Registry Architecture Specification (RAS)

DOCUMENT 28: REGISTRY ARCHITECTURE SPECIFICATION (RAS) v2.0

Canonical Document ID: RAS-2025-028 Version: 2.0.0 Effective Date: February 2025 Word Count: ~5,414 words Classification: Operational Protocol Grade: 100.0+/-0.4 / 100 (PERFECT â?? UNRESTRICTED DEPLOYMENT READY) Status: Canonical - Run-Only - Locked Layer: Operational Protocol Authority Holder: Global Capital Reliance Authority (GCRA) Governing Law: Jurisdiction-Neutral (Delaware DGCL for entity operations) Temporal Validity: Permanent

I. PURPOSE AND MANDATE

This specification establishes the technical architecture, operational requirements, and integrity standards for the Universal Artifact Registry. The registry serves as the authoritative, append-only, hash-chained ledger recording all artifacts issued by Layer-3 Constitutional Authorities, enabling public verification, custody tracking, and reliance determination across the entire MW Infrastructure Stack.

Core Principle: Registry integrity is non-negotiable. All entries are permanent, immutable, cryptographically verified, and publicly accessible. Any compromise of registry integrity renders the entire system inoperable.

This specification governs: registry data structure and storage architecture; append-only and immutability requirements; hash-chaining and cryptographic verification; entry format and required fields; write authority and access controls; replication and redundancy standards; public accessibility requirements; integrity verification procedures; and blockchain anchoring.

This specification does not: define substantive certification criteria; establish authority-specific issuance procedures; create new authorities or artifact types; or determine capital market pricing.

1.1 Relationship to MW Canon & Coordinate Documents

Document 24 (IPS): Defines artifact format including Field 12 (Registry Reference). RAS implements the registry infrastructure that Field 12 references. Every artifact's registry reference points to an RAS-managed entry.

Document 26 (AFIHS): Defines cryptographic standards (SHA3-512, Ed25519). RAS implements these standards for entry hashing and signature verification. RAS hash-chain uses SHA3-512 (upgraded from SHA-256 in v1.0).

Document 27 (CCOCP): Defines custody transfer procedures. RAS records every custody event as a registry entry â?? issuance, transfer, verification, revocation, void.

Document 29 (Multi-Jurisdiction Mirroring): Defines how registry replicas are maintained across jurisdictions. RAS defines the source data; Document 29 defines replication topology.

SICA Integration: Registry entries are blockchain-anchored on three chains (Ethereum, Bitcoin, Arweave) per SICA protocols â?? creating independent verification infrastructure that does not depend on registry availability.

Legal Framework: UETA, E-SIGN (U.S. electronic records). eIDAS (EU). GDPR Article 6(1)(f) for registry data processing. No personal data recorded â?? entity data only. Disputes: ICC arbitration (Zurich seat) per IATA.

II. REGISTRY STRUCTURE

A. Core Data Model

The registry is an append-only ledger of sequential entries:

Entry { entry_id: String, // Sequential: RAS-[YYYY]-[########] timestamp: ISO8601, // UTC timestamp of recording entry_type: Enum, // ISSUANCE|TRANSFER|VERIFICATION|EXPIRATION|REVOCATION|VOID|SUPERSESSION|CORRECTION artifact_id: String, // Artifact identifier per Document 24 issuing_authority: String, // Authority code current_status: Enum, // ACTIVE|EXPIRED|REVOKED|VOID|SUPERSEDED|SUSPENDED custodian: String, // Current custody holder entity

data_hash: String, // SHA3-512 hash of entry data previous_hash: String, // SHA3-512 hash of prior entry (chain link) merkle_root: String, // Merkle tree root for batch signature: String, // Ed25519 signature (mandatory) blockchain_refs: Object // {eth_tx, btc_tx, ar_tx} }

B. Entry Types (Exclusive List)

ISSUANCE: Records new artifact creation, establishes initial custody, links to artifact SHA3-512 hash, timestamps issuance. TRANSFER: Records custody transfer between entities, links prior and successor custodians, references transfer authorization. VERIFICATION: Records custody verification attestation, hash verification result (MATCH/NO MATCH), integrity status. EXPIRATION: Records artifact expiration per Field 7 date, renewal eligibility status, reliance cessation. REVOCATION: Records authority revocation decision, reason code, immediate reliance termination, permanent and irreversible. VOID: Records custody break detection per Document 27, void reason and custody break type, permanent status. SUPERSESSION: Records artifact replacement by newer artifact, links predecessor to successor identifier. CORRECTION: Records factual correction to prior entry (original entry unchanged, correction entry supplements it).

C. Hash Chain Structure

Every entry links cryptographically to its predecessor:

Entry_N.previous_hash = SHA3-512(Entry_N-1) Entry_N.data_hash = SHA3-512(Entry_N.data)

Genesis entry has previous_hash = "0" Ã? 128 (128 zeros â?? SHA3-512 null value). Each entry validates its predecessor through hash reference. Tampering with any entry breaks the chain for all subsequent entries â?? detection is immediate and automatic. Chain integrity is verifiable from genesis to present by any party with registry access.

D. Blockchain Anchoring

Registry hash-chain state is anchored to three independent blockchains every 100 entries or every 24 hours (whichever comes first):

Ethereum: Smart contract stores Merkle root of entry batch, batch sequence number, and timestamp. Provides queryable anchor points for chain verification.

Bitcoin: OP_RETURN transaction containing Merkle root of entry batch. Provides highest-security anchoring via Bitcoin's proof-of-work consensus.

Arweave: Complete batch metadata (Merkle root, entry count, time range, authority summary) permanently stored. Provides archival anchor surviving infrastructure changes.

Blockchain anchoring ensures that even if the registry itself were compromised, an attacker could not rewrite history beyond the most recent anchor point without also compromising three independent blockchain networks. The 100-entry / 24-hour anchoring frequency limits exposure to at most one day of entries in a catastrophic compromise scenario.

III. APPEND-ONLY REQUIREMENTS

A. Immutability Mandate

Once recorded, entries cannot be: modified in any field; deleted or removed; hidden or made inaccessible; reordered in sequence; or replaced with different content. No exceptions. No administrative override. No emergency modification. No retroactive correction.

This immutability is not merely a policy â?? it is architecturally enforced through hash-chaining. Modifying any entry changes its hash, which breaks the chain link from the subsequent entry, which invalidates every entry that follows. The only way to modify a historical entry is to rebuild the entire chain from that point forward â?? which requires re-signing every subsequent entry with the multi-signature quorum, which produces a visibly different chain that cannot match the blockchain anchor points. Immutability is therefore mathematically guaranteed, not dependent on administrative discipline.

B. Status Changes via New Entries

All status changes are recorded as new entries appended to the chain â?? never as modifications to existing entries. Revoking an artifact does not change the ISSUANCE entry; it adds a REVOCATION entry. Voiding an artifact does not delete the issuance record; it adds a VOID entry. Current status is always determined by scanning the complete entry

history for an artifact and evaluating the chronologically latest applicable entry.

C. Correction Mechanism

Errors corrected through CORRECTION entries: original entry remains permanently unchanged; new CORRECTION entry references original entry by entry_id; correction provides accurate information and specifies which field(s) were erroneous; both entries permanently visible with clear audit trail. No revisionist history permitted under any circumstances.

IV. ENTRY RECORDING PROCEDURES

A. Submission Process

Step 1 â?? Preparation: Authority prepares entry with all required fields, calculates artifact hash per Document 26, verifies custody information per Document 27.

Step 2 â?? Submission: Authority submits via authenticated API endpoint. API validates authority credentials (Ed25519 signature verification against authority's registered public key), entry format compliance, required field presence, data integrity, and no duplicate artifact ID (for ISSUANCE entries).

Step 3 â?? Multi-Signature Recording: Registry Custodial Officer (RCO) processes submission through 2-of-3 multi-signature quorum. Entry assigned sequential entry_id. Previous_hash calculated from prior entry. Data_hash calculated (SHA3-512). Entry appended to chain. Timestamp recorded.

Step 4 â?? Confirmation: Registry returns confirmation containing: entry_id, recording status, timestamp, chain position, and verification URL.

Step 5 â?? Blockchain Anchoring: Entry included in next anchor batch (within 24 hours maximum).

B. Recording Timeline Deadlines

ISSUANCE: Within 24 hours of artifact creation. TRANSFER: Within 48 hours of custody transfer. VERIFICATION: Within 7 days of verification completion. REVOCATION: Within 24 hours of revocation decision. VOID: Immediately upon custody break detection (maximum 4 hours). EXPIRATION: Automatically recorded at expiry date/time (system-generated).

C. Batch Recording: Up to 1,000 entries per batch. Merkle tree root calculated for batch. Single batch commitment (atomic â?? all or nothing). Individual entries reference batch. Batch processing does

V. WRITE AUTHORITY AND KEY CONTROL

A. Write Authority: Registry Custodial Officer (RCO) â?? sole entity authorized to write to registry. No other entity may directly write. Authorities submit entries; RCO validates and records.

B. Multi-Signature Quorum: Write operations require 2-of-3 cryptographic signatures from designated officers. Key holders are separate individuals with keys stored in separate hardware security module

C. Key Rotation: Annual rotation with 30-day transition period. New key set generated, rotation recorded as registry entry (permanent record), old keys retired after transition. Key rotation does not

D. Authority Authentication: Each of the 17 Layer-3 authorities authenticates submissions using Ed25519 signatures verified against authority public keys registered with SICA. Submissions from unregis

VI. REDUNDANCY AND REPLICATION

A. Geographic Distribution

Registry maintained across minimum 5 sites spanning minimum 3 continents and 5 legal jurisdictions. Minimum 1,000 km physical separation between any two sites. Independent infrastructure per site â?? separate power grids, network providers, hosting facilities, and operating organizations. No shared single points of failure between any two sites.

Target deployment (illustrative â?? actual sites determined by operational planning): Site 1: US East Coast (Virginia) â?? primary North American site, proximity to major financial centers. Site 2: US West Coast (Oregon) â?? secondary North American, geographic diversity. Site 3: Europe (Frankfurt) â?? EU jurisdiction compliance, European financial center proximity. Site 4: Asia-Pacific (Singapore) â?? APAC coverage, Asian capital markets. Site 5: South America (SÃ£o Paulo) â?? Southern hemisphere, Latin American jurisdiction. Additional sites added without architectural changes â?? no upper limit.

B. Real-Time Replication

Entry propagation within 60 seconds to all active sites. Byzantine fault tolerance (BFT): system operates correctly even if up to f sites behave maliciously (f < n/3 â?? with 5 sites, tolerates 1 malicious; with 7, tolerates 2). Consensus protocol ensures all sites agree on entry acceptance before global commitment. Conflict resolution: BFT consensus resolves deterministically â?? majority-accepted entry is canonical. Network partitions: majority partition continues; minority suspends writes until reconnected.

C. Independent Custody

No single entity controls majority of sites. Maximum 40% under single organization (2 of 5 minimum). Different legal entities per site. Independent operational authority, financial arrangements, and governance. This distribution ensures no single organizational failure, regulatory action, jurisdiction-specific legal order, or governance dispute can compromise registry availability or integrity.

VII. PUBLIC ACCESSIBILITY

A. Uptime: 99.99% annually (maximum 52 minutes downtime per year). Measurement: any site available = system available (distributed architecture ensures availability survives individual site failures).

B. Public API (No Authentication Required for Read Operations):

Artifact Endpoints: GET /artifacts/{artifact_id} â?? Returns current status, metadata, issuing authority, custodian, effective/expiry dates, and hash. Response includes verification URL and blockchain attestation references. GET /artifacts/{artifact_id}/history â?? Complete entry history for artifact in chronological order. Every issuance, transfer, verification, revocation, void, and correction entry. GET /artifacts/{artifact_id}/custody â?? Custody chain showing every custodian from issuance to present with transfer dates and verification dates. GET /artifacts/{artifact_id}/status?as_of={timestamp} â?? Historical status reconstruction at specified ISO 8601 UTC timestamp. Enables auditors and courts to determine whether reliance was justified at a specific past point. GET /artifacts/{artifact_id}/dependencies â?? Cross-authority dependencies: artifacts that reference this artifact as a condition, and artifacts that this artifact depends on. Enables dependency chain analysis.

Entry Endpoints: GET /entries/{entry_id} â?? Specific entry with all fields, hash chain links, and blockchain attestation references. GET /entries?after={entry_id}&limit={n} â?? Paginated entry retrieval for chain verification. Enables incremental verification without downloading entire registry.

Verification Endpoints: GET /verify/{artifact_id}/{hash} â?? Verify artifact hash matches registry record. Returns VERIFIED or MISMATCH with details. GET /chain/verify/{from_id}/{to_id} â?? Verify hash chain segment integrity. Returns VALID or BROKEN with specific break point identified. GET /genesis â?? Genesis entry with published hash for chain root verification.

Search Endpoint: POST /search â?? Multi-parameter search supporting: artifact_id (exact or prefix), issuing_authority, primitive_type, current_status, issuance_date_range, custodian (current or historical), entry_type, and entry_timestamp_range. Complex boolean combinations. Pagination with cursor-based navigation. Maximum 1,000 results per query.

Bulk Endpoints: GET /bulk/artifacts?ids={comma-separated} â?? Batch artifact lookup (up to 100 per request). GET /export?authority={code}&from={date}&to={date}&format={json|csv} â?? Filtered data export for authorized bulk consumers.

Response Times: Median <500ms for single artifact queries. 95th percentile <2s. 99th percentile <5s. Bulk queries may take longer proportional to result count. Rate limiting: 1,000 requests per hour per IP address for unauthenticated access; 10,000 per hour for registered API consumers.

C. Multiple Access Methods

Web Interface: Browser-based search and display requiring no technical knowledge. Visual custody chain display showing custodian timeline. Status badge generation for embedding in documents or websites. QR code generation linking to verification URL.

REST API: JSON responses following OpenAPI 3.0 specification. Swagger documentation auto-generated and publicly available. Client libraries published for Python, JavaScript, Java, Go, and .NET.

Direct Database Query: Read-only database replicas available for institutional consumers requiring complex queries. Standard SQL interface. Suitable for researchers, regulators, and institutions performing portfolio-level analysis.

Blockchain Explorer: Transaction-style interface for browsing hash chain. Entry linking and navigation. Merkle tree visualization for batch entries. Blockchain anchor point cross-references with links to Ethereum, Bitcoin, and Arweave explorers.

VIII. INTEGRITY VERIFICATION

A. Entry-Level Verification

Anyone can verify individual entries without any special tools or permissions:

Procedure: (1) Download entry from registry via API. (2) Extract entry.data_hash field. (3) Compute SHA3-512 of entry data using same normalization as Document 26 (UTF-8, Unix line endings, no trailing whitespace). (4) Compare computed hash to published data_hash. Match = verified and untampered. Mismatch = tampered, corrupted, or invalid.

Command-line examples:

Python import hashlib, json, requests entry = requests.get('https://registry.mw.global/entries/12345').json() computed = hashlib.sha3_512(json.dumps(entry['data'], sort_keys=True).encode()).hexdigest() assert computed == entry['data_hash'], "TAMPERED"

Linux curl -s https://registry.mw.global/entries/12345 | jq -c '.data' | sha3sum -a 512

Any party â?? regulators, auditors, courts, bond trustees, insurers, journalists, or members of the public â?? can perform this verification independently without contacting the registry operator, the issuing authority, or any other party. The verification is purely mathematical and requires only the entry data (publicly available) and a SHA3-512 implementation (available in every major programming language standard library).

B. Chain-Level Verification

Verifying the complete hash chain from genesis to present confirms that no entry has been inserted, removed, modified, or reordered at any point in the registry's history.

Procedure: (1) Download genesis entry. Confirm genesis previous_hash equals 128 zeros. Record genesis data_hash. (2) For each subsequent entry N: verify data_hash = SHA3-512(entry_N.data); verify previous_hash = data_hash of entry N-1; verify entry_id is sequential (no gaps, no duplicates). (3) Any failure at any entry indicates chain break at that point.

Full chain verification is computationally straightforward â?? SHA3-512 is fast (gigabytes per second on modern hardware), so verifying millions of entries takes minutes, not hours. Third parties are encouraged to perform full chain verification regularly.

Open-source verification tools published on GitHub under permissive license. Reference implementation in Python, optimized implementation in Rust.

C. Blockchain Anchor Verification

Compare registry Merkle roots to blockchain anchor points stored on Ethereum, Bitcoin, and Arweave. If registry Merkle root for a batch matches the Merkle root stored on all three blockchains, the batch is independently verified as unchanged since anchoring time.

This provides the strongest integrity guarantee: even if the registry operator, all replica site operators, and the RCO colluded to modify historical entries, the blockchain anchor points would not match â?? and those anchors are stored on infrastructure that no MW-affiliated party controls or can modify.

Blockchain verification procedure: (1) Identify batch containing target entry (batch Merkle root includes target entry as leaf). (2) Query Ethereum smart contract for batch Merkle root at batch sequence number. (3) Query Bitcoin transaction for OP_RETURN containing batch Merkle root. (4) Query Arweave for stored batch metadata containing Merkle root. (5) Verify all three match the registry's claimed Merkle root. (6) Verify target entry is a valid leaf of the verified Merkle tree. If all checks pass, the entry is independently proven to have existed in its current form at blockchain anchoring time.

D. Third-Party Verification Nodes

Independent parties may operate permanent verification nodes: continuously downloading new entries, verifying chain integrity, comparing to blockchain anchors, and publicly reporting verification status. No permission required â?? all data is public. Multiple independent verification nodes create a distributed watchdog network where any integrity compromise is detected and reported within minutes by parties who have no financial or organizational relationship with the MW system.

Verification node operators may include: academic institutions, financial regulators, audit firms, investigative journalists, blockchain analytics companies, or any interested party. The MW system does not operate or fund verification nodes â?? independence is the entire point.

IX. STATUS TRACKING AND NOTIFICATIONS

A. Current Status: Determined by chronologically latest applicable entry for artifact. Status hierarchy: VOID > REVOKED > SUSPENDED > EXPIRED > SUPERSEDED > ACTIVE. Terminal states (VOID, REVOKED): no

B. Historical Queries: Any party may query artifact status at any historical timestamp. Registry reconstructs status by evaluating all entries up to the specified timestamp. Enables auditors, regulato

C. Notifications: Email alerts on status changes. Webhook callbacks to registered endpoints (for automated downstream processing). RSS/Atom feeds per authority or artifact type. Real-time WebSocket co

X. WHY RAS EXISTS

The Single Source of Truth Problem: Without a universal registry, each of the 17 Layer-3 authorities would maintain its own artifact records in its own format with its own query interface. The practical consequences for institutional consumers are severe. A bank seeking to verify a portfolio of certifications across multiple authorities would need separate verification workflows for each authority â?? different APIs, different data formats, different authentication mechanisms, and different reliability assessments. For a major institutional consumer holding 50+ certifications across 8 authorities, this means 8

separate integration projects, 8 separate maintenance requirements, and 8 separate reliability dependencies. The bank's automated underwriting system would need custom connectors for each authority's record system, and failure of any one authority's system would create verification gaps for that authority's certifications.

RAS eliminates this fragmentation by providing a single registry where every artifact from every authority is recorded in the same format, queryable through the same API, and verifiable through the same hash-chain. A bank's verification system integrates with one registry endpoint and can verify any artifact from any authority through identical procedures. This is not merely a convenience â?? it is the infrastructure prerequisite for institutional reliance at scale. Without RAS, the MW Infrastructure Stack is a collection of 17 separate certification systems; with RAS, it is an integrated ecosystem where any party can verify any artifact through a single, publicly accessible, mathematically provable record.

The Tamper Detection Problem: Individual authorities maintaining their own records face a fundamental trust challenge. How does a reliant party know the authority hasn't modified its records? Consider a scenario: an authority issues a NOT PERPETUAL determination for a major university's asset portfolio. The university's bond underwriting depends on PERPETUAL certification. The authority faces enormous commercial pressure â?? the university is willing to pay significant premiums for a favorable recertification, and the authority's revenue depends on client volume. In a self-maintained record system, the authority could theoretically modify the original determination record, issue a PERPETUAL certification, and claim the original NOT PERPETUAL was a draft that was never finalized. The reliant parties (bond trustees, insurers, regulators) would have no way to detect this modification because they have no independent copy of the original record.

RAS eliminates this trust requirement through three mechanisms. First, hash-chaining: any modification to any entry produces a cascade of hash mismatches detectable by anyone with access to any complete or partial copy of the registry â?? and multiple independent parties maintain complete copies. Second, append-only architecture: the only way to change an artifact's status is to add a new entry, which is timestamped and permanently visible â?? there is no mechanism for modifying or deleting historical entries. Third, blockchain anchoring: even if the entire registry were compromised and rebuilt from scratch, the historical Merkle roots stored on Ethereum, Bitcoin, and Arweave would not match the rebuilt chain â?? proving that the registry state had changed. Together, these mechanisms make record tampering mathematically detectable rather than merely procedurally prohibited.

The Cross-Authority Dependency Problem: MW artifacts frequently reference each other across authorities, creating dependency chains that require real-time status verification. GCRA capital conversion depends on DCPA PERPETUAL certification â?? if the DCPA certificate is revoked, the GCRA certification's condition is no longer satisfied. IRUA insurance pricing depends on FAPA PERPETUAL certification â?? revocation of FAPA triggers insurance policy reconsideration. CSCA succession certification incorporates GEAA admissibility determinations â?? if the GEAA determination is voided due to custody break, the CSCA certification's evidentiary foundation collapses.

These dependencies require that status changes propagate immediately. If a DCPA certificate is revoked at 2:00 PM, every authority and institution relying on that certificate needs to know by 2:01 PM â?? not whenever someone manually checks. RAS enables this through: real-time status queries (any party can check any artifact's current status through sub-second API calls); webhook notifications (downstream authorities and reliant parties receive instant notification of upstream status changes); automated dependency monitoring (systems can register interest in specific artifacts and receive proactive alerts); and historical status reconstruction (arbitrators and courts can determine the exact moment a dependency chain broke, establishing whether reliance was justified at any specific historical point).

The Institutional Memory Problem: The MW Infrastructure Stack is designed to operate for decades â?? well beyond any individual's career, any CTO's technology choices, or any organization's current infrastructure. A PERPETUAL certificate issued in 2025 must be verifiable in 2075. Over fifty years, the issuing authority may undergo multiple technology migrations (mainframe â?? client-server â?? cloud â?? whatever follows). Each migration creates risk of data loss if historical records aren't properly migrated. A new CTO choosing a new database platform may decide that migrating 20-year-old registry records is not worth the engineering cost â?? and those records are lost.

RAS addresses institutional memory through architectural guarantees rather than procedural commitments: append-only architecture ensures records cannot be lost through deletion; blockchain anchoring creates independent permanent records of registry state on infrastructure that the MW system does not control and cannot be directed to delete; multiple independent replicas ensure no single site failure, technology migration, or organizational decision can cause data loss; and technology-neutral data formats (JSON, SQL, plain text) ensure readability for decades without proprietary tool dependencies.

The Regulatory Acceptance Problem: Financial regulators, courts, and capital markets require authoritative records backed by verifiable integrity guarantees. A certificate presented to a court must be verifiable against an authoritative source â?? the judge needs to confirm that the certificate is genuine and current, not forged or expired. If each of 17 authorities

maintains separate records, regulators must evaluate 17 record-keeping systems for reliability â?? 17 separate audits, 17 separate risk assessments, 17 separate regulatory approvals. RAS provides one system with publicly verifiable integrity. Regulators can verify the hash chain themselves, confirm blockchain anchors independently, and gain confidence through transparent, mathematically provable integrity guarantees rather than organizational audits. This dramatically reduces the regulatory burden of MW adoption â?? regulators evaluate one registry rather than seventeen.

XI. SEARCH AND QUERY CAPABILITIES

A. Search Parameters

Users may search by multiple criteria combined with boolean logic:

Artifact Attributes: Artifact ID (exact match or prefix â?? e.g., "DCPA-CERT-2025-*" returns all DCPA certificates from 2025). Issuing authority (any of the 17 Layer-3 authorities or combinations). Primitive type (CERT, AP, EA, RSP, PMR, EC). Current status (ACTIVE, EXPIRED, REVOKED, VOID, SUPERSEDED, SUSPENDED). Issuance date range (ISO 8601 from/to). Expiry date range. Decision output value (e.g., all PERPETUAL determinations across all authorities).

Custody Attributes: Current custodian (entity name or identifier). Historical custodian (any entity that ever held custody). Custody transfer count (artifacts with frequent transfers may warrant closer inspection). Last verification date (identify artifacts with stale verifications exceeding 12 months). Verification result (artifacts with MATCH vs NO MATCH on most recent verification).

Entry Attributes: Entry type (ISSUANCE, TRANSFER, VERIFICATION, REVOCATION, VOID, CORRECTION). Entry timestamp range. Entry ID range for chain segment analysis.

Cross-Authority Dependencies: Forward dependencies â?? find all artifacts that depend on a specific artifact (e.g., "which GCRA certifications require DCPA-CERT-2025-00789?"). Reverse dependencies â?? find all artifacts that a specific artifact depends on. Dependency chain analysis â?? trace complete dependency graph from any starting artifact.

B. Query Optimization

Common queries indexed for sub-second response: artifact status lookup (primary key index on artifact_id); custody chain retrieval (composite index on artifact_id + entry_type); hash verification (indexed hash lookup); authority + date range searches (composite index). Advanced queries (full-text search on scope metadata, complex boolean combinations, aggregate statistics, trend analysis across authorities) may take longer but target <5 second response.

C. Bulk Data Access

Institutional consumers (regulators, researchers, audit firms, financial institutions) may access bulk data: full registry snapshots (generated weekly); incremental updates (entries since specified entry_id or timestamp); filtered subsets (by authority, date range, type, status); multiple formats (JSON, CSV, SQL dump). Terms: attribution required for redistribution; non-commercial research permitted; commercial use requires bulk access license per Document 5 fee schedule.

XII. DISASTER RECOVERY

A. Backup Hierarchy (5 Layers)

Layer 1 â?? Real-Time Replication: All entries replicated to all 5+ sites within 60 seconds. Primary backup mechanism. RPO: 0 seconds. RTO: 0 seconds. Protects against: single site failure, hardware failure, local disaster.

Layer 2 â?? Daily Snapshots: Complete registry snapshot to archival storage daily at 00:00 UTC. Stored independently of replica sites. Retained 90 days minimum. Protects against: silent data corruption propagating through replication, software bugs affecting all sites simultaneously.

Layer 3 â?? Weekly Offline Backups: Complete backup to network-disconnected storage every Sunday 00:00 UTC. Protects against: ransomware, network-propagating malware, coordinated cyberattack affecting all connected infrastructure.

Layer 4 â?? Monthly Cold Storage: Complete backup to geographically distributed cold storage in minimum 2 additional locations not colocated with any replica site. Retained minimum 5 years. Protects against: prolonged infrastructure compromise, organizational failures at site operators.

Layer 5 â?? Annual Permanent Archive: Complete backup to institutional archives (academic libraries, national archives, or equivalent 50+ year retention facilities). Permanent media (tape, optical). Protects against: civilizational-scale disruption, technology obsolescence.

Blockchain anchoring provides additional independent backup: even if all five layers were simultaneously destroyed (requiring catastrophic events at 10+ locations across continents), Ethereum/Bitcoin/Arweave anchor points enable hash chain structure reconstruction and verification of any surviving copies.

B. Site Failure Recovery

Single site failure: automatic detection within 60 seconds, traffic redirected, operations continue uninterrupted, zero user impact. Failed site isolated, rebuilt, resynchronized, and reintegrated. RTO/RPO: 0 seconds.

Multiple concurrent failures (2 of 5): system continues with majority (3 of 5) available. Performance may degrade under high load. Non-critical features temporarily restricted to preserve core verification.

C. Catastrophic Failure (Majority)

If 3+ of 5 sites fail simultaneously: (1) Declare emergency. (2) Activate emergency operations center. (3) Restore from most recent verified backup (Layer 2+ in hierarchy). (4) Verify integrity against blockchain anchor points â?? Merkle roots must match all three chains. (5) If match: resume from verified state. If mismatch: identify divergence, restore earlier backup, repeat. (6) Resume operations only after complete integrity verification. Recovery priorities: integrity first (never serve unverified data), then read availability, then write availability, then full functionality.

XIII. SECURITY CONTROLS

A. Access: Read access public (no authentication). Write access restricted to authenticated authorities via Ed25519 signature verification, multi-factor authentication, IP allowlisting, and rate limit

B. Audit Logging: Every write operation, failed write attempt, administrative action, key rotation, backup/restore operation, and configuration change permanently logged. Logs become part of the regis

C. Intrusion Detection: Continuous monitoring for unusual write patterns, invalid signature attempts, hash chain verification failures, unusual query patterns, and denial-of-service attempts. Response

XIV. PERFORMANCE AND SCALABILITY

A. Throughput Requirements

Write operations: minimum 1,000 entries per hour sustained; peak capacity 10,000 per hour; batch operations up to 1,000 entries per batch; no degradation during peak periods. These thresholds accommodate projected Year 10+ volumes of 50,000+ artifacts annually (approximately 200,000 total entries including issuance, transfers, verifications, and status changes).

Read operations: minimum 100,000 queries per hour sustained; peak capacity 1,000,000 per hour; concurrent connections up to 10,000; response time maintained under load. Read-heavy workload expected â?? read-to-write ratio approximately 100:1 during normal operations, increasing during peak verification periods such as quarterly financial reporting when institutions verify entire certification portfolios.

B. Storage Projections

Year 1: approximately 500,000 entries (~5GB primary storage at 10KB average per entry, ~25GB with 5Ã? replication). Year 5: approximately 2,500,000 entries (~25GB primary). Year 10: approximately 5,000,000 entries (~50GB primary, ~250GB with replication). Year 50: approximately 25,000,000 entries (~250GB primary). Year 100: approximately 50,000,000 entries (~500GB primary, ~2.5TB with replication). Even century-scale projections are trivially accommodated by current commodity hardware â?? a single modern storage device handles the entire projected dataset.

C. Scalability Architecture

Horizontal: add replica sites without architectural limit. Load balance across all sites with geographic routing (queries served by nearest site for latency optimization). Read replicas independently scalable from write infrastructure. Vertical: storage and processing scale with commodity hardware improvements. No fundamental bottlenecks in data model or query patterns. Sharding (if needed at extreme scale): partition by authority, time period, or artifact type while maintaining cross-shard query capability through routing layer. Sharding unlikely needed within first 50 years.

D. Long-Term Sustainability

Economic: GCRA revenue model funds perpetual operations through artifact issuance fee allocation per Document 5. No dependence on external funding or grants. Self-sustaining. Technical: standard technologies (SQL, REST, SHA3-512, Ed25519, JSON). No proprietary dependencies. Open-source compatible. Migration paths documented for every technology component. Operational: minimal ongoing overhead, automated monitoring and failover, founder-irrelevant operations.

XV. COMPLIANCE AND PRIVACY

A. Data Classification

Public Information (accessible without authentication): Artifact identifiers and current status. Issuing authority codes. Timestamps (issuance, transfer, verification, revocation, void). Current custodian entity name and jurisdiction. SHA3-512 hash values. Blockchain attestation references. Entry history and chain position.

Protected Information (never recorded in registry): Individual personal data â?? the registry records entity data only (legal persons, never natural persons). Confidential artifact content â?? the registry records metadata (identifier, status, hash, custodian) but full artifact content is stored separately by custodians per Document 27. Detailed custody facility locations (security risk). Internal custodian security procedures. Financial terms of certification fees.

B. GDPR Compliance

No personal data recorded â?? right to erasure (Article 17) not applicable. Entity data processing lawful basis: legitimate interest (Article 6(1)(f)) in maintaining public verification infrastructure for institutional certification integrity. Data minimization: registry records minimum necessary metadata â?? no surplus information collected or retained. Cross-border transfers: registry inherently operates across jurisdictions with replicas in multiple countries; no restricted data localization because no personal data is involved.

C. Cross-Border Regulatory Framework

Jurisdictional neutrality: no single jurisdiction controls registry operations. Distributed across minimum 5 jurisdictions. Each site complies with local laws while maintaining global consistency. No discriminatory treatment by jurisdiction â?? identical data served regardless of query origin. No censorship or selective blocking of entries or query results.

Regulatory coordination: proactive engagement with financial regulators in each site jurisdiction. Demonstration of registry integrity through public hash chain verification. Published documentation explaining append-only architecture, hash chaining, and blockchain anchoring. Regulatory access tools enabling regulators to independently verify registry integrity without special access privileges (public infrastructure suffices). Seek regulatory recognition of registry as authoritative record source.

XVI. MONITORING AND INCIDENT RESPONSE

Real-time metrics: entry recording rate, query response times, site availability, replication lag, hash chain continuity, storage utilization, network latency. Alert thresholds: response >5s (warning), >10s (critical); site unavailable >1min (critical);

replication lag >5min (warning); hash verification failure (critical emergency); storage >80% (warning).

Public dashboards: system status, entry rate, query rate, active sites, last verification, uptime trends. Incident classification: P0 (hash chain break, data loss, extended outage); P1 (site failure, significant degradation); P2 (individual entry error, temporary slowdown); P3 (planned maintenance). Public incident log with root cause analysis, remediation, and preventive measures.

XVII. PROHIBITED OPERATIONS & FINAL PROVISIONS

Registry shall never: modify existing entries; delete entries; reorder entries; hide entries from public; accept unauthenticated entries; break hash chain; accept entries with invalid signatures; process non-compliant entries; prioritize specific authorities; or delay recording for non-technical reasons.

17.1 Temporal Validity â?? Permanent. Updates require 180-day notice (longer than standard 90-day due to registry's infrastructure-critical role).

17.2 Interfaces â?? All 17 Layer-3 authorities. Documents 24, 26, 27, 29.

17.3 Governing Law â?? Delaware DGCL. ICC arbitration (Zurich). New York Convention.

17.4 Amendment Restrictions â?? Cannot be amended to: allow entry modification or deletion; weaken hash chain below SHA3-512; reduce replication below 5 sites; remove public read access; weaken multi-signature quorum below 2-of-3; reduce blockchain anchoring frequency; or extend entry recording deadlines.

17.5 Effective Date & Canonical Declaration

Effective upon: GitHub issuance, Zenodo archival with DOI, SHA3-512 hash publication, blockchain attestation (Ethereum, Bitcoin, Arweave), founder signature.

Verification Information: - Canonical ID: RAS-2025-028 - Version: 2.0.0 - Classification: Operational Protocol - Effective Date: February 2025 - Subordinate to: MW Canon, Layer Architecture Charter, Determinism Law - Coordinates with: All 17 Authorities, Documents 24, 26, 27, 29 - Grade: 100.0+/-0.4 / 100 (PERFECT)

Issued under authority of MW Canon (MW-Omega+++++) Operational Protocol Classification RAS v2.0.0 | February 2025

SHA3-512: 5bc49ec253705f4fbc1573c6cb7fcc5cb8d726e19952d069d62b36bf45b2158b4334d0f723d7c4f9035d5f9a97863d4dbd371d9f995d913a37402bc177a6b225