Custody Protocol — Final Implementation Invariants

This document captures the final core architectural decisions before implementation begins.
These invariants are consensus-critical and must not be violated without a protocol version bump.

1. Cryptographic Standards

Hash Function: BLAKE3

All protocol hashing (transaction preimage hashing, ID derivation, reference hashing, optional state hashing)
must use BLAKE3 exclusively.

Transaction signing preimage:
BLAKE3("CUSTODY_TX_V1" || SCALE_ENCODE(envelope_without_signature))

No alternative hash functions permitted within consensus logic.

2. Nonce Rules (Strict Mode)

Nonce enforcement rule:

Incoming transaction nonce must equal exactly (last_nonce + 1).
Any deviation results in ERR_NONCE_MISMATCH.

Storage:
N|canon(signer) -> u64 last_nonce

This guarantees strict linear ordering of actions per signer and prevents replay or gaps.

3. Policy Snapshot Invariant

IntentStateV1 stores:
- policy_set_id
- policy_version
- required_threshold
- required_claims snapshot

All approval validation and execution validation MUST reference the snapshotted policy version, not the currently active policy pointer.

This ensures historical integrity and prevents governance changes from altering previously proposed intents.

4. Intent Index (Required)

An index is required to allow listing intents per vault in time order.

Index key layout:
IV|||| -> 1

Lexicographic ordering guarantees chronological iteration.

This index must be written atomically with IntentState creation.

5. Determinism Guarantees

• All RocksDB prefix scans must iterate in lexicographic order.
• No floating-point arithmetic in consensus code.
• All uint128 values must be encoded as decimal strings in JSON.
• No wall-clock time usage — only block time from TxContext.
• All vectors inside stored state must be canonically sorted.
• Enum tags must never be reordered (append-only).

6. Event Emission Rules (Future Push Model Compatibility)

All ABCI events must be derived strictly from post-state writes.

Event attributes must include stable identifiers only:
workspace_id, vault_id, intent_id, asset_id, destination_id, status, policy_set_id, policy_version.

No non-deterministic fields (timestamps beyond block time, random identifiers).

7. Security Boundaries

• Private keys are never stored inside the node.
• Signature verification only.
• All write operations must occur through a RocksDB WriteBatch.
• All state transitions must be idempotent and deterministic.

8. Upgrade Strategy

Any change to:
- Hash function
- Nonce rule
- Policy snapshot semantics

- Key layouts
- Enum tag ordering

requires a protocol version bump and coordinated upgrade.

Backward compatibility must not be assumed implicitly.