Post-Quantum Resilient Mesh Networking for Maritime and Tactical Edge Operations

Zach Kelling — Hanzo Team

Version 1.0 — January 15, 2026 January 2026

Abstract

We present a post-quantum resilient networking architecture designed for maritime, tactical edge, and disconnected/degraded/intermittent/limited-bandwidth (DDIL) environments. The system integrates the Reticulum Network Stack (RNS) as a native consensus transport layer alongside traditional IP networking, enabling validators and edge nodes to communicate over LoRa, packet radio, serial links, and mesh topologies without reliance on centralized infrastructure or DNS. All transport sessions employ hybrid post-quantum key exchange (X25519 + ML-KEM-768, FIPS 203) and hybrid authentication (Ed25519 + ML-DSA-65, FIPS 204), providing defense-in-depth against both classical and quantum adversaries. The architecture includes a zero-trust overlay network with identity-first micro-segmentation, a zero-copy binary protocol achieving 11.5 million transactions per second, and fault-tolerant streaming over NATS JetStream. We demonstrate sub-second consensus finality over mesh networks and analyze the system's suitability for shipboard, shore-to-ship, expeditionary, and submarine communication scenarios.

Executive Summary. This is a communications layer that keeps networked systems coordinated when the usual infrastructure—satellite links, cell towers, DNS, certificate authorities—is jammed, degraded, or simply absent. It runs over whatever carries bytes (long-range radio, serial cable, existing tactical radios, satellite) and falls back automatically as links drop, so nodes stay in agreement at sea, underground, or behind a contested boundary. Every link is encrypted with two independent locks at once—today's cryptography plus the new quantum-resistant standards mandated for national-security systems—so traffic recorded now cannot be decrypted later when quantum computers arrive. It is built for operators who must assume the network is hostile and the adversary is patient: defense program offices, government integrators, and operators of critical infrastructure that has to keep working when the connection does not. The performance figures and security properties stated below are measured, not aspirational.

Contents

1 Introduction

Naval operations increasingly depend on resilient, high-performance networking across heterogeneous and contested environments. The convergence of three trends—the quantum computing threat to classical cryptography, the proliferation of edge computing devices, and the operational necessity of DDIL communications—demands a fundamentally new approach to assured networking.

Traditional military networking architectures assume reliable, high-bandwidth links with centralized key management and DNS-based addressing. These assumptions fail in contested maritime environments where satellite communications may be jammed, shore infrastructure may be degraded, and adversaries may conduct harvest-now-decrypt-later (HNDL) attacks against encrypted traffic.

1.1 Challenges in Naval Networking

(i) **Disconnected Operations**: Vessels and expeditionary forces must maintain consensus and data synchronization during extended periods without connectivity.

(ii) **Medium Heterogeneity**: Communication may traverse satellite, HF radio, LoRa mesh, fiber, and IP networks within a single operational theater.

(iii) **Quantum Threat**: Nation-state adversaries are developing cryptographically relevant quantum computers, threatening all classical public-key cryptography.

(iv) **Centralized Fragility**: Dependence on DNS, certificate authorities, and centralized key servers creates single points of failure.

(v) **Latency Tolerance**: Store-and-forward capability is essential when links have multi-second or multi-minute round-trip times.

1.2 Contributions

This paper presents the following contributions:

1. **RNS-Native Consensus**: Integration of the Reticulum Network Stack as a first-class transport layer in the Hanzo consensus protocol, enabling validator communication over any physical medium.

2. **Hybrid PQ Transport**: End-to-end post-quantum key exchange and authentication for all mesh links using NIST FIPS 203 and FIPS 204 algorithms.

3. **Zero-Trust Overlay**: Identity-first networking with micro-segmentation, eliminating implicit trust from network position.

4. **High-Performance Data Distribution**: The ZAP zero-copy binary protocol achieving 2.2 ns parse latency with zero memory allocations.

5. **Tactical Deployment Analysis**: Concrete scenarios for shipboard mesh, shore-to-ship relay, and expeditionary force networking.

1.3 Strategic Context

Assured networking is the foundation on which everything else in a sovereign, disconnected AI stack rests: if nodes cannot reach agreement, neither the models nor the cryptography above them matter. Hanzo (American-owned, Techstars '17, founded 2014) builds this transport in-house because the same organization owns both the post-quantum cryptography securing each link and the frontier AI—the *Zen* model family—that runs over it. That vertical ownership is the strategic point. The center of gravity for "open" frontier AI has shifted offshore: the United States has narrowed to a closed two-vendor model duopoly, while the broad open ecosystem—and thus most openly available frontier weights—now sits abroad. Hanzo is the American open-source alternative that controls the model and the cryptography together, so defense and regulated operators can run auditable, sovereign AI on their own networks—offline, at the tactical edge, where the data lives—without renting from the closed duopoly or depending on foreign open weights. The mesh transport described here is what lets that stack operate when the network is contested rather than only in the data center.

2 Reticulum Network Stack Integration

2.1 Architecture Overview

The Reticulum Network Stack (RNS) provides medium-agnostic, delay-tolerant, encrypted networking without dependence on IP infrastructure, DNS, or certificate authorities. We integrate RNS as a native transport layer in the Hanzo consensus node, alongside traditional TCP/IP endpoints.

The Hanzo endpoint abstraction supports three addressing modes:

Type	**Format**	**Resolution**
IP	`203.0.113.50:9631`	Direct socket connection
Hostname	`validator.example.com:9631`	DNS resolved at dial time
RNS	`rns://a5f72c3d...`	Cryptographic destination hash, no DNS

Table 1: Unified endpoint addressing in Hanzo consensus.

RNS destinations are identified by 16-byte (128-bit) truncated hashes of the destination's public keys, providing cryptographic addressing that requires no external naming infrastructure.

2.2 Supported Physical Interfaces

RNS operates over any medium capable of carrying bytes:

- **AutoInterface**: Automatic detection of available local network media.
- **TCPClientInterface**: Standard TCP/IP for LAN and WAN connectivity.
- **LoRaInterface**: Long-range radio (LoRa) for low-bandwidth mesh operation, enabling validator communication across 10–50 km ranges at 1–100 kbps.
- **SerialInterface**: Direct serial connections for point-to-point links.
- **Packet Radio**: Amateur and military packet radio systems.

- **Satellite**: Via TCP or serial gateways to SATCOM terminals.

2.3 Software-Defined Radio Integration

RNS AutoInterface and SerialInterface provide natural integration points for Software-Defined Radio (SDR) platforms. SDR waveforms that output to serial or IP interfaces are consumed directly by RNS without modification to the network stack, enabling rapid adoption of new waveform types as they become available.

The ZAP binary protocol is well-suited as an application layer over SDR waveforms. A complete ZAP-encoded consensus call fits within 256 bytes, making it viable even over narrowband SDR channels where every byte of airtime is contested. This compactness is critical for HF and UHF waveforms with limited symbol rates.

RNS enables frequency-agile operation: when an SDR platform switches between channels, frequencies, or waveform modes, RNS transparently re-establishes links via its delay-tolerant transport without requiring upper-layer reconfiguration. This is particularly valuable in electronic warfare environments where frequency hopping and adaptive waveform selection are operational necessities.

The architecture is compatible with the Software Communications Architecture (SCA) and Joint Tactical Radio System (JTRS) radio families. SCA-compliant radios that expose serial or IP gateway interfaces can serve as RNS physical transports, bridging between the mesh networking layer and legacy tactical radio infrastructure.

LoRa, already supported as a first-class RNS interface, demonstrates the SDR integration model: it is a programmable spread-spectrum radio whose modulation parameters (spreading factor, bandwidth, coding rate) are software-configured at runtime. Extending this pattern to wideband SDR platforms such as USRP, LimeSDR, or military SCA radios requires only a serial or IP bridge—no changes to RNS, ZAP, or the consensus protocol.

2.4 Configuration

Listing 1: RNS transport configuration in the Hanzo node.

```
type RNSConfig struct {
    ConfigPath       string        // ~/.reticulum/
    IdentityPath     string        // ~/.reticulum/identity
    GatewayAddr      string        // Optional gateway
    AnnounceInterval time.Duration // Default 5 minutes
    Interfaces       []string      // AutoInterface, LoRaInterface
    LinkTimeout      time.Duration // Default 30 seconds
    Enabled          bool          // Default false
    PostQuantum      bool          // Enable hybrid PQ
    RequirePostQuantum bool        // Reject classical-only
}
```

2.5 Deployment Scenarios

The Pars Improvement Proposal PIP-7009 defines five deployment scenarios for RNS-based validator operation:

1. **Sovereign Infrastructure**: Validators operating in jurisdictions with internet restrictions, using mesh networks to reach global consensus.

2. **Disaster Resilience**: Mesh networks maintaining consensus during infrastructure outages, using LoRa and packet radio.
3. **Remote Deployment**: LoRa-connected validators in areas without internet access—remote bases, offshore platforms, polar stations.
4. **Mobile Validators**: Maritime, aviation, and vehicular validator nodes maintaining consensus during movement.
5. **Air-Gapped Security**: High-security validators with controlled, non-IP connectivity for maximum isolation.

3 Post-Quantum Hybrid Transport Security

3.1 Threat Model

We assume an adversary with the following capabilities:

- Full network observation and recording of all ciphertext (HNDL capability).
- Access to a cryptographically relevant quantum computer within 10–15 years.
- Ability to perform active man-in-the-middle attacks on initial connection.
- No compromise of endpoint hardware or key material.

3.2 Hybrid Identity

Each RNS node maintains a hybrid identity combining classical and post-quantum key pairs:

Definition 3.1 (Hybrid RNS Identity)**.** A hybrid identity $\mathcal{I}$ consists of:

$$\begin{aligned}\mathcal{I} = \big(&\mathrm{sk}_{\text{Ed25519}}, \mathrm{pk}_{\text{Ed25519}},\\ &\mathrm{sk}_{\text{X25519}}, \mathrm{pk}_{\text{X25519}},\\ &\mathrm{sk}_{\text{ML-DSA-65}}, \mathrm{pk}_{\text{ML-DSA-65}},\\ &\mathrm{sk}_{\text{ML-KEM-768}}, \mathrm{pk}_{\text{ML-KEM-768}}\big)\end{aligned} \tag{1}$$

3.3 Hybrid Key Exchange Protocol

Algorithm 1 RNS Hybrid Post-Quantum Key Exchange

Initiator A generates ephemeral keys:
  $(\text{ek}_A^X, \text{epk}_A^X) \leftarrow$ X25519.Generate()
  $(\text{ek}_A^M, \text{epk}_A^M) \leftarrow$ ML-KEM-768.Generate()
$A \to B$: $\text{epk}_A^X \| \text{epk}_A^M \| \text{pk}_A^{\text{DSA}} \| \sigma_A$
  where $\sigma_A = \text{ML-DSA-65.Sign}(\text{sk}_A^{\text{DSA}}, \text{epk}_A^X \| \text{epk}_A^M)$

Responder B verifies and computes:
  Verify σ_A under pk_A^{DSA}
  $\text{ss}^X \leftarrow \text{X25519.DH}(\text{sk}_B^X, \text{epk}_A^X)$
  $(\text{ct}^M, \text{ss}^M) \leftarrow \text{ML-KEM-768.Encaps}(\text{epk}_A^M)$
  $K \leftarrow \text{HKDF-SHA256}(\text{“RNS-HybridLink-v1”}, \text{ss}^X \| \text{ss}^M)$
$B \to A$: $\text{epk}_B^X \| \text{ct}^M \| \text{pk}_B^{\text{DSA}} \| \sigma_B$

Initiator A derives session key:
  $\text{ss}^X \leftarrow \text{X25519.DH}(\text{ek}_A^X, \text{epk}_B^X)$
  $\text{ss}^M \leftarrow \text{ML-KEM-768.Decaps}(\text{ek}_A^M, \text{ct}^M)$
  $K \leftarrow \text{HKDF-SHA256}(\text{“RNS-HybridLink-v1”}, \text{ss}^X \| \text{ss}^M)$
  Derive: $K_{\text{send}}, K_{\text{recv}}, K_{\text{MAC-s}}, K_{\text{MAC-r}} \leftarrow \text{HKDF-Expand}(K)$
Destroy all ephemeral private keys (forward secrecy)
Session: AES-256-GCM with counter-mode nonces

3.4 Wire Format Analysis

Component	Classical	Hybrid PQ
X25519 ephemeral public key	32 B	32 B
ML-KEM-768 ephemeral public key	—	1,184 B
ML-KEM-768 ciphertext	—	1,088 B
ML-DSA-65 public key	—	1,952 B
ML-DSA-65 signature	—	3,309 B
Ed25519 public key	32 B	32 B
Ed25519 signature	64 B	64 B
Total handshake	**∼256 B**	**∼7,500 B**

Table 2: Wire format comparison: classical vs. hybrid PQ handshake.

The 29× overhead is a one-time cost per link establishment. Session data uses AES-256-GCM with no additional overhead, making the amortized cost negligible for long-lived connections.

3.5 Security Properties

Theorem 3.2 (Hybrid Security)**.** *The hybrid key exchange provides IND-CCA2 security if either X25519 ECDH or ML-KEM-768 is secure. Formally, an adversary must break both the Computational*

Diffie-Hellman problem on Curve25519 and the Module-LWE problem with parameters $(k = 3, n = 256, q = 3329)$ *to compromise session keys.*

Theorem 3.3 (Forward Secrecy)**.** *Compromise of any long-term identity key does not compromise past session keys, as each session uses fresh ephemeral key pairs that are destroyed after derivation.*

4 Zero-Trust Overlay Network

4.1 Hanzo ZT Architecture

Hanzo Zero Trust (ZT) is a comprehensive zero-trust networking framework comprising 78 repositories, providing identity-first overlay networking with no implicit trust from network position.

4.1.1 Core Components

- **Identity**: Certificate-based cryptographic identity with HSM support (PKCS#11, YubiHSM, CloudHSM). Each endpoint possesses a unique cryptographic identity that must be verified before any communication.
- **Transport**: Pluggable transport layer supporting TCP, TLS, DTLS, WebSocket, and Transwarp (UDP-based with Dilithium flow control).
- **Channel**: Binary protocol framework for type-safe, encrypted message passing.
- **Fabric**: Overlay routing mesh that forwards traffic based on identity policies, not network addresses.

4.1.2 Transport Plugin Architecture

New transports are registered via a global parser registry:

Listing 2: Transport plugin registration pattern.

```
type Address interface {
    Dial(name string, id *identity.TokenId,
         timeout time.Duration,
         tcfg Configuration) (Conn, error)
    Listen(name string, id *identity.TokenId,
           acceptF func(Conn),
           tcfg Configuration) (io.Closer, error)
    Type() string
}

// Register at init time
transport.AddAddressParser(RNSAddressParser{})
```

This plugin architecture enables RNS to serve as a ZT underlay transport, combining zero-trust policy enforcement with mesh networking resilience.

4.2 Dilithium Fault-Tolerant Transport

The Dilithium protocol (distinct from the PQ signature scheme of the same name) implements Westworld3, an adaptive UDP-based flow control framework optimized for long-haul, dedicated links.

Parameter	Value
TxPortal start	96 KB
TxPortal max	4 MB
Retransmission start	200 ms
RTT probe interval	50 ms
Max segment size	1,450 B (UDP MSS)

Table 3: Dilithium/Westworld3 default baseline profile.

Key resilience mechanisms include adaptive congestion control (RTT-based), duplicate ACK detection with capacity reduction (0.9×), retransmission with aggressive capacity cut (0.75×), and pool-based buffer management.

5 High-Performance Data Distribution

5.1 ZAP Zero-Copy Binary Protocol

The ZAP (Zero-copy Application Protocol) provides extreme-performance serialization for consensus messaging, AI agent communication, and GPU cluster coordination. Two parallel implementations serve different ecosystems:

- **Rust implementation**: Multi-language SDK generation, MCP gateway bridging, PQ handshake.
- **Go implementation**: Consensus wire protocol, VM-to-VM communication, mDNS discovery.

5.1.1 Performance

Operation	ZAP	Protobuf	Speedup
Parse	2.2 ns	500 ns	172×
Serialize	44 ns	4,221 ns	96×
Consensus round	5.5 ns	—	—
Allocations/op	0	11–121	∞
End-to-end TPS	11.5M	246K	47×

Table 4: ZAP performance vs. Protocol Buffers.

5.1.2 Wire Format

ZAP uses a 16-byte little-endian header:

Listing 3: ZAP wire header format.

```
Magic:       "ZAP\x00"     (4 bytes)
```

```
Version:     1              (2 bytes)
Flags:       compressed|encrypted|signed (2 bytes)
Root Offset: offset to root struct (4 bytes)
Size:        total message size     (4 bytes)
```

All struct fields are accessed by fixed offset calculation with 8-byte alignment, enabling true zero-copy reads directly from the network buffer.

5.1.3 Post-Quantum Handshake

ZAP includes native PQ transport security using the same hybrid scheme as RNS: ML-KEM-768 + X25519 for key exchange, ML-DSA-65 for authentication.

5.2 Hanzo Stream: Resilient Event Distribution

Hanzo Stream provides Kafka wire protocol compatibility over NATS JetStream, enabling standard Kafka clients to connect without modification while gaining the resilience benefits of JetStream's distributed storage.

- **Stateless**: All storage and replication delegated to NATS JetStream.
- **Compression**: GZIP, Snappy, LZ4, ZSTD.
- **Topic mapping**: Kafka topic T, partition $P \rightarrow$ JetStream stream `kafka-`T`-`P.
- **Consumer offsets**: KV bucket with key format `{group}.{topic}.{partition}`.

6 Software-Defined Networking

6.1 Hanzo Ingress: L7 Reverse Proxy

Hanzo Ingress serves as the cloud-native front door for all production traffic:

- **Protocol support**: HTTP/1.1, HTTP/2, gRPC, WebSocket, QUIC-ready.
- **TLS management**: Automatic Let's Encrypt with DNS challenge, wildcard certificates, zero-downtime renewal.
- **Circuit breakers**: Configurable failure expression thresholds with fallback responses, preventing cascading failures.
- **Rate limiting**: Per-endpoint and global token bucket algorithms.
- **Observability**: Prometheus metrics, OpenTelemetry tracing, structured access logging.

6.2 Hanzo Gateway: Policy-Based Routing

The API gateway routes 147+ endpoints across production services with hot-reloadable YAML configuration:

- **JWKS authentication**: JWT validation against `hanzo.id/.well-known/jwks` with 10-second TTL caching.

- **Org-scoped identity**: `X-Org-Id`, `X-User-Id`, `X-User-Email` headers derived from JWT claims.
- **Per-endpoint rate limiting**: Global 5,000 req/s, per-IP 100 req/s.
- **Multi-level security**: Organization-scoped access via IAM + KMS integration, with 46+ services using KMSSecret CRDs.

7 Edge Tunnel System

The Hanzo Tunnel provides a Rust-based cloud relay for connecting edge devices to the platform:

- **Transport**: JSON frames over WebSocket Secure (WSS).
- **Resilience**: Automatic reconnection with exponential backoff (1s initial, 60s max), 30-second heartbeat, 10-second connect timeout.
- **Authentication**: Bearer token (JWT from hanzo.id OAuth or API key).
- **Frame types**: Register, Registered, Event, Command, Response, Ping, Pong.
- **App kinds**: Dev, Node, Desktop, Bot, Extension.

8 Kubernetes-Native Infrastructure

The production deployment runs on Kubernetes with 65+ deployments across multiple clusters:

- **High availability**: Multi-replica deployments (2+ for stateless services) with zero-downtime rolling updates (`maxSurge: 1, maxUnavailable: 0`).
- **Secrets management**: KMSSecret CRDs for declarative secret synchronization from Hanzo KMS to Kubernetes secrets. Zero plaintext in YAML manifests.
- **Storage**: Persistent volumes on block storage for stateful services (PostgreSQL, NATS JetStream, ClickHouse).
- **Multi-cluster**: `hanzo-k8s` (DigitalOcean SFO3) for services, `apps-gke` (GCP) for ARM64 CI runners.

9 Performance Analysis

9.1 Consensus over RNS

Transport	Finality	Bandwidth	Latency
TCP/IP (LAN)	500 ms	1–10 Gbps	<1 ms
TCP/IP (WAN)	700 ms	100 Mbps	50–200 ms
RNS (TCP)	900 ms	100 Mbps	50–200 ms
RNS (LoRa)	3–10 s	1–100 kbps	200–2000 ms
RNS (Satellite)	2–5 s	1–50 Mbps	500–1500 ms

Table 5: Consensus finality across transport media.

9.2 Protocol Overhead

Protocol	Parse	Allocs	Throughput	PQ Support
ZAP	2.2 ns	0	11.5M TPS	Yes
Protobuf	500 ns	11–121	246K TPS	No
JSON-RPC	5,579 ns	58	140K calls/s	No
Cap'n Proto	∼100 ns	0	∼2M TPS	No

Table 6: Protocol performance comparison.

10 Naval Application Scenarios

10.1 Shipboard Mesh Network

A carrier strike group deploys RNS mesh networking across vessels:

- **Inter-ship**: LoRa mesh (10–50 km range) for consensus messages, falling back to SATCOM for bulk data.
- **Intra-ship**: Ethernet/WiFi via AutoInterface for high-bandwidth local communication.
- **Security**: Hybrid PQ handshakes protect against adversary quantum capabilities; zero-trust overlay prevents lateral movement from compromised nodes.
- **Resilience**: If SATCOM is jammed, LoRa mesh maintains consensus among vessels within radio range.

10.2 Shore-to-Ship Relay

- Shore gateway nodes bridge between IP infrastructure and RNS mesh.
- ZAP protocol minimizes bandwidth usage (91% memory reduction vs. JSON-RPC).
- Hanzo Stream provides store-and-forward for delay-tolerant data synchronization.

10.3 Expeditionary Force Networking

- Man-portable LoRa nodes establish mesh network in austere environments.
- Edge AI models (Zen nano, 0.8B parameters) run on tactical hardware for local decision support.
- Consensus maintains data integrity even with intermittent connectivity.

10.4 Submarine Communications

- RNS serial interface connects to existing submarine communication systems.
- Store-and-forward ensures eventual consistency during extended silent running.
- Post-quantum encryption protects against long-term decryption of recorded traffic.

11 Transport Capability Matrix

Capability	TCP	DTLS	TLS	WS	Transwarp	RNS
Encryption	–	Yes	Yes	TLS	TLS	PQ
NAT Traversal	–	–	–	–	–	Yes
Mesh Routing	–	–	–	–	–	Yes
Store-Forward	–	–	–	–	–	Yes
Delay-Tolerant	–	–	–	–	–	Yes
LoRa Support	–	–	–	–	–	Yes
Packet Radio	–	–	–	–	–	Yes
Adaptive Flow	TCP	OS	TCP	TCP	W3	RNS

Table 7: Transport capability matrix across all supported protocols.

12 Conclusion and Roadmap

We have presented a comprehensive assured networking architecture that addresses the unique challenges of naval and tactical edge operations. The integration of RNS as a native consensus transport, combined with hybrid post-quantum cryptography, zero-trust overlay networking, and high-performance data distribution, provides a resilient foundation for DDIL environments.

12.1 Current Capabilities

- RNS transport fully integrated into Hanzo consensus with hybrid PQ handshake.
- ZAP protocol achieving 11.5M TPS with zero-copy design.
- Zero-trust overlay with 7 transport plugins and HSM support.
- Production Kubernetes deployment with 65+ services.

12.2 Roadmap

- **Q2 2026 (planned)**: SATCOM-optimized RNS interface with forward error correction.
- **Q3 2026 (planned)**: Multi-path simultaneous transport (IP + LoRa + satellite).
- **Q4 2026 (planned)**: COMSEC certification preparation for naval deployment.
- **2027 (planned)**: Integration with Navy tactical data links (Link 16/22).

References

[1] National Institute of Standards and Technology, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS 203, August 2024.

[2] National Institute of Standards and Technology, "Module-Lattice-Based Digital Signature Standard," FIPS 204, August 2024.

[3] National Institute of Standards and Technology, "Stateless Hash-Based Digital Signature Standard," FIPS 205, August 2024.

[4] National Security Agency, "Commercial National Security Algorithm Suite 2.0," CNSA 2.0, September 2022.

[5] R. Barnes, K. Bhargavan, B. Lipp, C. Wood, "Hybrid Public Key Encryption," RFC 9180, February 2022.

[6] M. Qvist, "Reticulum Network Stack: Cryptography-based networking for resilient communications," 2023.

[7] NetFoundry, "OpenZiti: Programmable Zero Trust Networking," 2023.

[8] Pars Network, "PIP-7009: Reticulum Network Stack Integration," February 2026.

[9] Hanzo Team, "Hanzo Consensus: Multi-Protocol Byzantine Agreement with Post-Quantum Finality," Hanzo, 2025.

[10] Hanzo Team, "Quasar: Hybrid BLS-Corona Consensus with Sub-Second Quantum Finality," Hanzo, 2025.

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

[12] Cloudflare, "CIRCL: Cloudflare Interoperable Reusable Cryptographic Library," v1.6, 2024.

[13] Synadia, "NATS: Cloud Native Messaging System with JetStream Persistence," 2024.

[14] K. Varda, "Cap'n Proto: Insanely Fast Data Interchange Format," 2013.

[15] Department of Defense, "DOD Strategy for Operations in Disconnected, Degraded, Intermittent, and Limited Bandwidth Environments," 2023.

Reproducibility

Source code. `github.com/hanzoai/zap` (commit `TBD`).
Build. cargo build –release (Rust 1.78+); Go binaries via make build.
Hardware tested. Apple M-series, Linux amd64/arm64 (edge: ARM Cortex-A), tested with mDNS multicast.
Standards referenced. ML-KEM-768, ML-DSA-65 (FIPS 203/204), W3C DID, DOD DDIL.
Contact. research@hanzo.ai.