

PATENT APPLICATION

INVENTOR(S)

[0001] KIMZEY, Samuel C

TITLE

[0002] One-Action-One-Mint Transaction Paradigm

TECHNICAL FIELD

[0003] This invention relates to blockchain data recording, distributed ledger execution efficiency, and automated agent coordination. More particularly, it describes a method for atomic event capture and propagation in decentralized, memory-driven systems where each action results in exactly one immutable mint.

BACKGROUND

[0004] Conventional decentralized systems often rely on redundant state logs, off-chain event queues, and fragmented proof mechanisms to track system activity. These lead to storage bloat, complex synchronization logic, and fragmented data for downstream consumers. In dynamic systems with autonomous agents or AI modules, this complexity impedes real-time coordination. There is a need for a unified, atomic event model that enables stateless replay, consistent proofs, and seamless downstream consumption.

SUMMARY

[0005] The One-Action-One-Mint paradigm includes:

1. Defining each action type with explicit input/output schemas and unique identifiers.

PATENT APPLICATION

2. Generating a cryptographic attestation binding inputs to outputs.
3. Minting a single on-chain transaction encapsulating the attestation and metadata.
4. Using the minted transaction as the sole source of truth for all downstream consumers.
5. Enabling full system state reconstruction by replaying these atomic mints through a stateless protocol engine.

DETAILED DESCRIPTION

[0006] Every system event – whether a memory entry, proof-of-memory, vault transaction, or cognitive reflection – is formalized as an action with a unique identifier and defined schemas for inputs and outputs. A stateless engine refers to a module or processor that does not store internal state between executions but derives behavior solely from replayed events and immutable inputs.

Upon action execution:

1. A cryptographic attestation (e.g., hash or signature) is computed over the combined input/output data.
2. A single mint transaction is produced on-chain embedding the attestation, action identifier, and any relevant metadata (e.g., timestamp, actor ID, context).

This atomic mint serves as:

1. The sole authoritative record of the action.
2. A trigger for downstream modules—agents, smart contracts, AI engines, or analytic pipelines—that subscribe to these mints to drive further logic.
3. A replayable proof unit for reconstructing state without side logs or mutable databases.

PATENT APPLICATION

METHOD FLOW

1. Action Definition – Each action is defined by its input/output schema and a unique identifier.
2. Attestation Generation – On execution, compute a cryptographic attestation (e.g., hash or signature) binding inputs to outputs.
3. Atomic Mint – A single on-chain transaction is minted containing the attestation, metadata, and identifiers.
4. Downstream Integration – Agents and systems subscribe to mint events for real-time or deferred processing.
5. State Reconstruction – Replay the mint sequence through a stateless engine to derive system state at any point.

NARRATIVE WORKED EXAMPLE

[0007] A vault credit issuance is processed. The action schema includes:

type: MintCredit

inputs: {userID, amount, timestamp}

outputs: {newBalance}

[0008] The system hashes these values to generate an attestation, then mints a single transaction embedding this data. Modules such as risk analyzers, dashboards, or reflection engines subscribe to MintCredit events and update their logic using this immutable source—no additional logs or queries are needed.

PATENT APPLICATION

ALGORITHMIC WORKED EXAMPLE

[0009] Pseudocode:

```
def process_action(action_type, input_data):  
  
    output_data = execute_logic(action_type, input_data)  
  
    attestation = hash(input_data || output_data || action_type)  
  
    mint_transaction('ActionMint', {  
  
        'type': action_type,  
  
        'attestation': attestation,  
  
        'inputs': input_data_metadata,  
  
        'outputs': output_data_metadata  
  
    })  
  
    notify_subscribers('ActionMint')
```

POTENTIAL EMBODIMENTS

[00010] Use of digital or multi-signature attestations for federated workflows.

[00011] Aggregation of multiple one-action-one-mint logs into a composite batch transaction for efficiency.

[00012] Embedding Merkle roots or trees to support sub-action nesting or structured workflows.

PATENT APPLICATION

[00013] Deployment across both PoS and PoW networks, or embedded within core VM logic.

IMPLEMENTATION NOTES

[00014] All system logic conforms to the one-action-one-mint rule: each action results in one immutable record.

[00015] Action schemas and attestation logic live in stateless modules or execution contexts.

[00016] No additional on-chain state is required beyond the mint; the system is fully reconstructable via mint replay.

[00017] This pattern ensures deterministic behavior, auditability, and frictionless downstream integration.

CLAIMS

1. A method for recording system events in a decentralized memory-driven ledger, comprising:
 - a. defining, by a processor, a plurality of action types, each with specified input and output schemas and unique identifiers;
 - b. generating, by the processor, a cryptographic attestation binding the inputs and outputs of an action execution;
 - c. minting, by the processor, a single on-chain transaction encapsulating the attestation, action identifier, and metadata;

PATENT APPLICATION

- d. subscribing, by downstream modules, to the minted transaction as the exclusive proof object for triggering subsequent processing;
2. The method of claim 1, further comprising reconstructing system state by replaying the sequence of minted transactions through a stateless engine.
3. The method of claim 1, wherein the cryptographic attestation is a hash of the concatenated inputs and outputs.
4. The method of claim 1, wherein multiple action mints are batched into aggregated transactions for efficiency.
5. The method of claim 1, wherein zero-knowledge proofs are used to attest action correctness without revealing underlying data.
6. The method of claim 1, wherein the action logic and minting execution occur within stateless protocol modules, with no retained on-chain memory or versioned state.
7. The method of claim 1, wherein downstream smart contracts or agents rely exclusively on the presence of an action's mint identifier to initiate logic, without inspecting on-chain state.
8. The method of claim 1, wherein agents or artificial cognition modules reconstruct execution context or synchronize behavior by replaying a sequence of minted transactions.
9. The method of claim 1, wherein action minting is authorized only upon the presence of multiple cryptographic attestations corresponding to required participant identities.
10. The method of claim 1, further comprising a validation layer in which trait modules assess attestation logic for coercion, drift, or scope mismatch prior to minting.

PATENT APPLICATION

ABSTRACT

[00018] A system and method for recording system events in decentralized memory-driven ledgers. Each discrete action is captured by a single on-chain mint transaction containing a cryptographic attestation, enabling atomic proof, reducing storage complexity, and allowing downstream agents and processors to subscribe to a single, immutable record as the authoritative source of truth. The system supports stateless replay, AI coordination, and full auditability without auxiliary logs or mutable state.