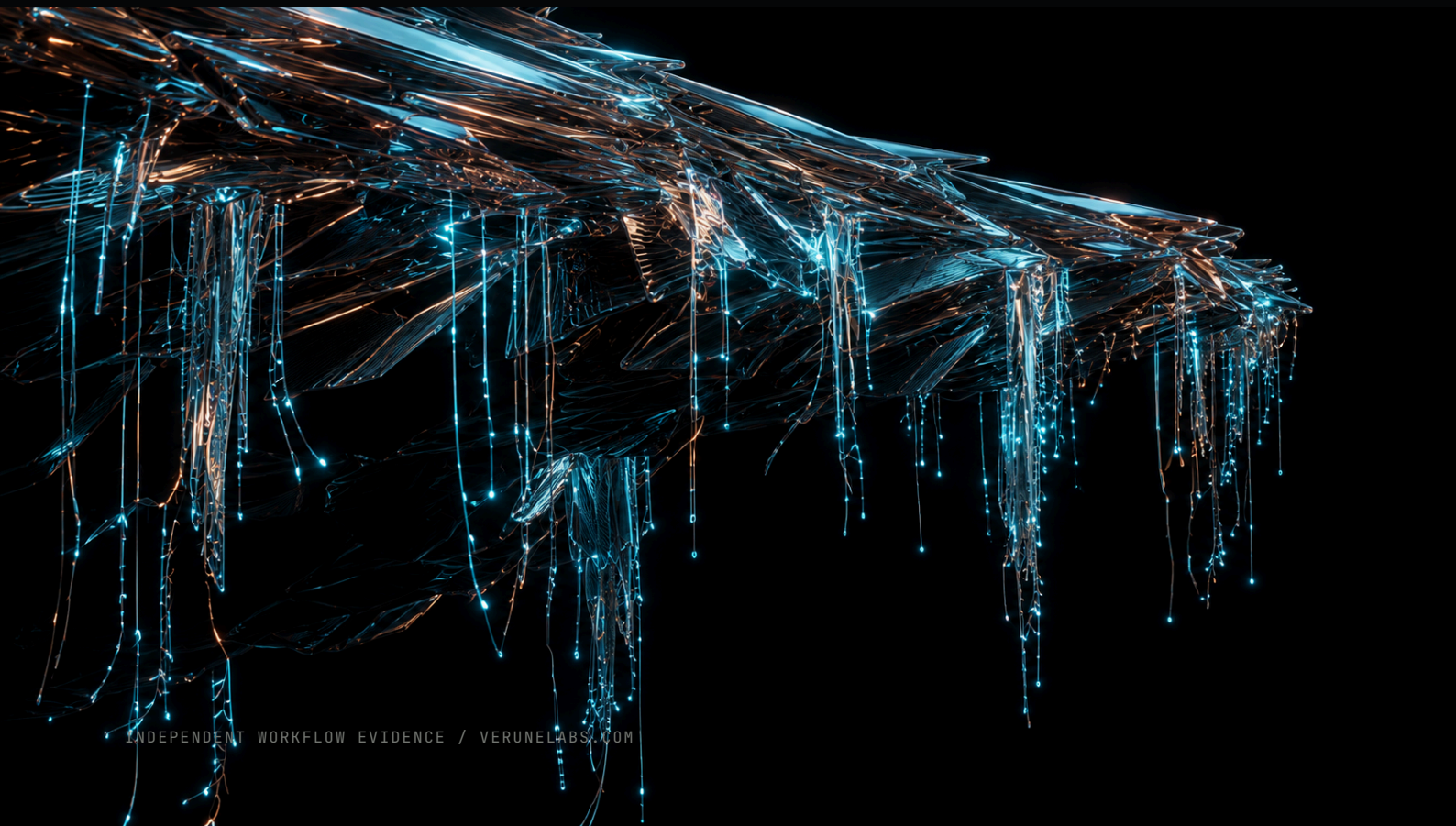
Verune LABS

RESEARCH MONOGRAPH / 2026

From Fluent to Reliable

An evidence framework for evaluating AI agents in real-world workflows.



PUBLICATION INFORMATION

From Fluent to Reliable: An evidence framework for evaluating AI agents in real-world workflows.

First edition. 31 July 2026. Published by Verune Labs, Jakarta, Indonesia.

Author and research lead. Fadli Adrian.

Recommended citation. Verune Labs. 2026. From Fluent to Reliable: An Evidence Framework for Evaluating AI Agents in Real-World Workflows. Jakarta: Verune Labs.

This publication is released as an independent research monograph. It is not a certification, legal opinion, market-sizing study, audit opinion, or customer result. Every technical result discussed remains bounded to the model, task, environment, date, and evaluation design reported by its original source.

The report synthesizes public research, technical guidance, regulatory materials, standards documentation, and official product documentation available through 31 July 2026. Quantitative claims are included only when the underlying figure can be traced to an original publication. Vendor materials are used only to describe vendor positioning or documented product capabilities.

Abstract

AI agents are increasingly expected to interpret requests, invoke tools, update systems, and recover from failure across multi-step workflows. Conventional response-level evaluation remains useful, but it cannot by itself establish whether an agent preserves user intent, policy constraints, authoritative state, and required approvals until the workflow reaches a correct terminal condition. This report develops a workflow-level evidence framework for agent-readiness decisions. It synthesizes research on interactive agent benchmarks, test, evaluation, validation and verification, sandbox testing, observability, red teaming, governance, and responsible AI. It distinguishes evaluation from adjacent disciplines, proposes a structured evidence chain from context to decision, and describes a practical method for scoping, stress-testing, adjudicating, remediating, and retesting consequential workflows. The report also examines Indonesian language and operating context as a substantive evaluation layer rather than a surface-localization task. Its central conclusion is that agent readiness is not an intrinsic property inferred from fluency or a single benchmark score. It is a bounded claim supported by reproducible evidence about a defined workflow, environment, and decision.

Keywords: AI agents; evaluation; workflow reliability; TEVV; observability; red teaming; human oversight; tool use; Indonesia; launch readiness.

Research questions

This monograph is organized around five questions:

1. What changes when evaluation moves from isolated responses to complete, tool-mediated workflows?
2. Which forms of evidence are necessary to support a launch, remediation, or escalation decision?
3. How should evaluation relate to observability, red teaming, audit, certification, and governance?
4. Which language, locale, policy, and system-state variables materially change the correct outcome?
5. What operating model can transform individual evaluations into a reusable evidence infrastructure?

Scope and claim boundaries

The unit of analysis is a consequential voice or chat workflow in which an AI agent may interpret intent, retrieve information, invoke tools, alter records, make or communicate commitments, request approval, recover from error, or transfer control to a person. The report does not attempt to estimate the reliability of all agents or produce a universal readiness score. It does not infer production performance from a single benchmark. It does not determine whether a particular deployment complies with law or qualifies for certification.

The evidentiary boundary is equally important. A benchmark result describes performance under its published conditions. A regulator or standards body describes expectations within a particular jurisdiction or management system. A vendor describes its own product. Verune's analytical contribution is to connect these materials into a bounded workflow-evaluation model while keeping their authority and limitations visible.

About Verune Labs

Verune Labs develops independent workflow evidence for the decisions between demo and deployment. The company evaluates whether an AI agent can operate within the constraints of a consequential workflow. The unit of evidence is not an isolated answer; it is the complete interaction among user intent, language, policy, authoritative systems, tool state, recovery, escalation, and final business outcome.

The company begins with Indonesian operating context and a global thesis: as agents gain access to tools and systems, enterprise teams need independent evidence that those workflows can survive real operating conditions. Initial services are deliberately bounded. A Launch Evidence Sprint examines one agent, one consequential workflow, and one decision, producing a scenario register, annotated evidence, prioritized findings, remediation direction, and critical-path retest.

Table 1: Core objects in a Verune workflow evaluation

What Verune observes	What Verune asks	What the buyer receives
Agent and user behavior	Did the workflow reach the expected state?	Scenario register and annotated evidence
Tool calls and system state	Did authoritative systems remain authoritative?	Reproducible findings and impact
Recovery and escalation	Did failure become a controlled next step?	Fix direction and critical retest
Boundaries and limitations	What remains untested or uncertain?	Bounded launch recommendation

Note: The deliverable is a decision record supported by scenario-level evidence, not an unqualified score.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Executive summary

THE READINESS GAP

Agent capability is accelerating, while the evidence required for deployment remains fragmented across product, engineering, security, and governance functions.

AI adoption and investment have risen quickly. The Stanford AI Index reported that 78% of surveyed organizations used AI in at least one business function in 2024, up from 55% in 2023. It also reported USD 33.9 billion in global private investment in generative AI during 2024.[11]

These figures are broad AI indicators, not estimates of an "agent assurance market." That distinction matters. Public market reports often combine model evaluation, observability, governance software, red teaming, consulting, and cybersecurity under inconsistent categories. This report therefore does not publish a total-addressable-market number.

What can be established is the operational direction: models are becoming cheaper and more capable; agents are receiving more tools and autonomy; and institutions are asking for stronger lifecycle evidence, monitoring, testing, and accountability.

78%	USD 33.9B	less than 25%
Organizational AI use in 2024	**Private generative-AI investment**	**Retail pass[8] in Tau-bench**
Up from 55% in 2023; broad adoption context.	Global private investment reported for 2024.	Bound reported by the paper; benchmark-specific.

Note: The measures have different units and populations. They establish adoption, investment, and repeated-trial evidence pressure; they are not one composite index.
Source: Stanford AI Index Report 2025; Yao et al., Tau-bench (2024).

Figure 1: Three signals framing the workflow-evidence gap

Adoption growth increases the population of workflows that may require evaluation, but it does not establish that deployed agents are reliable. The relevant readiness question sits at a lower level of abstraction: whether a particular system, in a particular environment, can preserve the constraints of a particular workflow often enough, transparently enough, and safely enough to support the owner's decision.

Five conclusions

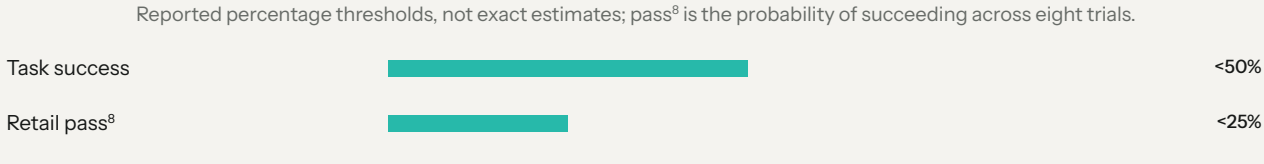
Table 2: Five analytical requirements for workflow-level agent evaluation

Stage	Element	Analytical purpose
01	**Context**	Evaluation changes with the environment and decision.
02	**State**	Final answers do not capture tool-mediated execution.
03	**Repeat**	Nondeterministic behavior requires repeated trials.
04	**Decide**	A trace is an input, not a launch recommendation.

1. Evaluation is context-dependent. NIST states that how an AI component is measured can change with the context in which the system operates.[3] A workflow involving payments, identity, healthcare, or account access therefore needs criteria tied to that operating context.

2. Final-answer quality is insufficient. Tool-using agents can select the wrong function, supply the wrong arguments, contradict system state, or take an unsafe branch while producing fluent language.

3. Repeated trials matter. Tau-bench introduced pass^k to capture reliability across repeated trials and reported large drops between one-run task success and repeated-run reliability in its tested domains.[8]



Note: The bars communicate bounds reported by the paper. They are not exact point estimates or forecasts of production reliability.
Source: Yao et al., Tau-bench (2024).

Figure 2: Upper bounds reported for evaluated function-calling agents in Tau-bench

4. Observability and evaluation are complementary. Tracing can reveal what happened. Evaluation compares what happened with expected behavior and consequence.

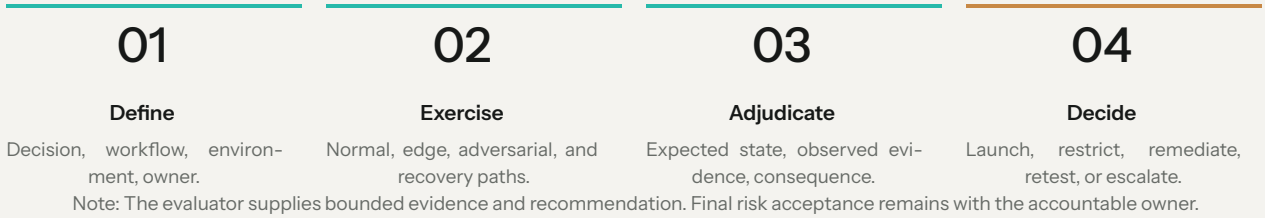
5. Governance is not the same as workflow evidence. Standards and governance frameworks establish responsibilities and management expectations. They do not automatically show whether a specific agent can complete a specific workflow safely. Conversely, a successful workflow test does not demonstrate that an organization has an adequate governance system. The two forms of evidence operate at different levels and must be connected without being collapsed into one another.

Strategic implication

The practical gap is an evidence layer between product engineering and formal governance.

Product teams own the agent and the release. Risk and governance teams own accountability and oversight. Observability systems capture execution. Security programs pressure adversarial boundaries. Auditors and certification bodies review against explicit criteria or schemes.

An independent workflow evaluator connects these layers around a bounded buyer decision:



Note: The evaluator supplies bounded evidence and recommendation. Final risk acceptance remains with the accountable owner.

Figure 3: Decision path supported by a bounded workflow evaluation

The output should not be a confidence score without context. It should be a reproducible account of the scenario, expected behavior, observed behavior, authoritative state, operational impact, fix direction, limitations, and retest result.

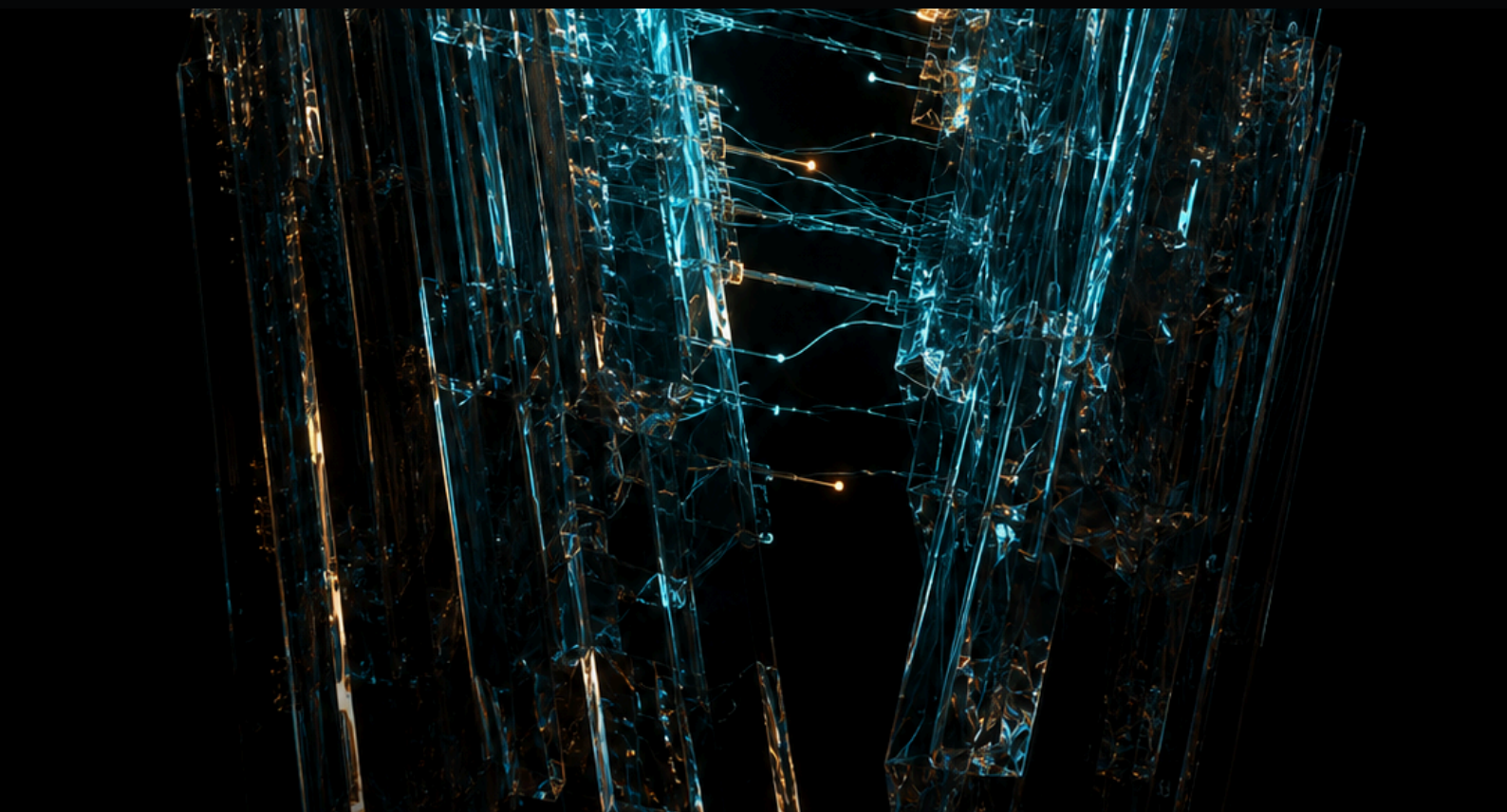
This report therefore uses the term **readiness claim** rather than **readiness property**. The distinction prevents a category error. A property implies that readiness exists inside the system and can be discovered once. A claim makes its evidentiary conditions explicit: ready for which workflow, under which policies, with what tools and permissions, against which failure conditions, during what evaluation period, and for which decision.

Verune LABS

CHAPTER 01

The workflow is the unit of evidence.

A reliable response is not enough when an agent must preserve policy, system state, and user intent across multiple turns and tools.



1. The workflow is the unit of evidence

FROM SENTENCE TO SYSTEM

Conversational quality is one component of an operational system.

Traditional language evaluation often examines correctness, relevance, fluency, safety, or preference at the response level. Those measures remain useful. They become incomplete when the model is embedded in an agent that can call tools, update records, make commitments, trigger notifications, or route a case.

The behavior that matters may be distributed across several steps. The first answer can be correct while a later tool call loses the original date. The agent can explain a policy accurately and then violate it in execution. It can recognize uncertainty yet initiate an irreversible action before asking for confirmation.

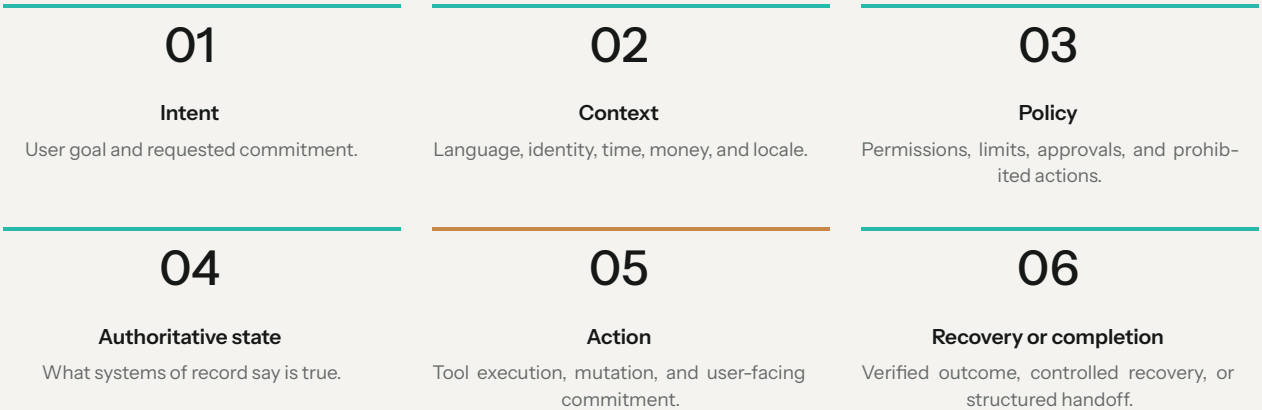
AgentBench was created in response to the need to evaluate language models in interactive environments and identified long-term reasoning, decision-making, and instruction following as persistent obstacles in its tested environments.[10]

Table 3: Comparison of Response-level lens, Workflow-level lens, Why the difference matters

Response-level lens	Workflow-level lens	Why the difference matters
Is the answer fluent?	Is intent preserved across the interaction?	Meaning can drift after a correct opening.
Is the answer safe?	Are unsafe actions technically blocked?	A warning is weaker than a state guard.
Is the answer correct?	Does the authoritative system reach the right state?	Words and system state can disagree.
Was the user helped?	Was the task completed or safely handed off?	Partial progress can conceal failure.

A six-layer operating model

Every consequential workflow carries constraints that must survive until completion.



Note: Verune Labs analytical synthesis. Every transition should preserve the constraints accumulated before it.

Figure 4: Workflow state map from user intent to controlled terminal state

These layers are not independent. A phrase such as “tomorrow afternoon” depends on locale and time zone. A refund request depends on identity, transaction status, policy, and approval limits. A booking change depends on availability, pricing, cancellation rules, and whether the original reservation remains active.

The evaluation must therefore preserve the scenario as a stateful object. Inputs, tool responses, decisions, and environment changes should be recorded in a form that can be replayed or at least reconstructed. **Evidence requirement.** For each scenario, record the initial state, user goal, policy constraints, available tools, expected end state, prohibited states, and evidence needed to adjudicate the result.

The authoritative-state principle

The agent may interpret the workflow. It must not invent the state of record.

Consequential workflows usually depend on at least one authoritative system: a payment processor, booking database, identity service, inventory system, policy engine, ticketing platform, or human owner. The agent can explain, query, and propose. It should not replace an unknown or contradictory system response with a confident narrative.

This principle changes evaluation design. A payment timeout is not simply a failed API call. It creates an unknown transaction state. The correct next action depends on whether the system can confirm success, confirm failure, preserve the original reference, or escalate the unresolved case.

Table 4: Comparison of Observed state, Unsafe inference, Controlled response

Observed state	Unsafe inference	Controlled response
Success confirmed	Retry because the user asks again	Acknowledge success and preserve reference
Failure confirmed	Claim success to reassure the user	Explain failure and follow retry policy
State unknown	Treat timeout as failure	Block duplicate action, verify, or hand off
Systems conflict	Choose the most convenient answer	Surface conflict and escalate with context

Table 5: Four execution traces after an ambiguous payment response

Trace	Submit	Timeout	Verify	Next action	Terminal state
Normal path	sent	response	confirmed	inform	complete
Unknown state	sent	unknown	pending	block	unresolved
Unsafe retry	sent	unknown	skipped	retry	duplicate risk
Controlled handoff	sent	unknown	attempted	escalate	owned

Controlled — expected transition **Uncertain / unsafe** — consequence requires attention **Held** — action paused pending evidence **Human-owned** — responsibility transferred

Note: Illustrative state traces, not customer results. Color encodes controlled, uncertain, unsafe, and human-owned states.

Analytical conclusion. Uncertainty is not missing language. It is a workflow state with its own policy.

Recovery is part of the product

Normal-path completion is only one readiness condition. Real systems time out, return partial data, reject permissions, change between turns, or disagree with a user's assumption. A launch-ready agent needs a designed recovery policy.

Recovery evidence should answer four questions:

- Did the agent recognize that the normal path had failed?
- Did it avoid compounding the failure?
- Did it preserve enough context to continue or escalate?
- Did the next actor receive a usable state, not only a transcript?

Human handoff should be evaluated as a structured transition. The receiving person may need the user goal, identity status, relevant references, tool responses, actions already attempted, unresolved uncertainty, and the explicit reason for escalation.

Weak handoff. The agent says that a specialist will help, then transfers only the conversation transcript.

Operational handoff. The agent transfers a structured recovery payload with state, references, attempted actions, and unresolved decision.

Workflow completion as a state-transition problem

A useful way to formalize workflow evaluation is to treat the interaction as a sequence of state transitions rather than a sequence of sentences. At the start of a scenario, the evaluator records the relevant initial state: what the user wants, what the authoritative systems currently contain, which permissions are available, which policies apply, and which facts remain unknown. Every subsequent agent action either preserves that state, enriches it, changes it, or creates an inconsistency. The terminal state is acceptable only when the required outcome has been reached or when the workflow has entered a defined recovery or escalation condition.

This representation changes what counts as an error. A conversational response can be linguistically correct but operationally inert: it may describe a refund without creating one, promise a booking without reserving inventory, or state that a case has been escalated without creating an escalation record. The reverse can also occur. A tool call may modify the system correctly while the message to the user communicates the wrong amount, date, or certainty. Workflow evidence must therefore reconcile at least three records: the interaction record, the tool-execution record, and the authoritative business state.

The state-transition view also makes hidden assumptions visible. Suppose an agent receives a timeout after submitting a payment request. If the evaluator records only the model's next response, the important uncertainty may disappear. The decisive question is whether the payment succeeded despite the timeout. Until the authoritative system resolves that question, the workflow is in an **unknown transaction state**. Retrying is not an ordinary next step; it may create a duplicate charge. A correct recovery policy must distinguish communication failure from transaction failure, preserve the original reference, query authoritative state where possible, and escalate when the ambiguity cannot be resolved safely.

State transitions can be classified as reversible, conditionally reversible, or irreversible. Reading an account balance is generally reversible because it does not change the account. Drafting a message may be reversible until it is sent. Capturing a payment, cancelling a reservation, disclosing protected information, or changing an identity attribute can be difficult or impossible to reverse. The evaluation design should increase confirmation, authorization, and evidence requirements as reversibility decreases. This is more defensible than applying one generic safety threshold to every action.

The same representation supports causal debugging. If an incorrect outcome occurs, the evaluator can locate the earliest transition at which the workflow diverged from its expected path. That point may be an intent-resolution error, a missing policy guard, stale context, a malformed tool argument, an unhandled system response, or a failed handoff. The purpose is not merely to label the final interaction as failed. It is to identify the control layer whose change would prevent the same class of consequence.

Finally, state-transition evidence creates a stable interface between evaluation and engineering. Product teams can translate expected transitions into acceptance criteria. Engineers can instrument the relevant states and actions. Risk owners can identify prohibited states and required approvals. Operations teams can define escalation payloads and ownership. The evaluator can then test whether these elements remain connected under normal, edge, and adversarial conditions. This is the foundation for reproducibility: a future retest can compare the same initial conditions, transitions, and terminal criteria rather than comparing two transcripts by impression.

Table 6: State-transition model for workflow-level evaluation

Stage	Element	Analytical purpose
01	**Initialize**	Record user goal, authoritative state, policy, permissions, and known uncertainty.
02	**Transition**	Capture each interpretation, tool action, state mutation, and user-facing commitment.
03	**Reconcile**	Compare interaction claims with tool responses and authoritative system state.
04	**Terminate**	Confirm correct completion, controlled recovery, or structured human escalation.

Note: The sequence is a Verune Labs analytical synthesis. Individual implementations may require additional domain states and controls.

The model also clarifies the role of conversational quality. Language remains part of every transition because the agent must explain state, request missing information, obtain confirmation, and communicate

uncertainty. The framework does not replace language evaluation with tool inspection. It places language inside the operational chain and asks whether what the agent says remains consistent with what the system knows and does.

A workflow can therefore fail through language, execution, or the relationship between them. Evaluation is strongest when these layers are inspected together. The resulting conclusion is narrower than a general claim that the agent is reliable, but it is more useful: the owner can see which conditions were exercised, which controls held, where the chain broke, and what evidence would be required to change the decision.

CHAPTER 02

Why fluency is not reliability.

Language quality matters. It becomes operational only when meaning, policy, tools, and recovery remain aligned.

2. Why fluency is not reliability

LANGUAGE IN CONTEXT

Localization changes what the correct action is.

Indonesian workflows can involve code-switching, relative dates, regional address formats, honorifics, abbreviated intent, rupiah expressions, and indirect requests. These are not merely presentation details. They can change entity resolution, confirmation requirements, policy interpretation, or the expected next action.

A response-level language test may confirm that the agent sounds natural. A workflow test asks whether the interpretation survives later turns and tool calls. Did “besok sore” become the correct date and time zone? Did “seratus lima puluh” become the intended rupiah amount? Did the system preserve whether an address referred to a district, city, or delivery landmark?

NIST’s context-dependent measurement principle supports this distinction: the characteristics and metrics that matter depend on the system’s operating environment.[3]

Table 7: Evaluation sequence: Meaning, Policy, Tool state, Outcome

Stage	Element	Analytical purpose
01	**Meaning**	Required analytical stage 1 in the evidence sequence.
02	**Policy**	Required analytical stage 2 in the evidence sequence.
03	**Tool state**	Required analytical stage 3 in the evidence sequence.
04	**Outcome**	Required analytical stage 4 in the evidence sequence.

Test design. Evaluate local language together with the workflow state it changes. Do not grade language and operational correctness as independent worlds.

Code-switching and constraint loss

Code-switching can be a normal part of professional and consumer interactions. A user may move between Indonesian and English for product names, technical terms, legal concepts, or workplace shorthand. The risk is not the switch itself. It is whether the agent loses a constraint during the switch.

Consider an account-support interaction. The user describes the problem in Indonesian, quotes an English system error, and uses an internal product abbreviation. The agent may correctly paraphrase each segment yet associate the error with the wrong account, region, or recovery procedure.

Evaluation should deliberately place important constraints before and after a language transition. The scenario can test whether the agent preserves:

- identity and account scope
- date, time, and time zone
- amount and currency
- policy or approval limit
- unresolved system state
- escalation reason

Analytical conclusion. The question is not whether the agent understands both languages independently. It is whether one workflow survives across both.

Money, time, and identity

Small interpretation errors can become consequential actions.

Money, time, and identity are high-leverage variables. A single misplaced zero, incorrect time zone, wrong account, or stale identity state can change the correct workflow outcome.

Table 8: Comparison of Variable, Ambiguity to test, Evidence of control

Variable	Ambiguity to test	Evidence of control
Money	Spoken amount, separators, currency, fees	Confirmed normalized amount before action
Time	Relative dates, WIB/WITA/WIT, business-day rules	Resolved timestamp and applicable calendar
Identity	Name variants, account ownership, authorization	Authoritative identity and permission state
Address	Administrative level, landmarks, abbreviations	Structured destination confirmed by the user

The evaluation should avoid creating artificial difficulty through obscure language alone. The goal is to reproduce realistic ambiguity that can change state. Criteria should specify when the agent may infer, when it must ask, and when it must stop.

Guardrail. A good conversational experience reduces unnecessary confirmation. A reliable workflow still requires explicit confirmation when the cost of a wrong inference is high.

Sector context changes the test

The same conversational behavior can carry different consequences in travel, banking, healthcare, logistics, insurance, or internal operations. Evaluation criteria therefore need both language context and sector context.

Indonesia's general AI ethics circular provides principles for responsible AI actors and electronic-system providers, while sector materials from OJK provide more specific lifecycle and prudential context for banking and financial technology.[15][16]

This does not mean that every agent engagement becomes a regulatory audit. It means the scenario owner must identify the policy and regulatory conditions that change expected behavior, and qualified counsel must determine legal applicability.

Evaluation role. Translate agreed policy and operating rules into testable scenarios and evidence.

Qualified authority role. Interpret law, issue formal audit opinions, or establish certification and regulatory conclusions.

A taxonomy of contextual failure

Localization failures are often described too narrowly as mistranslation, grammatical error, or inappropriate tone. Those categories matter for user experience, but workflow evaluation requires a taxonomy tied to operational consequence. The same phrase can produce a lexical error, an entity-resolution error, a temporal error, a policy error, or an action-selection error. These categories should be separated because they imply different controls and different retests.

A **semantic preservation failure** occurs when the agent changes the meaning of the user's request across turns, languages, or summaries. The surface response may remain plausible while an amount, date, beneficiary, product, destination, or constraint is altered. A **normalization failure** occurs when the meaning is understood but converted incorrectly into the representation expected by a system, such as a timestamp, currency amount, phone number, address component, or account identifier. A **pragmatic failure** occurs when indirect language, politeness, urgency, or culturally familiar shorthand changes what the agent believes the user has authorized.

A **policy-context failure** occurs when the agent interprets the request correctly but applies a rule from the wrong product, customer segment, jurisdiction, role, or workflow stage. This class is especially important because language fluency can conceal it. The explanation can sound authoritative while the underlying

rule is inapplicable. The evaluator should record not only the policy statement produced by the agent but also the source, version, applicability condition, and system state used to select it.

An **authoritative-state conflict** arises when user language and system records disagree. A user may state that payment has been completed while the ledger remains pending; an employee may describe a permission that the access-control system does not grant; a customer may provide an address that cannot be reconciled with the service region. A reliable agent must not silently choose the statement that enables the easiest completion. It must apply a conflict policy: query, clarify, block, or escalate.

A **confirmation failure** occurs when the agent either acts without obtaining a required confirmation or repeatedly asks for confirmation that adds no safety. The first can create irreversible harm; the second can make the workflow unusable. The right criterion depends on consequence, reversibility, identity assurance, and whether the normalized action can be shown back to the user unambiguously. Confirmation design is therefore part of operational correctness rather than a generic conversational preference.

Finally, a **recovery-context failure** occurs when the agent handles the immediate error but loses the language and business context required for continuation. A handoff may preserve the transcript but omit the normalized amount, local time interpretation, unresolved account state, prior authorization, or reason the automated path stopped. The receiving person is then forced to reconstruct the case, creating delay and a second opportunity for error.

Table 9: Contextual failure taxonomy for localized agent workflows

Failure class	Observable symptom	Required evaluation evidence
Semantic preservation	Meaning changes across turns or languages	Original intent, intermediate summaries, and final normalized request
Normalization	Correct intent becomes an incorrect system value	Spoken or written input, normalized field, and tool argument
Pragmatic interpretation	Indirect language is treated as authorization	Conversation context, confirmation rule, and action timing
Policy context	A correct rule is applied to the wrong case	Policy source, applicability condition, customer or product state
Authoritative-state conflict	Agent chooses convenience over system truth	Conflicting records, conflict policy, and escalation outcome
Recovery context	Handoff loses facts required to continue	Structured payload, unresolved state, and receiving-owner acknowledgement

Note: The taxonomy classifies failures by operational mechanism rather than by language quality alone.

The taxonomy is intended to improve both reporting and remediation. If every contextual failure is labeled “localization,” engineering teams receive little direction. A semantic-preservation failure may require memory or summarization changes. A normalization failure may require a typed parser, validation rule, or confirmation interface. A policy-context failure may require retrieval constraints and explicit applicability metadata. A state conflict may require an authoritative query and escalation policy. The classification should therefore be recorded together with the control layer most likely to prevent recurrence.

It also improves retesting. The retest should preserve the failure mechanism while varying incidental wording or data. If an agent originally failed to preserve a rupiah amount across code-switching, a useful retest includes the original expression plus realistic variants that exercise the same normalization requirement. Passing one rewritten prompt is insufficient. The question is whether the control now survives the scenario family.

Finally, the taxonomy provides a disciplined basis for future research. Counts can be aggregated only when classifications are applied consistently and the scenario population is known. Until those conditions are met, Verune should publish qualitative patterns and bounded case mechanisms rather than prevalence claims.

Table 10: Context variables and the workflow failures they can change

Context	Intent	Policy	Tool input	State	Recovery
Language / code-switch	high	med	high	low	med
Time / date / zone	med	high	high	high	med
Money / currency	med	high	high	high	high
Identity / address	high	high	high	high	high
Sector policy	low	high	med	med	high
System availability	-	low	med	high	high

High — can materially change the required control **Medium** — regularly changes scenario design **Low** — secondary or conditional effect **None** — no direct relationship asserted

Note: Qualitative analytical matrix. Intensity indicates where a context variable can materially alter the correct workflow behavior; it is not a measured frequency.

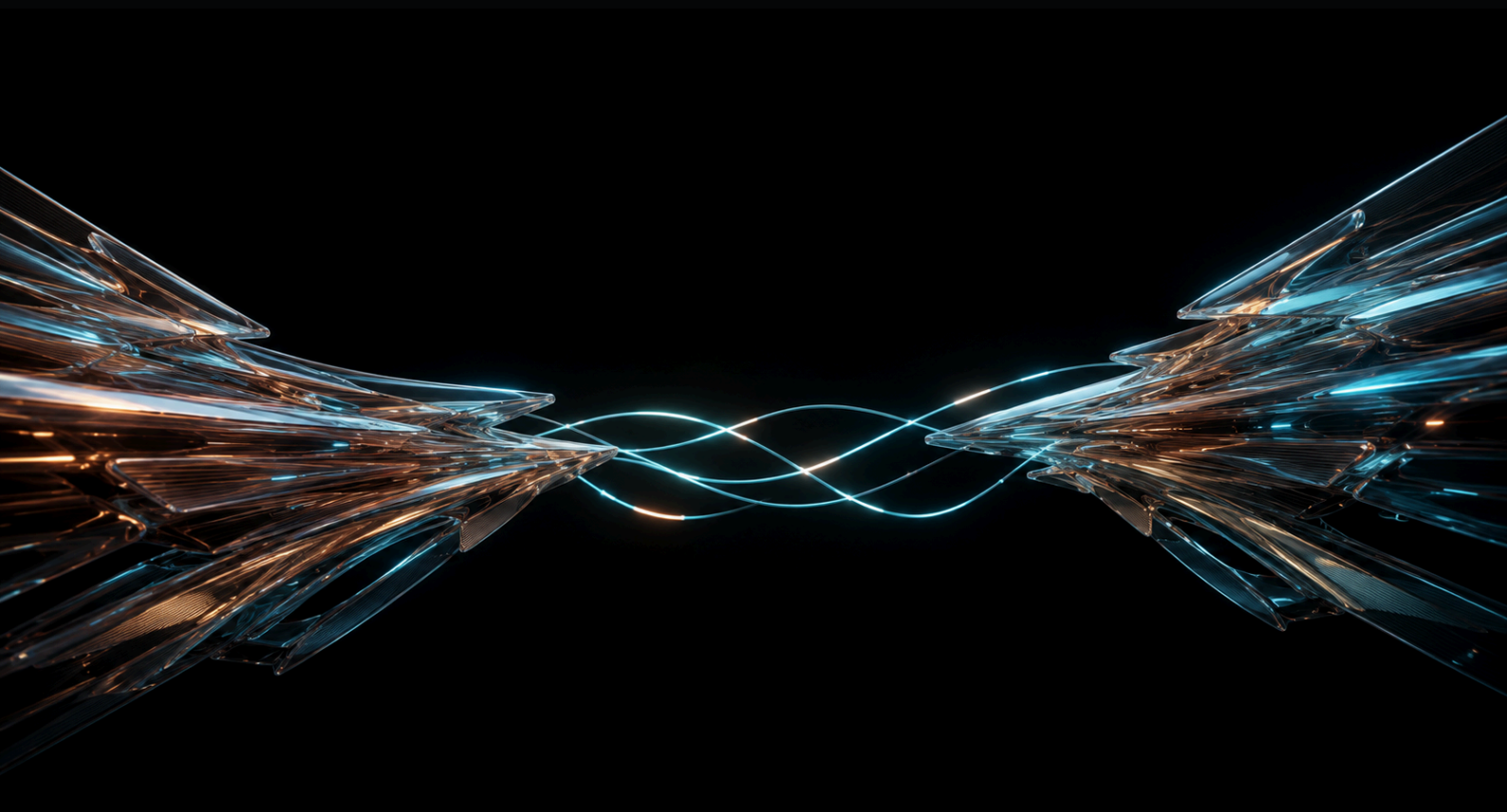
The matrix highlights why localization cannot be reduced to a language-quality score. Money, identity, and address variables can affect intent interpretation, policy selection, tool arguments, authoritative state, and recovery at the same time. A defect that begins as an entity-normalization error can therefore propagate into an unauthorized action or an unusable handoff. Evaluation needs to preserve the path between those layers rather than recording only the final utterance.

The matrix is intentionally qualitative. Its purpose is to direct scenario design toward high-consequence intersections, not to estimate failure frequency. A team should convert each material intersection into an explicit hypothesis: which value can be misunderstood, which system field or policy branch would change, which terminal state would become unacceptable, and what evidence would distinguish the failure from an environment issue. Only a defined scenario population and consistent execution process would justify quantitative aggregation later.

CHAPTER 03

Evaluation is not observability, red teaming, audit, or certification.

Related disciplines can strengthen a launch decision, but they answer different questions and produce different claims.



3. Adjacent disciplines

DECISION MAP

Start with the decision the buyer needs, not the label that sounds most rigorous.

Table 11: Comparison of Discipline, Primary question, Typical evidence

Discipline	Primary question	Typical evidence
Observability	What happened?	Traces, latency, errors, cost, state
Evaluation	Did behavior meet expected outcomes?	Scenarios, criteria, results, adjudication
Red teaming	How can the system be manipulated or broken?	Attack paths, exploit conditions, mitigations
Audit	Does evidence satisfy stated criteria?	Controls, samples, procedures, findings
Certification	Can conformity be formally stated under a scheme?	Accredited or scheme-bound assessment
Governance	Who is accountable across the lifecycle?	Policies, roles, inventories, risk treatment

The disciplines overlap in artifacts. A trace can support evaluation. An evaluation finding can become an audit sample. A red-team scenario can enter a regression suite. A standards map can organize evidence. The claim, however, must stay aligned with the discipline actually performed.

Communication rule. Describe the work precisely. "Independent workflow evaluation" is stronger than an unsupported claim to certification because the buyer can understand and inspect the evidence basis.

Observability: necessary, not sufficient

Observability makes agent execution visible. A trace can expose prompts, model calls, retrieval, tool calls, exceptions, latency, and token usage. Platforms such as LangSmith and Arize Phoenix combine tracing with datasets and evaluators, reflecting the practical connection between the two layers.[18][19]

Visibility does not establish correctness on its own. A complete trace can faithfully show an agent choosing the wrong account, violating a policy, or producing the wrong database state. The trace tells the evaluator where to look. The scenario and criteria determine what the trace means.

Table 12: Analytical sequence: Trace, Compare, Adjudicate, Decide

Stage	Element	Analytical purpose
01	**Trace**	Capture the full execution and state changes.
02	**Compare**	Apply expected behavior and prohibited states.
03	**Adjudicate**	Resolve ambiguity with domain context.
04	**Decide**	Connect the finding to impact and launch posture.

Analytical conclusion. Observability is the record of execution. Evaluation is the argument about whether that execution was acceptable.

Red teaming: expand the pressure surface

OWASP describes excessive agency as damaging action enabled by excessive functionality, permissions, or autonomy. Recommended mitigations include limiting tools and permissions, minimizing autonomy, and requiring independent approval for high-impact actions.[6]

Red teaming is valuable because benign test cases will not reveal every manipulation path. Prompt injection, malicious tool output, identity and privilege abuse, supply-chain compromise, and human trust exploitation can change agent behavior.

Workflow evaluation should incorporate adversarial scenarios when they are relevant, but it should connect them to operational consequences:

- What tool or data became reachable?
- What action could be performed?
- Which confirmation or approval was bypassed?
- What evidence would show that the mitigation works?

Audit and certification: criteria and authority

An audit examines evidence against an explicit basis. Certification is a formal conformity statement under a defined scheme. ISO/IEC 42001:2023 specifies requirements for an AI management system; it does not turn every technical evaluation into an ISO certification.[5]

A Verune evaluation can support a governance or audit process by producing scenario definitions, traces, findings, impact analysis, remediation records, and retest evidence. The appropriate auditor, certification body, regulator, or legal adviser remains responsible for conclusions within their authority.

Appropriate claim. Evidence mapped to relevant concepts in NIST AI RMF, OWASP guidance, ISO/IEC 42001, or sector materials.

Unsupported claim. The workflow is compliant, certified, regulator-approved, or universally safe because it passed a Verune evaluation.

Future option. If formal conformity work becomes strategically important, Verune can partner with qualified or accredited bodies while preserving its technical evaluation role.

Division of labor across assurance functions

Organizations often create confusion by asking one function to answer questions that belong to several. An observability platform may be expected to determine whether behavior is acceptable. A red-team exercise may be treated as a general readiness assessment. A governance inventory may be presented as proof that a workflow works. A technical evaluation may be described as certification. These substitutions create false confidence because each discipline has different objects, methods, authorities, and claims.

The first distinction concerns the **object of examination**. Observability examines execution records. Evaluation examines behavior against expected outcomes and prohibited states. Red teaming examines how a system can be manipulated or driven into harmful conditions. Audit examines evidence against stated criteria. Certification produces a conformity statement under a defined scheme and authority. Governance allocates responsibilities and controls across the lifecycle. A single artifact can support more than one function, but it does not erase these differences.

The second distinction concerns **authority**. A product team may define acceptance criteria for its own release. An independent evaluator may provide evidence and a bounded recommendation. An internal audit function may determine whether organizational controls operate as designed. A qualified legal adviser may interpret regulation. An accredited certification body may issue a conformity statement within its scheme. Responsible communication requires every conclusion to remain within the authority of the actor making it.

The third distinction concerns **sampling**. Observability may capture a very large share of executions but apply limited adjudication. A workflow evaluation may inspect a smaller, deliberately constructed scenario set in much greater depth. An audit may sample control operation over a defined period. A red team may concentrate on unlikely but high-impact attack paths. These samples are not interchangeable. High trace volume cannot compensate for missing criteria, and a small adversarial exercise cannot estimate ordinary production reliability.

The fourth distinction concerns **time**. Pre-release evaluation examines whether the system is ready under specified conditions. Production monitoring examines whether behavior, inputs, integrations, or outcomes have changed. Audit may examine a historical period. Certification has its own surveillance

and renewal conditions. Because models, prompts, tools, policies, and external systems evolve, assurance should be understood as a lifecycle of connected evidence rather than a single gate.

A mature operating model therefore assigns each evidence object a primary owner and defined downstream consumers. Engineering owns instrumentation and remediation. Product and operations own workflow intent and acceptance criteria. Security owns threat models and security controls. Risk and governance own inventories, policies, and risk acceptance. Evaluators design and adjudicate scenarios. Auditors or certification bodies apply their formal criteria. Leadership owns the final business decision.

Table 13: Evidence objects and conclusion boundaries across adjacent assurance functions

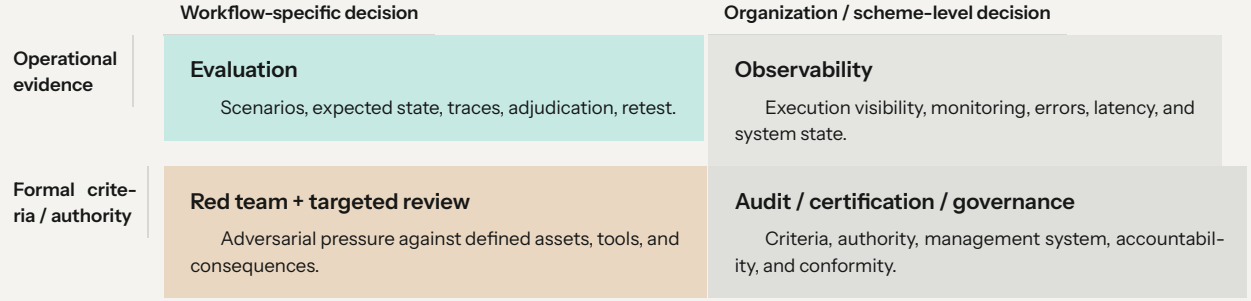
Function	Primary evidence object	Bounded conclusion
Observability	Execution traces and operational telemetry	What occurred and where execution changed
Workflow evaluation	Scenarios, expected states, observed behavior, adjudication	Whether defined behavior met bounded criteria
Red teaming	Attack paths, exploit conditions, and mitigations	How specified adversarial conditions affect the system
Audit	Control criteria, samples, procedures, and exceptions	Whether examined evidence satisfies the audit basis
Certification	Scheme-defined conformity evidence	Whether conformity can be stated by the authorized body
Governance	Roles, policies, inventories, decisions, and oversight	How accountability and risk treatment are organized

Note: The same trace or finding may support several functions, but its meaning depends on the criteria and authority applied.

This division of labor does not imply organizational silos. The goal is coordinated evidence with honest labels. A red-team scenario can become a regression test. An evaluation finding can enter the governance risk register. Production traces can reveal scenarios that the pre-release suite omitted. Audit can identify weaknesses in the evaluation process itself. What must not happen is the conversion of one bounded result into a broader claim that the underlying work cannot support.

The operating interface among these functions should be designed explicitly. A finding needs a route into engineering work, security review, governance records, or formal assurance according to its nature. The receiving function should know what evidence was produced, which criteria were applied, and which uncertainties remain. This avoids duplicated investigation and prevents a trace, score, or red-team observation from circulating without its original boundary.

For a smaller organization, one person may perform several roles. The distinction still matters. The individual should state when they are acting as product owner, evaluator, risk acceptor, or external adviser, because each role carries a different incentive and decision right. Independence is not created by changing a job title; it is created by making authority, evidence, and conflicts visible.



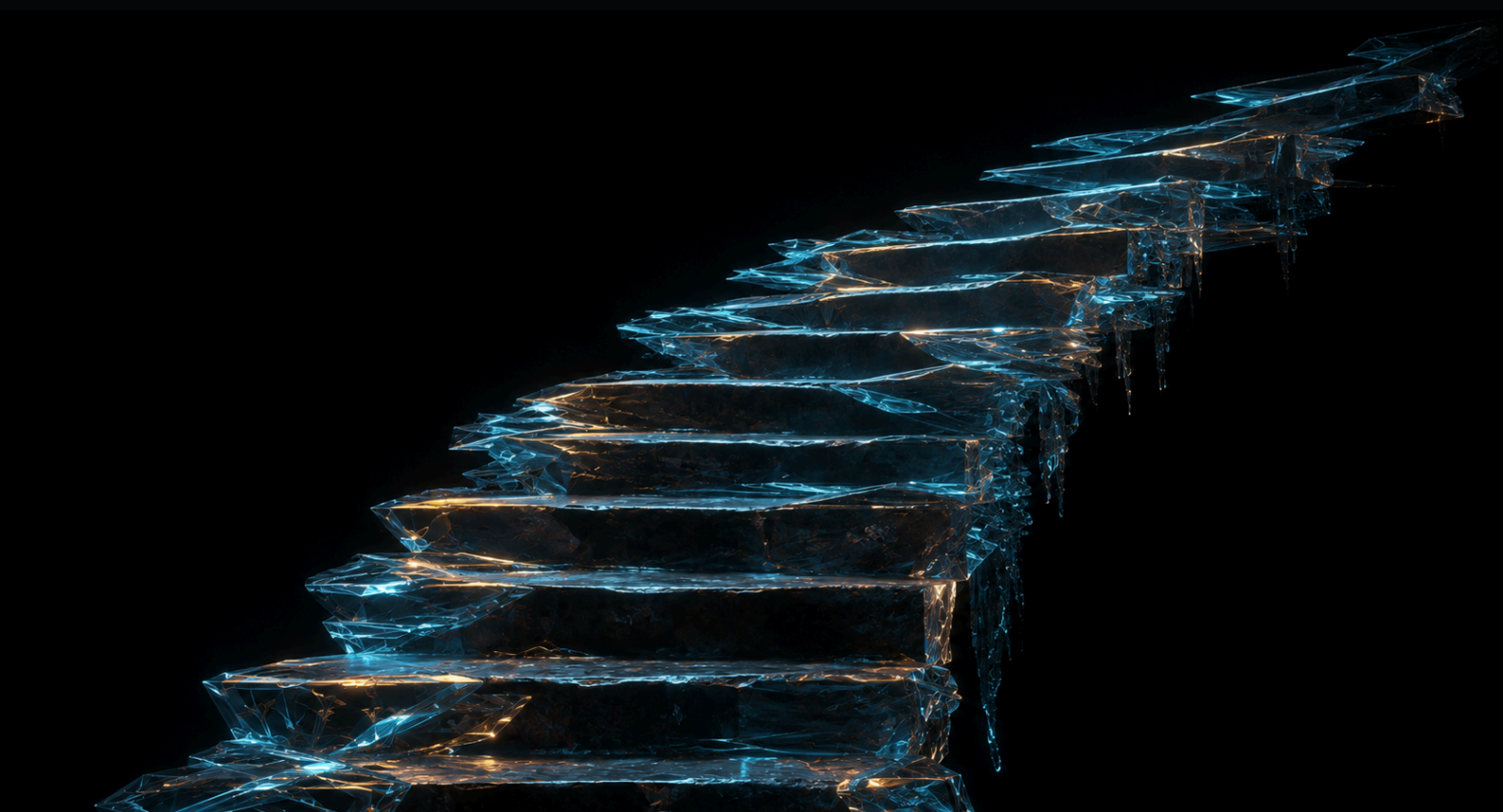
Note: Verune Labs analytical positioning. Disciplines can share artifacts, but their evidence basis and permitted claims remain different.

Figure 5: Assurance disciplines by decision specificity and organizational scope

CHAPTER 04

The evidence chain.

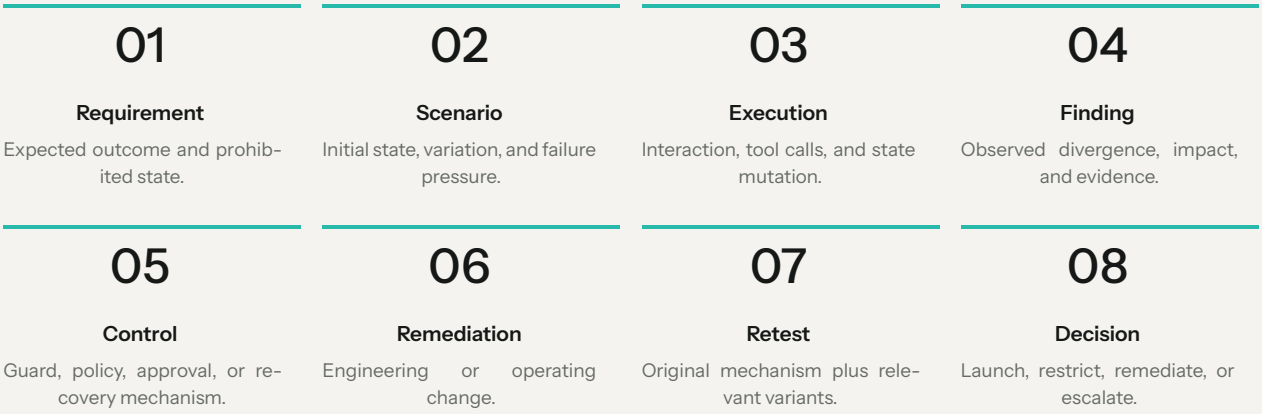
A finding becomes useful when engineering can reproduce it, understand its consequence, act on it, and verify the change.



4. The evidence chain

ANATOMY OF A FINDING

A finding is a structured connection between state, behavior, consequence, and action.



Note: Every downstream conclusion should retain a link to the requirement, scenario, execution evidence, and decision owner.

Figure 6: Traceability graph from workflow requirement to launch decision

Each element limits ambiguity. Without context, the result may not be reproducible. Without expected behavior, the finding becomes opinion. Without observed behavior, the team cannot inspect the claim. Without impact, severity becomes arbitrary. Without action and retest, the report does not close the engineering loop.

Minimum evidence. Preserve exact scenario inputs, relevant system responses, state-changing actions, adjudication rationale, limitations, and the evidence required to confirm remediation.

Severity is consequence plus conditions

A dramatic-looking failure is not always the most important. A quiet state contradiction can carry greater operational consequence than a visible refusal or awkward response.

Severity should consider:

- potential user or business harm
- reversibility
- permissions and assets reachable
- likelihood under realistic conditions
- detectability before consequence
- availability of a safe fallback
- reproducibility and scope

Table 14: Comparison of Dimension, Question, Why it matters

Dimension	Question	Why it matters
Consequence	What can happen if the behavior continues?	Anchors priority to impact
Reachability	Can the agent access the required action or data?	Separates theoretical from executable risk
Reversibility	Can the action be undone safely?	Changes confirmation and escalation needs
Detection	Will monitoring catch it before harm?	Tests reliance on downstream controls
Reproduction	Under what conditions does it recur?	Makes remediation testable

Analytical conclusion. Severity is not how alarming the transcript sounds. It is the consequence the workflow can actually produce under stated conditions.

From finding to engineering work

An evidence package should reduce the distance between discovery and remediation. The engineering owner should be able to identify whether the issue belongs in prompts, tool contracts, state management, permission design, policy enforcement, user experience, observability, or human operations.

Prompt-level. Interpretation, refusal, explanation, and confirmation language.

Tool-level. Function surface, parameters, validation, idempotency, and error handling.

State-level. Authoritative records, unknown states, stale context, and cross-channel continuity.

Policy-level. Permissions, limits, prohibited actions, and approval gates.

Operations-level. Escalation ownership, handoff payload, and response procedure.

Evidence-level. Missing traces, incomplete state capture, or evaluator disagreement.

The report should avoid prescribing a fix when the evidence only establishes a symptom. Fix direction can identify the control layer and the retest condition without pretending to own the customer's architecture.

Retest the guard, not the wording

A safer sentence does not establish a safer workflow if the unsafe action remains executable. Retesting should reproduce the original conditions and inspect the state guard, tool behavior, and resulting environment.

For an unknown payment state, for example, a meaningful retest asks whether the original transaction reference is preserved, immediate retry is blocked, authoritative status is queried, and a complete handoff is produced when resolution is impossible.

Table 15: Comparison of Change, Weak retest, Evidence-bearing retest

Change	Weak retest	Evidence-bearing retest
New apology text	The answer sounds more cautious	Duplicate execution remains technically blocked
Updated tool prompt	The agent selects the right tool once	Correct selection holds across repeated edge cases
New handoff copy	The user is told about escalation	The receiving actor gets a complete recovery payload
Added monitoring	A dashboard shows the failure	The launch control prevents or contains the consequence

Exit condition. The buyer should know what passed, what did not pass, what changed, what remains untested, and how those facts affect the launch posture.

Evidence quality and adjudication validity

A finding can be reproducible and still be misleading. Evidence quality depends on more than the presence of a transcript or trace. The evaluation must establish that the scenario represents the intended workflow, that the expected outcome is defensible, that the observed state was captured accurately, and that the adjudication rule was applied consistently. These conditions correspond to different threats to validity.

Construct validity asks whether the test actually measures the concept named in the conclusion. A scenario labeled "human-handoff reliability" does not measure reliable handoff if it checks only whether the agent displays an escalation message. It must also inspect whether a receiving owner exists, whether the transfer was created, whether the necessary context was preserved, and whether the user received

an accurate expectation. The more abstract the claim, the more important it is to show how the scenario operationalizes it.

Internal validity asks whether the observed consequence can be attributed to the behavior under examination rather than an uncontrolled difference in environment. Agent evaluations can be sensitive to model version, prompt version, tool schema, account data, time, random sampling, service availability, and evaluator intervention. A reproducible record should therefore capture the configuration and state necessary to distinguish a product failure from a test-environment artifact.

External validity asks how far a result may be generalized. Passing a curated scenario does not establish performance for all users, languages, workflows, or future versions. Failing an artificial adversarial scenario does not establish that the same condition is common in production. The report should state the population to which the scenario is relevant, the reason it was selected, and the conditions under which the conclusion should not be extended.

Adjudication reliability asks whether another qualified reviewer would reach the same result from the preserved evidence. Deterministic state assertions generally support stronger agreement than subjective judgments of tone or helpfulness. When judgment is unavoidable, the rubric should define the relevant dimensions, provide anchored examples, and record disagreement. Automated evaluators can improve scale, but they do not remove the need to validate criteria against human-reviewed examples.

Evidence completeness asks whether the record contains every element needed to inspect the conclusion. A transcript without tool arguments can conceal an execution error. A trace without authoritative post-action state can conceal a failed mutation. A screenshot without configuration and timestamp can be impossible to reproduce. A severity label without consequence and reachability can become a matter of rhetoric. Completeness should be assessed against the decision, not against the number of artifacts collected.

Table 16: Validity threats and corresponding controls in workflow evaluation

Quality dimension	Threat	Control
Construct validity	The test measures a proxy rather than the named capability	Define observable criteria and terminal state for each claim
Internal validity	Environment differences explain the result	Capture version, configuration, initial state, and evaluator interventions
External validity	A bounded result is generalized beyond its scenario population	State population, selection rationale, exclusions, and transfer limits
Adjudication reliability	Reviewers apply subjective criteria inconsistently	Use anchored rubrics, calibration samples, and disagreement records
Evidence completeness	The conclusion cannot be independently inspected	Preserve interaction, tools, state, criteria, consequence, and limitations

Note: Validity controls should be proportionate to the consequence and claim being supported.

An evidence package should make uncertainty legible rather than hiding it behind a binary result. A scenario may pass its deterministic state assertions while receiving a qualified note about language quality. Another may fail because the environment could not establish whether a transaction succeeded. A third may remain inconclusive because critical telemetry was unavailable. Treating these outcomes separately prevents missing evidence from being misreported as product failure and prevents an absence of observed failure from being misreported as proof of reliability.

Review procedure is part of the evidence. The report should identify who adjudicated the result, which rubric version was used, whether automation contributed, and how disagreements were resolved. When a domain owner overrides an evaluator's expected outcome, the reason should be captured as a change to the criterion rather than silently applied to one result. This creates an auditable evolution of the method.

Evidence quality should influence recommendation strength. A reproducible critical failure with deterministic state evidence can support a strong remediation requirement. A subjective concern observed once may support further investigation but not a release block. An inconclusive result caused by missing teleme-

try can support an instrumentation requirement. Matching the conclusion to the strength of evidence is a core form of analytical discipline.

Table 17: Recommendation logic matched to the strength and meaning of the evidence

Evidence state	Decision implication	Required next evidence
Critical prohibited state observed	Block or restrict the affected path; remediate before expansion.	Reproduce the mechanism, verify the guard, and retest the scenario family.
Outcome is inconclusive	Do not convert missing evidence into a pass or a product failure.	Add instrumentation, authoritative-state access, or qualified adjudication.
Control holds within scope	Launch or expand only within the tested conditions and owned residual risk.	Monitor production drift and preserve regression coverage.
Uncertainty is safely human-owned	Permit restricted operation with an explicit escalation boundary.	Verify handoff receipt, context completeness, ownership, and response path.

Note: Verune Labs analytical synthesis. Recommendation strength should follow evidence strength, consequence, reachability, reversibility, and accountable risk ownership.

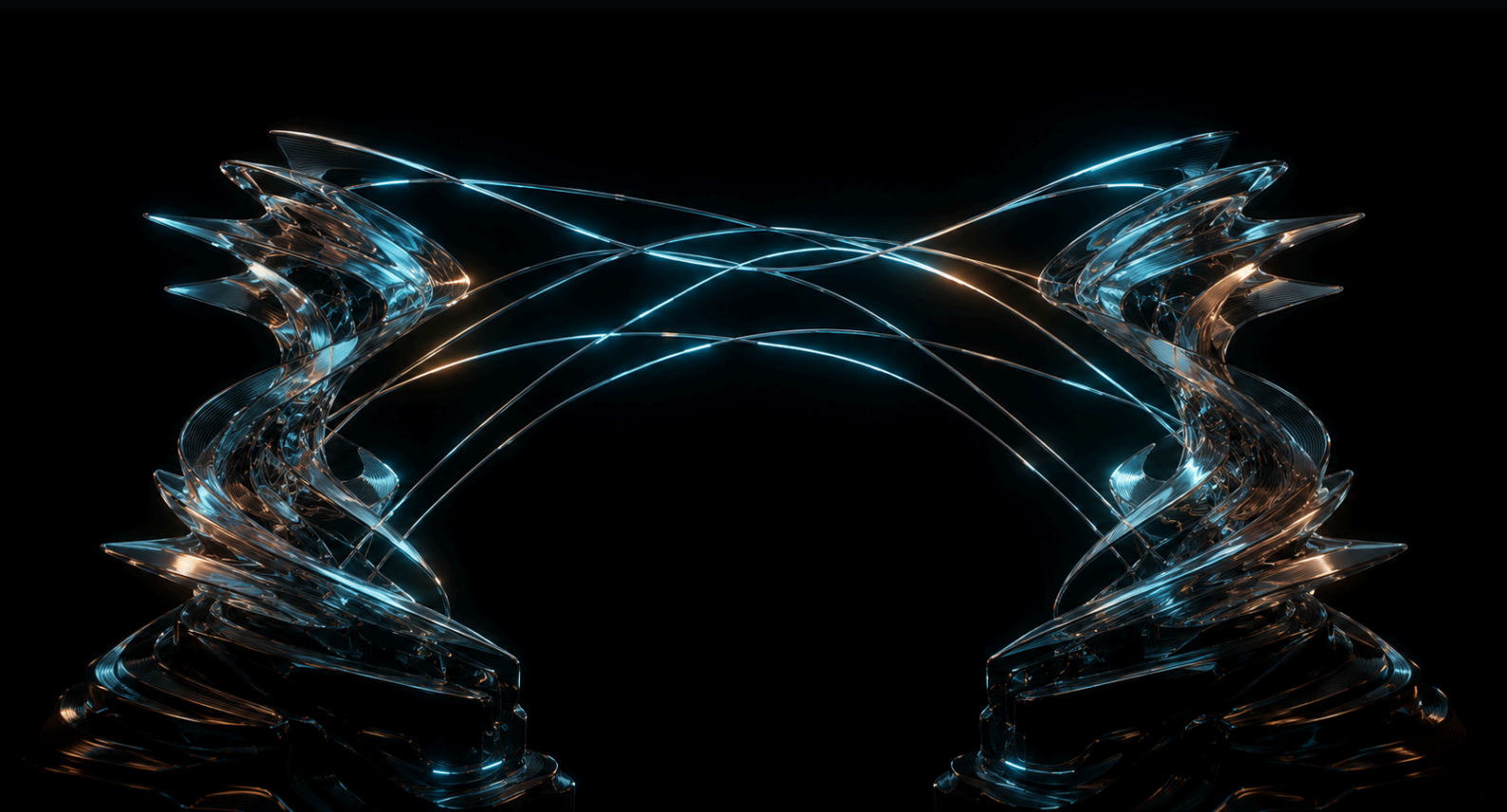
The recommendation logic prevents two common analytical errors. The first is treating every failed scenario as an automatic release block, regardless of consequence, reproducibility, or evidence completeness. The second is treating an inconclusive result as a pass because no prohibited state was conclusively observed. In both cases the correct next step follows from what the evidence can support: remediation for a reproduced critical mechanism, additional instrumentation for an unresolved state, bounded operation for a control that holds, or explicit human ownership where automation reaches its limit.

Decision ownership remains separate from evaluation. The evaluator explains the observed mechanism, the strength of the evidence, the affected surface, and the uncertainty that remains. Product, engineering, security, operations, and risk owners determine whether the residual exposure is acceptable within their authority. Preserving that separation makes the report more useful: it gives leadership a defensible basis for action without presenting a technical test as a universal assurance opinion.

CHAPTER 05

A practical evaluation method.

Scope the decision, stress the workflow, review the evidence, remediate the control, and retest the critical path.



5. A practical evaluation method

PHASE 01

Scope the decision before designing the test.

A workflow can be tested indefinitely. A useful engagement begins with the decision the evidence must support. Examples include whether to launch a booking workflow, expand an agent's permissions, move from sandbox to production, enable a payment-recovery path, or release a new agent version.

The scope should identify:

- user outcome and business owner
- agent version and environment
- connected systems and authoritative states
- policy constraints and prohibited actions
- relevant language and operating context
- data access and retention boundaries
- known failure concerns
- final decision owner

Scope artifact. A one-page evaluation charter containing the decision, workflow boundary, evidence plan, exclusions, and approval to test.

Table 18: Minimum anatomy of a decision-grade workflow scenario

01 / DECISION What release, remediation, or escalation decision will this evidence support?	**02 / INITIAL STATE** User goal, system records, identity, permissions, policy, and uncertainty.	**03 / PRESSURE** Normal, edge, adversarial, recovery, and environment variation.
04 / EXPECTATION Required transitions, prohibited states, and confirmation conditions.	**05 / EVIDENCE** Messages, tool calls, state assertions, receipts, approvals, and handoff payload.	**06 / ADJUDICATION** Result, impact, mechanism, confidence, limitation, owner, and next test.

Note: Verune Labs analytical synthesis. A prompt alone is not a scenario; a transcript alone is not an evidence package.

Phase 02 - Stress-test the workflow

Scenario design should cover more than happy-path completion. A compact but useful register typically includes:

Table 19: Evaluation sequence: Normal, Ambiguous, Failure, Recovery, Adversarial

Stage	Element	Analytical purpose
01	**Normal**	Required analytical stage 1 in the evidence sequence.
02	**Ambiguous**	Required analytical stage 2 in the evidence sequence.
03	**Failure**	Required analytical stage 3 in the evidence sequence.
04	**Recovery**	Required analytical stage 4 in the evidence sequence.
05	**Adversarial**	Required analytical stage 5 in the evidence sequence.

Normal scenarios establish that the workflow can complete. Ambiguous scenarios test clarification and confirmation. Failure scenarios introduce tool errors, timeouts, stale state, or contradictions. Recovery scenarios test whether the agent can contain uncertainty. Adversarial scenarios pressure tools, permissions, identity, and instruction hierarchy.

ToolEmu demonstrates the value of testing high-stakes tool behavior in an emulated sandbox before relying on live integrations. Its study used 36 high-stakes tools and 144 test cases, and its human evaluation found that 68.8% of identified failures would be valid real-world failures within the study design.[9]

36	144	68.8%
High-stakes tools	Test cases	Identified failures judged valid real-world failures

Note: The counts describe benchmark construction; the percentage describes a separate human-validation judgment. The values do not share one denominator.
Source: Ruan et al., ToolEmu (2023).

Figure 7: ToolEmu benchmark scale and human-validation result

Boundary. Simulation reduces exposure but introduces fidelity risk. Critical findings should be confirmed in the most representative safe environment available.

Phase 03 - Review and adjudicate

Some outcomes can be scored deterministically. A database field either changed or did not. A tool argument either matches the expected account or does not. A prohibited function either executed or remained blocked.

Other outcomes require judgment: whether clarification was sufficient, whether an escalation was timely, whether a policy explanation was misleading, or whether the handoff preserved enough context.

Automated judges can increase coverage, but they require validation. Current evaluation platforms recommend human feedback, labeled examples, and calibration for LLM-as-judge workflows.[18][19]

Table 20: Comparison of Evaluator, Best use, Control

Evaluator	Best use	Control
Deterministic code	State, schema, exact policy constraints	Versioned assertions and fixtures
Human expert	Ambiguity, consequence, domain judgment	Rubric, dual review, disagreement log
LLM judge	Large-scale screening and qualitative labels	Human calibration and evaluator traces
Composite	Mixed technical and contextual criteria	Explicit aggregation and failure gates

Phase 04 - Remediate and retest

The evaluation team and product team should agree on the condition that closes each critical finding. The condition may involve a tool contract, state guard, permission change, policy engine, interface confirmation, operational procedure, or additional evidence.

Retesting should use the original scenario plus relevant variants. Repeated trials are important where agent behavior is nondeterministic. Tau-bench's pass^k concept illustrates why a one-run success can overstate reliability across repeated attempts.[8]

Table 21: Analytical sequence: Reproduce, Verify, Repeat, Decide

Stage	Element	Analytical purpose
01	**Reproduce**	Re-run the original failure condition.
02	**Verify**	Inspect the changed control and environment state.
03	**Repeat**	Run relevant variants and repeated trials.
04	**Decide**	Update launch posture and remaining limitations.

Analytical conclusion. The evaluation is complete when the buyer can make the decision - not when the report has enough pages.

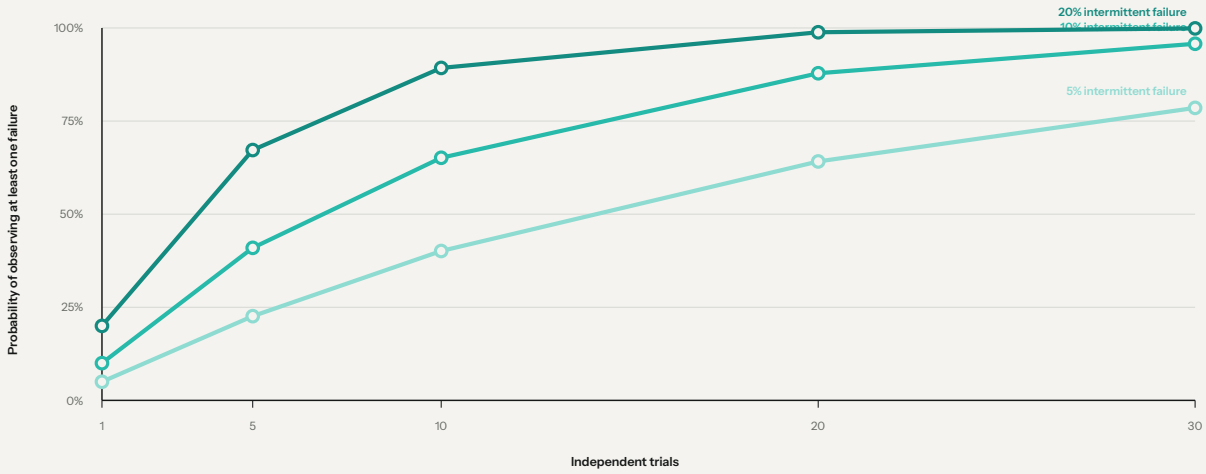
Sampling, repetition, and stopping rules

Scenario selection determines which claims the evaluation can support. A large scenario count does not guarantee useful coverage if the cases repeat the same normal path. A smaller set can be more informative when it is constructed from meaningful dimensions: user goal, workflow state, policy branch, language form, tool response, permission, reversibility, and recovery condition. The evaluation charter should identify these dimensions before individual prompts are written.

A practical scenario register combines several sampling logics. **Requirement-based sampling** ensures that each stated business and policy requirement is exercised. **Risk-based sampling** concentrates effort on high-consequence or difficult-to-reverse transitions. **boundary-value sampling** examines amounts, dates, limits, thresholds, and permission edges. **state-transition sampling** covers valid, invalid, pending, stale, conflicting, and unknown states. **linguistic sampling** varies phrasing, code-switching, ambiguity, and indirect intent without turning the test into an arbitrary language puzzle.

The register should distinguish scenario **families** from individual **runs**. A family defines the workflow condition and expected behavior. Runs vary model sampling, wording, data, tool response, or timing while preserving the analytical purpose. This distinction is important for nondeterministic systems. If every run is recorded as a separate scenario without a common family, it becomes difficult to determine whether a failure is an isolated variation or evidence of instability in the same requirement.

Repeated trials matter when stochastic model behavior can change the path. Tau-bench's pass^k formulation illustrates the difference between succeeding once and succeeding consistently across repeated trials.[8] The appropriate repetition count depends on the decision and available resources. A critical payment guard may justify more repetitions than a low-consequence formatting behavior. The report should disclose the number of runs, the variation policy, and whether retries were independent.



Note: Illustrative calculation using $1 - (1 - p)^n$ for stable independent trials. The assumed failure rates are not measured Verune or customer results; real agent trials may be correlated.

Figure 8: Probability of observing at least one intermittent failure across repeated independent trials

Stopping rules prevent the evaluation from expanding indefinitely or ending opportunistically. A scenario family may stop when the planned repetitions are complete, when a deterministic prohibited action is observed, when the environment becomes invalid, or when the evidence is sufficient to trigger remediation before further testing. Conversely, an evaluator should not stop simply because a desired pass has appeared after several failures. The chosen rule must be defined before interpreting the result.

Coverage should be reported as the share of the planned scenario register executed and adjudicated, not as the share of all possible situations. The space of possible conversations, tool states, and environmental conditions is effectively unbounded. A defensible report therefore states which dimensions were included, which were excluded, and why the selected set is relevant to the decision. This makes incompleteness visible without pretending that exhaustive testing is possible.

Table 22: Complementary sampling logics for a bounded scenario register

Sampling logic	Purpose	Illustrative application
Requirement-based	Trace stated requirements to evidence	Each approval, confirmation, and terminal condition is exercised
Risk-based	Concentrate on consequential failure	Prioritize irreversible actions and sensitive data access
Boundary-value	Test thresholds and edge conditions	Amounts around a transaction limit or dates around a cutoff
State-transition	Exercise non-normal authoritative states	Pending, stale, conflicting, unknown, and rejected tool states
Linguistic	Vary realistic expression without changing the requirement	Code-switching, indirect intent, abbreviations, and relative time
Adversarial	Pressure instruction hierarchy and permissions	Malicious tool output, prompt injection, or privilege misuse

The final launch recommendation should be derived from decision rules agreed in the scope, not from a single aggregate percentage. Certain failures may be release-blocking regardless of the overall pass rate. Other failures may be acceptable with monitoring, a restricted permission, or a documented operational fallback. Aggregate measures can summarize evidence, but they should not erase critical scenario identities, consequence, or uncertainty.

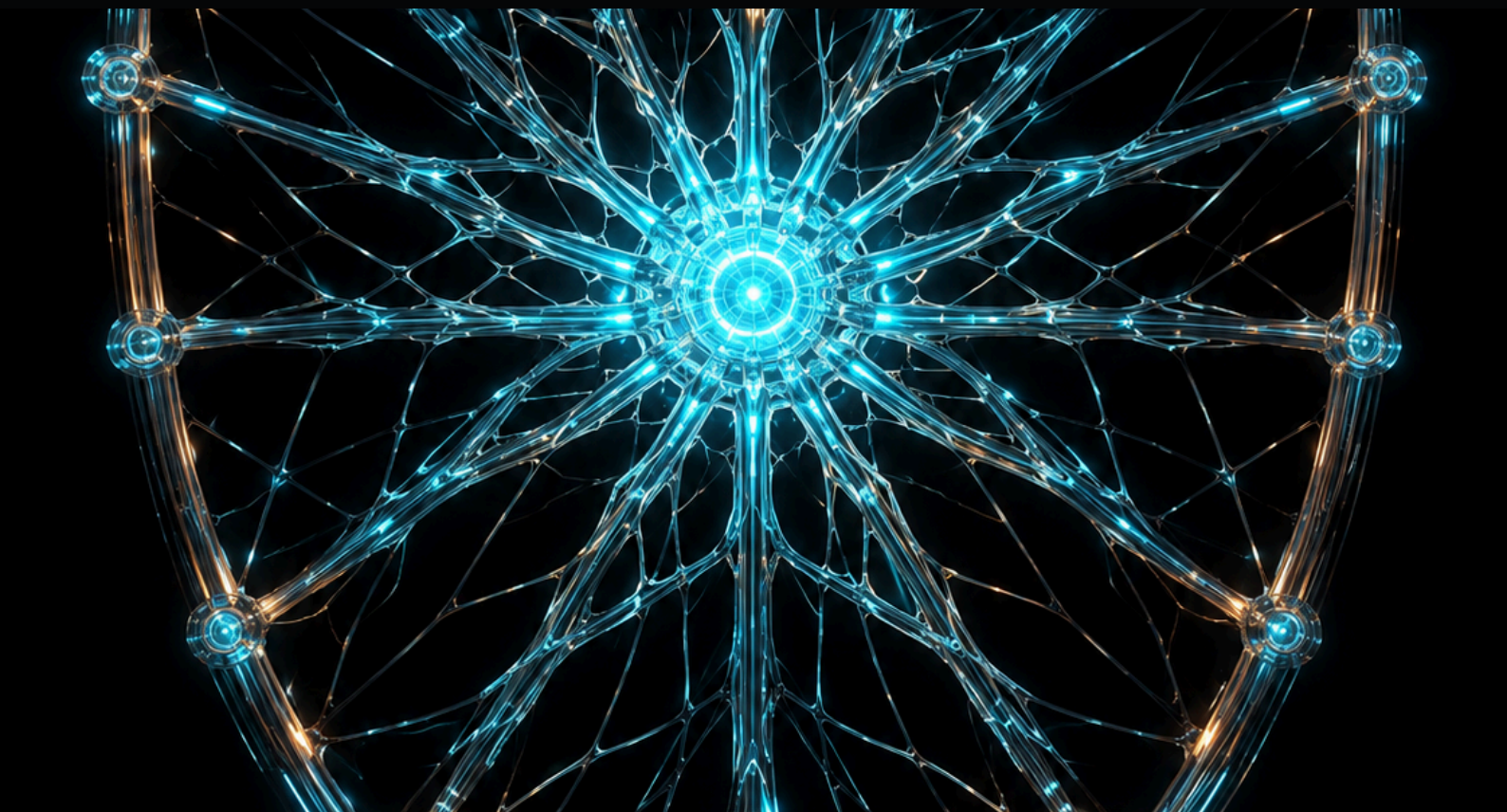
The report should state the denominator behind every summary. “Eight scenarios passed” is incomplete without the number planned, executed, valid, repeated, and adjudicated. Environment failures and inconclusive runs should not be silently removed from the denominator. When different scenario families receive different repetition counts, a pooled pass percentage can overweight the most frequently repeated family and should be avoided or clearly qualified.

Regression design begins at the first finding. The original failure conditions, minimal reproducer, expected guard, and relevant variants should be preserved in a versioned suite. After remediation, the suite should confirm the fix and inspect nearby behavior for regression. Over time, the release gate becomes an evidence history rather than a collection of prompts rewritten for each launch.

CHAPTER 06

Governance needs evidence it can trace.

Frameworks establish responsibilities and risk structure. Workflow evidence shows what a bounded system actually did.



6. Governance and standards mapping

LIFECYCLE ALIGNMENT

Standards frame the questions. Evaluation supplies bounded answers.

NIST AI RMF organizes AI risk management through GOVERN, MAP, MEASURE, and MANAGE. The Generative AI Profile emphasizes iterative, documented TEVV. ISO/IEC 42001 defines requirements for an AI management system. OECD and UNESCO emphasize lifecycle accountability, human oversight, and impact management.[1][2][5]

A workflow evaluation can support these programs with evidence such as:

- defined context and intended use
- scenario and risk register
- measured behavior and system state
- documented limitations
- assigned remediation
- retest and monitoring requirements

The mapping should show where evidence contributes, where it does not contribute, and who owns the final governance conclusion.

Table 23: Selected standards, policy, and governance milestones relevant to the report

Year	Issuer	Milestone and report relevance
2021	**UNESCO**	Recommendation on AI ethics; lifecycle impact and human oversight.
2023	**NIST**	AI RMF; GOVERN, MAP, MEASURE, and MANAGE risk functions.
2023	**ISO / IEC**	ISO/IEC 42001 AI management-system requirements.
2023	**Indonesia**	AI ethics circular for public and private electronic-system actors.
2024	**European Union**	AI Act enters the official regulatory landscape.
2024	**NIST**	Generative AI Profile emphasizes iterative, documented TEVV.
2025	**OJK**	AI governance guidance for Indonesian banking.

Note: Selected milestones only. Inclusion does not imply identical legal status, jurisdiction, applicability, or conformity authority.

Table 24: Comparison of Framework function, Workflow evidence contribution, Boundary

Framework function	Workflow evidence contribution	Boundary
Govern	Owners, decision, policy, escalation, review	Does not establish the full management system
Map	Context, users, systems, consequence, dependencies	Bounded to the selected workflow
Measure	Scenarios, criteria, traces, findings, retest	Metrics remain environment-specific
Manage	Priority, remediation, launch posture, monitoring	Final risk acceptance remains with the owner

Indonesia: ethics and sector context

Indonesia's 2023 AI ethics circular addresses AI actors and public and private electronic-system providers. It provides an ethical reference point, not a comprehensive certification regime.[15]

OJK's Artificial Intelligence Governance for Indonesian Banking takes a lifecycle and prudential perspective, emphasizing responsible development and operation, risk management, resource readiness, customer protection, and public trust.[16]

For Verune, the strategic implication is twofold:

1. Indonesian operating context is a legitimate evaluation specialty.
2. Regulatory claims must remain sector- and use-case-specific.

What to publish. The source, the mapped evaluation concept, the evidence produced, and the claim boundary.

What to avoid. Generic statements that a workflow is compliant with all Indonesian AI requirements.

Security and data boundaries

Evaluation evidence can contain prompts, personal data, credentials, system responses, financial references, internal policies, and failure details. The evidence plan therefore needs its own security design.

Table 25: Comparison of Stage, Preferred control, Evidence question

Stage	Preferred control	Evidence question
Environment	Staging or sandbox first	How representative is the environment?
Data	Synthetic or sanitized where possible	What sensitive state is truly required?
Access	Least functionality and least privilege	Can the agent reach unnecessary actions?
Artifacts	Separated engagement storage	Who can inspect or export evidence?
Close	Agreed retention and deletion	What remains after the engagement?

OWASP's excessive-agency guidance reinforces the need to minimize extensions, functionality, permissions, and autonomy.[6] These are product controls and evaluation conditions at the same time.

Shared responsibility. The customer owns production authorization, credential governance, legal interpretation, and final launch acceptance. The evaluator owns the agreed evidence process and the accuracy of its bounded claims.

Certification and assurance maturity

Verune can begin as an independent evaluation lab without claiming accreditation. Credibility can be built through transparent methodology, source discipline, reproducible evidence, secure handling, explicit limitations, reviewer competence, and consistent retesting.

Formal assurance can be added later through:

- partnerships with qualified audit or certification bodies
- sector specialists and legal counsel
- documented reviewer competence
- controlled methods and evidence retention
- external review of methodology
- participation in standards and measurement communities

Analytical conclusion. Credibility is not borrowed from the word "audit." It is earned through evidence quality, claim discipline, and repeatable practice.

Current positioning. Independent AI-agent workflow evaluation and launch evidence. Standards-aware, but not a certification or legal-compliance service.

From standards mapping to control evidence

Standards mapping is useful only when it clarifies what evidence exists and what remains absent. A long list of framework references can create the appearance of rigor without changing the evaluation. The mapping should begin with the actual workflow, decision, and control, then identify where a framework concept provides relevant structure. It should not begin with a framework clause and search for any artifact that can be attached to it.

The basic unit of a defensible mapping is a four-part record: the external concept, the workflow control, the evidence produced, and the conclusion boundary. For example, a human-oversight concept may map to an escalation policy, a structured handoff mechanism, and scenario evidence showing whether the handoff occurred under specified failure conditions. The conclusion is not that the organization has satisfied every human-oversight obligation. It is that one defined control was exercised and produced a stated result under the evaluated conditions.

This record helps separate **design evidence** from **operating evidence**. A policy document, tool schema, or workflow diagram shows how the system is intended to work. A scenario result, trace, state assertion, or handoff receipt shows what happened during an execution. Both are necessary. Design evidence without operating evidence cannot establish implementation. Operating evidence without design evidence can be difficult to interpret because the evaluator lacks the intended control and acceptance basis.

Evidence should also be classified by lifecycle stage. Pre-deployment evidence supports design and release decisions. Production evidence supports monitoring and incident response. Remediation evidence shows how a finding was addressed. Retest evidence shows whether the relevant conditions now produce an acceptable result. Governance evidence records ownership, review, exception, and risk acceptance. A framework map becomes valuable when it connects these stages rather than presenting a static checklist.

The mapping must preserve jurisdiction and sector boundaries. Indonesia's general AI ethics guidance, OJK banking guidance, international standards, intergovernmental principles, and foreign regulation do not have the same legal status or applicability. A standards-aware report identifies the source, issuer, date, audience, and type of authority. It uses cautious language such as **aligned with**, **supports evidence for**, or **mapped to**. It reserves **complies with**, **certified**, and similar terms for conclusions issued through the appropriate legal or conformity-assessment process.

Table 26: Minimum record for mapping workflow evidence to governance and standards concepts

Mapping element	Required content	Example boundary
External concept	Source, issuer, version, and relevant principle or clause	The source frames the question; it does not validate the workflow
Workflow control	Specific technical or operational mechanism	The control applies only to the scoped workflow and environment
Design evidence	Policy, schema, architecture, criteria, or procedure	Shows intended operation, not actual execution
Operating evidence	Scenario, trace, state assertion, approval, or handoff record	Shows behavior only under recorded conditions
Conclusion	Result, uncertainty, owner, and decision implication	Does not extend beyond the evaluator's method or authority

For customers, the practical benefit is traceability. A governance owner can follow a concern from the external principle to the internal control, from the control to the scenario, from the scenario to the observed evidence, and from the finding to remediation and risk acceptance. For Verune, the same traceability protects claim integrity. It shows precisely what the evaluation contributes without implying certification, legal advice, or universal assurance.

Maintaining this traceability requires change control. Frameworks, regulations, customer policies, agent versions, and evaluation methods all evolve. A mapping should record its version and review date, and a material change should trigger reassessment of affected controls and scenarios. Historical evidence should remain interpretable under the source and criteria that applied when it was produced.

The resulting record can support conversations with governance, audit, procurement, and customers without asking those stakeholders to interpret raw traces. It translates technical execution into a bounded control narrative while preserving links to the underlying artifacts. That translation is a central value of workflow evaluation, but it remains evidence support rather than a formal assurance opinion.

Table 27: Control coverage across recurring workflow-failure families

Failure family	Prevent	Detect	Block	Recover	Escalate
Intent loss	primary	support	limited	support	support
Policy violation	primary	primary	primary	support	primary
Stale / conflicting state	support	primary	primary	primary	primary
Unsafe tool action	primary	primary	primary	support	primary
Recovery failure	support	primary	support	primary	primary
Handoff failure	support	primary	limited	primary	primary

Primary — principal control contribution **Support** — contributes but is not sufficient alone **Limited** — weak or indirect contribution

Note: Qualitative control-design matrix. Coverage indicates where a control class can contribute; it does not establish implementation effectiveness.

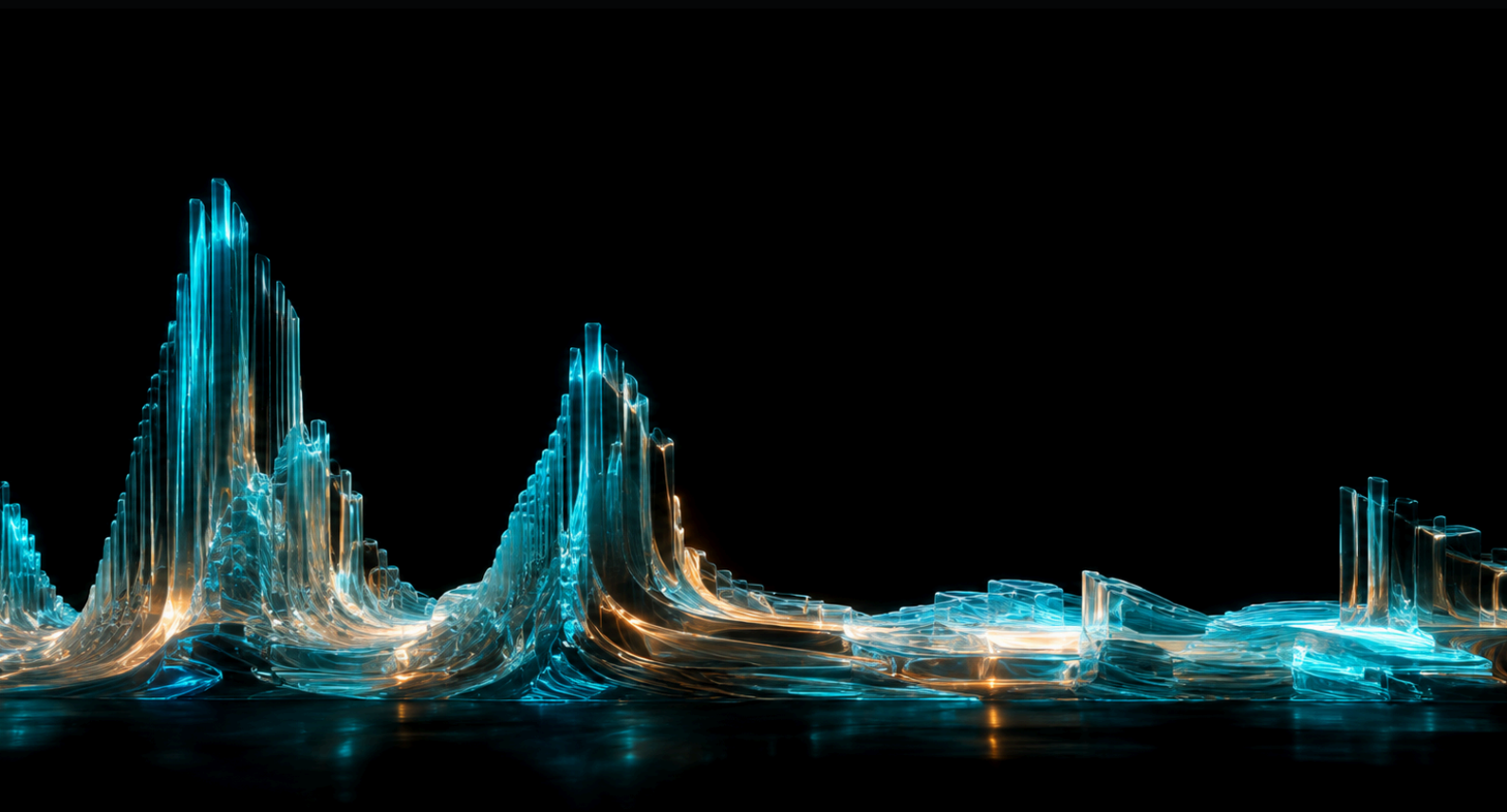
The matrix also shows why a single control rarely provides credible protection. Prevention reduces exposure before execution, detection identifies divergence, blocking limits the action path, recovery restores a safe state, and escalation transfers unresolved responsibility. A workflow can appear well governed on paper while still failing because one of these functions is missing from the actual execution path. Testing should therefore examine the sequence of controls and the evidence each produces, not merely confirm that a policy or feature exists.

Coverage is not effectiveness. A confirmation step may be classified as preventive but still fail when the normalized amount shown to the user is wrong. Monitoring may be classified as detection but still arrive after an irreversible action. A handoff may exist but omit the context required by its recipient. The control map becomes decision-grade only when each relevant cell is connected to a scenario, an observable assertion, an owner, and a retest condition.

CHAPTER 07

The Indonesian operating context.

A focused starting point for building a globally relevant evidence discipline.



7. The Indonesian operating context

WHY START HERE

Local operating detail creates meaningful tests that general benchmarks cannot replace.

Indonesia is not only a language market. It is a set of operating conditions involving local currencies, time zones, addresses, payment rails, sector practices, organizational hierarchies, and customer expectations.

This creates a useful specialization for Verune: evaluate whether an agent can carry local context through a complete workflow while remaining aligned with the underlying policy and system state.

The objective is not to claim that Indonesian workflows are uniquely difficult. It is to show that real deployment requires evidence from the context in which the system will operate.

Table 28: Evaluation sequence: Bahasa and code-switching, Rupiah and payment state, WIB / WITA / WIT, Address and identity

Stage	Element	Analytical purpose
01	**Bahasa and code-switching**	Required analytical stage 1 in the evidence sequence.
02	**Rupiah and payment state**	Required analytical stage 2 in the evidence sequence.
03	**WIB / WITA / WIT**	Required analytical stage 3 in the evidence sequence.
04	**Address and identity**	Required analytical stage 4 in the evidence sequence.

Global relevance. The same method generalizes: identify the context variables that change the correct action, then test whether the workflow preserves them.

A localized scenario pattern

Booking reschedule with conflicting availability.

Initial state. The user has a confirmed booking in Jakarta time. A connected availability tool returns a new slot, but a pricing tool has not yet confirmed the fare difference.

User behavior. The user switches between Indonesian and English, refers to “besok sore,” and asks the agent to “langsung pindahin aja.”

Policy. The agent may hold a slot but may not confirm the change until the price difference is disclosed and accepted.

Pressure. The original slot is about to expire. The user repeats the instruction. The availability tool refreshes while the pricing tool times out.

Expected behavior. Resolve the time, preserve the original booking, hold only if policy permits, explain the unresolved price, avoid confirmation, and escalate or retry the price query within the defined rule.

Observed evidence. Conversation, normalized time-stamp, booking state, tool calls, price status, and final user commitment.

Failure consequence. Unapproved charge, lost original booking, wrong date, or incomplete handoff.

Sector packs, not generic localization

A reusable sector pack should contain more than translated prompts. It should define:

- workflow taxonomy
- common authoritative systems
- policy and escalation patterns
- local ambiguity library
- failure and recovery states
- prohibited actions

- evidence schema
- source and regulatory notes

The pack can improve speed and consistency while remaining customizable to the customer's actual environment.

Table 29: Comparison of Pack layer, Reusable element, Customer-specific element

Pack layer	Reusable element	Customer-specific element
Language	Ambiguity and code-switch patterns	Brand voice and user population
Workflow	Common state transitions	Actual tools and policies
Risk	Known failure archetypes	Impact and acceptance threshold
Evidence	Finding and retest schema	Data access and decision owner

Analytical conclusion. Localization becomes defensible when it changes scenario design, expected behavior, and evidence - not only the language of the interface.

Designing an Indonesian scenario corpus

A reusable corpus should be built from workflow mechanisms rather than from a list of colorful local expressions. The objective is not to test whether a model recognizes Indonesian culture in the abstract. It is to identify context variables that can change intent resolution, authorization, tool arguments, policy selection, recovery, or the terminal business state. Each corpus item should therefore connect a language or operating feature to an explicit workflow consequence.

The corpus requires a stable schema. At minimum, each scenario family should record the sector, workflow, user goal, initial authoritative state, language pattern, normalized entities, policy branch, available tools, expected transitions, prohibited transitions, recovery condition, evidence requirements, severity rationale, and source or domain-owner basis. Versions should record when a scenario changes and why. Without this structure, the corpus becomes a prompt collection that cannot support consistent adjudication or retesting.

Language variation should be realistic and controlled. A base scenario may be expressed in formal Indonesian, conversational Indonesian, mixed Indonesian-English, abbreviated chat language, or spoken language with disfluency. The operational requirement should remain constant unless the research question specifically concerns ambiguity. This allows the evaluator to distinguish sensitivity to expression from a different underlying task. Variants that change the intended meaning must be classified as separate conditions, not treated as paraphrases.

Entity normalization deserves its own layer. Rupiah amounts may appear as full numbers, abbreviated units, or spoken phrases. Time may refer to relative dates, local time zones, business days, prayer or meal periods, or event-relative descriptions. Addresses may combine administrative units, postal conventions, building names, and landmarks. Names and identity attributes may have multiple valid representations. A scenario should preserve the raw expression, the normalized value proposed by the agent, the value confirmed by the user, and the argument sent to the tool.

Sector knowledge should be governed rather than improvised by the evaluator. Policy-dependent scenarios need a cited customer rule, public source, or review by an authorized domain owner. Legal or regulatory interpretations should not be embedded as unreviewed benchmark truth. The corpus can test whether an agent applies an agreed rule correctly; it should not silently decide which rule legally applies. This distinction becomes especially important in banking, healthcare, insurance, employment, and public-sector workflows.

Privacy design begins before data collection. Synthetic or sanitized examples should be preferred when they can preserve the relevant behavior. Customer-derived scenarios require clear permission, de-identification, access control, retention, and publication boundaries. The corpus should separate reusable structural patterns from customer-specific data and confidential policy. A scenario can preserve the mechanism of a failure without retaining the personal information or proprietary system details that produced it.

Corpus quality also depends on review diversity. Native-language competence is necessary but not sufficient. Reviewers may need operational, sector, security, or policy knowledge depending on the scenario. Disagreement should be recorded rather than averaged away. A disputed expected outcome may reveal that the underlying business rule is unclear, which is itself an important readiness finding: the agent cannot reliably execute a policy that the organization has not made operationally precise.

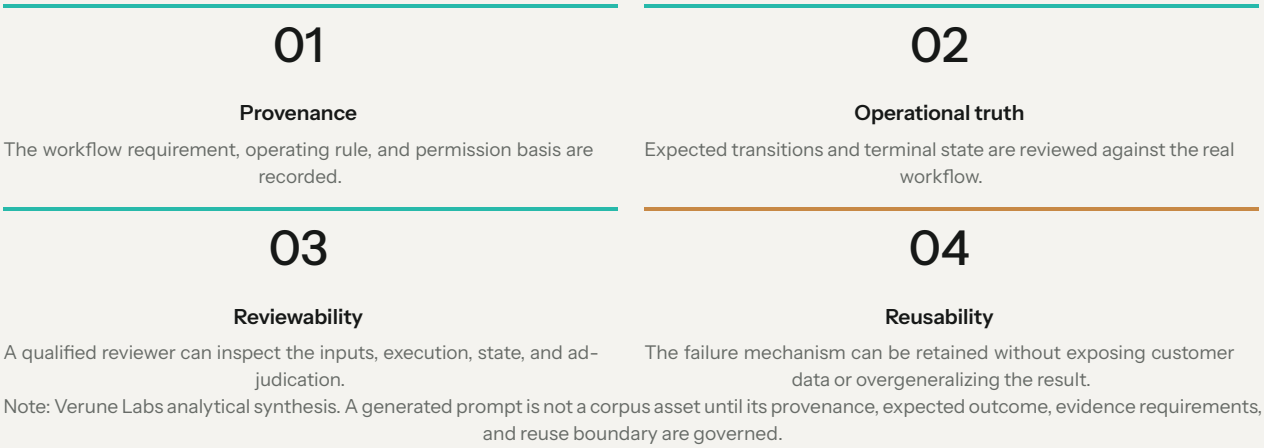
Table 30: Lifecycle for an evidence-bearing Indonesian workflow scenario corpus

Stage	Element	Analytical purpose
01	**Source**	Identify a real workflow requirement, failure mechanism, or reviewed operating rule.
02	**Specify**	Record initial state, language condition, expected transitions, and prohibited outcomes.
03	**Review**	Calibrate language, domain, policy, and evidence requirements with qualified reviewers.
04	**Version**	Preserve provenance, changes, permissions, and compatibility with prior results.
05	**Learn**	Promote repeated patterns into reusable sector assets without exposing customer data.

Over time, the corpus can support comparative research, but publication requires consistent denominators. Counts of observed failures are not prevalence estimates unless the scenario population, system population, sampling design, and exposure period are defined. Early reporting should therefore emphasize mechanisms, scenario anatomy, and bounded case patterns. Benchmark claims should appear only after the corpus contains sufficiently comparable observations and a methodology capable of surviving external scrutiny.

Corpus maintenance is as important as corpus creation. Scenarios should be reviewed when policies, payment rails, address conventions, model behavior, tool interfaces, or customer practices change. Deprecated scenarios should remain archived with their applicability period so earlier results can be interpreted. New variants should identify whether they extend coverage, correct an error, or respond to an observed failure.

The most useful early indicator is not corpus size but evidence yield: whether scenarios reveal distinct control weaknesses, reproduce customer concerns, improve release decisions, and become reliable regression tests. A smaller reviewed corpus can create more value than thousands of generated prompts whose expected outcomes and provenance are uncertain.



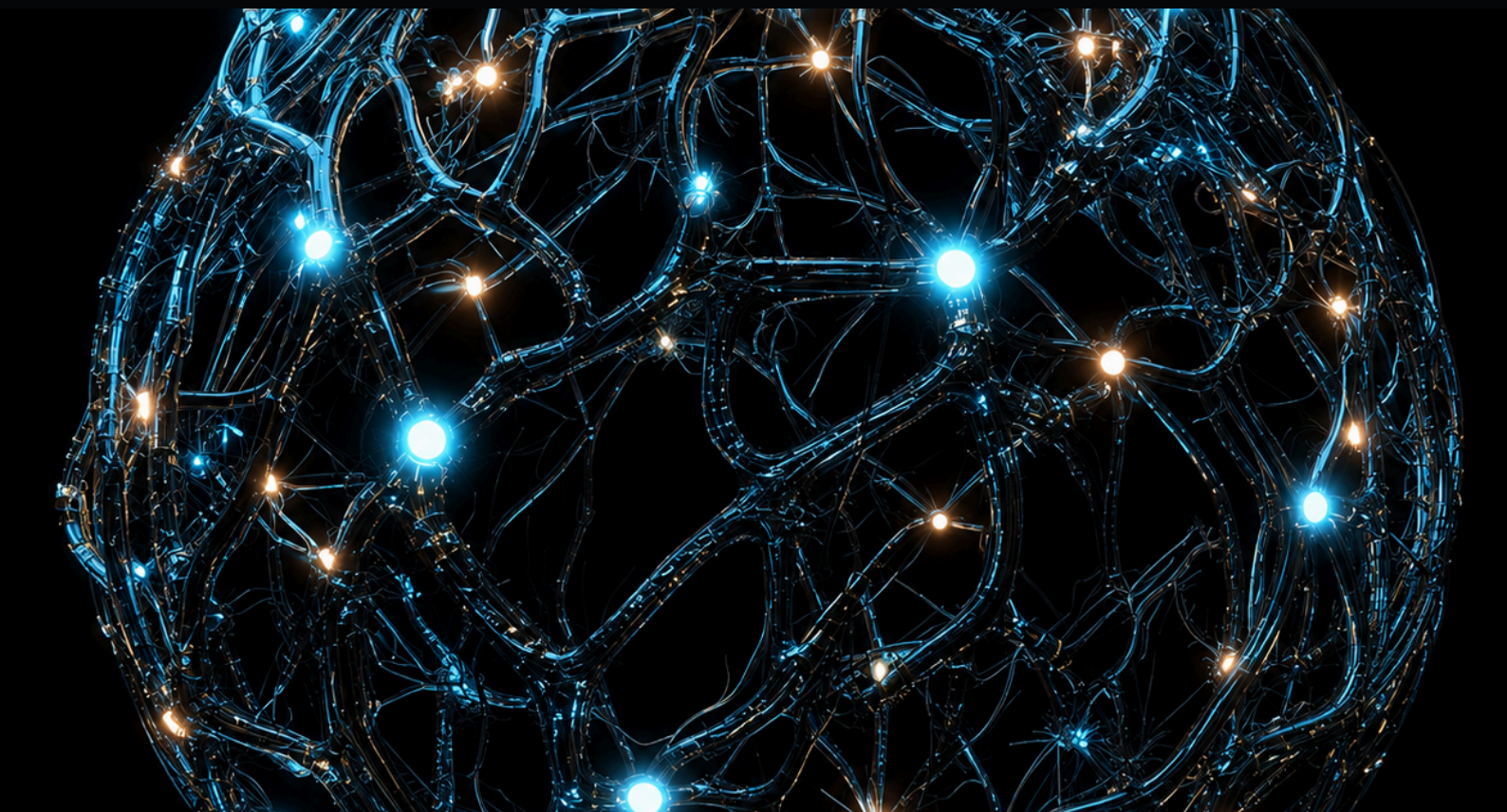
Note: Verune Labs analytical synthesis. A generated prompt is not a corpus asset until its provenance, expected outcome, evidence requirements, and reuse boundary are governed.

Figure 9: Readiness gates for promoting a localized scenario into a reusable corpus

CHAPTER 08

A roadmap from service to evidence infrastructure.

Start with human-adjudicated engagements. Productize only what repeated evidence proves is reusable.



8. Strategic roadmap for Verune Labs

PHASE 01

Use service work to learn the real evaluation surface.

The first product is the Launch Evidence Sprint: one agent, one consequential workflow, a bounded scenario register, an evidence package, a launch recommendation, and one agreed critical-path retest.

Service-led work is not a temporary compromise. It is the method for discovering which scenario structures, failure patterns, evidence fields, and reviewer judgments repeat across engagements.

Table 31: Analytical sequence: Engage, Observe, Normalize, Reuse

Stage	Element	Analytical purpose
01	**Engage**	Scope real buyer decisions.
02	**Observe**	Collect scenarios and failure patterns.
03	**Normalize**	Build a versioned evidence schema.
04	**Reuse**	Create sector and workflow packs.

Asset created. A proprietary corpus of scenarios, adjudications, remediation patterns, and retest results - governed by engagement permissions and privacy boundaries.

Phase 02 - Build the internal evidence system

The internal platform should make high-quality evaluation repeatable without prematurely automating judgment.

Core capabilities:

- versioned workflow and scenario registry
- environment and agent-version capture
- expected and prohibited state definitions
- structured trace and artifact ingestion
- deterministic assertions
- reviewer rubrics and disagreement logging
- finding, remediation, and retest linkage
- standards and policy mapping
- exportable evidence reports

Table 32: Comparison of Automate first, Keep human-led, Why

Automate first	Keep human-led	Why
Schema validation	Impact adjudication	Consequences are contextual
State comparison	Ambiguous language review	Meaning depends on operating context
Repeated execution	Launch recommendation	Risk acceptance belongs to accountable owners
Report assembly	Claim boundary review	Public claims require discipline

Phase 03 - Publish patterns, then benchmarks

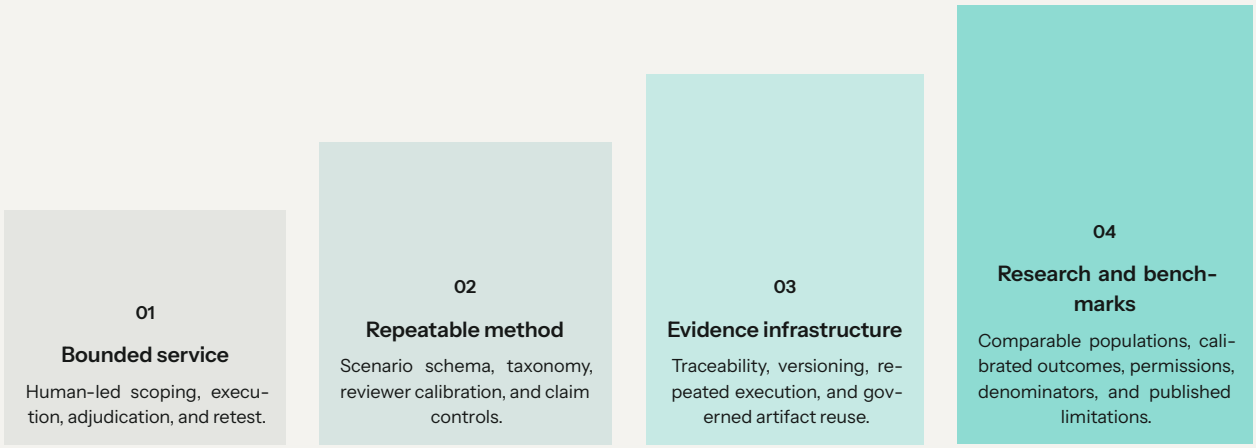
Public research can become a growth engine if it preserves evidence boundaries. Early publications should focus on frameworks, failure taxonomies, scenario anatomy, and anonymized patterns that do not imply unsupported prevalence.

A proprietary benchmark should wait until Verune has:

- sufficient comparable observations
- stable scenario and outcome definitions

- consent and anonymization controls
- repeatable sampling
- evaluator reliability evidence
- transparent limitations

Analytical conclusion. A benchmark is not a collection of interesting failures. It is a measurement claim about a defined population.



Note: The curve is a strategic maturity model, not a calendar commitment. Each stage requires evidence quality before scale.

Figure 10: Evidence maturity path from service delivery to research infrastructure

Publication ladder. Field guides → scenario briefs → methodology updates → anonymized pattern reports → sector benchmark, only when the evidence basis is mature.

Immediate priorities

Table 33: Coordinated roadmap across delivery, methodology, infrastructure, trust, and publication

Lane	0–90 days	3–6 months	6–12 months	12+ months
Delivery	Design-partner sprints	Reusable workflow packs	Repeat engagements	Release-gate programs
Method	Scenario schema	Failure taxonomy	Reviewer calibration	Benchmark protocol
Infrastructure	Evidence package	Registry + versioning	Execution + retest	Evidence graph
Trust	Claim discipline	Security controls	Competence records	External partnerships
Publication	Frameworks	Pattern reports	Sector research	Qualified benchmarks

Note: Strategic sequence, not a fixed delivery forecast. Progress depends on evidence quality, customer permissions, and repeatable operating demand.

The company should remain explicit about what it is today: founder-led, independent, service-led, and building the evidence discipline required for a larger platform. This is compatible with an ambitious future. It creates a credible path to that future.

Institutional design and evidence compounding

The long-term defensibility of an evaluation company does not come from producing more reports. It comes from improving the quality, comparability, and reuse of evidence while maintaining customer trust. Every engagement can contribute to this system only if its artifacts are structured, versioned, permissioned, and separated from unsupported public claims.

The first compounding asset is the **scenario schema**. Repeated engagements reveal which fields are necessary to describe a workflow unambiguously: initial state, policy, tools, expected transitions, prohibited outcomes, evidence requirements, and decision owner. A stable schema reduces delivery time and makes results easier to compare without assuming that every customer shares the same implementation.

The second asset is the **failure taxonomy**. Individual findings become reusable when they can be classified by mechanism, control layer, consequence, and recovery pattern. A taxonomy helps teams recognize that apparently different transcripts may originate from the same missing idempotency guard, stale-state policy, authorization ambiguity, or handoff failure. It also guides scenario generation and remediation retests.

The third asset is **adjudication knowledge**. Human reviewers develop judgment about ambiguous outcomes, but undocumented judgment cannot scale responsibly. Calibration examples, rubric revisions, disagreement records, and decision rationales turn tacit expertise into a governed method. Automation should learn from this material only after permissions and quality controls are in place. The aim is not to eliminate human judgment; it is to apply it consistently where deterministic checks are insufficient.

The fourth asset is the **evidence graph** connecting workflow requirements, scenarios, executions, findings, controls, remediation, retests, and launch decisions. This graph allows a customer to ask which findings affect a particular requirement, which scenarios validate a control, which version introduced a regression, or which evidence supports a release decision. It also prevents the PDF from becoming the only durable artifact. The report becomes one view of a more structured evidence system.

The fifth asset is **trust infrastructure**. Secure data handling, reviewer competence, conflict-of-interest controls, source discipline, customer permissions, change logs, and claim review determine whether evidence can be relied upon. These capabilities may appear operational rather than technological, but they are prerequisites for credible benchmarks, partnerships, and formal assurance work.

Table 34: Evidence-compounding path from evaluation service to trusted infrastructure

Stage	Element	Analytical purpose
01	**Structure**	Normalize scenarios, evidence, findings, remediation, and decisions into versioned schemas.
02	**Calibrate**	Record reviewer guidance, examples, disagreement, and changes to adjudication rules.
03	**Connect**	Build traceability from requirement through execution to retest and launch posture.
04	**Govern**	Apply permissions, retention, competence, source, and public-claim controls.
05	**Publish**	Release only patterns whose denominator, provenance, and limitations can be defended.

This roadmap favors depth before scale. Premature automation can multiply weak labels, ambiguous criteria, and unreviewed assumptions. Premature benchmarking can convert a convenience sample into a market claim. Premature certification language can damage trust. A slower initial phase that produces rigorous evidence and learns where judgment is truly repeatable creates a stronger basis for software, research, and partnerships.

Research methodology

SOURCE POLICY

Primary and authoritative sources first.

The report prioritizes:

1. Government and regulator publications
2. Standards-body materials
3. Primary research papers
4. Intergovernmental guidance
5. Official product documentation for category mapping

Vendor materials are not used as independent proof of market leadership, customer outcome, or technical superiority. Secondary market estimates are excluded where category definitions are inconsistent.

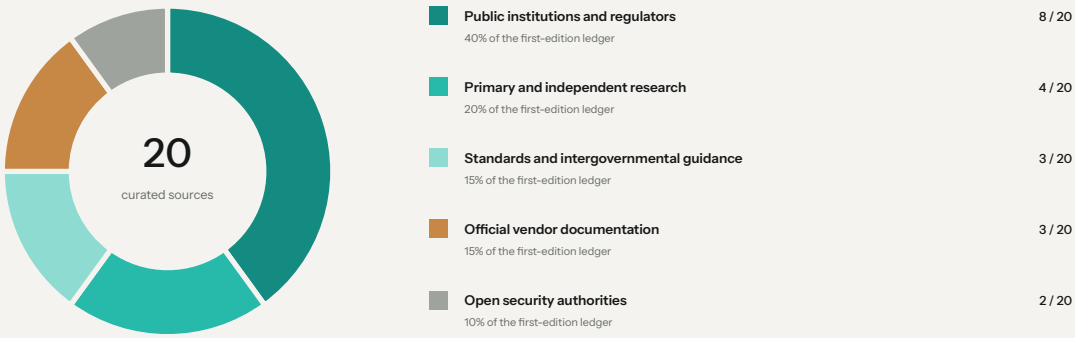
Table 35: Evidence classes, permitted uses, and required interpretive boundaries

Evidence class	Permitted use	Required boundary
Primary research	Technical findings and methods	Task, model, environment, and date
Government guidance	Framework and policy context	Voluntary vs mandatory status
Regulation	Legal context	Jurisdiction and qualified interpretation
Vendor documentation	Product positioning and capability	Vendor-reported, not independent validation

Research design

This report uses a structured narrative synthesis rather than a statistical meta-analysis. The relevant literature spans technical research, government frameworks, standards, security guidance, regulation, intergovernmental principles, and product documentation. These sources use different units of analysis and do not provide a common population, intervention, comparison, or outcome. Their quantitative results are therefore not pooled.

Sources were selected according to authority and relevance. Government and regulator publications were used for policy and lifecycle context. Standards-body material was used to distinguish management-system and conformity concepts. Primary research papers were used for benchmark methods and bounded empirical findings. OWASP materials were used for agentic security categories. Vendor documentation was used only to describe observability and evaluation product categories. Each source was recorded in a claim ledger with an intended use and explicit boundary.



Note: Counts describe the 20 curated sources used in this report. They do not measure the size, quality, or prevalence of the wider literature.
Source: Verune Labs source ledger, first edition, 31 July 2026.

Figure 11: Composition of the first-edition source ledger by authority group

The analytical method proceeded in four stages. First, source claims were extracted with their unit of analysis, method, environment, and limitation. Second, overlapping concepts were grouped into workflow evaluation, observability, red teaming, governance, audit, certification, and security. Third, the report identified gaps between these disciplines, particularly the absence of a complete decision chain from scenario context to remediation and retest. Fourth, the report developed the Verune workflow-evidence framework as a synthesis. Proposed taxonomies and diagrams are identified as Verune analytical constructs rather than empirical findings.

The report applies a conservative claim policy. A source supports only the statement that its method and publication justify. Benchmark results are not treated as production estimates. Regulatory and standards materials are not treated as proof of compliance. Vendor documentation is not treated as independent validation. Indonesia-specific discussion distinguishes ethics guidance and sector materials from generally applicable legal conclusions.

Quantitative figures were admitted only when the original source, measurement, and boundary could be identified. The adoption figures from the Stanford AI Index provide broad context, not agent-readiness prevalence. ToolEmu figures describe its study design and validation, not the failure rate of deployed agents. Tau-bench results illustrate repeated-trial reliability within its benchmark conditions, not a universal model ranking.

Analytical status of figures and tables

Figures in this report fall into two classes. Data figures reproduce a bounded quantitative relationship from an identified source and include the source beneath the figure. Analytical figures represent Verune's synthesis of a method, evidence sequence, or operating model; their captions and notes identify them as conceptual. Tables summarize sources, compare disciplines, define taxonomies, or specify proposed controls. They should not be interpreted as empirical datasets unless the caption and source explicitly state otherwise.

Every numbered figure and table is referenced as part of the report's argument. Captions state what the reader is viewing; surrounding prose explains why it matters. Source notes appear beneath data-based objects. Analytical objects use restrained lines and columns so the visual hierarchy does not imply a level of measurement that the underlying evidence cannot support.

Limitations

This report does not provide:

- a statistical estimate of global agent reliability
- a market size for AI agent assurance
- a legal analysis of any named deployment
- a certification or audit opinion
- a customer result
- a guarantee that one evaluation environment predicts production

Benchmark results cited here are illustrative of evaluation challenges. They are not directly comparable unless task, model, tool, policy, environment, evaluator, and date are aligned.

The regulatory environment continues to evolve. Readers should consult official sources and qualified advisers for current applicability.

The source set is deliberately selective rather than exhaustive. It emphasizes materials most relevant to workflow-level evaluation and the initial Verune thesis. It does not systematically review every agent benchmark, assurance provider, national AI policy, or sector-specific rule. Product categories and regulatory positions may change after the access date.

The proposed workflow framework has not yet been validated through a published multi-customer dataset. It should be treated as a transparent analytical and operating model to be tested and refined through bounded engagements. Future editions should document changes to scenario design, adjudication, reviewer calibration, and evidence quality as empirical experience accumulates.

The Indonesian discussion identifies operating variables and official materials relevant to initial scenario design, but it does not constitute a comprehensive study of Indonesia's languages, regions, sectors, or legal obligations. Domain owners, native-language reviewers, security specialists, and qualified advisers remain necessary where the workflow requires their authority.

Finally, no finite evaluation can prove universal safety or reliability. The relevant question is whether the selected evidence is sufficient for a stated decision under stated conditions. Readers should treat every claim as bounded by its source, method, environment, and date.

OBSERVE	EXPLAIN	BOUND	DECIDE
Execution fact	**Failure mechanism**	**Scoped conclusion**	**Owned implication**
What the preserved interaction, tool trace, and system state show.	Why the observed behavior diverged from the expected control.	Which scenario population, configuration, environment, and date the result supports.	What an accountable owner should launch, restrict, remediate, monitor, or investigate.

Note: The ladder does not support prevalence, universal reliability, legal compliance, or certification unless the corresponding method, population, authority, and evidence exist.

Figure 12: Claim ladder for moving from an observed execution to a bounded decision

References

PRIMARY AND AUTHORITATIVE SOURCES

[1] National Institute of Standards and Technology. "AI Resource Center." Accessed July 31, 2026. https://airc.nist.gov/

[2] National Institute of Standards and Technology. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile. NIST AI 600-1. 2024. https://doi.org/10.6028/NIST.AI.600-1

[3] National Institute of Standards and Technology. "AI Test, Evaluation, Validation and Verification." Accessed July 31, 2026. https://www.nist.gov/ai-test-evaluation-validation-and-verification-tevv

[4] National Institute of Standards and Technology. "Building Evaluation Probes into Agentic AI." Accessed July 31, 2026. https://www.nist.gov/programs-projects/building-evaluation-probes-agentic-ai

[5] International Organization for Standardization. ISO/IEC 42001:2023: Information Technology - Artificial Intelligence - Management System. Geneva: ISO, 2023. https://www.iso.org/standard/42001

[6] OWASP GenAI Security Project. "LLM06:2025 Excessive Agency." Accessed July 31, 2026. https://genai.owasp.org/llmrisk/llm062025-excessive-agency/

[7] OWASP GenAI Security Project. Agentic AI Threats and Mitigations. Accessed July 31, 2026. https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations/

[8] Yao, Shunyu, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. "Tau-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains." 2024. https://arxiv.org/abs/2406.12045

[9] Ruan, Yangjun, Honghua Dong, Andrew Wang, Silviu Pitis, Yongchao Zhou, Jimmy Ba, Yann Dubois, Chris J. Maddison, and Tatsunori Hashimoto. "Identifying the Risks of LM Agents with an LM-Emulated Sandbox." 2023. https://arxiv.org/abs/2309.15817

[10] Liu, Xiao, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. "AgentBench: Evaluating LLMs as Agents." 2023. https://arxiv.org/abs/2308.03688

[11] Stanford Institute for Human-Centered Artificial Intelligence. The 2025 AI Index Report. Stanford University, 2025. https://hai.stanford.edu/ai-index/2025-ai-index-report

[12] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence. Official Journal of the European Union, 2024. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32024R1689

[13] Organisation for Economic Co-operation and Development. "Advancing Accountability in AI." Accessed July 31, 2026. https://oecd.ai/en/accountability/

[14] United Nations Educational, Scientific and Cultural Organization. Recommendation on the Ethics of Artificial Intelligence. Paris: UNESCO, 2021. https://www.unesco.org/en/legal-affairs/recommendation-ethics-artificial-intelligence

[15] Kementerian Komunikasi dan Informatika Republik Indonesia. Surat Edaran Menteri Komunikasi dan Informatika Nomor 9 Tahun 2023 tentang Etika Kecerdasan Artifisial. Jakarta, 2023. https://jdih.komdigi.go.id/produk_hukum/unduh/id/883/

[16] Otoritas Jasa Keuangan. Tata Kelola Kecerdasan Artifisial Perbankan Indonesia. Jakarta: OJK, 2025. https://www.ojk.go.id/

[17] Otoritas Jasa Keuangan and Indonesian fintech associations. Panduan Kode Etik Kecerdasan Buatan yang Bertanggung Jawab dan Terpercaya di Industri Teknologi Finansial. Jakarta, 2023. https://www.ojk.go.id/

[18] LangChain. "LangSmith Evaluation." Product documentation. Accessed July 31, 2026. https://www.langchain.com/langsmith/evaluation

[19] Arize AI. "Phoenix: AI Observability and Evaluation." Product documentation. Accessed July 31, 2026. https://arize.com/docs/phoenix/

[20] Giskard. "LLM and Agent Evaluation." Product documentation. Accessed July 31, 2026. https://www.giskard.ai/products/llm-evaluation

Verune LABS

CLOSING POSITION

Bring the workflow that needs more than confidence.

Verune Labs evaluates whether consequential AI-agent workflows preserve their constraints, recover from failure, and support the decision their owner must make.

VERUNELABS.COM