

# One agent action. Reviewable evidence of every step.

Bind what an agent proposes to independently supplied authority, constrain the effect, and reconstruct what happened from the retained record.

BOUNDED SYNTHETIC REFERENCE

One refund workflow · synthetic SQLite destination · one host · no production credentials. Not a production deployment, OS security boundary or compliance certification. Public release is pending approval.

A WORKED EXAMPLE: ONE REFUND AGENT, FOUR REQUESTS. EACH LEAVES A RECONSTRUCTABLE RECORD.

|                                                                                                                       |                                                                                                                  |                                                                                                                                       |                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>\$250 refund requested</b><br><b>ALLOWED</b><br>Within the grant. Rechecked just before the effect; executed once. | <b>\$750 refund requested</b><br><b>DENIED</b><br>Outside the grant. No attempt is made; the denial is recorded. | <b>Same message delivered twice</b><br><b>ONE REFUND</b><br>Same proposal, same effect ID.<br>Redelivery returns the original record. | <b>Same refund, new proposal</b><br><b>TWO REFUNDS POSSIBLE</b><br>A different valid proposal gets a new effect ID. Effect-ID dedup is not business-intent dedup. |
|-----------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|

As agents move from drafting text to issuing refunds and changing records, a log line is no longer enough. **A chatbot is reviewed by what it said; an agent must be reviewed by what it was allowed to do, what it attempted and what the destination was observed to do.** The Open Control Stack keeps those facts separate and connected.

FOLLOW ONE ACTION ALL THE WAY THROUGH

| DECLARE                                                                                                 | CONTROL                                                                                                                                 | REPLAY                                                                                                           | EVIDENCE                                                                                                    |
|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| <b>What may the agent propose?</b><br>Action · target · scope · effect limits<br><b>Action Manifest</b> | <b>Is this effect allowed now?</b><br>Authority supplied outside the request · rechecked just before the effect<br><b>Control Plane</b> | <b>What does the record show?</b><br>Attempted · acknowledged · observed · reconstructed<br><b>Replay Bundle</b> | <b>What can reviewers trace?</b><br>Linked to source · gaps left visible<br><b>Governance Evidence Pack</b> |

PROPOSED → PERMITTED → ATTEMPTED → OBSERVED *Independent verification is outside the reference.*

## How it works

- **Authority comes from outside the request.** Authority Context is supplied by a trusted resolver; the incoming message cannot grant it.
- **Critical inputs are rechecked before each effect.** Per effect only: not whole-workflow or check-to-commit atomic.
- **Execution is bounded.** A changed payload, target or adapter produces no effect.
- **The record survives ambiguity.** A timeout is recorded as unknown, never as “nothing happened.”
- **Review is reconstructable.** Trace every step without rerunning anything.

**Evidence is not permission.**  
A receipt, signature, log or model response can be valid evidence and still authorize nothing.  
A denial is evidence too: it shows the control made a decision.  
**That separation is the point.**

## WHERE ENFORCEMENT STOPS

|                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Per effect, not per workflow.</b> Decide, execute and reconcile are not one atomic transaction; authority can change between the recheck and the destination commit.<br><b>Application-level, not OS confinement.</b> A privileged host process can bypass it. Live OpenShell confinement is unqualified. | <b>Effect ID, not business intent.</b> Redelivery is caught; a second, different proposal for the same refund is not prevented on the default path. An optional refund-intent profile, run separately, blocks it.<br><b>Synthetic, not production.</b> No production resolver authentication, real accounts, distributed budgets, remote finality or exactly-once delivery. |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Current release evidence:** all 35 acceptance scenarios resolved, required suites run twice · **fresh Oct 13 reproduction:** 1,009 Python tests × 2 runs, 20 Node tests. Public contracts, code, tests and retained failures, so you can rerun it yourself.

**Bring one workflow your agents should run.**  
We will map where its authority, destination and enforcement differ from the reference, and what evidence each difference needs.

**cogno.us**  
Learn more

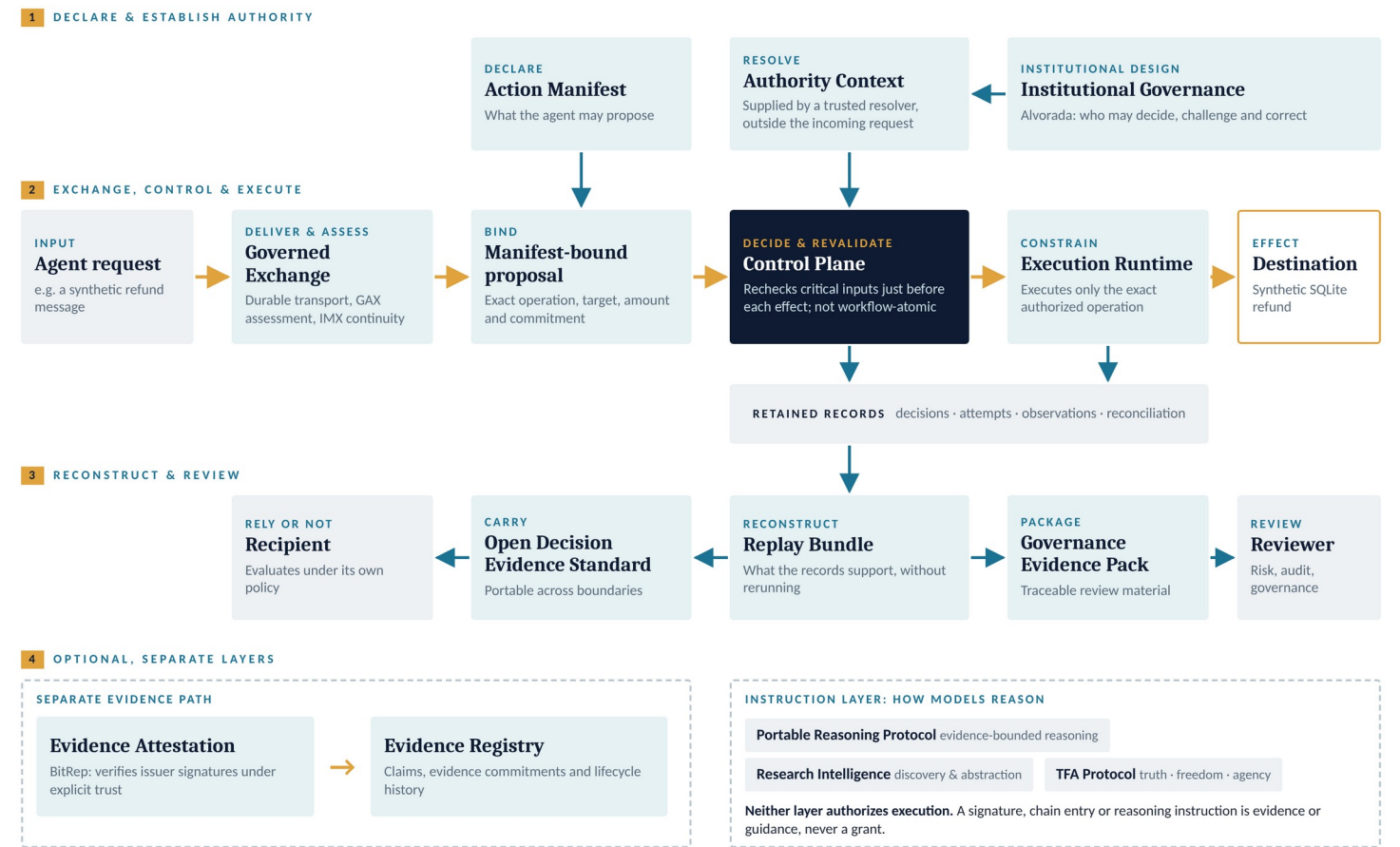
# Thirteen components. Six groups. One rule.

Each component owns one responsibility and states what it does not establish. Governed Exchange connects the groups. The rule: evidence is not permission.

|                                     |    |                                     |                                                                 |
|-------------------------------------|----|-------------------------------------|-----------------------------------------------------------------|
| 01 · KNOWLEDGE & PROVENANCE         | 01 | Cognous Evidence Registry           | Local-chain claims, evidence commitments and history.           |
|                                     | 02 | Cognous Evidence Attestation        | BitRep: verify issuer signatures under explicit trust.          |
| 02 · INSTITUTIONAL GOVERNANCE       | 03 | Cognous Institutional Governance    | Alvorada: authority, challenge and correction for institutions. |
| 03 · DECLARATION & POLICY CONTROL   | 04 | Cognous Action Manifest             | Declare the action before evaluating permission.                |
|                                     | 05 | Cognous Control Plane               | Evaluate proposals against authority; keep the decision record. |
| 04 · CONSTRAINED EXECUTION          | 06 | Cognous Execution Runtime           | Constrained execution beneath current authorization.            |
| 05 · REPLAY & EVIDENCE              | 07 | Cognous Replay Bundle               | Reconstruct what the retained records support.                  |
|                                     | 08 | Cognous Governance Evidence Pack    | Turn runtime records into reviewable governance evidence.       |
|                                     | 09 | Open Decision Evidence Standard     | Portable decision evidence across boundaries.                   |
| 06 · REASONING & HUMAN OVERSIGHT    | 10 | Portable Reasoning Protocol v1.0    | Instructions for evidence-bounded reasoning.                    |
|                                     | 11 | Research Intelligence Protocol v1.0 | Disciplined discovery and abstraction, kept separate.           |
|                                     | 12 | TFA Protocol (S43)                  | Truth · Freedom · Agency.                                       |
| CROSS-CUTTING · CONNECTS THE GROUPS | 13 | Cognous Governed Exchange           | Governed exchange and continuity for a bounded workflow.        |

## HOW THE COMPONENTS WORK TOGETHER

One synthetic refund, from request to reviewable evidence. Amber arrows carry the action; teal arrows carry authority in and records out. Dashed panels never sit on the authorization path.



# One refund agent, seven cases.

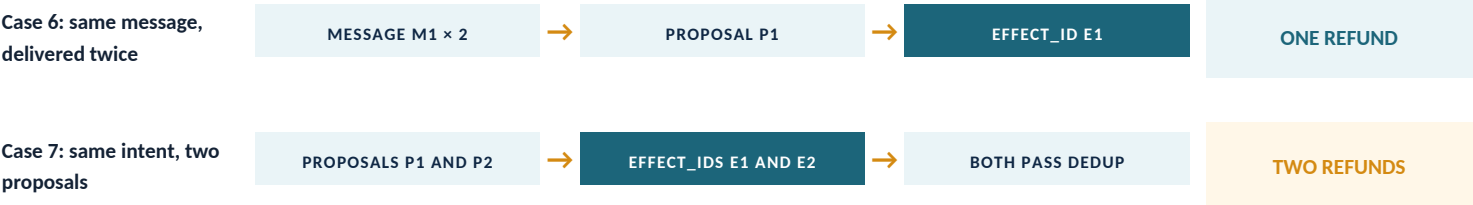
Each case runs in the bounded synthetic reference. Read the last column as carefully as the outcome.

THE SETUP

**Agent:** refund agent. **Declared action:** routine refund in the Action Manifest. **Authority:** a grant supplied by a trusted resolver, outside the incoming request. **Destination:** a synthetic SQLite refund table.

| CASE                                       | WHAT HAPPENS                                                                                                                     | WHAT THE RECORD SHOWS                                                                        | WHAT IT DOES NOT ESTABLISH                                                                                                      |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|
| 1 Refund within the grant                  | Control Plane allows; critical inputs are rechecked just before the effect; the runtime executes the frozen operation once.      | Decision, revalidation, attempt and an observed “applied” state.                             | Independent verification; that authority could not change between recheck and commit.                                           |
| 2 Refund outside the grant                 | Control Plane denies. No attempt is made.                                                                                        | The proposal, the denial and the authority it was evaluated against.                         | That the grant itself was well designed.                                                                                        |
| 3 Payload changed after approval           | Amount, target or adapter no longer matches the frozen envelope. The runtime rejects it.                                         | A rejected attempt and no effect.                                                            | OS confinement: a privileged host process could bypass the package.                                                             |
| 4 Timeout after the attempt                | Acknowledgement unknown. Reconciliation concerns the original effect; a later observed absence does not authorize a replacement. | Unknown acknowledgement, observations and reconciliation, with the interrupted history kept. | Remote finality or exactly-once delivery.                                                                                       |
| 5 Authority changes before recovery        | Recovery is denied under current authority.                                                                                      | Both facts, separately: the historical applied effect and the current denial.                | Rollback or cancellation of the original refund.                                                                                |
| 6 Same message delivered twice             | Same proposal, same effect_id. Redelivery returns the original retained artifacts.                                               | One effect, with the redelivery linked to it.                                                | Cross-host or remote exactly-once delivery.                                                                                     |
| 7 Same refund, new proposal<br>KNOWN LIMIT | A second valid proposal for the same customer refund has a new commitment, so a new effect_id. Both can execute.                 | Two separate effects, each fully traceable.                                                  | Business-intent deduplication on the default path. The optional refund-intent profile blocks it, but is not enabled by default. |

## WHY CASES 6 AND 7 DIFFER: EFFECT ID IS NOT BUSINESS INTENT



Deduplication keys on the effect a specific proposal names, not on what the business meant. The selected runtime includes an optional refund-intent profile that binds one intent to one effect, but the default path does not enable it. Otherwise a deployment must guard business intent itself, for example with an idempotency key on the destination.

## ASK SIX QUESTIONS OF ANY REFUND

1. Who supplies authority? Institution, mandate, grant, approval, policy, time.

2. What was proposed? Operation, target, amount and commitment.

3. What was allowed just before the effect? The decision and its revalidation.

4. What happened at the destination? Attempt, acknowledgement, observation.

5. Can the record be traced? Through Replay, Evidence Pack and ODES.

6. What remains to be qualified? The risk register and support status.

# What the reference shows, and where it stops.

Read this before the component pages. Each accepted claim is tied to executed evidence and a stated boundary.

## WHAT THE REFERENCE IS

|                                                                                |                                                                                                         |                                                                                         |                                                                                   |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>BOUNDED</b><br>One refund workflow and a single agent, followed end to end. | <b>SYNTHETIC</b><br>A SQLite destination. No real accounts, payment services or production credentials. | <b>LOCAL</b><br>Cooperating writers on documented local Linux filesystems, on one host. | <b>OPEN</b><br>Public contracts, code, tests and retained failures you can rerun. |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|

## WHERE ENFORCEMENT STOPS

|             | WHAT HOLDS IN THE REFERENCE                                                                                                 | WHAT DOES NOT                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| Authority   | Authority Context is supplied by a separately configured trusted resolver and resolved outside the incoming request.        | Production resolver authentication and credential custody.                                                    |
| Effect time | Authorization-critical inputs are revalidated immediately before each effect. Individual record transactions are protected. | Check-to-commit atomicity by default. The optional same-host profile (C1) is off by default; no C2/C3 claims. |
| Duplicates  | Same-effect deduplication by effect_id, including across processes sharing one SQLite file.                                 | Business-intent deduplication on the default path; the refund-intent profile is separate and optional.        |
| Execution   | Application-level restrictions on the supported path, using a frozen operation snapshot.                                    | OS confinement, live OpenShell and deployment-wide bypass resistance.                                         |
| Delivery    | Reconciliation of the original effect, with interrupted history kept.                                                       | Rollback, cancellation, remote finality and exactly-once delivery.                                            |

## ACCEPTED EVIDENCE

| EVIDENCE                                       | WHAT WAS DEMONSTRATED                                                                                                                                                               | OUTSIDE THE CLAIM                                                                                      |
|------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Current release gate                           | All 35 acceptance scenarios resolved for the selected generation, required suites run twice. Fresh Oct 13 reproduction: 1,009 Python tests in each of two runs, plus 20 Node tests. | Production authentication, real-world verification, live OpenShell and publication approval (on HOLD). |
| Shared-store persistence                       | Cooperating same-host record transactions preserve supported concurrent writes and process-interruption behavior on documented local Linux filesystems.                             | Whole-workflow atomicity, cross-host persistence and remote exactly-once delivery.                     |
| Optional same-host atomic profile (C1)         | 73 tests passed in a separate run; authority, claim, budget and effect ordered in one same-host SQLite transaction.                                                                 | Default enablement; external or distributed atomicity.                                                 |
| Recovery qualification                         | Same-effect continuity, original-effect reconciliation, accepted and rejected observations, and changed-authority denial kept separate from historical execution.                   | Retry permission from absence, rollback, cancellation or remote finality.                              |
| Protected-worker campaign (historical profile) | Twelve isolated cases and 17 verifier tests passed on Ubuntu 22.04.5, Linux 6.8.0-1064-azure, bubblewrap 0.6.1 and Python 3.11.16.                                                  | OpenShell, arbitrary agents, production credentials or general production non-bypassability.           |

### READ THE OUTPUT AS INPUT

The reference's output is input to human governance review. It does not amend policy, confer authority or approve deployment. This describes the selected lock, reviewed at hub cdbd4de on 14 October 2026; newer releases are not selected automatically. Publication and operational trust remain on HOLD; the earlier 915-test campaign is historical.

# Cognous Evidence Registry

Register claims and evidence, and keep the whole history.

A protocol and reference implementation for registering scientific claims, linking evidence commitments and preserving revision and lifecycle history. In the accepted slice a local blockchain contract is authoritative for those records, and Evidence Attestation verification runs independently off-chain.

THE PROBLEM

**Inclusion is mistaken for truth**  
A registry must show who registered a record, what evidence was committed and what changed. Chain inclusion cannot say whether a claim is true or the evidence independent.

→

WITH EVIDENCE REGISTRY

**Commitments and lifecycle, labeled honestly**  
Wallet attribution, chain inclusion, signature assurance and scientific interpretation each stay separately labeled.

WHAT IT PROVIDES

- **Claim registration:** content commitments and wallet attribution recorded through the local contract.
- **Evidence relationships:** supporting and contradicting evidence linked, without turning votes or duplicates into truth.
- **Revision and lifecycle:** prior content preserved through revision, withdrawal and supersession, with explicit ownership rules.
- **Read-only reconstruction:** state and history rebuilt by an indexer with documented reorganization handling.
- **Attestation binding:** content bindings recomputed and issuer signatures verified off-chain with the exact accepted verifier revision.

WHAT IT DOES NOT CLAIM

No public-chain deployment, audited production contract, remote consensus finality, authenticated RPC verification or on-chain attestation-verification bridge. Legacy FastAPI and store surfaces are non-authoritative demonstrations, and reputation-weighting goals are not implemented guarantees.

IN THE REFUND WORKFLOW

A researcher registers a claim commitment on the local chain and attaches an evidence commitment. An independent reader reconstructs the record, retrieves the expected evidence bytes and checks the signed statement with Evidence Attestation. Withdrawal or supersession preserves history rather than silently replacing it.



AT A GLANCE

|              |                                               |
|--------------|-----------------------------------------------|
| TYPE         | Protocol & reference implementation           |
| INTERFACE    | Local blockchain reference 0.1.0              |
| ACCEPTED PIN | d5e45d2                                       |
| SOURCE       | github.com/cogno-us/cognous-evidence-registry |
| FORMERLY     | The Index                                     |

TRY IT

Register a claim, revise it, then reconstruct its full history from the indexer.

COMPONENT01

# Cognous Evidence Attestation

Know who signed it, under trust you chose.

The BitRep binary-attestation protocol and its reference verifier. The accepted v1 interface checks Ed25519 signatures against an operator-configured issuer-key trust snapshot and evaluation time. Success means issuer\_signature\_only under those conditions, not factual truth or institutional authority.

THE PROBLEM

**Signatures without context mislead**  
Unverified identity flags, copied verification JSON and unsigned legacy imports cannot show who signed a statement or whether the key is still admissible.

→

WITH EVIDENCE ATTESTATION

**Strict statements, explicit trust**  
Signed statements checked against issuer keys you configure, at a stated evaluation time, with conservative handling of revocation.

## WHAT IT PROVIDES

- **Strict signed statements:** version, algorithm, issuer and key, subject, type, audience, content digest and validity times.
- **Explicit trust snapshot:** operator-controlled issuer keys; no default trusted issuer is supplied.
- **Current-state verification:** signature, key status, audience and time checks, with conservative revocation behavior.
- **Admission and duplicates:** v1 records kept apart from legacy data; identical resubmission is idempotent, conflicting content rejected.
- **Fail-closed assurance:** malformed or unsupported inputs rejected; placeholder privacy and platform checks kept outside accepted assurance.

## WHAT IT DOES NOT CLAIM

No reputation score, real zero-knowledge proof, external identity verification, blockchain inclusion verification, signed portable result receipt or caller-selected historical verification in v1. Issuer enrollment, secure time, snapshot distribution, anti-rollback and production transport authentication remain deployment work.

## IN THE REFUND WORKFLOW

A downstream system such as the Evidence Registry supplies the expected subject, audience and content binding. The verifier checks the statement locally or through the reference service. The consumer still compares the digest with the exact bytes it uses and decides how much weight to give the assertion. Two signatures over the same bytes are not independent corroboration.



## AT A GLANCE

|              |                                                  |
|--------------|--------------------------------------------------|
| TYPE         | Protocol & reference verifier                    |
| INTERFACE    | Verification contract v1 (Ed25519)               |
| ACCEPTED PIN | 5b5077d                                          |
| SOURCE       | github.com/cogno-us/cognous-evidence-attestation |
| FORMERLY     | BitRep                                           |

TRY IT

Configure a trust snapshot with one issuer key, then verify a valid, an expired and a wrong-audience statement.

COMPONENT02

# Cognous Institutional Governance

Decide who may decide, before any agent acts.

Alvorada is an open research framework for businesses, public bodies and communities, with or without AI. It combines proposed constitutional designs, proportional governance, domain guidance, decision templates and a public-stack implementation profile.

THE PROBLEM

**Authority is assumed, not designed**  
Institutions often cannot say who may decide, how a decision can be challenged or who verifies the correction. AI can help prepare evidence, but it cannot create that authority.

→

WITH INSTITUTIONAL GOVERNANCE

**Authority with challenge and remedy**  
A proposed charter, review paths matched to consequence, and a proposed Authority Context profile that maps established institutional authority into software.

## WHAT IT PROVIDES

- **Constitutional design:** a common charter, proposed profiles and the preserved original Constitution v0.1, open to inspection.
- **Proportional governance:** review intensity matched to consequence, reversibility and uncertainty.
- **Challenge and remedy:** dissent, review, succession, emergencies and correction made visible in the design.
- **Domain guidance:** the framework applied to specific contexts, with departures and local decisions recorded.
- **Authority Context profile:** institutional requirements downstream software can consume, without moving runtime enforcement into the framework.

## WHAT IT DOES NOT CLAIM

Proposed guidance and constitutional research, not adopted governance, operative law or demonstrated institutional superiority. Repository acceptance does not ratify a constitution. Synthetic adapters did not show better outcomes than conventional controls, and runtime enforcement belongs downstream.

## IN THE REFUND WORKFLOW

Choose a domain guide and a review path, then follow the invented service case through decision, challenge and correction. For software, independently established authority is mapped into the proposed Authority Context 0.1.0 profile. A trusted resolver supplies it to the Control Plane, outside any incoming request. The controller must still authenticate that resolver and enforce permissions before each effect.



## AT A GLANCE

|              |                                                      |
|--------------|------------------------------------------------------|
| TYPE         | Open research framework                              |
| INTERFACE    | Authority Context 0.1.0                              |
| ACCEPTED PIN | fb3d979                                              |
| SOURCE       | github.com/cogno-us/cognous-institutional-governance |
| FORMERLY     | Constitutional Governance for Institutions           |

TRY IT

Pick one domain guide and walk the invented service case from decision to correction. Then list where your own authority actually comes from.

COMPONENT03

# Cognous Action Manifest

Name the action before anyone asks whether it is allowed.

A lightweight format, Python validator and CLI for describing the actions an AI agent may propose. Manifest 1.1 adds deterministic runtime bindings, so a downstream controller can compare a proposal with one specific declared action.

THE PROBLEM

**A tool name is not a policy boundary**  
The same integration can read data, draft a message, issue a payment or delete a record. Reviewers cannot govern an action surface no one has written down.

→

WITH ACTION MANIFEST

**An explicit, reviewable action inventory**  
Stable action and adapter IDs, finite targets and effect limits that survive the handoff from design review to runtime control.

### WHAT IT PROVIDES

- **Action inventory:** tools, action types, owners and intended scope, identified before execution.
- **Governance requirements:** authority, review and reliance requirements declared, without issuing grants.
- **Payload handling:** sensitive fields and redaction hints described for downstream record handling.
- **Runtime binding:** action and adapter IDs, finite targets, effect limits and Authority Context references (1.1 profile).
- **Validation:** Pydantic models, JSON schemas and the aam CLI to validate, summarize and list actions.

### WHAT IT DOES NOT CLAIM

Declaration and validation, not a security boundary or runtime access control. A valid manifest does not establish institutional legitimacy, current approval, a completed effect or compliance. The exporting system must apply redaction hints itself.

### IN THE REFUND WORKFLOW

A reviewer declares a routine refund with its allowed adapter, target and effect limits. At runtime the proposal is bound to that manifest commitment. The Control Plane then obtains Authority Context from a trusted resolver, independently of the incoming request, and checks the actual proposal. Matching the declaration alone never authorizes the refund.



### AT A GLANCE

|              |                                             |
|--------------|---------------------------------------------|
| TYPE         | Format, Python validator & CLI              |
| INTERFACE    | Manifest 1.1                                |
| ACCEPTED PIN | 46c950b                                     |
| SOURCE       | github.com/cogno-us/cognous-action-manifest |
| FORMERLY     | Agent Action Manifest                       |

TRY IT

Write a manifest for one bounded action you already run, validate it with the CLI and review it with that action's owner.

COMPONENT 04

# Cognous Control Plane

An earlier yes is not a permanent credential.

A Python reference implementation for bounded agent authorization, effect-time revalidation and persistent runtime records. It connects declared actions, independently supplied authority, execution attempts and reconciliation.

THE PROBLEM

**Approvals go stale**  
A cached approval or a fluent explanation cannot show whether authority, evidence and approvals were still valid when the effect was attempted.

→

WITH CONTROL PLANE

**Critical inputs rechecked just before each effect**  
The controller decides, revalidates authorization-critical inputs immediately before invoking the effect, and keeps decisions, attempts, observations and reconciliation as separate records.

WHAT IT PROVIDES

- **Bounded authorization:** Manifest-bound proposals evaluated against explicit policy and an Authority Context supplied by a trusted resolver, outside the incoming request.
- **Effect-time checks:** authorization-critical inputs revalidated immediately before each effect is attempted.
- **Lifecycle records:** decisions, attempts, observations and reconciliation kept distinct, including unknown acknowledgements.
- **Persistent records:** reload-and-append under a Linux sidecar lock, with atomic replacement and explicit persistence failures.
- **Developer tools:** validation, policy examples, replay export, redaction and integrity helpers, each with documented assurance limits.

IN THE REFUND WORKFLOW

A synthetic refund is proposed under a bounded grant. The controller records its decision, rechecks current authority immediately before the effect and invokes the constrained adapter. After a timeout, reconciliation concerns the original effect: a fresh observed absence does not authorize a replacement, and accepted applied evidence can resolve delivery while preserving the interrupted history.



WHAT “EFFECT TIME” MEANS HERE

The recheck is per effect, on one host. Individual record transactions are protected, but decide, execute and reconcile are not one atomic transaction, and there is no distributed check-to-commit guarantee: authority can change between the recheck and the destination commit. Reconciliation records what happened; it does not undo it.

WHAT IT DOES NOT CLAIM

Supports cooperating writers on documented local Linux filesystems with shared canonical-path assumptions. Production resolver authentication, distributed budgets, remote finality and exactly-once delivery are not established.

AT A GLANCE

|              |                                           |
|--------------|-------------------------------------------|
| TYPE         | Python reference implementation           |
| INTERFACE    | Bounded authorization/effect 0.1          |
| ACCEPTED PIN | d3dadee                                   |
| SOURCE       | github.com/cogno-us/cognous-control-plane |
| FORMERLY     | Agent Control Plane                       |

TRY IT

Trace one refund’s decision, revalidation and attempt records, then run the timeout case to see how reconciliation treats the original effect.

COMPONENT05

# Cognous Execution Runtime

On the supported path, execute only the exact authorized operation.

The stack’s single-agent constrained execution boundary. Its Python execution layer (engine/) supplies the bounded synthetic destination and adapter boundary. It is a hardened derivative of Moltbot; the upstream TypeScript Moltbot application in the same repository is outside the reviewed boundary.

THE PROBLEM

Authorization and execution drift apart

Between an allow decision and the effect, a payload can change, a count can grow, or a caller can claim an institution it does not represent.

→

WITH EXECUTION RUNTIME

A frozen operation bound to one effect

The complete operation is snapshotted at entry; validation and execution use only that snapshot, and effect identity is bound to its digest.

WHAT IT PROVIDES

- **Execution Envelope 0.2.0:** a deep snapshot of the operation taken before any resolver or Control Plane callback runs.
- **Strict refund rules:** exactly one effect, a finite non-negative amount and a payload commitment that must match.
- **Trusted institution binding:** institution and domain taken from the trusted resolver, never from a caller’s label.
- **Durable attempts:** immutable attempt identities with append-only events; a reused attempt ID is rejected and never overwrites a success.
- **Producer profile 2.0.0:** versioned export of envelopes, results, effects, attempts and observations for reconstruction.

IN THE REFUND WORKFLOW

The Control Plane’s revalidating workflow calls the refund adapter, which applies strict local execution policy, writes a durable synthetic SQLite refund and supports destination observation and reconciliation. SQLite serialization handles same-effect deduplication across separate processes sharing one database file.



KNOWN LIMIT, SHOWN EXPLICITLY: EFFECT ID IS NOT BUSINESS INTENT

Deliver the same \$250 refund proposal twice and both deliveries carry the same effect\_id: one refund. Submit a second, different valid proposal for the same customer refund and it gets a new effect\_id: both can execute, producing two refunds. The selected runtime now includes refund-intent ownership (executor PR #14): run as a separate optional profile, it keeps one effect for two proposals of the same intent. The default release path does not enable it.

WHAT IT DOES NOT CLAIM

Application-level restrictions, not OS confinement: a sufficiently privileged host process can bypass the package. A SQLite synthetic destination only, with no real accounts, payment services or production credentials. Live OpenShell confinement is unqualified.

AT A GLANCE

|              |                                                  |
|--------------|--------------------------------------------------|
| TYPE         | Python execution layer (engine/)                 |
| INTERFACES   | Execution Envelope 0.2.0; producer profile 2.0.0 |
| ACCEPTED PIN | c3c3ee7                                          |
| SOURCE       | github.com/cogno-us/cognous-execution-runtime    |
| FORMERLY     | Moltbot Safe                                     |

TRY IT

Run the substitution cases (changed payload, target or adapter) and confirm each produces no effect. Then submit two different proposals for the same refund: both execute on the default path, and only one under the optional refund-intent profile.

COMPONENT06

# Cognous Replay Bundle

Rebuild what happened from the records, without running it again.

A portable reconstruction format, importer, validator and CLI for agent-run evidence. Reconstruction Bundle 0.2.0 imports exact supported producer revisions while keeping legacy formats and historical provenance distinct.

THE PROBLEM

Logs disagree, and stories fill the gaps

Records from authorization, execution and recovery can conflict or omit facts. A retrospective success story is easy to write and hard to check.

→

WITH REPLAY BUNDLE

One connected, honest reconstruction

Proposals, decisions, attempts, observations and source lineage in one bundle, with uncertainty and missing information left visible.

WHAT IT PROVIDES

- **Producer imports:** exact supported Control Plane and executor combinations; unknown revisions are rejected, not guessed.
- **Identity continuity:** proposal, decision and effect identity preserved, with attempt namespaces kept separate.
- **Recovery history:** unknown acknowledgements, rejected or null observations and evidence-only recovery lineage retained.
- **Loss and completeness:** declared coverage and missing information, without claiming the effect completed.
- **Portable review:** validate, summarize, redact and sign exports; HMAC export integrity is separate from production key management.

WHAT IT DOES NOT CLAIM

Reconstruction is non-effecting. It does not re-evaluate policy, renew authorization, repeat an effect or independently verify delivery. Completeness is not destination finality, and a valid digest or HMAC is not institutional authority.

IN THE REFUND WORKFLOW

After a bounded refund attempt, the importer consumes the actual Control Plane and executor records. It keeps whether the destination was observed applied, absent, partial or unresolved, and which source supplied each fact. Evidence Pack and ODES exporters then derive review material without rerunning the action or rewriting the originals.



AT A GLANCE

|              |                                           |
|--------------|-------------------------------------------|
| TYPE         | Format, importer, validator & CLI         |
| INTERFACE    | Reconstruction Bundle 0.2.0               |
| ACCEPTED PIN | 459e4ba                                   |
| SOURCE       | github.com/cogno-us/cognous-replay-bundle |
| FORMERLY     | Agent Replay Bundle                       |

TRY IT

Import the records from one reference run and check how an unresolved outcome is represented in the bundle.

COMPONENT07

# Cognous Governance Evidence Pack

A summary your reviewers can trace back to the source.

A reference format, validator, traceable importer and Markdown renderer for business-facing review of agent governance records. It keeps the link between a declared action surface and reconstructed runtime evidence.

THE PROBLEM

**Raw events overwhelm; summaries overclaim**

Reviewers need authority, attempted actions and unresolved outcomes without reading every event, but a summary must not turn declarations into verified facts.

→

WITH GOVERNANCE EVIDENCE PACK

**Review material with its derivation attached**

Every summarized decision or outcome links to its source, and declared controls, runtime assertions, tests and verification stay apart.

WHAT IT PROVIDES

- **Evidence structure:** the agent, its actions, governance context and source records in one review package.
- **Traceable imports:** Manifest and Reconstruction Bundle inputs transformed with explicit source references and versions.
- **Assurance separation:** declared controls, source-asserted facts, executed tests and independent verification kept distinct.
- **Continuity checks:** source identities, commitments and producer compatibility validated across the selected generation.
- **Business rendering:** validated Markdown summaries that keep unresolved and missing information visible.

WHAT IT DOES NOT CLAIM

Import success does not show that a test ran, that an institution authorized current reliance or that a control works in production. Not certification, legal advice, independent audit or a runtime enforcement layer.

IN THE REFUND WORKFLOW

The bounded refund produces retained reconstruction evidence. The importer combines it with the Manifest, records the derivation and renders business-facing review material. A reviewer can follow each item back to its source and tell historical authorization, a current recovery denial and the observed destination state apart.



AT A GLANCE

|              |                                                      |
|--------------|------------------------------------------------------|
| TYPE         | Format, validator, importer & renderer               |
| INTERFACES   | Evidence Pack schema 0.2.0; transformation 0.3.2     |
| ACCEPTED PIN | b4baccd                                              |
| SOURCE       | github.com/cogno-us/cognous-governance-evidence-pack |
| FORMERLY     | Agent Governance Evidence Pack                       |

TRY IT

Render a pack from a reference run and follow three summary lines back to their source records.

COMPONENT08

# Open Decision Evidence Standard

Let decisions cross boundaries with their evidence intact.

ODES is an open, vendor-neutral discussion draft and reference tooling for portable decision evidence. The v0.2 narrative and implementation profile keep the pder-v0.1 record schema. It gives recipients what they need to evaluate a decision; it does not decide whether they should rely on it.

THE PROBLEM

**Decisions travel; their basis does not**  
A decision can reach another system or organization without its authority basis, human disposition, evidence commitments, freshness or conditions of use.

→

WITH ODES

**Portable evidence, judged by the recipient**  
Structured decision evidence that recipients validate under their own policy, without mandatory disclosure of confidential raw evidence.

## WHAT IT PROVIDES

- **Portable records:** authority, machine role, human disposition, model and policy context, and evidence references.
- **Integrity packaging:** exported content bound to explicit commitments, keeping integrity distinct from authenticity.
- **Recipient evaluation:** package validity and declared consumption conditions checked under an explicit recipient policy.
- **Stack mapping:** export from supported Manifest and Replay inputs through a documented field mapping.
- **Open draft process:** schema, profiles, conformance material and lifecycle and reliance RFCs open for public review.

## WHAT IT DOES NOT CLAIM

Not a shared policy plane, blockchain requirement or certification, and not a finding that a decision is correct, lawful or suitable for reliance. It grants no execution authority. Reference compatibility is narrower than universal interoperability or standards adoption.

## IN THE REFUND WORKFLOW

A producer exports a decision-evidence package from retained reconstruction. The recipient validates the record and its commitments, then evaluates the package under its own policy. Missing authentication or current-authority evidence stays visibly missing: package validity cannot fill those gaps, and selective disclosure is not permission to infer omitted facts.



## AT A GLANCE

|              |                                                     |
|--------------|-----------------------------------------------------|
| TYPE         | Discussion draft + reference tooling                |
| INTERFACES   | Implementation profile 0.2.1; pder-v0.1             |
| ACCEPTED PIN | c5e9a0f                                             |
| SOURCE       | github.com/cogno-us/open-decision-evidence-standard |

TRY IT

Export one package from a reference run and evaluate it under a deliberately strict recipient policy.

COMPONENT09

# Portable Reasoning Protocol v1.0

Ask the model to show where evidence ends and inference begins.

PRP is a reusable SKILL.md-based instruction package for general-purpose AI work. It asks a model to calibrate rigor to consequence, distinguish evidence from inference, state material uncertainty and preserve the user’s agency.

THE PROBLEM

**Fluency hides uncertainty**  
Confident output can conceal unsupported certainty, invented sources, unstated assumptions or a premature conclusion.

→

WITH PORTABLE REASONING PROTOCOL

**Rigor scaled to consequence**  
A shared, portable instruction set that raises scrutiny as stakes rise, without turning routine tasks into long reports.

## WHAT IT PROVIDES

- **Adaptive rigor:** more scrutiny as consequence, uncertainty, novelty or irreversibility increase.
- **Claim discipline:** observed, inferred, estimated, hypothetical, normative and unknown statements kept apart.
- **Hallucination controls:** instructs the model not to fabricate facts, citations, capabilities, tool results or completed actions.
- **Agency preservation:** material tradeoffs presented, without manipulative pressure or unnecessary requests.
- **Optional handoff:** proposal and evidence distinctions preserved for downstream systems, without creating authority.

## WHAT IT DOES NOT CLAIM

Behavioral instructions, not runtime enforcement. PRP adds no persistent state, authenticates no institution, intercepts no tools, creates no sandbox and guarantees no truthfulness. The hub runs static checks only; model-behavior efficacy is unmeasured.

## IN THE REFUND WORKFLOW

A user brings a decision question and asks PRP to separate facts, assumptions and remaining uncertainty. The assistant proposes options at a depth suited to the task. If a consequential action follows, the deployment must still check authority and constrain execution separately. A confident conclusion is not an authorization.



## AT A GLANCE

|              |                                                 |
|--------------|-------------------------------------------------|
| TYPE         | SKILL.md instruction package                    |
| VERSION      | v1.0                                            |
| ACCEPTED PIN | cb137f0                                         |
| SOURCE       | github.com/cogno-us/portable-reasoning-protocol |

TRY IT

Install the skill in one assistant and compare its answers to the same consequential question with and without it.

COMPONENT 10

# Research Intelligence Protocol v1.0

Test competing explanations before abstracting from them.

A modular SKILL.md-based research workflow with two distinct components. Discovery structures observations, hypotheses and discriminating experiments; Abstractor of Abstractors (AoA) compares structure and proposes transferable invariants with explicit limits.

THE PROBLEM

**Favorite explanations win too early**  
Research fails by confirming the preferred story, or by generalizing from an analogy before the evidence justifies it.

→

WITH RESEARCH INTELLIGENCE PROTOCOL

**Discovery and abstraction, kept separate**  
Competing hypotheses, negative findings and missing results stay on the record while candidate abstractions carry their limits.

## WHAT IT PROVIDES

- **Discovery:** observation separated from interpretation; experiments chosen to distinguish competing explanations.
- **Uncertainty ledger:** assumptions, missing information, controls and stopping conditions retained.
- **AoA:** entities, relations, functions, dynamics, boundaries, governance and context compared.
- **Transfer discipline:** surface resemblance distinguished from structural equivalence, with the point of failure stated.
- **Optional proposal profile:** traceable, unaccepted proposals validated for downstream review, never accepted as Evidence Registry objects.

## IN THE REFUND WORKFLOW

Use Discovery alone to choose the next discriminating observation, or AoA alone when the evidence base already supports structural comparison. Combined, Discovery produces structured evidence and AoA proposes abstractions, each carrying its uncertainty and provenance. The optional handoff stays unaccepted until a receiving system makes its own decision.



## WHAT IT DOES NOT CLAIM

Runs no physical experiments, verifies no facts, supplies no missing findings and establishes no authority. A valid proposal is not an accepted Evidence Registry object. Behavioral benefits are unmeasured; static and schema checks do not demonstrate discovery or model efficacy.

## AT A GLANCE

|              |                                                    |
|--------------|----------------------------------------------------|
| TYPE         | Modular SKILL.md workflow                          |
| VERSION      | v1.0                                               |
| ACCEPTED PIN | 30c7274                                            |
| SOURCE       | github.com/cogno-us/research-intelligence-protocol |

TRY IT

Run Discovery on an open question your team disagrees about and review the uncertainty ledger together.

COMPONENT

11

# TFA Protocol (S43)

Say what is true. Ask for nothing. Protect their next move.

A lightweight, optional behavioral protocol for AI interaction, built on three rules: Truth, Freedom and Agency. Implementation and evaluation guidance explain those rules without turning them into an authority system.

THE PROBLEM

Help that costs the user

An assistant can burden users through unsupported certainty, unnecessary questions, validation pressure or closing off alternatives too soon.

→

WITH TFA PROTOCOL

A small vocabulary for interaction quality

Three rules reviewers can apply to any response, while still allowing the questions that are genuinely needed.

WHAT IT PROVIDES

- **Truth:** claims calibrated to evidence; no invented facts, sources, capabilities or authority.
- **Freedom:** no extracting unnecessary information, reassurance, labor or commitment as the price of help.
- **Agency:** meaningful alternatives kept open; action only within permission already granted.
- **Proportionate questions:** the minimum needed when accuracy, safety or authorization depends on missing information.
- **Evaluation guidance:** explicit cases and review criteria, rather than treating the three rules as a behavioral guarantee.

WHAT IT DOES NOT CLAIM

TFA authenticates no identities, issues or revokes no grants, enforces no tool permissions and establishes no external effect. The hub has not run model-behavior qualification. Historical PDFs keep their original provenance and are not new cryptographic guarantees.

IN THE REFUND WORKFLOW

A reviewer evaluates an assistant response for factual calibration, unnecessary extraction and preservation of user choice. A clarifying question is appropriate when it resolves decision-critical uncertainty or an authorization boundary. Once permission and evidence are sufficient, the assistant can proceed without repeatedly seeking reassurance. TFA never overrides institutional or runtime controls.



AT A GLANCE

|              |                                                   |
|--------------|---------------------------------------------------|
| TYPE         | Optional behavioral protocol                      |
| VERSION      | S43                                               |
| ACCEPTED PIN | 442d07b                                           |
| SOURCE       | github.com/cogno-us/truth-freedom-agency-protocol |

TRY IT

Score ten recent assistant replies against the three rules and discuss where your reviewers disagree.

COMPONENT12

# Cognous Governed Exchange

Receiving a message is not the same as being told to act.

An experimental GAX/IMX reference that connects governed messages, recipient assessment, bounded execution and retained evidence. It implements exchange behavior and local transport; the institution and its Alvorada constitution live separately, in Institutional Governance.

THE PROBLEM

**Delivery blurs into permission**

Receipt, understanding, acceptance, authorization and execution are different events. Duplicate messages, lost acknowledgements and changed authority make them easy to confuse.

→

WITH GOVERNED EXCHANGE

**Each state assessed and retained separately**

GAX assesses the message, durable transport handles delivery and redelivery, and IMX keeps continuity, so outcomes stay tied to the original artifacts.

## WHAT IT PROVIDES

- **GAX assessment:** message structure, supported profile, identity binding, references, freshness and permitted handling.
- **Local durable transport:** queue, deliver and redeliver messages while keeping their link to recipient outcomes.
- **Runtime integration:** calls the Control Plane and executor with a caller-supplied resolver, policies and trusted time.
- **Retained artifacts:** original Replay, ODES and IMX artifacts exposed with identities and content commitments.
- **Recovery and continuity:** original effects reconciled and derivative lineage retained; loading a successor is never an execution grant.

## WHAT IT DOES NOT CLAIM

A bounded refund profile, not full GAX conformance, a production network or institutional authentication. Message IDs and receipt do not authorize actions. Deferred Governed Exchange PR #2 is excluded from the hub.

## IN THE REFUND WORKFLOW

LocalDurableTransport delivers a synthetic refund message to the GAX recipient adapter, which assesses it and routes the Manifest-bound proposal through the trusted runtime. Redelivery of the same message returns the original retained artifacts, with no second effect. If evidence was never retained after an effect, evidence-only recovery creates an explicitly derived record, never a replacement effect.



## AT A GLANCE

|              |                                                   |
|--------------|---------------------------------------------------|
| TYPE         | Experimental GAX/IMX workbench                    |
| INTERFACES   | GAX retained artifacts 1.1.0; merged-producers-v1 |
| ACCEPTED PIN | a1cbc7b                                           |
| SOURCE       | github.com/cogno-us/cognous-governed-exchange     |
| FORMERLY     | Alvorada Experimental Workbench                   |

TRY IT

Run the transported refund in the reference and compare the first delivery with the artifacts a redelivery returns.

COMPONENT13

# One refund, reconstructed step by step.

The request moves through transport, assessment, binding, authority resolution, revalidation and constrained execution. The hub then checks the evidence it left behind. The diagram on page 2 shows the same path.

## FIVE FACTS THE STACK KEEPS SEPARATE

|                                                                                   |                                                                                       |                                                                                         |                                                                                     |                                                                                           |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|
| <b>1 PROPOSED</b><br>The Manifest-bound operation, target, amount and commitment. | <b>2 PERMITTED</b><br>The Control Plane decision, revalidated just before the effect. | <b>3 ATTEMPTED</b><br>Each durable attempt and its acknowledgement, or the lack of one. | <b>4 OBSERVED</b><br>What the destination was seen to do: applied, absent, partial. | <b>5 VERIFIED</b><br><i>Independent confirmation, which the reference does not claim.</i> |
|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------|

## THE SUPPORTED WORKFLOW

|                                                                                                                                                                                      |                                                                                                                                                                                   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1 DELIVER</b><br>LocalDurableTransport delivers the refund message to the GAX recipient adapter in Governed Exchange.                                                             | <b>5 DECIDE &amp; REVALIDATE</b><br>The Control Plane records its decision, then rechecks critical inputs immediately before this effect. Not atomic with the destination commit. |
| <b>2 ASSESS</b><br>GAX assesses structure, profile, identity binding, freshness and permitted handling. Receipt alone authorizes nothing.                                            | <b>6 EXECUTE</b><br>The Execution Runtime executes only the frozen, exact operation against the synthetic SQLite destination.                                                     |
| <b>3 BIND</b><br>The adapter produces a proposal bound to the declared action in the Action Manifest.                                                                                | <b>7 RETAIN</b><br>Decisions, attempts, observations and IMX continuity records are retained with identities and commitments.                                                     |
| <b>4 RESOLVE AUTHORITY</b><br>A separately configured trusted resolver supplies the Authority Context (institution, mandate, grant, approval, policy), outside the incoming request. | <b>8 RECONSTRUCT &amp; REVIEW</b><br>Replay reconstructs the run; the Evidence Pack and ODES carry it to reviewers and recipients.                                                |

## IDENTITIES THAT TRAVEL

|                            |                                                                                                   |
|----------------------------|---------------------------------------------------------------------------------------------------|
| <b>message_id</b>          | a delivery object                                                                                 |
| <b>proposal_commitment</b> | the exact operation                                                                               |
| <b>decision_id</b>         | a Control Plane decision                                                                          |
| <b>effect_id</b>           | the destination effect one proposal names, across delivery and recovery. Not the business intent. |
| <b>attempt_id</b>          | an attempt, in separate Control Plane and executor namespaces                                     |
| <b>bundle_id</b>           | a reconstruction; Evidence Pack and ODES IDs never mint an effect                                 |

## WHEN THINGS GO WRONG

- **Unknown acknowledgement:** does not imply absence; an in-flight effect can still commit.
- **Observed absence:** point-in-time evidence, not permission to retry.
- **Reconciliation:** concerns the original effect and keeps the acknowledgement history.
- **A denied recovery:** never erases a historical applied effect. Both facts stand, recorded separately.
- **Evidence-only recovery:** creates a clearly derived record, never a replacement effect.

### ON THE SIDE, NOT IN THE PATH

Evidence Attestation and the Evidence Registry form a separate evidence path for signed claims and commitments. PRP, Research Intelligence and TFA shape how models reason and respond. None is an enforcement dependency, and none can authorize an effect.

**Optional**  
by design

# Run it, inspect it, then decide.

*Runs locally with no production credentials, Docker or OpenShell. Run one bounded refund and read what it leaves behind.*

## FOR ENGINEERS: RUN THE REFERENCE

Use a disposable Linux checkout. These commands reproduce the October 13 verified run of the selected lock.

```
# Clone the hub and pin the Oct 13 verified source
git clone \
  https://github.com/cogno-us/cogno-us-open-control-stack.git
cd cogno-us-open-control-stack
git checkout 9a3772783e4f08d206268bd9650b16395adb546a

# Create and activate a Python 3.11 environment
python3.11 -m venv .venv
source .venv/bin/activate

# Run two isolated repetitions of the reference
python tools/reference_release.py run \
  --results-dir results/reference
```

The runner refuses a nonempty results directory and replaces `.reference-work/`, so give each run a new, empty results path.

## READ THE RESULT

Start with `results/reference/scenario-results.json`: check `release_qualified`, the pins, test totals and each required scenario. Then open:

- **scenario-matrix-results.json** — the resolved matrix
- **representative-repeatability.json** — run-to-run comparison
- **skip-accounting.json** — nothing silently skipped
- **artifact-index.json** — the retained evidence

*A missing, skipped, failed or unexecuted required scenario blocks the release gate. A passing characterization of a known limitation, such as case 7, does not mean the limitation was prevented.*

## WHAT YOU NEED

| PLATFORM                                     | RUNTIMES                                        | NETWORK                                             | NOT NEEDED                                                   |
|----------------------------------------------|-------------------------------------------------|-----------------------------------------------------|--------------------------------------------------------------|
| Linux with Git. Reference CI runs on Ubuntu. | Python 3.11 with venv and pip; Node 20 and npm. | GitHub clones and Python and npm package downloads. | Production credentials, Docker or an OpenShell installation. |



*It does not deploy services, write to a public chain or provision infrastructure. A command that returns before setup completes is not a passing qualification.*

## PLANNING YOUR OWN DEPLOYMENT

1. **Pick one workflow** with a real destination and a bounded effect.
2. **Map your authority**: who supplies it, and how the resolver is authenticated.
3. **List the differences** between your authority, destination, credentials and enforcement boundary and the reference, including business-intent duplicates.
4. **Qualify each difference** with its own acceptance evidence before extending any claim.

## WHERE TO GO NEXT

Developer quickstart <docs/quickstart.md>

Governance reviewer quickstart <docs/governance-quickstart.md>

Architecture and responsibility map <docs/architecture.md>

Support status <docs/release-status.md>

Optional execution profiles <docs/optional-execution-profiles.md>

*Existing licenses and notices apply; the selected stack is not wholly Apache-2.0, and the selected PRP and TFA revisions remain a rights-review exception. ISS, Navalia and private research are outside this public integration.*

## LET'S TALK

Bring one workflow your agents should run. Together we will map where its authority, destination, credentials and enforcement differ from the reference, and what evidence each difference needs before you rely on it.

**cogno.us**  
COGNOUS