M10-03 · GIBALJA VISUAL MANUAL

개인정보와 비밀정보를 안전하게 다루기

한 문장 목표: 실제 개인정보나 실제 key를 쓰지 않고, 합성 고객지원 서비스의 데이터가 어디로 퍼지는지 그린 뒤 24개 안전 계약과 증거로 release 여부를 판정합니다.

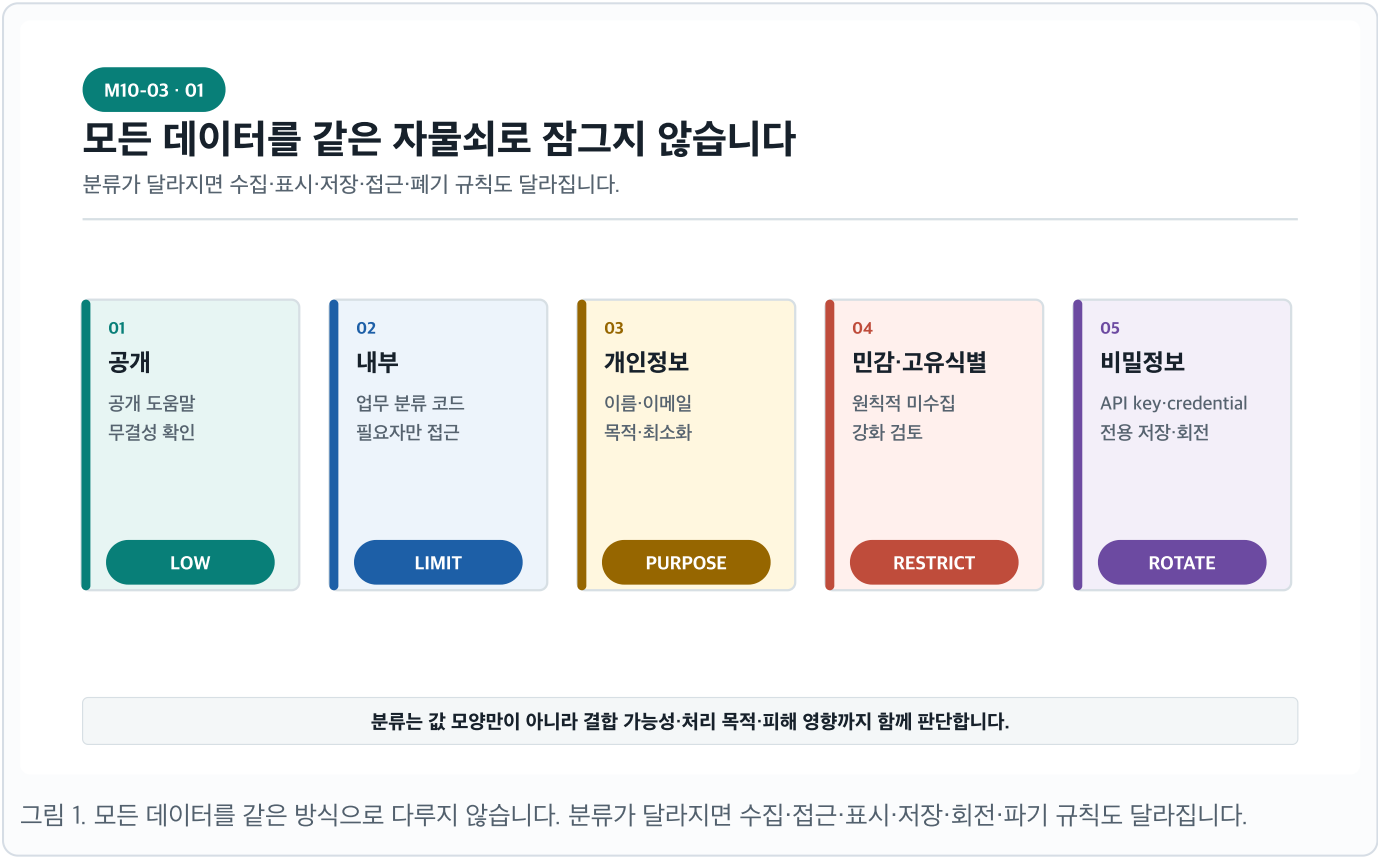


그림 1. 모든 데이터를 같은 방식으로 다루지 않습니다. 분류가 달라지면 수집·접근·표시·저장·회전·파기 규칙도 달라집니다.

분류가 달라지면 수집·표시·저장·접근·폐기 규칙도 달라집니다.

01

공개

공개 도움말
무결성 확인

LOW

02

내부

업무 분류 코드
필요자만 접근

LIMIT

03

개인정보

이름·이메일
목적·최소화

PURP

OSE

04

민감·고유식별

원칙적 미수집
강화 검토

RESTRICT

05

비밀정보

API key·credential
전용 저장·회전

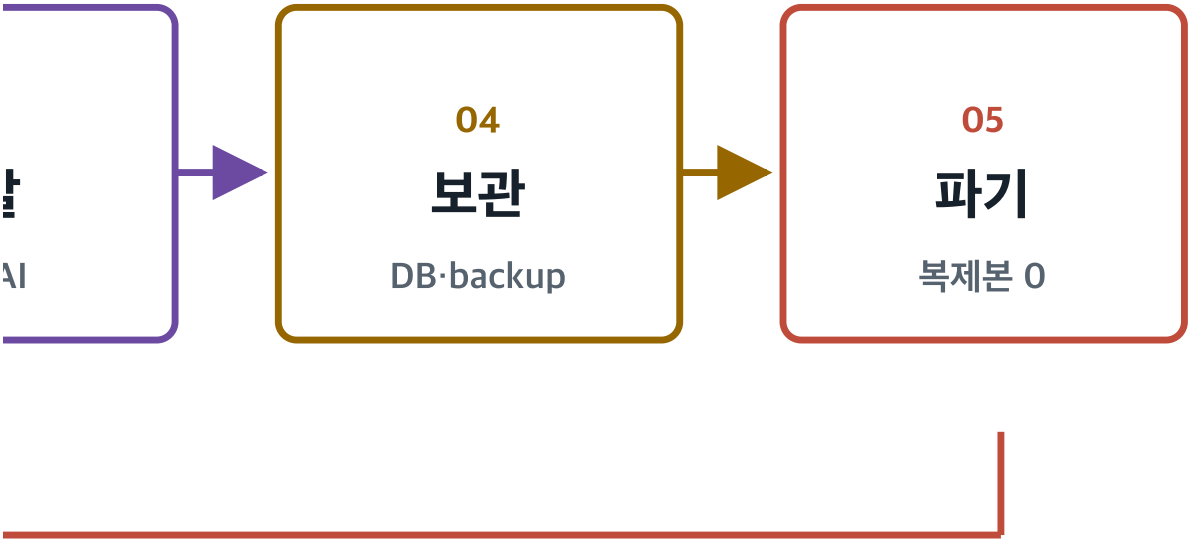
ROTATE

0. 한눈에 보는 두 번째 지도

그림 2. 개인정보는 한 칸에 머물지 않습니다. 수집부터 파기까지 원본과 복제본의 모든 이동 경로를 이어 봐야 합니다.

수집부터 파기까지 원본과 복제본의 모든 경계를 이어 봅니다.

변경·복원·사고가 생기면 전



체 생애주기를 다시 검수

난이도	그림 먼저	개념·판정	실습	셀프 테스트	최종 산출물
Level 2	35분	70분	90분	15분	처리 목록·흐름도·안전 처리 결과서

안전한 처리는 “암호화했다” 한 문장으로 끝나지 않습니다. 받지 않아도 되는 값을 받지 않고, 필요한 사람에게 필요한 일부만 보이고, 로그·오류·분석·알림·AI 도구로 원문이 새지 않게 하며, secret을 코드 밖에서 짧게 쓰고 회전하고, 기한이 끝난 원본과 복제본을 지웠다는 증거까지 이어져야 합니다. 이 교재는 교육용 개발 검수이며 법률 자문, 개인정보 영향평가, 보안 인증 또는 침투 테스트를 대신하지 않습니다.

0.1 첫 두 장에서 기억할 열 문장

```
분류하지 않은 데이터는 일관되게 보호할 수 없다.
처리 목록의 한 행은 항목·목적·주체·위치·전달·보존을 연결한다.
가장 안전한 필드는 목적이 없어 받지 않은 필드다.
마스킹·token·가명처리·암호화는 효과와 잔여 위험이 다르다.
업무 DB 밖의 로그·오류·분석·알림·AI·backup도 데이터 sink다.
좋은 로그는 원문이 적고 event·actor reference·result가 선명하다.
secret은 문자열이 아니라 생성부터 revoke까지의 생애주기다.
암호화 key는 암호문과 분리하고 접근·회전·폐기를 추적한다.
삭제 완료는 요청 접수가 아니라 적용 대상 copy가 0인 상태다.
Release Gate는 보호 조치 이름이 아니라 검증 가능한 증거 묶음이다.
```

1. 이 PDF를 공부하는 방법

1.1 1회차 · 그림 16장만 읽기 · 35분

그림 제목과 캡션만 읽고 다음 흐름을 말해 봅니다.

```
분류 → 목적·최소 수집 → 처리 목록
→ 요청·DB·응답·로그·분석·AI·backup 흐름
→ 표시·접근·암호화·key 분리
→ secret 주입·회전·revoke
→ 보존·파기·사고 대응 → Release Gate
```

1.2 2회차 · 민감 데이터 흐름 검수 스튜디오 · 55분

실습 생성기를 실행합니다.

```
./02_Labs/G10_Test_Quality_Security/L10-03_create-sensitive-data-secret-practice.sh
```

세 버전을 순서대로 비교합니다.

```
leaky-default-v1
  → partial-protection-v2
  → safety-candidate-v3
```

첫 버전은 6/24만 통과합니다. 두 번째는 16/24로 올라가지만 AI 전달, CI 출력, secret 회전·revoke, backup, 권리 요청, 대량 반출, 사고 증거에 8개 실패가 남습니다. 후보는 24/24와 6/6 회귀 계약을 만족해 `release_ready`가 됩니다.

1.3 3회차 · 내 서비스에 적용 · 120분

단계별 실습서를 따라 개인정보·비밀정보 안전 처리 결과서를 작성합니다. 낯선 표현은 개인정보·secret 안전 처리 용어집에서 찾습니다.

1.4 손의 순서를 고정합니다

먼저 하지 않을 것	먼저 할 것
“민감해 보이는 값”부터 암호화	처리 항목·목적·주체·흐름 목록 작성
화면 입력 필드만 확인	서버 허용 목록과 숨은 field 거절 확인
업무 DB만 점검	로그·오류·분석·알림·AI·backup sink 펼치기
마스킹과 암호화를 같은 뜻으로 사용	위협·목적·효과·재식별 가능성 구분
`.env`면 안전하다고 단정	저장·주입·출력·하위 process 노출 확인
삭제 API 응답으로 파기 완료 처리	원본·색인·cache·export·backup copy 확인
법률 조문을 체크박스로만 옮김	조직 적용 요건과 법무·개인정보 책임자 검토 연결

2. 학습 범위와 안전 경계를 고정합니다

2.1 이번 실습의 합성 시스템

```
service: synthetic_support_ticket
controls: 12
scenarios: 24
stages:
  inventory-collection: 6
  use-access-sharing: 6
  secret-lifecycle: 6
  retention-incident: 6
variants:
  leaky-default-v1: 6/24
  partial-protection-v2: 16/24
  safety-candidate-v3: 24/24
regression_contracts: 6/6
```

실 개인정보·실 secret·외부 네트워크·live AI 요청·공격 실행·실비용이 없습니다. 서버는 `127.0.0.1`에서만 열리고 trace에는 run ID·버전·개수·판정만 남습니다.

2.2 이번 매뉴얼이 주장하는 것과 주장하지 않는 것

증거로 주장할 수 있는 것	이 교재만으로 주장하지 않는 것
합성 계약에서 24개 expected가 충족됨	실제 서비스의 모든 개인정보 처리가 적법함
통제·시나리오·결과가 양방향 추적됨	개인정보 영향평가·인증·법률 검토가 완료됨
실데이터·실secret·live AI 사용 0	운영 인프라와 vendor가 모두 안전함
특정 version의 알려진 흐름을 검수함	알려지지 않은 취약점과 사고 가능성이 0

2.3 이전·다음 매뉴얼과의 경계

연결	가져오는 것	이번 매뉴얼이 더하는 것
M05-03 데이터 보존·백업·파기	일반 데이터 수명주기	개인정보·secret의 복제본·권리 요청·사고 증거 적용
M10-02 시나리오 검수	expected·actual·evidence·Gate	데이터 분류·흐름·노출·secret 생애주기 심화
M10-04 AI 위험 검수	다음 단계	여기서는 AI에 보내기 전 개인정보·secret 0만 확인

연결	가져오는 것	이번 매뉴얼이 더하는 것
법무·개인정보 보호책임자	조직별 적용 판단	교재 결과서를 검토 가능한 입력 자료로 제공

대한민국 개인정보 보호법과 하위 규정은 개정될 수 있고 산업·정보주체·처리 목적·국외 이전·위탁 구조에 따라 적용이 달라집니다. 결과서에는 확인일과 기준 version을 기록하고, 실제 배포 전 조직의 개인정보 보호책임자·법무·보안 담당자의 검토를 받습니다.

3. 다섯 분류로 처리 규칙을 시작합니다

3.1 공개·내부·개인정보·민감·비밀정보

분류	합성 예	핵심 질문	기본 처리 방향
공개	공개 도움말 제목	공개해도 되는가, 변조되지 않았는가	공개 범위와 무결성
내부	상담 분류 code	업무상 누가 알아야 하는가	필요자 접근·외부 전달 제한
개인정보	이름·email·문의	살아 있는 개인과 연결되는가	목적·최소 수집·권리·보존
민감·고유식별	건강 내용·고유식별정보	더 큰 피해와 별도 요건이 있는가	원칙적 미수집·강화 검토
비밀정보	API key·DB credential	시스템 권한을 열 수 있는가	전용 저장·최소 권한·회전·revoke

개인정보와 secret은 겹칠 수 있지만 같은 개념은 아닙니다. 사용자 password는 개인과 연결되고 인증 권한도 열 수 있으므로 두 관점의 통제가 함께 필요합니다.

3.2 값 하나가 아니라 맥락과 결합을 봅니다

`support-42` 만으로 특정인을 바로 알아보기 어려워도 다른 표와 결합하면 개인과 연결될 수 있습니다. 반대로 공개 이름이라도 건강 상담 내용과 결합되면 영향이 커집니다. 다음을 함께 봅니다.

- 직접 식별자와 간접 식별자.
- 다른 데이터와 결합 가능성.
- 처리 목적과 예상 사용자.
- 노출 시 개인·조직·시스템의 피해.
- 되돌릴 수 있는지와 회복 비용.

3.3 분류 결정에 owner와 날짜를 붙입니다

필드	예
Classification	`personal`
Reason	회신 email로 개인과 직접 연결
Owner	privacy-owner
Approved at	2026-07-16
Review trigger	목적·vendor·보존·schema 변경

분류는 영구 라벨이 아닙니다. 결합 데이터, 처리 목적, 외부 전달처가 바뀌면 다시 검토합니다.

4. 처리 목록의 한 행으로 전체 흐름을 연결합니다

그림 3. 목록에 없는 field·저장소·전달처는 통제되지 않는 처리입니다. 한 행에서 왜 받고 어디로 가며 언제 지우는지 답할 수 있어야 합니다.

개인정보 항목 분류체계를 만들어

항목 이름만 적지 말고 목적·주체·위치·전달·보존을 한 줄로 연결합니다

항목	분류	목적	주체
email	개인정보	문의 회신	고객·상담자

OWNER · evide

	위치	전달	보존
	support DB	메일 처리자	종료+30일

nce · 변경일 연결

4.1 최소 처리 목록

항목	질문	합성 예
Field	어떤 값인가	`email`
Classification	어떤 분류인가	personal
Purpose	왜 필요한가	문의 회신
Basis·approval	어떤 조직 근거인가	승인된 서비스 목적
Data subject	누구의 정보인가	고객
Actors	누가 보는가	배정 상담자
Source	어디서 들어오는가	support form
Stores	어디에 저장되는가	support DB
Sinks	어디로 전달되는가	mail processor
Protection	어떻게 줄이고 보호하는가	화면 masking·TLS·저장 보호
Retention	언제 어떤 예외로 지우는가	종료+30일·승인된 hold 예외
Owner	누가 변경과 증거를 책임지는가	privacy-owner

4.2 schema와 처리 목록을 비교합니다

다음 세 집합을 비교합니다.

```
A = 승인된 처리 목록의 field
B = request·database·event schema의 field
C = log·analytics·AI·export에 실제 나타난 field
```

`B − A`는 승인되지 않은 수집·저장 후보이고, `C − A`는 승인되지 않은 전달·복제 후보입니다. `A − B`는 문서와 구현의 불일치입니다.

4.3 보이지 않는 복제본을 찾습니다

- request body와 reverse proxy log.
- database와 검색 색인.

- application error와 APM event.
- analytics payload와 dashboard export.
- notification body와 delivery history.
- AI prompt·tool call·evaluation dataset.
- cache·temporary file·local download.
- backup·snapshot·restore environment.

5. 목적과 최소 수집으로 입구를 좁힙니다

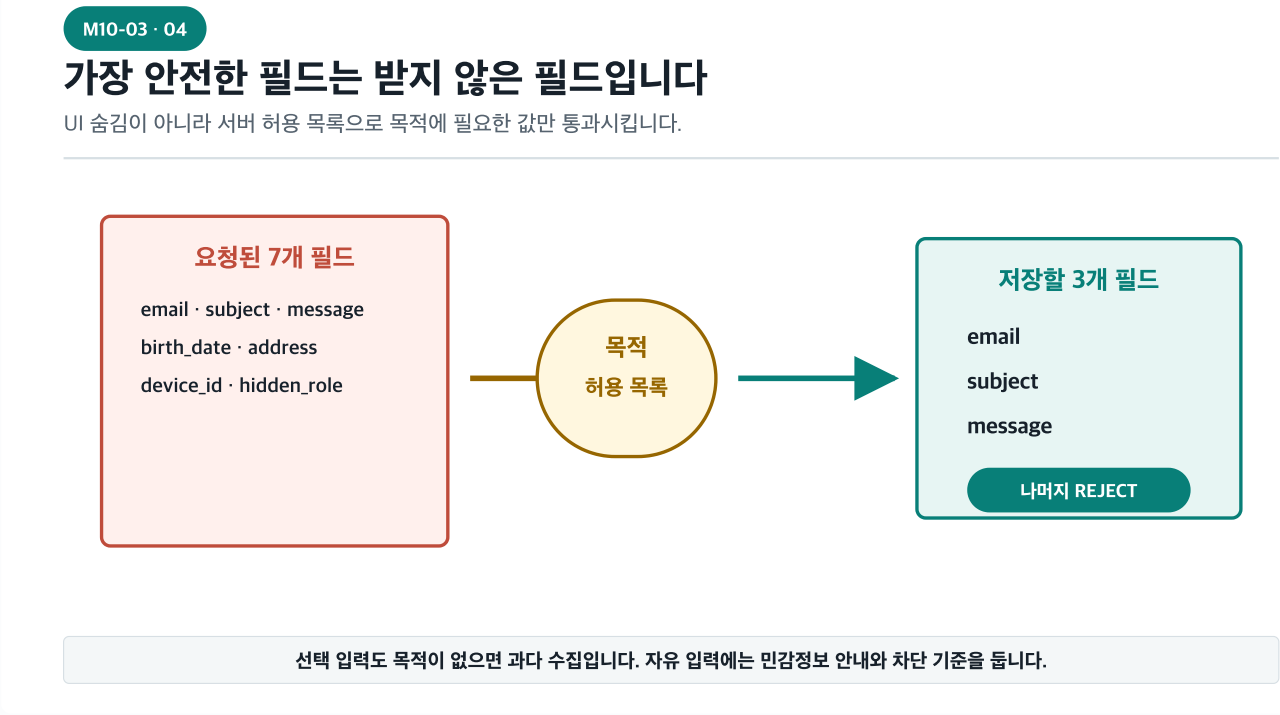


그림 4. 화면에서 숨기는 것만으로는 충분하지 않습니다. 서버가 목적에 필요한 field만 허용하고 숨은 field를 거절해야 합니다.

UI 숨김이 아니라 서버 허용 목록으로 목적에 필요한 값만 통과시킵니다

요청된 7개 필드

email · subject · message

birth_date · address

device_id · hidden_role

목적

허용 목록


저장할 3개 필드
email
subject
message
나머지 REJECT

5.1 목적 문장은 기능이 아니라 결과로 씁니다

나쁜 목적은 “서비스 제공”처럼 너무 넓습니다. 좋은 목적은 field가 왜 필요한지 판정할 수 있습니다.

넓은 표현	검수 가능한 표현
고객 관리	문의 답변을 email로 발송
서비스 개선	주간 오류율을 개인 식별자 없이 집계
AI 활용	문의의 제품 category를 합성·비식별 입력으로 분류

5.2 field마다 네 질문을 합니다

1. 이 field가 없으면 사용자 목표를 달성할 수 없는가.
2. 더 낮은 민감도의 대체값으로 바꿀 수 있는가.
3. 입력 시점이 아니라 나중에 필요할 때 받을 수 있는가.
4. 원문 대신 reference·범주·집계값을 쓸 수 있는가.

5.3 자유 입력은 별도 위험입니다

문의 본문 같은 자유 입력에는 사용자가 예상하지 못한 건강·금융·고유식별·secret을 붙일 수 있습니다. 안내 문구만 두지 말고 다음을 연결합니다.

- 수집하지 말아야 할 정보의 짧은 안내.
- 고위험 pattern의 합성 탐지와 저장 전 차단.
- 탐지가 틀릴 때 안전하게 수정할 경로.
- 차단 event에는 원문을 남기지 않는 로그.
- 운영 검토가 필요한 경우 최소 권한 queue.

5.4 SC-02와 SC-03의 판정

시나리오	안전 기대
SC-02 생년월일 과다 수집	`birth_date` 거절·세 필드만 수용
SC-03 건강 정보 자유 입력	저장 0·안내 표시·검토 evidence

6. 보호 기법을 목적에 맞게 구분합니다

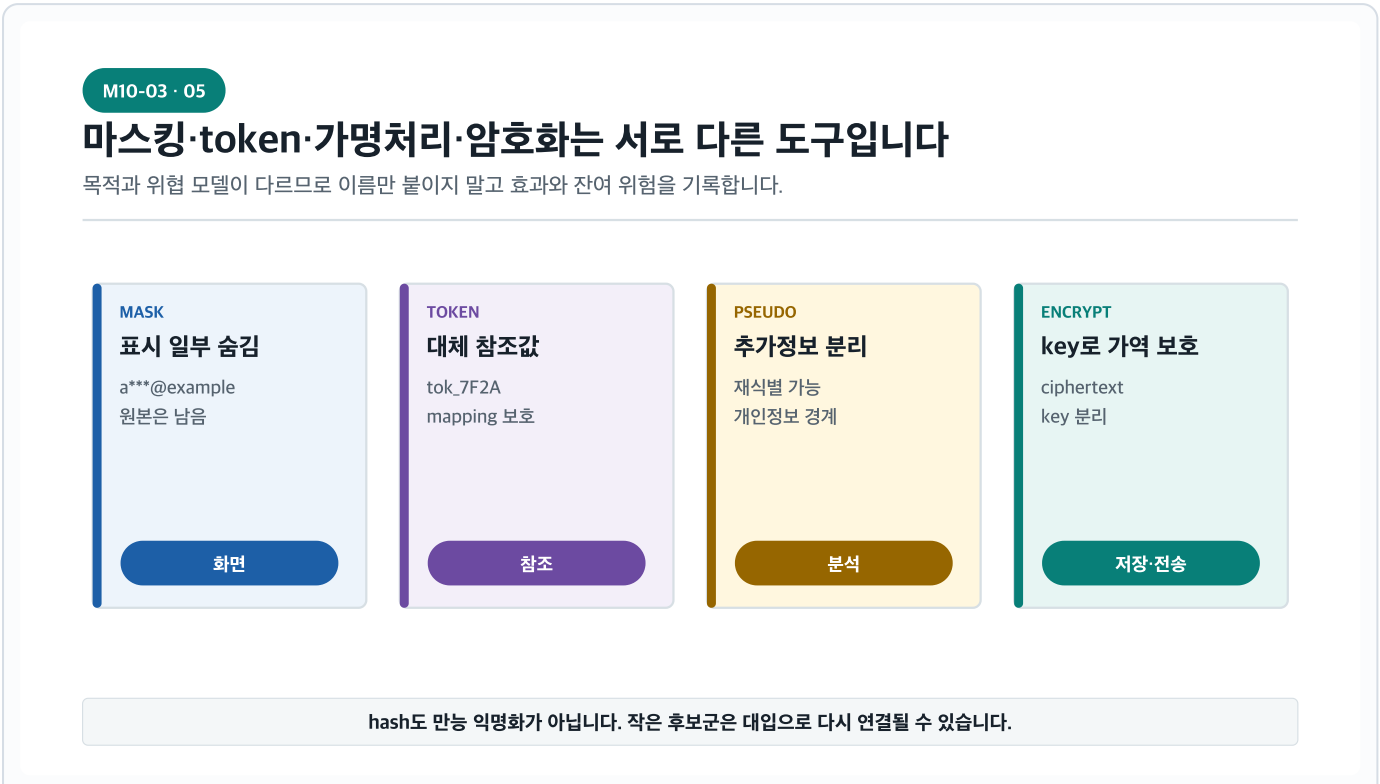


그림 5. 이름이 비슷해 보여도 보호 효과는 다릅니다. 원본이 남는지, 재연결할 수 있는지, key가 필요한지를 기록합니다.

목적과 위협 모델이 다르므로 이름만 붙이지 말고 효과와 잔여 위험을 ㅈ

MASK

표시 일부 숨김

a***@example

원본은 남음

화면

TOKEN

대체 참조값

tok_7F2A

mapping 보호

참조

기록합니다.

PSEUDO

추가정보 분리

재식별 가능
개인정보 경계

분석

ENCRYPT

key로 가역 보호

ciphertext
key 분리

저장·전송

6.1 네 기법 비교

기법	주 목적	원본 복원·연결	핵심 잔여 위험
Masking	화면·문서에 일부만 표시	원본은 다른 곳에 남음	backend·export·개발자 도구 노출
Tokenization	원본 대신 reference 사용	mapping으로 재연결 가능	mapping store와 권한 노출
가명처리	추가정보 없이는 개인 식별을 어렵게 함	추가정보로 재식별 가능	결합·추론·추가정보 노출
Encryption	key 없이는 원문 해석을 어렵게 함	key로 복호화 가능	key·권한·runtime compromise

6.2 hash를 익명화와 혼동하지 않습니다

Email·전화번호처럼 후보 공간이 작거나 예측 가능한 값은 단순 hash만으로 다시 대입될 수 있습니다. salt·keyed hash·tokenization·집계 등 목적에 맞는 방법과 재식별 위험을 검토합니다.

6.3 password는 되돌릴 필요가 없습니다

Password는 원문을 복원해 비교하지 않습니다. 검증된 password hashing 방식과 적절한 work factor를 사용합니다. 암호화와 password hashing은 목적이 다릅니다.

6.4 “암호화됨”을 검수 가능한 문장으로 바꿉니다

```
asset: support_message
state: at-rest
threat: storage media disclosure
control: approved authenticated encryption
key_location: separate key service
key_owner: security-owner
rotation_trigger: schedule or suspected exposure
evidence: configuration ID + test result
```

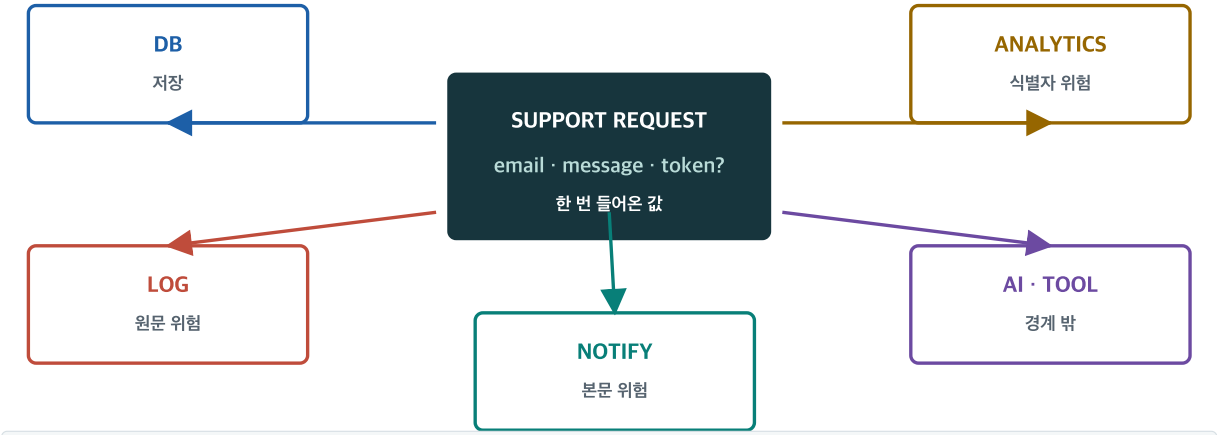
구체 알고리즘·key 길이·module 요구는 조직의 보안 표준과 현재 공식 지침을 따릅니다. 교재 예시는 암호 구현 코드를 직접 만들게 하지 않습니다.

7. 한 번 들어온 값의 확산 지도를 그립니다

M10-03 · 06

한 번 받은 값은 다섯 갈래로 복제될 수 있습니다

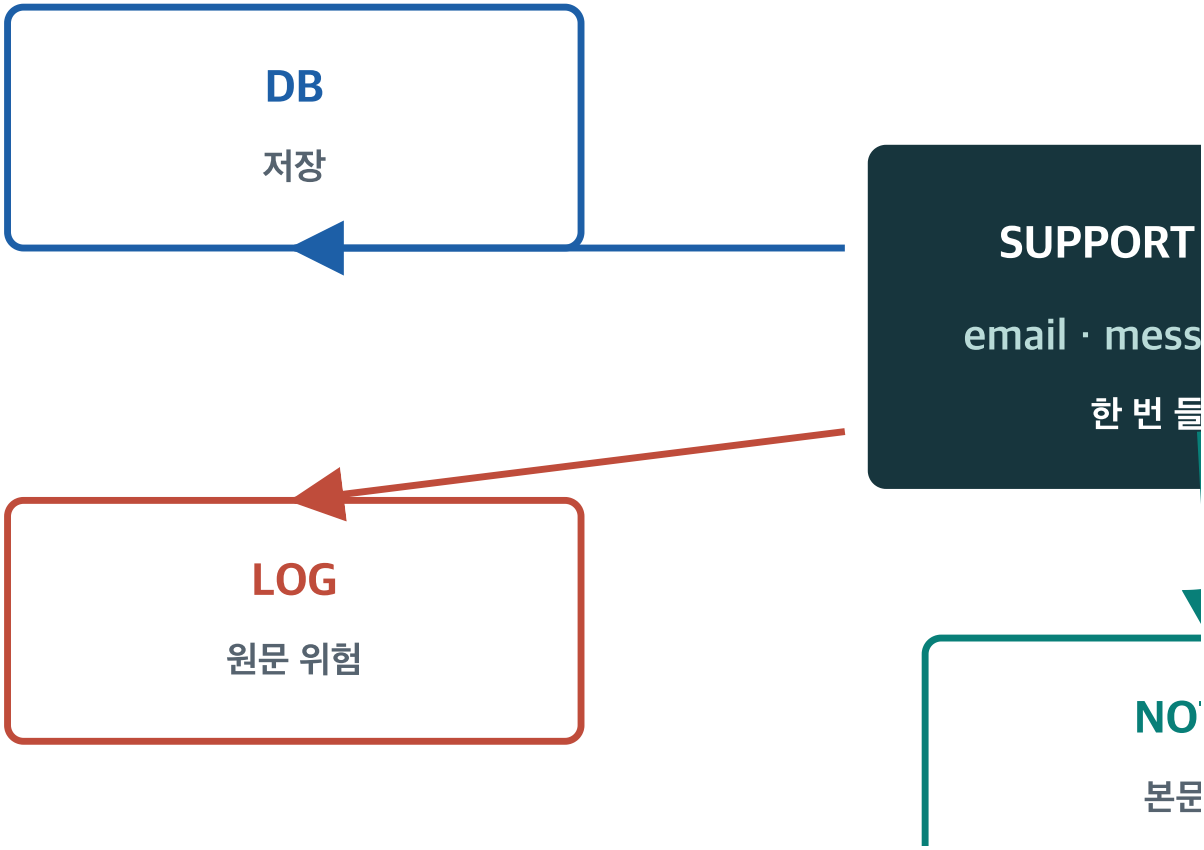
업무 DB만 보지 말고 로그·분석·알림·AI 도구까지 sink를 펼쳐 봅니다.

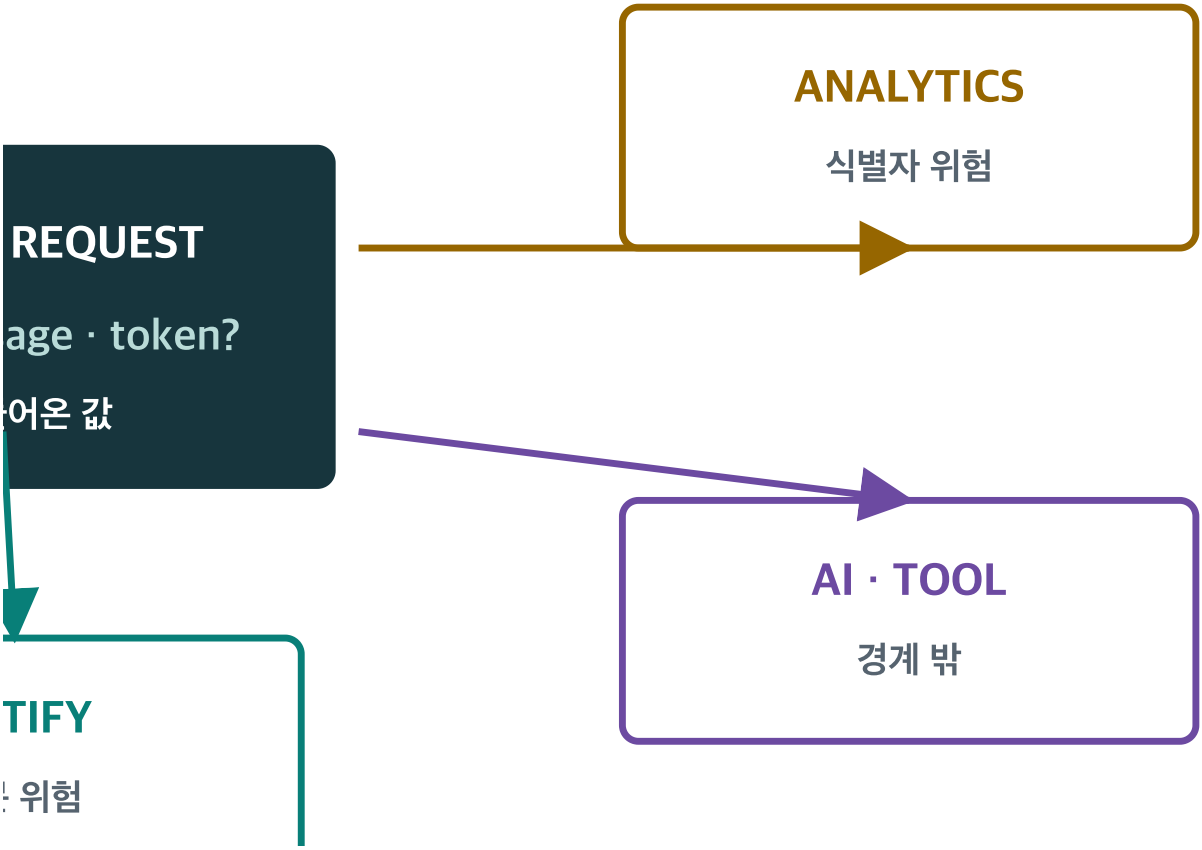


각 sink마다 필요한 필드, 보호 방식, 보존 시간, owner가 달라야 데이터 확산을 통제할 수 있습니다.

그림 6. 업무 DB만 안전해도 충분하지 않습니다. 같은 원문이 로그·분석·알림·AI에 복제되면 전체 노출면이 커집니다.

업무 DB만 보지 말고 로그·분석·알림·AI 도구까지 sink를 펼쳐 봅니다.




ANALYTICS
식별자 위험
REQUEST
age · token?
어온 값
AI · TOOL
경계 밖
TIFY
위험

7.1 흐름 화살표의 최소 표기

```
[source] -- field subset / purpose / protection --> [sink]
           owner / retention / evidence
```

예를 들면 다음과 같습니다.

```
support-form -- email,message / 문의 처리 / TLS --> support-api
support-api -- ticket_ref,status / 현황 표시 --> list-response
support-api -- event,actor_ref,result / 감사 --> security-log
support-api -- category only / 통계 --> analytics
```

7.2 sink마다 다시 묻습니다

Sink	최소 질문
Request	허용 field와 크기·형식·자유 입력 경계가 있는가
Database	column·row 접근과 저장 보호·보존 시계가 있는가
Response	화면 목적에 필요한 field만 반환하는가
Log·error	원문·token·key·stack을 제거하는가
Analytics	개인 식별자 없이 목적을 달성할 수 있는가
Notification	제목·본문·수신자에 원문이 필요한가
AI·external tool	목적·계약·보존·학습 사용·차단을 확인했는가
Backup·export	대량 복제 승인·만료·복원 뒤 재삭제가 있는가

7.3 Data Flow Diagram과 처리 목록을 왕복합니다

그림은 누락된 경로를 찾는 데 강하고 표는 owner·근거·보존을 관리하는 데 강합니다. 흐름도에 있는 화살표는 처리 목록의 행으로, 처리 목록의 sink는 흐름도의 화살표로 서로 연결합니다.

8. 로그와 오류를 사건 추적용으로 줄입니다

그림 7. 좋은 로그는 사건을 재구성할 수 있으면서 원문은 최소입니다. 로그 자체도 별도의 민감 저장소로 관리합니다.

[illegible]

password·token·key·민감 원문을 빼고 누가 무엇을 어떤 결과로 했는지

BEFORE

원문 로그

email=hana@example.test
message=환불 계좌…
Authorization=Bearer…
stack=/srv/app.py:88

REDU

로그입니다

디 남깁니다.

JCE

AFTER

사건 추적 로그

event=ticket_view
actor_ref=usr_42f
result=DENY
correlation_ref=req_91a

8.1 보통 직접 남기지 않을 값

- 인증 password·session ID·access token.
- API key·database connection string·암호화 key.
- 민감정보·고유식별정보와 필요 없는 개인정보 원문.
- 결제·계좌·카드 정보.
- 요청·응답 body 전체.
- stack trace·query value·내부 file path의 외부 응답.

8.2 사건 추적에 남길 구조

```
{
  "event": "ticket_view",
  "actor_ref": "usr_42f",
  "object_ref": "tkt_91a",
  "result": "DENY",
  "reason_code": "NOT_ASSIGNED",
  "correlation_ref": "req_7b2",
  "occurred_at": "synthetic-clock"
}
```

실습의 reference는 합성 값입니다. 운영에서는 reference 자체의 재연결 가능성, 접근 권한, 보존 기간을 별도로 검토합니다.

8.3 redaction은 입력점 하나에만 두지 않습니다

요청 수신, application logger, exception handler, APM agent, proxy, queue, analytics exporter의 경계를 각각 확인합니다. 한 logger가 안전해도 다른 library가 body를 자동 수집할 수 있습니다.

8.4 SC-10과 SC-11

시나리오	원문에서 제거	대신 남길 것
SC-10 로그 안전	email·message·token	event ID·result·reference
SC-11 오류 안전	stack·query value	일반화 code·correlation reference

9. 외부와 AI 전달은 새 경계로 봅니다

그림 8. 복사·붙여넣기도 처리 경계를 넘는 전달입니다. 보내기 전에 목적·필드·계약·보존·학습 사용·차단 기준을 확인합니다.

편리한 복사·붙여넣기 전에 목적·필드·계약·보존·학습 사용 여부를 확인

INSIDE

업무 경계

원본 문의

email · message

내부 분류

목적=문의 처리

BOUNDAR

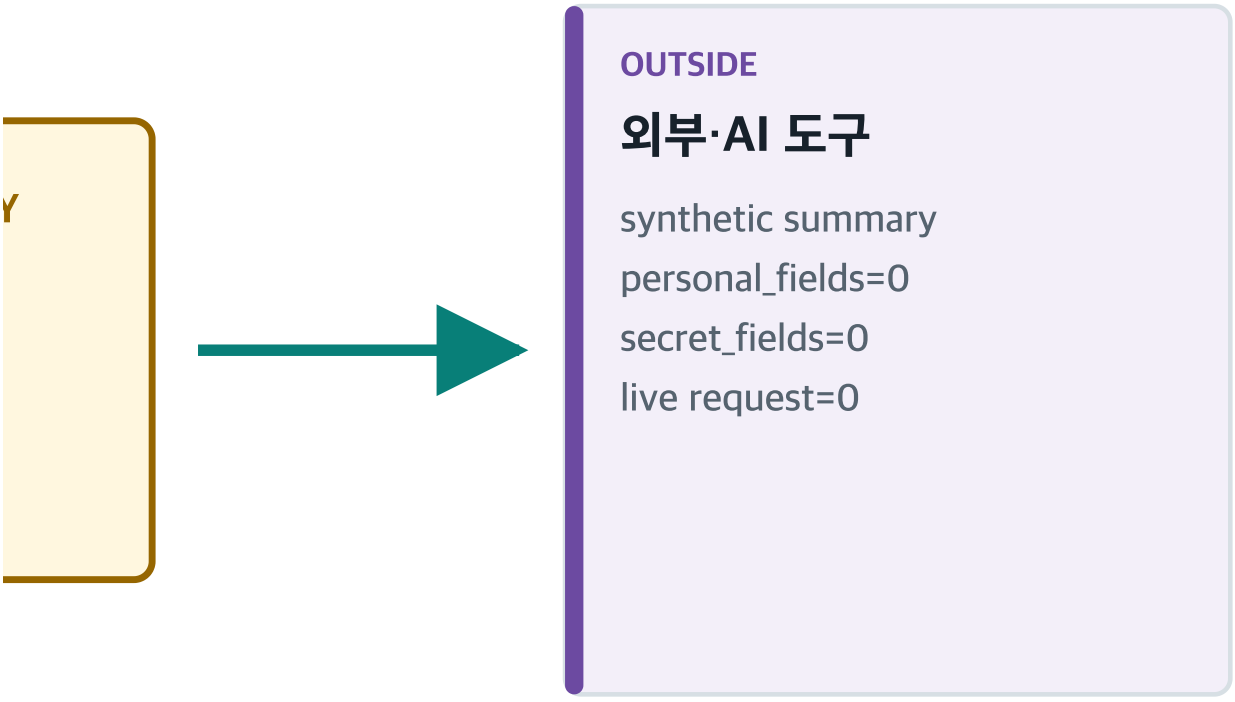
목적 확인

필드 제거

계약·보존

차단 기준

합니다.


OUTSIDE
외부·AI 도구
synthetic summary
personal_fields=0
secret_fields=0
live request=0

9.1 전달 전 여덟 질문

1. 외부 처리가 정말 필요한가.
2. 보내는 정확한 field는 무엇인가.
3. 개인 식별자를 제거하거나 local 처리할 수 있는가.
4. secret·credential·내부 문서가 섞이지 않는가.
5. 저장 위치·보존 기간·삭제·재처리 조건은 무엇인가.
6. 하위 처리자·국외 이전·계약·조직 승인 요건은 무엇인가.
7. 입력이 모델 학습·평가·운영 개선에 사용되는가.
8. vendor 설정·모델·목적이 바뀔 때 재검 trigger가 있는가.

9.2 SC-12의 안전 경계

```
input: synthetic support ticket
personal_fields: 0
secret_fields: 0
live_request: false
result: AI_PAYLOAD_REDACTED
```

이 시나리오는 AI 공격을 검수하지 않습니다. Prompt injection, 환각, tool misuse, model output 정보 유출은 M10-04의 범위입니다.

9.3 외부 처리자의 증거도 연결합니다

증거	예
처리 목적·field 목록	approved data-flow row
계약·승인	contract ID·privacy review ID
설정 snapshot	retention·training·region setting
전송 검증	synthetic payload test
삭제·반환	deletion evidence·termination plan
변경 감시	vendor·subprocessor·model change trigger

10. secret을 저장소 밖에서 실행 시 주입합니다

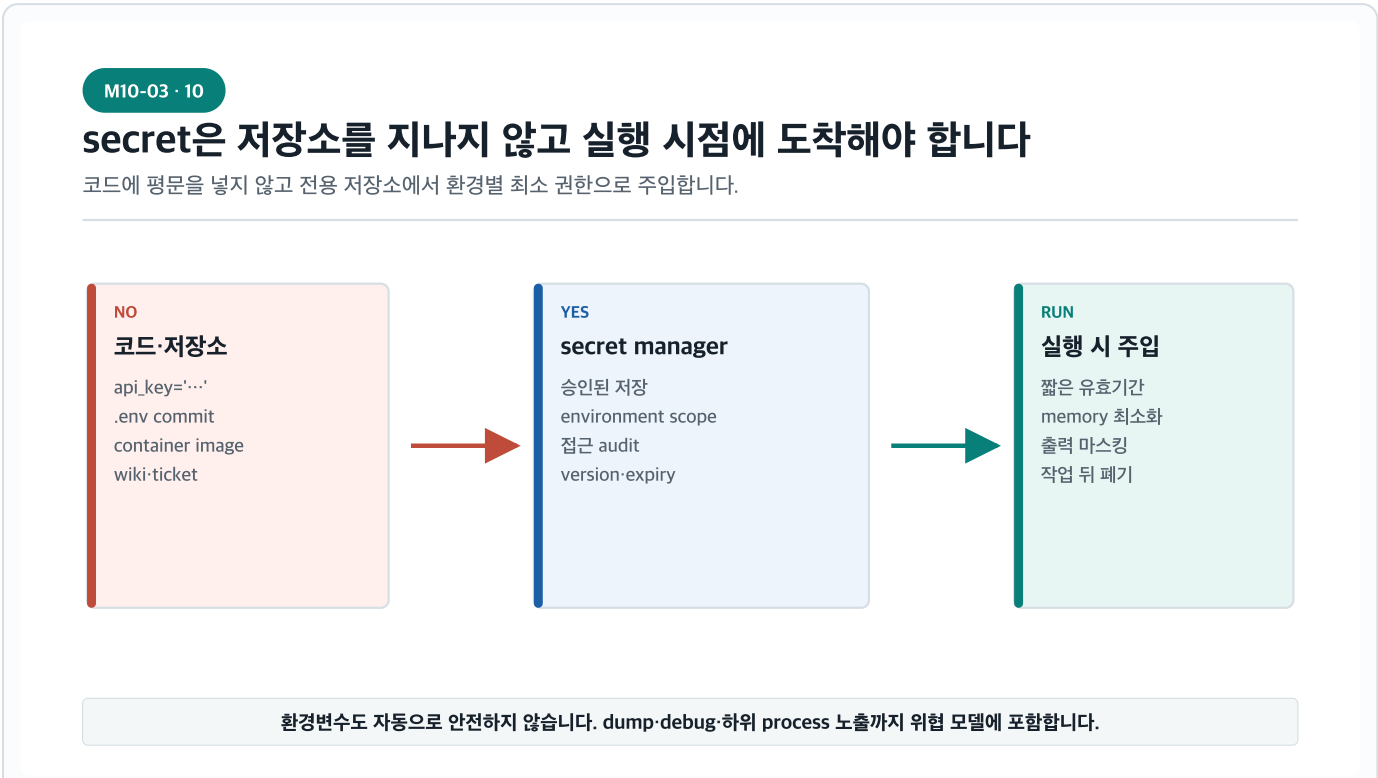


그림 9. secret은 source repository·image·wiki·ticket을 통과하지 않고 승인된 저장소에서 환경별 최소 권한으로 주입합니다.

코드에 평문을 넣지 않고 전용 저장소에서 환경별 최소 권한으로 주입합

NO

코드·저장소

api_key='…'

.env commit

container image

wiki·ticket

YES

secret mana

승인된 저장

environment sc

접근 audit

version·expiry

[illegible]니다.

[illegible]ger

[illegible]ope

RUN

실행 시 주입

짧은 유효기간
memory 최소화
출력 마스킹
작업 뒤 폐기

10.1 secret의 예

- API key와 access token.
- Database username·password·connection credential.
- SSH private key와 signing key.
- TLS private key와 client certificate.
- Webhook signing secret.
- Encryption key와 key-encryption key.
- CI/CD deploy credential.
- Break-glass credential과 recovery code.

Endpoint URL이나 public certificate처럼 공개 가능한 설정은 secret이 아닐 수 있습니다. 무엇이 권한을 열고 누출 시 어떤 blast radius를 만드는지로 분류합니다.

10.2 code와 configuration을 검사합니다

위치	확인할 것
Source	hard-coded 값·예제 key·test fixture
Git history	삭제 전 commit과 fork·mirror
Build artifact	container layer·package·source map
CI/CD	variable scope·fork·debug output·artifact
Runtime	process argument·environment dump·crash dump
Collaboration	ticket·wiki·chat·screenshot·recording

이미 commit된 secret은 파일에서 지우는 것만으로 끝나지 않습니다. 노출된 것으로 간주하고 revoke·재발급·영향 범위 확인·history와 artifact 처리·재발 방지를 진행합니다.

10.3 환경변수는 운반 방식이지 완전한 금고가 아닙니다

환경변수는 code hard-coding보다 낫지만 process dump, debug endpoint, 하위 process, `/proc` 같은 runtime 표면에 노출될 수 있습니다. 플랫폼의 전용 secret 기능, file mount, identity-based short-lived credential 등 현재 조직 표준과 위협 모델에 맞는 방식을 선택합니다.

10.4 SC-13~SC-15

ID	안전 계약
SC-13	repository hit 0·runtime injection true
SC-14	CI 출력 secret value false·metadata true
SC-15	dev·test·prod shared secret false·least privilege true

11. secret을 생애주기로 운영합니다

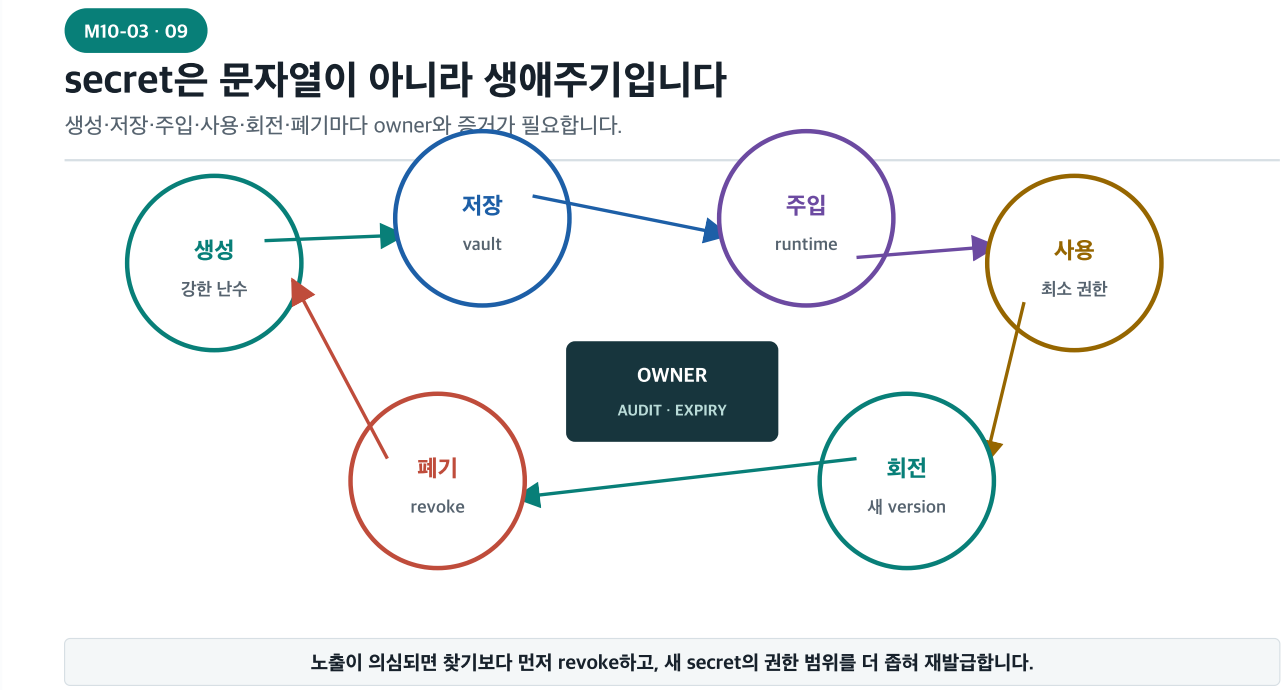
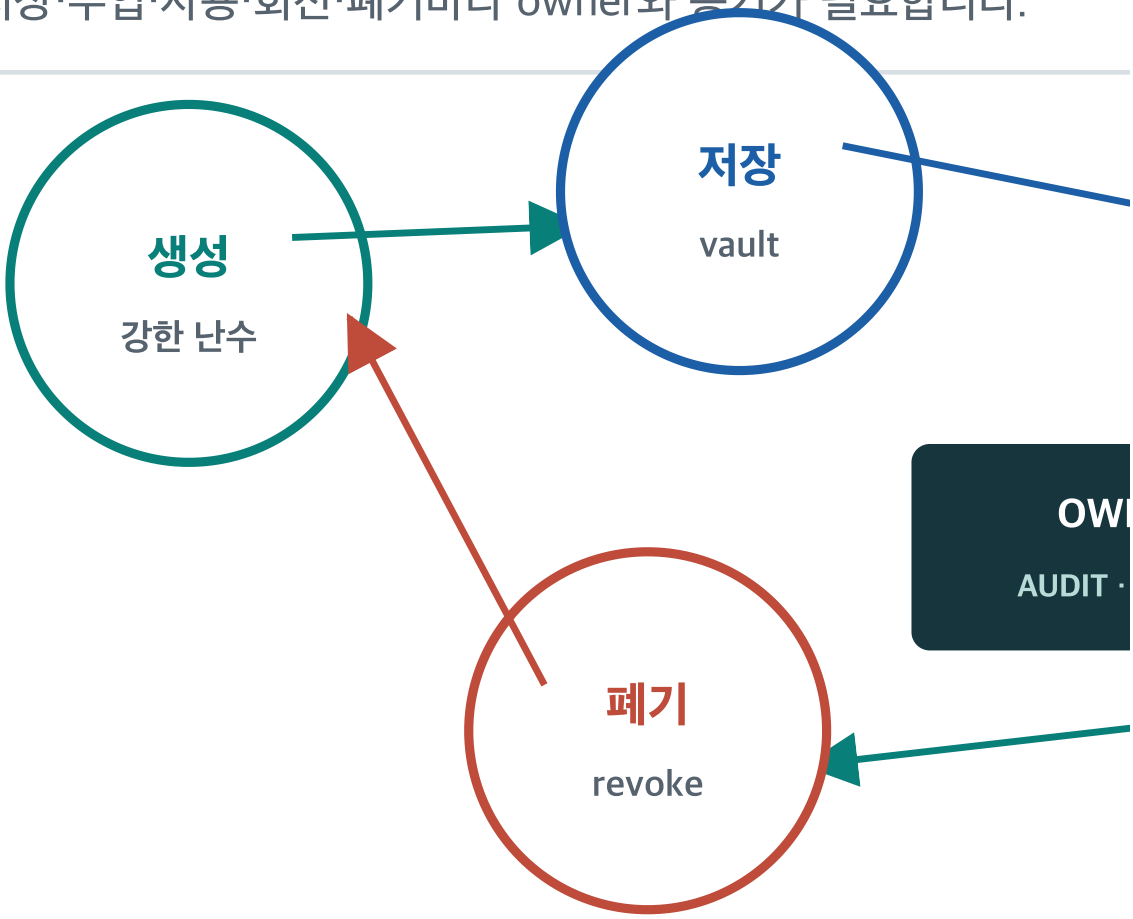
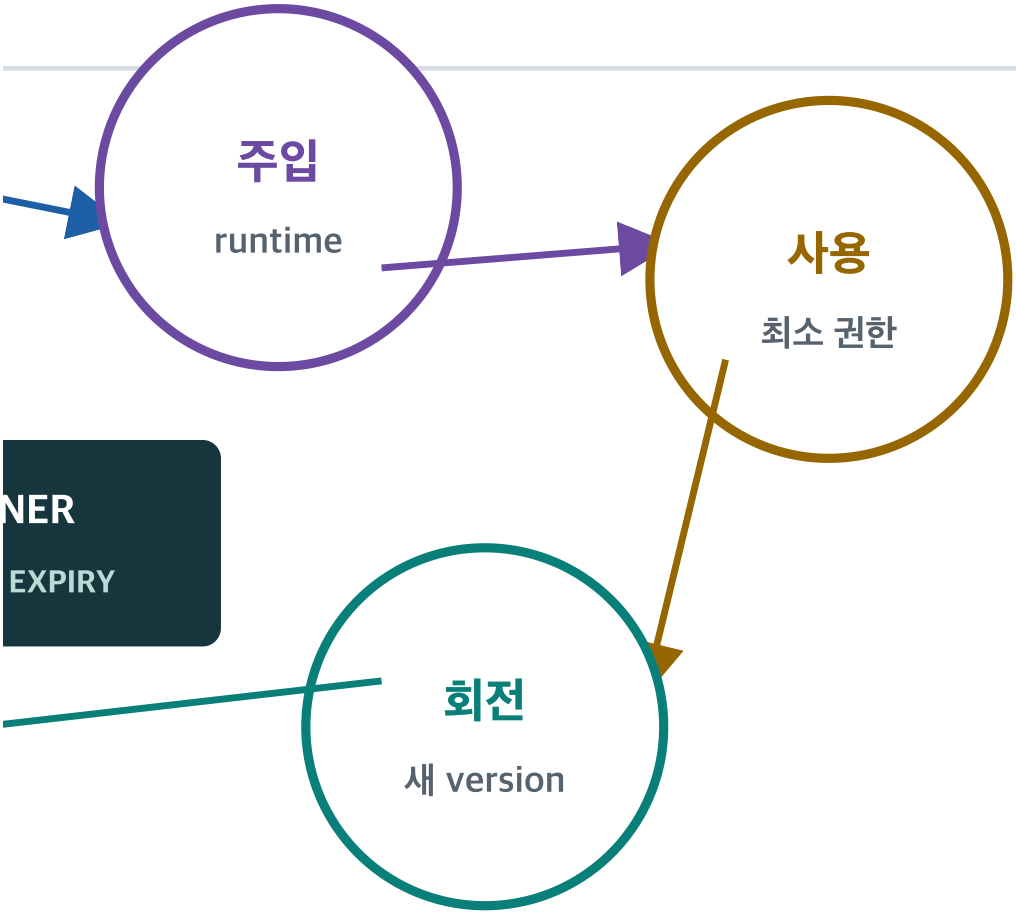


그림 10. secret은 owner·사용처·만료·version·회전·revoke·복구 증거를 가진 운영 자산입니다.

생성·저장·주입·사용·회전·폐기마다 owner와 증거가 필요합니다.




주입
runtime
사용
최소 권한
NER
EXPIRY
회전
새 version

11.1 여섯 단계

단계	검수 질문	증거
생성	충분히 강한 방식으로 생성됐는가	approved generator·policy
저장	전용 저장소와 접근 통제가 있는가	vault reference·ACL
주입	필요한 workload에 짧게 전달되는가	workload identity·mount
사용	목적·환경·권한 범위가 최소인가	audit·scope
회전	중단 없이 새 version으로 바꿀 수 있는가	rotation test
폐기	이전 version이 더 이상 유효하지 않은가	revoke probe·event

11.2 회전 trigger

- 정해진 cryptoperiod 또는 조직 정책 기한.
- 노출이 확인되거나 의심됨.
- 권한 보유자의 역할 변경·퇴사.
- Vendor·환경·목적·scope 변경.
- 알고리즘·library·key management 표준 변경.
- 예상보다 큰 사용량이나 비정상 접근.

11.3 회전은 새 값을 만드는 것보다 넓습니다

새 version 생성, consumer 갱신, 짧은 이중 유효 구간, 건강 확인, 이전 version revoke, cache·worker 반영, rollback 기준, 증거 보존을 한 runbook으로 묶습니다.

11.4 SC-16과 SC-17

시나리오	PASS 조건
자동 회전	new active·old retired·downtime false
노출 의심 revoke	old valid false·new scoped true·blast radius recorded

노출 의심 사건에서 “먼저 원인을 완벽히 찾고 나중에 회전”하지 않습니다. 재사용 가능한 secret이면 우선 접근을 봉쇄하고 revoke한 뒤 영향 범위를 조사합니다.

12. 암호화 key와 데이터를 분리합니다

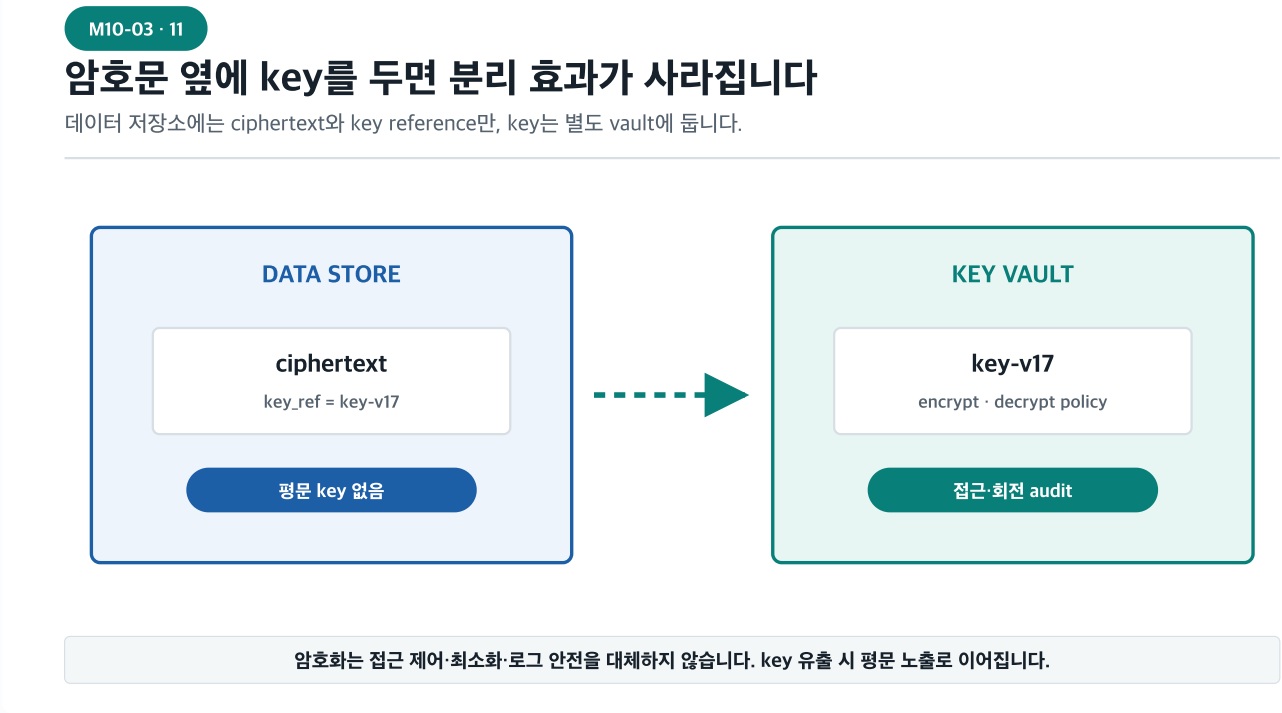
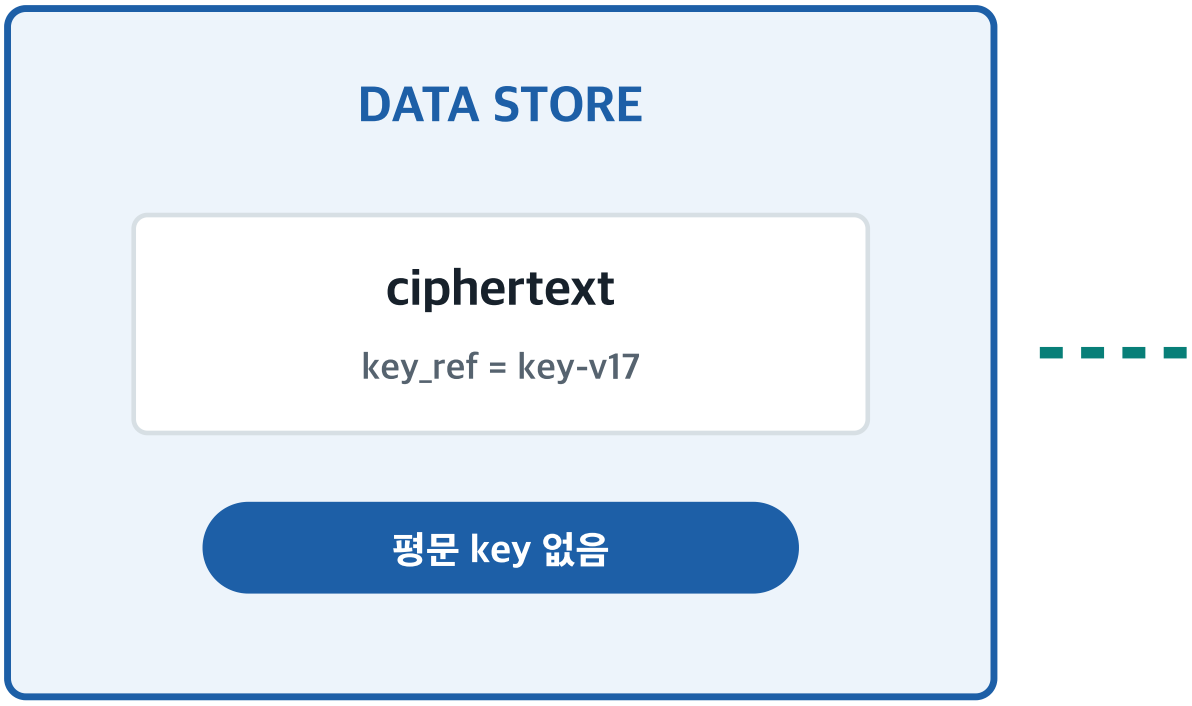
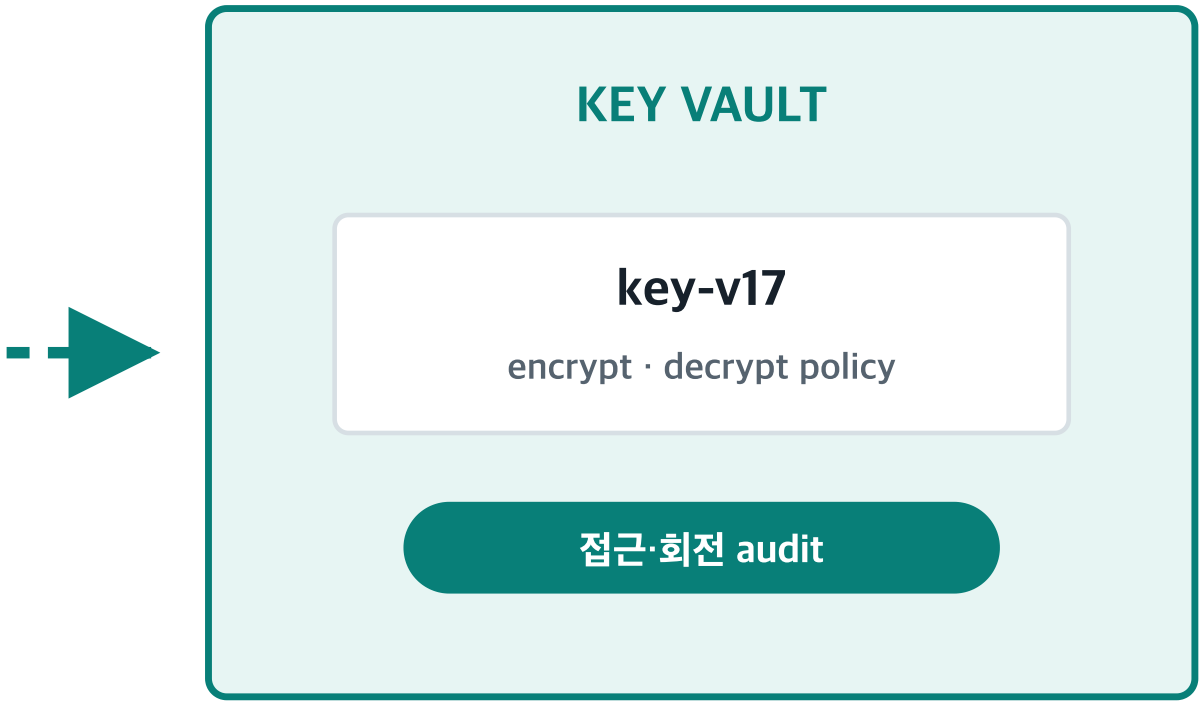


그림 11. 암호문 옆에 평문 key를 두면 한 번의 침해로 둘 다 노출됩니다. key의 저장·사용·회전·폐기를 데이터와 분리합니다.

데이터 저장소에는 ciphertext와 key reference만, key는 별도 vault에



둡니다.



12.1 위협 모델이 먼저입니다

Disk 수준 암호화는 물리 매체 유실에는 도움이 되지만 application 권한을 탈취한 원격 공격에는 충분하지 않을 수 있습니다. Application·database·filesystem·hardware 어느 층에서 보호할지는 다음을 보고 정합니다.

- 누가 무엇을 탈취하거나 볼 수 있다고 가정하는가.
- 어떤 component가 평문을 꼭 사용해야 하는가.
- key에 접근하는 workload와 사람이 누구인가.
- backup·restore·rotation 때 어떤 key가 필요한가.
- key를 잃거나 손상했을 때 복구 가능한가.

12.2 envelope 구조를 개념으로 읽습니다

```
plaintext -- DEK --> ciphertext
DEK -- KEK --> encrypted DEK
data store: ciphertext + encrypted DEK + key reference
key service: KEK + policy + audit
```

교재는 직접 암호 알고리즘을 구현하지 않습니다. 검증된 library·platform·key service와 조직의 cryptographic standard를 사용합니다.

12.3 key 관리 체크

- [ ] key ID·owner·purpose·environment를 기록했습니다.
- [ ] key와 보호할 data를 다른 경계에 두었습니다.
- [ ] application code에 key 원문이 없습니다.
- [ ] encrypt·decrypt 권한을 workload별로 줄였습니다.
- [ ] key access·rotation·failure를 audit합니다.
- [ ] rotation·retirement·destruction·recovery를 시험했습니다.
- [ ] backup 복호화에 필요한 key와 만료 정책을 연결했습니다.

13. 접근과 표시 범위를 최소화합니다

13.1 저장돼 있다는 사실이 모두에게 보일 이유가 되지 않습니다

화면·operation	필요한 field	제외할 field
문의 목록	ticket reference·status·category	email·message body
배정 상담 상세	masked email·subject·message	credential·불필요한 device ID
분석 dashboard	집계 count·category	raw identifier·free text
관리자 export	승인된 최소 column	default 전체 column

13.2 최소 권한과 목적 기반 접근

Role만 보지 않고 actor·purpose·resource relation·environment·time·operation을 함께 봅니다. 대량 조회·내보내기·복호화·secret 조회는 일반 조회보다 더 높은 영향의 별도 operation으로 둡니다.

13.3 가림은 backend 반환을 줄이는 것에서 시작합니다

CSS로 화면만 가리거나 frontend에서 문자열을 잘라도 network response에는 원문이 있을 수 있습니다. Server response serializer가 목적에 필요한 field만 반환하도록 합니다.

13.4 access evidence

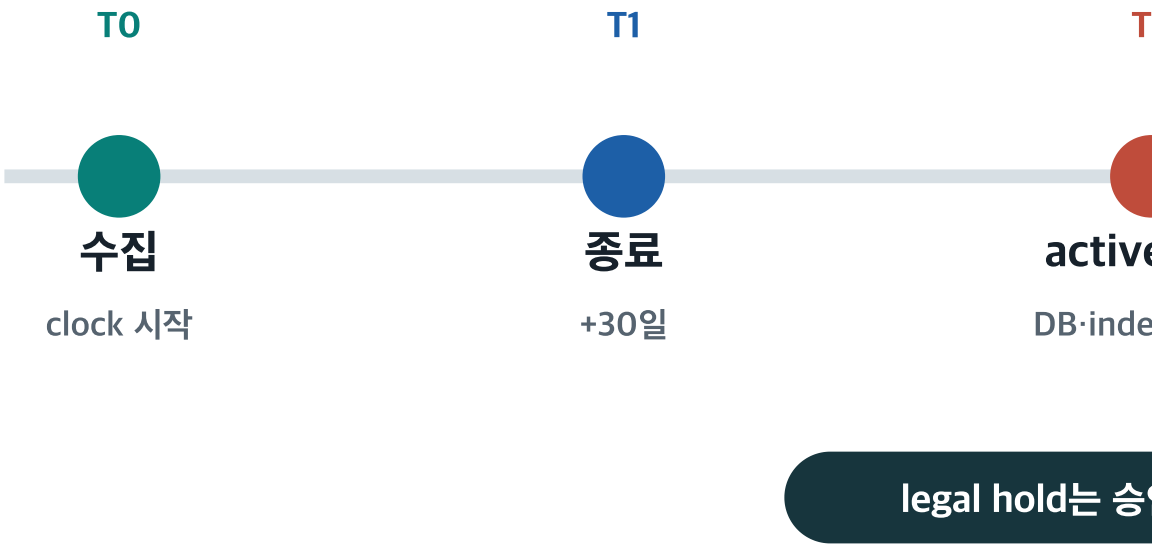
```
who: actor reference
what: operation
which: object or dataset reference
why: approved purpose
result: allow or deny
when: timestamp
evidence: audit reference
```

14. 보존·백업·파기를 하나의 시계로 연결합니다

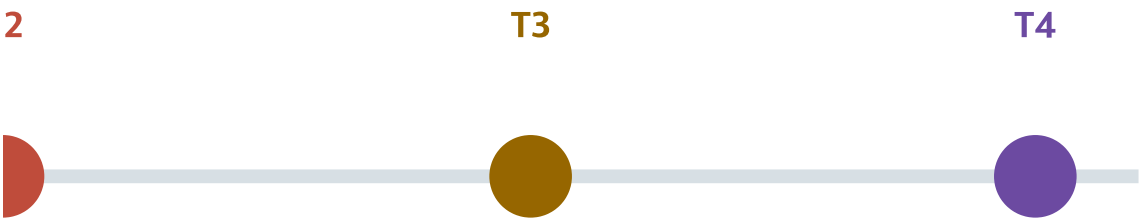
그림 12. 파기는 버튼 한 번이 아닙니다. 원본·색인·cache·export·backup의 서로 다른 시계를 연결합니다.

[illegible]

원본·색인·cache·export·backup의 만료와 복원 뒤 재삭제까지 확인합니



니다.



인된 예외로 분리

14.1 보존 rule의 최소 필드

필드	예
Data set	support message
Clock start	ticket closed
Active retention	30 days
Backup expiry	approved backup schedule
Exception	approved legal hold only
Delete method	primary·index·cache purge
Restore behavior	restore 뒤 expired item 재삭제
Owner	data-owner
Evidence	deletion event·copy count·restore test

14.2 copy inventory

- Primary database와 replica.
- Search index와 materialized view.
- Cache와 temporary file.
- Analytics warehouse와 feature store.
- User·admin export와 local download.
- Notification history와 external processor.
- Backup·snapshot·disaster recovery copy.
- AI evaluation·training·prompt archive가 있다면 그 처리 위치.

14.3 삭제와 권리 요청을 연결합니다

정보주체의 열람·정정·삭제 등 요청은 조직의 적용 법률·예외·절차에 따라 판단해야 합니다. 기술 흐름에서는 request ID가 identity verification, 대상 data set, 승인·예외, downstream processor, 처리 결과, 통지 evidence까지 이어지는지 확인합니다.

14.4 SC-19~SC-22

ID	검수할 증거
SC-19	primary·index·cache 0·deletion event
SC-20	expired backup copy false·restore re-delete true
SC-21	downstream pending 0·evidence linked
SC-22	export approval·purpose·expiry true·raw identifier false

15. 대량 export와 backup은 별도 고위험 operation입니다

15.1 화면 조회와 대량 반출을 나눕니다

한 건 조회 권한이 있다고 전체 고객 export 권한이 생기지 않습니다. 다음을 별도로 설계합니다.

- 명시적 역할과 승인.
- 목적·ticket·owner.
- column·row 최소 범위.
- 파일 보호와 전달 channel.
- 자동 만료와 수동 회수.
- 다운로드·재전송·삭제 audit.
- 이상 사용 탐지와 중단 조건.

15.2 backup도 개인정보 처리 위치입니다

Backup은 availability 통제이면서 장기 복제본입니다. 암호화, key 복구, 접근, 지역, 보존, 폐기, restore test, 복원 뒤 만료 data 재삭제를 함께 관리합니다.

15.3 legal hold는 무기한 보존의 다른 이름이 아닙니다

Hold에는 근거·범위·승인자·시작일·검토일·해제 trigger·접근 제한이 필요합니다. Hold가 끝나면 원래 retention clock과 파기 절차로 되돌립니다.

16. 노출 의심 사고를 증거로 닫습니다

그림 13. 탐지 후 우선 봉쇄하고, 영향 범위와 통지·신고 요건을 검토하며, owner와 기한이 있는 재발 방지로 닫습니다.

원인 분석을 기다리지 말고 접근 차단과 secret revoke를 우선 실행합니

signal → action → owner → time → evide

ㅣ다.

04

영향

data·주체·기간

05

검토

통지·신고 요건

06

재발 방지

owner·기한

nce → decision을 한 incident record에 연결

16.1 여섯 단계

1. **탐지:** 누가 어떤 신호를 언제 발견했는지 기록합니다.
2. **봉쇄:** 노출 경로·접근·export·session을 우선 차단합니다.
3. **회전:** 재사용 가능한 secret을 revoke하고 더 좁은 scope로 재발급합니다.
4. **영향 확인:** data category·정보주체·기간·system·recipient·copy를 확인합니다.
5. **통지·신고 검토:** 적용 법률·계약·정책에 따라 책임자와 전문가가 판단합니다.
6. **재발 방지:** 근본 원인·통제 개선·owner·기한·검증 시나리오를 연결합니다.

16.2 incident record의 최소 필드

묶음	필드
Signal	source·time·confidence·reporter
Scope	data·subject·system·period·recipient
Action	contain·revoke·preserve evidence
Decision	severity·notification review·owner
Recovery	restore·rotation·monitoring
Prevention	root cause·control·test·deadline

16.3 증거 보존과 원문 노출을 함께 다룹니다

사고 조사에 필요한 증거를 보존하되, 증거 bundle 자체에 불필요한 개인정보·secret 원문을 복제하지 않습니다.
Hash·reference·controlled snapshot·access log 등 조직의 forensic·privacy 절차를 따릅니다.

16.4 SC-23

`contained=true`, `scope_assessed=true`, `notification_reviewed=true`가 모두 증거에 연결돼야 합니다.
`notification_reviewed`는 통지가 필요 없다는 자동 판정이 아니라 책임자가 적용 요건을 검토했다는 기록입니다.

17. 검수 스튜디오에서 세 버전을 비교합니다

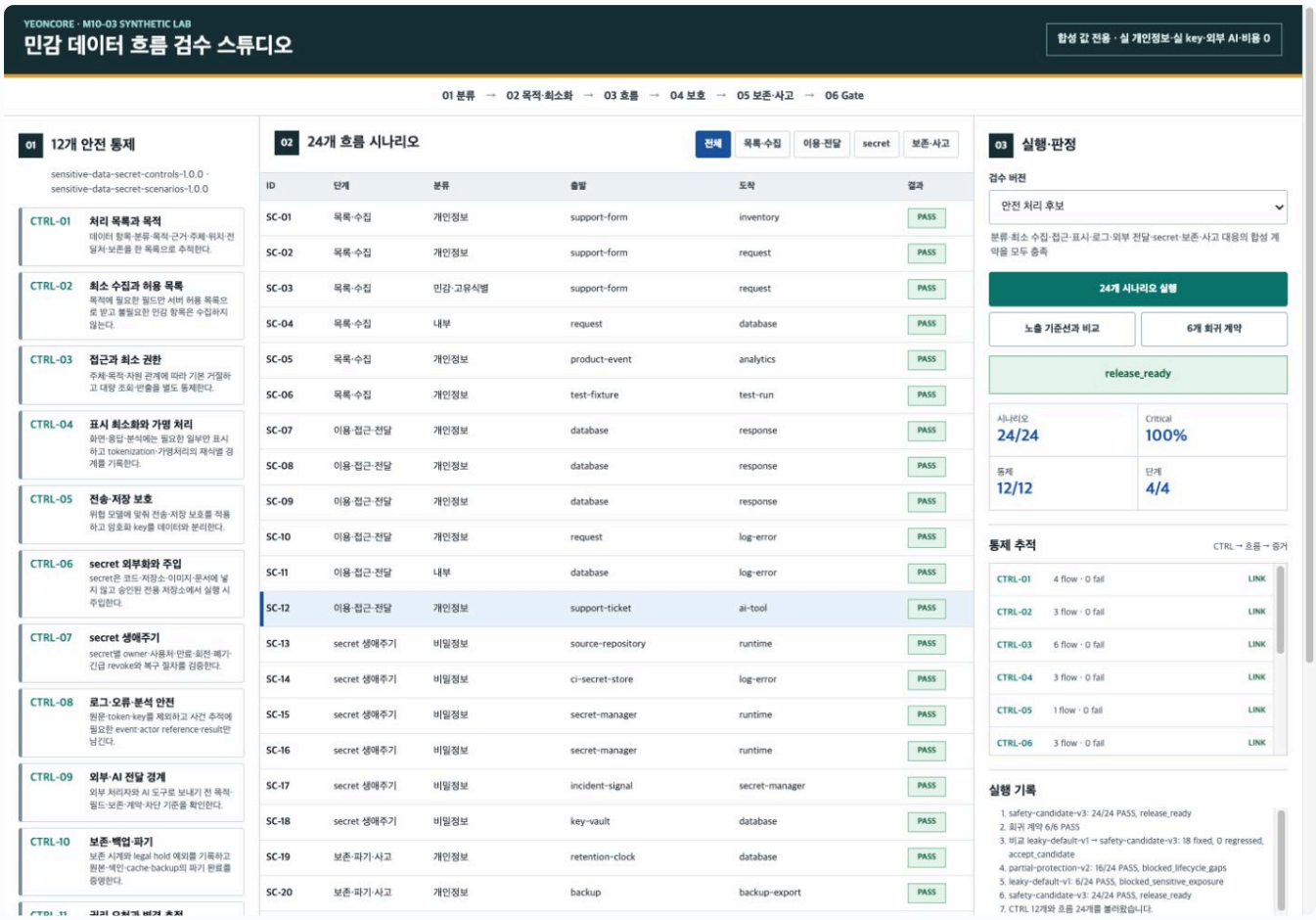


그림 14. 데스크톱 화면. 왼쪽 12개 통제, 가운데 24개 흐름, 오른쪽 Gate와 증거를 한 화면에서 연결합니다.

그림 15. 모바일 화면. 통제와 흐름은 세로로 읽고 넓은 표는 그 영역 안에서 가로 스크롤합니다.

17.1 세 버전의 학습 의미

Version	결과	남은 위험	Decision
leaky-default-v1	6/24	과다 수집·원문 로그·AI 전달·hard-code·장기 copy	blocked_sensitive_exposure
partial-protection-v2	16/24	외부·CI·회전·backup·권리·export·사고 증거	blocked_lifecycle_gaps
safety-candidate-v3	24/24	합성 범위 밖 잔여 위험 별도 기록	release_ready

17.2 고정 실패를 읽는 순서

```
scenario → data class → source → sink
→ expected field → actual field → mismatch
→ linked control → owner → retest
```

Pass rate 숫자만 보지 말고 어떤 sink와 생애주기 단계가 실패했는지 확인합니다.

17.3 비교와 회귀

```
baseline: leaky-default-v1
candidate: safety-candidate-v3
fixed: 18
regressed: 0
regression: 6/6 PASS
decision: accept_candidate
```

18. Release Gate를 증거 계약으로 운영합니다

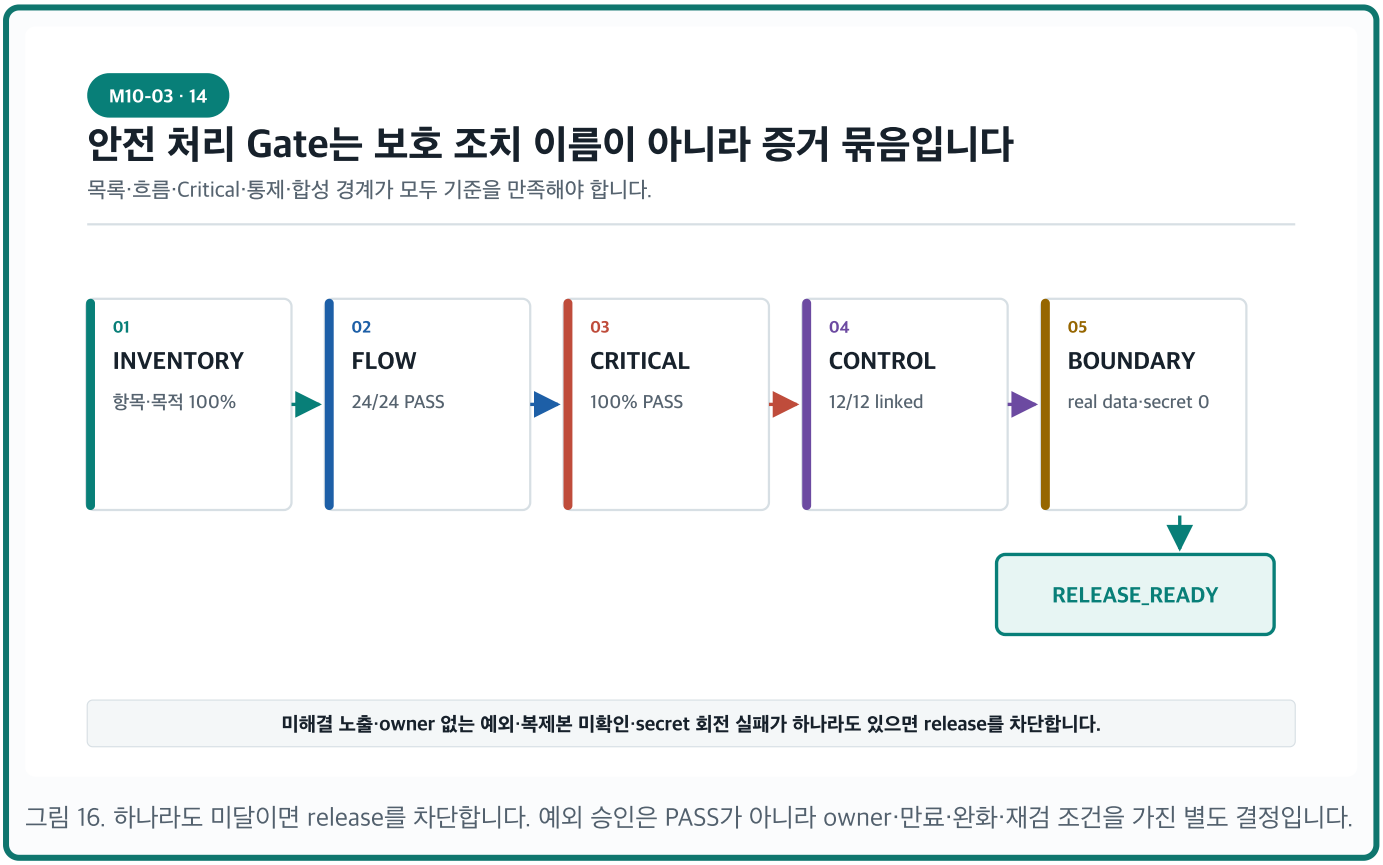


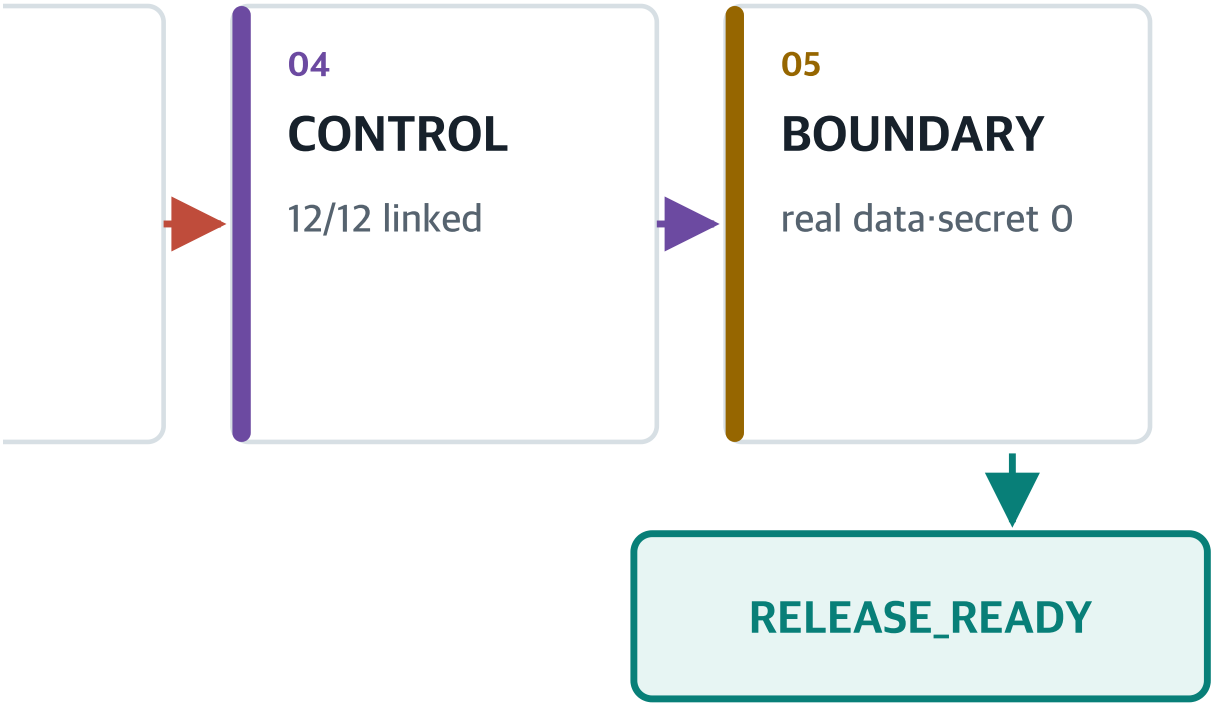
그림 16. 하나라도 미달이면 release를 차단합니다. 예외 승인은 PASS가 아니라 owner·만료·완화·재검 조건을 가진 별도 결정입니다.

목록·흐름·Critical·통제·합성 경계가 모두 기준을 만족해야 합니다.

01
INVENTORY
항목·목적 100%

02
FLOW
24/24 PASS

03
CRITICAL
100% PASS


04
CONTROL
12/12 linked
05
BOUNDARY
real data·secret 0
RELEASE_READY

18.1 실습 Gate

Gate	기준	후보
Scenario pass rate	100%	24/24
Critical pass rate	100%	100%
Stage coverage	100%	4/4
Control coverage	100%	12/12
Real personal data	0	0
Real secret	0	0
Live AI request	0	0
Orphan control·scenario	0	0

18.2 실제 결과서의 결론

```
scope + version
inventory + flow map
safeguards + scenarios
actual evidence + defects
residual risk + owner
decision + approver + expiry + retest
```

18.3 예외 승인을 숨기지 않습니다

기준 미달로 배포해야 한다면 `release_ready`로 바꾸지 않습니다. `blocked_with_exception`처럼 별도 판정을 쓰고 다음을 기록합니다.

- 실패한 Gate·시나리오·data flow.
- 개인·조직·시스템의 예상 영향.
- 임시 완화 통제와 관찰 지표.
- 승인 권한자·만료일·중단 조건.
- 수정 owner·기한·재검 시나리오.

19. 공식 기준을 실무 질문으로 바꿉니다

19.1 대한민국 개인정보 보호법·시행령

현행 법령의 개인정보 정의, 처리 원칙, 수집·이용·제공·파기, 안전조치, 정보주체 권리, 유출 대응 등을 조직의 처리 흐름에 맞게 검토합니다. 법률 조문을 교재의 일반 체크리스트로 자동 판정하지 않습니다.

공식 자료: 국가법령정보센터 개인정보 보호법 검색

19.2 개인정보의 안전성 확보조치 기준

개인정보보호위원회고시 제2026-9호는 2026년 7월 1일 시행본입니다. 내부 관리계획, 접근 권한·접근 통제, 접속기록, 보호조치, 파기 등 조직에 적용되는 요구를 현재 고시 원문과 안내서로 검토합니다.

공식 자료: 개인정보보호위원회 고시 제2026-9호

19.3 NIST Privacy Framework

Privacy Framework는 데이터 처리 생애주기와 시스템 개발 생애주기를 맞추고, 데이터 최소화·처리 관리·개인의 요청·audit·기술 통제 검증을 위험 기반으로 다루는 공통 언어를 제공합니다.

공식 자료: NIST Privacy Framework, Using Privacy Framework

19.4 NIST SP 800-122

NIST SP 800-122는 PII를 식별하고 맥락에 맞는 영향 수준과 보호조치를 정하며 부적절한 접근·이용·공개로부터 기밀성을 보호하고 사고 대응 계획을 갖추도록 안내합니다. 미국 연방기관 대상 문서이므로 한국 법률 요건을 대신하지 않습니다.

공식 자료: NIST SP 800-122

19.5 OWASP Logging Cheat Sheet

OWASP는 password·access token·민감 개인정보·database connection string·encryption key 같은 값을 보통 로그에 직접 기록하지 않고 제거·마스킹·가명처리·암호화하도록 안내합니다. 로그의 기밀성·무결성·가용성과 접근·보존도 함께 봅니다.

공식 자료: OWASP Logging Cheat Sheet

19.6 OWASP Secrets Management·Cryptographic Storage·Key Management

Secret을 중앙화·표준화하고 접근·audit·자동 회전·revoke·만료·backup·restore를 생애주기로 관리합니다. 민감정보 저장을 최소화하고 key를 data와 분리하며 검증된 platform·library를 사용합니다.

공식 자료: Secrets Management, Cryptographic Storage, Key Management

20. 현업 적용 체크리스트

20.1 설계 전

- [ ] 처리 항목·분류·목적·주체·위치·전달·보존을 목록으로 만들었습니다.
- [ ] 민감·고유식별정보는 받지 않는 대안을 먼저 검토했습니다.
- [ ] 외부·AI·분석·알림·backup sink를 모두 그렸습니다.
- [ ] 법률·계약·조직 정책의 확인일과 owner를 기록했습니다.
- [ ] 실제 개인정보·secret 대신 합성 검수 data를 준비했습니다.

20.2 구현·검수 중

- [ ] 서버 허용 목록이 숨은 field를 거절합니다.
- [ ] 목록·응답·export가 목적에 필요한 field만 반환합니다.
- [ ] 로그·오류·APM·CI 출력에 원문과 secret이 없습니다.
- [ ] 외부·AI payload의 개인정보·secret field가 0입니다.
- [ ] secret이 code·history·artifact·wiki·ticket에 없습니다.
- [ ] 환경별 secret scope·만료·회전·revoke가 검증됐습니다.
- [ ] 암호화 key와 data가 분리되고 접근이 audit됩니다.
- [ ] 대량 export에 추가 승인·목적·만료가 있습니다.

20.3 운영·파기·사고

- [ ] 원본·index·cache·export·processor·backup의 보존 시계가 연결됐습니다.
- [ ] restore 뒤 만료 data 재삭제를 검증했습니다.
- [ ] 권리 요청이 downstream 완료 evidence까지 추적됩니다.
- [ ] 노출 의심 시 봉쇄·revoke·영향 확인 runbook이 있습니다.
- [ ] 통지·신고 요건을 책임자와 전문가가 검토합니다.
- [ ] 잔여 위험·예외 승인·만료·재검 조건을 기록했습니다.

21. 30문항 셀프 테스트

21.1 문제

1. 개인정보와 secret을 같은 분류로만 다루면 안 되는 이유를 쓰십시오.
2. 처리 목록 한 행의 핵심 열 일곱 개 중 일곱 개를 쓰십시오.
3. 값의 모양만으로 데이터 분류가 끝나지 않는 이유는 무엇입니까.
4. 목적 문장을 검수 가능하게 쓰는 방법을 예로 설명하십시오.
5. 최소 수집을 판단하는 네 질문 중 두 가지를 쓰십시오.
6. UI에서 field를 숨기는 것만으로 과다 수집을 막을 수 없는 이유는 무엇입니까.
7. 자유 입력에 민감정보 안내만 두면 부족한 이유를 쓰십시오.
8. 마스킹과 암호화의 차이를 설명하십시오.
9. Tokenization의 핵심 잔여 위험은 무엇입니까.
10. 단순 hash가 항상 익명화를 보장하지 않는 이유는 무엇입니까.
11. 요청 원문이 퍼질 수 있는 sink 여덟 개 중 네 개를 쓰십시오.
12. 좋은 사건 추적 로그의 field 네 가지를 쓰십시오.
13. 로그에서 보통 직접 남기지 않을 값 네 가지를 쓰십시오.
14. 일반화 오류 응답이 숨겨야 할 정보 세 가지를 쓰십시오.
15. 외부·AI 전달 전에 확인할 질문 세 가지를 쓰십시오.
16. M10-03과 M10-04의 AI 검수 경계를 설명하십시오.
17. Secret이 source repository에 commit됐다면 파일 삭제 외에 무엇을 해야 합니까.
18. 환경변수가 완전한 secret vault가 아닌 이유를 쓰십시오.
19. Secret 생애주기 여섯 단계를 순서대로 쓰십시오.
20. 노출 의심 secret을 먼저 revoke해야 하는 이유는 무엇입니까.
21. Secret 회전의 성공 증거 세 가지를 쓰십시오.
22. 암호화 key와 ciphertext를 분리하는 이유를 설명하십시오.
23. Disk 암호화만으로 application compromise를 막지 못할 수 있는 이유는 무엇입니까.
24. 목록 API가 상세 API보다 더 적은 field를 반환해야 하는 이유는 무엇입니까.
25. 삭제 요청 접수와 파기 완료의 차이를 쓰십시오.
26. Backup restore 뒤 재삭제 검증이 필요한 이유는 무엇입니까.
27. 대량 export를 일반 조회와 별도 operation으로 두는 이유는 무엇입니까.
28. 노출 사고 대응의 여섯 단계를 쓰십시오.
29. `notification_reviewed=true`가 자동 법률 판정을 뜻하지 않는 이유는 무엇입니까.

30. 자신의 서비스 한 기능을 골라 source·field subset·purpose·sink·protection·retention을 한 줄로 쓰십시오.

21.2 모범 답안

1. 개인정보는 개인의 권리와 처리 목적·보존을, secret은 시스템 권한과 회전·revoke를 중심으로 통제하며 두 속성이 겹칠 수도 있기 때문입니다.
2. Field·분류·목적·근거·정보주체·actor·source·store·sink·보호·보존·owner 중 일곱 가지입니다.
3. 다른 데이터와의 결합, 처리 목적, 접근 주체, 노출 영향에 따라 같은 값의 위험이 달라지기 때문입니다.
4. “서비스 개선” 대신 “주간 오류율을 개인 식별자 없이 집계”처럼 field 필요성을 판정할 수 있게 씁니다.
5. 없으면 목표를 달성할 수 있는지, 더 낮은 민감도로 대체할 수 있는지, 나중에 받을 수 있는지, 원문 대신 reference·집계를 쓸 수 있는지 중 두 가지입니다.
6. 클라이언트가 숨은 field를 직접 보낼 수 있으므로 서버 허용 목록이 저장 전에 거절해야 합니다.
7. 사용자는 여전히 고위험 값을 입력할 수 있으므로 탐지·차단·수정 경로·원문 없는 event가 필요합니다.
8. 마스킹은 표시 일부를 숨기지만 원본은 남고, 암호화는 key 없이는 원문 해석을 어렵게 하는 가역 보호입니다.
9. Token과 원본의 mapping store가 노출되거나 과도한 권한으로 재연결되는 위험입니다.
10. 후보 공간이 작거나 예측 가능하면 원문 후보를 hash해 대입할 수 있기 때문입니다.
11. Request·database·response·log/error·analytics·notification·AI/tool·backup/export 중 네 가지입니다.
12. Event·actor reference·object reference·result·reason code·correlation reference·time 중 네 가지입니다.
13. Password·session ID·access token·API key·connection string·encryption key·민감 원문 중 네 가지입니다.
14. Stack trace·file path·query value·internal class·secret 중 세 가지입니다.
15. 필요성·정확한 field·비식별 대안·secret 혼입·보존·하위 처리자·학습 사용·변경 trigger 중 세 가지입니다.
16. M10-03은 보내기 전 개인정보·secret 0을, M10-04는 prompt injection·환각·tool misuse·output 유출을 검수합니다.
17. 노출된 것으로 보고 revoke·재발급·영향 확인·history와 artifact 처리·재발 방지를 해야 합니다.
18. Process dump·debug·하위 process·runtime 정보 표면에 노출될 수 있기 때문입니다.
19. 생성·저장·주입·사용·회전·폐기입니다.
20. 원인 분석 중에도 재사용될 수 있어 blast radius가 계속 커질 수 있기 때문입니다.
21. 새 version active, consumer 건강 확인, 이전 version invalid, downtime 없음, audit event 중 세 가지입니다.
22. 한 저장소의 침해로 암호문과 복호화 key를 동시에 얻지 못하게 하려는 것입니다.
23. Application이 정상 권한으로 평문을 읽을 수 있으면 탈취된 application 권한도 같은 경로를 사용할 수 있기 때문입니다.
24. 반복·대량 조회되는 목록은 불필요한 원문 노출과 blast radius가 더 크기 때문입니다.
25. 접수는 작업 시작이고 완료는 원본·색인·cache·processor·backup 적용 대상과 증거가 정책 기준을 충족한 상태입니다.
26. 만료된 data가 backup 복원으로 다시 active store에 나타날 수 있기 때문입니다.
27. 대량 복제·반출의 영향이 커 추가 승인·최소 column·만료·audit가 필요하기 때문입니다.
28. 탐지·봉쇄·회전·영향 확인·통지 검토·재발 방지입니다.

29. 적용 법률·사실관계·계약을 책임자와 전문가가 검토했다는 evidence일 뿐 교재 엔진의 자동 법률 결론이 아니기 때문입니다.
30. 답은 서비스에 따라 다르며 여섯 요소와 owner·evidence까지 연결되면 좋습니다.

22. 마지막 한 장 요약

```
1. 공개·내부·개인정보·민감·secret을 분류한다.
2. 항목·목적·주체·위치·전달·보존을 처리 목록에 연결한다.
3. 목적에 필요하지 않은 field는 받지 않는다.
4. 요청·응답·로그·분석·알림·AI·backup의 흐름을 그린다.
5. 마스킹·token·가명처리·암호화의 효과를 구분한다.
6. Secret을 code 밖에서 주입하고 회전·revoke·폐기한다.
7. 암호화 key와 data를 분리하고 접근·회전을 audit한다.
8. 원본·index·cache·export·processor·backup의 파기를 증명한다.
9. 사고는 봉쇄·회전·영향·검토·재발 방지 evidence로 닫는다.
10. 24개 흐름·12개 통제·잔여 위험을 Release Gate로 판정한다.
```

완료 기준: 자신의 서비스 한 기능에 대해 처리 목록, 흐름도, 최소 수집, 접근·표시·로그·외부 전달, secret·key, 보존·파기, 사고 대응, 잔여 위험, 릴리스 결론을 하나의 안전 처리 결과서로 설명할 수 있으면 이 매뉴얼을 마친 것입니다.