CODANCE PUBLICATION

# How Aurora Works

A plain-language account of controlled execution, observation and evidential reconstruction

| Control | Value |
|---|---|
| Document | COD-PUB-AUR-001 v0.1 \| 25 August 2026 |
| Owner | Codance Ltd |
| Platform | Project Aurora working alpha |
| Purpose | Public system overview |
| Status | First substantive draft for competent review |
| Review discipline | Distinguish demonstrated evidence, engineering proposition and programme intent |

**OPERATING IDEA**

Aurora turns an assurance method into a governed evidence-producing campaign: it controls what may enter, records how it is processed, observes declared consequences and preserves enough lineage for a competent reviewer to work backwards from a report to the material evidence.

## What Aurora is—and is not

Project Aurora is Codance's working alpha platform for modelling and operating assurance methods across AI-enabled systems. It is not a model monitor, a generic log collector, a replacement for system testing, or an automated authority that declares a system trustworthy. Its purpose is to coordinate bounded assurance work and preserve the identities, events, transformations, products and limitations needed to support a finding that is open to challenge.

Aurora is designed around a practical reality: the evidence required to answer an assurance question is distributed. Some comes from the System Under Test, some from independent sensors, some from deterministic tests, some from probabilistic analysis, and some from the assurance procedure itself. Aurora keeps those roles distinct while joining them through governed identity and lineage.

**PLAIN-LANGUAGE SUMMARY**

The platform does not inspect an AI's thoughts. It records the externally material pathway by which a probabilistic system affects human and technical systems, and the pathway by which an assurance procedure reaches its report.

## 1. Five responsibilities

| Responsibility | What it establishes |
|---|---|
| Commissioning | Whether the assurance question can be supported by the SUT's actual observable surfaces. |
| Governed admission | Whether a delivery is authorised, attributable and permitted to enter a campaign route. |
| Controlled processing | Which published sequence of versioned work operated and what each step produced. |
| Observation and retention | What material SUT events, actions and consequences were preserved at declared collection points. |
| Reporting and challenge | How the finding relates to source evidence, method influences, limitations and human authority. |

These responsibilities are separated so that no single component can quietly invent campaign identity, alter historical work or convert missing evidence into apparent confidence.

## 2. Roles and authority

A Campaign describes the governed assurance engagement. A Programme groups related method work. An Execution identifies one controlled application of a configuration and method. The System Under Test remains the article being examined; Aurora does not become part of the SUT merely because it observes or stimulates declared interfaces.

| Role | Authority and boundary |
|---|---|
| Campaign owner / method owner | Defines the question, authorised method, acceptance meanings and legitimate finding authority. |
| SUT owner or developer | Provides controlled article identity, documentation, interfaces and authorised changes. |
| Campaign Register | Maintains independent identities, placements, admission rules and execution attribution. |
| Sensor / adapter / actuator | Observes or stimulates only a commissioned interface and emits a governed transmission. |
| Aurora Core | Processes canonical messages through published routes without being told campaign UUIDs in SUT content. |
| Competent reviewer | Interprets whether retained evidence supports the bounded finding and its use. |

**HUMAN AUTHORITY**

Aurora may apply rules and generate assessments, but it cannot supply the legitimate authority that belongs to a competent person or institution.

## 3. From delivery to canonical evidence

1. A registered transmitter attempts a governed delivery under a Transmission Definition.
2. The Input Router resolves the declared actor, space and channel address to one active iDevice and entry stage.
3. Campaign Register checks ownership, execution, collection point, gate state and permitted transmission meaning.
4. If admitted, Core creates the canonical inbound message identity. Before this point the object is a Transmission; after persistence it becomes a Signal in Core.
5. The published Processing Route creates a Processing Episode and the required work records.
6. Versioned skills consume declared prior products, perform bounded work and emit typed products or attributable failures.
7. Terminal outputs close the sequence with is_final=true, allowing output and report lineage to be reconstructed.

A pre-Core refusal is retained outside Core as a Transmission Failure with a failure-ledger identity. A post-ingress processing defect belongs to the Processing Episode and must reach a typed terminal disposition. This distinction prevents failed admission from being misrepresented as processed evidence.

## 4. Processing routes, stages and skills

An iDevice is the logical entry point selected for a class of work. It resolves to an entry stage. A stage contains ordered or dependency-linked steps and may include sub-stages. Each step is bound to an installed skill package and declares what it requires and produces. The published route therefore acts as an executable representation of the assurance method rather than an informal prompt chain.

Each input initiates one Processing Episode. Work-in-progress records carry the message, route identity, dependencies and earlier products required by later steps. A step can use deterministic algorithms, controlled prompt guides, retained memory, retrieval or an authorised model call. Those capabilities do not erase provenance: inputs, outputs, skill identity and material configuration remain attributable.

| Object | Function | Evidence value |
|---|---|---|
| iDevice | Selects the commissioned entry route. | Shows why this processing path began. |
| Stage | Groups a meaningful phase of the method. | Preserves method structure and authorised sequence. |
| Step | Defines one bounded unit of work. | Identifies dependencies, product and outcome. |
| Skill | Implements the step on a device. | Provides versioned executable behaviour. |
| Product | Retains a typed result or evidence object. | Allows later work and reports to cite exact material. |
| Processing Episode | Joins all work for one input. | Supports reconstruction of completion, failure and outputs. |

## 5. Observation and provenance

Aurora does not collect every available datum. Commissioning maps evidence required by the assurance method to actual SUT surfaces. A Collection Point states where an observation has evidential meaning. A registered sensor instance watches that surface, validates source records and emits a governed transmission with source event identity, timestamps, configuration hash and acquisition details.

Independent placement matters. A hostile or defective SUT should not be able to declare that its own behaviour was properly observed. Sensor configuration is locked, execution identity is bound, and the Register records which instrument was placed at which collection point for which execution. Hash-linked operational journals and offsets allow later checking of order and continuity.

**EVIDENCE MINIMISATION**

Collect the minimum evidence needed to execute, reconstruct and challenge the commissioned method. More data is not automatically better evidence.

## 6. Deterministic and probabilistic work

Aurora permits both deterministic and probabilistic methods because assurance work may require exact validation and contextual interpretation. The two are not interchangeable. Identity, schema, hash, ordering, boundary, admission and policy constraints remain deterministic. A model may summarise, classify, compare or propose an interpretation only where the method authorises that role and its inputs, guide, model identity and outputs are retained.

| Deterministic control | Probabilistic contribution |
|---|---|
| UUID, schema and type validation | Contextual classification or synthesis |
| Hash and lineage verification | Natural-language comparison or explanation |
| Authorisation and gate state | Interpretive recommendation within a declared guide |
| Exact policy thresholds | Assessment where evidence does not reduce to one rule |
| Idempotency and duplicate handling | Candidate hypothesis for competent review |

The system does not convert model confidence into authority. A probabilistic product remains a product with attributable inputs and limitations, available to later deterministic tests and human review.

## 7. Reconstruction and reporting

Forward operation begins with a governed input and produces work, products and outputs. Assurance review often begins in reverse: a reader starts from a finding, follows its report hash and selected evidence index, identifies the relevant execution and root transmissions, then examines source events, acquisition records and the procedure that transformed them.

- The report identifies the question, selected controlled articles and conclusion.
- Lineage identifies the execution, root transmission and source evidence.
- Product hashes identify the exact retained snapshots and assessments.
- Work records show which steps completed, failed or were skipped and what they consumed.
- Collection-point and instrument identities show where and how observations were acquired.
- Limitations and deviations show where the finding must not be extended.

**RECONSTRUCTION TEST**

A report is not evidentially adequate merely because it contains hashes. A competent reviewer must be able to use the retained identities and products to understand the material route from source to conclusion.

## 8. Demonstrated example and present boundary

In the ClearLedger pair demonstration Aurora processed two completed execution reports, verified that they belonged to the same campaign contract and bounded fixture, gathered their registered evidence, checked controlled invariants, isolated one configuration difference, found the first material divergence and emitted a pair assessment and human-readable report. Both request and comparison routes completed all three published steps.

This demonstrates integrated operation of governed admission, versioned DAG processing, sensor evidence, execution reporting, human pair selection and comparative report generation. It does not establish universal observability,

accreditation, production-scale resilience or the validity of an arbitrary assurance method. Those are explicit development and external-challenge tasks.

## Document control and review

This is a first substantive public draft prepared for structured review. Before release, Codance will verify technical identities and claims against the controlled evidence, resolve reviewer comments, assign the approved publication version, regenerate the PDF and record the final file SHA-256.

| Review question | Required disposition |
|---|---|
| Is every statement clearly classed as demonstrated fact, engineering proposition or programme intent? | Confirm, narrow, source or remove. |
| Can a competent reader identify the controlled boundary and prohibited over-claims? | Make the boundary explicit. |
| Do technical identities and results match retained evidence? | Verify against controlled records. |
| Does the document disclose material limitations and credible contrary interpretations? | Add or strengthen limitations. |
| Is the document useful without access to confidential implementation detail? | Improve explanation or supporting public evidence. |