Identity-Bound Memory Rights

1. Abstract:

A method for binding immutable memory streams to unique, persistent identities on a decentralized memory-driven blockchain, ensuring that each identity's historical events are non-transferable, tamper-proof, and enforce rights to access, control, and preserve personal memory records.

2. Technical Field:

This invention relates to identity management and data rights on blockchain systems, and more particularly to methods for attaching memory rights to persistent, non-transferable identity anchors.

3. Background:

Decentralized ledgers record vast histories of transactions and events, but lack a mechanism to securely bind those records to unique, persistent identities in a way that enforces rights over personal data. There is a need for a system that ensures memory streams are irrevocably linked to identities, preventing unauthorized transfer or tampering, and preserving ownership.

4. Summary:

The Identity-Bound Memory Rights method comprises:

Generating a non-transferable identity anchor (e.g., soulbound address) for each user or entity.

Binding memory streams—chronological sequences of on-chain events—to the identity anchor via cryptographic links.

Assigning rights over memory streams, including Access Right, Control Right, and Preservation Right.

Enforcing rights by validating identity signatures before allowing memory retrieval or modification.

Logging any unauthorized access attempts or rights violations as immutable audit events.

5. Detailed Description:

At a conceptual level, each user or entity is issued a persistent, non-transferable identity anchor—such as a soulbound token or unique address—at network genesis or account creation. Memory streams, composed of on-chain minted events, are cryptographically linked to the identity anchor, forming an immutable record. Rights metadata stored alongside the stream define who may read, append, or delegate portions of the memory.

Any request to interact with the memory stream must present a valid signature from the identity anchor. Unauthorized requests generate immutable audit logs capturing the violation details.

6. Method Flow:

Step 1: Identity Anchor Creation – Generate a non-transferable identity token or address bound to the user.

Step 2: Memory Event Linking – For each new memory event, include the identity anchor in the event metadata and cryptographic hash.

Step 3: Rights Assignment – Define rights (Access, Control, Preservation) in on-chain metadata associated with the identity.

Step 4: Rights Enforcement – On memory retrieval or modification request, verify the request signature against the identity anchor and rights metadata.

Step 5: Audit Logging – If a request fails verification, mint an immutable audit event recording the attempted violation.

7. Narrative Worked Example:

Alice creates an identity anchor 'ID_Alice'. She performs two memory actions: posting a health record and a personal note. Each event is minted on-chain with 'ID_Alice' in the metadata. Later, Alice requests her personal history, signs the request with her private key, and retrieves the stream. An unauthorized request by Eve fails signature verification and generates an audit event 'UnauthorizedAccessAttempt(ID_Alice, Eve)'.

8. Algorithmic Worked Example:

Pseudocode:

```
1.  id_anchor = createSoulboundID(user_public_key)
2.  for event in memory_events:
3.     event.metadata.identity = id_anchor
4.     event.hash = sha256(event.data || id_anchor)
5.     mint_event(event)
6.  request = receiveRequest()
7.  if verifySignature(request.signature, id_anchor) and hasRight(id_anchor, 'Access'):
8.     return fetchMemoryStream(id_anchor)
9.  else:
10.    mint_audit_event('AccessViolation', id_anchor, request.origin)
```

9. Potential Embodiments:

Using biometric or multi-factor authentication bound to the identity anchor for enhanced security.

Employing zero-knowledge proofs to grant selective access without revealing full identity.

Multi-chain identity anchors where streams on different blockchains validate against a shared anchor.

Delegated memory rights allowing users to grant temporary, revocable access to third parties.

10. Implementation Notes:

Identity anchors and memory events follow a one-action-one-mint paradigm. All rights metadata and audit logs are stored on-chain. No off-chain services are required for core enforcement; signature verification and metadata checks occur within protocol modules.

11. Claims:

1. A method for binding memory streams to persistent identities on a blockchain, comprising:

a. generating, by a processor, a non-transferable identity anchor for a user or entity;

b. linking, by the processor, each memory event to the identity anchor via cryptographic metadata;

c. defining, by the processor, rights metadata (Access, Control, Preservation) associated with the identity anchor;

d. verifying, by the processor, a request signature against the identity anchor and rights metadata before memory access;

e. minting, by the processor, an immutable audit event upon rights violation.

2. The method of claim 1, wherein the identity anchor is a soulbound token minted at account creation.

3. The method of claim 1, wherein rights metadata is stored as on-chain attributes linked to the identity anchor.

4. The method of claim 1, further comprising allowing delegation of rights via revocable credential events.

5. The method of claim 1, wherein verification is performed within on-chain protocol modules without external calls.