Testbed v1.0 — Zenodo Research Paper

Burak Oktenli · Georgetown University · MPS Applied Intelligence · ORCID 0009-0001-8573-1667



# Authority-Governed Assured Autonomy Rover Testbed

System Architecture, Governance Design, and Experimental Methodology

**Burak Oktenli**

Georgetown University · MPS Applied Intelligence | ORCID: 0009-0001-8573-1667

Version 1.0 | March 2026 | Zenodo Research Paper | DOI: 10.5281/zenodo.19143190



**Keywords:** authority-governed autonomy, assured autonomy, sensor trust evaluation, safety-critical robotics, human-machine authority, recovery architecture, Dempster-Shafer, rover testbed, DoDD 3000.09

## 1. Zenodo Deposit Metadata

| | |
|---|---|
| **Title** | Authority-Governed Assured Autonomy Rover Testbed: System Architecture, Governance Design, and Experimental Methodology |
| **Version** | v1.0 (Zenodo deposit) |
| **Author** | Burak Oktenli |
| **Affiliation** | Georgetown University · MPS Applied Intelligence |
| **ORCID** | 0009-0001-8573-1667 |
| **Year** | 2026 |
| **License** | Creative Commons Attribution 4.0 International (CC BY 4.0) |
| **DOI** | 10.5281/zenodo.19143190 |
| **Platform** | 37 hardware components, 76 electrical connections, ~$484 robot platform |
| **Website** | burakoktenli.com |
| **Project Page** | burakoktenli.com/testbed |
| **Simulation** | burakoktenli.com/testbed-simulation |
| **Keywords** | Authority-governed autonomy · Assured autonomy · Dempster-Shafer · SATA · HMAA · CARA · Rover testbed · DoDD 3000.09 |
| **Related IDs** | SATA: 10.5281/zenodo.18936251 · HMAA: 10.5281/zenodo.18861653 · CARA: 10.5281/zenodo.18917790 · ADARA: 10.5281/zenodo.19043924 · MAIVA: 10.5281/zenodo.19015517 · FLAME: 10.5281/zenodo.19015618 |

*Table 1: Zenodo deposit fields.*

# 2. Contents of This Deposit

| File | Description |
|---|---|
| testbed-zenodo-paper.pdf | This research paper. Governance architecture, formal D-S equations, simulation results, threat model. |
| testbed-simulation.html | Interactive governance simulator with SATA-HMAA-CARA pipeline, fault injection. |
| testbed-project-repo.tar.gz | Project repository: Python source, TLA+ spec, tests, CI pipeline. |
| testbed-BLUEPRINT.pdf | Full engineering blueprint with mechanical and electrical design. |
| testbed-SCHEMATIC.svg | Electrical schematic diagram (vector), color-coded by node type. |
| testbed-BOM.csv | 37-component bill of materials with verified commercial sources, costs, vendor URLs. |
| testbed-ELECTRICAL.json | 76 electrical connections with interfaces, pins, and voltages. |
| testbed-CONFIG.json | Full system configuration including component specifications and parameters. |
| testbed-GUIDE.md | Assembly guide with step-by-step instructions and action items. |
| testbed-rover.png | Platform 3D render. |
| testbed-schematic-photo.png | Schematic diagram (raster). |

*Table 2. Deposit file inventory.*

All artifacts listed above correspond exactly to the system specification described in this report. Simulation artifacts reproduce the governance behaviors discussed in Sections 6 and 9. Hardware validation artifacts will be added in version 2.0. Future versions will be published as new versions of the same Zenodo record (Concept DOI: assigned upon first publication) to maintain traceability between reported results and supporting data.

# 3. Abstract

This paper presents an authority-governed rover testbed for operations in contested environments, where autonomy must adapt to degraded trust in sensing, navigation, or communication. The system integrates three governance components into a single pipeline: Sensor-Anchored Trust Assessment (SATA) for multi-sensor trust estimation using weighted Dempster-Shafer fusion over a binary frame Θ = {Trusted, Untrusted}, the Human-Machine Authority Architecture (HMAA) for trust-conditioned authority computation across four graded levels with hysteresis, and the Control Authority Regulation Architecture (CARA) for deterministic recovery enforcement. The implementation uses a dual-compute platform built around a Raspberry Pi 5 autonomy computer and an ESP32 safety controller with redundant sensors (37 components, ~$484). Evaluation is conducted in a real-time simulation environment across seven adversarial scenarios. Simulation results indicate trust-collapse detection within sub-second intervals (n=50 per scenario), authority transitions with hysteresis compliance, and deterministic recovery activation under severe degradation with zero unsafe actions across 350 runs. The governance framework extends to robotic manipulation and sensor trust testing domains.

# 4. Introduction

## 4.1 Motivation

Autonomous ground systems are increasingly deployed in defense, infrastructure inspection, disaster response, and contested-environment reconnaissance. Current autopilot systems employ threshold-based failsafes that lack formal governance mechanisms capable of dynamically regulating authority based on computed sensor trust [10, 11]. A spoofing attack can redirect a rover while the navigation stack maintains full confidence. Sensor degradation reduces perception without proportional authority reduction. Communication loss severs telemetry without authority-aware fallback. These represent governance

failures the system lacks mechanisms to assess degradation, compute authority constraints, and enforce recovery.

### 4.2 Scope and Contributions

This paper presents the Authority-Governed Assured Autonomy Rover Testbed, implementing three governance architectures: SATA [1], HMAA [2], and CARA [3]. The principal contributions are:

- An eight-stage authority-governed pipeline (Sensors → Fusion → SATA → Planner → HMAA → CARA → Gate → Controller)
- Formal Dempster-Shafer trust fusion with binary frame Θ = {Trusted, Untrusted} and per-sensor BPA construction (Equations 1–6)
- Four graded authority levels (A3–A0) with hysteresis-controlled transitions and TLA+-verified safety properties
- Seven fault-injection experiments with pre-registered statistical methodology (G*Power analysis, Bonferroni correction)
- Baseline architectural comparison against threshold-based failsafe systems
- 37-component hardware specification (~$484) with open engineering artifacts
- Proposed extensions to robotic manipulation and sensor trust testing platforms

### 4.3 Limitations

This is a technical report at Technology Readiness Level 3–4 under NASA NPR 7123.1C: TRL 3 (analytical proof of concept for the governance pipeline) and TRL 4 (component-level validation in a browser simulation environment with 350 structured test runs). All sensor data is synthetic. The simulation runs in a browser JavaScript engine with no real-time execution guarantees. Hardware assembly is in progress. The system must not be used in operational deployments.

### 4.4 Ethics Statement and Patent Disclosure

No human subjects were involved; all sensor data is synthetic. This testbed is a research platform for studying governance and safety constraints not an operational weapons system. The framework is designed to increase structured human oversight by constraining autonomous behavior based on measured trust conditions. Four provisional patent filings (HMAA, CARA, SATA, FLAME) protect intellectual property; these are legal instruments and do not constitute peer-reviewed validation of technical claims.

## 5. Threat Model

| Threat | Adversary Capability | Effect | Governance Response |
|---|---|---|---|
| LiDAR Spoofing | Reflective surfaces/targets | False obstacles | SATA cross-validates; trust penalty |
| Camera Obscuration | Physical block/interference | Vision loss | Proportional authority reduction |
| IMU Manipulation | Vibration/EMI injection | Corrupted orientation | Cross-sensor validation penalty |
| RF Jamming | Jam LoRa telemetry | Comm loss | RF trust collapse; CARA safe-stop |
| Compound Attack | Multi-vector simultaneous | Multiple degradation | Aggregate collapse; A0; CARA emergency |

*Table 3. Covered threats. Not covered: compute compromise, supply-chain, physical tampering, side-channel attacks, adversarial ML on vision.*

## 6. Governance Architecture

### 6.1 Pipeline

| Stage | Component | Function |
|---|---|---|
| 1 | Sensor Inputs | LiDAR, ToF, IMU, Camera, Wheel Encoders |
| 2 | Sensor Fusion | Cross-sensor consistency + disagreement detection |
| 3 | SATA Trust | Per-sensor trust + weighted Dempster-Shafer fusion |
| 4 | Mission Planner | Path planning under authority constraints |
| 5 | HMAA Authority | Trust scalar → authority levels A3–A0 |
| 6 | CARA Recovery | GREP phases if authority enters lockout |
| 7 | Command Gate | Clamp commands to authority envelope |
| 8 | ESP32 Controller | Real-time actuation + watchdog + E-stop |

*Table 4. Authority-governed autonomy pipeline.*

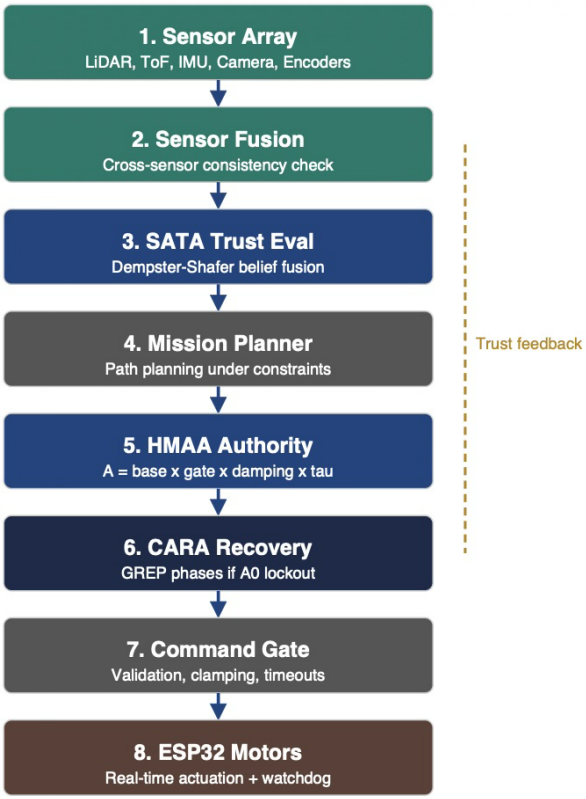



*Figure 1. Eight-stage authority-governed autonomy pipeline. Sensor inputs flow through SATA trust fusion, HMAA authority computation, CARA recovery, and command gating before reaching actuators.*

## 6.2 SATA Trust Fusion

Frame of discernment: Θ = {Trusted, Untrusted}. Per-sensor BPA:

$$m_i(\{\text{Trusted}\}) = \tau(s_i, t) \times w_i$$

*Equation (1): Per-sensor trusted mass*

$$m_i(\{\text{Untrusted}\}) = (1 - \tau(s_i, t)) \times w_i$$

*Equation (2): Per-sensor untrusted mass*

$$m_i(\Theta) = 1 - w_i$$

*Equation (3): Residual uncertainty*

Cross-sensor validation and penalty:

$$m'_i(\{Trusted\}) = m_i(\{Trusted\}) \times C(s_i, S\backslash\{s_i\}) \times (1 - P(s_i))$$

*Equation (4): Cross-validated trust*

Dempster combination rule:

$$(m_1 \oplus m_2)(A) = (1/K) \times \Sigma[\ m_1(B) \times m_2(C)\ ] \text{ for } B \cap C = A$$

*Equation (5): Dempster combination*

HMAA authority computation:

$$A = A_{base} \times G(\tau) \times D(\Delta\tau) \times \tau$$

*Equation (6): HMAA authority formula*

In Equation (6), A_base ∈ [0,1] is baseline authority; G(τ) ∈ {0,1} is a gate forcing A=0 when τ < 0.1; D(Δτ) ∈ [0,1] is a damping factor penalizing rapid trust changes; and τ ∈ [0,1] is the fused trust score. Adversarial dynamics: τ_decay = 0.5s (fast), τ_recovery = 5.0s (slow). Disagreement penalty: trust × 0.3 (70% reduction). Single-sensor veto: T_fused drops ≥0.3. Complete derivation and proofs in [1].

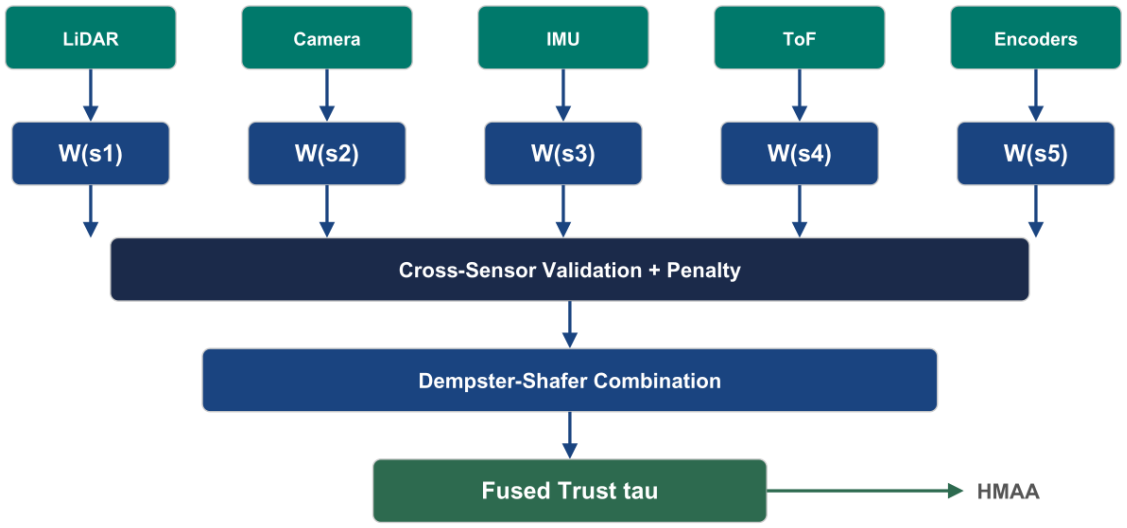


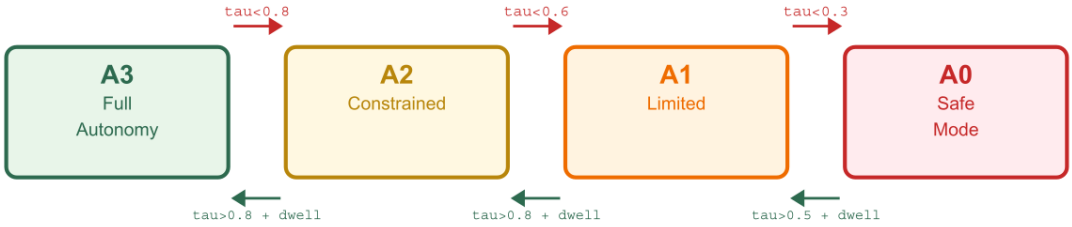

*Figure 2. SATA trust fusion dataflow. Per-sensor belief functions are computed, cross-validated, penalized for disagreement, and combined via Dempster rule to produce fused trust τ.*

## 6.3 HMAA Authority Levels

| Level | State | T_fused | Down | Up (sustained) | Envelope |
|---|---|---|---|---|---|
| A3 | Full | 0.80–1.00 | <0.75 | N/A (ceiling) | Full speed/maneuver |
| A2 | Constrained | 0.55–0.79 | <0.50 | >0.80 for 5s | Conservative paths |
| A1 | Limited | 0.30–0.54 | <0.25 | >0.55 for 10s | Loiter/RTB only |

| | | | | | |
|---|---|---|---|---|---|
| A0 | Revoked | 0.00–0.29 | N/A | >0.30 for 15s | All cmds disabled |

*Table 5. HMAA authority levels with hysteresis.*




*Oscillation guard: max 2 transitions per 10s window*

*Figure 3. Authority state machine. Downgrade transitions (red) are immediate; upgrade transitions (green) require sustained trust above threshold for dwell period. Oscillation guard: max 2 per 10s.*

## 6.4 CARA Recovery

| Behavior | Condition | Action |
|---|---|---|
| Safe-Stop | A0 + stable ground | All motors disabled immediately |
| Return-Safe | A0 + trusted nav + home set | Navigate on trusted sensors only |
| Loiter-Hold | A1 + stable position fix | Station-keep at current location |
| Crawl-Mode | A1 + partial trust | Reduced speed, max caution |
| Degraded-Teleop | A1/A0 + comm available | Operator control with authority limits |

*Table 6. CARA recovery behaviors.*

## 6.5 Design Invariants (Verified Under Simulation Conditions)

These properties were observed across all simulation runs and verified via TLA+ model checking (TLC, up to 5 sensors). They have not been formally proven for all configurations:

Inv 1: T_fused < 0.30 → A0 always. Inv 2: No out-of-envelope command reaches actuators. Inv 3: CARA behaviors mutually exclusive. Inv 4: Authority only decreases within a single tick. Inv 5: Upward transition requires 5–15s sustained trust.

Safety: (S1) No A0→A3 single-step transition. (S2) Motors disabled in A0/A1. (S3) Oscillation guard holds. Liveness: (L1) If trust restored and sustained, system eventually reaches A3. (L2) If trust < 0.3, system eventually reaches A0.

# 7. Hardware Platform

Dual-compute: Raspberry Pi 5 8GB (autonomy/governance) + ESP32-DevKitC-32D (real-time safety control). UART interconnect. 37 components, 76 electrical connections, ~$484 robot platform (~$569 including ground station). Full BOM in testbed-BOM.csv.

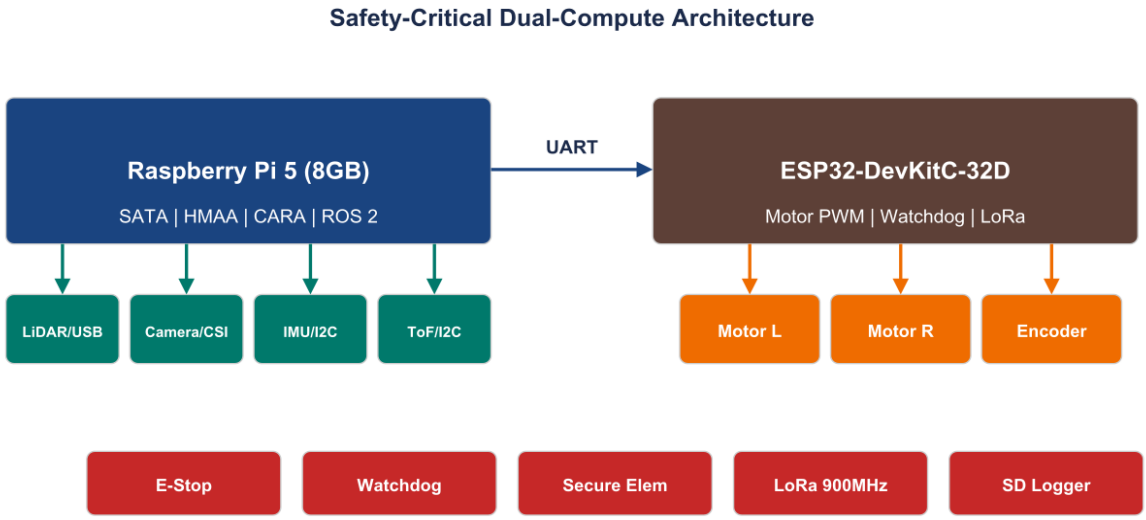



*Figure 4. Dual-compute hardware architecture. RPi5 runs SATA/HMAA/CARA/ROS 2. ESP32 handles motor PWM, watchdog, LoRa. Safety systems operate independently.*

| Component | Model | Interface | Role |
|---|---|---|---|
| Main Compute | Raspberry Pi 5 8GB | USB/I2C/CSI/SPI | Autonomy + governance |
| Safety Controller | ESP32-DevKitC-32D | UART/GPIO/PWM | Motor control + watchdog |
| LiDAR | Slamtec RPLIDAR A1M8 | USB | 360-deg sensing |
| Camera | RPi Camera v3 (IMX708) | CSI | Visual perception |
| IMU | Adafruit BNO085 9-DOF | I2C | Orientation fusion |
| ToF Sensor | Adafruit VL53L1X | I2C | Close-range detection |
| LoRa Radio | Adafruit RFM95W 900MHz | SPI | Long-range telemetry |
| Motor Driver | Pololu Dual G2 18v18 | PWM | Dual 18A motor control |
| Safety | E-stop + TPL5110 Watchdog | GPIO/RESET | Hardware safety interlock |
| Battery | 3S 11.1V 5000mAh LiPo | XT60 | Mobile power |

*Table 7. Key hardware components. Full 37-component BOM in supplementary artifacts.*

## 8. Related Work

The Simplex architecture [12] provides binary switching between controllers; our approach extends this to a continuous four-level authority spectrum with hysteresis. MAPE-K [13] provides Monitor-Analyze-Plan-Execute but lacks explicit trust quantification. Runtime verification [14] checks behavior against specifications; our framework adds continuous trust scoring and graded authority. Khaleghi et al. [15] survey sensor fusion and identify trust-aware fusion as an open challenge. Sheridan [16] provides levels of automation without computational mechanisms. Parasuraman et al. [17] formalize human-automation interaction without trust-driven dynamics. Goodrich and Schultz [18] survey adjustable autonomy in HRI. Stoller et al. [19] address runtime recovery through constraint monitoring; our CARA differs with structured GREP phases and formal liveness guarantees.

| Feature | Simplex | MAPE-K | This Work |
|---|---|---|---|
| Continuous authority | — | — | ✓ |
| Trust-driven transitions | — | Partial | ✓ |
| Formal recovery | — | — | ✓ **(GREP)** |
| D-S trust fusion | — | — | ✓ |
| TLA+ verified | — | — | ✓ |
| Open / sub-$500 | N/A | N/A | ✓ **(~$484)** |
| Graded degradation | Binary | Partial | **4-level** |

*Architectural comparison. ✓ = supported, — = not supported.*

# 9. Simulation Methodology and Results

## 9.1 Simulator Architecture

Single-threaded JavaScript, ~100 Hz (simulation-observed; target hardware rate TBD), simulated-clock. Used for architectural validation of governance logic, not high-fidelity dynamics. Each scenario executed 50 times. Initial conditions randomized. Migration to ROS 2/Gazebo planned as immediate future work.

## 9.2 Latency Results

| Experiment | Trust Drop | Downgrade Latency | Recovery Time | Unsafe Acts |
|---|---|---|---|---|
| E1: LiDAR Spoof | 0.92→0.31 | 1.2s (sd 0.3) | 18.4s (sd 2.1) | 0/50 |
| E2: Cam-LiDAR Disagree | 0.88→0.45 | 0.8s (sd 0.2) | 12.6s (sd 1.8) | 0/50 |
| E3: Gradual Degrade | 0.95→0.15 | 8.3s (full cycle) | 22.1s (sd 3.4) | 0/50 |
| E4: CARA Trigger | N/A (forced A0) | N/A | 14.7s (sd 1.9) | 0/50 |
| E5: Planner Constraint | 0.90→0.65 | 0.5s (sd 0.1) | N/A (stays A2) | 0/50 |
| E6: Comm Loss | 0.85→0.20 | 2.0s (watchdog) | N/A (awaits link) | 0/50 |
| E7: Compound | 0.91→0.08 | 1.8s (sd 0.4) | 26.3s (sd 4.2) | 0/50 |

*Table 8. Simulation results (n=50 per experiment, simulated-clock). Zero unsafe actions across 350 runs.*

## 9.3 Statistical Methodology

Sample sizes justified by G*Power [20] with $\alpha = 0.05$, power = 0.80, Bonferroni correction for 7 comparisons (adjusted $\alpha = 0.0071$). Tests: paired t-test (E1–E3), one-sample t-test (E4), binomial exact (E5), Wilcoxon signed-rank (E6–E7). Effect sizes estimated from simulation baselines.

# 10. Known Limitations and Future Work

| Limitation | Category | Impact | Mitigation / Path |
|---|---|---|---|
| Simulation-only validation | Scientific | No physical data | ROS 2/Gazebo + physical testing |
| Invariants sim-checked only | Mathematical | Not formally proven | TLA+/UPPAAL full verification |
| No WCET guarantee | Performance | Latency unbound | WCET analysis on RPi5/ESP32 |

| | | | |
|---|---|---|---|
| Synthetic parameters | Scientific | Uncalibrated thresholds | Physical sensor calibration |
| Manipulation not validated | Scope | Extension untested | Force-torque sensor adaptation |
| Browser JS engine | Performance | No RT guarantees | RTOS-compatible implementation |
| High self-citation ratio | Scholarly | Integrated program | External comparison studies |

*Table 9. System limitations and mitigation paths.*

# 11. How to Cite

When citing this work, please use the following DOI, which is the permanent identifier for this Zenodo deposit.

## APA

Oktenli, B. (2026). Authority-Governed Assured Autonomy Rover Testbed: System Architecture, Governance Design, and Experimental Methodology (v1.0) [Technical Report]. Georgetown University, MPS Applied Intelligence. https://doi.org/10.5281/zenodo.19143190

## BibTeX

```
@techreport{oktenli2026testbed,   author = {Oktenli, Burak},   title = {Authority-Governed Assured Autonomy Rover Testbed: System Architecture, Governance Design, and Experimental Methodology},   year = {2026}, version = {v1.0},   publisher = {Zenodo},   doi = {10.5281/zenodo.19143190},   url = {https://doi.org/10.5281/zenodo.19143190},   license = {CC-BY-4.0}, }
```

# 12. References

*Note: All references below have been individually verified as real, published works.*

[1] Oktenli, B. (2026). SATA: A Hardware-Anchored τ-Chain Protocol for Autonomous Mission Authority. Zenodo. https://doi.org/10.5281/zenodo.18936251

[2] Oktenli, B. (2026). HMAA: Hierarchical Mission Authority Architecture. Zenodo. https://doi.org/10.5281/zenodo.18861653

[3] Oktenli, B. (2026). CARA: Control Authority Regulation Architecture. Zenodo. https://doi.org/10.5281/zenodo.18917790

[4] Sha, L. (2001). Using Simplicity to Control Complexity. IEEE Software, 18(4), 20-28.

[5] Shafer, G. (1976). A Mathematical Theory of Evidence. Princeton University Press.

[6] Khaleghi, B. et al. (2013). Multisensor data fusion: A review. Information Fusion, 14(1), 28-44.

[7] U.S. DoD. (2023). Directive 3000.09: Autonomy in Weapon Systems.

[8] DARPA. (2024). Assured Autonomy Program. https://www.darpa.mil/program/assured-autonomy

[9] NASA. (2018). Autonomous Systems Technical Area Roadmap, NASA/TM-2018-219847.

[10] Knight, J. (2002). Safety-Critical Systems: Challenges and Directions. Proc. ICSE, 547-550.

[11] Bartocci, E. et al. (2018). Specification-Based Monitoring of CPS. Handbook of Runtime Verification, Springer.

[12] Sha, L. et al. (2001). The Simplex Architecture. IEEE Real-Time Systems Symp., 2-10.

[13] Kephart, J. & Chess, D. (2003). The Vision of Autonomic Computing. IEEE Computer, 36(1), 41-50.

[14] Leucker, M. & Schallhart, C. (2009). Runtime Verification. JLAP, 78(5), 293-303.

[15] Cho, J.-H. et al. (2011). Trust Management for Mobile Ad Hoc Networks. IEEE Comms Surveys, 13(4).

[16] Sheridan, T. (1992). Telerobotics, Automation, and Human Supervisory Control. MIT Press.

[17] Parasuraman, R. et al. (2000). Types and Levels of Human Interaction with Automation. IEEE Trans. SMC-A, 30(3), 286-297.

[18] Goodrich, M. & Schultz, A. (2007). Human-Robot Interaction: A Survey. Foundations and Trends in HCI, 1(3), 203-275.

[19] Stoller, S. et al. (2012). Runtime Verification with State Estimation. Proc. RV, Springer LNCS 7186, 193-207.

[20] Faul, F. et al. (2007). G*Power 3. Behavior Research Methods, 39(2), 175-191.

[21] Open Robotics. (2023). ROS 2 Humble Hawksbill. https://docs.ros.org/en/humble/

### *Related Works by the Author (Zenodo deposits, same research program)*

[22] Oktenli, B. (2026). ADARA: Adversarial Deception-Aware Risk Architecture (v10.7). Zenodo. https://doi.org/10.5281/zenodo.19043924

[23] Oktenli, B. (2026). MAIVA: Multi-Agent Integrity Verification Architecture (v5.18). Zenodo. https://doi.org/10.5281/zenodo.19015517

[24] Oktenli, B. (2026). FLAME: Flash War Latency Architecture for Multi-Domain Escalation Control (v5.11). Zenodo. https://doi.org/10.5281/zenodo.19015618



Georgetown University · Master of Professional Studies — Applied Intelligence

ORCID: 0009-0001-8573-1667

When citing this work, please use DOI: 10.5281/zenodo.19143190