Team Authority Governance Console

authrex.systems · browser simulation · seeded PRNG · synthetic data only · client-side execution · hardened interactive prototype v0.16

Scope. Everything in this document describes independent research artifacts: browser simulations, formal models, and unbuilt hardware reference designs. Maturity is self-assessed. Nothing here is a certification, an Authorization to Operate, a government endorsement, or evidence of a fielded system. Model-checking results apply to the stated finite model and assumptions, not to physical implementations.

PURPOSE

Team authority governance gate for a human-supervised team of people and machine agents: an independent arbiter over an untrusted allocator maintains exactly one eligible authenticated holder for each responsibility, never zero (a gap) and never two (split authority), with bounded reversible transfers, a safe-hold fallback when no eligible holder exists, and a supervisor override that applies only from an allowlist. Every decision is written to a SHA-256 hash-chained ledger, and the run replays deterministically from its seed; from seed 20260727 the v0.16 build reproduces the published reference ledger root.

CONTROLS AND DISPLAY

START, **STEP**, and **RESET RUN** drive the deterministic run; **DROP AGENT**, **DROP CTRL**, **DROP SUP**, **RECOVER ALL**, and **FORCE CONTENTION** inject membership faults and contention.

STALE GRANT, **FORGE ARBITER GRANT**, **OVERRIDE, NON-ALLOWLISTED**, and **MALFORMED REQUEST** inject adversarial requests; each is rejected and quarantined. **HUMAN OVERRIDE** applies only from the allowlist, and **EMERGENCY AUTHORITY** grants a bounded five-tick lease.

RUN ALL SCENARIOS executes the twenty scripted, seeded scenarios; **VERIFY CHAIN** confirms the SHA-256 hash-chained ledger, **TAMPER TEST** demonstrates detection while the live chain stays intact, and **RUN LEGITIMACY + CHAIN AUDIT** checks authority state, the grant chain, transition legitimacy, and replay.

SEAL EVIDENCE captures a run and **VERIFY EVIDENCE (FULL REPLAY)** replays it in full; an altered evidence payload is rejected. **EXPORT JSON** writes a local file, **FORGERY SELF-TEST** exercises the verifier, and the boot self-test reports engine encapsulation. The console is fully client-side and makes no network calls.

READING THE VERDICTS

Every requested change of holder is adjudicated against eligibility, authentication, epoch, and lease checks: **GRANT** installs a transfer chained to its predecessor grant; **SAFE HOLD** is the fallback when no eligible holder exists and is never a grant of new authority; **OVERRIDE** applies only from the allowlist, and a failed supervisor is treated as ineligible; **RECOVER** re-enters through the same gate; **REJECTED** refuses stale leases, epoch mismatches, malformed requests, forged arbiter grants, and forged capability tags, which are quarantined and never become the effective holder. Results apply to the modeled conditions only.

LIMITATIONS

AUTHREX-TEAM is a deterministic synthetic research prototype. It models the governance of decision authority in a human-supervised team, one eligible authenticated holder per responsibility with bounded reversible transfers, safe-hold fallback, and allowlisted supervisor override. It does not model physical dynamics, real actuation, targeting, weapons, certified hardware, operational identity infrastructure, or an accredited deployment environment. Demonstrated in a synthetic simulation environment; no formal TRL determination has been performed. Governance and assurance only, with no dynamics, no actuation, and no targeting scope.

AUTHREX-TEAM v0.16 is a hardened interactive prototype, not a reproducible signed release; there is no release archive, no detached signature, and no signing key for this artifact. The standalone sha256 797076b55de7684534aae13e74c677306eb4281a9911ed0875991d12ea01399e is the only integrity claim.

CANONICAL TERMINOLOGY

SATA: Sensor & Actor Trust Assessment · HMAA: Human-Machine Authority Allocation · ADARA: Adversarial Deception-Aware Reasoning · MAIVA: Multi-Agent Integrity Voting · FLAME: Forced Latency for Authority-Managed Escalation · CARA: Controlled Authority Recovery Architecture · ERAM: Escalation Risk Assessment Model

AUTHREX Systems · authrex.systems · Burak Oktenli · July 28, 2026 · Site documentation, CC BY 4.0 where stated; artifact licenses vary by record · Independent single-author research; self-assessed maturity; not a government publication