Quuipu

QUUIPU RESEARCH & ARCHITECTURE SERIES

WHITEPAPER 05

Trust Architecture

Provenance, Explainability, Human Control and Bounded Intelligence



Foundational Whitepaper · English

Quuipu Edition 2 · v1

Publication identity

Publication
Trust Architecture

Series
Quuipu Research & Architecture Series

Publication number
Whitepaper 05

Edition
Quuipu Edition 2 · v1

Language
English

Historical edition
Cognisphere Edition 1

This publication succeeds Cognisphere Edition 1. Historical PDFs remain immutable.

Contents

01. Trust as architecture

02. Trust Architecture

03. Provenance

04. Evidence

05. Confidence

06. Uncertainty

07. Explainability

08. Claimability

09. Human control

10. Fallback

11. Auditability

12. Bounded interaction

13. Policy architecture

14. Governance

15. Logging

16. Decision trace

17. Trust states

18. LLM integration

19. EU AI Act: transparency context

20. Transparency by design

21. Human oversight patterns

22. Trust in multi-tenant systems

23. Security and privacy

24. Testing trust

25. Trust metrics

26. Governance operating model

27. Trust invariants

28. The final promise

A. Decision Trace reference model

B. Trust failure matrix

C. EU transparency design checklist

D. Trust Architecture invariants

Preamble

Provenance, Explainability, Human Control and Bounded Intelligence

> This publication succeeds Cognisphere Edition 1. Quuipu Edition 2 preserves the historical thesis while updating active brand and publication identity.

Preamble

Provenance, Explainability, Human Control and Bounded Intelligence

FOUNDATIONAL PRINCIPLE

Trust without surrender Adaptive intelligence becomes trustworthy when source authority, system limits and human control remain visible and enforceable.

The experience may adapt. The source of truth does not.

Canonical Source ↓

Evidence & Provenance ↓

Bounded Runtime ↓

Policy & Claimability ↓

Decision Trace ↓

Human Control ↓

Adaptive Experience

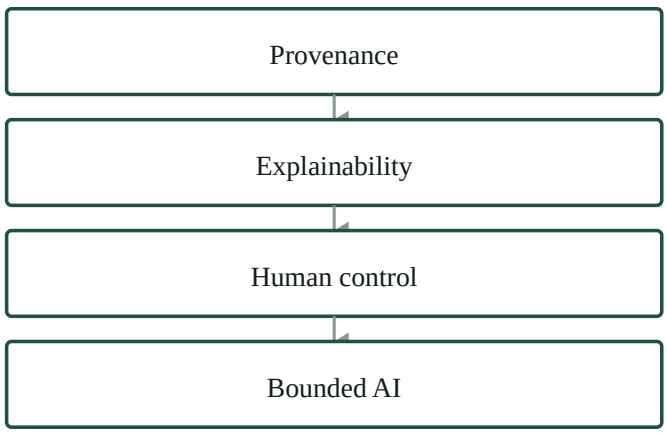


Figure 1 — Trust chain.

TRUST MODEL

Four trust planes Plane Primary responsibility Core artifacts

Source & Evidence Authority and support SourceRef, EvidenceRef, provenance

Runtime Decision Bounded resolution and policy Policy ID, reason code, integrity/trust state

Experience & Control Disclosure and user agency Source link, mode state, override, fallback

Governance & Audit Accountability and change control Logs, review gates, retention, incident trace

CHAPTER 01

Trust as architecture

Trust in an adaptive experience cannot depend on the impression of intelligence. It must derive from observable properties: information origin, system limits, the ability to explain a transformation, human control and the ability to return to source.

Trust is earned by structure Quuipu treats trust as the result of technical and governance contracts. Fluent output is not more reliable because it sounds convincing.

The invariant The experience may adapt. The source of truth does not. Adaptation may change form, journey and granularity; it does not transfer authority from source to runtime.

Trust surface The interface should make origin, state, qualifications and control perceptible without turning every interaction into a technical audit.

CHAPTER 02

Trust Architecture

Trust Architecture is the cross-cutting layer connecting runtime, experience and governance so that every material adaptation can be traced to sources, policies, decisions and controls.

Four planes The model distinguishes a Source & Evidence Plane, Runtime Decision Plane, Experience & Control Plane and Governance & Audit Plane.

Separation of concerns Evidence establishes support; confidence describes an internal assessment; explainability reconstructs why; governance establishes what is allowed. None of these concepts substitutes for the others.

Bounded intelligence Intelligence is useful when it operates within explicit scope, capabilities, policies and source authority.

Source & Evidence Plane ↓

Runtime Decision Plane ↓

Experience & Control Plane ↓

Governance & Audit Plane

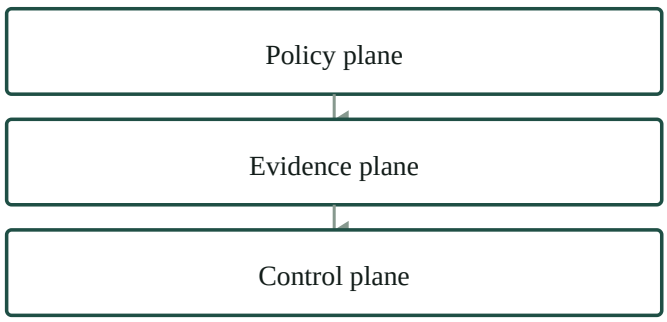


Figure 2 — Trust Architecture planes.

CHAPTER 03

Provenance

La provenance collega ciò che l'utente vede a ciò da cui deriva. È una condizione di fiducia, non un ornamento bibliografico.

Source provenance Identifies source, version, locator and authority scope.

Transformation provenance Records which capability or transformation produced the representation.

Decision provenance Connects source selection, policy, uncertainty and fallback to the emission decision.

Progressive disclosure The UI may show a simple source link while audit and debugging retain a richer trace.

CHAPTER 04

Evidence

Evidence is admissible support for claims and transformations. Its role is to prevent plausibility, model memory or semantic similarity from being confused with proof.

Evidence units Passages, records, structured fields or relationships can become evidence units when origin and scope are preserved.

Coverage A material claim needs sufficient coverage, not merely a generally relevant source.

Strength Evidence strength depends on authority, directness, version and domain policy.

Evidence gaps A gap should trigger qualification, clarification, scope reduction or fallback.

CHAPTER 05

Confidence

Confidence is an operational signal, not a universal measure of truth. It must be interpreted in the context of the decision type and available evidence.

Confidence is not authority A model may be highly confident and still lack authority to make a claim.

Calibrated use Confidence can support routing, escalation or fallback choice, but should not be the only gate.

User projection Numeric confidence can create false precision; qualified states are often more understandable.

Calibration Thresholds should be validated by domain and materiality rather than copied across use cases.

Signal Useful for Not sufficient for

Confidence Routing / escalation Truth or authority

Evidence strength Claim support Policy permission

Integrity state Semantic safety User disclosure by itself

Human review Additional assurance Permanent correctness

CHAPTER 06

Uncertainty

Quuipu must represent source, resolution, transformation and policy uncertainty without hiding it behind assertive language.

Known unknowns Ambiguity, missing data and conflicts can be explicitly modeled.

Uncertainty propagation If a premise is uncertain, derived representations retain that qualification.

User handling The system can expose alternatives, ask for clarification, reduce scope or return to Original.

Governance threshold High-materiality domains require more conservative thresholds.

CHAPTER 07

Explainability

Explainability means providing a useful, proportionate explanation of how the system arrived at a representation or decision.

Why this information Explain which context signals and sources made information relevant.

Why this representation Explain why Simple, Step-by-step, Explore or another composition was selected.

Why not Explain why a capability was refused, qualified or degraded.

No fake rationale Explanations derive from actual trace and decision data, not model-invented post-hoc stories.

CHAPTER 08

Claimability

Claimability defines what the system is authorized to assert. It is the bridge between evidence, policy and user-facing language.

Canonical claim An assertion directly attributable to an authoritative source.

Supported derivation A conclusion supported by sources under an allowed rule.

Qualified inference A useful but conditional inference requiring qualification.

Unsupported claim Must not be emitted as trusted output.

Materiality The more a claim affects rights, money, safety, obligations or decisions, the higher the threshold.

Claim class Trust treatment Typical action

Canonical Highest source attribution Show source

Supported derivation Evidence-backed Show provenance

Qualified inference Conditional Disclose qualification

Unsupported Below threshold Block / fallback

CHAPTER 09

Human control

Human control does not mean putting a person in every loop. It means designing real points of intervention, correction and accountability.

User override The user can change intent, mode or source, or return to the standard experience.

Operator override An authorized operator can disable capabilities, change policy, exclude sources or force fallback.

Review gates Some claims or domains may require human review before publication or action.

No dark autonomy The system does not autonomously expand its mandate beyond authorized capabilities.

CHAPTER 10

Fallback

Fallback is a trust property: it shows that Quuipu prefers a less adaptive experience to an insufficiently supported response.

Technical fallback Runtime unavailable, timeout or error: return to the canonical surface.

Semantic fallback Runtime available but integrity insufficient: Original, reduced scope, alternatives or refusal.

Policy fallback A policy prohibits transformation or requires human review.

Graceful degradation Degradation should be understandable and preserve the user's fundamental journey.

Normal adaptive experience ↓

Qualification ↓

Reduced scope ↓

Return to Original ↓

Human review ↓

Refusal

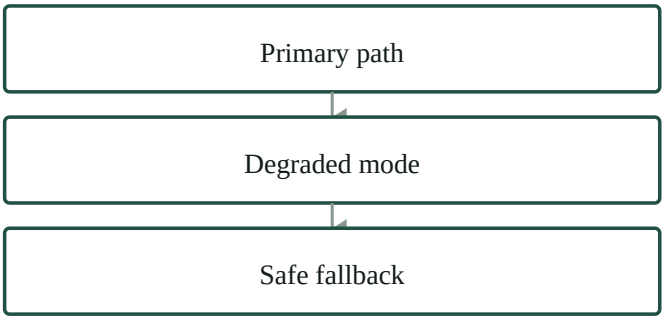


Figure 3 — Fallback ladder.

CHAPTER 11

Auditability

Auditability is the ability to reconstruct relevant events without turning the product into a surveillance system.

What to reconstruct Relevant context input, source set, policy, capability, decision state, output reference and fallback.

Minimality Trace collects what is needed for safety, quality and accountability while avoiding unnecessary data.

Immutability and retention For high-value events, log integrity and retention policies should be defined.

Audit views Technical debugging, governance audit and user explanation require different levels of granularity.

CHAPTER 12

Bounded interaction

Quuipu does not assume that more conversational freedom equals a better experience. Interaction can be bounded to preserve authority, safety and comprehensibility.

Declared intent Users state what they are trying to understand without automatic profiling.

Bounded topics Explore follows governed relationships rather than unlimited generated paths.

Capability boundaries Each capability exposes defined inputs, outputs, policy and failure modes.

Context boundary The system uses context needed for the current purpose, not an unlimited behavioral identity.

CHAPTER 13

Policy architecture

Policies turn trust principles into versionable operational rules.

Source policy Defines allowed sources and their authority.

Transformation policy Defines permitted transformations by content class.

Claim policy Defines claims requiring primary evidence, qualification, double-checking or refusal.

Interaction policy Defines modes and capabilities available in a scope.

Retention policy Defines which traces and state may be retained and for how long.

CHAPTER 14

Governance

Governance assigns responsibility and makes the rules bounding intelligence changeable in a controlled way.

Ownership Source owners, product owners, security, legal/compliance and engineering have distinct responsibilities.

Change control Changes to sources, models, prompts, policies or capabilities may require review and release gates.

Exception process Exceptions are explicit, preferably temporary and auditable.

Governance as runtime input Governance decisions become executable configuration and contracts rather than remaining only in documents.

CHAPTER 15

Logging

Quuipu logging must separate technical observability, decision trace and behavioral analytics.

Operational logs Errors, latency, timeouts, capability health and technical fallback.

Decision logs Applied policy, source class, integrity state, reason code and human override.

Behavioral analytics A separate optional category; not a prerequisite for Trust Architecture.

Privacy by design No telemetry/no persistence can be valid for cognitive context while retaining minimized health logging.

CHAPTER 16

Decision trace

The decision trace is the structured sequence explaining how the runtime moved from context and sources to an emitted experience.

Trace, not chain-of-thought The trace contains observable decision data - rules, source selection, state transitions and reason codes - not unnecessary hidden reasoning.

Deterministic anchors Policy ID, source version, capability version and integrity state make the trace verifiable.

Branching The trace records why a path was selected, qualified or discarded.

Replayability Where required, trace can reconstruct decision conditions without claiming perfect identity from a probabilistic model.

Trace anchor Example Purpose

Context declaredIntent + mode Why the request was resolved this way

Source sourceId + version What authority was used

Policy policyId + version What rule bounded the decision

Capability capability + version What transformation acted

Decision reasonCode + state Why emitted/qualified/blocked

Override actor + action What human control changed

CHAPTER 17

Trust states

An experience can expose trust states distinct from technical runtime state.

Trusted Source, evidence and policy meet the required threshold.

Qualified The experience is useful but contains uncertainty or inference requiring disclosure.

Human-reviewed An operator validated the result or result class.

Degraded Cognitive capability is reduced and the system uses a safer surface.

Blocked Policy or integrity prevents emission.

CHAPTER 18

LLM integration

If Quuipu uses LLMs, the model remains a capability subordinate to Trust Architecture.

Provider abstraction The runtime can change model without losing provenance, policy, fallback and trace.

Prompt is not governance A prompt instruction does not replace architectural enforcement.

Generated output as candidate Model output is treated as a candidate for validation, not as an authoritative source.

Disclosure Where interaction or content falls under applicable transparency duties, disclosure is designed into the experience.

CHAPTER 19

EU AI Act: transparency context

From 2 August 2026, Article 50 of the EU AI Act applies to certain categories of AI systems and content. The European Commission published dedicated guidelines on 20 July 2026 clarifying scope and application. This section provides design context, not legal advice.

Interactive AI For directly interactive AI systems, Article 50(1), where applicable, requires people to be informed that they are interacting with AI unless this is obvious.

Synthetic content Article 50(2), where applicable, concerns machine-readable marking of AI-generated or manipulated synthetic content, with specific scope, exceptions and transitional rules.

Deployers Additional duties concern, among other cases, deepfakes and certain AI-generated public-interest text without human review or editorial control.

Architecture implication For Quuipu, disclosure, provenance, content marking, logging and human control should be configurable capabilities rather than retrofits.

Legal boundary Concrete applicability depends on provider/deployer role, system type and deployment context and requires case-by-case assessment.

AI Act transparency area Architecture response Boundary

Direct AI interaction Visible AI invocation/disclosure Only where legally applicable or product-appropriate

Synthetic content marking Emission metadata/marking capability Depends on content/system scope and exceptions

Deepfake/public-interest disclosure Deploy-time disclosure controls Role and context specific

Compliance evidence Versioned policy + trace + logs Not a substitute for legal assessment

CHAPTER 20

Transparency by design

The most effective transparency is embedded in experience contracts before deployment.

AI interaction disclosure Users should be able to understand when an AI capability is actually invoked, where required or appropriate.

Derived-content disclosure A derived representation should be distinguishable from canonical content.

Machine-readable metadata Where required, the emission layer can attach markers or metadata without altering the canonical source.

First exposure User-facing disclosures should appear at a useful moment rather than only in policies or footers.

CHAPTER 21

Human oversight patterns

Different domains require different oversight patterns.

Human-on-request The user can escalate or request verification.

Human-on-exception The system involves an operator for conflict, low evidence or high-materiality claims.

Human-before-publish Output intended for external publication may require approval.

Human-in-governance Experts define source authority, policy and tests even when they do not intervene in every interaction.

Oversight pattern Trigger Latency

On request User asks for verification Asynchronous possible

On exception Conflict / low evidence Before trusted completion

Before publish External/high-impact publication Approval gate

In governance Policy/source/model changes Release cycle

CHAPTER 22

Trust in multi-tenant systems

In a multi-tenant deployment, trust requires isolation not only of data but also of authority, policy, trace and control.

Tenant policy Each tenant can have distinct source, claimability and retention rules.

Trace isolation Audit and decision traces do not cross tenant boundaries.

Shared capabilities Models and core may be shared while context, evidence and governance remain isolated.

Delegated administration Authorized roles manage policy without indiscriminate access to other tenants' content.

CHAPTER 23

Security and privacy

Trust Architecture and security architecture intersect but are not the same.

Least privilege Host, runtime and capabilities access only sources and context required for purpose.

Credential boundary Raw credentials do not become part of cognitive context.

Sensitive traces Decision traces and logs can contain sensitive metadata and need access control, retention and minimization.

No trust without security Provenance and explainability lose value if source identity or logs can be altered without control.

CHAPTER 24

Testing trust

Trust must be verified through failure injection, policy tests, provenance tests and human scenarios.

Policy tests Verify that prohibited claims or capabilities are actually blocked.

Trace tests Verify completeness and consistency of reason codes and anchors.

Fallback tests Simulate timeout, missing source, conflict, uncertainty and operator override.

Disclosure tests Verify that states and disclosures are perceivable, accessible and timely.

Red-team trust Adversarial tests seek authority confusion, prompt bypass, cross-tenant leakage and fabricated explanations.

CHAPTER 25

Trust metrics

Metrics should measure whether the system deserves trust, not only whether it is perceived as trustworthy.

Provenance completeness Share of derived outputs with a sufficient source path.

Unsupported claim rate Frequency of claims emitted without adequate evidence.

Correct fallback rate How often the system degrades when it should.

Override effectiveness Ability of human override to produce a real, traceable behavior change.

Disclosure coverage Coverage of situations where AI interaction or derived content should be visible.

Metric What it tests Failure signal

Provenance completeness Traceability Derived output without source path

Unsupported claim rate Evidence discipline Claims beyond support

Correct fallback rate Conservatism Unsafe continuation

Override effectiveness Human control Override has no effect

Disclosure coverage Transparency AI/derived state hidden

CHAPTER 26

Governance operating model

A Trust Architecture becomes operational when roles, review and incident processes are clear.

Trust owner Maintains invariants, risk taxonomy and metrics.

Domain authority Defines source authority and materiality within the domain.

Engineering Implements enforcement, logging, isolation and tests.

Compliance/legal Interprets applicable obligations and approves disclosure policy where needed.

Incident review Analyzes trust failures and produces verifiable remediation.

CHAPTER 27

Trust invariants

Invariants are the rules that must survive changes of model, host and deployment.

Source truth is stable Adaptation does not silently change the authoritative source.

Evidence precedes claim Material claims require support.

Uncertainty remains uncertainty The UI does not turn uncertainty into certainty.

Control is effective Override and fallback must actually change behavior.

Trace is factual Explainability derives from real events, not generated rationalizations.

Governance is enforceable Policies are enforced by runtime, not merely documented.

CHAPTER 28

The final promise

Quuipu Trust Architecture aims to make adaptive intelligence usable precisely by bounding its power. The system can help more when it can show where information came from, what it did, what it can support and when it must stop.

Adaptive experience Form can respond to context.

Stable authority The authoritative source retains its role.

Visible limits Uncertainty, fallback and policy become legible parts of experience.

Human sovereignty People and organizations retain real points of control.

The principle The experience may adapt. The source of truth does not.

CHAPTER A

Decision Trace reference model

Field Example Trust purpose

traceId immutable event ID Correlation

contextRef session-scoped snapshot Resolution context

sourceRefs IDs + versions Authority

evidenceRefs locators + strength Support

policyRef policy/version Governance

capabilityRef mode/model/version Transformation

decision emit/qualify/fallback/block Outcome

reasonCodes structured codes Explainability

overrideRef actor/action/time Human control

CHAPTER B

Trust failure matrix

Failure Risk Required response

Missing provenance Untraceable output Qualify or block

High confidence / low evidence False certainty Evidence gate wins

Policy bypass Unbounded intelligence Block + incident trace

Fabricated explanation False transparency Use factual reason codes only

Override ignored Loss of human control Fail safe + alert

Cross-tenant trace Privacy/authority leakage Reject + isolate

Hidden AI invocation Transparency failure Disclosure remediation

Stale source Wrong authority state Invalidate / refresh

CHAPTER C

EU transparency design checklist

This checklist translates current EU transparency context into architecture questions. It is not a legal compliance checklist.

Question	Architecture hook	Owner
Does the user directly interact with an AI capability?	Invocation state + disclosure surface	Product / legal
Can the system generate or manipulate synthetic content?	Marking metadata at emission	Engineering / compliance
Is derived content externally published?	Deploy-time disclosure policy	Publisher/deployer
Is human review/editorial control present?	Review state + decision trace	Governance
Can compliance measures be evidenced?	Versioned policy, logs, trace	Compliance / audit
Are pre-2 Aug 2026 systems relevant?	Transition/version inventory	Legal / operations

CHAPTER D

Trust Architecture invariants

TA-01 · Stable source of truth Adaptation never silently redefines canonical authority.

TA-02 · Evidence before material claim Support is required before assertion.

TA-03 · Confidence is not authority Confidence never overrides evidence or policy.

TA-04 · Uncertainty is preserved Qualification propagates into the experience.

TA-05 · Explainability is factual Explanations derive from real decision data.

TA-06 · Human control is effective Override changes runtime behavior and is traceable.

TA-07 · Fallback is first-class Safer degradation exists for technical, semantic and policy failure.

TA-08 · Intelligence is bounded Capabilities operate inside explicit scope and policy.

TA-09 · Governance is executable Policy is enforced by runtime contracts.

TA-10 · Logging is minimized Auditability does not require unlimited behavioral collection.

TA-11 · Tenant trust is isolated Authority, policy and trace remain tenant-scoped.

TA-12 · Transparency is designed in Applicable disclosure is an experience capability, not an afterthought.

APPENDIX E

Relationship to Semantic Integrity Semantic Integrity (WP04) Trust Architecture (WP05) Combined property

Preserves meaning Makes preservation visible/governable Trustworthy adaptation

Evidence & claimability Policy & decision trace Defensible claims

Integrity states Trust states + human control Safe experience behavior

Semantic fallback Technical/policy/human fallback Layered resilience

Source authority Governance ownership Stable source of truth

Provenance graph Auditability/explainability Reconstructable decisions

REFERENCES

Regulatory references European Commission, Guidelines on transparency obligations for providers and deployers of AI systems, published 20 July 2026. The Commission states that Article 50 transparency obligations apply from 2 August 2026. European Commission, Transparency obligations under Article 50 of the AI Act - Questions & Answers, 2026. This resource clarifies provider/deployer duties, direct AI interaction, synthetic-content marking, exceptions and transition rules. European Commission, Code of Practice on Transparency of AI-generated Content, 2026. The voluntary Code supports compliance with marking and labelling obligations under Article 50(2), (4) and (5).

Legal note The regulatory discussion in this whitepaper is architecture-oriented and does not constitute legal advice. Applicability should be assessed against the final legal text, Commission guidance, organizational role and deployment context.

Trust Architecture Provenance, Explainability, Human Control and Bounded Intelligence

THE EXPERIENCE MAY ADAPT. THE SOURCE OF TRUTH DOES NOT.