International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(22s) 2026 pp. 737-758
https://doi.org/10.70917/ijcisim-2026-5474
Article

Compliance-as-Code, Grounded in Observability: Continuous Controls Monitoring in AI-Driven Banking

Yashaswini Nalla

Independent Researcher, India
Email: ynalla.tech@gmail.com

Abstract: Banks increasingly place machine learning at the point of decision, and those systems change continuously through retraining, threshold tuning and challenger promotion. Control assurance has not kept pace. Controls are still validated discretely, at a quarterly review, an annual validation or a single audit sample, so an institution can hold a clean validation opinion and still be unable to demonstrate that a control operated on every decision taken between checkpoints. This paper presents a reference model for compliance-as-code grounded in observability, and a proof of concept that instantiates part of that model and is evaluated empirically. The reference model contributes a three-layer obligation to control to signal taxonomy with layered ownership, a machine-readable control signal descriptor that is the executable control rather than documentation about one, a many-to-many obligation mapping that makes coverage gaps visible by inspection, and a five-level maturity model. The proof of concept instruments a credit decisioning service with distributed tracing, evaluates five controls against live telemetry, and writes every verdict to a hash-chained, signed, append-only ledger. Across 1,220 decisions it produced 5,722 signed evidence records with no gaps, control observability coverage of 1.0, a baseline false-positive rate of 0.0, full-chain integrity verification, correct detection of a deliberately altered record, and per-control detection latency of 0 to 66 decisions. All evaluation used public and synthetically generated data.

Keywords: compliance-as-code, continuous controls monitoring, observability, OpenTelemetry, AI governance, model risk management, regulatory technology, continuous assurance, tamper-evident audit, algorithmic fairness, explainability, model drift, data drift, data lineage, agentic AI governance, policy-as-code, AI-driven banking

1. INTRODUCTION

A bank that puts a model at the point of decision acquires two obligations at once. It must make defensible decisions, and it must be able to show that the controls governing those decisions were operating. The first obligation is well served. Model validation, challenger testing, fairness measurement and explanation methods are mature enough that a competent institution can establish, at a moment in time, that a model behaves acceptably. The second obligation is served badly, and the reason is structural rather than a matter of diligence.

Modern decisioning systems change continuously. A model is retrained on fresh data. A score threshold is tuned in response to portfolio performance. A feature pipeline is refreshed. A challenger is promoted to champion. Input distributions shift without anyone acting at all. Against this, control assurance is organised around discrete events: a quarterly control review, an annual model validation, a sample of decisions pulled for an audit. Each of those events produces a defensible statement about a point in time. None of them produces a statement about the interval between points in time.

That gap is the subject of this paper. Its thesis is a single sentence. The bank's failure is rarely that it made a wrong decision; it is that it cannot demonstrate the control operated on every decision between checkpoints. Figure 1 shows the shape of the problem.

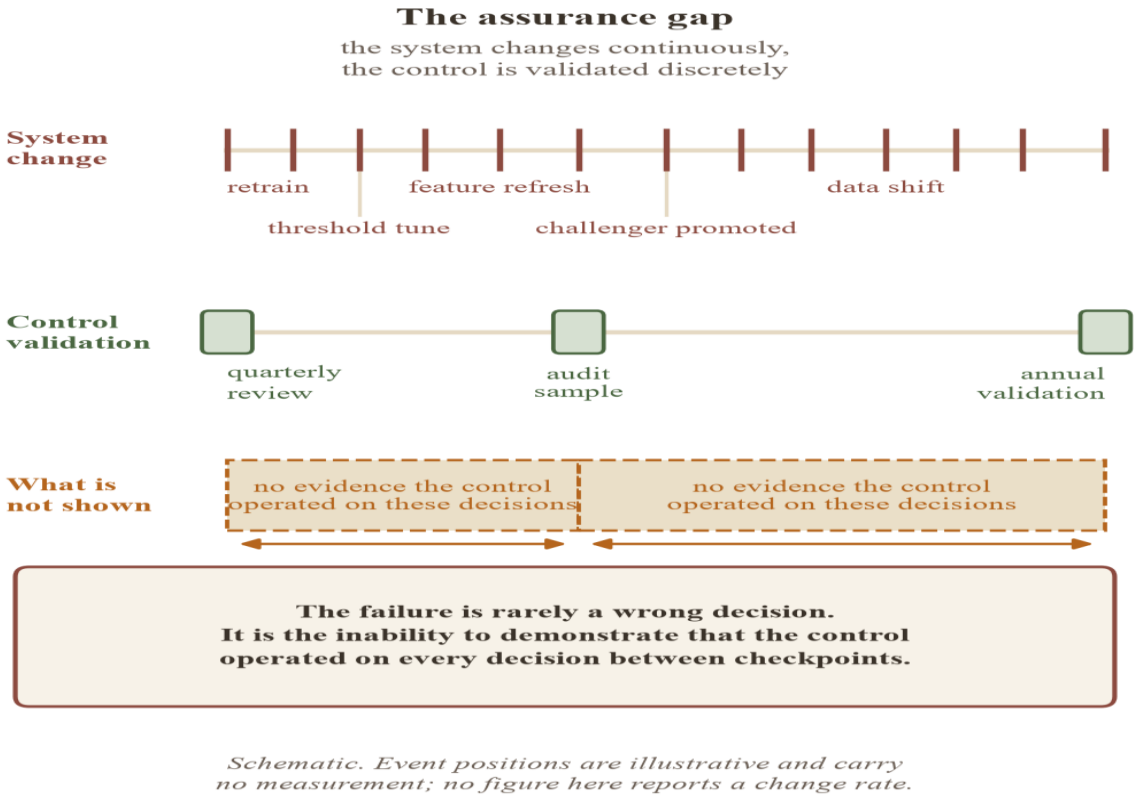

Figure 1. The assurance gap. The system changes continuously while the control is validated discretely, leaving intervals in which no evidence exists that the control operated. Schematic; event positions are illustrative and report no measurement.

The consequence is not hypothetical. Supervisory expectations are written in the language of continuity. The 2011 Supervisory Guidance on Model Risk Management states that ongoing monitoring "is essential to evaluate whether changes in products, exposures, activities, clients, or market conditions necessitate adjustment, redevelopment, or replacement of the model" [R1]. Audit standards for internal control over financial reporting require evidence that relevant controls "operated effectively during the entire period" upon which reliance is placed [R8]. The digital operational resilience regime for the European financial sector requires firms to "continuously monitor and control the security and functioning of ICT systems and tools" and to have "mechanisms to promptly detect anomalous activities" [R7]. The European artificial intelligence regulation frames risk management for high-risk systems as "a continuous, iterative process" [R6]. In every case the regulator asks about a period. In every case the institution answers with a point.

This paper argues that the answer is to make the control itself a piece of running software that reads the system's own telemetry, and to make its verdicts into signed evidence. It contributes the following.

1. A three-layer taxonomy separating obligations, controls and signals, with ownership assigned layer by layer, so that regulatory text and institutional implementation can evolve independently while the link between them stays explicit and recoverable.

2. A four-move translation procedure taking an obligation to an intent, a control, a signal and a pass or fail predicate, together with a machine-readable control specification schema in which the specification is the executable control rather than documentation describing one.

3. A many-to-many obligation mapping with a formal coverage metric, Control Observability Coverage, that reduces coverage assessment from expert judgement to the inspection of a matrix and makes a blind spot visible as an empty row.

4. A five-stage architecture from obligation translation through instrumentation, tamper-evident capture, near real-time evaluation and closed-loop response, including the instrumentation of agentic workflows at the granularity individual controls require.

5. A five-level maturity model for continuous compliance, L0 to L4, with an explicit statement of what changes at each boundary.

6. An empirical evaluation of a proof of concept that instantiates part of the reference model over five controls, reporting coverage, per-control detection latency, recovery, evidence completeness, ledger integrity and false-positive rate on public and synthetically generated data.

2. SCOPE AND EVIDENTIARY BASIS

The two contributions in this paper have different evidentiary status, and conflating them would misrepresent the work.

The reference model is a design contribution. It comprises the taxonomy, the translation procedure, the control specification schema, the mapping matrix and coverage metric, the five-stage architecture and the maturity model. It is argued for on the grounds of regulatory fit, mechanism and internal consistency. It is not the case that every element of the reference model corresponds to a deployed system.

The proof of concept is an implementation of part of the reference model, and it is evaluated empirically. It covers five controls, one live decisioning service, one evidence ledger and one enforcement path. Where the reference model describes capabilities beyond that scope, those capabilities are presented as design, not as measured behaviour, and the text says so at each point.

Neither contribution is a bank deployment. The proof of concept is not running in a production banking estate, it processes no customer data, and its results should be read as evidence that the mechanism works as designed at the scale tested, not as evidence of institutional outcomes.

The evaluation data basis is deliberate, and it is a strength rather than a caveat. The decision model is a logistic regression trained on the public UCI Statlog German Credit dataset, 1,000 records, with a protected attribute derived from the personal status and sex field for fairness monitoring [R12]. Control violations were introduced by controlled synthetic injection rather than found in the wild, which is what makes per-control detection latency measurable at all: the exact decision at which a violation begins is known, so the number of decisions to first breach is an observation rather than an estimate. No customer data and no proprietary banking data were used at any point. The choice was made for confidentiality and for reproducibility, and both are served: another researcher can obtain the dataset, reproduce the injection procedure and re-derive every number reported here.

Two dimensions were not evaluated, and the paper claims nothing about either. Transaction-monitoring performance was not evaluated. The controls implemented and measured here govern a credit decisioning workflow; no transaction-monitoring or anti-money-laundering control was implemented, instrumented or measured, and no figure of any kind is reported for it. It is stated as future work in Section 16. Auditor acceptance was not measured. Whether an assurance professional will accept continuously generated, cryptographically chained evidence in place of a sampled manual attestation is an open empirical question, treated as research agenda in Section 16 and as a

limitation in Section 15. No claim is made that any auditor reviewed, validated or accepted the evidence produced here.

3. THE ASSURANCE GAP AND ITS REGULATORY COST

Three properties of machine learning systems in production combine to produce the gap.

Change is continuous and partly unattended. Retraining pipelines, automated feature refresh and scheduled recalibration change model behaviour without a discrete human decision. Concept drift changes the relationship the model was fitted to without changing the model at all [14], [15]. The operational literature has documented for a decade that production readiness is a property of a system's tests and monitors rather than of model quality [13], and mature delivery practice treats continuous training and continuous monitoring as first-class stages [12].

Control validation is discrete and sampled. The control review, the validation cycle and the audit sample are all point observations, each defensible about the moment it observes and silent about the population of decisions between observations. This is not a criticism of audit method. Sampling infers a population property from a sample drawn from a stable population, and a continuously changing decision system is not that.

The obligations are written about periods. The 2011 model risk guidance places ongoing monitoring as one of the two core elements of validation, expects a programme of ongoing testing with procedures for responding to problems that appear, and expects that model code "cannot be altered except by approved parties, and that all changes are logged and can be audited" [R1]. Audit standards for internal control note explicitly that testing controls over a greater period of time provides more evidence than testing over a shorter one [R8]. The resilience regulation requires detection mechanisms that define alert thresholds and trigger response processes including automatic alerting [R7]. The artificial intelligence regulation requires high-risk systems to be designed with logging and record-keeping capabilities and to enable effective human oversight [R6].

Put together: the regulator asks whether the control operated throughout, and the institution can answer only whether the control existed and appeared to work when looked at. The cost is paid in examination findings, because continuity cannot be evidenced; in remediation, because a breach discovered at the next checkpoint has already touched every decision since the last one; and in managerial control, because the interval between checkpoints is precisely the interval in which nobody knows.

A note on the status of one instrument is necessary, because it bears on how the expectation should be cited. The 2011 Supervisory Guidance on Model Risk Management was issued as Federal Reserve SR letter 11-7 and OCC Bulletin 2011-12 on 4 April 2011 [R1]. On 17 April 2026 the Federal Reserve issued SR 26-2, which states that it "supersedes and replaces SR letter 11-7, Guidance on Model Risk Management (issued April 4, 2011)" and SR letter 21-8 [R2], and the OCC issued Bulletin 2026-13, which rescinds OCC Bulletin 2011-12 among other issuances [R3]. This paper cites the 2011 instrument, by its date, for the ongoing-monitoring expectation quoted above, because that is the document in which the quoted text appears. It makes no claim about the content of the 2026 successor beyond the fact and date of supersession, which was verified at source. The methodological point generalises: supervisory landing pages are revised in place, so a citation to a live URL can silently come to point at different text. Regulatory citation in this field should be to a dated instrument.

4. RELATED WORK

Continuous auditing and continuous assurance. The accounting information systems literature has argued for continuous auditing for over two decades. Chan and Vasarhelyi set out the shift from periodic to continuous audit practice and the reorganisation of audit work it implies [7]. Alles, Kogan and Vasarhelyi report on pilot implementations and are candid about what impeded them, notably the dependence on management's own systems for the data the auditor must rely on [8]. That work establishes the objective this paper pursues. What it largely predates is a standard, vendor-neutral telemetry substrate on which controls can be evaluated, and a cheap cryptographic

method for making the resulting evidence resistant to later alteration. Those two ingredients are why the objective is now tractable.

Policy-as-code and infrastructure-as-code governance. Treating configuration and policy as versioned, testable code is established practice and has been systematically surveyed [11]. The governance benefit is not automation speed; it is that the artefact under version control is the thing that actually runs, so the question "which rule was in force on this date" has a mechanical answer. This paper applies that property where the answer carries legal weight.

Observability and distributed tracing. Distributed tracing reconstructs the causal path of a request across services, and large-scale production systems have shown that per-request traces can be collected at acceptable cost and analysed in aggregate [5]. The OpenTelemetry Specification standardises the API, SDK, data model and wire protocol for traces, metrics and logs [R11]. The significance for compliance is that a span is a structured, timestamped, attributed record of a step actually taken, emitted by the system that took it, which is precisely the evidentiary form a control needs and precisely the form a manual attestation lacks.

Tamper-evident logging. Cryptographic timestamping and hash chaining were developed to make a document's contents at a point in time verifiable without trusting the custodian [1], using hash-tree constructions that let a large set be committed to compactly [2]. Schneier and Kelsey applied the idea to audit logs, so that an attacker who compromises a machine cannot undetectably alter records written before the compromise [3]. Certificate transparency demonstrates the pattern at internet scale, where an append-only, publicly auditable log makes misissuance detectable rather than preventable [4]. This paper uses these mechanisms unchanged; the contribution is in what is chained, namely control verdicts linked to the decisions that produced them.

Algorithmic fairness and disparate impact. Feldman and colleagues formalise disparate impact as the ratio of favourable outcome rates across a protected attribute, and connect it to the four-fifths screen used in United States enforcement practice [16]. That formalisation makes fairness expressible as a predicate over a population rather than an opinion about a model, and carries the property that matters most for control design here: it is a population measure, and cannot be evaluated on a single decision.

Explainability and its limits. Post-hoc explanation methods generate local surrogate accounts of individual predictions [19]. The critical literature is clear that such accounts are not the model's reasoning and that interpretability is not a single well-defined property [18], and argues that for high-stakes decisions an inherently interpretable model is preferable to an explained opaque one [17]. This bounds what a compliance control can honestly assert. A control can verify that a rationale was produced, in time, referencing the features the model used. It cannot verify that the rationale is a good one, and Section 15 treats that as a limitation rather than resolving it.

Drift detection. The concept drift literature provides the detector families, the change-point machinery and the taxonomy of drift types that a drift control instantiates [14], [15]. Its relevance here is that every practical detector operates over a window, which fixes a floor on how quickly a drift control can signal.

Data lineage and provenance. Provenance research supplies the models for recording where a data item came from and which transformations produced it [6]. Recording lineage is the mechanical part of discharging risk data aggregation expectations; the harder part, addressed in Section 7, is knowing which obligations a lineage record actually serves.

Model risk management, RegTech and algorithmic auditing. The supervisory model risk framework supplies the ongoing-monitoring expectation this paper operationalises [R1]. The regulatory technology literature examines what happens when requirements are expressed as computer code, including the risk that encoding a rule narrows it [10]. Raji and colleagues propose an end-to-end internal algorithmic auditing framework and are explicit that audit artefacts should be produced through the development lifecycle rather than assembled afterwards [9]. This paper is close in spirit and differs in mechanism: the artefacts here are emitted by the running system on every decision, and their integrity is cryptographic rather than procedural.

MLOps, production readiness and service-level objectives. Machine learning operations practice contributes the continuous-training and continuous-monitoring stages and a reference architecture for them [12]. The production readiness literature contributes the discipline of scoring a system on the tests and monitors it actually has [13]. Reliability engineering contributes the service-level objective, a numeric target on a measurable property of a service, which is the form the explainability control takes in Section 6.

Agentic AI governance. Where a system takes a sequence of actions rather than returning a single prediction, the governance surface expands with the number of steps and tool calls. The literature on increasingly agentic algorithmic systems identifies the resulting accountability difficulties, and argues that visibility into what the system actually did is a precondition for governing it [20]. Section 9 addresses that requirement at the granularity individual controls need.

5. FROM OBLIGATION TO EXECUTABLE CONTROL

The reference model separates three layers, shown in Figure 2, and assigns ownership to each.

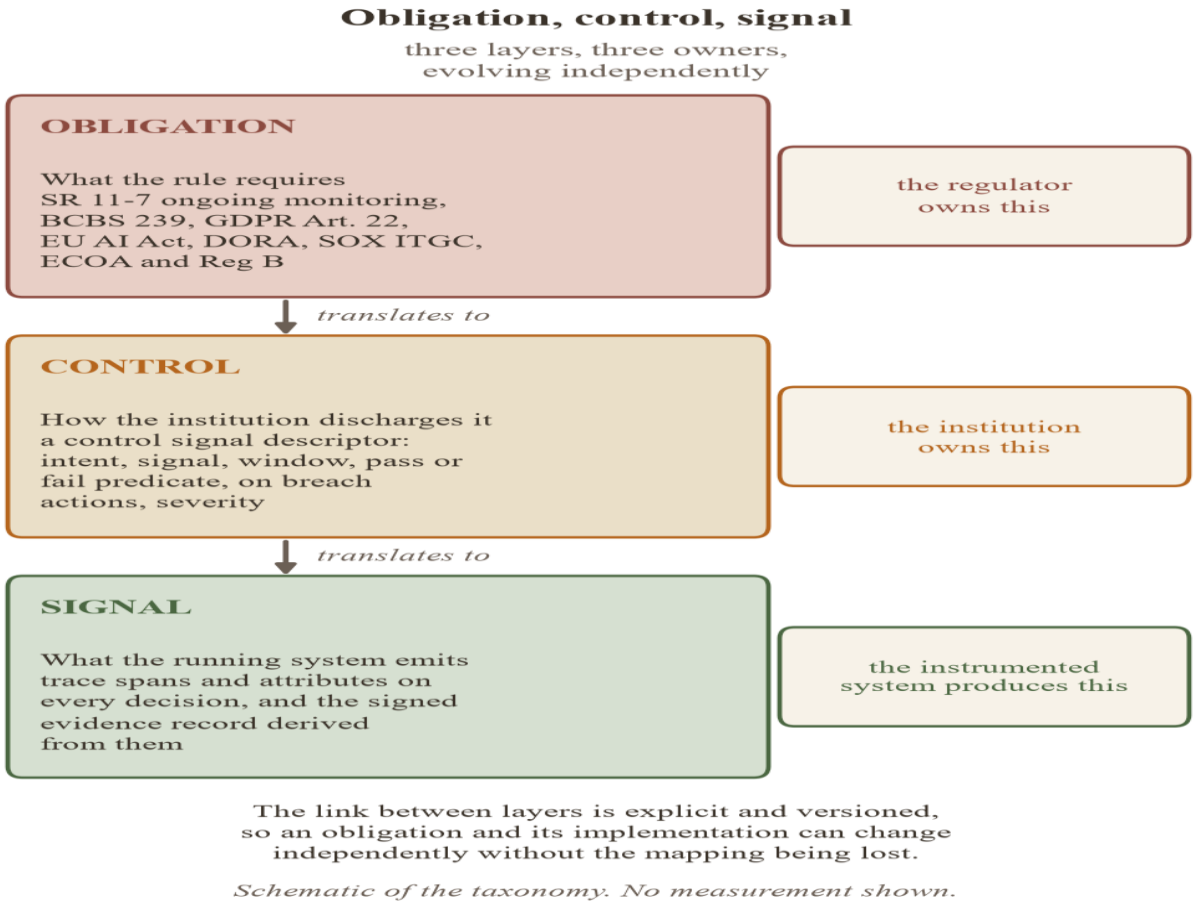

Figure 2. The three-layer obligation, control and signal taxonomy, with ownership assigned layer by layer. The link between layers is explicit and versioned, so obligations and implementation evolve independently without the mapping being lost. Schematic.

Obligations are what the rule requires. They are written by regulators and legislatures in natural language, they are stable over years rather than sprints, and the institution does not control their content. The regulator owns this layer.

Controls are how the institution discharges an obligation. A control is a choice: a specific measurable property, a specific threshold, a specific window, a specific response on breach. Two institutions facing the same obligation may implement different controls and both be defensible. The institution owns this layer.

Signals are what the running system emits. A signal is an observable property of an actual execution: an attribute on a trace span, a latency, a count, a ratio computed over a window of decisions. The instrumented system produces this layer, and it is not a matter of opinion at all.

The separation matters because the three layers change on different clocks and for different reasons. Regulation changes when a legislature acts. Controls change when the institution's risk appetite or architecture changes. Signals change when the system is re-engineered. Collapsing the layers, which is what a control narrative in a document does, means that any change to any layer requires rewriting the whole narrative and destroys the ability to answer which rule was in force when. Keeping them separate and explicitly linked means each can change while the mapping between them remains a first-class, versioned artefact.

5.1 The four-move translation

Translating an obligation into something a machine can evaluate proceeds in four moves. Obligation to intent, intent to control, control to signal, signal to a pass or fail predicate. The moves are worth naming separately because each is a distinct act with a distinct failure mode, and because the second and third are where institutional judgement enters and must be recorded.

Consider the automated decision provisions of the European data protection regulation. Article 22(1) establishes that a data subject has "the right not to be subject to a decision based solely on automated processing", and the transparency provisions require that the subject be given "meaningful information about the logic involved" [R5]. That is the obligation, and as written it is not machine-checkable. Nothing in a running system corresponds to "meaningful".

The **intent** is recoverable, however, and is narrower than the text: where an automated decision affects a person adversely, the reasoning behind that decision must be available to them. The **control** is the institution's chosen discharge of that intent: every denial must carry a rationale, and the rationale must be produced within a time budget short enough that it can be surfaced in the customer journey rather than reconstructed later. The **signal** is what the instrumented system emits: a boolean attribute recording whether a rationale artefact was attached to the decision span, and a duration recording how long its generation took. The **predicate** is the pass or fail rule: for every decision with an adverse outcome, the rationale attribute is present and the generation duration is within budget.

Two things about this translation should be stated plainly rather than glossed. First, it is lossy in a specific direction. The predicate checks that a rationale exists and arrived in time; it does not check that the rationale is meaningful, and given what is known about the limits of post-hoc explanation [17], [18], [19] no predicate could. The control therefore sits alongside human judgement about explanation quality rather than replacing it. Second, the narrowing is an institutional choice and it is now written down. A supervisor who disagrees can see exactly what was chosen, when and by whom, which is better than being handed a control narrative that asserts compliance without exposing the interpretation it rests on. The regulatory technology literature warns that encoding a rule can quietly narrow it [10]; the answer offered here is not to avoid encoding but to make the narrowing legible.

6. THE CONTROL SPECIFICATION SCHEMA

A control produced by the four-move translation is written as a control signal descriptor. Table 2 reproduces the descriptor for the fairness control implemented in the proof of concept.

Table 2. The control signal descriptor for C-FAIR-01, reproduced as the schema example. The descriptor is the executable control, held in version control, and is read directly by the evaluation engine.

Field	Value

Control_ID	C-FAIR-01
Serves	ECOA / Reg B, EU AI Act Art. 10
Intent	No disparate impact across sex
Signal	disparate_impact_ratio
Window	Sliding, last 200 decisions
Predicate	0.80 <= ratio <= 1.25 (four-fifths rule)
On_breach	emit_evidence, trip_circuit_breaker, route_human_review
Severity	high

The important claim about this artefact is easy to miss. The descriptor is not documentation about a control. It is the executable control. The evaluation engine reads the descriptor and evaluates it; there is no separate implementation that the descriptor describes and might drift from. The descriptor is held in version control alongside application code, which yields a property that manual attestation cannot provide: for any date, the exact rule that was in force on that date is recoverable from the repository, together with who changed it, when, and against what review. When an examiner asks what the fairness threshold was in the second quarter, the answer is a commit, not a recollection. This is the governance benefit of policy-as-code applied where it carries legal weight [11].

The fields do specific work. Serves carries the many-to-many link to obligations and is what makes the coverage matrix of Section 7 constructible. Intent records the human-readable purpose, so a later reader can judge whether the predicate still serves it. Signal names the measured quantity, which must correspond to something the instrumented system actually emits; a descriptor naming a signal nothing emits is detectable as a configuration error rather than discovered as an audit finding. Window distinguishes the two classes of evaluation analysed in Section 11 and is the field most responsible for detection latency. Predicate is the pass or fail rule, and the only field whose evaluation is mechanical. On_breach makes the response part of the control rather than a downstream operational choice, which is what closes the loop in Section 12. Severity drives routing and escalation.

The predicate in Table 2 requires the disparate impact ratio to fall between 0.80 and 1.25 inclusive. The provenance of those numbers deserves precision, because it is often asserted loosely. The 0.80 figure is the four-fifths screen from the Uniform Guidelines on Employee Selection Procedures, which state that a selection rate for a group "which is less than four-fifths (4/5) (or eighty percent) of the rate for the group with the highest rate will generally be regarded by the Federal enforcement agencies as evidence of adverse impact" [R10]. That instrument governs employee selection, not credit. The Equal Credit Opportunity Act and Regulation B prohibit discrimination against an applicant on a prohibited basis regarding any aspect of a credit transaction, and do not specify a numeric ratio [R9]. So the threshold in the descriptor is an institutional screen, chosen by analogy to a well-established enforcement heuristic and formalised in the fairness literature [16], rather than a numeric limit quoted from credit law. The upper bound of 1.25 is the reciprocal of 0.80, which makes the band symmetric under relabelling of which group is the reference. Recording the threshold's provenance in the descriptor's intent field, rather than presenting 0.80 as though it were statutory, is exactly the kind of visibility the schema exists to provide.

7. Many-to-Many Mapping and Coverage-Gap Detection

The naive approach to control coverage is one control per regulatory clause. It fails in both directions. It multiplies controls without adding assurance, because many clauses ask for the same underlying property. And it hides duplication, because nobody can see that four controls are all measuring one thing.

The reference model instead treats the relationship between obligations and controls as many-to-many, and holds the mapping as an artefact in its own right. Table 1 shows the structure for the five controls in the proof of concept.

Table 1. Obligation to control mapping matrix for the five controls in the proof of concept. A mark indicates that the control's Serves field declares that obligation. Reading down a column shows one control discharging several obligations; reading across a row shows one obligation needing several controls. An empty row is a compliance blind spot. The evidence ledger is cross-cutting and contributes to every row that requires a demonstration of operation across a period.

Obligation (dated instrument)	FAIR	EXPL	DRIFT	DATA	HUM
SR 11-7 (2011) ongoing monitoring [R1]			X	X	
BCBS 239 (2013) accuracy and integrity [R4]				X	
GDPR Art. 22 automated decisions [R5]		X			
EU AI Act Art. 10 data governance [R6]	X			X	
EU AI Act Art. 12 record keeping [R6]				X	
EU AI Act Art. 14 human oversight [R6]					X
DORA Art. 9 and 10 detection [R7]			X		
DORA Art. 11 response and recovery [R7]					
SOX ITGC operation across period [R8]				X	
ECOA and Reg B prohibited basis [R9]	X				

Read down a column and the many-to-one direction appears. The data lineage control, C-DATA-01, discharges obligations under three distinct regimes at once. Risk data aggregation expectations require that a bank generate accurate and reliable risk data, that data be aggregated on a largely automated basis to minimise the probability of errors, and that reports be reconciled to risk data [R4]. Internal control over financial reporting requires evidence that controls over the relevant data operated across the period [R8]. The model risk guidance requires process verification that internal and external data inputs continue to be accurate, complete and consistent with model purpose and design [R1]. One instrumented property, the presence and integrity of a lineage record on every decision, contributes to all three. Similarly, the fairness control serves both credit anti-discrimination obligations and the data governance provisions of the European artificial intelligence regulation.

Read across a row and the one-to-many direction appears. A single obligation may need several controls before it is discharged; ongoing monitoring under the model risk guidance is served by the drift control, the lineage control and the evidence ledger together, and by none of them alone.

The mapping is itself an audit artefact. It is the answer to the question an examiner actually asks, which is not "do you have controls" but "which of your controls addresses this requirement, and how do you know it is running". Because the mapping is derived mechanically from the Serves field of every deployed descriptor, it cannot drift from the deployed control set the way a maintained spreadsheet can.

7.1 Coverage-gap detection

The structural payoff is that a coverage gap becomes visible rather than inferred. An obligation with an empty row in the matrix is a compliance blind spot. Nothing the institution has instrumented bears on it. Table 1 includes such a row deliberately: the response and recovery provisions of the European resilience regulation, which require a business continuity policy and restoration capability [R7], are not served by any of the five controls in the proof of concept. The matrix says so without anyone forming a judgement.

This changes the character of coverage assessment. Under a narrative regime, deciding whether coverage is adequate requires an expert to read control documentation against regulatory text and form an opinion, which is expensive, unrepeatable and difficult to challenge. Under a matrix regime, it requires reading the matrix for empty

rows, which is mechanical, repeatable and challengeable by anyone who can read a table. Expert judgement is not eliminated; it moves to where it belongs, which is deciding whether a proposed control genuinely discharges the obligation it claims to serve.

The metric that formalises this is Control Observability Coverage, or COC. For a set of controls, COC is the proportion whose declared signal is in fact observable in the instrumented system and whose predicate is in fact evaluable on live telemetry. A control that names a signal nothing emits, or whose window can never be filled, counts against COC. The proof of concept reports COC of 1.0 across five controls, which states that every deployed control was genuinely evaluable against the running system rather than merely declared.

Regulatory change is handled by the same structure. A new instrument is decomposed into its obligations, each obligation becomes a row, and the existing Serves declarations are consulted to see which rows are already covered. What remains is the set of empty rows, so in place of a re-mapping exercise across the whole control estate the institution reads off a gap list. When the model risk guidance was superseded in April 2026 [R2], [R3], the mechanical part of assessing the impact under this structure would be exactly that.

7. ARCHITECTURE

The reference architecture has five stages, shown as a flowchart in Figure 3.

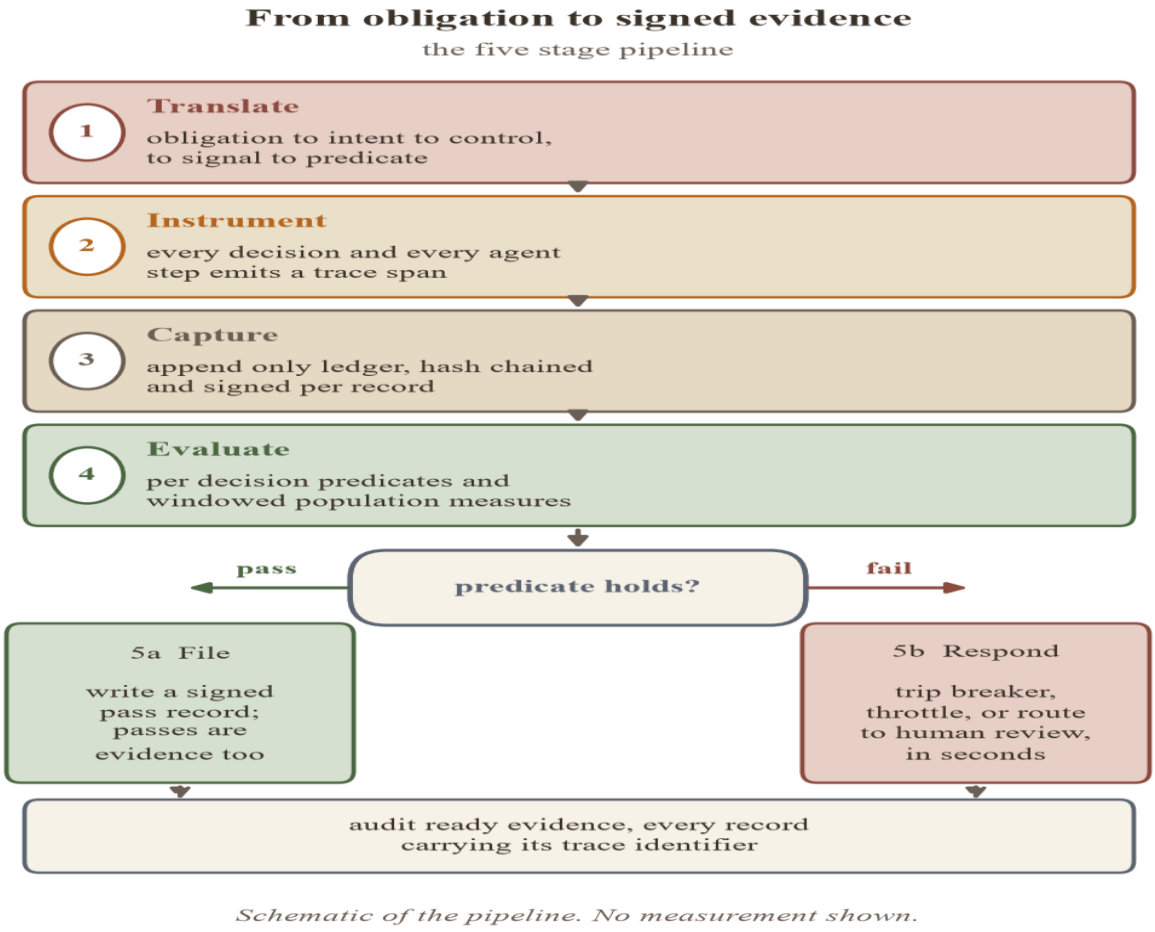

Figure 3. The five-stage pipeline from obligation to signed evidence. Note that both branches write to the ledger: a pass record is evidence that the control operated, which is what a failure log cannot show. Schematic.

Stage 1, translate. Obligations are decomposed and passed through the four-move translation of Section 5 into control signal descriptors held in version control. This stage is deliberately human. It is where interpretation happens, and its output is reviewable.

Stage 2, instrument. The decision workflow is instrumented so that every decision, and every step within a decision, emits telemetry carrying the attributes the descriptors reference. This is treated in Section 9.

Stage 3, capture. Telemetry relevant to control evaluation is captured as tamper-evident evidence in an append-only ledger. This is treated in Section 10.

Stage 4, evaluate. Controls are evaluated in near real time, per decision for predicate controls and per window for population controls. This is treated in Section 11.

Stage 5, respond and file. On breach, the declared On_breach actions execute. On pass, a pass record is written. Either way the verdict becomes filed, audit-ready evidence carrying the trace identifier of the decision that produced it. This is treated in Sections 10 and 12.

Two architectural choices deserve emphasis because they are what distinguish this from a monitoring dashboard.

First, the control specification is the input to the evaluation engine, not a description of it. Adding a control means adding a descriptor, not writing evaluation code. This is what keeps the control estate inspectable at the scale a bank actually operates at, and what makes the version history of the descriptor set the authoritative record of what was enforced when.

Second, evidence is written for passes as well as failures. This is discussed in Section 10 and is the single design decision most responsible for the architecture answering the regulator's actual question.

8. Instrumenting AI and Agentic Workflows

Instrumentation is the stage on which everything downstream depends, and the requirement is stricter than conventional application monitoring.

For a single-shot model inference the requirement is modest. The decision service emits a trace span for each decision, and that span carries compliance-relevant attributes: the model identifier and version, the score, the applied threshold, the outcome, whether a rationale artefact was produced and how long it took, the lineage identifier of the feature set used, the derived protected attribute where fairness monitoring requires it, and whether the decision was routed for human review. The standard specifies the data model, the attribute semantics and the wire protocol, so this is a matter of configuration and convention rather than of building bespoke plumbing [R11]. Aggregate analysis of per-request traces at production scale is established practice [5].

For an agentic workflow the requirement changes in kind. An agent does not return a single prediction. It takes a sequence of actions, calls tools, and chains steps whose composition is decided at runtime. Each of those steps becomes its own span. The consequence is that the telemetry records the path the system actually took and not merely the outcome it arrived at, and that is the level at which individual controls need to operate.

The distinction is load-bearing rather than decorative. Consider the human oversight control. An obligation to enable effective human oversight of a high-risk system [R6] is discharged by a control asserting that decisions meeting defined criteria were routed to a human. If the only telemetry is the final outcome, the control can observe that a decision was made and, at best, that a human was recorded as involved. If every step is a span, the control can observe whether the routing step was actually taken, in what order, before or after the action it was supposed to gate, and whether it was skipped on a retry path. The first is an assertion about the outcome. The second is an observation of the mechanism. Only the second can distinguish a control that operated from a control that was bypassed and happened to produce an acceptable result.

The same argument applies to the lineage control, where a multi-step agent may fetch data from several sources with different provenance and the composite lineage of a decision is reconstructable only if each fetch emitted its own span, and to explainability, where a rationale produced by one step and discarded by another never reached the customer.

This is the concrete form of the visibility requirement that the agentic governance literature identifies as a precondition for accountability [20]. The claim here is narrow and worth keeping narrow: per-step instrumentation makes the path observable, and observability of the path is what lets a control evaluate the mechanism rather than the outcome. It does not resolve who is accountable for the path.

10. Tamper-Evident Evidence Capture

Evidence that can be edited after the fact is not evidence in the sense a supervisor needs. The capture stage therefore uses an append-only ledger in which each entry is cryptographically chained to its predecessor and independently signed. Figure 4 shows why an edit is detectable.

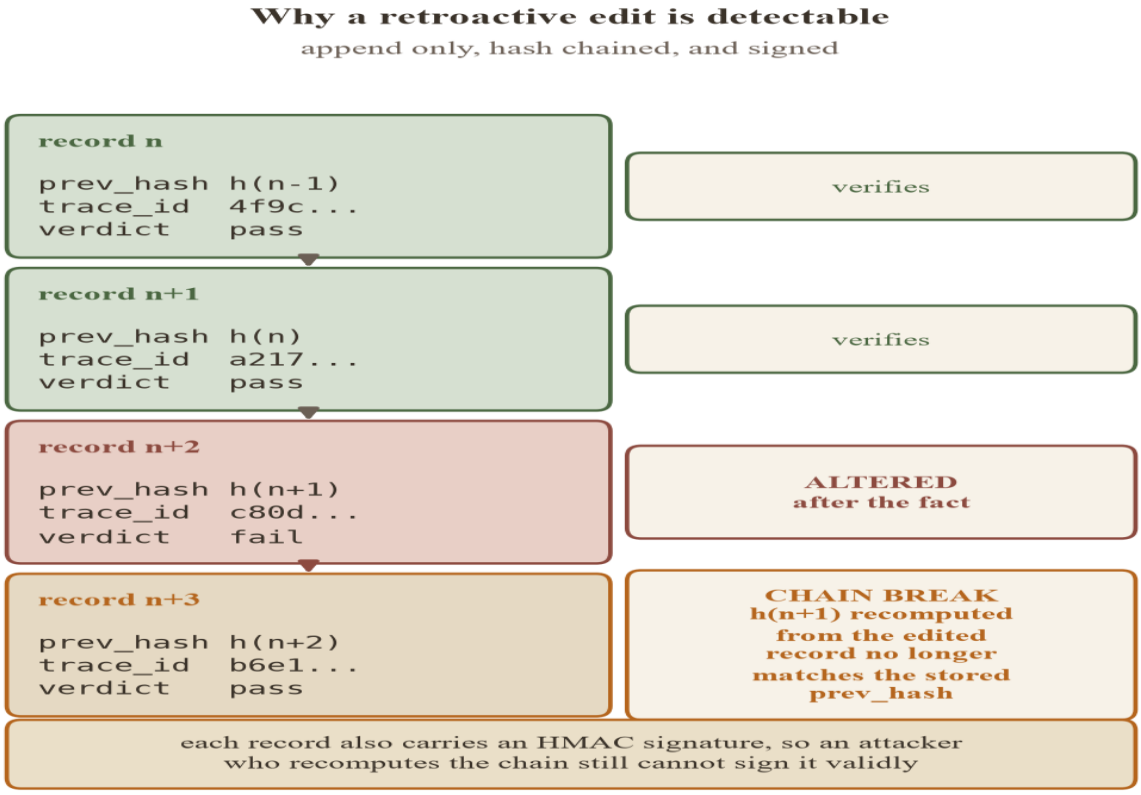

Figure 4. The hash-chained, signed, append-only ledger, and why a retroactive edit is detectable. Recomputing the chain from the altered record forward no longer matches the stored predecessor hash, and full-chain verification fails at a specific position. Schematic; hash values are placeholders. The chain verification and tampering-detection results are reported in Section 13.

Each record contains the hash of the previous record, so the records form a chain. Recomputing the chain from any point forward yields a hash sequence; if any record has been altered, the recomputed hash of the altered record no longer matches the value stored in its successor, and full-chain verification fails at a specific position. This is the hash-chaining construction from the digital timestamping literature [1], [2], applied to audit logs in the manner Schneier and Kelsey described [3], and demonstrated at internet scale by certificate transparency [4]. Each record additionally

carries a signature, in the proof of concept a keyed message authentication code, so an adversary who recomputes the entire chain to make it internally consistent still cannot produce valid signatures without the key.

The design goal is worth stating exactly, because it is often overstated. The ledger does not prevent tampering. It makes tampering detectable, and it localises it. That is the correct goal for a compliance evidence store: prevention would require removing the institution's own access to its own records, whereas detection is achievable, verifiable and sufficient to change the incentive.

Two properties of the record format do specific work.

Every record carries the trace identifier of the decision that produced it. A control finding is therefore not an isolated alert; it is a pointer back into the telemetry of the exact decision that triggered it, from which the model version, the feature set, the score, the threshold and the full step sequence are recoverable. An examiner presented with a finding can follow it to the decision. An engineer presented with the same finding can reproduce it. This bidirectional link between evidence and execution is what turns a monitoring alert into an audit artefact, and it is only available because the evidence is derived from tracing rather than from a separate compliance log.

Pass records are written, not only failures. This is deliberate and it is the crux of the argument. A log of failures demonstrates that the system detects failures. It does not demonstrate that the control was operating during the periods when nothing failed, and that is precisely the period the regulator asks about. A continuous stream of pass records, each signed, each chained and each pointing at a specific decision, is a positive demonstration that the control evaluated every decision throughout the period. It is what an attestation structurally cannot provide, because an attestation is one assertion about a period rather than a record per event within it. It is also why the evidence count in Section 13 exceeds the decision count: a decision evaluated by several controls generates several records, and completeness is measured over the records that should exist rather than over the decisions.

The cost of this choice is volume, and it should be acknowledged rather than waved away. Writing a signed record per control per decision produces evidence at the rate of business activity. The proof of concept operated at a scale where this was untroublesome; a production estate would need a retention policy and periodic checkpointing of the chain, so that verification does not require replaying the entire history. Those are engineering problems with known solutions, but they are real, and no measurement here speaks to ledger volume behaviour at production rates.

9. Near Real-Time Control Evaluation

The evaluation stage reads captured telemetry and evaluates every deployed descriptor against it. Controls fall into two classes, and the distinction is the sharpest analytical point in the evaluation.

Per-decision predicates are evaluable on a single decision. The explainability control asks whether this decision carries a rationale within budget. The lineage control asks whether this decision carries a lineage record. The human oversight control asks whether this decision, having met the routing criteria, was routed. Each reads one span, or one span tree, and returns a verdict. The window field of the descriptor is degenerate: it is the decision itself.

Population-level windowed measures are not evaluable on a single decision, even in principle. A disparate impact ratio is a property of a distribution of outcomes across a protected attribute; a single decision has no ratio. A drift statistic is a comparison between a reference distribution and a recent one; a single observation has no distribution. These controls declare a window, in the fairness descriptor of Table 2 a sliding window of the last 200 decisions, and cannot return a verdict until the window holds enough evidence to make the statistic meaningful.

This distinction explains the measured detection latencies directly, and it is why the paper reports them per control rather than as a single figure. Figure 5 plots the measured values.

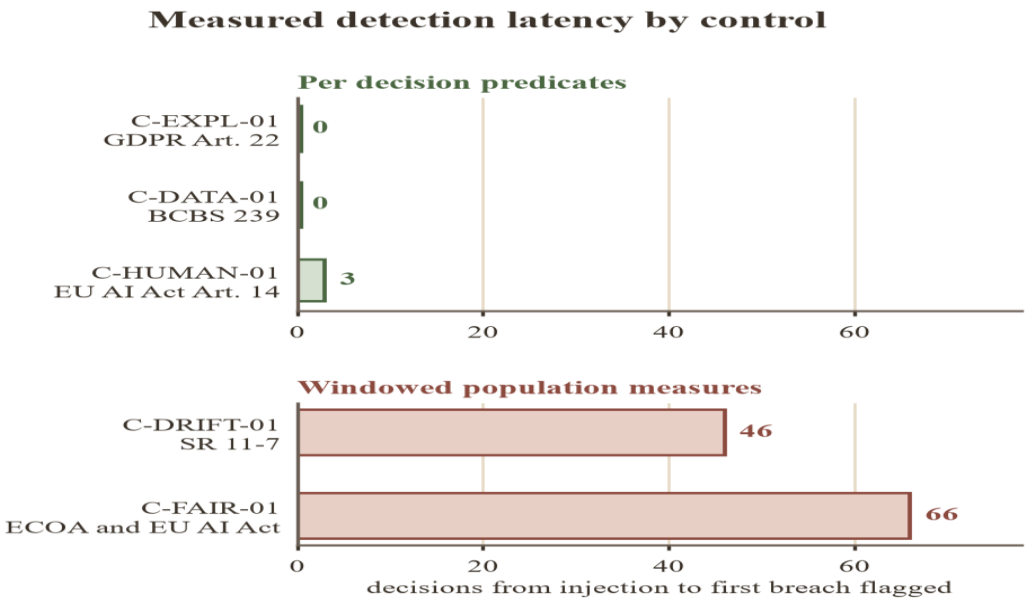

Figure 5. Measured per-control detection latency, in decisions from violation injection to first breach flagged, separating per-decision predicates from windowed population measures. These are the measured values from the proof of concept. The separation is the explanation: a predicate reading a single decision detects on that decision, while a measure over a sliding window cannot signal until the window carries enough evidence.

The two per-decision controls whose violations were injected as attribute-level defects detected on the very decision the injection began, at a latency of 0 decisions, because the predicate reads that decision and the defect is in it. The human oversight control detected at 3 decisions, consistent with a routing criterion that only bites on decisions meeting specific conditions, so the first qualifying decision after injection is not necessarily the first decision after injection. The two windowed controls detected at 46 decisions for drift and 66 decisions for fairness, because the injected shift had to accumulate within the window until the statistic crossed the threshold.

The practical reading is that detection latency for a windowed control is a design parameter, not a defect. It is set by the window length, the threshold, and the effect size of the violation. Shortening the window shortens latency and raises variance, which raises the false-positive rate; lengthening it does the reverse. The drift literature has characterised this tradeoff thoroughly [14], [15], and the fairness formalisation makes clear that the ratio is a population estimate with sampling error [16]. A framework that reported a single aggregate detection latency across both classes of control would be concealing this, and would invite the reader to conclude either that the per-decision controls are slow or that the windowed controls are fast. Neither is true, and the difference between 0 and 66 decisions is structural.

Near real-time is therefore the honest term rather than real-time. Per-decision predicates are effectively synchronous with the decision. Windowed measures are as fast as their statistics allow, and no faster.

10. Closed-Loop Enforcement

A control that detects and only records has improved the audit trail. It has not reduced the exposure. The On_breach field of the descriptor is what closes the loop, and it is part of the control specification rather than a separate operational runbook, so the response is versioned and reviewable alongside the rule that triggers it.

Three response classes are supported in the reference model, and the fairness descriptor in Table 2 declares all three.

Emit evidence. A breach record is written to the ledger, chained and signed, carrying the trace identifier and the measured value that violated the predicate. This happens in every case and is the minimum response.

Trip the circuit breaker. Automated decisioning on the affected path is halted. The proof of concept implements this, and the breaker tripped automatically on the first fairness breach without human involvement. The design intent is that a breach on a control the institution has classified as high severity should not continue to accumulate affected decisions while an alert waits in a queue.

Throttle or route to human review. Where halting is disproportionate, the response degrades the automation rather than stopping it: reduce the proportion of decisions taken automatically, or route the affected population to human adjudication. This preserves service while removing the automated exposure, and it produces its own evidence, since routed decisions carry oversight attributes the human oversight control can evaluate.

The temporal contrast is the point. A control finding in a quarterly report is acted on in weeks. A finding that trips a breaker is acted on in seconds. Between those two timescales lies the entire population of decisions taken between detection and response, and closing the loop is what makes that population empty. The obligation surface supports this reading: the resilience regulation expects detection mechanisms that trigger and initiate response processes, including automatic alerting [R7], and the model risk guidance expects ongoing testing together with procedures for responding to problems that appear [R1]. Neither instrument is satisfied by detection alone.

Closed-loop enforcement also introduces a governance problem that the reference model does not solve and should not pretend to. A system that halts its own decisioning is a system taking a consequential action on the basis of its own finding. That is a new object of governance, and Sections 15 and 16 treat it as unresolved.

13. Proof-of-Concept Evaluation

The proof of concept instantiates Stages 2 through 5 of the architecture over five controls. Section 2 states the scope and the data basis; this section reports what was measured.

Setup. A logistic regression credit decisioning model was trained on the public UCI Statlog German Credit dataset of 1,000 records, achieving a test area under the receiver operating characteristic curve of 0.77 [R12]. A protected attribute was derived from the dataset's personal status and sex field to support fairness monitoring. The decision service was instrumented so that every decision emits a trace span carrying compliance attributes [R11]. Five control signal descriptors were deployed: fairness, an explainability service-level objective, feature drift, data lineage presence, and human oversight routing. Control verdicts were written to a hash-chained, signed, append-only evidence ledger. Violations were introduced by controlled synthetic injection, so the decision at which each violation begins is known exactly.

The model's discriminative performance is reported for completeness and is not a result of this paper. An area under the curve of 0.77 on this dataset is unremarkable, deliberately so: the object under evaluation is the control layer, which should behave the same way over a mediocre model as over a good one.

Results. Table 3 reports the headline measures and Figure 5 plots per-control detection latency.

Table 3. Measured results from the proof of concept. Public UCI Statlog German Credit data with controlled synthetic violation injection; no customer or proprietary banking data.

Measure	**Value**
Decisions processed	1,220
Signed evidence records	5,722
Evidence completeness	5,722 of 5,722, no gaps
Control Observability Coverage (COC)	1.0 across 5 controls

Baseline false-positive rate	0.0
Ledger integrity, full-chain verification	Passed
Tampering test	Altered record correctly detected
Circuit breaker on first fairness breach	Tripped automatically
Mean time to compliance, fairness	184 decisions
Detection latency, C-EXPL-01	0 decisions
Detection latency, C-DATA-01	0 decisions
Detection latency, C-HUMAN-01	3 decisions
Detection latency, C-DRIFT-01	46 decisions
Detection latency, C-FAIR-01	66 decisions
Decision model, test area under the curve	0.77

Coverage was 1.0 across five controls, meaning every deployed descriptor named a signal the instrumented system actually emitted and a predicate that was actually evaluable on live telemetry. Evidence completeness was 5,722 of 5,722 signed records with no gaps, across 1,220 decisions. The baseline false-positive rate was 0.0: on a clean run with no injected violation, no control raised a breach. Full-chain ledger verification passed, and a tampering test in which a stored record was altered was correctly detected. The circuit breaker tripped automatically on the first fairness breach. Mean time to compliance for the fairness control, measured from breach to return to a compliant state, was 184 decisions. Per-control detection latency was 0 decisions for the explainability control, 0 decisions for the lineage control, 3 decisions for the human oversight control, 46 decisions for the drift control and 66 decisions for the fairness control.

Reading the numbers. Three observations are worth making, and one non-observation.

A baseline false-positive rate of 0.0 is a statement about a clean run at this scale with these thresholds, not a general property. It says the predicates do not fire spuriously on compliant traffic in this configuration, which is the necessary precondition for anyone trusting a breach signal. It does not establish a false-positive rate under production traffic, and nothing here should be read as doing so.

Mean time to compliance of 184 decisions against a fairness detection latency of 66 decisions is the more informative pair. Detection is fast relative to recovery. The interval between them is occupied by the response executing and by the sliding window flushing the violating decisions out of scope, which means recovery time for a windowed control is bounded below by its window length. An institution tuning these controls is trading detection latency against false positives, and recovery time against window length, and the descriptor makes both tradeoffs explicit parameters rather than emergent behaviour.

The tampering test result is a positive detection, not an absence of tampering. The test altered a record and confirmed that full-chain verification failed at the expected position. That establishes the detection mechanism works. It does not establish resistance to an adversary with key access, which the construction does not claim.

The non-observation: no transaction-monitoring measurement was taken, and none is reported. No anti-money-laundering or transaction-surveillance control was implemented or instrumented in this proof of concept. Table 4 records this explicitly alongside the dimensions that were measured, and Section 16 states it as future work. Likewise no auditor acceptance was measured, and no claim about assurance-professional acceptance of this evidence appears anywhere in this paper.

Table 4. Evaluation dimensions with measured status. The two dimensions marked as not evaluated are stated as future work in Section 16 and no figure is reported for either anywhere in this paper.

Dimension	Value	Status
Coverage (COC)	1.0 across 5 controls	Measured
Detection latency	0 to 66 decisions by control	Measured
Recovery (MTTC, fairness)	184 decisions	Measured
Evidence completeness	5,722 of 5,722 records, no gaps	Measured
Ledger integrity and tamper-evidence	Verified; tampering detected	Measured
False-positive rate	0.0 at baseline	Measured
Transaction-monitoring figures	Not applicable	**Not evaluated. Future work.**
Auditor acceptance	Not applicable	**Not measured. Future work.**

Maturity assessment. Against the model of Section 14, the proof of concept demonstrates L3 across all five controls: each is evaluated continuously against live telemetry and each generates its own tamper-evident evidence. It reaches L4 for those controls where closed-loop enforcement exists, which in this implementation is the fairness control with its circuit breaker. This is a per-control assessment, and stating it per control rather than as a single level for the system is the honest form.

11. A Continuous Compliance Maturity Model

The reference model includes a five-level maturity model, shown in Figure 6. Its value is not the levels themselves, which resemble other staged models, but the specificity of what changes at each boundary.

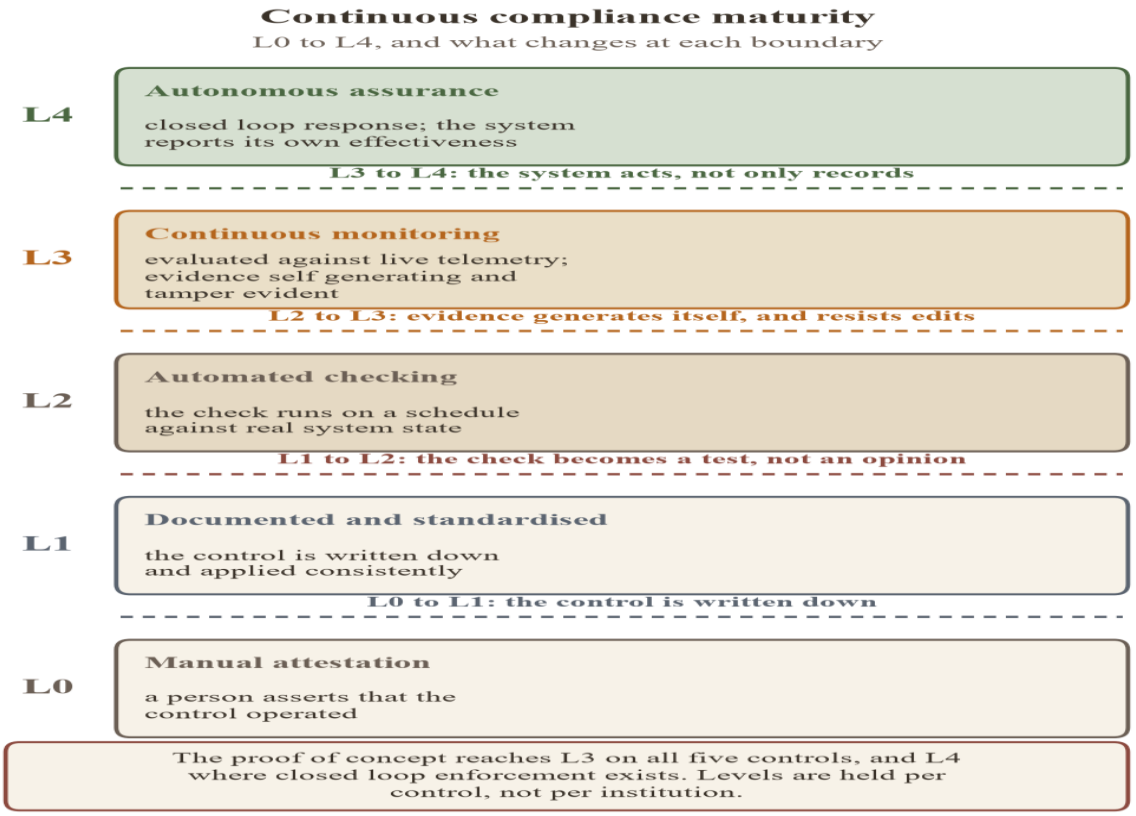

Figure 6. The L0 to L4 continuous compliance maturity model, with the dividing line stated at each boundary. Levels are held per control rather than per institution.

L0, manual attestation. A person asserts that the control operated. The evidence is the assertion.

L1, documented and standardised. The control is written down and applied consistently. What improves is repeatability. What does not improve is the evidence, which is still an assertion, now a better-organised one.

L2, automated checking on a schedule. A program runs the check against real system state. This is the boundary at which the check stops being an opinion and becomes a test. The distinction is categorical rather than incremental: at L1 the question "did the control operate" is answered by a person's judgement, and at L2 it is answered by a program's output. Everything above L2 is a refinement of a test; everything at or below L1 is a refinement of an opinion.

L3, continuous monitoring with self-generating tamper-evident evidence. Controls are evaluated against live telemetry rather than on a schedule, and the evidence is produced by the act of evaluation and is resistant to later alteration. Two things change at this boundary, and both are necessary. Continuity closes the interval between scheduled runs, which is where the assurance gap lives. Self-generating tamper-evident evidence removes the separate step in which evidence is assembled, and with it removes both the assembly cost and the opportunity for the record to be curated after the fact. A system with continuity but editable evidence is not L3, and neither is a system with tamper-evident evidence produced quarterly.

L4, autonomous assurance with closed-loop response. The system acts on its own findings rather than only recording them, and reports on its own effectiveness. The dividing line is action. At L3 a breach produces a signed record and an alert. At L4 a breach produces a signed record, an alert, and a change in system behaviour that removes the exposure.

Two properties of the model matter in use.

Levels are held per control, not per institution. An institution will have controls at different levels simultaneously, and that is normal rather than a failure of consistency. A fairness control may sit at L4 because halting automated decisioning is a proportionate response to a disparate impact breach, while a model documentation control sits at L1 because the obligation is qualitative and no useful test exists. Assigning a single maturity level to an institution averages away the information that matters, which is which specific obligations are covered by tests and which are covered by opinions. This is the same logic that makes production-readiness scoring useful at the level of individual tests and monitors rather than as a single score [13].

The levels are not uniformly desirable. L4 requires that the institution be willing to let the system act, which is appropriate for some controls and inappropriate for others. Progression should follow from the obligation and the risk appetite, not from a wish to be at the top of a ladder.

12. LIMITATIONS

Qualitative obligations resist automation. Where an obligation turns on a judgement of quality, no predicate can discharge it. The framework can check that a rationale is present and was produced within budget; it cannot check that the rationale is meaningful, and the literature on the limits of post-hoc explanation gives good reason to think no automated check could [17], [18], [19]. The framework therefore sits alongside human judgement rather than replacing it, and the honest position is that it converts a subset of obligations into tests and leaves the remainder as opinions, while making the boundary between the two visible.

Keeping controls current when regulation changes is a real and continuing burden. Version control makes the burden tractable and auditable; it does not remove it. Someone must read the new instrument, decompose it into obligations, decide whether existing controls discharge them and write descriptors for the gaps. That work is skilled, recurring, and not automated by anything in this paper. The supersession of the model risk guidance in April 2026 [R2], [R3] is an example of exactly the event that generates it.

A monitor can fail silently, and this is the most dangerous failure mode in the design. If telemetry stops arriving, no control raises a breach, and the absence of breaches is indistinguishable from compliance. Silence looks like compliance. A system whose instrumentation has been disabled or whose collector has failed will report a clean control estate. Controls that watch the controls are therefore required rather than optional: liveness checks on the telemetry pipeline, expected-volume checks that alarm when evidence arrives at less than the expected rate, and heartbeat records that make an absence of evidence itself an observable breach. The proof of concept does not implement this layer, and its absence is a genuine gap in the evaluated system rather than a theoretical concern.

Auditor acceptance is unstudied. Nothing is known from this work about whether an assurance professional will rely on continuously generated, hash-chained evidence, or how such evidence should be sampled, or whether existing audit standards accommodate it. The dimension was not measured. It is the largest open question about the practical value of the whole approach, because a control estate that produces evidence no auditor will rely on has improved internal management and not external assurance.

Governance of systems that act on their own findings is unresolved. A control that trips a breaker takes a consequential business action on the strength of its own measurement. Who approves that action class, what the reversal path is, how a false positive that halts lending is remediated, and how the acting system is itself supervised are open questions. The agentic governance literature identifies the general shape of the difficulty [20]; this paper does not solve it and its L4 level should be read as naming a capability that requires governance rather than as recommending it unconditionally.

This is a single proof of concept. One model, one dataset, one decisioning workflow, five controls, 1,220 decisions. No claim of generality across model families, decision types, control types or scale follows from it.

All evaluation data is public or synthetic. This is a deliberate and, for reproducibility, a positive choice, and it bounds what can be concluded. Synthetic violation injection makes latency measurable precisely because the injection point is known, which is exactly the condition that does not hold in production. Real violations arrive gradually, are confounded with other changes, and do not announce their start.

Not every element of the reference model is implemented. The proof of concept covers instrumentation, capture, evaluation and enforcement over five controls. It does not implement the full mapping and coverage tooling at institutional scale, the controls-watching-controls layer, checkpointing for ledger verification at production volumes, or any control outside the credit decisioning workflow.

13. RESEARCH AGENDA

Transaction monitoring. The controls evaluated here govern a credit decision. Transaction monitoring is the obvious next domain and was not evaluated in this work at all. It differs in ways that matter: the population is transactions rather than applications, the class imbalance is far more severe, the relevant controls concern alert quality and disposition timeliness rather than decision fairness, and the windowed-measure problem is sharper because typologies play out over long horizons. Whether the reference model transfers, and what per-control detection latencies look like in that setting, are open empirical questions.

Auditor acceptance. The question is empirical and answerable. It requires putting continuously generated, chained evidence in front of assurance professionals and internal audit functions and studying what they do with it: whether they rely on it, how they sample it, what additional properties they require first, and whether reliance changes the procedures they perform. This is the highest-value item on the agenda, because it determines whether the approach produces external assurance or only internal management information.

Silent-failure detection. Formalising the controls-watching-controls layer, including what liveness and volume properties are sufficient to make an absence of evidence a detectable breach, and what failure modes remain undetectable.

Latency and false-positive characterisation. Systematic characterisation of the window length, threshold and effect size surface for windowed controls, so that an institution can choose a detection latency budget the way it chooses a service-level objective, with the false-positive consequence quantified rather than discovered.

Governance of self-acting assurance. Approval, reversal, accountability and supervision models for controls that take enforcement actions on their own findings.

Coverage semantics. Control Observability Coverage as defined here measures whether declared controls are evaluable, not whether they adequately discharge the obligations they claim to serve, which remains a judgement. Whether any useful part of that judgement can be formalised is open.

Scale. Ledger volume, verification cost and checkpointing strategy at production decision rates, none of which this evaluation addresses.

14. CONCLUSION

The gap between how fast AI-driven banking systems change and how slowly their controls are validated is not closed by validating harder. It is closed by changing what a control is. A control expressed as a specification that an engine evaluates against the system's own telemetry, on every decision, and that writes a signed and chained record of each verdict, answers the question a supervisor actually asks. It says the control operated on this decision, and this one, and this one, and here is the evidence, and here is why that evidence cannot have been edited afterwards.

The reference model set out here makes that concrete: obligations, controls and signals separated and explicitly linked; a four-move translation that records where institutional judgement entered; a descriptor that is the executable control rather than documentation about one; a many-to-many mapping in which a blind spot is an empty row; a five-stage pipeline from obligation to signed evidence; and a maturity model whose boundaries name what actually changes. The proof of concept shows the mechanism works at the scale tested. Across 1,220 decisions and five controls it achieved full observability coverage, produced 5,722 signed evidence records with no gaps, raised no false positive on clean traffic, verified its own chain, detected a deliberately altered record, and tripped an automated breaker on first breach. Detection latency ranged from 0 decisions for controls that read a single decision to 66 decisions for controls that must fill a window, and that range is structural rather than a defect.

What the work does not show is equally definite. It shows nothing about transaction monitoring, which was not evaluated. It does not show that auditors will accept this evidence, which was not measured and remains the most consequential open question. And it does not show that a monitor cannot fail silently, because it can: if telemetry stops, silence looks like compliance, and controls that watch the controls are not optional.

Compliance-as-code grounded in observability does not make a bank compliant. It makes compliance a live, verifiable property of a running system rather than a periodic assertion about one, and it makes the boundary between what has been tested and what remains an opinion visible to everyone who has to rely on it.

Regulatory, standards and dataset sources

All items below were verified against the dated instrument, not a live landing page. Accessed 12 August 2026.

[R1] Board of Governors of the Federal Reserve System and Office of the Comptroller of the Currency, "Supervisory Guidance on Model Risk Management," SR letter 11-7 attachment and OCC Bulletin 2011-12, 4 April 2011. Verified against the dated 2011 attachment document. Superseded 17 April 2026; see [R2] and [R3].

[R2] Board of Governors of the Federal Reserve System, "Revised Guidance on Model Risk Management," SR letter 26-2, 17 April 2026. States that it supersedes and replaces SR letter 11-7 (issued 4 April 2011) and SR letter 21-8 (issued 9 April 2021).

[R3] Office of the Comptroller of the Currency, "Model Risk Management: Revised Guidance," OCC Bulletin 2026-13, 17 April 2026. Rescinds OCC Bulletin 2011-12 among other issuances.

[R4] Basel Committee on Banking Supervision, "Principles for effective risk data aggregation and risk reporting," January 2013 (BCBS 239). Principle 2 on data architecture and integrated data taxonomies including metadata; Principle 3 on accuracy and integrity; Principle 7 on reconciliation and validation of reports.

[R5] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation), OJ L 119, 4 May 2016. Article 22(1); Article 13(2)(f); Article 15(1)(h).

[R6] Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), OJ L, 2024/1689, 12 July 2024. Article 9(2); Article 10; Article 12; Article 14(1).

[R7] Regulation (EU) 2022/2554 of the European Parliament and of the Council of 14 December 2022 on digital operational resilience for the financial sector (DORA), OJ L 333, 27 December 2022, pp. 1 to 79. Article 9(1); Article 10(1) and 10(2); Article 11.

[R8] Sarbanes-Oxley Act of 2002, Public Law 107-204, section 404; and Public Company Accounting Oversight Board Auditing Standard AS 2201, "An Audit of Internal Control Over Financial Reporting That Is Integrated with An Audit of Financial Statements," paragraphs .52, .B2 and .B4.

[R9] Equal Credit Opportunity Act, 15 U.S.C. 1691 et seq.; Regulation B, 12 C.F.R. Part 1002, section 1002.4(a).

[R10] Uniform Guidelines on Employee Selection Procedures (1978), 29 C.F.R. section 1607.4(D), the four-fifths or eighty percent adverse impact screen.

[R11] OpenTelemetry Specification, version 1.60.0, Cloud Native Computing Foundation. Defines the API, SDK and data specifications for traces, metrics and logs.

[R12] H. Hofmann, "Statlog (German Credit Data)," UCI Machine Learning Repository, 1994. 1,000 instances, 20 features, including a personal status and sex attribute. doi:10.24432/C5NC77

REFERENCES

1. S. Haber and W. S. Stornetta, "How to time-stamp a digital document," Journal of Cryptology, vol. 3, no. 2, pp. 99-111, 1991. doi:10.1007/BF00196791
2. R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," in Advances in Cryptology, CRYPTO '87, Lecture Notes in Computer Science, pp. 369-378, 1988. doi:10.1007/3-540-48184-2_32
3. B. Schneier and J. Kelsey, "Secure audit logs to support computer forensics," ACM Transactions on Information and System Security, vol. 2, no. 2, pp. 159-176, May 1999. doi:10.1145/317087.317089
4. B. Laurie, "Certificate transparency," Communications of the ACM, vol. 57, no. 10, pp. 40-46, Sept. 2014. doi:10.1145/2659897
5. J. Kaldor, J. Mace, M. Bejda, E. Gao, W. Kuropatwa, J. O'Neill, K. W. Ong, B. Schaller, P. Shan, B. Viscomi, V. Venkataraman, K. Veeraraghavan, and Y. J. Song, "Canopy: An End-to-End Performance Tracing and Analysis System," in Proceedings of the 26th Symposium on Operating Systems Principles, pp. 34-50, Oct. 2017. doi:10.1145/3132747.3132749
6. M. Herschel, R. Diestelkämper, and H. Ben Lahmar, "A survey on provenance: What for? What form? What from?," The VLDB Journal, vol. 26, no. 6, pp. 881-906, Dec. 2017. doi:10.1007/s00778-017-0486-1
7. D. Y. Chan and M. A. Vasarhelyi, "Innovation and practice of continuous auditing," International Journal of Accounting Information Systems, vol. 12, no. 2, pp. 152-160, June 2011. doi:10.1016/j.accinf.2011.01.001
8. M. G. Alles, A. Kogan, and M. A. Vasarhelyi, "Putting Continuous Auditing Theory into Practice: Lessons from Two Pilot Implementations," Journal of Information Systems, vol. 22, no. 2, pp. 195-214, Sept. 2008. doi:10.2308/jis.2008.22.2.195
9. I. D. Raji, A. Smart, R. N. White, M. Mitchell, T. Gebru, B. Hutchinson, J. Smith-Loud, D. Theron, and P. Barnes, "Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing," in Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 33-44, Jan. 2020. doi:10.1145/3351095.3372873
10. E. Micheler and A. Whaley, "Regulatory Technology: Replacing Law with Computer Code," European Business Organization Law Review, vol. 21, no. 2, pp. 349-377, 2020. doi:10.1007/s40804-019-00151-1
11. A. Rahman, R. Mahdavi-Hezaveh, and L. Williams, "A systematic mapping study of infrastructure as code research," Information and Software Technology, vol. 108, pp. 65-77, Apr. 2019. doi:10.1016/j.infsof.2018.12.004
12. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," IEEE Access, vol. 11, pp. 31866-31879, 2023. doi:10.1109/ACCESS.2023.3262138

13. E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML test score: A rubric for ML production readiness and technical debt reduction," in 2017 IEEE International Conference on Big Data, pp. 1123-1132, Dec. 2017. doi:10.1109/BigData.2017.8258038
14. J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 1-37, 2014. doi:10.1145/2523813
15. J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under Concept Drift: A Review," IEEE Transactions on Knowledge and Data Engineering, 2018. doi:10.1109/TKDE.2018.2876857
16. M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and Removing Disparate Impact," in Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 259-268, Aug. 2015. doi:10.1145/2783258.2783311
17. C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," Nature Machine Intelligence, vol. 1, no. 5, pp. 206-215, May 2019. doi:10.1038/s42256-019-0048-x
18. Z. C. Lipton, "The mythos of model interpretability," Communications of the ACM, vol. 61, no. 10, pp. 36-43, Sept. 2018. doi:10.1145/3233231
19. M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135-1144, Aug. 2016. doi:10.1145/2939672.2939778
20. A. Chan, R. Salganik, A. Markelius, C. Pang, N. Rajkumar, D. Krasheninnikov, L. Langosco, Z. He, Y. Duan, M. Carroll, M. Lin, A. Mayhew, K. Collins, M. Molamohammadi, J. Burden, W. Zhao, S. Rismani, K. Voudouris, U. Bhatt, A. Weller, D. Krueger, and T. Maharaj, "Harms from Increasingly Agentic Algorithmic Systems," in Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, pp. 651-666, June 2023. doi:10.1145/3593013.3594033 Regulatory, standards and dataset sources All items below were verified against the dated instrument, not a live landing