

A clearer route to a safer launch.

A production-readiness audit of Asterpath, a fictional AI-assisted planning application.

- Product behaviour
- Engineering foundations
- Operating confidence

01	02	03	04
Understand	Verify	Prioritise	Retest

Fictional reviewed product Asterpath web application Staging candidate AP-RC-0911	Sample edition 11 September 2026 · Revision 4 17 assessment chapters · 70 pages
--	--

Fictional illustrative sample. Asterpath, its product, findings, tool results, screenshots, measurements, code extracts and estimates are invented. No Asterpath application was inspected and no scans were run against it. This demonstrates the deliverable; it is not a client result.

READING GUIDE

Find the area you care about.

Read the decision first for the immediate priorities. Each chapter explains the topic, the available evidence and the limits of the assessment.

START HERE	PAGE	FINISH WITH	PAGE
Executive decision	3	Action plan	64
Scope and evidence	4	Verification and handover	65
Coverage at a glance	5-6	Evidence and pack guide	66-67
Evidence-bounded scorecard	7	Terms and references	68-70

ASSESSMENT CHAPTERS		PAGES
01	Critical user journeys	8-10
02	Codebase, documentation & ownership	11-15
03	Authentication, accounts and permissions	16-18
04	Security and abuse protection	19-21
05	Payments & entitlements	22-24
06	UX, interface & visual consistency	25-29
07	Speed & performance	30-32
08	Backend, APIs and background processing	33-35
09	Database design and data integrity	36-38
10	Privacy and data lifecycle	39-41
11	Observability & support	42-45
12	Testing & change safety	46-49
13	External services & operating costs	50-53
14	Accessibility	54-56
15	Browser, device & language support	57-58
16	Search & AI discoverability	59-60
17	Release & recovery	61-63

EXECUTIVE DECISION

Do not launch yet.

Fix paid-access recovery, then verify a new candidate before release.

A customer can pay successfully and remain on the free plan after an interrupted entitlement update. That reproduced scenario is the single launch blocker, F01.



Foundations worth keeping

Server-owned identity and access controls, separate staging data, structured logs, a reproducible build and a versioned release handbook provide a useful base. These strengths remain bounded by the evidence described here.

Where confidence breaks down

Payment recovery is incomplete. AI retries lose user input, mobile page loading is slow in the illustrated lab run, and a failed alert delivery can remain unnoticed. Passing tests do not cover all of those scenarios.

What needs to happen next

WHEN	REQUIRED OUTCOME
Before launch	Resolve F01; name a new candidate, prove payment recovery and promote that verified artifact.
Before growth	Address test gaps, recoverable errors, slow loading, alert delivery, keyboard focus and incomplete AI cost attribution: F02, F03, F05-F08.
Planned improvement	Clarify competing workspace actions: F04.

Finding profile: 1 High, 6 Medium and 1 Low. Severity describes consequence and likelihood; the launch decision considers the agreed release boundary. Effort estimates are illustrative, not a quote.

Decision boundary

This recommendation applies only to AP-RC-0911, the four agreed journeys and the fictional evidence streams in this report. Unexamined systems remain unverified.

SCOPE AND EVIDENCE

What this assessment represents.

An audit is only useful when the reader can see which product, scenarios and environments its conclusions describe.

SCOPE ITEM	FICTIONAL ASSESSMENT BOUNDARY
Product and audience	English-language planning web app for individual customers; paid subscriptions and an AI plan-generation feature.
Reviewed version	AP-RC-0911, a fictional immutable staging candidate dated 11 Sep 2026.
Journeys	Sign in and manage a session; subscribe and unlock; generate a plan; export a plan.
Surface sample	Public landing page, sign-in, workspace, generation form, subscription flow and export dialog.
Runtime sample	Desktop Chrome, Firefox and Safari; iPhone Safari portrait/landscape. Exact illustrative profiles appear on page 57.
Operational sample	Read-only payment events, AI usage totals, crash/alert settings and a staging release record.
Excluded	Production changes, real charges, penetration testing, load testing, legal or accessibility certification, and real-user research.

Five kinds of evidence

EVIDENCE STREAM	WHAT IT CAN ESTABLISH
Source review	How the inspected code and configuration are intended to behave.
Automated checks	What a named command and its discovered tests actually checked.
Runtime validation	What happened for the sampled environment, account and scenario.
Provider consoles	What the inspected provider settings and event records showed.
Historical evidence	A dated previous change and verification record; not a substitute for current checks.

Illustration, not fabricated proof

Every result in this sample is an invented example. In a client engagement these entries are replaced with actual versions, dates, outputs and observations. Unavailable evidence is recorded explicitly.

COVERAGE AT A GLANCE / PRODUCT AND FOUNDATIONS

See what the evidence supports.

Coverage describes what was examined, not whether the area passed. New explanatory depth does not imply additional tests were completed.

ASSESSMENT AREA	COVERAGE	FINDING / DECISIVE LIMIT
01 Critical journeys	Partially reviewed	F01, F05, F08; selected recovery paths incomplete.
02 Codebase, quality and AI	Partially reviewed	Structure and setup sampled; independent handover and usable exit unverified.
03 Authentication and accounts	Partially reviewed	Session and selected isolation checks; OAuth, linking, recovery and full role lifecycle unverified.
04 Security and abuse	Partially reviewed	Trusted boundaries sampled; native security scans and wider abuse testing not performed.
05 Payments	Reviewed	F01: paid access can be stranded after a failed write.
06 UX and visual consistency	Reviewed	F04, F05; expert inspection, no real-user study.
07 Performance	Partially reviewed	F06; lab evidence only; field traffic and peak capacity unverified.
08 Backend and APIs	Partially reviewed	Payment/generation paths sampled; broader jobs, business rules and concurrent requests unverified.
09 Database and integrity	Partially reviewed	Ownership boundaries sampled; complete schema, constraints, query plans and concurrent-write behaviour unverified.

Read the disposition literally

- Reviewed:** agreed checks completed.
- Partially reviewed:** relevant evidence is missing.
- Not verified:** evidence cannot support a conclusion.
- Not applicable:** outside intended capabilities.

One home for each finding

A payment failure can affect a journey, a database write and a test. F01 remains one payment finding, cross-referenced elsewhere. The action plan still contains eight findings.

COVERAGE AT A GLANCE / RUNNING AND REACHING CUSTOMERS

Make the less visible risks explicit.

A product can work in a demonstration while delivery, recovery or support remains uncertain. These boundaries need named evidence and owners.

ASSESSMENT AREA	COVERAGE	FINDING / DECISIVE LIMIT
10 Privacy and lifecycle	Partially reviewed	Retention/deletion sampled; processor completion and restore drill unverified.
11 Observability and support	Partially reviewed	F07; alert delivery can fail silently; transactional email and sensitive support actions unverified.
12 Testing	Reviewed	F03; 202 illustrated passing checks omit key recovery scenarios.
13 Services and costs	Partially reviewed	F02; one invoice period; output quality, provider outage and peak demand unverified.
14 Accessibility	Partially reviewed	F08; selected manual checks, no complete conformance review.
15 Browser/device/language	Partially reviewed	Named profiles only; Android, tablet and translations unverified or inapplicable.
16 Search and AI discovery	Partially reviewed	Public-page technical checks; ranking and agentic tasks unverified.
17 Release and recovery	Partially reviewed	Version records sampled; full restore and independent handover unverified.

Useful evidence to obtain next

FOUNDER CONCERN	WHERE IT IS MADE VISIBLE
Can someone else run or recover the business?	Ownership and exit, page 15; recovery, page 63.
Can hidden work fail or charge twice?	Backend recovery, page 34; data consistency, page 37.
Will messages arrive and AI output be useful?	Delivery and support, page 45; output quality, page 53.

These follow-ups describe missing confidence, not newly discovered defects. No additional scans, customer research, load exercises or provider-console access were invented for this edition.

EVIDENCE-BOUNDED SCORECARD

Strengths and gaps, without an average.

The eight maturity dimensions summarise the 17 chapters. Chapter coverage is separate from maturity and finding severity.

Architecture and ownership

Reproducible setup and shared instructions; handover only partly exercised.

Ch 02, 08-09 · Medium evidence confidence

3/5

Managed

Critical journeys and user experience

F01 blocks paid access; recovery, focus and performance gaps also remain.

Ch 01, 06-07, 14-16 · High evidence confidence

2/5

Fragile

Security and data protection

Identity and isolation sampled; restore and wider attack scenarios unverified.

Ch 03-04, 09-10 · Medium evidence confidence

3/5

Managed

Payments and entitlements

F01 blocks launch: failed entitlement writes lack durable recovery.

Ch 05 · High evidence confidence

2/5

Fragile

Test and change safety

F03: 202 passing checks omit revenue and AI recovery scenarios.

Ch 12 · High evidence confidence

2/5

Fragile

Release and rollback

Candidate provenance documented; complete recovery rehearsal missing.

Ch 17 · Medium evidence confidence

3/5

Managed

Observability and incident response

F07: a failed alert delivery can remain unnoticed.

Ch 11 · High evidence confidence

2/5

Fragile

Cost and provider operations

F02: £77 in provider spend cannot be attributed to product activity.

Ch 13 · High evidence confidence

2/5

Fragile

1

Uncontrolled

2

Fragile

3

Managed

4

Resilient

5

Failure-tested

How to interpret the numbers

A launch blocker normally caps its affected dimension at 2. Missing material runtime or console evidence normally caps a dimension at 3 or makes it Not verified. No overall average is calculated: strong documentation cannot cancel a payment blocker.

Can customers complete the important tasks?

Critical journeys connect individual features into the outcomes customers expect. We consider the normal path, legitimate alternatives, failures and recovery.

Partially reviewed

Illustrative evidence: I01-I05

Four agreed journeys; runtime sampling does not cover every device, account history or provider failure.

J01 / Sign in and session

Partially verified. Sign-in, expiry and sign-out behave as expected in the sampled profiles. A dated manual recovery check exists; long-lived sessions across every supported browser remain outside this sample.

J02 / Subscribe and unlock

Finding raised: F01. Normal sandbox purchase unlocks paid access. An interrupted entitlement write leaves a paid account free. Duplicate and delayed events require explicit recovery evidence.

J03 / Generate an AI plan

Finding raised: F05. Normal generation completes. A timeout discards the customer's submitted brief; retry requires re-entry. Provider errors and cost are assessed in Chapters 11 and 13.

J04 / Export a plan

Finding raised: F08. A PDF is produced in the sampled successful flow. Keyboard focus returns to the page body after closing the export dialog, making the next step difficult to locate.

What is already working

- The product offers a clear first-use path from an empty workspace to a first plan.
- Writes use pending states to discourage duplicate submissions in the normal flow.
- Successful payment and generation states communicate the resulting access or saved output.

Where to read the detail

Payment recovery: page 23. Interface recovery: page 28. Keyboard focus: page 56. Their identifiers remain the same throughout the report.

The happy path is only the starting point.

The same journey can be supported by source review and a happy-path test while its recovery behaviour remains unverified.

JOURNEY	HAPPY	ALTERNATE	UNHAPPY	RECOVERY
J01 Sign in	Automated + runtime	Automated	Automated + runtime	Dated manual
J02 Subscribe	Automated + runtime	Partial automated	Runtime finding F01	Missing automation; F01
J03 Generate	Automated + runtime	Partial automated	Runtime finding F05	Dated manual; F05
J04 Export	Automated + runtime	Dated manual	Not verified	Focus finding F08

“Automated” means a relevant behaviour assertion is represented in the fictional check set. It does not certify the whole journey.
“Dated manual” means recorded validation against AP-RC-0911.

Evidence by journey

JOURNEY	SOURCE REVIEW	AUTOMATED CHECKS	RUNTIME VALIDATION	PROVIDER CONSOLES
J01	Reviewed	Reviewed	Partially reviewed	Not verified
J02	Reviewed	Partially reviewed	Reviewed; F01	Partially reviewed
J03	Reviewed	Partially reviewed	Partially reviewed; F05	Partially reviewed
J04	Reviewed	Partially reviewed	Partially reviewed; F08	Not applicable

Example: paid access after interruption

The test must establish a completed entitlement exactly once after recovery, with a visible failure before recovery. A provider returning success or a unit test passing is insufficient evidence that the customer received access.

Known limits

- No production charges or mutations are represented.
- The sample is an expert inspection, not a study with representative customers.
- Zero evidence is recorded as unknown; it never earns a pass.

Define completion from the customer's side.

A successful request is one event inside a journey. Completion means the customer has the intended result and can understand what to do next.

Partially reviewed

Illustrative evidence: I03-I05, I10

This interpretation uses the same four journey records. Additional failure scenarios are evidence follow-ups, not new completed tests.

JOURNEY BOUNDARY	CUSTOMER OUTCOME TO ESTABLISH	WHAT REMAINS OPEN
Session → permitted action	A valid customer can act; an expired or signed-out session cannot continue silently.	Long-lived and revoked sessions across the full support matrix are not verified.
Payment → usable access	The paid account receives the correct access, including after interruption.	F01 prevents this outcome; F03 lacks the complete recovery checks.
Provider response → saved plan	A usable result is saved, or the customer keeps enough information to retry safely.	F05 loses the brief after timeout; broader provider outage behaviour is unverified.
Export → continuing work	The file is produced and the customer returns to a clear, operable place.	F08 loses focus; unsuccessful export recovery has not been established.

Choose the next scenarios by consequence

Prioritise the point where external and application state can disagree, where work can be lost, or where an irreversible step occurs. Start with the controlled paid-access failure, then input preservation and dialog recovery. This targets the existing findings rather than extending every journey equally.

Evidence to preserve

Candidate, account state, initiating action, expected result and observed final state.
Connect the app record to a provider event when that boundary matters.

A useful recovery result

The same customer can continue without unexplained duplicate work, an incorrect entitlement or re-entering information the product could retain.

Keep coverage separate from the release decision

An untested scenario is marked Not verified. F01 is the demonstrated launch blocker; a long list of untested possibilities does not create additional confirmed defects.

Can this product be changed with confidence?

A maintainable codebase makes important behaviour understandable and changes verifiable. We assess the structure, setup and ownership that another engineer or AI agent would depend on.

Partially reviewed

Illustrative evidence: I01, I02, I11

Source and clean setup represented; a complete independent handover was not exercised.

AREA	ILLUSTRATIVE OBSERVATION	DISPOSITION
System boundaries	Client, server functions and shared contracts are separated; server code owns identity and paid access.	Strength
Reproducible setup	A locked package installation, environment catalogue and named build command reproduce staging.	Strength
Ownership	Repository, hosting, payment and AI-provider ownership are documented with recoverable team access.	Reviewed
Code health	Typecheck, lint and build pass. Knip flags 12 candidates; four unused exports survive manual triage.	Cleanup, no material finding
Change hotspots	Subscription handling crosses client, server and provider events; the boundary needs joint verification.	See F01 / F03
Continuity	Setup steps are exercised, but another operator has not completed a full release-and-recovery handover.	Not verified

How code quality is judged

Large files, unfamiliar libraries or generated code are not automatically defects. We look for concrete consequences: duplicated business rules, unclear state ownership, fragile changes, unbounded work or important checks that cannot be run.

Also in this chapter / AI-agent setup

Can a new agent find the right instructions, understand the boundaries and select the correct checks?

Also in this chapter / Repeatable workflows

Can routine testing, releases and AI content generation be repeated without relying on one person’s memory?

The next page examines code structure. The scanner inventory is on page 46; raw counts remain separate from validated risks.

Will the next change stay understandable?

Code quality is about how safely the product can change. We look for clear responsibilities, understandable data flow and business rules with an identifiable owner.

Partially reviewed Illustrative evidence: I01, I02, I04

Selected source paths and tests; historical change friction and the whole dependency graph are not verified.

STRUCTURAL QUESTION	ILLUSTRATIVE OBSERVATION	MEANING
Modularity and cohesion	Session, planning and payment concerns have separate modules. Each can be located from the system map.	Useful boundaries; keep them.
Coupling and ownership	The server owns paid access, but provider acknowledgement and persistence need coordinated handling.	Trace this boundary; F01 is the material failure.
Duplicated business rules	Shared contracts and a single entitlement policy are represented in the sampled source.	No competing policy found in the sample.
Control flow and side effects	Payment work continues after a success response; its failure cannot change the response.	The concrete risk is visible in the extract on page 24.
Readability and change safety	Names identify outcomes; tests cover selected state changes. Revenue recovery is missing from the gate.	F03 describes the assurance gap.
Dead and obsolete code	Four unused exports remain after scanner triage.	Cleanup; no demonstrated launch impact.

How we recognise tangled code

A small change requires editing unrelated areas; the same rule disagrees across copies; branches conceal important failure paths; or a function mixes several responsibilities so its outcome is hard to establish. We follow a representative change through the code and its checks.

A boundary should earn its place

A small, cohesive module can be easier to maintain than many tiny wrappers. File length, framework choice, nesting counts and scanner scores are signals to investigate, not automatic findings. Recommend refactoring when it reduces an evidenced risk.

The payment example illustrates an existing finding, F01. It does not create a second architecture finding or inflate the finding total.

Give the agent a reliable starting point.

Project instructions should help a coding agent understand the repository, follow its conventions and verify work without repeated explanation from the owner.

Reviewed

Illustrative evidence: I11

Illustrative fresh-session instruction discovery and a safe validation-command selection exercise; no autonomous release was attempted.

CHECK	ILLUSTRATIVE RESULT
Supported entry points	Root AGENTS.md supplies shared guidance; CLAUDE.md imports it and adds tool-specific notes. Both tools are used in this fictional team.
Useful context	The root guide describes product purpose, code locations, server-owned permissions and links to current architecture decisions.
Verification commands	Setup, lint, typecheck, tests and build instructions match the package scripts. Integration and end-to-end scope are stated explicitly.
Concise guidance	The shared entry file has 96 lines. That is a description, not a quality score. Detailed release steps live in a handbook.
Effective loading	The fresh-session exercise identifies the root guide and the payment-specific instructions in the intended working context.
Enforced boundaries	Test/release controls and provider permissions carry the important protections. Written instructions explain them.

What the official guidance contributes

OpenAI recommends practical repository context, working commands, constraints and completion criteria. Anthropic recommends concise, specific CLAUDE.md guidance and a target under 200 lines per file. We use roughly 100 lines only as a starting editorial aim; relevance and accuracy decide the assessment.

A useful acceptance exercise

A fresh agent can locate the affected subsystem, identify applicable instructions, explain the consequential boundaries and select the relevant checks. Report any missing access, broken commands or conflicting guidance.

Sources: OpenAI AGENTS.md and best practices; Anthropic memory and configuration guidance. Full links on page 69.

Make routine work repeatable.

A handbook, script or skill turns a repeated task into a process someone else can follow. Automation is useful when it reduces mistakes and preserves review at consequential steps.

Partially reviewed Illustrative evidence: I11, I12

Testing and generation dry-runs sampled; a complete release recovery rehearsal remains unverified.

WORKFLOW	DOCUMENTED AND AUTOMATED	EVIDENCE BOUNDARY
Testing	One handbook maps changed areas to checks; a script runs the standard validation set.	Discovery verified; scenario gaps remain F03.
Release	A release handbook names candidate, checks, migrations, smoke tests and rollback. A script records the candidate.	Staging record inspected; full rehearsal not verified.
AI content generation	A scoped skill links inputs, prompt/model version, cost cap, output checks and human acceptance.	Synthetic dry-run only; no public content published.
Incident response	A short runbook links symptoms, logs, provider events and escalation owner.	Alert delivery weakness is F07.

Generation workflow sampled

1 Prepare Validate source and synthetic inputs.	2 Generate Record version, attempts and budget.	3 Review Check output and flag uncertain cases.	4 Accept Human review before consequential use.
--	--	--	--

What makes an automation dependable

- Inputs, prerequisites, outputs and the responsible owner are explicit.
- Retries, partial completion and resuming a run do not silently duplicate work.
- Failure is visible; logs and retained outputs allow investigation.
- Dry-run, cost and access boundaries fit the task.
- Human review remains clear for publication, production data or other consequential changes.

Proportionate to the product

A short, accurate handbook may be enough. Skills, scheduled jobs or elaborate pipelines are recommended only where a repeated workflow benefits from them.

Keep the business operable without one person.

Ownership means more than paying the invoice. The company needs usable access, understandable records and a practical route to another operator or provider.

Partially reviewed

Illustrative evidence: I01, I11, I12

Ownership records and clean setup are represented. Independent operation, full account recovery and a usable data export have not been demonstrated.

CONTINUITY BOUNDARY	WHAT THE SAMPLE SUPPORTS	WHAT WOULD ESTABLISH CONFIDENCE
Company control	Repository, hosting, payment and AI-provider ownership are documented.	Verify authorised team access and recovery contacts for every critical account, including domain and database.
Another operator	Setup instructions and selected command discovery are exercised.	A second operator completes an agreed release, diagnosis and recovery task using the handbook.
Data and provider exit	The system map identifies dependencies.	Export representative data with relationships and attachments intact; establish how another system could use it.
Knowledge and decisions	Shared instructions link architecture and workflow guidance.	Locate the current business rules, migration history, provider contracts and consequential configuration.

Choose a practical handover exercise

Ask another authorised operator to trace a paid-access problem, identify the relevant deployment and explain the recovery path. Then exercise a contained task. Record missing access or undocumented decisions; reading a handbook alone does not establish independence.

Exit should preserve meaning

A file download is insufficient if identifiers, relationships, time zones or attachments are lost. Verify a representative exported record can be interpreted outside the original provider.

Keep the ownership boundary proportionate

Document who controls the accounts and how access is recovered. A small product does not need an elaborate enterprise process to avoid dependence on one personal login.

No provider migration or new recovery exercise was performed for this sample. Release and recovery evidence remains on page 63.

Can the right person get in—and stay in control?

Authentication establishes who someone is. Permissions decide what that person may do. Account lifecycle covers the changes between joining, recovering access and leaving.

Partially reviewed

Illustrative evidence: I01, I03

Existing session and selected isolation records only. OAuth configuration, account linking and a complete recovery lifecycle are not verified.

ACCOUNT AREA	ILLUSTRATIVE EVIDENCE	ASSESSMENT LIMIT
Sign-in, sign-out and expiry	Sampled staging sessions behave as expected in the named profiles.	Long-lived sessions and every browser/account history are not covered.
Trusted identity	The server derives the acting user from a verified session.	All endpoints and token-handling paths have not been inventoried.
Record permissions	Selected plan read/write tests reject another customer's access.	The full actor, record and action matrix remains incomplete.
Sensitive account changes	A dated manual recovery record exists.	Email changes, recovery links, revoked sessions and account-linking transitions need further evidence.
OAuth and invitations	No detailed provider or invitation evidence is represented.	Confirm which features exist before assessing redirects, linking, invite expiry and roles.

Why founders should care

A customer locked out of a paid account needs recovery that preserves ownership. Convenient recovery must not make it easy for someone else to take over that account.

What is already working

Server-derived identity and sampled cross-customer isolation are useful foundations. They do not establish that every account lifecycle path is safe.

A separate chapter with a clear boundary

Payment state belongs to Chapter 05; retention and deletion completion belong to Chapter 10. Their permission boundaries are cross-referenced here. No additional authentication finding is asserted.

Check the boundary when the interface is bypassed.

A permission check decides whether this account may perform this action on this record. The server must make that decision even when a request does not come from the expected screen.

Partially reviewed

Illustrative evidence: I01, I03

Selected plan read/write isolation and privileged-route evidence; no complete role-policy matrix or penetration test.

TRUST BOUNDARY	REPRESENTED CONTROL	EVIDENCE LIMIT
Session → acting identity	The server derives identity from the verified session.	Sensitive account changes and broad revocation scenarios are not fully exercised.
Customer → another customer's plan	Selected tests reject reading or writing the other customer's record.	These checks do not cover every route or policy combination.
Customer → administrative action	Normal routes are separate from administrative actions; server-side role checks protect privileged work.	No exhaustive adversarial review of administrative capabilities.
Client → paid feature	The server owns the entitlement decision.	Correct authority does not repair the failed entitlement update: F01.

Why the last row matters

The payment defect leaves a legitimate customer without access. The sampled evidence does not show unauthorised access. We retain F01 as a payment-recovery finding and avoid presenting it as a demonstrated permission bypass.

Extend coverage when the product boundary changes

- Adding teams or sharing creates new combinations of actor, role, record and action.
- Account recovery or role changes need checks for old sessions and cached permissions.
- A new privileged endpoint needs its own server-side decision and a meaningful denied-access check.

Assessment conclusion

Server-owned authority and sampled isolation are useful strengths. Broader policy coverage remains an explicit follow-up, with no additional confirmed security finding in this sample.

Check what happens when access changes.

The difficult account cases often occur after the first login: a lost device, changed email address, linked provider or removed administrator.

Not verified Illustrative evidence: I01, I03 provide limited context

The following are assessment questions and proposed acceptance evidence. They are not additional completed tests or confirmed weaknesses.

LIFECYCLE SCENARIO	WHAT A DEPENDABLE RESULT WOULD ESTABLISH
Password or magic-link recovery	The intended account is recovered; links expire and cannot be reused; old sessions follow an explicit revocation policy.
OAuth and account linking	The callback belongs to the initiated flow; redirects are constrained; identities cannot be joined solely on an untrusted matching email claim.
Email or role change	Consequential changes require appropriate confirmation; existing sessions and cached permissions reflect the new access.
Invitations and team removal	If teams exist, invitations have bounded authority and expiry; removing a member withdraws access to relevant records.
Admin access and recovery	Privileged accounts have suitable protection and recovery; support actions are authorised, attributable and bounded.

Inspect the permission decision in context

Teaching example only · not Asterpath source or a finding

```
1  # A request body is not proof of identity or permission.
2  actor = verify_session(request)
3  plan = load_plan(request.plan_id)
4  require_permission(actor, "update", plan)
5  apply_allowed_changes(plan, request.changes)
```

This sketch names the decisions rather than implementing them. A real review verifies token validation, tenant boundaries, allowed fields and the behaviour of the actual data access path.

Use the product's real account model
A single-customer product may have no teams or invitations. Record that as Not applicable after confirming the design; do not infer a missing feature from a generic checklist.

Protect the boundaries attackers can reach.

Security review follows exposed entry points, sensitive operations, secrets and dependencies. It considers how a legitimate feature could be misused as well as how a technical control could fail.

Partially reviewed

Illustrative evidence: I01, I03

Source, selected isolation tests and staging sessions; no penetration test or exhaustive adversarial coverage.

AREA	WHAT THE FICTIONAL EVIDENCE SUPPORTS
Account protection	Identity, session and permission evidence has its own chapter, starting on page 16.
API and data boundaries	Selected isolation tests reject cross-customer access. The complete endpoint and policy inventory remains unverified.
Input and exposed routes	Validation, unsafe redirects, uploads and public route exposure need explicit scope. No exhaustive review of those paths is represented.
Secrets	Privileged provider credentials are represented as server-only. No secret values are included in the report.
Dependencies and configuration	Locked dependency/configuration evidence was inspected; broader supply-chain investigation is outside this sample.
Abuse controls	Per-account limits exist for generation. Coordinated abuse and high-volume attack behaviour were not tested.

Strength to preserve

Identity and permission decisions are enforced on the server. Hiding a button in the client is not treated as access control.

No material finding in the sampled checks

This describes the inspected paths only. It is not a statement that the application is secure in every scenario.

Follow-up evidence worth obtaining

- Broaden role and policy coverage when new roles or sharing features are added.
- Recheck sensitive account changes and session revocation against the intended account lifecycle.
- Keep dependency/advisory review tied to the actual release candidate.

Consider misuse of an otherwise valid feature.

Abuse can exploit an intended capability: generating expensive work, submitting oversized files or repeatedly attempting account recovery. Security and capacity controls need to fit those entry points.

Partially reviewed Illustrative evidence: I01, I03

Per-account generation limits and selected permission checks are represented. No active attack, load test or exhaustive route review was performed.

BOUNDARY TO EXAMINE	EXISTING EVIDENCE OR FOLLOW-UP
Public and authenticated APIs	Selected identity and isolation checks exist. Inventory every route, including edge functions, webhooks and administrative endpoints.
Inputs and file handling	Full payload, upload and download validation is not verified. Where present, assess allowed types, sizes, ownership and safe processing.
Resource abuse	Per-account generation limits exist. Shared quotas, parallel attempts and costly anonymous endpoints remain unverified.
Browser and origin controls	Verify the controls applicable to the real session design. CORS alone does not establish who may perform an operation.
Sensitive information exposure	Server-only credentials and sampled log redaction are represented. Wider error, artifact and history exposure remains unverified.

Connect a technical weakness to an outcome

A suspicious route or scanner alert becomes a report finding only when the evidence establishes the affected path, mechanism and consequence. An unrestricted expensive operation might create an operating-cost risk; an ownership bypass could expose another customer’s data. They require different evidence and remedies.

Security checks cannot establish business correctness

A request can be authenticated and still deduct the wrong credits or produce duplicate work. Chapters 08 and 09 assess those backend and data behaviours.

Control the follow-up scope

Use agreed test accounts, synthetic records and bounded inputs. The absence of active attack testing stays visible and is not converted into a security pass.

Know what the release depends on.

Dependencies, build configuration and provider credentials extend the application’s trust boundary. Source inspection and vulnerability scanning contribute different evidence.

Partially reviewed

Illustrative evidence: I01, I02

Locked dependency/configuration evidence and server-only credential placement are represented. Native vulnerability and secret-scan exports do not exist for Asterpath.

AREA	SUPPORTED CONCLUSION	UNVERIFIED EVIDENCE
Dependency identity	A lockfile helps identify the versions used by the illustrated build.	No current advisory database or vulnerability scan was run against this fictional product.
Secret placement	Privileged provider credentials are represented as server-only.	No complete repository-history, artifact or secret-rotation scan.
Configuration	The inspected configuration belongs to the named staging candidate.	Wider deployed infrastructure and production exposure are outside this sample.
Generation abuse	Per-account limits exist.	Coordinated abuse and high-volume attacks were not tested.

What scanner evidence would add

A real dependency scan ties a finding to a package, version, advisory and the application’s use of it. A secret scan needs controlled handling and validation; a candidate match is not proof that a credential is live. Source-analysis results also need a reachable mechanism and consequence before becoming report findings.

Tooling represented honestly

SonarQube and JFrog Xray are Not run here. Their edition, licensing and export requirements appear in the attached tooling notes.

Useful triage outcome

Classify a candidate as a material issue, maintenance item, false positive or unresolved question. Record the reason and the reviewed version.

No invented clean bill of health

A locked build and server-only credentials do not imply zero vulnerabilities. The absence of scanner evidence remains visible in the run register.

Payment should produce the right access.

Entitlements are the features or limits a customer is allowed to use. We follow the relationship between the payment provider’s records and the access granted by the application.

Reviewed

Illustrative evidence: I04

Fictional sandbox only; no live charges. The failed-write recovery scenario raises F01.

LIFECYCLE SCENARIO	ILLUSTRATIVE OBSERVATION
Successful subscription	Payment succeeds and the server grants paid access.
Cancellation or pending purchase	The interface explains the state and does not falsely confirm access.
Restore / sign in again	Existing provider identity reconnects to the sampled paid account.
Duplicate or delayed event	A stable event identity exists, but complete recovery assertions are missing. See F03.
Failed entitlement update	Provider receives success before the access write is durable; the customer stays free. See F01.
Refund / expiry	Sampled state transitions remove access according to the fictional product rules.

The failing path

1 Payment Provider confirms purchase.	2 Event Subscription event arrives.	3 Early success Success is returned too soon.	4 Write fails Access is never granted.
--	--	--	---

Customer outcome
Paid externally, free inside the product. Support must manually reconcile the mismatch. The next page shows the complete finding and the evidence needed to close it.

One underlying defect, one finding: F01 is linked from journeys, testing and observability without being counted again.

05 / PAYMENTS & ENTITLEMENTS / FINDING F01

A paying customer can remain on the free plan.

The payment provider and the application disagree after an interrupted entitlement update.

F01 **HIGH** Open · 11 Sep 2026

Acknowledgement precedes the durable access update

OBSERVED IN THE FICTIONAL ASSESSMENT

In the invented staging scenario, the payment event returns success before the entitlement write completes. A controlled database-write failure leaves payment confirmed while application access remains free.

CONSEQUENCE / IMPACT

A customer pays but receives no value. Repeated incidents create support work, refunds and damaged trust at the point of conversion.

LIKELIHOOD / CONFIDENCE

Plausible whenever the write fails after acknowledgement; the scenario is reproduced in the illustrative evidence. Evidence confidence: High.

Evidence: I04: event response, entitlement state and sandbox customer journey, 11 Sep 2026. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Make failed grants retryable, identify duplicate events consistently and reconcile provider payments that have no matching entitlement. The user should receive accurate progress or support information.

RETEST PASSES WHEN

Success, duplicate, delayed, out-of-order and failed-write scenarios converge on the correct final entitlement without duplicate effects. Failed paid grants recover; cancellation and expiry preserve the correct access state. Failures are visible and manual reconciliation is unnecessary.

Launch blocker: Yes. Resolve and verify a new candidate before release; a paying customer can otherwise receive no access.

Owner: Backend / release engineer · **Effort:** 1-2 days

Depends on: Payment sandbox, staging failure control and a candidate-linked regression check

Release decision

Do not launch yet. An implementation alone does not close F01: the defined recovery scenario must pass against the candidate that will be released.

05 / PAYMENTS & ENTITLEMENTS / CODE EVIDENCE FOR F01

The response succeeds before the work does.

The source reveals a missing durable handoff. Logging the error cannot recover the customer's access.

Invented TypeScript · server/payments/webhook.ts:42-46 · AP-RC-0911

```
42 export async function receivePaidEvent(event) {
43   // Signature and event validation occur before this excerpt.
44   void writeEntitlement(event).catch(reportError);
45   return Response.json({ received: true });
46 }
```

Why line 44 matters

The detached write can fail after the provider receives success. Logging does not durably retain work or schedule recovery. The customer can remain free after paying: F01.

Proposed approach · pseudocode · not implemented or verified

```
1 const accepted = await paymentInbox.acceptOnce({
2   eventId: verifiedEvent.id,
3   payload: verifiedEvent
4 });
5 return acknowledgeDurableReceipt(accepted);
```

What must sit behind this design

- Acknowledge only after durable receipt; failed receipt must remain visible and retryable.
- The worker tracks pending, failed and completed work, with bounded retries and an operator recovery path.
- Apply the correct subscription state transactionally; duplicates and delayed events must not restore cancelled or expired access.

Acceptance evidence, not just a nicer function

Inject a failed write and prove recovery without duplicate effects. Cover cancellation, expiry and event ordering too. See F01 and the test example on page 49.

Original fictional excerpt linked to I04. A client report uses a permission-appropriate minimal extract with its actual file, lines and reviewed version.

Make the next action easy to recognise.

UX review considers whether people can understand the interface, act confidently and recover from mistakes. Visual consistency supports those tasks while leaving room for the product’s intended style.

Reviewed

Illustrative evidence: I06

Heuristic inspection of the named screens, informed by Nielsen’s 10 usability heuristics; no representative-user study.

Illustrative strengths: typography, spacing and colour roles form a coherent identity; equivalent forms and controls reuse shared components. Navigation/back paths, contextual help, labels, validation and cancel behaviour support the sampled tasks. The competing action emphasis below and timeout recovery on page 28 are the material gaps.

Illustrative current workspace

Your plans

Create plan

Explore templates

Upgrade

Import

Four equally prominent filled actions compete for attention.

Illustrative direction

Your plans

Create plan

Explore templates

Import and account upgrades remain available with lower emphasis.

Visual structure in the sampled screens

LENS	ILLUSTRATIVE ASSESSMENT
Hierarchy and type	A small set of readable type roles separates titles, supporting information and controls; action emphasis remains F04.
Grouping and layout	Related labels, fields and help sit together. Content-appropriate widths and shared edges support scanning.
Colour and personality	Brand, neutral, status and interaction colours have distinct roles. Expressive colour can be coherent; useful secondary text stays readable.

What the evidence supports

This is an expert inspection. F04 is a clarity recommendation; F05 is a demonstrated failure in the fictional scenario. Aesthetic preferences alone do not establish lost conversions or unusability.

Action-hierarchy finding: page 26. Component and content observations: page 27. Recovery finding and code: pages 28-29.

06 / UX, INTERFACE & VISUAL CONSISTENCY / FINDING F04

Give important actions appropriate emphasis.

The workspace comparison on the preceding page illustrates competing emphasis. This finding separates the observed design from its unmeasured effect on customers.

F04 LOW Open · 11 Sep 2026

Four workspace actions have equal visual priority

OBSERVED IN THE FICTIONAL ASSESSMENT

Create, templates, upgrade and import use the same filled treatment in the initial workspace. Expert inspection identifies competing emphasis around the intended first task.

CONSEQUENCE / IMPACT

The intended next step takes more interpretation. No abandonment or conversion loss was measured.

LIKELIHOOD / CONFIDENCE

Visible on first use; the impact on actual customers remains an expert judgement. Evidence confidence: Medium.

Evidence: I06: fictional workspace hierarchy comparison. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Give the primary task appropriate emphasis and group secondary actions by purpose. Preserve the brand's expressive palette where it remains legible and meaningful.

RETEST PASSES WHEN

The reviewed screen has a clear action hierarchy and consistent colour roles. Validate first-task comprehension with representative users when available.

Launch blocker: No. All actions remain available; this is a clarity improvement.

Owner: Product / frontend designer · **Effort:** Half a day

Depends on: Agreed first-use task and shared button styles

Keep the recommendation contextual

One dominant action is often useful for a task-focused screen. A dashboard can support several important tasks; assess their grouping, context and relative emphasis rather than enforcing one button per page.

Consistency should survive real content.

A component system is useful when the same meaning produces a predictable interaction. Its quality becomes visible with long text, empty data, failures and constrained space.

AREA	WHAT THE FICTIONAL INSPECTION SUPPORTS	BOUNDARY
Shared controls	Equivalent buttons, form labels and dialogs use shared components and purposeful variants.	The equal-priority workspace actions remain F04.
Loading indicators	Inline pending controls and page placeholders differ for a reason; each represents ongoing work consistently.	A spinner count is not a quality measure.
Text and content	Sampled plan titles wrap without hiding the action; labels and helpful empty states remain visible.	Long translations and additional locales are not verified.
Layering and affordances	Surface order and control styling distinguish dialogs, menus and interactive regions. A flat style can work too.	Focus restoration is assessed as F08.
Icons and imagery	Icons remain legible at their intended sizes; controls retain meaningful labels. Public hero treatment is sampled.	Image delivery performance is F06; no upload/gallery review.
Responsive composition	Grouping and order survive the sampled narrow and landscape profiles.	Broader devices and zoom checks are scoped in Chapters 14-15.

Stress the design with realistic examples

- Long titles, empty results and high item counts should preserve context and useful next steps.
- Test image crops, text contrast and missing-image treatment with realistic content.
- Truncation must keep essential content accessible; menus must work at viewport edges.
- Loading, success, error and disabled treatments need clear, consistent meanings.

Purposeful variation is healthy

Different loaders, card layouts or icon treatments can serve different contexts. Raise a finding when inconsistency misleads someone, hides an action or makes the system harder to maintain; do not enforce an arbitrary visual quota.

These are expert assessment observations, not measured customer outcomes. Design systems support judgement; they do not replace task validation or accessibility checks.

Design the states between the screens.

A usable interface explains what is happening, preserves work and offers a sensible next action when something goes wrong.

STATE	ILLUSTRATIVE ASSESSMENT
Loading / mutation pending	Progress appears; repeated submissions are disabled. Loader styles vary by context, with consistent meaning.
Empty / no results	First-use guidance differs from an empty search result; both offer a useful next step. Zero results must leave filters and reset actions available.
Error / retry	The generation timeout message appears, but the submitted brief is discarded: F05.
Success / partial / stale	Saved plans receive confirmation; partial outputs are labelled. Offline/resume is only partly verified.

F05 MEDIUM Open · 11 Sep 2026

A generation timeout discards the customer's brief

OBSERVED IN THE FICTIONAL ASSESSMENT

The invented timeout path clears the entered planning brief. The retry action opens a blank form rather than resuming from the customer's input.

CONSEQUENCE / IMPACT

Customers must reconstruct their work after a recoverable provider failure. It increases effort precisely when the product has already failed them.

LIKELIHOOD / CONFIDENCE

Occurs on the represented timeout path; normal generation is unaffected. Evidence confidence: High.

Evidence: I05 and I06: generation timeout and retry state. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Preserve recoverable input, explain the failure and make retry/cancel outcomes clear. Keep cost and duplicate-request controls intact.

RETEST PASSES WHEN

A timeout retains the brief; retry submits the intended request once; cancel leaves the customer in a clear state without losing saved work.

Launch blocker: No. Normal generation remains available; address before promoting acquisition.

Owner: Frontend engineer · Effort: Half to one day
Depends on: Controlled provider timeout and regression fixture

06 / UX, INTERFACE & VISUAL CONSISTENCY / CODE EVIDENCE FOR F05

An error message cannot restore erased work.

This cleanup erases the draft after both success and failure, making a recoverable timeout costly.

Invented TypeScript · ui/generation/submit.ts:71-79 · AP-RC-0911

```
71  try {
72    await generatePlan(brief);
73    setStatus("saved");
74  } catch {
75    setStatus("failed");
76  } finally {
77    setBrief("");
78    setPending(false);
79  }
```

The visible consequence

Line 77 always clears the draft. “Try again” therefore returns to an empty form. Consistent spinners and messages cannot repair this state-handling failure: F05.

Proposed approach · pseudocode · not implemented or verified

```
1  const submittedBrief = brief;
2  try {
3    await generateAndConfirmSaved(submittedBrief);
4    showSavedPlan();
5  } catch {
6    preserveDraftIfUnchanged(submittedBrief);
7    showRecoverableError();
8  } finally {
9    setPending(false);
10 }
```

Retest the user's outcome

Force a timeout, confirm the brief survives, then retry once for one result. Define cancel, partial completion and late-response behaviour; prevent duplicate charges and overwriting newer edits.

Original fictional excerpts for I05 / F05. The proposed sequence omits framework-specific state and request-deduplication details; it is a review direction, not a drop-in implementation.

Measure the wait customers experience.

Performance review covers loading, responsiveness and the work done by the browser, application and providers. Lab diagnostics help explain bottlenecks; field data reflects actual users when available.

Partially reviewed

Illustrative evidence: I07

Invented Lighthouse results for a public page; authenticated journeys sampled separately. No field data or load test.

5.1s

Mobile lab LCP

2.0s

Desktop lab LCP

0.04

Mobile lab CLS

Unknown

Field INP / real users

Illustrative run: public landing page, 11 Sep 2026, Lighthouse 13.4.1; emulated mobile with slow-4G/CPU throttling and desktop profile. Median of three separate cold runs. No score or measurement here comes from a real site.

PERFORMANCE LENS	WHAT THIS SAMPLE ESTABLISHES	BOUNDARY
Loading / LCP	The main content appears later on the illustrated mobile profile than on desktop.	One public page; F06 explains the image bottleneck.
Visual stability / CLS	The mobile lab example reports 0.04.	This result is limited to the sampled loading sequence.
Responsiveness / INP	No real-user dataset is available.	Field responsiveness remains Unknown; lab results do not supply a field INP.
App and provider work	Generation behaviour is sampled as a journey.	No authenticated performance budget, sustained-load or capacity result is represented.

What to prioritise

Fix the evidenced mobile loading bottleneck before paid acquisition. Keep the result tied to the same page and test profile; do not imply that this one improvement establishes the speed of every authenticated journey.

The next page follows the illustrated diagnostic evidence. The complete finding and acceptance budget follow on page 32.

Follow the resource that delays useful content.

A performance score points to symptoms. The request sequence and the content that appears on screen explain what is making the customer wait.

Partially reviewed Illustrative evidence: I07

Original fictional Lighthouse summary and request-waterfall notes; no native Lighthouse export or real network capture.

DIAGNOSTIC OBSERVATION	INTERPRETATION	REVIEW DIRECTION
The hero is the main content element.	Its appearance determines the illustrated LCP result.	Prioritise this element's delivery before unrelated micro-optimisations.
The image is 2.4 MB.	A large transfer adds work under constrained conditions.	Produce an appropriately sized, compressed asset for the displayed role.
The request starts late.	Reducing bytes alone may leave avoidable discovery delay.	Make critical content discoverable early; verify the resulting request sequence.
Mobile differs from desktop.	The sampled device/network conditions affect the result.	Preserve both profiles and investigate regressions in either.

Retest the same question

1

Name the build
Identify the new candidate.

2

Repeat profiles
Keep comparable run settings.

3

Inspect the wait
Compare LCP and the waterfall.

4

Check the result
Keep layout and content intact.

Proposed outcome
The agreed mobile LCP budget is 2.5 seconds or less, without new layout shift. This is the acceptance target, not an achieved result.

Separate evidence streams
Use field data when available to understand real users. A missing field dataset stays Unknown instead of inheriting the lab result.

Only F06 is raised from these diagnostics. The illustration contains no measured conversion effect, server-capacity estimate or post-fix improvement.

07 / SPEED & PERFORMANCE / FINDING F06

The main content arrives too late on mobile.

The illustrated hero-image delivery explains the mobile loading weakness. Its consequence and acceptance criteria remain bounded by the sampled lab conditions.

F06 MEDIUM Open · 11 Sep 2026

The landing-page hero delays the main content on mobile

OBSERVED IN THE FICTIONAL ASSESSMENT

In the fictional lab evidence, a 2.4 MB hero image is the main content element and is requested late. The mobile LCP is 5.1 seconds, compared with 2.0 seconds on desktop.

CONSEQUENCE / IMPACT

The first useful screen appears slowly under the sampled mobile conditions. Conversion impact and behaviour under real traffic have not been measured.

LIKELIHOOD / CONFIDENCE

Expected under similar constrained-device conditions; field prevalence is unknown. Evidence confidence: Medium.

Evidence: I07: three lab runs and request-waterfall inspection. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Deliver an appropriately sized, compressed hero and make critical content discoverable early. Investigate the bottleneck rather than chasing an overall score.

RETEST PASSES WHEN

Repeat the same profiles against the new candidate; the agreed mobile LCP budget of 2.5 seconds or less is met without new layout shift. Report field data separately when available.

Launch blocker: No. The page remains usable; prioritise before paid acquisition.

Owner: Frontend engineer · **Effort:** Half to one day

Depends on: Equivalent test profiles, revised image and page build

Retain the evidence behind the decision

The supporting pack records the original fictional lab metrics and limits. A client retest would add the new candidate's actual runs, with comparable settings and an explicit pass or fail.

Does the work behind the screen complete correctly?

The backend applies business rules, changes stored data and coordinates outside services. Serverless and edge functions perform the same responsibilities even when there is no traditional server to manage.

Partially reviewed

Illustrative evidence: I01, I04, I05

Source boundaries plus existing payment and generation scenarios. A complete API, job and business-rule inventory is not represented.

BACKEND RESPONSIBILITY	WHAT THIS SAMPLE ESTABLISHES	WHAT REMAINS OPEN
Identity and access	Server-owned identity and paid-access decisions are represented.	Full endpoint permissions are scoped in Chapter 03.
Request and rule validation	Shared contracts and separated modules are represented.	Boundary values, allowed transitions and all consequential calculations are not verified.
Payment event processing	F01 shows success acknowledgement before durable entitlement work.	Safe retries, ordering and reconciliation must close the existing finding.
Generation lifecycle	The normal path completes; F05 loses input after timeout.	Cancellation, late responses, worker interruption and wider provider outages are unverified.
Scheduled and background work	Logs can correlate sampled jobs or requests.	The full job inventory, schedules, ownership and failed-job recovery remain unverified.

What completion should mean

A successful HTTP response does not establish a completed customer outcome. A durable result should be discoverable after refresh or interruption, and the product should distinguish work that is pending, failed or complete.

Useful architectural strength

Identity and paid access are represented as server-owned. Business rules should remain authoritative even when the expected interface is bypassed.

One defect, one finding

The payment processing failure remains F01. This chapter explains its backend implications and the broader evidence needed; it does not duplicate the finding.

08 / BACKEND, APIS AND BACKGROUND PROCESSING / JOBS, RETRIES AND RECOVERY

Make unfinished work visible and recoverable.

Webhooks, queues and scheduled functions often run after the customer has left the page. Delivery can repeat, work can stop midway and an old event can arrive late.

Partially reviewed

Illustrative evidence: I04, I05, I08

Payment failure, generation timeout and alert correlation are the existing fictional records. No additional queue, cron or outage exercise was run.

FAILURE MODE	WHAT THE REAL ASSESSMENT WOULD ESTABLISH
The same request arrives twice	A stable operation identity and durable state prevent duplicate effects, including simultaneous attempts.
The worker stops midway	The system can identify incomplete work, resume safely or expose an owned failure.
An older event arrives late	Ordering and state-transition rules preserve the correct current subscription or task state.
The provider is slow or unavailable	Timeouts, bounded retries and cancellation avoid indefinite work and retry amplification.
A scheduled task does not run	Expected execution and completion are monitored; missed work has a documented recovery path.
A reconciliation finds disagreement	Application records and provider outcomes can be compared, with a controlled correction and audit trail.

Start with the demonstrated failure

For F01, prove that the paid customer eventually receives the correct access after the controlled write failure, without duplicate effects. Retain the event identity and final state. A test that only checks a webhook returned success would miss the failure again.

A queue is a design option, not an automatic requirement

The assessment asks for durable, bounded and recoverable work. A simple synchronous operation may be sufficient; a queue can help when execution outlives a request, but its retries and failures still need handling.

Observability should expose stuck or exhausted work. F07 covers the existing alert-delivery defect; complete job monitoring remains a follow-up.

Protect the outcome when requests overlap.

Correctness includes arithmetic, time and simultaneous activity. Capacity includes finite database connections, function execution time and provider quotas—not just how quickly a page loads.

Not verified

Illustrative evidence: I01, I04, I09 provide context

The rules and capacity scenarios below are follow-up criteria. No concurrency exercise, peak-load measurement or new calculation result is claimed.

BUSINESS BOUNDARY	QUESTION TO MAKE EXPLICIT
Balances and allowances	Where are credit deductions, refunds and quota resets defined? What prevents negative balances or duplicate charging?
Money and time	Are amounts, currency rounding, billing boundaries and time zones represented consistently?
Finite resources	What happens at connection, request-size, execution-time or provider limits? Can the product degrade usefully?
Caches and stale state	Can a cached price, entitlement or permission outlive the event that changed it? How is freshness bounded?

An unsafe read-then-write pattern

Teaching example only · not Asterpath source or a finding

```
1  credits = read_credits(account_id)
2  if credits >= 1:
3      write_credits(account_id, credits - 1)
4      start_generation()
```

Two overlapping requests can read the same balance before either update is stored. The real design must coordinate the balance change and the operation identity, then handle failure between reserving usage and starting the external work.

Evidence that would answer the question

Run controlled overlapping requests against synthetic state. Establish the expected final balance, number of effects, recovery behaviour and recorded attempts. Database transactions and constraints are part of the design; the next chapter explains their role.

Will the information stay correct as the product grows?

Database assessment examines how information is represented, related, changed and retrieved. It includes access control, but also whether valid users can accidentally create contradictory or unusable data.

Partially reviewed

Illustrative evidence: I01, I04, I10, I12

Existing ownership, entitlement-write, deletion and recovery records provide context. A complete schema and query-quality assessment is not represented.

DESIGN AREA	EVIDENCE AVAILABLE	FOLLOW-UP NEEDED
Entities and relationships	Accounts, plans and entitlement boundaries appear in the system map.	Inspect actual tables, relationships, data types and authoritative sources.
Ownership and isolation	Selected plan read/write isolation is represented.	Check tenant ownership and policy combinations across all relevant tables and storage.
Integrity rules	F01 shows an interrupted entitlement persistence path.	Inspect uniqueness, required values, references and valid-state constraints.
Growth and query cost	No query plans or representative-growth measurements are provided.	Examine important reads and writes with realistic data shape and volume.
Changes and recovery	Staging release and rollback documentation exist.	Verify schema migration safety, restored integrity and cross-system reconciliation.

Duplication needs a reason and an owner

Repeated data is not automatically poor design. A historical purchase price may need to remain unchanged; a cached total may make reads faster. The assessment asks which copy is authoritative, why another exists and how inconsistency is detected or prevented.

A model that matches the product

Prefer understandable relationships and explicit rules. Avoid recommending a complex redesign merely because a small schema differs from a textbook pattern. Establish the failure or maintenance cost that the proposed change would address.

Keep related changes consistent.

Constraints reject invalid stored states. Transactions can group related changes. Concurrent requests still need a design that preserves the product's rules when operations overlap.

Not verified

Illustrative evidence: I01, I04 provide limited context

No schema extract, transaction-isolation review or simultaneous-write test is represented. This page explains the assessment lens, not an additional database finding.

INTEGRITY CONCERN	WHAT TO INSPECT
Conflicting records	Unique identities and references that prevent unintended duplicate events or orphaned records.
Invalid values	Required fields, appropriate types and domain rules for quantities, timestamps and state transitions.
Partial updates	Related database changes commit together where required; failed work has an explicit recovery path.
Concurrent updates	The chosen update/locking strategy prevents lost updates and over-allocation under actual request patterns.

Reserve a finite balance with one guarded update

PostgreSQL teaching example · not an Asterpath migration

```
1  UPDATE usage_balances
2  SET remaining = remaining - 1
3  WHERE account_id = :account_id
4     AND remaining >= 1
5  RETURNING remaining;
```

This sketch assumes one row per account and suitable constraints. The caller must check whether a row was returned. It does not solve repeated operation IDs, permissions, refunds or the gap between a database commit and an external provider action.

Verify the final state, not just a response

Controlled competing requests should preserve the intended balance and number of effects. Durable operation identity and reconciliation may be needed across service boundaries; a database transaction cannot roll back an external action by itself.

Inspect the reads that become expensive with growth.

A query can feel fast with a handful of records while doing unnecessary work on every request. Good assessment connects query shape, data distribution and indexes to an important customer action.

Not verified

Illustrative evidence: I01, I12 provide context

No database timing, query-plan, index inventory or load measurement is available in this sample. F06 remains a browser-loading finding, not evidence of slow database queries.

QUERY OR CHANGE AREA	EVIDENCE WORTH OBTAINING
Bounded results	Pagination, selected fields and result limits that fit the action; avoid loading an entire collection to display one page.
Repeated reads	Trace a list or export for one additional query per item, redundant fetches and expensive relationship traversal.
Useful indexes	Compare filters, joins and ordering with query plans and representative data; weigh write and storage cost before adding indexes.
Cache correctness	Identify who owns freshness, invalidation and access isolation when cached data changes.
Migrations	Track schema changes in version control; inspect data backfills, compatibility, locks and recovery for destructive steps.
Restored usability	Verify recovered relationships and important queries, then reconcile payment and job state with external systems.

Measure an agreed scenario safely

Select an important read or write and a representative synthetic dataset. Record the query, parameters, plan and environment. Start with inspection; execution-based analysis needs appropriate scope because it runs the statement and can have side effects or material cost.

Avoid a universal index checklist

An index can improve one read while increasing write work. Table size, distribution and access patterns determine whether it helps.

Schema and application must move together

A rollback of application code may still encounter a changed database. Assess forward/backward compatibility and restoration separately; see Chapter 17.

Know what is kept, shared and recoverable.

Data review follows information through collection, storage, processors, retention and deletion. It also considers whether important customer data can be restored after a failure.

Partially reviewed Illustrative evidence: I01, I10, I12

Code and policy/configuration alignment sampled; full processor deletion and backup restore are not verified.

DATA FAMILY	ILLUSTRATIVE HANDLING	LIMIT
Account and profile	Account identifiers and minimum profile fields retained while active.	Processor deletion completion not fully verified.
Plans and inputs	Saved plans belong to the account; provider receives the input needed for generation.	Retention checked in source; vendor history limited.
Payments	Application retains provider references and entitlement state; card handling remains with provider.	Wider provider compliance is not assessed.
Diagnostics	Correlation identifiers and redacted event context support investigation.	Only sampled payloads inspected.
Backup and recovery	Backup configuration and recovery owner are documented.	A successful restore drill was not evidenced.

What is working

The deletion path removes the sampled application-owned records. Central sensitive-field redaction is present in the illustrated logging setup.

What remains unknown

A provider setting that says backups are enabled does not show that restoration works. Record the last successful restore, recovered version and acceptable data-loss window.

Assessment questions

- Does the product collect what it actually needs?
- Do retention and deletion behaviour match the product’s promises?
- Can support and test work happen without unnecessary production-data exposure?
- Are interrupted deletion and restoration jobs detectable and recoverable?

These are technical and operational observations. This report does not establish legal compliance.

Follow the data beyond the first database write.

A customer’s information can appear in application records, provider requests, diagnostics and backups. Deletion and retention need to account for those distinct copies.

Partially reviewed

Illustrative evidence: I01, I10

The sampled application-owned deletion path and sensitive-field redaction are represented; processor deletion completion is not fully verified.

LOCATION	WHAT IS REPRESENTED	WHAT STILL NEEDS EVIDENCE
Application records	Saved plans belong to the account; sampled application-owned records are removed by deletion.	The complete path through all related records and interrupted deletion jobs.
Generation provider	Only the input needed for generation is sent in the sampled flow.	Effective provider retention and completion of applicable deletion requests.
Diagnostics	Central redaction limits sensitive content in sampled event payloads.	Unsampled events, exports and support workflows may require separate inspection.
Backups	Backup configuration and a recovery owner are documented.	Retention behaviour across backup copies and a verified restoration procedure.

Make the product promise testable

For a real assessment, follow an agreed synthetic account through creation, generation and deletion. Record which systems retain a copy, why it remains, who owns the action and how completion is established. A completed application request alone cannot prove that every processor has finished.

Strength to preserve

The illustrated application deletion path and shared redaction reduce scattered handling of sensitive data.

Boundary to make clear

This sample does not establish legal compliance or complete processor erasure. An unresolved lifecycle question remains a scoped unknown.

Access to another customer’s data belongs to Chapter 03. Here the question is whether permitted data use and retention remain understandable throughout the lifecycle.

A backup setting is not a recovered product.

Recovery confidence comes from restoring usable data and reconciling it with the systems around it. A documented owner and enabled backup are necessary inputs, not the result of that exercise.

Not verified

Illustrative evidence: I12

Backup configuration, ownership and rollback documentation exist in the fictional record. No successful backup restore is evidenced.

RECOVERY QUESTION	CURRENT DISPOSITION	EVIDENCE THAT WOULD CLOSE THE GAP
What can be restored?	Configuration is represented; restored contents are not demonstrated.	A recovered dataset with the intended records, integrity checks and source backup identity.
How much data might be lost?	No measured recovery window in this sample.	Agreed acceptable data loss compared with the recovered point in time.
How long will service take to return?	No timed recovery rehearsal.	An exercised sequence with access, dependencies and elapsed recovery time.
Do related systems still agree?	Cross-system recovery is not verified.	Reconcile restored access and job state with payment/provider events.

Keep a rehearsal contained

Use an isolated destination and agreed data handling. Check the recovered product's important reads and writes, then establish how payment and generation events since the backup would be reconciled. Retain a record of the result and clean up the rehearsal environment.

A separate follow-up, not a new launch blocker

The missing drill stays visible as an evidence limitation. This sample has not demonstrated data loss or a failed restore, so it does not create a ninth finding. The launch decision remains tied to F01.

Chapter 17 distinguishes an application rollback from data restoration and connects the recovery procedure to a release owner.

Can you explain what happened to a customer?

Observability uses logs, metrics and connected events to understand behaviour inside the product. The practical test is whether a problem can be noticed, investigated and handed to someone who can respond.

Partially reviewed

Illustrative evidence: I04, I08

Structured logs and sample event correlation inspected; alert delivery has a material gap, F07.

Logs and errors

CAPABILITY	ILLUSTRATIVE OBSERVATION
Structured events	Events include a stable name, severity, environment and application version.
Correlation	A request or job identifier connects a customer action to its server and provider events.
Crash context	Errors include the release and useful stack context; sensitive fields are centrally redacted.
Support lookup	The sampled paid-access mismatch can be followed from customer report to payment event and access state.
Unknown states	Recorded unpriced units remain visibly unknown. Some failed attempts are absent from the ledger; that attribution gap is F02.

A support investigation should connect these records

1	2	3	4
Customer action	App event	Provider event	Resolution
Which account and task?	Which request and release?	Which attempt and outcome?	What recovered, and when?

Strength

The illustrated log structure is useful for diagnosis and avoids copying customer content into routine events.

Gap to close

Diagnosis is reactive if nobody receives an alert. Page 44 assesses delivery and ownership; a log entry alone does not prove that someone was notified.

A small system may achieve this with correlated logs. Distributed tracing is justified when the service boundaries and investigation needs warrant it.

Connect system health to customer outcomes.

Metrics show trends; tracing or correlation explains a particular flow; alerts route important changes to a person. Each needs to reflect the product’s actual failure modes.

Partially reviewed

Illustrative evidence: I04, I08, I09

Signal definitions and selected correlation records; complete reconciliation, attempt attribution and alert delivery remain open.

SIGNAL FAMILY	ILLUSTRATIVE COVERAGE
Health and customer outcomes	Error rate, generation completion and payment-to-access drift are identified; automated payment reconciliation remains F01.
Provider and cost metrics	Attempts, retries, quota pressure and spend are partly attributed. See F02.
Connected events	Request/job identifiers link the sampled service boundaries. Full distributed tracing is not required for this scope.

Separate three meanings of success

RECORDED EVENT	WHAT IT ESTABLISHES	WHAT IT DOES NOT ESTABLISH
A payment event was received.	The application received the provider’s message.	The customer has durable paid access; F01 breaks that outcome.
A generation completed.	A successful output exists for that action.	Every failed or retried provider attempt is attributed; F02 identifies missing cost.
An alert relay ran.	The local relay completed its invocation.	The intended destination received the notification; F07 demonstrates that gap.

Use tracing where it helps an investigation

For this small product, stable request and job identifiers connect the sampled flow. Add distributed tracing if service boundaries make that chain difficult to reconstruct; installing a tracing product is not itself evidence of useful coverage.

Outcome definitions belong with the team’s response and support instructions. Thresholds, sampling and retention should fit the expected traffic and data sensitivity; none are newly measured in this sample.

11 / OBSERVABILITY & SUPPORT / FINDING F07

Notice failures before support reports them.

An alert is useful only if its delivery is checked and a responder knows what to do. A successful local invocation can conceal a failed handoff.

F07 MEDIUM Open · 11 Sep 2026

A failed alert delivery can remain unnoticed

OBSERVED IN THE FICTIONAL ASSESSMENT

The fictional staging alert relay records a successful local run after the destination rejects its message. There is no retry record or independent failed-delivery signal.

CONSEQUENCE / IMPACT

Operators may assume an important notification reached its destination. Detection of incidents becomes dependent on manual log inspection or customer reports.

LIKELIHOOD / CONFIDENCE

Occurs when the notification destination rejects a delivery; rejection is represented in the controlled example.
Evidence confidence: High.

Evidence: I08: alert relay failure and receiving-channel check. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Check delivery outcomes, make exhausted retries visible and name the response owner. Separate a successful relay invocation from successful message delivery.

RETEST PASSES WHEN

A rejected test notification becomes a visible failed delivery, recovers according to the documented policy, and reaches the intended responder or fallback.

Launch blocker: No. Core customer tasks continue; close the detection gap before growth.

Owner: Backend / operations owner · **Effort:** Half to one day

Depends on: Staging alert channel and controlled destination rejection

Retest both sides of the handoff

Reject a staging notification deliberately, inspect the recorded failure and verify retry or fallback delivery to the intended channel. The acceptance result must include the receiving side.

Check that the customer receives the outcome.

Password resets, invitations and receipts can be essential parts of a journey. Support tools also need to resolve problems without introducing untracked changes to customer data.

Not verified

Illustrative evidence: I04, I08 provide related context

Existing payment lookup and alert evidence only. Transactional email configuration, inbox delivery and sensitive support-action execution have not been assessed.

OPERATIONAL AREA	WHAT THE ASSESSMENT SHOULD ESTABLISH
Message inventory	Confirm which critical emails or notifications exist and connect each one to its customer journey.
Sender and delivery setup	Inspect the applicable domain-authentication and sender configuration, including SPF, DKIM and DMARC. A setting alone does not prove inbox delivery.
Failure and recovery	Record rejection, bounce and suppression behaviour; provide useful recovery when a message never arrives.
Customer lookup	The sampled paid-access problem can be traced to its payment event and account state. Broader support paths remain unverified.
Consequential support actions	Check authorisation, confirmation and an attributable record for refunds, access changes or corrections. Avoid relying on undocumented production edits.

Verify both sides of delivery

In a real engagement, trigger an agreed synthetic reset or invitation and retain the sending result plus the receiving-side outcome. Test relevant expiry and recovery behaviour. A provider accepting a message does not establish that the intended person can use it.

Keep the finding boundary clear

F07 concerns an operational alert relay. It does not establish a customer-email defect. Transactional delivery remains an explicit evidence gap.

Support access should fit the task

Give operators the information and controlled actions needed to help a customer, with proportionate permissions and a traceable outcome.

Run the checks. Establish what they cover.

Typechecking, linting, builds and tests detect different kinds of problems. We inspect the actual commands and their scope, then retain their outcomes and limitations.

Reviewed

Illustrative evidence: I02

All commands and results below are invented sample evidence.

EXAMPLE COMMAND / TOOL	ILLUSTRATIVE RESULT	WHAT IT TELLS US
npm run typecheck	Pass · exit 0	Selected client/server types are consistent; runtime inputs still need validation.
npm run lint	Pass · exit 0	Configured rules pass; excluded trees and disabled rules remain visible.
npm run build	Pass · exit 0	The named candidate produces a build; this does not verify live journeys.
npm run test:unit	164 checks pass	Unit/component behaviours, including selected validation and state transitions.
npm run test:integration	34 checks pass	Selected service/data interactions; incomplete payment recovery coverage.
npm run test:e2e	4 checks pass	Four happy paths; failure and recovery coverage is incomplete.
npm exec --no -- knip	Exit 1 · 12 candidates	Report-only scan using the locked local version and explicit workspace configuration.

Knip results need interpretation

In this fictional example, eight candidates are legitimate dynamic or generated entries and four are confirmed unused exports. The four are cleanup opportunities with no demonstrated launch impact. The entry-point configuration and exclusions are retained with I02.

Knip examines unused files, dependencies and exports, plus unresolved imports and unlisted dependencies. Local unused imports and variables remain the responsibility of lint rules. We do not delete code based solely on a scanner result.

Audit execution

Use the project’s locked tools or an isolated pinned setup. Inspect scripts before running; avoid automatic fixes, production targets and destructive options. Record actual versions, commands, scope, exit status and output.

Different checks answer different questions.

A scanner result is a lead to investigate. The evidence pack keeps the supporting detail alongside the assessment, including what did not run.

TOOL OR EVIDENCE FAMILY	PURPOSE	REPRESENTATION IN THIS SAMPLE
Project checks and Knip	Types, lint, build, discovered tests and unused-code candidates.	Invented results from I02; native outputs do not exist.
Lighthouse / PageSpeed	Page diagnostics and real-user performance where available.	I07 illustrates lab results only; no field dataset or native export.
SonarQube / source analysis	Code quality, reliability and supported security rules.	Not run. Tooling note explains scope, licence and export conditions.
JFrog Xray / dependency analysis	Known dependency vulnerabilities and licence/policy evidence.	Not run. No vulnerability result or advisory is invented.
Secrets and configuration	Exposed credentials and applicable infrastructure risks.	No native scanner export represented; source-review limits remain.
Accessibility and browser tests	Automated checks plus manual journey and interaction evidence.	Selected manual/runtime examples; no claim of full automated coverage.

What a client receives after a completed check

- The available native report, with the reviewed version, run conditions and scope.
- A separate interpretation that maps tool candidates to material findings, cleanup or a reasoned dismissal.
- A visible record for failed, blocked, unsupported or excluded checks.

Coverage depends on the edition and the product

Base Xray focuses on dependencies; advanced first-party analysis is a separate capability. Sonar features, exports and permitted processing depend on the installed edition and licence. A tool name is not a promise of every feature.

The public ZIP contains GadiriLabs-authored illustrative records and tooling notes, not simulated vendor reports. The attached-pack guide is on page 67.

202 passing checks can still miss the expensive failure.

Test quality depends on meaningful assertions and the risks they cover. Test count and coverage percentage are supporting information, not a readiness verdict.

QUALITY QUESTION	ILLUSTRATIVE OBSERVATION
Are tests discovered?	164 unit/component + 34 integration + 4 end-to-end = 202; no zero-test success represented.
Do they assert outcomes?	Selected tests inspect persisted state and user-visible results; some provider edges remain mocked.
Are failures and recovery covered?	Revenue and AI recovery gaps remain despite four passing happy-path end-to-end checks.
Do failures stop release?	Required checks run, but the missing scenarios cannot block a release.

F03 MEDIUM Open · 11 Sep 2026

Recovery scenarios are missing from the enforced test set

OBSERVED IN THE FICTIONAL ASSESSMENT

The fictional command set passes 202 checks. Payment-write failure, complete event recovery and AI timeout/input preservation are not asserted by the release-gated suite.

CONSEQUENCE / IMPACT

A change can pass every required check while breaking revenue or losing customer work. F01 and F05 demonstrate the affected behaviour.

LIKELIHOOD / CONFIDENCE

Every relevant change currently passes through this incomplete gate; a future regression is plausible. Evidence confidence: High.

Evidence: I02, I04, I05: discovered checks, scenario map and CI scope. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Add behaviour assertions for high-consequence failure and recovery paths. Keep dated manual evidence for scenarios that cannot reasonably be automated; fail on accidental zero-test discovery.

RETEST PASSES WHEN

Controlled regressions in access recovery and input preservation fail the relevant checks; the release workflow executes those checks against the promoted candidate.

Launch blocker: No. F01 is the launch blocker; broaden the portfolio before growth.

Owner: Test / release engineer · Effort: 1-2 days

Depends on: F01/F05 fixes, deterministic fixtures and sandbox boundaries

12 / TESTING & CHANGE SAFETY / CODE EVIDENCE FOR F03

Assert the customer outcome, including recovery.

A test can pass while proving much less than its name suggests. We inspect assertions, fixtures and integration boundaries as well as the test result.

Invented test · tests/payments/webhook.test.ts:28-31 · AP-RC-0911

```
28  it("handles a paid event", async () => {
29    const response = await sendPaidEvent(paidEvent);
30    expect(response.status).toBe(200);
31  });
```

What this test leaves open

It asserts the HTTP response. It does not show that access was granted, that a failed write recovers or that a duplicate event is harmless. It can pass with the detached write on page 24. That missing assurance is part of F03.

Proposed recovery test · pseudocode · not implemented or verified

```
1  failNextEntitlementWrite();
2  await deliverVerifiedEvent(paidEvent);
3  await runNextEntitlementAttempt(); // controlled failure
4  expect(await entitlement(customer)).toBe("free");
5  expect(await pendingRecovery(paidEvent.id)).toBe(true);
6
7  await runRecoveryUntilIdle();
8  await deliverVerifiedEvent(paidEvent); // duplicate
9  await runRecoveryUntilIdle();
10 expect(await entitlement(customer)).toBe("paid");
11 expect(await successfulGrantCount(paidEvent.id)).toBe(1);
```

Prove that the check can detect a regression

- Use an isolated test database and controlled provider events; exercise persistence rather than mocking away the failing boundary.
- Intentionally remove recovery or input preservation and confirm the relevant check fails.
- Ensure the release workflow discovers and runs the test on the candidate that will be promoted.

This one scenario is not the complete suite. Cancellation, expiry, delayed and out-of-order events need their own expected final states. Original fictional code linked to I02/I04; no private product code is reproduced.

Understand the services the product relies on.

External services can fail, slow down or become unexpectedly expensive. We assess credentials, limits, retries, degraded behaviour and whether usage can be reconciled with bills.

Partially reviewed

Illustrative evidence: I01, I04, I05, I09, I11

Server-held credentials, per-account limits and one aligned invoice period are represented. Provider-wide outages, peak demand and adversarial AI behaviour are unverified.

DEPENDENCY	CUSTOMER CONSEQUENCE	ILLUSTRATIVE ASSESSMENT
Payment provider	Confirmed payment must lead to the correct access.	F01 exposes the gap between an external event and durable application state.
AI generation provider	The customer needs a saved plan or a recoverable failed request.	Normal generation completes; timeout loses input, F05.
Provider billing	Every attempt can contribute to the cost of serving a customer.	F02 identifies £77 of unattributed spend in the aligned period.
Operational signals	Support needs to identify the attempt and its outcome.	Correlated events help diagnosis; cost attribution and alert delivery are incomplete.

Controls already represented

- Privileged credentials remain on the server.
- Per-account generation limits provide a bounded control in the sampled flow.
- The generation handbook names model/prompt version, cost cap, output checks and human acceptance for its synthetic dry-run.

Controls have limits

A per-account limit does not establish an account-wide spending ceiling or resilience under coordinated demand. A workflow dry-run does not prove the quality, safety or reliability of all generated output.

The next page examines failure and usage boundaries. The cost reconciliation finding follows on page 52.

Keep retries from becoming hidden work.

Retries can recover an interrupted task, but they can also repeat charges, confuse state or hide how much work was attempted. The application needs an explicit result for each attempt.

Partially reviewed

Illustrative evidence: I04, I05, I09

Existing payment, timeout and ledger evidence only. No additional outage simulation, load test or provider failover test is represented.

FAILURE BOUNDARY	EVIDENCE IN THIS SAMPLE	WHAT TO VERIFY NEXT
Payment acknowledgement	F01 acknowledges before the entitlement write is durable.	Recovery and event ordering converge on the correct access without duplicate effects.
Generation timeout	F05 clears the brief and makes retry harder.	Retained input, one intended retry, clear cancel and late-response handling.
Attempt attribution	F02 leaves some retries, failures and missing telemetry out of product attribution.	Every attempt has an outcome; missing prices or units stay visibly unknown.
Wider service disruption	Provider-wide outage and peak demand are not tested.	Agreed timeout, quota, degraded-state and operator-response scenarios.

A useful cost record follows attempts to outcomes

Separate the customer action from each provider attempt it causes. Reconcile successful, failed and retried work with the bill before dividing by successful customer outcomes. Otherwise an apparently cheap success can hide the cost of work that failed before it.

Keep the denominator meaningful

The illustrated period has 10,912 successful plans. Include the cost of unsuccessful attempts when calculating what those successes cost to deliver.

Bound any follow-up exercise

Use agreed synthetic inputs, sandbox targets and cost limits. A resilience assessment does not require intentionally disrupting the live provider.

These implications cross-reference F01, F02 and F05; they do not add a separate finding for each affected subsystem.

Know what each successful customer action costs.

The product’s own ledger and the provider invoice describe the same period but disagree. The missing attribution weakens pricing and margin decisions.



F02 MEDIUM Open · 11 Sep 2026

The product ledger understates the provider invoice by £77

OBSERVED IN THE FICTIONAL ASSESSMENT

For the same invented 30-day period, the product ledger totals £351 while the provider invoice totals £428. Some retries, failures and missing telemetry are absent from product attribution.

CONSEQUENCE / IMPACT

Pricing and gross-margin decisions use incomplete costs. Across 10,912 successful plans, ledger cost is £0.032 per success; invoice cost is £0.039.

LIKELIHOOD / CONFIDENCE

The mismatch exists in the illustrated period; its behaviour at higher demand is unverified. Evidence confidence: High.

Evidence: I09: aligned provider invoice and product attempt ledger. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Record every provider attempt and its outcome. Preserve missing/unpriced units as unknown, reconcile totals on a schedule and alert on unexplained variance.

RETEST PASSES WHEN

A new aligned period explains every billed category, including retries and failed attempts; cost per successful action includes those costs. Remaining variance is classified and owned.

Launch blocker: No. The represented spend is bounded; fix attribution before scaling acquisition.

Owner: Product / backend engineer · **Effort:** 1-3 days

Depends on: Read-only billing totals, attempt identifiers and effective price sources

Additional sampled controls: server-held credentials and per-account limits. Provider-wide outages, peak capacity and adversarial AI behaviour remain outside this sample.

A completed generation should still be useful.

AI quality asks whether the result meets the product’s promise, handles uncertainty and remains controllable. A successful API response or a low unit cost cannot answer those questions.

Partially reviewed

Illustrative evidence: I05, I11

A successful generation and one synthetic workflow dry-run are represented. No representative output-quality benchmark, adversarial assessment or model-comparison result is claimed.

QUALITY BOUNDARY	WHAT TO ESTABLISH IN AN APPLICABLE AI FEATURE
Useful output	Define success against realistic inputs: completeness, correct structure, appropriate content and consistency with the customer’s brief.
Known difficult cases	Include ambiguous, incomplete and out-of-scope inputs; specify when the product should ask, decline or expose uncertainty.
User correction	Make editing, retrying and rejecting a result understandable; preserve useful input and distinguish saved from unsaved work.
Versions and regression	Record prompt, model and relevant configuration; compare representative cases before changing them.
Untrusted content and actions	Where external content or tools exist, assess instruction manipulation, sensitive-data exposure and the authority of AI-triggered actions.

Start with an agreed evaluation set

Choose synthetic or appropriately authorised examples that represent the intended use. Define a review rubric and acceptable outcomes before comparing versions. Preserve failures as well as successes; a hand-picked demonstration does not establish dependable quality.

Evidence already represented

The workflow names prompt/model version, a cost cap, output checks and human acceptance. Its synthetic dry-run establishes only that limited exercise.

Product consequence

If an output is wrong, can the customer recognise and correct it before relying on it? The answer should influence the interface, evaluation and permissions.

F05 remains the demonstrated loss-of-input defect. This broader quality review adds visible follow-up criteria without inventing poor outputs or new findings.

Keep important tasks usable with different access needs.

Accessibility review samples how people navigate, perceive information and operate controls, including keyboard and assistive-technology use. Automated checks support manual inspection.

Partially reviewed

Illustrative evidence: I06, I10

Selected keyboard, labels, contrast, zoom and screen-reader checks; no complete conformance assessment.

ASSESSMENT LENS	ILLUSTRATIVE OBSERVATION	COVERAGE LIMIT
Document structure	Headings are present in the scoped screens.	No exhaustive semantic inspection of every route.
Names and instructions	Form labels are present; validation messages are associated with their fields.	Additional content variants and every generated result are not represented.
Contrast and zoom	Sampled text and controls remain legible at the tested contrast and 200% zoom.	This is not a complete contrast, reflow or conformance assessment.
Keyboard and assistive technology	Core controls are reachable.	Dialog focus return is inconsistent; F08 follows on page 56.

Semantics serve more than visual appearance

Headings expose the information structure; labels identify a control’s purpose; associated errors help people find and correct a problem. A visually consistent component still needs the right name, role and behaviour when used with a keyboard or screen reader.

Keep the useful foundation

Retain shared controls and the sampled label/error associations while correcting the focus lifecycle.

Report evidence by task and state

An export dialog that can be opened is only part of the task. Operation, confirmation and returning to the workspace need their own observations.

Visual hierarchy and component consistency are discussed in Chapter 06. This chapter focuses on access to the same information and actions.

Inspect the interactions a score cannot explain.

Automated checks can find some structural and rule-based issues. Manual task inspection is needed to understand focus, reading order, instructions and recovery in context.

Partially reviewed

Illustrative evidence: I06, I10

Selected manual observations are represented. No native axe or Lighthouse accessibility report was generated for the fictional product.

METHOD	WHAT IT CONTRIBUTES	THIS SAMPLE'S BOUNDARY
Automated rules	Candidates for missing names, associations and other detectable rule failures.	No native automated result is attached; no score is invented.
Keyboard sequence	Reachability, visible position, operation, escape and return through a real task.	The export-close sequence demonstrates F08; other dialogs are not comprehensively assessed.
Screen-reader inspection	Names, roles, state and reading sequence in the sampled interaction.	Selected checks only; no comprehensive browser/assistive-technology matrix.
Visual and layout inspection	Legibility, contrast and content usability under agreed conditions.	Selected contrast and 200% zoom observations; broader reflow and device coverage remain limited.

Read the scope before interpreting the result

The report names the screen, state, environment and method supporting a conclusion. Passing an automated check cannot establish that focus returns to the expected place or that a message makes sense. Conversely, a useful manual sequence does not establish coverage of every applicable accessibility criterion.

Next evidence to obtain

Retest F08 through opening, operating, closing and continuing from the dialog. Extend manual and automated coverage according to the supported tasks and audiences, and retain the resulting environment details.

This is a technical accessibility review, not a certification or complete conformance claim. The same finding is linked from journeys and UX without being counted again.

Return the keyboard user to a meaningful place.

Closing an overlay should leave the customer oriented and able to continue. In the sampled export sequence, the focus position is lost.

F08 MEDIUM Open · 11 Sep 2026

Closing the export dialog loses the keyboard position

OBSERVED IN THE FICTIONAL ASSESSMENT

After the fictional export dialog closes, focus returns to the document body instead of the control that opened it. A keyboard user must navigate back through the page.

CONSEQUENCE / IMPACT

The next step becomes harder to locate, particularly when using keyboard or screen-reader navigation. Export remains possible but takes avoidable effort.

LIKELIHOOD / CONFIDENCE

Reproduced on the sampled dialog-close path; other dialogs were not comprehensively assessed. Evidence confidence: High.

Evidence: I10: keyboard and screen-reader export-dialog sequence. All evidence describes fictional staging candidate AP-RC-0911.

REQUIRED OUTCOME

Manage focus on entry and close, keep the active position visible and expose clear dialog semantics without trapping users.

RETEST PASSES WHEN

Keyboard and sampled assistive-technology users can open, operate and close the dialog, with focus returning to the initiating control and a clear success state.

Launch blocker: No. Export remains reachable; resolve this access barrier before growth.

Owner: Frontend engineer · **Effort:** Half a day
Depends on: Dialog component and keyboard regression check

Complete the interaction

Retest the initiating control, dialog entry, operation, close and returned focus as one sequence. Confirm that success feedback remains understandable without requiring the customer to locate a new starting point.

Name the combinations the product supports.

Compatibility review checks the experience across agreed browsers, devices and layouts. Language and regional behaviour are assessed against the audiences the product promises to serve.

Partially reviewed Illustrative evidence: I03, I06, I10

Illustrative profiles below; a responsive viewport is not evidence of testing every physical device.

ILLUSTRATIVE PROFILE	SAMPLED BEHAVIOUR	RESULT / LIMIT
Chrome 150 / Windows / 1440 × 900	All four core happy paths; hierarchy and recovery checks.	Core paths verified; findings remain.
Firefox 153 / Windows / 1366 × 768	Sign-in, generation and export.	Reviewed sample; payment recovery not repeated.
Safari 26 / macOS / 1440 × 900	Core navigation, purchase handoff and export.	Reviewed sample; long-lived sessions unverified.
iPhone / iOS Safari 26 / 390 × 844 and landscape	Forms, dialogs, virtual keyboard and orientation changes.	Reviewed sample; no claim about every iPhone.
Android Chrome / tablet layouts	No runtime profile represented.	Not verified.

What we look for within a profile

- Controls remain reachable when the viewport shrinks or the keyboard opens.
- Orientation and zoom do not hide essential information or lose entered work.
- Touch targets, hover-only interactions and browser permissions receive appropriate treatment.

Support claims follow evidence

Publish a realistic support policy and add targeted checks for unverified combinations before promising them to customers. A single desktop emulator is insufficient evidence for all mobile browsers.

Keep meaning intact as the context changes.

A narrower screen, a different orientation or a regional format can change how an interface is understood. Coverage follows the product’s actual audience and support promises.

Partially reviewed

Illustrative evidence: I03, I06, I10

Named desktop and iPhone profiles, selected English formats and sampled content. Android, tablet and additional locales have no runtime evidence here.

CONTEXT	ILLUSTRATIVE ASSESSMENT	NEXT USEFUL BOUNDARY
Narrow and landscape layouts	Grouping and order survive the sampled iPhone profiles.	Additional screen sizes and Android/tablet runtime behaviour remain Not verified.
Forms and changing viewport	Forms, dialogs, virtual keyboard and orientation are included in the iPhone sample.	This does not establish every device’s keyboard, permission or browser behaviour.
English and regional formats	Dates use an unambiguous month name; displayed prices include currency.	Wider time-zone and regional edge cases are not verified.
Translations and right-to-left layout	Not applicable: Asterpath does not promise translated or RTL interfaces.	Reopen this scope if the intended audience or product promise changes.

Expand coverage when a promise changes

A new locale requires more than translated labels: content length, direction, input formats, dates, currency and help may all need attention. A new device claim needs runtime evidence for important controls and journeys. Record the combinations actually exercised rather than converting one responsive screenshot into a broad support claim.

Keep the existing findings in view

Compatibility sampling does not clear F01, F05 or F08. Successful paths in several browsers can coexist with a recovery defect that affects the shared application logic.

Chapter 06 covers component/content resilience; Chapter 14 covers access needs. This chapter records where those experiences were exercised.

Help the right public content be found.

For a public website, technical discoverability helps search systems reach and interpret intended content. Private application screens should retain their access and indexing boundaries.

Partially reviewed

Illustrative evidence: I13

Public landing-page technical checks only; rankings, AI citations and agent-operated tasks are not verified.

AREA	ILLUSTRATIVE COVERAGE
Crawling and indexing	Public landing content is reachable; robots, canonical and index controls are coherent in the sample.
Content and navigation	Useful text, descriptive headings and internal links describe the product and its main destination.
Metadata and structured data	Title/description reflect the page; any structured data must match visible content and an applicable type.
Private and duplicate pages	Account content remains behind authentication; utility destinations are kept out of the intended search set.

What the inspected controls establish

The intended public page can be reached and its purpose is described coherently. That supports technical eligibility. The sample contains no account-console evidence showing actual indexing, queries, clicks or search outcomes.

Keep private content private

Authentication protects account information. Indexing directives describe crawler treatment; they do not replace an access-control boundary. Public technical discovery checks do not establish that private data is safe in every scenario.

Proportionate follow-up

Check real indexing evidence when Search Console access is available. Investigate errors and intended exclusions before adding new markup or files. Assess agent-facing capabilities only when the product has a meaningful use case for them.

No custom AI file, experimental integration or promised ranking is required to pass this bounded chapter. Sources on page 69.

Separate being found from completing a task.

AI search may surface public information. An agent using the product attempts actions through an interface or integration. They require different evidence and should not share a single success claim.

Not verified

Illustrative evidence: I13

No demonstrated AI citation, ranking outcome or end-to-end agent-operated task. This page explains the boundary of the public technical inspection.

Search and AI discovery

Can a system access and understand the intended public information? Google’s guidance retains the same foundational technical requirements for its AI search features.

Agents using the interface

Can a software agent identify controls and complete an authorised task? Semantic controls and stable layouts can help, but successful task execution needs separate validation.

QUESTION	EVIDENCE THAT WOULD BE NEEDED	CURRENT DISPOSITION
Is the page indexed as intended?	Dated search-console or equivalent indexing evidence for the actual public page.	Not verified; technical controls only.
Does an AI search result cite the product?	A dated query/result observation with the product and context identified.	Not verified; no visibility outcome claimed.
Can an agent use a meaningful workflow?	An agreed task, allowed actions, environment and observed final customer state.	Not verified; no agent task attempted.

Choose a useful follow-up

Start with the public information customers need and the indexing evidence available. Assess an agent-facing workflow only if it serves a product need, with explicit boundaries around account actions and consequential steps. Extra files or integrations should answer that need rather than stand in for evidence.

The absence of an experimental AI integration does not create a finding. The sample’s technical discovery strengths remain limited to I13 and do not imply search growth or agent reliability.

Ship the version you verified. Keep a way back.

A release process connects the reviewed source, checks and built artifact to what customers receive. Recovery planning explains how an operator restores a known-good state when a change fails.

Partially reviewed

Illustrative evidence: I02, I12

Fictional staging release record inspected; production promotion and full restore rehearsal not verified.

STAGE	ILLUSTRATIVE EVIDENCE / REQUIRED IMPROVEMENT
Prepare	Named candidate, migration notes and owner are recorded.
Validate	Typecheck, lint, build and current tests are required. Missing recovery scenarios remain F03.
Promote	Release record connects source and artifact. Close F01, identify a new candidate and verify it before promotion.
Confirm	Smoke checks cover sign-in, paid access, generation and export; monitoring includes release identity.
Recover	Known-good artifact and rollback instructions exist; database restoration still needs exercised evidence.

What CI/CD means here

Continuous integration runs agreed checks when code changes. Delivery automation prepares or releases the verified build. We assess whether important checks actually prevent unsafe releases, including any documented manual controls.

Acceptance for the next release candidate

- F01's payment recovery test passes against the exact candidate.
- The operator confirms that the deployed version matches the verified artifact.
- Post-release smoke checks and alert routing have a named owner.
- Migration compatibility and a usable rollback/recovery path are recorded.

A missing automation product is not automatically a finding. The material question is whether unchecked or unrecoverable changes can reach customers.

Make the release record follow the artifact.

A passing check is useful only when it applies to the version customers receive. Candidate provenance is the record connecting source, checks, build and promotion.

Partially reviewed

Illustrative evidence: I02, I12

A fictional staging source/artifact record and required checks are represented. Production promotion of a repaired candidate has not occurred.

CONNECTION	WHAT THE RECORD SUPPORTS	REMAINING DECISION
Source → checks	The candidate has named typecheck, lint, build and test commands.	F03 leaves consequential recovery scenarios outside that gate.
Checks → build	The staging record connects the checked source to its artifact.	Any repair creates a new candidate requiring its own relevant checks.
Build → promotion	The release handbook describes promotion and records an owner.	F01 must pass its acceptance scenarios before launch.
Promotion → observed service	Smoke checks and release identity are part of the documented process.	Production execution and outcome are not represented in this sample.

Look beyond the presence of a pipeline

Check what triggers the workflow, which commands actually run, whether failures stop promotion and whether manual steps are recorded. A CI/CD badge cannot prove that a test was discovered or that the checked artifact reached production. A proportionate manual control can be assessed with the same questions.

The next candidate needs a new record

Keep AP-RC-0911 as the assessed snapshot. After repairing F01, name the new candidate, attach its payment-recovery evidence and record its promotion decision. This expanded report is a presentation revision, not a retest or a changed verdict.

The sample’s 202 passing checks still omit the recovery scenarios in F03. The test chapter shows both that limitation and a proposed assertion pattern.

Know which recovery action fits the failure.

Returning to an earlier application build and restoring stored data solve different problems. The operator needs to know which action is safe, what it affects and how recovery will be verified.

Partially reviewed

Illustrative evidence: I11, I12

A known-good artifact, rollback instructions and named owner are documented. A complete restore and independent release/recovery handover remain unverified.

RECOVERY ACTION	ILLUSTRATIVE READINESS	VERIFICATION STILL NEEDED
Application rollback	A known-good artifact and instructions exist.	Exercise the procedure and verify that the resulting service matches the intended version.
Migration compatibility	Migration notes are part of the release handbook.	Establish whether the earlier build can operate safely with the current data/schema.
Data restoration	Backup configuration and a recovery owner are represented.	The isolated restore and reconciliation evidence described in Chapter 10.
Operational handover	A handbook describes release, response and escalation.	Another operator completes the full process with the necessary access.

Finish with the customer outcome

After a recovery action, establish the active version and repeat the relevant smoke paths. Check that paid access and provider events still agree, saved plans remain usable and the response owner can see failures. A completed command is only one part of that verification.

Preserve useful controls

Keep the named owner, versioned handbook and artifact record. They give the rehearsal a concrete starting point.

Keep the missing evidence visible

Do not report a successful restore or independent handover until the exercise has actually completed and its result is retained.

These are existing scope limitations, not newly reproduced failures. The action plan carries them as evidence follow-ups alongside the eight confirmed illustrative findings.

ACTION PLAN

One launch blocker. A sequenced improvement plan.

Each material finding appears once below. Estimates are illustrative engineering effort; scheduling depends on access, fixes and verification.

PRIORITY / FINDING	OUTCOME AND OWNER	EFFORT
Before launch · F01	Durable paid access and recovery. Backend/release · payment sandbox and regression evidence.	1-2 days
Before growth · F05	Preserve the brief after generation failure. Frontend · controlled timeout fixture.	½-1 day
Before growth · F03	Enforce high-consequence recovery checks. Test/release · follows F01/F05 and stable fixtures.	1-2 days
Before growth · F07	Detect and recover failed alert delivery. Backend/operations · staging channel and failure control.	½-1 day
Before growth · F02	Reconcile all provider attempts and costs. Product/backend · aligned billing and attempt records.	1-3 days
Before growth · F06	Meet the agreed mobile loading budget. Frontend · equivalent lab profile and image build.	½-1 day
Before growth · F08	Restore predictable keyboard focus. Frontend · shared dialog and regression check.	½ day
Planned · F04	Clarify workspace action hierarchy. Product/design · agreed first-use task.	½ day

Evidence follow-ups, separate from confirmed defects

- Exercise a backup restore and independent release/recovery handover.
- Validate additional browsers/devices before expanding support promises.
- Obtain real-user performance and search-console evidence when available.

Keep the good foundations

Retain server-owned access controls, environment separation, correlated redacted logs, reproducible setup and the shared development handbook while implementing the fixes.

VERIFICATION AND HANDOVER

Close findings with evidence.

A retest checks agreed changes against a named new candidate. It does not imply that every unchanged area was reviewed again.

STEP	WHAT IS RECORDED
Agree the retest	Finding IDs, required outcomes, environment, candidate and access.
Run the acceptance scenarios	The verification steps on each finding page; preserve outputs and runtime observations.
State the outcome	Pass, Fail or Not verified for each retested finding, with date and evidence.
Update the decision	Keep “Do not launch yet” while F01 remains unresolved. Reassess against the newly named candidate after its acceptance evidence is complete.
Update relevant scores	Change a maturity score only when evidence supports the new level; retain the previous score/date.
Hand over	Owners, remaining risks, operating instructions, access cleanup and optional follow-on scope.

What a private client report adds

This sample shows the complete reading structure and selected illustrative detail. The private deliverable replaces invention with the actual versioned evidence, every material finding, the full technical coverage disposition, command outputs, unresolved questions and a walkthrough of the recommended sequence.

Useful to the owner

A clear decision, understandable consequences, prioritised work and a way to judge whether the fixes succeeded.

Useful to the next engineer

Stable finding identifiers, source/version evidence, dependencies, acceptance scenarios and enough context to investigate without guessing.

Implementation remains a separate decision

A client can take the report to another engineer. Any implementation proposal or additional retest has its own agreed scope and acceptance criteria; it does not determine finding severity.

Disclosure: Asterpath is fictional. This report is a demonstration of structure and reasoning, not a testimonial, proof of an engagement or evidence that its measurements occurred.

APPENDIX / ILLUSTRATIVE EVIDENCE INDEX

Trace each conclusion to its evidence.

A real audit retains exact tool versions, commands, source references and dated observations. These fictional identifiers demonstrate that relationship without exposing a client’s internal systems.

ID	ILLUSTRATIVE EVIDENCE PACKET	USED IN
I01	AP-RC-0911 source/configuration snapshot; system map, data boundaries and ownership register.	Ch 02-04, 08-10, 13
I02	Locked tool versions; command, configuration, exit and discovery records; CI-required-check inventory; Knip triage.	Ch 02, 04, 12, 17; F03
I03	Sign-in/session runtime records for the named profiles and sample isolation scenarios.	Ch 01, 03-04, 15
I04	Payment-handler excerpt; sandbox events, access state and controlled write-failure/recovery records.	Ch 01-02, 05, 08-09, 11-13; F01
I05	Generation-state excerpt; success, timeout and retry sequences against synthetic input.	Ch 01, 06, 08, 12-13; F05
I06	Screen/state inventory; action hierarchy comparison and heuristic review notes.	Ch 06, 14-15; F04/F05
I07	Three-run lab performance bundle, profile/version settings and request-waterfall notes; no field dataset.	Ch 07; F06
I08	Redacted sample logs, correlation records and controlled alert-delivery failure.	Ch 11; F07
I09	Aligned invented 30-day invoice and product ledger; 10,912 successful plan outcomes.	Ch 11, 13; F02
I10	Keyboard, screen-reader, contrast/zoom notes; sampled retention/deletion and processor boundaries.	Ch 01, 10, 14-15; F08
I11	Instruction-loading exercise; command/link checks; testing/release handbook and synthetic generation dry-run.	Ch 02, 13, 17
I12	Staging release record and documented rollback; restore and full handover evidence absent.	Ch 02, 09-10, 17
I13	Public-page indexing controls, content/metadata checks; account-console and agentic task evidence absent.	Ch 16

Sample provenance

All packets refer to fictional candidate AP-RC-0911 and assessment date 11 Sep 2026. I09 represents the preceding 30-day billing period. There are no real client records behind these identifiers.

APPENDIX / SUPPORTING EVIDENCE PACK

Read the assessment. Inspect the evidence when needed.

The ZIP keeps this PDF, supporting records and editable priorities together. Start with the decision and action plan; use the detail to investigate or brief the next engineer.

INSIDE THE SAMPLE ZIP	HOW TO USE IT
START-HERE.html	An offline index that explains the contents and links to each supporting record.
Assessment PDF	This complete report, with chapter navigation, findings and retest criteria.
assessment-coverage.html / .json	A reader guide linking the 17 chapters and additional founder concerns to the existing evidence and explicit unknowns.
illustrative-evidence/	Readable examples and structured GadirLabs records for checks, Knip, performance, journeys, accessibility and costs.
tooling-notes/	SonarQube and JFrog Xray roles, availability and export notes. These are not scan results.
findings.csv	A portable list of the eight findings, their priorities, owners and required outcomes.
scan-run-register.csv	What each evidence entry represents, its scope and the explicit absence of actual sample scanner execution.
manifest.json	File inventory, hashes and provenance so the pack can be checked after handover.

Supporting records are not extra findings

The performance record expands I07 and F06. The recovery record supports F01 and F05. Several records can support one finding; their count does not increase its severity or the total number of defects.

This fictional sample

Every supporting record is clearly marked as an original GadirLabs illustration. There are no actual Asterpath scans or native Sonar/Xray results behind it.

An actual client handover

The pack contains the permitted native exports from completed checks, alongside scope, triage and manual evidence. Unavailable or withheld evidence has a stated reason.

Useful beyond this engagement

The report and evidence should help you work with another engineer, compare a later candidate or verify a proposed fix. You can download the PDF on its own when you only want the readable assessment.

APPENDIX / TERMS AND READING NOTES

A little context for the technical language.

These definitions connect engineering terms to the decisions in the report. The following page lists the sources that inform the assessment.

TERM	MEANING IN THIS REPORT
Entitlement	The features or usage a customer is allowed to access.
Authentication / permissions	Establishing who is acting; deciding what they may do.
OAuth / provider sign-in	Delegated access through another provider; a sign-in flow must also establish the account identity.
Backend / edge function	Server-side work that validates requests, applies rules and coordinates data or services.
Constraint / transaction	A rule that rejects invalid stored data; a group of related changes committed together.
Query plan	The database's approach to finding, joining and returning the requested records.
CI / release gate	Checks that run on changes and the controls that prevent an unverified release.
Idempotent recovery	Repeating a request or event safely, without duplicating its intended effect.
LCP / CLS / INP	Loading of the main content; unexpected layout movement; responsiveness to user interaction.
Heuristic review	Expert inspection against usability principles; distinct from observing representative users.
Modularity / cohesion	Related responsibilities live together with an understandable purpose.
Coupling	How much a change in one area forces other areas to change with it.
Design tokens	Shared values for recurring type, colour, spacing and other interface decisions.
Lab / field data	Controlled diagnostics versus measurements from actual user conditions.

Use a source extract to explain a mechanism

The code pages show original invented examples, with source locations and candidate identity to demonstrate evidence traceability. Proposed designs are incomplete teaching examples; fixes require implementation and verification against the real system.

APPENDIX / METHOD REFERENCES

The sources behind the assessment lenses.

These sources inform the questions we ask. They do not endorse this report, establish a universal checklist or replace evidence from the product being assessed.

Primary references · checked 11 Sep 2026

1. Nielsen Norman Group: 10 usability heuristics
Framework for expert usability inspection; not a substitute for user research.
2. NN/g: consistency and standards; visual-design principles
Consistent meaning and interaction. Visual design: <https://www.nngroup.com/articles/principles-visual-design/>
3. OpenAI: AGENTS.md and reusable guidance
Discovery, scoped instructions and verification. Practical contents: <https://learn.chatgpt.com/guides/best-practices>
4. Anthropic: Claude Code project memory
Concise CLAUDE.md files, supported locations and modular guidance. Loading: <https://code.claude.com/docs/en/debug-your-config>
5. Google: PageSpeed Insights
Lab versus field evidence and measurement limitations.
6. W3C: first accessibility review
Manual checks and their limits. Automated scoring: <https://developer.chrome.com/docs/lighthouse/accessibility/scoring>
7. Knip: issue types and analysis model
Scanner coverage. Configuration limits: <https://knip.dev/explanations/how-knip-works>
8. Google: AI features and your website
Technical search eligibility; no guaranteed visibility. Agentic browsing is separate and experimental: <https://developer.chrome.com/docs/lighthouse/agentic-browsing/scoring>
9. SonarQube: reports and permitted processing
Native report availability depends on edition. Current analyser licence: <https://www.sonarsource.com/license/ssal/>
10. JFrog Xray: dependency analysis and exports
Product scope and conditional features. Export reference: <https://docs.jfrog.com/security/docs/export-scan-results-source-code>

Additional design reference

Adam Wathan and Steve Schoger, *Refactoring UI*, v1.0.2. Used to inform visual hierarchy, layout, typography, colour, depth, imagery and interface-state assessment. Its design advice is applied contextually; book artwork and text are not reproduced.

The internal assessment uses one technical coverage register across these reader-facing chapters. Scope determines depth; unavailable evidence remains visible. This public sample does not publish the complete internal checklist.

APPENDIX / ENGINEERING REFERENCES

Primary guidance for the deeper engineering chapters.

These references explain the underlying concepts. The product's architecture, scope and evidence determine which checks apply and how deeply they are assessed.

1. OWASP Application Security Verification Standard

A structured reference for authentication, access control and application security assessment. This sample is not an ASVS certification.

2. OWASP API Security Top 10

API-specific risks, including object permissions, resource consumption and sensitive business flows.

3. Supabase: authenticating Edge Functions

Authentication still belongs at serverless function boundaries; implementation depends on the actual caller and route.

4. PostgreSQL: constraints

Required values, uniqueness, references and other rules that preserve valid stored data.

5. PostgreSQL: transaction isolation

Concurrent transaction behaviour and the guarantees associated with isolation levels.

6. PostgreSQL: using EXPLAIN

Interpreting query plans and the execution implications of EXPLAIN ANALYZE.

Apply guidance to the real system

The PostgreSQL examples teach transferable questions about integrity and concurrency. Another database may use different mechanisms; judge whether its actual design preserves the required customer outcome.

New chapter depth is explanatory and organisational. The fictional evidence remains I01-I13, and the finding register remains F01-F08. Missing OAuth, schema, query, job, delivery and AI-quality evidence is disclosed where relevant.