# Cybersecurity Under Change: Proof-Carrying Assurance via Frozen Records and a One-Residual, One-Clock Certificate

Camilla Josephson *

Posted Date: 2 January 2026

doi: 10.20944/preprints202601.0050.v1

Keywords: human-centred cybersecurity; proof-carrying assurance; frozen record; one-residual one-clock law; Metzler systems; input-to-state stability; Step 6; refinement ladders; AI agents; metric drift

 doi:10.20944/preprints202601.0050.v1



*Article*

# Cybersecurity Under Change: Proof-Carrying Assurance via Frozen Records and a One-Residual, One-Clock Certificate

**Camilla Josephson**

Independent Researcher, Sweden; cmkjosephson@gmail.com

**Abstract**

Cybersecurity assurance drifts under change. Tooling updates, policy revisions, monitoring redesigns, and AI-enabled automation can silently change what is measured, how it is measured, and which differences are treated as "the same," while human workflows adapt under staffing constraints, alert fatigue, incentives, and competing priorities. We introduce a human-centred, proof-carrying approach to security assurance: a certificate layer that freezes one operational record—system boundary, defect definitions, risk scoring ruler, neutrality conventions, audit window, upgrade path, and observation interfaces—so that "improvement under upgrades" has a precise and checkable meaning. Over time, the method combines multiple interacting risk channels into a single decision-ready assurance summary with an explicit improvement margin and an explicit disturbance allowance, designed to remain interpretable during incidents and operational spikes. Across versions and refinements, it enforces a vertical-coherence requirement: upgrade effects must have a finite total footprint so that claims do not drift without bound as systems evolve. We package the framework as four auditable obligations—controlling semantic and policy drift, maintaining a uniform improvement claim, ensuring upgrade coherence, and transporting guarantees to observable evidence—and prove a Master Certificate showing that passing these checks yields version-stable, mechanically verifiable assurance envelopes on the declared episode window. The resulting rates, budgets, and slack are human-centred objects: decision-ready summaries, governance-grade non-regression guarantees, and feasibility diagnostics under organisational constraints.

**Keywords:** human-centred cybersecurity; proof-carrying assurance; frozen record; one-residual one-clock law; Metzler systems; input-to-state stability; Step 6; refinement ladders; AI agents; metric drift

## Notation

| Symbol | Type / Domain | Meaning (declared once and frozen) |
|---|---|---|
| **Time, scale, and windows** | | |
| $T$ | scalar | Episode horizon / audit duration. |
| $[t_0, t_1] = [t_0, t_1]$ | interval | Audit window; typically $[0, T]$ after time shift. |
| $t$ | variable | Time parameter (continuous) or embedded turn index. |
| $s$ | variable | Scale/refinement coordinate (continuous); $s \le s'$ indicates refinement upgrade. |
| $s_0 < s_1 < \cdots$ | grid | Declared refinement grid with $s_K \to \infty$. |
| **Frozen record (certificate measurement geometry)** | | |
| $\mathfrak{P}$ | tuple | Frozen record $\mathfrak{P} = (X, \mathcal{R}, W, \Pi_{\mathcal{N}}, [t_0, t_1], (\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K \ge K_0}, \mathrm{obs})$ (Definition 3.3). |
| $X$ | space | System boundary/state space (socio-technical or agent boundary). |

 doi:10.20944/preprints202601.0050.v1



| Symbol | Type / Domain | Meaning (declared once and frozen) |
|---|---|---|
| $\mathcal{R}$ | space | Ambient measurement space (room) in which defects/representatives are typed. |
| $\mathcal{R}_p$ | space | Defect space on which the ruler acts (often $\mathrm{Ran}(\Pi_\perp) \subseteq \mathcal{R}$). |
| $W$ | operator | Bounded SPD ruler on $\mathcal{R}_p$; defines the frozen measurement norm $\|u\|_W^2 = \langle u, Wu\rangle$. |
| $\Pi_{\mathcal{N}}$ | projector/family | Neutrals (semantic equivalences) in the frozen record. |
| $\Pi_\perp$ | projector | Neutral-free projection $\Pi_\perp := I - \Pi_{\mathcal{N}}$. |
| obs | map | Observation/readout map (logs/traces) used for auditability. |
| **Frozen target spaces and refinement maps** | | |
| $\mathcal{X}_W$ | Banach space | Frozen target space $\mathcal{X}_W := L^\infty([t_0,t_1];\mathcal{R}_p)$ with $\|u\|_{\mathcal{X}_W} = \operatorname{ess\,sup}_{t\in[t_0,t_1]} \|u(t)\|_W$. |
| $\mathcal{X}_s$ | Banach space | Scale-indexed target space; in the simplest stance $\mathcal{X}_s \equiv \mathcal{X}_W$ for all $s$. |
| $\mathcal{P}_{s\to s'}$ | bounded map | Scale comparator $\mathcal{P}_{s\to s'} : \mathcal{X}_s \to \mathcal{X}_{s'}$ for $s \le s'$ (Eq. (73)). |
| $(\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K\ge K_0}$ | ladder datum | Refinement ladder (spaces, rulers, inter-level maps), either continuous in $s$ or discrete in $K$. |
| **Agent pack (one concrete system boundary instance)** | | |
| $x(t)$ | state | Tool-using agent state, e.g. $x(t) = (m(t),\sigma(t),\kappa(t),\mathcal{D}(t),\pi(t))$ (Eq. (1)). |
| | component | Memory/context state (conversation context, scratch, summaries). |
| $x(t) = (m(t),\sigma(t),\kappa(t),\mathcal{D}(t),\pi(t))$ (1) | | |
| $m(t)$ | | |
| $\sigma(t)$ | component | Tool state (bindings, permissions, rate limits). |
| $\kappa(t)$ | component | Retrieval/cache state (index snapshot, embedding version, top-$k$ set). |
| $\mathcal{D}(t)$ | component | Working document/context set given to the LLM. |
| $\pi(t)$ | component | Policy/guardrail configuration (system prompt, filters, redaction rules). |
| **Risk channels, defects, and ledgers** | | |
| $\mathcal{A}$ | finite set | Risk-channel index set (e.g. $\{\mathrm{unsafe}, \mathrm{exfil}, \mathrm{inj}, \mathrm{tool}, \mathrm{drift}\}$). |
| $e^a(t)$ | element | Channel-$a$ defect representative, measurable from $(x(t), \mathrm{obs}(x(t)))$ in the frozen record. |
| $W_a$ | operator | Channel ruler (bounded SPD) used to measure $e^a(t)$. |
| $R^a(t)$ | scalar | Channel ledger $R^a(t) := \|e^a(t)\|_{W_a}^2 \ge 0$ (Eq. (2)). |
| | scalar | Positive aggregation weight for channel $a$. |
| $R^a(t) := \|e^a(t)\|_{W_a}^2 \ge 0$ (2) | | |
| $w_a$ | | |
| $R_{\mathrm{tot}}(t)$ | scalar | Total risk ledger $R_{\mathrm{tot}}(t) := \sum_{a\in\mathcal{A}} w_a R^a(t)$ (Eq. (3)). |
| | element | Stacked defect $e_{\mathrm{tot}} := (\sqrt{w_a}\, e^a)_{a\in\mathcal{A}}$; equivalently $R_{\mathrm{tot}} = \|e_{\mathrm{tot}}\|_{W_{\mathrm{A}}}^2$. |
| $R_{\mathrm{tot}}(t) := \sum_{a\in\mathcal{A}} w_a R^a(t)$ (3) | | |
| $e_{\mathrm{tot}}(t)$ | | |
| $W_{\mathrm{A}}$ | operator | Block-diagonal combined ruler $W_{\mathrm{A}} := \bigoplus_{a\in\mathcal{A}} W_a$. |
| **One-clock law (time-domain core)** | | |
| $\alpha$ | scalar | Certified one-clock margin (Definition 3.2). |
| $\tau_{\mathrm{tot}}(t)$ | function | Nonnegative forcing budget in $L^1([t_0,t_1])$ for the one-clock law. |
| $\|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}$ | scalar | Time-budget footprint (audited constant). |
| **Step 6 (scale-domain core)** | | |
| $e(s)$ | map | Scale path $e : [0,\infty) \to \mathcal{X}_W$ (defect representative as scale varies). |
| $G(s)$ | function | Step 6 majorant with $G \in L^1([0,\infty))$ and $\|\partial_s e(s)\|_{\mathcal{X}_W} \le G(s)$. |
| $\|G\|_{L^1([0,\infty))}$ | scalar | Scale-budget footprint (audited constant). |
| **Scheme budgets and dictionary (O1/O4)** | | |

 doi:10.20944/preprints202601.0050.v1



| Symbol | Type / Domain | Meaning (declared once and frozen) |
|---|---|---|
| $\Delta_{\mathrm{sch}}^{(K)}$ | element | Scheme shift at level $K$ (policy/prompt/template/normalization change) measured in XW-K$_K$. |
| $\delta_K$ | scalar | Summable schedule with $\|\Delta_{\mathrm{sch}}^{(K)}\|_{\text{XW-K}_K} \le \delta_K$ and $\sum_{K\ge K_0}\delta_K < \infty$. |
| $R_{\mathrm{geo}}(t)$ | scalar | Geometric/internal ledger used for the dictionary (often $R_{\mathrm{tot}}$ or a designated sub-ledger). |
| $R_{\mathrm{info}}(t)$ | scalar | Audited readout ledger computed from $\mathrm{obs}(x(t))$. |
| $c_{\mathrm{dict}}, C_{\mathrm{dict}}$ | scalars | Dictionary constants: $c_{\mathrm{dict}}R_{\mathrm{geo}}(t) \le R_{\mathrm{info}}(t) \le C_{\mathrm{dict}}R_{\mathrm{geo}}(t)$. |

## Master definitions (used throughout)

| Entry | Definition / Formula | Role |
|---|---|---|
| Frozen record | $\mathfrak{P} = (X, \mathcal{R}, W, \Pi_{\mathcal{N}}, [t_0,t_1], (\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K\ge K_0}, \mathrm{obs})$ | Declares measurement geometry; fixes meaning of "uniform". |
| Total risk ledger | $R_{\mathrm{tot}}(t) = \sum_{a\in\mathcal{A}} w_a \, \|e^a(t)\|_{W_a}^2$ | Single scalar DSFL diagnostic for assurance. |
| Certified one-clock margin | $\dot{R}_{\mathrm{tot}}(t) \le -2\alpha R_{\mathrm{tot}}(t) + \tau_{\mathrm{tot}}(t)$ with $\tau_{\mathrm{tot}} \in L^1([t_0,t_1])$ | Time-domain contractive spine. |
| Tail-robust envelope | $R_{\mathrm{tot}}(t) \le e^{-2\alpha(t-t_0)}R_{\mathrm{tot}}(t_0) + \int_{t_0}^t e^{-2\alpha(t-s)}\tau_{\mathrm{tot}}(s)\,ds$ | Predictability bound on $[t_0,t_1]$. |
| Frozen target space | $\mathcal{X}_W = L^\infty([t_0,t_1];\mathcal{R}_p)$ with $\|u\|_{\mathcal{X}_W} = \operatorname{ess\,sup}_{t\in[t_0,t_1]}\|u(t)\|_W$ | Scale/coherence statements live here. |
| Projection/extension maps | $\mathcal{P}_{s\to s'} : \mathcal{X}_s \to \mathcal{X}_{s'}$ for $s \le s'$ | Scale-to-scale comparators. |
| Step 6 (scale generator bound) | $\|\partial_s e(s)\|_{\mathcal{X}_W} \le G(s)$ with $G \in L^1([0,\infty))$ | Vertical coherence criterion. |
| Summable increments | $\sum_K \|e(s_{K+1}) - e(s_K)\|_{\mathcal{X}_W} < \infty$ | Cauchy refinement drift. |
| Scheme-unpollutedness (O1) | $\|\Delta_{\mathrm{sch}}^{(K)}\|_{\text{XW-K}_K} \le \delta_K$ with $\sum_{K\ge K_0}\delta_K < \infty$ (or absorption) | Controls scheme/policy drift. |
| Dictionary (O4) | $c_{\mathrm{dict}}R_{\mathrm{geo}} \le R_{\mathrm{info}} \le C_{\mathrm{dict}}R_{\mathrm{geo}}$ on $[t_0,t_1]$ | Transports guarantees to auditable readouts. |
| Human-centred (HC0) | Definition (Human-centred cybersecurity, umbrella). Fix a frozen record $\mathfrak{P} = (X, \mathcal{R}, W, \Pi_{\mathcal{N}}, [t_0,t_1], (\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K\ge K_0}, \mathrm{obs})$ as in Definition 3.3. The assurance problem (and the resulting certificate) is *human-centred* if the HC1–HC4 conditions below hold. | Typing discipline ensuring operational meaning under change. |
| Human-centred (HC1) | **Human-facing semantics (boundary and neutrals).** $X$ and $\Pi_{\mathcal{N}}$ encode the organisation's operational equivalences (what counts as "the same") for decision and accountability. | Prevents semantic drift from being charged as risk. |
| Human-centred (HC2) | **Human-operable geometry (ruler as governance object).** $W$ is chosen so that $\|u\|_W$ measures material deviation from declared targets (policy/safety/risk), not a purely model-internal score. | Makes residual magnitude decision-meaningful. |
| Human-centred (HC3) | **Human-auditable evidence (observation map).** obs is declared so defects and ledgers are recomputable from finite, reviewable evidence (logs/traces/tickets/scans). | Enforces auditability from finite artefacts. |
| Human-centred (HC4) | **Human-usable outputs (decision-ready primitives).** The certificate exports a small stable interface (e.g. one scalar $R_{\mathrm{tot}}$, one margin $\alpha$, explicit time/upgrade budgets) usable during incidents without reconstructing the full state. | Ensures cognitive economy and operational usability. |
| Human-centred (Remark) | **Remark.** "Human-centred" is a typing requirement on the certificate data: $X$ and $\Pi_{\mathcal{N}}$ fix operational irrelevances, $W$ fixes material magnitude, and obs fixes auditability. If any drift, the human meaning of the assurance claim drifts. | Diagnostic: drift in atoms implies drift in meaning. |

 doi:10.20944/preprints202601.0050.v1



## 1. Introduction

Modern digital systems evolve continuously: models are retrained, retrieval indices and schemas change, policy templates and guardrails are revised, scanners and severity rubrics are replaced, tool permissions are widened or narrowed, and monitoring pipelines are re-instrumented [1,2]. A familiar failure mode follows: even when a control or mitigation is unchanged, the *meaning* of the assurance claim can drift because the *measurement geometry* drifts. In practice this happens whenever the evaluation pipeline changes—a new classifier, a new log schema, a new notion of what counts as a neutral reformulation of the same behaviour, or a new definition of the score itself [1]. Teams and regulators then face a moving-goalpost pathology: "we improved" is asserted in a new metric that is not comparable to the old one [3,4].

This is not merely organisational; it is a typed mathematical fact. In stability theory, an exponential decay statement is a statement about a *specific generator acting on a specific normed space* [3,4]. If the norm changes without explicit equivalence constants, statements like "uniform across discretisations" or "uniform across parameters" cease to be invariant [3,4]. The same logic governs assurance under upgrades: if the ruler, neutral conventions, or observation map drift, then the semantics of "risk is decreasing" drifts as well [3,4].

We introduce a proof-carrying certificate layer whose purpose is to prevent that drift. The core move is to declare and freeze a single operational record—the *frozen record*—so that "uniform across upgrades" has a precise, mechanically checkable meaning [1,2]. The frozen record fixes: (i) a system boundary, (ii) a defect typing space, (iii) a single risk ruler (a frozen SPD norm), (iv) a neutral convention, (v) an audit window, (vi) a refinement ladder, and (vii) an observation map. Once these are declared, all guarantees are stated and checked in that geometry (or transported with explicit norm-transfer constants) [2–4].

### 1.1. Why Cybersecurity Is Human-Centred by Construction

We use the term *human-centred* (rather than merely *socio-technical*) because the relevant security properties are *defined*, *enforced*, and *certified* inside a control loop whose decision and verification operators are ultimately human-facing [5,6]. A security property in practice is not a free-floating predicate of a codebase or protocol; it is a property of a deployed system *as operated*: policy authors declare constraints and acceptable risk; engineers implement mechanisms; operators triage and remediate; auditors demand evidence; and governance bodies revise requirements and priorities [5,6]. Thus the semantics of "secure" are fixed by human commitments, and the dynamics of "becoming secure" are constrained by human capacity [6].

Formally, the deployed system closes a feedback loop

$$\text{policy/targets} \longrightarrow \text{mechanisms} \longrightarrow \text{operations} \longrightarrow \text{evidence/audit} \longrightarrow \text{policy/targets},$$

where each arrow is mediated by human practices and organisational processes [5,6]. Workflow structure, staffing constraints, alert fatigue, incentives, and governance cadence bound what controls can be deployed, how quickly deviations can be corrected, and what evidence can be produced and trusted [6]. These constraints are not externalities; they determine the feasible set of implementations and therefore determine which formal guarantees remain valid after deployment [5].

Even when a technical guarantee exists in principle (e.g. a verified component, a cryptographic reduction, a formally proved invariant), it becomes a security guarantee *in the world* only after it is translated into procedures, interfaces, and audit evidence that humans can operate and maintain under continual change [5,6]. This translation introduces additional degrees of freedom (tooling updates, logging schemas, policy templates, exception processes) that can silently alter the meaning of a claimed guarantee unless the measurement and verification geometry are fixed and made checkable [1,3,4]. In this sense, the "human-centred" qualifier is a typing requirement on the guarantee: the claim must be stated in a form that survives the human-operated update loop [6].

 doi:10.20944/preprints202601.0050.v1



Credential hygiene provides a concrete illustration. The well-documented tendency to choose weak or reused passwords when policies are burdensome, together with exception handling and workarounds under time pressure, are not edge cases but primary drivers of realised security outcomes [7–9]. Any assurance architecture that treats these behaviours as exogenous anomalies will systematically mis-estimate risk and overstate robustness [6,9]. A human-centred formulation instead treats such effects as first-class: they appear as explicit forcing/injection budgets and as constraints on achievable improvement margins, making feasibility and auditability part of the guarantee rather than after-the-fact explanations [6,8].

Accordingly, an assurance claim is meaningful only if it is (i) *deployable* (compatible with constraints on time and attention), (ii) *operable* (actionable through a small set of decision variables), (iii) *auditable* (supported by evidence that can be checked), and (iv) *maintainable under change* (stable across tool upgrades, policy revisions, and environment drift) [1,2,5,6]. This paper treats these requirements as a typing discipline: the guarantee must be expressed in a frozen measurement geometry (ruler, neutrals, observation map, and ladder), with explicit budgets for what is not modelled, so that "improvement" remains comparable across versions [3,4].

At the same time, good human-centred security does *not* require all humans to be cybersecurity-centred. The design goal is that most people can act normally without continuous security vigilance, while a small set of roles (operators, engineers, auditors, leaders) interact with security through stable, decision-ready interfaces [6,9]. In DSFL terms, this means concentrating security cognition into auditable primitives (one residual, one clock, explicit budgets and slack), rather than distributing it across the entire user population [6,8].

**Short answer**

**Yes:** cybersecurity must be human-centred because security is defined, deployed, operated, audited, and updated inside human organisations and governance structures.
**No:** humans need not be cybersecurity-centred; the design objective is to make security implicit and low-cognitive-load for most people.

The certificate layer is introduced precisely as a *human-facing interface* for guarantees: it reduces drifting dashboards to a small set of typed, auditable objects (rates, budgets, slack) that can be checked and governed [1,2].

### *1.2. Problem Statement*

We study a verification problem common to modern human-centered security systems, including AI-enabled ones [5,6].

> *Given a deployed system that evolves in time and across refinement scales (tool versions, policy revisions, monitoring pipelines, model and retrieval updates), can we produce a proof-carrying, refinement-invariant safety envelope that is (i) meaningful under upgrades, (ii) compositional across coupled risk channels, and (iii) auditable from a finite bundle?*

Our target output is a *certificate* that can be shipped with a release and checked by a third party, in the spirit of proof-carrying and runtime-verifiable claims [1,2]. The certificate must be explicit about its typing: what is being measured (defects), in what geometry (ruler), modulo what equivalences (neutrals), over what interval (audit window), and along what upgrade path (ladder) [3,4].

### *1.3. Scope and Non-Claims*

Scope.

We develop a proof-carrying DSFL certificate calculus for assurance under change in human-centered security systems, with a primary focus on tool-using LLM agents. The calculus fixes a frozen measurement geometry (ruler, neutrals, observation map, audit window) and treats upgrades as a refinement ladder, so that "uniform across versions" has a checkable meaning [1–4]. As an

 doi:10.20944/preprints202601.0050.v1



auxiliary *avatar-chain* validation domain, we use quantum-computing refinement indices (code distance, concatenation level, circuit depth, decoder granularity) to stress-test the ladder semantics and the bundle-and-checker workflow in a controlled setting [10–12].

Non-claims.

We do *not* propose a new cryptographic primitive, a new intrusion-detection algorithm, or a new vulnerability scoring standard [5]. We do *not* claim to solve alignment or to give a complete semantics of natural language. We do *not* claim that quantum computing and cybersecurity share physical content; the connection is methodological (typed stability under refinement), not ontological [3,4,10]. Where a brick is not proved in-text (e.g. a decay/ringdown gap in a declared ruler, or a domain-specific scheme-control classification), we mark it as an explicit import and state the norm-typing requirements needed to use it [3,4,13–15].

### *1.4. Approach: Frozen Records, One-Clock Reduction, and Vertical Coherence*

The framework is a verification calculus with two orthogonal axes: *time* (how risk evolves on an audit window) and *scale* (how guarantees survive upgrades and refinement) [3,4]. The governing rule is that every claim is interpreted in a *frozen record* (ruler, neutrals, observation map, ladder), so that "uniform across versions" is a checkable statement rather than a narrative [3,4].

Time axis (one-clock safety envelope).

We model assurance as a *vector* of nonnegative ledgers capturing heterogeneous contributors to risk, including human/organisational channels (workload, exception pressure) and technical channels (misconfiguration, identity exposure, data-handling violations, unsafe tool use, prompt-injection success) [5,6]. Channels interact through *nonnegative injections*, yielding a cooperative comparison system of Metzler type [16,17]. Under a Hurwitz (small-gain) margin condition, positive-systems scalarisation collapses the ledger vector to a single scalar residual $R_{\mathrm{tot}}$ obeying a forced contraction law on the audit window:

$$\dot{R}_{\mathrm{tot}}(t) \;\le\; -2\alpha\, R_{\mathrm{tot}}(t) \;+\; \tau_{\mathrm{tot}}(t), \qquad \tau_{\mathrm{tot}} \in L^1([t_0,t_1]),\ \tau_{\mathrm{tot}} \ge 0. \tag{4}$$

The constants $(\alpha, \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])})$ are the certificate's time-domain content: a single improvement margin and a single disturbance footprint, stated in the frozen ruler. This is Lyapunov/semigroup stability expressed in proof-carrying form [3,4,16,18,19].

Scale axis (refinement ladders and Step 6).

Guarantees must survive upgrades: tool versions, policy templates, monitoring pipelines, model weights, retrieval indices, and schema changes [1]. We treat upgrades as movement along a refinement ladder with defect representatives typed in a frozen target space

$$\mathcal{X}_W := L^\infty([t_0,t_1];\mathcal{R}_p). \tag{5}$$

*Vertical coherence* is enforced by a Step 6 condition: refinement drift must have finite total footprint in the frozen norm (equivalently, summable increments on a discrete ladder, or an $L^1$ scale-majorant in continuous scale) [20]. This yields a projective-limit semantics: upgrade effects do not accumulate without bound, so the defect representative converges and the meaning of the certificate cannot drift arbitrarily across versions [20].

Proof-carrying posture (bundle-and-checker semantics).

A release ships a finite bundle: the frozen record, the constants/budgets instantiating the time- and scale-domain inequalities, and a minimal checker that verifies the inequalities in the declared geometry. This is the assurance analogue of proof-carrying code and runtime verification: ship evidence, verify cheaply [1,2].

 doi:10.20944/preprints202601.0050.v1



### 1.5. Two-Domain Positioning and Avatar-Chain Method

The paper is operationally anchored in *human-centred cybersecurity* and tool-using LLM agents, where the measurement-geometry drift problem is acute and governance demands auditable non-regression under change [6]. At the same time, we use *quantum computing* as a deliberately controlled "avatar chain": a domain with explicit refinement indices (code distance, concatenation level, circuit depth, decoder granularity) where frozen rulers, ladder uniformity, and PASS/FAIL certificate checking can be stress-tested numerically [10–12]. References to GR/QFT-in-curved-spacetime enter only as motivating instances of the same typed pathology (norm/gauge/scheme drift) and the same cure (frozen records + ladder coherence), not as a claim of shared physical content [13–15,21].

### 1.6. Main Contributions

**C1** **Frozen-record semantics for assurance under change.** We formalise the typing data required to make cross-version claims invariant: a frozen defect room, a frozen SPD ruler, a neutral convention, an audit window, a refinement ladder, and an observation map. This prevents metric drift from being mistaken for safety improvement [3,4].

**C2** **A one-clock certificate for coupled risk channels.** Under cooperative coupling and a Hurwitz margin condition, we prove a positive weighting reduces multiple ledgers to a single scalar residual obeying (4) with explicit budgets. The resulting envelope is tail-robust and mechanically checkable [16–19].

**C3** **A refinement-ladder coherence theorem (Step 6).** We show that a summable refinement drift condition (or an $L^1$ scale-generator bound) implies vertical coherence in the frozen target norm: upgrade effects have finite total footprint, yielding a well-defined limiting representative and an honest meaning for "uniform across upgrades" [20].

**C4** **A Master Certificate with auditable obligations.** We package the requirements as auditable obligations (scheme/dictionary control, uniform one-clock margin, vertical coherence, and dictionary transport). Passing these checks implies a refinement-invariant safety envelope on the declared audit window [1,2].

### 1.7. What This Paper Does not Claim

This paper does not claim to solve alignment, to provide a complete semantics of natural language, or to reduce all safety questions to a single scalar in an absolute sense. All claims are *relative to the declared frozen record* [3,4]. The point of the framework is to make that dependence explicit: when stakeholders disagree, the disagreement must appear as a disagreement about the frozen record (ruler, neutrals, observation map, ladder) or about which obligation fails—not as an unresolvable narrative dispute [1,2].

### 1.8. Roadmap

The paper is organised to move from motivation and typing discipline to certificate-level theorems, and finally to operational and governance implications.

Section 3 fixes the problem setting and design principles. It formalises human-centred cybersecurity as a dynamical verification problem, introduces residuals as distances to declared constraints, and explains why guarantees must be stated in a frozen measurement geometry to avoid moving-goalpost assurances [3–6].

Section 4 develops the certificate-layer mathematics. It defines defect rooms, frozen rulers, neutral conventions, and nonnegative ledger vectors, and establishes the comparison inequalities used throughout the paper [3,4].

Section 5 proves the core technical result: the one-residual, one-clock theorem. Starting from coupled nonnegative ledgers with cooperative (Metzler) structure, it shows how a single scalar residual with a certified contraction rate and explicit disturbance budget can be constructed, yielding tail-robust envelopes that are decision-ready and auditable [16–19].

 doi:10.20944/preprints202601.0050.v1



Section 6 instantiates the framework for governance and compliance. Policy–practice mismatch is formalised as a typed imbalance operator, and compliance assurance is reframed as a provable contraction margin rather than a snapshot checklist [5].

Section 7 introduces empirical coupling budgets. It explains how cross-channel interactions enter the certificate as explicit constants, and how conservative empirical bounds populate the Metzler comparison model used by the one-clock reduction [16,17].

Section 8 analyses feasibility under human and organisational constraints. It proves improvement-floor results that separate sustainable recovery rates from unavoidable disturbance footprints, making explicit when improvement targets are infeasible given staffing, workflow, and incident intensity [6,8,9].

Section 9 addresses assurance under upgrades. It formalises refinement ladders, contractive comparators, and the Step–6 summability condition, proving that summable refinement drift yields version-stable, projective-limit semantics in a frozen measurement geometry [20].

Section 10 specifies the proof-carrying audit architecture. It defines the certificate bundle, checker interface, PASS/FAIL semantics with slack, and the numerical diagnostics required to instantiate constants conservatively and reproducibly [1,2].

Section 11 demonstrates the workflow on a large block-structured avatar. The goal is not empirical validation of a threat model, but illustration of how frozen records, coupling budgets, gap estimates, and summability checks produce an auditable certificate decision [3,4].

Section 12 discusses implications for human-centred cybersecurity practice. It shows how the certificate layer yields decision-ready summaries, governance-grade non-regression guarantees, and a deployment model for adaptive AI and tool-using agents [1,2,6].

Section 13 records limitations and open problems. These include ruler and residual selection, observability constraints, empirical calibration challenges, and opportunities to tighten scalarisation bounds without losing auditability [3,4].

Section 14 summarises the contribution and outlines a concrete research and deployment agenda based on proof-carrying, frozen-record assurance under continual change [1,2].

## 2. Positioning and State of the Art

This paper sits at an intersection where several technically mature literatures still fail to *compose* into a single, upgrade-stable assurance object [1–4]. In practice, organisations operate security guarantees over long horizons while the measurement pipeline evolves: policies change, tools are replaced, monitoring is re-instrumented, models are retrained, and schemas drift [1,5,6]. The consequence is a moving-goalpost failure mode: improvements can be asserted in a new metric that is not typed back to the old one [3,4]. Our positioning is that this failure mode is not merely managerial; it is structural. A stability claim is typed by a specific evolution operator in a specific norm, and semantics drift is a loss of typing [3,4]. DSFL is offered as a compositional certificate layer whose purpose is to preserve meaning under change by freezing the measurement geometry and routing residual uncertainty into explicit budgets with PASS/FAIL slack [1,2,16,17].

### *2.1. The Position: Cybersecurity Is Human-Centered, but Guarantees Must Be Typed*

Cybersecurity outcomes are not solely technical properties of software stacks. They are embedded in human behaviour, work practices, organisational culture, incentives, and (inter)national governance [5,6]. In operational terms, this means that security claims are determined by:

- *design* (interfaces, defaults, and affordances that shape operator and user behaviour),
- *workflow integration* (how precautions compete with productivity and latency constraints),
- *governance* (policy revision, compliance instruments, accountability, and audit evidence),
- *adversarial and geopolitical context* (crime, conflict, regulation, and standards).

A human-centred approach therefore requires guarantees that remain legible to stakeholders and resilient to organisational change [5,6]. The DSFL stance is that this demand can be made mathe-

 doi:10.20944/preprints202601.0050.v1



matically precise: a guarantee is only comparable across time and versions if it is stated in a frozen measurement geometry (a ruler, a neutral convention, an observation map, and a declared upgrade path) or transported with explicit equivalence constants [3,4].

*2.2. State of the Art: Four Literatures That Do not yet Yield One upgrade-Stable Assurance Object*

(i) Stability and semigroups in fixed norms (typed meaning).

Classical stability theory makes explicit that decay and contractivity are statements about a specific generator acting on a specific normed space [3,4]. This fixed-norm discipline is routine in PDE stability and control, but assurance practice often violates it by changing metrics as tools and policies evolve [3,4]. DSFL imports the discipline as a governing rule: one frozen ruler (no moving goalposts), and cross-version comparisons must either be nonexpansive in that ruler or carry explicit norm-transfer constants [3,4].

(ii) Positive systems, small-gain, and compositional closure (coupled channels).

Multi-channel risk dynamics are naturally modelled as cooperative systems: one channel injects risk into another (workload into misconfiguration, identity drift into policy violations, tool use into exposure), and these injections do not cancel. Positive-systems theory provides sharp stability criteria (Metzler–Hurwitz / small-gain) and a scalarisation mechanism that collapses a nonnegative ledger vector to a single rate-and-budget envelope [16–19]. DSFL adopts these tools but changes the unit of output: coupling coefficients and Hurwitz margin become auditable certificate constants rather than modelling conveniences [16,17].

(iii) Proof-carrying and runtime-verifiable assurance (ship evidence, verify cheaply).

Proof-carrying code formalises a producer/consumer pattern: ship a compact artifact that a small checker can validate [2]. Runtime verification similarly emphasises monitorable properties and checkable traces [1]. Contemporary security assurance frequently lacks an analogous artifact for *assurance under change*: audits often remain narrative, or rely on dashboards whose semantics drift [1,5]. DSFL positions the certificate bundle (frozen record + constants + witnesses + trace commitments) as the corresponding assurance artifact: a third party can re-check the inequalities in the declared geometry and obtain PASS/FAIL with slack [1,2].

(iv) Human-centred cybersecurity and usable security (why drift and overload are first-order).

Usable security and human factors research has shown that security mechanisms fail when they impose cognitive overload, misaligned incentives, and unusable workflows, and that "noncompliance" is often an adaptive response to constraints rather than irrational error [5,6,8,9]. A recurring operational symptom is metric overload with semantic drift: practitioners cannot act on dozens of signals whose meaning changes as tools and policies evolve [6]. DSFL targets this failure mode by collapsing heterogeneous signals into one scalar residual with an explicit contraction margin, an explicit disturbance budget, and explicit coupling constants, so that decision-making can be supported by a small set of stable, auditable objects [16,17].

*2.3. Why a Compositional Certificate Is Still Missing*

Across these literatures, what is still missing is a *single composable certificate* that simultaneously:

(i) fixes the meaning of "small defect" under upgrades by freezing the measurement geometry (or proving explicit norm-transfer constants) [3,4];

(ii) closes multiple interacting channels to one decision-ready "clock" with explicit coupling budgets and Hurwitz slack [16–19];

(iii) separates global security state from what bounded observers can actually audit via an explicit observation map [1];

 doi:10.20944/preprints202601.0050.v1



(iv) yields a proof-carrying artifact (bundle + checker) that supports third-party verification and non-regression narratives as PASS/FAIL inequalities rather than interpretive reports [1,2].

The DSFL contribution is precisely this compositional layer: a frozen-record discipline plus one-clock routing and upgrade-coherence (Step–6) budgets, producing version-stable assurance envelopes with explicit slack [3,4,16,17,20].

### *2.4. Implications: Three Interfaces That Make Guarantees Operational*

To connect the mathematics to practice, we organise the implications around three linked interfaces that are legible to operators, governance bodies, and engineers [5,6].

#### 2.4.1. Decision Interface: Cognitive Economy Under Metric Drift

Problem.

Operators face metric overload and semantic drift: scanner scores, alert counts, compliance percentages, and ML risk scores change meaning as tooling and policies evolve [5,6].

DSFL response.

We replace drifting indicators by one scalar residual in a frozen ruler, together with decision-ready constants: a certified improvement margin (one clock), an $L^1$ disturbance footprint, and explicit coupling constants that explain why local improvements may fail to yield global improvement. The underpinning is the fixed-norm discipline of stability theory [3,4] combined with positive-systems scalarisation for coupled nonnegative ledgers [16,17].

#### 2.4.2. Governance Interface: Continuous Assurance as a Contract

Problem.

Traditional compliance is a point-in-time snapshot and does not control drift between audits [5].

DSFL response.

We model compliance as a typed policy–practice imbalance residual measured in a frozen geometry and certify a contraction margin with an explicit budget on each declared audit window. Auditors verify inequalities over an interval and report PASS/FAIL with slack, rather than relying on checklists whose semantics drift [1–4].

#### 2.4.3. Engineering Interface: Non-Regression for Adaptive AI and Evolving Systems

Problem.

Adaptive systems change: new prompts, tools, updated policies, model updates, retrieval/index snapshots, and telemetry schemas [1].

DSFL response.

We require proof-carrying semantics: upgrades are admissible only if their induced drift is controlled by explicit, summable/integrable budgets (vertical coherence), so that assurance meaning remains stable across versions and can be re-checked by a minimal checker, in the spirit of proof-carrying artifacts [1,2,20].

**Positioning summary**

**DSFL is a compositional certificate layer.** It does not replace cryptography, formal methods, or human-centred interventions. It provides a frozen-record, one-clock, proof-carrying semantics that makes multi-channel assurances comparable under change, auditable by third parties, and actionable by humans through explicit rates, budgets, and slack.

 doi:10.20944/preprints202601.0050.v1



## 3. Problem Setting and Design Principles

This section fixes the *typed* problem setting in which all later theorems live. The central claim is that human-centred cybersecurity is *necessarily* a dynamical, human-centered verification problem: a guarantee must remain meaningful under (i) disturbances, (ii) interventions, and (iii) upgrades that change the measurement pipeline [5,6]. We therefore set up (a) a controlled-process boundary, (b) a residual-based notion of equilibrium, and (c) a frozen measurement geometry that prevents "moving-goalpost" assurances. The mathematics is standard Lyapunov/semigroup stability in fixed norms [3,4,22] combined with a proof-carrying interface notion [1,2].

### *3.1. Human-Centred Cybersecurity as a Dynamical System*

Cybersecurity is inherently time-dependent and human-centered: human workflows, staffing constraints, alert fatigue, organisational incentives, and tool affordances interact with technical controls [5,6,8,9]. A mathematically honest assurance claim must therefore be a statement about the evolution of a system *over a window* rather than a static checklist item [5]. This motivates Lyapunov-style reasoning: one declares a target notion of "acceptable security" (equilibrium / compliance / safe operation), defines a scalar residual measuring distance to that target, and proves that the residual contracts (up to explicit disturbance budgets). This is the stability logic of dissipative dynamical systems and semigroup theory [3,4,18,22].

#### 3.1.1. System Boundary and State

We model a deployed security-relevant human-centered system as a state trajectory

$$x : [t_0, t_1] = [t_0, t_1] \to X, \tag{6}$$

where $X$ is a state space that includes both technical configuration and human/organisational state. The purpose of introducing $X$ is not to claim that all human variables are directly observable, but to make the system boundary explicit: security outcomes depend on internal state, external disturbances, and interventions, and must therefore be stated as properties of a dynamical evolution [5,6].

**Setup 3.1** (human-centered security system as a controlled process)**.** *Fix a finite audit window* $[t_0, t_1] = [t_0, t_1]$*. Let* $X$ *be a (typically high-dimensional) state space, and let* $x(t) \in X$ *denote the system state. We allow:*

(i) interventions $u(t)$ *(training, patching, policy changes, access reviews, monitoring changes);*

(ii) disturbances $d(t)$ *(attacks, outages, user error, vendor changes, workload spikes);*

(iii) observations $o(t)$ *produced by an observation map* obs *(logs, telemetry, tickets, scans), declared precisely in Section 3.2.*

Why this abstraction is necessary.

A human-centred model must explicitly admit that (i) actions are mediated by procedures and tools, (ii) disturbances are unavoidable, and (iii) the observables are not the full state but a logged readout [1,5,6]. This is exactly the setting in which Lyapunov and comparison principles give robust guarantees: one proves inequalities for nonnegative quantities extracted from the trajectory, rather than solving the full dynamics [3,4,18,19].

#### 3.1.2. Equilibria as Constraints and Residuals as Distances

The central object is a nonnegative scalar residual $R(t)$ that measures distance from a declared target set $E \subseteq X$ (e.g. compliance set, safe configuration set, or "acceptable risk" set) [18,19].

**Definition 3.1** (Security equilibrium set and residual)**.** *Let* $E \subseteq X$ *be the declared* equilibrium *(acceptable security) set. A* security residual *is a measurable function* $R : X \to \mathbb{R}_{\geq 0}$ *such that* $R(x) = 0$ *for all* $x \in E$ *and* $R(x) > 0$ *for* $x \notin E$*. Along a trajectory* $x(t)$ *we write* $R(t) := R(x(t))$*.*

 doi:10.20944/preprints202601.0050.v1



**Remark 3.1** (Residuals need not be literal distances). *In practice, $R$ is often derived from a defect map $\Phi : X \to \mathcal{R}_p$ and a ruler $W$ (see Section 3.3), so $R(x) = \|\Phi(x)\|_W^2$. This includes squared-distance-to-constraint as a special case (when $\Phi$ is a constraint operator and $E = \ker \Phi$) [18].*

To connect residuals to computation and audit, it is useful to give a canonical construction.

**Setup 3.2** (Constraint residual via Hilbert projection). *Let $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ be a Hilbert space, let $E \subseteq \mathcal{H}$ be nonempty, closed, and convex, and define*

$$R(x) := \operatorname{dist}(x, E)^2 := \inf_{y \in E} \|x - y\|^2. \tag{7}$$

**Proposition 3.1** (Projection characterisation of squared-distance residual). *Under Setup 3.2, for each $x \in \mathcal{H}$ there exists a unique projection $\Pi_E(x) \in E$ such that*

$$\|x - \Pi_E(x)\| = \operatorname{dist}(x, E),$$

*and $R(x) = \|x - \Pi_E(x)\|^2$. Moreover, $\Pi_E$ is nonexpansive: $\|\Pi_E(x) - \Pi_E(x')\| \le \|x - x'\|$ for all $x, x' \in \mathcal{H}$.*

**Proof.** This is the classical projection theorem in Hilbert spaces (existence/uniqueness for closed convex sets and nonexpansivity of metric projections). □

Human-centred reading.

This is the cleanest semantics for "distance to compliance": $E$ is the set of compliant states in the declared semantics; $\Pi_E(x)$ is the nearest compliant configuration; and $R(x)$ is the squared correction distance. The nonexpansivity of $\Pi_E$ is the formal template behind "admissible remediation steps do not worsen compliance" (see Section 3.3.3) [18,19].

3.1.3. Lyapunov Claims as Certificate Interfaces

We will aim to prove an *inhomogeneous Lyapunov inequality* on $[t_0, t_1]$:

$$\dot{R}(t) \;\le\; -2\alpha\, R(t) \;+\; \tau(t), \qquad \tau \in L^1([t_0, t_1]),\ \tau \ge 0, \tag{8}$$

where $\alpha > 0$ is a certified improvement margin and $\tau$ is an explicit disturbance budget [3,4,18,19].

**Definition 3.2** (Certified one-clock margin). *The constant $\alpha > 0$ in (8) is called the* certified one-clock margin.

**Remark 3.2** (Why "$L^1$ budgets" are the right contract). *The $L^1$ condition on $\tau$ expresses a finite footprint on the declared window. This matches operational practice: incidents can spike, but governance cares about cumulative impact and recovery [5,6]. Analytically, $L^1$ is exactly what yields tail-robust envelopes by the inhomogeneous Grönwall mechanism [18,23,24].*

**Theorem 3.1** (Tail-robust security envelope). *Assume $R$ is absolutely continuous on $[t_0, t_1]$ and satisfies (8). Then for all $t \in [t_0, t_1]$,*

$$R(t) \;\le\; e^{-2\alpha(t-t_0)} R(t_0) + \int_{t_0}^{t} e^{-2\alpha(t-s)}\, \tau(s)\, ds. \tag{9}$$

*In particular,*

$$R(t) \;\le\; e^{-2\alpha(t-t_0)} R(t_0) \;+\; \|\tau\|_{L^1([t_0,t_1])}. \tag{10}$$

**Proof.** Rewrite (8) as $\dot{R}(t) + 2\alpha R(t) \le \tau(t)$. Multiply by $e^{2\alpha(t-t_0)}$, integrate from $t_0$ to $t$, and divide by the integrating factor. This is the standard inhomogeneous Grönwall argument [23,24]. □

Human-centred meaning.

The envelope (9) is actionable: it tells an operator and an auditor (i) how fast the residual contracts when not being hit (margin $\alpha$) and (ii) how much disturbance the organisation absorbed (budget $\tau$) [5]. This is an explicit alternative to narrative assurance [1,2].

 doi:10.20944/preprints202601.0050.v1



### 3.2. Frozen Measurement Geometry ("No Moving Goalposts")

A central design principle is the *frozen record*. All claims are stated relative to a declared measurement geometry consisting of: a ruler (norm), a neutral convention (semantic equivalences), an observation map, and a refinement ladder. Freezing this geometry ensures that statements such as "risk is decreasing" have invariant meaning across tool upgrades and policy changes [3,4]. Without this discipline, guarantees are ill-posed in the same sense that exponential stability claims are ill-posed when the norm changes with time or with the discretization index [3,4].

#### 3.2.1. Definition of a Frozen Record

**Definition 3.3** (Frozen record)**.** *A* frozen record *is a tuple*

$$\mathfrak{P} = (X,\ \mathcal{R},\ W,\ \Pi_{\mathcal{N}},\ [t_0,t_1],\ (\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K\geq K_0},\ \mathrm{obs}), \tag{11}$$

*where:*

(i) $X$ *is the system state space (human-centered boundary);*

(ii) $\mathcal{R}$ *is an ambient measurement space in which defect representatives are typed;*

(iii) $W \succ 0$ *is a bounded SPD ruler on a defect space* $\mathcal{R}_p \subseteq \mathcal{R}$*, defining* $\|u\|_W^2 := \langle u, Wu\rangle$*;*

(iv) $\Pi_{\mathcal{N}}$ *is the* neutral convention *(semantic equivalences), and* $\Pi_\perp := I - \Pi_{\mathcal{N}}$ *is the neutral-free projection;*

(v) $[t_0,t_1] = [t_0,t_1]$ *is the audit window;*

(vi) $(\mathcal{H}_{p,K}, W^{(K)}, \Pi_K^{K+1})_{K\geq K_0}$ *is a declared refinement ladder (tool versions, policy versions, monitoring pipelines);*

(vii) obs *is an observation map that produces auditable readouts (logs/telemetry) from trajectories.*

*All residuals, norms, coupling constants, and budgets are interpreted in the same frozen record* $\mathfrak{P}$*.*

Why each component is necessary.

The ruler $W$ fixes the meaning of "small defect" [3,4]. Neutrals encode invariances (e.g. benign renamings, harmless formatting, approved compensating controls) so the residual does not punish representational accidents [6]. The ladder records the fact that systems evolve. The observation map makes the claim audit-friendly: guarantees must be transportable to what humans can actually see, a principle aligned with proof-carrying and runtime-verification workflows [1,2].

#### 3.2.2. A Formal "No Moving Goalposts" Impossibility Statement

If the measurement geometry is allowed to change arbitrarily, then stability claims can be made true by renorming. The following proposition formalises this pathology [3,4].

**Proposition 3.2** (Ill-posedness of decay claims under uncontrolled re-norming)**.** *Let* $X$ *be a vector space and let* $x(t)$ *be any trajectory with* $x(t) \neq 0$ *on* $[t_0,t_1]$*. There exists a time-dependent norm* $\|\cdot\|_t$ *on* $X$ *such that the function* $R(t) := \|x(t)\|_t^2$ *decays exponentially at any prescribed rate: for every* $\alpha > 0$ *there exists a choice of* $\|\cdot\|_t$ *with*

$$R(t) = R(t_0)e^{-2\alpha(t-t_0)} \qquad (t \in [t_0,t_1]). \tag{12}$$

**Proof.** Fix any reference norm $\|\cdot\|$ and define a scalar weight $c(t) := e^{-\alpha(t-t_0)}\|x(t_0)\|/\|x(t)\|$. Define $\|v\|_t := c(t)\|v\|$. Then $\|x(t)\|_t = c(t)\|x(t)\| = \|x(t_0)\|e^{-\alpha(t-t_0)}$, hence $R(t) = \|x(t)\|_t^2 = R(t_0)e^{-2\alpha(t-t_0)}$. □

Interpretation.

Without a frozen ruler, any improvement claim can be manufactured by changing the measurement geometry [3,4]. In security practice this corresponds to "dashboard illusions": changing a scanner, severity rubric, or instrumentation can create the appearance of improvement without changing reality [5]. Thus, freezing the measurement geometry is logically required, not stylistic.

 doi:10.20944/preprints202601.0050.v1



#### 3.2.3. Neutrals as Semantic Equivalence Classes

Humans routinely treat some differences as irrelevant: renaming roles, reordering log lines, or choosing among equivalent policy templates [6]. If a residual penalises such differences, it creates unnecessary work and reduces trust. We therefore explicitly declare neutrals [6,8].

**Definition 3.4** (Neutral transformations). *A* neutral family $\Pi_{\mathcal{N}}$ *is a set (or finite-dimensional subspace) of transformations on the defect space that represent semantic equivalences. A residual is* neutral-invariant *if* $R(x) = R(\varphi(x))$ *for all* $\varphi \in \Pi_{\mathcal{N}}$*, or at minimum* neutral-stable *if there exist* $0 < c \le C < \infty$ *with*

$$c\,R(x) \le R(\varphi(x)) \le C\,R(x) \qquad (\varphi \in \Pi_{\mathcal{N}}). \tag{13}$$

Design principle.

Neutrals make guarantees human-legible: they prevent certificates from depending on representational accidents [6]. They are the human-centered analogue of gauge/neutral mode removal in stability theory [3,4].

### *3.3. Residuals Versus Metrics*

We distinguish *residuals* from ad-hoc metrics. A residual measures distance to a declared constraint or target set in a declared ruler, while a metric may be an arbitrary score that is not tied to an equilibrium notion [3,4]. Residuals govern action: they decrease under admissible updates and increase only under explicit forcing. This distinction is essential for auditability, compositional reasoning, and for making guarantees meaningful to humans [5,6].

#### 3.3.1. Residuals as Distances to Constraints

**Definition 3.5** (Constraint-residual form). *Let* $E \subseteq X$ *be a declared equilibrium/constraint set and let* $\Phi : X \to \mathcal{R}_p$ *be a defect map such that* $\Phi(x) = 0$ *iff* $x \in E$*. Given a frozen ruler* $W \succ 0$ *on* $\mathcal{R}_p$*, define the residual*

$$R(x) := \|\Phi(x)\|_W^2. \tag{14}$$

Why this matters.

A constraint-residual has falsifiable semantics: $R(x)$ is small only if the constraint defect $\Phi(x)$ is small in the declared ruler. This makes the residual suitable for proof-carrying assurance: it can be recomputed from declared observations and checked [1,2].

#### 3.3.2. Metrics as Scores and the Danger of Non-Transportability

A generic metric $S(x)$ may correlate with security, but without a declared ruler and declared equilibrium set, a score does not support stability reasoning [3,4]. In particular, a score may improve under changes that do not improve the underlying system (e.g. changes in scanners, thresholds, reporting) [5]. Residuals avoid this by being typed in a frozen record (Definition 3.3) [3,4].

#### 3.3.3. Admissible Updates: Why Residuals Support Compositional Reasoning

A key property of residuals is that they can be required to be nonincreasing under admissible updates, while all other effects are budgeted [18,19].

**Definition 3.6** (Nonexpansive (admissible) update in a frozen ruler). *Let* $u \in \mathcal{R}_p$ *be a defect representative and let* $W \succ 0$ *be the frozen ruler. An update map* $\Psi : \mathcal{R}_p \to \mathcal{R}_p$ *is* $W$-nonexpansive *if*

$$\|\Psi(u)\|_W \le \|u\|_W \qquad \forall\, u \in \mathcal{R}_p. \tag{15}$$

**Proposition 3.3** (Residual monotonicity under nonexpansive updates). *Let* $R(u) = \|u\|_W^2$ *be a quadratic residual in the frozen ruler and let* $\Psi$ *be* $W$*-nonexpansive. Then* $R(\Psi(u)) \le R(u)$ *for all* $u$*.*

**Proof.** Immediate: $R(\Psi(u)) = \|\Psi(u)\|_W^2 \le \|u\|_W^2 = R(u)$. □

 doi:10.20944/preprints202601.0050.v1



Human-centred meaning.

An admissible update is a *control primitive* that cannot make the situation worse in the declared ruler [18,19]. This turns procedures into checkable contracts: a training module, an access-hardening script, or an automated remediation is admissible only if it is nonexpansive (or if its non-admissible effects are explicitly budgeted) [1,2,5]. This is how residuals compose across teams, tools, and organisational interventions.

#### 3.3.4. From Residuals to a Decision-Ready Scalar Ledger

Even when multiple residual channels are needed (identity, patching, misconfiguration, exfiltration, unsafe tool use), one can provide a single scalar ledger by aggregating them with positive weights and proving a one-clock inequality via positive-systems (Metzler) reduction [16–19]. The design principle is: use residuals (typed distances) as primitives so aggregation preserves meaning [3,4].

**Remark 3.3** (Design summary)**.** *The three design principles in this section are logically linked: (i) security must be treated dynamically (Lyapunov reasoning) [18,22], (ii) guarantees must be stated in a frozen measurement geometry (no moving goalposts) [3,4], and (iii) residuals (typed distances) are the correct primitives, because they admit nonexpansive admissible updates and explicit forcing budgets [16,17]. Together, these principles create the conditions under which human-centred assurance claims can be auditable and comparable [1,2].*

## 4. Mathematical Framework

This section states the certificate-level mathematics used throughout the paper. The goal is to formalise three objects that are routinely blurred in security practice: *(i)* what is being measured (defects / residuals), *(ii)* how it is being measured (the ruler / frozen measurement geometry), and *(iii)* how multiple channels compose into one auditable guarantee (one-clock closure). Our stance is functional-analytic: stability and decay are statements about an evolution in a fixed normed geometry [3,4]. The DSFL contribution is to make this typing explicit and to provide a reusable routing theorem that turns multiple nonnegative ledgers into one contraction law with explicit budgets.

### *4.1. Residual Spaces and Rulers*

We formalise defect representatives as elements of a Hilbert space equipped with a symmetric positive definite (SPD) operator defining the ruler. The induced norm quantifies the size of deviations. Neutral directions encode semantic equivalences (e.g. role renamings, benign configuration permutations) and are explicitly projected out.

#### 4.1.1. Hilbert Defect Room and the Frozen Ruler

**Definition 4.1** (Defect room and frozen ruler)**.** *A* defect room *is a pair $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ where $\mathcal{H}$ is a real Hilbert space. A* ruler *is a bounded, selfadjoint, strictly positive operator $W : \mathcal{H} \to \mathcal{H}$ (SPD), written $W \succ 0$. The induced inner product and norm are*

$$\langle x, y\rangle_W := \langle x, Wy\rangle, \qquad \|x\|_W := \sqrt{\langle x, x\rangle_W}. \tag{16}$$

*A* frozen record *declares $W$ once, and all certificate claims are interpreted in $\|\cdot\|_W$.*

**Explanation.** The insistence on a frozen ruler is not stylistic. In semigroup stability theory, exponential decay is a statement about a specific generator in a specific norm [3,4]. Changing the norm without explicit equivalence constants changes the meaning of the decay claim. The frozen ruler principle is the mechanism that prevents "metric overload" and "version drift" from becoming mathematically invisible.

 doi:10.20944/preprints202601.0050.v1



#### 4.1.2. Neutral Directions and Semantic Equivalence

In security, different configurations may be semantically equivalent even when they are not bitwise identical: renaming roles, permuting hosts, or changing identifiers may not change the risk semantics. We represent such equivalences by a finite-dimensional neutral subspace and remove it explicitly.

**Definition 4.2** (Neutral subspace and projection). *Let $\mathcal{N} \subset \mathcal{H}$ be a closed subspace (typically finite-dimensional), called the* neutral space. *Let $\Pi_\perp : \mathcal{H} \to \mathcal{H}$ denote the $W$-orthogonal projector onto $\mathcal{N}^{\perp_W}$, i.e. the complement with respect to $\langle \cdot, \cdot \rangle_W$. Given a raw defect $\tilde{e} \in \mathcal{H}$, the* neutral-free defect *is $e := \Pi_\perp \tilde{e}$ and the corresponding residual is*

$$R := \|e\|_W^2. \tag{17}$$

**Explanation.** Neutrals are part of the typing data. If one changes the neutral convention mid-argument, the residual changes meaning. Projecting neutrals out is the certificate analogue of quotienting by a semantic equivalence relation. (For the analogous role of "neutral modes" in stability statements, see standard semigroup/PDE treatments [3,4].)

#### 4.1.3. Norm Transfer and "No Moving Goalposts"

Even when multiple tools produce different measurements, one can compare them only after proving norm equivalence.

**Proposition 4.1** (Quantitative norm transfer between rulers). *Let $W_1, W_2 \succ 0$ be bounded SPD rulers on $\mathcal{H}$. Then there exist constants $0 < c_- \le c_+ < \infty$ such that for all $x \in \mathcal{H}$,*

$$c_- \|x\|_{W_1}^2 \le \|x\|_{W_2}^2 \le c_+ \|x\|_{W_1}^2. \tag{18}$$

*Moreover, $c_\pm$ may be chosen as the extremal spectral values of the bounded SPD operator $W_1^{-1/2} W_2 W_1^{-1/2}$.*

**Proof.** $A := W_1^{-1/2} W_2 W_1^{-1/2}$ is bounded, selfadjoint, and strictly positive, hence its spectrum lies in $(0, \infty)$ with finite infimum and supremum. The Rayleigh quotient bounds

$$\inf \sigma(A) \, \|y\|^2 \le \langle y, Ay \rangle \le \sup \sigma(A) \, \|y\|^2$$

applied to $y = W_1^{1/2} x$ yield the claimed inequalities. □

**Explanation.** Proposition 4.1 is the exact mathematical statement behind "metric drift": if you move between rulers without tracking $c_-, c_+$, you can manufacture apparent improvement or apparent degradation by changing the measuring stick. This is the same norm-typing phenomenon that governs transfer of decay estimates between energies in semigroup theory [3,4].

### *4.2. Multi-Ledger Systems*

Realistic security systems involve multiple nonnegative residuals $R^a(t) \ge 0$, corresponding to different channels such as configuration drift, identity exposure, operational workload, or AI agent misuse. These ledgers interact through coupling terms and forcing budgets. This subsection formalises such systems at the certificate layer using standard cooperative/positive systems ideas [16,17].

#### 4.2.1. Ledger Vector and Nonnegativity

**Definition 4.3** (Nonnegative ledger vector). *Let $\mathcal{A} = \{1, \dots, m\}$ index risk channels. A* ledger vector *is a map $r : [t_0, t_1] \to \mathbb{R}^m_{\ge 0}$ with components*

$$r_a(t) := R^a(t) \ge 0, \qquad a \in \mathcal{A}. \tag{19}$$

*We interpret $R^a$ as a squared $W^a$-norm of a neutral-free defect in a channel-specific defect room, and we regard the nonnegativity $R^a \ge 0$ as the fundamental typing constraint that enables positive-systems routing [16,17].*

 doi:10.20944/preprints202601.0050.v1



**Explanation.** Working with nonnegative ledgers rather than signed metrics prevents cancellation artifacts. At the certificate layer, we only claim what can be audited: nonnegative quantities and explicit budgets.

#### 4.2.2. Couplings and Forcing Budgets

Couplings represent how mismatch in one channel increases mismatch in another (e.g. operational overload increases misconfiguration risk; identity sprawl increases policy-violation likelihood). Forcing budgets represent exogenous injections (e.g. new vulnerabilities disclosed, business-driven policy change, staffing shocks). The cooperative comparison form below is standard in positive systems and input-to-state stability reasoning [16–19].

**Definition 4.4** (Coupled ledger inequality with budgets)**.** *We say the ledgers satisfy a* DSFL comparison model *if $r$ is absolutely continuous and*

$$\dot{r}(t) \;\le\; M(t)\, r(t) \;+\; \tau(t) \qquad \text{for a.e. } t \in [t_0, t_1], \tag{20}$$

*where:*

(i) $M(t) \in \mathbb{R}^{m\times m}$ *is measurable and* Metzler *for a.e. $t$ (off-diagonal entries $\ge 0$);*

(ii) $\tau(t) \in \mathbb{R}^m_{\ge 0}$ *is a nonnegative forcing vector with $\tau \in L^1([t_0, t_1])$.*

**Explanation.** The Metzler constraint encodes the principle that couplings act as *injections*: they can feed risk from one channel into another but cannot cancel it. The $L^1$ condition on $\tau$ is the precise integrability hypothesis needed to obtain explicit envelopes (convolution bounds) for a scalar one-clock ledger [18,19].

### *4.3. One-Clock Reduction (Metzler Closure)*

We introduce a comparison inequality in which the vector of ledgers is bounded by a Metzler matrix. Under a Hurwitz margin condition, the system admits a positive weighting that collapses all ledgers into a single scalar residual obeying a one-clock inequality. Intuitively, many interacting risks reduce to one certified rate of improvement. This is a classical positive-systems phenomenon [16,17].

#### 4.3.1. Metzler Matrices and the Hurwitz Margin

**Definition 4.5** (Metzler and Hurwitz)**.** *A matrix $M \in \mathbb{R}^{m\times m}$ is* Metzler *if $M_{ab} \ge 0$ for all $a \ne b$. It is* Hurwitz *if all eigenvalues have strictly negative real parts. Equivalently, its spectral abscissa $\mu(M) := \max\{\mathrm{Re}(\lambda) : \lambda \in \mathrm{spec}(M)\}$ satisfies $\mu(M) < 0$ [16,17].*

**Explanation.** Hurwitzness is the quantitative statement "the coupled system is globally stable". In DSFL, this is not assumed; it is certified by explicit inequalities (often LP-feasible conditions), as standard in control and positive systems [16–19].

#### 4.3.2. One-Clock Theorem

**Theorem 4.1** (One-clock reduction for Metzler-coupled ledgers)**.** *Let $r : [t_0, t_1] \to \mathbb{R}^m_{\ge 0}$ be absolutely continuous and satisfy (20) with $M(t) \equiv M$ constant Metzler and $\tau \in L^1([t_0, t_1]; \mathbb{R}^m_{\ge 0})$. Assume $M$ is Hurwitz. Then there exists a weight vector $w \in \mathbb{R}^m_{++}$ and a rate $\lambda_{\mathrm{eff}} > 0$ such that the scalar ledger*

$$R_{\mathrm{tot}}(t) := w^\top r(t) \tag{21}$$

*satisfies, for a.e. $t \in [t_0, t_1]$,*

$$\dot{R}_{\mathrm{tot}}(t) \;\le\; -2\lambda_{\mathrm{eff}}\, R_{\mathrm{tot}}(t) \;+\; w^\top \tau(t). \tag{22}$$

*Consequently,*

$$R_{\mathrm{tot}}(t) \le e^{-2\lambda_{\mathrm{eff}}(t-t_0)} R_{\mathrm{tot}}(t_0) + \int_{t_0}^{t} e^{-2\lambda_{\mathrm{eff}}(t-s)}\, w^\top \tau(s)\, ds. \tag{23}$$

 doi:10.20944/preprints202601.0050.v1



**Proof.** Because $M$ is Metzler and Hurwitz, there exists $w \gg 0$ and $\alpha > 0$ such that $M^\top w \leq -2\alpha w$ (a standard positive-systems Lyapunov characterisation; see, e.g., [16,17]). Multiply (20) by $w^\top$ to obtain

$$\frac{d}{dt}(w^\top r) \leq (M^\top w)^\top r + w^\top \tau \leq -2\alpha\, w^\top r + w^\top \tau, \tag{24}$$

using $r \geq 0$. Set $\lambda_{\text{eff}} := \alpha$ and integrate the scalar differential inequality to obtain (23) (inhomogeneous Grönwall; cf. [18,19]). □

**Explanation.** Theorem 4.1 is the certificate kernel: once the channel couplings can be bounded so that the Metzler matrix is Hurwitz, the many-ledger system collapses to one auditable scalar inequality with explicit budgets. This is the precise sense in which "many risks reduce to one certified rate of improvement."

4.3.3. Certificate-Friendly Feasibility Condition

The existence of $w \gg 0$ with $M^\top w \leq -2\alpha w$ is particularly useful because it is a linear feasibility problem.

**Proposition 4.2** (LP-form sufficient condition for a one-clock margin)**.** *Let $M$ be Metzler. If there exist $\alpha > 0$ and $w \in \mathbb{R}^m_{++}$ such that*

$$M^\top w \leq -2\alpha w \quad \textit{(componentwise)}, \tag{25}$$

*then $M$ is Hurwitz and the one-clock inequality (22) holds with $\lambda_{\text{eff}} = \alpha$.*

**Proof.** The inequality gives a linear copositive Lyapunov witness for Metzler stability [16,17], implying Hurwitzness and enabling the scalarisation argument in the proof of Theorem 4.1. □

**Explanation.** Proposition 4.2 is the computational handle: it turns the abstract condition "system is stable" into an auditable check (feasible $w$ and margin $\alpha$) that can be verified by linear programming on finite avatars, consistent with a proof-carrying workflow [2].

4.3.4. Two-Channel Closed Form (for Intuition)

For two ledgers $r = (r_s, r_f)$ with

$$M = \begin{pmatrix} -2\lambda_s & \eta_{s\leftarrow f} \\ \eta_{f\leftarrow s} & -2\lambda_f \end{pmatrix}, \qquad \lambda_s, \lambda_f > 0, \quad \eta_{\cdot\leftarrow\cdot} \geq 0, \tag{26}$$

Hurwitzness is equivalent to the small-gain inequality

$$4\lambda_s\lambda_f > \eta_{s\leftarrow f}\eta_{f\leftarrow s}, \tag{27}$$

and $\lambda_{\text{eff}}$ is determined by the spectral abscissa of $M$ (equivalently the dominant eigenvalue for Metzler $M$), yielding an explicit bottleneck formula (see standard treatments of $2 \times 2$ positive systems and small-gain conditions [16–19]). This is the simplest analytic avatar of the general Metzler closure.

**Remark 4.1** (Why this framework is reusable)**.** *Nothing in this section depends on the semantics of the ledgers (security, quantum computing, or quantum gravity). The only required inputs are: a frozen ruler, explicit neutral conventions, explicit coupling bounds, and integrable budgets. This is why the DSFL certificate layer can be reused across domains. In particular, the same one-clock reduction is standard in the analysis of cooperative systems and positive linear systems [16,17].*

## 5. The One-Residual, One-Clock Theorem

This section gives the core certificate-layer result of the paper. We start from multiple nonnegative ledgers (risk channels) that interact through nonnegative couplings and external forcing. Under a standard positive-systems (Metzler) hypothesis and a Hurwitz margin condition, we prove that there

 doi:10.20944/preprints202601.0050.v1



exists a single scalar residual—a positive weighting of the ledger vector—that obeys a *forced contraction inequality* with one explicit decay rate and one explicit disturbance budget. This is the mathematically precise version of the human-centred question: *"Are we improving, at what certified rate, and how much disturbance did we absorb?"*

The proof is purely functional-analytic: it uses the comparison principle for cooperative systems, Perron–Frobenius theory for Metzler matrices, and the inhomogeneous Grönwall mechanism [3,4,16–19,23–25].

### *5.1. Assumptions (Certificate Layer)*

#### 5.1.1. Ledgers, Regularity, and Forcing Budgets

Let $[t_0, t_1] = [t_0, t_1]$ be a fixed audit window. We consider $m$ nonnegative ledgers (channels) $R^a : [t_0, t_1] \to \mathbb{R}_{\geq 0}$, collected into a vector

$$r(t) := (R^1(t), \ldots, R^m(t))^\top \in \mathbb{R}^m_{\geq 0}. \tag{28}$$

The certificate layer makes only minimal regularity demands: the ledgers must be absolutely continuous so that their time derivatives exist almost everywhere and can be bounded by inequalities [18,24].

**Assumption 5.1** (A1: absolute continuity and nonnegativity)**.** *Each ledger $R^a$ is absolutely continuous on $[t_0, t_1]$ and satisfies $R^a(t) \geq 0$ for all $t \in [t_0, t_1]$.*

External disturbances (attacks, outages, measurement noise, workload spikes) are not required to be small pointwise; they are required to have finite total footprint on the audit window. This is the correct currency for forced-stability envelopes and input-to-state style guarantees [18,19].

**Assumption 5.2** (A2: integrable forcing)**.** *There exists a nonnegative forcing term $\tau : [t_0, t_1] \to \mathbb{R}^m_{\geq 0}$ with $\tau \in L^1([t_0, t_1]; \mathbb{R}^m)$ such that all disturbance contributions to ledger dynamics are upper-bounded by $\tau(t)$ componentwise.*

#### 5.1.2. Metzler Comparison Inequality and Stability Margin

Cross-channel effects in human-centered security systems are inherently cooperative in the sense that one channel can inject risk into another but cannot cancel it. This is captured by a Metzler comparison inequality, the standard abstraction in positive-systems theory [16,17,25].

**Definition 5.1** (Metzler and Hurwitz)**.** *A matrix $M \in \mathbb{R}^{m \times m}$ is* Metzler *if $M_{ab} \geq 0$ for all $a \neq b$. Its spectral abscissa is $\mu(M) := \max\{\operatorname{Re} z : z \in \operatorname{spec}(M)\}$. We call $M$* Hurwitz *if $\mu(M) < 0$.*

**Assumption 5.3** (A3: cooperative (Metzler) comparison inequality)**.** *There exists a constant Metzler matrix $M \in \mathbb{R}^{m \times m}$ such that, for a.e. $t \in [t_0, t_1]$,*

$$\dot{r}(t) \leq M r(t) + \tau(t) \qquad \text{(componentwise)}. \tag{29}$$

Interpretation of $M$.

Writing $M_{aa} = -2\lambda_a$ and $M_{ab} = \eta_{a \leftarrow b} \geq 0$ $(a \neq b)$, the diagonal rates $\lambda_a$ represent intrinsic dissipation/controls in channel $a$, while the off-diagonals $\eta_{a \leftarrow b}$ represent injections from channel $b$ into channel $a$. Hurwitzness is the small-gain condition: dissipation dominates injections [16–19].

**Assumption 5.4** (A4: Hurwitz margin)**.** *The Metzler matrix $M$ in Assumption 5.3 is Hurwitz: $\mu(M) < 0$.*

Why constant $M$ is acceptable at the certificate layer.

Allowing $M(t)$ time-dependent is possible, but the certificate layer aims for auditable constants. In practice, time dependence is either absorbed into the budget $\tau$ or replaced by a conservative constant upper bound. This is consistent with certificate semantics: we prefer a slightly pessimistic but checkable bound to a sharp but unstable one [17–19].

 doi:10.20944/preprints202601.0050.v1



*5.2. Main Theorem*

5.2.1. Positivity and the Forced Comparison Formula

**Lemma 5.1** (Positivity of the semigroup generated by a Metzler matrix)**.** *If $M$ is Metzler, then $e^{tM}$ has nonnegative entries for all $t \geq 0$.*

**Proof.** Choose $\beta$ such that $M + \beta I$ is entrywise nonnegative. Then $e^{t(M+\beta I)} = \sum_{k\geq 0} \frac{t^k}{k!}(M+\beta I)^k$ is entrywise nonnegative. Hence $e^{tM} = e^{-\beta t} e^{t(M+\beta I)}$ is entrywise nonnegative. □

**Lemma 5.2** (Forced comparison (Duhamel) bound)**.** *Assume Assumptions 5.1 and 5.3. Then for all $t \in [t_0, t_1]$,*

$$r(t) \;\leq\; e^{(t-t_0)M} r(t_0) \;+\; \int_{t_0}^{t} e^{(t-s)M} \tau(s)\, ds \qquad \textit{(componentwise)}. \tag{30}$$

**Proof.** Let $y$ solve $\dot{y} = My + \tau$ with $y(t_0) = r(t_0)$, so that Duhamel's formula gives the right-hand side [3,4]. Set $z := y - r$. Then $z(t_0) = 0$ and $\dot{z} \geq Mz$ a.e. Writing $q := \dot{z} - Mz \geq 0$ yields $z(t) = \int_{t_0}^{t} e^{(t-s)M} q(s)\, ds \geq 0$ componentwise by Lemma 5.1. Hence $r \leq y$. □

5.2.2. One-Clock Reduction

We now construct the one scalar ledger that collapses multi-channel risk dynamics into a single certified clock. The key point is that Hurwitzness of a Metzler matrix implies the existence of a strictly positive weighting vector that turns the vector inequality into a scalar inequality with an explicit decay margin [16,17,25].

**Theorem 5.1** (One-residual, one-clock theorem (certificate layer))**.** *Assume Assumptions 5.1–5.4. Then there exist weights $w \in \mathbb{R}^m_{>0}$ and a constant $\alpha > 0$ such that the scalar residual*

$$R_{\mathrm{tot}}(t) \;:=\; w^\top r(t) \tag{31}$$

*satisfies, for a.e. $t \in [t_0, t_1]$,*

$$\dot{R}_{\mathrm{tot}}(t) \;\leq\; -2\alpha\, R_{\mathrm{tot}}(t) \;+\; \tau_{\mathrm{tot}}(t), \qquad \tau_{\mathrm{tot}}(t) := w^\top \tau(t) \in L^1([t_0, t_1]),\ \tau_{\mathrm{tot}} \geq 0. \tag{32}$$

*Moreover, a sufficient (and checker-friendly) witness condition is the existence of $w \gg 0$ and $\alpha > 0$ such that*

$$M^\top w \;\leq\; -2\alpha\, w \qquad \textit{(componentwise)}. \tag{33}$$

**Proof.** Since $M$ is Metzler and Hurwitz, positive-systems theory implies that there exists $w \gg 0$ with $M^\top w \ll 0$ [16,17,25]. Fix any $\alpha > 0$ such that (33) holds. Multiply the comparison inequality (29) on the left by $w^\top$:

$$\dot{R}_{\mathrm{tot}}(t) = w^\top \dot{r}(t) \;\leq\; w^\top M r(t) + w^\top \tau(t) = (M^\top w)^\top r(t) + \tau_{\mathrm{tot}}(t). \tag{34}$$

Since $r(t) \geq 0$ componentwise and $M^\top w \leq -2\alpha w$ componentwise, we have

$$(M^\top w)^\top r(t) \;\leq\; (-2\alpha w)^\top r(t) \;=\; -2\alpha\, R_{\mathrm{tot}}(t), \tag{35}$$

which yields (32). Integrability of $\tau_{\mathrm{tot}}$ follows from $\tau \in L^1$ and $w > 0$. □

Security assurance interpretation.

Theorem 5.1 provides a certificate-level contract: a single scalar ledger decreases at a certified rate $\alpha$ up to an explicit disturbance budget $\tau_{\mathrm{tot}}$. The weights $w$ encode how different risk channels contribute to the decision-ready residual. Crucially, the proof uses only inequality routing, so it remains valid even when the underlying system is complex, non-linear, or partially observed, as long as its effect on the ledgers is upper-bounded by the certificate assumptions [18,19].

 doi:10.20944/preprints202601.0050.v1



LP-checkable certificate interface.

Condition (33) is linear in $w$ and $\alpha$. Thus a checker can validate (or produce) a one-clock margin by solving a linear feasibility problem: find $w \gg 0$ and $\alpha > 0$ such that $M^\top w + 2\alpha w \leq 0$. This makes the theorem operational for proof-carrying security bundles [2].

### *5.3. Tail-Robust Envelopes*

The one-clock inequality implies explicit envelopes that are robust to transient spikes. This is the mathematically precise reason the framework is human-centred: it distinguishes a persistent trend (rate $\alpha$) from accumulated disturbance (budget $\tau_{\mathrm{tot}}$) [18,19,23,24].

**Theorem 5.2** (Tail-robust envelope for the one-clock residual)**.** *Assume the hypotheses of Theorem 5.1. Then for all $t \in [t_0, t_1]$,*

$$R_{\mathrm{tot}}(t) \;\leq\; e^{-2\alpha(t-t_0)} R_{\mathrm{tot}}(t_0) + \int_{t_0}^{t} e^{-2\alpha(t-s)}\, \tau_{\mathrm{tot}}(s)\, ds. \tag{36}$$

*In particular,*

$$R_{\mathrm{tot}}(t) \;\leq\; e^{-2\alpha(t-t_0)} R_{\mathrm{tot}}(t_0) \;+\; \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}. \tag{37}$$

**Proof.** This is the inhomogeneous Grönwall argument applied to (32) [23,24]. □

#### 5.3.1. Interpretation: Incidents as Budgets, Recovery as Rate

Transient spikes.

A short, intense incident produces a large $\tau_{\mathrm{tot}}(t)$ on a small time interval. The envelope (36) shows that its impact is filtered by an exponentially decaying kernel: its long-term effect is proportional to its *integrated footprint* rather than its peak value [18,23,24].

Recovery.

When $\tau_{\mathrm{tot}}$ returns to baseline, the residual returns to exponential decay at rate $\alpha$. Thus the certificate separates "how hard we were hit" ($\|\tau_{\mathrm{tot}}\|_{L^1}$) from "how quickly we recover" ($\alpha$) [18,19].

Human-facing assurance.

In operational terms, Theorem 5.2 supports a minimal, interpretable assurance report:

$$\text{Certified rate } \alpha, \qquad \text{disturbance budget } \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}, \qquad \text{and initial condition } R_{\mathrm{tot}}(t_0). \tag{38}$$

These are the quantities a human decision-maker can use to compare interventions across versions without being misled by metric drift, provided the frozen record discipline holds (Section 3) [3–6].

## 6. Compliance and Assurance as Imbalance

This section instantiates the DSFL certificate layer in a governance/compliance setting. The central move is to treat compliance not as a collection of heterogeneous checklist items, but as a *typed imbalance* in a frozen measurement geometry. Policy is the blueprint; operational practice is the response. The defect is the policy–practice mismatch; the residual is its squared norm in a frozen ruler; and assurance is a *gap statement* (a provable contraction margin with explicit budgets) rather than a binary label.

Throughout, we keep the DSFL hygiene:

- every claim is stated in a declared normed space (frozen ruler), consistent with fixed-norm stability semantics [3,4];
- semantic equivalences are encoded as neutrals and projected out (cf. the necessity of quotienting/removing neutral modes in stability statements) [3,4];
- coupled compliance channels route through a positive-systems (Metzler) one-clock closure [16–19] (Section 4).

 doi:10.20944/preprints202601.0050.v1



### 6.1. Policy–Practice Imbalance Operators

We define imbalance operators for identity and access management, configuration state, data handling, and agent tool use. Each operator measures deviation between declared policy and observed practice and is designed to be compatible with auditability (finite evidence) and stability under change (frozen measurement geometry), in the proof-carrying spirit [1,2].

#### 6.1.1. Typing: Blueprint Space, Practice Space, and Observation Map

Fix a regime window $[t_0, t_1] = [t_0, t_1]$ (e.g. an audit period). Let $\mathcal{H}_s$ be the *policy blueprint* space and $\mathcal{H}_p$ the *practice* space, both Hilbert spaces, and let

$$\mathcal{I} : \mathcal{H}_s \to \mathcal{H}_p \tag{39}$$

be the calibration/interchangeability map that embeds a policy blueprint into the practice space. We also declare a neutral space $\mathcal{N} \subset \mathcal{H}_p$ representing semantic equivalences (role renamings, asset relabelings, benign permutations), and a $W$-orthogonal projector $\Pi_\perp$ onto $\mathcal{N}^{\perp_W}$. Finally, we declare an observation map $\mathcal{O}_t$ (the measurement pipeline) that extracts practice evidence from logs, identity graphs, configurations, and agent traces. Making $\mathcal{O}_t$ explicit is essential for auditability: guarantees must transport to what is observable and checkable from evidence, not merely to an unobserved internal state [1,2].

**Definition 6.1** (Policy–practice imbalance operator (abstract))**.** *A* policy–practice imbalance operator *is a map*

$$E_{\mathrm{pol}} : \mathcal{H}_s \times \mathcal{H}_p \to \mathcal{H}_p \tag{40}$$

*such that* $E_{\mathrm{pol}}(s, p) = 0$ *if and only if practice* $p$ *satisfies policy* $s$ *in the declared semantics. In DSFL form we take*

$$E_{\mathrm{pol}}(s, p) := \Pi_\perp \Big( p - \mathcal{I}s \Big), \tag{41}$$

*possibly after composing* $p$ *with the observation map* $\mathcal{O}_t$ *and any declared normalisation.*

**Explanation.** Definition 6.1 is a typed version of a common audit question: "how far is reality from policy, after quotienting away naming conventions?" The projector $\Pi_\perp$ enforces that only semantically meaningful mismatch is charged to the residual. The requirement that the operator be defined in a Hilbert space with a frozen ruler is exactly the fixed-norm discipline needed for meaningful contraction statements [3,4].

#### 6.1.2. Concrete Imbalance Operators by Channel

We now record canonical examples. In each case, the DSFL rule is: define a *typed defect* that lives in a Hilbert space, then measure it in one frozen ruler. This makes compliance a candidate Lyapunov quantity rather than a checklist score.

Identity and access management (IAM).

Let $G_t$ be the directed graph of identities, roles, entitlements, and resources at time $t$. Let $s^{\mathrm{IAM}}$ be the policy blueprint (allowed edges, separation-of-duty constraints, MFA rules) and let $p^{\mathrm{IAM}}(t)$ be the observed graph (from logs and identity systems). Define

$$E_{\mathrm{IAM}}(t) := \Pi_\perp^{\mathrm{IAM}} \Big( \Phi_{\mathrm{IAM}}(p^{\mathrm{IAM}}(t)) - \mathcal{I}^{\mathrm{IAM}} s^{\mathrm{IAM}} \Big), \tag{42}$$

where $\Phi_{\mathrm{IAM}}$ is a declared feature/embedding of the graph into a Hilbert space (e.g. adjacency features, risk-weighted walk counts), and $\Pi_\perp^{\mathrm{IAM}}$ removes neutral relabellings. Graph-to-Hilbert embeddings are a standard way to make structural constraints accessible to normed analysis; DSFL uses the embedding only as typing data (frozen in the record) so that cross-version comparisons remain meaningful.

 doi:10.20944/preprints202601.0050.v1



Configuration state (hardening posture).

Let $x(t)$ denote the measured configuration vector (CIS controls, patch state, service exposure, runtime settings). Let $s^{\mathrm{CFG}}$ encode the target baseline. Define

$$E_{\mathrm{CFG}}(t) := \Pi_{\perp}^{\mathrm{CFG}}\Big(x(t) - \mathcal{I}^{\mathrm{CFG}} s^{\mathrm{CFG}}\Big), \tag{43}$$

where neutrals can encode benign reparameterisations (e.g. different but equivalent hardening profiles).

Data handling and governance.

Let $d(t)$ be a measured data-flow and classification state (labels, access paths, retention). Let $s^{\mathrm{DATA}}$ encode policy constraints (e.g. retention limits, residency, encryption-at-rest). Define an imbalance $E_{\mathrm{DATA}}(t)$ as the neutral-free distance of observed flows from the policy manifold, for example by a constraint-violation vector or by a quadratic penalty embedding. Treating governance requirements as constraints whose violation has a normed magnitude is standard in control-style assurance: it makes "how far from policy" a quantitative object suitable for Lyapunov reasoning [18,19].

Agent tool use and AI governance.

Let $u(t)$ be a trace representation of agent actions (tools invoked, permissions used, destinations contacted). Let $s^{\mathrm{AG}}$ be the tool-policy blueprint (allowed tools, rate limits, data boundaries). Define

$$E_{\mathrm{AG}}(t) := \Pi_{\perp}^{\mathrm{AG}}\Big(u(t) - \mathcal{I}^{\mathrm{AG}} s^{\mathrm{AG}}\Big), \tag{44}$$

where neutrals may encode harmless trace reordering or equivalent tool aliases. This form matches the proof-carrying posture: the auditable object is the trace and the declared comparison map, verified by a small checker [1,2].

**Remark 6.1** (Why operators, not checklists)**.** *Each $E_\bullet$ is an* operator *because it produces an element of a Hilbert space whose norm can be used in stability claims. A checklist does not carry a geometry; it cannot support a one-clock inequality. The underlying logic is the same as fixed-norm stability theory: without a declared normed space, "decay" is not a typed claim [3,4].*

### *6.2. Compliance Residuals*

Compliance is quantified as a residual $R^{\mathrm{pol}} = \|E_{\mathrm{pol}}\|_W^2$ in the frozen ruler. This formulation supports auditability and comparability, and it aligns directly with Lyapunov-style reasoning for controlled systems [18,19].

#### 6.2.1. Frozen Ruler and the Compliance Residual

**Definition 6.2** (Compliance residual in a frozen ruler)**.** *Fix a frozen SPD ruler $W : \mathcal{H}_p \to \mathcal{H}_p$ and define the neutral-free imbalance*

$$e_{\mathrm{pol}}(t) := E_{\mathrm{pol}}(s, p(t)) \in \mathcal{H}_p. \tag{45}$$

*The* compliance residual *is*

$$R^{\mathrm{pol}}(t) := \|e_{\mathrm{pol}}(t)\|_W^2. \tag{46}$$

**Explanation.** $R^{\mathrm{pol}}$ is a single scalar that aggregates heterogeneous violations by a declared measurement geometry. Because $W$ is frozen, statements like "compliance improved by a factor of 10" have invariant meaning over time and across tool updates (up to explicit norm-equivalence constants), mirroring the fixed-norm discipline in stability theory [3,4]. This is also the correctness condition for proof-carrying assurance: a third party must be able to recompute the residual in the same ruler from the declared evidence [2].

 doi:10.20944/preprints202601.0050.v1



6.2.2. Comparability Under Tool Changes

If the measurement pipeline changes (scanner upgrade, new log source), then the corresponding change of ruler must be made explicit; otherwise one can "improve" compliance by redefining the measurement geometry. This is the same pathology as ill-posed decay claims under uncontrolled renorming in fixed-norm stability theory [3,4].

**Proposition 6.1** (Compliance comparability under ruler changes). *Let $W_1, W_2 \succ 0$ be two rulers representing two measurement pipelines. If there exist constants $c_-, c_+ > 0$ such that*

$$c_- \|x\|_{W_1}^2 \le \|x\|_{W_2}^2 \le c_+ \|x\|_{W_1}^2 \qquad \forall x \in \mathcal{H}_p, \tag{47}$$

*then residual comparisons across the two pipelines are meaningful up to the explicit distortion factor $c_+/c_-$. If no such equivalence constants are established, cross-version comparisons are not certificate-valid.*

**Proof.** This is the norm-transfer fact for bounded SPD rulers; it is the same functional-analytic phenomenon that governs stability transfer between energies [3,4]. □

**Explanation.** Proposition 6.1 formalises the audit intuition: a new scanner may change the scale, so claims must carry the conversion constants. DSFL enforces this as a typing rule, consistent with the proof-carrying stance (ship explicit constants and verify them) [2].

6.2.3. Multi-Channel Compliance as a Ledger Vector

Let $\mathcal{A} = \{\mathrm{IAM}, \mathrm{CFG}, \mathrm{DATA}, \mathrm{AG}, \dots\}$ index compliance channels. Define

$$R^a(t) := \|E_a(t)\|_{W^a}^2, \qquad a \in \mathcal{A}, \tag{48}$$

and assemble a residual vector $r(t) = (R^a(t))_{a\in\mathcal{A}} \in \mathbb{R}_{\ge 0}^{|\mathcal{A}|}$. Couplings (e.g. workload → misconfigurations) appear as nonnegative injections into the ledger inequalities, as in Section 4; the resulting closure uses positive-systems/ISS reasoning [16–19].

*6.3. Assurance as a Gap, not a Checklist*

Instead of binary compliance, assurance is expressed as a rate of improvement toward policy conformance, with explicit slack. This is aligned with the control-theoretic view of safety under forcing and with the fixed-norm semantics of stability theory [3,4,18,19].

6.3.1. Assurance Means a Contraction Margin in the Frozen Ruler

A checkbox view treats compliance as static: PASS/FAIL at a time. DSFL treats assurance as dynamic: does the compliance residual contract with a certified margin under admissible operations?

**Definition 6.3** (Assurance as a one-clock claim). *We say a system has* assurance with margin $\lambda_{\mathrm{eff}} > 0$ *on* $[t_0, t_1]$ *if there exists a scalar compliance ledger $R_{\mathrm{tot}}(t)$ (typically a positive weighting of channel residuals) and a budget $\tau_{\mathrm{tot}} \in L^1([t_0, t_1])$, $\tau_{\mathrm{tot}} \ge 0$, such that*

$$\dot{R}_{\mathrm{tot}}(t) \le -2\lambda_{\mathrm{eff}} R_{\mathrm{tot}}(t) + \tau_{\mathrm{tot}}(t) \qquad \text{for a.e. } t \in [t_0, t_1]. \tag{49}$$

**Explanation.** This is the typed analogue of "security is improving with provable slack, despite change." The inequality is meaningful only in a frozen ruler, exactly as stability statements are meaningful only in fixed norms [3,4].

6.3.2. Metzler Closure Yields Assurance from Coupled Channels

If the channel residuals satisfy a Metzler comparison inequality (Section 4), then assurance reduces to a Hurwitz-margin condition, as in the classical theory of positive linear systems and small-gain/ISS closure [16–19].

 doi:10.20944/preprints202601.0050.v1



**Theorem 6.1** (Assurance by one-clock reduction). *Assume the channel ledger vector $r(t)$ satisfies*

$$\dot{r}(t) \leq Mr(t) + \tau(t), \tag{50}$$

*with $M$ Metzler and $\tau \in L^1([t_0, t_1]; \mathbb{R}^m_{\geq 0})$. If $M$ is Hurwitz, then there exist $w \gg 0$ and $\lambda_{\text{eff}} > 0$ such that $R_{\text{tot}}(t) := w^\top r(t)$ satisfies the assurance inequality (49) with $\tau_{\text{tot}}(t) = w^\top \tau(t)$.*

**Proof.** This is exactly the Metzler one-clock reduction (Section 4), a standard positive-systems argument: Hurwitz Metzler $\Rightarrow$ existence of $w \gg 0$ with $M^\top w \ll 0$, then scalarisation yields the one-clock inequality [16–19]. □

**Explanation.** Theorem 6.1 is the mathematical core of "assurance as a gap": assurance is a provable margin $\lambda_{\text{eff}}$ obtained from explicit coupling bounds, not a checklist score.

6.3.3. Slack Is the Audit Object

In a compliance programme, the most important number is not a binary label but the slack:

$$\text{slack}(t) := 2\lambda_{\text{eff}} R_{\text{tot}}(t) - \tau_{\text{tot}}(t). \tag{51}$$

Positive slack means the system is contracting faster than it is being injected with mismatch; negative slack means the current policy–practice gap cannot be certified as closing under the declared operations.

**Remark 6.2** (Why this improves accountability). *Slack variables are auditable: they reduce disputes about interpretation to inequalities in a frozen record. This is exactly the DSFL posture: progress is "numbers with slack," not narrative conformance. This design aligns with proof-carrying assurance: the bundle ships constants and witnesses; the checker verifies inequalities and returns PASS/FAIL with slack [1,2].*

## 7. Empirical Coupling Budgets

This section explains how the DSFL certificate layer interfaces with empirical security operations. The core idea is simple: *multi-channel assurances are only as strong as the coupling bounds that connect the channels*. If workload spikes can inject risk into configuration drift, or if alert volume can inject risk into human error and policy exceptions, then a certificate that ignores these couplings is brittle. DSFL therefore treats coupling coefficients as first-class certificate constants: single numbers that must be declared, audited, and stress-tested.

Mathematically, the point is that the one-clock theorem (Sections 4–5) requires an explicit Metzler comparison system. Its off-diagonal entries are precisely the coupling budgets [16–19]. Hence, empirical coupling estimation is not "extra analytics"; it is the mechanism by which operational reality enters the certificate. Human factors motivate why coupling is first-order: under cognitive load and organisational constraints, interventions in one channel can amplify failures in another, so assurances that ignore coupling invite "local improvement / global regression" failure modes [5,6].

### *7.1. Why Coupling Matters in Security*

Human workload, alert volume, and configuration drift influence each other. Ignoring these couplings leads to brittle assurances [5,6].

7.1.1. Coupling Is the Difference Between Local and Global Guarantees

Consider two ledgers:

$$R^{\text{cfg}}(t) \quad \text{(configuration drift / misconfiguration residual)} \qquad \text{and} \qquad R^{\text{hum}}(t) \quad \text{(human workload / operational overload residual).} \tag{52}$$

Even if each ledger would decay in isolation (e.g. via automation or staffing interventions), coupling can defeat global contraction. A simple comparison model is

$$\dot{R}^{\text{cfg}} \leq -2\lambda_{\text{cfg}} R^{\text{cfg}} + \eta_{\text{cfg}\leftarrow\text{hum}} R^{\text{hum}}, \qquad \dot{R}^{\text{hum}} \leq -2\lambda_{\text{hum}} R^{\text{hum}} + \eta_{\text{hum}\leftarrow\text{cfg}} R^{\text{cfg}}. \tag{53}$$

 doi:10.20944/preprints202601.0050.v1



The coupled system contracts only if the induced Metzler matrix is Hurwitz (small-gain) and the forcing budgets are integrable, as in the stability theory of positive systems [16–19]. Thus, coupling constants are not optional: they are the difference between "each team metric improved" and "the organisation is provably safer" [5].

#### 7.1.2. Operational Meaning of Couplings

In practice, coupling mechanisms include:

- **Workload → misconfiguration:** high alert load increases the probability of rushed changes and policy exceptions, inflating configuration residuals.
- **Misconfiguration → workload:** drifting configurations create more alerts and incidents, inflating workload.
- **Identity sprawl ↔ policy violations:** expanding entitlements increases both violation frequency and response overhead.

DSFL formalises these effects as nonnegative injections between ledgers, i.e. off-diagonal entries of a Metzler matrix. The certificate is only meaningful once these numbers are bounded in the frozen record. This is aligned with the human-centred security viewpoint that many "technical" failures are mediated by workflow and governance constraints rather than by missing primitives [5,6].

### *7.2. Model-Free Empirical Coupling Estimate*

We define an empirical coupling constant $\kappa_{\mathrm{emp}}$ from observed time series and prove an immediate inequality bounding cross-channel injections. The goal is deliberately certificate-facing: produce conservative, auditable constants that can populate a Metzler comparison matrix and be used for one-clock closure [16,17].

#### 7.2.1. From Coupled Inequalities to an Observable Injection Trace

At the certificate layer, coupling appears as a term that contributes to the derivative of a ledger. For two channels $a$ and $b$, we write the abstract form

$$\dot{R}^a(t) \;\le\; \cdots \;+\; J_{a\leftarrow b}(t) \;+\cdots, \tag{54}$$

where $J_{a\leftarrow b}(t)$ is the *injection trace* measuring how channel $b$ feeds channel $a$ at time $t$. In empirical settings, $J_{a\leftarrow b}(t)$ is not derived from PDE identities; it is estimated from data by a declared protocol (e.g. regressions on interventions, causal models, or direct bookkeeping of change events). The DSFL requirement is not a particular statistical method; it is that the output be conservative and typed to the frozen record so that it can safely enter a Lyapunov/ISS-style inequality [18,19].

**Remark 7.1** (Typing requirement)**.** *The injection trace must be expressed in units compatible with the residuals it couples. If $R^a$ and $R^b$ are computed in a frozen ruler, then $J_{a\leftarrow b}$ must be derived from the same frozen data pipeline; otherwise the coupling estimate is not DSFL-typed.*

#### 7.2.2. Definition of the Empirical Coupling Constant

**Definition 7.1** (Empirical coupling constant)**.** *Fix a time grid $\mathcal{T} \subset [t_0, t_1]$ and observed time series*

$$\{J_{a\leftarrow b}(t)\}_{t\in\mathcal{T}}, \qquad \{R^a(t)\}_{t\in\mathcal{T}}, \qquad \{R^b(t)\}_{t\in\mathcal{T}}, \tag{55}$$

*with $R^a(t), R^b(t) \ge 0$. For a small stabilisation parameter $\varepsilon > 0$, define*

$$\kappa_{\mathrm{emp}}(a \leftarrow b) := \max_{t\in\mathcal{T}} \frac{|J_{a\leftarrow b}(t)|}{\sqrt{R^a(t)\,R^b(t)} + \varepsilon}. \tag{56}$$

**Explanation.** The normalisation $\sqrt{R^a R^b}$ is certificate-friendly because it matches the Cauchy–Schwarz structure of cross terms and yields a dimensionless coupling coefficient; it is also the scaling used when

 doi:10.20944/preprints202601.0050.v1



routing bilinear interactions into Metzler/ISS-style scalar envelopes [16–19]. The small $\varepsilon$ prevents division by zero and is recorded as part of the frozen audit protocol.

7.2.3. Immediate Inequality (the Certificate-Ready Bound)

**Proposition 7.1** (Empirical coupling inequality)**.** *Let $\kappa_{\mathrm{emp}}(a \leftarrow b)$ be as in Definition 7.1. Then for all $t \in \mathcal{T}$,*

$$|J_{a\leftarrow b}(t)| \;\le\; \kappa_{\mathrm{emp}}(a \leftarrow b)\,\sqrt{R^a(t)\,R^b(t)} \;+\; \varepsilon\,\kappa_{\mathrm{emp}}(a \leftarrow b). \tag{57}$$

**Proof.** By definition, for each $t \in \mathcal{T}$,

$$\frac{|J_{a\leftarrow b}(t)|}{\sqrt{R^a(t)R^b(t)}+\varepsilon} \le \kappa_{\mathrm{emp}}(a \leftarrow b). \tag{58}$$

Multiply both sides by $\sqrt{R^a(t)R^b(t)}+\varepsilon$ to obtain (57). □

**Explanation.** Proposition 7.1 is deliberately elementary: it shows that once the time series are declared, the empirical coupling constant immediately yields a certificate-ready inequality. This is the DSFL posture: coupling claims are not interpretive; they are inequalities with explicit constants, designed to be routed into a one-clock certificate check [1,2].

7.2.4. From $\sqrt{R^a R^b}$ to a Metzler Comparison Form

To route coupling into the one-clock theorem we typically use the bound

$$\sqrt{R^a R^b} \le \tfrac{1}{2}(R^a + R^b), \tag{59}$$

so the inequality (57) yields a linear-in-ledgers injection bound of Metzler type:

$$|J_{a\leftarrow b}(t)| \;\le\; \tfrac{1}{2}\kappa_{\mathrm{emp}}(a \leftarrow b)\,R^a(t) + \tfrac{1}{2}\kappa_{\mathrm{emp}}(a \leftarrow b)\,R^b(t) + \varepsilon\,\kappa_{\mathrm{emp}}(a \leftarrow b). \tag{60}$$

Thus empirical couplings become explicit off-diagonal entries and diagonal penalties in the Metzler matrix used for one-clock closure [16–19].

*7.3. Interpretation as a Certificate Constant*

The coupling constant enters assurance reports as a single auditable number, with robustness under resampling. This is the empirical analogue of theorem constants in proof-carrying and runtime-verifiable workflows: ship a conservative constant plus a declared protocol, and verify the resulting inequality cheaply [1,2].

7.3.1. Certificate Semantics: What $\kappa_{\mathrm{emp}}$ Means

In DSFL, $\kappa_{\mathrm{emp}}(a \leftarrow b)$ is interpreted as:

> *The worst-case observed rate (in the declared frozen record) at which channel b can inject mismatch into channel a, normalised by the geometric mean of the two residuals.*

Large $\kappa$ means "tight coupling" and predicts that local improvements may not yield global improvement unless diagonal gaps are large enough to dominate couplings (small-gain) [16–19].

7.3.2. Robustness Under Protocol Variation and Resampling

A single max-over-time estimate can be noisy. DSFL therefore treats robustness as part of the certificate, in the same spirit as conservative constant selection in verification and runtime monitoring [1,2].

 doi:10.20944/preprints202601.0050.v1



**Definition 7.2** (Robust empirical coupling constant (grid/seed worst case))**.** *Let $\mathcal{G}$ be a finite grid of protocol settings (window choice, smoothing, downsampling, feature set) and $\mathcal{S}$ a finite set of resampling seeds (bootstraps). Define*

$$\kappa_{\mathrm{rob}}(a \leftarrow b) := \max_{g \in \mathcal{G}} \max_{s \in \mathcal{S}} \kappa_{\mathrm{emp}}^{(g,s)}(a \leftarrow b), \tag{61}$$

*where $\kappa_{\mathrm{emp}}^{(g,s)}$ is computed from the time series produced by protocol $(g, s)$.*

**Explanation.** $\kappa_{\mathrm{rob}}$ is a conservative certificate constant: it is designed to be hard to "pass by luck". It preserves the DSFL typing discipline (same ruler, same semantics) while stabilising the estimate against plausible measurement perturbations [18,19].

7.3.3. How $\kappa$ Appears in an Assurance Report

In a proof-carrying assurance report, the couplings appear as explicit entries of the Metzler matrix $M$ used by the one-clock theorem (Section 4). Concretely:

- each off-diagonal coupling $M_{ab}$ is instantiated by a conservative bound such as $\frac{1}{2}\kappa_{\mathrm{rob}}(a \leftarrow b)$;
- diagonal terms include the certified gaps minus any diagonal penalties induced by coupling bounds;
- the report records the *slack* in the Hurwitz margin condition (LP-feasible $\alpha$ and witness $w \gg 0$), not just the raw couplings.

This is exactly the certificate posture: coupling constants are not narrative; they are inputs to a checkable stability test [2,16,17].

**Takeaway**

**Coupling as a single auditable number.** DSFL turns cross-channel interactions into a finite list of constants $\{\kappa(a \leftarrow b)\}$. These constants are not narrative; they are certificate inputs. They decide whether one-clock closure holds, hence whether an assurance claim is valid under change.

## 8. Floors and Feasibility Under Human Constraints

This section formalises a human-centred constraint that is routinely ignored in purely technical security models: *security improvement has floors*. Even when the underlying technical system is stabilisable, the achievable improvement rate is limited by human attention, staffing, organisational latency (change control, reviews), and workflow friction. From the DSFL certificate perspective, these limitations appear as bounds on (i) how large a *dissipation margin* can be sustained, and (ii) how large the *disturbance footprint* must remain. We therefore introduce mathematically precise floor quantities and prove a theorem that links persistent dissipation plus capacity-bounded forcing to a guaranteed improvement floor.

The underlying analytic tools are standard: inhomogeneous Grönwall inequalities [23,24], semigroup stability logic [3,4], and dissipativity/ISS-style reasoning [18,19]. The novelty is the *interpretation*: constants in these inequalities become organisational feasibility objects. The human-centred motivation follows the usable-security lesson that security outcomes are constrained by workflow, cognition, and incentives, and that operational reality creates hard limits on what "faster remediation" can mean in practice [5,6].

### *8.1. Why Security Improvement Has Floors*

8.1.1. Human and Organisational Mechanisms That Generate Floors

In a purely technical control model, one may imagine increasing a stabilising gain arbitrarily to make convergence arbitrarily fast. Human-centred cybersecurity cannot do this. Three structural mechanisms impose floors [5,6]:

 doi:10.20944/preprints202601.0050.v1



(i) Attention and throughput bounds.

Incident triage, access reviews, policy exceptions, and remediation require human attention. If the arrival rate of actionable items exceeds human throughput, backlog grows and risk cannot decay faster than the rate at which work can be processed [6].

(ii) Workflow latency and batching.

Many controls are constrained by change-management latency (testing, review boards, deployment windows). Thus even if a fix is known, it cannot be applied immediately; residual reduction is rate-limited by organisational cadence [5].

(iii) Fatigue and diminishing returns.

Training and behavioural interventions have diminishing returns under repeated exposure. In dynamical terms, the effective dissipation margin contributed by training saturates and cannot be increased indefinitely [6].

These mechanisms imply: there exists a maximal sustainable improvement rate, and a minimal unavoidable forcing footprint, even when the system is stable in principle. A certificate that ignores floors will propose infeasible interventions and create incentives for metric gaming [5].

#### 8.1.2. Formal Encoding at the Certificate Layer

In the certificate spine, the one-clock inequality

$$\dot{R}(t) \;\le\; -2\alpha(t)\,R(t) \;+\; \tau(t), \qquad \tau \ge 0, \tag{62}$$

has two interpretable degrees of freedom:

(i) $\alpha(t)$ is the *effective improvement margin* (how much dissipation we can sustain);

(ii) $\tau(t)$ is the *disturbance/arrival footprint* (how much risk is injected or how much work arrives).

Human constraints bound both:

$$\alpha(t) \;\le\; \alpha_{\max} \quad \text{(limited intervention strength)}, \qquad \int_t^{t+\Delta} \tau(s)\,ds \;\ge\; \underline{B}(\Delta) \quad \text{(unavoidable workload/disturbance)}. \tag{63}$$

Floors are therefore not metaphysical: they are explicit constraints on certificate parameters, and they should be interpreted through ISS-style feasibility logic [18,19].

### *8.2. Asymptotic Improvement Floors*

Floors are long-run statements. Instantaneous rates may fluctuate due to incidents, releases, or staffing changes. We therefore define floor quantities using time averages, which are robust and auditable, and standard in stability analysis where asymptotic rates are extracted from log-slopes and averaged dissipation [3,4,18].

#### 8.2.1. Effective Rate and Average Exponent

Let $R : [t_0, \infty) \to (0, \infty)$ be absolutely continuous. Define the *backward effective rate*

$$\alpha_{\text{eff}}(t) \;:=\; -\frac{1}{2}\frac{\dot{R}(t)}{R(t)} \qquad \text{for a.e. } t. \tag{64}$$

If $R$ obeys $\dot{R}(t) \le -2\alpha R(t)$ with constant $\alpha > 0$, then $\alpha_{\text{eff}}(t) \ge \alpha$ a.e. More generally, $\alpha_{\text{eff}}$ quantifies the *realised* contraction speed, a standard device in Lyapunov/ISS analyses [18,19].

**Definition 8.1** (Asymptotic average decay exponent)**.** *The* asymptotic average decay exponent *is*

$$\alpha_\infty \;:=\; \liminf_{T\to\infty} \frac{1}{T}\int_{t_0}^{t_0+T} 2\alpha_{\text{eff}}(t)\,dt \;=\; \liminf_{T\to\infty} \frac{1}{T}\Big(\log R(t_0) - \log R(t_0+T)\Big). \tag{65}$$

 doi:10.20944/preprints202601.0050.v1



Interpretation.

$\alpha_\infty$ is the long-run exponential slope of the residual in log scale. It is the correct object for governance: it measures the net improvement rate after accounting for transient incidents [18,19].

8.2.2. Feasible Floors

A *feasible floor* is a lower bound on $\alpha_\infty$ that holds across all trajectories consistent with declared human and organisational constraints.

**Definition 8.2** (Feasible improvement floor)**.** *A constant $\alpha_{\text{floor}} \geq 0$ is a* feasible improvement floor *on a class of deployments if every residual trajectory $R$ generated under that class satisfies*

$$\alpha_\infty \geq \alpha_{\text{floor}}. \tag{66}$$

We now state a theorem that gives a rigorous and auditable floor criterion. The theorem does *not* pretend that residuals always decay to zero; in realistic environments $\tau$ does not vanish. Instead, it proves: if the dissipation margin is persistently positive and the long-run forcing footprint is capacity-bounded, then the system admits a guaranteed improvement floor and an explicit asymptotic bound on the residual magnitude. This is an ISS-style statement: persistent inputs imply an ultimate bound, and sustained dissipation controls the return rate [18,19].

8.2.3. A Capacity Model for Forcing

We model the unavoidable long-run footprint of incidents and workload by a bound on average forcing density. This is the most operationally natural assumption: it says that on any long window of length $T$, the total injected disturbance is at most linear in $T$, matching standard "bounded average input" hypotheses in ISS [18,19].

**Assumption 8.1** (Capacity-bounded forcing density)**.** *There exists $B \geq 0$ such that for all $T > 0$,*

$$\int_{t_0}^{t_0+T} \tau(t)\, dt \;\leq\; B\, T. \tag{67}$$

Interpretation.

$B$ is an "incident/workload intensity" measured in the same currency as $R$. It represents the minimum unavoidable load the organisation must absorb per unit time. Importantly, (67) is audit-friendly: it can be estimated from logs and tickets, consistent with the proof-carrying posture that claims should be checkable from declared evidence [1,2].

8.2.4. A Persistent Dissipation Hypothesis

A floor requires persistent dissipation: some mechanism must continuously drive the residual down. In human-centred terms, this encodes sustainable governance practices rather than one-off improvements [5,6].

**Assumption 8.2** (Persistent dissipation margin)**.** *There exists $\underline{\alpha} > 0$ such that for a.e. $t \geq t_0$,*

$$\dot{R}(t) \;\leq\; -2\underline{\alpha}\, R(t) \;+\; \tau(t), \qquad \tau(t) \geq 0. \tag{68}$$

8.2.5. The Floor Theorem and Its Consequences

**Theorem 8.1** (Improvement floor under persistent dissipation and capacity-bounded forcing)**.** *Assume $R$ is absolutely continuous, $\tau \in L^1_{\text{loc}}$, and Assumptions 8.1 and 8.2 hold. Then:*

(i) ***Uniform ultimate bound.*** *For all $t \geq t_0$,*

$$R(t) \;\leq\; e^{-2\underline{\alpha}(t-t_0)} R(t_0) \;+\; \frac{B}{2\underline{\alpha}}. \tag{69}$$

 doi:10.20944/preprints202601.0050.v1



*In particular,* $\limsup_{t\to\infty} R(t) \le \frac{B}{2\underline{\alpha}}$.

(ii) ***Floor interpretation.*** *If* $B = 0$ *(no sustained forcing), then* $R(t)$ *decays exponentially with rate* $\underline{\alpha}$ *and*

$$\alpha_\infty \;\ge\; \underline{\alpha}. \tag{70}$$

*If* $B > 0$*, then the best possible long-run outcome is a neighbourhood of size* $\frac{B}{2\underline{\alpha}}$*, and the meaningful "floor" becomes a* floor on the transient recovery rate *back to this neighbourhood after spikes.*

**Proof.** Apply the tail-robust envelope (inhomogeneous Grönwall) to (68) [23,24]:

$$R(t) \;\le\; e^{-2\underline{\alpha}(t-t_0)}R(t_0) \;+\; \int_{t_0}^{t} e^{-2\underline{\alpha}(t-s)}\tau(s)\,ds. \tag{71}$$

Partition the integral into unit intervals and use (67) on each interval: for $s \in [t-k-1, t-k]$ we have $\int_{t-k-1}^{t-k} \tau \le B$. Thus

$$\int_{t_0}^{t} e^{-2\underline{\alpha}(t-s)}\tau(s)\,ds \;\le\; \sum_{k=0}^{\infty}\Big(\sup_{u\in[t-k-1,t-k]}\int_{u}^{u+1}\tau(s)\,ds\Big)e^{-2\underline{\alpha}k} \;\le\; B\sum_{k=0}^{\infty}e^{-2\underline{\alpha}k} \;=\; \frac{B}{1-e^{-2\underline{\alpha}}} \;\le\; \frac{B}{2\underline{\alpha}}, \tag{72}$$

where the last inequality uses $1 - e^{-x} \ge x/2$ for $x \in (0, 1]$ and is valid up to adjusting constants. This yields (69) and the lim sup claim.

If $B = 0$ then $\tau \equiv 0$ and (68) gives $R(t) \le e^{-2\underline{\alpha}(t-t_0)}R(t_0)$. Taking logs and dividing by $t$ yields $\alpha_\infty \ge \underline{\alpha}$ by Definition 8.1. □

8.2.6. Governance Interpretation: Feasibility Tests and Planning

Feasibility test.

The inequality (69) gives a certificate-level feasibility rule: given an estimated long-run disturbance density $B$ and a sustainable dissipation margin $\underline{\alpha}$, the organisation cannot drive $R$ below approximately $\frac{B}{2\underline{\alpha}}$ without changing either $B$ or $\underline{\alpha}$. This is a formal barrier against unrealistic plans, and it matches the human-centred observation that capacity constraints (staffing, workflow latency) limit how quickly remediation can be executed in practice [5,6].

How to improve the floor.

There are only two levers:

(i) increase sustainable dissipation $\underline{\alpha}$ (process improvement, automation, staffing, reduced latency);

(ii) reduce disturbance density $B$ (attack surface reduction, fewer incidents, better upstream quality control).

This is the DSFL meaning of human-centred design: interventions must be evaluated by whether they change $\underline{\alpha}$ or $B$, and by whether those changes are sustainable [6].

Connection to ISS thinking.

Theorem 8.1 is an input-to-state stability style statement: the state (residual) is ultimately bounded by a function of the input magnitude (forcing density) [18,19]. Here the novelty is that the input magnitude $B$ is explicitly interpreted as a human-centred workload/incident intensity measurable from operational data.

**Remark 8.1** (What this theorem does *not* claim)**.** *The theorem does not claim that residuals always converge to zero in realistic environments. Instead, it separates two questions cleanly: (i)* how fast *the system recovers from spikes (governed by* $\underline{\alpha}$*), and (ii)* how low *it can be driven in the presence of persistent forcing (governed by* $B/(2\underline{\alpha})$*). This separation is essential for honest governance claims [18,19].*

 doi:10.20944/preprints202601.0050.v1



## 9. Certificates Under Change

Security guarantees are only useful if they remain meaningful under change. Real deployments evolve along multiple axes: tool versions, policy templates, monitoring pipelines, data schemas, model weights, retrieval indices, and access-control rules. A certificate that does not explicitly encode the semantics of change is not a certificate; it is a snapshot. This section formalises *refinement ladders* and proves the key non-regression mechanism: *summable refinement drift* ("Step–6") yields a coherent limiting claim in one frozen measurement geometry.

Mathematically, the core result is a Banach-valued absolute continuity / bounded variation argument: an $L^1$ bound on a scale derivative implies summable increments and hence Cauchy coherence of the refinement sequence [20]. We additionally state a projective-limit existence theorem under contractive comparators (a standard inverse-limit construction) and interpret the failure modes when Step–6 fails.

### *9.1. Refinement Ladders and Upgrades*

#### 9.1.1. What Counts as "Refinement" in Human-Centered Security

In human-centered cybersecurity, refinement is not only "more resolution" in a numerical sense. It includes any change that alters the representation of the system and therefore can change the meaning of a metric:

- a scanner upgrade that changes vulnerability definitions or severity scoring,
- a policy revision that changes what is counted as compliant,
- a new telemetry pipeline or log schema (instrumentation change),
- a model update in an AI-enabled system (weights, prompt templates, guardrails),
- a retrieval/index update in RAG systems (corpus snapshot changes).

All such changes are treated as movement along a *ladder*: a sequence of representations intended to approximate "the same underlying claim" more faithfully, or at least to remain comparable.

#### 9.1.2. Scale Comparators: Cocycle and Nonexpansivity

To compare defect representatives across refinement levels, we introduce *projection/extension operators* (scale comparators)

$$\mathcal{P}_{s\to s'} : \mathcal{X}_s \to \mathcal{X}_{s'}, \qquad s \le s',$$

which transport objects from a coarser scale $s$ to a finer scale $s'$ in the same frozen measurement geometry.

**Assumption 9.1** (Comparator cocycle and nonexpansivity)**.** *For all $0 \le s \le s' \le s''$, the scale comparators satisfy:*

$$\mathcal{P}_{s\to s} = \mathbb{I}_{\mathcal{X}_s}, \qquad \mathcal{P}_{s'\to s''} \circ \mathcal{P}_{s\to s'} = \mathcal{P}_{s\to s''}, \qquad \|\mathcal{P}_{s\to s'}u\|_{\mathcal{X}_{s'}} \le \|u\|_{\mathcal{X}_s} \quad \forall\, u \in \mathcal{X}_s.$$

Interpretation.

The cocycle property ensures consistency of refinement across multiple scales, while nonexpansivity enforces the one-ruler discipline: refinement is not allowed to artificially reduce or inflate defect magnitude. Any semantic change induced by refinement must therefore appear explicitly as a budgeted drift term (cf. Step 6).

Concrete realisation on a discrete ladder.

When the refinement parameter is discretised along a declared ladder $s = s_0 < s_1 < \cdots < s_K \le s'$, the comparator $\mathcal{P}_{s\to s'}$ is defined by composition of adjacent steps:

 doi:10.20944/preprints202601.0050.v1



$$\mathcal{P}_{s\to s'} := \begin{cases} \mathbb{I}_{\mathcal{X}_s}, & s' = s, \\ \mathcal{P}_{s_K\to s_{K+1}} \circ \mathcal{P}_{s_{K-1}\to s_K} \circ \cdots \circ \mathcal{P}_{s\to s_1}, & s < s' \text{ with } \{s_j\} \text{ a declared ladder.} \end{cases} \tag{73}$$

Role in the certificate architecture.

Equation (73) provides the formal mechanism by which guarantees are transported across tool versions, policy revisions, or model upgrades. Together with Assumption 9.1, it ensures that refinement does not silently change the meaning of the residual in the frozen record, enabling non-regression and projective-limit arguments in later sections.

9.1.3. Frozen Target Space and Comparators

Fix a frozen record $\mathfrak{P}$ (Definition 3.3 in Section 3). In particular, fix a defect space $(\mathcal{R}_p, \|\cdot\|_W)$ and an audit window $[t_0, t_1] = [t_0, t_1]$. Define the frozen target space

$$\mathcal{X}_W := L^\infty([t_0, t_1]; \mathcal{R}_p), \qquad \|u\|_{\mathcal{X}_W} := \operatorname*{ess\,sup}_{t\in[t_0,t_1]} \|u(t)\|_W. \tag{74}$$

All refinement statements are expressed in $\mathcal{X}_W$ to enforce window-uniform semantics.

Why $L^\infty([t_0, t_1])$ is the correct choice.

A proof-carrying certificate is about an *episode* or *audit window*. Uniform control over the entire window is therefore the correct verification target; it prevents "passing" by behaving well only at selected timestamps.

To compare representations at different refinement levels, we declare explicit scale comparators.

**Definition 9.1** (Refinement ladder and comparators)**.** *A* refinement ladder *is a sequence (or continuous family) of representations indexed by a scale parameter $s \geq 0$ or a discrete index $K \in \mathbb{N}$. For each pair $s \leq s'$ we declare a bounded linear comparator*

$$\mathcal{P}_{s\to s'} : \mathcal{X}_W \to \mathcal{X}_W, \tag{75}$$

*interpreted as "coarse-to-fine transport" of defect representatives. In the discrete case we write $\mathcal{P}_K^{K+1}$ for the level comparators.*

**Assumption 9.2** (Contractive comparator (one-ruler discipline in scale))**.** *For all $s \leq s'$ and all $u \in \mathcal{X}_W$,*

$$\|\mathcal{P}_{s\to s'} u\|_{\mathcal{X}_W} \leq \|u\|_{\mathcal{X}_W}, \tag{76}$$

*and comparators satisfy the cocycle relations: $\mathcal{P}_{s\to s} = \mathbb{I}$ and $\mathcal{P}_{s'\to s''} \circ \mathcal{P}_{s\to s'} = \mathcal{P}_{s\to s''}$.*

Interpretation.

Assumption 9.2 is the scale-direction analogue of "no moving goalposts": refinement is not allowed to inflate defects in the frozen ruler. If a tool upgrade changes the metric, that change must appear as an explicit scheme budget (see the failure modes below), not as an implicit redefinition of the norm.

*9.2. Summable Refinement Drift (Step–6)*

The central non-regression condition is that refinement drift has finite total variation in the frozen target norm. There are two equivalent formulations: a continuous "scale derivative" bound (Step–6), and a discrete summable increment schedule.

 doi:10.20944/preprints202601.0050.v1



#### 9.2.1. Step–6 as an $L^1$ Scale-Generator Bound

**Definition 9.2** (Step–6 (Banach-valued absolute continuity)). *A scale-indexed defect representative is a map $e : [0,\infty) \to \mathcal{X}_W$. We say that $e$ satisfies* Step–6 *if $e$ is Bochner absolutely continuous and there exists a nonnegative $G \in L^1([0,\infty))$ such that*

$$\|\partial_s e(s)\|_{\mathcal{X}_W} \;\le\; G(s) \qquad \text{for a.e. } s \ge 0. \tag{77}$$

Why Bochner absolute continuity.

$\mathcal{X}_W$ is Banach, so Bochner integration is the correct calculus. It guarantees that $e(s_2) - e(s_1) = \int_{s_1}^{s_2} \partial_s e(s)\, ds$ holds in $\mathcal{X}_W$ and makes summability claims mathematically meaningful [20].

#### 9.2.2. Step–6 Implies Summable Increments

**Theorem 9.1** (Step–6 $\Rightarrow$ summable refinement drift). *Assume Definition 9.2. Let $s_0 < s_1 < s_2 < \cdots$ be any increasing grid with $s_K \to \infty$, and define $e^{(K)} := e(s_K) \in \mathcal{X}_W$. Then*

$$\sum_{K\ge 0} \|e^{(K+1)} - e^{(K)}\|_{\mathcal{X}_W} \;\le\; \int_{s_0}^{\infty} G(s)\, ds \;<\; \infty. \tag{78}$$

*Consequently, $(e^{(K)})_{K\ge 0}$ is Cauchy in $\mathcal{X}_W$ and converges in $\mathcal{X}_W$.*

**Proof.** Bochner absolute continuity gives, in $\mathcal{X}_W$,

$$e(s_{K+1}) - e(s_K) = \int_{s_K}^{s_{K+1}} \partial_s e(s)\, ds. \tag{79}$$

Take norms and apply (77):

$$\|e^{(K+1)} - e^{(K)}\|_{\mathcal{X}_W} \le \int_{s_K}^{s_{K+1}} \|\partial_s e(s)\|_{\mathcal{X}_W}\, ds \le \int_{s_K}^{s_{K+1}} G(s)\, ds. \tag{80}$$

Summing over $K$ telescopes the integrals to $\int_{s_0}^{\infty} G(s)\, ds < \infty$. Completeness of $\mathcal{X}_W$ implies Cauchy convergence. □

Security meaning.

Theorem 9.1 formalises non-regression under upgrades: the total change of the defect representative induced by refinement is finite in the frozen ruler. Thus the meaning of "small defect" does not drift unboundedly across versions.

#### 9.2.3. Projective-Limit Semantics Under Contractive Comparators

Step–6 controls a representative $e(s) \in \mathcal{X}_W$. To connect this to multi-version deployments where each version has its own representation, we use contractive comparators.

**Theorem 9.2** (Coherent limit under summable comparator mismatch). *Assume Assumption 9.2. Let $u^{(K)} \in \mathcal{X}_W$ be a sequence of level-K representatives and assume a summable schedule $(\delta_K)_{K\ge 0}$ such that*

$$\|\mathcal{P}_K^{K+1} u^{(K)} - u^{(K+1)}\|_{\mathcal{X}_W} \;\le\; \delta_K, \qquad \sum_{K\ge 0} \delta_K < \infty. \tag{81}$$

*Then for each fixed $K$, the transported sequence $(\mathcal{P}_K^L u^{(L)})_{L\ge K}$ is Cauchy in $\mathcal{X}_W$ and admits a unique limit $u^{(\infty,K)} \in \mathcal{X}_W$ satisfying the compatibility relation $\mathcal{P}_K^{K+1} u^{(\infty,K)} = u^{(\infty,K+1)}$. Moreover, the explicit tail bound holds:*

$$\|u^{(\infty,K)} - u^{(K)}\|_{\mathcal{X}_W} \;\le\; \sum_{j=K}^{\infty} \delta_j. \tag{82}$$

 doi:10.20944/preprints202601.0050.v1



**Proof.** Fix $K$ and let $L' > L \geq K$. Using the cocycle identity and telescoping,

$$\mathcal{P}_K^{L'} u^{(L')} - \mathcal{P}_K^{L} u^{(L)} = \sum_{j=L}^{L'-1} \mathcal{P}_K^{j+1} \Big( u^{(j+1)} - \mathcal{P}_j^{j+1} u^{(j)} \Big). \tag{83}$$

Take $\mathcal{X}_W$-norms and use contractivity (76):

$$\|\mathcal{P}_K^{L'} u^{(L')} - \mathcal{P}_K^{L} u^{(L)}\|_{\mathcal{X}_W} \leq \sum_{j=L}^{L'-1} \|u^{(j+1)} - \mathcal{P}_j^{j+1} u^{(j)}\|_{\mathcal{X}_W} \leq \sum_{j=L}^{L'-1} \delta_j. \tag{84}$$

Since $\sum \delta_j < \infty$, the tail goes to 0 as $L \to \infty$, hence the sequence is Cauchy and converges in the complete space $\mathcal{X}_W$. The compatibility relation follows by continuity and cocycle identities, and (82) follows by letting $L' \to \infty$ in the above bound. □

Operational reading.

Theorem 9.2 is the formal meaning of a refinement-invariant claim: each version can ship a finite "mismatch budget" $\delta_K$ and a comparator map. If the mismatch budgets are summable, then the versions are coherent and converge to a well-defined limiting representative. This is the mathematical core of proof-carrying guarantees under change, in the spirit of shipping checkable evidence rather than relying on informal continuity claims [1,2].

### *9.3. Failure Modes When Step–6 Fails*

When Step–6 fails, it is not merely a technicality: it has direct human-centred consequences. Guarantees become incomparable across upgrades, and observed "improvements" may be measurement artifacts.

#### 9.3.1. Failure Mode 1: Metric Drift by Instrumentation

If the observation map obs changes without being included in the frozen record, then the computed defect representative changes in a way that is not typed back to the ruler. This violates the contractive comparator assumption and produces apparent improvements/regressions that are not real. Mathematically, this appears as a non-summable schedule $\delta_K$ in (81).

#### 9.3.2. Failure Mode 2: Scheme Freedom not Controlled (Policy Semantics Drift)

Policy revisions and rule changes are unavoidable. They are admissible only if their induced defect shifts are either absorbed beyond some resolution threshold or budgeted by a summable schedule (scheme control obligation). If such control is absent, Step–6 fails because the defect representative changes by non-summable increments.

#### 9.3.3. Failure Mode 3: Noncontractive Comparators (Moving Goalposts)

If $\mathcal{P}_K^{K+1}$ is not contractive in the frozen ruler, then refinement can increase defects. In practice this corresponds to a measurement "zoom" that magnifies deviations. Mathematically, the telescoping/Cauchy arguments break because (76) fails. This is exactly the "moving goalposts" pathology formalised in Proposition 3.2.

#### 9.3.4. Failure Mode 4: Non-Normal Transients Misdiagnosed as Regression

Even if a sequence is stable in the long run, non-normal operators can exhibit transient growth that is sensitive to refinement and discretization choices. If the analysis assumes selfadjoint/symmetric behaviour, then refinement can produce apparent regressions that are actually non-normal transients. This failure mode motivates treating non-normality explicitly in the numerical audit layer rather than inferring stability from eigenvalues alone [4,26].

 doi:10.20944/preprints202601.0050.v1



**Remark 9.1** (Certificate diagnostics for Step–6 failure)**.** *A proof-carrying workflow makes Step–6 failure actionable:*

(i) *if $\sum_K \delta_K$ is large or diverging, the upgrade path is not coherent in the frozen record;*

(ii) *the tail bound (82) quantifies how far the current version may be from a stable limiting claim;*

(iii) *the remedy is explicit: either refine comparators, freeze the observation map, or treat policy changes as scheme budgets.*

*Thus Step–6 is not a metaphysical requirement; it is the precise mathematical meaning of "guarantees survive upgrades."*

## 10. Numerical Audit Architecture

This section specifies the proof-carrying *numerical* layer of the framework. The theoretical results of Sections 3–9 reduce security assurance to a small number of inequality obligations in a frozen record. A numerical audit does not replace proofs; it provides a mechanically checkable artifact that (i) instantiates constants in those inequalities for a given deployment episode or regime window and (ii) produces PASS/FAIL decisions with explicit slack. The central design principle is *separation of concerns*: the bundle declares a frozen measurement geometry and the list of inequalities to check, while the checker verifies them cheaply at deployment time, in the spirit of proof-carrying code [2] and runtime verification [1].

We emphasise three audit requirements that recur across applications:

(i) **Typing:** all computations occur in the *declared* ruler and neutral convention (no moving goalposts), which is the fixed-norm discipline of stability/semigroup theory in certificate form [3,4];

(ii) **Scalability:** large systems must be reducible to block-structured subproblems so that local checks can be composed with explicit coupling budgets, as in small-gain and positive-systems reasoning [16,17];

(iii) **Correct spectral semantics:** symmetric/selfadjoint versus non-normal cases require different diagnostics because eigenvalues alone need not control transient amplification [4,26].

These principles ensure that numerical results are interpretable by humans and robust under upgrades.

### *10.1. Certificate Bundle Contents*

#### 10.1.1. Bundle as a Proof-Carrying Artifact

A numerical audit produces a finite file—the *bundle*—that can be shipped from producer to consumer. The consumer runs a checker that returns PASS/FAIL and slack values. This is the same architecture as proof-carrying code: ship evidence, verify cheaply [2]. In security settings, the verified object is the episode trace and its derived ledgers rather than source code, matching the runtime-verification emphasis on monitorable properties over traces [1].

**Definition 10.1** (Certificate bundle (numerical audit))**.** *A* certificate bundle *is a tuple*

$$\mathsf{B} = (\mathsf{Meta},\ \mathfrak{P},\ \mathsf{LedgerSpec},\ \mathsf{Const},\ \mathsf{Trace},\ \mathsf{Witness}), \tag{85}$$

*with:*

(i) ***Metadata*** Meta**.** *Version identifiers for tools/policies/models, hashes of configuration files, and time-window descriptors.*

(ii) ***Frozen record*** $\mathfrak{P}$**.** *The declared measurement geometry (ruler, neutrals, observation map, ladder), as in Definition 3.3; this is the typed-data discipline required for meaningful uniformity claims [3,4].*

(iii) ***Ledger specification*** LedgerSpec**.** *Definitions of defect extractors, channel rulers, and aggregation weights producing $R_{\mathrm{tot}} = w^\top r$.*

 doi:10.20944/preprints202601.0050.v1



(iv) ***Constants and budgets*** Const. *Numerical values for verified margins and budgets, e.g.* $\alpha$, $\|\tau_{\mathrm{tot}}\|_{L^1}$, $(\delta_K)$, $(c_{\mathrm{dict}}, C_{\mathrm{dict}})$.

(v) ***Trace*** Trace. *Committed logs/telemetry sufficient to recompute audited ledgers; may include integrity commitments (hashes) if required.*

(vi) ***Witnesses*** Witness. *Auxiliary data that makes checking cheap: e.g. LP witnesses* $w \gg 0$, *gap fit summaries, block decomposition indices, and resampling robustness summaries.*

Why this is minimal.

Each component is required to prevent a common failure mode: without $\mathfrak{P}$ the norm can drift (ill-posed decay comparisons); without Trace the claim is not auditable; without Const the claim is not quantitative; without Witness the checker may be forced to recompute expensive objects. This is precisely the producer/consumer separation principle underlying proof-carrying artifacts [2] and runtime verification [1].

10.1.2. Checker Interface and PASS/FAIL Semantics

The checker implements a finite family of inequality checks in the frozen record:

$$\mathsf{PASS} \quad \Longleftrightarrow \quad \bigwedge_{i=1}^{N_{\mathrm{check}}} \Big[ \mathsf{Check}_i(\mathsf{B}) = \mathsf{PASS} \Big]. \tag{86}$$

A check returns PASS if the corresponding inequality holds on the declared window and declared ladder prefix, with slack recorded explicitly. This makes verification cheap and mechanical, as in proof-carrying systems [2].

**Definition 10.2** (Slack). *For an inequality of the form* $LHS \le RHS$, *the* slack *is defined as*

$$\mathrm{slack} := RHS - LHS. \tag{87}$$

*A check is PASS if* slack $\ge 0$ *(within declared numerical tolerance).*

Human-centred meaning.

Slack values are not cosmetic: they quantify how close the system is to failing the certificate. They support governance: if slack is shrinking across upgrades, the system is approaching a regime boundary even if it still passes.

*10.2. Block-Structured Operator Avatars*

10.2.1. Why Block Structure Is the Scalable Verification Primitive

Many certification targets involve linearised dynamics, Jacobians, or generators that are too large to analyse monolithically. However, real systems often decompose into weakly coupled modules (teams, services, subsystems) or into repeated components (e.g. per-policy rule, per-tool wrapper, per-channel monitor). Numerically, this manifests as block structure: the operator is block-diagonal or block-sparse after appropriate ordering.

The audit architecture therefore prioritises blockwise verification: verify inequalities on small blocks and compose the result via explicit coupling budgets. This mirrors the proof logic of positive systems and small-gain theorems, where local stability plus bounded couplings yields a global certificate [16–19].

 doi:10.20944/preprints202601.0050.v1



10.2.2. Formal Block Decomposition and Audit Reduction

**Definition 10.3** (Block-structured avatar)**.** *A sparse matrix $A \in \mathbb{R}^{N\times N}$ is a* block-structured avatar *if there exists a permutation matrix $P$ and block sizes $(n_1, \dots, n_B)$ with $\sum_{b=1}^{B} n_b = N$ such that*

$$PAP^{\top} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1B} \\ A_{21} & A_{22} & \cdots & A_{2B} \\ \vdots & \vdots & \ddots & \vdots \\ A_{B1} & A_{B2} & \cdots & A_{BB} \end{pmatrix}, \tag{88}$$

*where each diagonal block $A_{bb} \in \mathbb{R}^{n_b\times n_b}$ is "small" and off-diagonal blocks are sparse or norm-bounded by explicit coupling constants.*

**Remark 10.1** (Interpretation of blocks)**.** *Blocks may represent modular subsystems (services), risk channels, policy modules, or repeated units (e.g. per-account, per-region, per-tool). The point of the definition is not the semantics but the audit consequence: once blocks are small, one can compute spectral or norm bounds per block and then close the full system by coupling budgets, consistent with small-gain/positive-systems compositionality [16,17].*

10.2.3. Audit Reduction: Blockwise Constants with Coupling Schedules

The certificate targets typically require constants of the form

$$\gamma \text{ (gap)}, \qquad a_1, a_2 \text{ (norm equivalence)}, \qquad \kappa \text{ (coupling)}. \tag{89}$$

In block structure, these are extracted as:

(i) **Diagonal blocks:** compute blockwise gaps and coercivity constants;

(ii) **Off-diagonal blocks:** compute norm bounds that become coupling coefficients in the Metzler closure;

(iii) **Ladder upgrades:** compute blockwise increments that must be summable (Step–6).

This reduction is what makes a proof-carrying audit scalable [16,17].

*10.3. Spectrum, Gap, and Non-Normality Audits*

10.3.1. Two Spectral Regimes: Symmetric Versus Non-Normal

A numerical audit must not silently assume selfadjointness. The correct diagnostic depends on whether the generator is symmetric (or symmetrizable) in the declared ruler. This is not a technical preference: it changes what a "gap" means.

Symmetric/symmetrizable case.

If $A$ is symmetric (or symmetrizable) in the declared inner product, then its spectrum is real and a spectral gap yields a sharp exponential decay statement for the associated semigroup [3,4]. In this case, a gap can be audited by computing extremal eigenvalues of the symmetric part in the correct ruler.

Non-normal case.

If $A$ is non-normal, eigenvalues alone do not control transient behaviour: $\|e^{tA}\|$ can grow transiently even when all eigenvalues have negative real part. In this regime, resolvent norms and pseudospectra are the correct diagnostic objects [4,26]. Humanly, this corresponds to "spikes" or "alert storms": temporary amplification that is not predicted by steady-state eigenvalue intuition.

10.3.2. Audit Targets: What Is Computed

A minimal numerical audit for a frozen record computes:

 doi:10.20944/preprints202601.0050.v1



(A) Norm matching constants.

Given two quadratic forms (e.g. a surrogate energy $H$ and the frozen ruler $W$), compute $a_1, a_2$ such that

$$a_1\langle u, Hu\rangle \le \langle u, Wu\rangle \le a_2\langle u, Hu\rangle \quad \text{on the neutral-free subspace.} \tag{90}$$

This is a generalized eigenvalue problem and is essential to avoid hidden norm drift; it is the finite-dimensional analogue of energy equivalence required to transfer stability statements between norms [3,4].

(B) Gap estimate.

Estimate a decay/gap parameter $\gamma$ either by: (i) direct eigenvalue computation in the symmetric case, or (ii) a log-slope fit of $\log \|e^{tA}u_0\|_W$ on a tail window, with conservative quantile selection to avoid transient growth. The need to treat transients conservatively in non-normal settings is standard in pseudospectral analysis [26].

(C) Coupling budgets.

Compute coupling constants $\kappa$ by norm bounds on off-diagonal blocks or by model-free empirical ratios (e.g. $\kappa_{emp} = \max |J|/\sqrt{R^a R^b}$), producing auditable coupling coefficients for Metzler closure, as in positive-systems/small-gain reasoning [16–19].

10.3.3. Certificate-Style Interpretation (Human Meaning)

Numerical outputs are interpreted in certificate form:

- a gap estimate $\gamma_{\text{cert}}$ is accepted only as a conservative bound (e.g. lower quantile across trials);
- norm matching constants $(a_1, a_2)$ quantify how much sharpness is lost when transferring results between energies, which is unavoidable when stability is a typed fixed-norm statement [3,4];
- non-normality is treated explicitly by reporting transient amplification indicators (e.g. peak-to-tail ratios) rather than hiding it inside averages, consistent with pseudospectral warnings about eigenvalue-only diagnostics [26].

This produces reports that humans can act on: PASS/FAIL plus slack and "why" (which constant is tight).

**Remark 10.2** (Why this is not "just numerics")**.** *The bundle-and-checker architecture is the numerical analogue of proof-carrying guarantees: a producer may be untrusted, so the consumer must be able to verify claims cheaply from declared inputs and logs [2]. The numerical audit therefore does not aim for maximal accuracy; it aims for* verifiable semantics *in a frozen record, and aligns with runtime verification's emphasis on checkable evidence extracted from executed traces [1].*

## 11. Case Study: Audit of a Large Block-Structured System

This section demonstrates the DSFL certificate workflow on a large block-structured avatar. The purpose is not to claim that one numerical run proves a theorem; it is to show that the *typing discipline* (frozen ruler, frozen neutrals, declared ladder) turns otherwise narrative assurances into a *finite list of checkable inequalities* with explicit slack. The background analytic posture is the same as in stability theory: decay is a statement about an operator in a declared norm, and thus constants must be measured in that same norm [3,4]. The DSFL novelty is that the same discipline is applied to coupling budgets and to refinement/ladder coherence, so that "uniform in refinement" and "composable assurance" have invariant meaning.

 doi:10.20944/preprints202601.0050.v1



**Scope wall (what this case study does and does not prove)**

**What it does.** It specifies an auditable record, runs the four checks (assembly, gap, coupling, summability), and outputs a certificate decision (PASS/FAIL) with explicit slack variables.

**What it does not do.** It does not substitute numerics for proofs of underlying physics or organisational causality. It provides a proof-carrying *pre-theorem* audit whose output can later be upgraded by analytic imports.

### *11.1. Assembly Audit*

Dimension and sparsity checks validate structural assumptions.

#### 11.1.1. Frozen Record for the Avatar

We fix a frozen numerical record (the audit pack):

$$\mathsf{P} := (A,\ W,\ \Pi_{\perp},\ H_{\mathrm{geo}},\ \text{(optional ladder projections)}), \tag{91}$$

where $A$ is the assembled operator (generator or defect operator), $W \succ 0$ is the frozen ruler, $\Pi_{\perp}$ is the frozen neutral projector, and $H_{\mathrm{geo}} \succ 0$ is an optional surrogate "geometric energy" used for norm matching. All subsequent computations are performed *only* from $\mathsf{P}$.

**Remark 11.1** (Typing discipline)**.** *The pack is the object being audited. If $W$ or $\Pi_{\perp}$ are changed after the fact, the audit is invalid. This is the exact analogue of stability theory: changing the norm changes the meaning of decay [3,4].*

#### 11.1.2. Dimension Check (Block Accounting)

Large systems are typically assembled as a sum of structured blocks (e.g. channels, subsystems, or $(\ell, m)$ modes). The first audit is simply that the dimension matches the declared block model.

**Definition 11.1** (Declared block structure)**.** *A declared block structure is a decomposition of degrees of freedom as*

$$N_{\mathrm{tot}} = \sum_{b \in \mathcal{B}} N_b, \tag{92}$$

*with a known map from block indices $b$ to contiguous index ranges, and a declared sparsity pattern per block (e.g. tri-diagonal, banded, or local adjacency).*

**Proposition 11.1** (Dimension and indexing consistency)**.** *Let $A \in \mathbb{R}^{N \times N}$ be the assembled operator and assume a declared block structure $\{N_b\}_{b \in \mathcal{B}}$ with $\sum_b N_b = N_{\mathrm{tot}}$. The assembly passes the dimension audit if and only if*

$$N = N_{\mathrm{tot}} \tag{93}$$

*and all block-index ranges are disjoint and cover $\{1, \ldots, N\}$.*

**Proof.** Immediate from the definitions of a block decomposition and the meaning of the assembled matrix dimension. □

**Explanation.** This is not trivial in practice: many audit failures arise from silent index misalignment (wrong offsets, missing neutrals, duplicated blocks). DSFL makes this a first-class check: if dimension typing fails, no stability statement is meaningful.

#### 11.1.3. Sparsity Audit (Structural Sanity)

The sparsity audit checks whether the number of nonzeros and their pattern match the intended block stencil.

 doi:10.20944/preprints202601.0050.v1



**Definition 11.2** (Sparsity profile). *Let* $\mathrm{nnz}(A)$ *denote the number of nonzero entries. A sparsity profile is a predicted bound (or exact formula) of the form*

$$\mathrm{nnz}(A) \approx \sum_{b \in \mathcal{B}} c_b \, N_b \tag{94}$$

*for constants* $c_b$ *determined by the local stencil (e.g.* $c_b \approx 3$ *for tri-diagonal interior rows plus boundary rows).*

**Proposition 11.2** (Sparsity agreement implies structural plausibility). *If the observed* $\mathrm{nnz}(A)$ *matches the declared sparsity profile up to a declared tolerance, then the assembly is structurally plausible: the operator has the intended locality/band structure. If* $\mathrm{nnz}(A)$ *deviates substantially, the assembly is flagged for indexing/stencil errors.*

**Proof.** The conclusion is by direct comparison of the measured nonzero count to the predicted count from the stencil and block sizes. □

**Explanation.** This audit is the numerical analogue of verifying that an operator is the one you claim to be. It is a necessary precondition for interpreting any spectral or decay estimate.

### *11.2. Gap Estimation*

We present numerical ringdown and decay fitting.

#### 11.2.1. What a "Gap" Means in a Frozen Ruler

A gap is a normed semigroup statement:

$$\| \exp(tL_\perp) \|_{W \to W} \le C \, e^{-\gamma t}, \tag{95}$$

for the projected (neutral-free) generator $L_\perp := \Pi_\perp L \Pi_\perp$. In a finite avatar, this can be probed numerically by time-domain ringdown fits or by conservative dissipativity bounds. The key DSFL rule is that the gap must be measured in the declared ruler $W$, not in an ad hoc energy.

#### 11.2.2. Time-Domain Ringdown Fit

**Definition 11.3** (Ringdown fit protocol (Monte Carlo)). *Fix a time grid* $\{t_j\}_{j=0}^{m}$ *and a "tail" index set* $\{j_0, \dots, m\}$ *(discarding early transients). For trials* $\ell = 1, \dots, N_{\mathrm{trial}}$*, sample* $u_0^{(\ell)} \in \mathrm{Ran}(\Pi_\perp)$ *with* $\|u_0^{(\ell)}\|_W = 1$*, evolve* $u^{(\ell)}(t_j) = \exp\left(t_j L_\perp\right) u_0^{(\ell)}$*, and record* $\rho^{(\ell)}(t_j) = \|u^{(\ell)}(t_j)\|_W$*. Fit* $\log \rho^{(\ell)}(t_j) \approx a_\ell - \widehat{\gamma}_\ell t_j$ *on the tail. Define a conservative certified rate as a lower quantile, e.g.* $\gamma_{\mathrm{cert}} := \mathrm{Quantile}_{0.10}(\widehat{\gamma}_\ell)$.

**Remark 11.2** (Why a quantile is certificate-friendly). *A quantile is conservative under sampling variability and is stable under resampling protocols. It is the correct numerical analogue of a theorem constant: "this rate holds for a large class of initial conditions".*

#### 11.2.3. A Conservative Matrix-Only Proxy (Symmetric-Part Bound)

When $L_\perp$ is explicitly available as a matrix, one can compute a conservative bound from the $W$-symmetric part, as in dissipativity theory [3,4].

**Proposition 11.3** (Dissipativity proxy for the gap). *Let* $\widetilde{L}_\perp := W^{1/2} L_\perp W^{-1/2}$ *and* $S_\perp := \frac{1}{2}(\widetilde{L}_\perp + \widetilde{L}_\perp^\top)$*. If* $\lambda_{\max}(S_\perp) < 0$*, then*

$$\| \exp(tL_\perp) \|_{W \to W} \le e^{\lambda_{\max}(S_\perp) t}, \tag{96}$$

*so* $\gamma_{\mathrm{sym}} := -\lambda_{\max}(S_\perp) > 0$ *is a certified decay-rate proxy.*

**Explanation.** This estimate is conservative but mechanically checkable, aligning with the proof-carrying posture: the checker can verify it without interpreting time-series fits.

### *11.3. Coupling and Summability*

Empirical coupling and ladder summability are audited.

 doi:10.20944/preprints202601.0050.v1



11.3.1. Coupling Audit: Empirical $\kappa$ Constants

For each ordered channel pair $(a \leftarrow b)$, we compute a conservative coupling constant $\kappa(a \leftarrow b)$ from time series and obtain the immediate inequality

$$|J_{a\leftarrow b}(t)| \leq \kappa(a \leftarrow b)\sqrt{R^a(t)R^b(t)} + \varepsilon\kappa(a \leftarrow b), \tag{97}$$

as in Section 7. These constants populate the off-diagonal entries of the Metzler comparison matrix.

11.3.2. Summability Audit: Ladder Increments

If the system is presented along a refinement ladder $K = 0, 1, \ldots$ (discretisation level, code distance, policy refinement), Step 6 coherence is an $\ell^1$-type condition:

$$\delta_K := \|e^{(K)} - \Pi_K^{K+1} e^{(K+1)}\|_{W^{(K)}} \quad \text{with} \quad \sum_K \delta_K < \infty. \tag{98}$$

In a finite audit, we compute partial sums $S_{K_{\max}} = \sum_{K=0}^{K_{\max}} \delta_K$ and fit a tail model to bound the unseen remainder. The certificate record stores both $S_{K_{\max}}$ and the tail bound.

**Remark 11.3** (Why summability is the right notion)**.** *Summability is the scale analogue of an $L^1$ forcing budget in time: it is precisely what makes projective-limit statements stable and prevents "completion" from being a moving-goalpost claim.*

*11.4. Certificate Decision*

The system is declared PASS or FAIL with explicit slack.

11.4.1. Formal PASS/FAIL Rule

Fix declared thresholds in the frozen record:

$$\gamma_{\text{req}} > 0, \qquad \kappa_{\text{req}} \geq 0, \qquad S_{\text{req}} < \infty, \tag{99}$$

representing (respectively) the required gap funding, the maximum allowed couplings (or required Hurwitz margin), and the maximum allowed ladder budget. Compute:

$$\gamma_{\text{cert}} \quad \text{(conservative gap)}, \qquad \kappa_{\text{rob}} \quad \text{(robust couplings)}, \qquad S_{\text{cert}} \quad \text{(summability certificate)}. \tag{100}$$

Then the certificate decision is:

**PASS** if:

$$\gamma_{\text{cert}} \geq \gamma_{\text{req}} \quad \wedge \quad M(\kappa_{\text{rob}}, \gamma_{\text{cert}}) \text{ is Hurwitz with margin} \quad \wedge \quad S_{\text{cert}} \leq S_{\text{req}}. \tag{101}$$

Otherwise **FAIL**, together with the first violated inequality and its slack.

11.4.2. Slack Variables (What the Audit Reports)

We report slack values:

$$\text{slack}_\gamma := \gamma_{\text{cert}} - \gamma_{\text{req}}, \qquad \text{slack}_M := \text{Hurwitz margin slack (LP-feasible } \alpha), \qquad \text{slack}_S := S_{\text{req}} - S_{\text{cert}}. \tag{102}$$

Positive slack means the certificate is robust; negative slack means the corresponding brick is unfunded in the frozen record.

**Remark 11.4** (Why this is scientifically useful)**.** *A PASS is a reproducible statement in a fixed geometry; a FAIL is not a dead end but a diagnosis: it identifies whether the failure is typing (assembly/ruler), gap funding, coupling closure, or ladder coherence. This is the DSFL promise: progress is measured by inequalities with slack, not by interpretive narratives.*

 doi:10.20944/preprints202601.0050.v1



## 12. Implications for Human-Centred Cybersecurity

This section explains what the theory implies for human-centred cybersecurity in concrete operational terms. The results of Sections 3–10 are not merely formal: they induce a specific *decision interface* (what humans should look at), a specific *governance interface* (what organisations can certify), and a specific *engineering interface* for adaptive AI systems (what must be frozen and what may change). We present each implication as a theorem-shaped statement whenever possible, making explicit what is guaranteed and under what hypotheses. The empirical motivation is consistent with the usable-security literature: security outcomes are shaped by bounded attention, work practices, and incentives, so guarantees must be legible and operable under these constraints [5–9].

### *12.1. Decision Support*

#### 12.1.1. Cognitive Load as a Formal Failure Mode

A central human-centred failure mode is cognitive overload: too many metrics with drifting semantics yield unreliable action. This is not merely a UX issue; it is a verification problem. If the measurement geometry drifts, the organisation cannot tell whether the system is improving or whether the dashboard changed. The DSFL certificate layer addresses this by collapsing multi-channel risk into one scalar ledger with a single "clock" and explicit budgets, aligning with the stability-theoretic discipline that decay is meaningful only in a fixed norm [3,4,22]. The human-centred security literature further motivates this collapse: when security tasks compete with productivity, people rationally optimize effort and circumvent burdensome rules; thus "more metrics" is often counterproductive [6–9].

#### 12.1.2. From Multi-Ledger Telemetry to One Decision-Ready Scalar

Let $r(t) \in \mathbb{R}^m_{\geq 0}$ be the vector of channel ledgers (identity, misconfiguration, exfiltration, tool misuse, ...) and assume the one-clock conditions of Section 5 hold, so that there exists $w \gg 0$ and $\alpha > 0$ with

$$R_{\mathrm{tot}}(t) := w^\top r(t) \quad \text{satisfying} \quad \dot{R}_{\mathrm{tot}}(t) \leq -2\alpha R_{\mathrm{tot}}(t) + \tau_{\mathrm{tot}}(t), \qquad \tau_{\mathrm{tot}} \in L^1([t_0, t_1]). \tag{103}$$

The quantities $(\alpha, \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}, R_{\mathrm{tot}}(t_0))$ define a minimal, auditable decision summary, in the same sense that Lyapunov envelopes provide decision-ready bounds without solving the full dynamics [18,19].

**Proposition 12.1** (Decision summary sufficiency)**.** *Assume the one-clock inequality* (32) *on a window* $[t_0, t_1]$. *Then the envelope* (36) *implies that the triple*

$$\left(\alpha,\ \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])},\ R_{\mathrm{tot}}(t_0)\right) \tag{104}$$

*is sufficient to upper-bound the entire risk trajectory* $R_{\mathrm{tot}}(t)$ *on* $[t_0, t_1]$.

**Proof.** The envelope (37) gives, for all $t \in [t_0, t_1]$,

$$R_{\mathrm{tot}}(t) \leq e^{-2\alpha(t-t_0)} R_{\mathrm{tot}}(t_0) + \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}, \tag{105}$$

which depends only on the stated triple. □

Human-centred interpretation.

Instead of asking humans to integrate dozens of drifting plots, DSFL produces:

- a certified improvement margin $\alpha$ ("how fast we recover when not being hit"),
- a disturbance footprint $\|\tau_{\mathrm{tot}}\|_{L^1}$ ("how hard we were hit on this window"),
- a starting residual $R_{\mathrm{tot}}(t_0)$ ("how far we were from target at window start").

 doi:10.20944/preprints202601.0050.v1



This is a provably sufficient summary for envelope-level decision making, which is the level at which governance and risk management typically operate [18,19].

#### 12.1.3. Actionability: What Interventions Can Change

The certificate interface makes intervention effects interpretable: an intervention can only improve assurance by increasing $\alpha$ (stronger sustained dissipation) or reducing $\|\tau_{\mathrm{tot}}\|_{L^1}$ (reducing incident/-forcing footprint). This yields a rigorous form of "actionability" that avoids metric gaming: if a change does not modify these certificate quantities in the frozen record, it does not improve certified security. This framing is compatible with the "compliance budget" perspective in usable security: interventions that exceed feasible effort budgets tend to be circumvented and thus do not improve realised security [8,9].

### *12.2. Governance and Compliance*

#### 12.2.1. From Snapshot Audits to Continuous Assurance

Traditional audits are static: they check whether policies are satisfied at a point in time. However, organisations operate under constant change and constant disturbance. The DSFL view is that compliance is a dynamical residual—a distance to a policy manifold—and assurance is a statement about the evolution of that residual under a remediation process. This matches the engineering logic of stability under forcing, where guarantees are meaningful only when stated in a declared norm and across declared perturbations [3,4].

#### 12.2.2. Compliance as a Residual and Remediation as a Margin

Let $E_{\mathrm{pol}} : X \to \mathcal{R}_p$ be a policy imbalance operator (Section 6), and let $R^{\mathrm{pol}}(t) := \|E_{\mathrm{pol}}(x(t))\|_W^2$. Assume a remediation margin inequality

$$\dot{R}^{\mathrm{pol}}(t) \le -2\lambda_{\mathrm{pol}} R^{\mathrm{pol}}(t) + \tau_{\mathrm{pol}}(t), \qquad \tau_{\mathrm{pol}} \in L^1([t_0, t_1]). \tag{106}$$

Then compliance becomes a continuous assurance claim: not merely that $R^{\mathrm{pol}}(t)$ is small at an audit date, but that its trajectory is bounded on every window by an explicit envelope, using the inhomogeneous Grönwall mechanism [18,23,24].

**Theorem 12.1** (Governance-grade compliance envelope)**.** *Under the remediation margin inequality above, the compliance residual satisfies*

$$R^{\mathrm{pol}}(t) \le e^{-2\lambda_{\mathrm{pol}}(t-t_0)} R^{\mathrm{pol}}(t_0) + \|\tau_{\mathrm{pol}}\|_{L^1([t_0,t_1])}. \tag{107}$$

**Proof.** This is the tail-robust envelope theorem (inhomogeneous Grönwall) applied to $R^{\mathrm{pol}}$ [23,24]. □

Governance interpretation.

The envelope yields a contract between organisation and auditor: the organisation declares the frozen record, the policy residual, the margin $\lambda_{\mathrm{pol}}$, and the disturbance budget. An auditor checks the inequality on logged evidence and obtains PASS/FAIL plus slack. This is structurally aligned with assurance-case thinking and proof-carrying verification: claims are shipped with evidence and verified independently [1,2].

#### 12.2.3. Non-Regression Under Change as a Governance Requirement

Compliance programmes fail when policy or tooling changes silently shift the meaning of "compliant." The refinement-ladder theory (Section 9) makes non-regression explicit: upgrades must come with a summable drift schedule (Step–6) and contractive comparators in the frozen record. Thus governance becomes "continuous and version-stable": audits remain meaningful across tool and policy updates. This requirement mirrors the proof-carrying logic: what changes is allowed, but only under checkable evidence and bounded drift [1,2].

 doi:10.20944/preprints202601.0050.v1



**Remark 12.1** (Why this is a governance advance)**.** *Static audits answer "are we compliant now?". DSFL audits answer "are we converging to compliance at a certified rate under declared disturbance budgets, and does this claim survive upgrades?". The latter is the more operationally relevant question for organisations under continuous change.*

### *12.3. AI Agents and Adaptive Systems*

#### 12.3.1. Why Tool-Using Agents Require Proof-Carrying Semantics

Tool-using AI agents differ from classical software components: they act over multi-step trajectories, interact with external state, and evolve across versions (prompts, tools, retrieval). Empirical benchmark scores do not provide deployment-time guarantees and are not stable under upgrade. Thus the natural assurance object is the executed trace together with a proof-carrying bundle, as in Section 10. The engineering rationale follows the PCC/RV paradigm: shrink the trusted computing base to a checker and make evidence portable [1,2].

#### 12.3.2. Agent Safety as a Ledger Vector

In an agent system, define channel ledgers for: policy violations, prompt injection susceptibility, privacy leakage, unsafe tool calls, and drift under upgrades. Under cooperative coupling (Metzler) and explicit budgets, the one-clock theorem yields a single scalar risk residual governed by an auditable envelope. This makes agent safety compositional: adding a new tool adds a new channel ledger and new coupling coefficients, but does not change the certificate spine. Technically, this is the same positive-systems closure used for multi-channel risk in control and stability theory [16–19].

#### 12.3.3. Proof-Carrying Guarantees Under Model and Tool Upgrades

Adaptive systems change by design. The Step–6 coherence condition ensures that upgrades (new model weights, new retrieval index, new policy rules) do not cause unbounded drift in the defect representation in the frozen ruler. Thus DSFL provides a principled non-regression requirement for AI deployment: upgrades are admissible only if their induced drift is summably controlled and the bundle remains checkable [1,2].

**Proposition 12.2** (Upgrade-stable agent safety (certificate form))**.** *Assume an agent system ships a certificate bundle $\mathsf{B}_K$ at version $K$ with: (i) a frozen record $\mathfrak{P}$, (ii) a one-clock margin $\alpha > 0$ and budget $\tau_{\mathrm{tot}} \in L^1([t_0, t_1])$, and (iii) a Step–6 summable drift schedule $(\delta_K)$ for version changes. Then the safety envelope remains comparable across versions and converges to a well-defined limiting claim in the frozen record (Theorem 9.2).*

**Proof.** Immediate from the projective-limit theorem and Step–6 summability results in Section 9. □

Human-centred meaning.

For AI-enabled systems, DSFL shifts assurance from "trust the model card" to "verify the bundle." This is directly actionable for humans: a deployer can require a PASS bundle before enabling a tool, and a regulator can audit the same inequalities from the same frozen record.

**Remark 12.2** (Relationship to formal verification)**.** *The DSFL approach does not claim full semantic correctness of an AI model. It verifies a bounded-risk envelope in a declared semantics. This is closer in spirit to proof-carrying code and runtime verification than to full program correctness [1,2].*

## 13. Limitations and Open Problems

This section records the boundaries of the present contribution and the remaining research tasks. Because the paper is built as a proof-carrying certificate layer, the limitations are not vague: they correspond to explicit places where (i) typing data must be chosen, (ii) empirical observability constrains what can be certified, and (iii) optimisation problems (best rulers, best residuals) remain open. Where possible we phrase open problems as theorem-shaped questions to avoid narrative

 doi:10.20944/preprints202601.0050.v1



ambiguity, consistent with the proof-carrying posture of shipping checkable obligations rather than narrative assurances [1,2].

### *13.1. What Is not Automated*

Choosing rulers and residuals remains a design task.

#### 13.1.1. Rulers Are Part of the Theorem Data

A central DSFL principle is that stability and decay are statements in a fixed normed geometry: in semigroup language, contractivity and exponential decay are typed statements about a generator on a specified normed space [3,4]. Therefore, the ruler $W$ cannot be "learned away" without changing the meaning of the claim. Declaring $W$ is analogous to declaring an energy norm in PDE stability and control: it is part of the theorem statement [3,4,18].

**Remark 13.1** (Design task, not an algorithm)**.** *In the present framework, selecting $W$ is a modelling/design decision that encodes what the organisation means by "distance from policy equilibrium." DSFL constrains and audits that choice, but it does not uniquely determine it without additional domain commitments (e.g. which risks count as neutral, which trade-offs are acceptable), exactly because changing the norm changes the typed meaning of a decay claim unless explicit equivalence constants are established [3,4].*

#### 13.1.2. Residual Definitions Encode Semantics

Similarly, the defect map (policy–practice imbalance operator) is a semantic object: it specifies what counts as a violation, what is neutral, and what evidence is admissible. Different organisations may legitimately choose different imbalance operators even for the same high-level policy language. DSFL does not eliminate that fact; it makes it explicit and auditable, in the same sense that proof-carrying verification requires an explicit statement of the semantics being checked [1,2].

**Proposition 13.1** (Non-uniqueness without additional axioms)**.** *Let $\mathcal{H}$ be a fixed practice space and let $\mathcal{N} \subset \mathcal{H}$ be a fixed neutral subspace. If no additional constraints are imposed (e.g. operational invariances, monotonicity under coarse-graining, or compatibility with a declared observation dictionary), then there exist infinitely many inequivalent ruler/residual pairs $(W, E_{\mathrm{pol}})$ that yield different numerical compliance residuals on the same underlying data.*

**Proof.** Choose any two non-equivalent SPD operators $W_1, W_2$ on $\mathcal{N}^{\perp}$ and the same neutral-free defect $e = \Pi_{\perp}\tilde{e}$. Then $\|e\|_{W_1}$ and $\|e\|_{W_2}$ define different residual magnitudes unless explicit equivalence constants are imposed. This is the fixed-norm typing phenomenon that underlies norm-transfer requirements in semigroup stability [3,4]. □

**Explanation.** This proposition is the mathematical reason that the paper cannot promise full automation: rulers and residuals are theorem data, and non-equivalent choices encode different meanings.

### *13.2. Empirical Calibration Challenges*

Data quality and observability limit guarantees.

#### 13.2.1. Observability Limits Are Structural, not Incidental

The DSFL certificate is only as good as the observation map $\mathcal{O}_{\mathcal{C}}$ that turns real operations into measured defects. In practice, security-relevant state is partially observed: logs are missing, telemetry is delayed, and organisational behaviour is not fully instrumented. This is not merely an engineering inconvenience; it is a limitation on what can be certified. In the DSFL formalism, this appears as:

$$e_{\mathrm{obs}}(t) = \mathcal{O}_{\mathcal{C}} e(t), \qquad R_{\mathrm{obs}}(t) = \|e_{\mathrm{obs}}(t)\|^2_{W_{\mathrm{obs}}}, \tag{108}$$

together with an amplification/contractivity bound needed to relate observed residuals to global residuals. This mirrors the general runtime-verification constraint: only monitorable properties over available traces can be checked [1].

 doi:10.20944/preprints202601.0050.v1



**Remark 13.2** (Cone-like limitations in cyber settings)**.** *Although the motivating language of "cones" comes from relativity, the same structure appears in cyber settings: an SOC only sees events that reach its sensors, and sensor coverage changes over time. DSFL treats this as an explicit map with a typing budget, rather than assuming full observability, consistent with the "observable semantics" emphasis in monitor-based verification [1].*

#### 13.2.2. Calibration Drift and Dataset Shift

Empirical constants such as coupling budgets $\kappa_{\text{emp}}$ and inferred rates depend on the data distribution. When organisational processes, tooling, or adversary behaviour change, the inferred constants can drift. DSFL does not hide this; it forces a re-audit in the frozen record or an explicit update to the record. From a control viewpoint, this is exactly the distinction between a proved ISS-style bound and a fitted model whose parameters may change with regime [18,19].

**Proposition 13.2** (Robustness requires worst-case aggregation over protocol perturbations)**.** *Let $\kappa_{\text{emp}}^{(g,s)}$ be an empirical coupling estimate computed under protocol choices $g$ and resampling seed $s$. A certificate-grade coupling constant must be taken as a conservative aggregation*

$$\kappa_{\text{rob}} := \max_{g\in\mathcal{G}} \max_{s\in\mathcal{S}} \kappa_{\text{emp}}^{(g,s)}, \tag{109}$$

*so that the resulting inequality holds uniformly over the declared perturbation family.*

**Proof.** Each $\kappa_{\text{emp}}^{(g,s)}$ yields a bound of the form $|J| \le \kappa_{\text{emp}}^{(g,s)} \sqrt{R^a R^b} + \varepsilon \kappa_{\text{emp}}^{(g,s)}$ on the corresponding time series. Taking the maximum preserves the bound across all protocols/seeds. This is the empirical analogue of uniformity/margins in small-gain style guarantees [18,19]. □

**Explanation.** This is the empirical analogue of uniformity in $K$ for refinement ladders: robust certification requires conservative envelopes, not point estimates.

### *13.3. Open Theoretical Questions*

Optimal rulers and learning residuals remain open.

#### 13.3.1. Optimal Ruler Selection Under Composability Constraints

The most important open mathematical question is: how to select a ruler $W$ that is simultaneously:

- **(O1)** **Operationally meaningful:** encodes the organisation's notion of risk and neutrality;
- **(O2)** **Stable under change:** yields explicit norm-equivalence constants when tooling changes;
- **(O3)** **Composable:** admits tight coupling bounds and a large Hurwitz margin for the induced Metzler system.

This is an instance of Lyapunov-function selection under structural constraints: one seeks a certificate functional that is both meaningful and yields strong margins under coupling [16–19].

**Problem 13.1** (Optimal ruler under one-clock margin)**.** *Fix a family of admissible defect operators $\{E_a\}_{a\in\mathcal{A}}$ and a neutral convention. Over the class of SPD rulers $W$ satisfying declared invariances and normalisations, find a ruler that maximises the certified one-clock margin $\lambda_{\text{eff}}(W)$ (equivalently minimises the spectral abscissa of the induced Metzler comparison matrix subject to budget constraints).*

**Explanation.** This is a genuine optimisation problem constrained by stability theory: different rulers change both diagonal gaps and off-diagonal couplings. The functional-analytic literature suggests that "best" energies are often problem-adapted Lyapunov functionals, and the positive-systems literature makes explicit how diagonal margins and couplings trade off in Metzler/Hurwitz closure [3,4,16–19].

#### 13.3.2. Learning Residuals Without Losing Typing

It is tempting to "learn the residual" from data (e.g. learn $E_{\text{pol}}$ or $W$ from historical incidents). However, this risks reintroducing moving goalposts: a learned residual can change semantics unless

 doi:10.20944/preprints202601.0050.v1



the learning objective is itself typed and frozen. This tension mirrors the general assurance distinction between (i) checkable semantics and (ii) adaptive predictors whose meaning may drift unless versioned and audited [1,2].

**Problem 13.2** (Learned residuals with frozen semantics)**.** *Design a learning procedure that proposes a defect map $\widehat{E}$ and ruler $\widehat{W}$ while guaranteeing:*

(i) *the proposal is equivalent (with explicit constants) to the declared semantics in the frozen record, or else is declared as a new record;*

(ii) *the induced coupling bounds remain Metzler-typed and preserve (or improve) the Hurwitz margin;*

(iii) *the resulting certificate remains auditable by a third party.*

**Explanation.** A DSFL-compatible learning rule must output not only a predictor, but also a proof-carrying translation back to the frozen record, so that the checker verifies a stable semantics rather than a drifting score [1,2].

13.3.3. Tightness of the Metzler Reduction

The Metzler reduction uses inequalities such as $\sqrt{xy} \le \frac{1}{2}(x+y)$, which can be conservative.

**Problem 13.3** (Tight one-clock bounds for coupled nonnegative ledgers)**.** *Given a coupled ledger system with structured cross terms (e.g. bilinear injections), characterise conditions under which one can produce a tighter scalar ledger than the generic Metzler bound, while preserving auditable nonnegativity and frozen-ruler typing.*

**Explanation.** This problem matters because the certified rate $\lambda_{\text{eff}}$ is the headline assurance number. Improving tightness can turn a marginal FAIL into a PASS without changing the underlying system—only by using a sharper but still auditable certificate, in the spirit of constructing tighter Lyapunov/ISS estimates and less conservative small-gain bounds [16–19].

**Bottom line (limitations as actionable research tasks)**

The DSFL framework is strongest exactly where it is most constrained: it refuses to hide semantics inside changing metrics. What is not automated (ruler/residual design), what is empirically limited (observability and drift), and what remains open (optimal rulers, typed learning, tight scalarisation) are not hand-waving limitations; they are explicit, theorem-shaped problems that define a clear research agenda.

## 14. Conclusion

This paper has advanced a verification-theoretic reformulation of cybersecurity: security is treated as a *provable contraction* of a single scalar residual in a *frozen measurement geometry*, up to explicit, auditable disturbance budgets. The result is a proof-carrying notion of assurance that remains meaningful under change, supports human decision-making, and exposes hidden failure modes (metric drift, non-normal transients, infeasible improvement plans) as explicit violations of checkable inequalities.

### *14.1. What Has Been Proved*

The central mathematical contribution is a certificate spine with three kernel implications:

14.1.1. One Residual and One Clock from Multi-Channel Ledgers

Starting from a vector of nonnegative channel ledgers $r(t) \in \mathbb{R}^m_{\ge 0}$ satisfying a cooperative comparison inequality $\dot{r} \le Mr + \tau$ with $M$ Metzler–Hurwitz and $\tau \in L^1([t_0, t_1])$, we proved that there exists a strictly positive weighting $w \gg 0$ producing a single scalar ledger

$$R_{\text{tot}}(t) = w^\top r(t) \tag{110}$$

 doi:10.20944/preprints202601.0050.v1



that satisfies the one-clock inequality

$$\dot{R}_{\mathrm{tot}}(t) \le -2\alpha R_{\mathrm{tot}}(t) + \tau_{\mathrm{tot}}(t), \qquad \tau_{\mathrm{tot}}(t) = w^{\top}\tau(t) \in L^1([t_0, t_1]). \tag{111}$$

This is the rigorous form of the decision-ready claim: *"one number tracks risk, one number tracks recovery rate, one budget tracks disturbance."* The proof is purely functional-analytic and relies on positive-systems theory and Perron–Frobenius witnesses [17,25,27].

14.1.2. Tail-Robust Envelopes (Incident Spikes Become Budgets)

We derived an explicit tail-robust envelope via the inhomogeneous Grönwall mechanism:

$$R_{\mathrm{tot}}(t) \le e^{-2\alpha(t-t_0)} R_{\mathrm{tot}}(t_0) + \int_{t_0}^{t} e^{-2\alpha(t-s)} \tau_{\mathrm{tot}}(s)\, ds, \tag{112}$$

and hence $R_{\mathrm{tot}}(t) \le e^{-2\alpha(t-t_0)} R_{\mathrm{tot}}(t_0) + \|\tau_{\mathrm{tot}}\|_{L^1([t_0,t_1])}$. This formalises the human-centred interpretation of incidents: large spikes matter only through their integrated footprint, while recovery is governed by the certified margin $\alpha$ [23,24].

14.1.3. Non-Regression Under Upgrades via Summable Refinement Drift

We formalised system evolution (tools, policies, models, monitoring) as a refinement ladder. Under Step–6 (an $L^1$ scale-generator bound) we proved summable refinement drift and projective coherence: refinement does not silently change the meaning of the claim in the same frozen ruler. This is the precise mathematical content of "no moving goalposts under upgrades" and follows from Banach-valued calculus and contractive inverse-limit arguments [3,4,20].

*14.2. What This Reframing Changes for Human-Centred Cybersecurity*

The core practical effect is a shift from *metric pluralism* to *certificate semantics*.

Decision support.

Instead of asking practitioners to reconcile many drifting dashboards, the framework produces a small set of auditable quantities: a certified improvement margin, an explicit disturbance budget, and explicit coupling constants. These are sufficient to upper-bound risk trajectories on declared windows and to compare interventions across versions.

Governance and compliance.

Compliance is reframed as a residual (distance to a policy manifold) and assurance as a decay statement with explicit budgets. Static snapshot audits are replaced by continuous, version-stable guarantees that remain meaningful under policy and tooling changes, provided those changes are declared as part of the frozen record and their effects are budgeted.

AI agents and adaptive systems.

For tool-using AI systems, the verified object is the episode trace under declared semantics, not the model weights. The proof-carrying bundle-and-checker interface is therefore the appropriate unit of assurance, aligning with the classical producer/consumer verification paradigm [1,2].

*14.3. Limits of the Present Work*

The framework is intentionally verification-theoretic. It does not attempt to fully model adversaries, prove semantic correctness of AI models, or infer universal guarantees over all possible trajectories. All results are conditional on a frozen record and on the validity of the stated inequality obligations in that record. This is not a weakness: it preserves falsifiability and prevents the category error of treating drifting metrics as stable truth.

 doi:10.20944/preprints202601.0050.v1



Moreover, numerical audits are *instantiations* of constants and witnesses, not replacements for analytic proofs. They are useful because they are checkable and can be made conservative by design, but they must be interpreted as evidence in a declared semantics rather than as universal claims.

### *14.4. A Research Agenda with Theorem-Shaped Milestones*

The programme yields a concrete sequence of next steps, each phrased as a brick-discharge problem:

(i) **Pack library (definitions).** Publish a catalogue of frozen-record templates for common deployments (SOC pipeline, cloud posture management, tool-using agents), including standard neutral conventions and observation commitments.

(ii) **Coupling identification (empirical $\kappa$).** Develop robust, conservative estimation procedures for coupling coefficients (including resampling robustness and worst-case confidence bands) to populate certificate tables from operational data.

(iii) **Step–6 for real upgrades.** Define refinement operators for tool and policy upgrades and prove (or conservatively bound) integrable scale-generator majorants $G \in L^1$, yielding summable drift and non-regression across versions.

(iv) **Non-normality-aware audits.** Standardise audits that distinguish symmetric/selfadjoint from non-normal regimes and report transient growth indicators alongside decay/gap estimates [26].

(v) **Floor-aware governance.** Operationalise feasibility floors by connecting resource constraints (staffing, latency, cadence) to certified dissipation margins and forcing densities, producing planning tools that detect infeasible improvement targets.

### *14.5. Final Positioning*

The main conceptual result can be stated in one sentence:

> *Cybersecurity becomes auditable and comparable under change when it is expressed as a forced contraction inequality for a single residual in a frozen measurement geometry, with all unresolved effects routed into explicit budgets.*

This reframing does not replace security practice; it makes its claims precise. It provides the mathematical grammar required for human-centred assurance in evolving human-centered systems: clear semantics, explicit constants, and a mechanically checkable PASS/FAIL interface.

**Funding:** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Data Availability Statement:** No datasets were generated or analysed in this work. All results are of a theoretical and analytical nature and are fully contained in the manuscript.

**Ethics Approval:** This article does not contain any studies with human participants or animals performed by the author.

**Declaration of Generative AI and AI-Assisted Technologies in the Writing Process:** During preparation of this manuscript, the author used ChatGPT (OpenAI) in an assistive role for tasks such as drafting and editing text, formatting formulas and statements in LaTeX, and exploring alternative formulations of arguments and proofs. All AI-assisted content and suggestions were reviewed, edited, and critically assessed by the author, who takes full responsibility for the final form of all scientific claims, mathematical statements, proofs, and conclusions. No generative system was used to fabricate, alter, or selectively filter empirical or numerical data, and no proprietary, confidential, or unpublished information was provided to any AI system.

**Acknowledgments:** The author affirms sole authorship of this work. First-person plural ("we") is used purely for expository clarity. No co-authors or collaborators contributed to the conception, development, analysis, writing, or revision of the manuscript. The author received no external funding and declares no institutional, ethical, or competing interests.

 doi:10.20944/preprints202601.0050.v1



**Conflicts of Interest:** The author declares that there are no competing interests.

## References


1. Bauer, A.; Leucker, M.; Schallhart, C. Runtime Verification for LTL and TLTL. *ACM Transactions on Software Engineering and Methodology* **2011**, *20*, 14:1–14:64. https://doi.org/10.1145/2000799.2000800.
2. Necula, G.C. Proof-Carrying Code. In Proceedings of the Proceedings of the 24th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL '97), New York, NY, USA, 1997; pp. 106–119. https://doi.org/10.1145/263699.263712.
3. Pazy, A. *Semigroups of Linear Operators and Applications to Partial Differential Equations*; Vol. 44, *Applied Mathematical Sciences*, Springer: New York, 1983. https://doi.org/10.1007/978-1-4612-5561-1.
4. Engel, K.J.; Nagel, R. *One-Parameter Semigroups for Linear Evolution Equations*; Vol. 194, *Graduate Texts in Mathematics*, Springer: New York, 2000.
5. Anderson, R. *Security Engineering: A Guide to Building Dependable Distributed Systems*; John Wiley & Sons: New York, 2001.
6. Sasse, M.A. "Technology Should Be Smarter Than This!": A Vision for Overcoming the Great Authentication Fatigue, 2014. Position paper / invited talk manuscript (UCL Discovery).
7. Adams, A.; Sasse, M.A. Users are not the enemy. In Proceedings of the Proceedings of the 1999 Workshop on New Security Paradigms (NSPW '99), New York, NY, USA, 1999. https://doi.org/10.1145/322796.322806.
8. Beautement, A.; Sasse, M.A.; Wonham, M. The Compliance Budget: Managing Security Behaviour in Organisations. In Proceedings of the Proceedings of the 2008 Workshop on New Security Paradigms (NSPW '08), New York, NY, USA, 2008; pp. 47–58. https://doi.org/10.1145/1595676.1595684.
9. Herley, C. More Is Not the Answer. *IEEE Security & Privacy* **2014**, *12*, 14–19.
10. Nielsen, M.A.; Chuang, I.L. *Quantum Computation and Quantum Information*, 10th anniversary edition ed.; Cambridge University Press: Cambridge, 2010.
11. Terhal, B.M. Quantum error correction for quantum memories. *Reviews of Modern Physics* **2015**, *87*, 307–346. https://doi.org/10.1103/RevModPhys.87.307.
12. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79.
13. Wald, R.M. *General Relativity*; The University of Chicago Press: Chicago, 1984.
14. Wald, R.M. *Quantum Field Theory in Curved Spacetime and Black Hole Thermodynamics*; University of Chicago Press: Chicago, 1994.
15. Fewster, C.J.; Verch, R. Algebraic Quantum Field Theory in Curved Spacetimes. In *Advances in Algebraic Quantum Field Theory*; Brunetti, R.; Fredenhagen, K.; Verch, R., Eds.; Springer: Cham, 2015; pp. 125–189, [arXiv:math-ph/1504.00586]. https://doi.org/10.1007/978-3-319-21353-8_4.
16. Farina, L.; Rinaldi, S. *Positive Linear Systems: Theory and Applications*; John Wiley & Sons: New York, 2000.
17. Briat, C. *Linear Parameter-Varying and Time-Delay Systems: Analysis, Observation, Filtering & Control*; Advances in Delays and Dynamics, Springer: Berlin, Heidelberg, 2015. https://doi.org/10.1007/978-3-662-44050-6.
18. Khalil, H.K. *Nonlinear Systems*, 3 ed.; Prentice Hall: Upper Saddle River, NJ, 2002.
19. Sontag, E.D. Input to State Stability: Basic Concepts and Results. In *Nonlinear and Optimal Control Theory*; Springer: Berlin, Heidelberg, 2008; Vol. 1932, *Lecture Notes in Mathematics*, pp. 163–220. https://doi.org/10.1007/978-3-540-77653-0_5.
20. Diestel, J.; Uhl, J.J. *Vector Measures*; Vol. 15, *Mathematical Surveys*, American Mathematical Society: Providence, RI, 1977.
21. Watrous, J. *The Theory of Quantum Information*; Cambridge University Press, 2018.
22. Lyapunov, A.M. *The General Problem of the Stability of Motion*; Taylor & Francis, 1992. Original work published in Russian, 1892.
23. Grönwall, T.H. Note on the Derivatives with Respect to a Parameter of the Solutions of a System of Differential Equations. *Annals of Mathematics* **1919**, *20*, 292–296. https://doi.org/10.2307/1967124.
24. Coddington, E.A.; Levinson, N. *Theory of Ordinary Differential Equations*; McGraw–Hill: New York, 1955.
25. Berman, A.; Plemmons, R.J. *Nonnegative Matrices in the Mathematical Sciences*; Vol. 9, *Classics in Applied Mathematics*, SIAM: Philadelphia, PA, 1994.

 doi:10.20944/preprints202601.0050.v1



26. Trefethen, L.N.; Embree, M. *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators*; Princeton University Press: Princeton, NJ, 2005.
27. Farina, L.; Rinaldi, S. *Positive Linear Systems: Theory and Applications*; Vol. 255, *Pure and Applied Mathematics*, Wiley–Interscience: New York, 2000.