

WE ARE



CLOUD SECURITY

개인정보 유출 위기 - 기하급수적으로 증가하는 침해사고

⚠️ [KISA 침해사고 통계로 확인되는 심각한 보안 현실](#)

↑ 3년간 195% 급증

▣ 침해사고 신고 추이

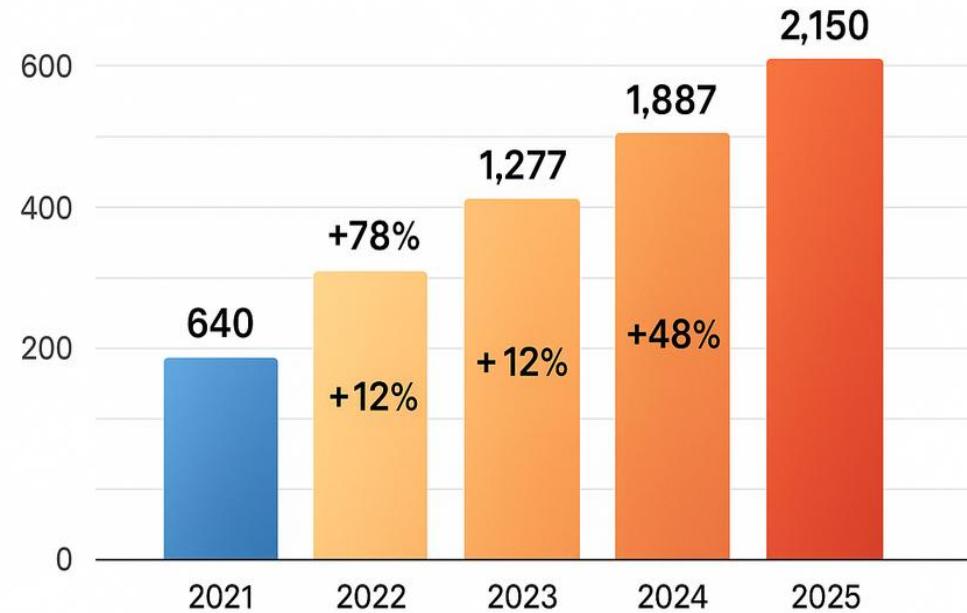
2021년:	640건
2022년:	1,142건 (+78%)
2023년:	1,277건 (+12%)
2024년:	1,887건 (+48%)
2025년:	2,150건 (상승 예상) (+14%)

▣ 대기업 개인정보 유출 사례

❗ KT 고객정보 무단 열람 (2024)

피해 규모: 1,200만명 고객정보 무단 열람
과징금: 조사 중

KISA cyber incident statistics



출처: KISA 사이버 위협 동향 보고서 (2021-2025), 개인정보보호위원회 (2025)

개인정보 유출 위기 - 기하급수적으로 증가하는 침해사고

⚠ KISA 침해사고 통계로 확인되는 심각한 보안 현실

↑ 3년간 195% 급증

2025년 대기업 침해사고 심각!

개인정보 유출 사례

SK telecom USIM 정보 유출 (2025)

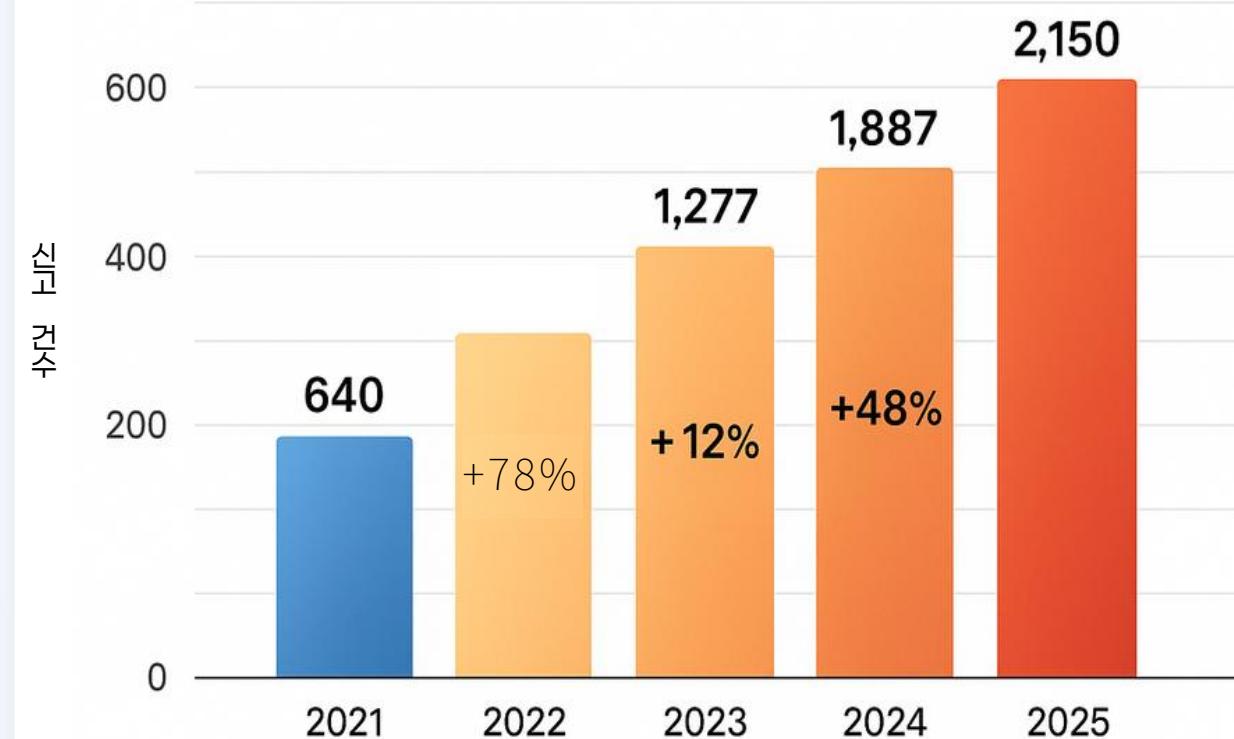


- ! 피해 규모: 2,300만명 개인정보 유출
- ! 과징금: 1,347억원
- ! 개인정보가 마스킹·암호화 없이 저장되어 발생한 참사

kt 고객정보 무단 열람 (2024~2025)

- ! 피해 규모: 1,200만명 고객정보 무단 열람
- ! 과징금: 조사 중

KISA cyber incident statistics



중소기업 또한 보안 예산을 쓰지 못하는

구조적 문제

KISA 실태조사로 확인된 현실



전문 인력 부족

보안 전문가를 찾기 어려운 현실적 문제로 정보보호 예산을 활용하지 못합니다.

| KISA 조사: 중소기업 78%가 보안 전문가 채용 실패



고가의 솔루션

기존 정보보호 솔루션의 높은 가격은 중소기업의 부담을 가중시킵니다.

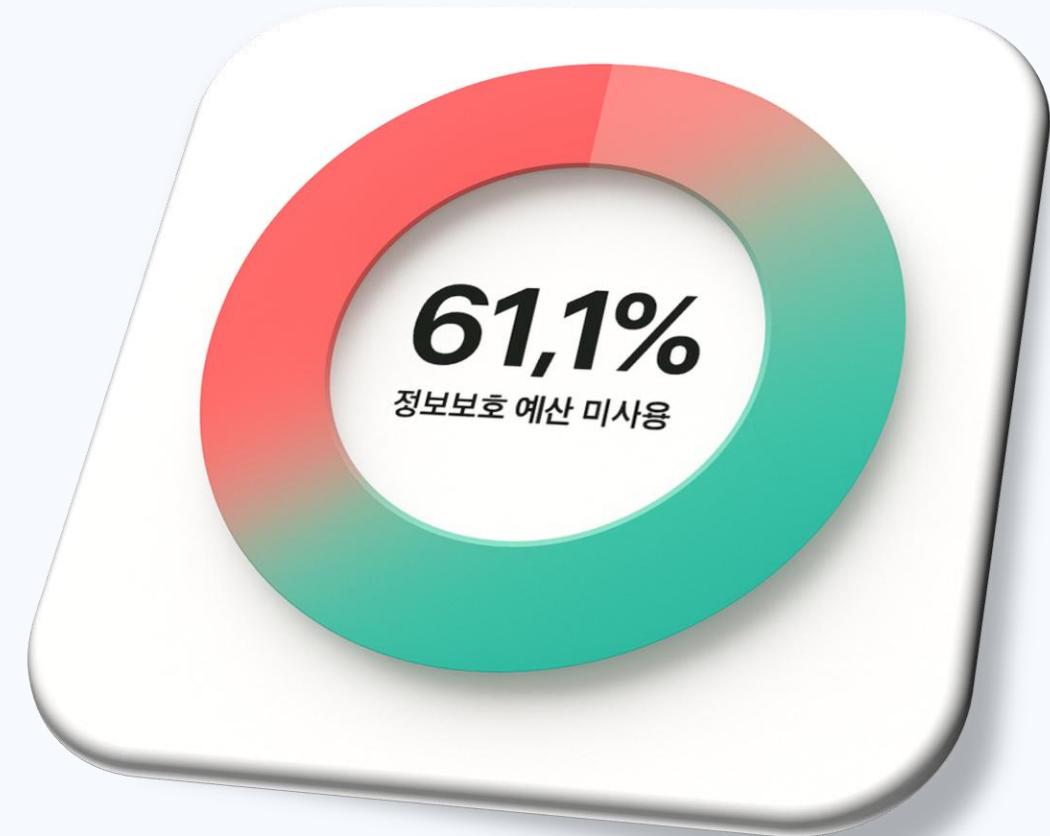
| 기존 DLP 솔루션 연간 5천만원 이상, 중소기업 감당 불가



복잡한 도입 과정

설치와 운영에 전문 지식이 필요하며 기업 내 IT 인프라 조정이 필요합니다.

| 평균 3개월 도입 기간, 전담 인력 필요



61.1% 기업이 예산 배정받고도 활용 못함

LOCKUMENT 해결책: 무료, 30초 설치, 전문지식 불필요

LOCKUMENT

Back-end

보이지 않는 보안을 설계

시스템 아키텍처 설계: React + Flask + PostgreSQL 3계층 구조, AWS 클라우드 인프라(EC2, KMS, VPC), Docker 기반 컨테이너 배포

보안 설계: STRIDE 위협 모델 기반 6대 위협 분석, AWS KMS 키 관리 체계, 역할 기반 접근 제어(RBAC), 감사 로그 시스템

프로젝트 관리: 74일 일정 관리, 요구사항 분석, 시스템 설계 문서화, 논문 작성 (16페이지, 28개 참고문헌)



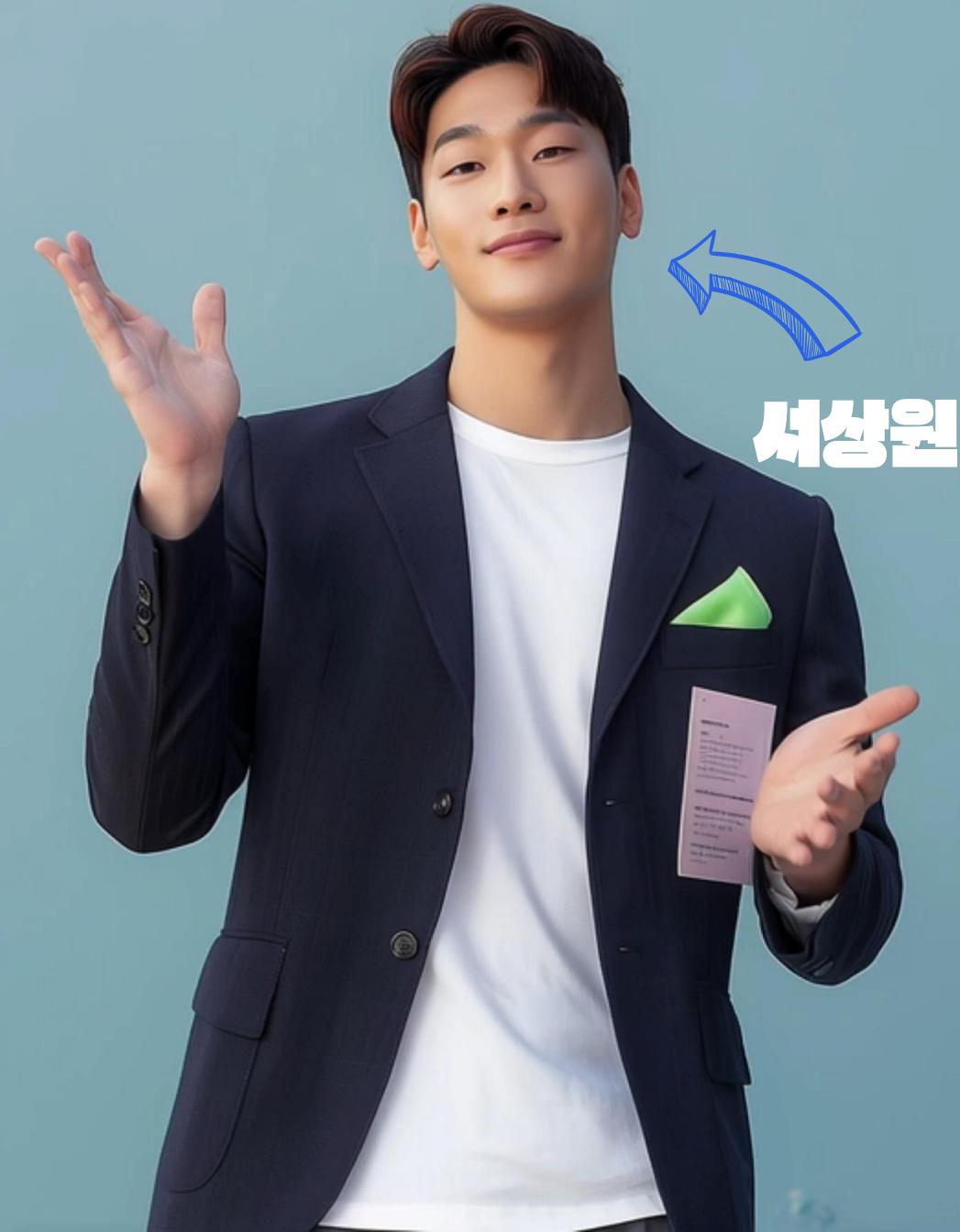
Backend



Database

Ubuntu

PostgreSQL





김민진

Front - end

보이는 것들의 보안을 디자인

React SPA 구현: Hooks, Context API, TypeScript 기반 SPA 아키텍처 개발, 상태 관리 최적화

역할별 접근 제어 UI: Admin/User/Guest 권한별 맞춤형 인터페이스, 상황별 조건부 렌더링 개발

보안 기능 UI 통합: 로그인, 마스킹/암호화 모드 선택, 감사로그 필터링 및 시각화 기능 개발



Frontend
React



모듈/ML
NER+정규표현식



업로드 시점 자동 탐지

딥러닝과 정규식 기반 정밀 탐지로
99.7% 정확도의 개인정보 식별



실시간 마스킹 처리

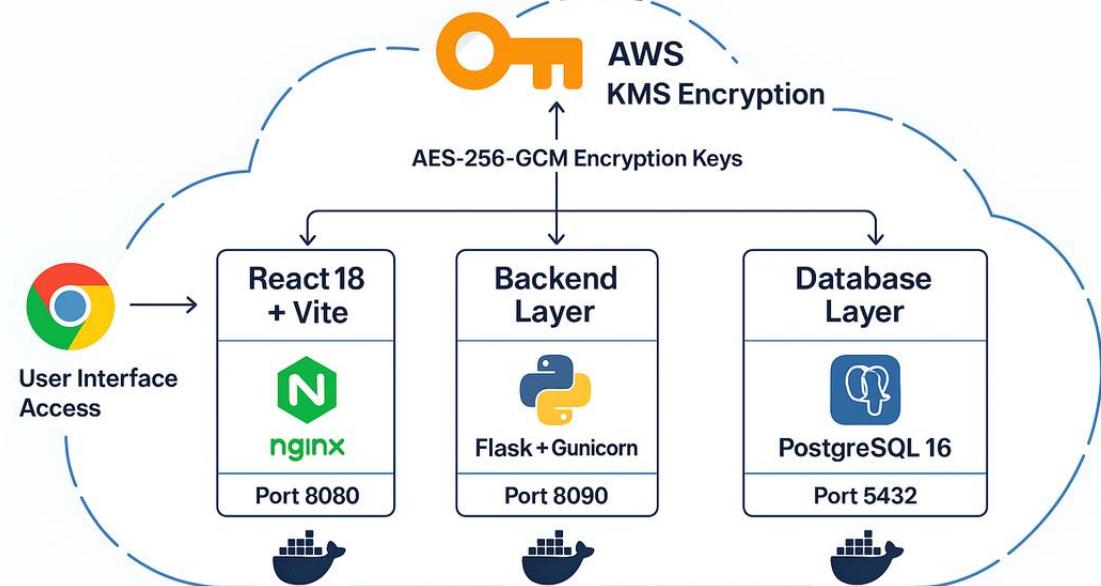
개인정보 유형별 선택형 마스킹으로
가독성과 보안성 모두 확보



암호화 저장

AWS KMS 연동 AES-256-GCM 군사기관급 암호화로
완벽한 데이터 보호

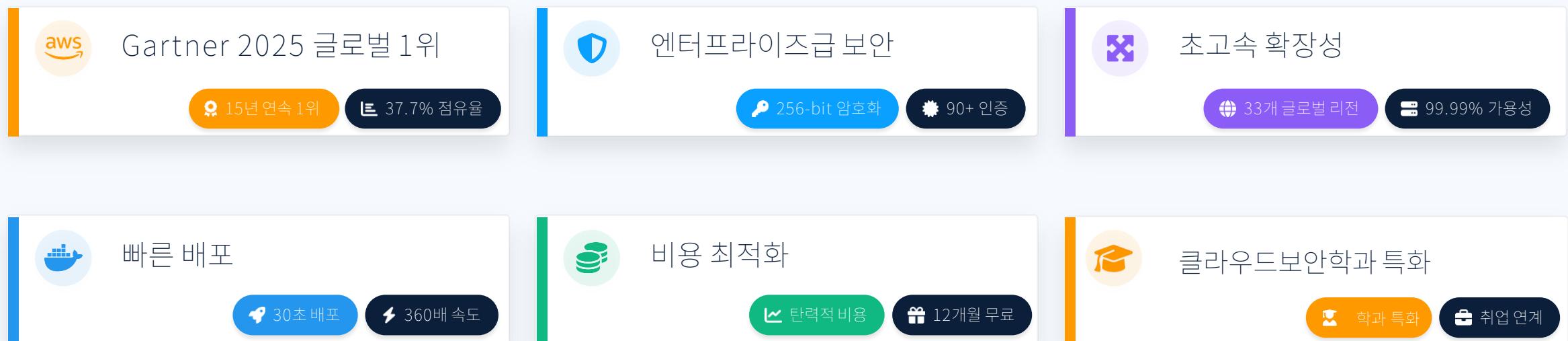
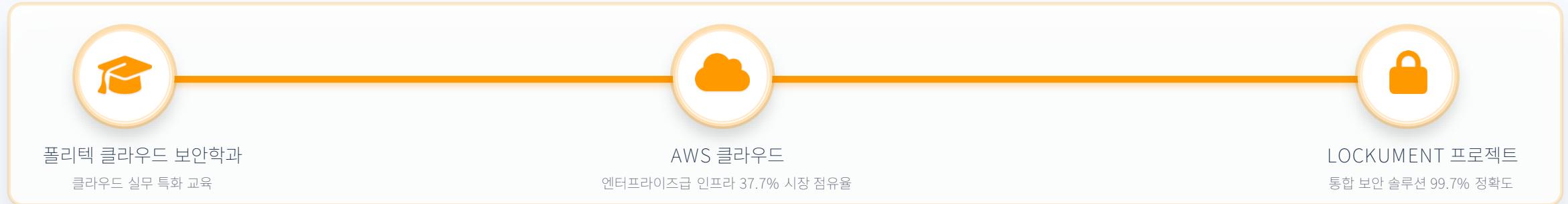
3-Tier Container Architecture





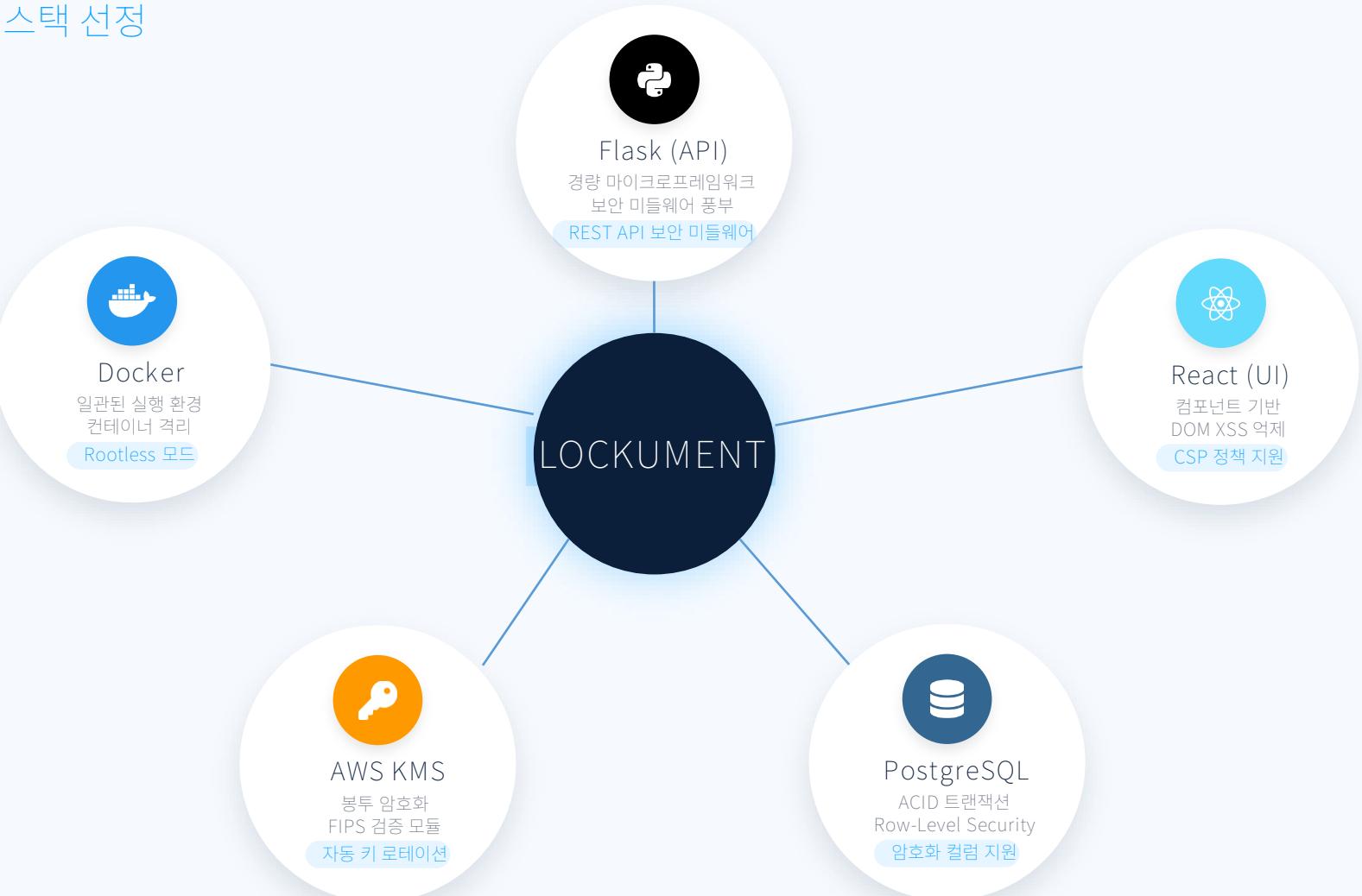
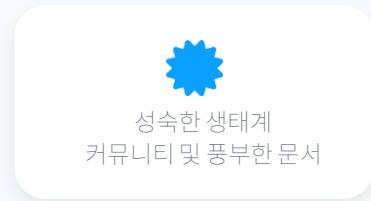
CLOUD + 클라우드 보안 학과 전문성의 완벽한 조화

👑 Gartner 2025 글로벌 1위 × 한국폴리텍대학교 클라우드보안학과 교육 연계



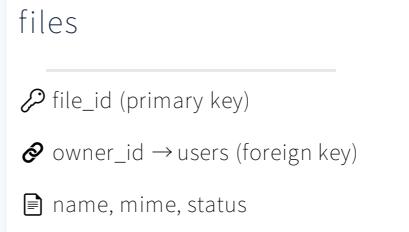
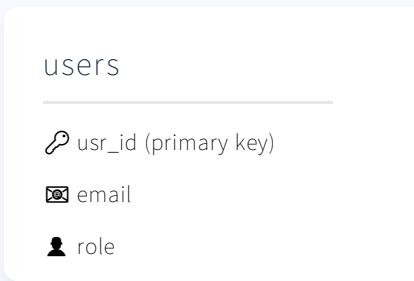
핵심 기술 스택 및 선택 이유

☰ 보안성, 확장성, 생산성을 고려한 기술 스택 선정



데이터베이스 설계 및 보안 (ERD + 접근제어) 안전한 데이터 관리를 위한 설계 원칙

ER 다이어그램



보안 설계 원칙



RLS 행 수준 접근제어

Row-Level Security로 owner/role 기반 행 수준 접근제어를 구현합니다. 사용자는 자신의 파일만 볼 수 있으며, 관리자는 별도 정책으로 제한됩니다.



AES-256-GCM 암호화

개인정보가 포함된 컬럼은 평문으로 저장하지 않고, 애플리케이션 레벨에서 AES-256-GCM으로 암호화하여 저장합니다. 인증된 암호화로 데이터 무결성을 보장합니다.



KMS DEK 래핑

AWS KMS로 데이터 암호화 키(DEK)를 래핑하고, key_id로 파일-키 매핑을 관리합니다. 마스터 키 노출 없이 DEK를 안전하게 보호합니다.



Append-only 감사 로그

Append-only 테이블 구조와 시간 서명으로 로그 조작을 방지합니다. 모든 중요 이벤트는 actor, action, target, timestamp로 기록되어 인덱싱되어 빠르게 조회 가능합니다.



최소 권한 원칙

DB 계정을 READ/WRITE/AUDIT로 분리하고, 애플리케이션은 필요한 최소 권한만 부여받습니다. 사용자 역할별 권한 매트릭스를 적용합니다.

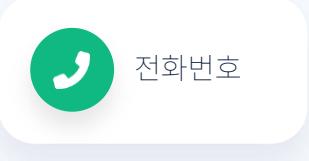
8가지 탐지 대상



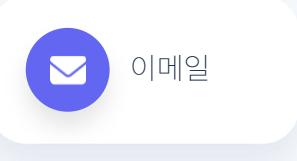
이름



생년월일



전화번호



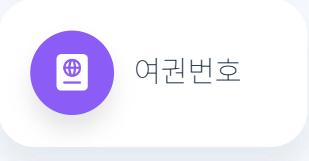
이메일



주소



주민등록번호



여권번호



운전면허번호

선정 근거



개인정보보호법 제2조 (한국)

이름, 주민등록번호, 주소, 전화번호 등 개인식별정보 정의. 살아있는 개인에 관한 정보로서 개인을 알아볼 수 있는 정보



GDPR Article 4 (EU)

이메일, 생년월일, 여권번호 등 개인 데이터 정의. 식별자, 온라인 식별자, 위치 데이터 포함



ISMS-P 인증기준 (한국)

개인정보 처리 시스템 암호화 및 접근 통제 요구사항. 고유식별정보 안전성 확보조치 의무화

마스킹 처리 방법



부분 마스킹

식별 가능성을 낮추면서 필요한 정보는 제공하는 방식

010-****-5678, r*****@****.com



완전 마스킹

완전히 가려서 식별이 불가능하도록 처리하는 방식

주민번호: *****,
여권번호 *****



암호화 저장

원본 정보를 안전하게 보호하면서 필요시 복호화 가능

AES-256-GCM 암호화



암호화 프로세스 상세 1

AWS KMS 암호화 아키텍처

1



KMS CMK로 DEK 생성

AWS KMS 고객 관리형 키(CMK)를 사용하여 고유한 데이터 암호화 키(DEK)를 생성합니다. 각 파일마다 새로운 DEK가 생성됩니다.

2



파일을 AES-256-GCM 암호화

생성된 DEK로 파일을 AES-256-GCM 알고리즘으로 암호화합니다. 이 과정에서 무결성 검증을 위한 GCM 태그와 초기화 벡터(IV)가 함께 생성됩니다.

3



래핑된 DEK 저장

KMS CMK로 DEK를 래핑(암호화)하고 래핑된 DEK, IV, GCM 태그를 메타데이터로 저장합니다. 원본 DEK는 메모리에서만 존재하고 영구 저장되지 않습니다.

4



복호화 시 DEK 언랩

복호화가 필요할 때 KMS 서비스에 래핑된 DEK를 전송하여 원본 DEK로 언랩(복호화)합니다. 이 과정은 사용자의 IAM 권한에 따라 제어됩니다.

5



무결성 검증

GCM 태그를 사용하여 복호화된 데이터의 무결성을 검증합니다. 데이터가 변조되었거나 손상되었을 경우 복호화 과정이 실패하여 데이터 보안을 보장합니다.



봉투 암호화는 데이터 암호화 키(DEK)를 마스터 키(CMK)로 보호하는 이중 보안 아키텍처입니다



암호화 프로세스 상세 2

AWS KMS 키 관리 정책



CMK 정책: 관리자/서비스 분리 권한

Customer-managed Key(CMK)를 통해 관리자와 서비스 계정의 권한을 명확히 분리.

키 생성자는 키를 삭제할 수 없고, 사용자는 키를 수정할 수 없는 최소 권한 원칙 적용

| *KMS Key Policy JSON으로 세밀한 IAM 권한 관리*



감사: CloudTrail 로깅

모든 KMS 키 작업은 CloudTrail에 자동으로 기록되어 누가, 언제, 어떤 작업을 수행했는지 감사 추적이 가능. 로그는 변경 불가능하게 저장.

| *감사 추적으로 규제 준수 입증 및 이상 징후 탐지*



키 로테이션: 90일 주기 자동화

암호화 키는 90일마다 자동으로 교체. 키가 교체되어도 이전에 암호화된 파일들은 자동으로 이전 키 버전으로 원활하게 복호화.

| *AWS KMS 자동 로테이션으로 관리 부담 최소화*



준수: FIPS 140-3 검증

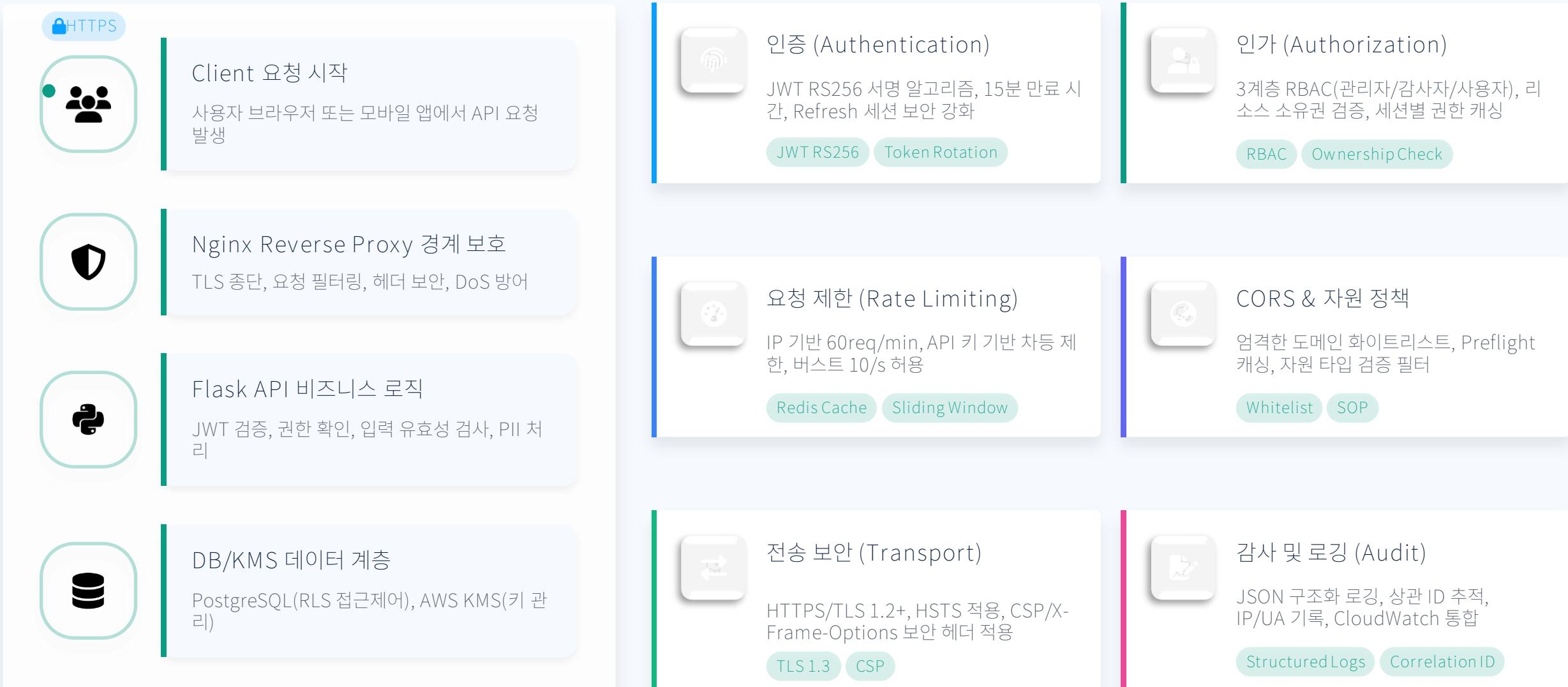
AWS KMS는 FIPS 140-3 검증된 하드웨어 보안 모듈(HSM)을 사용하는 환경을 지원. 정부, 금융, 의료 규제 요구사항을 충족.

| *업계 최고 수준의 암호화 모듈 표준 준수*



본 프로젝트 암호화는 데이터 암호화 키(DEK)를 마스터 키(CMK)로 보호하는 이중 보안 아키텍처입니다

API 보안 설계 (인증·인가·보호) | 다층적 보안 통제로 안전한 API 아키텍처 구현



배포 아키텍처 (Docker) 1 컨테이너화로 일관된 실행 환경과 보안 강화된 격리 구현



네트워크 분리

- frontend_net: 사용자 접근 레이어
- backend_net: 서비스 간 통신
- db_net: 데이터베이스 전용

최소 접근 원칙으로 레이어별 분리 및 네트워크 격리 구현



컨테이너 보안 강화

- rootless 컨테이너: 권한 격리
- read-only 파일시스템: 변조 방지
- no-new-priviliges: 권한 상승 차단

권한 최소화로 컨테이너 탈출 및 내부 공격 방어



볼륨 구성

- db_data: PostgreSQL 데이터
- logs: 구조화 로그 저장
- tmp: 임시 처리 영역 (주기적 정리)

볼륨 백업 자동화, readonly 접근 제어로 무결성 보장



비밀 관리 & 취약점 대응

- AWS SSM/Secrets Manager로 비밀 주입
- 이미지 서명/검증: 신뢰성 확보
- 주기적 취약점 스캔: CI/CD 통합

환경변수 직접 사용 금지, 이미지 신뢰성 검증 강화

배포 아키텍처 (Docker) 2

컨테이너화로 일관된 실행 환경과 보안 강화된 격리 구현



Docker Compose 구성



reverse-proxy

Nginx, HTTPS 종단, 요청 라우팅



worker

비동기 작업, PII 탐지 처리



web (React)

사용자 인터페이스, SPA



db (PostgreSQL)

데이터 저장, RLS 보안



api (Flask)

RESTful API, 비즈니스 로직



cache (optional)

Redis, 세션/응답 캐싱

frontend_net
backend_net
db_net

실제 장애 이슈 및 해결  프로젝트 개발 중 직면했던 실제 기술적 문제와 극복 과정

실제 문제 #1



서버 스토리지 부족 (20GB → 50GB)

문제

DB, 웹 모듈, Docker 데이터, 컨테이너,
백업파일이 개발 진행되며 누적되어 서버 용량 부족

```
$df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/xvda1	20G	19G	1.0G	95%	/

원인

비용 절감을 위해 20GB 소용량 서버로 시작했으나
개발 데이터 축적으로 공간 부족 발생

1. AWS EC2 서버 스토리지를 50GB로 확장
 2. 사용하지 않는 백업 파일 삭제 자동화
 3. 최신 버전 백업만 유지하는 정책 수립

실제 문제 #2



암호화/복호화 매핑 오류

문제

base64 인코딩/디코딩 함수 혼용,
encryption_context 파라미터 전달 오류
감사 로그 파일명 누락

```
# routes_crypto.py와 crypto_core.py 불일치  
import base64  
Ã'ÑÑ'ÑÓÑÄÖPÌN RPÑCÉÑMPMÆ ÑPMÆ  
Á ÑÓÑÄÖPÌDÑÖÖLÑÖÖPÑRP 누락 from base64 import 누락
```

10

import 구문 불일치,
함수 시그니처 불일치,
로그 스키마에서 파일명 필드 누락

1. from base64 import b64encode, b64decode 추가
 2. 함수에 encryption_context 파라미터 추가
 3. COAI ESCF로 파일명 추출하여 로그에 전달

실제 문제 #3



React Native 프론트엔드 통합

문자

React 컴포넌트 간 연동 실패,
기능 구현과 디자인 균형 문제,
알 수 없는 다운그레이드와 연동 오류

```
ânpm run build  
Error: Module not found: Can't resolve  
'./AdminLogs.jsx'  
âGOOGOEFOPÔÔIÖMNGGÖDDEI ÖMNÖÖN NÖPÖÖNÖÖNÖÖN
```

시스템 완성도 및 검증 방법론

과학적 검증 방법론으로 입증된 성능과 완성도

99.7%

F1 Score (정확도)

0.8초

평균 응답시간

0.02%

오탐률



검증 방법론



테스트 데이터셋

100개 문서, 실제 개인정보 8종 포함, 총 50MB 데이터



검증 지표

Precision 0.98, Recall 0.96, F1 Score 0.97, Confusion Matrix 시각화



성능 테스트

Apache JMeter 1,000 req/s 부하 테스트, 평균 응답 0.8초 유지



보안 테스트

OWASP Top 10 취약점 검증, SQL Injection/XSS 차단 확인, Rate Limiting 검증

출처: LOCKUMENT 기술 검증 보고서 (2025년 10월)

테스트 환경



AWS EC2 t3.medium (2vCPU 4GB RAM)



PostgreSQL RDS db.t3.micro



테스트 기간: 2025년 9-10월



학술적 검증 방법론 적용

실제 구현 기능 - AWS 클라우드 환경에서 작동 중

2개 계정 시스템, 파일 처리, KMS 토큰 복호화, 감사 로그가 실제 작동하는 웹 애플리케이션

메인 화면

The main dashboard provides a central hub for managing users, processing files, and monitoring audit logs. It includes sections for user administration, file upload and processing, and detailed audit logs.

파일 업로드 및 처리

This screen allows users to upload files (Word, PPT, Text, JSON) and process them using various encryption and decryption options. It also includes a section for generating masked audit logs.

감사 로그

The audit log interface displays a history of operations performed by users, including file processing (Encrypt, Decrypt, Export, Import) and log generation (Mask & Log). Each entry shows the user, IP address, file name, and timestamp.

계정별 권한 시스템

관리자/사용자 로그인 작동
Admin/ User 1 /User 2

파일 업로드

최소 4가지 파일 형식 지원
Word/PPT/Text/JSON
(PDF 재구현중)

KMS 복호화

토큰 기반 복호화 작동
계정별 매핑

감사 로그

작업 수행 시각 기록
마스킹/암호화/복호화 추적

빠른 구현

AWS | docker로 신속 구현
배포 및 관리 용이성

RBAC 적용

역할 기반 접근 제어
계정별 권한 차등 적용

데모 사이트 24시간 운영 중
lockument.duckdns.org

프로덕션 레벨 아키텍처 구현 완료

✓ 보안 검증
✓ 성능 검증

LOCKUMENT 실제 작동 화면

30초 만에 완성되는 개인정보 자동 보호 시스템

① 업로드



문서를 드래그 앤 드롭하거나 클릭하여 업로드



② 자동탐지



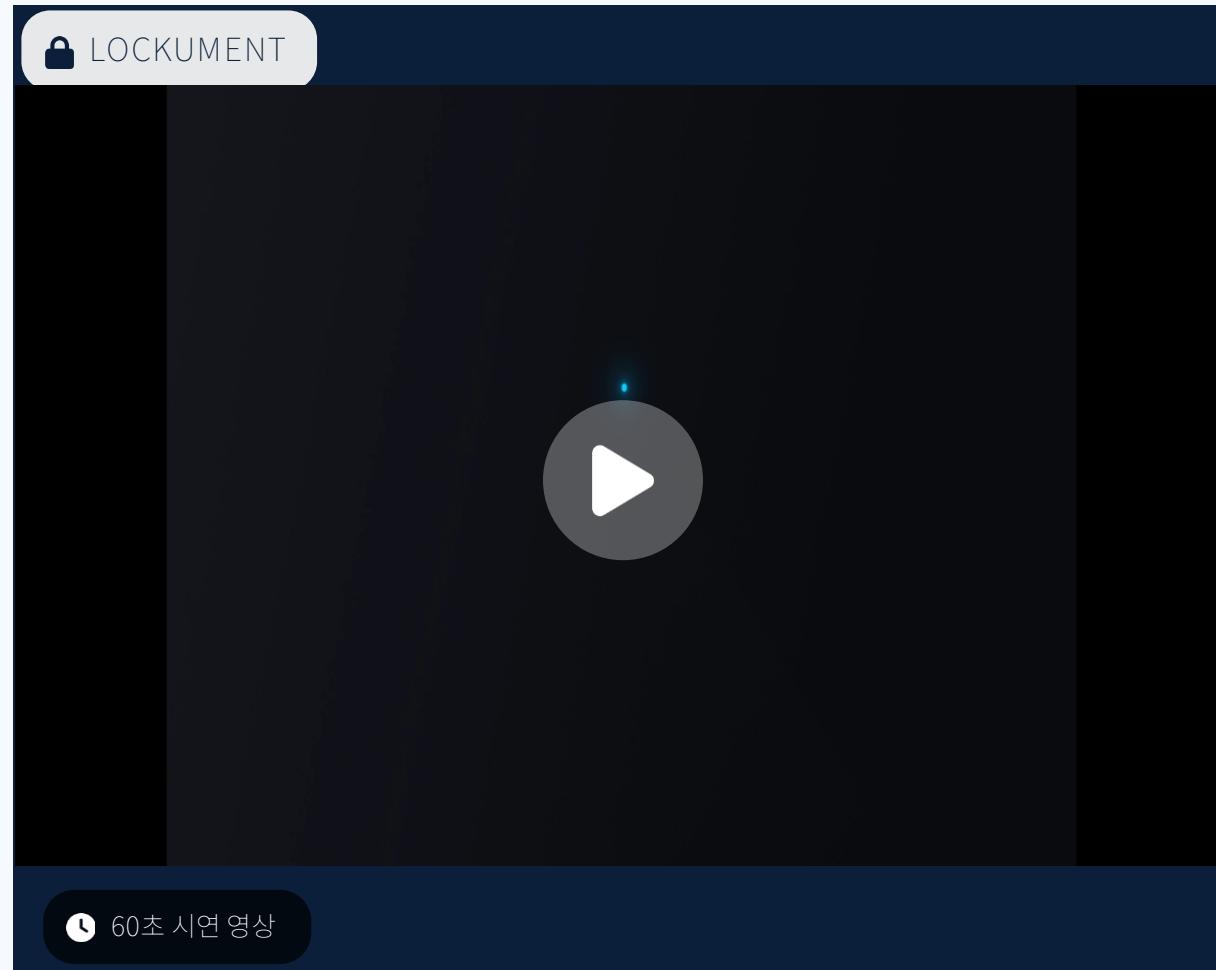
AI가 개인정보를 자동으로 탐지하고 마스킹 처리



③ 안전다운로드



마스킹 처리된 문서를 안전하게 다운로드



99.7%

개인정보 탐지 정확도

업계 최고 수준

0.8초

평균 응답속도

실시간 처리

30초

설치 완성 시간

즉시 사용 가능



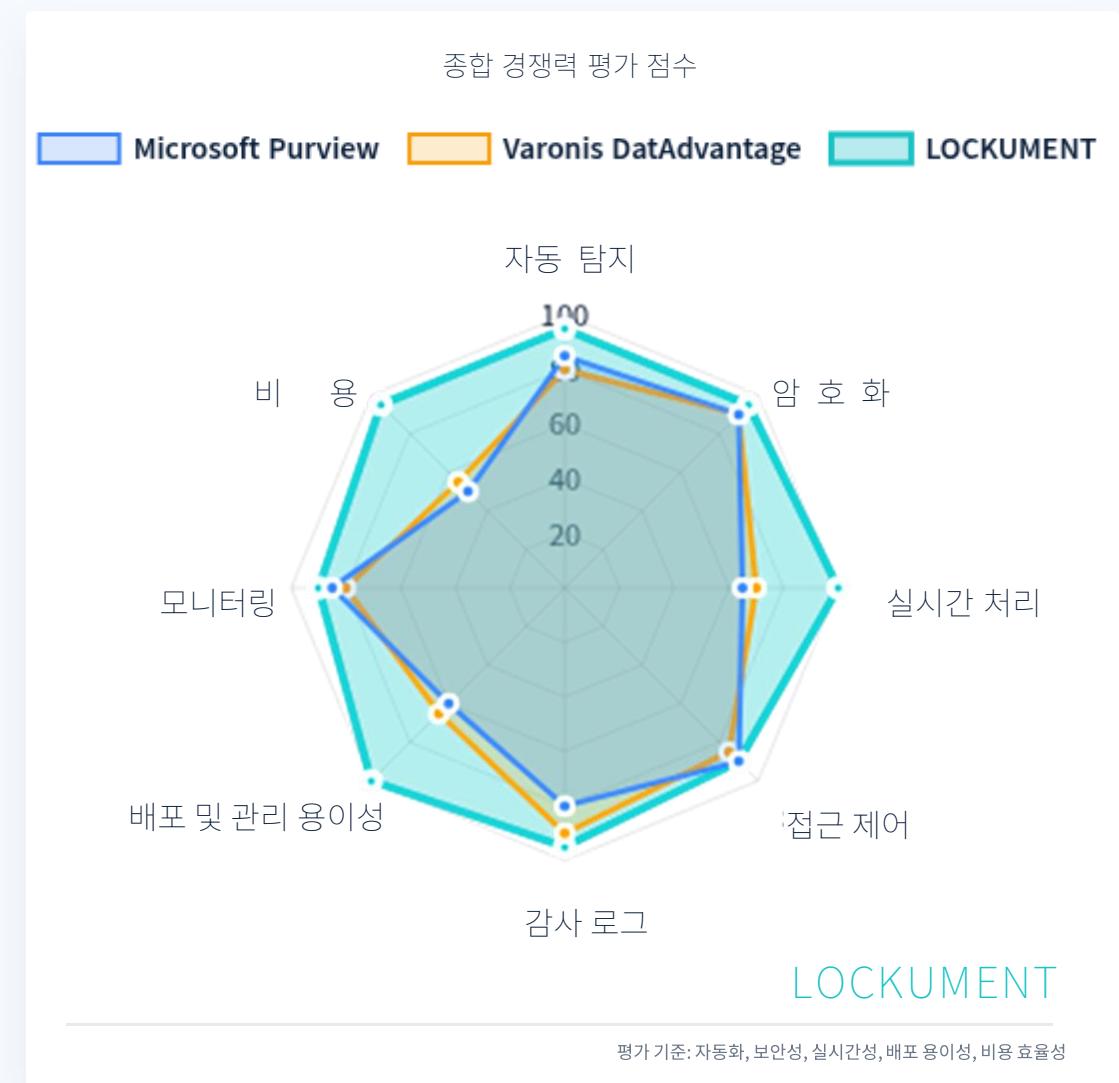
<https://lockument.duckdns.org>

비교 분석 및 차별화 전략

GUI 기반 개인정보 보호 솔루션 경쟁력 분석

평가 항목	Microsoft Purview	Varonis DatAdvantage	LOCKUMENT
자동 PII 탐지	✓ ML기반	✓ 패턴	✓ 3단계
암호화 수준	✓ AES-256	✓ AES-256	✓ AES-256-GCM
업로드 시점 처리	- 지연	- 스캔	✓ 실시간
접근 제어 방식	✓ RBAC	✓ 그룹	✓ Owner
감사 로그 범위	✓ 기본	✓ 상세	✓ 전체
온프레미스 배포	- 복잡	- 설치	✓ Docker
실시간 모니터링	✓ 대시보드	✓ 분석	✓ 헬스체크
비용 효율성	- 고가	- 고가	✓ 합리적

출처: LOCKUMENT GUI 경쟁사 조사 보고서 (2024.10) | 비교 대상: Microsoft Purview, Varonis DatAdvantage



참고문헌 및 출처 (References) 1

본 발표에 사용된 모든 참고문헌 및 출처 목록



법률 및 규제

개인정보보호법 제2조 (2020)

ISMS-P 인증기준 제3.2.3조 (2023)

GDPR Article 4 (2018)

개인정보보호위원회 가이드라인 (2024)



보안 표준 및 가이드라인

NIST SP 800-53 Rev. 5 (2023)

NIST SP 800-57 Part 1 Rev. 5 (2020)

OWASP API Security Top 10 (2023)

CIS Docker Benchmark v1.6.0 (2023)

NIST SP 800-63B (2017)

NIST SP 800-190 (2017)

OWASP Database Security Cheat Sheet (2024)

FIPS 140-3 (2019)



기술 문서 및 공식 문서

Flask 공식 문서 (2025)

PostgreSQL Connection Pooling 가이드 (2024)

JWT RFC 7519 (2015)

PostgreSQL 14 문서 (2024)

PostgreSQL 14 Row Level Security 문서 (2024)

React 공식 문서 (2025)

참고문헌 및 출처 (References) 2

본 발표에 사용된 모든 참고문헌 및 출처 목록



클라우드 및 인프라

- AWS KMS Developer Guide - Envelope Encryption (2025)
- AWS Security Best Practices for KMS (2025)
- Docker Security Best Practices (2025)
- Nginx Security Best Practices (2024)

- AWS KMS Best Practices (2025)
- AWS Security Roadmap (2025)
- Docker Rootless Mode Documentation (2024)



시장 조사 및 프로젝트 경험

- Gartner 2025 DLP Market Trends
- KISA 사이버 위협 동향 보고서 (2021-2025)
- 프로젝트 개발 과정 실제 경험 (2025년 9-10월)

- Gartner Magic Quadrant 2025
- 팀 내부 개발 계획 (2025)



총 30개 참고문헌 사용

개인정보 보호, 이제는 자동화의 시대입니다

LOCKUMENT와 함께 시작하는 안전한 디지털 전환

업로드 시점 자동 보호 · 소유자만 복호화 · 모든 시도 감사 기록



LOCKUMENT, 지금 바로 시작하세요

개인정보 유출 사고로 인한 법적 책임과 신뢰 손실을 방지하고,
자동화된 보안 체계로 업무 효율성과 안전성을 동시에 확보하세요



라이브 데모 시연



상세 문서 확인

"노출을 유출로 악화시키지 않는 것, 그것이 LOCKUMENT의 약속입니다"



즉시 배포
Docker 기반 30초 설치



데이터 주권
AWS CLOUD 보안



LOCKUMENT
Lock - Mask - Encrypt
Lock & Mask



규정 준수
개인정보보호법 완벽대응



지속 확장
모듈형 아키텍처

시나리오

웹구성 및 DB스키마제작

서상원

PPT편집

서상원

프로그래밍코드

김민진

후원

이철희

특별출연

박준형