

PRODUCT HANDBOOK · P02

The Loom product handbook

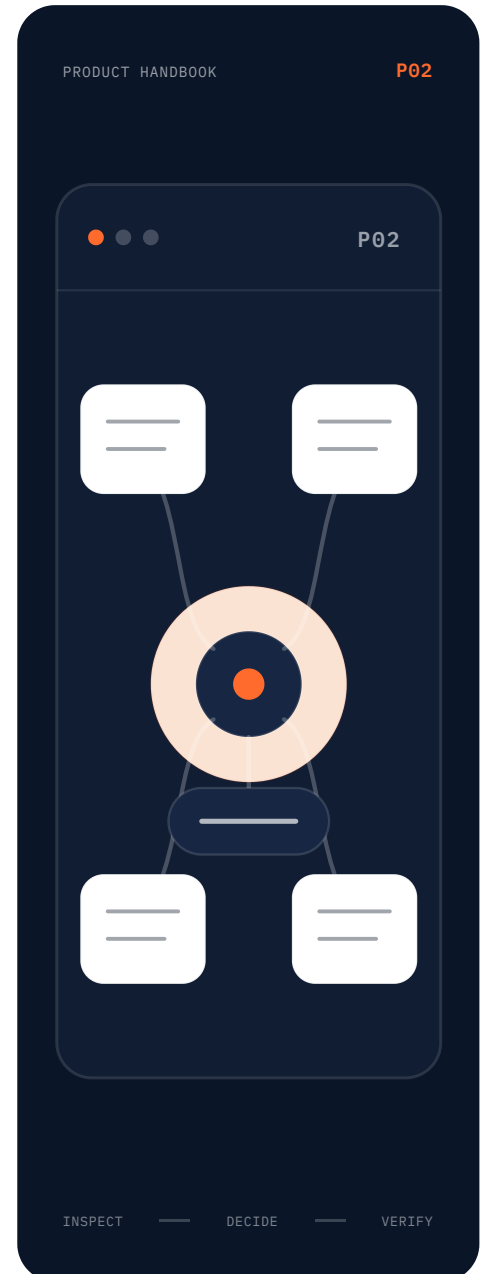
Cognition as a traceable path through perception,
memory and search, sparse expert routing,
reasoning, exact solver use, and expression.

FOUR WEAVES

SPARSE ROUTING

CONSTRAINT TRACE

AION CERTIFICATE



Loom does not hide cognition behind one undifferentiated model. It records the selected experts,
source and memory references, constraints, solver path, expression, and proof state.

P02

CONTENTS

Loom, from typed question to checked artefact

This handbook follows one eligibility explanation through the four weaves, expert selection, constraint conflict, exact computation, expression, trace, certificate, and operational proof.

PRIMARY READERS

Product, architecture, delivery and operations teams

READING DEPTH

Professional + technical

DECISION THIS SUPPORTS

A product-specific handbook for teams evaluating Loom as a source-aware cognitive runtime with inspectable experts, constraints, solver artefacts, and proof material.

01

The cognitive artefact

03–04

What Loom owns and what it must not claim

02

Route and reason

05–09

Experts, corpus, constraints, memory, and solver boundaries

03

Trace and verify

10–14

Expression, AION, replay, failure, and deployment

04

Prove the product

15

Acceptance corpus, production record, sources, and next action

05

Subject index

16–17

An alphabetical finder for concepts, controls, products, and decisions.

HOW TO USE IT

Replace the illustrative eligibility question with one workload you own. Verify current model, expert, catalogue, solver, benchmark, interface, deployment, and commercial terms against the maintained routes and written agreement.

SOURCE DISCIPLINE

Dweve-specific facts are taken directly from the English website text audited 2026-07-20; web-navigation wording is humanised for print. Evaluation guidance is editorial, and scenarios and derived visuals are illustrative. Unsupported synthesis is replaced by canonical copy or omitted.

THE COGNITIVE ARTEFACT

Loom owns the cognitive artefact, not the decision

Loom can perceive, retrieve, route, reason, solve, and express. A human or domain system still owns the authority and consequence of acting on the result.

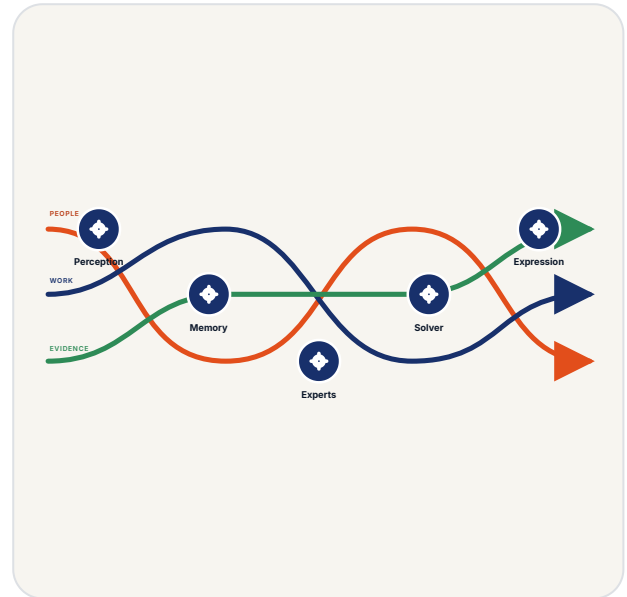
An eligibility workflow asks: “Why was this applicant placed in review?” Loom receives a typed question, authorised case projection, policy references, constraints, and requested mode. It may route a small subset of specialists, inspect source-linked knowledge, invoke an exact solver, and return a reason or typed refusal with the cognitive and proof material attached.

Cognition has a boundary

The product specification gives Loom ownership of perception, memory, search, reasoning, exact solver use, and expression. It does not own team staffing, source certification, human authority, downstream side effects, or the numeric/runtime implementation beneath exact work. That boundary is the product, not a disclaimer added after an answer.

LOOM TEST

If a reviewer cannot distinguish what Loom inferred, what the solver calculated, what the source said, and who approved the consequence, the cognitive artefact is incomplete.

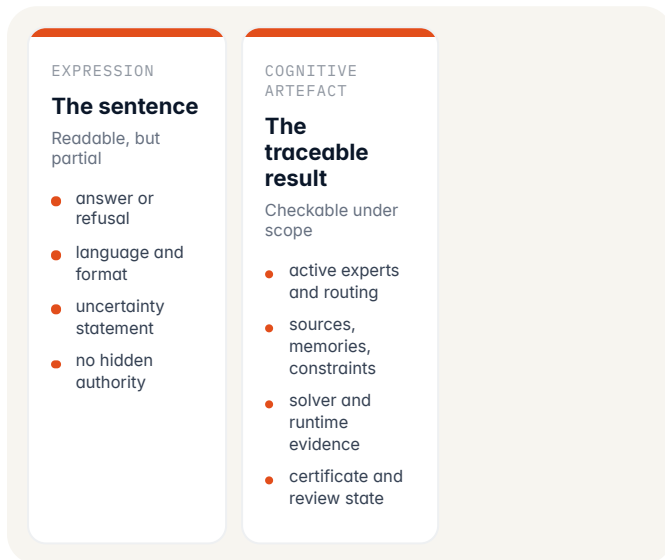


The fluent sentence is the last expression of several interleaved paths with different evidence and authority boundaries.

THE COGNITIVE ARTEFACT • RESULT

A fluent answer is not a complete trace

Loom should return an answer or typed refusal with source references, expert and constraint identity, solver artefacts where used, and trace material that a reviewer can inspect.



Expression is a view over cognitive and proof artefacts, not a replacement for them.

For the eligibility question, a useful result can say which policy condition was not met, cite the relevant source, show whether a numeric solver was used, list material constraints, expose uncertainty, and stop short of making the officer's decision. A generated reason that cannot point back to its cognitive path is only a plausible explanation.

Keep four questions separate

- What did the selected experts and retrieval path contribute?
- Which constraints fired, conflicted, or remained unresolved?
- What exact computation or solver produced a numeric artefact?
- Who or what owns the final human or operational consequence?

NO AUTHORITY BY FLUENCY

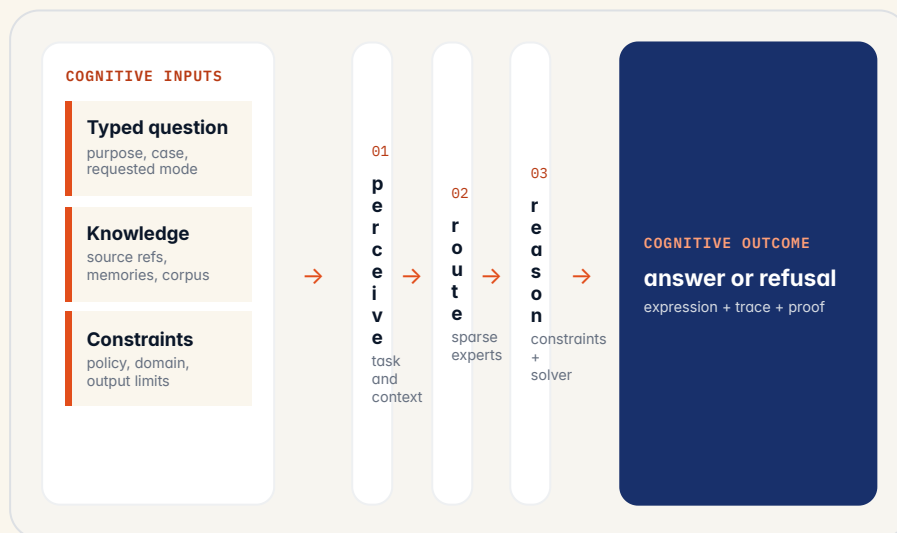
Loom can express a recommendation. Fluency does not grant it policy authority, professional competence, or permission to change an external system.

01

ROUTE AND REASON

Keep the sparse path visible

Loom describes four weaves: perception, memory and search, reasoning and solvers, and expression. The route through them should be inspectable rather than hidden behind a single model label.



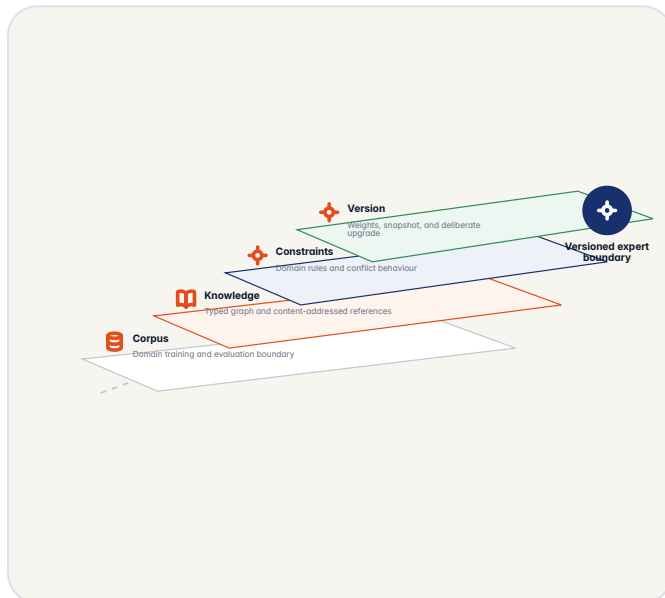
The four weaves form a cognitive path; they do not erase the boundary between reasoning and authority.

ROUTE AND REASON · CATALOGUE

Each expert carries its own reason to be active

The website describes 528 specialists across 10 domain clusters: 456 in the production base and 72 expansion modules in training through the stated pipeline. The catalogue is not one anonymous capability.

An expert in Loom is described as carrying its own domain corpus, knowledge graph, constraint set, evaluation suite, and content-addressed version. The route should record why the active set was selected and what candidate specialists were excluded, rather than reducing the event to “the model answered.”



An expert is a versioned participant with a domain boundary, not a hidden slice of a giant model.

Routing questions

- Which catalogue version and cluster were searched?
- Which signatures and gate subsets selected the active experts?
- Which experts were candidates but not activated, and why?
- What happens when routing confidence is low or specialists disagree?
- Can the same route be inspected later without assuming today's catalogue is unchanged?

CATALOGUE BOUNDARY

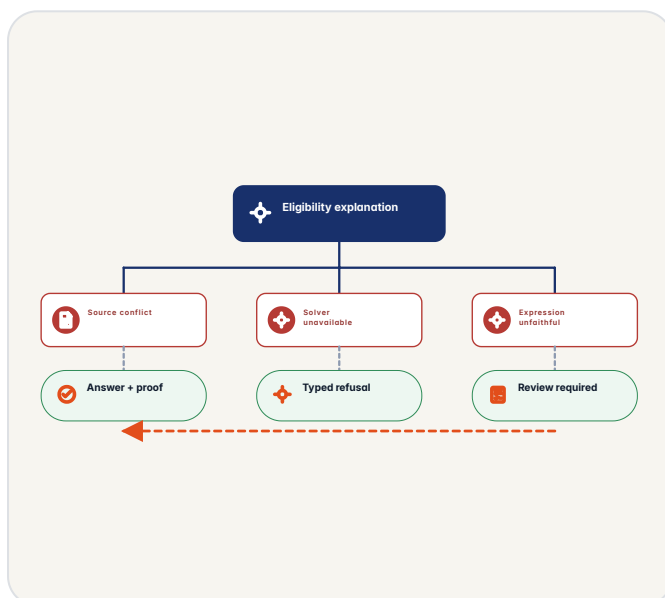
Adding a specialist should be a pinnable catalogue event. It should not silently rewrite historical routing or make every answer depend on the newest expert.

ROUTE AND REASON • DOMAIN RULES

Constraints should surface conflict instead of averaging it away

Loom carries constraints with the expert and source path so a hard conflict can become a typed refusal, review task, or explicit limitation.

The eligibility case may contain a policy threshold, a source freshness rule, an exception lane, and a communication constraint. If the source says one thing and the active policy says another, fluent synthesis is the wrong success condition. The result should identify the conflict, affected scope, owner, and safe next state.



A constraint failure becomes a named result state instead of being averaged away behind fluent language.

Conflict record

- source and constraint identities, versions, and authority
- the exact input or clause that caused conflict
- which expert or solver detected it
- whether the conflict is resolvable, escalatable, or outside scope
- the owner and permitted next action

REFUSAL IS USEFUL

A refusal that names the missing source, conflicting constraint, and responsible owner is more valuable than an answer that makes the conflict invisible.

ROUTE AND REASON · COGNITION AND COMPUTATION

Exact solver use is a separate contract

Loom can select or explain a solver. The solver and runtime still need their own input, numeric, verification, and output boundaries.

The eligibility explanation may require an affordability or threshold calculation. Loom can route the task to a typed solver and preserve its artefact. It should not imply that natural-language reasoning and exact computation are the same operation, or that a proof certificate validates the policy itself.

Illustrative Loom cognitive record

```
task = eligibility-explanation-4f9e
question = why was the case placed in review?
catalogue = loom://production-456@manifest
active_experts = [finance-eligibility-07, explanation-02]
sources = spindle://policy-6.2 + case-snapshot
constraints = [freshness, threshold, disclosure]
solver = numerus://affordability@profile-2026
route = seeded + gate-subset + sorted-state
result = sourced_reason | typed_refusal
proof = aion://certificate
```

Illustrative fields. The deployed Loom interface and supported proof format own the actual schema.

Solver questions

- Were units, precision, rounding, and input commitments declared?
- Which solver, version, runtime, and target performed the calculation?
- Was the result independently compared against a high-precision reference?
- Can a reviewer distinguish solver output from Loom expression?
- What happens when the exact path is unavailable or outside the requested determinism mode?

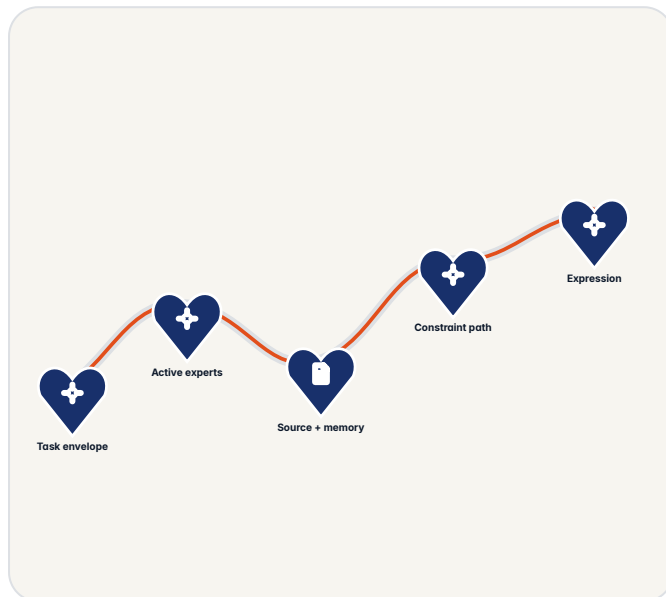
NUMERICAL BOUNDARY

A correctly rounded result can still be based on the wrong source, threshold, or purpose. Numeric proof is necessary for the calculation, not sufficient for the decision.

TRACE AND VERIFY • EXPRESSION

The trace names what fired

An explanation is stronger when the reviewer can inspect the constraints, active specialists, source references, solver artefacts, and expression that produced it.



Each stop answers a different question. A fluent paragraph is not a substitute for the route.

Expression checks

- Does the answer preserve the source and constraint meaning?
- Are uncertainty, refusal, and missing evidence visible?
- Can a reviewer separate retrieved text from reasoning and solver output?
- Does the language expose the human or operational gate that remains?
- Is the expression versioned and joined to the cognitive artefact?

FAITHFULNESS BOUNDARY

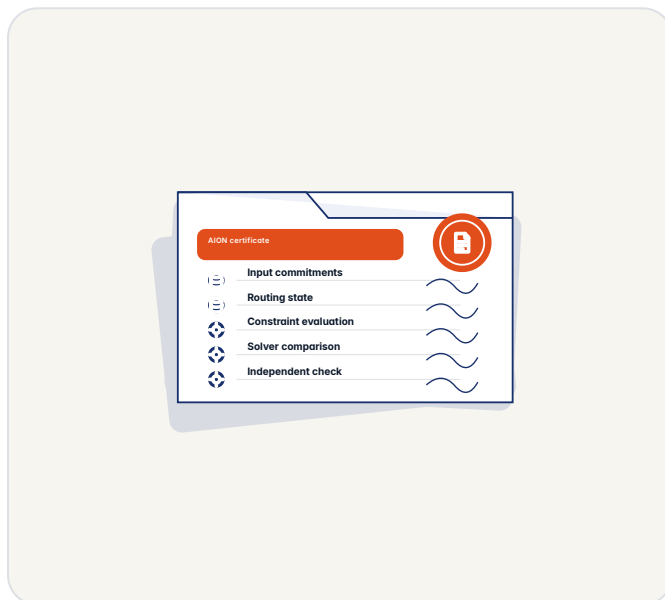
A trace can show what happened inside Loom. It cannot make an expression faithful merely because the trace exists.

TRACE AND VERIFY • CERTIFICATE

AION verifies the declared path, not the truth of the world

The AION proof certificate can make a selected cognitive result portable and independently checkable under its supported format.

The Loom route describes an AION certificate attached on emit and verification with the customer's own tooling. That is valuable at the edge of trust: a reviewer can check commitments to inputs, constraints, route, result, and proof material outside the producing service. It does not certify that the source was correct, the policy was legitimate, or the answer was suitable for every person.



The verifier checks the declared packet without gaining authority to change the source or downstream decision.

Certificate limits

- It does not make a bad source authoritative.
- It does not prove that the active constraints represent a fair or lawful policy.
- It does not replace model, expert, solver, or domain validation.
- It does not prove that the expression was understandable or that a human approved the consequence.

INDEPENDENT CHECK

Can a reviewer inspect the Loom certificate with retained inputs and verifier material without giving the verifier permission to run a new cognitive task?

TRACE AND VERIFY • REPEATABILITY

Determinism is a declared replay class

The website describes fixed-seed, bit-identical replay, integer and bitwise inference, sorted iteration, seeded random draws, and no wall-clock dependency. A proof must state where those claims apply.

Loom's determinism story is useful because it names concrete sources of drift: floating-point non-associativity, reduction order, vendor math kernels, iterable ordering, random seeds, and time. It still needs a supported target matrix, catalogue and source snapshot, runtime identity, resource boundary, and test corpus.

REPLAY CLASS	PIN	EXPECTED OBSERVATION
Cognitive route	catalogue, active experts, source, constraints	same route or explicit supported difference
Numeric solver	inputs, profile, runtime, precision, seed	same result within declared exact class
Expression	language, prompt/flow, model, output mode	same or bounded expression difference
Certificate	packet, verifier, schema, key	independent check passes or names drift
Environment	CPU, OS, package, compiler, target	supported result and resource state

"Replayable" must name the part of Loom that is expected to repeat and the part allowed to vary.

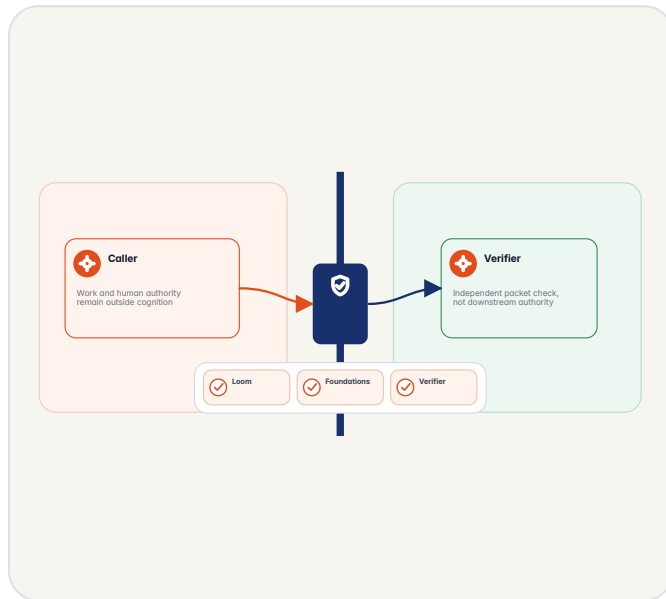
Replay questions

- Is the exact catalogue and source snapshot available?
- Were seed, ordering, time, and resource assumptions satisfied?
- Did routing change because the candidate pool or expert version changed?
- Can divergence be localised to source, route, constraint, solver, expression, or environment?
- Does the packet record a supported replay class instead of making a global claim?

COMPOSITION • FABRIC, SPINDLE, NEXUS, CORE

Loom can be evaluated without moving the authority boundary

Loom is a cognitive component in a larger system. A useful integration makes the surrounding work, sources, tasks, runtime, and human authority explicit.



Loom cognition becomes trustworthy when the surrounding authority, source, runtime, and work boundaries remain separate.

The first integration should request one cognitive artefact from a Fabric or Nexus workflow, use a bounded Spindle source projection, select a declared expert and solver path, and return an answer or refusal with trace. The caller owns the human gate and external effect. Loom owns the cognitive path and its artefacts.

DELETION TEST

Remove Loom from the design and name the cognitive state, routing, reasoning, solver relationship, and expression contract that disappear. Do not assign those responsibilities to a generic chat box.

02

PROVE LOOM

Make the cognitive runtime answerable

A Loom proof tests routing, sources, constraints, exact work, expression, certificate, deployment, resource use, and refusal rather than rewarding a fluent demo answer.

LOOM PROOF · ELIGIBILITY From task envelope to independently checked artefact			
01	TYPED TASK question, purpose, and authority are bounded	caller	bound
02	CATALOGUE + ROUTE active experts and candidates are fixed	Loom	pinned
03	CONSTRAINTS + SOURCES domain and knowledge context are inspectable	source / domain owner	linked
04	SOLVER + RUNTIME exact work meets declared profile	method owner	verified
05	EXPRESSION + REFUSAL the result remains faithful to the path	Loom	reviewed
06	CERTIFICATE + REPLAY a supported independent check succeeds	verifier owner	portable

A Loom proof is a cognitive-path proof, not a single benchmark or confidence score.

PROVE LOOM · ACCEPTANCE CORPUS

Run the cognitive runtime against hard cases

The corpus should include ordinary answers, sparse-route changes, constraint conflict, solver failure, source withdrawal, expression drift, adversarial input, and replay.

CORPUS SLICE	INCLUDE	EXPECTED OBSERVATION
Normal task	typed question, current source, eligible experts	answer, trace, and certificate close
Route change	catalogue or expert version changes	active set and historical route remain visible
Constraint conflict	policy and source disagree	typed refusal or named escalation
Solver boundary	missing, invalid, or unsupported exact path	numeric claim is withheld or limited
Expression drift	same path under changed expression conditions	difference is classified, not hidden
Adversarial input	prompt/source injection, poisoned context, broad request	scope and authority hold
Replay	fixed seed, snapshot, supported environment, verifier	declared replay class passes or explains drift
Recovery	catalogue unavailable, resource exhaustion, restore	safe failure and packet retention

The corpus is part of the cognitive product decision. A fluent happy path alone is not acceptance.

Measure the path, not only the sentence

- routing candidates, active experts, catalogue version, and route cost
- source coverage, constraint conflict, solver output, and expression fidelity
- certificate verification, replay class, environment, and resource state
- refusal quality, human review, downstream authority, and affected-person explanation
- access denial, incident record, restore outcome, and exit package

Record raw inputs, routes, traces, outputs, failed assertions, reviewer decisions, resource use, and gaps. A cognitive runtime earns reliance when its limitations are as inspectable as its answers.

PRODUCTION RECORD

The organisation decides where Loom may be trusted

A product proof ends with a bounded adoption decision, not with the last successful generated answer.

The product owner, ML engineer, domain owner, model-risk reviewer, security owner, source owner, runtime owner, human approver, procurement/legal owner, and affected-user representative may each own a different condition. They decide task scope, expert and catalogue boundary, source projection, solver class, expression mode, human gate, resource budget, deployment, support, retention, export, and exit state.

LOOM PRODUCTION RECORD	Eligibility explanation proof decision
TASK	One source-aware eligibility explanation BOUNDED
ROUTE	Named catalogue, expert set, source, and constraints SCOPED
COMPUTE	Solver, runtime, seed, precision, and certificate LINKED
AUTHORITY	Recommendation, reviewer, refusal, and consequence gate REVIEWED
OPERATION	Latency, capacity, incident, restore, support OWNED
EXIT	Packet, verifier, source, model, and transition export DATED
DECISION	Go / narrow / hold / stop SIGNED
The record should remain understandable after the original model, benchmark, and integration team have moved on.	

The smallest credible next step

Run one eligibility or policy explanation through a fixed catalogue, source projection, constraint set, solver boundary, review task, certificate, and supported replay environment. Make the domain owner, ML engineer, security owner, reviewer, and affected-user representative inspect the same cognitive packet. If they cannot answer their different questions without reconstruction, narrow the proof.

PRIMARY ACCEPTANCE CRITERION

Loom identifies the task, sources, memories, active experts, constraints, solver and runtime, expression, refusal or uncertainty, human gate, certificate, and supported replay result for every selected cognitive artefact.

That proves one cognitive responsibility. It is not permission to generalise from one route to every domain, model, expert, solver, user, policy, or consequence.

FIND A SUBJECT

Subject index

Page references point to the substantive explanation, worked example, control, boundary, or decision record, not merely to a passing label.

A Acceptance criteria 14–15	A Adoption 15	A AION 8, 10, 12
A Authority 3–5, 7, 10, 12–15	B Benchmarking 13, 15	B Boundaries 3, 5–6, 8–9, 11–12, 14–15
C Capabilities 6	C Contracts 8, 12	C Cost and budgeting 14–15
D Dependencies 11	D Deployment 13, 15	D Determinism 8, 11
E Evidence 3–4, 9	E Exit 14–15	F Fabric 12
F Failure 7, 14	H Human authority 3, 12, 14	I Identity 4, 11
I Incident response 14–15	I Inputs 7–8, 10, 14	I Integration 12, 15
K Kera 12	K Keys and signing 6	L Ledger 15
L Limits and exclusions 5, 7, 10	L Loom 3–15	M Manifests 8
M Models and solvers 3–15	N Nexus 12	N Numerus 8, 12
O Operations 8, 15	O Outputs 5, 8–9, 11, 14	O Ownership 3, 7, 13, 15
P Policies 3–5, 7–8, 10, 14–15	P Procurement 15	P Proof 3–5, 7–8, 10–11, 13, 15

SUBJECT INDEX CONTINUED

Subject index · R–W

Terms are grouped alphabetically. Consecutive page references are collapsed into ranges.

R Recovery 14–15	R Replay 11, 13–15	R Responsibility 12, 15
R Retention 14–15	R Risk 15	R Rules 7
S Security 15	S Sources 3–5, 7–15	S Sovereignty 4, 10
S Spindle 8, 12	S State 4, 7–8, 10–12, 14–15	T Testing 3, 11–12
T Trace 4–5, 9–12, 14	V Validation 10	V Verification 8–11, 14
V Versioning 6, 8, 11, 14	W Workflows 3, 12	

CLOSING PRODUCT RECORD

Start with the answer you most dread defending

Choose one cognitive task, name its authority boundary, and make the reasoning path checkable before the reviewer asks why the answer was produced.

Write the task identity, purpose, context projection, catalogue and active expert set, source and memory references, constraints, solver and runtime, seed and replay class, expression, refusal state, certificate, reviewer, deployment posture, recovery, retention, export, and stop conditions.

Then choose the hard case: a conflicting source, routing change, unavailable expert, solver failure, poisoned document, expression drift, unsupported replay, resource exhaustion, or certificate mismatch. A product handbook is useful when it makes the failure concrete enough to run.

TASK

One bounded cognitive question or artefact

Do not begin with every model workload

ROUTE

Catalogue, experts, sources, constraints, and solver

Each has an effective version

EXPRESSION

Answer, refusal, uncertainty, and trace

Fluency is not authority

EVIDENCE

Cognitive artefact, certificate, verifier, and replay

Scope matters

DECISION

Go, narrow, hold, or stop

Signed with review triggers

THE LOOM STANDARD

An answer can be fluent. A dependable cognitive artefact shows the task, source path, active specialists, constraints, exact work, expression, authority boundary, and proof state that existed when Loom responded.