CEED in Practice

One change from permission to production

Public Review Draft 2 · October 26, 2026

Viktor Barakoski

A support team adds ticket priorities. Its coding agent finds a billing dependency, and passing tests leave a permission question unresolved. This companion follows the team’s decision through review, delivery, and later investigation. It then gives readers a matched comparison, a measurement worksheet, calibration cases, and a synthetic coverage declaration to use in their own trials.

The CEED Framework™ connects authority, delivery checks, and retained evidence. The name CEED™ stands for Contract-Enforced, Evidence-Driven.

All people, repositories, events, records, and numerical exercise values in this companion are synthetic. They demonstrate the specified process, not an operating deployment or measured benefit.

How to use this companion

A CEED Delivery Contract records what one change may touch, which evidence it needs, and how it may be released. The CEED Router checks that authority at delivery gates. A CEED Change Passport retains the decisions and links to their evidence. The short names Contract, Router, and Passport refer to those components throughout this example.

Developers can follow the feature from its assigned requirement to its first blocked candidate. Authorizers can inspect the two permission decisions. Platform and operations teams can trace the gate connections and retained records. Assurance readers can start with the investigation and then follow its links backward.

The CEED Specification and Change Passport Schema, Public Review Draft 2, governs the rules. This companion explains them. The worked examples trace decisions to their authority and evidence. Chapter 2 explains how the records connect, using a complete Contract object and summaries of the retained history.

Reader	Useful starting point
Developer or proposer	1 The worked lifecycle; 5 Calibrate the decisions
Authorizer	1.3 Correct the proposal and release a hold; 1.5 Authorize the billing exception
Platform or operations team	2 Read the stored records; 6 Map actual enforcement
Engineering or assurance lead	3 Compare the complete work; 4 Record effort as it happens

Table 1. Reading paths
Reading paths connect each role to a practical task.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

Contents

How to use this companion 2

1 The worked lifecycle 3

2 Read the stored records 7

3 Compare the complete work 10

4 Record effort as it happens 11

5 Calibrate the decisions 14

6 Map actual enforcement 15

7 Run the first trial 17

Publication information 17

1 The worked lifecycle

The support team wants its ticket-management API to store low, medium, or high priority. The product owner records that intent and the acceptance criteria in REQ-1432. The identifier names the work item, not the full requirement. Its tracker remains the authoritative source.

Figure 1 follows the decisions. The detailed story below explains when each role acts and what its decision permits.

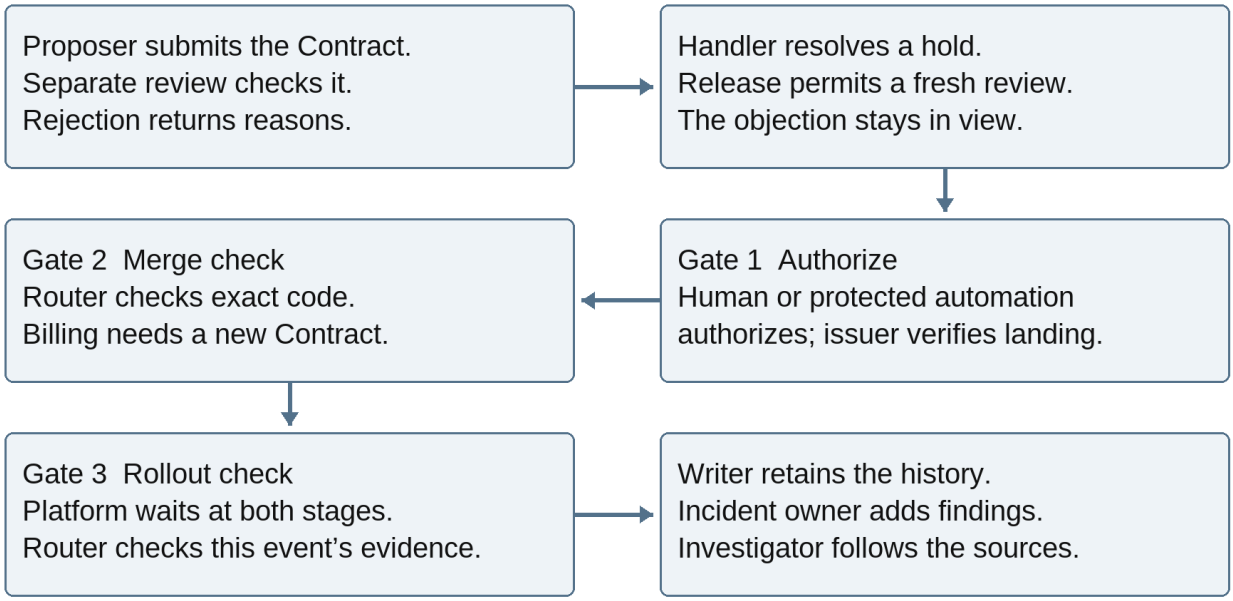


Figure 1. Decisions carry the change through delivery.
Review and operational recovery lead back to authorization; retained history supports investigation.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

1.1 Name the people and protect their roles

The coding agent is the Contract proposer and implements the feature. A separate review service checks routine Contract proposals. A human review operator monitors that service. The protected issuer applies repository policy and records issuance. Maya, the engineering lead, can independently authorize an exception. Elena handles operational holds. The incident owner records later associations and retractions.

The organization protects the requirement assignment, policy, enrollment, issuing credentials, and record writer. The coding agent can prepare a proposal and read review reasons, but it cannot make its own proposal authoritative. Platform governance declares who controls those resources. The operations owner checks actual credential custody against those declared control relationships before relying on the separation.

Route A requires an independent human authorizer. Route B uses authority already delegated through protected policy; the separate AI reviewer checks the proposal but cannot authorize it. The ticket-feature grant is that saved permission for routine ticket work. For this exercise, the policy selects the standard tier, separate AI review, and a rejection limit of two. Protected Router configuration sets a ten-minute waiting limit; it is not a Policy field. The issuer uses these settings to decide when an attempt can proceed and when a person must intervene.

1.2 Prepare Contract version 1

The coding agent draws the allowed paths from the API, migration, service, and test work. Authentication and billing remain protected. The Contract names unit tests, migration validation, and the API contract test. Its release rule resolves a canary into initial exposure and full promotion before authorization.

Decision	Contract version 1
Requirement	REQ-1432; add low, medium, or high ticket priority
Allowed paths	src/api/schema/ticket.graphql src/db/migrations/** src/services/ticket_service.py src/services/priority_service.py src/services/notification_service.py tests/**
Protected paths	src/auth/** src/billing/**
Merge evidence	unit-tests; migration-validation; api-contract-test
Release	Canary; deployment smoke tests at initial exposure; canary error-rate evidence at full promotion

Table 2. Contract version 1
Contract version 1 gives the feature a boundary and a release plan.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

1.3 Correct the proposal and release a hold

The agent opens a Contract-only pull request at .ceed/contracts/REQ-1432.yaml. Its first proposal accidentally omits the notification service path. That proposal has the right schema shape, but the separate AI review finds that it does not cover the assigned feature. The Router records the completed rejection as review-1 and retains objection O1 with the full explanation outside the Passport. The rejection count becomes one of two.

The agent reads the reasons and adds the exact notification service path. It pushes the corrected proposal to the same pull request. The Router starts review-2 with new inputs and a saved deadline. The reviewer service then loses access. When the deadline passes, the Router closes that review’s authority, retains HOLD-1432, and alerts Elena. The check remains non-passing. No replacement review starts while the hold is open.

A late answer from review-2 goes to the internal log. It creates no rejection, objection, count change, or Passport entry. Elena restores service access and uses her enrolled account to post the supplied command:

```
/ceed resume HOLD-1432 Restored reviewer service access
```

The Router verifies Elena’s account, records who released the hold, what she resolved, and when, once. A later valid duplicate receives an already-released reply. This overdue-review release leaves the rejection count at one. Release permits another attempt; it does not approve the Contract.

Review-3 checks the corrected proposal and the retained objection. Its authenticated result binds O1 and records that the added notification path addresses it. The Router confirms that the review is still authoritative and no hold is open when it records the issuance decision. The earlier rejection remains readable.

1.4 Issue the Contract and check the code

Protected automation authorizes Contract version 1 under route B after the required checks. The approved Contract-only commit lands as a merge commit. The Router verifies that the second parent is the authorized commit, that exactly the canonical Contract path changed, and that the landed bytes equal the approved bytes. Only then does the issuer append contract-issued.

The agent now implements the feature. Its notification path reads a flag in src/billing/config.yaml, so the agent changes that file too. Unit tests, migration validation, and the API contract test pass for the exact base and head. The Router still blocks the merge: Contract version 1 protects billing. The stored decision preserves both the passing evidence and the unauthorized path.

The team now has a precise question for Maya. Does this feature justify changing that billing file? The block is useful because it asks for that decision without discarding the completed implementation or mistaking its tests for permission.

1.5 Authorize the billing exception

Maya confirms the dependency. Repository policy protects authentication and the control plane, but permits independent authorization of this billing exception. The routine grant does not permit it. Contract version 2 therefore uses route A.

The coding agent adds exactly src/billing/config.yaml to allowed scope and adds billing-integration-test to the required merge evidence. The broad src/billing/** protection remains. The exact allowed path wins over that broader Contract pattern for this file alone. Repository-policy protection would win over either Contract pattern.

Maya approves the exact Contract-only proposal. After landing, the Router repeats all three verification steps and the issuer records Contract version 2. The next code revision receives a fresh check with the billing test. It passes under the new authority. Contract version 1’s earlier block stays correct and remains in the Passport.

The sample uses a final candidate that is known before landing and whose checked source becomes the build source. A default-branch position alone is insufficient. If squash or rebase creates another final commit, deployment waits for that commit’s own approved check, using its first parent as the base. That check connects the code being released to the authority and evidence for that exact revision.

1.6 Keep combined work separately authorized

For a separate tabletop variation, combine REQ-1432 with a separately authorized search feature. The main Passport retains the single-change history. In the variation, the integration attributes every changed path to exactly one constituent. REQ-1432 keeps its billing exception; the search change gains no permission from it.

Before the combined commit advances, the Router checks that the subsets are disjoint and complete. It checks each constituent against its own issued Contract and evidence. Their strategy and ordered checkpoint names must match; the release evidence is the union at each matching checkpoint. A changed group receives fresh checks.

If both changes touch the same file, the group waits while the overlap is resolved. If the integration cannot establish exact attribution or cannot let blocked work merge separately, configuration falls back to single-change merges. Each change can then advance under its own authority and checks.

1.7 Advance the canary in order

The release platform builds the artifact from the approved source and creates deployment SYN-DEP-5501. The artifact digest identifies its bytes; the deployment identifier names this release event. A second release of the same artifact would receive another deployment identifier.

At initial exposure, the Router verifies deployment smoke tests and the authenticated strategy assertion for this event. The example accepts self-asserted strategy evidence from the deploying workload because the policy bound at issuance explicitly permits it. The evidence connects the assertion to the same provider event and records the source honestly.

After the canary serves traffic, the observer supplies canary error-rate evidence for full promotion. Evidence from another deployment or another checkpoint cannot stand in for this result. The Router first finds the earlier approved checkpoint for the same deployment, Contract version, and artifact. The platform waits for the new approval before expanding the release.

1.8 Preserve later knowledge and choose the repair scope

After the policy’s 24-hour observation window, the writer records a healthy outcome, completed checkpoints, and no rollback. Nineteen days after deployment, the incident owner associates SYN-INC-77 with it. The original healthy observation remains. The association records the incident system’s opening time and does not designate a primary cause.

Further analysis clears the association. The incident owner appends a retraction for the same deployment and incident, with retrievable evidence. The latest appended association or retraction determines the effective relationship. The investigator can see both the first finding and the correction.

If a repair were needed solely to restore REQ-1432’s intended outcome, the team would issue Contract version 3 and repeat merge, rollout, and observation within this Passport. A fix with independent intent, one spanning several original changes, or one governed through a separate work item needs its own Contract and Passport. The incident association in this example is retracted. For the next genuine repair, the rule tells the team which history to continue.

2 Read the stored records

The examples in this chapter describe wholly synthetic Contract, Policy, Passport, authorization-evidence, deployment-evidence, and local-report records. They illustrate the record structure and relationships; they do not establish live authentication or gate enforcement. The example Passport has ten entries, summarized below so the reader can follow each decision through the retained history. This edition presents the examples within the document; no separate records download accompanies it.

Entry	Record	What the next reader learns
1	Review rejection	The first proposal omitted notification scope.
2	Contract version 1 issued	Route B checked the corrected proposal and retained O1.
3	Merge blocked	Billing crossed the boundary despite passing tests.
4	Contract version 2 issued	Maya authorized the exact billing exception through route A.
5	Merge approved	New candidate and all four named tests pass.
6–7	Rollout approved	Initial exposure precedes full promotion.
8	Healthy observation	The first 24-hour outcome was healthy.
9–10	Association and retraction	Later analysis changes the effective incident relationship.

Table 3. Passport history
The append sequence keeps decisions and corrections in one history.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

2.1 Read a complete Contract

The following complete object represents Contract version 1. The canonical repository location uses YAML; JSON is also a YAML-compatible serialization. The file used for issuance must have the exact bytes identified by its recorded digest. A Contract version identifier belongs in the issuance record, not as an eighth Contract field.

```
{
  "schema_version": "1.0",
  "requirement": {
    "id": "REQ-1432",
    "summary": "Add low, medium, or high priority to support-ticket records.",
    "source": "tracker://REQ-1432"
  },
  "allowed_scope": [
    "src/api/schema/ticket.graphql",
    "src/db/migrations/**",
    "src/services/ticket_service.py",
    "src/services/priority_service.py",
    "src/services/notification_service.py",
    "tests/**"
  ],
  "protected_scope": [
    "src/auth/**",
    "src/billing/**"
  ],
  "required_merge_evidence": [
    "unit-tests",
    "migration-validation",
    "api-contract-test"
  ],
  "risk_tier": "standard",
  "release_rule": {
    "strategy": "canary",
    "checkpoints": [
      {
        "name": "initial-exposure",
        "strategy_verification_required": true,
        "required_release_evidence": [
          "deployment-smoke-tests"
        ]
      },
      {
        "name": "full-promotion",
        "strategy_verification_required": false,
        "required_release_evidence": [
          "canary-error-rate"
        ]
      }
    ]
  }
}
```

Contract version 2 adds the one billing path and its test to the object above. The policy retains the routine grant and the independent authorizer predicate. The standard tier’s policy does not change to make the exception routine.

2.2 Follow an approval to its evidence

Each issuance entry points to the authorization evidence for its Contract version. That evidence binds the exact Contract commit and bytes to the requirement assignment, policy bytes, tier, actors, control declarations, review, objection history, and landing evidence. Authorized commits and landed merge commits remain distinct.

In this example, evidence references use a local relative URI and a SHA-256 digest, a fingerprint of the exact resource bytes. An optional JSON pointer selects a particular objection only after the whole resource digest is verified. Resolving a reference requires checking those bytes before

interpreting the selected content. Changing a referenced file requires updating the dependent references; changing an issued version requires a new authorization.

The operational review log illustrates configuration and lifecycle evidence outside the standardized Passport format. It explains the review attempts; it is not another Passport entry. The declaration scenario in §6 illustrates a separate internal coverage record. Readers should distinguish these teaching examples from the complete Contract object printed above and from the requirements that govern an implementation.

2.3 Read the local report as a snapshot

The example local report is calculated from a snapshot of the retained history without network access. Its September period uses UTC, and its October 2 generation time allows the 30-day cohort to mature. The report identifies the selected store snapshot. A rejection does not count as an issued version or merge evaluation.

Measure	Synthetic result	Interpretation
Changes evaluated	1	One change enters the cohort.
Merge evaluations	2	Blocked first candidate; approved later candidate.
Attempt block rate	1 / 2	One of the two merge evaluations blocks for authority.
Change block rate	1 / 1	The first evaluation of the only change blocks.
Contract versions issued	2	Rejection is excluded.
Recorded deployments	1	One fresh production outcome; later findings do not create deployments.
Active late associations	0	The retraction clears the association while preserving its history.

Table 4. Local report
The local report separates repeated decisions from changes and deployments.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

The report cannot discover an incident or deployment absent from its input history. A missing source link remains a retrieval problem even when the Passport is well formed. The investigator starts with the billing block, follows Contract version 2 to Maya’s exact decision, then follows the deployment and incident evidence. That route connects the summary counts to the decisions and source facts behind them.

3 Compare the complete work

The engineering lead wants to know what changes when the team adopts the framework. The comparison begins with a competent current workflow: an assigned tracker item, repository guidance, review, tests, protected releases, and retained incident records. That workflow can catch the billing edit and preserve its approval well.

3.1 Set matched conditions

Use the same feature, initial repository state, billing dependency, test obligations, canary stages, and investigation questions in both runs. Give both teams equivalent access to the tracker, CI results, release records, and incident findings. Choose the same target authority, retention, and recovery capabilities. Record any capability the current configuration lacks and the work needed to add it.

Choose comparable participants and record familiarity with each method. Alternate run order or use comparable separate groups to reduce learning carryover. Keep the requirement and planted dependency fixed. Freeze the target before comparing effort; a cheaper workflow that omits a required control is not the same comparison.

Work category	Competent current or extended workflow	CEED workflow
Setup and maintenance	Configure guidance, reviewers, release controls, retention, and any local authorization model.	Connect protected policy, issuance, exact-candidate gates, source evidence, and retained records.
Each authority version	Clarify scope and preserve the review decision; include repeated guidance.	Prepare and authorize each Contract version; include separate AI review, retries, and issuance verification.
Each candidate and checkpoint	Run existing checks and verify the required permissions and release conditions.	Run the same specialist checks; add the Router’s authority and evidence decision.
Exceptions and rework	Reviewer catches billing; Maya decides; team repairs and retains the decision.	Gate blocks billing; Maya authorizes Contract version 2; team checks the revision again.
Later investigation	Retrieve permission, candidate, release, and incident records from the existing arrangement.	Follow Passport references to the same source facts and inspect corrections.
More volume or assurance	Estimate added integrations, queues, retention, verification, and staffing for the common target.	Estimate the same pressures, including Router service load and record maintenance.

Table 5. Matched comparison
Matched comparison follows the same obligations through both workflows.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

3.2 Run delivery and investigation in both arms

In the current-workflow arm, the reviewer finds the billing edit before merge. Maya considers the dependency, authorizes the exception through the existing protected process, and requires the billing test. The team retains the approval and the tested revision, then performs the two-stage canary. Later, the investigator reconstructs why billing changed and which evidence supported release.

In the CEED arm, use the lifecycle in Chapter 1. Retain the rejected proposal, the operational hold, both issued Contract versions, the blocked candidate, and the approved revision. During

investigation, ask the same questions and require the same source evidence. Include time spent repairing a broken link or resolving an ambiguous approval in either arm.

The observer records active human minutes, elapsed waiting, and machine consumption separately. AI input and output tokens, CI minutes, and storage have different units; retain the provider, model or service, date, and price basis before converting them to money. Shared tests, tracker work, and incident analysis count once per arm. A trace identifier links a shared activity to every use without charging it repeatedly.

The useful result is a map of effort and missing capability. One workflow may need more preparation and less retrieval, while another already provides the needed history. The comparison lets the lead decide where a common Contract and Passport add enough value to justify their operation.

4 Record effort as it happens

An observer records each activity when it occurs. Include the workflow arm, change and activity identifier, actor, phase, start and end, active effort, wait reason, machine units, and source evidence. Tag first-trial familiarization separately from recurring work. Mark process trial or Router pilot so manual rehearsal is never pooled with operating enforcement.

4.1 A filled worksheet

Table 6 uses invented values to demonstrate recording. C means the competent current workflow; F means the CEED workflow in an illustrative Router pilot. Both arms follow the same comparison target. The proposer’s active minutes record the person supervising Contract preparation; the agent’s tokens are recorded separately. In the F arm, the Router requests AI review, waits for the service, and holds the attempt for Elena to release, as in §1.3. These are imagined pilot observations, not measurements from a running Router. The observer records active review, service waiting, and machine consumption separately so the lead can account for the full review cycle.

Arm / activity	Actor	Phase	Active min	Wait min	Machine units
C / learn	Observer	First trial	20	0	—
F / learn	Observer	First trial	35	0	—
C / guidance	Developer	Recurring	18	5	2,000 input / 700 output tokens
F / Contract version 1	Proposer	Recurring	22	10	2,600 input / 900 output tokens
C / review	Reviewer	Recurring	17	25	30 CI minutes
F / review and hold	Review operator / Elena	Recurring	14	40	3,400 input / 800 output tokens; 30 CI minutes
C / exception	Maya	Recurring	12	30	—
F / Contract version 2	Maya	Recurring	15	30	—
C / rework	Developer	Recurring	25	0	12 CI minutes
F / rework	Developer	Recurring	25	0	12 CI minutes
C / retrieval	Investigator	Recurring	24	8	2 MB retained
F / retrieval	Investigator	Recurring	18	12	3 MB retained

Table 6. Filled measurement worksheet
Invented values compare the current workflow (C) with a Router pilot (F), separating effort, waiting, and consumption.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

The paired rework rows deliberately use the same effort and CI consumption. The worksheet does not credit the framework for implementation work both teams must do. In the F review row, the 40 minutes record elapsed waiting, including the service interruption and hold. The 14 active minutes record the review operator’s and Elena’s work; the tokens belong to the review service. Dashes mark machine use not recorded for this illustration. Keeping these columns separate makes the interruption’s cost visible.

Attach the activity log to add timestamps, wait causes, pricing sources, and evidence references. Do not sum overlapping waits into end-to-end duration. Derive elapsed delivery time from the actual start and finish events. If several people work together, sum their active person-minutes and retain the elapsed interval separately.

4.2 A blank worksheet for the next change

Trial header: change _____; workflow arm _____; process trial or Router pilot _____; observer _____; comparison target _____; date _____; participant familiarity _____. Use the same header for both arms and keep shared-control identifiers visible.

Activity / source	Actor	Phase	Start / end	Active / wait	Machine units
Repeated guidance / ____	____	____	____	____	____
Authorization / ____	____	____	____	____	____
Review / ____	____	____	____	____	____
Rework / ____	____	____	____	____	____
Evidence retrieval / ____	____	____	____	____	____
Setup or maintenance / ____	____	____	____	____	____
Candidate or checkpoint / ____	____	____	____	____	____

Table 7. Blank measurement worksheet
A reusable worksheet records the same categories in each comparison arm.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

For each row, retain a unique activity ID, the wait reason, evidence location, any shared-control ID, and the unit price source if money is calculated. Add rows for AI review retries, operational recovery, source retention, or stronger assurance work. Review first-trial learning and recurring operation separately before drawing an investment conclusion.

5 Calibrate the decisions

Before the trial, the proposer, independent authorizer, checker, and operations owner review the cases below. Each participant first records a decision and reason without seeing the answer. Then the group compares the boundary, evidence, route, and responsible person. The aim is a usable shared standard for the next real proposal.

Synthetic proposal	Expected decision and reason
Allow src/** for the ticket-priority feature.	Narrow the proposal to the intended paths. A correct path check can enforce a broad permission perfectly; it cannot decide that the permission was sensible.
Allow only the API file; omit the migration and notification service.	Return for correction. The proposed boundary cannot support the assigned feature. Add justified paths rather than granting the whole service tree.
Permit billing through the routine ticket grant.	Escalate to route A. The routine grant excludes billing; an independent person can consider the exact exception if repository policy permits it.
Use passing unit tests from the previous head.	Block this candidate until required evidence names both its actual base and head. A passing result for different code is insufficient.
Promote canary with evidence from the previous deployment.	Wait for evidence bound to this event and checkpoint. Check the earlier checkpoint approval too.
A required reviewer times out; choose mechanical-only.	Keep the hold. An enrolled handler resolves the problem and releases it; fresh review follows the policy-selected method.
Maya is unavailable for the billing exception.	Use another eligible independent authorizer if policy permits. Record waiting and staffing; delay is not authority to downgrade the route.

Table 8. Calibration cases
Calibration tests scope, evidence, escalation, and operational ownership.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

The group records who receives each escalation and who can cover an absence. It agrees a local service expectation for route A waiting, then measures the actual delay in the trial. A missed expectation prompts staffing or process improvement; it never grants automatic approval.

Finish by revisiting REQ-1432. Keep unresolved disagreements with an accountable owner before relying on the process. Ask the trial’s human proposer to justify every allowed path, the checker to identify the evidence for each gate, Maya to explain her authorization, and Elena to explain her hold release. Their answers give the group a common next action and a named person for each decision.

6 Map actual enforcement

An Enforcement Coverage Declaration is the organization’s protected internal account of its adopted delivery scope, gate connections, owners, verification, and gaps. It exists before reliance on governed actions or a coverage claim. It is maintained after relevant changes, rather than prepared for every build.

This wholly synthetic declaration describes a separate tabletop exercise: a faulty configuration and its repair. It is not the configuration history of the main Passport. Real declarations, validation views, histories, and protected references remain internal. The example lets platform teams practice finding gaps and tracing the evidence needed to close them.

6.1 Read the synthetic inventory

SYN-TICKET-DELIVERY covers one synthetic ticket repository’s main ref, staging and production, Contract issuance, code merge, rollout, and Passport writing. The platform owner maintains the map. The search repository is outside this inventory, and undocumented administrator routes remain an explicit unknown. An exclusion cannot waive a required gate inside the adopted scope.

Connection and owner	Coverage	Verification in revision 1
Contract issuance platform-owner	mandatory	verified-for-stated-test-scope Wrong proposal bytes and unrelated landed files prevent issuance.
Code merge platform-owner	mandatory	verified-for-stated-test-scope Missing approval prevents merge for the stated base and head.
Initial exposure release-owner	mandatory	declared-only Workflow is configured to wait; refusal has not yet been exercised.
Full promotion release-owner	mandatory	failed Simulated timeout permits promotion in configuration rev1.
Passport append records-owner	mandatory	verified-for-stated-test-scope Producer credential cannot append or change write permissions.
Agent local path hint developer-owner	advisory	declared-only Hook offers feedback but cannot authorize delivery.
Emergency release operations-owner	absent	unknown No Router invocation; authoritative platform audit required.

Table 9. Synthetic enforcement inventory
Synthetic revision 1 exposes a promotion failure instead of claiming complete enforcement.
CEED in Practice, Public Review Draft 2, 2026 · CC BY 4.0 · https://ceedframework.com/

Mandatory describes the intended enforcement connection. Verified describes a property checked within a stated scope. Those are separate questions. In revision 1, the promotion gate is mandatory but its synthetic timeout test fails. The owner withholds reliance on rollout enforcement until the fault is repaired and tested.

6.2 Retain the evidence and revision history

In this scenario, each coverage record identifies its property, assessment date, method, assessor and control relationship, exact configuration reference, and supporting evidence reference. The platform owner also administers configuration, so the assessment is not independent. Those references identify configuration and evidence locations that an internal implementation would protect and resolve.

Revision 1 applies from September 1 until September 3. Configuration changes on September 2. The owner reassesses revision 2 on September 3 and retains the earlier record, its period, and the intervening unassessed change. The later check does not retroactively verify that interval. The retained dates let an investigator identify the gap.

In revision 2, both rollout gates pass the stated synthetic refusal and timeout cases. The history still retains revision 1’s failed promotion test. Scope, policy, enrollment, credentials, store protection, providers, integrations, bypass routes, and facts underlying verification all trigger reassessment when relevant.

6.3 Protect the map and use its gaps

Only authenticated, explicitly authorized governance identities can maintain the real declaration and its access policy. Producer execution credentials cannot create, approve, replace, or delete authoritative revisions or weaken that protection. Authorized internal reading remains possible. A trusted administrator’s combined roles are disclosed rather than described as independent.

Table 9 summarizes selected parts of the declaration scenario; Specification, Public Review Draft 2, §16.2 defines the complete required content. A generated view would also identify its exact source revision and byte digest, with correspondence established. The Draft 2 Conformance Pack does not include a separate declaration validation schema. This scenario illustrates the requirements without claiming validation against such a schema.

Path authorization makes the approved boundary enforceable; software security determines whether the implementation inside it is acceptable. The platform owner keeps authority and retained evidence outside the producing agent’s control, while security staff assess behavior within the allowed files.

Missing or stale declaration evidence prevents a current conformant deployment-coverage claim. It does not become a new per-change authorization input or automatically create a delivery hold. Required merge and rollout checks retain their own obligations. The owner uses this map to direct inspection and repair, then records what was actually verified.

7 Run the first trial

Choose a familiar change and retain its real requirement in the existing tracker. Name the human proposer, independent authorizer, checker, operations owner, and observer. Prepare the boundary with the sources the team already uses. Use the calibration cases to expose disagreement before it reaches a delivery gate.

Run the process trial with existing approvals and label its retained decisions as trial material. Record preparation, repeated guidance, review, waits, rework, and retrieval as they happen. Compare a competent alternative under the same target conditions. The team should finish with an actual decision about scope and a history another reader can follow.

A live Router pilot adds protected integrations and demonstrated waiting. Start with issuance and merge, verify the failure cases, and maintain the internal declaration. Add rollout only after testing event-bound evidence and each required checkpoint. Verify retention, recovery, and authorized writing as part of that same operational responsibility.

Review the results with the people who made and consumed the decisions. Decide which preparation can be reused, which escalation needs a different owner, and which evidence was difficult to retrieve. Use those findings to improve the next proposal, assign the follow-up work, and choose the next trial.

Publication information

CEED in Practice · Public Review Draft 2 · October 26, 2026. Author: Viktor Barakoski. This companion is explanatory; the CEED Specification and Change Passport Schema governs normative definitions. Document editions, Contract versions, schema version 1.0, and Router software versions identify different things.

Copyright © 2026 Viktor Barakoski. This document is licensed under Creative Commons Attribution 4.0 International (CC BY 4.0). The CEED Router implementation source will be licensed under Apache License 2.0. These licenses do not grant endorsement, official status, trademark rights, or certification by the CEED Project.

CEED™ and CEED Framework™ are trademarks of Viktor Barakoski. Third-party names identify their respective owners and imply no affiliation, sponsorship, or endorsement. The framework is independently authored; the author’s views are personal.

Related sources: CEED Specification and Change Passport Schema, Public Review Draft 2, §§4–12 and 16.2; The CEED Framework Whitepaper, Public Review Draft 2, Chapters 2–10; CEED Conformance Pack, Public Review Draft 2, schema manifest and test coverage map. Official resources: https://ceedframework.com/.