How Gradia works: technical architecture

From approved source evidence to executable, reviewable tests

Data contracts, authority, state transitions, replay and native evaluation admission.

Two execution paths share governance infrastructure

Observer workflow world

Reviewed case
Owner-approved WorldPlan
Compiled EnterpriseTwinSpec
WorkflowWorld command receipts

Native evaluation factory

Interview or structured Spec
IR and Build
Environment and TaskSpec
Run, Episode and Grade evidence

A downstream evidence contract must explicitly admit the inputs it consumes.

Capture has separate permission and membership layers

Object	Purpose	Who supplies authority
Capture scope	Allowed accounts, resources, content and retention	Named enterprise and participant approval
Session and observation	Selected activity and explicit coverage gaps	Collector/source identity within scope
Business case	Observations assigned to one workflow instance	Reviewable membership decisions
Verified case edition	Exact included history and review state	Authenticated human review

The captured event, its case membership and its human review remain distinct.

Each receiver has a declared acquisition boundary

Receiver	Collected surface	Important boundary
Browser	Categorical click/navigation metadata	Exact approved HTTPS pages, explicit start
macOS 15.2+	Selected approved native window	Whole-window screenshot consent and review
Service adapters	Selected provider records and optional content	Pinned account/resource and provider limits
Microsoft renewal	Existing delegated mail/Teams access	No new consent or scope expansion

Screenshots and sanitized text require their own permission and review.

Encryption protects specific storage boundaries

Storage	Protection	Practical limit
Browser local store / connector SQLite	Local metadata remains readable	Use a controlled participant device
Desktop vault / optional content	Encrypted local persistence	Masking and redaction still need review
Managed Observer state	AES-256-GCM with org/project AAD	Bounded serialized workspace state
Graph refresh-token store	Encrypted, scope-bound private POSIX store	Access token in memory, durable halt on terminal failure

AAD means authenticated additional data. It binds ciphertext to its declared context.

An outbox preserves exact delivery through uncertainty

01

Persist export

The collector holds an immutable pending export before delivery.

02

Send exact bytes

An uncertain response retains the same pending payload.

03

Resume acquisition

The receiver must acknowledge the outbox before new collection.

Foreground --interval collection applies bounded retries and retention sweeps.

Human review binds an exact edition

What the reviewer sees

Included observations
Content review state
Coverage gaps and decisions
The exact current case edition

Case review or rule approval

New or removed evidence
Changed membership or content
A changed rule plan
Stale source or rights dependencies

The visual editor invalidates preflight when rule edits remain unsaved or change the plan.

Automatic relevance requires measured admission

Step	Contract
Policy	Freeze model/provider/prompt policy and allowed use
Separate labels	Use human-labeled examples with blind heldout evidence
Admission	Measure class precision and require the configured gates
Application	Apply admitted membership edits with exact evidence binding
Verification	Keep human verification separate from model suggestions

A detour can be excluded from a case without asserting that the source was deleted.

The automatic proposal preserves observation lineage

10,000 included observations
At most 100 chronological action groups

Every group retains up to 100 exact event references.
Source, session and UTC-day boundaries guide grouping.
Balanced chronological batches handle larger boundary counts.

Grouping preserves order and attribution. The owner supplies business semantics.

WorldPlan makes the proposed rules explicit

Contract	Representative fields
World identity and time	case_id, start_at, horizon
Actors and source rights	principal_ids, source_visibility, rights_basis, allowed_uses
Modeled state	initial_state, terminal_states, initial_facts
Action definition	from_state, to_state, actor_ids, required_event_ids
Action conditions and effects	not_before, deadline, requires, effects

Pydantic contracts reject undeclared fields and invalid relationships.

A transition checks authority before changing state

```
allowed = state_matches and actor_allowed
allowed &= time_in_range and required_events_have_occurred
allowed &= all(type(fact) is type(expected) and fact == expected)

if act and allowed:
    state = action.to_state
    facts.update(action.effects)
```

An unavailable action refuses without committing a workflow frame.

State changes remain inside the declared-rule engine.

Time and record visibility constrain model context

Clock and evidence

An advance command requires clock < at <= horizon.

Only records at or before the clock enter the temporal projection.

Actor view

Record viewer IDs constrain which source records are readable.

Modeled facts are shared within the plan, with no per-fact ACL.

The model receives current available actions and the actor’s permitted records.

Receipts bind command history to resulting state

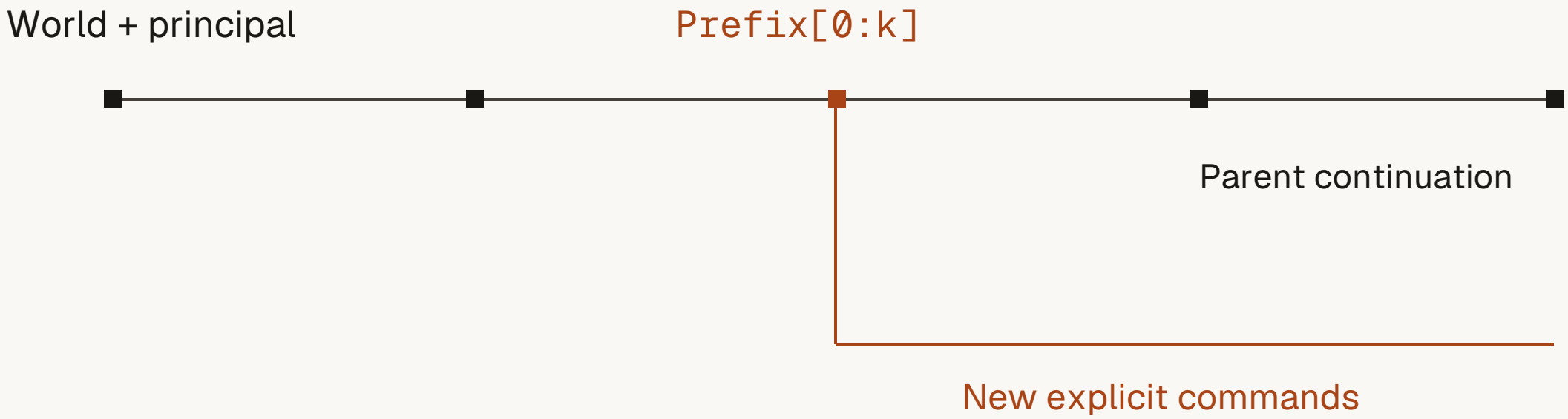
```
root = SHA256(canonical_json({
  world, principal_id, clock, state, facts, commands
}))

frame = { sequence, command, before, after, previous }
frame.sha256 = SHA256(canonical_json(frame))
```

The first frame links to the compiled world digest.
Each later frame links to the previous frame digest.

Receipt includes terminal_root, completed, business_seconds and the explicit claim boundary.

A branch reconstructs the saved command prefix



`fork(k)` creates a fresh engine and replays `commands[:k]`.

This is deterministic replay. It does not estimate counterfactual real-world outcomes.

A governed attempt has a durable network boundary

Before dispatch	After provider response
Revalidate case, world, policy and independent approval	Revalidate current scope, case review and world evidence
Reserve the next bounded model call durably	Record call outcome separately from accepted transition
Send only current actor-visible context	Validate the proposed command through WorkflowWorld
Enforce approved model/data/spend limits	Commit the resulting frame or preserve refusal state

A model call receipt, a valid command and a completed workflow are separate facts.

Guard makes the assurance boundary explicit

Surface	What the evidence can establish
Portable process or SDK recorder	Integrity of recorded covered events, with named gaps
Governed Gateway	Decisions and calls that cross the configured gateway
Runtime and egress composition	Only the exact measured isolation/enforcement boundary
Universe execution	Recorded state transitions under the admitted environment rules

A stronger assurance claim needs the corresponding independently checkable runtime proof.

The native factory freezes a different execution contract

01

Specification

Interview or structured input becomes a validated, frozen Spec and IR.

02

Build

An exact build binds the environment and declared assets.

03

Native execution

Tasks run under the admitted runner, evaluator and conditions.

Observer receipts need explicit translation and native admission before this protocol can use them.

Evaluation preserves the denominator and its exclusions

Binding	Why it matters
Task and environment	The test case and executable environment have exact identities
Model and execution policy	The system configuration and conditions stay comparable
Evaluator and sampling panel	The measure and task/seed cells freeze before eligible runs
Run, Episode and Grade	Comparison derives from persisted execution and grading
Exclusions and missing scores	Infrastructure censors remain distinct from model failures

A single workflow demonstration does not establish evaluator calibration or model quality.

Four immutable families carry the policy lifecycle

Family	Admission derives from
assessment-contracts	Pre-frozen tasks, environments, conditions and methods
release-comparisons	Eligible paired native runs and recomputed result counts
technical-evidence-packages	Current dependency graph, assets, decisions and signatures
remediation-corpora	Native failures, item review, scoped rights and disjoint splits

Primary writes require exact independent human decisions and current dependency closure.

Comparison and package signatures bind the final body

Before issuance

Rehash referenced assets.
Resolve current rights and approvals.
Bind actual human identities.
Sign the final immutable body.

At verification

Recompute source/result bindings.
Recheck dependency closure.
Verify configured workspace keys.
Refuse stale or invalid evidence.

A valid digest or signature alone does not establish business truth, rights or calibration.

Remediation admission separates repair from training

Required evidence	Admission boundary
Recorded failure	Materialize an actual eligible native failure
Item review	An independent reviewer decides on the exact proposed item
Source rights	Recipient scope and separate training use where required
Split integrity	Keep declared tasks, splits and holdouts disjoint
Privacy and answer checks	Bounded literal checks supplement, but never replace, review

Creating a corpus does not dispatch training or prove that a proposed repair works.

Runtime services preserve explicit project boundaries

Component	Responsibility
Next.js Console	Authenticated user journeys and API proxy
FastAPI API	Scope, current approval, contract and project-context checks
Worker	Authorized queued execution with current dependency validation
PostgreSQL	Forced organization/project RLS and append-only evidence tables
Assets and configured keys	Content identity, encryption and exact signature verification

This architecture describes implemented boundaries, not a general compliance certification.

Capacity limits refuse incomplete admission

Path	Current bounded admission
Managed Observer workspace	32 MB plaintext serialized state per project
Automatic world proposal	10,000 observations, 100 groups, 100 references per group
Workflow plan and execution	200 actions per plan, 200 commands per run
Agent attempt	1–8 model calls per attempt
Nested policy evidence	8 MB per asset, 32 MB unique reads, 512 references, 10,000 cells

Named limits and refusal codes make qualification testable.

An enterprise integration starts with declared invariants

Who owns the workflow and each source?
Which evidence can each actor see, and when?
What exactly changes state, and who approves the rule?
Which execution and evaluator contract will judge the attempt?

The implementation guide maps these questions to source files, API endpoints, schema names and a worked synthetic case.