M13-04 · GIBALJA VISUAL MANUAL

인수인계와 유지보수 준비하기

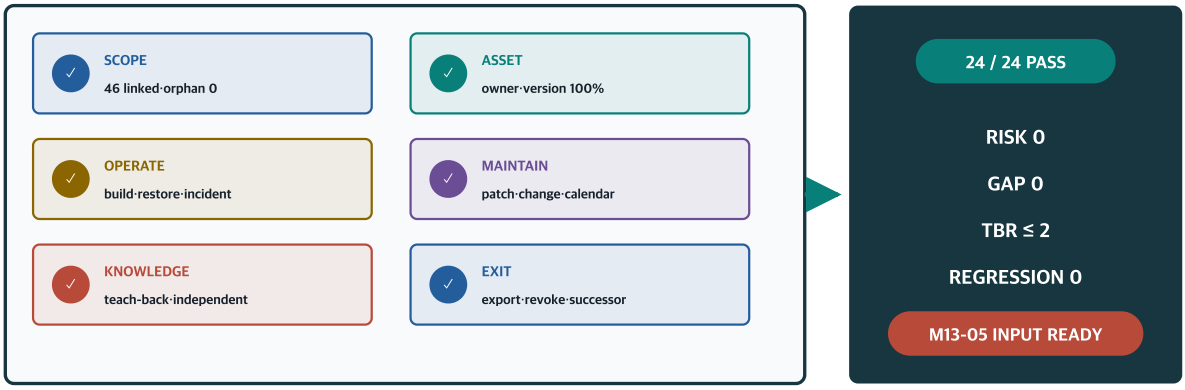
한 문장 목표: `M12·M13 evidence 46개 → scope·ownership·asset manifest → build·access·data·integration·operations rehearsal → maintenance backlog·calendar → teach-back·reverse shadow → Handover Acceptance Gate → M13-05 고객 가치·사업 언어` 를 한 trace로 연결합니다.

학습·권한 경계: 이 장의 조직·사람·계정·데이터·장애·계약·수치는 모두 합성입니다. `handover_acceptance_package_ready` 는 실제 서비스 이전·credential/secret 전달·유지보수 계약·검수·법률/규제 적합·production acceptance·배포·release·가용성 보장이 아닙니다. 실제 판단과 전달은 조직 policy와 자격·권한 있는 서비스·보안·개인정보·법무·계약·운영 담당자가 수행해야 합니다.

M13-04 · 15

Handover Acceptance Gate는 받은 것보다 독립 수행과 남은 위험을 봅니다

24개 scenario와 12개 control을 통과하고 critical asset·risk·gap·TBR·exit·후속 입력을 확인합니다.

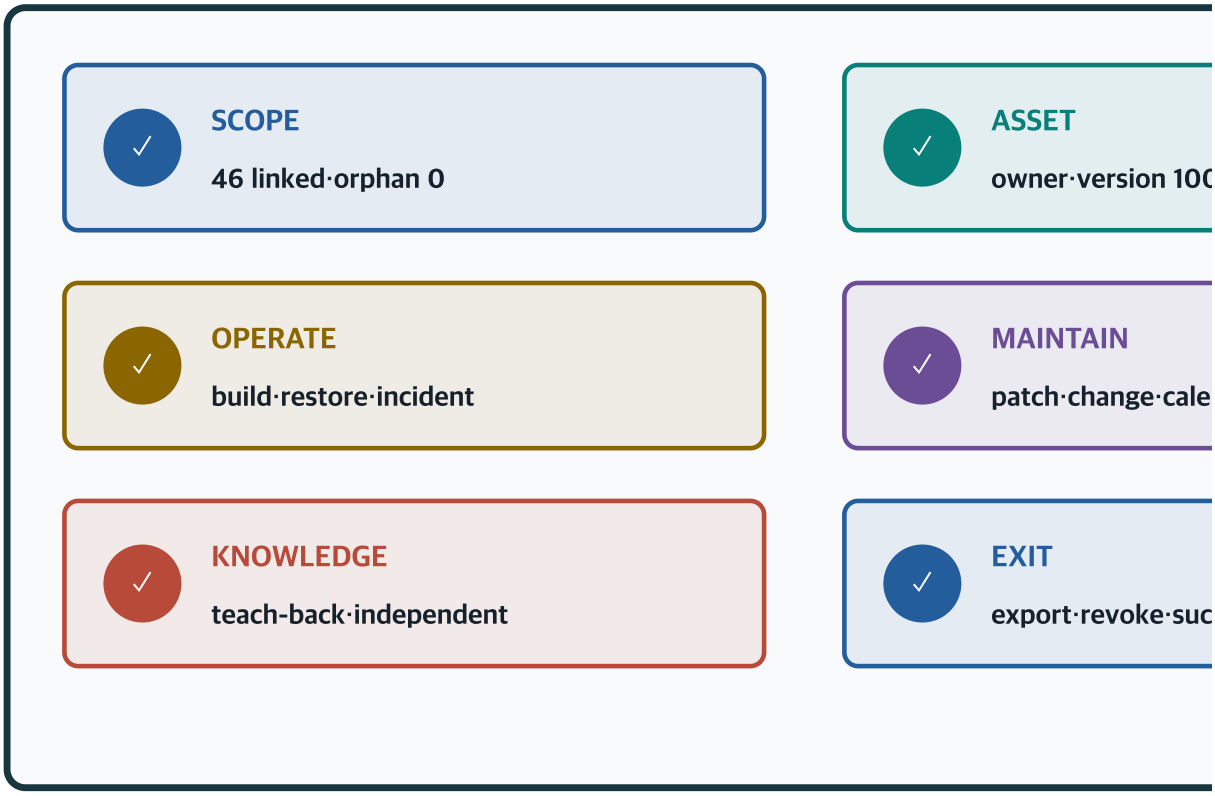


교육 Gate 통과 뒤에도 실제 검수·서비스 이전·계약·credential 전달·production release는 별도 권한으로 결정합니다.

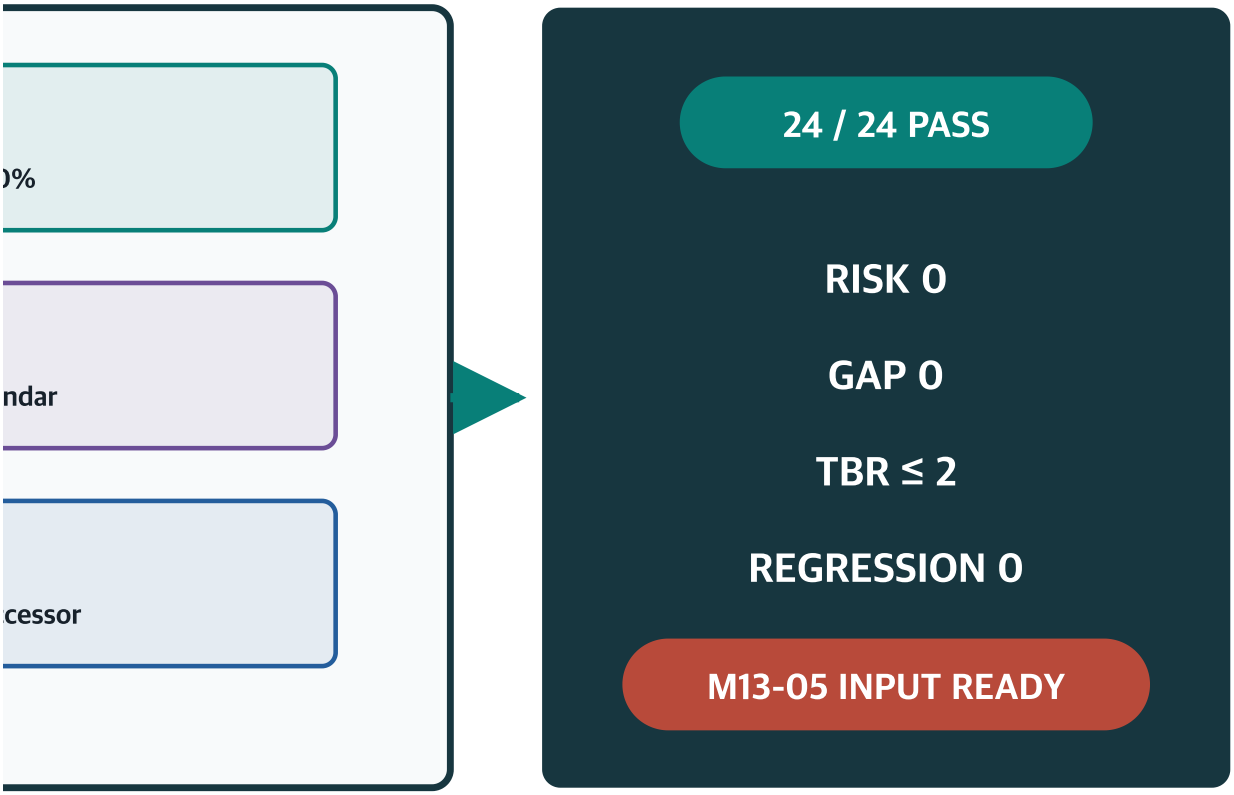
한눈에 보기. 받은 파일 수보다 독립 build·restore·incident·maintenance·teach-back·exit evidence와 남은 risk·gap·TBR을 봅니다.

Handover Acceptance Gate는 인증 체크

24개 scenario와 12개 control을 통과하고 critical asset·risk·gap·TBR·exit·후



속 입력을 확인합니다.


0%
ndar
cessor
24 / 24 PASS
RISK 0
GAP 0
TBR ≤ 2
REGRESSION 0
M13-05 INPUT READY

1. 이 장에서 완성할 것

산출물	핵심 내용	합성 완료 신호
Handover input	M12·M13 source·scope·authority	46 linked·orphan0·conflict0
Asset/ownership	RACI·repo·artifact·config·access	critical owner/version 100%
Operability	build·deploy·restore·incident	receiver independent pass
Maintenance	4 types·patch·change·calendar	owner·test·rollback
Knowledge/exit	teach-back·reverse shadow·export·revoke	hidden critical unknown0
Gate	24 scenario·12 control·risk/gap/TBR	24/24·0/0/2·M13-05

2. 그림부터 읽는 5분 지도

그림	먼저 볼 질문	설명할 수 있어야 할 것
01~03	무엇을 어떻게 넘기나	evidence journey·authority·mode
04~06	누가 무엇을 같은 상태로 만들 수 있나	RACI·manifest·build/rollback
07~09	접근·데이터·연계가 끊기지 않나	credential lifecycle·restore·dependency
10~12	장애와 변경을 누가 처리하나	observability·maintenance types·patch cycle
13~15	지식·반복 업무·종료가 이어지나	teach-back·calendar·Gate·M13-05

3. 합성 사례 카드

항목	합성 값	경계
Service	회의 후 실행 기록 서비스의 인수인계와 유지보수 준비	실제 조직·서비스 아님

항목	합성 값	경계
Source	M12 + M13-01~03 = 46 linked	실제 source·계약 자료 아님
Mode	HND-C evidence-led candidate	실제 transfer 판정 아님
Receiver	8명 signal·3회 session	실제 인원·교육 아님
Service signals	p95 800ms·RTO4h·RPO1h·response30m·critical patch7d	SLA·보장 아님
Portfolio	12 controls·24 scenarios·12 evidence	compliance·certification 아님
Target	M13-05 customer value/business language input	사업 승인 아님

4. 안전 경계와 금지선

이 실습이 하는 것	하지 않는 것	실제 상황의 다음 행동
합성 metadata와 fault rehearsal	실제 개인정보·계정·secret 사용	승인된 sandbox·vault·privacy procedure 사용
교육용 readiness Gate	실제 검수·서비스 이전	계약·acceptance authority 승인
합성 patch·restore·incident	production change·release	change/release policy와 운영 window 적용
공식 source를 applicability reference로 사용	자동 준수·인증·법률 의견	최신판·범위·전문가 검토
M13-05 입력 준비	사업성·예산 승인	고객·재무·경영 authority 판단

5. 인수인계서의 12개 evidence 구조

묶음	포함 evidence	수신 팀이 답할 질문
Scope/authority	EVD-01~02	무엇을 누구 권한으로 받는가
Asset/build/access	EVD-03~05	같은 artifact를 만들고 안전하게 접근하는가
Data/dependency	EVD-06~07	복구·대조·실패·만료를 처리하는가
Operations/support	EVD-08~09	장애를 탐지·소통·복구·escalate하는가

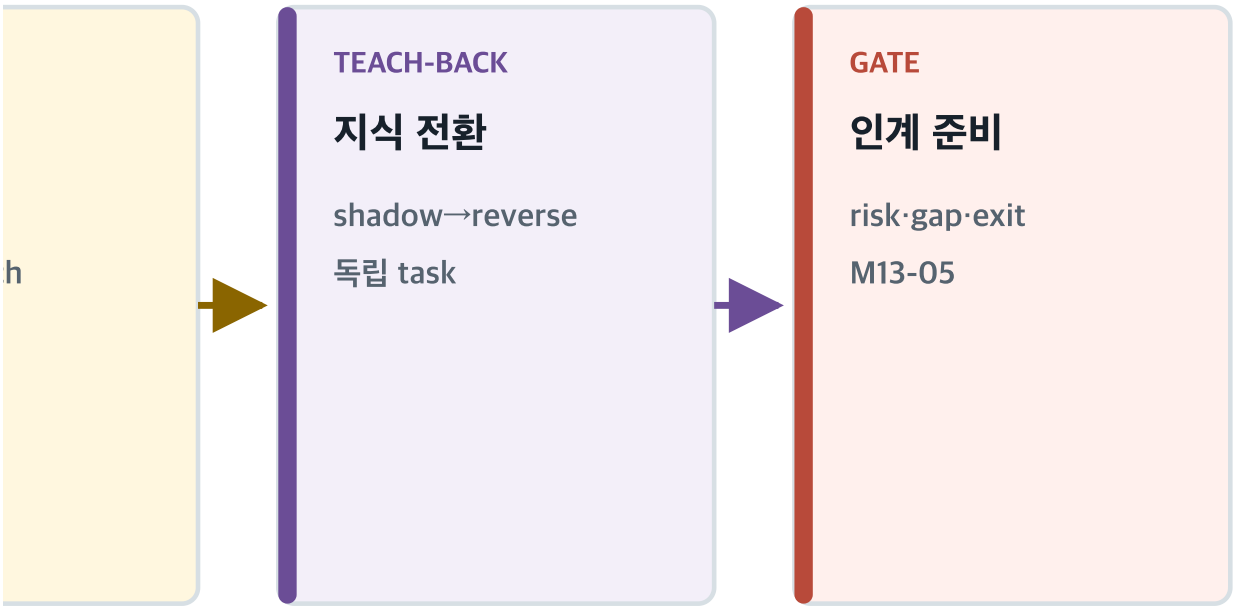
묶음	포함 evidence	수신 팀이 답할 질문
Maintenance/knowledge	EVD-10~11	변경을 관리하고 독립 수행하는가
Gate/exit	EVD-12	남은 risk·gap·TBR·종료·후속이 보이는가

6. 파일 전달을 운영 능력의 이동으로 바꾸기

그림 1. Input→manifest→rehearsal→teach-back→Gate를 같은 ID·owner·version으로 잇습니다.

받은 자료를 manifest로 정규화하고 수신 팀의 독립 재현 증거로 바꿉니다.

목록 → 재현 → 설명 →


h
TEACH-BACK
지식 전환
shadow→reverse
독립 task
GATE
인계 준비
risk·gap·exit
M13-05

독립 수행 → 잔여 위험

> **핵심 원칙:** 인계 완료는 '보냈다'가 아니라 수신 팀이 설명·실행·복구할 수 있다는 evidence입니다.

좋은 문서도 실제 build·restore·incident·patch 절차와 연결되지 않으면 특정 사람의 기억을 대신하지 못합니다. 합성 사례는 이전 장의 46개 artifact를 12개 handover evidence와 24개 rehearsal로 바꿉니다.

관점	합성 사례	확인할 evidence
Input	46 linked artifacts	version·orphan0·conflict0
Manifest	scope·asset·authority	owner·freshness·TBR
Rehearsal	build·restore·incident	receiver executed
Knowledge	teach-back·reverse shadow	independent task
Gate	risk·gap·exit·M13-05	24/24

> **흔한 오답:** 공유 폴더에 source와 매뉴얼을 올린 뒤 인수인계 완료 메일을 보냅니다.

> **고친 예:** 항목마다 owner·version·location·rehearsal·acceptance·TBR을 붙이고 수신 팀의 독립 수행 결과를 기록합니다.

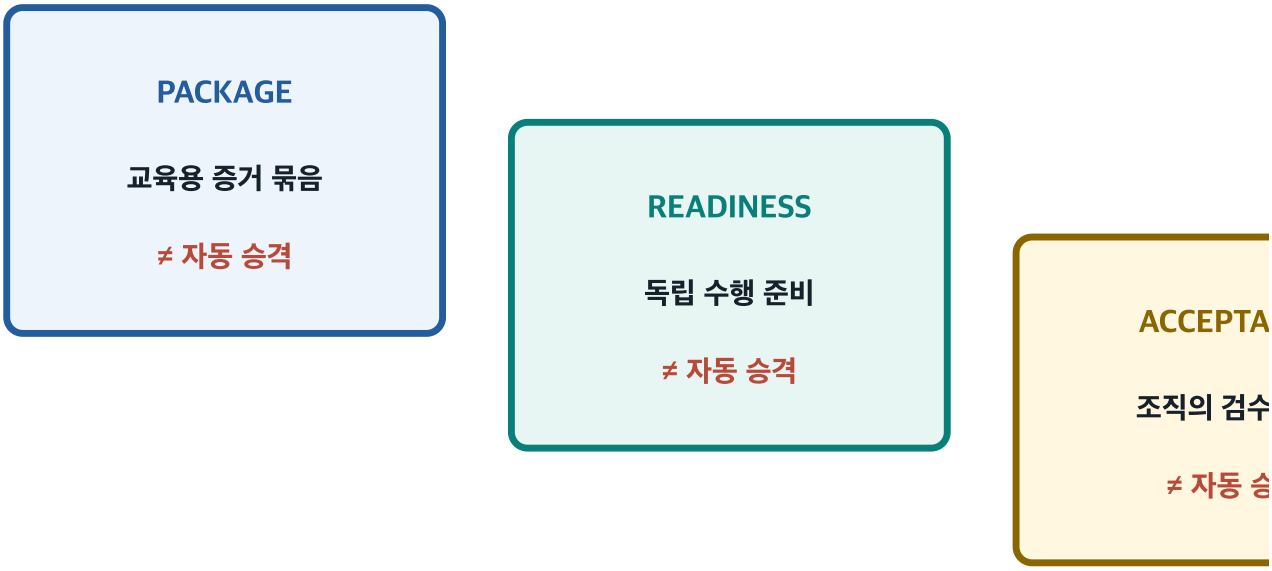
그림을 보며 4단계 연습

1. 46개 입력의 version을 확인합니다.
2. 인계 항목을 manifest로 정규화합니다.
3. 수신 팀이 P0 task를 독립 수행합니다.
4. 잔여 risk·gap·TBR과 후속 owner를 남깁니다.

7. 증거 묶음과 실제 이전·계약·release 권한 분리하기

그림 2. 교육 package·독립 수행 준비·검수·실제 이전·계약/release는 서로 다른 결정입니다.

각 층의 결정권자를 나눠 교육 Gate를 계약 승인이나 production release로 오



해하지 않습니다.


NCE
판단
격
TRANSFER
계정·자산·책임 이전
≠ 자동 승격
CONTRACT·RELEASE
계약·production 권한
≠ 자동 승격

핵심 원칙: Evidence는 결정 입력이지 계정·자산·책임·계약을 자동 이전하는 권한이 아닙니다.

준비도가 높아도 실제 credential 전달·계약 검수·production 운영은 조직 정책·계약·보안 절차의 권한자가 승인해야 합니다. 문서의 answered question과 unanswered authority를 첫 페이지에 함께 씁니다.

관점	합성 사례	확인할 evidence
Package	교육 evidence	author·reviewer
Readiness	독립 수행 가능성	receiving owner
Acceptance	검수 기준 충족 판단	acceptance authority
Transfer	계정·자산·책임 이전	security·service authority
Contract/release	법적·production 결정	qualified authority

흔한 오답: Gate가 PASS이므로 실제 서비스와 credential이 자동 인수됐다고 선언합니다.

고친 예: 교육 Gate·조직 검수·실제 transfer·maintenance contract·production release를 별도 상태와 authority로 기록합니다.

그림을 보며 4단계 연습

1. 현재 문서가 답하는 질문을 씁니다.
2. 답하지 않는 실제 결정을 나열합니다.
3. 결정별 authority와 source를 연결합니다.
4. 오해 가능한 완료 문구를 경계 문장으로 고칩니다.

8. 네 가지 인수인계 방식의 사람 종속 비교하기

그림 3. 파일 존재·관찰·독립 수행·공급자 의존을 서로 다른 handover mode로 구분합니다.

[illegible]

문서 덤프·동행 중심·증거 중심·공급자 의존 모델을 독립 운영 가능성으로 비교[illegible]

HND-A

문서 덤프

파일 존재

재현 증거 없음

BLOCKED

6/24

HND-B

Shadow 의존

시연 관찰

혼자 수행 안 함

PARTIAL

15/24

관찰한 것과 독립적으로 수[illegible]

합니다.

HND-C

Evidence-led

teach-back

reverse shadow

CANDIDATE

24/24

HND-D

공급자 Black-box

초기 부담 작음

진단·exit 잠김

TRIGGER

조건부

행할 수 있는 것은 다릅니다

핵심 원칙: 관찰한 것과 혼자 수행할 수 있는 것은 다릅니다.

문서 덤프는 빠르지만 재현 증거가 없고 shadow-only는 현장 맥락을 보지만 기존 담당자 종속을 남깁니다. Evidence-led 방식은 수신 팀의 행동을 확인하고 supplier black-box는 diagnosis·exit trigger를 명시합니다.

관점	합성 사례	확인할 evidence
HND-A	문서 덤프	6/24·blocked
HND-B	shadow 의존	15/24·partial
HND-C	evidence-led teach-back	24/24·candidate
HND-D	supplier black-box	exit trigger 필요
선택	HND-C	실제 승인 아님

흔한 오답: 기존 담당자의 시연이 매끄러웠으므로 새 팀도 운영할 수 있다고 간주합니다.

고친 예: Shadow 뒤 reverse shadow와 independent build·restore·incident task를 실행해 사람 종속을 측정합니다.

그림을 보며 4단계 연습

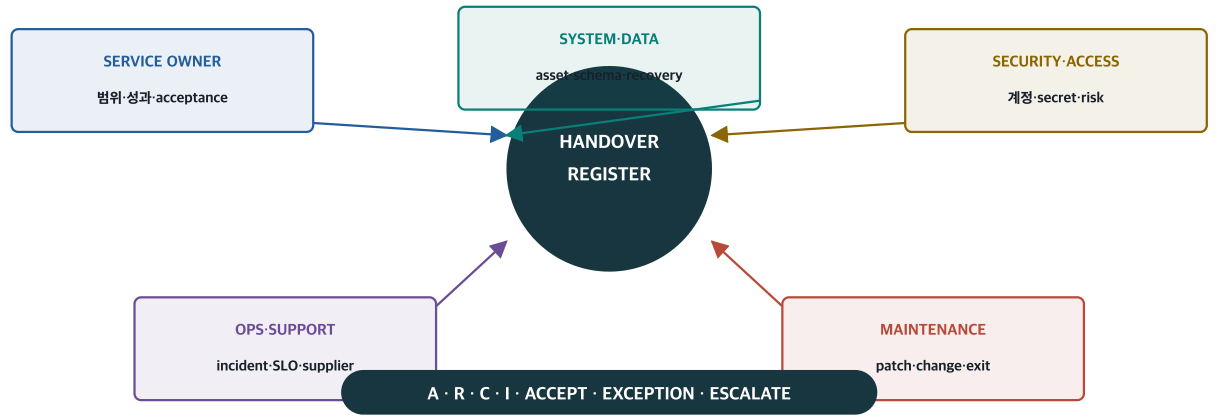
1. 현재 방식이 네 mode 중 어디인지 고릅니다.
2. 사람·공급자 종속을 적습니다.
3. 독립 수행으로 바꿀 P0 task를 고릅니다.
4. 실패 시 보완·중단 trigger를 정합니다.

9. 소유권·RACI·수락·예외 권한을 한 지도에 놓기

M13-04 · 04

모든 인계 항목에는 소유자와 수락·예외 권한이 필요합니다

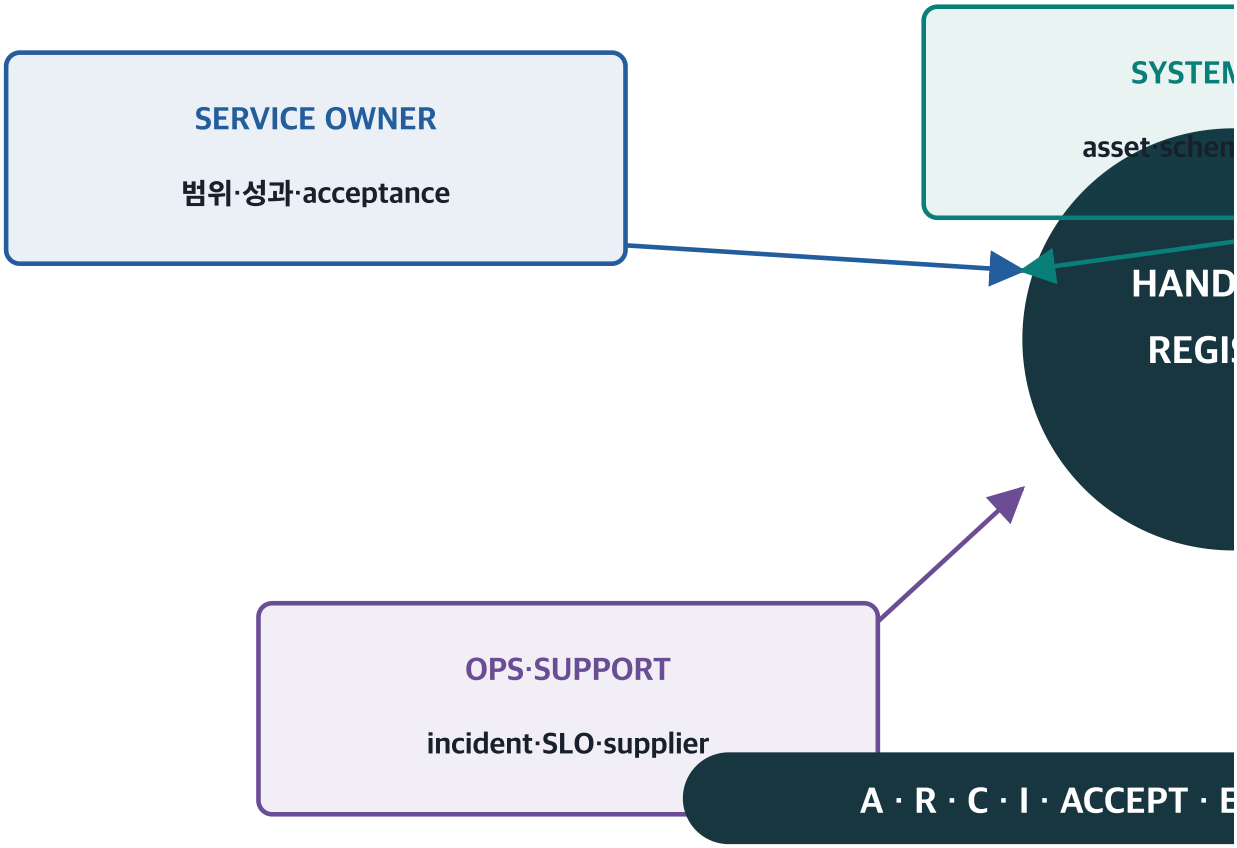
서비스·시스템·데이터·보안·운영·지원·유지보수 역할을 한 등록부에 연결합니다.

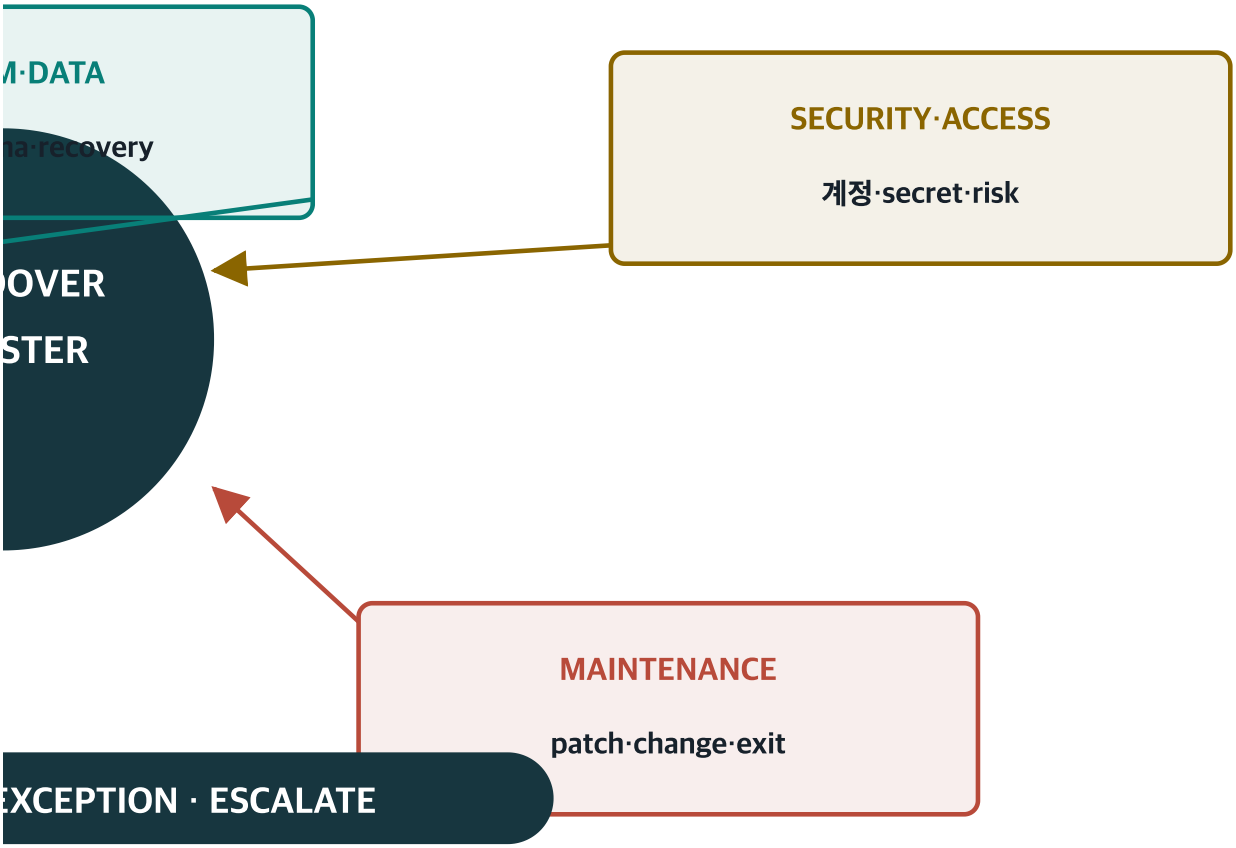


검토자가 많아도 accountable owner가 없으면 문제가 발생한 뒤 책임과 결정 경로가 다시 끊깁니다.

그림 4. 인계 항목마다 accountable owner와 receiving responsibility·acceptance·exception 권한을 연결합니다.

서비스·시스템·데이터·보안·운영·지원·유지보수 역할을 한 등록부에 연결합니다




·DATA
a·recovery
OVER
STER
SECURITY·ACCESS
계정·secret·risk
MAINTENANCE
patch·change·exit
XCEPTION · ESCALATE

핵심 원칙: 모두가 검토해도 최종 설명 책임과 예외 결정은 명확해야 합니다.

Service·system·data·security·operations·support·supplier 역할이 섞이면 장애와 변경 때 결정이 멈춥니다. 항목마다 A는 한 명으로 두고 실제 권한과 문서 책임의 conflict를 0으로 닫습니다.

관점	합성 사례	확인할 evidence
Service	scope·outcome	acceptance
System/data	asset·schema·restore	technical owner
Security/access	credential·risk	exception authority
Ops/support	incident·SLO·supplier	receiving owner
Maintenance	patch·change·exit	calendar·successor

흔한 오답: 운영팀·개발팀·보안팀을 모두 공동 책임으로 적고 예외 승인자는 비웁니다.

고친 예: 항목별 RACI·decision right·escalation·acceptance·exception authority를 별도 열로 둡니다.

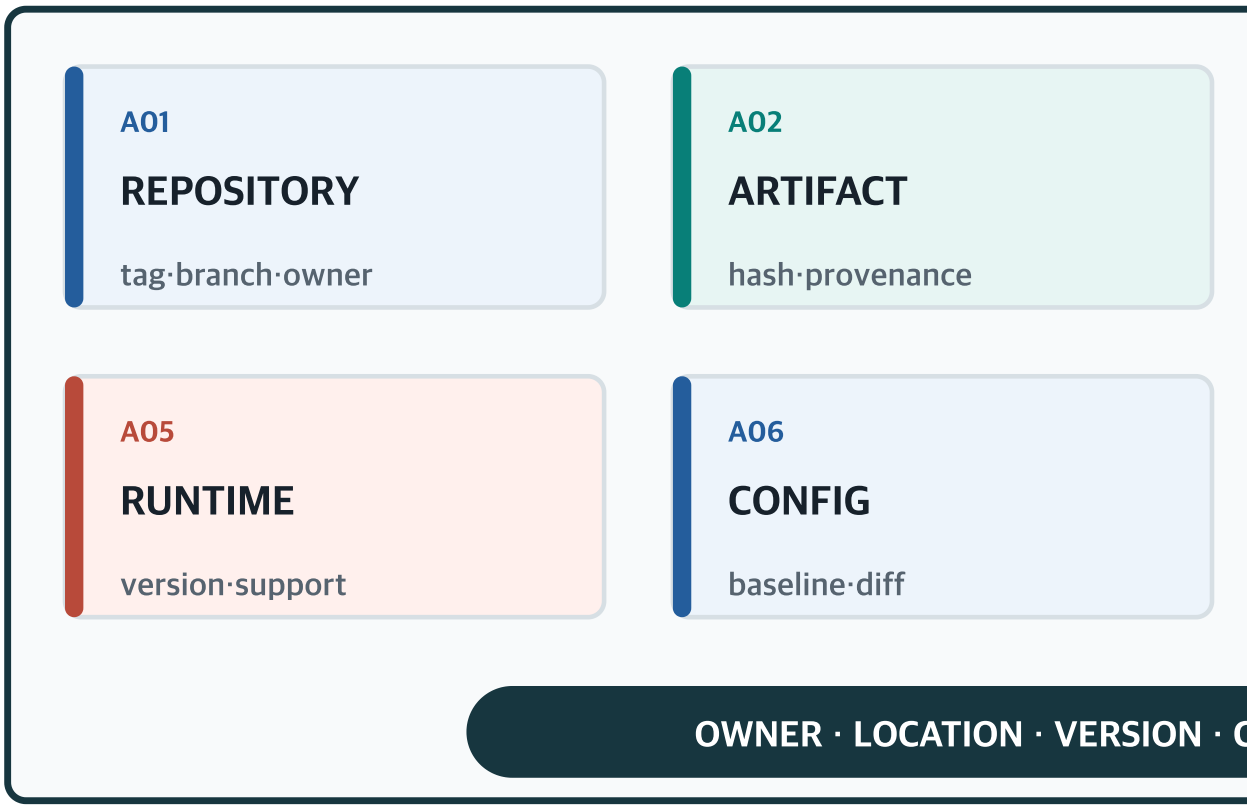
그림을 보며 4단계 연습

1. 인계 항목과 stakeholder를 나열합니다.
2. A·R·C·I를 배정합니다.
3. 수락·예외·risk authority를 확인합니다.
4. authority conflict를 rehearsal합니다.

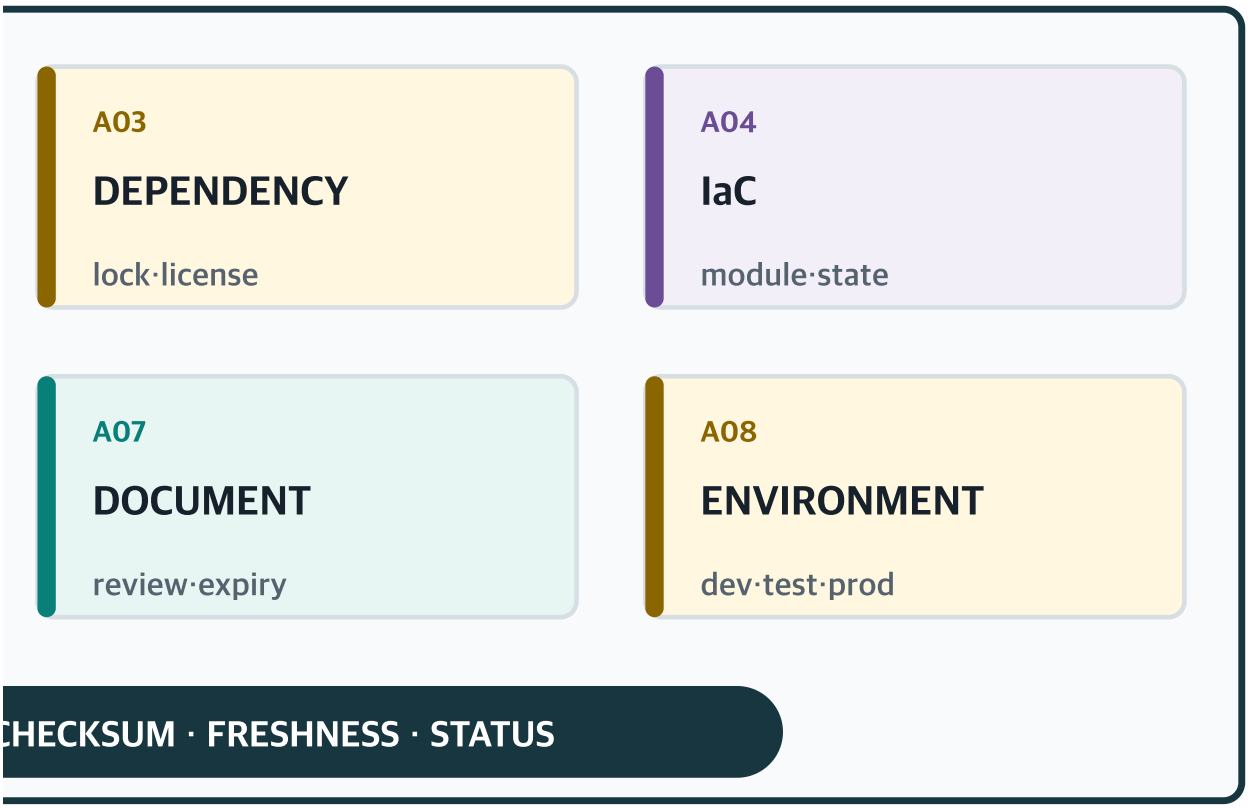
10. 자산·구성·버전을 재현 가능한 manifest로 만들기

그림 5. 자산마다 owner·location·version·checksum·freshness·criticality를 고정합니다.

Repository부터 문서·환경까지 owner·version·checksum·freshness를 한 ma



nifest로 묶습니다.



핵심 원칙: 자산 목록은 위치 표가 아니라 변경·복구·종료의 기준선입니다.

Repository와 artifact만 받아도 dependency lock·IaC state·runtime·환경 변수·문서·license가 빠지면 동일 환경을 재현할 수 없습니다. Critical asset의 owner와 version이 없는 항목은 인계 완료가 아닙니다.

관점	합성 사례	확인할 evidence
Source	repo·branch·tag	owner·protection
Artifact	binary·container	hash·provenance
Platform	dependency·IaC·runtime	lock·version·support
Configuration	baseline·environment	diff·secret class
Knowledge/license	document·license	review·expiry·successor

흔한 오답: 폴더 경로와 시스템 이름만 적고 실제 version·checksum·owner를 생략합니다.

고친 예: Criticality·owner·location·version·checksum·freshness·status·successor를 한 행으로 관리합니다.

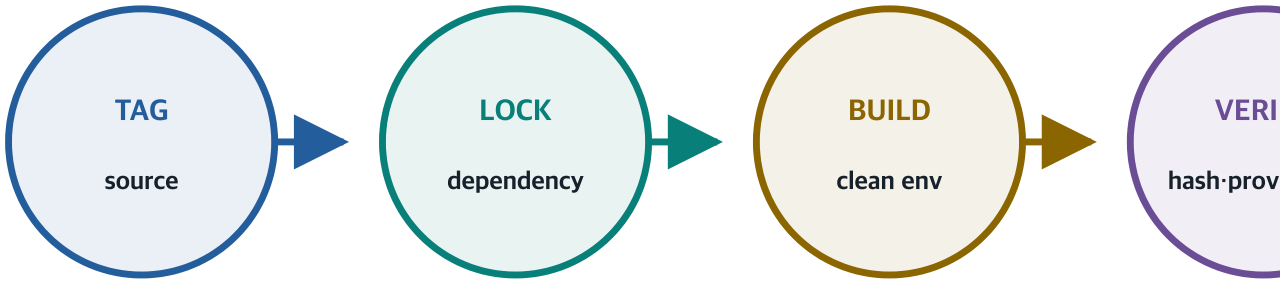
그림을 보며 4단계 연습

1. 8개 asset domain을 inventory합니다.
2. Critical asset을 표시합니다.
3. Version·checksum·freshness를 확인합니다.
4. Orphan·drift·TBR을 owner와 due로 보냅니다.

11. Source에서 build·deploy·rollback까지 수신 팀이 재현하기

그림 6. Tag→lock→clean build→hash verify→synthetic deploy→smoke→rollback을 수신 팀이 실행합니다.

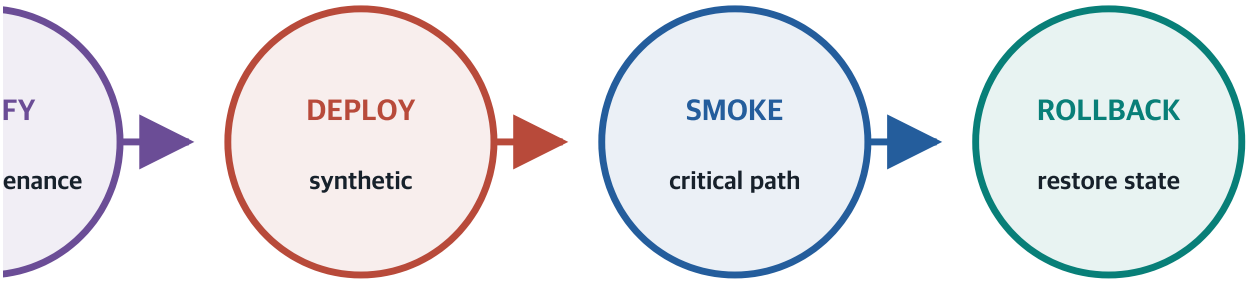
Tag·dependency lock·artifact hash·synthetic deploy·smoke·rollback을 끊김



RECEIVER EXECUTES · HIDDEN STEP

[illegible]

[illegible]킬 없이 실행합니다.



[illegible]0 · HASH MATCH · ROLLBACK PASS

핵심 원칙: 빌드 문서보다 숨은 단계 0인 clean build가 강한 evidence입니다.

기존 담당자 노트북에서만 되는 build는 인계가 아니라 환경 종속입니다. 수신 팀이 깨끗한 합성 환경에서 tagged source와 locked dependency로 artifact를 만들고 hash·smoke·rollback까지 확인합니다.

관점	합성 사례	확인할 evidence
Source	protected branch·tag	provenance
Build	clean environment·lock	hidden step0
Verify	artifact hash	expected match
Deploy	synthetic environment	smoke pass
Rollback	previous state	receiver pass

흔한 오답: 기존 담당자가 build한 artifact와 성공 화면만 전달합니다.

고친 예: 수신 팀이 tag부터 rollback까지 실행하고 command·version·hash·result·owner를 evidence로 남깁니다.

그림을 보며 4단계 연습

1. Release tag와 lock을 고릅니다.
2. Clean build를 실행합니다.
3. Artifact hash와 provenance를 확인합니다.
4. Synthetic deploy·smoke·rollback을 재현합니다.

12. Account·secret·certificate·license 생애주기 인계하기

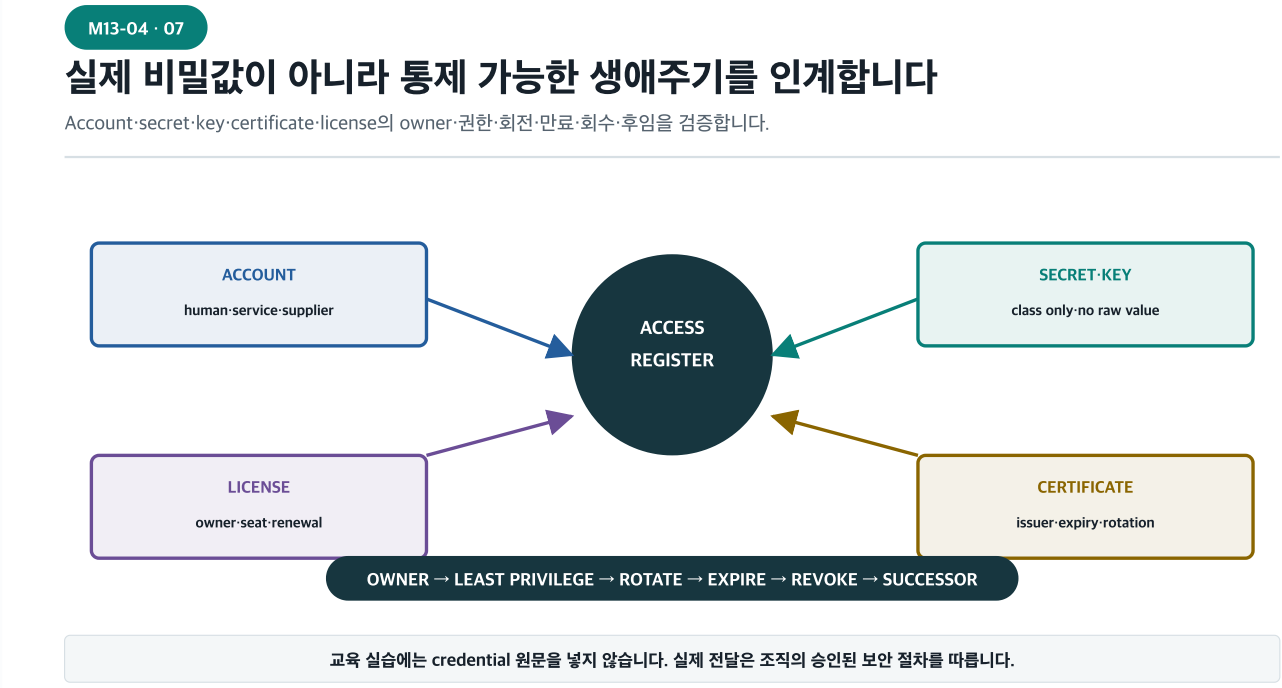
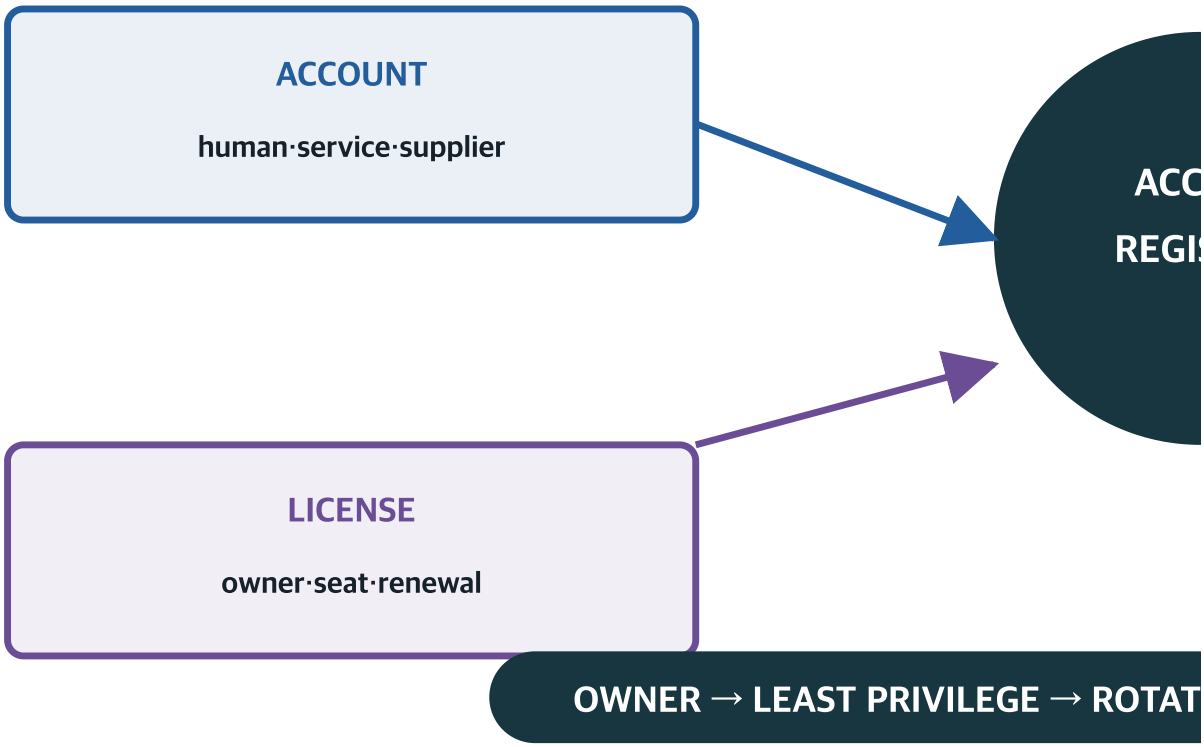
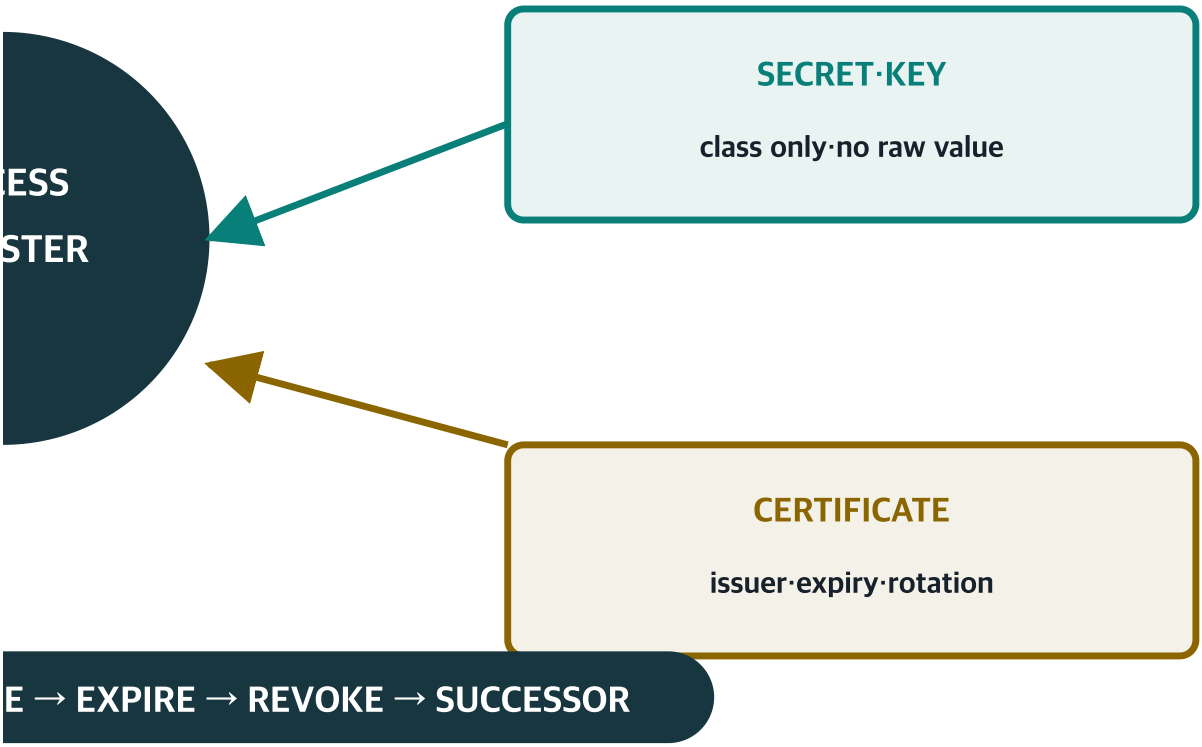


그림 7. 실제 비밀값 없이 owner·권한·rotation·expiry·revocation·successor를 검증합니다.

Account·secret·key·certificate·license의 owner·권한·회전·만료·회수·후임을



검증합니다.


ESS
STER
SECRET·KEY
class only·no raw value
CERTIFICATE
issuer·expiry·rotation
E → EXPIRE → REVOKE → SUCCESSOR

핵심 원칙: Credential 원문이 아니라 통제 가능한 생애주기를 인계합니다.

계정 접근만 열어 주면 service account·certificate·API token·license 갱신이 특정 사람에게 남을 수 있습니다. 교육 실습은 secret class와 metadata만 쓰며 실제 전달은 조직의 승인된 보안 절차를 따릅니다.

관점	합성 사례	확인할 evidence
Human account	role·least privilege	join/review/revoke
Service/supplier	owner·purpose	rotation·orphan0
Secret/key	class only	raw value0
Certificate	issuer·expiry	alert·renewal
License	seat·term·owner	renew·exit

흔한 오답: 비밀번호와 key를 문서나 메신저에 적어 넘기고 인계 evidence라고 부릅니다.

고친 예: 실제 값은 승인된 vault 절차로 다루고 문서에는 class·owner·scope·rotation·expiry·revoke 결과만 남깁니다.

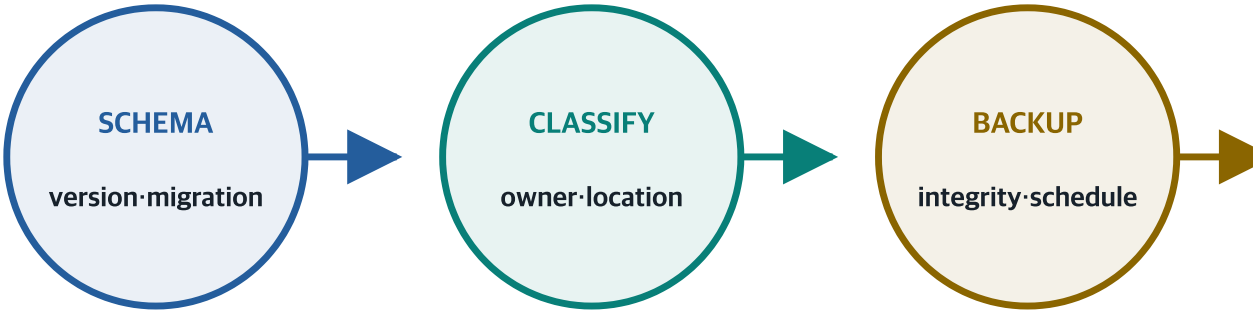
그림을 보며 4단계 연습

1. Account·credential·license inventory를 만듭니다.
2. Owner·least privilege·expiry를 붙입니다.
3. 합성 rotation·revoke를 실행합니다.
4. Orphan·expired·raw secret를 0으로 닫습니다.

13. Data schema·backup·restore·retention을 하나로 잇기

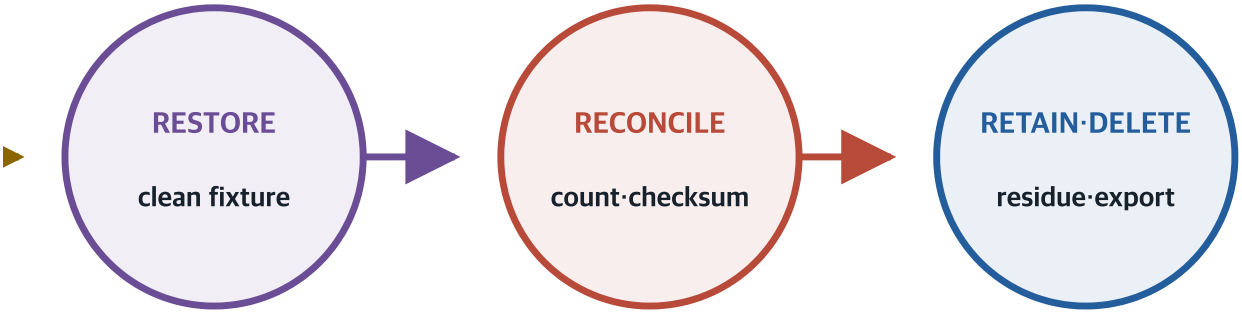
그림 8. Schema·migration·location·backup·restore·reconcile·retention·delete·export를 연결합니다.

Schema·migration·분류·위치·backup·restore·reconciliation·보존·삭제·반출



RESTORE PASS · RTO ≤ 4h ·

을 연결합니다.



RPO ≤ 1h · RECONCILE 100%

핵심 원칙: 백업 파일이 아니라 복구된 결과와 데이터 생애주기가 evidence입니다.

Backup이 있어도 schema version·encryption key·migration 순서·복구 권한·reconciliation이 빠지면 사용할 수 없습니다. 합성 데이터로 restore하고 RTO·RPO·count·checksum·retention·residue를 확인합니다.

관점	합성 사례	확인할 evidence
Schema	version·migration	owner·compatibility
Data map	class·location·copy	retention·export
Backup	schedule·integrity	owner·key class
Restore	clean fixture	RTO≤4h·RPO≤1h
Reconcile/exit	count·checksum·delete	100%·residue decision

흔한 오답: 백업 작업 성공 로그만 보고 복구 가능하다고 선언합니다.

고친 예: 수신 팀이 synthetic backup을 restore하고 schema·record·checksum·RTO/RPO·retention·export를 검증합니다.

그림을 보며 4단계 연습

1. Schema·migration·copy를 map합니다.
2. Backup integrity를 확인합니다.
3. Clean restore와 reconciliation을 실행합니다.
4. Retention·deletion·export·residue를 결정합니다.

14. 연계·dependency의 실패·만료·대조 경로 인계하기

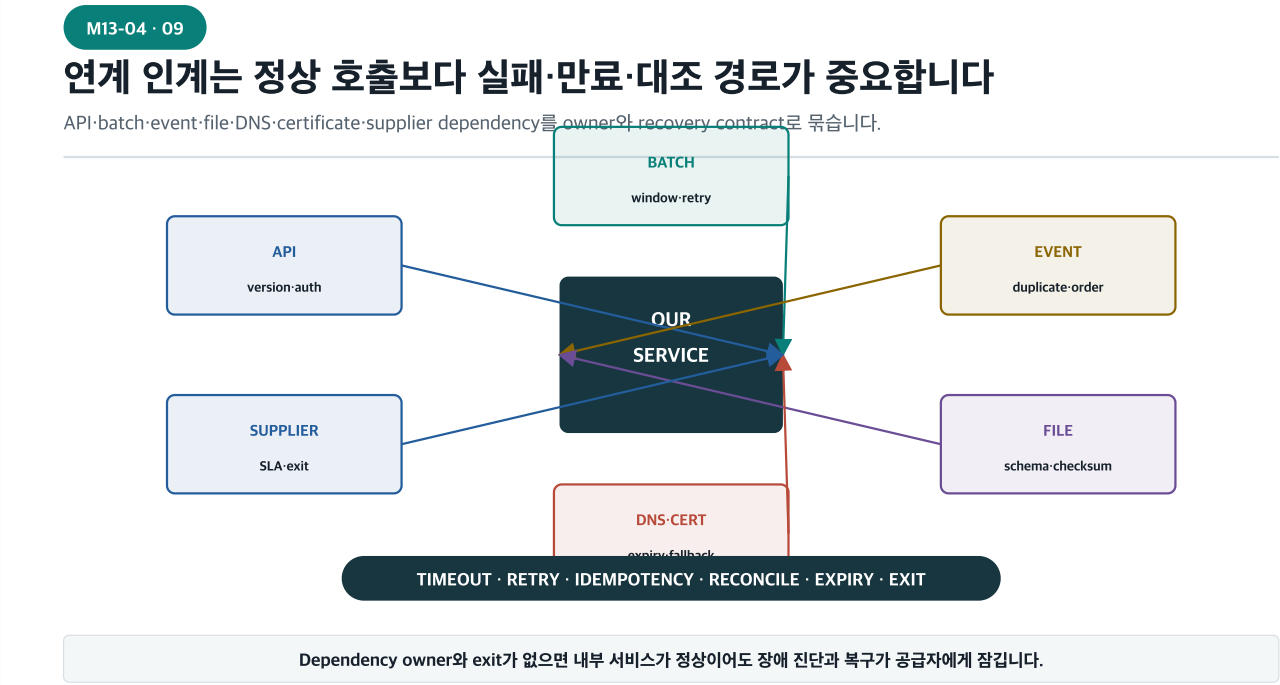
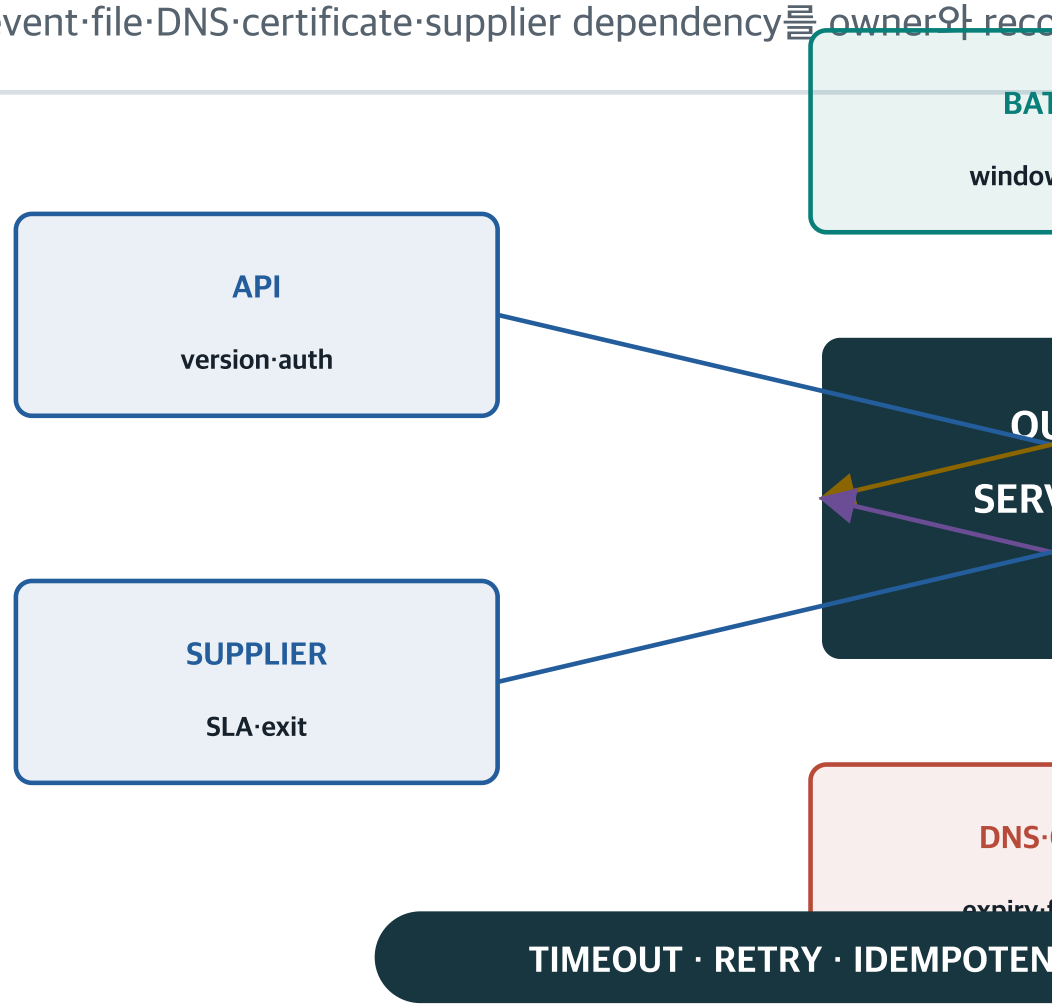
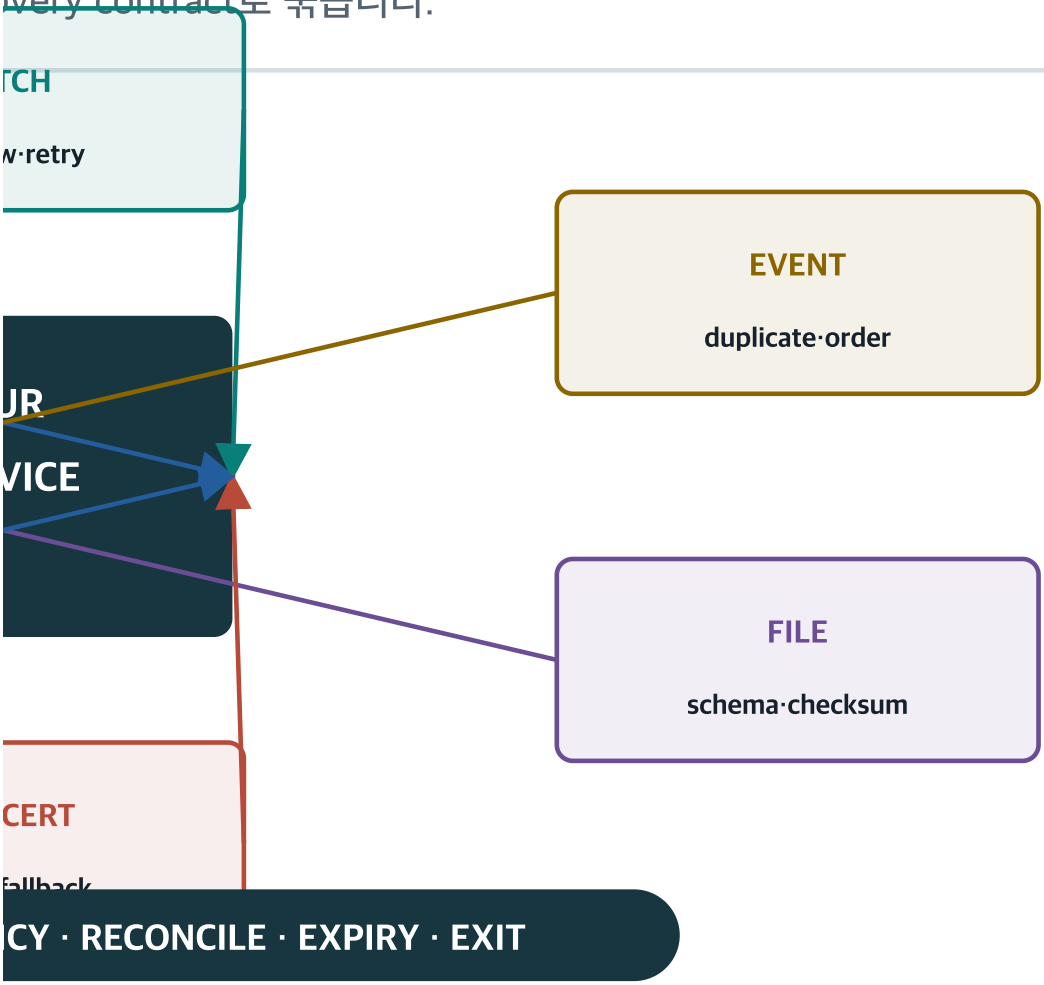


그림 9. Dependency마다 owner·version·timeout·retry·idempotency·reconcile·expiry·exit를 둡니다.

API·batch·event·file·DNS·certificate·supplier dependency를 owner와 recc



very contract로 묶습니다.



핵심 원칙: 정상 호출보다 실패 뒤 상태와 책임이 일치하는지가 중요합니다.

API·batch·event·file·DNS·certificate·supplier는 서로 다른 failure mode를 가집니다. Timeout·duplicate·expiry를 주입하고 side effect 0·reconcile 100%·escalation·exit를 수신 팀이 재현합니다.

관점	합성 사례	확인할 evidence
API	schema·auth·version	timeout·retry
Batch/file	window·format·checksum	restart·reconcile
Event	order·duplicate	idempotency
DNS/cert	endpoint·expiry	alert·fallback
Supplier	SLA·support	escalation·exit

흔한 오답: API 문서와 정상 response 한 건만 전달합니다.

고친 예: Dependency contract에 owner·version·failure·retry·reconcile·expiry·support·exit를 넣고 fault rehearsal을 남깁니다.

그림을 보며 4단계 연습

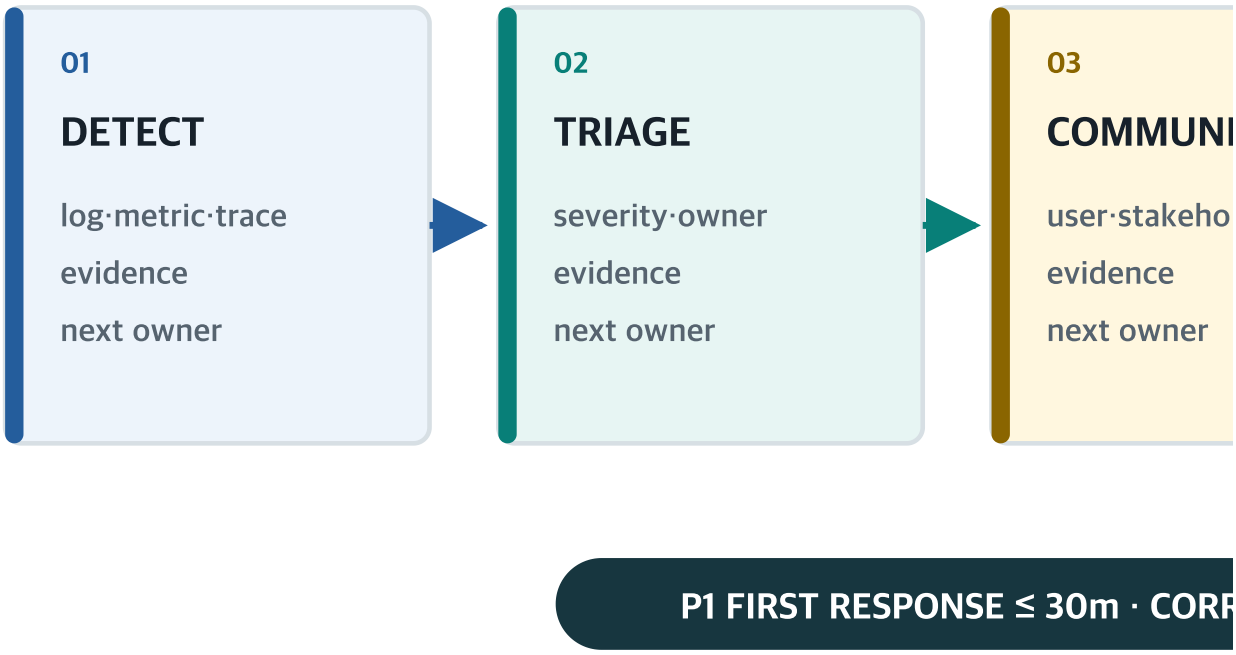
1. 여섯 dependency domain을 inventory합니다.
2. Failure·expiry·owner를 연결합니다.
3. Timeout·duplicate·certificate expiry를 주입합니다.
4. Reconcile·escalation·exit evidence를 닫습니다.

15. 관측·장애·지원 흐름을 수신 팀 행동으로 검증하기

그림 10. Log·metric·trace에서 triage·communication·restore·postmortem까지 이어집니다.

운영 프로세스 체계가 있는 상태

Log·metric·trace·alert에서 triage·communication·restore·postmortem까지



| receiving team이 수행합니다.

RELATION · ESCALATION · OWNER

핵심 원칙: Dashboard가 아니라 사람이 조치할 수 있는 signal과 recovery path를 인계합니다.

Alert가 울려도 severity·on-call·runbook·escalation·communication owner가 없으면 장애 대응은 멈춥니다. 합성 P1 incident에서 수신 팀의 first response·correlation·restore·postmortem owner를 검증합니다.

관점	합성 사례	확인할 evidence
Detect	log·metric·trace·alert	correlation
Triage	severity·impact	incident commander
Communicate	user·stakeholder	cadence·template
Restore	fix·workaround·rollback	owner·time
Learn	problem·postmortem	action owner

흔한 오답: Dashboard URL과 alert 목록을 넘기고 운영 준비가 끝났다고 봅니다.

고친 예: P1 fault를 주입하고 수신 팀이 탐지·분류·소통·복구·학습 action까지 수행합니다.

그림을 보며 4단계 연습

1. P0/P1 user outcome과 signal을 고릅니다.
2. Alert route·severity·role을 확인합니다.
3. 합성 incident를 실행합니다.
4. Communication·restore·postmortem evidence를 남깁니다.

16. 네 가지 유지보수를 하나의 backlog로 관리하기

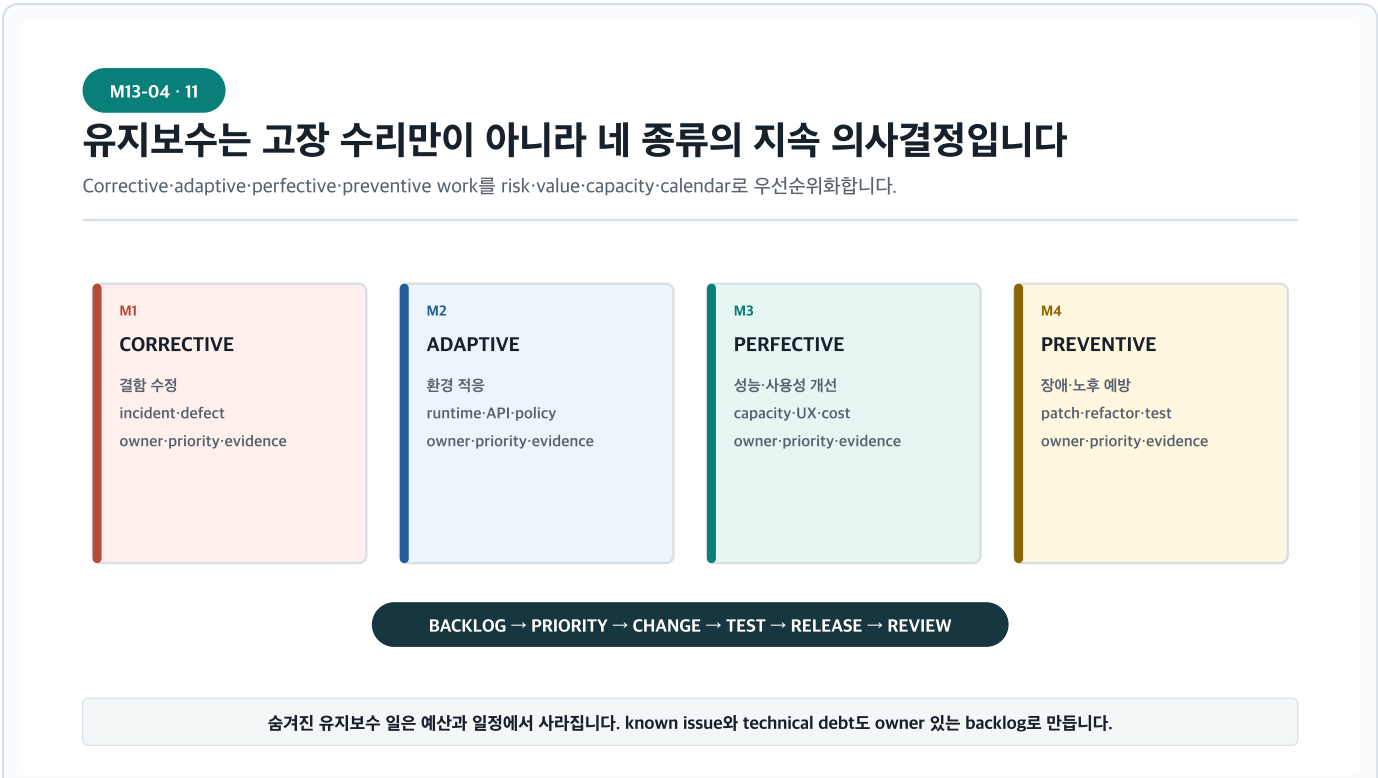


그림 11. 결함 수정·환경 적응·가치 개선·예방 작업을 같은 backlog와 우선순위로 관리합니다.

유지보수형 개선 우선순위 체계

Corrective·adaptive·perfective·preventive work를 risk·value·capacity·cale

M1

CORRECTIVE

결함 수정

incident·defect

owner·priority·evidence

M2

ADAPTIVE

환경 적응

runtime·API·policy

owner·priority·evidence

BACKLOG → PRIORITY → CHANG

ndar로 우선순위화합니다.

M3

PERFECTIVE

성능·사용성 개선

capacity·UX·cost

owner·priority·evidence

M4

PREVENTIVE

장애·노후 예방

patch·refactor·test

owner·priority·evidence

E → TEST → RELEASE → REVIEW

핵심 원칙: 유지보수는 고장 수리만이 아니라 네 종류의 지속 의사결정입니다.

Corrective만 처리하면 runtime EOL·API 변경·성능·기술 부채가 뒤로 밀립니다. 네 유형을 risk·customer value·capacity·cost·deadline으로 비교하고 owner·test·release·review evidence를 붙입니다.

관점	합성 사례	확인할 evidence
Corrective	defect·incident	restore correctness
Adaptive	runtime·API·policy	environment fit
Perfective	performance·UX·cost	value improvement
Preventive	patch·refactor·test	future risk reduction
Backlog	priority·owner·calendar	evidence·review

흔한 오답: 운영 중 생긴 일은 모두 버그로 부르고 긴급한 요청 순서로만 처리합니다.

고친 예: 유형·risk·value·effort·deadline·owner·acceptance·release window를 backlog에 기록합니다.

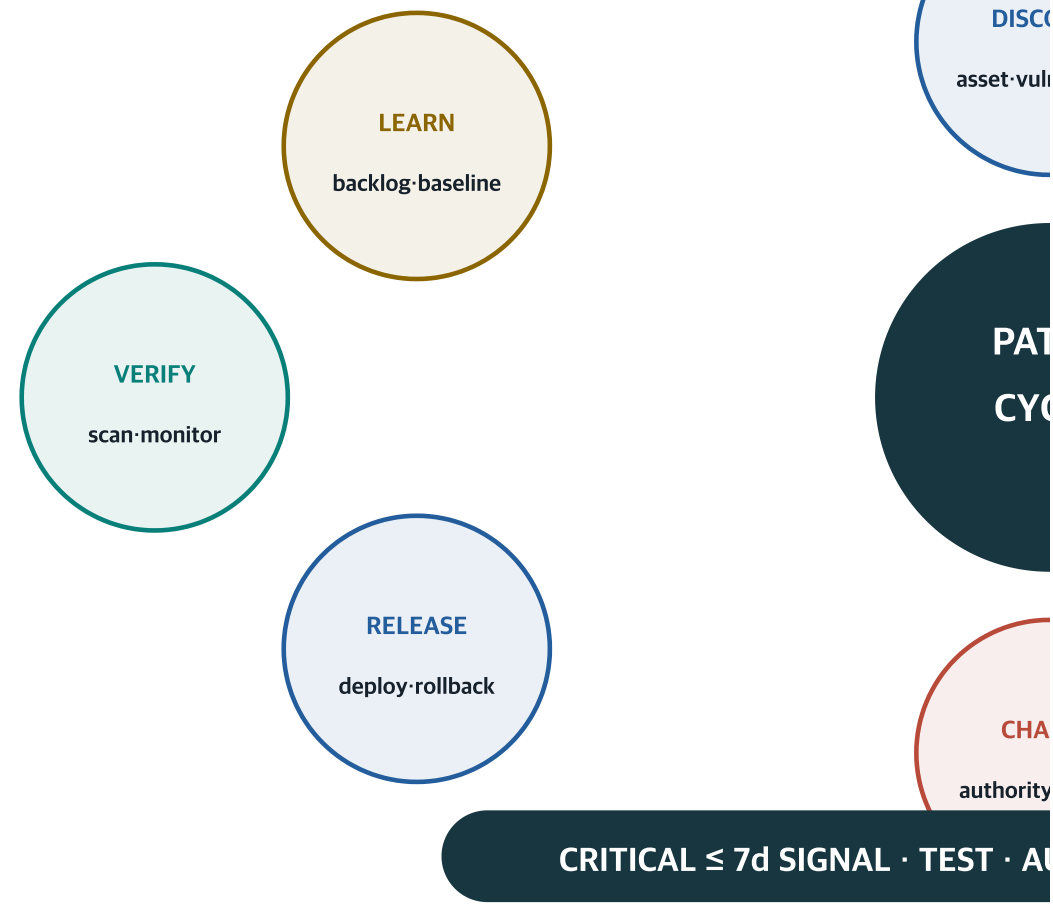
그림을 보며 4단계 연습

1. 현재 작업을 네 유형으로 분류합니다.
2. Risk·value·deadline을 평가합니다.
3. Owner·test·release evidence를 붙입니다.
4. 월간 backlog review 기준을 정합니다.

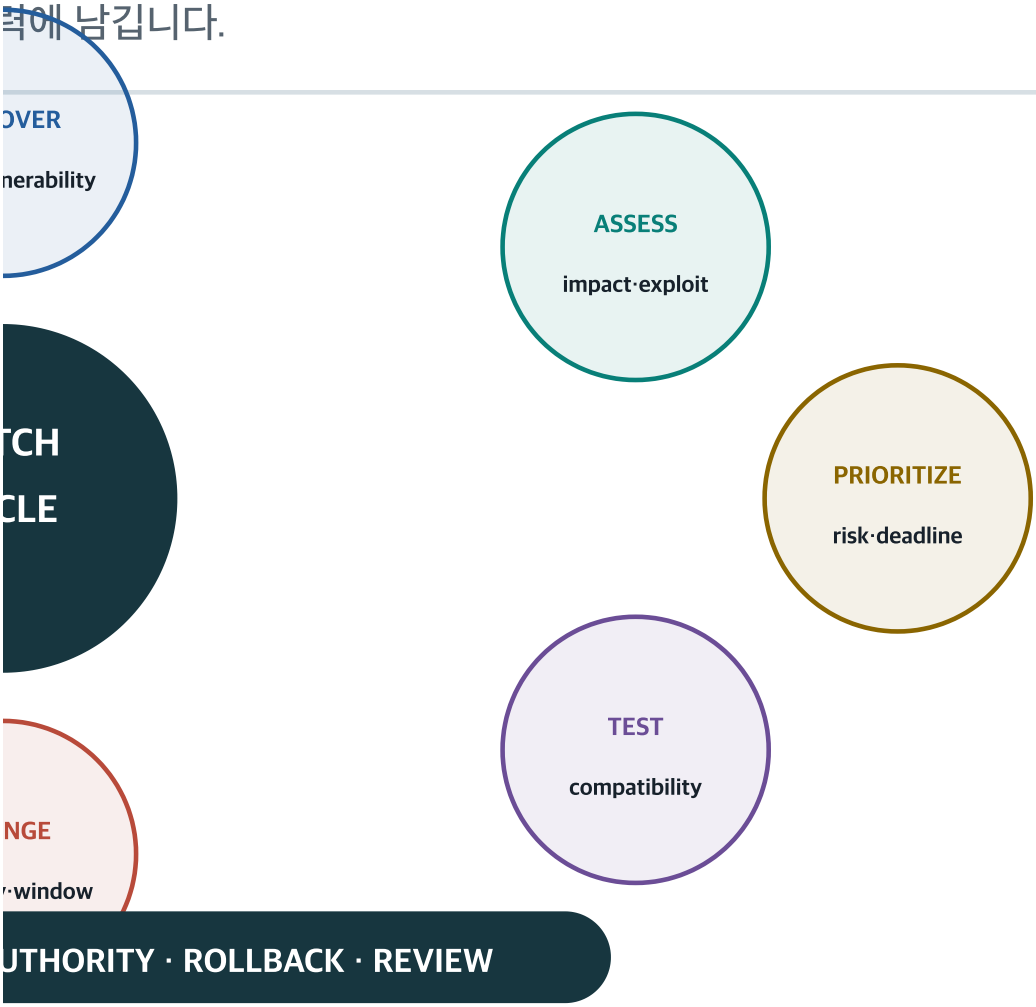
17. Patch·취약점·change·release를 닫힌 주기로 운영하기

그림 12. 자산·취약점 발견부터 test·change·release·rollback·검증·학습까지 닫습니다.

발견부터 영향 평가·우선순위·test·change·release·verify·학습을 owner와 달



력에 남깁니다.


OVER
nerability
ASSESS
impact·exploit
TCH
CLE
PRIORITIZE
risk·deadline
TEST
compatibility
NGE
·window
UTHORITY · ROLLBACK · REVIEW

핵심 원칙: Patch는 설치 버튼이 아니라 위험·변경·복구를 함께 다루는 주기입니다.

긴급 patch도 호환성·업무 영향·change authority·rollback을 무시하면 더 큰 장애를 만들 수 있습니다. 합성 사례의 critical 7일은 학습 signal이며 실제 기한은 exposure·impact·exploit·보완 통제로 정합니다.

관점	합성 사례	확인할 evidence
Discover/assess	asset·CVEs·exposure	impact owner
Prioritize	risk·deadline	exception authority
Test	compatibility·security	pass evidence
Change/release	window·approval·deploy	rollback
Verify/learn	scan·monitor·baseline	backlog update

흔한 오답: 심각도 숫자만 보고 production에 즉시 설치하거나, 위험을 이유로 무기한 미룹니다.

고친 예: 자산·노출·영향·기한·test·change authority·rollback·검증·예외 expiry를 한 record로 묶습니다.

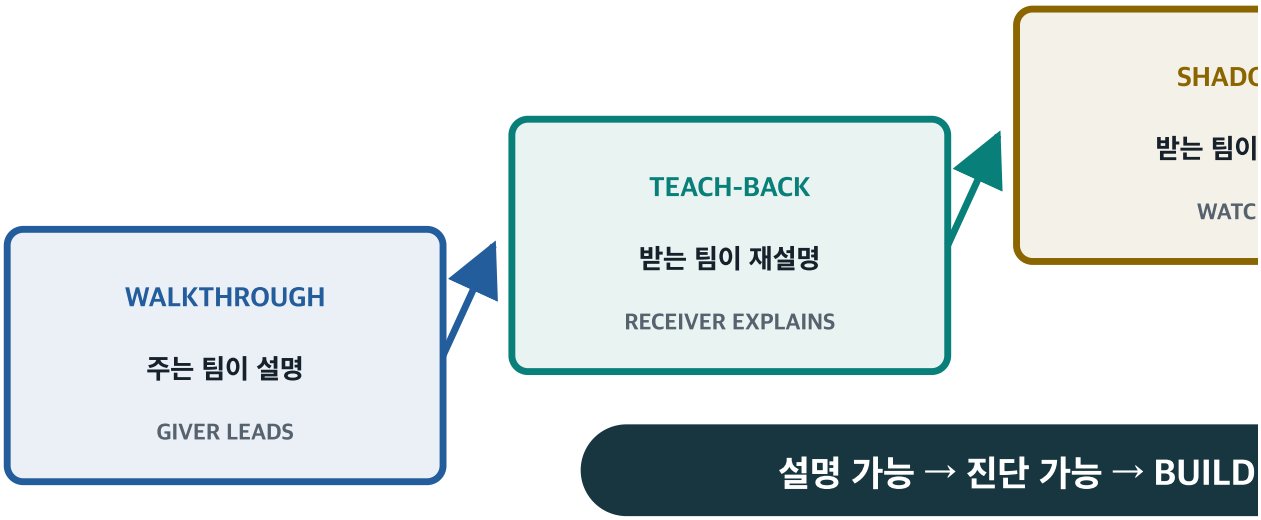
그림을 보며 4단계 연습

1. 합성 critical vulnerability를 고릅니다.
2. Exposure·impact·deadline을 평가합니다.
3. Test·change·rollback을 실행합니다.
4. Verify·baseline·backlog를 갱신합니다.

18. Walkthrough를 teach-back·reverse shadow·독립 수행으로 높이기

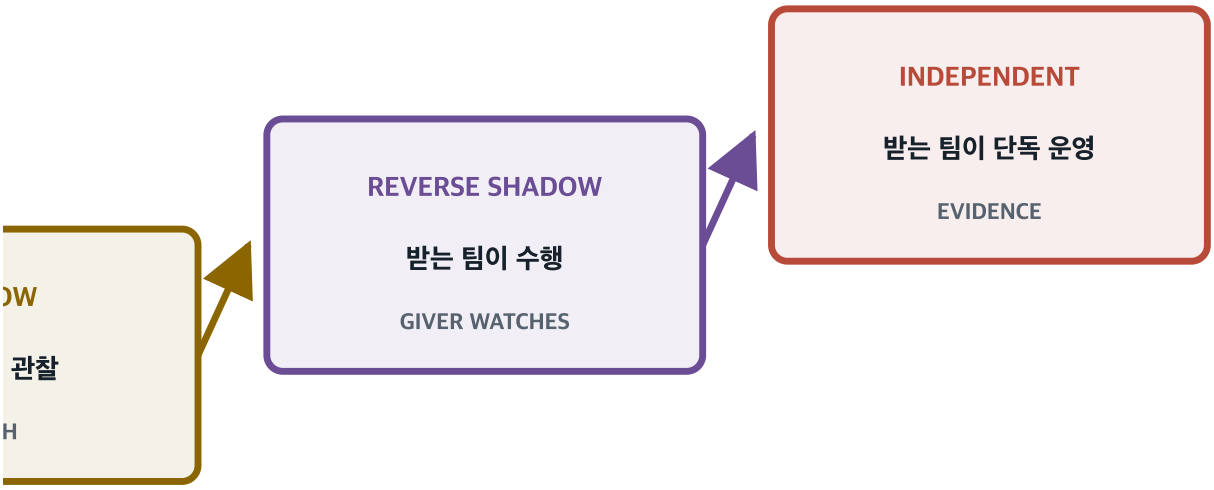
그림 13. 주는 팀의 설명에서 받는 팀의 설명·수행·독립 운영으로 evidence 강도를 높입니다.

Walkthrough에서 teach-back·shadow·reverse shadow·independent task로



[illegible]

[illegible] 증거 강도를 높입니다.



·RESTORE·INCIDENT 독립 수행

핵심 원칙: 지식 이전은 회의 참석 시간이 아니라 수신 팀의 행동 수준으로 측정합니다.

Runbook을 읽어도 architecture 이유·known issue·diagnostic clue는 암묵지로 남을 수 있습니다. 세 session에서 explain·diagnose·reverse shadow·independent build/restore/incident를 확인합니다.

관점	합성 사례	확인할 evidence
Walkthrough	giver explains	attendance
Teach-back	receiver explains	misunderstanding0
Shadow	receiver watches	question log
Reverse shadow	receiver acts	giver observes
Independent	receiver alone	build·restore·incident pass

흔한 오답: 교육 참석자 명단과 녹화 파일을 지식 이전 완료 evidence로 사용합니다.

고친 예: 수신 팀이 구조를 설명하고 장애를 진단하며 P0 task를 독립 수행한 결과와 보완 action을 기록합니다.

그림을 보며 4단계 연습

1. P0 knowledge와 task를 고릅니다.
2. Teach-back 질문을 만듭니다.
3. Shadow와 reverse shadow를 실행합니다.
4. Independent task와 correction count를 기록합니다.

19. 유지보수 달력·known issue·기술 부채·exit를 시간축에 놓기

그림 14. 반복 작업과 expiry·EOL·계약 종료 trigger를 같은 calendar에서 관리합니다.

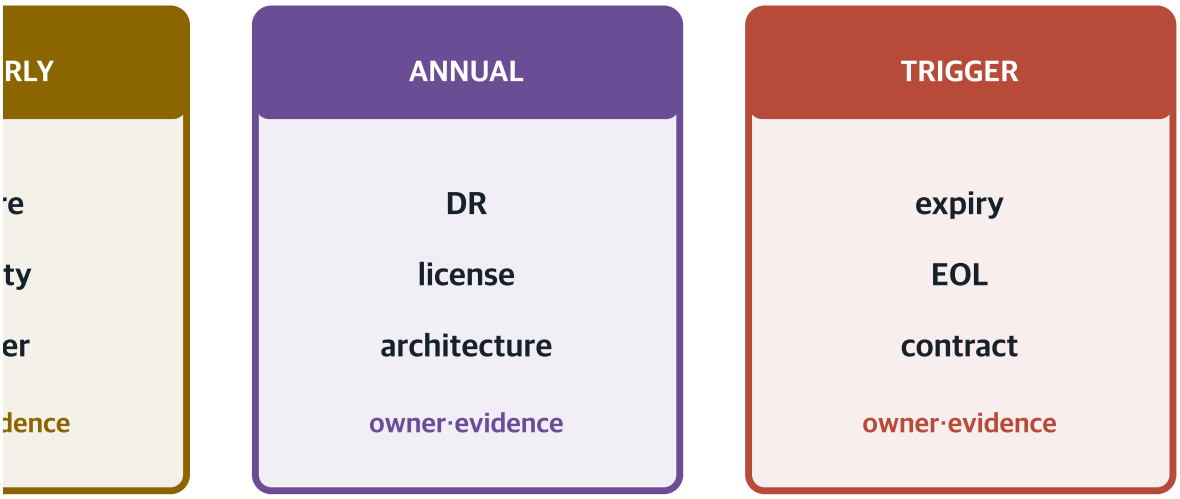
주간·월간·분기·연간·event trigger에 patch·복구·비용·공급자·license·exit를 ㅂ

WEEKLY	MONTHLY	QUARTE
alert	patch	restor
backup	access	capaci
backlog	cost	suppli
owner·evidence	owner·evidence	owner·evi

KNOWN ISSUE · TECH DEBT · RISK · TBR ·

[illegible]

[illegible]배치합니다.



SUCCESSOR · EXPORT · REVOKE · DELETE

핵심 원칙: 보이지 않는 반복 업무와 종료 조건을 달력에 올려야 owner·capacity·budget이 생깁니다.

Patch·restore drill·access review·certificate·license·supplier·capacity·cost는 서로 다른 주기로 돌아옵니다. Known issue·technical debt·residual risk·TBR을 숨기지 않고 expiry·successor·export·revoke·decommission trigger와 연결합니다.

관점	합성 사례	확인할 evidence
Weekly	alert·backup·backlog	ops owner
Monthly	patch·access·cost	maintenance owner
Quarterly	restore·capacity·supplier	service owner
Annual	DR·license·architecture	authority review
Trigger	expiry·EOL·contract·exit	successor·decommission

흔한 오답: 문제가 생기면 대응한다는 문장만 있고 반복 일정·예산·후임·종료 조건은 없습니다.

고친 예: 작업마다 cadence·owner·input·evidence·escalation·successor를 정하고 trigger event를 같은 달력에 둡니다.

그림을 보며 4단계 연습

1. 반복 업무와 법정·계약 일정을 모읍니다.
2. Cadence·owner·capacity를 배정합니다.
3. Known issue·debt·risk·TBR을 연결합니다.
4. Expiry·EOL·exit rehearsal을 추가합니다.

20. Handover Acceptance Gate를 닫고 M13-05로 넘기기

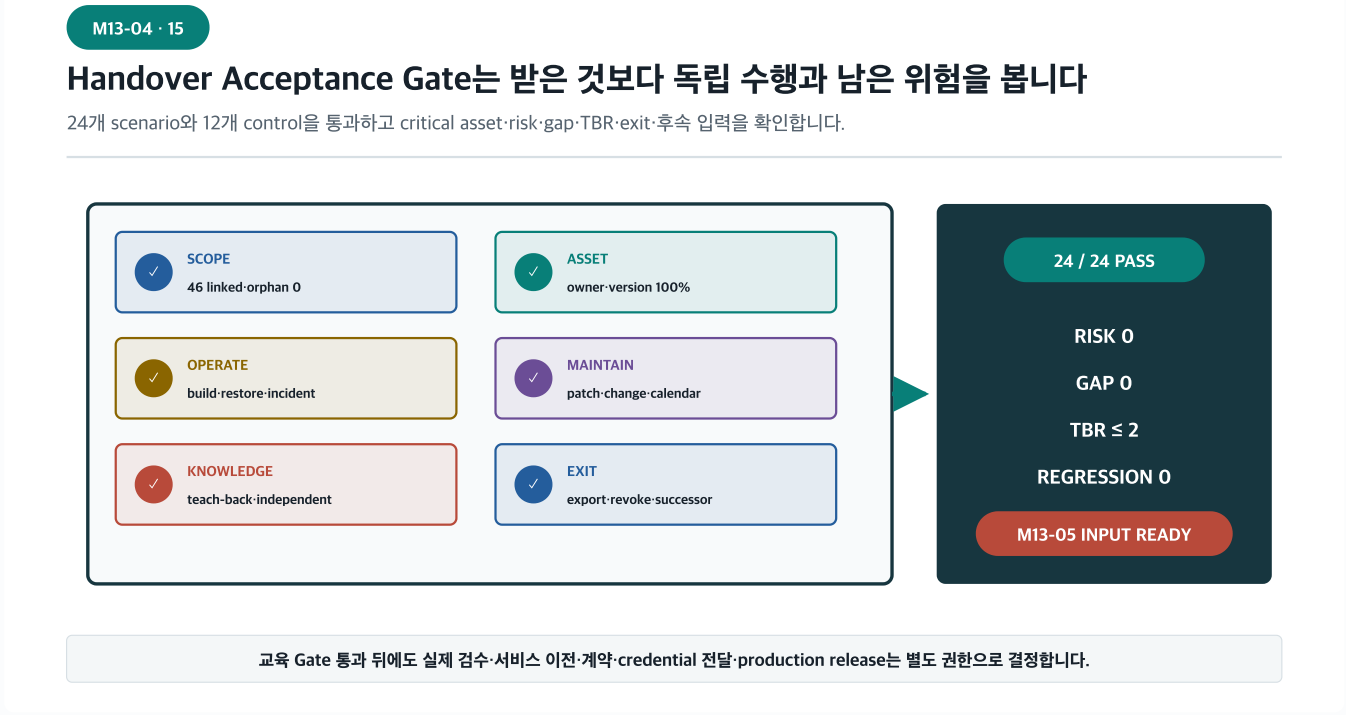
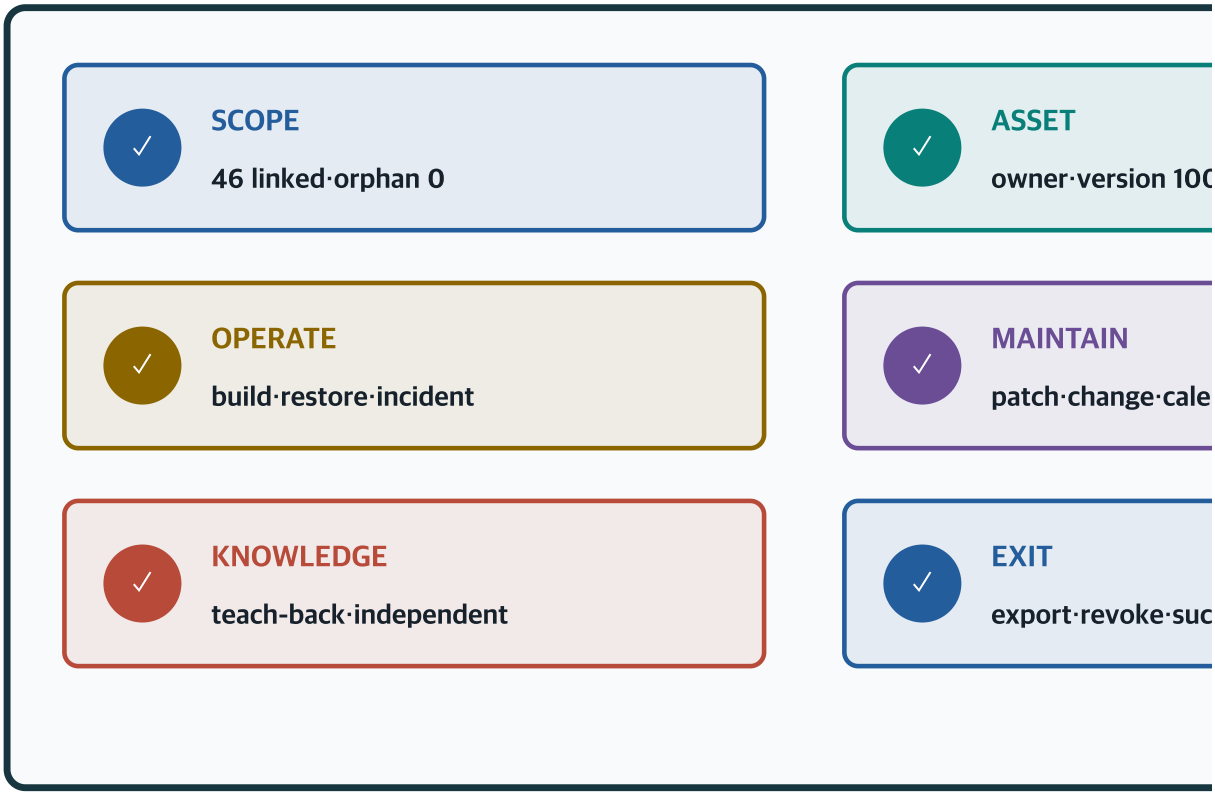


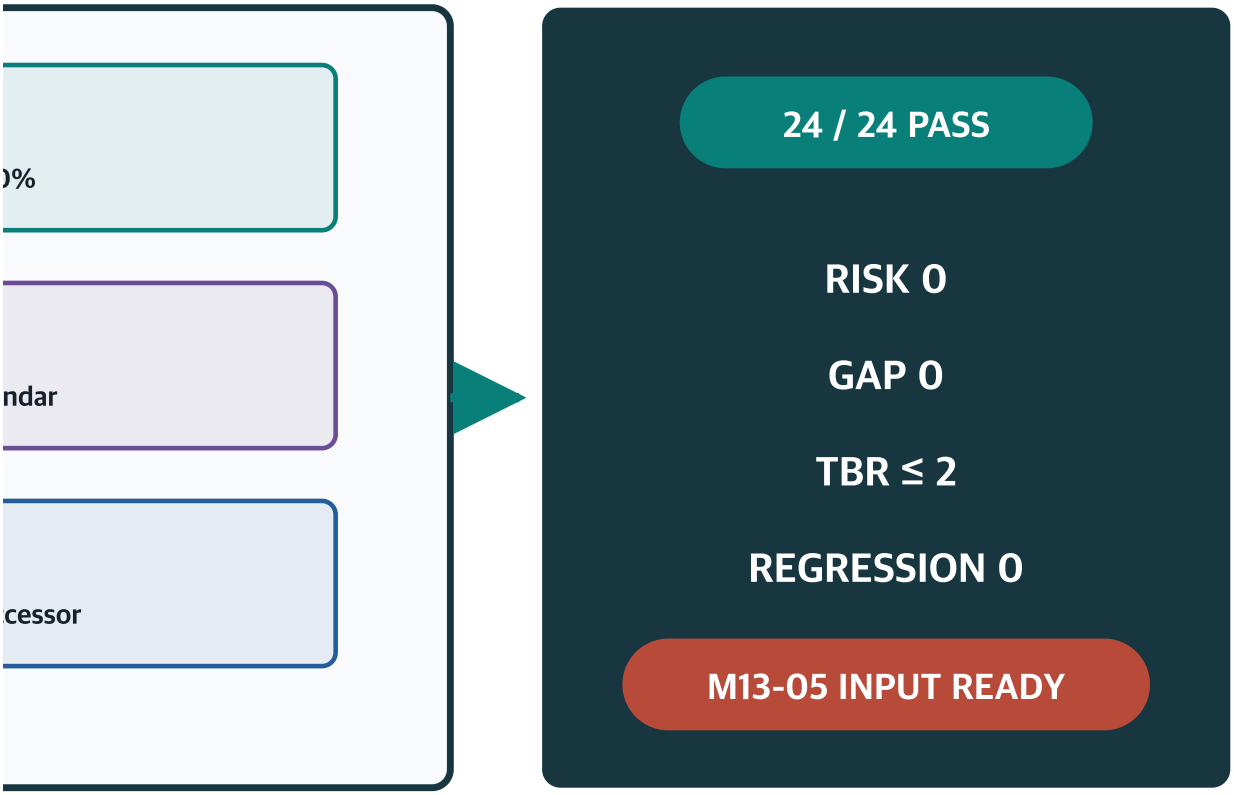
그림 15. Scope·asset·operate·maintain·knowledge·exit를 확인하고 잔여 위험과 M13-05 입력을 남깁니다.

Handover Acceptance Gate는 인증 체크

24개 scenario와 12개 control을 통과하고 critical asset·risk·gap·TBR·exit·후



속 입력을 확인합니다.


0%
ndar
cessor
24 / 24 PASS
RISK 0
GAP 0
TBR ≤ 2
REGRESSION 0
M13-05 INPUT READY

핵심 원칙: Gate는 받은 수보다 독립 수행·critical asset·남은 위험·exit·후속 owner를 봅니다.

최종 합성안은 24/24 scenario, 12/12 control, 네 lane과 12 coverage domain 100%, authority conflict·orphan critical asset·critical risk·gap 0, TBR 2를 요구합니다. PASS 뒤 실제 검수·transfer·contract·release는 별도 권한입니다.

관점	합성 사례	확인할 evidence
Scope/asset	46 linked·owner/version	orphan0
Operate	build·restore·incident	receiver pass
Maintain	backlog·patch·change	calendar owner
Knowledge/exit	teach-back·export·revoke	successor
Gate/handoff	24/24·risk0·gap0·TBR2	M13-05 input

흔한 오답: handover_acceptance_package_ready를 실제 인수 검수와 production release 승인으로 사용합니다.

고친 예: 교육 Gate 결과·잔여 TBR·authority boundary·12 evidence·receiving owner·M13-05 business narrative input을 함께 남깁니다.

그림을 보며 4단계 연습

1. 세 variant를 실행합니다.
2. Fixed·regressed·risk·gap·TBR을 비교합니다.
3. 여섯 Gate와 authority boundary를 확인합니다.
4. M13-05 value·cost·risk 설명 입력을 작성합니다.

21. 12개 Handover·Maintenance Control

ID	통제	최소 evidence
CTRL-01	M12·M13 source·scope·authority handoff	Handover input manifest
CTRL-02	Ownership·RACI·acceptance authority	Ownership and authority matrix
CTRL-03	Asset·configuration·version manifest	Asset and configuration manifest
CTRL-04	Repository·build·deploy·rollback reproducibility	Build release reproducibility pack
CTRL-05	Account·secret·certificate·license transfer	Access credential license transfer record
CTRL-06	Data·schema·backup·restore·retention	Data recovery and retention rehearsal
CTRL-07	Integration·dependency·reconciliation contract	Dependency and integration pack
CTRL-08	Observability·alert·incident·problem rehearsal	Operations incident rehearsal
CTRL-09	Service desk·supplier·warranty·escalation	Support and supplier map
CTRL-10	Maintenance backlog·patch·vulnerability·change·release	Maintenance backlog and calendar
CTRL-11	Knowledge transfer·teach-back·shadow	Knowledge transfer and teach-back record
CTRL-12	Handover Acceptance Gate·exit·M13-05	Handover Acceptance Gate

22. 24개 합성 시나리오 포트폴리오

ID	Lane	시나리오	위험	Source → Evidence	Controls
SC-01	governance-asset-transfer	M12·M13 46개 source를 handover input으로 인수	ownership-authority-gap	product architecture vendor readiness package → handover input manifest	CTRL-01, CTRL-02
SC-02	governance-asset-transfer	교육 Gate와 실제 service transfer·계약 권한 분리	ownership-authority-gap	handover question → authority boundary	CTRL-01, CTRL-02, CTRL-12

ID	Lane	시나리오	위험	Source → Evidence	Controls
SC-03	governance-asset-transfer	Service·system·data·security·operations owner와 RACI	ownership-authority-gap	organization readiness authority → ownership matrix	CTRL-02
SC-04	governance-asset-transfer	Repository·artifact·IaC·runtime·document·license inventory	asset-configuration-drift	current assets and delivery evidence → asset manifest	CTRL-03
SC-05	governance-asset-transfer	Configuration baseline·checksum·environment drift 검증	asset-configuration-drift	manifest and synthetic environment → configuration status record	CTRL-03, CTRL-04
SC-06	governance-asset-transfer	Handover item을 owner·rehearsal·acceptance·TBR로 정규화	ownership-authority-gap	handover claims → acceptance register	CTRL-01, CTRL-02, CTRL-03
SC-07	build-access-data	Repository access·branch·tag·provenance 인수	access-secret-license-gap	source and repository policy → repository access record	CTRL-04, CTRL-05
SC-08	build-access-data	Clean environment에서 build·artifact hash 재현	asset-configuration-drift	tagged source and locked dependencies → build reproducibility evidence	CTRL-03, CTRL-04
SC-09	build-access-data	Synthetic deploy·smoke·rollback을 receiving team이 수행	recovery-integration-blindspot	verified artifact and deployment procedure → release rollback evidence	CTRL-04, CTRL-08
SC-10	build-access-data	Account·secret class·certificate·license rotation과 revoke	access-secret-license-gap	access and credential inventory → transfer rotation record	CTRL-05
SC-11	build-access-data	Data schema·migration·classification·retention map	recovery-integration-blindspot	organization readiness data lifecycle → data operation map	CTRL-06
SC-12	build-access-data	Backup restore·RTO·RPO·reconciliation rehearsal	recovery-integration-blindspot	synthetic backup fixture → restore evidence	CTRL-06, CTRL-08
SC-13	operations-support-maintenance	API·batch·event·file·DNS·supplier dependency owner	recovery-integration-blindspot	integration and environment inventory → dependency contract	CTRL-07
SC-14	operations-support-maintenance	Timeout·duplicate·certificate expiry·reconciliation rehearsal	recovery-integration-blindspot	synthetic dependency faults → integration recovery evidence	CTRL-05, CTRL-07, CTRL-08

ID	Lane	시나리오	위험	Source → Evidence	Controls
SC-15	operations-support-maintenance	Log·metric·trace·dashboard·alert·on-call coverage	support-maintenance-gap	P0 and P1 service outcomes → observability coverage record	CTRL-08
SC-16	operations-support-maintenance	P1 incident triage·communication·restore·postmortem	support-maintenance-gap	synthetic P1 incident → incident rehearsal evidence	CTRL-08, CTRL-09
SC-17	operations-support-maintenance	Service desk·supplier warranty·severity·escalation 검증	support-maintenance-gap	support and contract inputs → support operating model	CTRL-09
SC-18	operations-support-maintenance	Maintenance type·patch·vulnerability·change·release calendar	support-maintenance-gap	known issues and risk register → maintenance backlog and calendar	CTRL-10
SC-19	knowledge-continuity-exit	Architecture·decision·runbook·FAQ·known issue 연결	knowledge-exit-dependency	M12 and M13 information items → operator knowledge map	CTRL-03, CTRL-11
SC-20	knowledge-continuity-exit	Receiving team teach-back과 diagnostic task	knowledge-exit-dependency	walkthrough and runbooks → teach-back evidence	CTRL-11
SC-21	knowledge-continuity-exit	Shadow→reverse shadow→독립 운영 rehearsal	knowledge-exit-dependency	synthetic operations tasks → receiver independence record	CTRL-08, CTRL-11
SC-22	knowledge-continuity-exit	Known issue·technical debt·residual risk·TBR backlog	knowledge-exit-dependency	defect risk and exception records → maintenance decision backlog	CTRL-10, CTRL-11, CTRL-12
SC-23	knowledge-continuity-exit	Capacity·cost·license·review·maintenance calendar와 successor	support-maintenance-gap	service signals and obligations → maintenance calendar	CTRL-05, CTRL-09, CTRL-10, CTRL-11
SC-24	knowledge-continuity-exit	Exit·export·revoke·decommission·M13-05 Handover Gate	knowledge-exit-dependency	all handover evidence → Handover Acceptance Gate	CTRL-01, CTRL-12

네 lane은 각각 6개 scenario를 가집니다: `governance-asset-transfer`, `build-access-data`, `operations-support-maintenance`, `knowledge-continuity-exit`.

23. 4개 mode·8개 outcome·12개 evidence 한눈표

ID	Mode	형태	강점	위험	상태
HND-A	문서 덤프형	file share에 source·manual을 한꺼번에 전달	빠르게 파일 존재를 확인	owner·version·reproduction·receiving proof가 없음	blocked-mode
HND-B	Shadow 의존형	기존 담당자 시연과 동행 중심	현장 맥락을 직접 관찰	receiving team이 혼자 실행하지 않아 사람 종속이 남음	partial-mode
HND-C	Evidence-led teach-back형	manifest·rehearsal·teach-back·reverse shadow·Gate	독립 build·restore·incident·maintenance를 증명	실제 transfer·contract·approval은 별도 authority 필요	synthetic-learning-candidate
HND-D	공급자 black-box형	supplier console·문서·인력에 지속 의존	초기 내부 부담이 작음	access·diagnosis·exit·successor capability가 잠김	deferred-trigger-mode

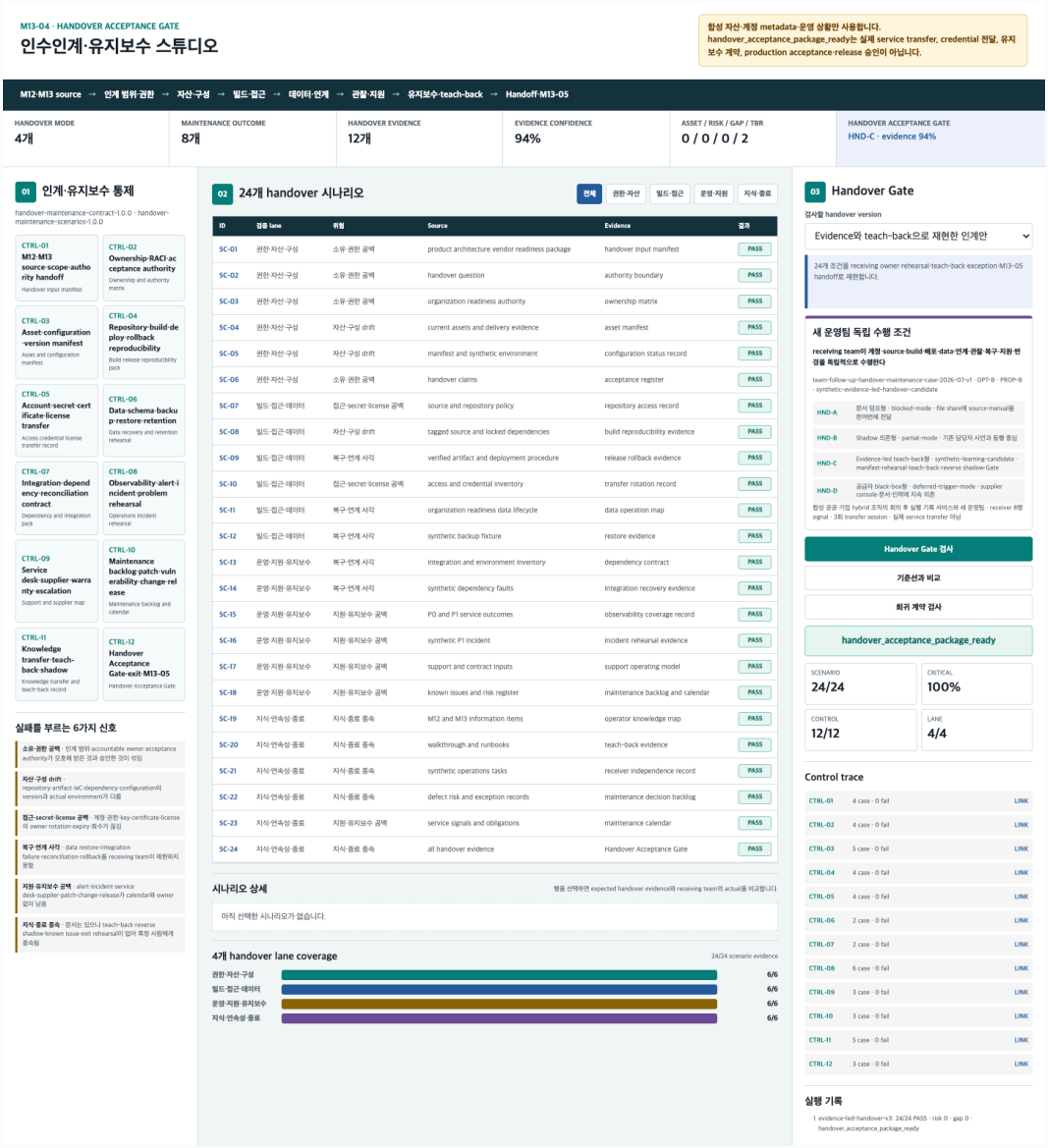
ID	Outcome	Stimulus	Artifact	Measure
OUT-01	authority and inventory	accepts one handover item	source·scope·asset·RACI	46 linked; orphan0; conflict0; 100% critical item owner and version
OUT-02	source build release	builds, deploys and rolls back a tagged release	repo·lock·artifact·IaC·config	clean build pass; hash match; smoke pass; rollback pass; manual hidden step0
OUT-03	access credential license	maps, rotates and revokes synthetic access	account·role·secret class·certificate·license	critical owner100%; orphan account0; expired certificate0; raw secret0
OUT-04	data recovery	restores, reconciles and exports a synthetic dataset	schema·migration·backup·retention	restore pass; RTO≤4h; RPO≤1h; reconcile100%; residue decision present
OUT-05	integration dependency	injects timeout, duplicate and certificate expiry	API·batch·event·DNS·certificate·supplier	duplicate side effect0; reconcile100%; dependency owner100%; expiry alert present
OUT-06	operations support	responds to a P1 synthetic incident	log·metric·trace·alert·runbook·service desk	first response≤30m; correlation present; escalation complete; postmortem owner present
OUT-07	maintenance change security	prioritizes and releases a synthetic critical patch	vulnerability·test·change·release·rollback	critical patch≤7d signal; test pass; approval boundary; rollback pass; calendar owner

ID	Outcome	Stimulus	Artifact	Measure
OUT-08	knowledge continuity exit	performs teach-back, reverse shadow and exit rehearsal	runbook·known issue·backlog·export·decommission	3/3 sessions; independent task pass; critical unknown0; export readable; M13-05 linked

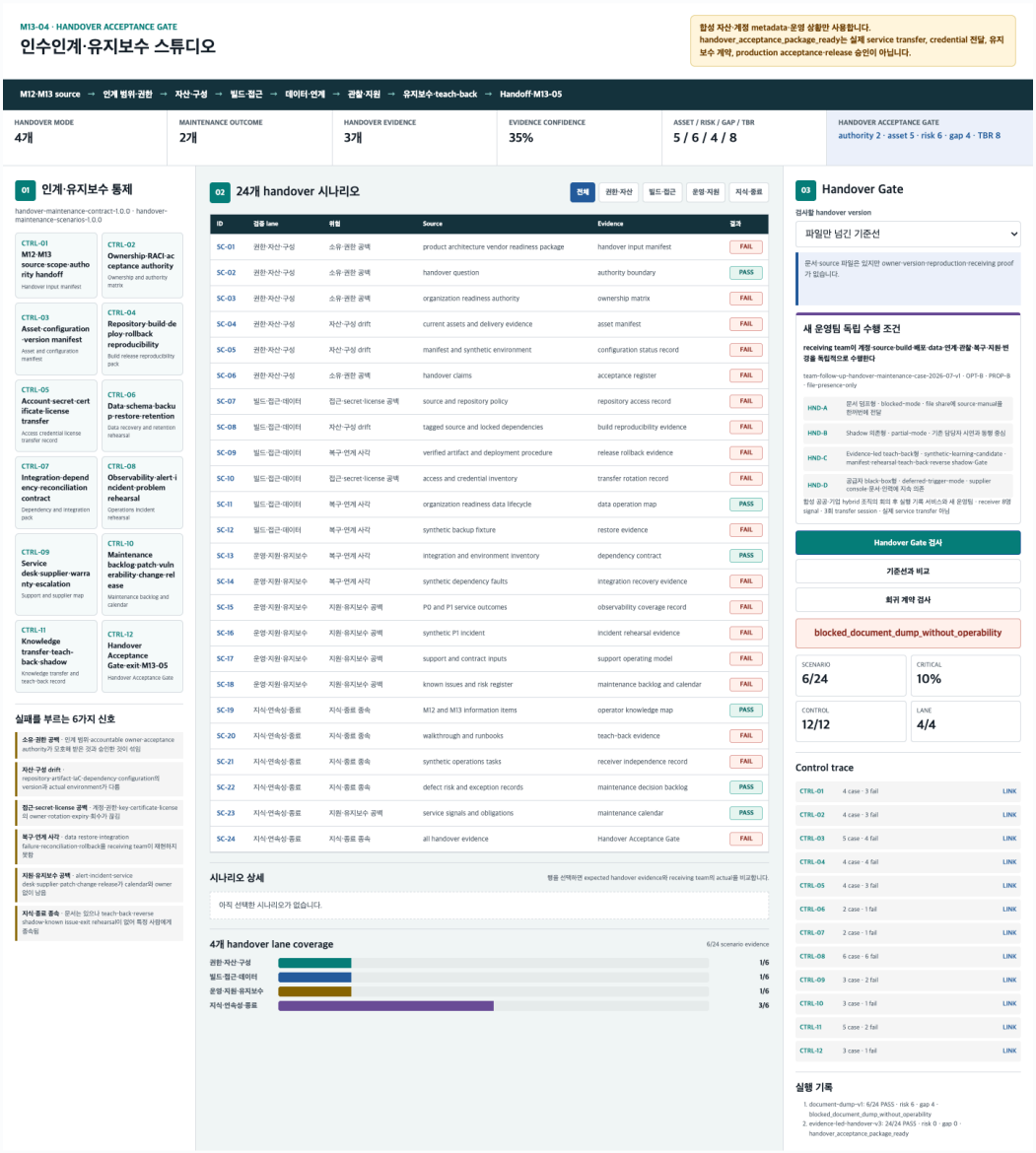
ID	Evidence	최소 proof
EVD-01	Handover input manifest	M12·M13 version·scope·46 linked·orphan·conflict·TBR
EVD-02	Ownership and authority matrix	service·system·data·security·operations·supplier·acceptance·exception
EVD-03	Asset and configuration manifest	repo·artifact·dependency·IaC·runtime·config·docs·license·checksum
EVD-04	Build release reproducibility pack	clean build·artifact hash·synthetic deploy·smoke·rollback·receiving owner
EVD-05	Access credential license transfer record	account·role·secret class·certificate·license·rotation·expiry·revoke
EVD-06	Data recovery and retention rehearsal	schema·migration·backup·restore·RTO/RPO·retention·deletion·export
EVD-07	Dependency and integration pack	API·batch·event·file·DNS·certificate·supplier·failure·reconcile·exit
EVD-08	Operations incident rehearsal	log·metric·trace·alert·on-call·incident·communication·postmortem
EVD-09	Support and supplier map	service desk·severity·response·escalation·warranty·supplier evidence
EVD-10	Maintenance backlog and calendar	maintenance type·patch·vulnerability·capacity·cost·change·release·rollback
EVD-11	Knowledge transfer and teach-back record	runbook·walkthrough·teach-back·shadow·reverse shadow·independent task
EVD-12	Handover Gate and successor record	critical asset·risk·gap·TBR·exception·exit·decommission·M13-05

24. 실습 스튜디오: 세 handover version 실행

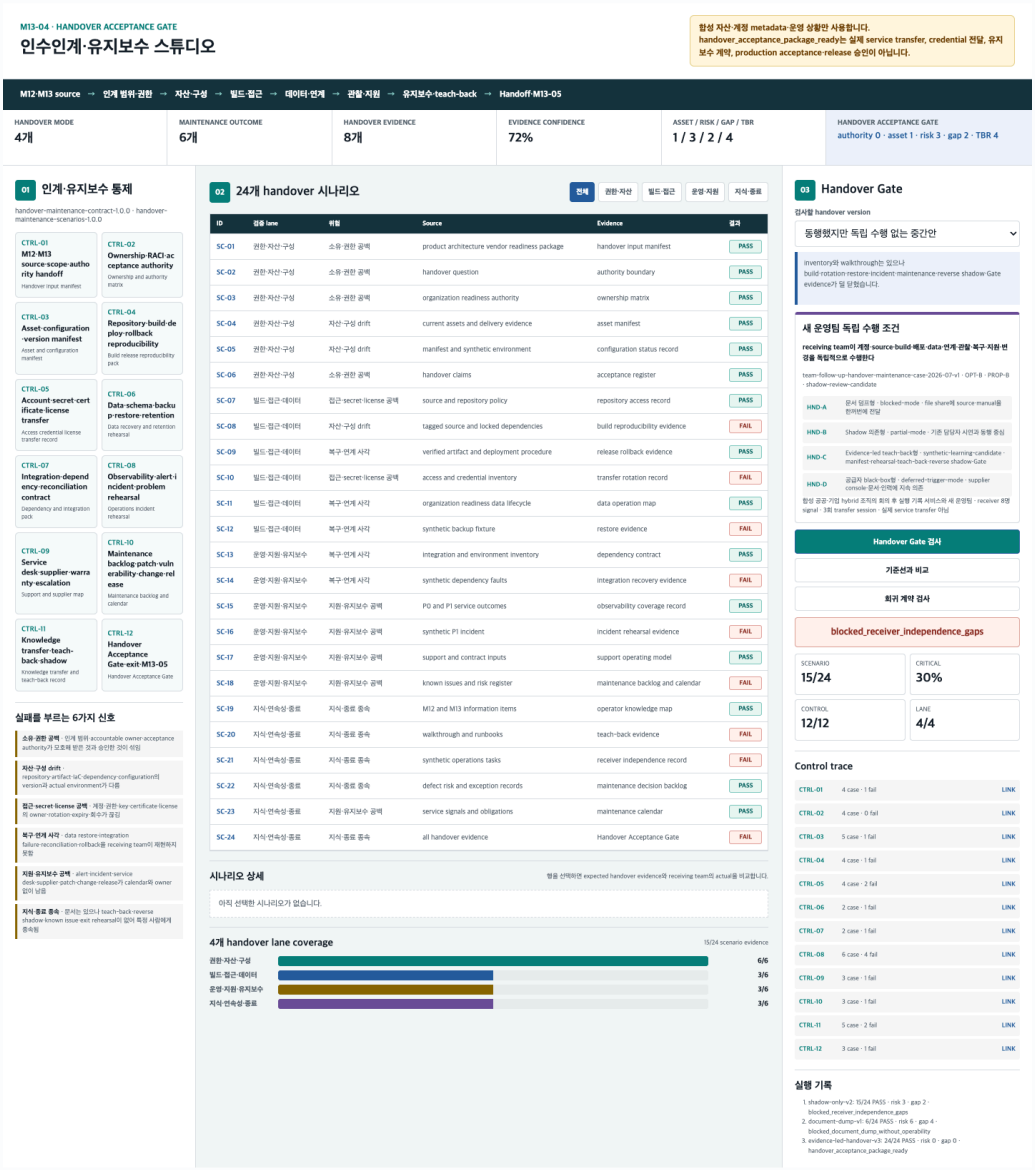
실습은 loopback-only·standard-library-only이며 실제 조직·사람·개인정보·계정·secret·production resource·외부 network를 사용하지 않습니다. 같은 24 scenario에서 파일 존재, shadow 의존, evidence-led 독립 수행의 expected/actual 차이를 비교합니다.



실습 화면 1. Candidate는 24/24, evidence12, outcome8, confidence94%, asset/risk/gap/TBR 0/0/0/2를 표시합니다.

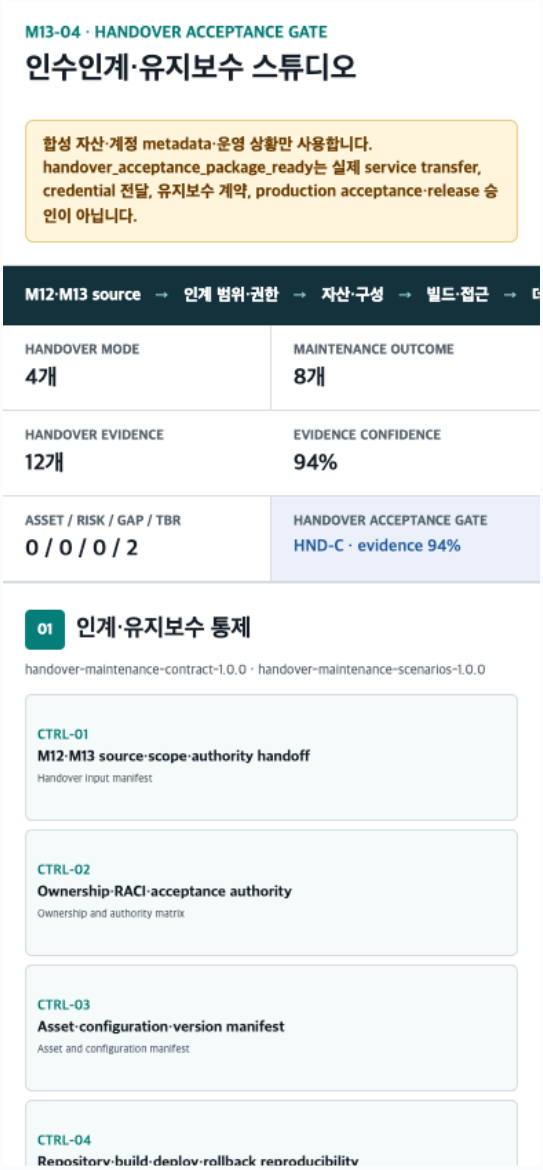


실습 화면 2. Baseline은 6/24, evidence3, outcome2, risk6·gap4·TBR8로 문서 덤프의 공백을 드러냅니다.



실습 화면 3. 중간안은 15/24이며 clean build·rotation·restore·incident·maintenance·reverse shadow·Gate가 덜 닫혔습니다.

실습 화면 4. SC-16에서 first response·severity·communication·restore·postmortem owner를 나란히 봅니다.



실습 화면 5. 모바일에서도 실제 transfer·credential·계약·release가 아님을 먼저 확인합니다.

실습 화면 6. SC-21 shadow→reverse shadow→독립 운영의 expected/actual을 좁은 화면에서도 확인합니다.

Version	Pass	Fail	주요 상태	판정
document-dump-v1	6	18	evidence3·outcome2·confidence35%·risk6·gap4·TBR8	blocked_document_dump_without_operability
shadow-only-v2	15	9	evidence8·outcome6·confidence72%·risk3·gap2·TBR4	blocked_receiver_independence_gaps
evidence-led-handover-v3	24	0	evidence12·outcome8·confidence94%·risk0·gap0·TBR2	handover_acceptance_package_ready

25. 90분 실습 워크북

시간	행동	남길 evidence	통과 질문
0~10분	46 source·scope·authority	input manifest	교육 Gate와 실제 transfer를 구분했나
10~20분	RACI·asset/config manifest	ownership·asset record	critical owner/version이 있는가
20~30분	Repo·clean build·hash	build reproducibility	hidden step 0인가
30~40분	Deploy·smoke·rollback·access	release/access record	수신 팀이 직접 했나
40~50분	Data·backup·restore·retention	recovery evidence	RTO/RPO·reconcile가 닫혔나
50~60분	Dependency·expiry·reconciliation	integration pack	timeout·duplicate·exit를 다뤘나
60~70분	Observability·incident·support	incident rehearsal	탐지·소통·복구·학습이 이어지나
70~80분	Maintenance·patch·change·calendar	backlog·calendar	네 유형과 rollback이 보이나
80~90분	Teach-back·exit·Gate·M13-05	independence·Gate	risk0·gap0·TBR≤2·authority가 명시됐나

```
Input source IDs·versions + answered/unanswered authority:
Ownership RACI + acceptance/exception/escalation:
Asset/config manifest + criticality + freshness:
Repository/tag/lock + clean build + artifact hash:
Synthetic deploy + smoke + rollback + receiving executor:
Account/secret class/certificate/license + rotate/revoke:
Schema/backup/restore/RTO/RPO/retention/export:
Dependency failure/expiry/reconcile/escalation/exit:
Observability/incident/support/supplier evidence:
Maintenance type/backlog/patch/change/release/calendar:
Teach-back/reverse shadow/independent task/known issue:
Gate result + residual risk/gap/TBR + exit + M13-05 input:
```

26. Handover Maintenance Package 템플릿

별도 템플릿 `03_Templates/T13-04_handover-maintenance-package.md`를 사용합니다. 15개 section은 metadata/authority·source/scope·RACI·asset/config·source/build/release·access/credential/license·data/recovery·dependency/integration·observability/incident·support/supplier·maintenance backlog/change·knowledge transfer·calendar/risk/TBR·exit/decommission·Gate/M13-05입니다.

템플릿 영역	먼저 채울 최소 열	빈칸 처리
Scope/authority	source·owner·accepted question	미결정은 TBR+due+authority
Asset/build/access	owner·version·location·rehearsal	critical blank는 Gate fail
Data/dependency/ops	failure·restore·escalation·evidence	추측 대신 synthetic test
Maintenance/knowledge	type·priority·calendar·teach-back	hidden work는 backlog
Exit/Gate	export·revoke·successor·risk/gap/TBR	실제 transfer 권한 분리

27. 공식 자료와 적용 경계

아래 자료는 인수인계·유지보수 설계의 **적용 가능성 참고용 일차 자료**입니다. 목록에 있다는 이유만으로 특정 조직·계약·서비스에 자동 적용되거나 적합·인증·승인되는 것은 아닙니다. 확인 기준일은 2026-07-16이며 실제 적용·최신판·계약·법률 판단은 조직과 자격 있는 전문가가 재확인해야 합니다.

공식 자료	이 장에서 확인할 원리	현재판·적용 경계
ISO/IEC/IEEE 14764:2022	software maintenance process와 disposal	current; backup·system administration 같은 operations 전체를 직접 다루지는 않음
ISO/IEC/IEEE 12207:2026	acquisition부터 operation·maintenance·disposal까지 lifecycle	2026-04 published; 조직·계약 context에 tailoring
ISO/IEC 20000-1:2018	service management system 요구사항	2023 confirmed current; 특정 SLA나 운영 승인을 자동 보장하지 않음
ISO/IEC 20000-2:2019	ISO 20000-1 적용 guidance	Amd 1:2020 포함; certification 범위와 실제 service evidence 분리
ISO 10007:2017	configuration management guidance	2023 confirmed current; 조직 baseline·authority 필요
ISO/IEC/IEEE 15289:2019	lifecycle information item과 documentation	2025 confirmed current; 모든 문서를 동일하게 만들라는 뜻 아님

공식 자료	이 장에서 확인할 원리	현재판·적용 경계
ISO/IEC 27002:2022	information security control guidance	적용 범위·risk assessment·조직 policy에 맞춘 선택 필요
NIST SP 800-40 Rev.4	enterprise patch management planning	guidance; 실제 patch deadline·change authority는 조직이 결정
NIST SP 800-61 Rev.3	cybersecurity risk management과 incident response 통합	2025-04 final; Rev.2 superseded
NIST SP 800-53 Release 5.2.0	patch·update·reliability를 포함한 control catalog	2025 release; 조직 context에 tailoring
NIST SP 800-128	security-focused configuration management	2019 update; 조직 baseline과 change process 필요
NIST SP 800-34 Rev.1	contingency planning·recovery	업무 영향·RTO/RPO·조직 절차와 함께 적용
NIST SP 800-57 Part 2 Rev.1	key management organization practice	current final under review; latest status와 Part 1 current final 재확인
NIST SP 800-88 Rev.2	media sanitization과 disposal	2025-09 final; media·data classification·policy에 맞춤
NIST SP 800-137	continuous monitoring strategy	system·organization risk context에 맞춘 metric·frequency 필요
NIST Log Management Project	log generation·storage·analysis·disposal	SP 800-92 Rev.1은 draft 상태; current final과 project status 재확인
NIST SP 800-126 Rev.4	SCAP 기반 security automation content	2026-06 final; 모든 자산·취약점 처리를 자동 대체하지 않음
행정기관 및 공공기관 정보시스템 구축·운영 지침	공공 정보시스템 구축·운영·산출물 맥락	2025-01-02 시행판; 적용 기관·사업과 최신 개정 재확인
소프트웨어사업 계약 및 관리감독 지침	software 사업 계약·관리감독의 공공 범위	2023-05-15 자료; 현재 효력·적용 범위 반드시 재확인
정보시스템 감리기준	공공 정보시스템 감리·검수 참고	공공 적용 대상·최신판·사업 범위 별도 확인

28. 셀프 테스트 30

1. 인수인계가 파일 전달과 다른 이유는 무엇인가요?

정답 보기
정답: 파일 존재는 입력일 뿐입니다. 수신 팀이 source·build·배포·복구·장애 대응·유지보수를 설명하고 독립 실행한 evidence가 있어야 운영 능력이 이동합니다.

2. handover_acceptance_package_ready가 뜻하지 않는 것을 세 가지 쓰세요.

정답 보기
정답: 실제 서비스 이전, credential·secret 전달, 유지보수 계약·검수, production acceptance·배포·release·가용성 보장을 뜻하지 않습니다.

3. 합성 사례가 인수하는 이전 산출물은 몇 개인가요?

정답 보기
정답: M12와 M13-01~03에서 이어진 46개 linked artifact이며 orphan과 conflict는 각각 0입니다.

4. HND-A와 HND-C의 가장 큰 차이는 무엇인가요?

정답 보기
정답: HND-A는 파일 존재를 확인하지만 HND-C는 manifest·rehearsal·teach-back·reverse shadow·Gate로 수신 팀의 독립 수행을 증명합니다.

5. 인계 범위와 실제 서비스 이전 권한을 분리해야 하는 이유는 무엇인가요?

정답 보기
정답: 교육 evidence나 검토 결과가 계정·자산·책임·계약을 자동 이전하지 않기 때문입니다. 실제 이전은 조직의 승인 절차와 권한자가 결정합니다.

6. RACI에서 accountable owner를 한 명으로 명확히 하는 이유는 무엇인가요?

정답 보기
정답: 결정과 결과의 최종 설명 책임이 분산되거나 서로 미뤄지는 것을 막기 위해서입니다.

7. Asset manifest에 최소 어떤 필드를 기록하나요?

정답 보기
정답: 자산 ID·종류·owner·location·version·checksum/provenance·환경·중요도·freshness·상태·후임을 기록합니다.

8. Configuration baseline과 실제 환경 차이를 왜 검토하나요?

정답 보기
정답: 문서상 설정과 실제 runtime 차이가 hidden step·보안 공백·배포 실패·복구 실패를 만들 수 있기 때문입니다.

9. Clean build가 강한 인계 evidence인 이유는 무엇인가요?

정답 보기
정답: 기존 담당자의 로컬 상태나 기억에 기대지 않고 tag·lock·환경만으로 같은 artifact를 만들 수 있음을 수신 팀이 증명하기 때문입니다.

10. Artifact hash와 provenance가 각각 답하는 질문은 무엇인가요?

정답 보기
정답: Hash는 artifact가 같은지, provenance는 어떤 source·dependency·builder·절차로 만들어졌는지를 답합니다.

11. 교육 실습에서 실제 secret 값을 다루지 않는 이유는 무엇인가요?

정답 보기
정답: 학습에 원문 credential이 필요하지 않고 노출 위험만 키우기 때문입니다. 종류·owner·rotation·expiry·revocation metadata만 사용합니다.

12. Orphan account와 orphan critical asset의 공통 위험은 무엇인가요?

정답 보기

정답: 유효한 책임자가 없어 접근 회수·변경·복구·갱신·종료 결정을 제때 수행할 수 없다는 점입니다.

13. Backup 존재와 restore readiness의 차이는 무엇인가요?

정답 보기

정답: Backup 존재는 copy가 있다는 주장이고 restore readiness는 무결성을 확인한 copy로 수신 팀이 목표 RTO·RPO 안에 복구하고 reconcile한 evidence입니다.

14. RTO와 RPO를 설명하세요.

정답 보기

정답: RTO는 중단 뒤 서비스를 복구해야 하는 목표 시간이고 RPO는 복구 시 허용 가능한 데이터 손실 시점입니다.

15. Integration dependency에 expiry와 exit를 함께 적는 이유는 무엇인가요?

정답 보기

정답: Certificate·token·계약·API version 만료는 장애 trigger이고, 대체·종료 경로가 없으면 공급자와 기술 종속이 지속되기 때문입니다.

16. Idempotency와 reconciliation이 각각 줄이는 위험은 무엇인가요?

정답 보기

정답: Idempotency는 재시도의 중복 side effect를 줄이고 reconciliation은 두 시스템 결과가 최종적으로 일치하는지 확인합니다.

17. 관측 가능성 신호가 actionable하려면 무엇과 연결돼야 하나요?

정답 보기

정답: Log·metric·trace·dashboard·alert가 severity·on-call·runbook·escalation·communication·restore·postmortem owner와 연결돼야 합니다.

18. Incident management와 problem management의 차이는 무엇인가요?

정답 보기

정답: Incident management는 영향을 줄이고 서비스를 빨리 복구하며, problem management는 반복 incident의 근본 원인과 영구 개선을 다룹니다.

19. Service desk와 supplier escalation 경계를 왜 문서화하나요?

정답 보기

정답: 사용자 접수·내부 진단·보증·공급자 대응·계약 authority가 섞이면 지연과 책임 공백이 생기기 때문입니다.

20. 네 가지 유지보수 유형을 쓰세요.

정답 보기

정답: Corrective(결함 수정), adaptive(환경 적응), perfective(성능·사용성·가치 개선), preventive(미래 결함·장애 예방)입니다.

21. Patch management가 설치 작업 하나가 아닌 이유는 무엇인가요?

정답 보기

정답: 자산 식별·취약점 평가·우선순위·호환성 test·change authority·배포·rollback·검증·baseline 갱신이 이어지는 주기이기 때문입니다.

22. Critical patch 7일은 무엇을 뜻하나요?

정답 보기

정답: 이 합성 사례의 학습용 signal입니다. 실제 기한은 취약점 심각도·악용 가능성·노출·영향·보완 통제·조직 policy로 정합니다.

23. Known issue와 technical debt를 숨기지 않아야 하는 이유는 무엇인가요?

정답 보기

정답: 수신 팀이 유지보수 비용·위험·우선순위·우회책을 모르면 같은 장애와 예상 밖 변경 비용을 반복하기 때문입니다.

24. Walkthrough와 teach-back의 차이는 무엇인가요?

정답 보기

정답: Walkthrough는 주는 팀이 설명하고, teach-back은 받는 팀이 구조·절차·위험을 자기 말로 재설명해 이해를 증명합니다.

25. Reverse shadow가 shadow보다 강한 evidence인 이유는 무엇인가요?

정답 보기

정답: 받는 팀이 직접 수행하고 주는 팀이 관찰하므로 암묵지 공백과 독립 실행 능력을 실제 행동에서 확인할 수 있기 때문입니다.

26. 유지보수 달력에 넣을 반복 항목을 네 가지 쓰세요.

정답 보기

정답: Patch·취약점, access·secret·certificate·license review, backup restore·DR rehearsal, capacity·cost·supplier·architecture review 등이 있습니다.

27. Exit plan에 data export 외에 무엇이 필요한가요?

정답 보기
정답: 계정·secret 회수, integration 해제, 데이터 삭제·잔존 결정, license 종료, decommission, 후임·successor owner와 evidence가 필요합니다.

28. 세 handover version의 pass 수와 판정을 쓰세요.

정답 보기
정답: document-dump-v1은 6/24·blocked_document_dump_without_operability, shadow-only-v2는 15/24·blocked_receiver_independence_gaps, evidence-led-handover-v3는 24/24·handover_acceptance_package_ready입니다.

29. 최종 Handover Acceptance Gate의 핵심 수치를 쓰세요.

정답 보기
정답: 24/24 scenario, critical 100%, 12/12 controls, 4/4 lanes, 12 coverage domain 100%, authority conflict·orphan critical asset·critical risk·critical gap 0, TBR 2 이하입니다.

30. M13-05에 넘기는 핵심은 무엇인가요?

정답 보기
정답: 기술 evidence를 고객 가치·사업 성과·비용·위험·운영 가능성으로 설명할 수 있도록 verified capability·constraint·risk·owner·proof를 구조화해 넘깁니다.

29. 용어집 300 학습 순서

`04_Glossary/GLOSSARY_handover_maintenance.md`에는 15개 묶음·300개 unique term이 있습니다. M13-03의 조직 운영 개념을 누적하고, 유지보수·patch·기술 부채·business narrative input을 추가했습니다.

묶음	핵심 질문
01~03	조직 맥락·authority·asset inventory를 설명하는가

묶음	핵심 질문
04~07	Identity·access·network·사용 조건을 인계하는가
08~10	Data·privacy·security·risk lifecycle을 구분하는가
11~13	Integration·observability·SLO·restore evidence를 연결하는가
14~15	Maintenance·exception·exit·Gate·M13-05를 설명하는가

30. 다음 단계 · 기술을 고객 가치와 사업 언어로 바꾸기

M13-05에서는 이 인수인계 evidence를 1쪽 사업 설명서로 바꿉니다. 기술 항목을 그대로 나열하지 않고 고객 문제·제공 가치·운영 가능성·비용·위험·차별점·측정 가능한 outcome으로 번역합니다.

```
M12 product·requirement·document package
→ M13-01 architecture context·quality scenarios·ADR
→ M13-02 vendor·acceptance·delivery evidence
→ M13-03 organization readiness conditions
→ M13-04 ownership·asset·operability·maintenance·knowledge·exit evidence
→ verified capability·constraint·cost·risk·proof
→ M13-05 one-page customer value and business narrative
```

기술 evidence	고객·사업 질문	M13-05 표현
Clean build·rollback	변경과 장애 위험을 얼마나 줄이나	예측 가능한 배포·복구 능력
RTO/RPO·incident	업무 중단 영향을 어떻게 제한하나	복구 가능성과 대응 속도
Patch·maintenance calendar	품질과 보안을 지속할 수 있나	지속 운영 비용과 책임
Teach-back·independent task	특정 사람·공급자 종속을 줄였나	내부 운영 자립도
Risk·gap·TBR·exit	무엇을 아직 모르고 어떻게 빠져나오나	투명한 위험·선택권·종료 가능성

마지막 확인: 좋은 인수인계서는 문서가 많은 자료가 아닙니다. 새로운 팀이 서비스의 구조와 위험을 설명하고, build·restore·incident·maintenance를 독립 수행하며, 모르는 것과 실제 결정 권한을 분명히 말할 수 있게 하는 자료입니다.