

# MAIVA Byzantine Consensus Simulation

## Simulation User Guide

Version: v5.18  
Document: MAIVA-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.19015517](https://doi.org/10.5281/zenodo.19015517)

# Document Control

<b>Document ID</b>	MAIVA-SIM-UG-001
<b>Simulation Version</b>	v5.18
<b>Author</b>	Burak Oktenli — Georgetown University — ORCID 0009-0001-8573-1667
<b>Date / Revision</b>	April 2026 / Rev 1.0 — Initial Release
<b>Classification</b>	UNCONTROLLED — Research Artifact
<b>Related Artifact</b>	DOI: 10.5281/zenodo.19015517

## 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 MAIVA (Multi-Agent Integrity Verification Architecture) simulation. MAIVA implements Byzantine-resilient swarm trust aggregation using trimmed weighted median, CUSUM anomaly detection, graduated action gates, and HMAC-signed peer trust attestation. The simulation visualizes agent roster, NxN trust matrix, network topology, timeline, audit trail, invariant violations, and full tick-level replay.

**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 MAIVA Byzantine Consensus Simulation 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 `maiva-simulation.html` in Chrome over HTTPS.

**Step 2.** Configure agent count and Byzantine fraction ( $f$ ) in the control panel.

**Step 3.** Click RUN to start — agents exchange trust attestations each tick.

**Step 4.** Switch to Matrix tab to see the NxN peer trust matrix with HMAC verification.

**Step 5.** Switch to Violations tab to monitor all 8 formal safety invariants in real time.

### 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 uses a React PDR (Preliminary Design Review) briefing format. **Top Bar:** Simulation controls (Run/Pause/Step), agent count configuration, Byzantine fraction (f) setting, and real-time swarm authority ( $A_{\text{swarm}}$ ) display. **Tab Bar:** Seven tabs — Roster, Matrix, Network, Timeline, Audit, Violations, Replay. **Main Panel:** Changes with selected tab. Roster shows agent list with authority/trust/quarantine status. Matrix shows NxN peer trust grid. Network shows topology graph. Timeline shows authority history chart. Violations shows 8 invariant monitors.

### 4.2 Navigation Tabs

Roster	Agent list with authority, trust, health, quarantine status, and enrollment tokens
Matrix	NxN peer trust matrix with HMAC attestation verification status
Network	Graph visualization of agent network topology with trust-weighted edges
Timeline	Authority and trust score history over simulation ticks
Audit	Sequential audit log with role-based entries and monotonic IDs
Violations	Real-time invariant monitoring (8 safety properties)
Replay	Tick-by-tick replay scrubber with full state reconstruction

### 4.3 Panel Descriptions

**Swarm Authority ( $A_{\text{swarm}}$ ).** Aggregated authority across all non-quarantined agents

**Agent Roster.** Per-agent: ID, authority, trust, CUSUM score, quarantine status

**Trust Matrix.** N x N matrix showing peer-to-peer trust scores with HMAC signatures

**Invariant Monitor.** 8 safety properties checked every tick: bounded authority, action gates, Byzantine tolerance, quorum, temporal decay, partition safety, sequential audit, attestation integrity

**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 maiva-simulation.html.
2. Configure agent count and Byzantine fraction ( $f$ ) using the top control panel.
3. Click RUN to start the simulation. Agents begin exchanging trust attestations.
4. Switch to the Matrix tab to observe the NxN peer trust matrix. Byzantine agents will show divergent trust scores.
5. Switch to the Violations tab to monitor all 8 invariants. Green = holding, Red = violated.
6. Quarantine a suspected Byzantine agent by clicking its entry in the Roster tab. Observe how swarm authority adjusts.
7. Use the Replay tab to scrub through past ticks and inspect historical trust states at any point.
8. Run Monte Carlo to generate statistical analysis of Byzantine detection rates across varied configurations.

### 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

### A\_swarm

Aggregated swarm authority computed from non-quarantined agent authorities using trimmed weighted median (resistant to  $f$  adversaries in  $3f+1$  rosters).

### Peer Trust Matrix

$N \times N$  matrix where  $M[i][j]$  = agent  $i$ 's trust assessment of agent  $j$ . Each entry signed with HMAC-SHA256.

### CUSUM Score

Cumulative sum anomaly detection per agent. Persistent drift in trust scores triggers Byzantine detection when CUSUM exceeds threshold.

### Action Gate Status

Per-action-class (OBSERVE, COMMUNICATE, NAVIGATE, ENGAGE, CRITICAL) permission based on current swarm authority level.

### Byzantine Detection Rate

Percentage of compromised agents correctly identified and quarantined. Target: 100% detection within 20 ticks for  $f \leq \text{floor}((n-1)/3)$ .

### Invariant Status

8 formal safety properties monitored every tick. Any violation triggers alert and is logged to the audit trail.

## 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

MAIVA Byzantine Consensus	MAIVA (Multi-Agent Integrity Verification Architecture)
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). MAIVA Byzantine Consensus Simulation. DOI: 10.5281/zenodo.19015517.
- [2] Oktenli, B. (2026). Simulation artifact. <https://burakoktenli.com/maiva-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](https://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](https://burakoktenli.com).

---

End of Document