PATENT APPLICATION

INVENTOR(S)

0001 KIMZEY, SAMUEL C

TITLE

0002 BlockMesh Memory-Mesh Architecture, Stateless Protocol, Epoch Coordination, and Deterministic On-Chain Virtual Machine Execution

CROSS-REFERENCE TO RELATED APPLICATIONS

0003 This application claims the benefit of priority under 35 U.S.C. § 119(e) to the following provisional patent applications:

a. U.S. Provisional Application No. 63/810,843, filed 05/23/2025, titled “BlockMesh Decentralized Memory-Mesh Architecture”
b. U.S. Provisional Application No. 63/810,834, filed 05/23/2025, titled “Memory Field Pressure & Tension System”
c. U.S. Provisional Application No. 63/810,841, filed 05/23/2025, titled “Non-Generative Artificial Cognition Engine”
d. U.S. Provisional Application No. 63/810,837, filed 05/23/2025, titled “Proof of Memory Flow System”
e. U.S. Provisional Application No. 63/811,761, filed 05/23/2025, titled “Composite Cognition Pools for Emergent Memory-Driven Systems”
f. U.S. Provisional Application No. 63/811,758, filed 05/25/2025, titled “Genesis Document Anchor Method for AI Behavior”
g. U.S. Provisional Application No. 63/811,763, filed 05/25/2025, titled “On-Chain VM for Deterministic Language Model Execution”
h. U.S. Provisional Application No. 63/812,634, filed 05/27/2025, titled “Stateless Action Engine & Minimal Proof Logging”
i. U.S. Provisional Application No. 63/812,620, filed 05/27/2025, titled “One Action-One Mint Transaction Paradigm”

j. U.S. Provisional Application No. 63/812,645, filed 05/27/2025, titled “Epoch Rhythm & Synchronization Mechanism”
k. U.S. Provisional Application No. 63/814,230, filed 05/29/2025, titled “Agent Lifecycle Dormancy Management”
l. U.S. Provisional Application No. 63/814,244, filed 05/29/2025, titled “Selective Memory Disclosure via Zero-Knowledge Proofs”
m. U.S. Provisional Application No. 63/814,257, filed 05/29/2025, titled “Identity Bound Memory Rights”
n. U.S. Provisional Application No. 63/816,926, filed 06/03/2025, titled “Memory Driven Autonomy for Physical Agents”
o. U.S. Provisional Application No. 63/816,924, filed 06/03/2025, titled “Post-Generative Decision Support Mesh”
p. U.S. Provisional Application No. 63/816,907, filed 06/03/2025, titled “Surplus Flow Redistribution System and Method”

All of the above applications are incorporated herein by reference in part or in their entirety.

TECHNICAL FIELD

0004 This invention relates to distributed ledger technology, decentralized memory systems, event-driven architectures, network topology management, and on-chain computational infrastructure. It encompasses methods for creating multi-layered blockchain meshes, atomic event attestation, stateless protocol execution, memory-based coordination, distributed storage orchestration, peer discovery mechanisms, protocol versioning, and deterministic virtual machine operation. The invention applies to blockchain networks, distributed databases, decentralized applications, federated systems, edge computing networks, IoT meshes, and any distributed system requiring scalable, auditable, and deterministic operation.

BACKGROUND

0005 Conventional blockchain architectures are limited by single-chain bottlenecks, stateful smart contract execution, and high storage or gas costs. They require external scheduling, off-chain compute, centralized storage, or static peer discovery, resulting in fractured state, complex audit trails, and weak data provenance. Existing systems lack integrated solutions for multi-chain coordination, atomic event capture, stateless execution, distributed storage with self-healing, dynamic mesh formation, versioned governance, and autonomous on-chain computation.

0006 No existing protocol provides a unified framework for:

a. Multi-layered blockchain mesh with independent yet anchored chains
b. Atomic event recording ensuring single authoritative records
c. Stateless engines reconstructing state from event history
d. Distributed storage with sharding, encryption, and self-healing
e. Dynamic peer discovery and mesh topology management
f. Protocol evolution through versioned on-chain governance
g. Deterministic on-chain VM execution triggered by memory events
h. Epoch coordination based on memory pressure patterns
i. Cross-layer coordination with cryptographic boundaries

DETAILED DESCRIPTION

0007 This invention provides a comprehensive framework for blockchain mesh architectures with memory-driven coordination and deterministic computation.

0008 <u>BlockMesh Memory-Mesh Architecture:</u>

0009 Systems implement hierarchical blockchain topologies with multiple independent chains forming a mesh. Each chain operates autonomously while maintaining cryptographic anchoring. The architecture supports:

a. 2 to N layers (L1-L6 or more) with designated purposes
b. Independent consensus per chain with cross-chain anchoring
c. Memory-based coordination between chains

d. Privacy boundaries enforced cryptographically
e. Subchains for identity, computation, storage, or governance
f. Flexible topology supporting various blockchain types (PoW, PoS, PoA, PoM)

0010 <u>One-Action-One-Mint Paradigm:</u>

0011 Every system action generates exactly one immutable on-chain mint transaction. This ensures complete auditability, eliminates auxiliary logs, enables stateless reconstruction, provides atomic proof, and supports downstream subscription. The paradigm works across any blockchain implementation.

0012 <u>Stateless Protocol Engine:</u>

0013 System state is reconstructed by replaying mint sequences rather than storing mutable state. The engine supports:

a. Deterministic replay from genesis or checkpoints
b. Parallel reconstruction across nodes
c. Point-in-time state queries
d. Verification without full history
e. Multiple blockchain compatibility

0014 <u>Memory-Based Epoch Coordination:</u>

0015 System periods emerge from memory accumulation patterns rather than fixed time:

a. Memory pressure triggers epoch transitions
b. Event density determines epoch boundaries
c. Pattern detection enables coordination
d. Reconciliation occurs at epoch boundaries
e. Adaptive timing based on activity

0016 <u>Distributed Storage Layer:</u>

0017 Data is sharded and distributed across mesh nodes:

a. Content-addressed storage with CID
b. Encryption before distribution
c. Configurable replication (1-N copies)
d. Automatic rebalancing and healing
e. Support for distributed or blockchain-based file systems
f. Capacity management across heterogeneous nodes

0018 <u>Dynamic Mesh Formation:</u>

0019 Nodes discover and organize through multiple mechanisms:

a. DNS-based bootstrap discovery
b. Blockchain-anchored peer lists
c. Gossip protocol propagation
d. DHT-based peer finding
e. Direct peer introduction
f. Reputation-based selection

0020 <u>On-Chain Virtual Machine Execution:</u>

0021 Deterministic computation triggered by blockchain events:

a. WASM, EVM, or custom VM support
b. Memory-pressure triggered execution
c. Cryptographically attested code loading
d. Deterministic execution across nodes
e. No external oracles required
f. Gas-free or alternative fee models

0022 <u>Protocol Versioning and Governance:</u>

0023 System rules anchored on-chain at genesis:

a. Immutable core documents with hashes
b. Machine-readable protocol specifications
c. Parameter adjustment through governance
d. Fork management and resolution
e. Upgrade coordination across mesh

ENABLEMENT

0024 The inventions described herein are enabled at the system and protocol level. A person of ordinary skill in distributed systems, decentralized computation, multi-agent coordination, and privacy-preserving cryptography can practice the claimed methods using the architectural descriptions in these specifications together with any suitable software/hardware stack.

0025 A working, non-limiting embodiment is publicly available at:

https://github.com/zeam-labs/zeam-testnet

0026 This implementation comprises a production-grade, modular codebase demonstrating the claimed protocol flows across initialization/governance, event logging, stateless replay, pressure/memory dynamics, storage orchestration, identity/privacy, deterministic VM execution, trait-based observation, and economic coordination.

0027 Function, file, and module names below are illustrative. Equivalent structures, languages (e.g., Rust, C++, Python, Solidity), runtimes (WASM/EVM/custom), and storage/consensus substrates may be substituted without departing from the inventions. The cited repository is only one embodiment; the claims cover all protocol-equivalent realizations.

0028 <u>System-Level Enablement</u>

0029 Architecture & Flows. The reference implementation demonstrates:

a. Initialization/Governance: anchoring of immutable documents (core/traits/protocol) with version identifiers and verification at runtime

b. Atomic Event Logging ("one-action-one-mint"): each operation emits one immutable record with input/output attestation and metadata
c. Stateless Replay: deterministic state reconstruction by replaying minted events from genesis or checkpoints; parallel verification supported
d. Pressure/Memory Dynamics: conversion of events/sensor data to memory-flow; detection of pressure/tension patterns; reflex orchestration and epoch coordination
e. Deterministic Compute: pressure-triggered, on-chain VM (e.g., WASM) loading attested code/model shards; deterministic transformations; mint-logged proofs
f. Distributed Storage: content addressing (CID), sharding, encryption, replication, capacity management, periodic verification, and self-healing
g. Identity/Privacy: non-transferable anchors, rights (access/control/preservation/return), layer-specific privacy, ZK selective disclosure, recovery across forks
h. Traits/Observation: domain observers (audit/ethics/health/finance/etc.) that monitor and recommend without direct control; all activity is mint-logged
i. Economic Coordination: surplus computation, pooling, middle-out redistribution, non-token flows, privacy proofs, and anti-manipulation measures
j. Peer Discovery & Topology: decentralized bootstrap (e.g., DNS/TXT, gossip), dynamic peer pools, and mesh self-healing
k. Protocol Versioning: genesis-anchored document hashes, on-chain version history, and upgrade/rollback mechanics
l. Agent Lifecycle: activation/dormancy/reactivation tied to memory-flow/pressure; lifecycle transitions are mint-logged
m. Sensor Integration: physical or external observations converted to memory-flow with equal footing to digital events
n. Health Visualization: real-time mesh/pressure visualizations, magnetic field feedback, and audit-logged metrics
o. Consent & Delegation: cryptographic consent flags, programmable delegation, return/recovery rights alongside selective disclosure

0030 Reproducibility. Determinism is enforced by:

a. Attested inputs/outputs per operation; no hidden state

b. Prohibition of non-deterministic sources (unseeded RNG, external oracles) in core paths
c. Checkpointing and replay yielding identical state for identical event sequences

0031 Portability. The methods are implementation-agnostic:

a. Any consensus (PoW/PoS/PoA/PoM/hybrid), storage substrate (IPFS-like, ZFS-like, cloud, on-prem), VM (WASM/EVM/custom), or language may be used
b. Modules labeled here (e.g., "storage," "runtime," "cognition," "vault") represent roles; organization and naming may vary

0032 <u>Practicing the Inventions Without the Reference Code</u>

0033 A skilled practitioner can implement the inventions by:

1. Anchoring immutable governance and protocol documents at system initialization and enforcing verification before execution
2. Emitting one atomic record per operation (input/output attested) and replaying those records for deterministic state
3. Computing memory-flow/pressure from events/sensors and triggering reflexes/epochs via pattern detection
4. Orchestrating storage via content addressing, sharding, encryption, replication, verification, and self-healing
5. Executing deterministic compute via any VM (WASM/EVM/custom) that loads attested modules/shards and produces mint-logged proofs
6. Anchoring identity & privacy with non-transferable credentials, on-chain rights, layered privacy, and ZK selective disclosure
7. Coordinating economics with surplus calculation, pooling, middle-out weighting, privacy proofs, and audit-ready distribution—all protocol-enforced

0034 These steps can be realized in any mature stack (e.g., Rust + Substrate/IPFS, Python + Tendermint, Solidity + rollups, C++ + custom VM), using equivalent cryptographic, storage, and coordination primitives.

0035 Scope, Extensibility, and Best Mode

a. The reference code shows one best-mode embodiment at filing. The claims are not limited to that code, file structure, storage system, consensus, VM, or language
b. Modules that serve as structural templates illustrate how protocol concepts are realized and may be substituted or extended without departing from the inventions
c. All major protocol flows (anchoring, minting, replay, pressure, privacy, VM, storage, economics, traits) are disclosed and demonstrated in a working system, enabling immediate practice and independent re-implementation

0036 The inventions are fully enabled through the protocol methods and flows set forth in these specifications and through a public, working embodiment. A skilled person can implement the inventions using the above guidance—either by adapting the reference code or by building equivalent systems on alternative platforms.

CLAIMS

1. A method for creating hierarchical blockchain meshes with immutable governance, comprising:
   a. establishing multiple independent chains organized hierarchically;
   b. defining cryptographic anchoring relationships between chains;
   c. embedding governance documents as cryptographic commitments at system initialization;
   d. recording all operations as atomic one-action-one-mint events;
   e. reconstructing system state through stateless replay of mint sequences;
   f. enforcing privacy and trust boundaries across mesh layers;
   g. enabling heterogeneous consensus mechanisms per chain;
   h. maintaining governance continuity through versioned on-chain registries; and
   i. ensuring all cross-chain interactions are cryptographically verifiable and auditable.

2. A method for synchronizing distributed chains through memory-driven pressure dynamics, comprising:

a. accumulating memory events across chains into continuous pressure signals;
b. detecting temporal and spatial patterns including clustering, frequency, divergence, or silence;
c. defining epoch boundaries based on pressure accumulation and release patterns;
d. coordinating reconciliation and synchronization at epoch transitions;
e. generating reflexive system responses proportional to detected memory pressure;
f. triggering composite reflexes through constructive or destructive interference of multiple pressures;
g. adjusting epoch duration dynamically based on system activity levels;
h. recording all epoch boundaries and reflex events as immutable on-chain entries; and
i. enabling system rhythm without reliance on fixed time intervals or external oracles.

3. A method for executing deterministic computation and distributed storage in a blockchain mesh, comprising:
   a. triggering virtual machine execution from blockchain events;
   b. loading code or model fragments from cryptographically attested distributed storage;
   c. executing deterministic transformations in WASM, EVM, stack-based, or custom environments;
   d. prohibiting non-deterministic sources including unseeded randomness, timestamps, or external oracles;
   e. generating cryptographic proofs of execution for each transformation;
   f. sharding data across mesh nodes with encryption and replication;
   g. rebalancing and healing storage automatically through protocol-defined assignments;
   h. enabling concurrent model versions and partial updates through versioned registries; and
   i. recording all compute and storage operations as immutable mint events.

4. The method of claim 1, wherein governance documents include ethics charters, operational policies, regulatory requirements, safety protocols, or value statements, immutably anchored at genesis.

5. The method of claim 1, wherein one-action-one-mint ensures each system action produces exactly one atomic record containing full input and output attestation.

6. The method of claim 1, wherein stateless replay supports deterministic reconstruction from genesis or from checkpoint states.

7. The method of claim 1, wherein heterogeneous consensus includes work-based, stake-based, authority-based, memory-based, or hybrid mechanisms.

8. The method of claim 1, wherein cross-chain interactions use anchored proofs for selective data sharing, bridge operations, or rollback management.

9. The method of claim 2, wherein memory pressure signals are computed using event density, flow velocity, value magnitude, or protocol-defined weighting.

10. The method of claim 2, wherein silence, contradiction, or drift in memory flow generates reflexive alerts and epoch realignment.

11. The method of claim 2, wherein reflexes are triggered by interference patterns including resonance, phase alignment, or harmonic reinforcement.

12. The method of claim 2, wherein epoch boundaries create checkpoints for reconciliation, recovery, and stateless replay.

13. The method of claim 2, wherein composite reflex orchestration includes weighted fusion of digital and physical memory flows.

14. The method of claim 3, wherein virtual machines include bytecode interpreters, stack machines, or symbolic reasoning engines, each producing deterministic outputs.

15. The method of claim 3, wherein model fragments use binary formats, neural network exchange formats, tensor serialization formats, or protocol-specific representations with cryptographic attestation.

16. The method of claim 3, wherein deterministic execution is reproducible across nodes such that identical inputs yield identical outputs.

17. The method of claim 3, wherein distributed storage integrates with IPFS-like or ZFS-like systems using cryptographic content addressing.

18. The method of claim 3, wherein self-healing storage includes re-pinning, replication factor enforcement, and load balancing across heterogeneous nodes.

19. The method of claim 3, wherein execution proofs include cryptographic signatures, Merkle trees, zero-knowledge attestations, or equivalent mechanisms.

20. The method of claim 3, wherein compute and storage events are surfaced as audit-ready logs for real-time visualization and governance verification.

ABSTRACT

0037 A comprehensive framework for blockchain mesh architectures featuring hierarchical multi-chain topologies, one-action-one-mint atomic recording, stateless protocol execution, memory-based epoch coordination, distributed storage orchestration, dynamic peer discovery, on-chain virtual machine execution, and versioned governance. The invention enables scalable blockchain systems that maintain complete auditability, support heterogeneous consensus mechanisms, coordinate through memory pressure patterns, execute computation deterministically on-chain, and evolve through governed updates. Compatible with existing blockchain platforms while enabling novel architectures.