PATENT APPLICATION

INVENTOR(S)

0001 KIMZEY, SAMUEL C

TITLE

0002 Surplus Flow Redistribution and Civic Economic Coordination in Decentralized Systems

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 their entirety.

TECHNICAL FIELD

0004 This invention relates to decentralized economic flow management, surplus redistribution, vault coordination, economic privacy, memory-anchored value flows, pressure-triggered economic events, and protocol-enforced economic automation in distributed systems. The invention encompasses methods for recording agent income and borrowing, calculating surplus, pooling and redistributing value via non-tokenized flows, triggering economic events through memory pressure, protecting economic privacy while maintaining auditability, and ensuring protocol-compliance without central control or token mechanisms.

BACKGROUND

0005 Conventional redistribution systems in distributed networks rely on flat rewards, token incentives, staking, or administrative control, producing perverse incentives, centralization risks,

or privacy violations. Token-based systems create speculation and manipulation opportunities. Staking models favor early adopters. Existing protocols fail to protect privacy while enabling fair redistribution, cannot prevent manipulation without central oversight, and require external triggers for economic events.

0006 No existing system provides protocol-anchored, memory-driven surplus redistribution that:

a. Anchors all economic events in immutable vault memory
b. Classifies surplus and borrowing through cryptographic attestation
c. Pools value in a non-tokenized manner
d. Distributes using weighted functions favoring median performers
e. Triggers economic events through memory-pressure patterns
f. Maintains full auditability with economic privacy
g. Operates without token emissions or administrative override
h. Integrates with memory-flow pressure for autonomous coordination

DETAILED DESCRIPTION

0007 This invention provides a protocol-enforced, privacy-protected, pressure-triggered, and auditable surplus redistribution system for decentralized economic coordination.

0008 <u>Surplus Classification and Vault Anchoring:</u>

0009 Each agent maintains a protocol-enforced vault recording all economic operations. Income streams from completed actions, contributions, or external integrations are recorded as cryptographically attested events. Borrowing represents temporary value allocation for operations. Surplus is computed per epoch as Income minus Borrowed Amount. All vault operations follow atomic paradigms where each economic event produces exactly one immutable record. Vaults maintain running balances, transaction histories, and cryptographic proofs.

0010 Memory-Pressure Triggered Economic Events:

0011 Economic operations integrate with memory-flow pressure signals. When pressure metrics indicate epoch boundaries, surplus calculations trigger automatically. High pressure in economic layers can accelerate redistribution cycles. Low pressure periods enable accumulation phases. Economic events generate pressure signals, creating feedback loops that stabilize system economics. Pressure-triggered events include epoch determination, surplus calculation, pool formation, borrowing adjustments, and policy updates.

0012 Pooling and Middle-Out Redistribution:

0013 Each epoch, a protocol-defined fraction of eligible agent surplus contributes to a redistribution pool. The system computes the statistical median of participating agent incomes. A non-linear weighting function assigns redistribution influence with full reward near median, smooth transitions, and decay at extremes. Redistribution is capped at the minimum of contributed surplus, borrowing capacity, and protocol maximum. Distribution calculations mint as atomic events for traceability.

0014 Non-Tokenized Flow and Economic Privacy:

0015 Redistribution operates using protocol-internal units rather than external tokens, representing computational resources, storage capacity, network bandwidth, or participation credits. Vault balances, borrowing amounts, and transaction patterns are protected using range proofs for balance verification, homomorphic encryption for private computation, differential privacy for statistics, and ring signatures for anonymity. Agents prove economic properties without revealing amounts.

0016 Integration with Memory Layers:

0017 Economic operations span multiple memory layers with public economic events, private vault state, execution logic, economic narratives, and long-term patterns distributed across

layers. Cross-layer economic flows require consent and generate pressure signals that influence system-wide economic rhythm.

0018 Adaptive Economic Policy:

0019 Redistribution parameters adapt through protocol governance including epoch length, pooling fraction, median zone, decay coefficient, and borrowing limits. Policy changes require multi-signature approval, grace periods, parallel testing, rollback capability, and full audit trails.

0020 Trait-Based Economic Observation:

0021 Economic observation modules monitor patterns including wealth concentration, flow velocity, borrowing stress, hoarding detection, and extraction patterns. Observations surface without overriding protocol logic, enabling early warning of imbalances, policy recommendations, manipulation detection, and struggling agent identification.

0022 Anti-Manipulation Mechanisms:

0023 The system prevents gaming through identity anchoring for resistance, temporal smoothing of spikes, reputation weighting, anomaly detection, mandatory participation windows, and progressive accumulation caps.

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 System-Level Enablement

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 surplus classification and vault anchoring in decentralized systems, comprising:
   a. maintaining protocol-enforced vaults for each agent to record income, borrowing, and surplus;
   b. recording all vault operations as cryptographically attested atomic events;
   c. computing surplus as income minus borrowed amount within each epoch;
   d. anchoring running balances and transaction histories immutably on-chain;
   e. supporting vault recovery through identity-based return rights;
   f. enabling aggregation of multiple income sources into unified surplus;
   g. recording surplus classifications as immutable entries;
   h. enforcing cryptographic proofs of balance and transaction integrity; and
   i. preserving deterministic replay of vault operations across nodes for audit, recovery, and independent verification.

2. A method for memory-pressure triggered pooling and redistribution of surplus value, comprising:
   a. accumulating memory-flow pressure from economic events and transactions to determine epoch boundaries;
   b. contributing a protocol-defined fraction of surplus to a redistribution pool;
   c. calculating statistical medians of participating agent surpluses;
   d. applying non-linear weighting functions that maximize redistribution near the median and decay at extremes;
   e. capping redistribution at the minimum of contributed surplus, borrowing capacity, and protocol-defined maximum;
   f. generating reflexive redistribution events when pressure accumulation, thresholds, or equivalent triggering conditions are met;
   g. redistributing pooled surplus proportionally based on weighted functions;
   h. recording pooling, weighting, and redistribution events immutably on-chain; and
   i. enabling dynamic adjustment of redistribution frequency based on pressure accumulation.

3. A method for privacy-preserving and auditable surplus coordination in decentralized systems, comprising:

a. obscuring vault balances using range proofs, homomorphic encryption, or differential privacy;
b. enabling transaction anonymity through ring signatures, mixing, or stealth addressing;
c. preventing manipulation through identity anchoring and anomaly detection;
d. deploying observation modules to monitor wealth concentration and hoarding patterns;
e. supporting programmable consent for redistribution participation;
f. enabling adaptive policy updates through governance mechanisms;
g. maintaining audit access with privacy-preserving aggregation; and
h. recording all economic operations for deterministic replay enabling independent verification, audit, and regulatory compliance.

4. The method of claim 1, wherein vaults enforce surplus computation through atomic one-action-one-mint records.

5. The method of claim 1, wherein recovery of vault balances requires cryptographic proof of identity continuity.

6. The method of claim 1, wherein surplus aggregation supports multiple income sources including digital, physical, or external integrations.

7. The method of claim 1, wherein deterministic replay reconstructs vault state from genesis or from checkpointed states.

8. The method of claim 2, wherein pooling fractions, weighting functions, and redistribution caps are defined in protocol-governed parameters.

9. The method of claim 2, wherein non-linear weighting curves include sigmoid, logarithmic, or polynomial functions.

10. The method of claim 2, wherein redistribution frequency is accelerated under high economic pressure and slowed during low pressure.

11. The method of claim 2, wherein reflexive redistribution events generate secondary pressure signals for system-wide coordination.

12. The method of claim 2, wherein surplus pooling and redistribution are capped to prevent over-distribution beyond contributed surplus.

13. The method of claim 3, wherein range proofs verify balances without exposing exact amounts.

14. The method of claim 3, wherein transaction anonymity uses ring signatures, stealth addressing, or equivalent techniques to obscure linkages.

15. The method of claim 3, wherein anomaly detection includes temporal smoothing, extraction pattern recognition, surplus spike identification, or equivalent mechanisms.

16. The method of claim 3, wherein programmable consent supports time-limited, conditional, or delegated participation authority.

17. The method of claim 3, wherein governance updates require multi-signature approval, grace periods, and rollback capability.

18. The method of claim 3, wherein observation modules monitor borrowing stress, flow velocity, and economic imbalances without enforcing corrections.

19. The method of claim 3, wherein audit access provides role-based permissions with aggregated views preserving individual privacy.

20. The method of claim 3, wherein deterministic replay enables independent verification of all surplus operations across epochs.

ABSTRACT

0037 A protocol-enforced surplus redistribution and economic coordination system for decentralized networks integrated with memory-flow pressure dynamics. The invention records agent income, borrowing, and surplus as cryptographically attested atomic events, automatically triggers economic events through memory-pressure patterns, pools surplus for redistribution using non-linear weighting curves favoring median performers, and distributes flow based on protocol-governed privacy-protected logic without tokens. Vaults and borrowing are obscured using cryptographic proofs while maintaining full auditability. Economic operations span multiple memory layers with consent-based cross-layer flows. Redistribution logic is protocol-configurable and pressure-adaptive, with observation modules, anti-manipulation mechanisms, and regulatory compliance support. The system achieves fair economic coordination without central control, external triggers, or token-based incentives.