BLADE-EDGE Governance Simulator

Simulation User Guide

Version: v5.0.3
Document: EDGE-SIM-UG-001
Date: April 2026
Revision: 1.0

Author: Burak Oktenli
Institution: Georgetown University, School of Continuing Studies
Program: M.P.S. Applied Intelligence (STEM)
ORCID: 0009-0001-8573-1667

Related Artifact: DOI: 10.5281/zenodo.19177472

Document Control

Document ID	EDGE-SIM-UG-001
Simulation Version	v5.0.3
Author	Burak Oktenli — Georgetown University — ORCID 0009-0001-8573-1667
Date / Revision	April 2026 / Rev 1.0 — Initial Release
Classification	UNCONTROLLED — Research Artifact
Related Artifact	DOI: 10.5281/zenodo.19177472

Table of Contents

1. Purpose, Scope, and Assumptions 3
2. Quick Start (5-Step) 3
3. System Requirements and Security 4
4. Interface Layout and Navigation 4-5
5. Operating Procedures 5-6
6. Parameter Reference 6-7
7. Scenario Reference 7
8. Metrics, Formulas, and Verification 7-8
9. Data Export and Reproducibility 8-9
10. Limitations and Threat Considerations 9
11. Troubleshooting 9-10
12. Glossary and References 10

1. Purpose, Scope, and Assumptions

This guide provides operating procedures for the BLADE-EDGE (Directed-Energy Weapon Governance) simulation. It implements the complete 9-module governance pipeline (SATA-ADARA-IFF-HMAA-MAIVA-FLAME-CARA-BDA-EFFECTOR) with Hungarian algorithm weapon-target assignment, 3D engagement visualization, 8-factor beam suitability computation, IFF classification, and Monte Carlo campaign analysis. The simulation demonstrates how authority-governed autonomy prevents unauthorized engagement.

Intended Audience: EB2-NIW petition evaluators, defense/aerospace reviewers, academic peers, and technical collaborators seeking independent verification of governance pipeline behavior.

Scope: Operation of the BLADE-EDGE Governance Simulator simulation. Does not cover mathematical theory (see published paper) or hardware specifications (see Blueprint.am).

1.1 Assumptions and Constraints

- The user has a modern browser with JavaScript enabled and HTTPS access.
- All parameter values are synthetic research placeholders, not calibrated against physical hardware.
- The simulation models governance logic only. Physical dynamics (aerodynamics, acoustics, electromagnetics) are simplified.
- Cryptographic operations (SHA-256 audit chain) use the WebCrypto API, not hardware TPM/HSM.
- Results are valid for demonstrating architectural behavior, not for operational safety assessment.

IMPORTANT

This simulation is a research prototype. Not for operational planning, safety-critical decisions, or system certification. All parameters are synthetic.

2. Quick Start

Step 1. Open the simulation HTML file in Chrome over HTTPS.

Step 2. Review the interface panels and identify the main controls.

Step 3. Click RUN or START to begin the simulation loop.

Step 4. Observe the governance pipeline processing sensor inputs through all modules.

Step 5. Use export/download buttons to save session data as JSON for verification.

NOTE

All computation runs client-side. No data leaves your browser. Requires HTTPS (not file://).

3. System Requirements and Security Considerations

Browser	Chrome 90+, Firefox 88+, Safari 15+, Edge 90+
Protocol	HTTPS required (WebCrypto API for SHA-256 audit hashing)
Display	Min 1280x720; recommended 1920x1080+
CPU/Memory	Any modern processor. Monte Carlo (100+ runs): multi-core recommended, ~200MB RAM peak
GPU	WebGL-capable recommended for 3D visualizations (Three.js)
Network	Internet for initial CDN load (~500KB). All computation client-side after load.
Installation	None — zero install, no login, no backend, no database, no cookies

3.1 Security Considerations

• **No data exfiltration:** All computation runs in the browser. No data is sent to any server.

• **CDN dependencies:** React, ReactDOM, and Babel load from cdnjs.cloudflare.com (Cloudflare CDN with SRI hashes where available).

• **Audit integrity:** SHA-256 hash chain via WebCrypto API. Each audit entry links to the previous entry's hash. VERIFY button recomputes the entire chain.

• **No authentication:** The simulation has no login system. All state is ephemeral in browser memory.

4. Interface Layout and Navigation

4.1 Panel Layout

The interface is a React application with 3D engagement visualization. **3D View:** Three.js rendering of engagement zone with threat tracks (colored by IFF classification), effector positions, and beam paths. **Left Panel:** Sensor trust array (Radar, EO/IR, LiDAR, GPS, IMU, Atmospheric), authority gauge, and pipeline status (9 modules: SATA through EFFECTOR). **Right Panel:** Track list with IFF classification, beam suitability scores, WTA assignment, and BDA results. **Controls:** START/STOP, threat injection, Monte Carlo campaign, BYPASS GOVERNANCE toggle.

> **TIP**
> Hover over interface elements for tooltips. Most gauges include ARIA labels for screen reader accessibility.

5. Operating Procedures

5.1 Startup

1. Navigate to the simulation URL or click Launch Simulation from burakoktenli.com.
2. Wait for loading (2-5 seconds). CDN scripts load from cdnjs.cloudflare.com.
3. Verify interface loads completely. All panels should be visible.

5.2 Standard Operation

1. Open the simulation via Launch Simulation or navigate to blade-edge-simulation.html.

2. Click START to begin the engagement simulation. The 3D view shows the engagement zone.

3. Threat tracks automatically appear in the engagement zone. The governance pipeline processes each track.

4. Monitor the pipeline status: each of the 9 modules (SATA -> ADARA -> IFF -> HMAA -> MAIVA -> FLAME -> CARA -> BDA -> EFFECTOR) shows PASS/FAIL/SKIP.

5. The beam suitability score (beta) combines 8 factors: line-of-sight, atmospheric, track stability, dwell feasibility, thermal margin, collateral clearance, target suitability, handoff availability.

6. IFF classification determines Friend/Unknown/Hostile. Friendly tracks are automatically excluded from engagement.

7. The Hungarian algorithm computes optimal weapon-target assignment for multi-target scenarios.

8. Run Monte Carlo campaign (100 trials) for statistical engagement outcome analysis.

9. Toggle BYPASS GOVERNANCE (Unsafe Demo) to see what happens without governance constraints.

5.3 Shutdown

1. Export session data. **2.** Close browser tab (all state discarded).

> **IMPORTANT**
> State is not persisted. Export before closing to preserve results.

6. Parameter Reference

Parameters are pre-configured per scenario. See Section 4 for interface controls and the published paper for parameter derivation.

7. Scenario Reference

Continuous mode. Inject faults manually using controls in Section 5.

8. Metrics, Formulas, and Verification

8.1 Key Metrics

Pipeline Status

9-module status: SATA (sensor trust), ADARA (deception check), IFF (identification), HMAA (authority), MAIVA (consensus), FLAME (deliberation), CARA (recovery), BDA (damage assessment), EFFECTOR (final gate).

Beam Suitability (beta)

8-factor quality score: line-of-sight, atmospheric transmission, track stability, dwell feasibility, thermal margin, collateral clearance, target suitability, handoff availability.

IFF Classification

P(FRIENDLY), P(HOSTILE), P(UNKNOWN) probabilities for each track.

WTA Assignment

Hungarian algorithm optimal weapon-target assignment for multi-track engagement.

Authority Level

HMAA-computed engagement authority. Must exceed threshold for effector activation.

BDA Score

Post-engagement Battle Damage Assessment based on velocity drop and RCS change.

8.2 Verification Checklist

Perform the following checks to verify correct simulation behavior:

Start simulation (RUN/START)	Interface loads. Governance pipeline begins processing.
Observe default state	Authority at nominal level. All pipeline stages PASS.
Inject a fault or attack	Authority reduces proportionally. Affected stage shows FAIL.
Monitor recovery	If CARA active, observe GREP recovery phases.
Export session data (JSON)	File downloads with parameters, history, and audit trail.
Reload and verify reproducibility	Same seed + params = identical outputs.

9. Data Export and Reproducibility

Click export/download to save session JSON with parameters, history, and audit trail.

Verification: 1) Export JSON. 2) Note PRNG seed. 3) Reload with same seed/params. 4) Verify bit-exact match.

9.1 Reproducibility Guarantee

PRNG	Mulberry32 (32-bit seeded)
Math.random()	Zero calls in computation paths
Cross-Browser	Verified: Chrome, Firefox, Safari, Edge
Cross-Platform	Verified: Windows, macOS, Linux
Audit Chain	SHA-256 via WebCrypto (SubtleCrypto API)

10. Limitations and Threat Considerations

Simulation-Only Evidence	Browser-based computation. No physical sensor data or hardware measurements.
Uncalibrated Parameters	All values are synthetic research parameters, not empirically derived.
No Real-Time Guarantees	JavaScript engine provides no timing guarantees for safety-critical operations.

Simulated Cryptography	SHA-256 uses WebCrypto. TPM/HSM operations are modeled, not hardware-backed.
Single-Session State	All state held in memory. Closing the tab discards all data.

10.1 Threat Considerations

• **CDN compromise:** React/Babel load from cdnjs.cloudflare.com. A CDN compromise could inject malicious code. Mitigation: Subresource Integrity (SRI) hashes on script tags where available.

• **Browser extensions:** Malicious extensions could modify simulation DOM/state. Mitigation: test in Incognito mode for clean results.

• **Local modification:** Users can modify simulation code via DevTools. Exported data should be verified against the published source on burakoktenli.com.

11. Troubleshooting

Black screen after loading	React render error or CSP violation	Open F12 Console for error details. Try Chrome Incognito mode.
Simulation runs slowly	CPU-intensive Monte Carlo or 3D rendering	Close other browser tabs. Reduce sample count.
Controls not responding	Browser tab lost focus	Click inside the simulation window. Ensure tab is active.
Export button not working	Pop-up/download blocked	Allow downloads from the simulation domain in browser settings.
Loading screen never completes	CDN scripts blocked by firewall/extension	Disable ad blockers. Allow cdnjs.cloudflare.com.

12. Glossary and References

12.1 Glossary

BLADE-EDGE	BLADE-EDGE (Directed-Energy Weapon Governance)
SATA	Sensor Attestation and Trust Anchoring — trust fusion
HMAA	Human-Machine Authority Architecture — authority computation
CARA	Control Authority Regulation Architecture — recovery protocol
Authority Level	Computed governance authority (0.0-1.0) governing operational actions
PRNG	Pseudo-Random Number Generator — Mulberry32 seeded for reproducibility
Governance Pipeline	Sequential processing chain: SATA -> HMAA -> MAIVA -> FLAME -> CARA

12.2 References

[1] Oktenli, B. (2026). BLADE-EDGE Governance Simulator. DOI: 10.5281/zenodo.19177472.

[2] Oktenli, B. (2026). Simulation artifact. https://burakoktenli.com/blade-edge-simulation.html

[3] Oktenli, B. (2026). Research Portfolio. https://burakoktenli.com

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

[5] NIST (2023). AI Risk Management Framework (AI RMF 1.0).

12.3 Contact

Burak Oktenli — Georgetown University, School of Continuing Studies
Website: **burakoktenli.com |** ORCID: **0009-0001-8573-1667**
For questions about this simulation or the governance architecture research program, use the contact form at burakoktenli.com.

End of Document