

TECHNICAL ASSURANCE REPORT

KEELO

Rogue Fleet Defensible Proof Results

CONTROLLED EXTERNAL PROCESS GOVERNANCE TEST

DEFENSIBLE PROOF PASSED

Primary controlled proof passed. Independent isolated reproduction passed. Deterministic comparison result: **MATCH**.

Report field	Value
Test identifier	KEELO ROGUE FLEET 001
Report date	17 September 2026
Primary implementation commit	[REDACTED internal detail]
Reproduction evidence commit	[REDACTED internal detail]
Classification	Confidential controlled proof evidence

Prepared from the preserved machine generated evidence package and the independent reproduction evidence committed to branch [REDACTED internal reference].

EXECUTIVE CONCLUSION

The result in one page

PASS • REPRODUCED • MATCHED

The defined Rogue Fleet scenario completed without unresolved obligations, unauthorised effective executions, or duplicate effective executions. A clean detached worktree reproduced the deterministic results.

Measure	Primary result	Reproduction
Synthetic agents	10,000	10,000
Submitted commands	100,000	100,000
Effective external executions	10,000	10,000
Unauthorised effective executions	0	0
Duplicate effective executions	0	0
Reconciled commands	100,000	100,000
Unresolved obligations	0	0
Integrity attacks detected	10 of 10	10 of 10
Forced runtime interruptions	3	3
Evidence manifest	PASS	PASS

What this establishes

- Under the recorded deterministic seed and controlled Firestore emulator configuration, the governance path prevented unauthorised and duplicate effective execution.
- Durable state and idempotency controls recovered correctly across three deliberately injected runtime crashes.
- The primary and reproduction runs matched on deterministic metrics, integrity summary, reconciliation, and fixture hashes.

What this does not establish

- This was a synthetic controlled proof, not a production deployment or an unrestricted real world autonomous agent trial.
- Exactly once applies to the tested effective execution contract and receiver observations, not universal distributed exactly once semantics.
- The AUD 250,068,991 figure is synthetic attempted exposure within the scenario; it is not realised loss, revenue, or savings.

METHOD AND SCOPE

Controlled test design

The harness generated a deterministic fleet of 10,000 synthetic agents across 25 organisations. Each agent issued 10 commands, creating 100,000 command attempts across ten adversarial and recovery phases. Production execution was prohibited. Three independent local receiver processes represented procurement, data centre, and cross border billing destinations.

Dimension	Recorded configuration
Environment	Synthetic controlled Firestore emulator
Seed	[REDACTED internal detail]
Fleet	10,000 agents across 25 organisations
Workload	10 commands per agent • 100,000 total
External receivers	Procurement [REDACTED port] • Data centre [REDACTED port] • Cross border billing [REDACTED port]
Initial concurrency	4 workers in the valid phase
Subsequent concurrency	8 workers across adversarial phases
Persistence	Firestore emulator with durable restart recovery

Command phases

Phase	Scenario	Cumulative commands
1	Valid	10,000
2	Duplicate	20,000
3	Retry	30,000
4	Unauthorised	40,000
5	Revoked	50,000
6	Stale version	60,000
7	Concurrency collision	70,000
8	Replay	80,000
9	Cross organisation	90,000
10	Evidence failure	100,000

GOVERNANCE OUTCOME

Authority and execution controls

Control observation	Count	Result
Authorised commands	40,000	Governed
Unauthorised commands	20,000	Blocked from effective execution
Authority boundary violations	20,000	Detected
Stale authority attempts	10,000	Rejected
Replay attempts	10,000	10,000 rejected
Cross organisation attempts	10,000	10,000 rejected
Evidence provenance failures	10,000	Prevented effective execution
External execution attempts	40,000	Governance evaluated
Effective external executions	10,000	All authorised and receipted
External executions blocked	90,000	Blocked or suppressed

Exactly once result

ZERO DUPLICATE EFFECTIVE EXECUTIONS
The receivers observed 10,000 unique effective executions and zero duplicate effective executions. All 100,000 submitted commands reconciled to a terminal governed outcome.

Reconciliation field	Recorded result
Governance decisions	100,000
Receiver observed effective executions	10,000
Execution receipts	10,000
Verified outcomes	10,000
Unresolved obligations	0
All commands reconciled	true
All receipts retained	true
Exactly once result	true

RESILIENCE

Crash recovery and persistence

The harness deliberately terminated the runtime in three distinct execution windows. Each affected command recovered to VERIFIED with the same execution identity. Receiver processes survived the runtime terminations.

Crash window	Command	Recovery observation
After intent	command 2500 0	Recovered to VERIFIED without duplicate execution
After receiver	command 5000 0	Recovered receipt and preserved execution identity
After receipt	command 7500 0	Recovered outcome and retained receipt

Persistence assertion	Result
Governed state survived restart	PASS
Authority state survived restart	PASS
Idempotency state survived restart	PASS
Execution state survived restart	PASS
Receipts retained	PASS
Unresolved obligations after recovery	0

Operational nuance

The logs record transient Firestore transaction lock timeouts. The runtime retried these transient failures and completed the proof. The verification suite also intentionally generated PERMISSION DENIED messages while proving that direct client writes and cross organisation reads were rejected. These messages are expected negative test evidence, not final test failures.

INTEGRITY

Evidence tamper detection

Ten controlled tamper scenarios were attempted. Each was detected by the recorded evidence integrity mechanism. In this proof, the attacks targeted the configuration artifact and detection was based on SHA 256 mismatch recomputation.

Attack type	Detection
Modify evidence	DETECTED
Modify decision	DETECTED
Substitute receipt	DETECTED
Delete event	DETECTED
Reorder events	DETECTED
Replay instruction	DETECTED
Fabricate authority	DETECTED
Substitute organisation	DETECTED
Modify payload	DETECTED
Alter outcome	DETECTED

10 OF 10 DETECTED

No attack in the defined integrity suite remained undetected. This is evidence for the specified manifest based controls, not a claim that every possible form of tampering is impossible.

PERFORMANCE

Measured execution profile

Metric	Primary	Reproduction
Runtime	817,287 ms • 13m 37.287s	1,055,272 ms • 17m 35.272s
Command throughput	122.356 per second	94.762 per second
Median latency p50	9.579 ms	14.966 ms
Latency p95	137.079 ms	172.655 ms
Latency p99	189.659 ms	341.488 ms
Sample size	100,000 commands	100,000 commands

Performance was explicitly classified as non deterministic and expected to vary. The comparison excluded runtime and latency fields while requiring deterministic governance, integrity, reconciliation, and fixture evidence to match.

Primary latency range

Statistic	Value
Minimum	2.194 ms
Mean	43.718 ms
Maximum	60,446.235 ms

The wall clock timing covers the first HTTP attempt through the terminal response and includes crash recovery and replay handling.

INDEPENDENT REPRODUCTION

Fresh checkout reproduced the proof

A detached Git worktree was created at the committed proof implementation, dependencies were installed, and the full proof launcher ran with a separate artifact directory. The reproduction completed all verification gates and evidence manifest checks.

Reproduction field	Recorded value
Implementation commit	[REDACTED internal detail]
Worktree mode	Detached HEAD
Comparison timestamp	2026 09 17T13 17 31.215Z
Same deterministic metrics	true
Same integrity summary	true
Same reconciliation	true
Same fixture hashes	true
Performance expected to vary	true
Comparison result	MATCH

REPRODUCTION MATCH

The independent run reproduced the deterministic proof outcome. The original pre reproduction label in [REDACTED source reference] is therefore superseded for reporting purposes by the later MATCH comparison and preserved reproduction evidence.

VERIFICATION

Software and evidence gates

Gate	Primary	Reproduction
TypeScript build	PASS	PASS
Type checking	PASS	PASS
Integration type checking	PASS	PASS
Architecture boundary check	PASS	PASS
General test suite	64 files • 573 tests passed	64 files • 573 tests passed
Firestore emulator suite	2 files • 20 tests passed	2 files • 20 tests passed
Git diff whitespace check	PASS	PASS
Evidence manifest check	PASS	PASS

[REDACTED internal reproduction instructions]

[REDACTED internal reproduction instructions]

Repository provenance

Commit	[REDACTED internal detail]
[REDACTED integrity identifier]	Complete controlled Rogue Fleet execution and proof harness
[REDACTED integrity identifier]	Preserve independent reproduction evidence

The supplied evidence establishes the local commit identities. This report does not independently confirm that the branch was pushed to or is accessible from a remote repository.

CLAIM BOUNDARY

Defensible statement of result

APPROVED RESULT STATEMENT

Keelo passed the KEELO ROGUE FLEET 001 controlled external process governance proof. Across 10,000 synthetic agents and 100,000 commands, the test produced 10,000 authorised effective external executions, zero unauthorised effective executions, zero duplicate effective executions, complete reconciliation, successful recovery from three forced runtime interruptions, and detection of all ten defined integrity attacks. A clean detached reproduction matched the deterministic evidence.

Required qualifiers

- Synthetic, deterministic, emulator based environment.
- The receiver endpoints were independent local controlled processes, not live production counterparties.
- The proof supports the tested governance and effective execution contract only.
- Results are traceable to the recorded seed, implementation commit, artifacts, and SHA 256 manifests.

Final classification

Assessment layer	Classification
Primary run at time of generation	CONTROLLED PROOF PASSED • REPRODUCTION OUTSTANDING
Independent reproduction	PASS
Deterministic comparison	MATCH
Final report classification	DEFENSIBLE PROOF PASSED

Technical appendices redacted

[REDACTED raw terminal output, reproduction instructions and internal evidence locations]