

M12-04 · GIBALJA VISUAL MANUAL

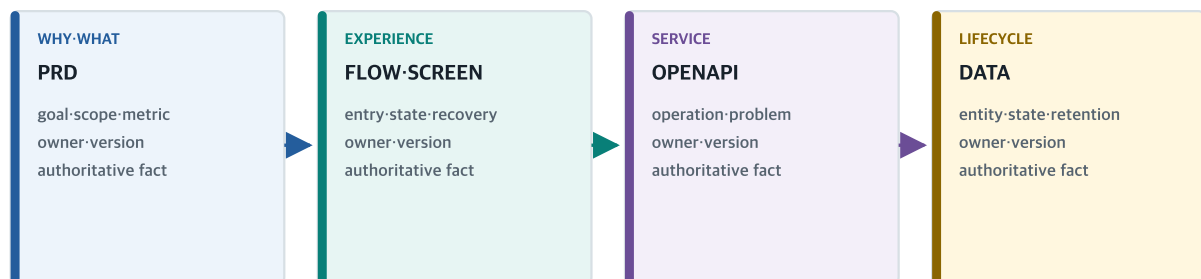
PRD·화면·API·데이터 문서 연결하기

한 문장 목표: M12-03 goal·scope·quality·metric → authoritative fact + canonical ID → PRD → flow·screen·UI state → API operation·problem → data entity·lifecycle → acceptance·test evidence → orphan·conflict·change impact → Document Linkage Gate → M13-01 handoff 를 끊김 없이 연결합니다.

M12-04 · 01

연결은 같은 내용을 네 번 복사하는 일이 아닙니다

Fact마다 authoritative location을 하나 정하고 다른 문서는 canonical ID와 version으로 참조합니다.

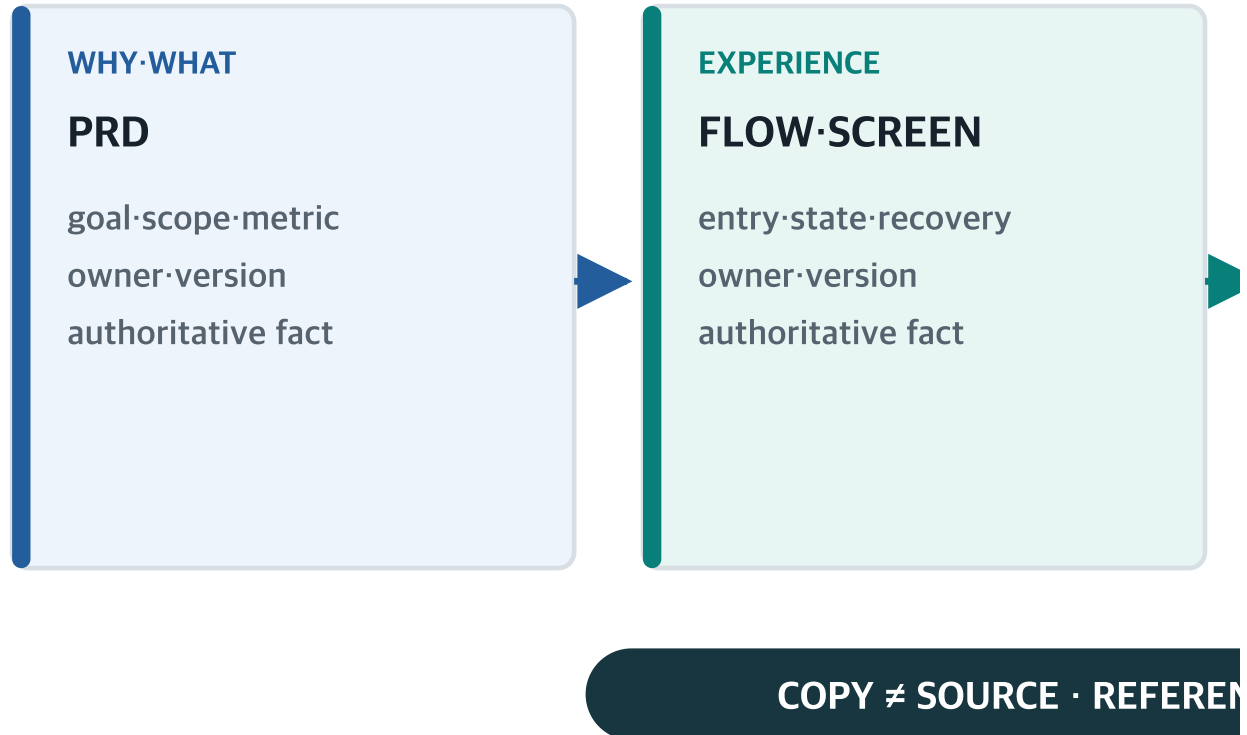


COPY ≠ SOURCE · REFERENCE + SNAPSHOT VERSION

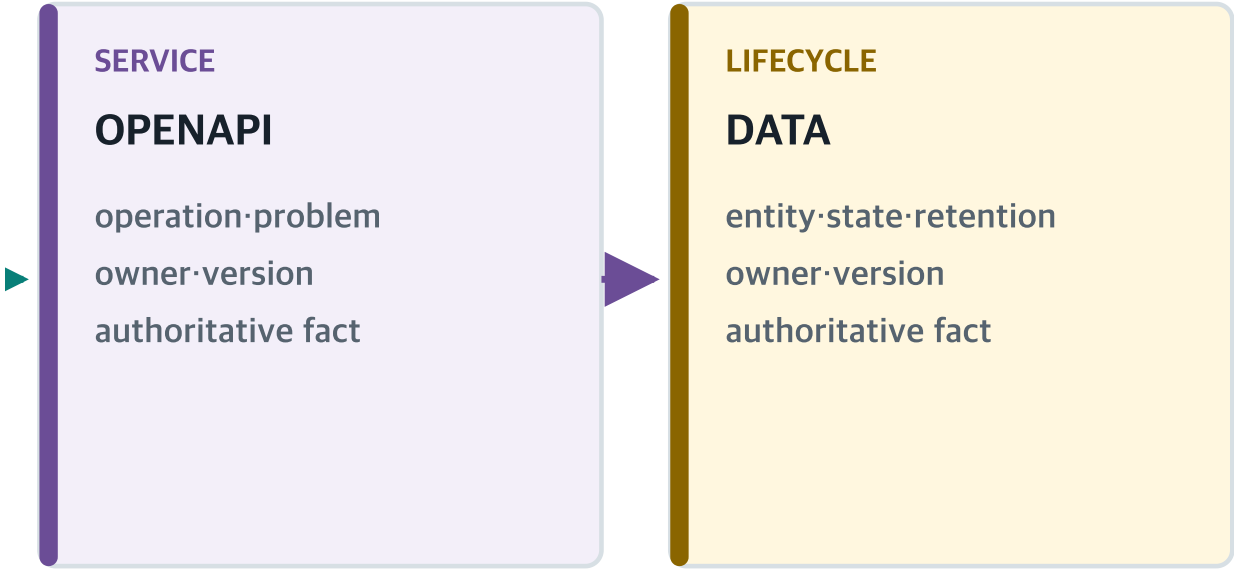
Source of truth를 하나로 정해야 drift와 승인 착각을 줄일 수 있습니다.

그림 1. 문서 연결은 같은 정보를 반복 작성하는 것이 아니라 authority를 나누고 ID·version으로 참조하는 작업입니다.

Fact마다 authoritative location을 하나 정하고 다른 문서는 canonical ID



과 version으로 참조합니다.



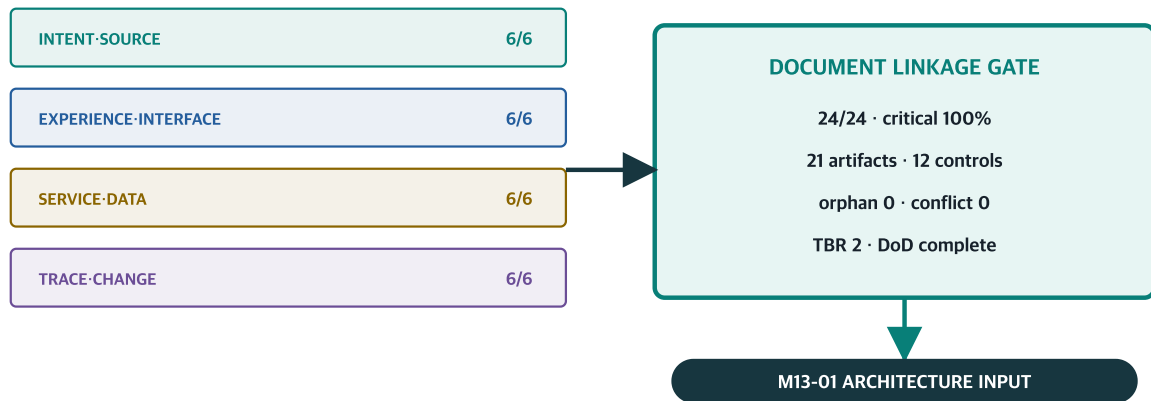
NCE + SNAPSHOT VERSION

0. 한눈에 보는 Document Linkage Gate

M12-04 · 14

네 lane이 모두 통과해야 architecture 입력 후보가 됩니다

24 scenario·12 control과 canonical·semantic·UI·API·data·acceptance·change coverage, orphan·conflict·TBR를 함께 봅니다.

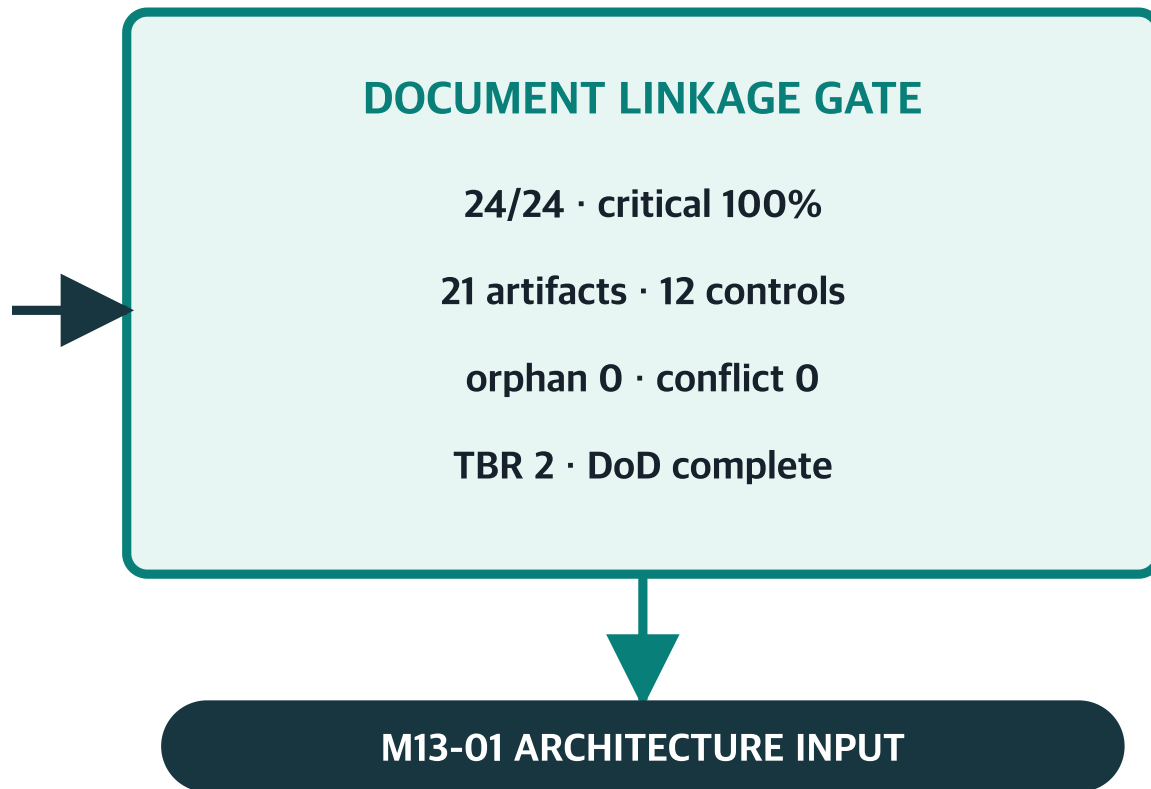


document_linkage_ready는 M13-01 선택 입력이며 실제 architecture·build·release 승인이 아닙니다.

그림 14. 네 lane 6/6, 21 artifact, 12 control, orphan 0, conflict 0, TBR 2일 때 architecture 입력 후보가 됩니다.

INTENT·SOURCE	6/6
EXPERIENCE·INTERFACE	6/6
SERVICE·DATA	6/6
TRACE·CHANGE	6/6

change coverage, orphan·conflict·TBR를 함께 봅니다.



학습 순서	시간	남기는 evidence
그림 14장	45분	authority·trace·state·contract·Gate 전체 지도
개념·합성 사례	105분	21 artifact·8 state·7 operation·5 entity
실습 스튜디오	90분	24 scenario·57 audit·6 regression
템플릿·셀프 테스트	60분	내 linked package·30문항 답

본문의 사용자·조직·요구사항·문서·API·데이터·수치·시스템은 모두 허구의 합성 학습 자료입니다.

`document_linkage_ready` 는 승인된 PRD·디자인·API·데이터 계약·architecture·build·pilot·release·예산·투자·계약·보안·접근성 인증을 대신하지 않습니다.

0.1 먼저 기억할 열두 문장

연결은 복사가 아니라 authority와 reference의 계약이다.
 PRD는 왜·누구·무엇·어떤 결과를 정하고 endpoint·table 상세를 중복 source로 만들지 않는다.
 Canonical ID는 안정적으로, version은 변경 이력으로 관리한다.
 Goal에서 test로 가고 test에서 goal로 돌아온다.
 화면은 loading·empty·error·denied·conflict까지 포함한 상태 계약이다.
 같은 사건은 UI·API·data에서 같은 의미를 가져야 한다.
 API problem은 UI recovery action까지 이어진다.
 Data contract는 field뿐 아니라 identity·state·history·retention을 가진다.
 Vocabulary registry는 문서 사이의 의미 API다.
 Acceptance·test·evidence는 구분하고 같은 ID로 잇는다.
 Orphan과 conflict는 계산하고 변경 영향은 graph로 찾는다.
 document_linkage_ready는 architecture·build·release 승인이 아니다.

1. 이 PDF를 공부하는 방법

1.1 1회차 · 그림과 caption만 읽기 · 45분

그림마다 `source`는 어디인가 → 어떤 ID로 이어지는가 → 어떤 mismatch를 막는가 → 무엇을 승인하지 않는가 를 손으로 짚습니다.

이 그림의 연결이 없으면 ____ 문서가 orphan이 되고, ____ 사건에서 ____와 ____가 충돌한다.

1.2 2회차 · 합성 사례 실습 · 90분

문서 연결 실습 생성기를 실행합니다.

```
./02_Labs/G12_Product_Definition/L12-04_create-document-linkage-practice.sh
```

Version	Pass	Artifacts	Orphan	Conflict	Decision
copy-paste-docs-v1	6/24	18	12	9	blocked_document_drift
linked-incomplete-pack-v2	15/24	21	3	2	blocked_contract_gaps
canonical-linked-pack-v3	24/24	21	0	0	document_linkage_ready

1.3 3회차 · 내 linked package 만들기 · 120분

문서 연결 package 템플릿을 source
registry→PRD→flow/screen/state→operation/problem→entity/lifecycle→acceptance/evidence→change/Gate
순서로 채웁니다.

1.4 막힐 때

문서 연결 용어집 300에서 지금 그림과 같은 묶음의 20개만 읽습니다. 용어마다 PRD·FLOW·SCR·OP·ENT·AC·SC·CTRL ID 중 하나를 붙입니다.

2. 연결된 문서 package를 다시 정의하기

LINKED DOCUMENT CONTRACTProduct
intent·experience·service·data·lifecycle·acceptance·change fact가 각자의 authoritative location과 stable ID·owner·status·version을 가지고 양방향 edge로 연결되며, orphan·missing·conflict·TBR와 residual risk를 계산할 수 있는 합성 architecture 입력 후보입니다.

2.1 합성 case contract

Case: team-follow-up-document-pack-2026-07-v1

M12-03 handoff: team-follow-up-mvp-scope-2026-07-v1

Product goal: 팀이 회의에서 결정한 후속 조치를 빠짐없이 책임·기한과 함께 실행한다

PRD: PRD-TFU-001 · team-follow-up-prd-0.1.0

UI / API / Data: team-follow-up-ui-0.1.0 / team-follow-up-api-0.1.0 / team-follow-up-data-0.1.0

OpenAPI / JSON Schema / WAI-ARIA: 3.2.0 / 2020-12 / 1.2

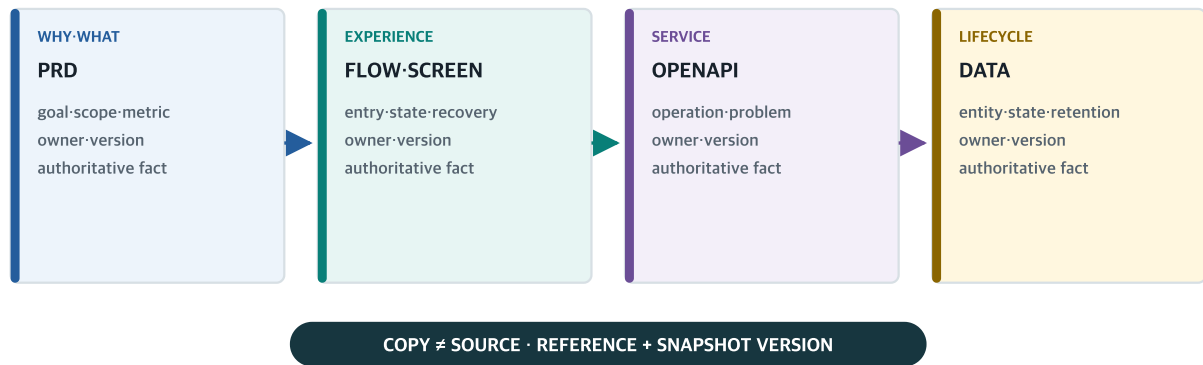
중요: 본 매뉴얼의 package는 실제 사람·요구사항·API·database·production resource를 사용하지 않습니다. 합성 Gate를 통과해도 실제 architecture·build·pilot·release에는 별도의 보안·privacy·접근성·비용·운영·법률·권한 검토가 필요합니다.

3. Fact의 authoritative location 정하기

M12-04 · 01

연결은 같은 내용을 네 번 복사하는 일이 아닙니다

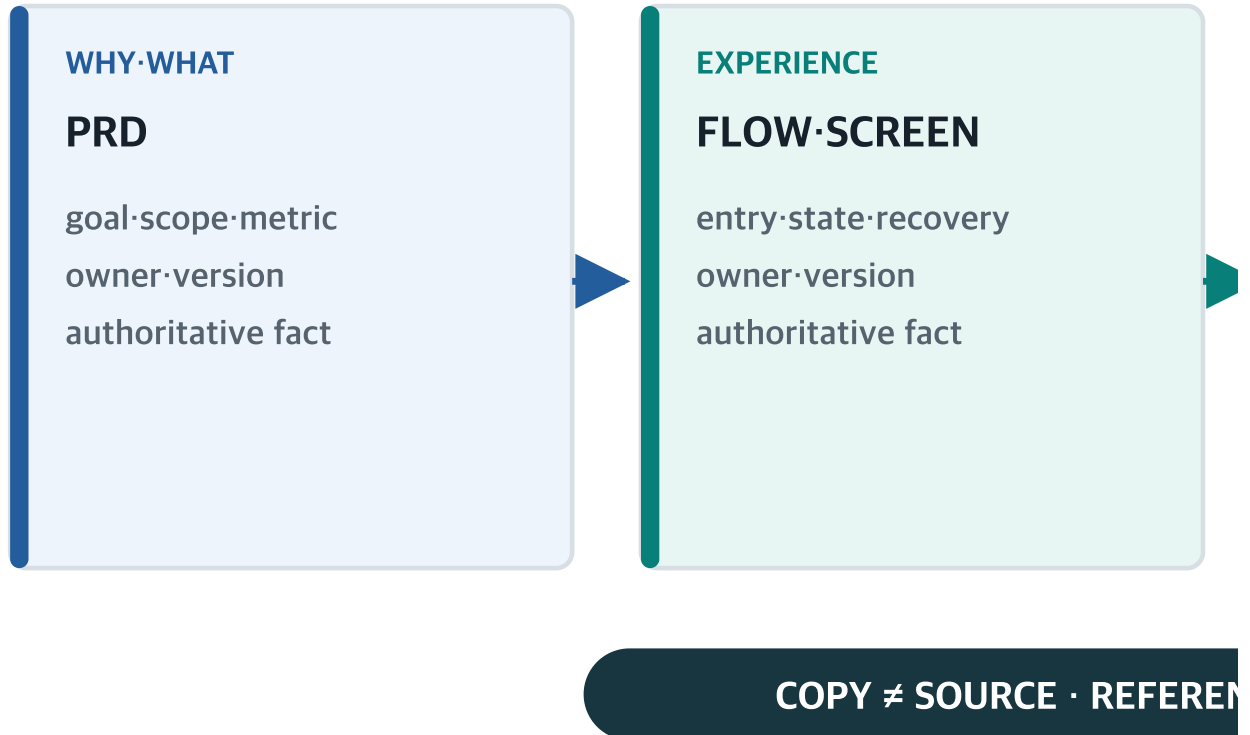
Fact마다 authoritative location을 하나 정하고 다른 문서는 canonical ID와 version으로 참조합니다.



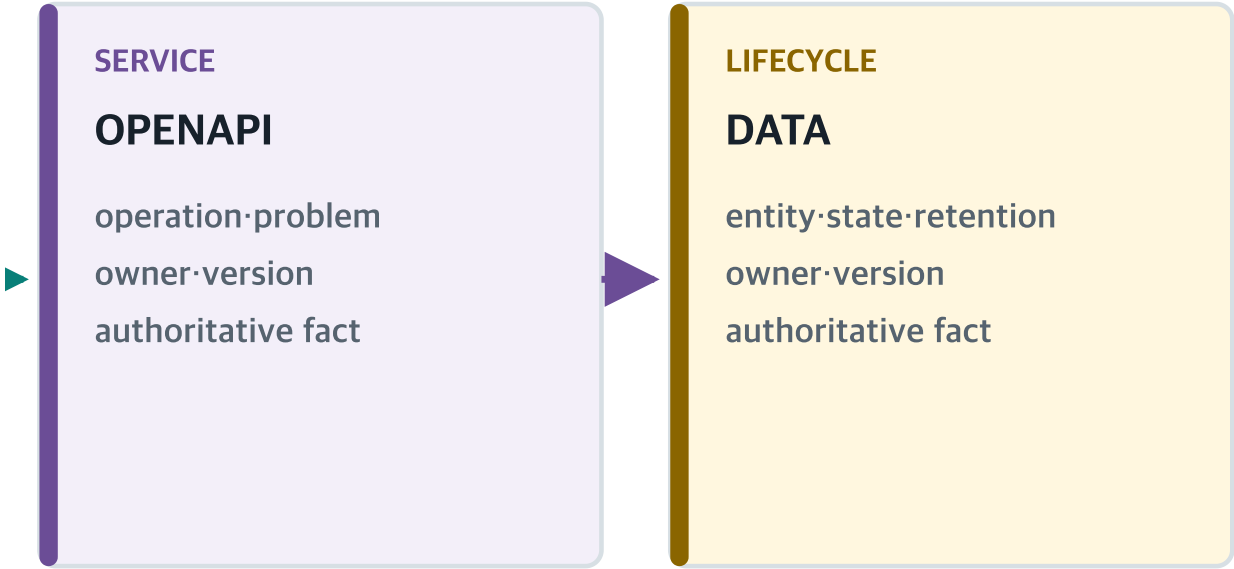
Source of truth를 하나로 정해야 drift와 승인 착각을 줄일 수 있습니다.

그림 1. 같은 내용을 반복 입력하지 않고 PRD·experience·service·data가 맡을 fact를 정한 뒤 reference와 version으로 잇습니다.

Fact마다 authoritative location을 하나 정하고 다른 문서는 canonical ID



과 version으로 참조합니다.



NCE + SNAPSHOT VERSION

CORE IDEA 연결의 시작은 복사가 아니라 authority 배분입니다.

PRD와 화면 문서와 OpenAPI와 데이터 사전에 같은 field 설명을 복사하면 네 곳이 모두 맞는 동안만 시스템이 맞습니다. 변경 한 번이면 최신값이 갈립니다. 합성 package는 목표·범위는 PRD, interaction은 flow·screen, HTTP behavior는 OpenAPI, lifecycle은 data contract를 source로 둡니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
PRD	user·problem·goal·scope·metric	PRD-TFU-001
Flow·screen	entry·exit·state·interaction	FLOW-01-SCR-03
OpenAPI	operation·request·response·problem	OP-02
Data	entity·field·state·history	ENT-03
Reference	canonical ID·snapshot version	override 금지

틀린 예와 고친 예

틀린 예: Due date의 의미와 validation을 PRD·화면·API·DB 문서에 똑같이 복사합니다.

고친 예: Meaning·unit은 vocabulary registry, server validation은 API·schema, display와 recovery는 screen state에 두고 canonical field ID로 있습니다.

5분 연습

1. Fact 열 개를 적습니다.
2. 각 fact의 authoritative document를 하나 고릅니다.
3. Owner·status·version을 붙입니다.
4. 다른 문서의 copy를 reference로 바꿉니다.

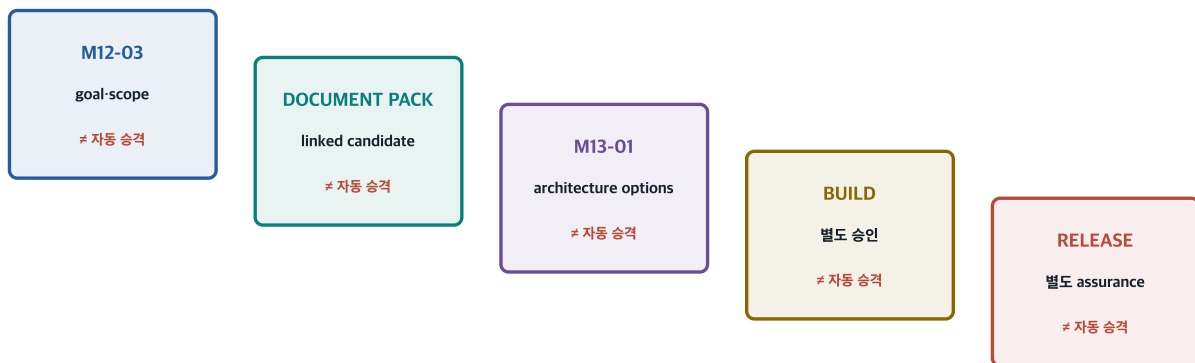
체크포인트: 연결의 시작은 복사가 아니라 authority 배분입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

4. 문서 package와 승인 경계 세우기

M12-04 · 02

문서 연결 후보와 승인된 설계는 같은 것이 아닙니다

현재 package가 답하는 질문과 architecture·build·release에서 별도로 답할 질문을 분리합니다.



document_linkage_ready는 M13-01 입력 후보이지 실제 PRD·API·architecture 승인 증거가 아닙니다.

그림 2. Linked document candidate는 architecture 선택 입력이며 다음 단계의 승인과 운영 책임을 자동으로 얻지 않습니다.

현재 package가 답하는 질문과 architecture·build·release에서 별도로 답

M12-03

goal·scope

≠ 자동 승격

DOCUMENT PACK

linked candidate

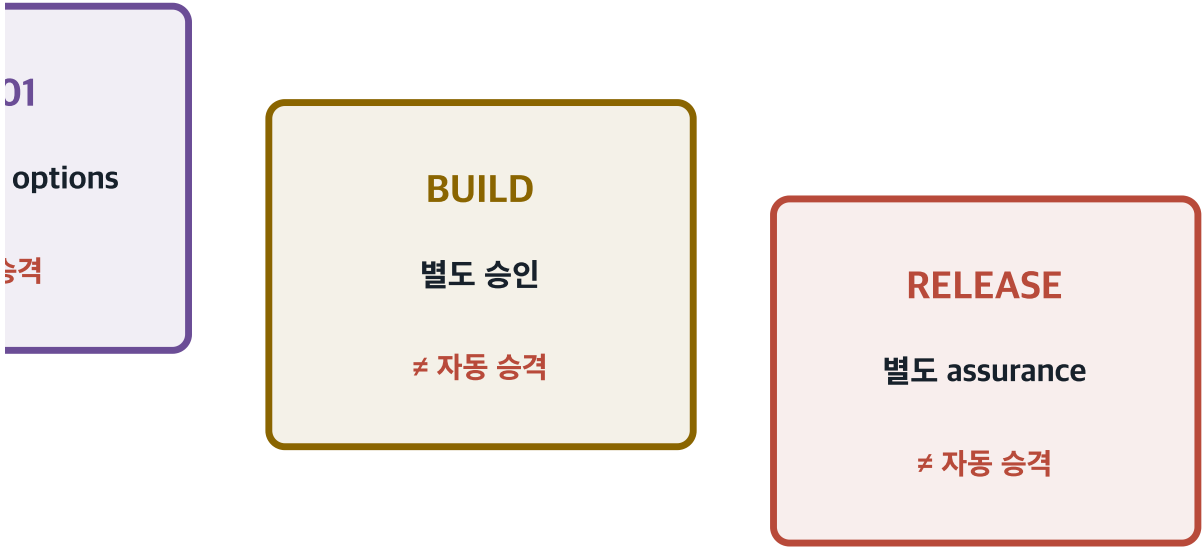
≠ 자동 승격

M13-0

architecture

≠ 자동 승격

할 질문을 분리합니다.



CORE IDEA 좋은 문서와 승인된 제품은 다른 상태입니다.

문서가 서로 맞다는 것은 중요한 준비 증거지만 성능 capacity, infrastructure, threat, privacy, procurement, 운영 support를 자동으로 해결하지 않습니다. Package header에 status·purpose·answered question·unanswered question·approval boundary를 표시해야 독자가 후보를 승인본으로 오해하지 않습니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
M12-03	왜·무엇을 먼저 배울까	scope candidate
M12-04	문서 계약이 서로 맞는가	linked candidate
M13-01	어떤 architecture가 맞는가	option·trade-off
Build	구현·검증 가능한가	별도 authority
Release	지속 운영 가능한가	별도 assurance

틀린 예와 고친 예

틀린 예: 문서 링크가 모두 있으므로 architecture와 개발을 승인합니다.

고친 예: Document Linkage Gate가 답한 질문과 남은 architecture·security·cost·operation 질문을 분리해 다음 Gate 입력으로 넘깁니다.

5분 연습

1. 현재 package status를 씁니다.
2. 답한 질문 세 개를 씁니다.
3. 답하지 못한 질문 세 개를 씁니다.
4. 승인할 권한과 승인하지 않는 항목을 적습니다.

체크포인트: 좋은 문서와 승인된 제품은 다른 상태입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

5. Canonical ID registry 만들기

M12-04 · 03

Canonical ID는 문서 사이의 공용 손잡이입니다

이름이나 페이지 위치가 바뀌어도 같은 의미를 가리키도록 stable ID-owner-status-version을 등록합니다.

ID	TYPE	OWNER	VERSION	SOURCE
SCR-03	screen	design	0.1.0	FLOW-01
OP-02	operation	api	0.1.0	SCR-03
ENT-03	entity	data	0.1.0	OP-02
AC-02	acceptance	qa	0.1.0	SCR-03-OP-02

STABLE ID · STATUS · REVIEW · NO REUSE

ID는 version과 분리하고 삭제된 ID를 다른 의미로 재사용하지 않습니다.

그림 3. Stable ID를 중심으로 type-owner-status-version-source를 모아 문서 위치가 바뀌어도 trace를 유지합니다.

이름이나 페이지 위치가 바뀌어도 같은 의미를 가리키도록 stable ID·owner

ID	TYPE	OWNER
SCR-03	screen	design
OP-02	operation	api
ENT-03	entity	data
AC-02	acceptance	qa

STABLE ID · STATUS

er·status·version을 등록합니다.

VERSION	SOURCE
0.1.0	FLOW-01
0.1.0	SCR-03
0.1.0	OP-02
0.1.0	SCR-03·OP-02
REVIEW · NO REUSE	

CORE IDEA ID는 정체성이고 version은 변경 이력입니다.

‘확인 화면’처럼 제목으로 연결하면 이름이 바뀌거나 같은 제목이 생길 때 링크가 흔들립니다. **SCR-03**, **OP-02**, **ENT-03**, **AC-02** 처럼 namespace를 가진 ID를 사용하고 artifact의 version·status는 별도 필드로 둡니다. 삭제 ID는 tombstone으로 보존합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
SCR-03	Owner·due confirmation	experience owner
OP-02	confirmOwnerAndDue	API owner
ENT-03	AssignmentConfirmation	data owner
AC-02	confirm success·denied·conflict	QA owner
Registry	unique·resolvable·not reused	document owner

틀린 예와 고친 예

틀린 예: 화면 v2를 만들면서 ID를 SCR-08로 바꾸고 과거 link를 삭제합니다.

고친 예: 같은 screen 정체성이면 SCR-03을 유지하고 version을 올리며 change impact와 compatibility를 기록합니다.

5분 연습

1. Artifact type namespace를 정합니다.
2. 21개 artifact에 unique ID를 줍니다.
3. Owner·status·version·source를 채웁니다.
4. 중복·broken link·ID reuse를 검사합니다.

체크포인트: ID는 정체성이고 version은 변경 이력입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

6. Goal에서 test evidence까지 왕복 trace 만들기

M12-04 · 04

한 요구를 목표에서 테스트 증거까지 왕복 추적합니다

Goal→scope→screen→operation→entity→acceptance edge를 연결하고 test에서 source로도 되돌아갑니다.

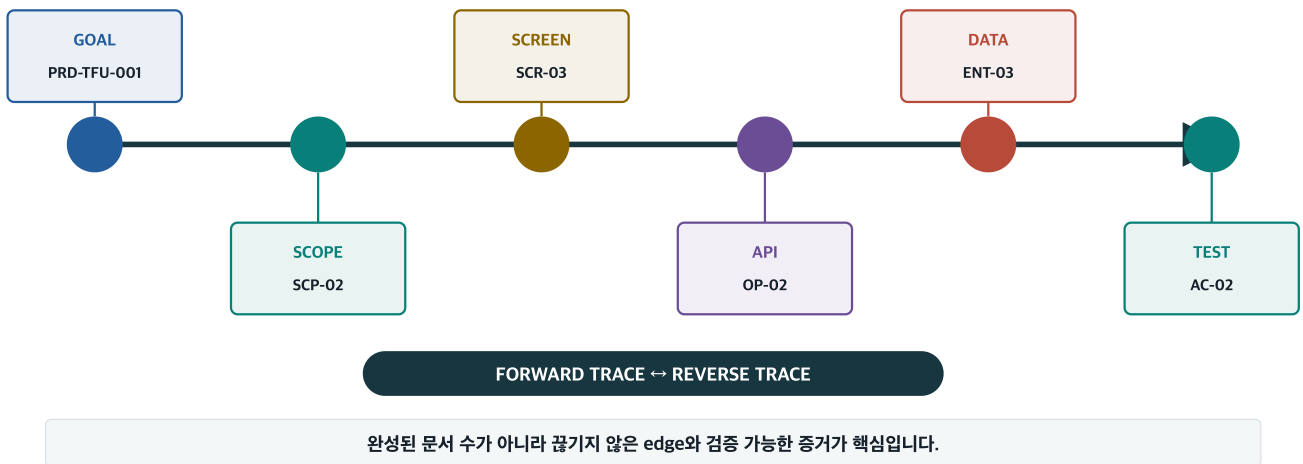
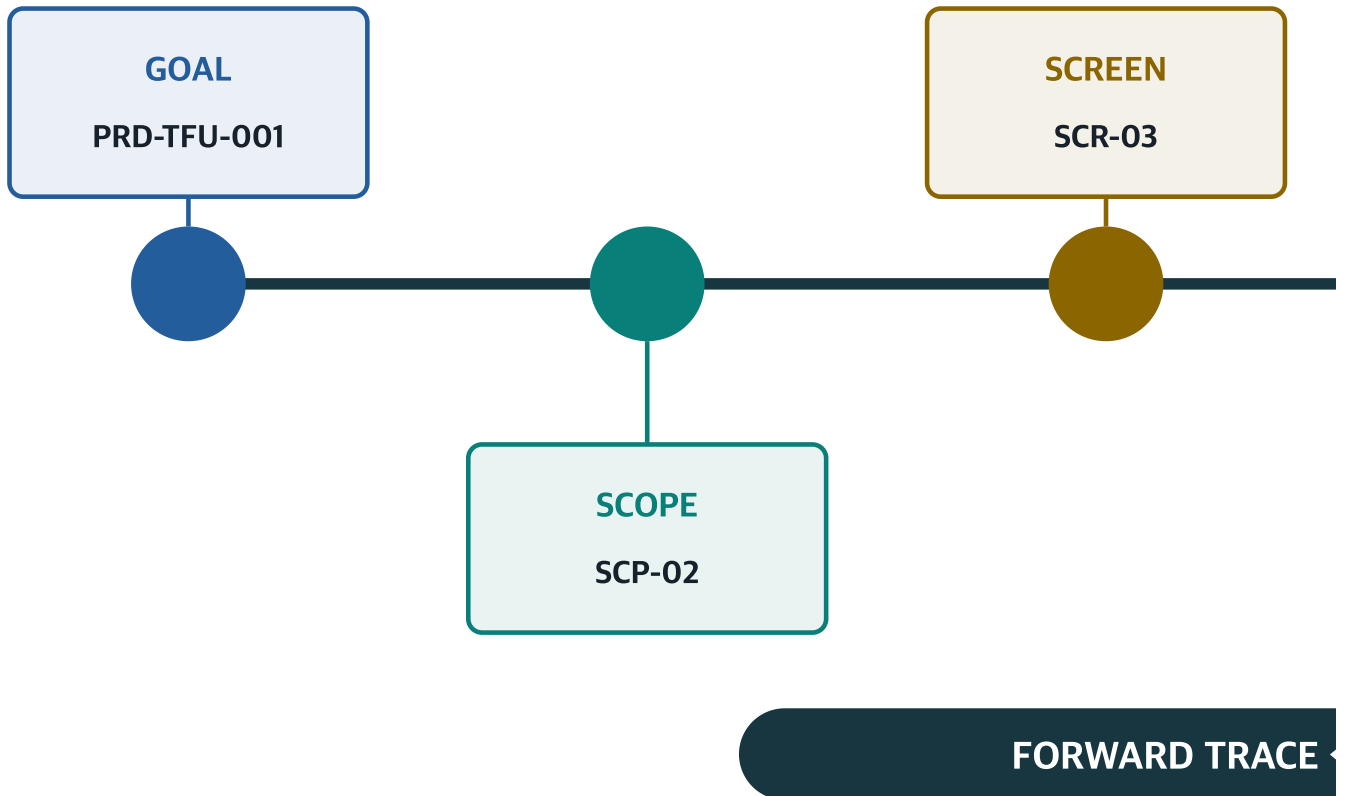


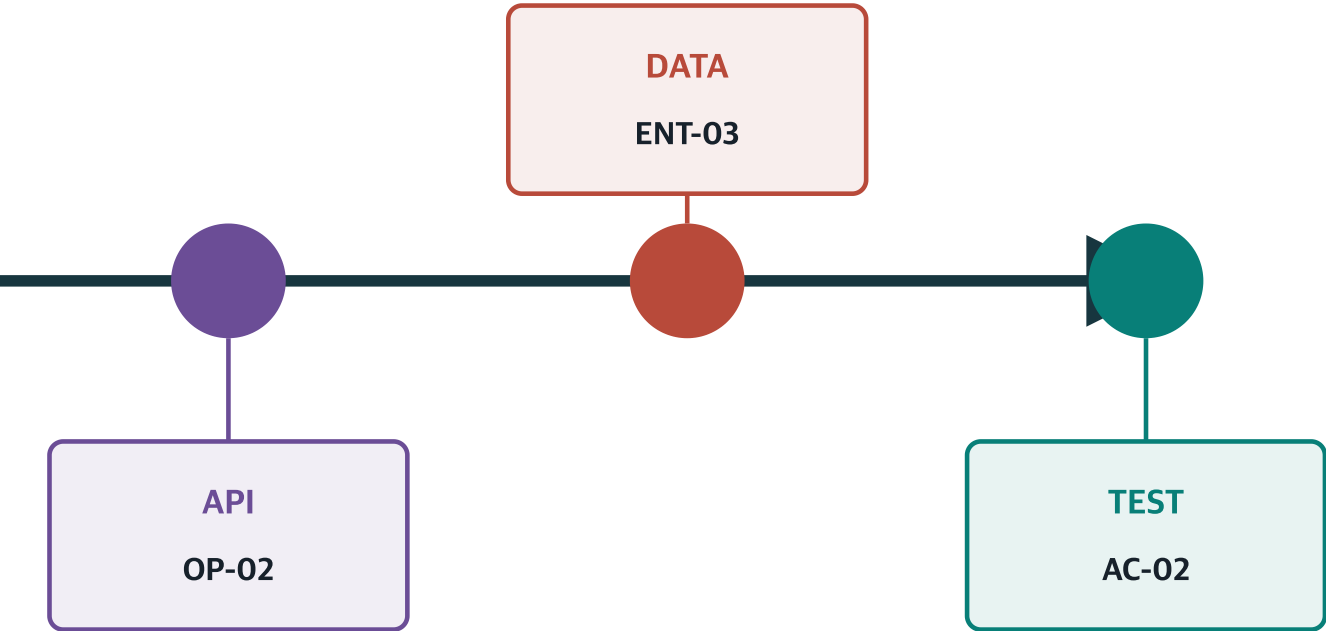
그림 4. Goal에서 acceptance로 내려가고 test에서 source decision으로 되돌아가는 두 방향이 모두 있어야 합니다.

Goal→scope→screen→operation→entity→acceptance edge를 연결하고





test에서 source로도 되돌아갑니다.



↔ REVERSE TRACE

CORE IDEA Trace는 목차가 아니라 검증 가능한 edge graph입니다.

Forward trace는 선택한 scope가 실제 screen·operation·entity로 구현 가능한지 보여 줍니다. Reverse trace는 test failure가 어떤 user outcome과 decision에 영향을 주는지 설명합니다. 합성 사례는 PRD-TFU-001 → SCP-02 → SCR-03 → OP-02 → ENT-03 → AC-02 를 하나의 chain으로 관리합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Goal	실행 가능한 기록	PRD-TFU-001
Scope	담당·기한 확인	SCP-02
Experience	확인 화면·states	SCR-03
Contract	PATCH·transition	OP-02·ENT-03
Evidence	positive·denied·conflict	AC-02·test run

틀린 예와 고친 예

틀린 예: PRD에 화면 링크를 한 번 붙였으니 trace가 끝났습니다.

고친 예: 각 edge의 type·source·sink·version·owner를 기록하고 reverse query와 orphan query를 실행합니다.

5분 연습

1. Scope 하나를 고릅니다.
2. Goal에서 screen·API·data·acceptance로 잇습니다.
3. Test에서 goal까지 역방향으로 읽습니다.
4. 끊긴 edge와 owner를 표시합니다.

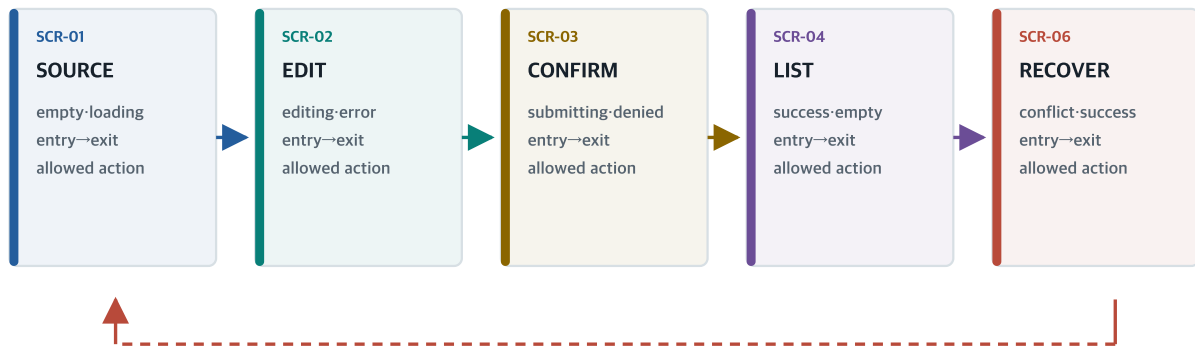
체크포인트: Trace는 목차가 아니라 검증 가능한 edge graph입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

7. Flow·screen·UI state를 한 지도에 놓기

M12-04 · 05

화면 목록보다 상태와 복구가 있는 사용자 흐름이 먼저입니다

각 screen은 purpose·entry·exit·UI state·trigger·visible result·allowed action·recovery를 가집니다.



Loading·empty·error·denied·conflict가 빠지면 정상 화면만 있는 거짓 계약이 됩니다.

그림 5. 화면 목적과 이동뿐 아니라 loading·empty·error·denied·conflict와 복귀 경로까지 flow에 넣습니다.

각 screen은 purpose·entry·exit·UI state·trigger·visible result·allowed :



action·recovery를 가집니다.



CORE IDEA 화면은 정지 이미지가 아니라 상태가 변하는 계약입니다.

Happy path 화면 여섯 장만 있으면 사용자는 network 지연, 빈 결과, invalid input, 권한 거부, 동시 수정에서 무엇을 해야 할지 모릅니다. Screen inventory는 purpose·entry·exit·trigger·state·visible result·allowed action·recovery를 가져야 합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Loading	source 또는 record 조회 중	progress·cancel boundary
Empty	기록 없음	create action
Editing	draft 변경	validation hint
Denied	권한 없음	safe exit·request path
Conflict	version 불일치	compare·preserve·retry

틀린 예와 고친 예

틀린 예: SCR-03에는 확인 버튼과 성공 화면만 정의합니다.

고친 예: Submitting·success·field error·denied·conflict state마다 trigger·announcement·focus·allowed action·recovery를 씁니다.

5분 연습

1. Flow start와 end를 적습니다.
2. Screen 여섯 개를 순서로 놓습니다.
3. 각 screen에 여덟 상태를 검토합니다.
4. 분기와 recovery가 flow로 돌아오는지 확인합니다.

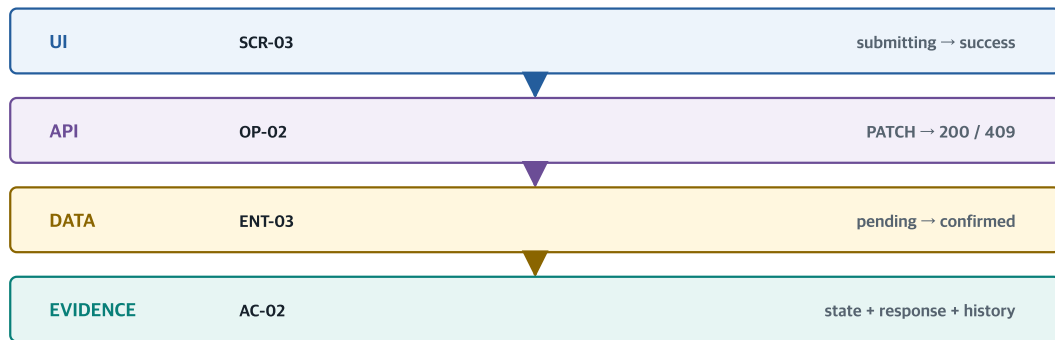
체크포인트: 화면은 정지 이미지가 아니라 상태가 변하는 계약입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

8. UI·API·데이터 상태 정합성 맞추기

M12-04 · 06

한 사건은 UI·API·데이터에서 같은 의미를 가져야 합니다

사용자 행동·HTTP 결과·entity transition·acceptance evidence를 한 줄로 맞춰 contract mismatch를 찾습니다.

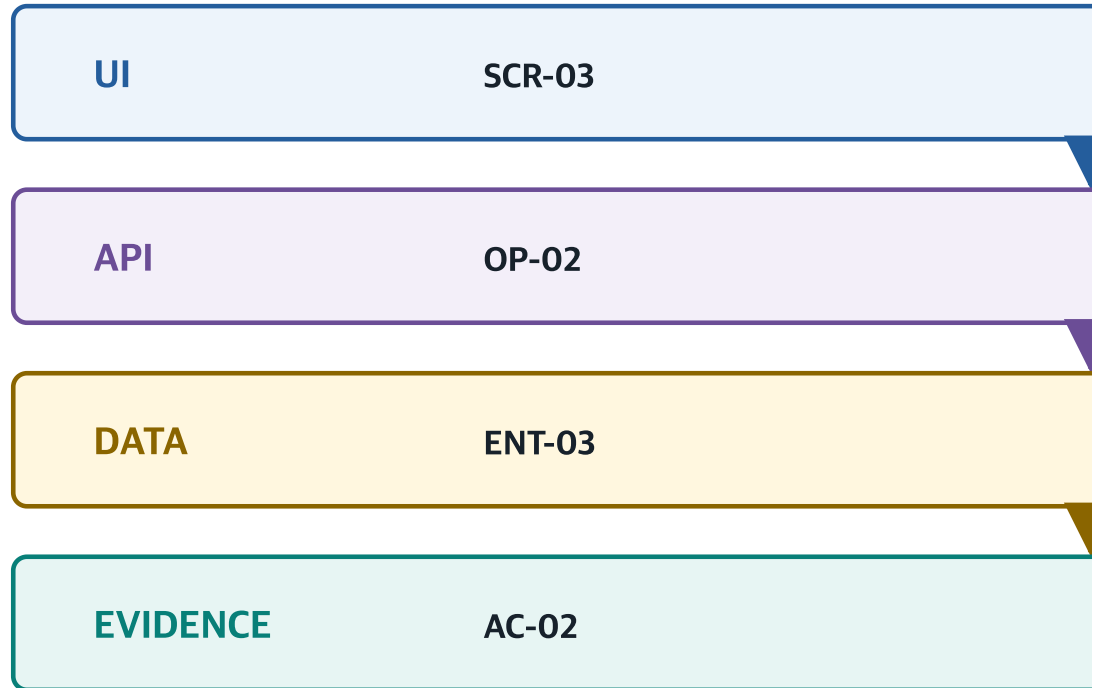


SAME EVENT · SAME MEANING · SAME RECOVERY

클라이언트 성공이 데이터 confirmed와 다르면 연결이 아니라 충돌입니다.

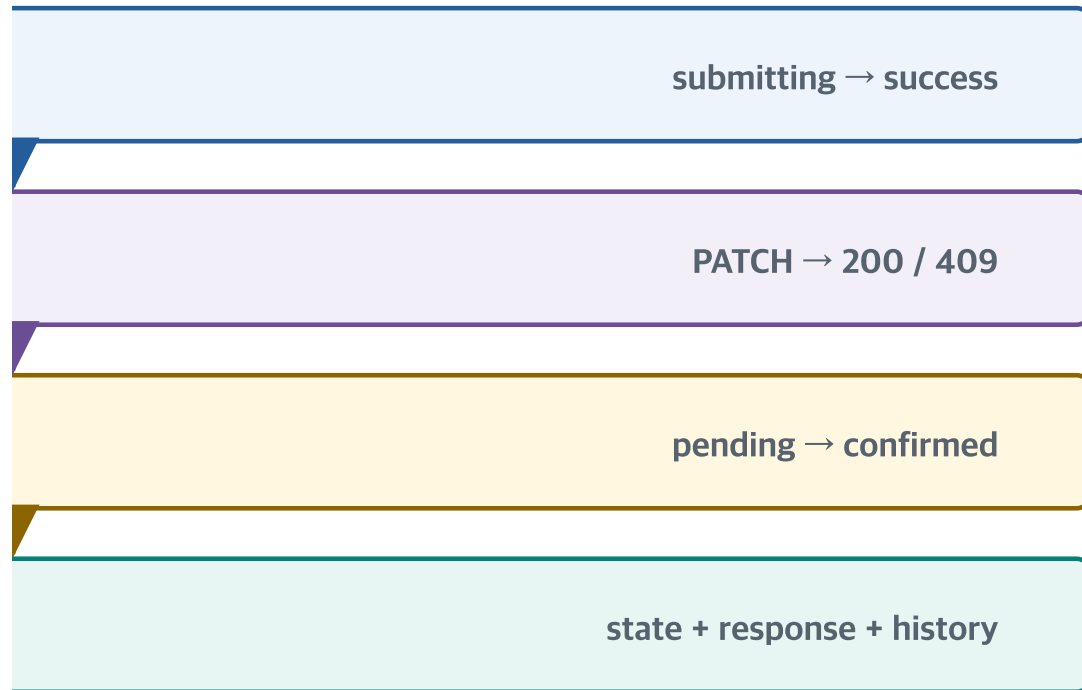
그림 6. 한 confirm 사건이 UI state·HTTP status·entity transition·evidence에서 같은 의미를 갖는지 비교합니다.

사용자 행동·HTTP 결과·entity transition·acceptance evidence를 한 줄로



SAME EVENT · SAME MEANING

크 맞춰 contract mismatch를 찾습니다.



ANING · SAME RECOVERY

CORE IDEA 같은 사건은 모든 layer에서 같은 결과와 복구를 약속해야 합니다.

UI가 success를 보여 주지만 API가 conflict를 반환하거나 데이터가 pending에 남으면 각 문서는 개별적으로 그럴듯해도 제품 계약은 실패합니다. Action ID와 operationId, response status와 UI state, entity transition과 acceptance를 한 행에 놓아 비교합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
UI	submitting→success 또는 conflict	SCR-03
API	PATCH→200 또는 409	OP-02
Data	pending→confirmed	ENT-03
History	actor·time·version	ENT-04
Evidence	visible state+response+transition	AC-02

틀린 예와 고친 예

틀린 예: 각 팀이 자기 문서만 검토하고 모두 완료라고 표시합니다.

고친 예: 같은 scenario ID로 UI·API·data expected를 나란히 검토하고 actual evidence가 모두 일치할 때만 pass합니다.

5분 연습

1. User action 하나를 고릅니다.
2. UI expected를 씁니다.
3. HTTP와 data transition을 맞춥니다.
4. Mismatch 하나를 일부러 넣고 detector로 찾습니다.

체크포인트: 같은 사건은 모든 layer에서 같은 결과와 복구를 약속해야 합니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

9. Problem detail에서 UI recovery까지 연결하기

M12-04 · 07

HTTP 오류는 UI 복구 행동까지 연결해야 완성됩니다

RFC 9457 problem type-status-field issue를 screen state-draft preservation-focus-next action과 맵핑합니다.

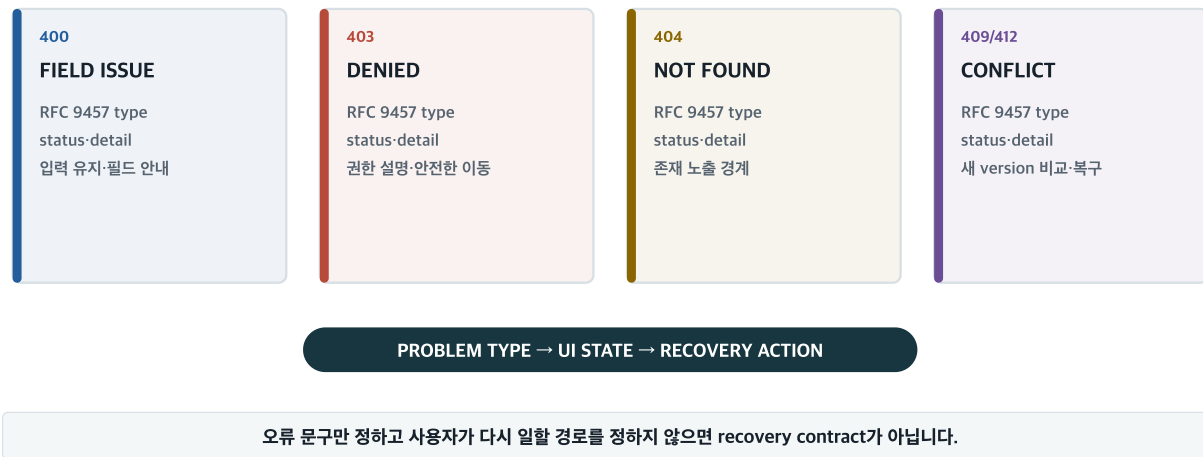


그림 7. API problem은 status 설명에서 끝나지 않고 입력 보존·focus·다음 행동까지 연결됩니다.

400

FIELD ISSUE

RFC 9457 type
status·detail
입력 유지·필드 안내

403

DENIED

RFC 9457 type
status·detail
권한 설명·안전한 이동

PROBLEM TYPE → UI STA

ration·focus·next action과 맵핑합니다.

404

NOT FOUND

RFC 9457 type
status·detail
존재 노출 경계

409/412

CONFLICT

RFC 9457 type
status·detail
새 version 비교·복구

TE → RECOVERY ACTION

CORE IDEA 오류 계약의 끝은 메시지가 아니라 다시 일할 수 있는 상태입니다.

RFC 9457 problem details는 machine-readable type과 HTTP status를 일관되게 전달하는 바탕이 됩니다. 그러나 UI 문서가 field issue·draft preservation·focus·allowed action을 연결하지 않으면 사용자는 복구하지 못합니다. 403과 404는 resource disclosure 경계도 함께 검토합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
400	invalid field	draft 유지·field focus
403	not authorized	안전한 이동·권한 경로
404	resource unavailable	노출 경계·return
409	state conflict	최신값 비교·merge
412	version precondition	refresh·preserve·retry

틀린 예와 고친 예

틀린 예: API 문서에 409가 있고 화면에는 ‘오류가 발생했습니다’라고 씁니다.

고친 예: Problem type을 conflict UI state와 연결하고 최신 version·내 draft·재시도 선택을 보존합니다.

5분 연습

1. Operation 하나의 problem 목록을 씁니다.
2. 각 problem type을 UI state로 잇습니다.
3. Draft·focus·announcement를 정합니다.
4. 사용자가 정상 flow로 돌아오는 exit를 확인합니다.

체크포인트: 오류 계약의 끝은 메시지가 아니라 다시 일할 수 있는 상태입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

10. 데이터 entity와 lifecycle 계약하기

M12-04 · 08

데이터 계약은 필드 표에서 생명주기로 확장됩니다

Identity·relationship·state transition·actor·time·reason·history·retention·provenance를 함께 연결합니다.

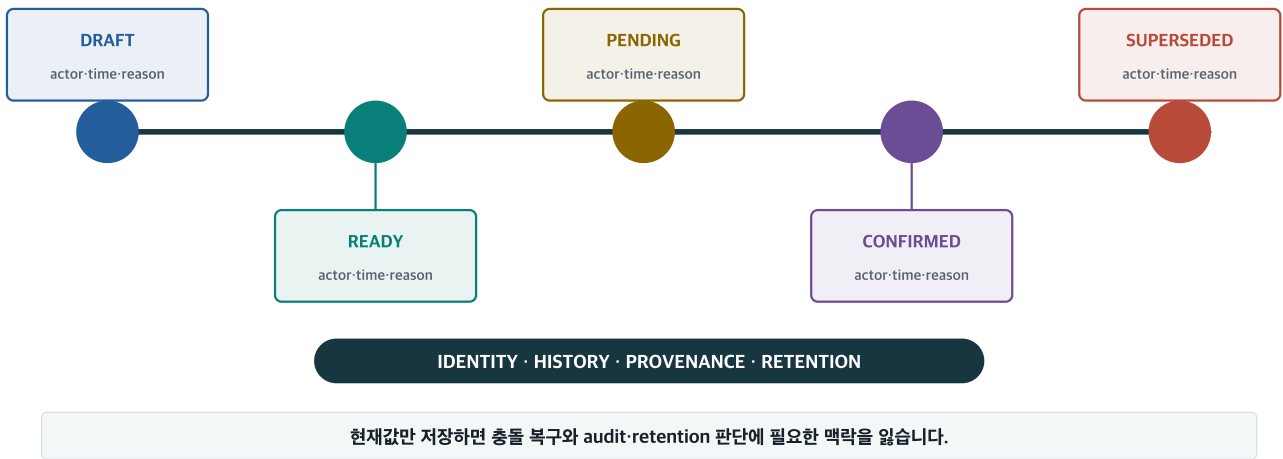
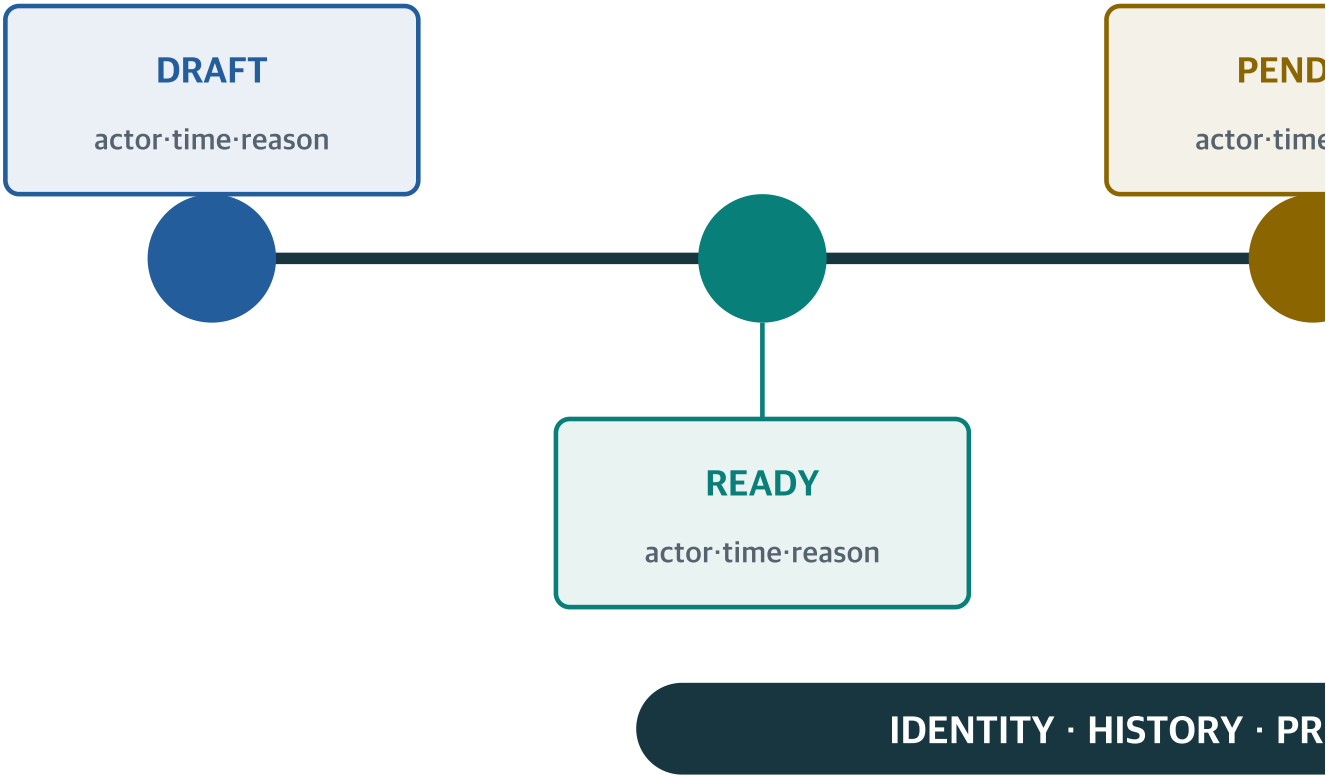
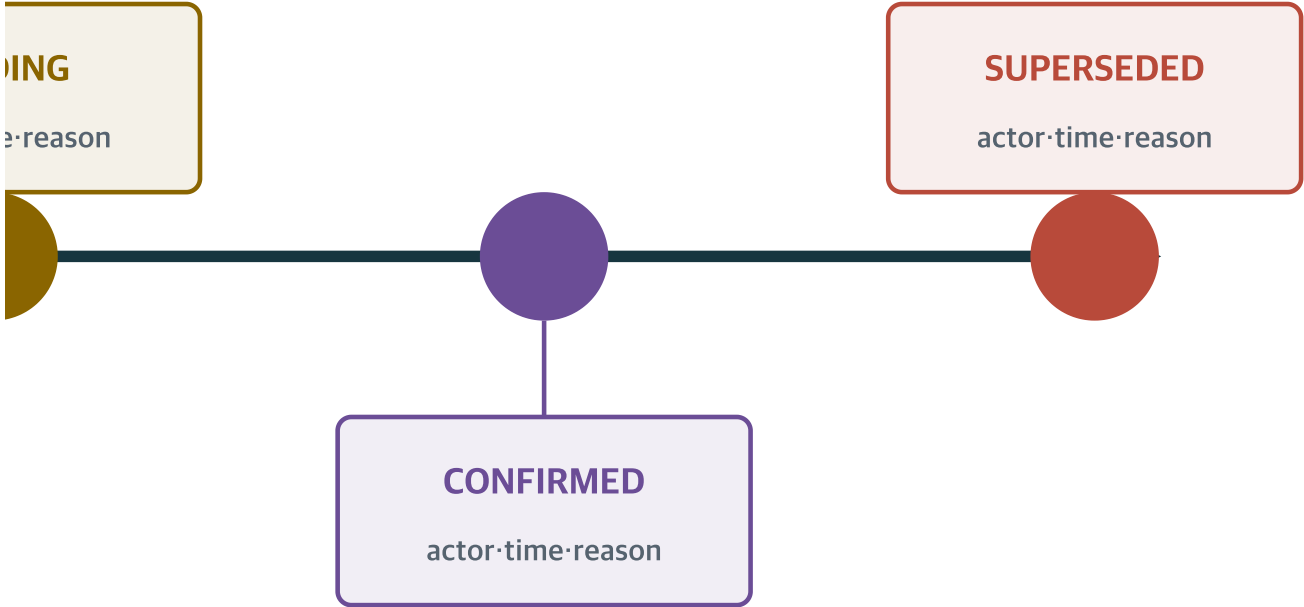


그림 8. Entity는 field 묶음이 아니라 identity·transition·history·provenance·retention이 있는 생명주기입니다.



ion·provenance를 함께 연결합니다.



OVENANCE · RETENTION

CORE IDEA 현재값만 맞는 데이터는 변경과 복구에서 충분하지 않습니다.

합성 사례의 ActionRecord와 AssignmentConfirmation은 draft·ready·pending·confirmed·rejected를 가집니다. 어떤 operation과 actor가 transition을 허용하는지, 이전 version과 source는 무엇인지, retention과 disposal은 어떻게 적용되는지 문서화해야 UI와 API가 같은 상태를 말합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Identity	action_record_id	stable·unique
Transition	pending→confirmed	OP-02·authorized actor
History	RecordVersion	current·superseded·restored
Provenance	meeting source·actor·time	source trace
Retention	policy·review·disposal	separate authority

틀린 예와 고친 예

틀린 예: DB column 표에 type과 nullable만 쓰면 data contract가 끝납니다.

고친 예: Field 의미와 identity·relationship·state
machine·history·provenance·retention·owner·version을 함께 씁니다.

5분 연습

1. Entity 다섯 개를 적습니다.
2. Identity와 relationship을 잇습니다.
3. Allowed transition과 actor를 씁니다.
4. History·retention·provenance evidence를 붙입니다.

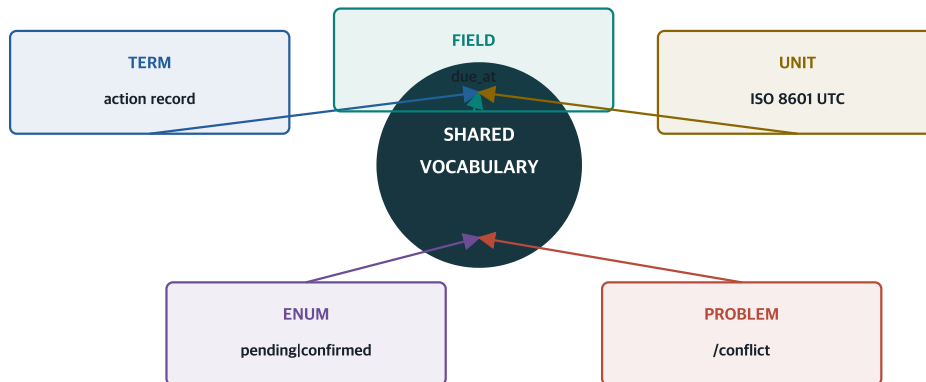
체크포인트: 현재값만 맞는 데이터는 변경과 복구에서 충분하지 않습니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

11. 공유 vocabulary·field·unit·enum 정렬하기

M12-04 · 09

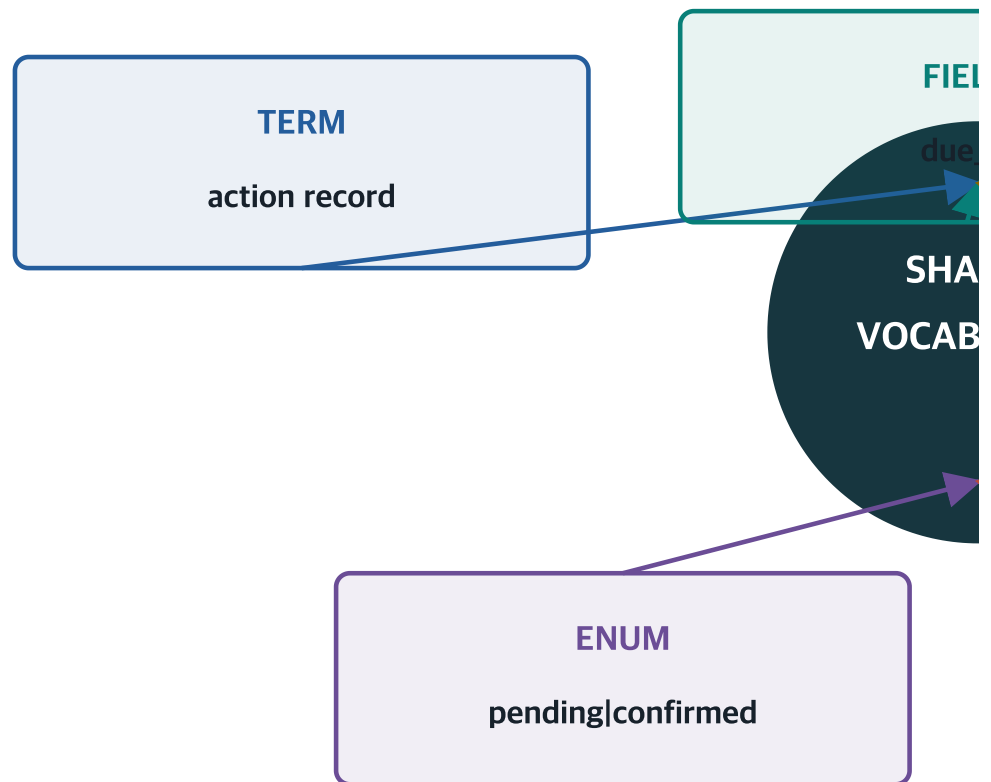
공유 용어집은 문서 사이의 의미 API입니다

Term·field·unit·time zone·enum·state·problem type의 canonical meaning과 alias를 한 registry에서 관리합니다.

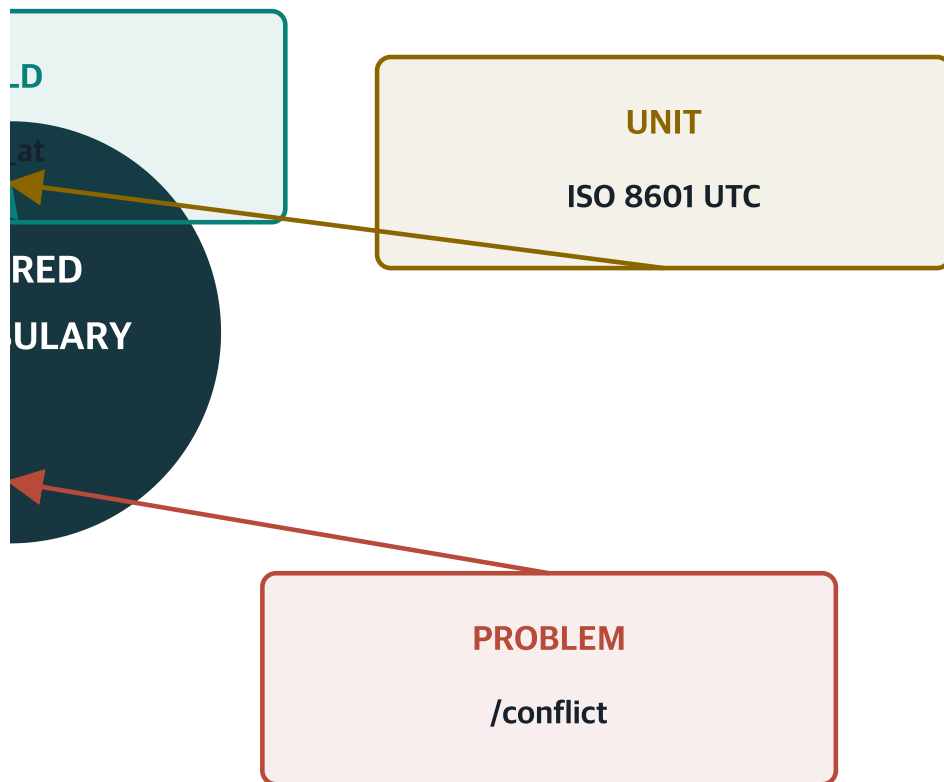


같은 이름의 다른 의미와 다른 이름의 같은 의미를 모두 collision으로 검토합니다.

그림 9. 공용 registry가 term·field·unit·time zone·enum·state·problem type의 의미와 alias를 관리합니다.



ring과 alias를 한 registry에서 관리합니다.



CORE IDEA 공유 용어집은 문서 사이의 의미 API입니다.

‘담당자’, ‘owner’, ‘assignee’가 같은지, ‘due date’가 날짜인지 UTC timestamp인지, **confirmed** 가 UI와 데이터에서 같은 상태인지 명시해야 합니다. ISO/IEC 11179 계열의 metadata naming·definition 원리를 참고해 name·definition·context·identifier·status를 관리합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Term	action record	정의·context·alias
Field	due_at	type·required·meaning
Unit	UTC instant	display time zone 별도
Enum	pending	confirmed
Problem	/problems/conflict	status·recovery map

틀린 예와 고친 예

틀린 예: 문서마다 자연스러운 표현을 사용하고 독자가 알아서 같은 뜻으로 이해합니다.

고친 예: Canonical term과 alias·deprecated name을 registry에 두고 field·enum·problem reference가 그 ID를 사용합니다.

5분 연습

- 충돌하기 쉬운 term 열 개를 모읍니다.
- Canonical name·definition·context를 씁니다.
- Alias와 금지 표현을 표시합니다.
- 모든 문서 reference가 같은 version인지 검사합니다.

체크포인트: 공유 용어집은 문서 사이의 의미 API입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

12. Acceptance-test-evidence matrix 만들기

M12-04 · 10

Acceptance는 화면·API·데이터를 같은 시나리오로 묶습니다

Positive·negative·boundary·authorization·accessibility acceptance를 test와 evidence 위치까지 연결합니다.

	POSITIVE	NEGATIVE	BOUNDARY	AUTH	ACCESS
SCR-03	✓	✓	✓	✓	✓
OP-02	✓	✓	✓	✓	✓
ENT-03	✓	✓	✓	✓	✓
AC-02	✓	✓	✓	✓	✓

CONTRACT → TEST → EVIDENCE LINK

한 layer의 unit test만으로 사용자 행동과 state transition의 연결을 증명할 수 없습니다.

그림 10. Positive만이 아니라 negative·boundary·authorization·accessibility를 같은 contract와 test evidence에 연결합니다.

Positive·negative·boundary·authorization·accessibility acceptance

	POSITIVE	NEGATIVE	BC
SCR-03	✓	✓	
OP-02	✓	✓	
ENT-03	✓	✓	
AC-02	✓	✓	
			CONTRACT → TEST

test와 evidence 위치까지 연결합니다.

BOUNDARY	AUTH	ACCESS
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
→ EVIDENCE LINK		

CORE IDEA 기대·검사·실행 증거는 서로 다르지만 같은 ID로 이어져야 합니다.

Acceptance는 관찰 가능한 expected outcome, test는 확인 절차, evidence는 실행 결과 위치입니다. 셋을 구분하면 requirement 변경 시 어떤 test를 다시 실행해야 하는지 계산할 수 있습니다. 합성 portfolio는 24 scenario 모두에 expected oracle과 evidence 위치를 둡니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Positive	authorized confirm→200	state confirmed
Negative	invalid due→400	field recovery
Boundary	version mismatch→412	draft preserved
Authorization	other org→403/404	no disclosure
Accessibility	keyboard·focus·announce	screen state evidence

틀린 예와 고친 예

틀린 예: API unit test가 있으므로 screen과 data acceptance도 검증됐다고 표시합니다.

고친 예: 같은 acceptance ID에 user-visible result·HTTP response·entity transition·accessibility evidence를 각각 연결합니다.

5분 연습

1. Core action 하나를 고릅니다.
2. 다섯 acceptance 관점을 씁니다.
3. Test type과 evidence path를 붙입니다.
4. Requirement change 시 rerun 목록을 만듭니다.

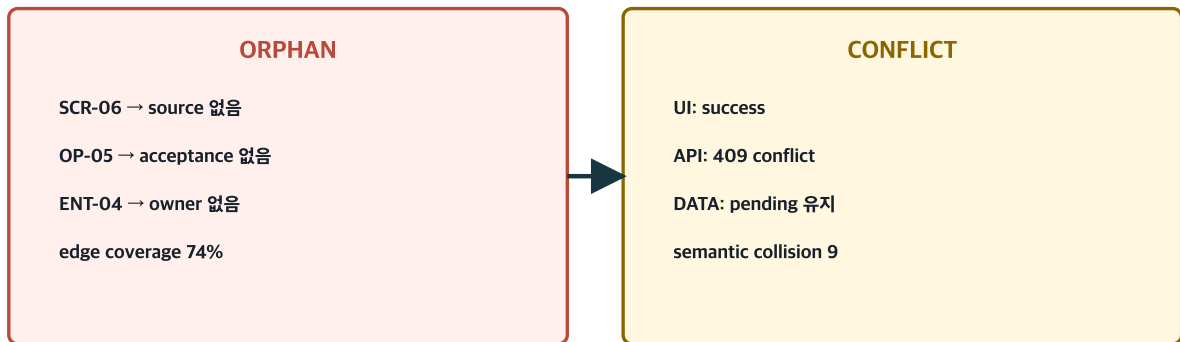
체크포인트: 기대·검사·실행 증거는 서로 다르지만 같은 ID로 이어져야 합니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

13. Orphan·missing·conflict를 계산해 찾기

M12-04 · 11

고아와 충돌은 눈대중이 아니라 계산해서 찾습니다

Source 없는 artifact와 같은 사건의 서로 다른 약속을 consistency matrix에서 분리해 owner에게 보냅니다.



RESOLVE → RE-RUN → ZERO / ZERO

Candidate Gate는 orphan 0·conflict 0을 요구하지만 그 수치 자체가 실제 승인 증거는 아닙니다.

그림 11. 존재 이유가 없는 artifact와 같은 사건의 다른 약속을 분리해 측정하고 owner에게 보냅니다.

Source 없는 artifact와 같은 사건의 서로 다른 약속을 consistency matrix

ORPHAN

SCR-06 → source 없음

OP-05 → acceptance 없음

ENT-04 → owner 없음

edge coverage 74%

RESOLVE → RE-RU

에서 분리해 owner에게 보냅니다.

CONFLICT

UI: success

API: 409 conflict

DATA: pending 유지

semantic collision 9

UN → ZERO / ZERO

CORE IDEA 문서 품질은 파일 수보다 끊긴 edge와 모순 수로 봅니다.

Orphan은 source·owner·test 중 필요한 연결이 없고, missing은 expected field나 state가 비어 있으며, conflict는 서로 다른 actual을 약속합니다. 합성 candidate는 canonical·source·semantic·UI·API·data·acceptance·change coverage 100%, orphan 0, conflict 0을 요구합니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Orphan	SCR-06 source 없음	source edge 추가
Missing	OP-05 acceptance 없음	AC ID 연결
Conflict	UI success·API 409	same event 정렬
Semantic	owner≠assignee	vocabulary resolution
Gate	0 orphan·0 conflict	re-run evidence

틀린 예와 고친 예

틀린 예: 모든 문서가 존재하고 리뷰 체크가 있으므로 일관성이 있다고 판단합니다.

고친 예: Canonical graph에서 broken edge·orphan node·field/state conflict query를 실행하고 0이 될 때까지 re-run합니다.

5분 연습

1. Artifact graph를 만듭니다.
2. In-degree·out-degree 0 node를 찾습니다.
3. Same event의 UI·API·data 값을 비교합니다.
4. Resolution owner와 evidence를 기록합니다.

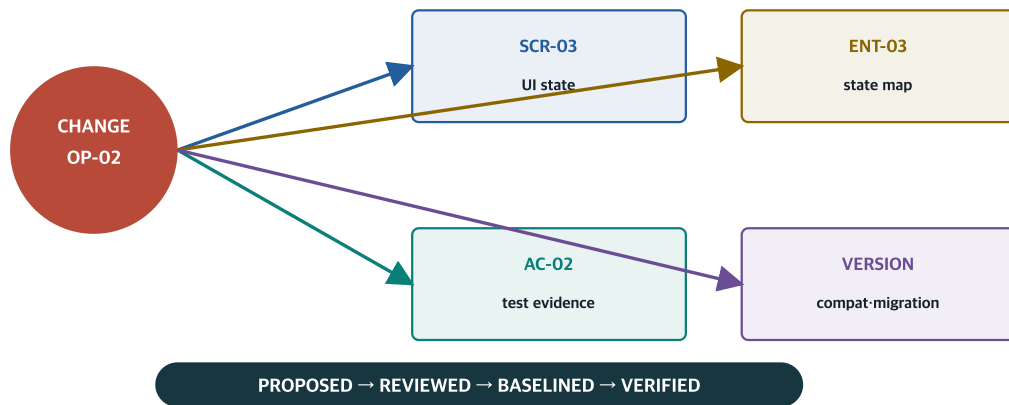
체크포인트: 문서 품질은 파일 수보다 끊긴 edge와 모순 수로 봅니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

14. Change impact·version·TBR 관리하기

M12-04 · 12

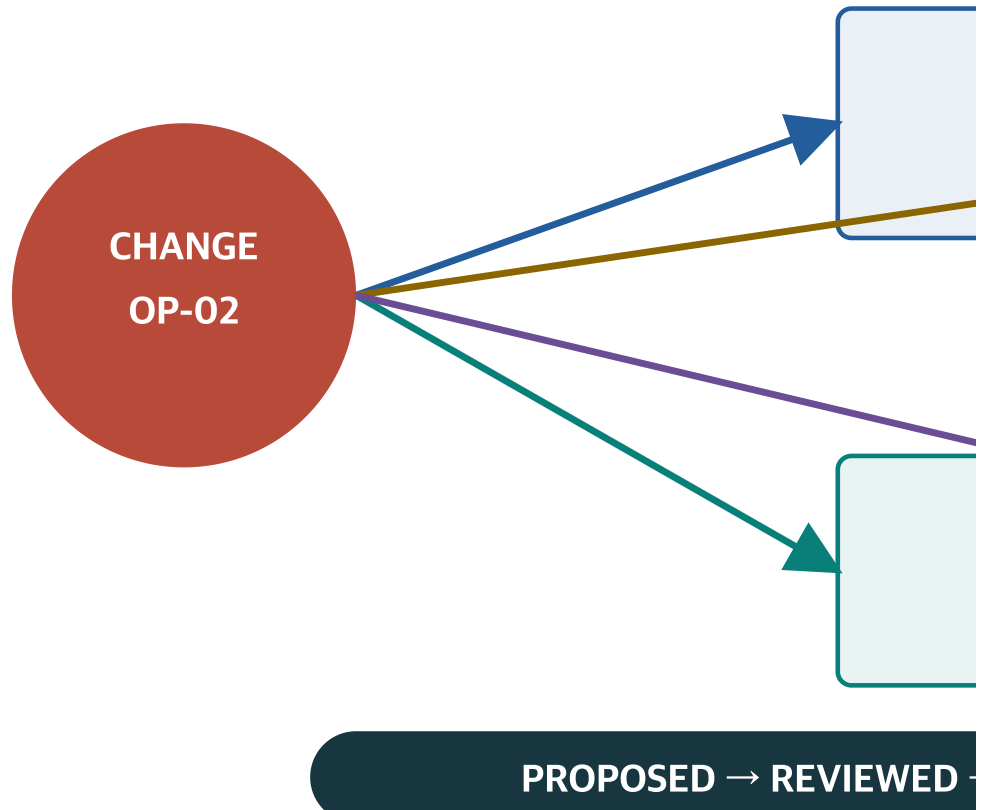
변경은 문서 하나가 아니라 연결된 edge 전체를 움직입니다

Source-affected IDs·compatibility·migration·test·owner·status·TBR·decision authority를 change record에 남깁니다.

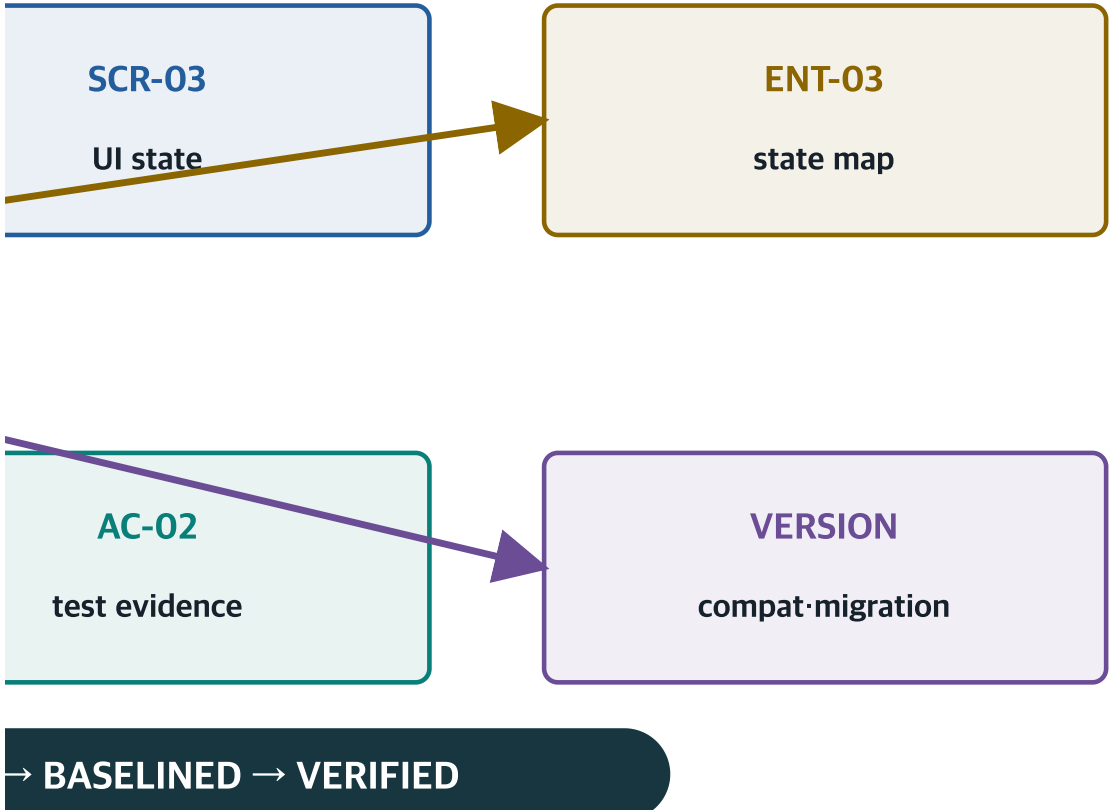


Version을 올렸다면 영향받은 screen·operation·entity·acceptance의 상태를 함께 검토합니다.

그림 12. Operation 하나의 변경이 screen·entity·acceptance·compatibility·migration·version에 미치는 영향을 계산합니다.



decision authority를 change record에 남깁니다.



CORE IDEA 변경은 파일 하나가 아니라 연결된 edge 전체를 움직입니다.

OP-02 request field를 바꾸면 **SCR-03** form, JSON Schema, **ENT-03** transition, **AC-02** tests, consumer compatibility와 migration이 함께 움직입니다. Proposed·reviewed·baselined·verified 상태와 TBR owner·due·authority를 기록해야 변경이 조용히 기준선을 깨지 않습니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Source	change request CR-07	reason·evidence
Affected	SCR-03·OP-02·ENT-03·AC-02	edge query
Compatibility	request·response consumer	breaking 여부
Migration	data·client·test	plan·fallback
Governance	owner·status·TBR·authority	baseline update

틀린 예와 고친 예

틀린 예: API 문서 version만 0.2.0으로 올리고 다른 문서는 다음 sprint에 고칩니다.

고친 예: Affected ID를 먼저 계산하고 compatibility·migration·test를 검토한 뒤 관련 기준선과 trace version을 함께 갱신합니다.

5분 연습

1. 가상 change 하나를 만듭니다.
2. Affected IDs를 graph로 찾습니다.
3. Compatibility·migration·test를 씁니다.
4. Status·TBR·authority와 baseline update를 기록합니다.

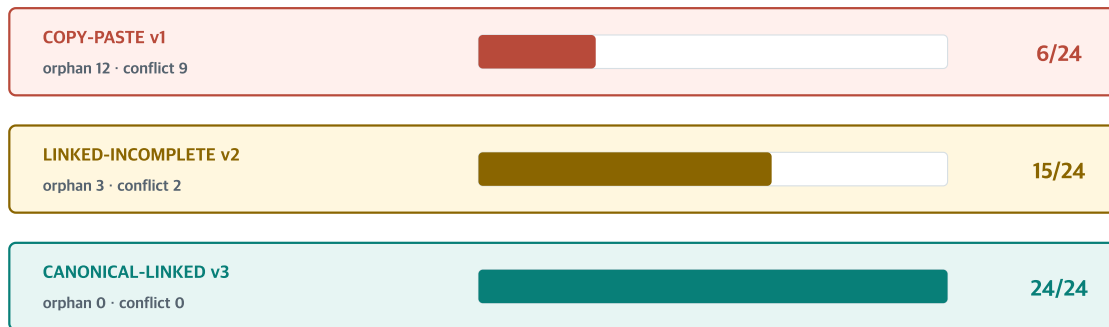
체크포인트: 변경은 파일 하나가 아니라 연결된 edge 전체를 움직입니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

15. 세 문서 package를 비교해 drift를 학습하기

M12-04 · 13

세 패키지를 비교하면 연결 품질의 차이가 보입니다

복사본 묶음에서 부분 연결을 거쳐 canonical source·semantic·contract·trace가 완성된 후보로 개선합니다.



FIXED 18 · REGRESSED 0 · QUALITY +100%

문서 수가 18→21로 늘어난 것보다 orphan 12→0·conflict 9→0이 더 중요한 변화입니다.

그림 13. 세 version을 실행하면 복사본·부분 연결·완전 연결 후보가 어떤 실패를 남기는지 수치로 보입니다.

복사본 묶음에서 부분 연결을 거쳐 canonical source·semantic·contract·t

COPY-PASTE v1

orphan 12 · conflict 9

LINKED-INCOMPLETE v2

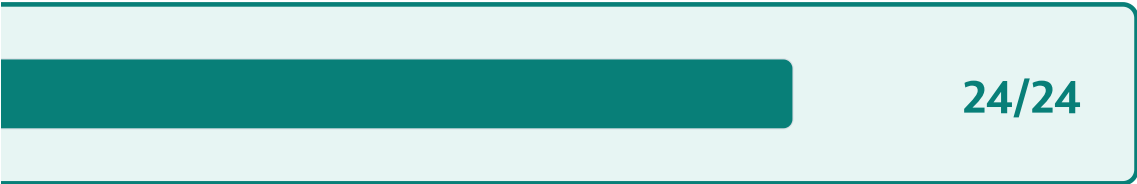
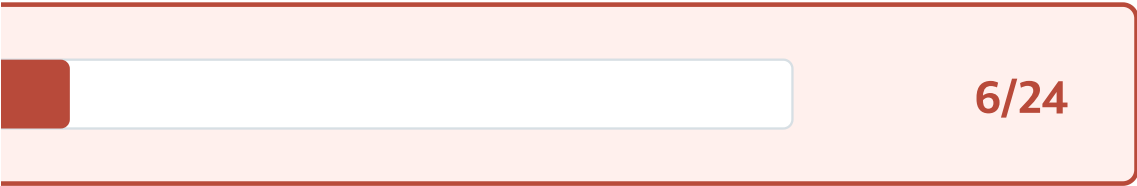
orphan 3 · conflict 2

CANONICAL-LINKED v3

orphan 0 · conflict 0

FIXED 18 · REGRESSED

race가 완성된 후보로 개선합니다.



00 · QUALITY +100%

CORE IDEA 문서 개수보다 source·semantic·contract·trace의 닫힘이 중요합니다.

기준선은 18개 artifact가 있어도 6/24, orphan 12, conflict 9입니다. 중간안은 21개를 모두 갖췄지만 state·problem·lifecycle·change edge가 덜 닫혀 15/24입니다. Candidate는 21개 artifact를 같은 ID·meaning·test·change record로 연결해 24/24를 만듭니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
copy-paste v1	6/24	orphan 12·conflict 9
linked incomplete v2	15/24	orphan 3·conflict 2
canonical linked v3	24/24	orphan 0·conflict 0
Comparison	fixed 18	regressed 0
Quality	coverage 100%	not approval

틀린 예와 고친 예

틀린 예: 문서 21개가 모두 있으므로 중간안도 complete입니다.

고친 예: Artifact 존재와 edge quality를 분리해 보고, fail scenario의 expected/actual mismatch를 해결한 뒤 다시 검사합니다.

5분 연습

1. 세 variant를 차례로 실행합니다.
2. Fail 18→9→0을 기록합니다.
3. Orphan·conflict 변화를 비교합니다.
4. 고친 edge와 남은 approval boundary를 설명합니다.

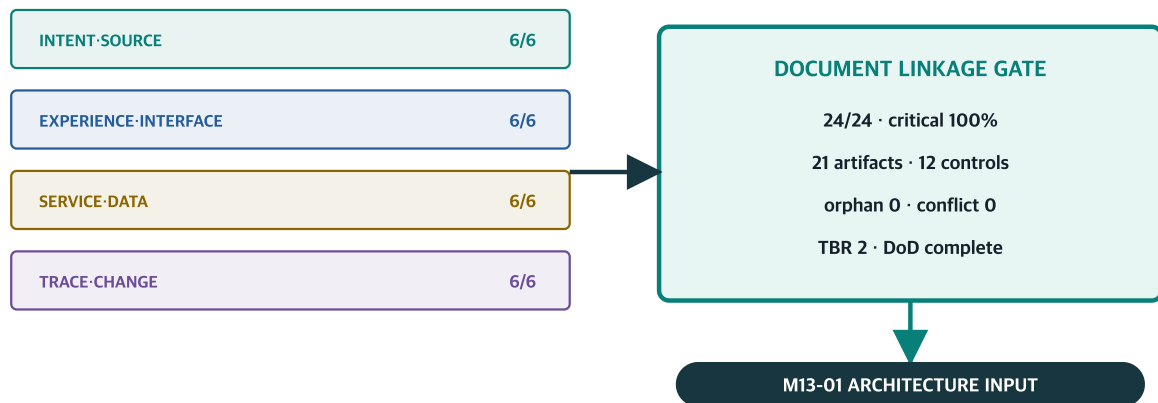
체크포인트: 문서 개수보다 source·semantic·contract·trace의 담힘이 중요합니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

16. Document Linkage Gate와 M13-01 handoff 만들기

M12-04 · 14

네 lane이 모두 통과해야 architecture 입력 후보가 됩니다

24 scenario·12 control과 canonical·semantic·UI·API·data·acceptance·change coverage, orphan·conflict·TBR를 함께 봅니다.

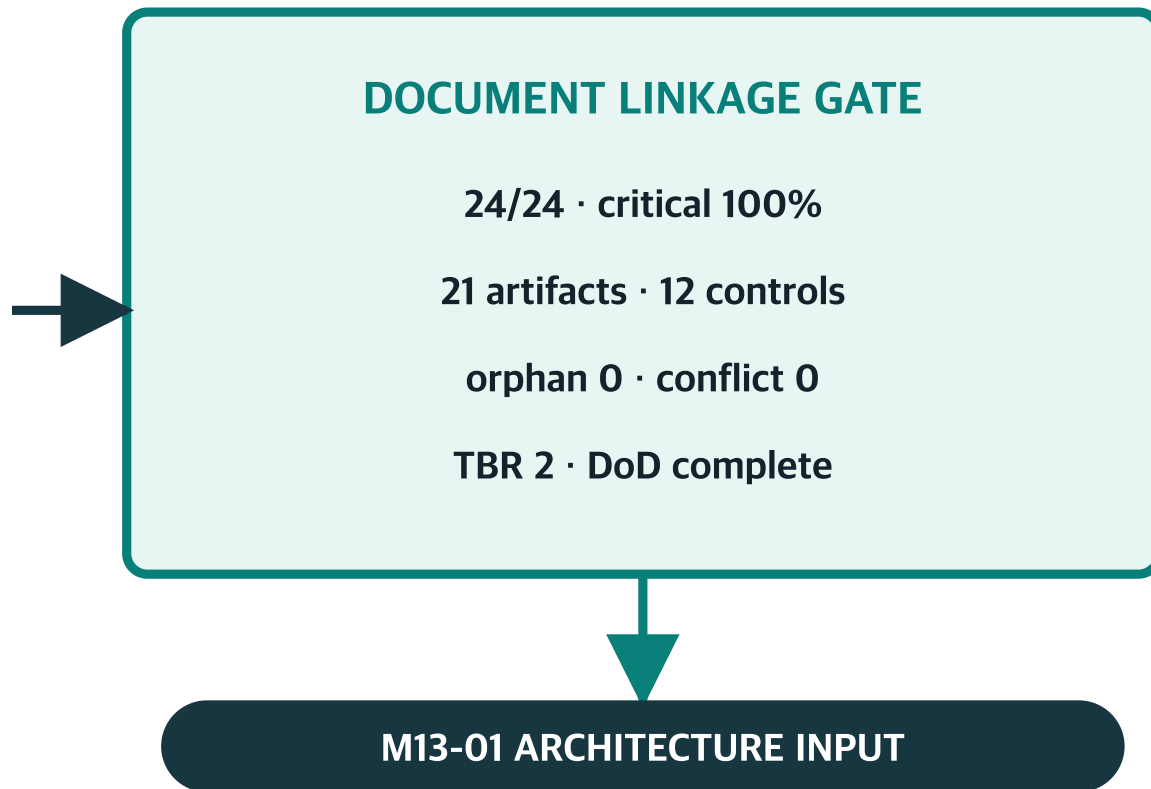


document_linkage_ready는 M13-01 선택 입력이며 실제 architecture·build·release 승인이 아닙니다.

그림 14. 네 lane과 비기능 coverage·orphan·conflict·TBR·DoD를 함께 통과해야 architecture 입력 후보가 됩니다.

INTENT·SOURCE	6/6
EXPERIENCE·INTERFACE	6/6
SERVICE·DATA	6/6
TRACE·CHANGE	6/6

change coverage, orphan·conflict·TBR를 함께 봅니다.



CORE IDEA Gate는 좋은 인상을 versioned evidence와 잔여 위험으로 바꿉니다.

합성 Gate는 24/24 scenario, critical 100%, 12 controls, 네 lane 각 6개,
canonical·source·semantic·UI·API·data·acceptance·change 100%, orphan 0, conflict 0, TBR 2와 DoD·conflict
resolution을 요구합니다. M13-01에는 quality constraints와 interface boundary, unresolved risk를 숨기지
않고 넘깁니다.

합성 사례로 읽기

요소	합성 사례	검토 evidence
Intent·source	6/6	handoff·authority·PRD·semantics
Experience·interface	6/6	flow·screen·state·access
Service·data	6/6	operation·problem·schema·lifecycle
Trace·change	6/6	acceptance·matrix·impact·Gate
Handoff	M13-01	constraint·interface·risk·TBR

틀린 예와 고친 예

틀린 예: document_linkage_ready가 나왔으므로 선택한 architecture로 개발을 시작합니다.

고친 예: Linked package를 M13-01 option comparison 입력으로 넘기고,
architecture·security·cost·operation·release authority를 별도 Gate로 둡니다.

5분 연습

1. 네 lane을 판정합니다.
2. Coverage·orphan·conflict·TBR를 기록합니다.
3. Residual risk와 decision authority를 씁니다.
4. M13-01 constraint·interface handoff를 만듭니다.

체크포인트: Gate는 좋은 인상을 versioned evidence와 잔여 위험으로 바꿉니다. 지금 쓴 항목의 source·ID·owner·status·version·evidence를 다시 확인합니다.

17. 합성 candidate의 21개 document artifact

아래 목록은 21개의 독립 문서를 만들라는 뜻이 아닙니다. 각 artifact가 어떤 fact를 authoritative하게 갖고 어떤 source ID를 참조하는지 보여 주는 logical inventory입니다.

ID	Type	Title	Source IDs	Owner
PRD-TFU-001	prd	회의 후 실행 기록 PRD	SCP-01, SCP-02, SCP-03, SCP-04, SCP-05, SCP-06, SCP-07, SCP-08, SCP-09	product-owner
FLOW-01	flow	Create·confirm core flow	SCP-01, SCP-02, SCP-03	service-designer
FLOW-02	flow	Correct·conflict·recover flow	SCP-04, SCP-05	service-designer
SCR-01	screen	Meeting source picker	SCP-01	product-designer
SCR-02	screen	Action record editor	SCP-01	product-designer
SCR-03	screen	Owner·due confirmation	SCP-02	product-designer
SCR-04	screen	Current meeting action list	SCP-03	product-designer
SCR-05	screen	Record detail·version history	SCP-04	product-designer
SCR-06	screen	Conflict recovery	SCP-05	product-designer
OP-01	api-operation	Create action record	SCP-01, SCR-02	api-owner
OP-02	api-operation	Confirm owner and due	SCP-02, SCR-03	api-owner
OP-03	api-operation	List meeting action records	SCP-03, SCR-04	api-owner
OP-04	api-operation	Read action record detail	SCP-04, SCR-05	api-owner

ID	Type	Title	Source IDs	Owner
OP-05	api-operation	Update with version precondition	SCP-04, SCP-05, SCR-06	api-owner
OP-06	api-operation	Read version history	SCP-04, SCR-05	api-owner
OP-07	api-operation	Restore selected record version	SCP-05, SCR-06	api-owner
ENT-01	data-entity	MeetingSource	SCP-01, OP-01	data-owner
ENT-02	data-entity	ActionRecord	SCP-01, SCP-03, OP-01, OP-03	data-owner
ENT-03	data-entity	AssignmentConfirmation	SCP-02, OP-02	data-owner
ENT-04	data-entity	RecordVersion	SCP-04, SCP-05, OP-05, OP-06, OP-07	data-owner
ENT-05	data-entity	OutcomeEvent	SCP-06	analytics-owner

17.1 7개 API operation

ID	Method	Path	Success	Problem status
OP-01	POST	/action-records	201	400, 403, 409
OP-02	PATCH	/action-records/{id}/confirmation	200	400, 403, 404, 409
OP-03	GET	/meetings/{id}/action-records	200	403, 404
OP-04	GET	/action-records/{id}	200	403, 404
OP-05	PATCH	/action-records/{id}	200	400, 403, 404, 409, 412
OP-06	GET	/action-records/{id}/versions	200	403, 404
OP-07	POST	/action-records/{id}/restore	200	400, 403, 404, 409

17.2 5개 data entity

ID	Entity	Identity	States
ENT-01	MeetingSource	meeting_source_id	available, unavailable
ENT-02	ActionRecord	action_record_id	draft, ready, pending, confirmed, rejected
ENT-03	AssignmentConfirmation	confirmation_id	pending, confirmed, rejected
ENT-04	RecordVersion	record_version_id	current, superseded, restored
ENT-05	OutcomeEvent	outcome_event_id	accepted, excluded

18. 열두 개 문서 연결 통제

Control	통제 제목	최소 evidence
CTRL-01	M12-03 outcome·scope·version handoff	M12-03 handoff manifest
CTRL-02	Canonical ID·source of truth·metadata	Canonical registry
CTRL-03	PRD intent·scope·metric·decision contract	Versioned PRD
CTRL-04	User flow·screen inventory·entry/exit	Flow and screen map
CTRL-05	UI state·interaction·accessibility	UI state contract
CTRL-06	API operation·auth·request·response·error	OpenAPI operation contract
CTRL-07	Data entity·field·state·history·retention	Data contract and state map
CTRL-08	Shared vocabulary·unit·enum·error semantics	Shared vocabulary registry
CTRL-09	Acceptance·test·evidence linkage	Acceptance and evidence matrix
CTRL-10	Cross-document consistency matrix	Cross-document trace matrix
CTRL-11	Change impact·version·status·TBR·conflict	Change impact and decision log
CTRL-12	Document Linkage Gate·M13-01 handoff	Document Linkage Gate

19. 24개 합성 scenario portfolio

각 lane에는 정확히 6개 scenario가 있습니다. Critical scenario 하나라도 실패하면 평균 coverage가 높아도 candidate는 blocked입니다.

ID	Lane	Scenario	Risk	Source → Sink	Control
SC-01	intent-source	M12-03 goal·scope·metric·decision handoff	orphan-artifact	m12-03-handoff → document-manifest	CTRL-01
SC-02	intent-source	Document package ID·owner·status·version metadata	copy-drift	package-registry → document-header	CTRL-02, CTRL-11
SC-03	intent-source	PRD user·problem·goal·scope·floor ·metric·decision	orphan-artifact	m12-03-contract → prd	CTRL-03
SC-04	intent-source	Fact별 source of truth와 snapshot reference	copy-drift	canonical-registry → document-references	CTRL-02
SC-05	intent-source	Canonical artifact ID와 stable link	orphan-artifact	id-registry → trace- edges	CTRL-02, CTRL-10
SC-06	intent-source	Glossary·field·unit·time·enum· problem type 의미	semantic-collision	vocabulary-registry → all-documents	CTRL-08
SC-07	experience-interface	Whole problem start/end·actor·branch·recover y flow	hidden-state	m12-03-journey → user- flows	CTRL-04
SC-08	experience-interface	Screen inventory·purpose·entry·exit·f low links	orphan-artifact	user-flows → screen- inventory	CTRL-04, CTRL-10
SC-09	experience-interface	Loading·empty·editing·submit ting·success·error·denied·conf lict	hidden-state	screen-inventory → ui- state-contract	CTRL-05
SC-10	experience-interface	Interaction·keyboard·focus·na me/role/value semantics	contract-mismatch	ui-state-contract → accessible-interaction	CTRL-05
SC-11	experience-interface	Field input·client hint·server validation·field error	contract-mismatch	field-registry → validation-contract	CTRL-05, CTRL-06, CTRL-07

ID	Lane	Scenario	Risk	Source → Sink	Control
SC-12	experience-interface	Screen action→operation→entity state trace	orphan-artifact	screen-actions → service-data-edges	CTRL-04, CTRL-06, CTRL-07, CTRL-10
SC-13	service-data-contract	Operation ID·method·path·request·success·version	contract-mismatch	screen-actions → openapi-operations	CTRL-06
SC-14	service-data-contract	Authentication·authorization·401/403/404 boundary	contract-mismatch	actor-permission-map → operation-security	CTRL-06
SC-15	service-data-contract	Request·response JSON Schema·required/null/enum examples	semantic-collision	field-registry → schema-contract	CTRL-06, CTRL-07, CTRL-08
SC-16	service-data-contract	RFC 9457 problem type·status·field issue·UI recovery	contract-mismatch	error-registry → problem-ui-map	CTRL-05, CTRL-06, CTRL-08
SC-17	service-data-contract	Entity·field identity·type·required/null·enum·relationship	semantic-collision	domain-vocabulary → data-contract	CTRL-07, CTRL-08
SC-18	service-data-contract	State transition·version history·provenance·retention	hidden-state	data-contract → lifecycle-contract	CTRL-07
SC-19	trace-change-governance	Positive·negative·boundary·authorization·accessibility acceptance	orphan-artifact	linked-contracts → acceptance-matrix	CTRL-09
SC-20	trace-change-governance	Goal→scope→flow→screen→operation→entity→test matrix	orphan-artifact	document-package → trace-matrix	CTRL-09, CTRL-10
SC-21	trace-change-governance	Field·state·enum·error cross-document consistency	contract-mismatch	linked-artifacts → consistency-report	CTRL-08, CTRL-10
SC-22	trace-change-governance	Change source·affected IDs·compatibility·migration·test	copy-drift	change-request → impact-report	CTRL-11
SC-23	trace-change-governance	TBR·conflict·decision authority·status·review log	false-document-approval	review-findings → decision-log	CTRL-02, CTRL-11

ID	Lane	Scenario	Risk	Source → Sink	Control
SC-24	trace-change-governance	Document Linkage Gate·residual risk·M13-01 handoff	false-document-approval	document-evidence-pack → document-linkage-gate	CTRL-01, CTRL-09, CTRL-10, CTRL-11, CTRL-12

20. 세 document package를 비교해 배우기

관찰	Copy-paste 기준선	부분 연결	Canonical candidate
Scenario	6/24	15/24	24/24
Artifact	18	21	21
Canonical ID	26%	86%	100%
Semantic alignment	25%	83%	100%
UI·API·data	17·29·20%	67·71·80%	100·100·100%
Orphan	12	3	0
Conflict	9	2	0
Decision	blocked_document_drift	blocked_contract_gaps	document_linkage_ready

20.1 기준선이 실패하는 이유

문서는 여러 개 있지만 authoritative source·stable ID·shared semantics·state·problem·lifecycle·test·change edge가 없습니다. 복사한 사실이 많아질수록 최신값은 더 쉽게 갈라집니다.

20.2 중간안이 실패하는 이유

Artifact inventory와 주요 ID는 갖췄지만 source of truth·flow recovery·UI states·problem mapping·lifecycle·trace·change impact가 부분적입니다. 연결선이 있다는 사실과 완전한 계약은 다릅니다.

20.3 Candidate가 통과하는 이유

모든 문서가 길어서가 아니라 fact authority·canonical identity·shared meaning·same-event contract·acceptance evidence·change impact가 달혀 orphan 0·conflict 0을 재현하기 때문입니다.

21. PRD·화면·API·데이터 연결 스튜디오 읽기

M12-04 · DOCUMENT LINKAGE GATE

PRD·화면·API·데이터 연결 스튜디오

교육용 합성 문서만 사용됩니다. document_linkage_ready는 실제 문서 설계·이력·개발·배포 승인이 아닙니다.

M12-03 목표·범위 → PRD → 흐름·화면·상태 → API·Problem → 데이터·생명주기 → Acceptance·Test → Linkage Gate → M13-01

DOCUMENT ARTIFACT

21개

PRD / FLOW / SCREEN

1 / 2 / 6

API / ENTITY / UI STATE

7 / 5 / 8

CANONICAL / SEMANTIC

100% / 100%

ORPHAN / CONFLICT

0 / 0

LINKAGE CHECK

21 artifact · TBR 2

01 연결 통제

document-linkage-contract-1.0.0 · document-linkage-scenarios-1.0.0

CTRL-01
M12-03
outcome-scope-version handoff
M12-03 handoff manifest

CTRL-02
Canonical ID source of truth-metadata
Canonical registry

CTRL-03
PRD
intent-scope-metric-decision contract
Versioned PRD

CTRL-04
User flow-screen inventory entry/exist
Flow and screen map

CTRL-05
UI
state-interaction-accessibility
UI state contract

CTRL-06
API
operation-auth-request-response-error
OpenAPI operation contract

CTRL-07
Data
entity-field-state-history retention
Data contract and state map

CTRL-08
Shared vocabulary unit-environment-error semantics
Shared vocabulary registry

CTRL-09
Acceptance-test-evidence linkage
Acceptance and evidence matrix

CTRL-10
Cross-document consistency matrix
Cross-document trace matrix

CTRL-11
Change impact-version status-TBR conflict
Change impact and decision log

CTRL-12
Document Linkage Gate-M13-01 handoff
Document Linkage Gate

끊어진 연결의 6가지 신호

복사본 drift · 같은 outcome-field-environment-error를 여러 문서에 복사해 서로 다른 확인없이 생성

고아 산출물 · Screen operation entity acceptance가

02 24개 연결 시나리오

현재

의도·기준정보

흐름·화면

API 데이터

추적·변경

ID	연결 영역	위험	Source	Sink	결과
SC-01	의도·기준정보	고아 산출물	m12-03-handoff	document-manifest	PASS
SC-02	의도·기준정보	복사본 drift	package-registry	document-header	PASS
SC-03	의도·기준정보	고아 산출물	m12-03-contract	prd	PASS
SC-04	의도·기준정보	복사본 drift	canonical-registry	document-references	PASS
SC-05	의도·기준정보	고아 산출물	id-registry	trace-edges	PASS
SC-06	의도·기준정보	의미 충돌	vocabulary-registry	all-documents	PASS
SC-07	흐름·화면	숨은 상태	m12-03-journey	user-flows	PASS
SC-08	흐름·화면	고아 산출물	user-flows	screen-inventory	PASS
SC-09	흐름·화면	숨은 상태	screen-inventory	ui-state-contract	PASS
SC-10	흐름·화면	계약 불일치	ui-state-contract	accessible-interaction	PASS
SC-11	흐름·화면	계약 불일치	field-registry	validation-contract	PASS
SC-12	흐름·화면	고아 산출물	screen-actions	service-data-edges	PASS
SC-13	API 데이터	계약 불일치	screen-actions	openapi-operations	PASS
SC-14	API 데이터	계약 불일치	actor-permission-map	operation-security	PASS
SC-15	API 데이터	의미 충돌	field-registry	schema-contract	PASS
SC-16	API 데이터	계약 불일치	error-registry	problem-ui-map	PASS
SC-17	API 데이터	의미 충돌	domain-vocabulary	data-contract	PASS
SC-18	API 데이터	숨은 상태	data-contract	lifecycle-contract	PASS
SC-19	추적·변경	고아 산출물	linked-contracts	acceptance-matrix	PASS

03 문서 패키지 판정

검사할 package version

Canonical ID·evidence 연결 후보

Source of truth shared semantics-UI/API/data contracts-acceptance-trace-change impact-M13-01 handoff를 versioned evidence로 담은 상태

회의 후 실행 기록

팀이 회의에서 결정된 후속 조치를 백업없이 해당·기준과 함께 실행한다

team-follow-up-document-pack-2026-07-v1 · PRD-TFU-001

PRD-TFU-001 prd · 회의 후 실행 기록 PRD

FLOW-01 flow · Create-confirm core flow

FLOW-02 flow · Correct-conflict-recover flow

SCR-01 screen · Meeting source picker

SCR-02 screen · Action record editor

SCR-03 screen · Owner-due confirmation

+ 15 linked artifact

Primary hypothesis · 운영 조정 담당자가 한 회의 source에서 실행 기록을 만들고 제안된 담당자가 담당·기준을 확인하면 한한 기록들이 놓여지고 제곱력 시간이 들어 든다.

선택 패키지 검사

기준선과 비교

회귀 계약 검사

document_linkage_ready

SCENARIO

24/24

CRITICAL

100%

실습 화면 1. 왼쪽은 12 control과 6 failure type, 가운데는 24 scenario, 오른쪽은 package version·artifact preview·Gate·trace입니다.

21.1 한 scenario를 evidence contract로 읽기

SC-16 · RFC 9457 problem type·status·field issue·UI recovery

행을 선택하면 계약과 실제값을 비교합니다.

BOUNDARY · 문서 권한 <pre>{ "zone": "service-contract", "controls": ["CTRL-05", "CTRL-06", "CTRL-08"] }</pre>	FLOW · source에서 sink <pre>{ "source": "error-registry", "sink": "problem-ui-map", "lane": "service-data-contract" }</pre>
EXPECTED · linkage oracle <pre>{ "code": "PROBLEM_DETAILS_ALIGNED", "field_issues_present": true, "http_status_aligned": true, "safe_detail_present": true, "stable_problem_code_present": true, "type_uri_present": true, "ui_recovery_linked": true }</pre>	ACTUAL · 현재 package <pre>{ "code": "PROBLEM_DETAILS_ALIGNED", "field_issues_present": true, "http_status_aligned": true, "safe_detail_present": true, "stable_problem_code_present": true, "type_uri_present": true, "ui_recovery_linked": true }</pre>

실습 화면 2. Scenario를 선택하면 document authority·source→sink·expected oracle·actual package를 나란히 비교합니다.

1. `copy-paste-docs-v1` 을 실행해 18개 FAIL과 orphan 12·conflict 9를 봅니다.
2. `linked-incomplete-pack-v2` 로 바꿔 남은 9개 contract gap을 봅니다.
3. `canonical-linked-pack-v3` 에서 24/24·artifact 21·orphan 0·conflict 0을 확인합니다.
4. 기준선 비교에서 fixed 18·regressed 0을 확인합니다.
5. 6개 regression contract를 실행합니다.

21.2 모바일에서 세 구역 복습하기

① 통제·위험

document-linkage-contract-1.0.0 · document-linkage-scenarios-1.0.0

CTRL-01

M12-03 outcome-scope-version handoff

M12-03 handoff manifest

CTRL-02

Canonical ID-source of truth-metadata

Canonical registry

CTRL-03

PRD intent-scope-metric-decision contract

Versioned PRD

CTRL-04

User flow-screen inventory-entry/exit

Flow and screen map

CTRL-05

UI state-interaction-accessibility

UI state contract

CTRL-06

API operation-auth-request-response-error

OpenAPI operation contract

CTRL-07

Data entity-field-state-history-retention

Data contract and state map

CTRL-08

Shared vocabulary-unit-enum-error semantics

Shared vocabulary registry

CTRL-09

Acceptance-test-evidence linkage

② 시나리오·evidence

복사본 drift

canonical

SC-05

의도·기준정보

고아 산출물

id-registr

SC-06

의도·기준정보

의미 충돌

vocabular

SC-07

흐름·화면

숨은 상태

m12-03-k

SC-08

흐름·화면

고아 산출물

user-flow

SC-09

흐름·화면

숨은 상태

screen-im

SC-10

흐름·화면

계약 불일치

ui-state-c

SC-11

흐름·화면

계약 불일치

field-regis

SC-12

흐름·화면

고아 산출물

screen-ac

SC-13

API-데이터

계약 불일치

screen-ac

SC-14

API-데이터

계약 불일치

actor-per

SC-15

API-데이터

의미 충돌

field-regis

SC-16

API-데이터

계약 불일치

error-regi

SC-17

API-데이터

의미 충돌

domain-v

SC-18

API-데이터

숨은 상태

data-cont

SC-19

추적·변경

고아 산출물

linked-cor

SC-20

추적·변경

고아 산출물

document

SC-21

추적·변경

계약 불일치

linked-art

SC-22

추적·변경

복사본 drift

change-re

SC-23

추적·변경

거짓 문서 승인

review-fir

SC-24

추적·변경

거짓 문서 승인

document

시나리오 상세

행을 선택하면 계약과 실제값을 비교합니다.

③ Gate·추적

선택 패키지 검사

기준선과 비교

회귀 계약 검사

document_linkage_ready

SCENARIO

24/24

CRITICAL

100%

CONTROL

12/12

LANE

4/4

Control trace

CTRL-01

2 case · 0 fail

LINK

CTRL-02

4 case · 0 fail

LINK

CTRL-03

1 case · 0 fail

LINK

CTRL-04

3 case · 0 fail

LINK

CTRL-05

4 case · 0 fail

LINK

CTRL-06

6 case · 0 fail

LINK

CTRL-07

5 case · 0 fail

LINK

CTRL-08

5 case · 0 fail

LINK

CTRL-09

3 case · 0 fail

LINK

CTRL-10

6 case · 0 fail

LINK

CTRL-11

4 case · 0 fail

LINK

CTRL-12

1 case · 0 fail

LINK

실행 기록

1. canonical-linked-pack-v3: 24/24 PASS · orphan 0 · conflict 0 · document_linkage_ready

세 이미지를 왼쪽부터 보며 authority → scenario → expected/actual → Gate → trace 를 말로 설명합니다. 전체 화면을 한 번에 축소하지 않고, 각 학습 구역을 읽을 수 있는 크기로 나눠 복습합니다.

22. 90분 실습 workbook

시간	행동	남길 evidence	통과 질문
0~10분	M12-03 manifest·package metadata	source·owner·status·version	무엇이 authoritative한가
10~20분	PRD intent·scope·metric	PRD ID·reference map	구현 상세를 중복 source로 만들지 않았나
20~35분	flow·screen·8 UI state	entry·exit·branch·recovery	숨은 상태가 없는가
35~50분	7 operation·problem mapping	auth·schema·status·recovery	UI와 HTTP가 같은가
50~65분	5 entity·lifecycle·vocabulary	identity·transition·history	Data meaning이 같은가
65~75분	acceptance·test·evidence	다섯 관점 matrix	양방향 trace가 되는가

시간	행동	남길 evidence	통과 질문
75~85분	orphan·conflict·change impact	0/0·affected IDs·TBR	변경이 기준선을 깨지 않나
85~90분	Gate·M13-01 handoff	pass/fail·risk·boundary	architecture 입력과 승인을 구분하나

22.1 실습 중 반드시 남길 아홉 줄

M12-03 source version:
 Package ID + owner + status + version:
 Fact → authoritative location:
 Goal → scope → flow → screen → operation → entity → acceptance:
 UI state ↔ HTTP problem ↔ data transition:
 Vocabulary + field + unit + enum + problem type:
 Orphan / conflict / TBR:
 Change impact + compatibility + migration + test:
 Gate result + residual risk + M13-01 handoff:

23. 제출 전 한 장 점검표

영역	PASS 질문	증거
Source	Fact별 authoritative location·owner·version이 하나인가	Source registry
PRD	User·problem·goal·scope·metric·decision이 있는가	Versioned PRD
Identity	Canonical ID가 unique·stable·resolvable한가	ID registry
Flow	Start·end·branch·recovery가 있는가	Flow map
UI	8 state·keyboard·focus·name/role/value가 있는가	UI state contract
API	Operation·auth·schema·status·problem이 있는가	OpenAPI
Data	Identity·state·history·retention이 있는가	Data contract
Semantics	Term·field·unit·enum·problem 의미가 같은가	Vocabulary registry
Evidence	다섯 acceptance 관점과 test run을 잇는가	Evidence matrix
Consistency	Orphan 0·conflict 0인가	Consistency report

영역	PASS 질문	증거
Change	Affected IDs·compatibility·migration·TBR가 있는가	Impact record
Governance	Gate·risk·authority·승인 경계가 명시됐는가	Decision log

24. 셀프 테스트 30

답을 보기 전 종이에 한 문장과 합성 사례 ID 하나를 씁니다. 정답을 외우기보다 내 package의 source와 evidence 위치를 표시합니다.

1. 문서 연결이 같은 내용을 여러 파일에 복사하는 것과 다른 이유는 무엇인가요?

정답과 해설

연결은 fact의 authoritative location을 하나 정하고 다른 문서가 canonical ID와 snapshot version으로 참조하는 일입니다. 복사본을 각각 고치면 최신값이 갈라집니다.

2. Source of truth의 최소 필드는 무엇인가요?

정답과 해설

Fact ID, authoritative location, owner, status, version, review 기준, conflict resolution rule입니다.

3. PRD가 authoritative해야 하는 정보와 아닌 정보를 구분해 보세요.

정답과 해설

PRD는 user·problem·goal·outcome·scope·quality floor·metric·decision rule의 source가 됩니다. 화면 픽셀, endpoint schema, table field 상세를 중복 source로 두지 않습니다.

4. Canonical ID와 version을 분리하는 이유는 무엇인가요?

정답과 해설

같은 artifact의 정체성은 유지하면서 내용의 변경 순서를 표현하기 위해서입니다. ID가 version마다 바뀌면 trace history가 끊깁니다.

5. 삭제된 canonical ID를 재사용하면 안 되는 이유는 무엇인가요?

정답과 해설

과거 evidence와 link가 새 의미를 가리켜 audit·change impact·test history가 오염되기 때문입니다.

6. Forward trace와 reverse trace의 차이는 무엇인가요?

정답과 해설

Forward trace는 goal·scope에서 screen·operation·entity·test로 가고, reverse trace는 test나 field에서 source requirement와 decision으로 돌아옵니다.

7. Orphan artifact란 무엇인가요?

정답과 해설

Source scope·flow·owner·acceptance 중 필요한 연결이 없어 왜 존재하고 어떻게 검증되는지 설명할 수 없는 artifact입니다.

8. Conflict와 missing link의 차이는 무엇인가요?

정답과 해설

Missing은 필요한 edge나 값이 없고, conflict는 같은 사건·field·state에 서로 다른 약속이 존재합니다.

9. 사용자 흐름에 필요한 최소 계약을 쓰세요.

정답과 해설

Actor, precondition, start, screen sequence, branch, failure, recovery, end outcome, channel, linked IDs입니다.

10. 화면 목록만으로 UI 명세가 충분하지 않은 이유는 무엇인가요?

정답과 해설

Loading·empty·editing·submitting·success·error·denied·conflict 상태와 trigger·visible result·allowed action·recovery가 없기 때문입니다.

11. 접근 가능한 interaction contract의 핵심은 무엇인가요?

정답과 해설

Keyboard path, focus order와 이동, accessible name·role·value, status announcement, error association, recovery focus입니다.

12. UI action과 API operation을 어떻게 연결하나요?

정답과 해설

Screen ID와 action ID를 operationId에 연결하고 method·path·actor·authorization·request·success·problem·idempotency·version을 맞춥니다.

13. 401·403·404를 문서에서 구분해야 하는 이유는 무엇인가요?

정답과 해설

Authentication·authorization·resource disclosure 경계가 다르며 UI recovery와 보안 의미도 달라지기 때문입니다.

14. JSON Schema 2020-12를 쓰더라도 별도 의미 계약이 필요한 이유는 무엇인가요?

정답과 해설

Schema는 구조와 제약을 표현하지만 product meaning·owner·unit·time zone·state transition·retention·authority를 모두 대신하지 않습니다.

15. RFC 9457 problem과 UI recovery의 연결 필드는 무엇인가요?

정답과 해설

Problem type·title·status·detail·instance·field issue를 UI state·message·draft preservation·focus·allowed next action과 연결합니다.

16. 409 conflict와 412 precondition failed를 합성 사례에서 어떻게 다루나요?

정답과 해설

현재 version과 request precondition의 차이를 problem으로 반환하고, UI는 최신값 비교·내 입력 보존·재시도 또는 복구 경로를 제공합니다.

17. 데이터 entity contract의 최소 범위를 쓰세요.

정답과 해설

Canonical name·meaning·identity·field type·required/null·enum·relationship·state transition·history·retention·provenance·owner·version입니다.

18. 현재값만 저장하면 어떤 연결 증거가 사라지나요?

정답과 해설

누가 언제 왜 상태를 바꿨는지, 어떤 version이 현재인지, conflict와 restore가 어떻게 발생했는지, retention 판단이 무엇인지가 사라집니다.

19. 공유 vocabulary registry가 관리할 여섯 종류를 쓰세요.

정답과 해설

Term, field, unit, time zone, enum·state, problem type입니다. Alias와 deprecated term도 관리합니다.

20. Semantic collision의 두 형태는 무엇인가요?

정답과 해설

같은 이름이 다른 의미를 갖는 경우와, 다른 이름이 사실상 같은 의미인데 연결되지 않은 경우입니다.

21. Acceptance matrix의 다섯 관점은 무엇인가요?

정답과 해설

Positive, negative, boundary, authorization, accessibility입니다.

22. Acceptance와 test evidence를 분리하는 이유는 무엇인가요?

정답과 해설

Acceptance는 관찰 가능한 기대이고 test는 그것을 확인하는 절차이며 evidence는 실행 결과의 위치와 version입니다. 셋을 섞으면 재현성과 책임이 흐려집니다.

23. Change impact record의 최소 필드를 쓰세요.

정답과 해설

Change source, affected IDs, compatibility, migration, tests, owner, status, TBR, version, decision authority입니다.

24. SemVer가 문서 변경의 의미를 자동으로 결정하지 못하는 이유는 무엇인가요?

정답과 해설

Major·minor·patch 규칙은 팀 계약에 따라 해석되어야 하며 schema·behavior·data migration·consumer impact를 직접 검토해야 하기 때문입니다.

25. 합성 package 세 version의 pass 수를 쓰세요.

정답과 해설

copy-paste-docs-v1은 6/24, linked-incomplete-pack-v2는 15/24, canonical-linked-pack-v3는 24/24입니다.

26. 합성 candidate의 artifact 구성을 쓰세요.

정답과 해설

PRD 1, flow 2, screen 6, API operation 7, data entity 5로 총 21개입니다.

27. Document Linkage Gate의 네 lane을 쓰세요.

정답과 해설

Intent·source, experience·interface, service·data contract, trace·change governance입니다.

28. 합성 candidate의 핵심 Gate 수치를 쓰세요.

정답과 해설

24/24 scenario, critical 100%, 12/12 control, 네 lane 6/6, canonical·source·semantic·UI·API·data·acceptance·change 100%, orphan 0, conflict 0, TBR 2입니다.

29. document_linkage_ready 가 뜻하지 않는 것은 무엇인가요?

정답과 해설

승인된 PRD·디자인·API·데이터 계약·architecture, production build·pilot·release, 예산·투자·계약·보안·접근성 인증을 뜻하지 않습니다.

30. M13-01에 넘길 최소 묶음은 무엇인가요?

정답과 해설

Product goal·scope·quality constraints, actor·flow·screen states, operation·error·authorization, entity·state·history·retention, acceptance·evidence, version·TBR·residual risk·change impact와 interface boundary입니다.

25. 공식 근거와 적용 경계

근거	이 매뉴얼에서 사용한 원리	적용 경계
ISO/IEC/IEEE 29148:2018	요구사항 engineering information item	원문 구매·조직 tailoring은 별도
ISO/IEC/IEEE 15289:2019	Life-cycle information item의 목적과 내용	특정 문서 세트를 강제하지 않음
ISO 10007:2017	Configuration identification·status·change control	조직 CM plan을 대체하지 않음
ISO/IEC 11179-6:2023	Metadata item identification·naming·registration	합성 ID namespace는 교육용

근거	이 매뉴얼에서 사용한 원리	적용 경계
ISO/IEC 11179-4:2004	Data definition 작성 원리	전체 표준 적합성 claim 아님
OpenAPI Specification	Operation·request·response·security contract	구현·보안 검증을 자동 보장하지 않음
JSON Schema 2020-12	JSON instance 구조·validation vocabulary	Product semantics·lifecycle을 대체하지 않음
RFC 9110	HTTP method·status semantics	Application authorization policy는 별도
RFC 9457	HTTP API problem details	UI recovery 설계는 별도 연결
RFC 6901	JSON document 내부 위치 참조	Canonical artifact ID와 같은 개념이 아님
WAI-ARIA 1.2	Name·role·value·state semantics	Native HTML 우선·적합성 claim 별도
WAI-ARIA APG	Accessible interaction pattern 예시	Informative guidance·실제 사용자 검증 필요
JSON-LD 1.1	Linked data context·identifier 사고	이 매뉴얼이 JSON-LD 도입을 요구하지 않음
Semantic Versioning 2.0.0	Version 의미를 공개 API contract로 표현	문서 impact를 자동 판정하지 않음
GOV.UK · Start by learning user needs	User need에서 service decision으로 trace	정부 service 절차를 그대로 복제하지 않음

25.1 이 매뉴얼의 합성 규칙

- 실제 개인정보·참여자·meeting 원문·requirement 원문·secret·production payload를 사용하지 않습니다.
- 외부 network·production resource·live side effect·public pilot·release를 사용하지 않습니다.
- 실제 document·architecture·budget·investment·contract·security·accessibility approval claim을 하지 않습니다.
- 24 scenario·12 control·21 artifact·8 state·7 operation·5 entity·57 audit는 이 합성 사례의 교육 계약이며 보편 표준 임팩트가 아닙니다.

26. 다음 단계 · M13-01 handoff

M13-01에서는 연결된 문서 package를 source로 삼아 규모·위험·변경 가능성·운영 책임에 맞는 architecture option을 비교합니다.

M12-04 product intent + scope + quality constraints

- actor + flow + screen states
- operation + authorization + problem + schema
- entity + state + history + retention
- acceptance + test evidence + change impact
- interface boundary + residual risk + TBR
- M13-01 architecture options + trade-off + decision record

Handoff item	M12-04 source	M13-01 use
Product goal·scope	PRD-TFU-001	architecture quality priorities
Flow·screen states	FLOW-01·02·SCR-01~06	component·channel boundary
Service contract	OP-01~07·problem registry	interface·integration options
Data lifecycle	ENT-01~05	storage·consistency·retention choices
Quality·access·security	UI/API/data contract	quality attribute scenarios
Evidence·change	AC·trace·impact·TBR	risk·trade-off·ADR input

완료 문장: 나는 PRD·flow·screen·UI state·API operation·problem·data entity·lifecycle·vocabulary·acceptance·test·change를 authoritative source와 canonical ID로 연결하고 orphan·conflict를 계산하며, **document_linkage_ready** 가 실제 architecture·build·release 승인이 아님을 구분할 수 있습니다.

부록 A. 용어집 300 학습 지도

문서 연결 용어집 300은 15개 묶음으로 구성됩니다. 한 번에 20개만 읽고 합성 사례의 ID를 붙입니다.

묶음	핵심 질문
01~03	Authority·PRD·canonical ID를 설명하는가
04~06	Trace·flow·UI state·access를 잇는가
07~09	API problem·schema·data lifecycle·semantics를 맞추는가
10~12	Acceptance·evidence·orphan·conflict·change를 계산하는가

목록	핵심 질문
13~15	Version·Gate·M13-01 handoff와 승인 경계를 구분하는가