The CineNexus Pro Case Study on heain-core

Whole-System Validation of the HEAIN Framework through an 11-Node Simulation, 25 Test Scenarios and a 24-Hour Soak under Random Faults

Jakkrit Boonma (จักรกฤษ บุญมา)

Independent Researcher · ORCID: 0009-0000-7165-3787 · jaggid.boonma@yahoo.co.th

Date: 12 October 2026

Implementation: heain-core v1.3 (commit 53f398e, Go 1.26), HP Z840 Workstation

Repository: github.com/JakkritB/heain-core (casestudies/cinenexus/)

Abstract

The HEAIN paper series has proposed a fractal architecture that joins the cloud and Edge AI [1][2], specified it as a protocol [3][4], proven its mechanisms one at a time on real hardware [5][6][9][10], and laid down a theory of structural immunity [7][8]. Mechanism-level tests, however, do not show that all the mechanisms work together in an organization with many sites, sensitive data and an unreliable network. This paper reports the first case study of the HEAIN Framework: CineNexus Pro, a hypothetical film post-production company with a central machine room, five studios connected by VPN, and a customer portal in the cloud. The author turned the company's node map into an 11-node simulation that runs the real heain-core in containers on an HP Z840, and broke the requirements down into 25 test scenarios (T1–T25). The scenarios cover data sovereignty, survival when the Master is lost, work while the network is cut, four-eyes governance, PDPA erasure, active security, live upgrades and long-run endurance. In the final run all 25 scenarios passed all 244 checks, and the system ran for 24 hours under 70 random faults, processing 216,080 jobs with none lost or run twice, flat memory on every node, and an audit chain that verified on every node. The case study also found problems that paper design and per-mechanism tests could not see, and led to nine fixes to heain-core. All results can be reproduced from the scripts published in the repository.

Keywords: *HEAIN Framework, heain-core, Case Study, Data Sovereignty, Network Partition, Failover, Soak Test, Chaos Testing, Film Post-Production, Edge AI*

1. Introduction

1.1 Background

The HEAIN Framework grew through eight stages: from the idea that the cloud and the edge are one fractal tree, to the theory of the Hybrid Edge AI Node [1][2], the HEAIN Core Protocol specification [3][4], the implementation in heain-core v1.1 [5][6], multi-level succession in v1.2 [9], the Triad of Structural Immunities [7][8], standalone survival and self-governance in v1.3 [10], and finally the HEAIN Framework as a whole [11]. Every stage so far proved its mechanisms one at a time, on small topologies built for

each mechanism. That does not answer the question real users and funders care about most: when all the mechanisms run at once in a real organization, does the system still keep every promise?

This paper answers that question with the first case study of the HEAIN Framework. The author chose film post-production because it gathers the hardest problems of distributed systems in one place: large customer source files that must not leave the studio, heavy work that needs GPUs, many sites joined by a network that can drop, and legal requirements on personal data.

1.2 Status of the case study

> CineNexus Pro is a hypothetical company. All of its details come from a node map designed by the author; it is not a real company, and no customer data of any kind was used in the tests. Every node runs as a container on a single machine, and application work is simulated. These limits are explained in Sections 4.3 and 9.

1.3 Research questions

- **RQ1 Data sovereignty.** Does each studio's raw data stay in the studio at all times, even when the studio is busy and the machine room is idle?
- **RQ2 Survival.** Does the system keep working when the Master is lost, when a studio is cut off, or when every studio is cut off at once, and does it rejoin with no work lost or duplicated?
- **RQ3 Governance and security.** Do important decisions really require two people, and does the system resist stolen certificates and unauthorized machines?
- **RQ4 Endurance.** Does the system keep all of its promises over a long run under load and random faults?

1.4 Contributions

1. A translation of a multi-site organization's node map into an 11-node simulation running the real heain-core, with a mapping of the organization's requirements onto HEAIN mechanisms (Section 3).
2. A set of 25 test scenarios and a method that checks evidence the system records itself (Section 4).
3. Results that pass all 244 checks, performance results, and a 24-hour soak under 70 random faults (Sections 5 and 6).
4. A record of the nine heain-core fixes the case study found, and lessons for building distributed systems (Sections 7 and 8).
5. Limitations and open items before the Alpha release, and how to reproduce every result (Sections 9 and 10).

2. Background

The HEAIN Framework [11] arranges every node as a fractal tree; every tier above Worker is a Master that differs only in scope. Work follows a seven-phase protocol: P1 intake and encryption, P2 allocation

and leasing, P3 in-memory processing, P4 bounded result buffering, P5 the four-eyes policy gate, P6 role change when a parent fails, and P7 sovereignty-respecting knowledge distribution. Automatic survival is governed by three mechanisms: Role Ceiling, Duty Profile and Escalation Saturation [7][8]. heain-core v1.2 added multi-level succession [9], and v1.3 added automatic role return, witness-based partition decisions, STANDALONE mode with a signed journal, a two-tier Configuration API, and the data lifecycle [10].

Testing distributed systems. Unit tests and per-mechanism tests often miss problems that come from unusual orderings of events, long runs and real load. Chaos engineering [14] therefore injects random faults into a running system and checks that it still keeps the properties that matter. This case study applies the same idea in the 24-hour soak (T25), checking evidence the system records itself rather than API responses alone.

3. The CineNexus Pro Case

3.1 Organization and requirements

CineNexus Pro provides film post-production services. It has one central machine room and five studios, SB, N1N, OC, KT and Film Archive, which connect to the machine room over VPN (WireGuard) with Cloudflare Mesh as a backup path. A Web Portal on Cloudflare lets member customers submit work. Four requirements make this case hard.

1. **Sensitive data.** Customer source files (RAW), such as footage, audio and film scans, may not leave the studio unless an approver releases each job, even when the studio is busy and the machine room is idle.
2. **An unreliable network.** A studio's VPN can drop; the studio must keep working and merge its results back without losing work.
3. **Critical machines can fail.** When the Master is lost, a machine-room node must take over, but studios must never take authority themselves.
4. **Law and oversight.** PDPA erasure [15], legal hold, licensing, live upgrades, backup and restore, and protection against stolen certificates.

3.2 Node map

The map has 11 HEAIN nodes in three roles. One Master acts as both Primary and Zone Master. Four Worker Farm nodes in the same room run portal jobs and stand ready to take over as Master. Six studio Workers (one per studio, two for Film Archive) run only their own studio's work. Governance has a single zone, the Master, but each studio carries its own origin_zone label so that P2 and P7 keep RAW files inside the studio they came from.

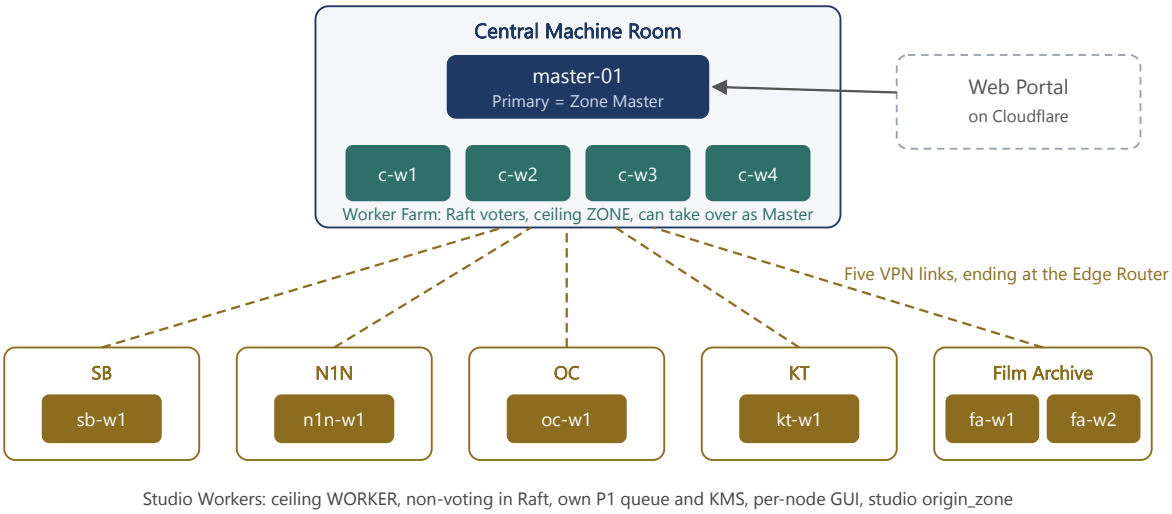


Figure 1 — The CineNexus Pro node map used in the simulation

Node	Role	Ceiling	Raft	On failure
master-01	Primary = Zone Master	none	voter	Down: P6 promotes a Farm node; the role returns once the Master is stable again
c-w1 to c-w4	Worker Farm, Master successors	ZONE	voter	Promoted to Master: Duty Profile hands GPU work to other Farm nodes
sb-w1, n1n-w1, oc-w1, kt-w1	Studio Worker	WORKER	non-voter	Link lost: STANDALONE and journal; never stands for Master
fa-w1, fa-w2	Studio Worker (pair)	WORKER	non-voter	One down: the other carries the zone's work; both cut off: STANDALONE and journal

Table 1 — Node register and duties

3.3 Data policy

Every studio's raw files live only on that studio's machines. Five kinds of data may cross between a studio and the Master; everything else stays in the studio.

Data	Direction	HEAIN mechanism	Condition
Job commands	Master → studio	P2 lease and Envelope	No raw files; states only the job, node and policy
Reports, status, audit and journal	Studio → Master	Audit chain and Escalation Saturation	Results, timing, resource use, and the journal sent on reconnect

Learned knowledge	Studio → Master → all zones	P7 Sanitizer and P5 KNOWLEDGE_UPDATE	DISTILLED form only; customer data, raw file names and node IDs removed
Approved policies and settings	Master → studio GUI	Config API 2a/2b via P5	The GUI keeps using them while the link is down
Approved model versions	Model Store → studio	Policy setting 2b via P5	Changing versions needs an approver, because it changes customer output
Raw files, outputs, fine-tuned models, embeddings and studio job keys	Never leave	P3, P7 RAW stays in zone, EmbeddingGuard, KMS (I1b)	Studio only

Table 2 — Data that may and may not leave a studio

3.4 Mapping the requirements onto HEAIN mechanisms

- **Data sovereignty.** Every studio node is granted zone:<id> through P5 and runs with p2.zone_affinity enabled. A job tagged with an origin zone is leased only to nodes of that zone; a job that waits longer than p2.zone_wait is raised to P5 under the DATA_LOCALITY category [10].
- **Survival.** A Role Ceiling of WORKER keeps studios from taking authority; the Master's escrow is delivered only to the Worker Farm (I1b). P6 promotes a Farm node and returns the role automatically when the Master comes back [9][10].
- **Network cuts.** Witnesses decide that a studio has been cut off; the studio enters STANDALONE, keeps a signed journal, and reconciles under a per-zone rate limit [10].
- **Governance.** Policy settings, cross-zone data release, erasure, certificate revocation, rollouts and role-ceiling raises all pass through P5, where the proposer cannot approve their own request [3][7].
- **Security.** The node plane accepts only node certificates admitted through P5, Raft runs over mTLS, and revocation takes effect at the TLS handshake [10].

4. Method

4.1 Environment

Every scenario runs against the real heain-core on 11 nodes, as containers from the same image on a single HP Z840. Each studio's network is a separate Docker network; "cutting the VPN" means detaching the node from that network. Machine-room nodes sit on every studio network behind a simulated Edge Router. The platform's real applications run in the network namespace of the node they belong to, because the core hands keys and jobs only to applications on the same node. The 11 nodes together use about 272 MiB of memory.

4.2 Procedure

1. **Break the requirements into scenarios.** The author broke the requirements into 25 scenarios. T1–T15 come from the original node map; T16–T25 were added on 9 October 2026 and cover performance, upgrades, licensing, PDPA, active security, backup and restore, resource exhaustion, container death and the soak.
2. **Eight rounds, each from zero.** Each round tests several scenarios, starts from empty data, and configures zones and applications through P5 as a real installation would.
3. **Check evidence.** Checks read evidence the system records itself, such as each node's audit, network captures and searches of files on disk, and never trust API responses alone.
4. **A new round must not break an old one.** Every heain-core fix reruns all earlier rounds, together with 19 real-system regression suites and the unit tests of every package.
5. **Commit only when everything passes.** Each step's script stops at the first failed check, and every push is confirmed with git ls-remote.

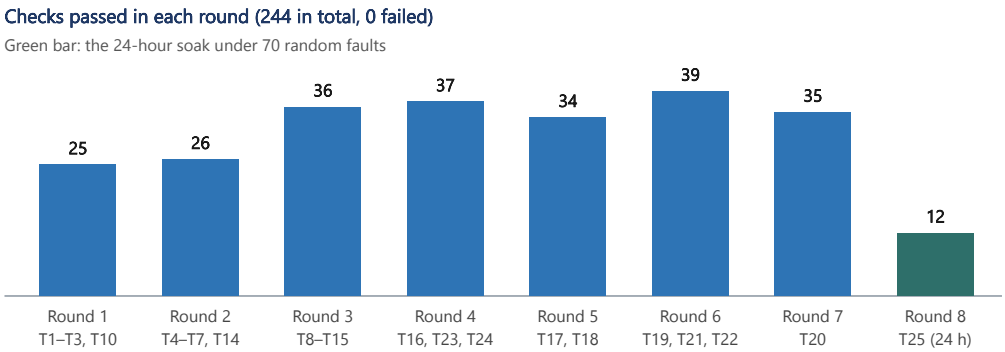


Figure 2 — The eight test rounds and the checks passed in the final run

4.3 Limits of the method

All nodes run on one machine and share one clock, the network has none of the real latency of a WAN, and application work is simulated with delays rather than real rendering. The results therefore confirm that the system behaves correctly, but they are not yet performance results on a real network.

5. Results by Scenario

All 25 scenarios passed in the final run on 11–12 October 2026, on heain-core commit 53f398e: 244 checks with none failed. The table is ordered by scenario number; the evidence column is what the scripts actually checked.

Test	Subject	Round	Evidence that passed
T1	Studio work stays in the studio	1	The Master never saw the ticket; no RAW on any disk, including sb-w1; network capture found no RAW

T2	Studio busy, machine room idle	1	The job was not sent out of the zone; after the wait it rose to P5; the approver declined and the job waited in the studio
T3	Portal work runs only on the Farm	1	No studio received work; the result could be read once, then the key was destroyed
T4	Master lost	2	c-w1 took over in about 20 s; the 6 studio nodes stopped at the WORKER ceiling
T5	Master returns	2	The role returned after a stable period of about 50 s; at no point did two Masters serve the same work
T6	Escrow to the Farm only	2	Escrow delivered to c-w1 to c-w4; studios that asked were refused
T7	KT's VPN cut	2	KT entered STANDALONE and kept working; a journal of 12–14 entries merged in full and in order
T8	Every studio cut at once	3	787 entries merged in 51 s; the Master rate-limited with 429 within the token-bucket bound
T9	Admin widens a P7 rule twice	3	The rule went Overdue for review; the admin could not acknowledge it, the approver could
T10	App certificate on the Admin API	1	Refused with 403 and recorded as admin.denied without flooding the audit
T11	New technique from Film Archive	3	Held at P5; other studios received only DISTILLED form, with no customer IDs or file paths
T12	Raise fa-w1 above its ceiling	3	Held at P5; the admin could not self-approve; once declined, it stayed WORKER
T13	Cloudflare down	3	The portal could not submit work, but studios kept working over VPN
T14	Jobs in flight when the Master is lost	2	SB's job ran in SB; the pending portal job was taken over and run by c-w1
T15	OC cut while using the GUI	3	Resubmitting with the same Idempotency-Key returned the same job; synced once after reconnect
T16	Performance	4	Every job ran once; the core added p95 219 ms (Farm) and 273 ms (studio)
T17	Live upgrade	5	11 nodes in 477 s, one at a time; a broken release was stopped at the canary; 1,124 jobs not lost
T18	Licensing	5	One node short limited only the newest node; after expiry, accepted work still finished
T19	PDPA erasure	6	Approved once, erased on all 11 nodes with key fingerprints; legal hold wins

T20	Active security	7	Certificate revocation took effect within 5 s; an unlisted machine was refused; Raft over mTLS
T21	Break-glass	6	Usable only when cut off, on the node itself; logged every time; acknowledged by the approver
T22	Backup and restore	6	Encrypted in-zone backup; after restore the node rejoined the Farm in 3 s; erased data did not return
T23	Resource exhaustion	4	Disk full: every change refused until the audit could be written; jobs with no runnable node raised an alert
T24	Container death	4	The lease expired in 21 s and another node ran the job; the result was delivered once
T25	24-hour soak	8	216,080 jobs, none lost or duplicated; 70 faults; flat memory

Table 3 — Results of scenarios T1–T25 in the final run

6. Performance and the 24-Hour Soak

6.1 Performance (T16)

T16 measured 64 KB jobs with 50 ms of simulated processing. With one job at a time, the core adds less than 0.3 s at p95. With many concurrent jobs, throughput is bounded by the Master's CPU.

Path	Single job: end-to-end p50 / p95 (ms)	Core overhead p95 (ms)	Concurrent (jobs/s)
Portal → Master → Farm	254 / 269	219	41.3 (theoretical ceiling 320)
Studio GUI → studio	258 / 323	273	49.0 (theoretical ceiling 80)

Table 4 — Core performance in the simulation

6.2 The 24-hour soak (T25)

From 11 to 12 October 2026, the portal at the Master and the Studio GUIs at sb-w1 and kt-w1 submitted work without pause. All four Farm nodes and the QC applications of both studios ran the jobs, and each GUI wrote an edit to the journal every 10 seconds. Meanwhile a script injected 70 random faults.

Aspect	Result
Jobs	Portal 138,644 jobs (p50 231 ms, p99 345 ms); studio SB 38,725 and KT 38,711 jobs; 216,080 in total, every one delivered once and run once
Journal	8,635 GUI entries per studio, every one acknowledged by the Master
70 faults	36 studio network cuts, 16 Farm node restarts, 10 application kills and 8 fast link flaps; no Master change; every STANDALONE entry matched a real cut
Resources	Memory, heap, goroutines and open files flat on all 11 nodes; on the Master, goroutines 57 → 56 and open files 57 → 56
Data growth	Master +64 MB per hour, of which audit +51 MB (about 1.2 GB per day); 754,243 audit records; a full chain check took at most about 7.7 s
Audit chain	Verified on every node

Table 5 — Results of the 24-hour soak under random faults

The first 24-hour run (10–11 October 2026) met every criterion that is the core's responsibility: 138,428 portal jobs and 77,460 studio jobs, none lost or run twice, flat memory, and no Master change. It failed in the final minutes because the simulation never confirmed its application certificates, so the core revoked them when the 24-hour probation period ended, exactly as designed. The test harness was then changed to confirm certificates the way heain-sdk does, and the rerun passed in full.

7. What heain-core Fixed Because of the Case Study

The case study led to nine fixes to heain-core, each caused by a problem a scenario actually found. The table is ordered from the most recent fix.

Round (tests)	Commit	What was fixed
8 (T25)	53f398e	Escrow no longer scans every job ever held; input dropped once a job completes; incremental audit check without locking writes (full check every 6 hours)
7 (T20)	90ecc88	The node plane accepts only node certificates; the admitted-node list and revocation go through P5; Raft over mTLS; application uses enforced
6 (T19, T21, T22)	c74e00c	Erasure by data subject through P5; legal hold beats erasure; per-node encrypted backup and restore
5 (T17, T18)	a434ee0	Rollout skips cut-off nodes and upgrades them on return; fixed a connection leak that exhausted the Master's file descriptors
4 (T16, T23, T24)	3dcfbdf	Audit that cannot be written means every change is refused; minimum disk floor; active-job index (lease and result fell from about 80 to about 6.5 ms per job)
3 (T8–T15)	7e398d9	P7 relayed through the Global Primary; closed a path by which application certificates could push knowledge into P7; role-ceiling raises go through P5

2 (T4–T7, T14)	73bc4be	The Farm follows the new slot holder when the Master is lost; ceiling-bound nodes watch their parent's holder
1 (T1–T3, T10)	01386af	Refused Admin API calls recorded in the audit
Before round 1	787c9de	P2 zone affinity (work never leaves its zone); per-zone journal intake rate limit

Table 6 — heain-core fixes arising from the case study

8. Discussion

8.1 Answers to the research questions

RQ1. Studio raw data stayed in the studio throughout, confirmed by three independent kinds of evidence: the Master's audit never saw the ticket, there was no RAW on any disk, and network capture found no RAW (T1). Even with the machine room idle, work was not sent out of the zone without an approver (T2). **RQ2.** The system kept working under every kind of failure tested: Master loss (T4, T14), one studio cut off (T7, T15), every studio cut off at once (T8) and container death (T24). It rejoined with no job or journal entry lost or duplicated. **RQ3.** Every important decision tested really did require two people, and the proposer could not approve their own request (T2, T9, T12, T19). Stolen certificates and unlisted machines were refused at the TLS handshake (T20). **RQ4.** The system kept all of its promises throughout 24 hours under 70 faults, without memory growth (T25).

8.2 Lessons

- **The most serious problems appear only in long runs under load.** A connection leak that exhausted the Master's file descriptors, and escrow that slowed with the whole history of jobs, did not appear in unit tests or per-mechanism tests. A soak is therefore a required part of the proof, not an option.
- **Testing as an attacker finds what a normal user does not.** Every kind of certificate is issued by the same CA, so checking that a certificate chains to the CA is not enough; planes must be separated by certificate type, and nodes admitted through P5.
- **A system that follows its design can still fail because a user skipped a step.** The first 24-hour run failed because the harness never confirmed its application certificates, while the core behaved exactly as designed. Documentation and the SDK must make mandatory steps hard to miss.
- **Every round made the core stronger.** Each round found a weakness or limit that paper design could not see, and every fix passed a rerun of all earlier rounds.

8.3 Threats to validity

Internal. The test scripts and the simulation were written by the developer, so they may not cover cases nobody anticipated. **External.** All nodes run on one machine with one clock, and the network is virtual, so the results say nothing about real latency, packet loss or clock skew. **Measurement.** Performance figures come from simulated delay jobs and a single run per configuration, so they are not yet repeated quantitative measurements with distributions.

9. Limitations and Open Items before Alpha

- **It is a simulation.** All 11 nodes run as containers on one machine; VPN and Cloudflare are simulated with Docker networks. It has not yet run at real studios or on a real network.
- **Audit grows with workload.** About 1.2 GB per day at this load, and there is no retention policy yet. Pruning or archiving old segments must be designed so that the hash chain can still be verified.
- **Job records are kept forever.** Input is now dropped, but job records still accumulate.
- **Security defaults are still loose.** Without security.fleet_nodes, any node with a certificate from the CA can join; apps.enforce_uses is off by default; and a cut-off node learns of certificate revocations only when it reconnects. The Alpha release must decide which of these to turn on by default.
- **The Master is a CPU bottleneck** at about 41–49 jobs per second. That is enough for CineNexus Pro, but larger loads and multiple Masters have not been tested.
- **Base applications must be retested.** Applications that declare uses should run their test suites against this core, because enforcing uses changes earlier behavior.

The next step is to close the open items above and then run the next case study, the southern border provinces, which tests different requirements: human rights, independent approvers and data lanes that must never be linked, before the Alpha release [11].

10. Evidence and Reproducibility

Every core commit has been confirmed with git ls-remote on every remote; the final core commit is 53f398e. The test scripts are in the repository under casestudies/cinenexus/: lib.sh, sim.sh and round1.sh to round8.sh. Each round commits only when every check passes, and round 8 is locked with flock to prevent overlapping runs.

To reproduce, check out heain-core commit 53f398e, run casestudies/cinenexus/sim.sh up, then run each round's script. The test machine needs only Go and Docker.

Result file on the test machine	Contents
/mnt/xfs_nvme/heain-cn0/sim	The 11-node simulation environment (label heain.case=cn0)
~/cn9.out, ~/cn10.out, ~/cn11.out	Output of rounds 6, 7 and 8 with the regression suites
~/cn11-soak.json	Summary of the 24-hour soak
~/cn11-soak-nodes.txt	Per-node resource readings throughout the soak
~/cn11-soak-events.log	The 70 injected faults
~/cn11-perf.json	T16 performance results

Table 7 — Evidence files from the tests

11. Conclusion

The CineNexus Pro case study is the first proof that the HEAIN Framework works as a whole system in an organization with many sites, sensitive data and an unreliable network. heain-core passed all 25 scenarios and 244 checks on an 11-node simulation, and ran for 24 hours under 70 random faults without losing or duplicating any of 216,080 jobs. Raw data never left its studio, cut-off studios kept working and rejoined in full, a Farm node took over a lost Master within about 20 seconds, and every important decision required two people. Just as important as the passing results, the case study found problems that paper design and per-mechanism tests could not see, and made the core stronger through nine fixes. All results are reproducible, and they form the basis for the public-sector case studies to follow, which share the same requirements: sensitive data, many sites and an unreliable network.

References

[1] Boonma, J. "Hybrid Edge AI Node: A Fractal Governance Architecture for Sovereignty-Aware Distributed AI Processing." Zenodo, 2026. DOI: 10.5281/zenodo.22851257

[2] Boonma, J. "Hybrid Edge AI Node: สถาปัตยกรรมการกำกับดูแลแบบยืดหยุ่นสำหรับการประมวลผล AI แบบกระจายศูนย์ที่เคารพอธิปไตยข้อมูล." (Thai edition) Zenodo, 2026. DOI: 10.5281/zenodo.22851587

[3] Boonma, J. "HEAIN Core Protocol: A Normative Specification for Collective Intelligence in Fractal Edge AI Governance." Zenodo, 2026. DOI: 10.5281/zenodo.22858845

[4] Boonma, J. "HEAIN Core Protocol: ข้อกำหนดเชิงบรรทัดฐานสำหรับปัญญาร่วมในการกำกับดูแล AI แบบยืดหยุ่นที่ขอบเครือข่าย." (Thai edition) Zenodo, 2026. DOI: 10.5281/zenodo.22858994

[5] Boonma, J. "The Intelligent Network of the Future — Empirical Proof of the HEAIN Core Protocol (HEAIN-CORE-1.1)." Zenodo, 2026. DOI: 10.5281/zenodo.23060556

[6] Boonma, J. "โครงข่ายอัจฉริยะแห่งอนาคต บทพิสูจน์เชิงประจักษ์ของ HEAIN Core Protocol (HEAIN-CORE-1.1)." (Thai edition) Zenodo, 2026. DOI: 10.5281/zenodo.23061474

[7] Boonma, J. "The Triad of Structural Immunities: A Theoretical Framework for Preventing Cascading Failures, Authority Creep, and Alert Storms in Fractal Edge AI." Zenodo, 2026. DOI: 10.5281/zenodo.23172868

[8] Boonma, J. "สามประสานภูมิคุ้มกันระดับสถาปัตยกรรม: กรอบทฤษฎีเพื่อป้องกันความล้มเหลวแบบลูกโซ่ การลุกลามของสิทธิ์ และ Alert Storm ในปัญญาประดิษฐ์ที่ขอบเครือข่ายแบบแฟร็กทัล." (Thai edition) Zenodo, 2026. DOI: 10.5281/zenodo.23172979

[9] Boonma, J. "Multi-Level Role Succession: HEAIN Core Protocol 1.2 (HEAIN-CORE-1.2)." 2026 (companion paper).

[10] Boonma, J. "Standalone Survival and Self-Governance: HEAIN Core Protocol 1.3 (HEAIN-CORE-1.3)." 2026 (companion paper).

[11] Boonma, J. "HEAIN Framework: An Infinitely Scalable Fractal Intelligent Network Seamlessly Linking Cloud and Edge AI, Resilient to Failure and Multi-Dimensional Fraud." 2026 (companion paper).

[12] Boonma, J. "HEAIN-SD-1.0: Policy-Governed Subtitling and Dubbing for Thai on Hybrid Edge AI Nodes." Zenodo, 2026. DOI: 10.5281/zenodo.23276542 (Thai edition: DOI: 10.5281/zenodo.23276891)

[13] Boonma, J. "design-notes/n-tier-generalization.md", "docs/progress-log.md" and "casestudies/cinenexus/REPORT.md." heain-core repository, 2026.

[14] Basiri, A., Behnam, N., de Rooij, R., Hochstein, L., Kosewski, L., Reynolds, J., and Rosenthal, C. "Chaos Engineering." IEEE Software, 33(3), 2016, pp. 35–41.

[15] Personal Data Protection Act, B.E. 2562 (2019), Thailand.

Author and copyright holder: Mr. Jakkrit Boonma (Independent Researcher) | ORCID: 0009-0000-7165-3787 |
Email: jaggid.boonma@yahoo.co.th

© 2026 Jakkrit Boonma. All rights reserved. Ownership and copyright in this document belong to Mr. Jakkrit Boonma from the date of publication, under applicable copyright law.