Atho Transaction Reference (Current)

Date: 2026-04-05

This is the authoritative transaction behavior reference for current development builds.

Canonical Policy Metric

Atho uses SegWit-style policy metrics:

- weight = base_bytes * 3 + total_bytes
- vsize = ceil(weight / 4)

vsize is used for:

- fee floor checks,
- mempool admission,
- miner packing,
- transaction size policy limits.

Active policy constants:

- fee floor: 500 atoms/vB
- minimum tx fee: 200,000 atoms
- tx confirmations required for regular spend visibility: 10

Binary Transport Codec

Current serializer emits compact binary payload magic:

- ATX2 (current)

Decoder accepts:

- ATX2
- ATX1 (legacy decode compatibility)

Code:

- Src/Transactions/txbinary.py
- Src/SigWit/sigwit.py

Field Strategy (Compact)

Inputs:

- spend reference: tx_out_id (<96-hex-txid>:<vout>)
- optional non-zero sequence via bitmap + varint in binary codec
- no full signature embedded per-input

Outputs:

- amount encoded from integer atoms
- destination binding encoded as 48-byte raw HPK digest
- lock flags compacted with bitmap

Witness:

- signature and public key are raw bytes in binary payload
- API/UI wire display remains canonicalized for compatibility/human tooling

Signature Reference Token

Input script_sig is a compact reference token, not a full signature blob.

Current canonical prefix:

- SR:<16-hex>

Legacy SIG_REF:* acceptance is policy-gated and disabled in current defaults.

- Integer Accounting
Consensus-critical tx value logic uses integer atoms. Display conversion to decimal strings is
presentation-only.

Examples:
- amount_atoms
- fee_atoms
- exact conservation checks in verifier paths
- Witness Size Policy
Current Falcon witness policy bounds:
- signature target ~666 bytes (compressed range accepted by policy)
- canonical public key length 897 bytes
- legacy public key length acceptance is migration-policy controlled
- Measured Size Profile (Hard Data)
There is no single fixed tx size. Size varies with:
- input count
- output count
- private vectors
- metadata
- witness payload length

The table below is measured from the live serializer + policy path:
- Transaction.to_dict()
- serialize_tx_v1(...)
- Constants.tx_policy_metrics(...)

Snapshot date: 2026-04-05

Flow Shape	Base Bytes	Total Bytes	Weight	vsize	Binary Bytes
coinbase (1 out)	290	290	1160	290	290
public 1 in / 1 out	121	1634	1997	500	1634
public 1 in / 2 out	174	1687	2209	553	1687
public 2 in / 2 out	223	1736	2405	602	1736
public 3 in / 2 out	272	1785	2601	651	1785
public -> private 1 in / 1 private out	2092	3605	9881	2471	3605
private -> public 1 private in / 1 public out	2329	3842	10829	2708	3842
private -> private 1 private in / 1 private out	4294	5807	18689	4673	5807
private -> public + private change 1 private in / 1 private out + 1 public out	4301	5814
18717	4680	5814
- Private Multipliers vs Standard Public Send
Using public 1 in / 2 out (553 vB) as baseline:
- public -> private: 2471 / 553 = 4.47x
- private -> public: 2708 / 553 = 4.90x
- private -> private: 4673 / 553 = 8.45x
- private -> public with private change: 4680 / 553 = 8.46x

This confirms current private flows are not ~21x for representative
single-input/single-recipient shapes. They are currently ~4.5x to ~8.5x, with larger multiplie
when private input/output counts increase.
- Private Size Sensitivity (Measured)
From the same measurement run:

- private -> private without view-tag note message: 4675 vB
- private -> private with view-tag: 4679 vB (+4 vB)
- private -> private with 32-byte note message: 4702 vB (+27 vB)
- private -> private with 2 private outputs: 6644 vB (+1969 vB)
- private -> private with 2 private inputs: 6931 vB (+2256 vB)

Interpretation:
- each additional private vector has a clear linear-ish cost,
- the private path overhead is dominated by encrypted payload and private proof material,
- witness still matters, but private vectors are the primary driver.
- Hashing Model
- txid: no-witness hash
- wtxid: full-transaction hash (with witness)
- signing digest: deterministic no-witness signing body hash

Hashing and signing internals:
- Src/Utility/txdhash.py
- Src/Transactions/tx.py
- Src/Transactions/txvalidation.py
- Validation Invariants
Transaction acceptance requires:
- valid structure and network identity,
- valid signature over no-witness digest,
- existing/unspent inputs,
- no double spend,
- atom-exact value conservation,
- fee >= required fee from canonical vsize.

11.1) Role-Aware Lockup Flows (Bond and Stake)
Bonding and staking are represented as normal transaction flows to deterministic role-derived destinations, with additional consensus state handling:
- bond deposit/top-up to bond-derived address,
- stake deposit/top-up to stake-derived address,
- exit/withdraw paths gated by state and height rules.

Consensus rejects malformed or unauthorized role-path interactions, even when base transaction format is otherwise valid.

Key point:
- display addresses can vary by UI form,
- consensus ownership/eligibility checks are driven by deterministic pubkey-derived role digests.
- Throughput Mapping (Measured)
At 3,500,000 vB per block and 120s block time:

Formula:
- TPS_ideal = 3,500,000 / (120 * avg_vsize)
- TPS_95 = TPS_ideal * 0.95 (practical packing headroom)

Flow Shape	vsize	TPS (ideal)	TPS (95% pack)
public 1 in / 2 out	553	52.74	50.11

public -> private	2471	11.80	11.21
private -> public	2708	10.77	10.23
private -> private	4673	6.24	5.93

Conclusion:
- public throughput envelope remains high for standard payments,
- private throughput is intentionally lower due stronger privacy payloads,
- real chain TPS depends on the live tx mix in each block.