[CAPABILITY PROFILE · AUTHREX-02]

HMAA · HUMAN-MACHINE AUTHORITY ALLOCATION

Convert trust and operational context into an explicit, graded, enforceable authority state so that human control remains meaningful when the machine acts faster than a supervisor can evaluate.

MODELED **TESTED** **IMPLEMENTED**

Name variants: Historical expansions: Human-Machine Authority Architecture (evidence site), Hierarchical Mission Authority Architecture (title of Zenodo record 18861653). Original artifact titles are preserved unchanged. Tier names: the current HMAA page uses Full Autonomy, Restricted, Minimal, Revoked; the archived V4 catalog used Full, Standard, Restricted, Safe, which is not carried.

PRIMARY FUNCTION	Graded four-level authority state (A3 full autonomy, A2 restricted, A1 minimal, A0 revoked; design thresholds on the fused trust scalar of 0.80, 0.55 and 0.30 on the HMAA page) computed deterministically from the trust scalar and operational context, with asymmetric hysteresis: downgrade is immediate on trust loss, upgrade is stepwise and delayed.
MISSION AREAS	Any platform where a human must retain meaningful control at machine speed: UAS engagement decisions, autonomous ground platforms, AI-enabled command-and-control recommendations.
EVIDENCE ANCHORS	TLA+ model checked with TLC (scope clause below); 42-file Python package with 98 tests and 7 adversarial experiments; seeded browser simulation; Zenodo DOI 10.5281/zenodo.18861653.
CURRENT STANDING	Formally analyzed at stated bounds and demonstrated in simulation; HMAA_Refine is FAITHFUL on the discrete side and BOUNDED on the continuous side in the formal-coverage record; not built on hardware; not independently reproduced.
SELF-ASSESSED TRL	TRL 3. No formal TRL determination has been performed.
INTELLECTUAL PROPERTY	U.S. provisional application 63/999,105, filed March 7, 2026 (receipt 74759595), reported as filed; unexamined; confers no enforceable rights.

[MISSION PROBLEM]

Human oversight of a machine is nominal when the machine acts faster than the human can evaluate. A binary permit-or-deny cannot express partial trust and cannot degrade gracefully, so operators either over-trust (leave full authority in place while evidence erodes) or over-restrict (lock the system out when a partial contraction would keep the mission alive). Oscillation is the second failure: a system that restores authority as soon as a signal recovers can flap between states at the sensor noise rate. The consequence is either an unsafe action at full authority or a mission lost to an unnecessary lockout.

[OPERATIONAL RELEVANCE]

- Engagement-authority chains for uncrewed aircraft where the confirm-or-abort decision must be explicit at a defined tier.
- Ground robotic movement and convoy operations under lidar or vision degradation (speed and route constrained at A2 and A1).
- AI-enabled command and control where a recommendation must carry the authority tier under which it was generated.
- Human-machine teaming and collaborative combat aircraft concepts, where the authority held by the machine must be legible to the pilot at all times.

[CAPABILITY DESCRIPTION]

HMAA is the single component of the pipeline that issues an authority tier, so a reviewer tracing any decision has one place to look. The automaton takes the trust scalar from SATA and the operational context (consequence tier, operator availability) and computes a target tier. Downgrade to the target is immediate and may skip tiers; upgrade is stepwise and delayed by a hysteresis interval of roughly 5 to 15 seconds of sustained trust so that a transient recovery cannot restore authority. At A1 (minimal) the operator confirms where policy requires it, or the predeclared outcome applies; at A0 (revoked) actuation is withheld and a safe state is commanded, which invokes CARA. The automaton is deterministic: identical inputs produce identical tiers, which is what makes deterministic replay of a recorded run possible.

[HMAA · ARCHITECTURE, ASSURANCE, EVIDENCE]

[TECHNICAL ARCHITECTURE]

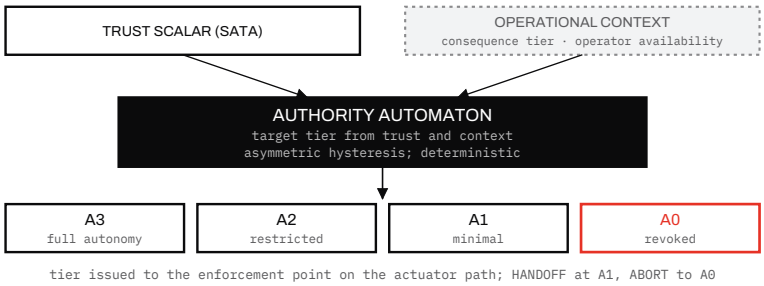

Figure 1. HMAA technical architecture. Encoding: solid = implemented; dashed = specified; gray = external; filled = enforcement point.

[HOW IT WORKS]

1. Receive the per-source trust scalar from SATA and the current operational context.
2. Compute the target tier from trust and context using the configured, versioned policy thresholds.
3. If the target is below the current tier, downgrade immediately (tiers may be skipped) and record the transition with its reason.
4. If the target is above the current tier, hold; upgrade one tier only after the hysteresis interval has elapsed with trust sustained above threshold.
5. At A1 (minimal), route the pending action to the operator for confirmation where mission policy requires it; if confirmation is absent when the bounded hold expires, the predeclared outcome (delay, handoff, or abort) applies.
6. At A0 (revoked), withhold actuation, command the safe state, and hand control to CARA for structured recovery.
7. Write every tier change and outcome (EXECUTE, DELAY, HANDOFF, ABORT) to the hash-chained record.

[TECHNICAL DIFFERENTIATION]

A kill switch has two states and no memory; role-based access control grants permission by identity, not by the current evidence. HMAA makes authority a computed, time-varying, enforceable state with a formal model behind it, and it separates the rate of downgrade from the rate of upgrade so that safety and availability are governed by different rules. Against Simplex-style runtime assurance it adds a gradient between full autonomy and full intervention; against policy-as-code it supplies the runtime state that the policy acts on. It composes with both and replaces neither.

[ASSURANCE MECHANISMS: DESIGN PROPERTY VERSUS ESTABLISHED PROPERTY]

Mechanism	Status
Tier confinement: no tier outside A3 to A0	Held in TLA+ at stated bounds (invariant)
Immediate downgrade on trust loss	Held in TLA+ at stated bounds (invariant)
Hysteresis on upgrade; no restoration by timeout	Held in TLA+ at stated bounds (invariant); timing values are design parameters
Stepwise upgrade path (UpgradeIsStepwise)	Vacuous at this bound: not established
Trust-authority weak consistency	Vacuous at this bound: not established
Liveness: an eventual outcome is issued	One liveness property held at stated bounds
Model-to-code conformance	By construction and by the 98-test package; not by refinement proof (Obligation 6 open)
Human override and handoff at A1	Design property; exercised in the seeded console

[TECHNICAL EVIDENCE]

Evidence	Result	Source / artifact
TLC model check of the HMAA automaton	23,748 distinct reachable states at depth 9 (26,397,356 states generated); the result describes the discrete authority automaton under the assumption that instantaneous authority equals its target and does not cover continuous behaviour between decision instants; of 8 stated properties, 5 invariants and 1 liveness property held, 2 vacuous at this bound (UpgradeIsStepwise, TrustAuthorityWeakConsistency)	HMAA_verification_v2_deposit.zip (deposit of record); Zenodo 10.5281/zenodo.18861653
Faithfulness of HMAA_Refine.tla	FAITHFUL on the discrete side; BOUNDED on the continuous side	authrex.systems/formal-coverage
Python package tests	42 files; 98 tests; 7 adversarial experiments; deterministic traces	Zenodo deposit
Seeded browser simulation	Deterministic replay from seed; synthetic data	burakoktenli.com/hmaa-simulation
Earlier figure of 48,751 states	WITHDRAWN; never to be reasserted	Correction log CL-01

[EVIDENCE SNAPSHOT]

EVIDENCE SNAPSHOT	
Implemented	Yes (Python package)
Executable artifact	Yes
Formal model	Yes
Simulation evidence	Yes
Hardware validation	No
Field validation	No
Independent validation	No
Government evaluation	No
Operational deployment	No

[LIMITATIONS]

- Bounded model checking of the discrete automaton is not a proof of the physical system and does not cover continuous behaviour between decision instants.
- Two of eight stated properties are vacuous at the checked bound; they are not established.
- Obligation 6 (GLUE between discrete and continuous models) is open; initial-state correspondence was withdrawn as bound-dependent (held at MaxTick = 3, violated at 4).
- Hysteresis timing (5 to 15 s) is a design parameter tuned in simulation; not characterized against any real control loop.
- Simulation and formal-model evidence only; all data synthetic; no hardware-in-the-loop test, no field validation, no independent replication, no deployment.
- Not independently reproduced or technically verified by another party; second-environment reruns are confirmation by the same producer.

[TECHNOLOGY READINESS (SELF-ASSESSED; NO FORMAL TRA)]

CURRENT ASSESSED TRL	TRL 3 (self-assessed against the DoD nine-level scale; no formal technology readiness assessment has been performed).
BASIS	The authority automaton has a formal model checked at stated bounds, a tested implementation, and seeded simulations exercising the critical function under trust loss and recovery. Laboratory validation on a representative controller loop (TRL 4) has not been performed.
EVIDENCE SUPPORTING	concept defined and specified; formal model (TLA+/TLC) at stated bounds; software implementation with test suite; simulation demonstration
EVIDENCE REQUIRED FOR NEXT TRL	(1) Software-in-the-loop with a representative controller or flight-stack interface (ArduPilot or ROS 2) in a laboratory environment. (2) Independent re-execution of the deposited TLC configuration and the 98-test package. (3) Closure or explicit scoping of the two vacuous properties at a larger bound, and a decision on Obligation 6.

[INTEGRATION PROFILE]

INPUTS	SATA trust scalar; consequence tier (from ERAM or mission policy); operator availability; mission clock.
OUTPUTS	Authority tier A3 to A0; outcome (EXECUTE, DELAY, HANDOFF, ABORT); ledger entry.
INTERFACES	Authority interface to the enforcement point on the actuator path; operator confirmation channel; ledger.
DEPLOYMENT	Software module beside the controller; the automaton is small enough for embedded targets but no embedded build exists.
COMPUTE REQUIREMENTS	Not characterized on target hardware.
SECURITY BOUNDARY	Inside: automaton, policy thresholds (versioned), ledger. Outside: trust inputs, operator channel, actuator interlock.
DATA REQUIREMENTS	Synthetic; government-defined scenarios can be run on the seeded console without data transfer.

[TRANSITION ROADMAP]

Phase	Activity for this capability	Who can execute it	Exit evidence
1 Technical review	Review of the artifacts listed in the evidence table	Program alone, with a government reviewer	Reviewer findings; correction log entries
2 Government-defined simulation	Sponsor scenarios on the seeded console	Program with sponsor scenarios	Deterministic, replayable results; hash-chained records
3 Degradation and failure injection	Evidence lapse, spoofing, delayed human response	Program with sponsor parameters	False-restriction, miss, and contraction-latency figures
4 Government testbed / SIL	Integration with a representative controller or interface	Requires sponsor interface and testbed	SIL integration report
5 Hardware-in-the-loop	Representative interference; timing on real hardware	Requires sponsor laboratory or rig	Measured latencies; TRL 4 to 5 evidence
6 Relevant-environment demonstration	Independent evaluation and red team	Requires independent evaluator	Independent report
7 Program integration	Only after earlier gates succeed	Requires transition partner	Not claimed to be available

[GOVERNMENT ENGAGEMENT REQUEST (SPECIFIC)]

- Independent review of the TLA+ model and the 98-test package (Phase 1) by a formal-methods-capable government laboratory.
- A government-defined mission scenario with an explicit confirm-or-abort tier so that policy thresholds can be set by the sponsor rather than by the author.
- Access to a software-in-the-loop flight or vehicle stack for Phase 4.
- Guidance on operator-comprehension metrics for authority state, a success metric the program cannot evaluate alone.

[REVIEWER QUESTIONS THIS PROFILE ANSWERS]

What is it: see capability description. Why does DoD care: see operational relevance. Where would it fit: see integration profile. Does it exist: see evidence snapshot. What evidence supports it: see technical evidence. Has anyone independent tested it: no. Has it been tested in the real world: no. What does it rely on: see limitations and security boundary. What can cause it to fail and what does it do then: see assurance mechanisms and the published counterexamples. How mature: see the TRL block. What would the government need to provide: see the request above. Next measurable milestone: first item under evidence required for next TRL.