04 LDIP 보고서 A — 플랫폼/아키텍처/운영

진행 상태	읽기 전

프로젝트: Project 4 · 기반: Flask(Python) · 작성: 2025-09-15

0. 문서 목적

- "문서통합플랫폼"의 총괄 개요, 시스템 아키텍처, 공통 선행작업, 데이터 모델 요약, 배포·보안·운영 원칙 정리함.
- 상세 모듈 사양은 **LDIP 보고서 B — 모듈별 상세 사양** 문서 참조함.

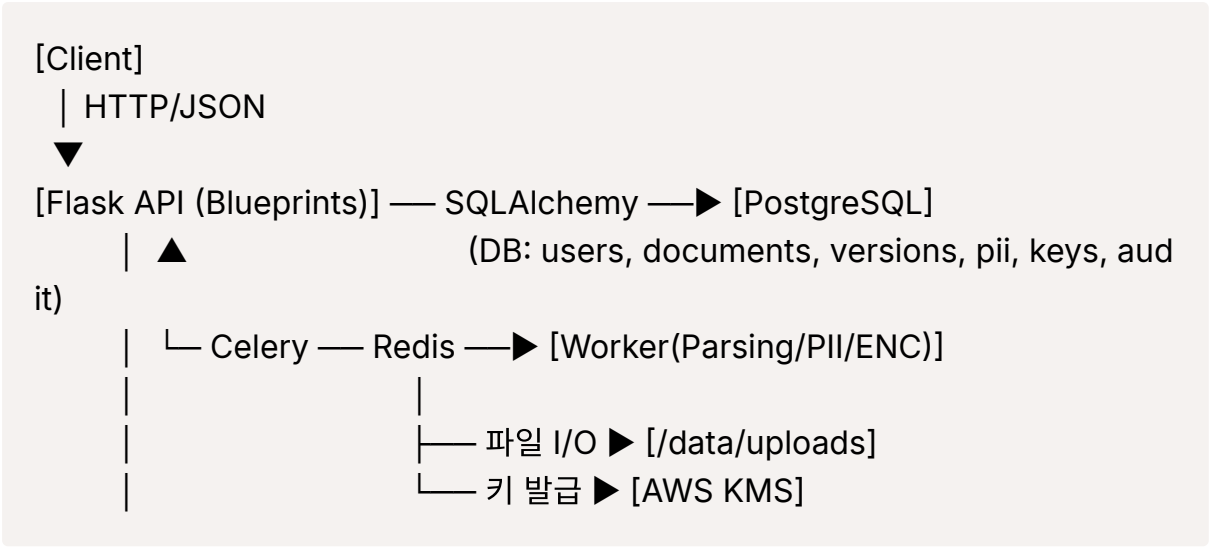
1. 명칭·약어

- 공식 한글 명칭: **문서통합플랫폼**
- 공식 영문 명칭: **Lockument Document Integration Platform**
- 약어: **LDIP**

2. 개요·범위·원칙

- 방식: **Flask(Python) + Blueprint** 모듈화, **Celery + Redis** 비동기 처리, **PostgreSQL** 영속 저장, **AWS KMS** 키관리 채택함.
- 범위: 업로드, 버전, 파싱, PII 탐지/마스킹, ENC(암·복호화), 감사, 관리자 조회 포함함.
- 비범위(초기): 항목별 암호화, TokenizedRecovery 재조립, OCR, 외부 DLP 연동 제외함.
- 원칙: **정본=암호문**, **마스킹본=별도 버전**, **작성자만 복호화** 원칙 적용함.

3. 시스템 아키텍처 요약



- 포트: API 8080, Redis 6379, DB 컨테이너 5432(호스트 5433 매핑) 구성함.

3.1 공통 선행작업 체크리스트(비전공자용)

- Docker/Compose 설치 확인함.
- Git 클라이언트 설치 확인함.
- `.env` 파일 작성함: `DATABASE_URL, REDIS_URL, UPLOAD_ROOT, ENCRYPTION_MODE, AWS_REGION, KMS_KEY_ID`.
- `docker compose up --build -d` 실행함 → `http://<호스트>:8080/healthz` 확인함.
- DB 초기화: `flask db upgrade` 또는 제공 SQL 적용함.
- AWS KMS 권한 부여함(`kms:GenerateDataKey`, `kms:Decrypt`).
- 샘플 문서 3~5개 준비함(PDF/DOCX/XLSX, 1~20MB 권장).

---

4. 데이터 모델 요약(최소)

- 핵심 테이블: `users`, `documents`, `document_versions`, `pii_rules`, `pii_findings`, `doc_keys`, `audit_log` 구성함.
- 관계(ER) 요약: 문서(1) — 버전(N), 버전(1) — 키(1), 버전(1) — 탐지결과(N) 구성함.
- 인덱스: `document_versions(doc_id,v_no)` 유니크, `pii_findings(doc_id,v_no)` 인덱스, `audit_log(ts)` 인덱스 구성함.

---

5. 환경 설정·배포

5.1 환경 변수(.env)

```
FLASK_APP=app.api:app
DATABASE_URL=postgresql+psycopg2://app:app_pw@db:5432/appdb
REDIS_URL=redis://redis:6379/0
UPLOAD_ROOT=/data/uploads
ENCRYPTION_MODE=KMS
AWS_REGION=ap-northeast-2
KMS_KEY_ID=arn:aws:kms:ap-northeast-2:123456789012:key/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

5.2 배포(Compose)

- 서비스: `api`, `worker`, `db`, `redis` 정의함.
- 볼륨: `./data/uploads:/data/uploads`, `./data/pg:/var/lib/postgresql/data` 마운트함.
- 헬스체크: `/healthz` 제공함.

---

6. 보안·운영 가드레일

- 입력 검증: 파일 크기/확장자/매직넘버 검사 적용함.
- 키·비밀: IAM Role 우선, Access Key 지양함.
- AAD 고정: `{user_id, doc_id, version, ts}` 적용함.
- 레이트리밋: 복호화/다운로드 시도 제한 구성함.
- 로깅: 예외 메시지 민감정보 노출 금지함.
- 백업/복구: 암호문·`doc_keys`·`audit_log` 주간 백업 + 복구 리허설 운영함.

---

7. 성능·품질 목표(초기 기준선)

항목	목표
업로드→암호화(10MB)	평균 ≤ 2초
파싱(텍스트 PDF)	성공률 ≥ 95%
KMS 호출 실패율	< 0.1%
복호화 권한 위반	403 반환 100%

---

8. 테스트 전략·수용 기준(AC)

- 단위: 정규식 규칙, AES-GCM, KMS 래퍼 테스트 수행함.
- 통합: 업→암→파→PII→마→복 엔드투엔드 시나리오 수행함.
- 보안: 권한 우회, AAD 불일치, 태그 검증 실패 케이스 포함함.
- 성능: 100MB×10 파일 배치, KMS 호출 지표 보고서 산출함.

---

9. 운영 Runbook(요약)

1. 배포: 이미지 빌드 → Compose 적용 → DB 마이그레이션 → 헬스체크 순서 수행함.
2. 장애: KMS 오류 재시도 3회 → 실패 시 알림 및 복구 가이드 참조함.
3. 모니터링: 실패 작업 수, KMS 오류율, API p95, 저장소 사용량 대시보드 운영함.
4. 백업: 주간 스냅샷 및 복구 리허설 수행함.

---

10. 일정·역할(RACI)

- W3: DB 스키마 안정화, ENC 경로, `/encrypt` 워커화, 파싱 연결 수행함.
- W4: 마스킹본 다운로드 정책 적용, 성능·품질 리포트 제출함.
- 역할: PM(상원) 기획·품질, 서버(준형) 인프라·API, 알고리즘(재석) PII 규칙·마스킹 담당함.

---

11. 리스크·대응

- 바이너리 포맷 편집 난이도 높음 → 원본 재조립 금지·암호문 정본 원칙 적용함.
- OCR 미지원 → 차기 단계 Tesseract 컨테이너 계획 수립함.
- KMS 비용/지연 → 배치 묶음 처리·재시도·지수백오프 적용함.

부록 A. 초기 스키마 목록

- `users`, `documents`, `document_versions`, `pii_rules`, `pii_findings`, `doc_keys`, `audit_log`.

부록 B. .env 키 목록

- `DATABASE_URL, REDIS_URL, UPLOAD_ROOT, ENCRYPTION_MODE, AWS_REGION, KMS_KEY_ID`.