Omne: A Dual-Layer Blockchain Architecture for Commerce-Grade Decentralized Infrastructure

Greg B — Omne Foundation (OmneDAO)

March 2026

Abstract

We present Omne, a layer-one blockchain designed to serve as infrastructure for global digital commerce. Omne introduces a dual-layer consensus architecture comprising a 3-second commerce layer for point-of-sale finality and a 9-minute security layer for settlement-grade guarantees, unified through Proof of Validated Execution and Resource Attestation (PoVERA). The protocol employs a dual-token model: OMC, an inflationary commerce token with utilization-responsive fee burning and 18-decimal (quar) precision enabling sub-cent transaction costs, and OGT, a deflationary governance token with a 21-million-unit hard cap governed by a fixed-supply halving schedule. A native smart contract pipeline executes WASM bytecode deterministically via the Axiom Runtime across heterogeneous hardware, with SHA-256 state commitments embedded in every security block. The Omne Orchestration Network (OON) extends the validator infrastructure into an off-chain distributed compute marketplace, producing cryptographic attestation certificates and cross-subsidizing on-chain transaction fees at a rate of 30% of compute revenue. The Omne Media Protocol (OMP) provides decentralized file storage through a hybrid architecture: asset manifests, SHA-256 merkle roots, and storage proofs live on-chain in consensus state, while raw chunk data is distributed across dedicated storage nodes with configurable redundancy and erasure coding. Anti-spam protections rely exclusively on non-fee controls—rate limiting, stake-weighted quality-of-service, throttling, and circuit breakers—ensuring that microscopic fees remain a permanent architectural property rather than a temporary subsidy. We describe the protocol's consensus mechanism, token economics, fee model, execution environment, smart contract system, compute orchestration layer, decentralized storage protocol, and security model in sufficient detail for independent implementation.

1 Introduction

The promise of blockchain-based commerce has been constrained by a fundamental tension between the properties that make distributed ledgers trustworthy and the requirements that make payment systems usable. A merchant processing a point-of-sale transaction requires confirmation within seconds. A developer deploying a token contract should not pay more in gas than the contract's first hundred users will transact. A small business owner should not watch their $50 transfer lose $4.70 to fee volatility between mempool submission and block inclusion.

Existing layer-one protocols have optimized for one end of this spectrum at the expense of the other. High-throughput chains achieve low latency by weakening finality guarantees, creating reorg risk that is unacceptable for commerce. Security-first chains achieve strong finality at the cost of confirmation times measured in minutes and fee markets that auction block space to the highest

bidder. No production chain has solved this trilemma: sub-second commercial finality, settlement-grade security, and deterministically microscopic fees.

Omne addresses this problem through three architectural decisions. First, a dual-layer consensus mechanism separates the commerce confirmation path (3-second blocks with instant finality) from the security settlement path (9-minute blocks binding 180 commerce blocks into a single cryptographic commitment). Second, a dual-token economic model decouples the medium of exchange (OMC) from the governance and staking instrument (OGT), eliminating the single-token trilemma in which a token cannot simultaneously serve as a scarce store of value, a low-friction medium of exchange, and a governance weight. Third, anti-spam protection is implemented entirely through non-fee controls—rate limiting, stake-weighted quality-of-service, throttling, and circuit breakers—so that transaction fees can be set at the cost floor rather than the market-clearing price.

Our central hypothesis is that these three constraints—dual-layer consensus, dual-token economics, and non-fee anti-spam—are jointly sufficient to achieve commerce-grade latency without compromising settlement-grade security, and that transaction fees can be made permanently microscopic through structural economic design rather than temporary subsidization. We further hypothesize that compute orchestration and decentralized storage can be integrated at the protocol level, producing utilization-driven revenue streams that reinforce fee sustainability without introducing external dependencies.

This paper describes the Omne protocol in sufficient detail for independent implementation and economic analysis. Each section presents the design rationale, the implemented mechanism, and the observable properties that follow from the design.

2 Architecture Overview

Omne is structured as a single logical chain with two block-production cadences operating in parallel. The commerce layer produces blocks every 3 seconds, targeting point-of-sale and e-commerce confirmation latency. The security layer produces blocks every 540 seconds (9 minutes), each embedding a cryptographic commitment over the 180 commerce blocks produced during that interval. Validators participate in both layers through a unified Proof of Validated Execution and Resource Attestation (PoVERA) consensus mechanism.

Transaction execution occurs in the Axiom Runtime, a WASM-based deterministic execution environment that guarantees identical state transitions across heterogeneous hardware. Every security block contains an `ExecutionStateCommitment` comprising a SHA-256 state root, an execution trace hash, and a per-tier gas budget report, binding the execution history of the preceding commerce epoch to the security chain.

The Omne Orchestration Network (OON) extends the validator infrastructure into an off-chain compute marketplace. Registered nodes rent idle capacity for workloads such as AI training, 3D rendering, and data processing. Compute revenue cross-subsidizes on-chain transaction fees at a protocol-defined rate of 3,000 basis points (30%), creating a sustainable funding mechanism for microscopic fees without relying on inflationary pressure.

The Omne Media Protocol (OMP) extends the network with decentralized file storage. Files are serialized into 256 KB chunks, content-addressed via SHA-256, and distributed across dedicated storage nodes with configurable redundancy (2–7$\times$ replication) and optional Reed-Solomon erasure coding. Storage nodes bond 500 OGT, build reputation through proof-of-storage challenges,

and earn fees via escrow-based periodic settlement. Asset manifests and merkle roots are maintained on-chain; raw chunk data lives off-chain on storage nodes. This hybrid architecture avoids bloating consensus state while providing on-chain integrity guarantees for all stored data.

3 Dual-Layer Consensus

3.1 Commerce Layer

The commerce layer produces blocks at a fixed cadence of 3 seconds (`COMMERCE_BLOCK_TIME_SECS = 3`). Each commerce block contains transactions prioritized by a three-tier system: Commerce priority (weight 3), Standard priority (weight 2), and Compute priority (weight 1). Block execution budgets allocate 70% of available time to FastVM transactions (commerce), 20% to StandardVM, and 10% to ComputeVM.

Commerce blocks achieve probabilistic finality upon production and deterministic finality upon inclusion in a security block. For point-of-sale use cases, the 3-second probabilistic finality is sufficient; the merchant receives confirmation before the customer leaves the counter.

3.2 Security Layer

The security layer produces blocks every 540 seconds (9 minutes), with each security block spanning exactly 180 commerce blocks:

$$\frac{540 \text{ s}}{3 \text{ s/block}} = 180 \text{ commerce blocks per security block}$$

Each security block contains:

Field	Description
`commerce_state_commitment`	Merkle commitment over 180 commerce blocks
`execution_state_commitment`	SHA-256 state root, execution trace hash, budget report hash
`validator_rewards`	Per-validator reward breakdown for the epoch
`block_fees`	Aggregated fee accounting (collected, burned, distributed)
`slashed_stake_ogt`	Total OGT slashed during the epoch
`vdp_commitments`	Verifiable-Deterministic-Parallelizable attestations
`computational_revenue_quar`	OON compute revenue for cross-subsidization
`storage_fee_revenue_quar`	OMP storage fee revenue for the epoch

The 9-minute cadence was chosen to balance two constraints: the interval must be long enough to amortize the cost of BFT finality across many commerce transactions, and short enough that the security chain provides meaningful settlement guarantees within a time frame comparable to the ~10-minute cadence characteristic of security-first proof-of-work chains.

3.3 Commerce-to-Security Rollup

At the close of each 180-block commerce epoch, the block producer constructs an `Execution-StateCommitment`:

```
ExecutionStateCommitment {
    state_root:           SHA-256(state trie),
    execution_trace_hash: SHA-256(deterministic execution trace),
    budget_report_hash:   SHA-256(per-tier gas budget report),
    source_block_height:  commerce block height
}
```

This commitment is embedded in the subsequent security block, creating an immutable binding between the commerce execution history and the security chain. Any validator can independently reconstruct the commitment by replaying the 180 commerce blocks, providing a deterministic verification path.

4 PoVERA Consensus

Proof of Validated Execution and Resource Attestation (PoVERA) is Omne's consensus mechanism. It combines four components:

1. **Proof of Stake (PoS):** Validators bond OGT within the dynamic stake range of 15–28 OGT (`MIN_VALIDATOR_STAKE = 15`, `MAX_VALIDATOR_STAKE = 28`). The active validator set is capped at 100 nodes. Delegators may participate with a minimum of 1 OGT.
2. **Proof of Valuable Computation (PoVC):** Validators that contribute compute capacity through the Omne Orchestration Network earn attestation certificates, which factor into block producer selection. This creates a direct incentive for validators to make surplus capacity available for off-chain workloads.
3. **RANDAO Randomness:** Block producer selection uses a RANDAO-derived beacon, weighted by stake, to ensure pseudorandom and manipulation-resistant leader election.
4. **BFT Finality:** The security layer applies Byzantine fault-tolerant finality, ensuring that once a security block is committed, it cannot be reverted without collusion of more than one-third of staked weight.

Validators are rewarded per security block according to a deterministic schedule (Section 5). Misbehavior is penalized through slashing: minor violations (downtime exceeding 100 blocks) incur a 5% stake penalty, while major violations (double signing) incur a 50% penalty. Slashed validators are jailed for 1,000 blocks (~50 minutes at 3-second commerce cadence).

5 Dual-Token Economics

5.1 Design Rationale

A single-token protocol forces one asset to serve simultaneously as a medium of exchange, a store of value, and a governance weight. These roles impose contradictory pressures: a medium of exchange must be abundant and low-friction; a store of value must be scarce; a governance weight must be illiquid enough to align long-term incentives. Omne separates these roles across two tokens.

5.2 OGT: Omne Governance Token

OGT is Omne's deflationary governance and staking token, governed by a fixed-supply halving schedule with hard cap.

Supply schedule. The initial block reward is 50 OGT per security block. The reward halves every 233,760 security blocks, an interval derived from the 9-minute block cadence:

$$233{,}760 \text{ blocks} \times 540 \text{ s/block} = 126{,}230{,}400 \text{ s} \approx 4.0 \text{ years}$$

The halving schedule proceeds through four reductions, after which a permanent floor of 3 OGT per block is maintained:

Epoch	Blocks	Reward (OGT)	Cumulative Supply (OGT)
0	0–233,759	50	11,688,000
1	233,760–467,519	25	17,532,000
2	467,520–701,279	12	20,333,120
3	701,280–935,039	6	21,735,680
4+	935,040+	3 (floor)	Asymptotic toward cap

The hard cap is 21,000,000 OGT (`OGT_MAX_SUPPLY = 21_000_000`). After the four halvings, the 3 OGT floor ensures continued validator compensation while maintaining supply within the cap through a diminishing issuance rate.

Staking and bonding. Validators must bond between 15 and 28 OGT. This dynamic range prevents both barrier-to-entry problems (stake too high) and Sybil attacks (stake too low). Delegation is supported with a minimum of 1 OGT. Beyond validator staking, OGT serves as the bonding currency for protocol infrastructure operators: OON compute nodes bond 1,000 OGT and OMP storage nodes bond 500 OGT, creating a unified economic commitment model across all network roles.

5.3 OMC: Omne Commerce Token

OMC is the medium of exchange and fee currency. It is inflationary by design, with a burning mechanism that provides deflationary counterbalance proportional to network utilization.

Genesis supply. 1,000,000 OMC is minted at genesis to the OmneDAO treasury (`omne1000000000000000000000`). There is no maximum supply.

Block reward. The initial OMC reward is 2 OMC per security block. Rewards follow the same halving schedule as OGT (233,760 blocks, 4 halvings):

Epoch	OMC Reward/Block
0	2.000
1	1.000
2	0.500
3	0.250

Epoch	OMC Reward/Block
4+	0.125 (floor)

Precision. OMC uses 18-decimal precision:

$$1 \text{ OMC} = 10^{18} \text{ quar}$$
$$1 \text{ quar} = 10^{6} \text{ micro-quar}$$

The smallest unit of account is 1 micro-quar (10^{-24} OMC). This precision enables fee calculations at scales where rounding errors would otherwise dominate.

Treasury faucet and buy-back. The OmneDAO treasury operates a band peg mechanism for OMC price stability. The treasury sells OMC at a fixed ceiling price (faucet), providing a supply expansion valve when demand exceeds target. The treasury buys back and burns OMC at a fixed floor price, providing a contraction valve when supply exceeds demand. This mechanism bounds OMC volatility within a governance-defined band without requiring external collateral.

6 Fee Economics

6.1 Design Principles

Omne's fee model is built on the premise that transaction fees should cover the marginal cost of execution, not the market-clearing price of block space. Fee revenue is explicitly not a protocol goal. The system cross-subsidizes low fees from three sources: OMC block rewards to validators, OON compute revenue (30% of which flows to fee subsidization), and OMC burning that reduces circulating supply.

6.2 Base Fee

The initial base fee is 20 micro-quar per unit of gas, dynamically adjusted based on block utilization with a target utilization of 50%:

$$\text{base\_fee} = 20 \text{ micro-quar/gas}, \quad \text{clamped to } [1, 100] \text{ micro-quar/gas}$$

In OMC terms, the base fee is:

$$20 \text{ micro-quar} = 20 \times 10^{-6} \text{ quar} = 20 \times 10^{-24} \text{ OMC}$$

At typical gas costs for a simple transfer (~21,000 gas), the fee is approximately 4.2×10^{-19} OMC—effectively zero in human-denominated terms.

Commerce discount. Transactions on the commerce layer receive a 50% fee reduction (`commerce_multiplier = 0.5`), reflecting the protocol's prioritization of commercial use cases.

6.3 Fee Burning

Omne implements utilization-responsive fee burning with a congestion-dependent burn rate:

Block Utilization	Burn Rate
Below 50% (low)	10% of fees burned
50%–80% (moderate)	70%–95% of fees burned
Above 80% (high)	95% of fees burned

Under sustained high utilization, the burning rate approaches 95%, creating strong deflationary pressure on OMC supply. The remaining fees flow to validators and the DAO treasury according to the protocol's fee split configuration.

6.4 Revenue Cross-Subsidization

The Omne Orchestration Network generates revenue from off-chain compute jobs. A protocol-defined subsidy rate of 3,000 basis points (30%) of this revenue is routed to fee subsidization via PoVERA's `record_attested_job()` pathway. Revenue is smoothed over a rolling window of 12 security blocks (`REVENUE_SMOOTHING_WINDOW = 12`) to prevent fee volatility from compute demand spikes.

Storage fees collected by the Omne Media Protocol contribute to the same economic model. Escrow payments from asset uploaders flow to storage nodes over the lifetime of the stored assets, and the utilization-responsive burn mechanism applies to the on-chain component of storage manifest transactions. The combination of compute revenue (OON) and storage revenue (OMP) produces two independent, utilization-driven revenue streams that reinforce the system's ability to maintain microscopic commerce fees without subsidy expiration.

7 Deterministic Execution: The Axiom Runtime

7.1 Design Constraints

A commerce-grade blockchain requires that every validator executing the same transaction sequence arrives at the same state root, regardless of hardware architecture, operating system, or compiler version. Nondeterministic execution is not merely a bug—it is a consensus failure. If two validators compute different state roots for the same block, the chain forks.

7.2 WASM Execution

The Axiom Runtime executes WebAssembly (WASM) bytecode in a sandboxed environment with deterministic memory allocation, floating-point behavior, and gas metering. WASM was chosen because it provides a well-specified, portable instruction set with no implicit platform dependencies. The runtime strips all sources of nondeterminism: system clocks, random number generators, and environment variables are replaced with deterministic host functions.

7.3 State Commitment

After each commerce block, the runtime produces a SHA-256 hash of the complete WASM memory state. At the security layer boundary, the `ExecutionStateCommitment` binds the state root, the execution trace hash, and the gas budget report into the security block:

$$\text{state\_root} = \text{SHA-256}\big(\text{state trie after 180 commerce blocks}\big)$$

Any validator can verify the commitment by replaying the commerce epoch and comparing the resulting hash. Disagreement triggers the slashing protocol.

7.4 Three-Tier VM

The Axiom Runtime provides three execution tiers, each with distinct resource budgets:

Tier	Timeout	Gas Limit	Memory	Block Budget
FastVM	400 ms	1,000,000	100 MB	70%
StandardVM	3 s	10,000,000	1 GB	20%
ComputeVM	30 s	100,000,000	4 GB	10%

Transactions are assigned to a tier at submission. Commerce-priority transactions execute in FastVM; general application logic in StandardVM; heavy computation (machine learning inference, cryptographic proofs) in ComputeVM. Tier budgets are enforced per-block, preventing compute-heavy workloads from crowding out commerce transactions.

8 Smart Contracts

8.1 Contract Deployment

Smart contracts are deployed via signed execution plans submitted through the `omne_deployContract` RPC endpoint. An execution plan contains the WASM bytecode, a designated entry function, the target VM tier, and deployment metadata. The contract address is derived deterministically:

$$\text{address} = \text{SHA-256}\big(\text{bytecode} \| \text{name} \| \text{entry\_function} \| \text{export} \| \text{nonce}\big)$$

Deployed contracts are stored in the `ContractBytecodeRegistry` with the following fields:

Field	Description
`wasm_bytecode`	Compiled WASM binary (max 16 MB)
`bytecode_hash`	SHA-256 of the bytecode
`vm_tier`	Execution tier (FastVM, StandardVM, ComputeVM)
`entry_function`	Entry point function name
`export_name`	Exported WASM function identifier
`deploy_block`	Block height at deployment

Field	Description
`deployment_nonce`	Deduplication counter

8.2 Omne ABI

Omne uses a custom binary wire format for contract invocation rather than adopting existing account-model ABI conventions. Because the Axiom Runtime executes WASM natively, a purpose-built encoding that maps directly to WASM function signatures avoids the complexity of virtual machine translation layers designed for stack-based architectures. The wire format is:

```
[MAGIC: 4 bytes "OMNE" (0x4F4D4E45)]
[VERSION: 1 byte (0x01)]
[METHOD_NAME_LENGTH: 2 bytes, big-endian]
[METHOD_NAME: UTF-8 encoded string]
[ARG_COUNT: 2 bytes, big-endian]
[ARGUMENTS: type_tag (1 byte) + length (4 bytes) + data ...]
```

Nine argument types are supported:

Tag	Type	Size
0x01	U32	4 bytes
0x02	U64	8 bytes
0x03	I32	4 bytes
0x04	I64	8 bytes
0x05	String	Variable (length-prefixed)
0x06	Bytes	Variable (length-prefixed)
0x07	Bool	1 byte
0x08	Address	20 bytes
0x09	U128	16 bytes

Maximum calldata size is 4 MB. Maximum argument count per call is 64. Encoding and decoding are deterministic across all validators.

8.3 Contract Interaction

State-modifying calls. A transaction targeting a deployed contract address submits ABI-encoded calldata in the transaction's `data` field via `omne_sendTransaction`. The chain loads the contract's WASM bytecode from the registry, decodes the method name and arguments from the ABI envelope, and routes execution to the appropriate entry function within the Axiom Runtime.

Read-only queries. The `omne_call` RPC endpoint executes contract methods without committing state changes or consuming gas. This enables free balance lookups, metadata queries, and view functions.

Cross-contract calls. The `omne_host::call_contract` host function enables one deployed contract to invoke another, providing the foundation for composable multi-contract applications.

8.4 SDK

The TypeScript SDK provides the `OmneContract` class for client-side contract interaction:

- `query(method, args)` executes a read-only call via `omne_call` and returns the decoded result.
- `call(options)` submits a state-modifying transaction and returns a `TransactionReceipt`.
- `AbiEncode` builders and `encodeContractCall()` produce correctly formatted ABI payloads.
- `isAbiEncoded()` validates that a byte array conforms to the Omne ABI wire format.

9 Omne Orchestration Network

9.1 Architecture

The Omne Orchestration Network (OON) is an off-chain distributed compute layer built directly into the protocol. Six modules—orchestrator, node registry, job distribution, coordination, VDP compliance, and attestation—are wired through a central `CoordinationManager`. Four RPC endpoints expose the system: `omne_submitJob`, `omne_getJobStatus`, `omne_registerNode`, and `omne_getNodeInfo`.

The design rationale is straightforward: the same validator infrastructure that secures the chain possesses compute capacity that sits idle between block production duties. Rather than leaving this capacity unused, OON allows validators (and third-party compute providers) to rent it for off-chain workloads, creating a revenue stream that directly subsidizes on-chain transaction fees.

9.2 Job Submission and Packaging

Jobs are submitted as self-describing `ComputeJobSubmission` envelopes containing:

- WASM bytecode (max 50 MB) with a SHA-256 integrity hash
- Input data (max 10 MB) with a SHA-256 integrity hash
- Entry function name
- Job type classification (Rendering3D, AiTraining, DataProcessing, Simulation, Cryptographic, or Custom)
- Resource hints (CPU cores, memory, GPU availability, storage, bandwidth)
- Maximum execution time and expected output size

Both the WASM bytecode hash and input data hash are verified at submission time. Any mismatch triggers deterministic rejection.

9.3 Chunked Execution

Long-running jobs execute in iterative 30-second time slices within the ComputeVM tier. Each chunk produces an `ExecutionCheckpoint` containing a memory snapshot that enables resumption after preemption or node reassignment. Budget enforcement operates across three dimensions:

Budget	Default Limit
Maximum chunks	4,320 (= 36 hours at 30 s/chunk)
Total gas	Configurable per job
Total wall time	Configurable per job

This model allows compute-intensive workloads—AI training runs, physics simulations, cryptographic proof generation—to execute within the protocol's deterministic framework without blocking commerce transactions.

9.4 Node Registry and Reputation

Compute nodes register with a bond of 1,000 OGT and declare their capabilities:

Capability	Description
`cpu_cores`	Available CPU cores
`memory_gb`	Available RAM
`gpu_available`	GPU compute support
`storage_gb`	Available storage
`bandwidth_mbps`	Network bandwidth
`vdp_compliant`	Supports Verifiable-Deterministic-Parallelizable framework

Nodes build reputation through job completion. The `ReputationHistory` track record factors reliability (completed vs. failed and timed-out jobs), performance, and quality into an overall reputation score. The minimum threshold for job assignment is 0.7 (`min_reputation_score = 0.7`). Nodes falling below this threshold are excluded from job distribution until their score recovers.

9.5 Dispatch and Verification

The `NodeDispatcher` routes sub-tasks to registered nodes via their endpoint URLs, enforcing a per-node concurrency cap of 10 concurrent jobs and a maximum payload size of 50 MB. Results are collected with timeout enforcement and retry logic (exponential backoff, max 5 attempts, 15-second initial / 900-second maximum backoff).

Every job result undergoes VDP compliance verification using one of four methods:

Method	Description
ZkProof	Zero-knowledge proof of correct execution
StateHash	SHA-256 hash comparison of output state
Replication	Independent re-execution on a second node
Benchmark	Performance comparison against known baselines

9.6 Attestation and Settlement

Completed and verified jobs produce cryptographic `ProofOfWork` certificates containing the work description, resource usage metrics, performance metrics, a computational proof, a VDP verification result, and an ed25519 signature. Certificates are valid for 24 hours.

Settlement occurs deterministically via the `SettlementManager`:

1. At job submission, the submitter's payment is locked in escrow.
2. Upon successful verification, a batch settlement transaction releases payment to the compute node. Transaction IDs are derived deterministically: `SHA-256("omne:oon:settlement:{batch_id}:{j`
3. If VDP verification fails post-settlement, a dispute mechanism claws back payment.

Compute revenue flows to fee subsidization at 3,000 BPS (30%) via PoVERA's `record_attested_job()` pathway. Consecutive job timeouts (default threshold: 3) trigger validator slashing through the PoVERA consensus mechanism.

10 Omne Media Protocol

Prior work in decentralized storage has produced protocols that operate as separate networks, each with its own token, consensus mechanism, and validator set. An application that requires both computation and storage must bridge between a smart contract chain and a storage chain, introducing latency, trust assumptions, and token conversion friction. We observe that this separation is an artifact of historical protocol design, not a fundamental constraint. Omne eliminates the boundary by building file storage directly into the protocol. Storage nodes participate in the same OGT staking and reputation infrastructure as validators and compute nodes, creating a unified economic model where compute, consensus, and storage are facets of a single network.

10.1 Architecture

OMP uses a hybrid on-chain / off-chain architecture. *Manifests* — asset metadata, SHA-256 content hashes, merkle roots, chunk assignments, and storage proof records — live on-chain in consensus state, benefiting from the security layer's finality guarantees. *Raw chunk data* lives off-chain on dedicated storage nodes, avoiding the state bloat that would make on-chain storage prohibitively expensive at scale. The chain enforces data integrity; storage nodes provide data availability.

Nine coordinating modules — coordinator, storage registry, manifest manager, chunk distributor, chunk store, storage proofs, retrieval coordinator, settlement manager — are wired through a central `OmpCoordinator` that provides thread-safe access from the RPC layer and block execution pipeline. Six RPC endpoints expose the protocol: `omne_ompStoreAsset`, `omne_ompFinalizeAsset`, `omne_ompGetManifest`, `omne_ompRegisterStorageNode`, `omne_ompGetRetrievalPlan`, and `omne_ompStorageStats`.

10.2 Asset Lifecycle

The lifecycle of a stored file proceeds in four stages:

1. Manifest creation. The uploader submits a `storeAsset` request specifying the file's SHA-256 content hash, total size, chunk count, merkle root, redundancy level (2–7×, default 3×), storage

tier (Hot, Warm, or Cold), and an optional erasure codec (Reed-Solomon or none). The protocol validates size constraints (maximum 1 GB per asset, `MAX_ASSET_SIZE = 1{,}073{,}741{,}824` bytes), redundancy bounds, and hash lengths, then creates an on-chain manifest in `Uploading` status and locks an OGT escrow for the estimated storage cost.

2. Chunk submission. The file is serialized into 256 KB chunks (`CHUNK_SIZE = 262{,}144` bytes, maximum 4,096 chunks per asset). Each chunk is content-addressed via SHA-256 and submitted to the chunk store with integrity verification on write. The chunk distributor assigns each chunk to `redundancy` distinct storage nodes using round-robin placement sorted by available capacity, ensuring uniform load distribution.

3. Finalization. Once all chunks are submitted, the uploader calls `finalizeAsset`. The protocol recomputes a binary SHA-256 merkle tree from the submitted chunk hashes and verifies the root against the declared merkle root in the manifest. A mismatch — indicating data corruption or tampering during upload — causes deterministic rejection. On success, the manifest transitions to `Finalized` status and becomes eligible for retrieval and proof-of-storage challenges.

4. Retrieval. Clients request a `RetrievalPlan` that maps each chunk index to an ordered list of storage node endpoints holding that chunk. Chunks can be fetched in parallel from multiple nodes. The client verifies each downloaded chunk against the SHA-256 hash from the on-chain manifest, providing end-to-end integrity without trusting any individual storage node.

10.3 Storage Nodes and Reputation

Storage nodes register by bonding a minimum of 500 OGT (`STORAGE_NODE_BOND_OGT = 500`) and declaring capabilities:

Capability	Description
`storage_gb`	Available storage capacity
`upload_bandwidth_mbps`	Upload throughput
`download_bandwidth_mbps`	Download throughput
`supports_hot_tier`	Immediate retrieval support
`supports_cold_tier`	Archival storage support
`region`	Geographic region for latency optimization

The maximum storage node set is 10,000 nodes (`MAX_STORAGE_NODES = 10{,}000`). Each node begins with a reputation score of 1.0 and is recalculated based on proof-of-storage outcomes:

$$\text{reputation} = \frac{\text{proofs\_passed}}{\text{proofs\_passed} + \text{proofs\_failed}}$$

Nodes falling below a reputation threshold of 0.3 are automatically suspended and excluded from new chunk assignments. Only nodes with reputation ≥ 0.5 receive chunk distribution assignments. The reputation system creates a natural selection pressure: reliable nodes accumulate chunks and earn fees; unreliable nodes lose assignments and eventually forfeit their bond.

10.4 Proof of Storage

Validators periodically challenge storage nodes to prove they still hold specific chunks. The challenge-response protocol operates as follows:

1. The challenger (a validator) generates a random nonce and constructs a `StorageChallenge` with a deterministic challenge ID:

$$\text{challenge\_id} = \text{SHA-256}(\text{node\_id} \| \text{asset\_id} \| \text{chunk\_index} \| \text{nonce} \| \text{timestamp})$$

2. The challenged storage node computes the expected proof response:

$$\text{proof} = \text{SHA-256}(\text{chunk\_hash} \| \text{nonce})$$

3. The validator verifies the response by recomputing the same hash from the on-chain chunk hash (known from the manifest) and the nonce it issued. Three outcomes are possible:

Verdict	Condition	Consequence
Valid	Proof matches expected hash	Proofs passed counter incremented
Invalid	Proof does not match	Reputation penalty; consecutive failures trigger suspension
Expired	Response received after deadline	Treated as failure; reputation penalty

The default challenge interval is 3,600 seconds (`PROOF_CHALLENGE_INTERVAL_SECS = 3{,}600`). Hot-tier assets are challenged more frequently than cold-tier, reflecting the higher availability requirements.

10.5 Storage Tiers

OMP supports three storage tiers, each with distinct performance and cost characteristics:

Tier	Retrieval Latency	Cost	Proof Frequency	Use Case
Hot	Immediate	Highest	Most frequent	Real-time media, application assets
Warm	Moderate	Balanced	Standard	Documents, historical data
Cold	High	Lowest	Least frequent	Backups, archival, compliance records

Tier selection is made at manifest creation and affects pricing, proof challenge frequency, and the storage node eligibility filter (only nodes declaring tier support receive assignments for that tier).

10.6 Settlement

Storage fees use an escrow-based periodic drip model. When an asset manifest is created, the uploader's OGT is locked in escrow. Over the storage period, escrow funds are released incrementally to the storage nodes holding the asset's chunks. Settlement IDs are derived deterministically:

$$\text{settlement\_id} = \text{SHA-256}\big(\texttt{"omne:omp:settlement:\{asset\}:\{node\}:\{block\}"}\big)$$

This deterministic derivation ensures that any validator can independently reconstruct the settlement history from the chain state. When the escrow is fully released, the settlement record is marked complete. If a storage node fails proof-of-storage challenges, the corresponding portion of the escrow is slashed rather than released, aligning storage node incentives with data availability.

10.7 Design Rationale

Building storage into the protocol rather than delegating it to a separate network eliminates three classes of problems:

1. **Bridge risk.** External storage networks require bridge contracts, oracles, or multi-sig committees to relay storage proofs to the application chain. Each bridge is an attack surface. OMP's proofs are verified natively by the same validators that produce blocks.
2. **Token friction.** Separate storage networks use separate tokens. Users must acquire, manage, and convert between tokens to use storage and compute together. OMP uses OGT for bonding and OMC for fees — the same tokens used for everything else on the network.
3. **Incentive fragmentation.** When storage and compute are on different networks, there is no mechanism for cross-subsidization. OMP storage fees contribute to the same economic system that keeps commerce fees microscopic: storage manifest transactions contribute to fee burning, and storage node bonds contribute to the OGT staking economy.

11 Security Model

11.1 Anti-Spam Controls

Omne does not rely on fee escalation to deter spam. Instead, the protocol enforces layered non-fee controls:

Mempool rate limiting. Transactions are rate-limited at three granularities:

Scope	Rate	Burst
Per account	60 tx/s	20 tx burst
Per source IP	300 tx/s	75 tx burst
Global	8,000 tx/s	1,000 tx burst

The mempool is capped at 10,000 pending transactions with a maximum payload of 512 KB per transaction.

Stake-weighted QoS. Validators with higher stake receive proportionally higher priority in block production and message propagation, ensuring that well-capitalized honest validators can always outcompete an attacker with equal or lesser stake.

Circuit breakers. Under sustained overload, the protocol activates circuit breakers that shed low-priority traffic while preserving consensus and security message throughput.

11.2 Slashing

Violation	Penalty	Jail Duration
Minor (downtime > 100 blocks)	5% of stake	1,000 blocks (~50 min)
Major (double signing)	50% of stake	1,000 blocks (~50 min)

Slashed OGT is burned and tracked in the security block's `slashed_stake_ogt` field, providing a public audit trail.

11.3 Cryptography

Omne standardizes on Ed25519 for all signature operations across the protocol: transaction signing, block signing, validator identity, OON attestation certificates, OMP storage proof verification, and cross-contract message authentication. Ed25519 was chosen for its performance characteristics (batch verification), resistance to side-channel attacks, and absence of the nonce-reuse vulnerabilities that affect elliptic-curve schemes requiring per-signature randomness.

12 Governance

The Omne protocol is governed by OmneDAO, a Swiss Foundation. Protocol parameters—fee multipliers, burn rates, stake ranges, slashing penalties, OON subsidy rates—are adjustable through OGT-weighted governance proposals. The DAO treasury, seeded with 1,000,000 OMC at genesis, funds protocol development, ecosystem grants, and the OMC band peg mechanism.

OGT holders participate in governance with voting power equal to their balance plus any locked (staked) tokens. This ensures that validators—who have the most direct operational interest in the protocol's health—hold proportional governance weight.

13 Ecosystem: The Three-Tier Token System

Omne supports a three-tier token architecture:

1. **OMC (Infrastructure).** The base-layer medium of exchange and fee currency. Used for all on-chain transactions, smart contract interactions, OON compute payments, and OMP storage manifest fees.
2. **OGT (Governance).** The staking, slashing, and governance token. Used for validator bonding (15–28 OGT), OON compute node registration (1,000 OGT), OMP storage node registration (500 OGT), and protocol-level voting.

3. **ORC-20 (Application).** A token standard for application-specific utility tokens deployed as smart contracts. ORC-20 tokens inherit OMC-denominated gas costs automatically. The standard defines:
    - Core operations: `balance_of`, `transfer`, `approve`, `transferFrom`
    - Governance integration: `lock_tokens`, `unlock_tokens`, `voting_power`
    - Configuration: maximum supply, mintability, burnability, transfer controls (blacklists, per-transaction limits, daily volume caps)
    - Fee sharing: configurable platform revenue sharing and transfer fee rates

This layered design enables application developers to issue tokens with custom economics while inheriting Omne's fee infrastructure and security guarantees.

14 Observations and Conclusions

The architecture described in this paper yields several observable properties that address the hypotheses stated in Section 1.

Observation 1: Latency-security decoupling is achievable without finality compromise. The dual-layer design separates the 3-second commerce confirmation path from the 9-minute security settlement path. Commerce blocks provide the latency characteristics required for point-of-sale and e-commerce. Security blocks provide the settlement guarantees required for high-value transfers and institutional use cases. The 180:1 rollup ratio binds the two layers cryptographically without requiring either to weaken its guarantees.

Observation 2: Dual-token economics eliminate the single-token trilemma. By separating the medium of exchange (OMC) from the governance and staking instrument (OGT), the protocol avoids the contradictory pressures that force single-token designs to choose between scarcity, liquidity, and governance alignment. OGT follows a deflationary halving schedule; OMC is inflationary with a utilization-responsive burn counterbalance. Neither token is asked to serve a role that conflicts with its supply mechanics.

Observation 3: Microscopic fees are sustainable as a structural property. Transaction fees at the cost floor—on the order of 10^{-20} OMC per simple transfer—are maintained not through temporary subsidy but through three reinforcing mechanisms: OMC block rewards to validators, utilization-responsive fee burning that contracts supply under load, and revenue cross-subsidization from the Omne Orchestration Network (30% of compute revenue) and the Omne Media Protocol (storage fee settlement). The combination of two independent, utilization-driven revenue streams reduces the risk that any single demand source failing would compromise fee sustainability.

Observation 4: Protocol-integrated compute and storage eliminate bridge risk. The Omne Orchestration Network and the Omne Media Protocol demonstrate that off-chain compute and decentralized storage can share a single staking, reputation, and settlement infrastructure with the consensus layer. This eliminates the token conversion friction, bridge trust assumptions, and incentive fragmentation that arise when these services operate as separate networks. Smart contracts execute deterministically across heterogeneous hardware, files are stored with cryptographic proof of availability, and off-chain compute workloads settle on-chain with verifiable attestation—all within a unified economic model.

These findings support the hypothesis that a commerce-grade layer-one protocol is achievable without sacrificing the security properties that make blockchains trustworthy. The design space explored here—dual-layer consensus, dual-token economics, non-fee anti-spam, integrated compute, and integrated storage—represents one feasible point in that space. We invite independent analysis, formal verification of the economic properties described, and adversarial testing of the security model.

References

[1] A. Kiayias, A. Russell, B. David, R. Oliynykov, "Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol," Crypto 2017.

[2] E. Buchman, J. Kwon, Z. Milosevic, "The Latest Gossip on BFT Consensus," 2018.

[3] D. J. Bernstein, N. Duif, T. Lange, P. Schwabe, B.-Y. Yang, "High-Speed High-Security Signatures," Journal of Cryptographic Engineering, 2012.

[4] A. Haas et al., "Bringing the Web up to Speed with WebAssembly," PLDI 2017.

[5] R. C. Merkle, "Protocols for Public Key Cryptosystems," IEEE Symposium on Security and Privacy, 1980.

[10] NIST, "Secure Hash Standard (SHS)," FIPS PUB 180-4, 2015.