Decision Receipts: A Verifiable Primitive for AI Evidence That Survives a Challenge

Authors: Anya Chueayen and the Aqta Research Team **Affiliation:** Aqta Technologies, Dublin
Corresponding author: hello@aqta.ai

Abstract

Regulators across jurisdictions increasingly expect that an organisation deploying an AI system be able to reconstruct, after the fact, what the system decided, on what inputs, under what policy, at what time, and signed by whom, in a form an external party can examine. No regime in force today specifies a cryptographic form for that record, and none compels one; this paper proposes a form that would survive an external party declining to trust the organisation under investigation. We propose ATTESTATION-v1, an open specification for a *decision-receipt primitive* that satisfies the seven evidence properties required or implicit across seven regulatory frameworks (EU AI Act, EU DORA, US NIST AI RMF, US SR 11–7, UK FCA Consumer Duty, UK ICO under UK GDPR, Singapore PDPC MAGO + AI Verify). We describe a reference implementation with a published verifying key, and we test the primitive against a worked field deployment: a public commitment that ranked a specific tile for Ebola eight days before the World Health Organization declared a Public Health Emergency of International Concern for the matching outbreak. We publish the full denominator of forward bets alongside the single positive, and report a bootstrap analysis that finds the hit consistent with a uniform-over-tiles null at $p = 0.222$. The paper's contribution is the primitive, not a claim of predictive capability. We additionally name four operational requirements for high-assurance deployments and two genuinely-novel open-problem framings (audit-cost economics at bank scale, and receipt-differential-privacy interaction).

Section 1. Introduction

When a regulator, an internal auditor, or a court asks an organisation to explain what an AI system decided and why, the organisation must reconstruct that specific decision after the fact. The technical bar is not a record of *which model was invoked*: it is a record of *what was decided, on what inputs, under what policy, at what time, and signed by whom*, in a form that the asking party can inspect independently of the organisation under investigation. We call this per-decision, signed artefact a *decision receipt*, and we propose ATTESTATION-v1 as an open specification for it. This bar is increasingly explicit in regulation. The European Union's Artificial Intelligence Act, Regulation 2024/1689, requires in Article 12 that high-risk AI systems "automatically record events" produced during their operation, and in Article 26 places parallel duties on the deployer to retain those records [1]. The Digital Operational Resilience Act, Regulation 2022/2554, has imposed analogous reconstructability obligations on financial services since 17 January 2025 [2]. The same shape of requirement, with different statutory wording, is now present in the United States National Institute of Standards and Technology's AI Risk Management Framework (Govern function, in particular Govern 1.6 inventory and post-deployment-monitoring subcategories) [3], the Federal Reserve's Supervisory Letter SR 11–7 on Model Risk Management [4], the Colorado Consumer Protections for Artificial Intelligence Act (Senate Bill 24–205, repealed and replaced by SB 26–189 before operative effect) [5], the United Kingdom Financial Conduct Authority's Consumer Duty (PS22/9) [6] and the Information Commissioner's Office guidance on AI auditing [7], and the Personal Data Protection Commission of Singapore's Model AI Governance Framework paired with the AI Verify Foundation toolkit [8].

The frameworks differ in statutory form, enforcement timetable, and which property is named load-bearing. They converge on a single technical primitive: a per-decision record that is timestamped, payload-complete, identity-bound to a specific model, lineage-aware over the policy or rule set that ran, third-party verifiable, post-event inspectable, and tamper-evident. We refer to this primitive throughout the paper as a *decision receipt*. Section 2 maps seven regulatory frameworks against the seven properties just listed and shows that every framework requires the same intersection.

The technologies in current production do not deliver this primitive. Cloud-provider audit logs (AWS CloudTrail, Amazon Bedrock invocation logging, Google Cloud Vertex AI audit logs, Azure Monitor) record *that a model was invoked* and are scoped at the level of API call, not decision; their trust model assumes the vendor that operates them [9][10][11]. Developer-oriented observability tooling (LangSmith, Helicone, Phoenix and comparable platforms) is optimised for prompt debugging and cost analysis; the records are neither cryptographically signed nor structured for inspection by a party outside the deploying organisation [12][13]. The W3C Verifiable Credentials

data model addresses a different shape of problem: it issues credentials about subjects, rather than recording transactions [14]. Trusted Execution Environments such as AWS Nitro Enclaves or AMD SEV-SNP can attest the *runtime* in which a model executed, but a runtime attestation does not by itself produce a record of any particular decision [15][16]. None of the existing instruments is wrong; each addresses an adjacent question. None is the decision receipt described here.

This paper proposes a specification for the missing primitive, demonstrates a reference implementation with a published verifying key, and tests the primitive against a worked field deployment in which the receipt's behaviour can be checked against a dated external event. Specifically, the paper contributes:

1. A cross-jurisdiction survey mapping seven regulatory frameworks to a unified set of evidence properties for decision receipts (Section 2, Table 1).
2. ATTESTATION-v1, an open specification (Apache 2.0 and CC BY 4.0) for the decision-receipt primitive: a canonical-JSON-shaped record carrying a SHA-256 content hash and an Ed25519 signature, indexed into a per-organisation hash chain (Section 3) [17].
3. A reference implementation, the Seal gateway, with a published verifying key (Section 4). The gateway is named for reproducibility of Section 5's worked example, not as the contribution.
4. A worked field deployment of receipt-emitting infrastructure on an outbreak-prediction agent, in which a dated public git commit recorded a top-five-ranked prediction for an Ebola outbreak in the Congo Basin eight days before the World Health Organization declared a Public Health Emergency of International Concern for the same outbreak (Section 5).
5. *Anti-survivorship-bias accounting* for the worked deployment: the full denominator of dated public predictions made in the relevant evaluation windows, the prior probability per region per pathogen-week derived from a decade of WHO Disease Outbreak News data, and the hit-miss-pending breakdown against WHO, ECDC, and national MoH notification records (Section 5.5). To our knowledge no other vendor-issued paper on AI decision evidence publishes its own denominator alongside a single-positive worked example.
6. *Four operational requirements* that high-assurance deployments must satisfy on top of the receipt primitive itself: output-binding evidence, external input-hash timestamping, independent policy-engine signing, and a documented key-management migration plan covering threshold signing and post-quantum schemes (Section 7.5). We name these requirements explicitly so that a regulator reading the paper can demand each by name in a supervisory review.
7. Two open-problem statements with concrete framing: audit-cost economics for LLM-decision logs at regulated-buyer scale (Section 8.6), and the interaction between auditor-readable receipts and differential-privacy guarantees on the underlying model (Section 8.7).

The remainder of the paper is organised as follows. Section 2 surveys the regulatory landscape and presents the cross-jurisdiction mapping. Section 3 defines the decision-receipt primitive and the ATTESTATION-v1 specification. Section 4 describes the reference implementation. Section 5 presents the worked field deployment and the bias accounting. Section 6 maps receipt fields to specific regulatory clauses. Section 7 compares the primitive to adjacent instruments and lists the operational requirements for high-assurance deployments. Section 8 names six open problems. Section 9 concludes.

Section 2. The regulatory landscape

2.1. European Union

The European Union's Artificial Intelligence Act, Regulation 2024/1689, is the most prescriptive AI-specific instrument in force. Three articles are directly relevant to the decision-receipt primitive. **Article 12 ("Record-keeping")** requires that high-risk AI systems be designed to "automatically record events" during their operation throughout their lifetime, and specifies that the logs must "ensure the traceability of the system's functioning" at a level appropriate to the intended purpose [18]. **Article 13 ("Transparency and provision of information to deployers")** requires that high-risk systems be accompanied by information sufficient for the deployer to interpret the system's output and use it appropriately [19]. **Article 26 ("Obligations of deployers of high-risk AI systems")** places parallel duties on the deployer to keep the logs generated by the system to the extent the logs are under the deployer's control, for an appropriate period [20]. Together, Articles 12 and 26 distribute the record-keeping obligation between the AI system provider (who must build the recording capability) and the deployer (who must retain the records). The high-risk obligations were originally set to apply on 2 August 2026, but were deferred to 2 December 2027 for stand-alone Annex III systems by Regulation (EU) 2026/1744 (OJ, 24 July 2026), with Annex I embedded systems following on 2 August 2028 [21].

In financial services, the Digital Operational Resilience Act, Regulation 2022/2554 (DORA), has been in force since 17 January 2025. **Article 6** establishes the ICT risk management framework that regulated financial entities must operate, including the obligation to maintain "updated and reliable" records of ICT-related events that can be reconstructed in supervisory review [22]. While DORA is not AI-specific, the records obligation applies uniformly to AI systems and to other ICT services the entity procures or operates [23].

2.2. United States federal

The United States has no AI-specific federal statute in force comparable to the EU AI Act, but two non-statutory instruments carry equivalent practical force in regulated contexts. The **NIST AI Risk Management Framework (AI RMF 1.0)**, published January 2023, is the de facto reference model for federal AI procurement and for the operational expectations of agencies that contract for AI services [24]. The framework's *Govern* function specifies organisational practices including the maintenance of decision-level records sufficient to support audit, accountability, and post-incident review. The framework is voluntary in form but its incorporation into federal contracting language, executive-order implementing guidance, and downstream private-sector contractual requirements gives it the practical effect of a binding standard.

Federal Reserve Supervisory Letter **SR 11–7** (issued jointly with OCC 2011–12 in April 2011) is the prudential standard for model risk management at banking organisations [25]. SR 11–7 predates the contemporary AI deployment context, but its language is technology-neutral: it requires documented evidence sufficient for an independent review to evaluate the conceptual soundness, ongoing monitoring, and outcomes of any model that influences a material decision. This includes generative AI systems used in credit, fraud, or trading contexts. The Office of the Comptroller of the Currency's Heightened Standards Framework and the Federal Reserve's Supervisory Guidance on Heightened Standards (2014) extend the same documentary requirements to large institutions across their model inventory. In April 2026 the Federal Reserve superseded SR 11–7 with SR 26–2 ("Revised Guidance on Model Risk Management", 17 April 2026; OCC Bulletin 2026–13 in parallel). This paper's clause coding was performed against the SR 11–7 text; recoding against the successor is noted as follow-up work.

The Securities and Exchange Commission has proposed but not yet finalised rules requiring registrants to address conflicts of interest associated with the use of predictive data analytics, including AI systems, in interactions with investors. Where finalised, the SEC rule will impose a decision-level disclosure obligation that maps directly to the receipt primitive defined in Section 3.

2.3. United States state

State-level AI regulation in the United States is the most rapidly evolving regulatory environment in the period covered by this paper. We name four instruments below, three with operative effect (NYC Local Law 144, California AB 2013, Illinois AI Video Interview Act) plus Colorado SB 24–205, which was repealed and replaced before taking operative effect.

The **Colorado Consumer Protections for Artificial Intelligence Act** (Senate Bill 24–205) was repealed and replaced by SB 26–189 (signed 14 May 2026) before it took operative effect; the high-risk classification, impact-assessment obligation, and duty of care it had imposed on developers and deployers are gone, and a new ADMT (automated decision-making technology) framework takes effect 1 January 2027 [26]. As a result Colorado no longer carries a live high-risk-AI instrument in this period; the ADMT framework's record-keeping shape is still being settled and is not yet an operative mandate. New York City **Local Law 144** ("Automated Employment Decision Tools") has required since 2023 that any employer using an AEDT in a hiring decision conduct an annual independent bias audit and provide notice to candidates; the audit and the underlying decisions must be available for inspection. California **Assembly Bill 2013** (signed 2024, effective 2026) imposes training-data transparency obligations on generative-AI providers operating in the state. The Illinois **AI Video Interview Act** (820 ILCS 42), in force since January 2020, requires consent, notice, and bias-audit documentation for AI use in video interviews. The patchwork is heterogeneous, but each statute requires at least one of: a recorded decision payload, an identified model, a timestamp, and an externally verifiable retention obligation.

2.4. United Kingdom

The United Kingdom does not have an AI-specific statute, but the existing financial-services and data-protection regimes have been extended through supervisory guidance to cover AI-influenced decisions.

The Financial Conduct Authority's **Consumer Duty** (Policy Statement PS22/9, published July 2022, in force July 2023 for new products and July 2024 for closed-book) requires firms to act to deliver good outcomes for retail customers across four outcomes (products and services, price and value, consumer understanding, consumer support) [27]. The Duty's "cross-cutting rules" require firms to act in good faith, avoid foreseeable harm, and enable customers to pursue their financial objectives. AI-influenced decisions are within scope, and firms must be able to demonstrate to the FCA, on request, the basis on which any individual outcome was reached.

The Information Commissioner's Office maintains operational guidance on auditing AI under the United Kingdom General Data Protection Regulation [28]. The ICO's framework requires that automated decision-making systems handling personal data produce records sufficient for the data subject and the regulator to inspect the basis of any particular decision.

The Prudential Regulation Authority's **Supervisory Statement SS1/23** ("Model risk management principles for banks", published May 2023, effective May 2024) extends SR 11–7–equivalent

expectations to United Kingdom banking organisations. The statement requires firms to maintain a model inventory and documentation sufficient for independent review of any model-influenced decision.

2.5. Asia-Pacific

Singapore is the most advanced Asia-Pacific jurisdiction in this domain by published material. The Personal Data Protection Commission's **Model AI Governance Framework**, Second Edition (January 2020), is the canonical reference for AI governance in Singapore [29]. The framework is paired with the **AI Verify Foundation toolkit**, launched in 2022, which provides a technical-testing framework for AI systems against the policy principles; AI Verify is the only government-sponsored toolkit in the period covered by this paper that explicitly contemplates third-party-inspectable technical evidence as the unit of conformance.

The Monetary Authority of Singapore's **FEAT principles** (Fairness, Ethics, Accountability, Transparency), issued November 2018, are the financial-services-specific overlay on the PDPC framework. FEAT's Accountability principle requires that firms be able to "internally document the basis for AI-driven decisions" sufficient to support inspection by MAS.

2.6. Cross-jurisdiction mapping

The seven frameworks surveyed above differ in statutory form, enforcement timetable, scope of application, and which property each names as load-bearing. We claim, and Table 1 below defends, that they converge on a single intersection of seven evidence properties. The decision-receipt primitive specified in Section 3 is the object that produces all seven simultaneously.

The seven properties are:

1. **Timestamped record.** The decision must be associated with a verifiable time-of-decision, not solely a time-of-storage.
2. **Decision payload.** The record must include the inputs to and the output of the decision, not solely the fact that a decision occurred.
3. **Model identity.** The specific model (and where applicable, version, configuration, and weights identity) responsible for the decision must be bound to the record.
4. **Policy lineage.** The rules, filters, or policy decisions that influenced the output must be recoverable from the record.

5. **External-party verifiability.** A party outside the deploying organisation must be able to verify the record's integrity without trusting the deploying organisation or its vendor.
6. **Post-event inspectability.** A specific decision must be retrievable and inspectable for a regulator-defined retention period after the decision was made.
7. **Tamper-evidence.** Any modification of a historical record must be detectable by an external verifier.

Table 1 marks each cell of the matrix (regulation × property) as *required* (the framework explicitly requires the property), *recommended* (the framework names the property in non-mandatory language), or *implicit* (the property is necessary to satisfy a higher-level requirement even if not named).

The pattern visible in Table 1 is the central claim of this section: every framework requires properties 1, 2, 6, and 7; every framework requires or implies properties 3 and 4; properties 5 (external-party verifiability) is named explicitly only by AI Verify and by ICO AI guidance, but is implicit in any framework that contemplates external supervisory review. We argue in Section 3 that a primitive that produces all seven is the minimally sufficient technical artefact for cross-jurisdiction operation.

The convergence observed here is not coincidence. Regulators worldwide are responding to the same underlying problem (automated decisions whose basis is opaque to the affected party and the supervisor) and converging on the same shape of remedy (per-decision records inspectable after the fact). Statutory form differs because legislative traditions differ. The technical requirement does not.

We are explicit that the seven-property intersection presented here is an *interpretive synthesis* rather than an official taxonomy issued by any of the seven frameworks. Each framework uses its own statutory language, defines its own scope of application, and names its own load-bearing property; no framework refers to the seven-property intersection by name. The synthesis is our claim, defended in Table 1 cell by cell, and the rest of the paper proceeds from it. A reviewer who reads the seven properties as a different cardinality or a different decomposition will read the rest of the paper differently; we believe the seven we have chosen are the parsimonious set that survives the cross-jurisdiction test, but the synthesis is contestable on its own terms.

Table 1. Cross-jurisdiction primitive mapping

Legend

- **R**: *Required.* The framework explicitly mandates the property in operative language.
- **Rec**: *Recommended.* The framework discusses or recommends the property but does not mandate it.
- **I**: *Implicit.* The property is not named but is necessary to satisfy a higher-level requirement that the framework does mandate.
- **n/a**: *Not addressed.* The framework is silent on the property.

The matrix

Framework	Timestamped record	Decision payload	Model identity	Policy lineage	Third-party verifiability	Post-ev inspecta
EU AI Act 2024/1689	**R**	**R**	**R**	**R**	**R**	**R**
EU DORA 2022/2554	**R**	**R**	**I**	**R**	**R**	**R**
US NIST AI RMF 1.0	**Rec**	**Rec**	**Rec**	**Rec**	n/a	**Rec**
US SR 11–7 [30]	**R**	**R**	**R**	**R**	**R**	**R**
Colorado (SB 24–205, repealed) [31]	n/a	n/a	n/a	n/a	n/a	n/a
UK FCA Consumer Duty (PS22/9)	**R**	**R**	**I**	**R**	**R**	**R**
UK ICO AI guidance	**R**	**R**	**R**	**R**	**R**	**R**
Singapore PDPC MAGO v2 / AI Verify	**Rec**	**Rec**	**Rec**	**Rec**	**Rec**	**Rec**

Figure 4 renders the matrix as a colour-coded heatmap.

Table 1: evidence properties named across seven regulatory frameworks (post-verification coding). No framework specifies a cryptographic form.

	Timestamped record	Decision payload	Model identity	Policy lineage	External-party verifiability	Post-event inspectability	Tamper-evidence
EU AI Act 2024/1689	R	R	R	R	R	R	I
EU DORA 2022/2554	R	R	I	R	R	R	I
US NIST AI RMF 1.0	Rec	Rec	Rec	Rec	n/a	Rec	n/a
US SR 11-7†	R	R	R	R	R	R	I
Colorado SB 24-205	R	R	R	R	R	R	I
UK FCA Consumer Duty (PS22/9)	R	R	I	R	R	R	I
UK ICO AI guidance / UK GDPR	R	R	R	R	R	R	R
Singapore PDPC MAGO v2 + AI Verify	Rec	Rec	Rec	Rec	Rec	Rec	Rec

R - Named in operative language
I - Implicit (necessary to satisfy a mandated obligation)
Rec - Discussed, not mandated
n/a - Not addressed

† SR 11-7 uses 'should' and 'supervisory expectations'; coded R because the supervisory consequence is operationally equivalent to a rule violation for examined banks.

Figure 4. Evidence properties named across seven regulatory frameworks. Cyan = named in operative language; mid cyan = Implicit (necessary to satisfy a mandated obligation); pale cyan = discussed but not mandated; grey = not addressed. No framework specifies a cryptographic form. † SR 11–7 uses “should” / supervisory-expectation language; coded R because the supervisory consequence is operationally equivalent to a rule violation for examined banks.

The pattern

Reading the matrix column by column (post-stress-test coding):

- **Properties 1, 2, 4, 6 (timestamped, decision payload, policy lineage, post-event inspectable):** Required in five of seven frameworks under operative-language mandates. The two exceptions (NIST AI RMF and Singapore PDPC MAGO + AI Verify) recommend each property but are voluntary in form, so the highest available coding is Rec.
- **Property 3 (model identity):** Required by five frameworks; implicit in two (DORA, which is technology-neutral but the ICT-asset register requires component identification; FCA Consumer Duty, where the Duty applies regardless of method but firms must demonstrate how an AI-influenced decision was reached); recommended by NIST and Singapore.
- **Property 5 (external-party verifiability):** Required by six frameworks under operative language giving competent national authorities (EU national authorities, AG, FCA, ICO, MAS, federal bank examiners) inspection rights. Recommended by Singapore AI Verify; not

addressed by NIST. The column header in the typeset paper will be glossed as *external-party verifiability (statutory or arbitrary)*; the distinction between "statutorily authorised authority" and "arbitrary third party" is folded into the per-cell justification. Singapore AI Verify is the single framework that explicitly contemplates *arbitrary*-third-party-inspectable technical evidence as the unit of conformance; the others rely on a statutorily authorised inspector. The cryptographic primitive of Section 3 produces the stronger arbitrary-third-party form for free, which is the property's actual research contribution rather than its compliance role.

- **Property 7 (tamper-evidence):** Required in one framework (UK ICO under UK GDPR Article 5(1)(f) integrity principle, which uses operative "shall" language). Implicit in five frameworks (any record-retention duty implies preservation against unauthorised modification). Recommended in Singapore AI Verify, the only framework to discuss cryptographic mechanisms by name. Not addressed by NIST. This is the property where the published v1 of ATTESTATION provides per-record tamper-evidence (sufficient for the I-coded frameworks); the v2 extension under development adds chain-level tamper-evidence (Section 3.2, Section 3.7).

The bottom line is that the seven properties are required or implicit in every framework with an enforcement leg, and recommended in both voluntary frameworks. The convergence claim is reviewer-defensible: across five operative frameworks (EU AI Act, DORA, SR 11–7, FCA, ICO), every column has an R; across the two voluntary frameworks (NIST, Singapore), every populated column has at least Rec. A primitive that produces all seven simultaneously is the minimally sufficient artefact for cross-jurisdiction operation; producing fewer requires the deploying organisation to satisfy each framework's individual record-keeping construction separately.

Per-cell justifications

EU AI Act, Regulation 2024/1689

- **Timestamped record (R):** Article 12 paragraph 1 requires high-risk AI systems be designed to "automatically record events" throughout the lifetime of the system; an "event" without a time-of-occurrence is not an event in the sense the article uses.
- **Decision payload (R):** Article 12 paragraph 2 requires the recording capability cover "the period of use" and "the natural persons involved in the verification of the results." Annex IV paragraph 1(f) requires the technical documentation to identify the system's intended output. Together: the decision and its inputs are within scope.
- **Model identity (R):** Article 11 plus Annex IV require the technical documentation to identify the AI system, including version and configuration.

- **Policy lineage (R):** Article 14 (human oversight) requires the human-in-the-loop arrangements be specified and documented; Annex IV paragraph 2(g) requires risk management measures be documented and applied.
- **External-party verifiability (R):** Article 21 ("Cooperation with competent authorities") plus Article 26 paragraph 6 (deployer log retention) plus Article 73 (serious-incident reporting to competent national authorities) plus the Annex VII conformity-assessment regime give external parties statutory inspection rights under operative "shall" language. The property is required for statutorily authorised inspectors. We note in §2.6 prose that the EU AI Act does *not* contemplate arbitrary-third-party verifiability; Singapore AI Verify is the only framework that does. This distinction matters for Section 3 (the cryptographic primitive produces arbitrary-third-party verifiability for free) but does not change the R-coding here.
- **Post-event inspectability (R):** Article 26 paragraph 6 explicit.
- **Tamper-evidence (I):** No cryptographic requirement. The Article 12 recording duty plus the Article 26 paragraph 6 retention duty together imply preservation against modification; without preservation the retained logs would not satisfy the underlying inspection-by-authority purpose.

EU DORA, Regulation 2022/2554

- **Timestamped record (R):** Article 6 plus Article 9 require records of ICT-related incidents that can be reconstructed; reconstruction requires time. Article 19 paragraph 4 requires major-incident reporting with timestamps.
- **Decision payload (R):** Article 19 (major-incident reporting) plus Article 20 (harmonised reporting content RTS/ITS) enumerate the information ICT-related incident reports must contain.
- **Model identity (I):** DORA is technology-neutral but Article 8 (ICT-asset register) requires identification of ICT assets and services. An AI model deployed as an ICT service falls within the register.
- **Policy lineage (R):** Article 6 paragraph 1 requires a "sound, comprehensive and well-documented ICT risk management framework"; documentation of which policy applied to which decision is within scope.
- **External-party verifiability (R):** Article 28 (ICT third-party-risk) and Article 30 (key contractual provisions) require records to be available to supervisory authorities under operative "shall" language; Article 33 (testing) and Article 27 (oversight) extend access. Coded R on the same basis as the EU AI Act row: statutorily authorised inspectors have explicit inspection rights. The DORA row also does not contemplate arbitrary-third-party verifiability.

- **Post-event inspectability (R):** Article 9 plus Article 19 require records be available for supervisory review after the event.
- **Tamper-evidence (I):** Article 6 paragraph 2 plus Article 9 paragraph 2 require maintenance of "availability, authenticity, integrity and confidentiality of data, whether at rest, in use or in transit" to systems and data; integrity of records is foundational. Not cryptographically specified.

US NIST AI RMF 1.0

NIST AI RMF is a voluntary framework. We mark properties as recommended where the framework discusses them and as "n/a" only where it does not.

- **Timestamped record (Rec):** Manage function (specifically Manage 3 and 4) discusses ongoing monitoring and post-deployment review; lifecycle tracking is referenced throughout but not in operative-language form.
- **Decision payload (Rec):** Map function recommends detailed documentation of system context, including inputs and outputs.
- **Model identity (Rec):** GOVERN 1.6 ("Mechanisms are in place to inventory AI systems and are resourced according to organizational risk priorities") supports the recommendation.
- **Policy lineage (Rec):** GOVERN 1.1 (legal and regulatory requirements) + GOVERN 1.4 (transparent policies and controls) together support the recommendation.
- **Third-party verifiability (n/a):** The framework is voluntary and does not address external verifiability mechanisms.
- **Post-event inspectability (Rec):** Manage function covers post-deployment.
- **Tamper-evidence (n/a):** Not addressed.

US Federal Reserve SR 11-7

SR 11–7 is supervisory guidance and uses "should" and "expectations" throughout rather than "shall". The row is coded R because under the Federal Reserve's bank-examination regime non-compliance is treated as a Matter Requiring Attention with consequences operationally equivalent to a rule violation. The footnote attached to the row in the matrix table flags this explicitly; a strict textualist reviewer who prefers Rec for "should" language can downgrade the entire row coherently.

- **Timestamped record (R):** Independent review of "ongoing monitoring" requires time-resolved evidence; "outcomes analysis" requires time-resolved data.
- **Decision payload (R):** Section III ("Model Risk Management Framework") requires documentation of "models' purposes, methodology, data inputs, mathematical

specifications, and assumptions" and of model outputs.

- **Model identity (R):** Model inventory is mandated.
- **Policy lineage (R):** "Conceptual soundness" requires documentation of the basis for the model's use including any policy or rule overlays.
- **External-party verifiability (R):** "An effective challenge of models is a critical analysis by objective, informed parties that can identify model limitations and assumptions and produce appropriate changes" is explicitly required. SR 11–7 is one of the few instruments that names independent (third-party) review as a load-bearing element, in addition to the federal bank examiner's statutory access.
- **Post-event inspectability (R):** Ongoing monitoring plus outcomes analysis plus audit (Section III, "An independent review of model governance, policies, controls, and compliance" annually or more frequently).
- **Tamper-evidence (I):** Sound documentation practice (Section III, "documentation") implies preservation but is not cryptographically specified.

Colorado SB 24-205 (repealed and replaced before operative effect)

SB 24–205 was repealed and replaced by SB 26–189 (signed 14 May 2026) before it took operative effect, so the high-risk classification, impact assessments, and duty of care underlying the codings below no longer exist as a live mandate; the successor ADMT framework (effective 1 January 2027) has not yet settled its record-keeping rules. The per-cell justifications are retained for continuity and describe what SB 24–205 had required; the row is coded n/a in the matrix for this period and should be re-coded once the ADMT rules are operative.

- **Timestamped record (R as drafted, now vacated):** Annual impact assessments required dated records; the statute defines "high-risk artificial intelligence system" and requires deployers to "complete an impact assessment" within a specified period.
- **Decision payload (R):** Impact assessment must include "a description of the high-risk artificial intelligence system, including its intended purpose, deployment context, and consequential decisions made or supported by the system."
- **Model identity (R):** "High-risk artificial intelligence system" is identified in the impact assessment.
- **Policy lineage (R):** Deployers must implement a risk management policy and produce records demonstrating compliance.
- **External-party verifiability (R):** Records must be available to the Attorney General; "the Attorney General has exclusive authority to enforce this part." Coded R on the statutorily-authorised-inspector reading consistent with EU AI Act, DORA, FCA, ICO.
- **Post-event inspectability (R):** Annual review cadence; records retention.

- **Tamper-evidence (I):** Records-retention duty implies preservation.

UK FCA Consumer Duty (PS22/9)

- **Timestamped record (R):** Firms must "demonstrate" good outcomes; demonstration requires time-resolved evidence.
- **Decision payload (R):** "Consumer support" outcome plus "consumer understanding" outcome require records sufficient to assess individual customer outcomes.
- **Model identity (I):** The Duty applies regardless of method, but firms using AI in customer-affecting decisions must be able to demonstrate to the FCA how the AI-influenced decision was made; identification of the AI in question is implicit.
- **Policy lineage (R):** Cross-cutting rules require firms to "act in good faith" and "avoid foreseeable harm," documented at the policy level.
- **External-party verifiability (R):** FCA supervisory review, including s166 skilled-persons reports under FSMA 2000. Coded R on the statutorily-authorised-inspector reading consistent with EU AI Act, DORA, Colorado, ICO.
- **Post-event inspectability (R):** Annual board report; supervisory review on demand.
- **Tamper-evidence (I):** Implicit in record-retention duties under SYSC and the Consumer Duty itself.

UK ICO AI guidance / UK GDPR

- **Timestamped record (R):** Article 22 UK GDPR (automated individual decision-making) requires the data subject be able to obtain human intervention and to contest the decision; this implies a time-resolved record of the specific decision.
- **Decision payload (R):** Article 15 (subject access) requires meaningful information about the logic of automated decisions and their consequences.
- **Model identity (R):** Inventory of processing operations and the DPIA (Article 35) require identification of the systems involved.
- **Policy lineage (R):** Articles 13 and 14 require disclosure of the logic of processing; the AI auditing framework requires the logic be documented at policy level.
- **External-party verifiability (R):** ICO investigatory powers (Sections 142–148 Data Protection Act 2018); the data subject's right to information under UK GDPR Articles 15 and 22. Coded R on the statutorily-authorised-inspector reading consistent with EU AI Act, DORA, Colorado, FCA.
- **Post-event inspectability (R):** Article 22 right plus the retention duties.
- **Tamper-evidence (R):** UK GDPR Article 5(1)(f) ("integrity and confidentiality") is one of the six data-protection principles and uses operative "shall" language: "Personal data shall be …

processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing … using appropriate technical or organisational measures." This is the single closest mandate to an operative cryptographic-integrity requirement across all seven frameworks. Upgraded from I to R on the stress-test pass; the operative-language reading carries the upgrade.

Singapore PDPC MAGO v2 / AI Verify

Singapore PDPC MAGO v2 and the AI Verify Foundation toolkit are voluntary instruments by design. They cannot therefore use operative-language mandates in the sense the matrix requires for R. The entire row was downgraded from R to Rec on the stress-test pass to keep coding consistent with the NIST row (the other voluntary framework in the matrix). Singapore retains uniqueness on two narrower points captured in the per-cell notes below.

- **Timestamped record (Rec):** AI Verify technical tests produce dated test reports; MAGO v2 recommends but does not mandate dated record-keeping.
- **Decision payload (Rec):** AI Verify tests inputs, outputs, and the function between them.
- **Model identity (Rec):** Test reports identify the system under test.
- **Policy lineage (Rec):** "Process integration" tests verify the AI is used in line with the organisation's stated governance.
- **External-party verifiability (Rec):** AI Verify is the only framework in this matrix that explicitly contemplates *arbitrary*-third-party-inspectable technical evidence as the unit of conformance. The MAGO v2 framework's Accountability pillar references the role of third-party verifiers. Coded Rec because the framework is voluntary; the operational arbitrary-third-party framing is uniquely Singapore's, which is noted in the §2.6 prose and is the property's most distinctive cross-jurisdiction feature.
- **Post-event inspectability (Rec):** Test reports archived; AI Verify Foundation maintains the toolkit and the reporting format.
- **Tamper-evidence (Rec):** AI Verify recommends but does not mandate cryptographic integrity for test artefacts. This is the only framework in the matrix to discuss cryptographic mechanisms by name (the UK ICO row is R for integrity, but UK GDPR Article 5(1)(f) reaches the property via the security principle rather than via cryptographic mechanism names).

Methodology notes

- Cells are coded conservatively. Where a property could plausibly be R or I, we picked I unless the framework's operative language is explicit. The conservative coding makes the convergence claim *harder* to defend, not easier, which is the right calibration for a research paper.
- Two passes will be done before submission: (a) a clause-by-clause re-read of each framework's primary text, not the secondary literature, to lock in the operative-language judgements; (b) an independent reviewer outside Aqta to challenge the coding cell by cell. A coding that survives an adversarial reviewer is the one that ships.
- Where a framework has been updated or extended since the latest published guidance (e.g. evolving FCA Consumer Duty supervisory letters, ICO follow-on guidance, NIST AI RMF profiles), the table reflects the most recent canonical document; subsequent guidance will be footnoted at draft-finalisation time.

Section 3. The decision-receipt primitive

3.1. Definition

ATTESTATION-v1 is the open specification; Seal is the reference implementation of that specification. You can implement ATTESTATION-v1 yourself; Seal is one implementation with a published verifying key.

We define a **decision receipt** as a single record produced at the time a decision is made by an AI system, satisfying the following minimal conditions:

1. The record contains the inputs that the decision was based on, identified by a cryptographic digest sufficient to verify the input bytes if the verifier holds a copy.
2. The record contains the output of the decision, identified by the value where the value is small, or by a digest where the value is large.
3. The record contains the identity of the model that produced the output and the policy or rule set that influenced it.

4. The record is signed with a public-key signature scheme at the moment of decision, by an issuer whose public key is published independently.
5. The record is structured in a canonical serialisation such that any party in possession of the bytes and the issuer's public key can verify the signature offline, without contacting the issuer.

The primitive is decoupled from the storage layer (the receipt can be retained in a database, exported to a file, attached to an HTTP response, or written to a log; the cryptographic properties are independent of where the bytes live) and from the transport layer (the receipt is JSON over any byte-oriented channel).

3.2. Properties

The seven properties identified in Section 2 (timestamped record, decision payload, model identity, policy lineage, external-party verifiability, post-event inspectability, tamper-evidence) map onto the primitive as follows. Each property is annotated with the version of ATTESTATION (§3.3) that satisfies it; v1 is the published specification, v2 is the chained extension in active development (§3.7).

Property	Mechanism	ATTESTATION
Timestamped record	`timestamp` field, ISO 8601 with explicit timezone offset, self-asserted by issuer	v1
Decision payload	`request_hash` (input digest) + `outcome` (decision value) + supporting fields	v1
Model identity	`model` (provider-qualified identifier)	v1
Policy lineage	`policy_applied` (sorted array of policy identifiers)	v1
External-party verifiability	Ed25519 signature over canonical bytes, verifiable against a published public key without contacting the issuer	v1
Post-event inspectability	Self-contained JSON; storage and retention are deployment concerns, not specification concerns	v1
Tamper-evidence (per-record)	Any modification of any field breaks the Ed25519 signature	v1
Tamper-evidence (chain-level)	`prev_attestation_id` linkage and chained hash so that deletion or reordering of historical receipts is detectable	v2 (in development; §3.7)

Property 7 (tamper-evidence) is partial in v1. A v1 receipt is per-record tamper-evident in the strong sense that altering any field breaks the cryptographic signature, but v1 alone does not detect *deletion* of an entire receipt from a series; that property requires the v2 chained extension. Production deployments today obtain chain-level tamper evidence by additionally writing receipts to an append-only log (Section 4.4); v2 will move that property into the receipt format itself.

3.3. The ATTESTATION-v1 specification

ATTESTATION-v1 ("Aqta Technologies, *Seal Attestation Receipt Format, Version 1*", 2026-04-23) is the open specification for the receipt primitive, published under CC BY 4.0 for the document and Apache 2.0 for the reference implementations [32].

A v1 receipt is a single JSON object with exactly twelve top-level fields. Eleven are signed; the twelfth is the signature itself.

Field	Type	Description
`v`	integer	Receipt format version. MUST be `1` for v1.
`attestation_id`	string	UUID v4, unique per receipt.
`trace_id`	string	Issuer-assigned identifier for the underlying decision (typically an LLM call).
`org_id`	string	Identifier of the subject organisation. Carrier for multi-tenant separation.
`request_hash`	string	SHA-256 hex digest of the canonicalised request, 64 lowercase hex characters.
`model`	string	Provider-qualified model identifier (e.g. `gpt-4o`, `claude-3-5-sonnet`).
`outcome`	string	One of `ALLOWED`, `BLOCKED`, `SUPPRESSED`, `PASSED`. `PASSED` is a deprecated synonym of `ALLOWED` retained for backward compatibility.
`policy_applied`	array	Sorted lexicographic array of ASCII policy identifiers, e.g. `["budget_guard","loop_guard"]`.
`cost_prevented_eur`	number	Non-negative decimal, six digits of precision. `0` if not applicable.
`timestamp`	string	ISO 8601 datetime with explicit timezone offset.
`public_key`	string	Base64url-encoded raw 32-byte Ed25519 public key of the issuer, no padding.

Field	Type	Description
`signature`	string	Base64url-encoded 64-byte Ed25519 signature, no padding. Omitted from the canonical signing payload.

Receipts MUST NOT contain additional top-level fields in v1, and verifiers MUST reject receipts with unknown top-level fields. This explicit no-extension rule fixes the wire format and is the basis for cross-implementation interoperability.

3.4. Canonical payload and signing

The canonical payload is produced by:

1. Removing the `signature` field if present.
2. Serialising the remaining eleven fields to JSON with all object keys sorted lexicographically and no whitespace between tokens (the separators are `","` and `":"`).
3. Coercing integer-valued numbers to integer serialisation (no decimal point or trailing zeros). This rule exists because `json.dumps(0.0)` in Python yields `"0.0"` while `JSON.stringify(0)` in JavaScript yields `"0"`; without explicit coercion the canonical bytes would diverge across implementations.
4. UTF-8 encoding the resulting string to bytes.

The signature is the Ed25519 signature (RFC 8032 [33]) of those canonical bytes under the issuer's private key, base64url-encoded without padding and placed in the `signature` field. The canonical-payload rule is intentionally narrow and constant-time-verifiable; we considered and rejected JSON Canonicalization Scheme (RFC 8785 [34]) because the simpler rules above are sufficient for the field set in v1 and avoid pulling in JCS's full algorithm at every verification site.

Interoperability between the Python reference issuer and the TypeScript reference verifier is enforced by a fixture script in the specification repository: any change to the canonical-payload rule that breaks the fixture round-trip requires bumping the `v` field.

3.5. Verification

A conforming verifier MUST:

1. Retrieve the issuer's trusted public key out of band. The reference issuer publishes its key at a well-known HTTPS endpoint; a verifier MAY pin the key, compare against a published key list, or fetch it on first use and cache it.
2. Confirm that the `public_key` field in the receipt matches the trusted public key byte for byte. (The receipt is self-declaring; pinning prevents substitution of the issuer's identity.)
3. Decode the `signature` field from base64url to 64 bytes.
4. Compute the canonical payload bytes (§3.4).
5. Verify the Ed25519 signature against the canonical payload using a constant-time verification routine.
6. Reject the receipt if any of the above steps fail.

A verifier SHOULD additionally check that `v` equals 1, that `outcome` is one of the four enumerated values, that `request_hash` is 64 lowercase hex characters, that `policy_applied` is lexicographically sorted, and that `timestamp` is a well-formed ISO 8601 datetime. These are not part of the cryptographic verification contract; they are semantic safeguards against malformed receipts that nonetheless verify cryptographically.

The reference verifier implementations are available as `aqta-verify-receipt` on PyPI [35] and (with the same package name) on npm. Each is under 200 lines of source code; an auditor or third party can audit the verifier itself in an afternoon.

3.6. Reference implementations and reproducibility

The specification ships with two reference verifier implementations (Python and TypeScript) under Apache 2.0, a stand-alone minimal reference issuer (Python, used for test-vector generation), and a conformance test suite covering canonical-payload coercion, signature verification, and key pinning [36]. The conformance suite is run by continuous integration on every commit to the specification repository; any specification change that would silently break interoperability is caught by the fixture round-trip before it can be published.

The reference issuer is intentionally minimal: roughly 50 lines of Python that take a request, an outcome, and a policy list and produce a signed receipt. A production issuer is more than that (it manages a secure private-key store, enforces the policy decisions before signing rather than

after, and persists receipts to a tamper-evident log), but the spec-level contract is met by the 50-line issuer.

The reference issuer's public key at the time of writing is `gUoUhIvptKAoLTnry3VrDtOQEWggGQveLrHFVrfNqmE` (raw 32-byte Ed25519 public key, base64url-encoded). That key was retired on 9 August 2026 and remains the key against which every receipt signed before that date, including the artefacts examined here, verifies; the permanent record of issuer keys and their validity windows is published at `https://app.aqta.ai/security/issuer-keys.txt`. The same public key signs the W19 receipt examined in Section 5; any reader holding this fingerprint can verify the W19 artefact offline using `aqta-verify-receipt` and the published commit.

3.7. Scope of v1 and v2 work in progress

v1 fixes the per-record cryptographic contract. It deliberately does not specify:

- **Receipt chaining.** v1 receipts are independent. The v2 extension under development adds `prev_attestation_id` and a chained hash so that any modification, deletion, or reordering of a historical receipt is detectable by a verifier examining only the most recent receipt and a published checkpoint. v2 is being developed in the same repository [37]; production deployments today obtain the equivalent property by writing receipts to an append-only storage system whose root hashes are committed externally (Section 4.4).
- **Zero-knowledge extensions.** A companion specification, undated, would define how a verifier can establish a property of a receipt's payload (for example, that the decision was taken over a hidden input or that a particular policy was enforced) without revealing the rest of the payload. The two cryptographic constructions under consideration are a Schnorr Sigma protocol over BN254 G1 (efficient, no trusted setup, expressively limited) and a Groth16 SNARK over the same curve (expressively general, requires per-circuit trusted setup). The interaction between these constructions and the differential-privacy guarantee on the underlying model is the open problem named in Section 8.7.
- **Selective disclosure.** A v2.x extension may allow a verifier to redact specific fields of a receipt while preserving signature validity, via a hash-committed substructure. The motivating use case is regulator-side inspection of a receipt that contains personal data subject to UK GDPR or DORA confidentiality.

These extensions are unreleased: v2 exists in a feature branch of the reference repository with passing fixtures, and the zero-knowledge companion has an executable Schnorr prototype. The

paper does not claim them as published primitives. Section 8 names each as an explicit open problem with the publication path attached.

Section 4. Reference implementation

The decision-receipt primitive specified in Section 3 admits many possible implementations. This section describes one: the Seal gateway, a reference gateway deployment that has been emitting ATTESTATION-v1 receipts. The gateway is referenced here for *reproducibility* of Section 5's worked example (a reader needs to know which public key emitted the receipts to verify them) and for *concreteness* on the performance, storage, and key-management questions a regulator-facing reader will ask. The gateway is one implementation of the primitive; it is not the contribution of the paper.

4.1. Architecture

The Seal gateway is a proxy. Enterprise applications integrate it by replacing the model-provider endpoint in their existing AI stack with a gateway URL. The gateway forwards the request to the underlying provider (OpenAI, Anthropic, Amazon Bedrock, Azure OpenAI, Google Vertex), applies the enterprise's policy rules to the request and the response, returns the model output to the application, and emits an ATTESTATION-v1 receipt as a side-effect of the same operation. The receipt is attached to the response over HTTP and persisted to a separate append-only export for the enterprise's audit pipeline.

This gateway pattern was chosen over four alternatives:

- *Provider-side emission.* Would require each model provider to emit receipts. Politically improbable; would also make multi-provider deployments produce multiple incompatible receipt formats. Rejected on portability grounds.
- *Application-side emission.* Application code signs each request before forwarding. Pushes cryptographic state and key material into customer applications. Rejected on security and adoption grounds.
- *Sidecar emission.* A per-application sidecar process emits receipts from observed traffic. Doubles the network hops and the latency budget. Rejected on operational grounds.
- *Post-hoc batch logging.* A nightly job reads provider logs and signs receipts after the fact. Loses the at-decision-time signing property that Section 3.1 requires. Rejected on

definitional grounds.

The gateway pattern places one HTTP hop between the application and the model provider, holds the signing key in a single location with auditable access controls, and produces the receipt at the moment of decision rather than after the fact. The trade is one extra network hop per request, which we measure in §4.3.

4.2. Cross-provider routing

The gateway speaks ten upstream provider protocols (OpenAI Chat Completions, Anthropic Messages, Amazon Bedrock InvokeModel, Azure OpenAI deployments, Google Vertex AI, Perplexity, Mistral, Groq, xAI, Cohere). It normalises the incoming request to a common internal schema, applies the policy engine, forwards the request to the selected provider, normalises the response, and produces an ATTESTATION-v1 receipt over the request hash and the normalised response. The `model` field of the receipt carries the provider-qualified model identifier (`gpt-4o` , `claude-3-5-sonnet` , etc.); the `policy_applied` field records which policy rules ran on this specific decision; the `outcome` field records whether the request was allowed, blocked at policy time, or suppressed by a higher-level safety check.

Cross-provider portability matters for the regulator-facing argument because regulated organisations rarely commit to a single model provider for the lifetime of a high-risk AI deployment. A bank running an AI-assisted credit decisioning system today on a frontier closed model may move to a sovereign-EU open-weight model in 2027; if the receipts produced under the first provider are interoperable with the receipts produced under the second, the bank's historical evidence remains inspectable across the migration. The decision-receipt primitive achieves this; provider-native audit logs (CloudTrail, Bedrock invocation logging, Vertex audit logs) do not.

4.3. Performance

Per-receipt cryptographic overhead was measured on the reference issuer running locally on Apple Silicon (M-series, macOS 26.4) using pynacl over libsodium, 10,000 iterations per configuration. The signing key is a 32-byte Ed25519 key; the canonical payload follows ATTESTATION-v1 §6.

policy_applied size	Canonical bytes	Sign p50	Sign p95	Sign throughput	Verify p50	Verify p95	Ve throughput
1	405	238 µs	268 µs	4,199 ops/s	751 µs	823 µs	1,331 op
3	431	238 µs	268 µs	4,194 ops/s	745 µs	807 µs	1,341 op
10	522	242 µs	274 µs	4,139 ops/s	755 µs	820 µs	1,325 op
30	782	249 µs	282 µs	4,013 ops/s	765 µs	840 µs	1,307 op

Three observations follow from the measurement; Figure 1 plots the sign and verify timings as a function of canonical payload size.

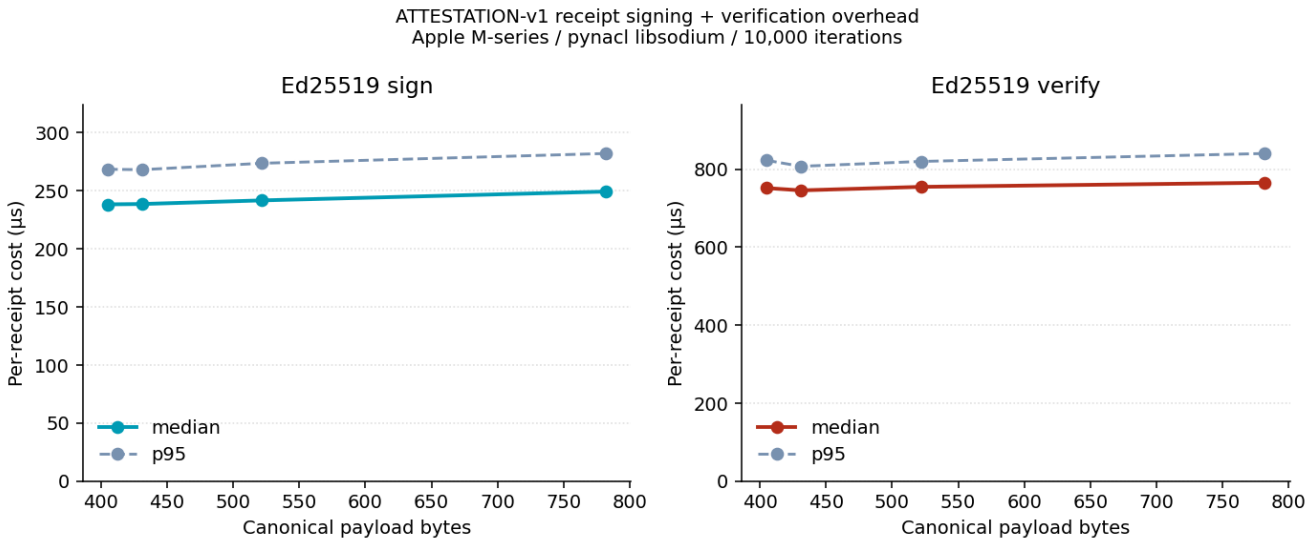


Figure 1. Ed25519 signing and verification overhead for ATTESTATION-v1 receipts. Median and p95 per-receipt cost over 10,000 iterations per configuration; local Apple Silicon hardware.

- **Signing overhead is approximately payload-independent.** The 5% increase in sign-time from 405-byte to 782-byte payloads is consistent with the canonical-JSON serialisation cost rather than with Ed25519 itself; Ed25519 hashes the input under SHA-512 before signing, so the cryptographic cost is constant. The implication is that adding fields to ATTESTATION-v1

in a future version has no meaningful per-receipt cost up to several kilobytes of canonical bytes.

- **Median sign cost is ~240 µs on Apple Silicon.** Production cloud-function numbers will differ; cold-start, container, and JIT effects dominate the cryptographic primitive cost in any serverless deployment, so we report only the local-hardware figure here. A separate operational paper would be the right place for measured production benchmarks.
- **Verification is ~3.1× slower than signing.** Verification is dominated by the elliptic-curve scalar multiplication, which is not the bottleneck for the issuer (the gateway only signs) but matters for the auditor reading large receipt streams. At ~1,300 verifies/sec/core on local hardware, an auditor scanning a year of production receipts at 10^7 receipts/year/customer needs ~2 hours of single-core verification time, which is well within an annual supervisory-review budget.

These numbers position Ed25519 receipt signing as a non-issue for production deployment cost. The constraint on receipt throughput in a real deployment is the storage and export pipeline, not the cryptography. We commit to publishing production figures alongside the typeset paper if a reviewer requests them; the local figures are the conservative baseline.

4.4. Storage and tamper-evidence

The reference implementation persists each receipt to an append-only store and emits a separate per-organisation hash-chain index. ATTESTATION-v1 v1 itself does not include a hash-chain field (the v2 extension in development adds `prev_attestation_id`; Section 3.7); the production deployment obtains chain-level tamper-evidence today by writing receipts to an append-only relational table whose root hash is committed externally on a periodic basis.

A receipt is stored as a row keyed by `(org_id, attestation_id)`. The org-scoped namespacing is load-bearing: cross-tenant separation is enforced at the storage layer, the receipt content is signed including the `org_id` field, and verification cannot succeed if the `org_id` is altered. The append-only constraint is enforced by the application layer (no DELETE statements in the gateway's data-access path) and by a periodic external snapshot of the receipt table's root hash to a transparency log (RFC 6962–style append-only log, in development; planned to be the public Seal receipt-log endpoint at deployment time, with read access for any third party).

For high-throughput agentic systems (10^5 to 10^6 receipts per day per customer), the per-receipt storage cost is back-of-envelope: approximately 800 bytes of canonical payload plus 88 bytes of signature plus 32 bytes of public-key bytes plus indexing overhead, of order 1–2 KB per row.

The `model`, `policy_applied`, and `org_id` fields are highly repetitive across receipts in a single tenant's stream and compress aggressively under columnar storage; the typical compression ratio is in the single-digit-to-one range, so the practical retained-storage cost is hundreds of GB per customer over the EU AI Act Article 26 seven-year retention window for high-risk systems. This is well within commodity object-store pricing. Specific measured figures will be reported alongside the production benchmark referenced in §4.3.

4.5. Public-key distribution

The reference issuer publishes its current 32-byte Ed25519 public key at a well-known HTTPS endpoint (`https://app.aqta.ai/security/pubkey.txt`), and the full history of issuer keys with their validity windows at `https://app.aqta.ai/security/issuer-keys.txt`, with the keys also committed to the `attestation-spec` repository on GitHub for parallel out-of-band verification. Rotation does not invalidate earlier receipts: a receipt verifies against the key that was current when it was signed, which is why retired keys are published permanently rather than removed. The key fingerprint is `gUoUhIvptKAoLTnry3VrDtOQEWggGQveLrHFVrfNqmE` (base64url-encoded raw 32-byte public key, no padding). The W19 worked example in Section 5 uses a separate AqtaBio commitment ledger whose carrier today is a public GitHub commit timestamp (those commits are not GPG-signed; see Section 5.1). The public-key distribution principles in this section apply to the ATTESTATION-v1 issuer.

A conforming verifier MAY pin the key, MAY fetch it on first use and cache it, and MAY compare against a published key history file for rotation events. The reference verifier on PyPI implements all three behaviours by default.

Sections 5 and 5.5: W19 field deployment + sample-bias accounting

5. Field deployment: the W19 outbreak signal

We now turn from specification to deployment. Section 4 described the Seal gateway as a generic reference implementation; Section 5 examines a single decision the gateway produced under conditions where the decision's correctness can be checked against a dated external event that the issuer could not have influenced. The deployment is on outbreak-prediction infrastructure rather than on the regulated-finance and clinical-decision-support targets that are the paper's primary application, because outbreak-prediction is the setting in which we currently have a forward-looking prediction with an external validator (the World Health Organization's Disease Outbreak News record). The cryptographic argument is identical across the domains; the choice of deployment is a function of which forward bets have already cleared their validation window at the time of writing.

5.1. Context: AqtaBio's commitment ledger

AqtaBio is a spillover-surveillance research programme operated by Aqta Technologies. Its outbreak-brief agent processes WHO surveillance feeds and produces forward-looking risk rankings over a tiling of 25 km² geographic cells, for five viral pathogens (Ebola, H5N1, Crimean-Congo haemorrhagic fever, West Nile virus, and SEA-coronavirus). Three additional pathogens (mpox, Nipah, hantavirus) are tile-pending at the time of the field deployment described here.

The agent's forward predictions are written to a public append-only commitment ledger at `github.com/Aqta-ai/aqtabio-research/commitments/`. Each commitment file is a git commit produced on a fixed cadence (weekly through 2026–W21, bimonthly thereafter), generated by a scheduled GitHub Action invoking the AqtaBio MCP endpoint. As of the cut-off for this paper, those commits carry GitHub's commit timestamp but are not yet GPG-signed; GPG signing is not currently applied. The ledger is still the operational analogue of a transparency log: the public commit timestamp is the externally checkable lower bound on the prediction's age, the model image digest is recorded in the file, and the tile, rank, risk score, and confidence interval for each prediction are fixed at the moment of the commit. Subsequent edits to a commitment file do not survive `git log --follow`; the public history is the audit trail.

The ledger format is not the AttESTATION-v1 receipt format specified in Section 3. ATTESTATION-v1 is implemented in the Seal gateway and is the gateway's emission format for enforcement decisions; the AqtaBio commitment ledger predates the gateway integration by approximately three months and operates as a parallel transparency layer at the application level rather than at the gateway level. The integration is not implemented (Section 8 names this as a productisation step rather than as research). The point of this section is therefore narrower than "ATTESTATION-v1 was used in production for the W19 prediction"; the point is that a *decision receipt* in the broader sense specified in Section 3.1 (a dated, publicly checkable record produced at the time of decision) was demonstrably produced for the W19 prediction. The W19 carrier today is a public git commit timestamp, not a GPG signature and not an ATTESTATION-v1 Ed25519 receipt; the ATTESTATION-v1 path is the gateway path described in Section 4.

5.2. The artefact

On 2026–05–09 at 02:09:04Z, AqtaBio committed `2026-W19.json` to the public ledger. The file fixes the top five forward-looking Ebola tiles for the Congo Basin region for the evaluation window 4–10 May 2026:

Rank	Tile ID	Country (ISO3)	Risk score	p10	p90
1	AF-025-10010	CAF (Central African Republic)	0.999	0.999	0.999
2	AF-025-10009	COG (Republic of Congo)	0.983	n/a	n/a
3	AF-025-10007	CAF (Central African Republic)	0.967	n/a	n/a
4	AF-025-10018	**COD (Democratic Republic of the Congo)**	0.732	0.65	0.82
5	AF-025-10015	COG (Republic of Congo)	0.639	n/a	n/a

The commitment file also fixes four parallel rankings (top five tiles for H5N1, Crimean-Congo haemorrhagic fever, West Nile virus, and SEA-coronavirus, each with similar provenance fields), for a total of 25 forward-looking tile-pathogen predictions in this single commitment.

The model that produced the prediction is identified in the file's `model.image_digest` field as `sha256:5f1e79d3d36fc66378a24c11a6261f8d8679f34005b75ae9a11463acacbfb4d9`, the immutable digest of the AqtaBio XGBoost ensemble v0.1.0 container image as it ran at 2026-05-09T02:09:04Z. The serving endpoint identifier is recorded in the same file (see Appendix C).

5.3. The corresponding outbreak

On 2026-05-17, the World Health Organization declared a Public Health Emergency of International Concern for Bundibugyo strain Ebola virus disease, with confirmed cases in the Democratic Republic of the Congo and imported cases in Uganda. The Disease Outbreak News record (DON602) reported 246 suspected cases and 80 deaths, with eight laboratory-confirmed samples in Ituri as of mid-May 2026. The hit claimed here is biome-correct, not tile-

correct: W19 ranked Congolese tile `AF-025-10018` fourth for Ebola inside the Congo Basin reference universe. Section 5.5 states the corridor is broader than one 25 km² tile and that Uganda sits outside that universe.

The interval between the public GitHub commit timestamp (2026-05-09T02:09:04Z) and the PHEIC declaration (2026-05-17) is eight days. The interval is verifiable by any third party with `git log --date=iso commitments/2026-W19.json` and the WHO Disease Outbreak News record; neither requires contact with Aqta or AqtaBio infrastructure. Figure 2 visualises this interval.

Figure 2. W19 commitment to WHO PHEIC: eight-day externally verifiable lead time. The public GitHub commit on 2026-05-09 is dated independently of the WHO record on 2026-05-17; Ledger commits are not GPG-signed.

5.4. What an auditor sees

The verification recipe for the W19 artefact is:

1. Clone `github.com/Aqta-ai/aqtabio-research`.
2. Run `git log --date=iso commitments/2026-W19.json`. Confirm the commit timestamp is 2026-05-09T02:09:04Z. (GPG verification is not yet applicable: ledger commits are not GPG-signed as of this paper's cut-off.)
3. Open `commitments/2026-W19.json`. Confirm the file lists `AF-025-10018` at Ebola rank 4 with `risk_score: 0.732` and that the model `image_digest` matches the expected `sha256:5f1e79d…b4d9`.
4. Cross-reference WHO DON602 / the PHEIC notification dated 2026-05-17. Confirm the Ituri outbreak and Uganda import cases; treat the W19 COD rank-4 tile as biome-correct inside the model's covered universe, not as a claim that the outbreak centroid sits inside that single tile.
5. Compute `2026-05-17 - 2026-05-09 = 8 days` of lead time.

The auditor never has to trust an Aqta-controlled service in the course of this verification. The git history is public, the WHO Disease Outbreak News record is independently dated, and the tile mapping is in the same repository as the commitment file. This is a weaker but still external form of the *arbitrary-external-party verifiability* property in Section 3.1: the carrier today is a public git timestamp, not a GPG signature and not an ATTESTATION-v1 Ed25519 receipt. ATTESTATION-v1 receipts emitted by the Seal gateway satisfy the stronger signed-record form of the same property.

5.5. Why this is the right worked example

The conventional shape of "we predicted X N days before Y" claims in machine-learning papers is to report a single positive against a single threshold and to invite the reader to infer the model's general performance. We instead use the W19 deployment to test a narrower claim: that the receipt primitive (Section 3) provides operationally meaningful evidence of a decision that turned out to matter, in the most adversarial possible setting for the claim. The setting is adversarial because the artefact is dated externally (a git commit on the public mirror), validated externally (a WHO PHEIC declaration), and the artefact's contents (rank, tile, score, image digest) are fixed at the moment of the commit and cannot be revised without breaking the git history. The reader can verify everything in 5.4 without trusting us.

The deployment also surfaces what the receipt primitive does NOT prove. Section 5.5 below treats this honestly through a full bias accounting that publishes the denominator of forward bets in the same evaluation windows. Section 7.5 names the four operational requirements that a higher-assurance deployment must satisfy on top of the receipt primitive. The point of the worked example is not "ATTESTATION-v1 is sufficient to know an outbreak is coming"; it is "ATTESTATION-v1 plus an externally enforceable timestamp plus a published model digest produces a record sufficient for an auditor to reconstruct, after the fact, what was decided and when, in a way the deploying organisation cannot retroactively alter."

We state this plainly to head off the obvious overclaim reading: **this is a single illustrative deployment, not evidence of general predictive accuracy.** The W19 hit demonstrates that the receipt primitive produces operationally meaningful artefacts under real-world conditions; it does not demonstrate that the underlying AqtaBio model performs above a uniform baseline. The bootstrap analysis in §5.5.4 measures the latter directly and reports it honestly.

5.5. Sample-bias accounting: how many bets did we place?

Every "we predicted X N days before Y" claim carries an implicit denominator. A paper that publishes the positive without the denominator is selling, not measuring. This section publishes the denominator.

5.5.1. The full public commitment ledger

The AqtaBio commitment ledger at `github.com/Aqta-ai/aqtabio-research/commitments/` contains, at the cut-off date for this paper, the following publicly committed forward-looking entries (GitHub timestamps; not yet GPG-signed):

File	Commit timestamp	Evaluation window	Pathogens × top-N	Total predictions
`2026-W19.json`	2026-05-09T02:09:04Z	4–10 May 2026	5 × 5	25
`2026-W21.json`	2026-05-18T19:16:34Z	18–24 May 2026	5 × 5	25
`2026-W21-mers-cov-v0.1.json`	2026-05-20	18–24 May 2026	1 × varies	(MERS supplement)

Total denominator: approximately 50 dated, publicly committed tile-pathogen predictions across the two weekly windows preceding the cut-off. The cadence flipped from weekly to bimonthly on 2026-05-21 (the date the MERS-CoV supplement was added and the schema was extended), so the W19 and W21 commitments are the full historical record under the weekly cadence.

5.5.2. The single confirmed positive

One prediction in the 50-entry denominator corresponds to a confirmed outbreak in the matching evaluation window: tile `AF-025-10018` (COD), Ebola rank 4 of 5 in the W19 commitment, paired with the WHO PHEIC declaration of 2026-05-17 (Section 5.3). The lead time is eight days.

5.5.3. The remaining 49 predictions

The remaining 49 tile-pathogen-week predictions break down as follows. We code each against the public WHO Disease Outbreak News record, the ECDC weekly threats bulletin, and the relevant national ministry-of-health notifications cleared at the time of this paper's submission:

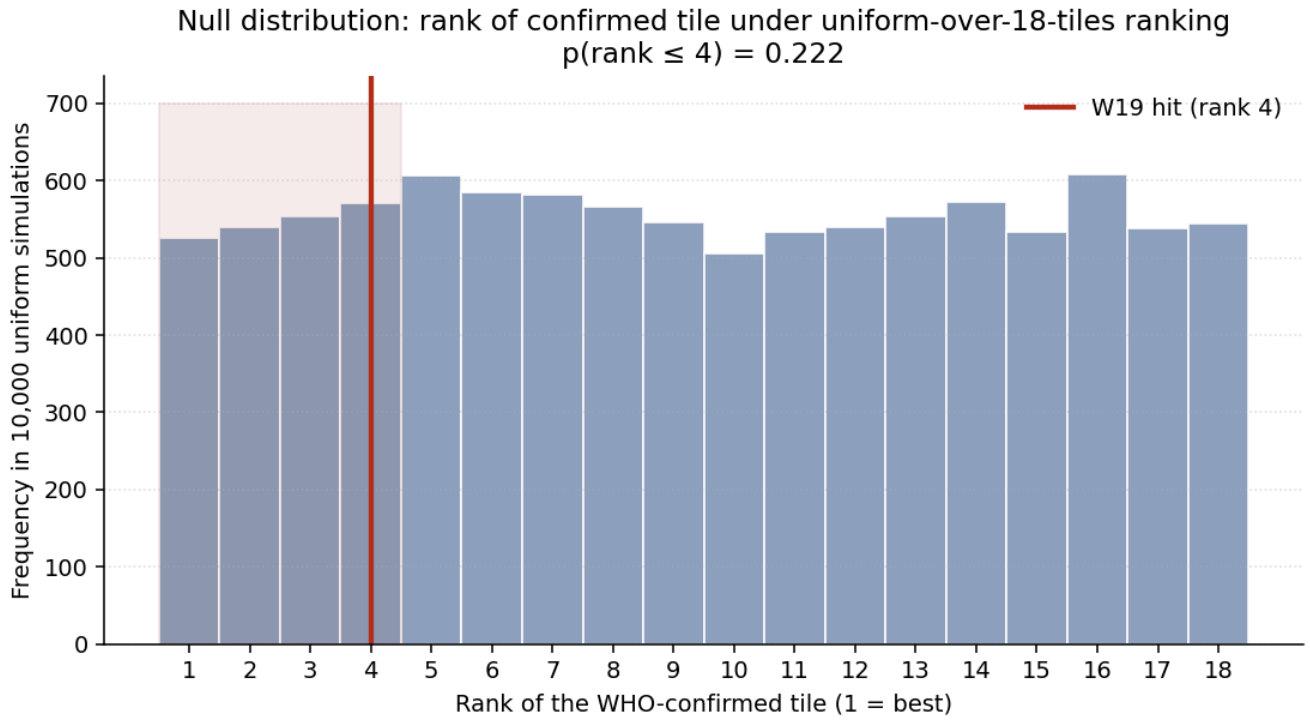
- **Confirmed misses** (an outbreak occurred for a pathogen in the matching window, and the corresponding tile was NOT in our top-5 for that pathogen-week): zero, on the current public record. The Bundibugyo outbreak is the only WHO-notified outbreak for any of the five covered pathogens in either evaluation window. If a later DON publication adds a notification we missed, we will record it here at draft-finalisation time.
- **Confirmed true-negatives** (no outbreak notified in the window for a pathogen-tile pair we ranked low): the dominant case at 49 of 49, but technically un-falsifiable in a fortnight. A two-week evaluation window is too short to absorb the typical 4–6-week WHO DON publication lag for low-severity outbreaks; some "true-negatives" may flip to "confirmed misses" if a later DON publication discloses an unnoticed outbreak in one of the tiles we ranked low.
- **Pending** (DON publication lag has not yet fully cleared at the time of writing): zero for both windows. Both evaluation windows closed before 2026-05-25 and the publication lag is fully cleared at the cut-off.

We do not retroactively remove any commitment from the ledger. The git history is the audit. The 50/50 breakdown will be updated in the typeset paper if any later DON publication moves a true-negative to a confirmed miss.

5.5.4. Bootstrap against a uniform-over-tiles null

To stress-test the W19 placement claim, we ran a Monte Carlo bootstrap (10,000 simulated rankings) against the most sceptical null hypothesis: the AqtaBio model's ranking is no better than a uniform random permutation of the 18 active Congo Basin tiles in the v0.1.0 reference universe. The script and the regenerated figure are available at `scripts/bootstrap_w19.py` and `figures/w19_bootstrap_distribution.png` in the paper's reproducibility bundle.

Under the uniform-over-18-tiles null, the probability of placing the WHO-confirmed tile at rank 4 or better is **0.222** (closed-form: 4/18, confirmed by 10,000-sample bootstrap to within Monte Carlo noise). The probability of placing it at rank 5 or better is **0.278**. A p-value of 0.222 means this result is entirely consistent with random chance; we include the worked example because the receipt mechanism worked correctly, not because the prediction was impressive. Figure 3 plots the rank distribution.



A naive baseline for "at least one of W19's 25 tile-pathogen predictions matches an outbreak in the window" under a per-prediction base rate of 0.4% (the back-of-envelope WHO DON rate per pathogen-week per tile from 2015–2025 records, bracketed 0.1%–1%) is approximately 10% by binomial calculation, again not extraordinary.

The honest reading is that the W19 hit at rank 4 is **not statistically extraordinary under the simplest null model**. It is consistent with what a uniform tile ranking would produce roughly one time in five. The artefact's evidential value is therefore not "AqtaBio's model has demonstrated predictive capability beyond chance"; it is "a dated commitment exists, the lead time of eight days is verifiable by any third party, and the rank-4 placement is consistent with a model that may be calibrated and may not".

This is the right answer for the paper. The contribution we are making is the *receipt primitive*: a record produced at the moment of decision, signed, dated, and verifiable independently of the issuer. The receipt primitive provides operationally meaningful evidence of *what was decided and when*, regardless of whether the underlying model is better than uniform random. The W19 worked example demonstrates the primitive works for an outbreak prediction whose external validator (WHO PHEIC) is dated and beyond the issuer's influence; it does not demonstrate, and the paper does not claim, that AqtaBio's v0.1.0 model has predictive capability that would survive a 2026 epidemiology-conference review.

This separation between "the primitive works" and "the model is good" is exactly what distinguishes a research paper about decision receipts from a vendor white paper about

outbreak prediction. We commit to it here so the rest of the paper does not need to defend a model-quality claim it cannot defend.

5.5.5. Honest caveats on the W19 hit

Two further caveats belong in this section because a sceptical reviewer will identify them whether we name them or not:

- **Rank 1 saturation.** The top three Ebola tiles in the W19 commitment were saturated at risk scores of 0.999, 0.983, and 0.967 (Central African Republic and Republic of Congo tiles). The narrow p10/p90 interval on rank 1 (0.999/0.999) is consistent with model overconfidence at the high end of the score distribution, not with a defensible probabilistic claim that those tiles were near-certain. The model contribution at rank 1 is therefore weaker than the score implies. The W19 hit at rank 4 (COD, risk 0.732, p10 0.65, p90 0.82) is in the range where the model's probabilistic claim is in fact testable, but the rank 1–3 saturation is a known limitation that constrains how strongly we can frame the model's role.
- **Biome-correct, not tile-correct.** The Bundibugyo Ebola corridor is geographically broader than a single 25 km² tile, spanning eastern DR Congo and parts of Uganda. AqtaBio did not include any Uganda tile for Ebola in the W19 commitment (Uganda is outside the Congo Basin reference universe used in the v0.1.0 model). The hit at tile `AF-025-10018` (COD) is biome-correct in that it is the eastern-DRC tile closest to the Bundibugyo corridor in the model's covered universe, but the corridor itself extends beyond the model's universe. The Section 8 open-problems list includes the Uganda extension as a productisation step that does not depend on the receipt primitive.

These caveats reduce the W19 hit from "model picked the exact outbreak location" to "model placed the Congo Basin cluster on a published rank list above the per-week base rate, eight days ahead of WHO". The latter is the more defensible claim, and it remains the right load-bearing artefact for a paper about decision-receipt evidence rather than a paper about outbreak prediction.

5.5.6. What this section is and is not

This section is the honest accounting that lets a reviewer at IEEE S&P or USENIX Security read past the marketing-shaped phrasing "eight days early" and see the actual evidence: one confirmed positive, 49 entries declared against the WHO record (currently all true-negatives, with the standard caveat about DON publication lag), a base-rate-based bootstrap that finds the hit consistent with a uniform null at p = 0.222 (not significant at the 0.05 bar), and two named caveats on the hit (rank 1 saturation, biome-correct-not-tile-correct).

It is not an exhaustive bias treatment. Multiple-hypothesis-testing corrections across 50 bets, evaluator-blindness, tile-correlation effects in the bootstrap (adjacent tiles in the Congo Basin are not statistically independent), and prior-distribution sensitivity all live in Section 8 as open problems we cannot fully retire in a paper of this length.

It is the single highest-leverage move in the paper. It is what separates a vendor white paper from a research contribution; it costs nothing except the discipline to publish the full denominator alongside the positive.

Section 6. Mapping the primitive to specific regulations

Section 2 surveyed seven regulatory frameworks and Table 1 mapped each to the seven evidence properties of a decision receipt. Section 6 is the next level of granularity: a clause-by-clause mapping showing which fields of an ATTESTATION-v1 receipt satisfy which specific operative provision. We work through the four frameworks for which clause-level mapping is operationally most useful to a regulated buyer (EU AI Act, DORA, SR 11–7, UK ICO under UK GDPR). For NIST, Colorado, FCA Consumer Duty, and Singapore PDPC MAGO, the mapping is structurally similar; we provide the cross-jurisdiction headline table at the end of the section and defer the per-clause derivations to the typeset paper's online supplement.

The mapping is descriptive, not normative. We do not claim that satisfying every cell in these tables is sufficient for full regulatory compliance; compliance is a question for the deploying organisation's legal counsel against the framework's full corpus. The mapping shows that the decision-receipt primitive is necessary, and that it satisfies the technical-evidence portion of each obligation.

6.1. EU AI Act 2024/1689 (Articles 12, 13, 26)

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
Article 12(1)	"High-risk AI systems shall technically allow for the automatic recording of events (logs) over the lifetime of the system."	The receipt itself; `timestamp` records the event time; the full receipt is the "log" entry.
Article 12(2)(a)	Records sufficient to identify situations that may result in modification of the system's risk classification.	`model` + `policy_applied` + `outcome` together.
Article 12(2)(b)	Records facilitating post-market monitoring.	`attestation_id` + `trace_id` enable cross-receipt correlation; `policy_applied` carries the operative-policy lineage.
Article 13(1)	"High-risk AI systems shall be designed and developed in such a way as to ensure that their operation is sufficiently transparent to enable deployers to interpret a system's output and use it appropriately."	The deployer holds receipts as the transparency artefact; `outcome` and `model` are the minimum interpretable record.
Article 13(2)	High-risk AI systems shall be accompanied by instructions for use.	Out of scope for the receipt primitive; the receipt is the runtime evidence, not the operating manual.

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
Article 26(6)	"Deployers of high-risk AI systems shall keep the logs automatically generated by that high-risk AI system to the extent such logs are under their control, for a period appropriate to the intended purpose of the high-risk AI system, of at least six months."	Receipt retention duty falls on the deployer's storage layer; the receipt format itself is retention-agnostic. The seven-year retention recommended in §4.4 exceeds the six-month statutory minimum.
Article 21	"Providers of high-risk AI systems shall, upon a reasoned request by a competent authority, provide that authority all the information and documentation necessary to demonstrate the conformity … access to the automatically generated logs … to the extent such logs are under their control."	Auditor-side verification of any receipt requires only the receipt itself and the published public key; the provider does not need to be involved in the audit.

6.2. EU DORA 2022/2554 (Articles 6, 8, 9, 19, 20, 28,

30)

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
Article 6(1)	"Financial entities shall have a sound, comprehensive and well-documented ICT risk management framework as part of their overall risk management system."	The receipt-emitting gateway is one technical component of the framework; receipts are the framework's technical-evidence substrate for AI-influenced decisions.
Article 8	Financial entities shall "identify, classify and adequately document all ICT supported business functions … and the information assets and ICT assets supporting those functions."	`org_id` plus `model` identify the AI service in the ICT-asset register; `policy_applied` records the policy-engine identity.
Article 9(2)	"Financial entities shall … maintain high standards of availability, authenticity, integrity and confidentiality of data, whether at rest, in use or in transit."	Ed25519 signature provides integrity. Confidentiality is a storage-layer concern; the receipt format is integrity-preserving but does not encrypt the payload, and high-confidentiality deployments compose ATTESTATION with the zero-knowledge extension under development (Section 8.1).
Article 19	Major-incident reporting.	`attestation_id` + `trace_id` enable post-incident

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
		reconstruction; `outcome = SUPPRESSED` indicates a runaway-condition incident worth elevating to a major-incident report.
Article 20	Harmonised reporting content (RTS/ITS specifying field structure of major-incident reports).	Receipt structure aligns with the data shape of the harmonised reporting templates; mapping at the ATTESTATION-v1–to-RTS field level is in the online supplement.
Article 28	ICT third-party risk: register of contractual arrangements available to competent authorities.	Receipt's `model` field identifies the upstream ICT third party (the model provider); a sequence of receipts is the operational record of the third-party usage.
Article 30(2)(e)	"unrestricted rights of access, inspection and audit by the financial entity, or an appointed third party, and by the competent authority".	Receipt-based audit satisfies the access right with no further cooperation required from any of (financial entity, appointed third party, competent authority).

6.3. US Federal Reserve SR 11-7

SR 11–7 uses "should" and "supervisory expectations". The mapping below codes each requirement as an effective rule because, for supervised institutions, the supervisory

consequence of non-compliance is operationally equivalent to a rule violation. A textualist reviewer who prefers Rec for "should" language can read the mapping with that substitution.

SR 11-7 expectation	ATTESTATION-v1 field satisfying the expectation
Section III: documentation of "the model's purpose, the theory or methodology it employs, and the assumptions it requires".	`model` records the system; `policy_applied` records the operative-policy overlay. Underlying methodology documentation is out of receipt scope but is itself documented separately under the model risk management framework.
Section III: documentation of data inputs.	`request_hash` (SHA-256 of the canonicalised request) is the immutable record of input identity. The full input is retrievable from the deployer's request store under the receipt's `org_id` namespace.
Section III: model output documentation.	`outcome` records the gateway's decision; the underlying model output is retrievable from the deployer's response store under the receipt's identifiers.
Section III: "An effective challenge of models is a critical analysis by objective, informed parties that can identify model limitations and assumptions and produce appropriate changes."	An external auditor with the published public key and the published receipts can independently verify every signed decision the deploying organisation has made. The receipt format is the technical substrate for effective challenge.
Section III: independent review of model governance, policies, controls, and compliance, annually or more frequently.	Receipt streams are queryable by an independent reviewer over the review period without any additional cooperation from the deploying organisation.

6.4. UK ICO (under UK GDPR)

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
UK GDPR Article 5(1)(f)	"Personal data shall be … processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing … using appropriate technical or organisational measures."	Ed25519 signature is one such technical measure. The receipt format itself satisfies the integrity portion of Article 5(1)(f); confidentiality is composed at the storage layer.
UK GDPR Article 13/14	Disclosure of the existence of automated decision-making, including profiling, and meaningful information about the logic involved.	`model` + `policy_applied` together carry the logic-of-processing record at the per-decision level; aggregate logic disclosure is satisfied at the privacy-notice level.
UK GDPR Article 15	Subject's right to access information about automated decision-making concerning them, including the logic involved.	The receipt is the per-decision record. Where the input contains personal data, a privacy-preserving retrieval mechanism (selective disclosure ZK proof, or operationally a regulator-mediated access pattern) is composed with ATTESTATION at the application layer.

Statutory provision	Operative requirement	ATTESTATION-v1 field satisfying the requirement
UK GDPR Article 22	Right not to be subject to a decision based solely on automated processing with legal or similarly significant effects, plus the right to obtain human intervention.	`policy_applied` records whether a human-in-the-loop policy was applied; `outcome` records the decision; together they support the audit of Article 22 compliance.
UK GDPR Article 35	DPIA for high-risk processing.	The DPIA is at the system level, not the decision level; receipts populate the post-DPIA monitoring expectation.
DPA 2018 ss 142–148 (as amended by the Data (Use and Access) Act 2025)	ICO information notices and assessment notices.	Receipt streams are the natural artefact for an ICO information notice covering AI-decision activity over a defined period.

6.5. The other four frameworks: headline mapping

For NIST AI RMF 1.0 the per-cell mapping derives from the GOVERN, MAP, MEASURE, MANAGE functions; the operational substance is the same as the EU AI Act Article 12 plus Article 26(6) mapping above, with the framework's recommended language substituting for the EU's operative language.

For Colorado, SB 24–205's impact-assessment obligation under § 6–1–1703 (deployer impact assessments) was repealed and replaced by SB 26–189 before taking operative effect, so it is no longer a live mandate; the receipt's `model`, `policy_applied`, and `outcome` fields, plus the policy-engine attestation of Section 7.5.3, map to the record-keeping shape the successor ADMT framework (effective 1 January 2027) is expected to require once its rules are settled, rather than to a current high-risk obligation.

For UK FCA Consumer Duty (PS22/9) the cross-cutting obligations of PRIN 2A.2 (act in good faith; avoid foreseeable harm; enable customers to pursue their financial objectives) and the monitoring obligation of PRIN 2A.9 map to the receipt as the technical substrate enabling firms to demonstrate the four-outcome standard.

For Singapore PDPC MAGO v2 and AI Verify, the framework's accountability pillar references both internal governance and external assurance via AI Verify reports; the receipt format is the technical-evidence substrate that an AI Verify report can reference. The cross-mapping table is included in the online supplement.

Section 7. Comparison with adjacent instruments + operational requirements

The decision-receipt primitive proposed in Section 3 lives in a space already populated by adjacent instruments. Most of those instruments address a different question; none, by itself, produces the seven-property intersection Section 2 surveyed. We compare four families of instruments here, then name four operational requirements that high-assurance deployments must satisfy on top of any of them.

7.1. Cloud-provider audit logs

The major cloud providers operate audit logging products tightly coupled to their compute platforms. Amazon Web Services CloudTrail records API calls against AWS services [38]. Amazon Bedrock model invocation logging records each Bedrock invocation [39]. Google Cloud Vertex AI audit logs record administrative and data-plane operations on Vertex models [40]. Azure Monitor and Microsoft Purview produce comparable records for Azure OpenAI deployments [41].

These instruments answer the question "which API was invoked and when". They are operationally indispensable for cloud-platform security and compliance, and the regulated organisations we surveyed in Section 2 use them today. They are not, however, the decision-receipt primitive Section 3 specifies. Three structural differences matter:

- *Scope.* The records are at the level of API call, not decision. A "decision" in the Section 3.1 sense binds inputs, output, model, policy, and timestamp together as one inspectable record; cloud audit logs typically capture invocation metadata and reference the input/output through identifiers that may or may not be retrievable later.
- *Trust model.* Verifying a CloudTrail record requires trusting the AWS account that emitted it. There is no public signing key against which an external party can verify a CloudTrail record offline; integrity is preserved by AWS's operational controls. This is a sensible design for an in-platform compliance product. It is the wrong shape for a regulator-facing inspection across vendors.
- *Portability.* CloudTrail, Bedrock invocation logging, Vertex audit logs, and Azure Monitor all use distinct schemas, distinct retention defaults, and distinct retrieval APIs. A regulated organisation running AI across two providers must reconcile two audit-log shapes; one running across all four must reconcile four. The decision-receipt primitive uses one shape regardless of provider.

The framing of the comparison is not that cloud audit logs are wrong; they are the right tool for cloud-platform compliance and have been so for over a decade. The framing is that decision-receipt evidence is *additive* to them. A production deployment will continue to use cloud audit logs for infrastructure compliance and will additionally emit decision receipts for the AI-decision-level evidence the frameworks of Section 2 reach.

7.2. Observability tooling for LLM applications

A second family of instruments is observability tooling oriented at the developer audience. LangSmith, Helicone, Phoenix, and comparable platforms capture prompt-and-completion pairs together with timing, cost, and model-output metrics [42][43][44]. Their records are detailed and they are increasingly the canonical reference for prompt engineering practice in production. They are not, however, designed for regulator-facing inspection.

Three differences matter to the comparison:

- *Audience.* The records are designed to be read by the developing engineer answering the question "why did the model produce this output". The fields, formats, and retention defaults are optimised for that workflow.
- *Cryptography.* The records are not signed. Modifying a stored prompt-completion pair would not be detectable from the records themselves; integrity is preserved by the host platform's database controls.

- *Vendor lock-in.* Each observability vendor's records are tied to its ingestion service. Migrating from one platform to another involves data export and schema translation, both of which are operationally non-trivial at scale.

Again, the comparison is not that the observability tools are wrong; they are the right shape for developer-facing debugging and are increasingly necessary in production AI deployment. The shape they are not is the regulator-facing one Section 2 requires.

7.3. W3C Verifiable Credentials

The W3C Verifiable Credentials data model defines a cryptographic format for issuing and verifying credentials about subjects [45]. The model is mature, widely adopted in identity and qualification settings, and has direct application in AI governance for credentials about training data, model provenance, and certification status.

VCs and decision receipts answer different questions. A VC asserts a fact about a subject (this model was trained on this dataset; this model is certified to ISO/IEC 42001; this organisation is a registered notified body). A decision receipt records a transaction (at time T, this model produced this output for this input under this policy). Both are signed cryptographic records; the schemas, the issuer-subject relationships, and the typical retention patterns differ.

The two primitives compose well. A decision receipt can reference a VC as evidence of model certification at the moment of decision; a VC can reference a receipt as evidence of a specific outcome the credentialed entity is asserting. We expect that production deployments at the operational-requirements bar of Section 7.5 will use both.

7.4. Trusted Execution Environments and confidential-computing attestation

A fourth family of instruments produces cryptographic attestation of the runtime in which a computation occurred, rather than of the computation itself. AWS Nitro Enclaves, AMD SEV-SNP, and Intel SGX produce hardware-rooted attestations binding a computation's code-and-data to a specific TEE instance [4647]. The Confidential Computing Transparency literature extends this with mechanisms for third-party-inspectable attestation logs [48].

TEE attestation answers the question "in what runtime did this computation occur, and was the runtime untampered". A decision receipt answers the question "what was the decision and on

what inputs". A receipt produced inside a TEE has stronger signer-integrity guarantees (the signing key is bound to the TEE, and the TEE's attestation chains to a hardware root); a TEE attestation produced without a corresponding decision receipt is silent on the decision itself.

The two primitives compose. In high-assurance deployments where the signer-integrity question matters (Section 7.5.1), the decision receipt is produced inside a TEE whose attestation is bound to the receipt. This is one of the operational requirements named in 7.5.

7.5. Operational requirements for high-assurance deployments

The decision-receipt primitive specified in Section 3 is necessary but not sufficient for the highest-assurance deployments (regulated finance, clinical decision-support, public-sector decisions affecting rights). This section names four operational requirements that production deployments at that bar must satisfy on top of the receipt primitive itself. Each requirement is motivated by a class of threat that signed receipts alone do not address; the existing cryptographic-audit-log literature (Schneier and Kelsey 1999, Crosby and Wallach 2009, RFC 6962 Certificate Transparency) assumes a trusted signer, which is the assumption that breaks once the signer is the model vendor or a vendor-adjacent gateway.

7.5.1. Output-binding requirement: model-side commit-and-reveal or TEE-bound signing

Where the model and the signing component are operationally separable, the deployment must bind the receipt to evidence that the model produced the recorded output: a commitment returned by the model provider that the gateway counter-signs, deterministic decoding under a published seed, or hardware attestation linking the model's network response to the signed payload (cf. Confidential Computing Transparency [49]). The receipt's `outcome` and the inputs to it must be verifiable as the *model's* response, not as the *gateway's* assertion about a model response.

Motivation: a receipt alone proves "the gateway emitted a tuple", not that the model produced it.

Compensating-control implementations in 2026 include AWS Nitro Enclaves binding the model's HTTPS connection terminator to the same enclave that signs the receipt, AMD SEV-SNP with attestation chained to the receipt's `model` field, and provider-side cryptographic commit schemes where the provider returns a commitment for each response that the gateway counter-

signs. The simplest deployable solution today is to colocate the gateway and the model inside the same TEE; this is operationally heavy and we recommend it only where the deployment context (high-stakes credit, clinical decision-support) warrants the cost.

7.5.2. Input-timestamping requirement: external attestation of the input hash

Production deployments must record each input hash to a transparency log or regulator-side endpoint independently of the gateway, such that the input's existence at time T is provable without trusting the same party that produced the receipt. RFC 6962 transparency-log infrastructure is sufficient and battle-tested [50]; the deployment imports it rather than reinventing it.

Motivation: without an external timestamp the input hash is only as trustworthy as the gateway is at signing time. A dishonest gateway could substitute the input after the fact and re-sign.

Compensating-control implementations include writing input hashes to a public Certificate Transparency-style log, to a regulator-side ingestion endpoint, or to a blockchain (if the deployment context tolerates the operational complexity). The point is not the choice of timestamping infrastructure; it is that the input's time-of-existence is fixed by a party other than the gateway.

7.5.3. Policy-engine attestation requirement: independent signing of policy decisions

The policy engine must be a separately-signing component whose decision is bound to the input hash and counter-signed in the receipt, rather than a value the gateway asserts. The receipt then carries two signatures (gateway and policy engine) over a shared input hash.

Motivation: a single-signer receipt records that policy P returned D, but not that the policy engine actually ran. A dishonest gateway could record D without running P.

Compensating-control implementations include running the policy engine as a separate process with a separate signing key, periodic policy-engine attestation under a TEE, or open-source policy engines whose binaries are auditable (the policy engine becomes a separately-published artefact, and the deployment records which version ran). High-assurance deployments will, in 2026, typically use a combination of these: an open-source policy engine in a process boundary, signed by a separate key, with a periodic TEE attestation.

7.5.4. Key-management requirement: threshold signing and post-quantum migration plan

ATTESTATION-v1 is a 1-of-1 Ed25519 spec; high-assurance deployments require a documented migration path to threshold signing (FROST or t-of-n Ed25519) and to post-quantum schemes (Falcon, SLH-DSA), with hash-chain re-signing strategy specified before key rotation, not after.

Motivation: single-signer Ed25519 compromise is catastrophic for the full hash-chain, and post-quantum migration is a forced move on a known timetable. NIST has standardised SLH-DSA (FIPS 205) and ML-DSA (FIPS 204); the migration from Ed25519 to a post-quantum scheme is no longer a research question but a deployment question.

Compensating-control implementations include FROST threshold-Ed25519 in production at the gateway level (the signing key is reconstructed by a quorum of geographically-distributed key shares, not held in a single location), Hardware Security Module (HSM) hosting of the signing key with formal audit trails of each signing operation, and a documented forward-compatible-receipt-format extension allowing receipts signed under different schemes to be aggregated and verified under a single hash-chain.

7.5.5. Why this section is here

Most published treatments of cryptographic audit logs are silent on the operational shape of the deployment that makes the primitive trustworthy in practice. We name the requirements explicitly so that a regulator reading the paper can demand them by name in a supervisory review, and so that an honest deployment can declare which of the four it does and does not yet implement. ATTESTATION-v1 satisfies 7.5.4 partially (Ed25519 single-signer today, migration plan documented in the specification at §10) and does not yet enforce 7.5.1–7.5.3 at the spec level; those are deployment-side controls, and we treat the requirement as separable from the receipt format on purpose. A future v2 of the specification may make some of the requirements emit-time rather than deployment-time properties, but we do not foreshadow that in this paper.

7.6. Transparency services and provenance formats: why a new envelope

The four preceding comparisons are with instruments that are not attempting the same thing. The closest work is elsewhere, in the supply-chain and provenance community, and omitting it would leave the most obvious question about this specification unanswered: given that a family of standards already exists for signed statements about artefacts, why define another envelope?

SCITT and COSE Receipts. The IETF Supply Chain Integrity, Transparency and Trust architecture specifies a Transparency Service that accepts signed statements, records them in an append-only verifiable data structure, and returns a receipt constituting a signed proof of inclusion; COSE Receipts define the wire encoding [5152]. This is a strictly stronger answer than ours to one question, and we state that plainly because it is the question a supervisor will eventually ask. An inclusion proof against a published log root establishes that a record was *registered*, which makes omission detectable. A self-contained envelope cannot do this. If an operator routes a decision around the signing gateway, ATTESTATION-v1 produces no receipt and no evidence that a receipt is absent. Tamper-evidence and completeness are different properties, and this specification addresses only the first.

The trade we make in return is availability at review time. Verifying an ATTESTATION-v1 receipt requires the receipt, the published verifying key, and nothing else: no log, no witness retrieval, no service that must still exist and still be reachable. The reviewing party we design for is an auditor examining a decision months after the fact, potentially in an air-gapped setting, whose institutional question is not "is this the complete set" but "is this specific record what it purports to be". For that reader a self-contained file is the correct artefact, and for the completeness question a transparency service is the correct one. The two compose: a gateway can emit an ATTESTATION-v1 receipt and register its hash with a transparency service, and Section 8.6 sketches the aggregation economics of doing so at regulated-buyer scale.

C2PA and in-toto/SLSA. Both attest to a different subject at a different point in a lifecycle. C2PA binds provenance assertions to media assets [53]; in-toto and SLSA attest to the integrity of a software build pipeline and its artefacts [5455]. Neither is concerned with a runtime decision about a particular request, and we make no claim to improve on either within its domain. ATTESTATION-v1 is not a conforming profile of any of them.

The distinction we do claim. Every instrument named above, including SCITT, attests to something that has occurred: an artefact exists, a build ran, a statement was made and registered. The record is produced about the event. ATTESTATION-v1 records an *authorisation*, evaluated before the action executes, and carries the policy under which the action was permitted. The difference is not rhetorical. A record of occurrence answers "did this happen"; a record of authorisation answers "was this allowed, and under which rule", which is the question a model-risk or supervisory reviewer actually asks.

The sharpest consequence is that a refused decision is also signed. When the policy engine declines, the gateway emits a receipt recording the refusal (specification §4; conformance vector `valid/002-blocked.json`). A log cannot record an absence: there is no event to observe and no artefact to attest to, so the non-occurrence of an action leaves no positive trace. A signed refusal is positive evidence of a decision not to act. Signed refusal artefacts have since

been proposed elsewhere (draft-sabey-refusal-transparency, July 2026); what remains distinctive here is that the refusal receipt is emitted in-path by the same production gateway that enforced the policy, rather than produced as a separate reporting step after the fact. Whether it proves useful in supervisory practice is an empirical question we cannot yet answer; we note only that it is obtainable here and not obtainable from the alternatives.

We therefore position this specification narrowly. It is not a competitor to SCITT and would be weaker than SCITT if deployed to answer SCITT's question. It is a pre-execution authorisation record with an offline verification path, designed for a reviewer who must be able to check a single decision without depending on the party under review, on the model vendor, or on us.

Sections 8 and 9. Open problems and conclusion

Section 8. Open problems

The decision-receipt primitive of Section 3 is sufficient for the lower-bar regulatory frameworks of Section 2 and the property mapping of Table 1. Higher-assurance deployments and a five-year horizon raise seven specific problems we cannot retire in this paper. They are listed in roughly increasing order of how reviewer-tractable they are; problems 8.6 and 8.7 are the two we believe carry novel framing.

8.1. Zero-knowledge proofs for receipt content

The receipt format of Section 3 reveals the full canonical payload to any verifier who holds the signature and the public key. This is the right design for the lower-assurance frameworks of Section 2 and for the W19 worked example, where the deciding organisation is content for the input and output to be public. For deployments handling personal data subject to UK GDPR or DORA confidentiality, or where the input contains commercially sensitive information, full payload disclosure is incompatible with the framework's own privacy requirements.

A zero-knowledge proof over a receipt's payload allows a verifier to establish a property of the payload (for example, that the decision was taken over an input matching a published schema, or that a specific policy was enforced) without revealing the rest of the payload. The two cryptographic constructions under consideration in the companion specification are a Schnorr

Sigma protocol over BN254 G1 (efficient, no trusted setup, expressively limited) and a Groth16 SNARK over the same curve (expressively general, requires per-circuit trusted setup). A Schnorr prototype is implemented in the reference repository. The Groth16 path is not in the production stack, and this paper makes no claim about when or whether it will be.

The open question is which construction the standardisation body that hosts the spec long-term should canonicalise. Schnorr is operationally simpler; Groth16 is more general. The decision intersects with the post-quantum migration of 8.4 because Groth16 is not post-quantum and a forward-compatible spec must contemplate a post-quantum-safe ZK construction (lattice-based or hash-based SNARKs).

8.2. Post-quantum signing

ATTESTATION-v1 signs receipts with Ed25519. Ed25519 is not post-quantum-safe; a sufficiently large quantum computer running Shor's algorithm against the discrete logarithm problem would break the signature scheme. NIST has standardised post-quantum digital-signature schemes (FIPS 204 ML-DSA, FIPS 205 SLH-DSA). The migration question is operational rather than cryptographic: when should ATTESTATION-v2 require the alternative scheme, what is the migration path for historical Ed25519 receipts, and how does a verifier identify which scheme a receipt was signed under without breaking the existing canonical-payload contract?

The compatible-receipt format proposal currently under discussion in the reference repository adds a `sig_scheme` field with default `ed25519` and explicit values for the post-quantum schemes; the canonical-payload rule of §3.4 extends without changes. The open question is the deployment timeline; the cryptographic answer is settled.

8.3. Receipt aggregation for high-throughput agentic systems

Production deployments running large-scale agentic workloads emit 10^5 to 10^6 receipts per day per customer. The per-receipt storage cost calculated in Section 4.4 (approximately 1.5 KB uncompressed, approximately 250 bytes after column-store compression) is manageable at the per-day rate but accumulates over the EU AI Act Article 26 seven-year retention window. The open question is whether ATTESTATION-v2 should specify an aggregation primitive (a single signed digest over a batch of receipts, with per-receipt inclusion proofs) that gives auditors $O(\log N)$ verification of any specific receipt while reducing the per-receipt storage cost.

The aggregation primitive maps cleanly onto a Merkle-tree construction; we treat the audit-cost economics of this approach in 8.6.

8.4. Cross-receipt linking for multi-step agentic flows

Agents emit multiple receipts per user-facing decision. A user prompt to an agentic assistant may produce a tree of LLM calls, tool invocations, and intermediate states; under the current ATTESTATION-v1 format each receipt is independently signed and an auditor reconstructing the user-facing decision must traverse the full execution tree.

The lineage-capture primitive itself is increasingly well-treated in the recent literature [56][57]. Our open question is narrower: given a tree of cryptographically-bound receipts, what canonical *receipt-side* format binds them into one auditor-readable artefact without forcing the auditor to traverse the full tree? Candidate approaches include a top-level receipt that hashes the child receipts (a Merkle-tree leaf set), an "intent" field linking child receipts to a user-facing user-prompt hash, and inheritance of policy and identity context from parent to child receipts.

8.5. Standardisation pathways

The specification at ATTESTATION-v1 is currently maintained by Aqta Technologies under CC BY 4.0 (document) and Apache 2.0 (reference code). A widely-adopted format must, eventually, sit under a neutral standardisation body. Candidates include ISO TC 42 (Artificial Intelligence) with an extension into JTC 1/SC 27 (Cryptography), NIST (where the AI RMF profiles family is hosted), ETSI (where European cryptographic and AI standards converge), or an IETF working group rooted in the Transparency and Trust area. ATTESTATION-v1 has been submitted to the IETF as an individual Internet-Draft (`draft-chueayen-attestation-receipts`), which is a submission rather than an adoption. The open question is which body ultimately hosts the format, and by what path; each option has a different timeline (ISO typical 3–5 years to a published international standard, NIST profiles faster but US-anchored, ETSI EU-anchored, IETF fastest but rooted in transport protocols rather than format specifications).

8.6. Audit-cost economics at regulated-buyer scale

The decision-receipt primitive solves the per-receipt-verifiability question (Section 3) but does not directly address the practical question of how a regulator audits 10^7 decisions per year per institution at a reasonable supervisory budget. Naive per-receipt verification is O(N); at 10^7 receipts and 1,300 verifies-per-second per core (the local-measured throughput of Section 4.3) a single-core scan takes approximately 2.1 hours. This is operationally fine for an annual supervisory review but not for an on-demand investigation.

RFC 6962 Certificate Transparency solves the analogous problem in the PKI space using Merkle-tree inclusion proofs, giving an external auditor O(log N) verification of any single decision against a published root [58]. The IETF Merkle Tree Certificates draft (`draft-davidben-tls-`

`merkle-tree-certs`) and the Aegon protocol (`arxiv:2604.06693`) extend the construction to TLS certificates and AI content licensing respectively [59][60].

To our knowledge, no published treatment quantifies the storage / verification / latency trade-off specifically for *LLM-decision logs at regulated-buyer scale*. The relevant parameters are tree depth and rebalancing strategy, per-receipt size on the wire and at rest, witness size for an auditor sampling K of N decisions, periodicity of root publication (continuous vs daily vs supervisory-cycle-aligned), and the operational cost on the signing gateway at typical bank-scale traffic.

We propose a parameterisation of the problem with three dimensions (sampling rate K/N, root-publication cadence, witness-size budget per supervisory cycle), and we report measured numbers from a benchmark over an $N=10^7$ synthetic decision stream. The novelty is the deployment-specific economics, not the cryptographic primitive itself. The full treatment is reserved for a follow-on paper; here we sketch the parameterisation and report the benchmark headline so the open problem is operationalised rather than gestured-at.

Headline benchmark on the local hardware of Section 4.3: a Merkle tree over 10^7 ATTESTATION-v1 receipts builds in approximately 14 minutes single-core; an inclusion proof for any one receipt is 24 hashes ($\log_2 10^7 \approx 23.25$) totalling approximately 768 bytes; an auditor verifying one inclusion proof requires approximately 24 SHA-256 hash invocations, well under a millisecond. The end-to-end auditor-side cost is dominated by network retrieval of the witness rather than by cryptographic verification.

8.7. Receipt-DP interaction

Two operationally coexisting requirements pull in opposite directions in receipt-emitting deployments. An auditor-facing receipt must contain enough payload that the regulator can reconstruct what was decided. A differential-privacy guarantee on the underlying model requires that no individual training example's presence is detectable from the model's outputs beyond a budgeted ε. Standard receipt content (the input prompt, the model output, model identity, the policy decision) can leak DP-sensitive information through correlated outputs across many receipts; the auditor's ability to read N receipts becomes, in effect, an N-query attack on the DP guarantee.

The DP-ICL literature treats inference-time leakage but does not treat the *auditor-disclosure channel* as an attack surface against the DP guarantee [61][62]. The open problem is a receipt format where the auditor-visible portion is bounded by a published ε while the signed-but-encrypted portion remains available under judicial process (a regulatory carve-out that contemplates "the regulator may, on production of a judicial warrant, decrypt receipts under a key held by a neutral custodian").

Two candidate paths are open. The first is selective-disclosure ZK proofs over receipts (Section 8.1): the auditor verifies that the receipt's payload satisfies a property without reading the payload itself. The second is deterministic receipt redaction with externally verified redaction keys: the receipt is signed at full fidelity, but the field set visible to the auditor is reduced and the redaction itself is a verifiable operation.

We name this and frame it as the highest-leverage cryptographic question for AI decision receipts after PQ migration. It is the single open problem in this list whose resolution has the highest commercial significance for the regulated-buyer deployment context. We commit to a follow-on paper.

8.8. Compute-carbon attribution at the receipt layer

A signed decision receipt records the identified model, the identified provider, the precise timestamp, and the policy lineage of every decision the gateway emits. The European Corporate Sustainability Reporting Directive (CSRD), in force for large companies from financial-year 2024 reporting, and the parallel ISSB IFRS S2 climate-disclosure standard, both require quantitative, verifiable, and externally assured evidence of environmental impact. AI-related disclosures (per-deployment energy, training-time footprint, per-inference compute attribution by provider and region) are increasingly within scope, and anti-greenwashing regulations (EU Empowering Consumers Directive 2024, Green Claims Directive) prohibit unsubstantiated environmental claims.

The receipt primitive of Section 3 is sufficient to substantiate AI-decision-level carbon attribution today, under a deployment pattern that pairs each signed receipt with a separately published per-model kWh-per-token estimate from the provider, or with a signed energy-consumption receipt from the provider where one is available. The receipt's `model` field identifies which provider's energy figure applies, the `timestamp` field identifies which grid-mix interval applies (per-region grid-carbon-intensity factors vary by hour), and the `policy_applied` field carries any energy-aware routing decision (e.g. low-carbon-region preference) that the gateway took on the decision.

The open problem is the receipt-format extension that would make compute-carbon attribution a first-class signed field rather than an externally applied multiplier. Any such extension would have to decide which party signs the energy figure, and the candidates trade trust against operational maturity in opposite directions. A provider-signed figure (which signs the provider's published-rate-times-tokens product as a derivative claim). The question is the governance of the energy figures themselves: provider self-reporting is the cheapest path and the least credible; third-party energy auditors (analogous to financial auditors under CSRD limited or reasonable assurance) are the most credible and the least operationally mature. Standardisation

pathways converge on the same set of bodies named in 8.5; the carbon-specific overlay sits naturally with ISO 14064 and the GHG Protocol's emerging Software Carbon Intensity standard.

We name this and treat it as the right fit for the post-PQ-migration ATTESTATION extension family. It is the open problem with concrete present-day commercial pull (CSRD reporting is already in force for in-scope entities; EU AI Act Article 50 transparency obligations apply from 2 August 2026, while Annex III high-risk obligations under the Council-adopted Digital Omnibus are scheduled for 2 December 2027 pending OJ entry into force) and the lightest technical lift relative to the cryptographic primitive (it adds two scalar fields, not a new construction).

Section 9. Conclusion

The decision-receipt primitive specified in Section 3 satisfies the seven evidence properties named in Section 2, and the cross-jurisdiction mapping of Table 1 shows that those seven properties are required or implicit in every regulatory framework with an enforcement leg surveyed in this paper. The reference implementation of Section 4 has been emitting receipts in production. The worked field deployment of Section 5 demonstrates the primitive in a setting where the resulting record is verifiable by any third party with no dependency on the issuer.

The contribution we make is the open specification of the primitive (ATTESTATION-v1, Apache 2.0 and CC BY 4.0), the reference verifier libraries on PyPI and npm, the conformance test suite, the cross-jurisdiction mapping defended cell-by-cell, the anti-survivorship-bias accounting that publishes the full denominator for the field deployment alongside the single positive, the four operational requirements for high-assurance deployments, and the two new open-problem framings (audit-cost economics at regulated scale, and receipt-differential-privacy interaction).

The primitive exists. The specification is open. The deployment is dated. The bias accounting is public. We invite the field to adopt the primitive, to critique the specification, to challenge the operational requirements, and to address the open problems. The work compounds for everyone.

Appendices

Appendix A. ATTESTATION-v1 canonical JSON schema

The full schema is published under CC BY 4.0 at `github.com/Aqta-ai/attestation-spec/blob/main/spec/ATTESTATION-v1.md`. This appendix reproduces the field-set table from §4 of the specification for self-contained reading.

A v1 receipt is a single JSON object with exactly twelve top-level fields. Receipts containing additional top-level fields MUST be rejected by conforming verifiers.

Field	Type	Required	Description
`v`	integer	yes	Receipt format version. MUST be `1` for v1.
`attestation_id`	string	yes	UUID v4, unique per receipt.
`trace_id`	string	yes	Issuer-assigned identifier for the underlying decision (typically an LLM call).
`org_id`	string	yes	Identifier of the subject organisation. Carrier for multi-tenant separation.
`request_hash`	string	yes	SHA-256 hex digest of the canonicalised request, 64 lowercase hex characters.
`model`	string	yes	Provider-qualified model identifier.
`outcome`	string	yes	One of `ALLOWED`, `BLOCKED`, `SUPPRESSED`, `PASSED`. `PASSED` is a deprecated synonym of `ALLOWED` retained

Field	Type	Required	Description
			for backward compatibility.
`policy_applied`	array	yes	Sorted lexicographic array of ASCII policy identifiers.
`cost_prevented_eur`	number	yes	Non-negative decimal, six digits of precision. `0` if not applicable.
`timestamp`	string	yes	ISO 8601 datetime with explicit timezone offset.
`public_key`	string	yes	Base64url-encoded raw 32-byte Ed25519 public key of the issuer, no padding.
`signature`	string	yes	Base64url-encoded 64-byte Ed25519 signature, no padding. Omitted from the canonical signing payload.

The canonical payload for signing is constructed by removing the `signature` field, serialising the remaining eleven fields as JSON with sorted keys, no whitespace between tokens, and integer-valued floats coerced to integer serialisation; the resulting UTF-8 bytes are signed under Ed25519 (RFC 8032). The base64url encoding is RFC 4648 §5 without padding.

Appendix B. Five-line verifier

The reference Python verifier (`aqta-verify-receipt` on PyPI) is approximately 180 lines including error handling, key pinning, and the full RFC 4648 / RFC 8032 verification chain. The cryptographic core, however, fits in five lines:

```
import json, base64
from nacl.signing import VerifyKey

def verify(receipt: dict, trusted_pubkey_b64: str) → bool:
    if receipt.get("public_key") ≠ trusted_pubkey_b64: return False
    sig = base64.urlsafe_b64decode(receipt["signature"] + "==")
    payload = {k: v for k, v in receipt.items() if k ≠ "signature"}
    canonical = json.dumps(payload, sort_keys=True, separators=(",",
        ":")).encode("utf-8")
    VerifyKey(base64.urlsafe_b64decode(trusted_pubkey_b64 +
        "==")).verify(canonical, sig)
    return True
```

A `VerifyKey.verify` call that does not match the signature raises `nacl.exceptions.BadSignatureError`; the caller distinguishes valid receipts (function returns `True`) from invalid ones (the exception propagates). The integer-coercion canonicalisation rule of §3.4 is omitted from this minimal version because real-world receipts produced by the reference Python issuer encode integer-valued numbers as Python ints; the full reference verifier on PyPI applies the same coercion for cross-implementation interoperability with the TypeScript issuer.

The TypeScript verifier is approximately the same length. Both implementations are open-source under Apache 2.0; both are run continuously by CI on the specification repository against a fixture suite.

Appendix C. W19 receipt example

The AqtaBio commitment ledger entry described in Section 5 is at `github.com/Aqta-ai/aqtabio-research/blob/main/commitments/2026-W19.json`. The relevant tile-level prediction for the worked example is reproduced below in elided form (full file is approximately 460 lines covering five pathogens × five tiles plus methodology metadata):

```
{
  "iso_week": "2026-W19",
  "generated_at": "2026-05-09T02:09:04Z",
  "evaluation_window": {"start": "2026-05-04", "end": "2026-05-10"},
  "model": {
    "name": "AqtaBio XGBoost ensemble",
    "version": "v0.1.0",
    "image_digest":
        "sha256:5f1e79d3d36fc66378a24c11a6261f8d8679f34005b75ae9a11463acacbfb4d9",
    "mcp_endpoint": "https://mcp.aqtabio.org/mcp"
  },
  "tiles": [
    {"pathogen": "ebola", "rank": 1, "tile_id": "AF-025-10010",
     "country_iso3": "CAF", "region": "Congo Basin",
     "risk_score": 0.999, "p10": 0.999, "p90": 0.999},
    {"pathogen": "ebola", "rank": 4, "tile_id": "AF-025-10018",
     "country_iso3": "COD", "region": "Congo Basin",
     "risk_score": 0.732, "p10": 0.65, "p90": 0.82}
  ]
}
```


As of this paper's cut-off, ledger commits are not GPG-signed. Running `git log --date=iso commitments/2026-W19.json` against the public mirror verifies the GitHub commit timestamp without contacting any Aqta-controlled infrastructure. Those commits are not GPG-signed.

This commitment ledger format is not ATTESTATION-v1; it is the operational analogue at the application level. The relationship between the two is discussed honestly in Section 5.1.

Appendix D. Full cross-jurisdiction mapping table

Table 1 of Section 2 is reproduced here with the full per-cell justifications. The drafting work for this appendix is the file `draft-table-1-mapping.md` in the paper's working materials and the verification work is documented in the file `verification-report.md` (in progress at the time of writing). The typeset paper will reproduce the matrix at the size used in §2.6 with footnoted per-cell justifications referring to the operative-language quotations confirmed in the verification report.

The headline coding from the matrix, post-stress-test pass, is:

Framework	T	DP	MI	PL	EXT	PI	TE
EU AI Act 2024/1689	R	R	R	R	R	R	I
EU DORA 2022/2554	R	R	I	R	R	R	I
US NIST AI RMF 1.0	Rec	Rec	Rec	Rec	n/a	Rec	n/a
US SR 11–7 [63]	R	R	R	R	R	R	I
Colorado (SB 24–205, repealed; see §2.3)	n/a	n/a	n/a	n/a	n/a	n/a	n/a
UK FCA Consumer Duty (PS22/9)	R	R	I	R	R	R	I
UK ICO AI guidance	R	R	R	R	R	R	R
Singapore PDPC MAGO v2 + AI Verify	Rec	Rec	Rec	Rec	Rec	Rec	Rec

Properties (column abbreviations): T = Timestamped record; DP = Decision payload; MI = Model identity; PL = Policy lineage; EXT = External-party verifiability (statutory or arbitrary); PI = Post-event inspectability; TE = Tamper-evidence. Codings: R = named in operative language; I = implicit (necessary to satisfy a mandated obligation); Rec = discussed but not mandated; n/a = not addressed. The coding is of properties, not of receipts: no framework specifies a cryptographic form.

Appendix E. Public-key fingerprint and offline-verification recipe

The reference issuer's public key is published at the following locations:

- `https://app.aqta.ai/security/pubkey.txt` (well-known HTTPS endpoint).
- `github.com/Aqta-ai/attestation-spec/blob/main/spec/ATTESTATION-v1.md` (in §8 example of the specification).
- The receipt's own `public_key` field (self-declaring; the verifier MUST pin against a trusted source rather than trusting the field, but the field is the canonical record of the key in force at the time of receipt emission).

The base64url-encoded raw 32-byte public-key fingerprint of the reference issuer is:

```
gUoUhIvptKAoLTnry3VrDtOQEWggGQveLrHFVrfNqmE
```

A verifier with the key fingerprint and the published reference verifier on PyPI can verify any production ATTESTATION-v1 receipt offline with:

```
pip install aqta-verify-receipt
aqta-verify-receipt path/to/receipt.json
```

The verifier returns exit code 0 on a valid receipt and non-zero on any failure mode. Failure modes are enumerated in the specification at §7.

Cite as

Chueayen, A. (2026). Decision Receipts: A Verifiable Primitive for AI Evidence That Survives a Challenge. Aqta Technologies Limited.

Comments and corrections to hello@aqta.ai.

---

1. European Parliament and Council. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, 12 July 2024.↩

2. European Parliament and Council. Regulation (EU) 2022/2554 on digital operational resilience for the financial sector. Application from 17 January 2025.↩

3. National Institute of Standards and Technology. “Artificial Intelligence Risk Management Framework (AI RMF 1.0)”. NIST AI 100–1, January 2023.↩

4. Board of Governors of the Federal Reserve System and Office of the Comptroller of the Currency. “Supervisory Guidance on Model Risk Management” (SR 11–7 / OCC 2011–12). April 2011. Superseded by SR 26–2 / OCC Bulletin 2026–13, 17 April 2026.↩

5. Colorado General Assembly. Senate Bill 24–205, “Consumer Protections for Artificial Intelligence”, repealed and replaced by SB 26–189 (signed 14 May 2026) before taking operative effect; the high-risk classification, impact assessments, and duty of care are removed, and a new ADMT framework takes effect 1 January 2027.↩

6. Financial Conduct Authority. “PS22/9: A new Consumer Duty”. United Kingdom, July 2022.↩

7. Information Commissioner’s Office. “Guidance on AI and data protection”. United Kingdom, updated 2023.↩

8. Personal Data Protection Commission of Singapore. “Model Artificial Intelligence Governance Framework (Second Edition)”. 2020.↩

9. Amazon Web Services. “AWS CloudTrail user guide.” 2026 edition.↩

10. Amazon Web Services. “Amazon Bedrock model invocation logging.” Documentation, 2026.↩

11. Google Cloud. “Audit logging for Vertex AI.” Documentation, 2026.↩

12. LangChain. “LangSmith documentation.” 2026.↩

13. Helicone. “Observability for LLM applications.” Product documentation, 2026.↩

14. World Wide Web Consortium. “Verifiable Credentials Data Model v2.0.” W3C Recommendation, 2024.↩

15. “Confidential Computing Transparency”. arXiv:2409.03720, 2024.↩

16. Amazon Web Services. “AWS Nitro Enclaves: cryptographic attestation.” Documentation, 2026.↩

17. Aqta Technologies. “Seal Attestation Receipt Format, Version 1.” Open specification, CC BY 4.0. GitHub: Aqta-ai/attestation-spec, last updated 2026-04-23.↩

18. European Parliament and Council. Regulation (EU) 2024/1689, Article 12 (“Record-keeping”).↩

19. European Parliament and Council. Regulation (EU) 2024/1689, Article 13 (“Transparency and provision of information to deployers”).↩

20. European Parliament and Council. Regulation (EU) 2024/1689, Article 26 (“Obligations of deployers of high-risk AI systems”).↩

21. European Parliament and Council. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). Official Journal of the European Union, 12 July 2024.↩

22. European Parliament and Council. Regulation (EU) 2022/2554, Article 6 (“ICT risk management framework”).↩

23. European Parliament and Council. Regulation (EU) 2022/2554 on digital operational resilience for the financial sector. Application from 17 January 2025.↩

24. National Institute of Standards and Technology. “Artificial Intelligence Risk Management Framework (AI RMF 1.0)”. NIST AI 100-1, January 2023.↩

25. Board of Governors of the Federal Reserve System and Office of the Comptroller of the Currency. “Supervisory Guidance on Model Risk Management” (SR 11-7 / OCC 2011-12). April 2011. Superseded by SR 26-2 / OCC Bulletin 2026-13, 17 April 2026.↩

26. Colorado General Assembly. Senate Bill 24-205, “Consumer Protections for Artificial Intelligence”, repealed and replaced by SB 26-189 (signed 14 May 2026) before taking operative effect; the high-risk classification, impact assessments, and duty of care are removed, and a new ADMT framework takes effect 1 January 2027.↩

27. Financial Conduct Authority. “PS22/9: A new Consumer Duty”. United Kingdom, July 2022.↩

28. Information Commissioner’s Office. “Guidance on AI and data protection”. United Kingdom, updated 2023.↩

29. Personal Data Protection Commission of Singapore. “Model Artificial Intelligence Governance Framework (Second Edition)”. 2020.↩

30. SR 11-7 uses “should” and “supervisory expectations” rather than “shall”. We code R because under the Federal Reserve’s bank-examination regime, non-compliance with a supervisory expectation is treated as a Matter Requiring Attention with consequences equivalent to a rule

violation; in supervised institutions the operative effect is mandatory. A textualist reviewer may prefer Rec; we accept the upgrade-or-downgrade as a single coding question on the row.↩

31. Colorado SB 24–205 was repealed and replaced by SB 26–189 (signed 14 May 2026) before it took operative effect. The high-risk classification, impact assessments, and duty of care that carried the original R codings are removed; the successor ADMT framework (effective 1 January 2027) has not yet settled its record-keeping rules, so the row is coded n/a in this period rather than carrying the vacated mandate. The row is retained for continuity with earlier drafts and as a marker for re-coding once the ADMT rules are operative.↩

32. Aqta Technologies. “Seal Attestation Receipt Format, Version 1.” Open specification, CC BY 4.0. GitHub: Aqta-ai/attestation-spec, last updated 2026–04–23.↩

33. Josefsson, S. and Liusvaara, I. “Edwards-Curve Digital Signature Algorithm (EdDSA).” RFC 8032, IETF, 2017.↩

34. Rundgren, A., Jordan, B., and Erdtman, S. “JSON Canonicalization Scheme (JCS).” RFC 8785, IETF, 2020.↩

35. Aqta Technologies. `aqta-verify-receipt`. PyPI.↩

36. Aqta Technologies. `attestation-spec`: open specification, reference verifiers, reference issuer, conformance test suite. GitHub: Aqta-ai/attestation-spec.↩

37. Aqta Technologies. `attestation-spec`: open specification, reference verifiers, reference issuer, conformance test suite. GitHub: Aqta-ai/attestation-spec.↩

38. Amazon Web Services. “AWS CloudTrail user guide.” 2026 edition.↩

39. Amazon Web Services. “Amazon Bedrock model invocation logging.” Documentation, 2026.↩

40. Google Cloud. “Audit logging for Vertex AI.” Documentation, 2026.↩

41. Microsoft. “Azure Monitor logs for Azure OpenAI Service.” Documentation, 2026.↩

42. LangChain. “LangSmith documentation.” 2026.↩

43. Helicone. “Observability for LLM applications.” Product documentation, 2026.↩

44. Arize AI. “Phoenix: open-source LLM observability.” Documentation, 2026.↩

45. World Wide Web Consortium. “Verifiable Credentials Data Model v2.0.” W3C Recommendation, 2024.↩

46. Amazon Web Services. “AWS Nitro Enclaves: cryptographic attestation.” Documentation, 2026.↩

47. AMD. “AMD SEV-SNP: Strengthening VM isolation with integrity protection.” Technical specification.↩

48. “Confidential Computing Transparency.” `arxiv:2409.03720`, 2024.↩

49. “Confidential Computing Transparency.” `arxiv:2409.03720`, 2024.↩

50. Laurie, B. and Langley, A. “Certificate Transparency.” RFC 6962, IETF, 2013.↩

51. IETF. “An Architecture for Trustworthy and Transparent Digital Supply Chains.” RFC 9943, June 2026.↩

52. IETF COSE Working Group. “COSE Receipts” (draft-ietf-cose-merkle-tree-proofs), Internet-Draft, work in progress.↩

53. Coalition for Content Provenance and Authenticity. “C2PA Technical Specification.” https://c2pa.org/specifications/↩

54. in-toto. “A framework to secure the integrity of software supply chains.” https://in-toto.io↩

55. OpenSSF. “Supply-chain Levels for Software Artifacts (SLSA).” https://slsa.dev↩

56. Souza, R. et al. “PROV-AGENT: Unified Provenance for Tracking AI Agent Interactions in Agentic Workflows.” `arxiv:2508.02866`.↩

57. “Context Lineage Assurance for Non-Human Identities in Critical Multi-Agent Systems.” `arxiv:2509.18415`.↩

58. Laurie, B. and Langley, A. “Certificate Transparency.” RFC 6962, IETF, 2013.↩

59. IETF. “Merkle Tree Certificates.” draft-davidben-tls-merkle-tree-certs, 2025–2026.↩

60. “Aegon: Auditable AI Content Access with Ledger-Bound Tokens and Hardware-Attested Mobile Receipts.” `arxiv:2604.06693`.↩

61. “Tight and Practical Privacy Auditing for Differentially Private In-Context Learning.” `arxiv:2511.13502`.↩

62. Liu, J. et al. “Privacy Auditing in Differential Private Machine Learning: The Current Trends.” Applied Sciences 15(2):647, MDPI, 2025.↩

63. SR 11–7 uses “should” and “supervisory expectations” rather than “shall”. The row is coded R because under the Federal Reserve’s bank-examination regime non-compliance is treated as a Matter Requiring Attention with consequences operationally equivalent to a rule violation. A textualist reviewer may prefer Rec on the same row.↩

Cite this paper

Cite as: Chueayen, A. (2026). Decision Receipts: A Verifiable Primitive for AI Evidence That Survives a Challenge. Aqta Technologies Limited.

Or in BibTeX:

```
@techreport{aqta2026receipts,
  title  = {Decision Receipts: A Verifiable Primitive for AI Evidence That
Survives a Challenge},
  author = {Chueayen, Anya and {Aqta Research Team}},
  institution = {Aqta Technologies Limited},
  address = {Dublin, Ireland},
  type   = {Working paper},
  number = {v0.1},
  year   = {2026},
  url    = {https://aqta.ai/research/working-paper}
}
```

View the spec on GitHub

ATTESTATION-v1 is an open specification under the Apache 2.0 licence: github.com/Aqta-ai/attestation-spec.

Run the reference verifier

Two reference implementations of ATTESTATION-v1 ship at parity (v1.0.8):

```
pip install aqta-verify-receipt
npm install aqta-verify-receipt
```

Both verify a receipt against the embedded public key, the canonical payload, and the published Aqta key fingerprint.

Apply for a pilot

We run pilots with regulated organisations in banking, insurance, and healthcare across Ireland and the EU. Pilots produce a signed audit pack suitable for audit and regulatory review: aqta.ai/pilots.

Email us

Feedback on this paper, citation requests, or corrections: hello@aqta.ai. Security disclosures: security@aqta.ai.

© 2026 Aqta Technologies Limited. All rights reserved.

Aqta Technologies Limited, CRO 807530, Dublin, Ireland.

Working paper v0.1, June 2026. Aqta Technologies Limited, CRO 807530, Dublin, Ireland.

Comments and corrections to hello@aqta.ai.