ATHERON

AI-POWERED. DATA-DRIVEN. PROFIT-FOCUSED.

ADVANCED AI TRADING PLATFORM

Technical Whitepaper

A Layer 1 whose graphics-card mining lane performs the same arithmetic as transformer inference, the compute network that runs on it, and the on-chain ledger that makes an AI trading engine's record checkable.

Document	Network	Status	Date
Technical Whitepaper v1.0	Atheron · ATON	Pre-testnet. Open parameters marked.	28 August 2026

00 Contents

– Important notice

This document is not an offer. Nothing in it is an offer to sell, or a solicitation of an offer to buy, any security, token or other instrument, in any jurisdiction. There is no public sale of ATON, there has been no private round, and none is planned. ATON is acquired by mining it or, once listed, by buying it on the open market from somebody who did.

Nothing here is investment or financial advice. This document describes a protocol and a piece of software. It does not recommend any course of action and it makes no claim about the future value of anything. Automated trading, mining and holding a digital asset all carry the risk of losing money.

Forward-looking statements may not happen. Sections describing what will be built, when it will arrive and how it will behave are statements of present intention against a plan. They are not promises. Several depend on results that do not yet exist, and the document says so where that is the case. Dates in this paper are expressed as months from the start of the build rather than as calendar dates, for exactly that reason.

The regulatory position of ATON is unsettled and untested. Atheron Network Inc. is a British Columbia corporation. How ATON is classified in Canada, the United States or anywhere else has not been determined by any regulator or court, and nothing in this document is a legal opinion on that question. The project has taken

the design decisions it believes reduce that exposure, which are described in this paper, and those decisions are not a substitute for a determination nobody has made.

Some parameters are not final, and are marked. Consensus parameters that depend on measurement rather than preference are given as targets with their resolution path stated. Section 30 lists every one of them in a single table so that no reader has to hunt for what is settled and what is not.

No warranty. The software described here is under development. It is provided, when it exists, without warranty of any kind. Access may be restricted in some jurisdictions and it is the reader's responsibility to know the rules that apply to them.

This document will be revised. Every revision is recorded in the project's public changelog with a date against it. The current version is stated on the cover.

01 Abstract

Atheron is a Layer 1 blockchain whose graphics-card mining lane performs the same arithmetic as transformer inference, so the hardware that secures the chain is the hardware that runs the network's AI workloads. This paper specifies the protocol, the execution environment, the compute network layered on it, the decision-recording ledger that makes an AI trading engine's claims checkable, and the token that pays for all of it.

Proof of work is usually criticised for spending energy to produce a number that has no use beyond proving it was expensive to find. Atheron's second mining lane, Atheron-TMH, is built from FP16 matrix multiplication over a working set sized past typical GPU cache. Winning that lane requires the same hardware profile that runs transformer inference well: tensor cores and high memory bandwidth. That is the mechanical basis for the network's central claim, which is that a leased GPU can mine and serve inference from the same silicon, and that the operator can choose between them on price.

On top of that sits a dual virtual machine execution layer, giving both an EVM for compatibility with existing tooling and a native WebAssembly engine with precompiles reaching directly into chain-specific state. A Compute Mesh schedules inference and oracle jobs across staked nodes with redundant execution and majority agreement. A Signal Ledger commits the decisions of the network's trading engine on-chain at the moment they are made, which converts an unverifiable claim about a track record into a record anybody can check.

The token, ATON, has a fixed hard cap of 15,000,000,000 and no public sale at any point. Eighty per cent is issued by mining on a smooth emission curve equivalent to a halving every four years. Section 23 gives every figure in units rather than percentages.

CONSENSUS	MINING	EXECUTION	HARD CAP
GhostDAG	**Two lanes**	**Dual VM**	**15,000,000,000**
PHANTOM family, account-based state	BLAKE3 ASIC and tensor-memory-hard GPU	revm EVM and wasmtime WASM	ATON, no public sale

02 The problem

Three unrelated inefficiencies, which turn out to share a solution.

Proof of work produces heat and nothing else

The security of a proof-of-work chain comes from the cost of the work. That cost is real and the security is real, but the work itself is discarded the moment it has served its purpose. Attempts to fix this by making the work useful have generally failed on one of two counts: either the useful work is not verifiable cheaply enough to serve as consensus, or making it useful makes it centralising, because the useful version favours one operator's specialised hardware.

Atheron does not claim to have solved that problem in general. It makes a narrower move: the work stays conventional hash-finding, and is verified exactly as a hash is verified, but the *shape* of the computation is chosen so that the hardware which wins at it is the hardware the network separately needs for something else. The work is still thrown away. The machines are not.

Decentralised AI inference has no economic anchor

Several projects have built networks for distributing machine-learning inference across independently operated hardware. The recurring difficulty is demand: a network of GPUs with nothing to run is a network of idle GPUs, and operators leave. Bootstrapping requires a workload large enough to keep nodes paid before third-party demand exists.

Atheron's mesh has such a workload from the first day it runs, because the network's own trading engine is a continuous consumer of inference, and because the same redundant-execution machinery serves oracle requests for any contract on the chain. Mining rewards provide a floor under node economics while that demand grows.

An AI trading product cannot prove anything it claims

Every automated trading product in the market reports its own results. A reader has no way to distinguish a good record from a selectively presented one, because the party with the strongest incentive to shade the number is also the only party holding it. Screenshots, dashboards and self-published statistics are all the same claim wearing different clothes.

If a decision is committed to a public chain at the moment it is made, before its outcome is known, and if that commitment is produced by a redundant process the operator does not unilaterally control, the resulting record is checkable by anybody. That does not make a strategy good. It makes the *claim about its past* falsifiable, which is a different and much rarer property.

> **WHAT TIES THEM TOGETHER**
>
> One machine can secure the chain, serve the inference and earn from both. The chain gives the mesh its economic floor, the mesh gives the trading engine an execution environment nobody controls alone, and the trading engine gives the mesh its founding demand. Each piece is buildable on its own; the argument for building them together is that each one solves the hardest problem of the next.

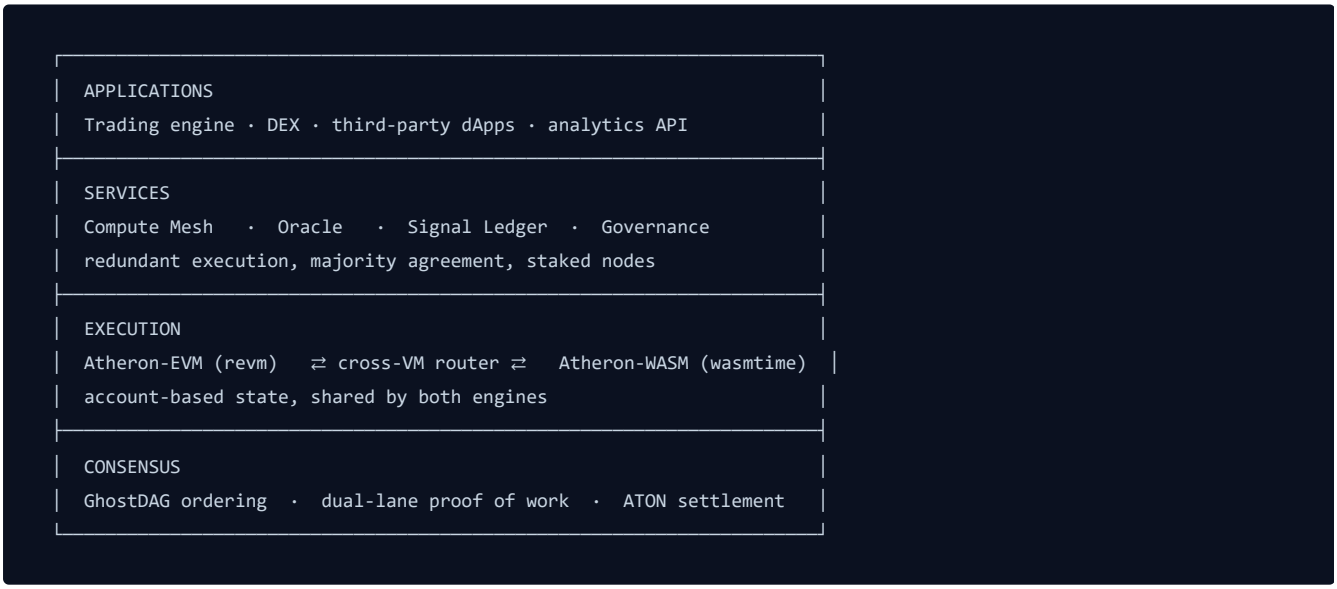
03 Design principles

Five commitments that constrain everything after this point. Where a later section makes a choice that looks inconvenient, one of these is usually why.

1. **Decentralised from the first block, not eventually.** The devnet's founding Compute Mesh nodes are consumer graphics cards. Initial model fine-tuning runs on the mesh rather than on rented cloud capacity. This costs real capability at the start, and Section 11 states plainly what it costs.
2. **Custody is never taken.** The trading engine holds no funds and no keys. A user's exchange credentials are encrypted on their own machine with a key derived from a passphrase the project never sees. The project cannot recover them, which is the only durable evidence that it does not hold them.
3. **Claims must be checkable rather than trusted.** Where the project asserts something about its own behaviour, the mechanism that makes the assertion verifiable is built alongside it. The Signal Ledger is the largest instance of this principle.
4. **Novel components are named as novel.** Atheron-TMH and the cross-VM router are the two genuinely new pieces here. Both carry disclosed emergency-patch capability under timelock, and both are described in this paper as unproven.
5. **Nothing ships to mainnet unaudited.** Section 26 lists every component whose audit is a hard gate on the mainnet date rather than a task that can follow it.

04 System architecture

Four layers, each usable without the ones above it.



The consensus layer is a true Layer 1 in the ordinary sense: its own validator and miner set, its own native settlement asset, its own security, derived from no other chain. Supporting EVM bytecode is a property of the execution layer and does not change that, in the same way that it does not for any other EVM-compatible Layer 1.

Each layer is a genuine dependency of the one above and not of the one below. The chain functions with no mesh. The mesh functions with no trading engine. The trading engine is the mesh's first large customer rather than its justification.

05 Consensus

GhostDAG ordering over an account-based ledger, tuned for high throughput from genesis rather than ramped into it later.

Why a DAG rather than a chain

A single-chain protocol must discard every block that arrives in parallel with the winner. That is what limits its block rate: raise the rate and the orphan rate rises with it until security degrades. GhostDAG, from the PHANTOM family, keeps parallel blocks instead of discarding them. It identifies a well-connected subset of the DAG, the blue set, using an anticone constraint, and derives a total order over every block from it. Blocks that would have been orphaned still contribute their transactions and still earn.

The practical consequence is that block rate stops being bounded by propagation delay in the way it is on a single chain. The cost is a more complex ordering rule and a parameter, the anticone constraint, that must be tuned against real propagation behaviour rather than chosen from preference.

State model

GhostDAG governs ordering, not state representation. The closest reference implementation uses a UTXO ledger, but that is a choice layered on top rather than a requirement of the algorithm. Because the EVM cannot function without account-based state, **Atheron's base ledger is account-based from genesis.** Mining rewards, mesh payouts and staking all credit and debit account balances directly.

> **THIS IS A REAL DEVIATION, NOT A CONFIGURATION FLAG**
>
> An account-based state machine has to be built into the node's core from the first commit. It is not a module that can be added to a UTXO client afterwards. Anybody estimating the work from the existence of a reference GhostDAG implementation should discount that estimate accordingly, and this paper would rather say so than let the reference implementation imply a shorter path than exists.

Parameters

The project has decided its tuning philosophy: target high throughput from genesis rather than launching conservatively and raising the rate later by network upgrade. That decision is made. The numbers it implies are not, because they are not the kind of thing that can be decided by preference.

PARAMETER	POSITION	HOW IT SETTLES
Consensus family	GhostDAG / PHANTOM. **Decided.**	Reference implementation exists in Rust and is adapted rather than written from zero.

PARAMETER	POSITION	HOW IT SETTLES
State model	Account-based from genesis. **Decided.**	Required by the EVM; built into the core state machine.
Genesis block rate	Target 4 to 10 blocks per second. **Target, not a commitment.**	An output of Phase 1 testnet measurement. If the rate is not safely achievable, the honest response is to delay mainnet or lower the genesis target.
Anticone constraint, k	Exploration begins at k=24, likely higher. **Open.**	Load-bearing testnet deliverable. Higher block rates need more tolerance for parallel blocks to stay safe.
Block reward split	50% Lane A, 50% Lane B. **Decided.**	Fixed at the protocol level. See Section 07.
Finality	Virtual chain selection with pruning-point checkpoints. **Decided.**	Mechanism fixed; the pruning window is a testnet tuning item.
Pruning window	**Open.**	Testnet tuning against real storage growth.

> **WHY THESE ARE PUBLISHED AS RANGES**
>
> A whitepaper that states a block rate before any testnet has measured one is stating a preference in the grammar of a fact. The parameters above are the ones the project has committed to determining by measurement, and Section 30 collects them, with the rest of the open items, into a single table. When each is settled the value is published and the change goes in the public changelog with a date on it.

06 Block structure and validation

What a block commits to, how a node validates one, and what changes because blocks arrive in parallel rather than in a line.

Header

The header is what a miner searches over and what a verifier checks. It commits to the parent set rather than to a single parent, which is the structural difference between a DAG protocol and a chain.

```
BlockHeader
  version           u16     protocol version
  parents           [Hash]  the DAG parent set, not a single parent
  merkle_root       Hash    transactions in this block
  state_root        Hash    account state after applying this block's order
  utxo_commitment   —       absent; the ledger is account-based
  timestamp         u64     milliseconds
  lane              u8      0 = Atheron-B3, 1 = Atheron-TMH
  bits              u32     compact difficulty target for that lane
  nonce             u64     the searched value
  blue_score        u64     cumulative blue-set weight, from ordering
  pruning_point     Hash    the checkpoint this block builds past
```

Two fields deserve comment. `parents` is a set because a block may reference every tip it saw, and referencing more of them is what lets parallel work be kept rather than orphaned. `lane` selects which proof-of-work function the header is validated under, and is committed inside the header so a block cannot be replayed against the other lane's difficulty.

Validation order

1. **Structural checks.** Header well-formed, parents known, timestamp within the permitted drift, lane value valid.
2. **Proof of work.** Recompute the lane's function over the header and check the result against `bits`. For Lane A this is a BLAKE3 evaluation. For Lane B it is the full tensor-memory-hard derivation, which is expensive to search and cheap to verify only once.
3. **Difficulty.** Check `bits` against the retarget schedule for that lane independently of the other lane.
4. **Ordering.** Compute the block's position in the virtual chain by GhostDAG's rule and derive `blue_score`.
5. **Execution.** Apply the block's transactions in the derived order and check the resulting `state_root`.
6. **Pruning consistency.** Check the block builds past a pruning point the node accepts.

> **VERIFICATION COST IS ASYMMETRIC, AND THAT IS THE POINT**
>
> Lane B's function is deliberately expensive to evaluate: a large working set and many rounds of matrix multiplication. A verifier evaluates it once per block; a miner evaluates it once per nonce attempt. That asymmetry is what makes it usable as proof of work at all.
>
> It does mean Lane B verification costs more than a hash check, and that a node validating history is doing real arithmetic rather than trivial work. Whether that cost is acceptable at the target block rate is one of the things Phase 1 testnet has to measure, and it is on the open-parameter list in Section 30 for that reason.

Ordering and finality

GhostDAG produces a total order over the DAG by identifying the blue set and walking it. A node's view of that order can change as blocks arrive, in the same way a chain can reorganise, but the depth at which it stops changing is a function of the anticone constraint and the block rate rather than of a fixed confirmation count.

Finality in practice is therefore probabilistic and deepening, with pruning-point checkpoints providing a floor past which reorganisation is not accepted. **The confirmation depth that corresponds to a given confidence is a measured property, not a design choice**, and it depends on the same parameters Section 30 lists as open. Publishing a confirmation count before measuring one would be stating a preference as a fact.

Data availability and pruning

Full history growth at a high block rate is a real operational constraint, not a footnote. Nodes prune past the pruning point, retaining headers and the state needed to validate forward. Archive nodes retaining everything are expected to exist but are not required for consensus. The pruning window is a tuning item measured against real storage growth on testnet.

07 Dual-lane proof of work

Two independent proof-of-work functions, each producing half the block reward, each retargeting separately, with a dampener that stops either lane taking the network.

Lane A — Atheron-B3

A BLAKE3-based proof of work. Nonce search over a block header committing to the DAG parent set and the state root. It is deliberately ASIC-favourable. That is the lane's purpose: specialised hardware is efficient, it is durable, and a network with an ASIC lane has a class of participant whose capital is committed to that network specifically.

Lane B — Atheron-TMH, tensor-memory-hard

The novel component. The function is constructed so that the hardware which computes it efficiently is the hardware that runs transformer inference efficiently.

1. BLAKE3 expands the block header and nonce into a seed.
2. The seed deterministically generates a large pseudo-random weight matrix, with a working set of 512MB to 1GB, sized past typical GPU L2 cache so that memory bandwidth rather than cache residency is the binding constraint.
3. N rounds of FP16 or BF16 matrix multiplication run over that matrix. This is the same arithmetic and the same memory access pattern as transformer inference.
4. A final BLAKE3 pass compresses the resulting tensor state into the candidate block hash.

Verification is cheap in the way proof of work requires: a verifier reruns the derivation from the header and nonce and checks the result. The expensive part is finding a nonce, not confirming one.

ATHERON-TMH IS NOVEL AND UNPROVEN, AND THIS PAPER WILL NOT PRETEND OTHERWISE

No proof-of-work function of this shape has been run adversarially at scale. It requires independent cryptographic and consensus audit, and a long public testnet under adversarial conditions, before mainnet. Specific risks that audit must address include: whether the matrix generation admits a shortcut that skips the memory-bound work; whether reduced-precision arithmetic introduces cross-vendor non-determinism that would fork the network along hardware lines; and whether an optimised implementation collapses the intended hardware profile onto a narrower one than intended.

A governance-level fallback exists to swap the algorithm if it is found to be gameable, held under the scoped, time-locked and publicly visible arrangement described in Section 22. A reader who considers this risk unacceptable should weigh Lane A, which uses a standard and well-understood function, accordingly.

Anti-monopoly balancing

Each lane retargets difficulty independently, approximately every 2,880 blocks. On top of that sits a lane-balance dampener: if either lane produces more than 65% of blocks over a trailing 2,016-block window, its next retarget receives an additional corrective multiplier. Both thresholds are stated here as the current design and both need testnet simulation before they are fixed.

The purpose is not to force an even split, which would be arbitrary, but to stop one lane's economics collapsing the other's. A network where one lane has become irrelevant has the security properties of a single-lane network while carrying the complexity of two.

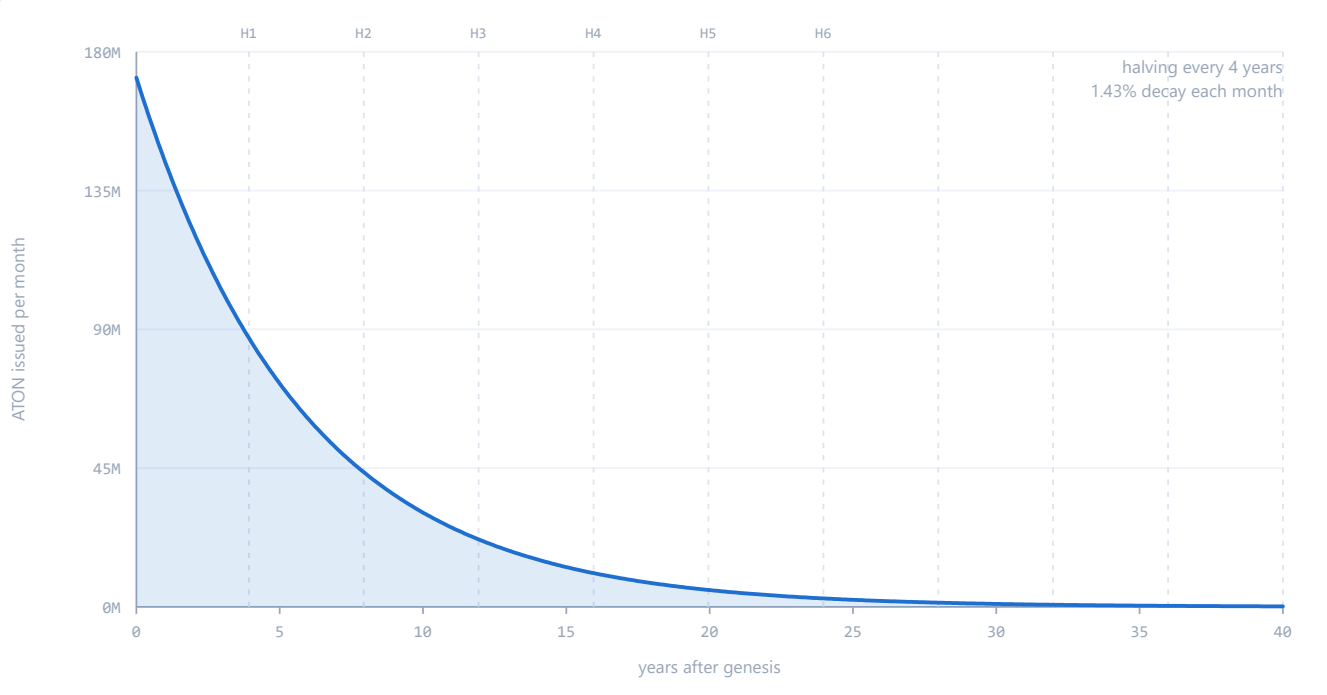


Figure 1 Monthly mining subsidy across forty years, computed from the 1.43% monthly decay. Halving-equivalents are marked H1 to H6. The two lanes divide every month's subsidy equally.

08 Difficulty and lane balance

Each lane retargets on its own schedule, with a corrective term that engages only when one lane is taking the network.

Independent retargeting

Each lane retargets approximately every 2,880 blocks *of that lane*, against that lane's own observed rate and its own share of the target block interval. The lanes do not share a difficulty. An ASIC farm arriving on Lane A raises Lane A's difficulty and leaves Lane B untouched, which is the intended behaviour: the lanes are separate markets for separate hardware.

The dampener

Independent retargeting alone permits a stable state where one lane produces nearly every block, because each lane retargets to its own rate regardless of the other's share. The dampener is what prevents that.

```
at each retarget for lane L:
    share = blocks_from_L over the trailing 2016-block window
                  ÷ all blocks in that window

    base  = standard retarget factor for lane L

    if share > 0.65:
        factor = base × corrective(share)      # raises L's difficulty further
    else:
        factor = base

corrective() is monotonic in share and equals 1.0 at the threshold,
so the dampener engages smoothly rather than as a step.
```

The 65% threshold and the 2,016-block window are the current design and both need testnet simulation against realistic two-lane economics before they are fixed. They are on the open-parameter list.

> **WHY NOT FORCE AN EVEN SPLIT**
>
> A hard 50/50 enforcement would mean that if one lane's hardware became genuinely scarce or unprofitable, the network would stall waiting for blocks that were not coming. The dampener corrects a drift toward monopoly without pretending the two markets must always be the same size. An uneven but stable split is a healthy outcome. A lane at 95% is not.

09 Execution layer

Two virtual machines over one account-based state: an EVM for the tooling that already exists, and a WebAssembly engine for the things only this chain can offer.

Atheron-EVM

Built on **revm**, a mature Rust implementation of the EVM also used by Reth and Foundry. Embedding a known-good implementation is a substantially lower risk than writing one, and the EVM is not a component where novelty is rewarded.

- Standard Ethereum JSON-RPC over the `eth_*` namespace, so MetaMask, Hardhat, Foundry, ethers.js and web3.js work against Atheron with no new tooling for a Solidity developer.
- Native ATON is the gas token inside the EVM environment, represented as an 18-decimal wei-equivalent. This is the same pattern every non-Ethereum EVM Layer 1 uses.
- The chain ID is reserved early through the public registry at chainid.network, so wallets, explorers and tooling can be built against a stable, collision-free identifier well before mainnet rather than in the days before it.

> **THREE CHAIN IDS, NOT ONE**
>
> Devnet, testnet and mainnet each get their own registered chain ID. Sharing an ID across networks would make a transaction signed on one replayable on another, which makes this a security item rather than administration. The devnet and testnet identifiers are reserved as part of Phase 0.

Atheron-WASM

Built on **wasmtime**, with an Atheron-specific contract ABI and a Rust SDK. Developers write contracts in Rust, compile to `wasm32-unknown-unknown` and deploy through a native CLI. The design is in the spirit of CosmWasm and ink! rather than a new paradigm.

What the WASM engine has that the EVM side does not is **native precompiles** reaching directly into chain-specific state: Compute Mesh job submission, staking operations, mining-lane statistics and the Signal Ledger. A contract that needs to schedule an inference job and act on the result is a WASM contract. A contract that needs an existing Solidity library and an audience of Solidity developers is an EVM contract. Both are first class.

The Compute Mesh's own staking, job-assignment and slashing logic is itself implemented as native WASM contracts rather than hardcoded consensus rules. That is a deliberate trade: a newer and less battle-tested execution path now, in exchange for the ability to change mesh mechanics by contract upgrade rather than by hard fork later. It also makes the WASM engine's audit a hard gate on mainnet rather than a desirable extra.

Cross-VM router

By default the two engines are separate state machines. Letting an EVM contract call a WASM contract, or the reverse, requires a router that translates calls and return data between two ABIs with different type systems, different gas accounting and different failure semantics.

The router ships at mainnet. That is a decision, and it is a decision that moves the mainnet date rather than one that can be deferred past it. Every comparable dual-VM chain has found its hardest bugs in exactly this translation layer.

- Multiple independent audit passes on the router specifically, not only on the two engines individually.

- A public testnet period long enough for adversarial cross-VM call fuzzing, rather than each engine tested in isolation.
- A conservative allowlist of permitted cross-VM call patterns at mainnet, widened after a stability period, rather than an unrestricted router on day one.

Gas abstraction

A native account-abstraction layer following the well-established ERC-4337 pattern lets a dApp sponsor its users' gas, or lets a user pay gas in a token other than ATON with conversion handled underneath. A new application can therefore onboard somebody who has never held ATON. The pattern is proven elsewhere, so this is engineering effort rather than novel risk.

	ATHERON-EVM	ATHERON-WASM
Runtime	revm	wasmtime
Language	Solidity, Vyper	Rust
Tooling	Existing Ethereum tooling, unchanged	Native CLI and Rust SDK
Mesh access	Through the cross-VM router	Direct, via precompiles
Staking and lane data	Through the router	Direct, via precompiles
Best for	Porting existing contracts, reaching Solidity developers	Anything that integrates with what makes this chain different
Audit status at mainnet	Hard gate	Hard gate, and the mesh depends on it

10 State, gas and transactions

The account model both engines share, how work is metered, and what a transaction can be.

Account model

One account space, shared by both virtual machines. An account holds a balance in native ATON, a nonce, and, where it is a contract, code and storage. Mining rewards, mesh payouts, staking movements and slashing all credit and debit these balances directly. There are no UTXOs anywhere in the system.

Because the space is shared, an address means the same thing to both engines. A balance is not duplicated per VM and does not need bridging between them. What needs translating at the router is *calls*, not *value*, which is a materially smaller problem than a chain running two ledgers would have.

Gas

- **One gas token.** Native ATON, represented inside the EVM as an 18-decimal wei-equivalent, the same pattern every non-Ethereum EVM Layer 1 uses.

- **Two schedules.** The EVM meters by the Ethereum gas schedule so that existing contracts behave predictably. The WASM engine meters its own instruction and host-call costs. A cross-VM call is metered on both sides of the boundary, and the router's own translation cost is charged to the caller.
- **Fees go to miners**, with a portion to stakers under the fee-sharing parameter, which is governable rather than fixed in this paper.
- **Gas abstraction** lets a dApp sponsor its users' gas, or lets a user pay in another token with conversion underneath. The pattern is ERC-4337-shaped and well proven elsewhere, so it is engineering rather than novel risk.

Transaction types

TYPE	PURPOSE
Transfer	Move ATON between accounts.
EVM call and deploy	Ordinary Ethereum-shaped transactions, unchanged.
WASM call and deploy	Native contract invocation and deployment.
Cross-VM call	Routed invocation between engines, subject to the allowlist.
Stake and unstake	Compute Mesh bonding and release.
Job submission	Enqueue a mesh job with its class, redundancy requirement and escrowed fee.
Sponsored transaction	Executed on behalf of an account whose gas another party pays.
Governance	Proposal, vote and timelocked execution.

State growth

A high block rate produces state and history faster than a slow one, and this is a genuine operational cost rather than a rounding error. Nodes prune past the pruning point, keeping headers and the state needed to validate forward. The pruning window is measured on testnet against real growth rather than assumed, and it appears on the open-parameter list for that reason.

11 The Compute Mesh

A network of staked nodes that executes inference and oracle work redundantly, agrees on results by majority, and pays for them in ATON.

How a job resolves

1. A job is submitted, either by the trading engine, by a third-party dApp through the Developer Platform, or by any contract needing an oracle reading.
2. The job is replicated to **N nodes**, N=5 by default.
3. Each node computes independently and returns a result hash.

4. **Three matching hashes out of five** finalises the job and releases payment.
5. A node returning a non-matching result is **slashed 2% of its stake**.
6. For trade-triggering work, N rises to **9 with a threshold of 6**.

The redundancy factor is what turns a set of independently operated machines into something whose output can be relied on without relying on any one of them. It is also what makes the same machinery usable as an oracle with no new trust assumption: a price feed answered by five independent nodes that must agree is exactly the structure already built for inference.

Node economics

ITEM	VALUE	REASONING
Minimum stake	1,000 ATON	A deliberate middle: low enough that somebody running one or two consumer cards can take part, high enough that operating fraudulent nodes at scale costs real money. Higher stake unlocks higher-value job tiers.
Default redundancy	N = 5, threshold 3	Tolerates two faulty or dishonest nodes per job.
Trade-triggering redundancy	N = 9, threshold 6	Two-thirds agreement where a wrong result would reach somebody's account.
Slashing	2% of stake per disagreement	Enough to deter dishonest results, not so punitive that an honest operator with a hardware fault is destroyed by it.
Income	Block rewards from Lane B, plus per-job fees	Two independent income streams from one machine. The mining reward is the floor under node economics while third-party job demand grows.

The local scheduler

Each node runs a scheduler that arbitrates its GPUs between Lane B mining and compute jobs, on operator-set profitability thresholds. This is the point where the whole design becomes concrete: the operator does not choose between mining and serving inference when they buy the card. They choose continuously, on price, and the card does whichever is worth more at that moment.

Reputation

Beyond raw stake, nodes accumulate a record of historical accuracy, uptime and latency, which feeds job-matching priority. Higher-reputation nodes are matched preferentially to higher-value and latency-sensitive work, including oracle requests where staleness matters most. Reputation is a reliability mechanism layered on the stake model rather than a replacement for it, and it is its own audit item.

Third-party access from mainnet

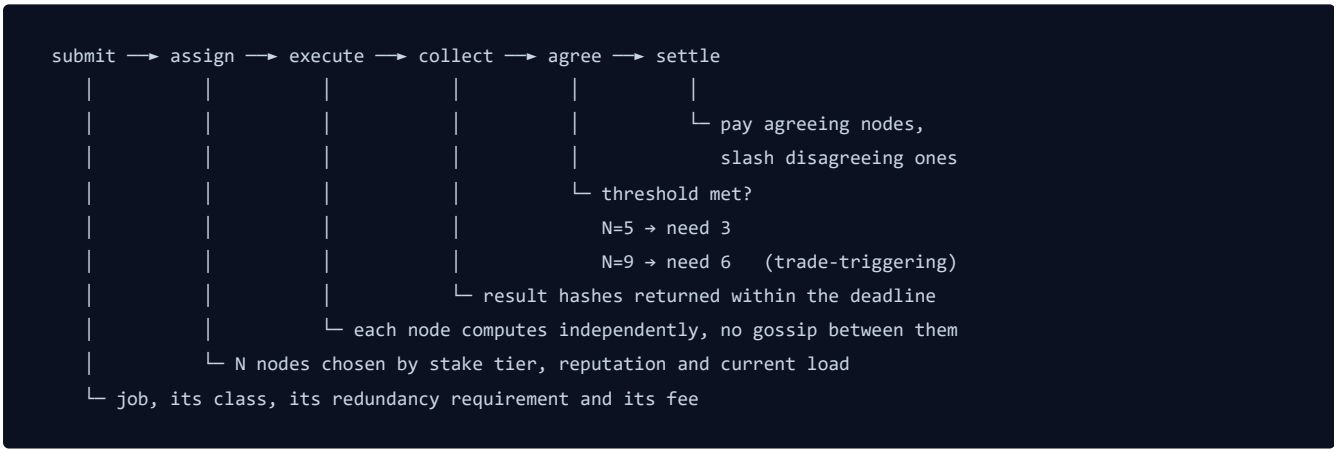
Third-party developers can submit arbitrary inference jobs from mainnet day one, not as a later phase, and can eventually deploy their own fine-tuned models for the network to serve. The mesh is intended as a general

decentralised compute utility rather than the trading engine's private backend. Service level tiers, offering guaranteed redundancy and latency for paying workloads, attach to that.

> **THE HONEST STATE OF THE MESH AT GENESIS**
>
> The founding devnet mesh is two consumer graphics cards, an RTX 3060 and an RTX 5070, 12GB each. That is not a figure of speech and it is not a placeholder for a data centre. It is a deliberate consequence of the first design principle, and it costs real capability: initial model fine-tuning starts on a 7 to 9 billion parameter model that fits a single 12GB card's QLoRA budget, rather than the 27 billion parameter variant the architecture ultimately wants.
>
> The larger model becomes practical when enough community capacity has joined to make distributed training across a homogeneous pool realistic. That is a genuine future milestone with a genuine dependency, not a downgrade being described as a plan.

12 Job lifecycle and scheduling

How a unit of work moves from submission to payment, and what happens at each way it can fail.



Assignment

Nodes are selected on three criteria: **stake tier**, which bounds the value of work a node may be given; **reputation**, being the accumulated record of accuracy, uptime and latency; and **current load**, so that a single fast node does not accumulate the whole queue. Selection is deliberately not purely highest-stake, because that converges on centralisation among the largest operators.

Isolation during execution

Assigned nodes do not communicate. Each computes independently from the job specification and returns only a result hash. If nodes could see each other's results before committing, the majority requirement would collapse into whichever node answered first, and the redundancy would be theatre.

Failure modes and what happens

FAILURE	DETECTION	CONSEQUENCE
A node returns a result disagreeing with the majority	Hash comparison at the agreement step	2% of stake slashed. Reputation falls. The job still settles from the majority.
A node does not answer before the deadline	Deadline expiry	No payment, reputation falls. Not slashed: a missed deadline is not evidence of dishonesty.
Fewer than the threshold agree	Agreement step	Job does not settle. No node is slashed on a split with no majority, because the job specification is as likely to be at fault as any node. The job is reissued to a fresh set and the specification is flagged.
A node is assigned work above its stake tier	Assignment-time check	Rejected at assignment; this is a scheduler bug rather than a node behaviour
The submitter cannot pay	Fee escrow at submission	The job is never assigned. Fees are escrowed before assignment rather than collected after.

Determinism is a requirement, not a preference

Redundant execution with hash comparison only works if honest nodes computing the same job produce byte-identical output. For inference this is a real engineering constraint: floating-point reduction order, library versions and hardware differences can all produce results that differ in the last bits and therefore differ entirely once hashed.

> **THIS IS ONE OF THE HARDEST PROBLEMS IN THE DESIGN**
>
> The mesh's correctness rests on independent nodes agreeing bit for bit. That requires pinned model versions, pinned runtime versions, a fixed reduction order, and a defined numerical mode, all committed as part of the job specification rather than left to the node. The `model version` field in a Signal Ledger receipt exists partly for this reason: it identifies exactly what was supposed to run.
>
> Where bit-exactness cannot be achieved for a class of job, the honest options are to quantise results to an agreed precision before hashing, or to accept a tolerance band and change the agreement rule accordingly. Both weaken the guarantee and both are preferable to claiming an exactness the hardware does not provide. Which applies to which job class is a testnet deliverable.

13 Oracle service

The same redundant-execution machinery, pointed at data instead of inference, with no new trust model.

A contract needing an external reading, a price, a rate, anything off-chain, submits an oracle request as a mesh job. The request is replicated across N nodes, each fetches and returns, and a majority must agree before the reading is written. The economics, the staking, the slashing and the reputation weighting are the ones already described in Section 11.

Two things follow. The chain gets an oracle without importing a separate trust assumption or a separate set of paid operators, and GPU lessors get a second revenue line from infrastructure the project is already paying to audit. Oracle requests are latency-sensitive and stale readings are the failure mode that matters, which is why reputation weighting is applied most heavily here.

> **WHAT AN ORACLE CANNOT FIX**
>
> Majority agreement establishes that five independent nodes saw the same thing. It does not establish that the thing was true. If every node reads the same upstream source and that source is wrong, the mesh will agree on a wrong number with complete integrity. Source diversity is a configuration property of each feed and a matter for whoever specifies it, and no consensus mechanism substitutes for it.

14 The Signal Ledger

An on-chain record of every decision the trading engine makes, committed at the moment it is made, containing nothing that identifies the person it was made for.

Mechanism

When the trading engine's fusion layer produces an actionable decision, that inference job has already passed through the mesh's redundant-execution pipeline: nine nodes computed it, at least six agreed on a result hash. That finalised, already-decentralised result is the natural anchor. Rather than using it only to settle mesh payment, the engine commits a structured **decision receipt** into the Signal Ledger, a native WASM contract.

The property that matters is not that a record exists. It is that the record was produced by a process the operator does not unilaterally control and cannot quietly revise afterwards. A company writing its own results to its own database has produced a claim. A receipt committed through majority agreement, timestamped before the market moved, is a fact about what was decided and when.

What is recorded, and what is not

RECORDED ON-CHAIN	NEVER RECORDED ON-CHAIN
Asset and pair	The user's exchange identity or API key
Direction: buy, sell or hold	Position size
Confidence score	Fill price
Per-engine breakdown: quant score, sentiment score, fusion weighting	Account balance
Model and version identifiers	Profit or loss
Timestamp	The raw market and social input data
A hash commitment of the input snapshot	Anything identifying the person the decision was made for

A public record of decisions is useful to everybody. A public record of positions would be a permanent liability attached to a named person, and no amount of usefulness would justify creating it. The input snapshot is committed as a hash rather than as data, which is enough to prove afterwards which inputs a decision was made from without publishing a market data feed.

What this does and does not establish

- **Establishes:** that a specific decision was made at a specific time, before the outcome was known, by a process that required independent agreement.
- **Establishes:** that the published history of decisions is complete, because an omitted decision would be a gap in an on-chain sequence.
- **Does not establish:** that the decisions were good. A verifiable record of being wrong is still a record of being wrong.
- **Does not establish:** what any individual made or lost. Two people acting on the same signal at the same moment can end up in very different places depending on size, venue, fees and timing, and the ledger deliberately holds none of those.

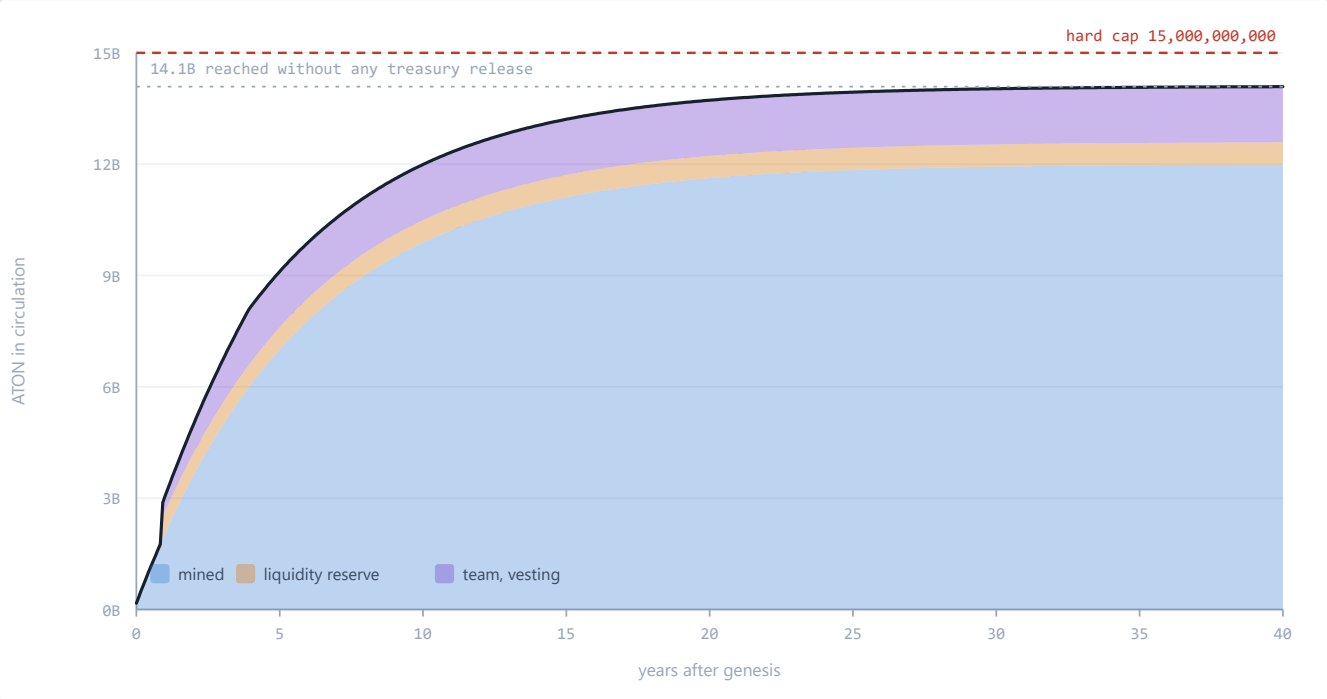


Figure 2 Circulating supply by source. Mined issuance dominates throughout. The dotted ceiling is where circulating supply settles without any treasury release, which is 14.1 billion of the 15 billion cap.

15 Receipt schema and verification

What a decision receipt contains, and the exact procedure a third party follows to check one.

Schema

```
DecisionReceipt
  asset             AssetId    the pair the decision concerns
  direction         u8         0 buy · 1 sell · 2 hold
  confidence        u16        fixed point, 0 to 10000
  breakdown
    quant_score     i16        signed, fixed point
    sentiment_score i16        signed, fixed point
    fusion_weight   u16        the weighting applied at decision time
  model_version     [u8;32]    identifier of the exact model and runtime
  input_commitment  Hash       hash of the input snapshot, not the snapshot
  mesh_job          Hash       the mesh job whose agreement produced this
  timestamp         u64        milliseconds, from the ordering layer
  publisher         Option     present only for marketplace-published strategies
```

There is no field for the user, the account, the size, the price or the outcome. Not omitted at write time: absent from the structure, so a future change of policy cannot begin populating one.

Verifying a published record

1. Read the receipts for the strategy or publisher in question directly from the chain. Do not read a number from an interface.
2. Check each receipt's `timestamp` against the ordering of the block that contains it. A receipt cannot claim a time earlier than the block that carries it.
3. Check each receipt's `mesh_job` resolves to a settled job whose agreement threshold was met. A receipt whose job did not reach agreement is not a valid receipt.
4. Check the sequence for gaps. The record is a sequence, and a missing decision is visible as a discontinuity rather than as an absence.
5. If the input snapshot is later published, hash it and compare against `input_commitment` to confirm the decision was made from the data claimed.

Steps one to four require nothing but a node and are available to anybody. Step five requires the snapshot to be published, which is at the discretion of whoever holds it, and is the only step that is not unilaterally verifiable.

> **WHAT A VERIFIED RECORD IS WORTH**
>
> It establishes that a decision was made at a time, by a process requiring independent agreement, before the outcome was known, and that the published history is complete. It establishes nothing whatever about whether the decisions were good. A complete and tamper-evident record of poor decisions is exactly as verifiable as a record of good ones, and this paper would rather say that than let verifiability be read as endorsement.

16 Trading engine

Two scoring engines feeding a fusion layer, executing non-custodially against the user's own exchange accounts, with every decision committed to the Signal Ledger.

> **WHAT THIS SECTION DOES AND DOES NOT CONTAIN**
>
> The trading engine's implementation is not open source and will not be. This section specifies its architecture, its interfaces, its custody model and what it commits on-chain, because all of that is verifiable by anybody using it and belongs in a technical paper. It does not contain model weights, training data, feature definitions or the inference path. That is a deliberate boundary and it is stated rather than left for a reader to discover by its absence.

Quant engine

Reads market structure across multiple timeframes and reduces it to a direction and a confidence. Its inputs are price and volume derived; it holds no view about news or sentiment. It is available on every subscription tier because a decision layer with only one engine is still a usable product.

Sentiment engine

A fine-tuned open-weight language model producing a structured sentiment and narrative score rather than free text. The base is from the Qwen family under Apache 2.0. Two properties drove that selection: strong multilingual reasoning, because sentiment that only reads English is sentiment that misses most of the market, and dense single-GPU deployability, which matches the mesh's actual hardware rather than requiring data-centre multi-GPU hosting.

Output is a structured score with a breakdown, not prose. A trading decision layer cannot consume an essay, and a system that produces one invites somebody to read it as advice.

> **ON THE BASE MODEL, PLAINLY**
>
> Apache 2.0 imposes no obligation to publicly attribute a base model, and the project takes no position beyond that in its marketing. But silence about provenance and a claim of building from scratch are different things. If asked directly, by a user, a journalist or a regulator, the answer is that the sentiment engine is built on an open-weight model from the Qwen family, fine-tuned on the mesh. Staying quiet is available; a false claim about provenance is not, and would be a misleading-advertising exposure rather than a licensing one.

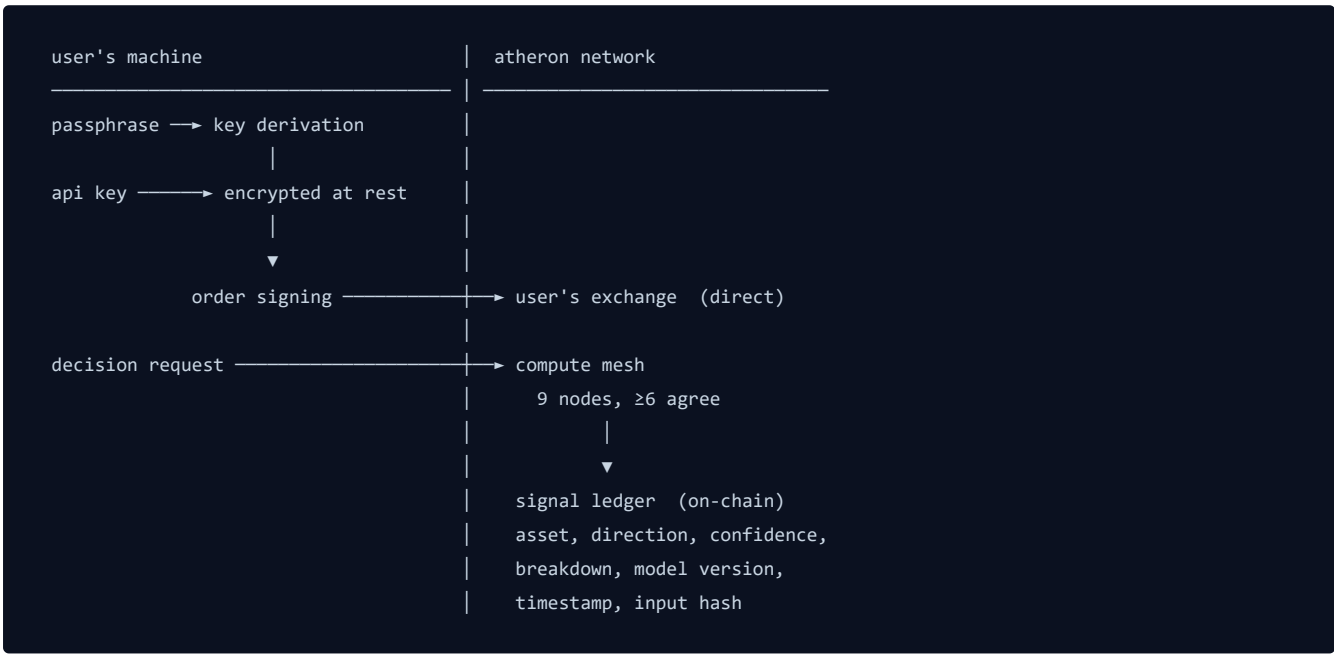
Fusion and policy

The fusion layer combines the two scores under weights the user sets, then applies a policy layer holding the user's risk configuration: position sizing, per-asset and total exposure caps, and a drawdown threshold. The policy layer is the last thing between a score and an order, and it can only reduce. It never increases a position beyond what the user's settings allow, and where the configured limits leave no room, the decision recorded is to do nothing.

Custody

Exchange API credentials are encrypted with a key derived from a passphrase the user holds, and written to local storage on the user's own device. They are not synchronised, not backed up to any infrastructure the project operates, and not present in any database it runs. Orders are signed on the user's machine and sent to the user's exchange account. No funds pass through anything the project operates at any point.

The client checks the permissions attached to a submitted key and **refuses to use a key carrying withdrawal permission**, rather than accepting it and relying on a promise never to exercise it. The test of whether a non-custodial claim is real is whether the company can recover a lost key. Here it cannot, and that is the evidence rather than a caveat attached to it.



17 Verifiable backtesting

Testing a rule set against a record that was committed before anybody knew the answer, which is a different exercise from testing against a vendor's own simulation.

A conventional backtest evaluates rules against data the vendor supplies, using a simulation the vendor wrote. Both halves are under the control of the party whose product is being evaluated. This is not usually fraud; it is that the degrees of freedom available to somebody producing a favourable curve are enormous and mostly invisible.

Backtesting against the Signal Ledger changes what is fixed. The decisions are already on-chain with timestamps that precede their outcomes, and they cannot be revised. A user varies their own policy layer, being sizing, caps and the drawdown threshold, and sees how the same immutable sequence of decisions would have been treated under those settings.

- **Fixed:** the decisions, their timing, their confidence and their breakdown.
- **Variable:** the user's own risk configuration.

- **Not modelled and not claimed:** fill prices, slippage, fees and venue behaviour, which are properties of the user's exchange and not of this system.

> **WHY NO PROFIT-AND-LOSS CURVE IS PRODUCED**
>
> Two people acting on the same decision at the same moment can end in very different places depending on their size, their venue, their fees and their timing. A single headline figure would necessarily be untrue for at least one of them, and probably for both. The backtest reports how a configuration would have treated the decision sequence, not what it would have earned, and no amount of user demand for the second number will produce one.

18 Decentralised exchange

A hybrid venue, spot only, launching same-VM and connecting across the two engines once the router's allowlist widens.

Trading model

A hybrid design: a central-limit order book implemented natively in WASM, and automated market maker pools on both VM sides. The order book gives price-time priority for participants who want it; the pools give continuous liquidity for assets and sizes where a book would be thin.

Rollout shape

At mainnet the DEX trades **same-VM only**. EVM-side pools trade EVM-native assets; the WASM-side order book and pools trade WASM-native assets including ATON. Both are fully live. They are not yet cross-connected.

Cross-VM pairs, which unify liquidity across both engines, go live when the cross-VM router's conservative allowlist is widened after its stability period. This is not a reduction of ambition. It is that ambition sequenced behind a dependency the project already decided to de-risk, and pretending the two were independent would be the dishonest version.

Scope

Spot only. No leverage, no perpetuals, no synthetics, and no route to any of them. The regulatory position for *decentralised* derivatives venues is unsettled in a way that spot is not, and 2023 enforcement precedent against DeFi platforms for unregistered derivatives activity still stands. Building the capability first and asking afterwards is how projects arrive somewhere they did not intend to be. Derivatives are revisited only after dedicated legal review shows the picture for decentralised venues has actually settled, which is not a calendar commitment.

Design notes

- **Oracle-secured pricing** from the mesh for slippage protection on both the book and the pool side, reusing infrastructure already being audited rather than adding a trust assumption.

- **Fair ordering** is a stretch goal rather than a mainnet requirement. Because Atheron controls its own base consensus, batch-auction or encrypted-mempool ordering is genuinely achievable here in a way it is not for a venue bolted onto somebody else's chain. It is also cutting-edge work industry-wide, with real research risk, and this paper declines to promise it.
- **Permissionless contracts, gated official front end.** The contracts stay open to anybody. The project's own interface applies geo-blocking and address screening, the same split several established venues use.
- **Fee sharing with ATON stakers**, not only with market makers and liquidity providers. Fee parameters are governable rather than fixed in this paper.

19 Developer platform

What a third party can build on this chain, and which of the two engines they should reach for.

Surfaces

SURFACE	AVAILABLE THROUGH	WHAT IT IS FOR
Standard Ethereum JSON-RPC	Atheron-EVM	Existing tooling, wallets, indexers and Solidity contracts with no changes
Native Rust SDK and CLI	Atheron-WASM	Contracts that use precompiles: mesh jobs, staking, lane statistics, the Signal Ledger
Compute Mesh job submission	WASM precompile, or the router from EVM	Submitting inference or oracle work and acting on an agreed result
Oracle requests	Same machinery as mesh jobs	Price and data feeds with no additional trust assumption
Public analytics API	Off-chain read API	Reading Signal Ledger and DEX data without running a node
Model deployment	Mesh, from mainnet	Deploying a fine-tuned model for the network to serve

Choosing an engine

Use the EVM when the value is in existing code, existing libraries or an existing audience of Solidity developers. Use WASM when the contract's purpose involves something only this chain has: scheduling inference, reading which lane produced what, staking into the mesh, or writing to and reading from the Signal Ledger. The cross-VM router means the choice is not permanent, but it is cheaper to start on the right side.

Contribution policy

The project's own repositories are closed to outside pull requests during development. Interfaces are moving week to week and several components are heading into independent audit; merging outside changes into that puts both the reviewer and the contributor in a poor position. This changes when the specification settles and the first audits return. It is stated here because a developer platform that quietly closes its own repositories is worse than one that says so.

20 Taking part

What each role requires in practice, so a reader can work out whether any of it applies to them before reading further.

ROLE	NEEDS	EARNS	NOTES
Lane A miner	ASIC hardware for a BLAKE3-based function	Half of each month's subsidy, shared by lane work	Conventional mining. No stake, no registration, no permission.
Lane B miner	A GPU with tensor cores and 512MB to 1GB of spare working memory beyond the model	The other half of the subsidy	Can be the same machine that serves compute. The scheduler decides between them continuously.
Compute Mesh node	A GPU, plus a stake bond of at least 1,000 ATON	Per-job fees, plus Lane B rewards when idle	Higher stake unlocks higher-value job tiers. Reputation accrues from accuracy, uptime and latency.
Full node	Storage and bandwidth for the pruned chain	Nothing directly	Required for anybody verifying independently rather than trusting an interface.
Archive node	Storage for unpruned history	Nothing directly	Expected to exist, not required for consensus.
Contract developer	The EVM or WASM toolchain	Whatever the contract earns	No permission needed. See Section 19.
Trading engine user	A subscription and an exchange account	Nothing from the protocol	Holds their own keys. The protocol is not involved in their trades except to record decisions.

Hardware, honestly

Lane B is designed so that a consumer card with tensor cores is a viable participant. The founding mesh is two such cards. That said, mining is a market: a viable participant is not the same as a profitable one, profitability depends on electricity price and on what everybody else brings, and this paper makes no claim about returns to any hardware.

The working set is deliberately sized past typical GPU L2 cache so that memory bandwidth binds rather than cache residency. A card with high compute and narrow memory will underperform its raw specification here, which is intended: the hardware profile being selected for is the inference profile, and inference is bandwidth-bound too.

21 Interoperability

What connects to other networks, and the significant thing that deliberately does not.

What exists

- **Ethereum tooling compatibility** at the JSON-RPC layer, so wallets, explorers, indexers and development frameworks work unchanged.
- **Registered chain IDs** for all three networks, so signed transactions cannot be replayed across them.
- **Exchange connectivity in the trading engine**, which is not a chain interoperability feature at all: the engine talks to centralised exchanges through the user's own API credentials, off-chain, and nothing about that touches the protocol.

What does not exist, on purpose

> **THERE IS NO BRIDGE, AND NONE IS PLANNED FOR MAINNET**
>
> A cross-chain bridge is the single most attacked category of infrastructure in this industry, and the losses have not been marginal. Building one is a decision to hold a large, permanently attractive target with a security model that is usually weaker than either chain it connects.
>
> Atheron launches without one. Assets native to Atheron are mined or issued on Atheron. This costs real liquidity access and real composability with other ecosystems, and that cost is the point: it is being paid deliberately rather than discovered later. If a bridge is ever added it will be a governed decision with its own audit programme, announced as its own thing, and it is not on the roadmap in Section 28.

22 Governance

Stake-weighted on-chain voting for treasury and mesh parameters, plus a disclosed, scope-limited emergency capability over the two novel components.

The governance module

A native WASM contract following the governor pattern: proposal, stake-weighted vote, timelock, execution. Its scope at version one is treasury disbursement and Compute Mesh parameter changes, meaning stake amounts, slashing percentages and redundancy factors. That scope is not incidental: the treasury is already committed to milestone or governance-gated release, so a governance mechanism is a dependency of the tokenomics rather than an addition to them.

The safety valve, described rather than hidden

Atheron-TMH and the cross-VM router are the two components this paper has repeatedly named as novel and unproven at scale. Deploying either with no ability to patch means that a critical bug found after launch can never be fixed. A founder-controlled multisig therefore retains a scoped emergency-patch capability over those two components.

PROPERTY	HOW IT IS CONSTRAINED
Scope	Atheron-TMH and the cross-VM router only. No power over the Compute Mesh, the DEX, the Signal Ledger, the governance module or any other contract.
Timelock	Every action passes through an on-chain delay before taking effect, so the network sees a pending action before it executes rather than afterwards.
Visibility	Every use, and every period of non-use, is publicly visible on-chain.
Sunset	Hands to full community governance once a defined decentralisation threshold is met: a minimum number of independent miners and validators, and a stability record for both components post-mainnet. The exact thresholds are a later governance decision and are not fixed here.

This is standard progressive decentralisation rather than something unique to this project, and it is described in every public document rather than mentioned in one. A reader who regards any admin capability as disqualifying now has the information to conclude that, which is the point of stating it.

Consensus-level changes beyond those two components stay with the core team until the network has enough independent validators that governance capture is not a realistic concern. Moving that boundary is itself a later governance decision.

23 Tokenomics

A fixed cap of fifteen billion ATON, no sale of any kind at any point, and eighty per cent of it issued by mining on a schedule fixed at genesis.

HARD CAP	ISSUED BY MINING	PUBLIC SALE	EMISSION LIFETIME
15,000,000,000	**12,000,000,000**	**None**	**~132 years**
ATON, fixed at genesis	80%, across two lanes	no private round either	33 halving-equivalents

Allocation, in units

Figure 3 Allocation of the hard cap, in ATON. Every figure is exact rather than rounded, and the notes state the release condition for each tranche.

ALLOCATION	SHARE	ATON	RELEASE
Mining	80%	12,000,000,000	Issued by the emission curve over roughly 132 years. Divided equally between Lane A and Lane B, so 6,000,000,000 to each.
Team	10%	1,500,000,000	Twelve-month cliff, then linear vesting to month 48. Nothing is released before month 12.
Treasury and ecosystem	6%	900,000,000	Multisig held, released against milestones or by governance vote. No fixed schedule, which is why it does not appear as a curve in Figure 2.
Initial liquidity	4%	600,000,000	Locked and vested for market making at listing. Not sold to the public at any point.
Total	**100%**	**15,000,000,000**	

Emission

The subsidy decays by **1.43% per month**, smoothly, rather than halving in discrete steps. Over 48 months that compounds to a factor of **0.500897**, which is a halving to within 0.18%. Over roughly 132 years it produces about 33 halving-equivalents. The cadence and the total lifetime therefore match Bitcoin's closely, and can be explained to anybody in those terms, while avoiding a discrete cliff.

The smoothness is not cosmetic. A literal 50% cliff applied across two independent mining lanes risks destabilising whichever lane is more economically marginal at that moment. A monthly decay reaches the same place without ever presenting that edge.

QUANTITY	VALUE
Month one subsidy	171,600,000 ATON
Month twelve subsidy	146,456,822 ATON

QUANTITY	VALUE
Month forty-eight subsidy	87,200,875 ATON
Monthly decay factor	0.9857
Compounded over 48 months	0.500897, against 0.5 for an exact halving
Error against an exact halving	0.179%
Halving-equivalents to exhaustion	≈ 33
Emission lifetime	≈ 132 years

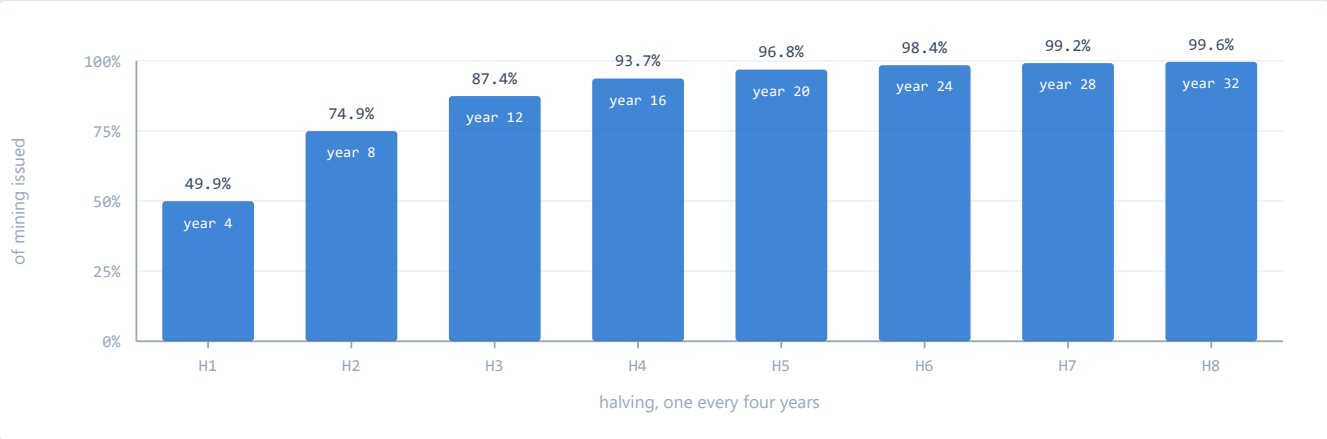


Figure 4 Share of the mining allocation issued by each halving-equivalent. Half of all mining issuance occurs in the first four years, which is the same front-loaded shape Bitcoin has and is stated here rather than left to be discovered.

Vesting and unlocks

Figure 5 Team vesting. Nothing is released before month twelve. From the cliff the schedule is linear to month forty-eight.

The treasury is deliberately absent from the supply curve in Figure 2. It has no schedule: it releases against milestones or by governance vote, and drawing a curve for it would imply a timetable that does not exist. The

consequence is visible in that figure as the gap between where circulating supply settles, about 14,100,000,000, and the 15,000,000,000 cap.

Where fees go

FEE	PAID BY	PAID TO
Transaction gas	Anybody transacting	Miners, with a portion to stakers under the fee-sharing parameter
Compute Mesh job fees	The trading engine, third-party dApps, oracle consumers	The node operators that executed and agreed on the job
DEX trading fees	Traders	Market makers and liquidity providers, with a share to ATON stakers
Subscription revenue	Trading engine subscribers	The company. This is off-chain revenue and is not a token flow.
Marketplace fee	Strategy publishers who charge	The company takes 15% of what a publisher earns. Nothing is taken where a strategy is published free.

> **TWO THINGS THE TOKEN DELIBERATELY DOES NOT DO**
>
> ATON is not a claim on the company's revenue and confers no entitlement to any share of it. Subscription income is ordinary corporate revenue and is not routed to holders. Separately, the token confers no governance power over the company, only over the on-chain parameters described in Section 13.
>
> This is a design choice with a reason. A token that pays holders a share of a company's profits is, in most jurisdictions that have looked at the question, something other than a utility token. The project would rather have a token that clearly does one job than one that invites the argument.

24 Network economics

How the pieces pay for each other, and where the assumptions are.

The three income streams a node can hold

1. **Lane B block rewards.** Half of each month's subsidy, divided among GPU miners by share of that lane's work.
2. **Compute Mesh job fees.** Paid per job by the trading engine, by third-party dApps and by oracle consumers, to the nodes that executed and agreed.
3. **Reputation-weighted priority**, which is not itself income but determines access to the higher-value job tiers.

A Lane A miner holds only the first. That asymmetry is intentional: the GPU lane is the one whose hardware has a second use, and the design's central claim is about that second use existing.

The scheduler's decision

A node's local scheduler compares, continuously, the expected value of mining the next interval against the expected value of taking available compute work, and allocates the GPU accordingly against operator-set thresholds. Three consequences follow, and they are worth stating because they are the mechanism rather than the marketing:

- **Mining sets a floor under compute pricing.** A node will not accept inference work priced below what mining would return, so compute fees cannot fall below mining economics without capacity leaving for the lane.
- **Compute demand raises the cost of attacking Lane B.** Capacity absorbed by paid inference is capacity not available to rent for an attack, and the attacker is bidding against a real buyer rather than against a subsidy alone.
- **The two markets are coupled, in both directions.** A collapse in compute demand pushes capacity back to mining and raises lane difficulty. A collapse in token price pushes capacity toward whichever is still paying. This coupling is the design's strength and is also a transmission path for shocks, which Section 31 lists as an economic risk rather than omitting.

Where the fees actually go

Section 23 gives the fee table. The point worth repeating here is what is *absent*: no fee stream is routed to token holders as a share of company revenue. Subscription income is ordinary corporate revenue. Staker fee-sharing applies to on-chain fees, being gas and DEX fees, which are network fees rather than company earnings. The distinction is deliberate and Section 23 explains why.

An honest gap

> **NOBODY HAS MEASURED THE COST OF A MESH JOB**
>
> Every statement in this section about relative pricing is structural rather than quantitative, because the cost per job on real hardware has not been measured and cannot be until the mesh runs. Subscription pricing, mesh fee floors and the point at which compute outbids mining all depend on that measurement.
>
> This is on the open-parameter list. If the measurement shows the published subscription levels are not sustainable, they move before launch rather than quietly afterwards.

25 Related work

What this borrows, what it does differently, and where somebody else got there first.

AREA	PRIOR WORK	WHAT ATHERON TAKES, AND WHAT IT CHANGES
DAG consensus	PHANTOM and GhostDAG; Kaspa as the production reference	Takes the ordering protocol and adapts a Rust reference rather than writing one. Changes the state model from UTXO to account-based, which is a core rewrite rather than a setting.

AREA	PRIOR WORK	WHAT ATHERON TAKES, AND WHAT IT CHANGES
Useful proof of work	Primecoin, Folding-style schemes, and a long line of attempts	Does not claim the work itself is useful. Keeps conventional hash-finding and chooses the arithmetic so the winning hardware profile matches the profile the network needs elsewhere. A narrower claim, and a defensible one.
Memory-hard proof of work	Ethash, Cuckoo Cycle, RandomX	Shares the goal of binding to a hardware profile. Differs in choosing tensor arithmetic over a large working set, specifically to match inference rather than to resist ASICs in general.
Decentralised inference	Several networks distributing ML work across independent hardware	Shares the structure. Differs in having a founding workload and a mining subsidy underwriting node economics before third-party demand exists, which is the failure point those networks usually hit.
Redundant execution and slashing	Standard in oracle and off-chain compute designs	Takes the pattern directly. Applies the same machinery to inference and to oracle work so both are secured by one audited mechanism.
Dual virtual machines	Several chains run an EVM beside a native engine	Takes the pattern. Ships the cross-VM router at mainnet behind a conservative allowlist rather than deferring it, and names it as the highest-risk component for exactly that reason.
On-chain provenance for model output	Verifiable-inference and zkML research	Does not claim cryptographic proof of correct inference. Claims something weaker and available today: a majority-agreed, timestamped commitment of a decision made before its outcome was known.

> **WHAT THIS PAPER DOES NOT CLAIM NOVELTY FOR**
>
> The consensus protocol, the EVM, the WASM runtime, the account-abstraction pattern and the redundant-execution model are all existing work, used as such and credited here. The genuinely new components are Atheron-TMH and the specific coupling of a mining lane to an inference market. Those two are also the ones this paper spends the most words warning about, which is not a coincidence.

26 Security and audit

Which components must be independently audited before mainnet, which are continuing commitments rather than one-time checks, and what is being relied on that has never been attacked.

Hard gates on the mainnet date

Each of the following must have completed independent audit before mainnet. These are not tasks that can follow the launch, and the mainnet date moves if one of them is not finished.

COMPONENT	WHY IT GATES	PARTICULAR CONCERN
Atheron-TMH	Novel proof-of-work function with no adversarial history	Shortcut attacks that skip the memory-bound work; cross-vendor non-determinism in reduced-precision arithmetic; whether an optimised implementation narrows the intended hardware profile
Cross-VM router	Ships at mainnet by decision, and is where comparable chains have found their worst bugs	Call and return-data translation between two ABIs, gas accounting across the boundary, and failure semantics when one side reverts
Atheron-WASM engine	The Compute Mesh runs on it as contracts rather than as protocol rules	A WASM engine bug is a Compute Mesh bug, so this audit carries the mesh's weight as well as its own
Compute Mesh contracts	Hold stake, assign work and execute slashing	Job assignment manipulation, slashing conditions triggering incorrectly, stake accounting
Signal Ledger contract	Writes the record the product's central claim depends on	Whether a receipt can be written without the majority agreement it purports to represent
Governance module and timelock	Controls the treasury and the emergency capability	Proposal and execution paths, timelock bypass, quorum manipulation
CLOB matching engine	Handles order matching and settlement	Ordering fairness, settlement correctness, economic attacks on matching
AMM pool contracts, both VMs	Standard but high value	Reentrancy, price manipulation including flash-loan-style attacks, on both sides
Referral rewards registry	Touches value-adjacent state	Eligibility accounting and redemption paths

Continuing commitments, not one-time checks

- **Cross-VM call fuzzing** through a public testnet period long enough to be adversarial, rather than each engine tested alone.
- **A standing bug bounty**, separate from the pre-launch per-component bounties, so that the incentive to report rather than exploit persists after launch.
- **Red-teaming of the fraud and sybil scoring** as an ongoing commitment. Reward-farming technique evolves, and a system audited once is audited against last year's attacks.
- **Re-audit on change.** Any change to an audited component returns it to the queue. Audit is a property of a specific commit, not of a component name.

WHAT IS BEING RELIED ON THAT HAS NEVER BEEN ATTACKED

Two things, and this paper has now said both several times because a reader who takes one thing from this section should take this. Atheron-TMH is a new proof-of-work function. The cross-VM router is a new translation layer shipping at mainnet rather than after it. Audit reduces the probability of an unknown flaw in each; it does not eliminate it, and no amount of process substitutes for adversarial exposure over time.

The scoped, time-locked emergency capability described in Section 22 exists precisely because these two components are what they are. That capability is itself a risk, and the project would rather hold a disclosed and constrained one than deploy an unpatchable component and hope.

Operational security posture

- No component of the trading engine's model or inference path is published, and no key material of any kind is held by the project on a user's behalf.
- Security reports are handled through a published disclosure process and never through public channels. A vulnerability posted in a chat room is a vulnerability disclosed to whoever intends to use it.
- The decentralisation statement in Section 03 is a security property, not marketing: the chain, the mesh, the DEX and the Signal Ledger depend on no infrastructure the company operates, so the company failing does not take them down.

27 Threat model

Who might attack this, what they would attack, what stops them, and where the honest answer is that nothing yet does.

Assumptions

- An adversary may control a substantial fraction of one lane's hashrate, may operate many mesh nodes, may submit arbitrary transactions and jobs, and may observe all network traffic.
- An adversary cannot break BLAKE3, cannot forge signatures, and cannot compel an honest node to run modified software.
- A user's own machine is outside the protocol's control. A compromised user device is a compromised user, and no protocol property recovers from it.

Attacks on consensus

ATTACK	WHAT IT WOULD ACHIEVE	WHAT STANDS AGAINST IT
Majority hashrate on one lane	Reorganisation and censorship of that lane's contribution	Each lane produces half the blocks, so controlling one lane is not controlling the network. This is the structural argument for two lanes, and it is why a lane collapsing to irrelevance is treated as a security event rather than an economic curiosity.

ATTACK	WHAT IT WOULD ACHIEVE	WHAT STANDS AGAINST IT
Majority across both lanes	Full reorganisation and censorship	Nothing structural. This is the standard proof-of-work assumption and it applies here as everywhere. It requires simultaneously dominating two different hardware markets, which is harder than dominating one but is not a proof.
A shortcut in Atheron-TMH	Cheap Lane B hashrate, then the row above	**Unresolved by design alone.** Independent cryptographic audit and an adversarial testnet, plus the scoped emergency capability to replace the algorithm. This is the single largest technical risk in the document.
Cross-vendor numerical divergence in Lane B	A network partition along hardware lines	A named audit item. The function must be bit-exact across the hardware that mines it, and a tolerance is not acceptable in a consensus function.
Withholding blocks to game the anticone rule	Improved relative reward, or ordering influence	GhostDAG's blue-set selection penalises poorly connected blocks, and k bounds the tolerated parallelism. The effective bound is a measured property, which is why k is a load-bearing testnet deliverable rather than a preference.
Timestamp manipulation	Difficulty distortion	Permitted drift bounds and per-lane retargeting over a long window.

Attacks on the Compute Mesh

ATTACK	WHAT IT WOULD ACHIEVE	WHAT STANDS AGAINST IT
Sybil nodes to capture a job's quorum	Forcing an agreed but wrong result	Stake bond per node, so identities cost money rather than being free. Assignment weights reputation and load as well as stake, so buying a quorum requires buying reputation, which takes time and correct answers.
Collusion among assigned nodes	The same, more cheaply	Nodes do not communicate during execution and return only hashes. Collusion requires coordination arranged before assignment, against an assignment the colluders do not control.
Lazy nodes copying a peer's answer	Payment without doing the work	Isolation during execution is what prevents this, and it is why the design does not gossip intermediate results. It is also why determinism matters: without it, honest nodes disagree and the signal that distinguishes lazy from honest is lost.
Griefing by returning garbage	Denying settlement, wasting fees	2% slashing per disagreement makes it costly, and reputation decay makes it self-limiting.
Submitting jobs designed to be non-deterministic	Forcing splits, then slashing honest nodes	A split with no majority slashes nobody, precisely so that a malicious specification cannot be used as a weapon against honest nodes. The specification is flagged instead.
Exhausting the queue with cheap jobs	Denial of service against real work	Fees escrowed at submission rather than collected at settlement, so the attack costs money up front.

Attacks on the Signal Ledger

ATTACK	WHAT IT WOULD ACHIEVE	WHAT STANDS AGAINST IT
Writing a receipt without a real mesh agreement	A fabricated track record	A receipt carries the `mesh_job` it came from, and verification resolves that job and checks its threshold was met. This is the specific thing a verifier should check and the reason the field exists.
Backdating a decision	Claiming foresight	A receipt cannot predate the block carrying it, and block ordering comes from consensus rather than from the writer.
Publishing selectively	A flattering subset	Receipts form a sequence, so an omission is a discontinuity rather than an absence. This is the property that makes the record's completeness checkable and not merely its contents.
Deanonymising a user from receipts	Linking decisions to a person	No field identifies a user, an account, a size or a price. The risk that remains is correlation against public exchange data by somebody who already suspects a link, which the protocol cannot eliminate and does not claim to.

Attacks on users

These are outside the protocol and are the ones most likely to actually cost somebody money, which is why they appear in a technical paper rather than only in support documentation.

- **Impersonation and phishing.** The most common and most effective attack on any crypto community. The project's standing rule is that no member of its team sends a first direct message to anybody, stated everywhere so that a message claiming otherwise is self-refuting.
- **A compromised user device.** Keys live on the user's machine, so malware there is inside the boundary the custody model describes. This is the cost of non-custodial design and it is not recoverable by anything on the protocol side.
- **A key with withdrawal permission.** The client refuses such a key rather than accepting it and promising restraint. That refusal is a product decision doing security work.
- **Exchange failure.** The user's venue holds their funds. Non-custodial design removes the project as a counterparty and does not remove the exchange.

> **WHERE THIS MODEL IS WEAKEST**
>
> Every row above that says something structural stands against an attack is describing a mechanism that has not yet been attacked in production. The two components carrying the most weight, Atheron-TMH and the cross-VM router, are also the two with no adversarial history at all.
>
> A threat model written before a network exists is a statement of what the designers thought of. Its value is that it can be checked against, and added to, by people who think of something else. Findings should go through the published security disclosure process and never through a public channel.

28 Delivery

Eight phases. Dates are expressed as months from the start of the build, because a calendar date implies a confidence nobody has.

PHASE	WINDOW	WHAT HAS TO BE TRUE TO CALL IT DONE
0 · Foundation	Months 0 to 1	Legal entity, counsel engaged, repositories and CI gates, public site and build tracker, chain IDs reserved, community infrastructure live, devnet mesh bootstrapped on the founding GPUs
1 · Core protocol	Months 1 to 4	GhostDAG node with account-based state, both mining lanes implemented, internal devnet running, consensus parameters measured rather than assumed, invite-only testnet alpha
2 · Execution layer	Months 2 to 6, overlapping	Both VMs, the cross-VM router behind its allowlist, gas abstraction, public testnet beta with contracts live
3 · Mesh, governance, oracle	Months 4 to 7, overlapping	Mesh contracts on WASM, staking and slashing live, oracle service answering, governance module and timelock deployed
4 · Mainnet	Months 7 to 9	Every hard-gate audit closed, adversarial testnet period completed, genesis parameters locked and published
5 · Trading engine v1	Months 5 to 9, overlapping	Quant engine, non-custodial connector and Signal Ledger in closed beta
6 · Full launch	Months 7 to 12, overlapping	Sentiment engine, fusion, desktop client, marketplace, DEX
7 · Scale	Month 12 onward	Third-party model marketplace, larger model variants as mesh capacity allows, progressive handover of the safety valve

> **THE TRACKER, NOT THIS TABLE, IS THE CURRENT POSITION**
>
> This schedule is the plan as it stands at the version on the cover. The project publishes a build tracker computed from the repositories themselves, and a changelog entry for every merged change. Where this table and the tracker disagree, the tracker is right, because it is derived from the work rather than written about it.

29 The testnet programme

What each network stage is for, and specifically which open parameters it is expected to close.

STAGE	ACCESS	WHAT IT EXISTS TO DETERMINE
Devnet	Internal, founding nodes only	That the node runs, that both lanes produce blocks, that the mesh assigns and settles a job at all. Correctness before performance.

STAGE	ACCESS	WHAT IT EXISTS TO DETERMINE
Testnet Alpha	Invite only, Deep Freeze and Absolute Zero backers	Anticone constraint under real propagation. Achievable block rate. Lane balance behaviour with more than one participant per lane. First adversarial attention on Atheron-TMH.
Testnet Beta	Public, contracts live	Both VMs under load. Cross-VM router fuzzing against the allowlist. Mesh determinism across heterogeneous hardware, which is the item Section 12 flags as hardest. Storage growth against the pruning window.
Mainnet	Public	Nothing. By this point every parameter on the open list is closed and every hard-gate audit is complete, or the date moves.

> **WHY TESTNET ACCESS IS STAGED RATHER THAN OPEN FROM THE START**
>
> Atheron-TMH needs adversarial attention, and adversarial attention is most useful when the people giving it can be reached and can be asked what they did. A small invited group first, then an open network, gets more usable findings than opening to everybody on day one and reading crash reports.
>
> The alpha gate is stated on the public site and is not a marketing device: it is the group who committed to the project before there was anything to use, and the cohort size is small enough to work with directly.

What a failed measurement means

If testnet shows the target block rate is not safely achievable at any anticone value, the response is to lower the genesis target or to delay mainnet. It is not to launch on the target and hope, and it is not to redefine safety. That is stated here so that a reader can hold the project to it later, which is the only reason to write such a sentence down.

30 Open parameters

Everything in this paper that is not yet fixed, in one table, with what settles it. Nothing on this list is resolvable by preference.

ITEM	SECTION	CURRENT POSITION	SETTLED BY
Anticone constraint, k	Section 05	Exploration begins at 24, likely higher	Phase 1 testnet simulation under real propagation conditions
Genesis block rate	Section 05	Target 4 to 10 per second	Phase 1 testnet measurement. If unsafe, the response is to delay mainnet or lower the target, not to launch on the target anyway
Pruning window	Section 06	Mechanism fixed, window open	Testnet tuning against real storage growth
Lane-balance dampener thresholds	Section 08	65% over a trailing 2,016 blocks	Testnet simulation of two-lane economics

ITEM	SECTION	CURRENT POSITION	SETTLED BY
Difficulty retarget interval	Section 08	Approximately 2,880 blocks per lane	Testnet tuning
Cross-VM allowlist contents	Section 09	Conservative set at mainnet	Audit findings, then a stability period before widening
Compute Mesh job pricing	Section 11	Fee market, floor set by mining economics	Real cost-per-job measurement, which requires the mesh to exist
DEX fee parameters	Section 18	Governable, not fixed here	Governance vote post-launch
Safety valve sunset thresholds	Section 22	Minimum independent miners and a stability record	A later governance decision
Subscription pricing	—	Researched levels published on the site	Cost-per-job measurement. If it moves, it moves before launch rather than quietly afterwards

31 Risks

The failure modes a reader should weigh. This section is not a formality and it is not exhaustive.

Technical

- **Atheron-TMH is unproven.** If a shortcut attack exists, the GPU lane's security assumption fails and the algorithm has to be replaced under the emergency capability, which is disruptive even when it works.
- **Reduced-precision non-determinism.** FP16 and BF16 behaviour differs across vendors and driver versions. If the function is not bit-exact across the hardware that mines it, the network forks along hardware lines. This is a specific and named audit item.
- **The cross-VM router ships at mainnet.** It is the component most likely to carry a serious bug, and it is being launched rather than deferred.
- **The target block rate may not be achievable safely.** The stated response is to delay or lower the target. That response has a cost and it is the honest one.
- **Account-based state on a GhostDAG client is a genuine deviation** from the reference implementation, and the core state machine has to be built rather than adapted.

Economic

- **Emission is front-loaded.** Half the mining allocation is issued in four years. If network value has not grown by the time the subsidy has fallen substantially, security spend falls with it. This is the same shape Bitcoin has and the same risk it carries.
- **Two lanes can become one.** If one lane's economics collapse, the network has the security of a single-lane chain and the complexity of two. The dampener is intended to prevent it and has not been tested against real

markets.

- **Mesh demand may not arrive.** The trading engine and mining rewards underwrite node economics at the start. If third-party demand does not follow, the mesh is an expensive way to run one product's inference.
- **The treasury has no schedule.** That is deliberate, and it means the market cannot price a known unlock in advance. Governance discretion is a governance risk.

Regulatory

- **ATON's classification is untested** in every jurisdiction that matters. The design decisions in this paper are intended to reduce that exposure and are not a determination by anybody with the authority to make one.
- **The strategy marketplace may constitute regulated activity.** Listing strategies for others to follow, and ordering them, resembles activity that several regimes treat as investment advice or as managing money. It needs dedicated counsel review in both Canada and the United States before it opens, and that review has not concluded.
- **Operating exchange infrastructure raises its own questions**, distinct from token classification: money-transmitter characterisation in the United States and dealer or marketplace registration in Canada.
- **Jurisdictional access may have to be restricted**, and the official front ends apply geo-blocking and address screening accordingly.

Execution

- **The founding hardware is two consumer graphics cards.** The scale-up path is real and is dependent on community capacity arriving.
- **The schedule is estimates against a plan** and plans in this field slip. The tracker exists so that slippage is visible rather than announced.
- **Several components are being built simultaneously** by a small team, and the mainnet gate requires all of the hard-gate audits to close.

32 Glossary

Anticone	The set of blocks in a DAG that are neither ancestors nor descendants of a given block. Its permitted size, k, is the parameter that bounds how much parallelism the ordering rule tolerates.
Blue set	The well-connected subset of the DAG that GhostDAG identifies and orders. Blocks outside it still contribute transactions.
Cross-VM router	The translation layer letting a contract on one virtual machine call a contract on the other.
Compute Mesh	The network of staked nodes executing inference and oracle jobs redundantly, with majority agreement.
Decision receipt	The structured record the trading engine commits to the Signal Ledger when the fusion layer produces an actionable decision.
GhostDAG	The consensus protocol, from the PHANTOM family, that orders a directed acyclic graph of blocks rather than a single chain.
Lane A, Atheron-B3	The BLAKE3-based, ASIC-favourable mining lane.

Lane B, Atheron-TMH	The tensor-memory-hard, GPU-favourable mining lane whose arithmetic matches transformer inference.
Non-custodial	The property that the operator holds neither funds nor keys, evidenced here by the operator being unable to recover a lost key.
Precompile	A native function exposed to contracts, used on the WASM side to reach chain-specific state directly.
Pruning point	A checkpoint past which historical DAG data may be discarded by a node.
Redundant execution	Running the same job on N nodes and requiring a threshold of matching results before accepting it.
Signal Ledger	The on-chain contract holding decision receipts.
Slashing	Confiscation of a portion of a node's stake for returning a result that disagrees with the majority.
Virtual chain	The totally ordered sequence GhostDAG derives from the DAG, against which state transitions are applied.

33 Document control

This paper is revised as the project moves. Every revision is recorded in the project's public changelog with a date. The version in force is stated on the cover. Where this document and the public build tracker disagree about status, the tracker is correct.

A plain-language companion covering the same project without the specification detail is published alongside this paper. Where the two disagree on a fact, this one is the reference, and the disagreement is a defect to be reported.

34 Appendix A — Emission schedule

Year by year, computed from the 1.43% monthly decay. Subsidy is the amount issued in the final month of that year. Circulating includes mined issuance, vested team tokens and the liquidity reserve, and excludes the treasury, which has no schedule.

YEAR	SUBSIDY THAT MONTH	MINED TO DATE	OF MINING	CIRCULATING	OF CAP
1	146,456,822	1,904,721,028	15.87%	2,879,721,028	19.20%
2	123,210,206	3,507,111,873	29.23%	4,857,111,873	32.38%
3	103,653,450	4,855,160,423	40.46%	6,580,160,423	43.87%
4	87,200,875	5,989,237,605	49.91%	8,089,237,605	53.93%
5	73,359,763	6,943,306,399	57.86%	9,043,306,399	60.29%
6	61,715,606	7,745,938,952	64.55%	9,845,938,952	65.64%

YEAR	SUBSIDY THAT MONTH	MINED TO DATE	OF MINING	CIRCULATING	OF CAP
7	51,919,688	8,421,172,246	70.18%	10,521,172,246	70.14%
8	43,678,645	8,989,227,953	74.91%	11,089,227,953	73.93%
9	36,745,675	9,467,118,022	78.89%	11,567,118,022	77.11%
10	30,913,154	9,869,154,152	82.24%	11,969,154,152	79.79%
11	26,006,409	10,207,376,393	85.06%	12,307,376,393	82.05%
12	21,878,496	10,491,913,717	87.43%	12,591,913,717	83.95%
13	18,405,794	10,731,287,355	89.43%	12,831,287,355	85.54%
14	15,484,302	10,932,665,992	91.11%	13,032,665,992	86.88%
15	13,026,529	11,102,080,453	92.52%	13,202,080,453	88.01%
16	10,958,870	11,244,604,307	93.71%	13,344,604,307	88.96%
17	9,219,404	11,364,505,812	94.70%	13,464,505,812	89.76%
18	7,756,038	11,465,375,740	95.54%	13,565,375,740	90.44%
19	6,524,947	11,550,234,913	96.25%	13,650,234,913	91.00%
20	5,489,264	11,621,624,664	96.85%	13,721,624,664	91.48%
24	2,749,555	11,810,472,980	98.42%	13,910,472,980	92.74%
28	1,377,243	11,905,066,510	99.21%	14,005,066,510	93.37%
32	689,857	11,952,448,112	99.60%	14,052,448,112	93.68%
40	173,084	11,988,069,342	99.90%	14,088,069,342	93.92%
50	30,735	11,997,881,467	99.98%	14,097,881,467	93.99%
60	5,458	11,999,623,811	100.00%	14,099,623,811	94.00%
80	172	11,999,988,138	100.00%	14,099,988,138	94.00%
100	5	11,999,999,626	100.00%	14,099,999,626	94.00%
132	0	11,999,999,999	100.00%	14,099,999,999	94.00%

READING THIS TABLE

Half of all mining issuance has occurred by year four and three quarters by year eight. Circulating supply settles at about 14.1 billion rather than at the 15 billion cap, because the 900,000,000 ATON treasury releases only against milestones or by governance vote and is therefore not part of any schedule.

Every figure here is produced by the same function that draws Figures 1, 2 and 4. There is no separate spreadsheet, so the table and the charts cannot disagree.

35 Appendix B — Parameter reference

Every numeric parameter in this document in one place, with its status.

PARAMETER	VALUE	STATUS
Max supply	15,000,000,000 ATON	Fixed at genesis
Mining allocation	12,000,000,000 ATON, 80%	Fixed
Team allocation	1,500,000,000 ATON, 10%	Fixed
Treasury allocation	900,000,000 ATON, 6%	Fixed
Liquidity allocation	600,000,000 ATON, 4%	Fixed
Monthly subsidy decay	1.43%	Fixed
Month one subsidy	171,600,000 ATON	Derived
Halving equivalent	48 months, factor 0.500897	Derived
Emission lifetime	≈132 years, ≈33 halvings	Derived
Team cliff	12 months	Fixed
Team vest end	Month 48, linear from the cliff	Fixed
Lane reward split	50% Lane A, 50% Lane B	Fixed
Consensus family	GhostDAG / PHANTOM	Fixed
State model	Account-based	Fixed
Genesis block rate	Target 4 to 10 per second	**Open** — testnet
Anticone constraint k	From 24, likely higher	**Open** — testnet
Pruning window	—	**Open** — testnet
Retarget interval	≈2,880 blocks per lane	**Open** — testnet

PARAMETER	VALUE	STATUS
Lane dampener threshold	65% over 2,016 blocks	**Open** — testnet
Mesh minimum stake	1,000 ATON	Fixed
Mesh default redundancy	N=5, threshold 3	Fixed, governable
Trade-triggering redundancy	N=9, threshold 6	Fixed, governable
Mesh slashing	2% of stake	Fixed, governable
Mesh job pricing	Fee market	**Open** — needs measurement
Marketplace take rate	15% of publisher earnings	Fixed
DEX fee parameters	—	**Open** — governance
Safety valve sunset	—	**Open** — governance