Automated Verification and Validation Framework with Digital Twin Environments

Zach Kelling — Hanzo Team

Version 1.0 — February 3, 2026 February 2026

Abstract

We present an integrated verification and validation (V&V) framework combining continuous observability, behavioral analytics, ML pipeline orchestration, digital twin environments, and formal cryptographic proofs. The observability platform ingests logs, metrics, and distributed traces into a ClickHouse OLAP backend, with LLM-specific telemetry tracking token usage, cost, latency, and quality metrics across 100+ AI providers. Temporal workflows orchestrate LLM evaluation, trace clustering, and sentiment analysis, while Dagster DAGs manage ETL pipelines for training data preparation. A network simulation environment supports configurable topologies with 38 registered virtual machines, enabling mission rehearsal and system qualification. The framework includes 17 formal cryptographic proofs, 28+ consensus tests, and automated CI/CD with multi-platform builds. Anomaly detection, feature flag-controlled rollout, and A/B testing with statistical significance provide continuous data-driven decision support.

Executive Summary. This is the test, evaluation, and monitoring layer that tells operators whether a complex AI and cryptographic system actually works—before it is fielded and continuously once it is. Three capabilities work together: full instrumentation, so every action the system takes is recorded and can be inspected after the fact; a high-fidelity "digital twin" that replicates the real deployment, so new builds and stressful conditions (jamming, link loss, hostile inputs) can be rehearsed safely before they reach the field; and automated checks—including machine-checked cryptographic proofs—that run on every change and block anything that regresses. For a program office, this is the qualification and continuous-assurance evidence a system needs to earn and keep an authorization to operate, with specialized tracking for AI workloads (cost, latency, output quality) that conventional IT monitoring does not capture. The capability is built to run where the system runs, including disconnected at the edge with store-and-forward when connectivity returns.

Contents

1 Introduction

Modern naval systems integrate AI/ML, post-quantum cryptography, and distributed computing at unprecedented scale. Traditional V&V approaches—manual test plans, periodic audits, and waterfall certification—cannot keep pace with continuous deployment and evolving threats.

This paper presents an automated V&V framework built on three principles:

1. **Continuous observability**: Every system interaction is instrumented, traced, and queryable.
2. **Automated evaluation**: ML models, cryptographic implementations, and consensus protocols are continuously verified against formal specifications.
3. **Digital twin fidelity**: System qualification occurs in high-fidelity simulated environments before live deployment.

1.1 Strategic Context

A verification and validation framework is only as credible as the systems it has actually carried into production under real scrutiny. Hanzo (American-owned, Techstars '17, founded 2014) is an applied AI lab and venture studio behind 100+ venture-funded startups and multiple unicorns; its founder is a recognized expert in *both* AI/ML and cryptography, and the company builds both the *Zen* frontier model family and the post-quantum cryptographic stack it runs on. The same agentic engineering stack described in these papers delivered a securities trading platform—a full ATS/BD/TA stack consolidated from 200+ microservices into 3 compiled binaries—through FINRA/SEC approval and SOC2 audit into production, in record time with a small team. That is the operative proof point for test and evaluation: regulated, externally audited software shipped at scale, the same discipline a defense accreditation demands. It also anchors a broader position. With American frontier AI consolidated into a closed two-vendor duopoly and the open frontier ecosystem now largely offshore, Hanzo is the American open-source alternative that owns the model and the cryptography together—and the V&V framework here is how that stack is qualified to run sovereign and disconnected, where the data lives, rather than rented from a black box.

2 Observability Platform

2.1 Architecture

Hanzo O11y provides unified observability across three telemetry signals:

Signal	Capabilities	Storage
Logs	Full-text search, structured filters, correlation	ClickHouse
Metrics	Time-series, histograms, percentiles, Apdex	ClickHouse
Traces	Flamegraphs, Gantt charts, span detail	ClickHouse

Table 1: O11y three-signal observability architecture.

2.2 LLM-Specific Observability

AI workloads require specialized telemetry beyond traditional APM:

Listing 1: LLM span attributes in O11y trace.

```
{
  "name": "llm.completion",
  "duration_nano": 1500000000,
  "attrs": {
    "llm.model": "zen4-pro-27b",
    "llm.token_input": "2048",
    "llm.token_output": "512",
    "llm.temperature": "0.7",
    "llm.cost": "0.0035",
    "llm.provider": "hanzo-engine"
  }
}
```

- **Token tracking**: Input/output token counts per request, aggregated by model, provider, and team.
- **Cost analysis**: Real-time cost tracking with budget alerts per organization.
- **Quality metrics**: BLEU, ROUGE, exact match scores via evaluation workflows.
- **Latency breakdown**: Time-to-first-token, tokens-per-second, queue wait time.
- **Error analysis**: Provider failure rates, timeout patterns, content filter triggers.

2.3 Alert and Anomaly Detection

- **Threshold-based**: Static rules on any metric (latency $>$ 2s, error rate $>$ 5%).
- **Anomaly detection**: Statistical deviation from baseline patterns.
- **Composite rules**: AND/OR combinations with flapper suppression.
- **Multi-channel**: Email, Slack, PagerDuty, MS Teams, webhooks.
- **Escalation**: Time-based severity escalation with on-call integration.

3 Behavioral Analytics

3.1 Hanzo Insights Platform

Insights provides behavioral analytics with IAM-integrated multi-tenancy:

- **Event ingestion**: Rust capture service for high-throughput event collection, Kafka-compatible streaming via Hanzo Stream.
- **OLAP engine**: ClickHouse columnar storage for sub-second queries across billions of events.
- **Cohort segmentation**: Group users by behavior patterns, demographics, or temporal characteristics.

- **Funnel analysis**: Multi-step conversion tracking with statistical significance testing.
- **Retention modeling**: Cohort-based retention curves with churn prediction.
- **Feature flags**: A/B testing with gradual rollout, kill switches, and variant-level metrics.
- **Dashboards**: Time-series, heatmaps, pie charts, bar charts with custom SQL queries.

3.2 LLM Analytics Module

- **Provider adapters**: Unified client for 100+ LLM providers (OpenAI, Anthropic, local Hanzo Engine).
- **Evaluation framework**: LLM-as-judge with configurable metrics and datasets.
- **Trace clustering**: Semantic grouping of conversations using embeddings.
- **Sentiment classification**: Automated sentiment analysis on model outputs.
- **Model registry**: Version tracking, provider key management, deployment tagging.

4 ML Pipeline Orchestration

4.1 Temporal Workflows

Temporal provides durable execution for long-running ML operations:

Workflow	Purpose
RunEvaluationWorkflow	LLM model evaluation with judge activities
BatchTraceSummarizationWorkflow	Summarize conversation traces in batches
TraceClusteringWorkflow	Cluster traces by semantic similarity
DailyTraceClusteringWorkflow	Scheduled daily clustering
ClassifySentimentWorkflow	Sentiment analysis on LLM outputs

Table 2: Temporal ML workflow catalog.

Algorithm 1 LLM Evaluation Workflow

```
Fetch evaluation configuration from database
for each trial i = 1, ..., N do
    Execute LLM judge activity (retry: 3 attempts, exponential backoff)
    Emit evaluation result event to analytics stream
    Update evaluation state in key-value store
end for
Compute aggregate metrics (mean, std, percentiles)
Emit completion telemetry (success/error, duration)
return evaluation summary with per-trial results
```

4.2 Dagster DAG Pipelines

Dagster manages ETL and data preparation workflows:

- **PostgreSQL → ClickHouse ETL**: Real-time event ingestion with sensor triggers.
- **Historical backfill**: Time-partitioned asset materialization for training data.
- **Materialized columns**: Pre-compute expensive aggregations for query performance.
- **Person resolution**: User identity unification across events.
- **Operational alerts**: Slack notifications on DAG failures.

4.3 Experiment Tracking

- Model comparison with A/B testing framework.
- Statistical significance (t-test, chi-square) for metric differences.
- Prompt version tracking with performance diffs.
- Feature flags for gradual model rollout.

5 Digital Twin Environments

5.1 Netrunner: Network Simulation

Netrunner provides configurable network simulation for system qualification:

- Configurable validator count, network topology, and fault injection.
- Consensus protocol selection (Quasar, Wave, Photon, Nova, Nebula).
- Byzantine validator simulation for adversarial testing.
- Performance benchmarking under controlled conditions.

5.2 Kubernetes-Based Environment Cloning

- **Namespace isolation**: Each test environment in a dedicated K8s namespace.
- **Parallel execution**: Multiple test environments run simultaneously.
- **Docker Compose stacks**: Dev stack (single node) and Universe (5-validator production replica).
- **Multi-cluster**: hanzo-k8s (DigitalOcean) and apps-gke (GCP ARM64).

5.3 Virtual Machine Registry

38 registered VMs with pluggable lifecycle:

VM	Chain	Test Focus
PlatformVM	P-Chain	Staking, validator set management
ExchangeVM	X-Chain	UTXO asset trading
EVM	C-Chain	Smart contracts, precompiles
DexVM	D-Chain	Orderbook, perpetuals, AMM
ThresholdVM	T-Chain	FHE, threshold signing
SessionVM	S-Chain	PQ messaging
BridgeVM	B-Chain	Cross-chain verification
AIvm	A-Chain	AI agent consensus
QuantumVM	Q-Chain	PQ key operations

Table 3: Selected VMs from 38 registered (security-relevant subset).

6 Automated Testing Frameworks

6.1 Test Coverage

Component	Tests	Status
Quasar consensus	28+	All passing
Corona threshold	21	All passing
SessionVM E2E	6	All passing
DEX precompiles	68	All passing
NIST PQ vectors	Per-algorithm	All passing
MPC keygen/signing	33 files	All passing

Table 4: Automated test suite coverage.

6.2 CI/CD Pipeline

- **Multi-platform builds**: `linux/amd64` + `linux/arm64` via native runners (no QEMU).
- **ARC scale sets**: Up to 30 ARM64 runners on GKE T2A Ampere nodes.
- **Tags**: `:main`, `:test`, `:dev` (branch-based), `:latest` (default branch).
- **Shared workflow**: `hanzoai/.github/.github/workflows/docker-build.yml@main`.

7 Formal Verification

7.1 Cryptographic Proofs

17 formal proofs cover cryptographic primitives, consensus protocols, and cross-chain messaging:

- **proof-crypto-bls**: BLS12-381 aggregate signature unforgeability.
- **proof-crypto-cggmp21**: CGGMP21 threshold ECDSA correctness under UC framework.
- **proof-crypto-mldsa**: ML-DSA EU-CMA reduction to Module-LWE hardness.
- **proof-crypto-tfhe**: TFHE semantic security under LWE assumption.
- **proof-crypto-verkle**: Verkle tree binding and hiding properties.
- **proof-protocol-wave**: Wave consensus liveness and safety.
- **proof-protocol-nova**: Nova linear consensus termination bound.
- **proof-bridge-teleport**: Cross-chain message integrity guarantee.
- **proof-warp-delivery**: Warp messaging delivery with finality binding.

7.2 Gas Determinism

All EVM precompiles provide O(1) gas cost computation from input header bytes:

Theorem 7.1 (Gas Determinism)**.** *For any precompile P with input x, the gas cost $G(P, x)$ is computable in constant time from the first k bytes of x (where $k \leq 8$), independent of the remaining input. No unbounded computation occurs during gas estimation.*

8 Performance Measurement

8.1 Consensus Performance

Network	Validators	Finality	TPS	Config
Mainnet	21	9.63 s	1,091	K=21, α=15
Testnet	11	6.3 s	1,094	K=11, α=8
Local	5	3.69 s	438–1,094	K=5, α=3

Table 5: Consensus performance across deployment sizes.

8.2 Wire Protocol

Operation	ZAP	MCP JSON-RPC	Speedup	Allocs
Single tool call	199 ns	3,939 ns	20×	2 vs. 58
20-tool orchestration	5,104 ns	73,685 ns	14×	60 vs. 1,060
Parse (buffer read)	2.2 ns	—	—	0
Build (message construct)	38.9 ns	—	—	0
Consensus round	4.6 ns	—	—	0

Table 6: ZAP benchmarks (Apple M1 Max, Go 1.26.1). MCP comparison from measured test suite.

9 Data Flow Architecture

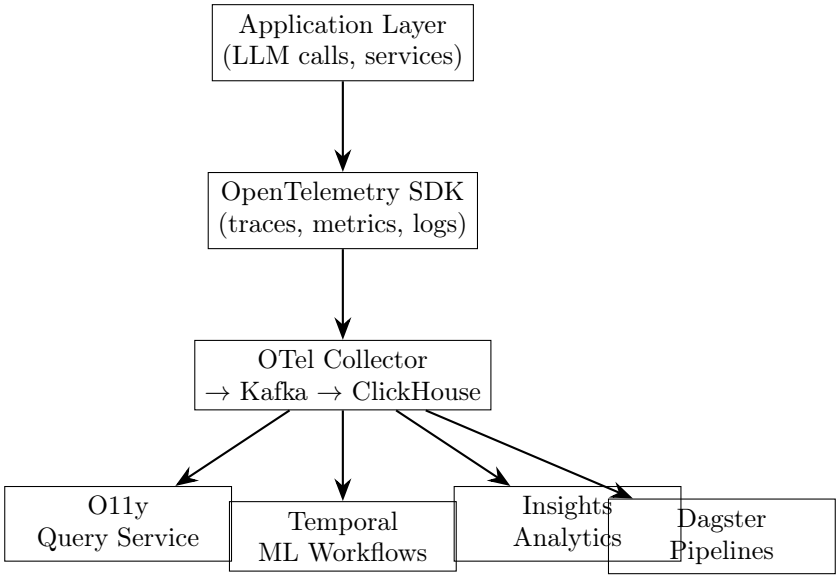

Figure 1: V&V data flow from application to analytics.

10 Naval Application Scenarios

10.1 System Qualification

Digital twin environments replicate production configurations for pre-deployment testing:

- Deploy identical K8s stack in isolated namespace.
- Run full consensus with simulated network conditions (latency, jitter, packet loss).
- Execute automated test suites against all cryptographic primitives.
- Verify performance meets operational requirements before fleet deployment.

10.2 Mission Rehearsal

- Configure Netrunner with theater-specific network topology.
- Simulate DDIL conditions (satellite jamming, link degradation).
- Validate AI model performance under constrained bandwidth.
- Test graceful degradation and recovery procedures.

10.3 Continuous Monitoring at Sea

- O11y agents deployed on each node collect telemetry locally.
- Store-and-forward synchronization when connectivity resumes.
- Anomaly detection runs locally with pre-trained models.
- Alert escalation via available communication channels.

11 Conclusion

We have presented an automated V&V framework that provides continuous verification of AI/ML systems, cryptographic implementations, and consensus protocols through integrated observability, behavioral analytics, ML pipeline orchestration, and digital twin environments. The framework's 17 formal proofs, comprehensive test suites, and production observability stack enable data-driven decision-making for naval system qualification and mission validation.

References

[1] OpenTelemetry, "Observability Framework for Cloud-Native Software," CNCF, 2024.

[2] ClickHouse Inc., "ClickHouse: Fast Open-Source OLAP Database Management System," 2024.

[3] Temporal Technologies, "Temporal: Durable Execution Platform," 2024.

[4] Elementl, "Dagster: Cloud-Native Data Orchestration," 2024.

[5] Hanzo Team, "Hanzo O11y: Unified Observability Platform," Hanzo Industries, 2024.

[6] Hanzo Team, "Hanzo Insights: Behavioral Analytics Platform," Hanzo Industries, 2024.

[7] Prometheus Authors, "Prometheus: Monitoring System and Time Series Database," CNCF, 2024.

[8] Kubernetes Authors, "Kubernetes: Production-Grade Container Orchestration," CNCF, 2024.

[9] Hanzo Team, "Quasar Consensus," Hanzo, 2025.

[10] "Practical Two-Round Threshold Signature from LWE," IACR ePrint 2024/1113.

[11] NIST, "Module-Lattice-Based Digital Signature Standard," FIPS 204, 2024.

[12] NIST, "Post-Quantum Cryptography Standardization," 2024.

[13] M. Grieves, "Digital Twin: Manufacturing Excellence through Virtual Factory Replication," 2015.

[14] G. Kim *et al.*, "The DevOps Handbook," IT Revolution Press, 2016.

[15] R. Kohavi *et al.*, "Trustworthy Online Controlled Experiments," Cambridge University Press, 2020.

Reproducibility

Source code. `github.com/hanzoai/platform, hanzoai/agents` (commit `TBD`).
Build. make test per component; CI via GitHub Actions on amd64/arm64 runners.
Hardware tested. Linux amd64/arm64; macOS for local dev.
Standards referenced. IEEE 29119, NIST SP 800-160, DevSecOps.
Contact. research@hanzo.ai.