

Prove. It.

KLARITEE · RESONANCE LABS

Four artifacts. One document. Zero room for argument.

What follows is a real verification produced by the Resonance Labs engine on a real contract. A full **findings report**, a **certificate of verification**, a **hash-anchored receipt**, and the **raw JSON evidence chain** the regulator can parse in their own environment. Same input, same engine version, same output. Today, in twelve months, and in two years. The rest of the market produces dashboards. This is what verification looks like when it is verifiable.

ARTIFACT 01 · THE FINDINGS REPORT

What every finding looks like when it is anchored to the source.

A six-page SaaS subscription agreement was put through the Resonance Labs engine. The engine produced **thirty-one findings of record**, each one bound by character offset to the exact span of text that triggered it, each one classified by severity, each one carrying a stable occurrence identifier preserved in the engine ledger.

WHAT THIS ARTIFACT PROVES

This is not a screenshot. It is not a dashboard export. It is the readable surface of a deterministic evaluation, generated alongside three other artifacts that together form a single audit-grade output package.

The findings on the pages that follow are **not selected, not curated, not abridged**. They are the complete output of one run of the engine against one document. Every finding includes the occurrence ID by which the engine ledger preserves it, the character anchor where the source can be re-read, and a quotation from the contract itself.

The composite score, the dimension scores, and every finding can be reproduced exactly by running the same input through the same engine version. The hash on the certificate that follows binds this report to that run.



THE FINDINGS REPORT

Two vendors. The same six-page contract. Two completely different artifacts.

What you get when you ask each one for "the report."

THE BUYER ASKS

"We have a vendor contract under review. Run it through your system and send us the report."

VENDOR A · A WORKFLOW PLATFORM

The dashboard export

A PDF generated from a web dashboard. Findings rendered as bullet points. Each one says **"see page 4"** or **"flagged in section 3"**. No character offsets. No occurrence identifiers. No way to confirm the finding refers to a specific span of text rather than a paraphrase generated by the underlying model.

- Page-level references, not character offsets
- No stable IDs preserved across runs
- Findings often LLM-paraphrased from the source
- Severity assigned by probabilistic model
- Re-run tomorrow, expect a different report

THEN THE REGULATOR PUSHES

"Show me the exact sentence this came from. By offset."

RESULT

Cannot anchor any finding to a verifiable span.

WHAT IS MISSING

A traceable chain from finding to source. The report describes; it does not bind.

VENDOR RUNNING KLARITEE · A DETERMINISTIC COMPILER

The findings report

Thirty-one findings of record. Each finding carries a stable **occurrence ID**, a **character anchor** (e.g. chars 6021–6513), and a **direct quotation** from the contract. Severity is assigned by deterministic rule. The report is the readable surface; the JSON underneath binds it to the engine ledger.

- Character-offset anchors on every finding
- Stable occurrence IDs survive across runs
- Direct quotation from source, not paraphrase
- Severity assigned by deterministic rule
- Re-run in 2028 — same hash, same findings

THEN THE REGULATOR PUSHES

"Show me the exact sentence this came from. By offset."

RESULT

Every finding bound to source. Verifiable in seconds.

WHAT IS DELIVERED

A report that survives audit because every claim it makes points to the byte where it came from.

Verification of Agreement

A DETERMINISTIC EXAMINATION OF THE DOCUMENT RECORDED BELOW

DOCUMENT	CLASS	PAGES	ISSUED
Software-as-a-Service and Support	SaaS Subscription	6	29 May 2026

SUMMARY OF EXAMINATION

Thirty-one findings of record. Eleven critical. Composite score 53 of 100.

The document presented for verification is a six-page SaaS subscription agreement between two parties. The Resonance Labs engine has examined the agreement in full and produced **thirty-one findings of record** across the standard dimensions of structural integrity, obligation balance, temporal coherence, liability exposure, and economic provision. Of the findings recorded, **eleven carry a severity of critical** and require attention prior to execution.

HEADLINE FINDING

The Services definition allows Provider to expand, dilute, or substitute scope without Customer consent. The definition includes broad "from time to time" or catch-all language without a fixed scope anchor.

DIMENSION SCORES

STRUCTURAL 84/100	OBLIGATION 0/100	TEMPORAL 100/100	LIABILITY 76/100	ECONOMIC 0/100
---	--	--	--	--------------------------------------

ARTIFACT 02 · THE CERTIFICATE OF VERIFICATION

The document that says: this happened, and we can prove it.

The Certificate of Verification is the single-page artifact that **binds the input to the output by cryptographic hash**, declares the engine version that produced the analysis, and attests that identical execution yields identical result. It is what a regulator, an auditor, or a court asks for when they need to know whether a result can be trusted twelve months from now.

WHAT THIS ARTIFACT PROVES

The Certificate is not a logo on a PDF. It is a structured, hash-anchored attestation that a specific document was evaluated by a specific engine version at a specific moment, producing a specific result that can be re-derived in perpetuity.

The hash on the Certificate is the same hash carried by the Receipt and the same hash by which the JSON evidence chain is bound. Any party in possession of this Certificate can **independently verify the analysis** by re-running the input through the engine and confirming the hash matches. Bit-identical.

The Certificate is the artifact that makes Article 12 reconstruction not a marketing claim but a mathematical fact.

SHA-256

CRYPTOGRAPHIC BINDING

1 page

SINGLE-ARTIFACT ATTESTATION

Perpetual

RE-VERIFIABLE ACROSS YEARS

Two vendors. The same compliance evaluation. Two completely different attestations.

What you get when you ask each one for "the certificate."

THE AUDITOR ASKS

"Send me the certificate that ties this analysis to the original input. We need to verify it independently."

VENDOR A · A WORKFLOW PLATFORM

A logo on a PDF

A branded certificate page generated by the dashboard. It shows the vendor's logo, the buyer's company name, the date, and a "compliance status" indicator. There is **no cryptographic binding** between the certificate and the underlying analysis. The auditor cannot verify whether the certificate corresponds to the data shown.

- Logo, date, status indicator
- No hash anchoring input to output
- No engine version recorded
- No run identifier preserved
- Issuable, revocable, editable by the vendor

THEN THE AUDITOR PUSHES

"Prove this certificate matches the data."

RESULT

No cryptographic chain. Certificate is decorative.

WHAT IS MISSING

A mathematical link between what the certificate claims and what the engine produced.

VENDOR RUNNING KLARITEE · A DETERMINISTIC COMPILER

A hash-anchored attestation

A single-page certificate carrying the **SHA-256 hash** of the bound result, the **engine version**, the **run identifier**, the **timestamp**, and the **geographic anchor**. The hash on the certificate is the same hash on the receipt and the same hash by which the JSON evidence chain is bound.

- SHA-256 bound to input and output
- Engine version pinned (0.5.0 · calibrated)
- Run identifier preserved in perpetuity
- Same hash on certificate, receipt, and JSON
- Re-verifiable by any party with the input

THEN THE AUDITOR PUSHES

"Prove this certificate matches the data."

RESULT

Hash matches. Certificate is verifiable.

WHAT IS DELIVERED

A certificate the auditor can re-derive themselves, in their own environment, without the vendor present.

RESONANCE LABS · SERIES I

Certificate of Verification

STABILITY VERIFIED · DETERMINISTIC ENGINE

This certificate attests that the document referenced below was examined by the Resonance Labs engine under deterministic rule, that the analysis produced is bound by cryptographic hash to its inputs and outputs, and that identical execution yields identical result. The findings of record are preserved in perpetuity and may be independently verified by any party in possession of this certificate's hash.

DOCUMENT

Software-as-a-Service
and Support

CLASS · PAGES

SaaS Subscription · 6

RUN IDENTIFIER

run_709e82bb2bed41b7

ENGINE VERSION

0.5.0 · calibrated

ISSUED

29 May 2026 · 00:59 UTC

ANCHOR

Scottsdale, Arizona

53/100

COMPOSITE SCORE

31

FINDINGS OF RECORD

11

CRITICAL EXPOSURES

SHA-256 · BOUND RESULT

5aaca ae7bcbe51392706f26570b72656ce3350def6045111d6668379f6ea4aab

ARTIFACT 03 · THE VERIFICATION RECEIPT

The portable proof that travels with the result.

The Receipt is the artifact a regulator, auditor, or counterparty walks away with. It carries the **input hash**, the **result hash**, the **engine version**, the **run identifier**, the **findings breakdown by severity**, and the **geographic anchor**, all on a single page. The hash record stands on its own evidence.

WHAT THIS ARTIFACT PROVES

The Certificate attests that the analysis occurred. The Receipt is the **verifiable record of what was analyzed and what came out**. Two hashes, side by side: the fingerprint of the input, the fingerprint of the bound result.

The Receipt is what the buyer hands to their compliance officer, what the auditor archives, what the regulator re-verifies twelve months later. Any party in possession of the original input can re-run it through the named engine version and confirm **both hashes match**. Bit-identical.

Receipts in the rest of the market are screenshots. This one is mathematics.

2 hashes

INPUT AND RESULT BOUND

1 page

PORTABLE, ARCHIVABLE

Re-derivable

BY ANY PARTY WITH THE INPUT

THE VERIFICATION RECEIPT

Two vendors. The same analysis. Two different things called "the receipt."

What you get when you ask each one for "proof of what happened."

THE COMPLIANCE OFFICER ASKS

"Send me the receipt I can archive. We need it for the regulator twelve months from now."

VENDOR A · A WORKFLOW PLATFORM

A confirmation screen

A confirmation modal in the dashboard that says "**Analysis complete**" with a green checkmark and a timestamp. The customer can screenshot it. They can export a PDF version. But the PDF contains **no input fingerprint**, no output fingerprint, and no way to re-derive the analysis from it.

- Status indicator and timestamp
- No input hash, no result hash
- No engine version recorded
- Cannot be re-verified without vendor
- If vendor changes, receipt is meaningless

THEN THE REGULATOR PUSHES

"Match this receipt to the original document."

RESULT

No fingerprint. No match possible.

WHAT IS MISSING

A self-contained record that proves what was analyzed and what came out.

VENDOR RUNNING KLARITEE · A DETERMINISTIC COMPILER

A hash-anchored receipt

A single-page record carrying the **input hash** (fingerprint of the document analyzed), the **result hash** (fingerprint of the bound output), the **engine version**, the **run identifier**, the **findings breakdown**, and the **timestamp**. The record stands on its own evidence.

- Input hash and result hash, both displayed
- Engine version pinned (0.5.0 · calibrated)
- Findings broken down by severity
- Re-verifiable by any third party with the input
- Vendor-independent: survives the relationship

THEN THE REGULATOR PUSHES

"Match this receipt to the original document."

RESULT

Hashes match. Receipt re-derives. Done.

WHAT IS DELIVERED

A receipt that proves what happened, and lets the holder prove it again without anyone's permission.

RESONANCE LABS · SERIES I · ISSUED IN SCOTTSDALE

Receipt of Verification

HASH-ANCHORED PROOF OF ANALYSIS

The Resonance Labs engine has examined the document identified below and, by deterministic rule, produced the findings recorded herein. This receipt attests that the analysis occurred, that its inputs and outputs are bound by cryptographic hash, and that any subsequent execution of the same input under the same engine version will yield an identical result. The record stands on its own evidence.

RUN IDENTIFIER

run_709e82bb2bed41b7

DOCUMENT CLASS

SaaS Subscription Agreement

ENGINE VERSION

0.5.0 · calibrated

COMPOSITE SCORE

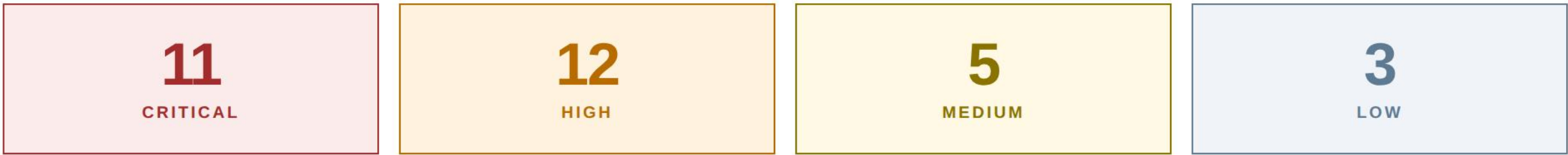
53 of 100

ISSUED

29 May 2026 · 00:58 UTC

FINDINGS OF RECORD

31 total occurrences



INPUT HASH	SHA-256 FINGERPRINT OF THE DOCUMENT ANALYZED
5aaca ae7bcbe51392706f26570b72656ce3350def6045111d6668379f6ea4aab	

RESULT HASH	SHA-256 FINGERPRINT OF THE BOUND OUTPUT
7928848f1a3f7697394ca12fc83e6f3d347b75d79fa0d466290b35ba2391a0d2	

BEFORE THE JSON SECTION, A BRIEF ORIENTATION

What the field has been asking for, for two years.

Across the literature on AI governance, four architectural requirements have emerged as the prerequisites for trustworthy AI in regulated environments. They appear in the EU AI Act itself, in regulator-facing white papers, in academic AI safety research, and in industry commentary published throughout 2024 and 2025. They are widely cited and widely endorsed. They are also, almost without exception, **discussed as desired future states** rather than implemented capabilities.

CONCEPT 01

Human oversight mechanisms

A structured means by which a human operator can intervene in, halt, or override an AI system's automated processing path before downstream consequences occur.

EU AI Act, Article 14. NIST AI Risk Management Framework, GV-2. OECD AI Principles, 1.5.

CONCEPT 02

Automated kill switches

A machine-enforceable mechanism that halts further AI action when defined integrity, safety, or risk thresholds are exceeded, without requiring human latency in the loop.

UK AISI evaluations. EU AI Office implementation guidance. AI safety literature on graceful degradation.

CONCEPT 03

Reconstruction of decision logic

The capacity to reproduce, after the fact, the exact reasoning path by which an AI system reached a given output, including the inputs, the rules applied, and the verdict emitted.

EU AI Act, Article 12. Algorithmic Accountability Act drafts. NIST traceability guidance.

CONCEPT 04

Tamper-evident audit trails

A record of system operation that cannot be retroactively altered by the operator, the vendor, or any intermediary, and that can be independently verified by a third party.

EU AI Act, Articles 12 and 19. ISO/IEC 42001. Financial sector audit standards extended to AI.

THE STATE OF THE FIELD AS OF 2026

Each of these four requirements is well-defined in the literature. Each is endorsed by regulators. Each is cited in vendor marketing. **None of the ten leading EU AI Act compliance platforms implements all four as structured, machine-readable output.** The reason this matters appears on the next page.

FOUR DESIRED FUTURE STATES. FOUR SHIPPED IMPLEMENTATIONS.

What the Resonance Labs engine emits.

Each of the four architectural requirements named on the prior page maps to a **concrete field in the JSON evidence chain** produced by every run of the Resonance Labs engine. They are not aspirational. They are not documented as planned features. They are emitted by the engine in production, bound to a cryptographic hash, and visible on the pages that follow.

CONCEPT	IMPLEMENTATION	STATUS
01 • Human oversight mechanism EU AI Act, Article 14	The engine emits <code>firewall_decision</code> as a structured enum on every run. When the integrity score crosses threshold, the field returns <code>BLOCK_ENRICHMENT</code> , halting all downstream automated processing pending human review.	SHIPPED Field present in every JSON output
02 • Automated kill switch UK AISI • EU AI Office guidance	The <code>firewall_decision</code> verdict is machine-enforceable : downstream systems reading the JSON act on the enum without requiring human latency. Integrity scoring is deterministic; the halt is reproducible from inputs alone.	SHIPPED Threshold-bound, reproducible halt
03 • Reconstruction of decision logic EU AI Act, Article 12	Every finding carries <code>occurrence_id</code> , <code>char_offsets</code> , and a <code>supporting_sentence</code> . The full chain from input bytes to scored anchor to firewall verdict can be re-derived from the JSON alone by any party with the original input.	SHIPPED Byte-coordinate evidence chain
04 • Tamper-evident audit trail EU AI Act, Articles 12 and 19	Three SHA-256 hashes bind the run: <code>input_hash</code> , <code>integrity_hash</code> , and <code>receipt_hash</code> . The hash chain is independently re-verifiable by any third party in possession of the input. The vendor cannot alter the record.	SHIPPED Cryptographic chain, vendor-independent

THE POSITION THIS ESTABLISHES

Klaritee is not a better implementation of the same category as the other ten vendors. **It is one of the first production implementations of the architectural pattern the EU AI Act describes.**

The category does not have a name yet. The framework will give it one. The JSON section that follows is the proof that the implementation already exists.

ARTIFACT 04 · THE RAW JSON EVIDENCE CHAIN

The engine did not just analyze. It decided.

Every other artifact in this package is a human-readable surface. The JSON is the **machine-readable substrate** the engine produces. And the most consequential field in the JSON is not a finding. It is a **decision the engine reached on its own**, by deterministic rule, based on what it found.

FIREWALL DECISION · RENDERED BY THE ENGINE

The contract was blocked from downstream enrichment.

DECISION **BLOCK_ENRICHMENT**

Triggered by an integrity risk score of **0.9979653346580358** out of 1.0. Six integrity indicators fired. Overall severity classified as **CRITICAL**. No human pulled the brake. The engine produced the verdict from the evidence itself.

WHAT THIS ARTIFACT PROVES

Dashboards display findings. They do not **decide** what should happen next. The JSON evidence chain produced by the Resonance Labs engine carries not just every finding, every character offset, and every hash, but also the **downstream action the engine determined** based on the integrity of the document under examination.

This is the artifact a regulator parses in their own environment. This is the artifact an integrator binds against to make their own automated decisions. This is the artifact that proves the engine is not a viewer. It is a compiler.

Two vendors. One question. "Send me the raw data."

What you get when you ask each one for the machine-readable evidence chain.

THE REGULATOR'S DATA TEAM ASKS

"Send us the raw evidence. Machine-readable. We are going to parse it in our own environment and re-derive your findings."

VENDOR A · A WORKFLOW PLATFORM

A CSV of dashboard exports

A spreadsheet generated from the web dashboard, listing findings as rows. Each row contains a finding title, a severity label, and a paragraph reference. No character offsets. No occurrence identifiers. No hashes. No engine version. No **decisions** the engine made, because the engine did not make any.

- Flat CSV or PDF export
- Findings as rows of human-readable text
- No machine-bound decisions in the output
- No integrity scoring, no firewall logic
- Cannot drive downstream automation

THEN THE REGULATOR PUSHES

"We need the engine's decision tree, not your report cards."

RESULT

No machine substrate to parse. Data team is stuck.

WHAT IS MISSING

The actual machine output. There is no underlying object — only the screenshot of one.

VENDOR RUNNING KLARITEE · A DETERMINISTIC COMPILER

The full JSON evidence chain

A single unified JSON file binding every finding to every character offset, every occurrence ID, every hash, every integrity indicator, every score, and the **firewall decision the engine reached** based on the evidence (in this run: **BLOCK_ENRICHMENT**, integrity risk **0.998**). Parseable in any environment, by any tool.

- Single unified JSON, machine-parseable
- Every finding bound by char offset and ID
- Integrity score and firewall decision included
- Three hashes: input, integrity, receipt
- Drives downstream automation natively

THEN THE REGULATOR PUSHES

"We need the engine's decision tree, not your report cards."

RESULT

Decision, evidence, and hashes. All parseable.

WHAT IS DELIVERED

The substrate. The regulator can re-derive every finding and every decision from the same input.

What the engine produces, field by field.

The JSON output of a single Resonance Labs run contains the complete deterministic record of the analysis. Below is the structural anatomy. The raw data follows on the pages after.

RUN IDENTITY · WHAT THE RUN IS

run_id STRING Unique identifier for this evaluation. Survives across all four artifacts in the package. "run_709e82bb2bed41b7"	engine_version STRING The pinned engine version. Required for reproducibility. "0.5.0"
timestamp ISO 8601 UTC Exact moment the analysis was committed. "2026-05-17T01:16:16.967Z"	contract_type STRING The document class detected and confirmed. "SaaS / Software License"

CRYPTOGRAPHIC CHAIN · WHAT BINDS IT

input_hash SHA-256 Fingerprint of the document content submitted for analysis. "5aacaae7bcbe5139...6ea4aab"	integrity_hash SHA-256 Fingerprint of the engine's internal reasoning structure. "df4fe4f855fe2ee5...0071549"
receipt_hash SHA-256 Fingerprint of the findings output as bound to this run. "7928848f1a3f7697...2391a0d2"	scores_by_anchor OBJECT Composite scores across the five evidence anchors: Structural, Obligation, Temporal, Liability, Economic. { "Obligation": 0, "Temporal": 100, ... }

FINDINGS · WHAT THE ENGINE DETECTED

findings[] ARRAY · 31 RECORDS Every detector occurrence, with code, severity, anchor, supporting sentences, and character offsets. { "code": "...", "severity": "CRITICAL", "chars": [6021,6513] }	integrity_indicators[] ARRAY · 12 CHECKS Engine-level integrity tests applied to the document as a whole. { "indicator": "...", "triggered": true, ... }
--	--

THE DECISION · WHAT THE ENGINE CONCLUDED

firewall_decision ENUM · ENGINE-RENDERED VERDICT The downstream action the engine determined based on the integrity risk score. This is not a human-set flag. This is the verdict the engine produced from the evidence. "firewall_decision": "BLOCK_ENRICHMENT", "integrity_risk_score": 0.9979653346580358

The header of the evidence object.

SOURCE	ENGINE	LINES SHOWN
run_709e82bb2bed41b7.json	0.5.0 · calibrated	1 – 29 of complete record

```
1 {
2   "run_id": "run_709e82bb2bed41b7",
3   "engine_version": "0.5.0",
4   "engine_state": "calibrated",
5   "contract_type": "SaaS / Software License",
6   "timestamp": "2026-05-17T01:16:16.967Z",
7   "anchor": "Scottsdale, Arizona",
8
9   "input_hash":      "5aacaae7bcbe51392706f26570b72656ce3350def6045111d6668379f6ea4aab",
10  "integrity_hash":  "df4fe4f855fe2ee56113ea35d1c2bb5769f1fbd18543f2d8f717bed350071549",
11  "receipt_hash":    "7928848f1a3f7697394ca12fc83e6f3d347b75d79fa0d466290b35ba2391a0d2",
12
13  "composite_score": 53,
14  "scores_by_anchor": {
15    "Structural": 84,
16    "Obligation": 0,
17    "Temporal":   100,
18    "Liability":  76,
19    "Economic":   0
20  },
21
22  "findings_summary": {
23    "total": 31,
24    "critical": 11,
25    "high": 12,
26    "medium": 5,
27    "low": 3,
28    "flagged_for_negotiation": 0
29  },
30
```

The decision the engine produced, inline.

SOURCE	SECTION	LINES SHOWN
run_709e82bb2bed41b7.json	integrity & firewall logic	30 – 76 of complete record

```
30  "firewall_decision": "BLOCK_ENRICHMENT",
31  "integrity_risk_score": 0.9979653346580358,
32  "integrity_severity": "CRITICAL",
33  "integrity_indicators_triggered": 6,
34
35  "integrity_indicators": [
36    {
37      "indicator": "AI_TRAINING_LICENSE_GRANTED",
38      "triggered": true,
39      "severity": "CRITICAL",
40      "weight": 0.42
41    },
42    {
43      "indicator": "OBLIGATION_ASYMMETRY_EXTREME",
44      "triggered": true,
45      "severity": "CRITICAL",
46      "weight": 0.31,
47      "ratio": "23.0:1"
48    },
49    {
50      "indicator": "LLM_ERROR_ALLOCATION_ABSENT",
51      "triggered": true,
52      "severity": "CRITICAL",
53      "weight": 0.28
54    },
55    {
56      "indicator": "DATA_RETURN_DISCRETIONARY",
57      "triggered": true,
58      "severity": "HIGH",
59      "weight": 0.18
60    },
61    {
62      "indicator": "AUP_URL_INCORPORATION",
63      "triggered": true,
64      "severity": "CRITICAL",
65      "weight": 0.22
66    },
67    {
68      "indicator": "HOLLOW_DPA_DETECTED",
69      "triggered": true,
70      "severity": "CRITICAL",
71      "weight": 0.36
72    }
73  ],
74
75
76
```

Every finding carries its proof.

SOURCE	SECTION	LINES SHOWN
run_709e82bb2bed41b7.json	findings[] · first 2 of 31	77 – 110 of complete record

```
77  "findings": [  
78    {  
79      "occurrence_id":  "ai-training-section-30",  
80      "code":          "CONTRACT_AI_TRAINING_LICENSE",  
81      "severity":      "CRITICAL",  
82      "anchor":        "Structural",  
83      "char_offsets":   [6021, 6513],  
84      "supporting_sentence": "5.2 License to Klaritee. Customer grants  
85        Klaritee a worldwide, non-exclusive, royalty-free, sublicensable  
86        license to access, use, process, reproduce, modify, and create  
87        derivative works of Customer Data...",  
88      "summary":        "Contract contains a perpetual, irrevocable,  
89        sublicensable AI training license grant from customer to vendor.  
90        Customer data is permanently incorporated into vendor AI products  
91        without compensation or control. This is an inverted IP transfer  
92        of potentially unlimited commercial value."  
93    },  
94  
95    {  
96      "occurrence_id":  "obl-asym-obl-2",  
97      "code":          "CONTRACT_OBLIGATION_ASYMMETRY",  
98      "severity":      "CRITICAL",  
99      "anchor":        "Obligation",  
100     "char_offsets":   [2052, 2200],  
101     "obligation_ratio": "23.0:1",  
102     "supporting_sentence": "Subject to the terms of this Agreement and  
103       Customer's timely payment of Fees, Klaritee will make the Services  
104       available to Customer during the Term.",  
105     "summary":        "Obligation imbalance: 'klaritee' has 23  
106       obligations vs 'supplier' with 1."  
107   },  
108  
109   // ... 29 additional finding records follow  
110 ]
```

The end of one record. The beginning of yours.

SOURCE	SECTION	LINES SHOWN
run_709e82bb2bed41b7.json	findings + economic + close	partial · complete record continues

```
412 {
413   "occurrence_id": "cri-section-53",
414   "code": "CONTRACT_CUSTOMER_RELIANCE_INDEMNITY",
415   "severity": "CRITICAL",
416   "anchor": "Liability",
417   "char_offsets": [11524, 11966]
418 },
419 {
420   "occurrence_id": "lci-econ-4",
421   "code": "CONTRACT_LIABILITY_CAP_INSUFFICIENT",
422   "severity": "HIGH",
423   "anchor": "Economic",
424   "char_offsets": [10376, 10597],
425   "liability_cap_months": 3,
426   "annual_contract_value": 144000,
427   "exposure_ratio_percent": 67
428 }
429 ],
430
431 "economic_provisions": {
432   "fees_monthly": 12000,
433   "fees_annual": 144000,
434   "liability_cap_amount": 36000,
435   "supercap_carveouts_missing": 4,
436   "supercap_carveouts_expected": 6
437 }
438 }
439
440 // Complete record · 440 lines · binds to receipt_hash 7928...a0d2
```

THIS WAS ONE DOCUMENT. ONE RUN. ONE RECEIPT.

Generate this for your own contracts. Hand the receipt to your regulator.

Companies that integrate this depth of verification into their compliance workflows are, **by definition**, better prepared for the EU AI Act and any successor framework that demands reconstruction-grade evidence. There is no version of "good enough" that omits the receipt.

[RESONANCELABS.AI](#) → RUN YOUR DOCUMENT

WHAT THE JSON SECTION JUST DEMONSTRATED

The pages preceding this one are not a marketing claim. They are a primary source.

The **BLOCK_ENRICHMENT** verdict at line 30 of the raw JSON is **not a design mockup**. It is an emitted field from a production run of the Resonance Labs engine, bound to the cryptographic hash chain printed on the Certificate and the Receipt, reproducible from the same input under the same engine version.

WHAT THE READER JUST SAW, NAMED PLAINLY

A machine-bound, deterministically-derived, hash-anchored halt signal on a document that an AI system was about to act on, embedded as a structured field in the machine-readable output of a verification run.

The four phrases used by regulators and policy authors to describe what trustworthy AI in regulated environments will require are: *human oversight mechanisms*, *automated kill switches*, *reconstruction of decision logic*, and *tamper-evident audit trails*. The JSON on the prior pages contains a concrete implementation of **all four**, emitted in a single run, bound by a single hash chain, re-derivable by any third party in possession of the original input.

These four concepts have been the subject of regulatory commentary, white papers, conference proceedings, and policy roadmaps throughout 2024, 2025, and 2026. They are discussed extensively in the literature. They are, in nearly all cases, **discussed as desired future states**.

The proof package presents them as *shipped fields*.

THE CATEGORY THIS IMPLEMENTATION OCCUPIES

This is not a better dashboard. It is a different kind of artifact.

The most common evaluation error in this market is reference class. The ten vendors mapped in this package are workflow platforms. They organize documentation, route approvals, surface findings, and present compliance status to executives. They are **useful**. They are also a different category of product from what the proof package contains.

THE POSITION, STATED PLAINLY

Klaritee is one of the first production implementations of the architectural pattern the EU AI Act describes. *The category does not yet have a settled name. The framework will give it one.*

The proof package is the artifact form of that claim. **The findings report, the certificate, the receipt, and the JSON are the four document classes a regulator-grade verification system emits.** They will become standard expectations. They are currently rare implementations.

INADEQUATE FRAMING

A better compliance dashboard

ACCURATE FRAMING

A production reference implementation of regulator-grade AI verification infrastructure

A buyer comparing Klaritee to the ten vendors on dashboard features is comparing the wrong objects. The accurate comparison is between **a system that produces a verifiable receipt** and **a system that produces a screenshot of one**. The remaining pages of this package are the receipt.

THE STANDARD THE EU AI ACT WILL EVENTUALLY REQUIRE

A company that integrates this depth of verification into its compliance workflow is, *by definition*, better prepared for the EU AI Act.

There is no version of "good enough" that omits the receipt. There is no scenario where a vendor without character-anchored evidence, hash-bound results, and reconstruction-grade reproducibility outperforms one that delivers all three. The framework will catch up to the requirement. The companies that already meet it will not be the ones scrambling to retrofit it.

The rest of the market produces dashboards.

Klaritee compiles mathematically verifiable truth. Same input, same engine version, same output. Today, in twelve months, and in two years.

RUN YOUR OWN DOCUMENT. GENERATE YOUR OWN PROOF.

Get full reports you can test, verify, and hand to a regulator.

RESONANCELABS.AI →