostgreSQL 이중화를 자동으로 구성합니다
- Keycloak 그룹 구조와 팀스페이스의 대응 관계
- CogentAI 모델 서빙 위치(LLM, 임베딩, 재순위, 가드레일)와 AAS 연결 계정
- 외부 검색 같은 선택 기능을 켤지 여부(기본은 꺼짐) — 도입 협의 시 확인

12.1.2 첨부 정책과 디스크 용량 계획을 먼저 세우는 운영 준비

첨부 파일은 S3 호환 저장소에 두며 고객사가 이미 쓰는 스토리지를 연결할 수도 있는데, 여기서 놓치기 쉬운 점은 백업이 용량을 몇 배로 키운다는 사실입니다. 6.5.2에서 정리한 대로 데이터베이스와 첨부 파일은 매일 백업해 7일, WAL은 따로 2일 보관합니다. 지금 공유 드라이브나 파일 서버에 쌓인 파일을 그대로 올리면 원본에 백업분까지 더한 용량이 필요합니다.

벡터 검색에 쓰는 pgvector의 HNSW 인덱스도 검색 속도와 재현율이 좋은 대신 인덱스를 만드는 시간과 메모리가 더 들기 때문에[S2], 문서 양이 늘어날수록 저장 용량과 함께 데이터베이스 메모리도 계획에 넣어야 합니다. 첨부를 모두 위키에 올릴지, 일부는 기존 저장소에 두고 링크만 걸지는 팀마다 업무 방식이 달라 한 번에 정하기 어렵습니다. 팀별로 운영 방식을 정하되, AI가 근거로 써야 하는 한글(HWP/HWPX)·DOCX 문서는 위키에 올리고, 영상이나 대용량 설계 파일은 기존 저장소에 두는 기준에서 출발하기를 권합니다.

- 팀별 첨부 운영 방식(위키 업로드 또는 링크) 결정
- 원본 용량 × 백업 보관 기간으로 계산한 디스크 계획
- 문서 수 증가에 따른 데이터베이스 메모리 계획
- 암호화 키 백업 담당자 지정 — 키를 잃으면 암호화 속성 값을 되살릴 수 없습니다

12.2 기존 문서 이관 — 사람이 옮기는 부분과 관리 API·관리자 스킬로 옮기는 부분

기존 위키와 노트 도구에 쌓인 문서는 한꺼번에 기계로 옮기기 어렵습니다. 조직 구조와 권한은 사람이 판단해야 하고, 문서 본문과 첨부는 도구로 옮길 수 있기 때문입니다. 이관 계획은 사람이 맡을 일과 도구가 맡을 일을 먼저 나누는 데서 시작합니다.

12.2.1 사람이 옮기는 문서와 관리 API로 옮기는 문서의 구분

부서를 합치거나 권한 그룹을 새로 짜는 일은 시스템이 대신 정할 수 없습니다. 예를 들어 두 부서가 하나로 합쳐지면, 합친 부서 이름으로 권한을 다시 매핑하는 일은 기존 도구를 관리하던 사람이 직접 해야 합니다. 반면 위키 안에서 데이터베이스를 다른 팀스페이스로 옮기는 일은 화면에서 끌어서 놓으면 끝납니다. 조직 구조에 관한 결정은 사람이, 반복적인 문서 복사는 도구가 맡는다고 보면 됩니다.

도구 쪽은 두 갈래로 나뉘는데, 에이전트가 쓰는 MCP에는 읽기 도구 19개와 쓰기 도구 17개만 있고 지우기 도구는 없습니다. 팀 생성, 계정 삭제, 팀스페이스 삭제 같은 관리 작업은 MCP에 넣지 않고 별도의 REST 관리 API와 관리자 스킬로 분리했습니다. MCP 규격도 사용자가 모든 데이터 접근과 작업에 명시적으로 동의해야 하며, 도구는 임의 코드 실행과 같으므로 신중히 다뤄야 한다고 정합니다[S4]. 이 분리 덕분에 일상 업무를 돕는 에이전트가 실수로 팀스페이스를 통째로 지우는 사고를 구조적으로 막습니다.

12.2.2 관리자 스킬과 한글 문서 변환을 쓰는 일괄 이관

대량 이관은 관리자 스킬을 붙인 에이전트가 관리 API로 페이지와 데이터베이스를 만들고, 첨부 문서는 MindWiki가 스스로 변환하는 흐름으로 진행합니다. 한글(HWP/HWPX)이나 DOCX 파일을 올리면 원본 보기

를 유지한 채 본문을 하위 페이지로 변환하고, 저장하는 순간 키워드·벡터·그래프 색인까지 만듭니다. 이 변환 덕분에 1만 건, 2만 건의 문서를 사람이 마크다운으로 바꾸고 임베딩을 따로 돌리던 작업이 사라집니다.

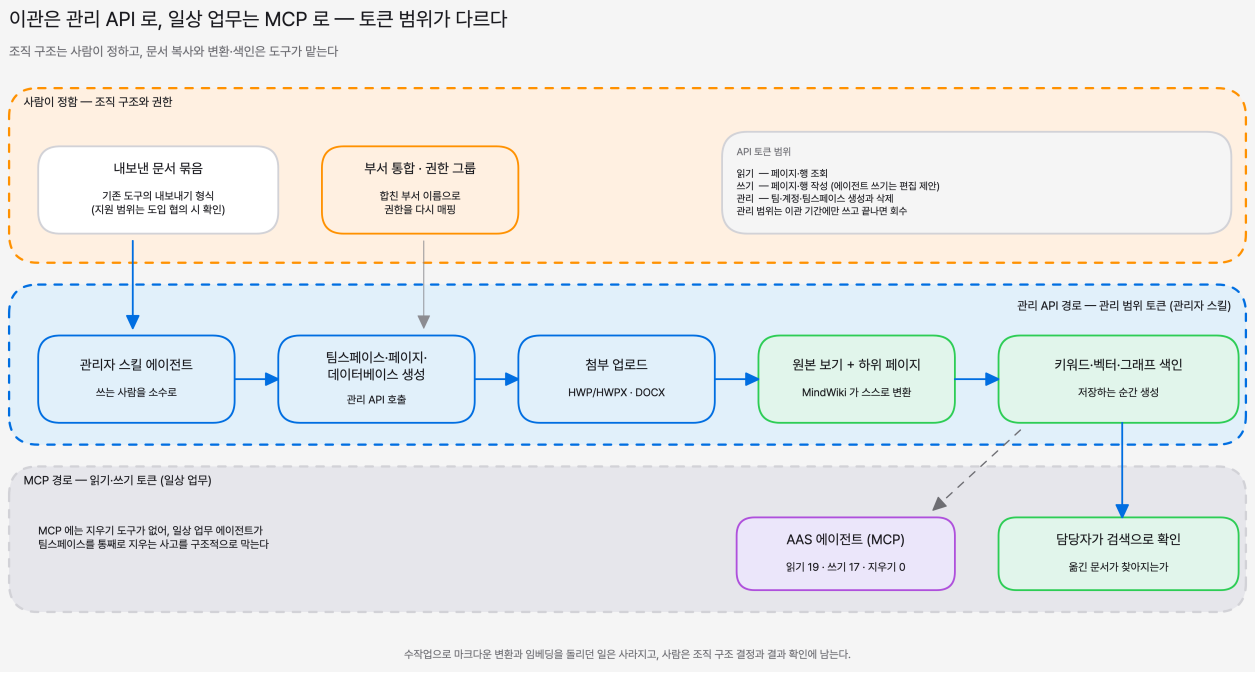

그림. 관리 API와 MCP로 나뉜 문서 이관 흐름

다만 관리자 스킬은 사용자, 워크스페이스, 팀스페이스까지 지울 수 있으므로 쓰는 사람을 소수로 정해야 합니다. API 토큰은 읽기, 쓰기, 관리 범위로 나눠 발급할 수 있으니, 이관 기간에만 관리 범위 토큰을 쓰고 이관이 끝나면 회수하는 방식이 안전합니다. 기존 도구별 내보내기 형식과 이관 도구의 지원 범위는 도입 협의 시 확인합니다.

12.3 데이터 정돈 원칙 — 질문 유형별 답 방법을 정리한 FAQ와 하나의 데이터베이스

문서를 많이 옮겼다고 AI 답변이 저절로 좋아지지는 않습니다. 정돈되지 않은 데이터는 근거가 되지 못하고, 아무 데나 늘어놓은 문서는 아무리 잘 검색해도 어떤 문서가 기준인지 알려 주지 못합니다. 파일럿 전에는 다음 두 가지 정돈 원칙을 정해 둘 것을 권합니다.

12.3.1 질문 유형별 답 방법을 정리한 FAQ가 높이는 AI 답변 품질

MindWiki의 검색은 키워드 검색과 벡터 검색 결과를 RRF로 합친 뒤 재순위를 거쳐 근거를 고릅니다. RRF는 서로 범위가 다른 점수 대신 순위를 합쳐 하나의 목록을 만드는 방식입니다[S1]. 이 방식은 관련 문서를 잘 찾아 주지만, 찾은 문서 가운데 어느 것이 조직의 공식 답변 기준인지는 판단하지 못합니다. 따라서 「이런 질문에는 이렇게 답한다」를 적어 둔 FAQ가 있어야 AI가 그 문서를 가장 먼저 근거로 삼습니다.

공공기관 민원 사례에서도 질문 유형별 답변 방법과 법령 근거를 정리한 FAQ 데이터베이스가 답변 초안의 품질을 좌우했습니다. FAQ 한 건에는 미리 정한 분류 가운데 하나인 질문 유형, 그 유형에 답하는 방법과 답변 예시, 관련 법령 조문이나 사내 규정 링크, 마지막으로 검토한 담당자와 날짜를 속성으로 두면 충분합니다.

12.3.2 속성 설계와 하나의 데이터베이스 원칙으로 시작하는 정돈

두 번째 원칙은 「데이터베이스는 하나, 보기는 여럿」이며, 그 이유는 다음과 같습니다. 팀마다 고객 목록을 따로 만들면 같은 고객이 세 군데에 다른 이름으로 남고, 그래프는 그 셋을 서로 다른 고객으로 봅니다. 일반 RAG는 흩어진 정보를 공통 속성으로 이어야 답이 나오는 질문에 약합니다[S5]. 그래서 MindWiki는 담당자가 정한 관계 속성을 그대로 그래프의 연결선으로 씁니다. 고객사와 제품을 자유 입력 글자 대신 공용 데이터베이스를 가리키는 관계 속성으로 만들어야 관계를 묻는 질문에 답할 수 있습니다.

속성을 설계할 때는 암호화 속성도 함께 정합니다. 암호화 속성은 칼럼 단위로 암호화하고 색인과 그래프에서 빠지므로, 검색해야 하는 값과 숨겨야 하는 값을 미리 나눠야 합니다.

- 고객사·제품·담당자는 공용 데이터베이스를 가리키는 관계 속성으로 만든다
- 같은 대상을 담는 데이터베이스는 하나만 두고 팀별 차이는 보기로 나눈다
- 주민등록번호처럼 숨길 값만 암호화 속성으로 지정한다
- 빈 속성은 AI가 값을 제안하고 담당자가 수락하는 방식으로 채운다

12.4 민원 답변과 기술지원, 회의록 중 첫 업무 하나로 정하는 파일럿 범위

파일럿은 업무 하나로 시작해야 효과를 재기 쉬운데, 업무를 여러 개 한꺼번에 맡기면 결과가 나빠도 데이터 정돈이 부족했는지, 자동화 설계가 잘못됐는지 가려내기 어렵습니다. 시험 환경에서 이미 흐름을 확인한 민원 답변, 기술지원, 회의록 가운데 하나를 고르기를 권합니다.

12.4.1 근거 문서 양과 답변 기한, 승인 단계로 보는 파일럿 업무 선정 기준

세 후보는 근거 문서가 얼마나 정돈돼 있는지, 답변 기한이 얼마나 촉박한지, 사람이 어디서 승인하는지에 따라 준비 부담이 다르며, 아래 표에서 그 차이를 나란히 비교할 수 있습니다.

기준	민원 답변	기술지원	회의록
필요한 근거 데이터	질문 유형별 FAQ, 법령 조문 데이터베이스	고객사·제품·버전·설치 환경, 과거 기술지원 케이스	회의 녹음 파일, 참석자 정보
답변 기한	기관 규정에 따름(사례 기관은 근무일 7일)	고객 계약의 응답 기준에 따름	회의 직후
사람의 승인 단계	담당자가 초안을 검토·승인한 뒤 게시판 등록	엔지니어가 답변 초안을 검토한 뒤 회신	회의 주관자가 회의록과 할 일 카드를 확인
필요한 연결	게시판 플러그인(5분 주기 가져오기, 개인정보 가림)	회사 세트의 기술지원 팀스페이스	받아쓰기와 화자 분리, 칸반 데이터베이스
시작 전 준비 부담	FAQ 정돈이 가장 큼	고객 환경 속성 입력	가장 적음

근거 데이터가 이미 정돈돼 있고 외부로 나가기 전에 사람이 반드시 승인하는 업무일수록 파일럿에 알맞습니다. 공공기관이라면 국가정보원이 2025년 12월 배포한 AI 보안 가이드북[S7]과 개인정보보호위원회가 2025년 8월 공개한 생성형 AI 개인정보 처리 안내서[S8]도 함께 놓고 업무를 고르기를 권합니다. 민원처럼 개인정보가 섞이는 업무는 가림 처리와 승인 단계를 파일럿 범위 안에 처음부터 넣어야 하기 때문입니다. 반대로 데이터 정돈에 시간을 쓰기 어렵다면 준비 부담이 가장 적은 회의록에서 시작해 할 일 카드가 쌓이는 흐름부터 확인하는 방법도 있습니다.

세 후보 밖의 업무를 고를 때는 에이전트에게 맡기기 좋은 업무의 네 가지 조건으로 걸러 볼 수 있습니다. 첫째, 매일 또는 매주 같은 형태로 되풀이됩니다. 둘째, 검색·문서·메신저·업무 시스템 조회처럼 에이전트가 쓸 수 있는 도구로 처리됩니다. 셋째, 결과물을 사람이 보고 맞는지 틀린지 판단할 수 있습니다. 넷째, 다루는 비밀 정보의 범위가 작아 조회 권한만으로 일을 끝낼 수 있습니다. 네 조건을 모두 갖춘 업무는 승인 단계를 두기 쉽고, 파일럿 결과가 나빴을 때 원인을 데이터 정돈과 자동화 설계 가운데 어디에서 찾을지도 분명합니다.

12.4.2 파일럿 성공 판단 지표와 도입 협의 때 확인할 항목

파일럿을 시작하기 전에 성공을 무엇으로 판단할지 정해 두어야 나중에 확대 여부를 숫자로 설명할 수 있습니다. 11장에서 다룬 지표 후보를 그대로 쓰되, 성과로 단정하지 않고 측정 후보로 두고 파일럿 전후를 비교합니다. 예를 들어 민원 답변은 시험 환경 기준으로 초안 한 건에 약 2.5분이 걸렸으므로, 실제 업무에서 초안 작성 시간과 담당자가 고친 비율을 함께 재면 되며, 결과는 문서 양과 업무 환경에 따라 달라집니다.

업무별 측정 후보로는 민원 답변의 초안 작성 시간·담당자 수정 비율·기한 안 처리 비율, 기술지원의 첫 응답 시간과 고객에게 다시 물은 건수, 회의록의 작성 시간과 할 일 카드 누락 건수를 들 수 있습니다.

파일럿 범위를 확정하기 전에 다음 항목은 도입 협의 시 확인합니다.

- 문서 수정 이력, 감사 로그, 보존 기간 정책
- 기존 도구에서 옮길 때 쓸 이관 도구와 절차
- 사내 LLM을 다른 모델로 바꿀 수 있는 범위
- 데이터베이스 행이 1만 건을 넘을 때의 성능
- 문서번호 자동 부여와 회사 서식 PDF 내보내기

이제 남은 결정은 여러분의 조직에서 민원 답변, 기술지원, 회의록 가운데 근거 데이터가 가장 잘 정돈된 업무 하나를 고르고, 그 업무를 MindWiki와 AAS 에이전트로 처리할 파일럿 범위를 정하는 일입니다. 범위가 정해지면 세 단계 설치와 권한 매핑은 짧은 기간에 끝나고, 그 뒤로는 저장하는 순간 쌓이는 문서가 AI 답변의 근거가 됩니다.

부록 — 참고문헌과 본문 용어 풀이

참고문헌

본문의 [S#] 표기는 아래 번호와 같습니다. 공개된 외부 자료만 실었습니다.

- [S1] Microsoft Learn. (2026-11-09 갱신). 「Hybrid search scoring (RRF)」. https://learn.microsoft.com/en-us/azure/search/hybrid-search-ranking (접속 2026-11-27)
- [S2] pgvector 프로젝트. 「pgvector — Open-source vector similarity search for Postgres」(README). https://github.com/pgvector/pgvector (접속 2026-11-27)
- [S3] Apache Software Foundation. 「Apache AGE — Graph database for PostgreSQL」. https://age.apache.org/ (접속 2026-11-27)
- [S4] Model Context Protocol 프로젝트. (2025-06-18). 「Model Context Protocol Specification」. https://modelcontextprotocol.io/specification/2025-06-18 (접속 2026-11-27)
- [S5] Microsoft Research. (2024-02-13). 「GraphRAG: Unlocking LLM discovery on narrative private data」. https://www.microsoft.com/en-us/research/blog/graphrag-unlocking-llm-discovery-on-narrative-private-data/ (접속 2026-11-27)
- [S6] Edge, D. 외, Microsoft Research. (2024-04). 「From Local to Global: A Graph RAG Approach to Query-Focused Summarization」. arXiv:2404.16130. https://arxiv.org/abs/2404.16130 (접속 2026-11-27)
- [S7] CIO Korea. (2025-12-10). 「국정원, AI 보안 가이드북 배포」(원 발표: 국가정보원 국가인공지능안보센터). https://www.cio.com/article/4103809/ (접속 2026-11-27)
- [S8] 개인정보보호위원회. (2025-08-06). 「생성형 AI 개발·활용을 위한 개인정보 처리 안내서」 공개 보도. https://byline.network/2025/08/6-277/ · https://smartcity.go.kr/?p=48889 (접속 2026-11-27)
- [S9] OpenSearch 프로젝트. 「Reciprocal Rank Fusion (RRF)」 문서. https://docs.opensearch.org/latest/vector-search/ai-search/hybrid-search/rrf/ (접속 2026-11-27)

본문에 나온 약어와 전문 용어 풀이

용어	정의
가드레일	질문과 답변을 검사해 대량 개인정보 추출처럼 위험한 요청을 막는 사내 AI 모델
데이터 주권	데이터가 저장·처리되는 위치와 접근 통제를 조직이 스스로 정하는 권리

용어	정의
암호화 속성	칼럼 단위로 값을 암호화해 저장하고 검색 색인과 그래프에서 빼는 데이터베이스 속성
임베딩	문장의 뜻을 숫자 벡터로 바꾸는 일. 뜻이 비슷한 문장끼리 가까운 벡터가 된다
재순위	1차 검색 결과를 질문과의 관련도로 다시 정렬하는 단계와 그 모델
재현율@10	찾아야 할 정답 문서 가운데 상위 10건 안에 들어온 비율
청킹	긴 문서를 검색 단위가 되는 작은 조각(블록)으로 나누는 일
화자 분리	회의 녹음에서 말한 사람을 구분해 받아쓰는 기능
회사 세트	팀스페이스·공용 데이터베이스·공유 규칙·자동화를 한 번에 설치하는 MindWiki의 업무 체계 묶음
AAS(AI Agent Studio)	업무별 AI 에이전트를 만들고 운영하는 오픈마루의 에이전트 플랫폼
AES-256-GCM	256비트 키를 쓰는 대칭 암호 방식으로, 암호화와 함께 위변조 검사를 제공한다
Apache AGE	PostgreSQL 안에서 그래프 데이터와 Cypher 계열 질의를 쓰게 하는 오픈소스 확장
BM25	낱말이 문서에 얼마나 자주, 얼마나 드물게 나오는지로 순위를 매기는 키워드 검색 알고리즘
Cypher	노드와 관계의 경로를 패턴으로 적어 조회하는 그래프 질의 언어
HNSW(Hierarchical Navigable Small World)	여러 층의 그래프로 가까운 벡터를 빠르게 찾는 근사 최근접 검색 색인
Keycloak	사내 계정과 그룹을 관리하고 SSO를 제공하는 오픈소스 인증 서버
LLM(Large Language Model)	대량의 글로 학습해 문장을 이해하고 만들어 내는 대규모 언어 모델
MCP(Model Context Protocol)	LLM 애플리케이션과 외부 데이터·도구를 표준 방식으로 잇는 공개 규약
MRR(Mean Reciprocal Rank)	첫 번째 정답이 나온 순위의 역수를 질문마다 구해 평균한 검색 품질 지표

용어	정의
p50	응답 시간을 줄 세웠을 때의 중간값. 요청의 절반이 이 시간 안에 끝난다
pgvector	PostgreSQL에 벡터 유사도 검색과 HNSW 색인을 더하는 오픈소스 확장
RAG(Retrieval-Augmented Generation)	질문과 관련된 문서를 먼저 찾아 그 내용을 근거로 LLM이 답하게 하는 방식
RRF(Reciprocal Rank Fusion)	여러 검색 결과의 순위 역수를 더해 하나의 순위로 합치는 방법
SSO(Single Sign-On)	한 번 로그인으로 여러 시스템을 쓰게 하는 인증 방식
WAL(Write-Ahead Log)	데이터베이스가 변경 내용을 먼저 기록해 두는 로그. 특정 시점 복구에 쓴다



저장하는 순간 지식이 된다 — 회사 문서가 AI의 근거가 되는 위키, MindWiki

CONTACT

WEB
msap.ai
www.msap.ai/

EMAIL
hello@msap.ai

TEL
02-6953-5427
0269535427

YOUTUBE
@msaptv
www.youtube.com/@msaptv

LINKEDIN
linkedin.com/showcas...
www.linkedin.com/showcase/msap-ai/

FACEBOOK
facebook.com/opennaru
www.facebook.com/opennaru

SCAN

© 2026 SMART CF, MSAP.ai. All rights reserved.