Human Seal

HUMAN SEAL · GUARDRAILS · W156

# Oversight-Failure Guardrails

Making the Human Gate proof against rubber-stamping, grounded in the meaningful-human-control and automation-bias literature.

Built 2026-09-20. Live status is in the open items register. Reference implementation: FLOCORE.

Status: PART-SHIPPED. The Seal rails and the rubber-stamp measure are LIVE on production (verified 2026-09-20); the gate-side user-interface guardrails and the effectiveness test are scoped for the gate lane. Grounded in two research packs (fetched primaries, tagged): docs/research/guardrails_mhc_requirements.md (meaningful human control, R1 to R20) and docs/research/guardrails_antirubberstamp.md (anti-rubber-stamp, G1 to G12).

## Purpose

A human approval that is a reflexive click is worse than none: it launders a machine decision and drops blame on the nearest person (Elish 2019, moral crumple zone, SOURCED). This wave makes the Human Gate proof against rubber-stamping by construction, and, decisively, makes oversight QUALITY provable from the receipt rather than assumed. The core finding of both packs is the same: the receipt must record the decision conditions, and only then can rubber-stamping be measured instead of hoped against (Green 2022 VERIFIED; Langer, Lazar, Baum 2025 VERIFIED: presence of a control is not evidence it works).

## What is LIVE (this wave, VERIFIED in code and tested)

- The Seal already enforced authority (P3, tracing property 3) and non-bypass (P4, positive human action).
- NEW: the Seal now records signed, tamper-evident decision-conditions evidence under an optional `oversight` object (apps/api/app/services/human_seal.py, OVERSIGHT_FIELDS), backward compatible with existing Seals:
  - context_hash (R9): the sha256 of the exact context the human was shown.
  - decision_latency_ms (R16, G3): time from AI proposal to decision, near-zero is a stamp signature.
  - pre_ai_decision (G1, commit-first): the human's own call before seeing the AI recommendation.
  - risk_tier (G9), ai_version (R6, dual attribution), rationale (G4), evidence_panels (G2), second_approver_ref (R4, dual control), rules_version (R20).
  - quality_flags (G8): the low-oversight-quality flags for the decision.
- NEW: rubber_stamp_indicators() computes the six measurable indicators over a set of Seals for one approver or queue, and quality_flags() derives the per-decision flags. Both tested (tests/test_human_seal_guardrails.py), including that a forged fast-decision fails signature verification, and that a stamp set is flagged while a healthy reviewer is not.

## The unified guardrail set (C1 to C8): structure, context, authority

Each guardrail: the requirement, the source (tagged from the packs), the mechanism, and where it is enforced (SEAL = the receipt rails, LIVE; GATE-UI = the approval interface, scoped for the gate lane; PROC = a process or release gate).

| ID | Guardrail | Source | Enforced | Status |
|---|---|---|---|---|
| C1 | Fail-closed HITL: a positive human approve recorded before execution; timeout or error is non-execution | Article 36 2013 (VERIFIED); EU AI Act 14(4)(d) (VERIFIED) | GATE + SEAL | SEAL live, gate scoped |
| C2 | Explicit registry of gated (world-affecting) action types | Siebert 2022 property 1 (VERIFIED) | GATE | scoped |
| C3 | Commit-first: capture the human's own decision before the AI recommendation is shown | Bucinca 2021 update (VERIFIED) | GATE-UI + SEAL (pre_ai_decision) | field live, UI scoped |
| C4 | Reveal-gated evidence: approve disabled until required panels surfaced, per-panel dwell recorded | Green 2022 (VERIFIED); Bucinca on-demand (VERIFIED) | GATE-UI + SEAL (evidence_panels) | field live, UI scoped |
| C5 | Risk-scaled minimum decision-view time, recorded (not a flat 30s) | Bucinca wait (VERIFIED); Parasuraman/Manzey 2010 (SOURCED) | GATE-UI + SEAL (decision_latency_ms) | field live, UI scoped |
| C6 | Context shown to the human, hashed onto the receipt | Santoni de Sio and van den Hoven 2018 tracing (VERIFIED); Art 12 (VERIFIED) | SEAL (context_hash) | live |
| C7 | Felt accountability: named authenticated approver distinct from AI and requester, rationale required, outcome fed back | Skitka 1999 (SOURCED); Green/Elish caveat | SEAL (approver_ref, rationale, ai_version) + PROC | fields live, feedback loop scoped |
| C8 | Authority commensurate with severity, checked at approval time | Siebert property 3 (VERIFIED); Art 14(5) (VERIFIED) | GATE (P3 authorize) | live |

# The unified guardrail set (C9 to C16): friction, measure, process

| ID | Guardrail | Source | Enforced | Status |
|---|---|---|---|---|
| C9 | Dual control (two distinct approvers) for the highest tier | Art 14(5) (VERIFIED) | GATE + SEAL (second_approver_ref) | field live, gate scoped |
| C10 | Contestable-by-construction: reject/modify as prominent as approve, no default, timeout fails to no action | Green 2022 (VERIFIED); Elish 2019 (SOURCED) | GATE-UI | scoped |
| C11 | Independent stop into a safe state, and override/reverse, each sealed | Art 14(4)(e) and (d) (VERIFIED); ICRC 2021 (SOURCED) | GATE | scoped |
| C12 | Volume and repetition control: gate only what warrants it, cap interruptions per approver per hour | Felisberto 2024, Ancker 2017 (SOURCED); Bucinca (VERIFIED) | GATE + metric | scoped |
| C13 | Adaptive friction: heaviest forcing functions only on low-confidence, high-blast, novel cases | Bucinca 2021 (VERIFIED) | GATE (risk_tier routing) | field live, routing scoped |
| C14 | Rubber-stamp monitor: per-approver, per-queue agreement/refusal/dwell with threshold alarms | Green 2022 (VERIFIED); Parasuraman (SOURCED); alert-fatigue lit (SOURCED) | SEAL metric (rubber_stamp_indicators) | live |
| C15 | Outcome-linked receipt: record succeeded/failed/reversed against the decision id | Siebert property 4 (VERIFIED); Article 36 accountability (VERIFIED) | SEAL | field reserved, wiring scoped |
| C16 | Pre-deployment oversight-effectiveness test: seed bad AND good proposals, measure catch rate and false-block rate, set a ship criterion per tier | Green 2022 (VERIFIED); Langer/Lazar/Baum 2025 (VERIFIED) | PROC (release gate + drill) | function built (W161.1); human drills are tracked in the register |

## The risk-tier table

Each tier of action has its own requirements. This is the one table; the code is the source.

The approver count, the authority check, the maximum Seal age and the minimum review time before approve are enforced today (by the gate and the approval page). The friction column and the drill thresholds are proposals for Floris and HGI to set: they are read only by the effectiveness drill and do not change what the gate allows.

Enforced today by the gate and the approval page:

| Tier | Approvers | Authority | Maximum Seal age | Minimum review time before approve |
|---|---|---|---|---|
| low | 1 | the action's min_level, checked at approval | 24 hours | 0 seconds |
| medium | 1 | the action's min_level, checked at approval | 4 hours | 3 seconds |
| high | 1 | the action's min_level, checked at approval | 1 hours | 8 seconds |
| critical | 2 | the action's min_level, checked at approval | 15 minutes | 15 seconds |

Proposed, not yet enforced (read only by the effectiveness drill):

| Tier | Friction | Minimum catch rate | Maximum false-block rate | Minimum cases per side |
|---|---|---|---|---|
| low | none beyond the enforced checks | 60% | 30% | 20 |
| medium | rationale recorded (C7) | 75% | 20% | 30 |
| high | rationale recorded (C7); commit-first (C3); evidence panels surfaced (C4); risk-scaled decision-time floor (C5) | 90% | 15% | 40 |
| critical | rationale recorded (C7); commit-first (C3); evidence panels surfaced (C4); risk-scaled decision-time floor (C5); context hash of what was shown (C6) | 95% | 10% | 50 |

The effectiveness drill (C16) reports two numbers, not one: the catch rate on seeded bad proposals and the false-block rate on seeded good ones (W161.1). A gate that refuses everything catches every bad proposal, so the catch rate alone cannot tell a working gate from a useless one. A drill with too few cases on either side does not pass; it reports not run.

## The measurable rubber-stamp indicators (computable from the receipt)

All six are computed by rubber_stamp_indicators() from the oversight fields, so oversight quality is auditable, not assumed:

- Agreement / approval rate (approve-unchanged fraction): high on a high-risk queue is the primary alarm.
- Refusal / override rate: trending to zero is a complacency signature.
- Decision latency (dwell): near-zero is a stamp signature.
- Below-latency-floor count and empty-rationale count: per-decision low-quality flags.
- Interruptions per approver per hour: the alert-fatigue signature (C12, scoped).
- Pre-AI vs post-AI divergence (C3): direct evidence the human held an independent view.

Standing rule, from the research and from FLOCORE house rule: every indicator is a QUESTION, not a verdict. A high agreement rate can be a genuinely good model on a low-risk queue; rule out the mundane cause before naming rubber-stamping. rubber_stamp_indicators() returns this caveat in its own output.

## Where each guardrail is enforced (and why not the static anti-pattern suite)

The static anti-pattern suite (scripts/check-antipatterns.sh) catches CODE-SHAPE anti-patterns at commit time. The rubber-stamp guardrails are RUNTIME properties over receipts and UI behavior, not code shapes, so forcing them into a static grep would be a greppable-proxy anti-pattern (checking something next to the claim, not the claim). They are therefore enforced as:

- a self-test (tests/test_human_seal_guardrails.py, in the test suite), and
- a runtime monitor (rubber_stamp_indicators), destined for the daily drill and the governance owner's board.

The gate-UI guardrails (C1 gate side, C2, C3 UI, C4, C5 UI, C10, C11, C12, C13 routing) are the gate lane's to build against this rails contract; FLOCORE supplies the receipt schema and the measure, not the tenant gate UI.

## Honest limits (from both packs)

- Forcing functions reduce but do not eliminate overreliance (Bucinca 2021, VERIFIED). A gate lowers the stamp rate; it cannot guarantee a correct human judgment.
- The friction that works is the friction people dislike (Bucinca 2021, VERIFIED). Expect pressure to weaken C3 and C5; weakening them quietly is itself a rubber-stamp risk. C13 (adaptive friction) rations the friction to where it changes outcomes.
- Presence is not effectiveness (Langer, Lazar, Baum 2025, VERIFIED). A system can pass every schema check and still deliver nominal oversight. Only C16 empirical testing tells us the gate actually works.
- Accountability without real control is a moral crumple zone (Elish 2019, SOURCED). C4, C5, C11 exist to give real control; where they cannot, the honest answer (Green 2022, VERIFIED) may be not to human-gate that action at all.
- A human gate can legitimize a system that should not exist (Green 2022, VERIFIED). No guardrail here answers whether the automated action should exist; that is the governance question above the gate.
- Present-human vs a delegated agent is a GAP (Level 3, W149): a WebAuthn or hardware step-up proves credential possession and a live gesture, not that a script is not driving the session. Defense in depth, not one check.
- The strongest quantitative figures (about 90 percent alert override; Skitka effect sizes) are SOURCED from abstracts and meta-summaries; read the primaries in full before any client-facing or regulatory use.

## Definition of done, status

- Spec written, each guardrail tied to a fetched source and a mechanism. DONE.
- The Seal records the decision-conditions evidence; the rubber-stamp measure computes from it; both tested. DONE (LIVE).
- The gate-UI guardrails and the C16 effectiveness test scoped for the gate lane. DONE (scoped).
- Rendered as a branded PDF into the Human Seal folder. DONE on render.