PRODUCT INTRODUCTION

Argus RAG Studio 제품 소개

RAG 파이프라인의 구축 · 검색/생성 · 평가 · 운영 · 배포를 한 곳에서 다루는 플랫폼입니다.

"한 번 동작하는 RAG 앱"이 아니라, 측정 → 최적화 → 개선 루프와 원격 배포·운영까지 갖춘 RAG 스튜디오

제품 유형
셀프호스티드 RAG 플랫폼 — 온프레미스 · 에어갭(폐쇄망) 지원

핵심 축
Build · Retrieve & Generate · Evaluate · Operate · Deploy

기술 스택
FastAPI · Next.js 16 · PostgreSQL + pgvector · MinIO

문서 기준일
2026-07-17

Data Dynamics Inc · Argus Project

EXECUTIVE SUMMARY

Argus RAG Studio

무엇

문서 인제스천부터 하이브리드 검색·답변 생성, 품질 평가, 파이프라인 버전 관리, 에이전트 기반 원격 배포까지 — RAG 시스템 전 주기를 다루는 통합 스튜디오입니다.

왜

자체 구축 RAG는 대부분 "일단 동작"에서 멈춥니다. 품질을 숫자로 측정할 수 없고, 설정 변경의 영향을 추적할 수 없으며, 폐쇄망 배포와 한국어 문서(HWP)는 처음부터 다시 만들게 됩니다.

얻는 것

골든셋·LLM Judge로 품질을 수치화하고, 설정 스윕으로 최적 조합을 자동 탐색하며, 피드백이 골든셋으로 환류되는 개선 루프 — 그리고 에어갭 환경까지 커버하는 배포 체계.

8종
청킹 전략 (semantic·markdown 포함)

5종
파싱 전략 (VLM · 한글 HWP 전용 포함)

3축
LLM-as-Judge 품질 평가

에어갭
폐쇄망 모델 반입 · 오프라인 서빙

※ 성숙도 자체 평가: 견고한 T3(RAG 스튜디오) + T4(엔터프라이즈) 일부 — 평가·최적화·파인튜닝 축은 상용 플랫폼 평균 이상

CONTENTS

목차

제품 포지셔닝 — RAG를 엔지니어링 가능한 자산으로

"문서를 넣으면 답이 나오는" 단일 기능이 아니라, RAG 파이프라인을 버전 가능한 설정으로 정의하고, 평가 지표로 검증하며, 추적 가능하게 서빙합니다.

Build

멀티포맷 인제스천 — 수집 → 파싱 → 청킹 → 임베딩 → 색인. 문서 라우팅과 소스 워치(무인 수집)까지.

Retrieve & Generate

하이브리드 검색(벡터+렉시컬+RRF) · 리랭킹 · 인용 답변 · 멀티턴 챗(SSE) · 페더레이션 질의.

Evaluate

골든 데이터셋 · Hit Rate/MRR · LLM-as-Judge 3축 — 품질을 숫자로 측정하고 회귀를 방지.

Operate & Deploy

파이프라인 버전·롤백·비교, 트레이스·p50/p95, 피드백 루프, 에이전트 기반 원격 배포·에어갭.

핵심 차별점 대부분의 자체 구축 RAG가 빠뜨리는 "측정 → 최적화 → 개선 루프"를 제품의 중심에 둡니다 — 평가·설정 스윕·피드백이 기본 기능입니다.

플랫폼 아키텍처 — 데이터 평면

Next.js 대시보드에서 FastAPI 백엔드(:4700)로 — 인제스천·질의·운영 세 흐름이 PostgreSQL+pgvector와 MinIO/S3, 그리고 분리 배포 가능한 추론 서버 위에서 동작합니다.

※ 임베딩·리랭킹은 백엔드 내 로컬(FastEmbed)로도 동작 — 추론 서버 분리는 선택이며, 규모에 따라 단계적으로 확장합니다.

기술 스택과 화면 구성

레이어	구성
Backend	Python 3.11+ · FastAPI(async) · SQLAlchemy 2.0 · Pydantic v2
데이터	PostgreSQL + pgvector (벡터·FTS 일원화) · MinIO/S3
Frontend	Next.js 16 (App Router) · React 19 · Tailwind 4 + shadcn/ui
추론	FastEmbed(ONNX) 또는 torch(cu128 · GPU/Blackwell)
배포	Agent(FastAPI) · Docker/Podman · zot 레지스트리 · buildx bake
인증	로컬 JWT(HS256) · Keycloak OIDC · API 키(서비스 계정)

대시보드 화면 (주요 메뉴)

지식베이스	Playground	챗
파이프라인	평가	관측성
피드백	문서 라우팅	파인튜닝
어노테이션	모델 관리	서버 관리
소스 워치	S3 스토리지	작업(Jobs)
사용자/권한	API 키	PII 규칙

※ 모든 기능이 하나의 대시보드로 — 구축(지식베이스)부터 운영(서버 관리)까지 별도 도구 없이 단일 UI에서 수행합니다.

⚙ 관리 화면 외 자동화용 REST API 제공 — 서비스 계정(API 키) 기반 M2M 연동을 지원합니다.

인제스천 파이프라인 — 문서가 검색 가능한 지식이 되기까지

멀티포맷(txt/pdf/docx/xlsx/pptx/hwp/hwpx …) 문서를 비동기 워커가 파싱 → 청킹 → 임베딩 → 색인합니다. 실패 시 멱등 재처리, 대량 처리 시 워커 분리 배포가 가능합니다.

1. 업로드/수집
직접 업로드 · 소스 워치(드롭존 주기 스캔·무인 수집) · S3 register

2. 파싱
text · layout · docai · vlm · rhwp — 문서 특성별 전략, 미설치 시 폴백

3. 청킹
8종 전략 · char/token 단위 · 품질 가드

4. 임베딩
컬렉션별 모델·차원 고정 · 배치 처리 · 차원 검증

5. 색인
pgvector(벡터) + tsvector(렉시컬) 동시 색인

- **파싱 전략 5종 —** text(플레인) · layout(pdfplumber) · docai(docling) · vlm(비전 LLM — 스캔·복잡 레이아웃) · rhwp(한글 HWP/HWPX 전용 Rust 파서). 일반 프레임워크에 없는 VLM·HWP 파싱이 강점입니다.
- **한국어 특화 —** kss 기반 한국어 문장 분리(대용량은 규칙 기반 폴백), HWP/HWPX 네이티브 처리, 다국어 임베딩 기본(bge-m3).
- **RAG 문서 라우팅 —** 경로·메타 규칙 + 내용 임베딩 유사도로 문서를 컬렉션에 자동 배정 — 대량 유입 문서의 무인 분류.
- **엔지니어링 견고성 —** 비동기 워커(CPU 작업 to_thread 오프로드), 멱등 재인덱싱, 작업(Jobs) 화면에서 상태 추적.

청킹 8종 — 검색 품질을 좌우하는 디테일

청킹은 RAG 품질의 절반입니다. 단순 분할 wrapper가 아니라, 표 보존·의미 경계·오버랩 처리까지 촘촘하게 지원합니다.

recursive / fixed / sentence / paragraph / section

기본 전략군 — 문서 구조와 길이 예산에 따라 선택. char/token(tiktoken) 단위 교체로 언어별 문자↔토큰 비율을 흡수합니다.

markdown

표·코드블록을 원자 단위로 보존하고, 헤딩 breadcrumb(상위 제목 경로)를 각 청크에 전파 — 표가 반토막 나는 고전적 문제를 방지.

semantic

문장 임베딩의 인접 코사인 거리 백분위 임계로 의미 경계를 분할 — 문서별 적응형.

auto

문서 특성에 따라 전략 자동 선택.

공통 품질 가드

- 스마트 오버랩 — 꼬리를 문장→단어 경계로 스냅해 깨진 조각 방지
- 작은 청크 이웃 병합 — 노이즈 벡터 방지
- 청크 예산 캡 — 폭주 방지
- 위치 메타 부착 — section_path · char_start/end (부모 문맥 확장의 기반)

※ 자체 평가(경쟁 비교 문서): LangChain RecursiveCharacterTextSplitter · LlamaIndex SemanticSplitter 수준을 충족하거나 상회 — 특히 한국어 처리와 표 보존 디테일.

지식베이스 설계 — 컬렉션은 보안 격리 경계다

대부분의 RAG가 컬렉션을 "주제 묶음"으로만 다룹니다. Argus는 컬렉션을 접근/격리 경계(fail-closed)로 설계했습니다.

물리 격리 검색

모든 쿼리가 collection_id로 물리 필터링 — "보안등급이 다른 문서를 한 컬렉션에 넣고 쿼리 필터로 거르기"를 명시적으로 금지해, 필터가 조용히 무시되는 fail-open 유출을 원천 차단합니다.

벡터공간 정합성

임베딩 프로바이더·모델·차원·거리 메트릭을 컬렉션 생성 후 불변으로 고정 — 차원 불일치는 런타임에서 예외로 차단. 프레임워크에선 사용자가 직접 관리해야 하는 정합성을 스키마 레벨에서 보장합니다.

결정적 문서 라우팅

문서→컬렉션 라우팅을 방화벽 룰처럼 우선순위·first-match-wins·fail-closed로 데이터화 — 보안등급이 불확실하면 최고 등급으로 배정하는 특칙까지.

소스 워치

드롭존(공유 폴더·버킷)을 주기 스캔해 무인 수집 — 라우팅 규칙과 결합하면 "넣으면 알아서 분류·색인"이 됩니다.

※ 이종 임베딩 컬렉션 간에는 페더레이션 검색이 RRF로 결과를 병합합니다 — 격리와 통합 질의를 동시에 처리

하이브리드 검색 — 벡터 + 렉시컬 + RRF, 그리고 리랭킹

의미(벡터)와 키워드(렉시컬)를 각각 검색해 RRF로 융합하고, 필요 시 리랭커로 재정렬합니다. 인용 포함 답변과 멀티턴 챗(SSE 스트리밍)까지 이어집니다.

1. 질의
사용자 질문 → 컬렉션별 질의 임베딩

2. 병렬 검색
vector: pgvector ANN(코사인) · lexical: tsvector FTS

3. RRF 융합
두 순위 리스트를 Σ 1/(k+rank)로 병합

4. 리랭킹
none · llm · cross_encoder 선택

5. 생성
컨텍스트 조립 → 인용 포함 답변 (SSE)

- **3-모드 검색 —** vector / lexical / hybrid를 파이프라인 설정으로 선택 — 워크로드별 최적 모드를 평가로 검증.
- **리랭킹 3종 —** LLM 리랭커 또는 로컬 cross-encoder(:8081 분리 배포 가능) — 정밀도가 중요한 상위 결과 재정렬.
- **페더레이션 질의 —** 여러 지식베이스를 한 번에 검색 — 이종 임베딩 컬렉션도 RRF로 안전하게 병합.

Playground & 챗

Playground에서 검색·생성 설정을 바꿔가며 즉시 실험하고, 챗에서는 멀티턴 대화(SSE 스트리밍)로 사용 — 모든 질의는 트레이스로 기록되어 운영 화면에서 추적됩니다.

※ 본문·문서명 등 정본 데이터는 PostgreSQL에서 hydrate — 벡터 인덱스와 정본 저장소의 역할을 분리해 일관성을 지킵니다.

모델 유연성 — 임베딩 · LLM · VLM을 워크로드에 맞게

특정 모델·벤더에 잠기지 않습니다. 프로바이더 추상화 위에서 컬렉션·파이프라인 단위로 모델을 선택하고, 폐쇄망에서는 모델 레지스트리로 오프라인 서빙합니다.

레이어	프로바이더	비고
임베딩	local(FastEmbed ONNX) · OpenAI 호환(TEI/vLLM/Ollama)	기본 bge-m3(1024d, 다국어) — 컬렉션별 모델·차원·거리 지정, 차원 자동 감지
리랭커	LLM 리랭커 · 로컬 cross-encoder 서버	기본 ms-marco-MiniLM — 설정으로 교체 가능
생성 LLM	Claude(기본) · OpenAI 호환 · Ollama · vLLM	고난도/일반/저비용 모델을 용도별로 구분 운용
VLM(비전)	vLLM 배포(에이전트 경유)	스캔 문서·이미지 파싱(vlm 전략)과 이미지 분석에 사용
OCR 검출	PaddleOCR(korean) · EasyOCR(ko,en)	detection_server(:8082) — 이미지 텍스트 검출·라벨링 연계

▪ **검색 임베딩 파인튜닝까지 —** extensions/retrieval-fine-tuning(base multilingual-e5-large)으로 도메인 용어에 맞춘 임베딩·리랭커 튜닝을 지원 — 대부분의 RAG 제품이 다루지 않는 영역입니다(14장).

※ 임베딩 서버는 OpenAI 호환 API라 요청 model로 임의 HuggingFace 모델을 지정할 수 있습니다 — 표의 모델명은 기본값이며 전부 설정으로 교체 가능합니다.

평가 하니스

"좋아진 것 같다"가 아니라 지표로 말합니다. 골든 데이터셋 기반 검색 지표와 LLM-as-Judge 생성 지표로, 설정 변경·모델 교체의 효과를 회귀 없이 검증합니다.

검색 품질 — 골든 데이터셋

- **골든셋 관리 —** 질문–정답 문서 쌍을 데이터셋으로 관리 — 피드백(👍 / 👎)에서 승격해 지속 확충.
- **Hit Rate · MRR —** 정답 문서가 상위 결과에 포함되는지, 몇 위에 오는지 정량 측정.
- **holdout · 과적합 플래그 —** 평가셋 분리와 과적합 감지로 "평가에만 좋은 설정"을 걸러냅니다.

생성 품질 — LLM-as-Judge 3축

- **Faithfulness —** 답변이 검색된 컨텍스트에 충실한가 — 환각(hallucination) 감지.
- **Answer Relevance —** 답변이 질문에 실제로 대답하는가.
- **Correctness —** 골든 정답과 비교해 사실적으로 맞는가.
- **Judge 게이팅 —** 비용이 큰 LLM 평가는 상위 N개 설정에만 적용 — 평가 비용 통제.

※ 평가 결과는 파이프라인 버전과 연결됩니다 — "이 설정 변경으로 MRR이 얼마나 움직였는가"를 이력으로 추적합니다.

PART 4 · EVALUATE & IMPROVE — 02

설정 스윕과 개선 루프 — 최적 조합을 자동 탐색

청킹·검색 모드·top-k·리랭커 조합을 수동으로 실험하는 대신, 스윕이 자동 탐색하고 리더보드로 비교합니다 — 자체 구축 RAG가 대부분 빠뜨리는 최적화 루프입니다.

1. 스윕 정의

쿼리 축(검색 모드·top-k·리랭커) + 인덱스 축(청킹 전략 — 임시 컬렉션 생성)

2. 자동 실행

골든셋으로 각 조합을 평가 — 스마트 탐색·judge 게이팅으로 비용 절약

3. 리더보드

Hit Rate·MRR·Judge 점수로 정렬 비교 — holdout·과적합 플래그 확인

4. 적용

우승 설정을 파이프라인 새 버전으로 반영 · 언제든 롤백

개선 선순환 (피드백 루프)

운영 트레이스(질의량·성공률·지연 p50/p95·토큰·인기 질의)에서 문제 질의를 찾고 → 사용자 피드백 👍/👎를 수집해 → 좋은 질문·정답 쌍을 골든셋으로 승격하고 → 다음 스윕·평가의 기준으로 삼습니다. 쓸수록 평가셋이 두꺼워지고 품질이 올라가는 구조입니다.

※ 경쟁 비교 문서의 자체 평가: 평가/최적화 축은 "평균 이상" — 설정 스윕은 상용 플랫폼에서도 드문 강점입니다.

검색 파인튜닝 파이프라인 — 도메인 용어까지 학습

일반 임베딩 모델은 사내 용어·약어에 약합니다. 용어사전에서 출발해 학습 데이터를 만들고, 외부 트레이너로 임베딩/리랭커를 튜닝한 뒤 모델 레지스트리로 배포합니다.

1. 용어사전
도메인 용어·약어·동의어 정의

2. 합성 질의 · 라벨링
용어 기반 질의 생성 → 라벨링 UI로 검수

3. 트리플 빌드
(질의 · 정답 · 오답) 학습 데이터셋 구성

4. JSONL 내보내기 · 학습
외부 트레이너 — 잡 전용 콜백 토큰 (M2M 인증)

5. 등록 · 배포
모델 레지스트리 등록 → 임베딩 서버 교체 배포

- **베이스 모델 —** intfloat/multilingual-e5-large (sentence-transformers/torch) — 다국어 검색 임베딩 기준.
- **잡 추적 —** 파인튜닝 잡의 상태·이력을 대시보드에서 추적 — 학습은 GPU 서버에서, 관리는 스튜디오에서.
- **정직한 현재 상태 —** 잡 추적 + 콜백은 MVP 수준이며, 임베딩 교체 시 전체 재인덱싱 비용이 트레이드오프입니다(불변 벡터공간 설계의 대가).

※ 파인튜닝 화면 구성: 개요 · 용어사전 · 라벨링 · 데이터셋 · 잡 · 모델 — 전 과정이 대시보드 메뉴로 제공됩니다.

파이프라인 버전 관리와 관측성

검색·리랭크·생성 설정을 버전 가능한 자산으로 다룹니다. 모든 질의는 트레이스로 남고, 지연·토큰·성공률이 대시보드에 집계됩니다.

파이프라인 버전

- **버전 · 스테이지 · 활성 —** 설정 변경은 새 버전으로 — 스테이징을 거쳐 활성화, 문제 시 즉시 롤백.
- **diff · 비교 —** 두 버전의 설정 차이와 평가 지표를 나란히 비교 — "무엇을 바꿨더니 무엇이 좋아졌나"에 답합니다.
- **평가 연동 —** 버전마다 평가 결과가 연결되어 회귀를 사전에 차단.

관측성 (Observability)

- **질의 트레이스 —** 검색 결과·리랭크·생성까지 단계별 기록 — 느린 질의·이상 답변의 원인 추적.
- **지표 대시보드 —** 질의량 · 성공률 · 지연 p50/p95 · 토큰 사용량 · 인기 질의.
- **API 키 —** 서비스 계정 단위 발급 — 외부 애플리케이션 연동과 사용 주체 구분.
- **작업(Jobs) —** 인제스천·평가·스윕 등 백그라운드 작업 상태 추적.

※ 트레이스·지표·피드백이 같은 데이터 평면(PostgreSQL)에 쌓입니다 — 별도 APM 없이 RAG 품질 운영이 가능합니다.

에이전트 기반 원격 배포 — 여러 호스트를 한 화면에서

각 호스트의 Argus Agent(:4501)가 스튜디오의 명령을 받아 워커·임베딩·리랭커·검출·VLM 서버를 Docker 또는 systemd로 배포·관리합니다. 이미지는 zot OCI 레지스트리에서 받습니다.

RAG Studio Server
servermgr — 등록·배포·프록시
heartbeat 수신 · 서비스 현황 통합

zot OCI 레지스트리
buildx bake — amd64 + arm64 멀티아키

→ **호스트 A (CPU)**
Agent :4501 → worker ×N (systemd 권장)

→ **호스트 B (GPU)**
Agent :4501 → embedding · reranker · detection

→ **호스트 C (DGX/arm64)**
Agent :4501 → vlm(vLLM) · gpu-torch 변형

- **GPU 변형 자동 선택 —** amd64 GPU=gpu(onnx), arm64/Blackwell=gpu-torch — 호스트 아키텍처로 이미지 변형 자동 결정.
- **설정 자동 주입 —** 배포 완료 시 embedding.server_url 등 RAG 설정이 자동 반영 — 수동 배선 없음.
- **통합 서비스 관리 —** 관리형 + 수동(MANUAL) + K8s 배포까지 하트비트/프로브로 한 화면에 — 배포 유형·모델·디바이스·자원 사용률·재시작.

※ 인제스천 워커를 별도 호스트로 분리 배포하면 대량 색인 시에도 질의 응답성이 유지됩니다 — 워커는 systemd 방식을 권장합니다.

에어갭(폐쇄망) — 모델 레지스트리와 오프라인 서빙

인터넷이 없는 환경을 기본 시나리오로 설계했습니다. 모델은 이미지에 동봉하지 않고, 팩(pack)으로 반입해 Model Repository에서 배포 시 자동 설치합니다.

1. 모델 팩 생성
온라인 환경에서 HF 모델을 팩으로 패키징

2. 반입 · 등록
관리 > 모델 관리(레지스트리) 등록 → argus-models 버킷(Model Repository) 업로드

3. 배포 시 자동 설치
에이전트가 대상 서버 볼륨에 모델 자동 설치

4. 오프라인 서빙
임베딩·리랭커·VLM(vLLM)이 외부 연결 없이 서빙

- **이미지 파이프라인 —** docker buildx bake 매트릭스로 amd64+arm64 멀티아키 이미지를 빌드해 폐쇄망 zot 레지스트리(zot.airgap.local:5000)로 push — 컨테이너까지 완전 오프라인.
- **온라인 폴백 —** 인터넷이 되는 환경에서는 HuggingFace 런타임 다운로드로 폴백 — 같은 구성으로 온/오프라인 모두 동작.
- **적용 대상 —** 임베딩·리랭커·검출(OCR)·VLM(vLLM) 서버 전부 동일한 모델 관리 방식 — 규제 산업(금융·공공·국방)의 망분리 요건에 대응합니다.

※ 인증도 폐쇄망 친화적입니다 — 로컬 JWT 기본, 사내 Keycloak(OIDC) 연동 선택.

Extensions — 독립 배포 가능한 구성요소

각 구성요소는 백엔드와 분리된 독립 배포물이며, 에이전트가 Docker로 배포합니다. 모델은 기본값일 뿐 — 전부 설정으로 교체 가능합니다.

구성요소	기본 모델 / 엔진	백엔드	포트	GPU 변형
embedding_server	mxbai-embed-large-v1 (1024d)	FastEmbed(ONNX) / torch	8080	cpu · gpu · gpu-torch
reranker_server	ms-marco-MiniLM-L-6-v2	FastEmbed(ONNX) / torch	8081	cpu · gpu · gpu-torch
detection_server	PaddleOCR(korean) / EasyOCR	paddleocr / easyocr	8082	cpu · gpu
hwp_render_server	Chromium 렌더 (ML 없음)	Node + @rhwp/core	8085	단일
retrieval-fine-tuning	base multilingual-e5-large	sentence-transformers	—	일회성 배치
zot-registry	OCI 레지스트리	zot	5000	—
nifi-cluster-deploy	NiFi 클러스터 설치	Ansible	—	—

※ 임베딩 서버는 OpenAI 호환(/v1/embeddings)이라 스튜디오 밖의 다른 애플리케이션에서도 재사용할 수 있습니다.

🐳 공개 이미지: hub.docker.com/repository/docker/datadynamics/argus-rag-studio

어노테이션과 이미지 파이프라인

문서 속 이미지·스캔본도 지식이 됩니다. OCR 검출 → 라벨링 → 학습 데이터화, 그리고 VLM 기반 이미지 분석까지 — 문서 AI 워크플로를 내장했습니다.

이미지 OCR 라벨링

bbox + 텍스트 어노테이션 UI — AI-Hub JSON 형식 입출력으로 한국 공공 데이터셋 생태계와 호환. 검출 서버(PaddleOCR/EasyOCR)가 초벌 라벨을 제안합니다.

이미지 탐색기 · 추출 · 분석

문서에서 추출된 이미지를 탐색·검색하고, VLM으로 이미지 내용을 분석해 색인 — 표·차트·도면이 있는 문서의 검색 품질을 끌어올립니다.

HWP 미리보기

hwp_render_server(Chromium + @rhwp/core)가 한글 문서를 브라우저에서 렌더 — 변환 왕복 없이 원문 확인.

VLM 파싱 연동

vlm 파싱 전략이 스캔 PDF·복잡 레이아웃을 비전 LLM으로 텍스트화 — vLLM 서버는 에이전트로 배포·모델 관리.

※ 한국 문서 환경(HWP · 공공 스캔 문서 · AI-Hub 데이터셋)에 특화된 축 — 일반 오픈소스 RAG 프레임워크에는 없는 영역입니다.

보안 · 거버넌스

인증·인가부터 격리 설계까지 — 지금 제공하는 것과 로드맵에 있는 것을 구분해서 말합니다.

인증

로컬 JWT(HS256) 기본, Keycloak OIDC 연동, API 키(서비스 계정) — 최초 로그인 비밀번호 강제 변경.

인가 (RBAC)

역할 기반 접근 제어 + 행 단위 소유권 가드 — 사용자/권한 관리 화면 제공.

격리 설계

컬렉션 = fail-closed 보안 격리 경계(9장) — 검색이 항상 컬렉션 단위로 물리 격리.

식별자 위생

외부 노출 식별자는 UUID로 통일 — 내부 순번 ID 노출로 인한 열거 공격 차단.

PII 규칙

PII 규칙 관리 화면 — 개인정보 패턴 정의·관리(마스킹 고도화는 로드맵).

로드맵

문서 단위 ACL·권한 필터드 검색(P2), 감사로그 완결성, 멀티테넌시(테넌트 격리·쿼터)는 로드맵 항목입니다.

※ 온프레미스 셀프호스팅이라 데이터가 조직 밖으로 나가지 않습니다 — LLM까지 내부 서빙(vLLM·Ollama)하면 완전한 데이터 주권이 확보됩니다.

경쟁 비교 — 어디가 세고, 어디가 빈칸인가

코드 근거 기반 자체 분석(design/competitive-analysis.md) — 기술 품질은 상용 코어와 동급이고, 격차는 "엔터프라이즈 운영 표면적"입니다. 알면서 비워둔 빈칸이지 모르는 결함이 아닙니다.

축	수준	근거
지식베이스 구성	상위권	보안 격리(fail-closed) · 벡터공간 정합성 · 결정적 라우팅 — 상용 코어 이상
문서 임베딩 · 인제스천	우위	청킹 8종 + 품질 가드, 한국어/HWP, VLM 파싱 — 표준 충족 + 우위
검색 파이프라인	표준 충족	하이브리드 + RRF + 리랭크. 고급 검색(쿼리 재작성·HyDE·GraphRAG)은 로드맵
평가 · 최적화	평균 이상	골든셋 · LLM Judge · 설정 스윕(상용에서도 드문 강점)
운영 경화	T4 보완	분산 워커 · 권한 필터드 검색 · 자동 커넥터 · 임베딩 캐싱 — 로드맵 진행 중

- **vs 오픈소스 프레임워크(LangChain·LlamaIndex) —** 프레임워크는 조립 부품 — 평가·버전·운영·배포·UI를 전부 직접 만들어야 합니다. Argus는 그 결과물을 제품으로 제공합니다.
- **vs SaaS RAG 플랫폼 —** 데이터가 외부로 나가고 폐쇄망 불가 — 온프레미스·에어갭·한국어 문서가 요건이면 선택지가 급격히 좁아집니다.

성숙도와 로드맵 — 현재 위치와 다음 단계

RAG 솔루션 성숙도 4단계 기준, 현재는 견고한 T3(RAG 스튜디오) + T4 일부입니다.

T1 토이/데모
임베드+검색+답변
✓ 상회

T2 기능 제품
청킹·하이브리드·리랭크·인증
✓ 충족

T3 RAG 스튜디오
평가·버전·스윕·관측성·피드백
✓ 현 위치

T4 엔터프라이즈
고급검색·스케일·거버넌스
△ 일부

순위	로드맵 항목	내용
P1	검색 고도화	쿼리 재작성·multi-query, parent-document 검색, contextual compression
P2	권한 필터드 검색	문서 메타 기반 ACL을 검색 단계 pre-filter로 — 엔터프라이즈 필수
P3	운영 경화	분산 워커(Arq/Celery), 임베딩·결과 캐싱, 레이트/비용 통제
P4 · P5	커넥터 · 고급 검색	S3·웹크롤·Confluence 자동 수집 / GraphRAG · 에이전트형 반복 RAG

※ 한 줄 평(설계 문서): "상용 RAG 플랫폼의 코어를 자체 구축했고, 평가·최적화·파인튜닝 축에선 평균 이상 — 고급 검색과 운영 경화를 채우면 T4로 올라선다."

도입 시나리오와 제공 서비스

사내 지식 Q&A

규정·매뉴얼·기술 문서를 지식베이스로 — 인용 포함 답변으로 신뢰할 수 있는 사내 챗봇.

폐쇄망 문서 검색

금융·공공·국방 망분리 환경 — 에어갭 모델 반입과 온프레미스 LLM으로 완전 내부 운영.

한국어 문서 자산화

HWP/HWPX·스캔 문서·이미지가 많은 조직 — rhwp 파서 + OCR/VLM 파이프라인으로 색인.

구축 · 컨설팅

PoC 설계부터 프로덕션 구축까지.

- ✓ 지식베이스·라우팅 설계
- ✓ 파싱·청킹 전략 컨설팅
- ✓ 에어갭 배포 설계

기술지원

운영 전반에 대한 기술지원.

- ✓ 설치·업그레이드 지원
- ✓ 품질 튜닝(평가·스윕)
- ✓ 장애 대응·에스컬레이션

교육 · 커스터마이징

운영 교육과 도메인 특화 개발.

- ✓ 관리자·운영자 교육
- ✓ 커넥터·파서 확장 개발
- ✓ 검색 임베딩 파인튜닝

☎ 문의: 유지관 · 010-2356-5921 · trendy.ryu@data-dynamics.io

NEXT STEPS

결론 및 다음 단계

Argus RAG Studio는 "한 번 동작하는 RAG 데모"가 아니라, 품질을 측정하고 개선하며 폐쇄망까지 운영하는 RAG 스튜디오입니다.

평가·설정 스윕·피드백 루프, 한국어/HWP 특화, 에이전트 배포·에어갭 — 자체 구축으로는 가장 오래 걸리는 부분이 이미 제품 안에 있습니다.

1

요건 정의 · 데모

대상 문서·보안 요건(망분리 여부)·목표 워크로드 정의, 표준 데모 시연

2

PoC — 고객 문서로 검증

실제 문서 인제스천 → 골든셋 구축 → 평가 지표로 품질 확인 · 설정 스윕

3

단계적 도입

파일럿 조직 오픈 → 피드백 루프 정착 → 워커·추론 서버 확장 배포

Data Dynamics Inc

문의: 유지관 · 010-2356-5921 · trendy.ryu@data-dynamics.io · www.data-dynamics.io