Positive Human Control Console

authrex.systems · browser simulation · seeded PRNG · synthetic data only · client-side execution · hardened standalone prototype v0.13

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

Positive human control authority governance: the critical authorization can be held only by two distinct authenticated humans, a machine is never eligible, and the failure safe state is NO-GO. The two keys, R_AUTH_A and R_AUTH_B, are human only; the GO state is representable only when both are held by two distinct authenticated humans and no hold is in force. Supervisor vetoes and separately attributed arbiter fail safe holds move the system to NO-GO, every authority change is appended to a SHA-256 hash chained ledger, and the run replays deterministically; at seed 20260728 over 400 ticks the build reproduces the published reference ledger root.

CONTROLS AND DISPLAY

RUN, **PAUSE**, **STEP**, and **RESET** drive the deterministic run across six tabs: OPERATIONS with a live two dimensional schematic, 3D MODEL, SCENARIOS, LEDGER, EVIDENCE, and ABOUT; both visual layers are read only.

AUTHENTICATE KEY A and **KEY B**, **REVOKE KEY A** and **KEY B**, and **COMMIT (REQUIRES GO)** exercise the two person gate; **DROP AN AUTHENTICATOR** and **RECOVER** inject availability faults.

MACHINE ATTEMPTS AUTH KEY, **SAME HUMAN BOTH KEYS**, and **INJECT FORGED AUTH GRANT** inject adversarial requests; each is refused or quarantined. **SUPERVISOR VETO** and **CLEAR VETO** are supervised human actions that name the available supervisor.

RUN ALL SCENARIOS executes the 27 bundled deterministic scenarios; **VERIFY CHAIN**, **TAMPER TEST**, and **RUN AUTHORITY, TRANSITION, REPLAY AUDITS** drive the six audits. **SEAL EVIDENCE** and **VERIFY EVIDENCE (FULL REPLAY)** produce and check a portable record, **EXPORT JSON** writes a local file, and **FORGERY SELF-TEST** exercises the verifier. Fully client side; no network calls; no browser storage; a secure random source is required and the console refuses to start without one.

READING THE VERDICTS

GO is representable only with both keys held by two distinct authenticated humans and no hold in force; a commit is recorded only in GO and is voided the moment its authorization state changes. **NO-GO** is the failure safe default for every failure path. **VETO** is an explicit, bounded, supervised human action; **FAIL SAFE HOLD** is an arbiter action recorded with its trigger, never as a human action. **REJECTED** covers machine key attempts, the same human on both keys, and injected, stale, cross run, or forged grants, which are quarantined and never become the effective holder. Results apply to the modeled conditions only.

LIMITATIONS

AUTHREX-REDLINE is a deterministic synthetic research prototype. It models the authority governance principle of positive human control, that a machine is never eligible to hold the critical authorization and that two distinct authenticated humans are required to authorize it, as an abstract state machine. It does not model real nuclear weapons, NC3 systems, launch or release mechanics, targeting, authorization codes, command procedures, certified hardware, operational identity infrastructure, or an accredited deployment environment. Behavior shown is demonstrated in simulation. No formal TRL determination has been performed.

This build does not include formal model checking, property based adversarial testing beyond the bundled batteries, mutation testing, a digital signature, human credential authentication, implementation identity, a sealed release manifest, or an independent verifier. There is no signed release. The published file has SHA-256 480459efb3cfc1023c6b50d843c98befa9ed3d340f8a00eeb6308f0743914164; recompute it locally to confirm the file you received matches the file described here.

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