PATENT APPLICATION

INVENTOR(S)

0001 KIMZEY, SAMUEL C

TITLE

0002 Memory-Flow, Field Pressure, and Reflex Event Orchestration in Distributed Ledgers

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 decentralized system coordination, field-effect memory dynamics, pressure and tension modeling, pattern-based orchestration, distributed health monitoring, feedback management, reflex automation, sensor integration, autonomous control, and analytical decision support in distributed systems. The invention encompasses methods for generating memory-flow signals from digital and physical sources, detecting system state via pressure patterns, managing feedback dynamics, coordinating without algorithms, managing agent lifecycles, implementing mesh health feedback, triggering reflexes, enabling autonomous navigation through memory, and providing analytical insights without prediction.

BACKGROUND

0005 Existing distributed systems rely on tokens, algorithmic polling, schedules, predictive models, or external triggers for coordination, ignoring organic informational dynamics in event logs and sensor streams. They lack unified mechanisms for generating continuous memory-flow from events and observations, detecting field pressure, managing feedback loops, implementing self-healing topologies, orchestrating reflexes based on patterns, enabling navigation through accumulated experience, and providing insights without forecasting. Systems cannot learn from patterns, adapt to conditions, maintain stability through self-modulation, or make decisions based solely on history rather than prediction.

0006 No system provides memory-driven, pressure-based coordination enabling:

a. Generation and propagation of memory-flow signals from any source
b. Detection and visualization of field pressure patterns with feedback
c. Integration of physical sensor data as memory contributors
d. Autonomous navigation through accumulated route memory
e. Analytical decision support without predictive modeling
f. Adaptive peer management through field dynamics
g. Pattern learning and adaptation over time
h. Post-generative decision making without prediction
i. Autonomous reflex orchestration from emergent triggers
j. Self-stabilizing pressure equilibrium
k. Full auditability and deterministic replay
l. Coordination without tokens, oracles, or central control

DETAILED DESCRIPTION

0007 This invention provides a comprehensive framework for memory-driven coordination through field effects and emergent patterns across digital and physical domains.

0008 <u>Memory-Flow and Field Pressure Generation:</u>

0009 All protocol events, sensor observations, and data inputs contribute atomic impulses to continuous memory-flow signals. Flow propagates between nodes through configurable mechanisms. Physical sensors generate memory-flow equally with digital events. Autonomous systems accumulate route memory as flow patterns. Analytical platforms convert data streams into memory pressure. Flow is computed using metrics including density, frequency, spatial distribution, temporal clustering, and event significance. Multiple streams aggregate into multi-dimensional pressure fields where patterns emerge organically.

0010 <u>Physical Sensor Memory Integration:</u>

0011 Sensor observations from autonomous systems are converted directly into memory-flow signals. Each reading becomes a memory atom contributing to system pressure. Navigation decisions emerge from accumulated route memory rather than path planning. Safety responses trigger from memory pressure thresholds. Sensor memories are weighted by criticality and recency. Systems share memory patterns for collective learning.

0012 <u>Analytical Decision Support Without Prediction:</u>

0013 Analytics platforms surface patterns from accumulated memory without generating predictions. The system identifies tensions, contradictions, and alignments in historical data converted to memory-flow. Decisions are supported by pressure patterns from actual events. Uncertainty is represented as absence of memory pressure. Analysts can trace insights to specific memory-flow sources. The system refuses to extrapolate beyond accumulated memory.

0014 <u>Pressure Feedback Dynamics:</u>

0015 Pressure events generate secondary pressure through feedback mechanisms including self-reinforcing loops with controlled amplification, dampening functions preventing runaway conditions, equilibrium detection identifying stable states, automatic modulation maintaining stability, cascade prevention through refractory periods, and pressure memory enabling pattern recognition. Physical events create stronger feedback than digital events when safety is involved.

0016 Autonomous Navigation Through Memory:

0017 Vehicles and robots navigate by following memory-flow gradients from previous successful routes. Challenging locations accumulate higher memory pressure, triggering careful behavior. Smooth routes show low pressure, enabling confident navigation. Fleet memory-flow enables learning from collective experiences. New routes generate exploratory pressure patterns. Emergency maneuvers create immediate high-pressure memories shared across fleets.

0018 Adaptive Mesh Topology:

0019 Nodes participate in field dynamics where health metrics define attraction/repulsion strengths. Healthy nodes attract connections within capacity. Unhealthy nodes are gradually isolated. Partitioned nodes receive recovery assistance. Load balancing occurs through field redistribution. Topology optimization via field gradients. Physical agents use field dynamics for formation and swarm coordination.

0020 Pattern Recognition and Adaptation:

0021 The system recognizes and learns from pressure patterns whether from digital events, physical sensors, or analytical data. Pattern templates are extracted from historical data. Similarity matching occurs with configurable tolerance. Priority weighting based on pattern significance. False positive suppression through validation. Pattern evolution tracking over time. Anomaly detection for unknown patterns. Cross-domain pattern recognition between physical and digital sources.

0022 Post-Generative Decision Support:

0023 Decisions emerge from accumulated memory without prediction. Memory tension indicates unresolved states. Contradiction detection surfaces conflicts. Silence patterns reveal missing information. Protocol drift identifies misalignment. Decisions based solely on historical patterns. No probabilistic inference or speculation. Physical actions emerge from memory pressure. Analytical insights derive from pattern recognition.

0024 Agent Lifecycle Management:

0025 Agent states respond to pressure dynamics. Activation thresholds adapt to system load. Dormancy preserves resources during calm. Strain detection prevents overload. Graceful degradation under pressure. Hierarchical activation cascades. Context preservation across transitions. Autonomous systems enter dormancy during low-activity periods. Analytical agents activate when data pressure accumulates.

0026 Epoch Synchronization:

0027 System rhythm emerges from pressure patterns. Primary epochs from major pressure cycles. Micro-epochs from local fluctuations. Nested hierarchies with different sensitivities. Checkpoint generation at stability points. Reconciliation during pressure valleys. Variable duration based on activity. Physical systems may have faster micro-epochs than digital layers.

0028 Composite Reflexes:

0029 Multiple pressure sources combine through constructive and destructive interference. Harmonic resonance creating standing waves. Phase relationships determining timing. Weighted fusion based on source authority. Threshold aggregation for activation. Physical sensor pressure overrides digital pressure for safety. Analytical pressure guides long-term planning while sensor pressure handles immediate response.

0030 Visualization and Monitoring:

0031 Real-time system state representation including pressure heat maps, flow vectors, node health indicators, pattern recognition overlays, anomaly highlighting, and privacy-preserving aggregation. Operators see memory-flow visualizations for situational awareness. Analysts view pressure patterns across data domains. Planners observe memory accumulation.

ENABLEMENT

0032 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.

0033 A working, non-limiting embodiment is publicly available at:

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

0034 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.

0035 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.

0036 <u>System-Level Enablement</u>

0037 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

0038 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

0039 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

0040 Practicing the Inventions Without the Reference Code

0041 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

0042 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.

0043 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

0044 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 generating and managing memory-flow pressure in decentralized systems, comprising:
   a. monitoring cryptographically attested events and sensor observations across distributed nodes;
   b. converting digital and physical inputs into continuous memory-flow signals;
   c. accumulating flows into pressure fields using metrics including density, frequency, clustering, magnitude, or significance;
   d. propagating flow between nodes with distance-based decay or governance-defined attenuation functions;
   e. amplifying flow through resonance with similar patterns and dampening runaway conditions;
   f. aggregating flows across domains into multi-dimensional pressure fields;
   g. detecting equilibrium, divergence, or anomalies in accumulated pressure;
   h. recording all flow generation and propagation events immutably on-chain; and
   i. enabling system coordination without reliance on tokens, external triggers, probabilistic models, or predictive forecasting.

2. A method for reflex orchestration and epoch synchronization in decentralized systems, comprising:
   a. defining epochs through accumulation and release of memory pressure;
   b. detecting micro-epochs, nested epochs, or reconciliation points from local fluctuations;
   c. generating reflexive system responses proportional to pressure accumulation;
   d. triggering composite reflexes through constructive or destructive interference of multiple pressures;
   e. ensuring interference patterns are mathematically deterministic and reproducible across nodes given identical inputs;
   f. weighting reflexes based on domain authority, safety criticality, or governance rules;
   g. recording epoch boundaries and reflex activations as immutable events;
   h. synchronizing chains across epochs without fixed time intervals;
   i. supporting reconciliation and recovery at pressure valleys or checkpoints; and
   j. ensuring reflex orchestration is deterministic, auditable, and reproducible across all participants.

3. A method for adaptive topology, decision support, and system resilience in memory-driven networks, comprising:
   a. adjusting peer connectivity based on node health, utilization, or pressure contribution;
   b. isolating unhealthy nodes and assisting partitioned nodes with recovery;
   c. detecting and learning from recurring pressure patterns, anomalies, or drift;
   d. surfacing contradictions, silences, or unresolved tensions as decision inputs;
   e. enabling post-generative decision support based solely on accumulated historical memory patterns without generation or prediction;
   f. managing agent lifecycle states including dormant, active, strained, or degraded based on pressure;
   g. maintaining stability through equilibrium detection, feedback modulation, and cascade prevention;
   h. visualizing system pressure, reflexes, and memory patterns for operators; and
   i. recording all topology adjustments, decisions, and lifecycle transitions immutably on-chain.

4. The method of claim 1, wherein sensor data includes visual, distance, environmental, biometric, chemical, or radiation readings converted into memory-flow signals.

5. The method of claim 1, wherein attenuation functions include exponential decay, inverse square law, or custom governance-defined patterns.

6. The method of claim 1, wherein feedback loops include self-reinforcing amplification, dampening functions, and refractory periods to maintain equilibrium.

7. The method of claim 1, wherein anomalies include unknown patterns, drift, or silence indicating missing information.

8. The method of claim 1, wherein equilibrium detection identifies stable attractor states and bifurcation points for stability management.

9. The method of claim 2, wherein micro-epochs provide faster reflex synchronization for physical agents relative to digital layers.

10. The method of claim 2, wherein reflexes are weighted to prioritize physical sensor pressure over digital memory pressure for safety.

11. The method of claim 2, wherein composite reflex orchestration uses harmonic resonance, phase alignment, or threshold aggregation.

12. The method of claim 2, wherein reconciliation checkpoints include cryptographic commitments, Merkle proofs, or state snapshots.

13. The method of claim 2, wherein memory-flow pressure directly triggers the reflex orchestration through pressure accumulation, threshold detection, pressure gradient recognition, or equivalent triggering mechanisms.

14. The method of claim 3, wherein topology adjustment attracts healthy nodes within capacity limits and repels unhealthy nodes.

15. The method of claim 1, wherein the system excludes probabilistic forecasting and relies solely on accumulated historical pressure data.

16. The method of claim 3, wherein pattern recognition includes extracting templates, matching with tolerance, weighting by significance, and tracking evolution over time.

17. The method of claim 3, wherein lifecycle transitions include pressure-driven dormancy during low activity and graceful degradation under strain.

18. The method of claim 3, wherein equilibrium management includes dampening oscillations, preventing cascades, and restoring stability through control functions.

19. The method of claim 3, wherein visualization includes pressure heat maps, flow vectors, anomaly overlays, and privacy-preserving aggregation.

20. The method of claim 3, wherein recorded logs enable deterministic replay, independent verification, and audit of all memory-flow and reflex operations.

ABSTRACT

0045 A framework for memory-flow generation, field pressure modeling, feedback dynamics management, topology coordination, and reflex orchestration in distributed systems integrating digital and physical domains. The invention transforms events and sensor observations into continuous pressure patterns with self-modulating feedback, implements adaptive topology management, enables pattern learning, supports post-generative decision making without prediction, triggers emergent coordination through field effects, enables autonomous navigation through memory patterns, and provides analytical insights without forecasting. All operations maintain stability through equilibrium management, provide real-time visualization, and create

immutable audit trails, enabling resilient coordination without central control or external dependencies.