MetaFlux: A Layer-1 Derivatives Exchange

MetaFlux Foundation

hello@mtf.exchange · mtf.exchange

Whitepaper v1.0 · 2026

Contents

1. Abstract

MetaFlux is an independent Layer 1 blockchain built for a single purpose: operating a derivatives exchange. A central limit order book and clearinghouse execute deterministically inside Byzantine-fault-tolerant consensus; an inline EVM shares each block, so smart contracts and exchange actions settle together with immediate finality; MetaFlux's own bridge, MetaBridge, moves assets across chains under a two-thirds validator co-signature — the chain's exact trust assumption, with no administrative key; and a single gateway serves the exchange wire protocol alongside standard EVM JSON-RPC. The native token MTF secures the network through proof of stake, and protocol revenue funds an open-market buyback whose purchased tokens are locked permanently in a keyless address.

The design pursues five properties that general-purpose chains cannot deliver together: immediate and absolute finality, so liquidations, funding, and bridge releases never model a reorg; deterministic execution, enforced mechanically down to the prohibition of floating point in the consensus path; cross-asset capital efficiency through scenario-based portfolio margin; resistance to ordering-based value extraction through batch auctions, private quotation, and encrypted order intents; and permissionless market listing through burned-bid auctions. Each mechanism is specified below with its governing parameters.

2. Introduction

Existing on-chain derivatives venues compete primarily on price: marginal reductions in taker fees, volume subsidies funded by token emissions, and replicated liquidity programs. Competition of this kind converges to zero margin and rewards whichever operator can subsidize losses longest. MetaFlux is designed on a different premise: professional trading flow is won on structural capability rather than on fees.

The properties that determine venue selection for a professional desk are structural. Whether a hedged portfolio margins as one exposure or as several, whether a large order can execute without revealing itself to the market, and whether an on-chain program can read canonical market state at native cost are all set by protocol design, not by a fee schedule. The design effort therefore concentrates on three axes:

Axis 1 / Market structure
A deterministic price-time order book extended with matching modes that resist ordering-based extraction: frequent batch auctions where fairness should beat latency, request-for-quote for size, and threshold-encrypted order intents so inclusion is fixed before contents are readable.

Axis 2 / Risk & margin
A scenario-based cross-asset portfolio-margin engine, dynamic risk parameters tuned from realized volatility, a graduated liquidation ladder that bounds cascades by construction, and auto-deleveraging formulated as an online-learning problem with a provable regret bound.

Axis 3 / Execution
A parallel EVM executing under Block-STM so contract calls do not queue behind unrelated transactions, and derivatives-aware precompiles that expose the clearinghouse's own margin, pricing, and depth primitives to Solidity at native cost.

Beneath all three sits a deliberately conservative safety kernel. Consensus follows formally studied protocols rather than novel ones, and every MetaFlux-specific mechanism resides strictly above the commit rule — in admission, proposal validity, or block-boundary effects — so the chain inherits well-understood safety properties while all differentiation occurs in market structure, risk management, and execution.

3. Consensus: MetaFluxBFT

MetaFlux is secured by MetaFluxBFT, a Byzantine-fault-tolerant proof-of-stake engine in the HotStuff / Jolteon family of pipelined BFT protocols [1, 2] — the research line that also produced DiemBFT [3]. It is leader-based and partially synchronous, with a pipelined two-chain commit rule. The engine is a MetaFlux-native implementation built on those published foundations rather than a fork, in order to satisfy the determinism requirements imposed by the rest of the system. The choice is deliberate on both sides: an exchange whose clearinghouse lives inside consensus cannot afford a novel commit rule, and it equally cannot afford a general-purpose engine whose execution semantics it does not control end to end.

Element	MetaFluxBFT
Commit rule	Two-chain, pipelined. Finality is immediate and absolute: zero confirmations, no probabilistic settlement, no reorg window.
Quorum	At least two-thirds of total staked voting power; voting is stake-weighted, and the protocol tolerates up to one-third Byzantine stake.
Validator set	Fixed within an epoch, re-derived from stake at epoch boundaries; the leader role rotates across the set each round.
Signatures	secp256k1 over an EIP-712 envelope in the MetaFlux signing domain — the same key family that authorizes every user action and every bridge release.
State commitment	A per-block `app_hash` — a compact fingerprint of resulting state, compared across validators every block, so a divergent node is detected within one round.
Execution discipline	Deterministic by construction: floating point, unordered iteration, and wall-clock time are prohibited in the consensus path. Block timestamps are consensus-derived.

3.1 System model

Let V be the epoch's validator set, s_v the stake bonded behind validator v, and $S = \Sigma_v\, s_v$ the total stake. Voting power is proportional to stake, and the adversary may control validators holding any combined stake strictly below $S/3$. The network is partially synchronous: messages may be delayed arbitrarily before some unknown global stabilization time, after which they arrive within a bounded delay. Under this model MetaFluxBFT guarantees **safety unconditionally** — no two honest validators ever commit conflicting blocks, regardless of network behaviour — and **liveness after stabilization**: the chain keeps committing once the network is synchronous, however long the preceding partition lasted.

A *quorum certificate* (QC) for a block is a set of round-votes from validators holding at least $2S/3$ combined stake. Quorums are the safety instrument, because any two of them intersect in enough stake to contain an honest validator:

$$s(Q_1 \cap Q_2) \geq s(Q_1) + s(Q_2) - S \geq S/3 > s(\text{Byzantine}) \tag{1}$$

An honest validator votes at most once per round, so two conflicting blocks can never both be certified in the same round — the intersection in (1) would have to contain an honest double-voter. Every consensus message is signed with the validator's secp256k1 key over the EIP-712 envelope, which makes equivocation self-evidencing: two conflicting signed votes at the same height and round are, by themselves, a complete slashing proof (§3.5).

3.2 The pipelined two-chain commit

Time is divided into rounds, each with a deterministically assigned leader. The leader proposes a block extending the highest certified block it knows; a validator votes for the proposal only if it extends the validator's own highest quorum certificate, and never votes twice in a round. When the proposal gathers a quorum of votes it becomes certified and the protocol advances. A block is committed by the two-chain rule — a certified block whose direct successor in the immediately following round is also certified is final:

$$\mathrm{QC}(B_r) \wedge \mathrm{QC}(B_{r+1}) \wedge \mathrm{parent}(B_{r+1}) = B_r \Longrightarrow \mathrm{commit}(B_r) \tag{2}$$

The protocol is pipelined: a single vote simultaneously certifies the block it names and forms the second link that commits that block's parent, so consensus settles one block per round at steady state with one message pattern per round, rather than running a separate multi-phase agreement per block. Finality is therefore immediate and absolute — no confirmation depth, no probabilistic settlement, no reorg window. A client that has seen a committed block has seen the permanent state.

This matters more for a derivatives venue than for a general-purpose chain. Liquidations, funding settlements, and bridge releases all execute against finalized state, so no downstream system — market maker, custodian, or counterparty chain — needs to model a rollback, and the risk engine never has to reason about a liquidation that might be unwound.

3.3 Liveness, rotation, and epochs

Progress is protected by round timeouts. If a leader fails to produce a certifiable proposal in time — because it crashed, is partitioned, or is censoring — validators time out and advance to the next round with the next leader. Leadership rotates deterministically across the validator set every round, so a faulty leader costs one round of latency, never the chain: no validator, however large its stake, holds a position the protocol cannot rotate away from within a round. Once the network is synchronous, a round with an honest leader certifies and commits, which yields the liveness guarantee of the partial-synchrony model.

The validator set is fixed within an epoch and re-derived from bonded stake at epoch boundaries: joining, leaving, and stake changes queue during the epoch and take effect at the rollover. Consensus therefore always runs against a closed, known set with fixed voting power, which keeps quorum arithmetic — and the slashing evidence it supports — unambiguous at every height.

3.4 Determinism and state commitment

Agreement on block *order* is only half of consensus for a chain whose state machine is a clearinghouse; the other half is agreement on *state*. Every validator executes the same deterministic transition, and each block carries an `app_hash` — a compact fingerprint of the resulting state — that validators compare every round, so an implementation divergence is detected within one block rather than accumulating silently. Determinism is enforced as a coding discipline with mechanical checks: floating-point arithmetic, unordered iteration, and wall-clock reads are prohibited in the consensus path, all monetary arithmetic is exact fixed-point, and block timestamps are consensus-derived. The consensus model is formally specified, with TLA+ and Stateright verification covering its safety and liveness invariants, and the production implementation is checked against that specification.

A design rule follows from treating the commit rule as a finished safety kernel: **the kernel never changes**. Every MetaFlux-specific mechanism — order admission, the matching and risk engines, forced inclusion of time-critical actions (§11), begin-block effects such as funding and liquidation — lives strictly above it, in admission rules, proposal-validity rules, or the state machine itself. The chain differentiates in market structure and risk while its agreement layer remains the well-studied object the proofs were written about.

3.5 Economic security

Validator misbehaviour carries protocol-defined economic cost. Double-signing — provable from two conflicting signed messages at the same height and round — slashes 5% of stake and jails the validator. Voting for an invalid fork slashes 5% and removes the validator permanently. Sustained unavailability slashes 0.1% and jails. Delegated stake is slashed pro-rata, so delegators bear the same incentives as the validators they select. Unbonding takes a seven-day lock during which the stake remains fully slashable, closing the withdraw-then-attack window. Validators post a minimum self-bond of 100,000 MTF and may charge at most 20% commission on delegator rewards.

4. Architecture

MetaFlux Core is a single deterministic state machine: matching engine, clearinghouse, pricing, risk, and economics are one program whose state transitions are agreed by consensus. There is no off-chain matcher posting settlements back on-chain, and no privileged sequencer. Beside the Core, in the same block, runs a general-purpose EVM.

4.1 The unified block

Each consensus round produces exactly one block at sub-second cadence, and exchange actions and EVM transactions execute together inside it — there is no separate slow lane for contract deployment and no second fee market. Value moves between the two domains as deterministic system transactions: EVM-to-Core transfers dispatch through a system contract, and Core-to-EVM credits materialize in the next block, exactly once, in FIFO order within each round. The EVM fee market is a single EIP-1559 mechanism [9] whose base fee is burned; the block's gas budget is elastic — widening under load, shrinking when idle — above a hard floor, with per-transaction caps sized so that contract deployment confirms in the next block without displacing trading flow.

4.2 Accounts and authorization

Accounts are standard 20-byte addresses derived from secp256k1 keys, and every action — an order, a transfer, a validator vote, a bridge withdrawal — is authorized by an EIP-712 typed-data signature [8]. The signature is the authentication; there are no API keys and no bearer tokens. Every signature commits to the hash

$$h = \text{keccak256}(\ \texttt{0x1901} \,\|\, D \,\|\, \text{keccak256}(m)\ ) \tag{3}$$

where m is the canonical serialization of the action and D is the domain separator binding the protocol name and chain identifier, so no signature can be replayed across networks; per-account monotonic nonces prevent replay within one. Sub-account addresses are derived deterministically as the first 20 bytes of `keccak256(master ‖ uint64(n))`. Three account primitives structure operational risk:

Primitive	Design
Agent wallets	Keys approved by a master account to sign trading actions on its behalf, with per-agent expiry. Agents can trade and cancel; they can never withdraw, transfer out, or manage other agents. Compromise of an agent key can disrupt a trading book; it cannot move funds.
Sub-accounts	Protocol-derived child addresses (up to 32 per master) with fully segregated positions and margin. Funds move only between a master and its sub-accounts, and a sub-account's loss can never involuntarily draw on the master or its siblings. Fee-tier volume aggregates across the family.
Native multi-sig	An account's master key can be irreversibly replaced by an M-of-N signer set ($N \leq 16$) at the protocol layer — the primitive itself, not a contract wrapper — after which every action requires M inner signatures over the same typed-data envelope.

4.3 Gateway

One gateway endpoint per network serves the entire client surface: `POST /info` for reads, `POST /exchange` for signed actions, `GET /ws` for streaming, and `POST /evm` for EVM JSON-RPC. The node itself is not internet-facing. The wire protocol is MTF-native and compact enough to implement directly from the typed-data signing specification; prices and sizes are transmitted as fixed-point integers, never as floating point, so a client can reproduce the chain's arithmetic exactly. A self-hosted node serves the identical surface.

5. Market structure

The base market is a central limit order book with strict price-time priority, matched deterministically inside the block. On top of it the protocol addresses the two questions market microstructure decides: how information is incorporated into price, and whether transaction ordering can be exploited for value extraction.

5.1 Order set

Orders carry a time-in-force of GTC, IOC, or FOK, or rest as ALO post-only orders that cancel rather than cross and never pay taker fees. Market orders are IOC orders bounded by a band around the committed mark price, so no single order can sweep the book beyond a bounded distance from the mark. Trigger orders — stop-loss, take-profit, and their limit variants — arm against the mark price, survive restarts, and evaluate on every commit. Grouping links an entry to its bracket (one-cancels-other) or attaches take-profit/stop-loss to the position itself, surviving partial closes. TWAP execution slices a parent order into scheduled IOC children with randomized timing jitter; scale orders ladder a size across a price range in flat, linear, or geometric shapes. Five self-trade-prevention modes govern what happens when an account would cross itself, down to `DecrementAndCancel`, which nets the overlap without printing a trade. Reduce-only orders are re-validated at commit time against committed state, not at admission.

5.2 Price formation

Three prices govern every market, each with a distinct failure model.

The **oracle price** o_t is a weighted median over up to ten external spot venues, recomputed every block from consensus timestamps:

$$o_t = \text{wmedian}\{\,(p_{v,t},\, w_v) : v \in V_t\,\} \tag{4}$$

where V_t is the set of surviving feeds: a quote older than a staleness bound (60 seconds by default) is dropped, a quote deviating more than 5% from the cross-venue median is rejected, and if the surviving weight falls below half the configured total the oracle holds its previous value rather than update from thin evidence.

The **mark price** — the price at which margin, liquidation, funding, and triggers are evaluated — is the median of three independent components:

$$m_t = \text{median}(\; o_t + \text{EMA}_\beta(mid_t - o_t),\;\; (b_t + a_t)/2,\;\; \text{median(external perp mids)}\;) \tag{5}$$

The first component anchors the mark to the oracle plus an exponentially smoothed basis (decay β = 0.9548 per update — a half-life of roughly 150 seconds); the second is the venue's own book mid, defined only when both sides are quoted and uncrossed; the third is the median of external perpetual mids, defined when at least two of five venues report. An absent component drops out of the median, and the internal book mid alone can never set the mark, so a manipulated spread cannot define its own liquidation price.

The **funding rate** settles discretely per asset — hourly by default, with the period, cap, and multiplier governed per market. The premium is measured against impact prices — the volume-weighted cost of executing a fixed reference notional Q against each side of the book — rather than against the last trade:

$$prem_t = (\;(bid_Q + ask_Q)/2 - o_t\;) \;/\; o_t \tag{6}$$

$$f_t = \bar{P}_t + \text{clamp}(\; r - \bar{P}_t\;;\, \pm c\;), \quad |f_t| \le 2\% \text{ per period} \tag{7}$$

where $\bar{P}_t$ is an exponential moving average of the premium, r is an interest-rate anchor (0.01% per 8 hours by default), and c bounds the per-step correction. If the oracle is stale or the book too thin to fill Q on both sides, the premium sample is zero and the index decays toward zero — an untrusted input never sets a funding payment. At each period boundary a position pays $s \cdot o_t \cdot (F_t - F_{\text{entry}})$, where F is the cumulative funding index and s the signed size. Funding is exchanged directly between longs and shorts; the protocol retains no portion of funding payments.

5.3 Ordering-resistant matching

A deterministically ordered block removes discretionary MEV within the block, but ordering itself still carries information. MetaFlux's answer is a per-market menu rather than one global mechanism, all clearing through the single core matching path. Markets default to the continuous book and opt in per market by governance.

Mode	Mechanism
Frequent batch auctions	Orders collected within a batch interval — one second by default, configurable per market — clear simultaneously at the single uniform price that maximizes crossed volume, with the over-supplied side filled pro-rata and price ties resolved toward the prior mark. Within a batch, latency confers no advantage: all crossing orders clear at the same price [5].
Request-for-quote	A taker publishes a request — side, size, reference price, slippage bound, and a time-to-live of a few seconds — and registered makers respond with firm quotes visible only to the taker. Acceptance settles atomically at the quoted price in the next block, margin-checked on both sides, and prints to the public tape after the fact. Large orders therefore execute without progressive information leakage.
Encrypted order intents	Order contents are threshold-encrypted to the validator set and remain sealed until the order's inclusion and position in the block are already fixed, removing the proposer's ability to act on what it orders. Composed with batch auctions — sealed contents, uniform price — this eliminates the remaining ordering-based strategies.
Verifiable fair ordering	For continuous markets, intra-block placement is seeded from the parent block's hash, so a proposer cannot bias where an order lands within the round.

In batch-auction markets the clearing price maximizes executable volume against the batch's aggregate demand $D(\cdot)$ and supply $S(\cdot)$ curves,

$$p^* = \arg\max_p \min(D(p), S(p)) \tag{8}$$

with ties resolved toward the prior mark. Every buy at or above p^* and every sell at or below it fills at p^*; the over-supplied side is rationed pro-rata, each order filling the fraction $D(p^*)/S(p^*)$ (or its inverse) of its size, so queue position within a batch is irrelevant by construction.

6. Risk & margin

Margin, liquidation, and loss allocation are protocol design surfaces, each specified as a deterministic function over exact fixed-point state. This section defines the margin system, the liquidation ladder, and the loss-allocation mechanism in turn.

6.1 Margin system

Each position margins in one of three modes: **cross**, against the account's pooled collateral; **isolated**, against a pre-allocated bucket that caps the position's maximum loss; or **strict-isolated**, which additionally excludes the position from any portfolio netting. Governance can also place an entire market under strict isolation — a containment tool for a market in distress that takes effect without cancelling a single resting order. Hedge mode lets an account hold long and short legs of the same market simultaneously, each leg margined independently. Initial margin on a position of price p and size s is notional over leverage, rounded up:

$$\mathrm{IM} = \lceil\, |p \cdot s| \,/\, \lambda \,\rceil, \quad \lambda \le 50 \tag{9}$$

where the leverage cap λ steps down through notional-banded tiers as position size grows. Ongoing solvency is measured by account health — account value over maintenance requirement:

$$H = \mathrm{AV} / \mathrm{MM}, \quad \mathrm{MM} = \Sigma_i \, |N_i| \cdot \rho_i \tag{10}$$

with per-asset maintenance ratios ρ_i (300 bps baseline) that step up in the same notional bands. Health is evaluated at every block boundary after the mark update and again after every account-touching action. The bands are governed on-chain and auto-tune from 30-day realized volatility, as does a per-asset funding multiplier, so risk parameters track prevailing market conditions without manual intervention.

6.2 Cross-asset portfolio margin

For multi-asset books, a SPAN-style scenario engine [6] replaces the per-asset maintenance sum. The portfolio is revalued across the scenario grid $G = \{\pm 5\%, \pm 10\%, \pm 20\%\} \times \{\pm 20\%, \pm 50\%\}$ — six price shocks crossed with four volatility shocks, 24 scenarios — and margined on the worst case plus a concentration penalty π:

$$\mathrm{PM} = \max\!\left(0, -\min_{(\delta p, \delta \sigma) \in G} \Sigma_i \left[s_i m_i \cdot \delta p + \tfrac{1}{2} \, |s_i| m_i \, \sigma_i \cdot \delta \sigma \cdot \delta p^2 \right] \right) + \pi \tag{11}$$

$$\pi = 0.10 \cdot \max\!\left(0, \max_i |N_i| - \tfrac{1}{2} \, V\right) \tag{12}$$

where s_i, m_i, σ_i, N_i are position size, mark, implied volatility, and notional, and V is the portfolio's net value: each scenario sums a linear (delta) and a convex (gamma) term per position, and 10% of any single-asset notional above half of net value is charged on top. A portfolio long one correlated asset and short another therefore margins on the residual exposure rather than paying full isolated margin on each leg; typical capital efficiency for a hedged, multi-leg book is 2–5× the classical requirement. Enrollment is opt-in and reversible above a 100,000 USDC equity floor, and the same engine is exposed to smart contracts through a precompile (§7.2), so on-chain programs compute margin with the same arithmetic the clearinghouse applies.

6.3 Tiered liquidation

Most on-chain liquidation engines fire the moment an account crosses the maintenance boundary. That single tick of mark price depresses the book, which trips neighbouring accounts across the same boundary, which depresses the book further — cascade depth is a function of how many accounts cluster at the edge when the shock lands. MetaFlux inserts a grace tier above the cliff and bounds every step below it. The ladder is driven by account health as defined in (10), and every decision is a pure function over exact thresholds.

Tier	Health band	Action
T0 — yellow card	`1.0 ≤ health < 1.1`	Force-cancel post-only resting orders to free parked collateral; notify the account. **No position is touched.**
T1 — partial	`0.8 ≤ health < 1.0`	Close 50% via a price-floored IOC limit — never a market sweep — with a 30-second cooldown before the next step.
T2 — full	`0.667 ≤ health < 0.8`	Full close under the same price floor.
T3 — netting	`health < 0.667`	Position transferred at the committed mark to the most profitable opposite-side accounts — no book interaction, no fees, value-neutral by construction.
T4 — deficit waterfall	negative equity	Auto-deleverage haircut (§6.4), then the insurance fund, then the treasury, in that order.

T0 provides hysteresis: an account pushed into the warning band by a transient price movement receives a consensus round to add margin or reduce exposure before the engine escalates. The forced closes in T1 and T2 are never market sweeps: they are IOC limit orders floored off the committed mark,

$$p_{\text{close}} = m_t \cdot (1 \mp \varphi), \quad \varphi = \rho/2 \text{ by default} \tag{13}$$

so a liquidation for a market with a 3% maintenance ratio can execute no more than 1.5% below the mark; whatever cannot fill at the floor escalates to netting instead of cascading through the order book. A liquidation fee — 50 bps of closed notional by default, charged only from remaining positive equity so it can never create a deficit — capitalizes the insurance fund.

6.4 Auto-deleveraging as online learning

When the ladder still leaves negative equity, the loss must be allocated to profitable counterparties. The industry-standard heuristic — rank profitable accounts by a return-on-equity score and haircut down the queue — exhibits documented failure modes: under stress it over-liquidates profitable accounts, and its ranking is unstable, so accounts with identical profit can be treated very differently. MetaFlux formulates deleveraging as a regret-minimizing online-learning problem [7], split into two independent sub-problems: *how much* to haircut, and *who* bears it.

Severity is a scalar $\theta_t \in [0, 1]$ setting the round's haircut budget $B_t = \theta_t \cdot D_t$ against the observed deficit D_t, learned by projected online gradient descent:

$$\theta_{t+1} = \Pi_{[0,1]}\left(\theta_t - \eta_t \cdot D_t \cdot \operatorname{sign}(\theta_t - \theta^*_t) \right) \tag{14}$$

where θ^*_t is the severity that would have exactly covered the round in hindsight and the adaptive step size η_t tracks realized deficit volatility. The controller carries a dynamic regret guarantee against the best time-varying severity path of total variation P_T:

$$\mathrm{Reg}^{\mathrm{dyn}}_T \le \sqrt{\left((1 + 2P_T) \cdot \Sigma_t D_t^2 \right)} \tag{15}$$

Given the budget, allocation is a deterministic pro-rata rule over each profitable account's realizable capacity u_i:

$$x_i = \min(\, u_i,\ \lfloor\, u_i \cdot B_t \,/\, \Sigma_j\, u_j \,\rfloor\,) \tag{16}$$

with integer rounding dust redistributed one unit at a time in deterministic account order — no solver, no floating point, and min-max fairness by construction: every account is reduced by the same fraction of capacity, and reducing one account's haircut would necessarily raise another's.

In a replay of a major venue's October 2025 deleveraging event, this controller over-liquidated by approximately $3.4M where the return-on-equity heuristic overshot by approximately $45M — a 13× reduction, within roughly 30% of the offline optimum computed with full hindsight — with zero monotonicity violations and near-perfect rank stability, against the baseline's 11.4% violation rate and 0.34 stability [7].

6.5 Insurance, vaults, and the lending pool

The protocol vault backstops the ladder: it is the netting counterparty of last resort, and a governed share of its net asset value — 10% by default — is reserved as the insurance fund that absorbs T4 deficits after ADL. User vaults extend the same share-accounting machinery to delegated trading: depositors mint shares at net asset value, managers hold trading authority but never withdrawal authority, and performance fees (capped at 30%) accrue only above a high-water mark, with management fees capped at 5% annualized and lockups at 30 days. Earn, the protocol lending pool, funds spot margin: suppliers deposit USDC and earn the borrow interest as continuous share-price appreciation — the pool's share price p_s = NAV / shares rises monotonically while loans perform, and supplier yield follows utilization:

$$\mathrm{APY}_{\mathrm{supply}} \approx \mathrm{APR}_{\mathrm{borrow}} \cdot u \cdot (1 - \rho_f), \quad u = \text{borrowed / supplied} \tag{17}$$

with a reserve factor ρ_f retained by the protocol, interest accrued per block, withdrawals bounded by idle liquidity, and borrower positions force-closed by the protocol's own liquidator at a maintenance floor. Yield derives from borrow interest rather than from token emissions.

7. Execution layer

The EVM is the prevailing execution environment for on-chain strategies, but a serial, general-purpose EVM is a poor fit for a high-throughput derivatives chain. MetaFlux retains full EVM compatibility — existing wallets, hardware signers, and trading frameworks work unmodified — and makes the execution layer parallel and derivatives-aware.

7.1 Parallel execution

Transactions execute optimistically in parallel under a Block-STM executor [4]. Every node derives the same conflict partition from transaction content, runs non-conflicting strata concurrently, and catches any mis-estimated access set by read-set re-validation and re-execution. The committed result is byte-identical to a serial execution of the same block; the partition is advisory, so correctness never depends on it. A contract call does not queue behind unrelated transactions, and throughput scales with the block's parallel width rather than a single lane.

7.2 Derivatives-aware precompiles

A Solidity contract that needs a portfolio-margin figure or top-of-book depth otherwise faces two unattractive options: reimplement the logic in bytecode over data it pushed on-chain — expensive, and a second implementation that drifts from the canonical one — or round-trip through an action and wait a block. MetaFlux instead exposes the clearinghouse's own primitives as precompiles, priced far below the equivalent bytecode. The margin a contract quotes is computed by the same engine that will margin the account.

Band	Primitives
Derivatives `0x0900`–`0x0904`	Portfolio-margin evaluation, vault NAV, ADL pro-rata clearing price, mark-to-market settlement, RFQ book depth.
Market data `0x0906`–`0x0908`	Best bid/offer, aggregated L2 depth, inventory-risk read for on-chain market makers.
CoreWriter `0x3333…3333`	The write path: a system contract through which contracts submit native exchange actions — orders and cancels, transfers, staking, vault operations, RFQ submission, batch-auction configuration, encrypted order submission, and cross-chain sends — twenty actions in all.

Write semantics are explicit: a CoreWriter call emits its action and returns, and the action executes on the Core after a short deterministic delay, with the emitted event as the causal link. Core-to-EVM credits land as system transactions in subsequent blocks, exactly once, ordered by round. Governance can disable an individual MetaFlux precompile by stake-weighted vote; the Ethereum-standard precompiles are outside governance reach entirely.

8. MetaBridge

Assets enter and leave MetaFlux through MetaBridge — all of them, including USDC. There is deliberately no third-party bridge and no external attestation service on the critical path: the bridge's trust assumption is exactly the chain's own two-thirds of stake, enforced by the same secp256k1 keys that sign consensus. There is no admin account anywhere in the design.

8.1 Deposits

A deposit on a source chain — EVM corridors such as Base and Arbitrum, or the Solana corridor — is observed by every validator, but only once the source chain reports it *finalized*: a reorged deposit can never mint an unbacked balance. Validators attest independently, and the credit lands when attestations reach a two-thirds stake-weighted quorum. Every transfer carries a deterministic message id

$$id = \text{keccak256}(\ \text{chain} \Vert \text{direction} \Vert \text{user} \Vert \text{asset} \Vert \text{amount} \Vert \text{destination} \Vert \text{nonce}\ ) \tag{18}$$

and each id is honored exactly once, so replays and duplicate attestations are idempotent by construction, even across validator-set changes.

8.2 Withdrawals

Withdrawals are two-phase. The user's request is co-signed by validators to the same two-thirds threshold and queued on the destination chain behind a dual dispute window — 300 seconds *and* 150 destination-

chain blocks, both of which must elapse — before anyone can claim. During the window, any single validator can dispute a suspect withdrawal or pause deposits; reversing that — unpausing, invalidating a withdrawal, or rotating the validator set — requires a two-thirds quorum of cold keys. Hot/cold key separation with two-phase rotation keeps a validator compromise recoverable without ever creating a unilateral spender. Batches settle with partial success: one bad entry is skipped and flagged, never reverting honest withdrawals beside it.

From a contract's point of view, cross-chain movement is a native action, not a convention: a Solidity contract queues an outbound transfer through the CoreWriter's cross-chain send, and MetaBridge releases on the destination chain after quorum and the dispute window. Bridged USDC on MetaFlux is a claim on the source-chain custody balance; the protocol makes this custody relationship explicit rather than obscuring it.

9. Markets & permissionless listing

Every perpetual is a linear, USDC-margined contract with no expiry, funded trader-to-trader. Spot markets trade token-for-token on the same book machinery with escrowed resting orders and no leverage. Spot margin extends spot with leverage funded by the Earn lending pool, isolated per pair, liquidated by the same floored-close discipline as perpetuals. Options and credit-protection markets are designed as later instrument classes on the same clearing stack — the order book, margin engine, and liquidation ladder are instrument-agnostic by construction.

9.1 The MIP suite

Market listing is a protocol mechanism rather than an operator decision. The MetaFlux Improvement Proposals define the market-creation surface:

MIP	Scope
MIP-1	The native spot token standard and permissionless spot pair deployment.
MIP-2	Metaliquidity — the protocol liquidity vault: pooled LP capital with share accounting and NAV marked to oracle, operated by whitelisted strategies.
MIP-3	Permissionless perpetual deployment: anyone can list a perp market by winning an on-chain auction.
MIP-4	A protocol-operated liquidity aggregator offering batch (uniform-price) and instant execution tiers over the same assets.
MIP-5	Earn — the spot lending pool whose utilization curve sets borrow rates for spot margin.
MIP-6	Outcome and prediction markets, once external-event resolution machinery matures.

9.2 Burned-bid deploy auctions

Market creation runs through recurring on-chain gas auctions — one slot per interval, hourly by default, with a linearly decaying reserve. Bids are escrowed and refunded on loss; the winning bid is **burned**, so the cost of listing accrues to all token holders rather than to any operator. The winner supplies the market specification within governance bounds — leverage between 1 and 50×, maintenance ratio of at least 1%,

funding cap of at most 0.5% per hour, a maker-rebate share of up to 2 bps, and a deployer fee under a governance-set cap — and the market accepts orders from its first block. The protocol arbitrates the auction and clears the market; attracting liquidity remains the deployer's responsibility. Because every deployer-set parameter is bounded by governance, a deployed market cannot become a fee trap.

10. Economics

MTF is the native token: it secures the chain through staking, it is the unit in which validators and their delegators are paid, and protocol revenue routes back to holders through an open-market buyback. The design rests on three mechanisms — a supply that tracks an external reference rather than an arbitrary cap, a fee engine that converts net revenue into buy pressure, and a staking ladder that weights committed stake over raw balance.

10.1 Supply: population-pegged

MTF has neither a fixed maximum supply nor an open-ended emission schedule. Total supply is pegged to the population of China: genesis supply is **1,404,890,000 MTF** — one token per person counted in the 2026 population — re-pegged once a year. In year y the reference figure P_y is the median across multiple authoritative Chinese central-government data sources, so no single source's revision can move the peg, and the validator set votes to settle the difference against the current supply target:

$$\Delta S_y = P_y - S_{y-1}, \quad \text{minted to the treasury if positive, burned from it if negative} \tag{19}$$

The adjustment touches only the governed treasury pool — never user, contributor, or staker balances — and at under one percent a year against a billion-token base, circulating supply is effectively stable while remaining anchored to an external, publicly verifiable reference. The peg is not a yield mechanism; the protocol does not fund rewards through perpetual inflation.

10.2 Allocation

Allocation	Share	Terms
TGE airdrop	30%	421,467,000 MTF, distributed at the token generation event to active traders, market makers, and points holders.
Core contributors	20%	280,978,000 MTF. One-year cliff with zero unlock, then six-year linear vesting. No contributor tokens unlock during the first year, and none can be sold into the airdrop.
Treasury · community · validators	50%	702,445,000 MTF. A single governed pool funding grants, liquidity programs, validator-reward bootstrap, and the supply re-peg reserve.

10.3 The fee flywheel

Every protocol fee follows one mechanical path. Maker rebates, referral credits, and builder credits are paid first, off the top, so liquidity providers are compensated before any distribution occurs. The remaining net revenue is split once, three ways:

70%	Buyback — purchases MTF on the open market under a manipulation-resistant price ceiling (in a dislocated market the buyback lags rather than chases), and every purchased token is locked permanently in a keyless address. The tokens are removed from circulation irreversibly: economically equivalent to a burn, executed as continuous open-market demand.
20%	Validators — routed to validators, who pass it to their delegators as the staking dividend. Fees fund validators; validators fund stakers.
10%	Treasury — the governed pool that funds development, security, and ecosystem growth.

The flywheel converts trading volume directly into MTF demand and float contraction while paying the stakers who secure the chain from the same revenue. Because the buyback executes on the open market and the locked address is keyless and public, fee accrual is fully auditable on-chain and cannot be diluted by off-book issuance. Before fee revenue matures, staking rewards bootstrap from the treasury on a square-root curve over total stake S:

$$\mathrm{APR}(S) = 8\% \cdot \sqrt{(\,S_0 \,/\, \max(S, S_0)\,)}, \quad S_0 = 5{\times}10^7 \text{ MTF} \tag{20}$$

— flat at 8% until 50M MTF is staked and decaying as the square root of stake beyond it, funded by allocation, never by new issuance.

10.4 Fee structure

A trader's effective fee is set by three independent tier systems that stack, each rewarding a different contribution: the volume tiers reward flow, the maker-share tiers reward depth, and the staking tiers reward committed stake.

30-day volume	Taker	Maker
< $5M	0.0350%	0.0100%
≥ $5M	0.0300%	0.0080%
≥ $25M	0.0270%	0.0060%
≥ $100M	0.0250%	0.0040%
≥ $500M	0.0220%	0.0020%
≥ $2B	0.0200%	0.0000%

On top of the volume tiers, makers providing at least 0.5%, 1.5%, or 3% of total maker volume earn an additional rebate of 0.0010%, 0.0020%, or 0.0030% respectively, and stakers earn a taker-only discount d of 5% to 50% through the ladder of §10.5. The three systems compose as

$$taker_{\text{eff}} = taker_{\text{tier}} \cdot (1 - d), \quad maker_{\text{eff}} = maker_{\text{tier}} - rebate \tag{21}$$

with the maker rebate paid off the top, before the split, so a top maker's net fee becomes negative. Two bounded redirections complete the schedule: a referrer earns a single-level, immutably-set share carved

from the protocol's cut rather than added to the user's price, and a builder code lets a third-party frontend charge a hard-capped per-fill fee under the trader's explicit on-chain approval. Every redirection is capped, disclosed pre-trade, and paid before the split — the 70/20/10 division applies exactly once, to net revenue.

10.5 Staking: the administrative ladder

Staking MTF confers two benefits: a share of the validator dividend, and a taker-fee discount that escalates with the size and conviction of the stake. The discount is delivered through a ten-rung ladder named for the Chinese administrative hierarchy, from a township office to the top of the state.

Rung	ve weight	Taker discount	Seat
Section Chief (Township Head)	> 100	5%	threshold
Deputy Section Chief	> 500	8%	threshold
Division Chief (County Chief)	> 2,000	12%	threshold
Deputy Division Chief	> 8,000	15%	threshold
Director-General (Mayor)	> 30,000	20%	threshold
Deputy Director-General	> 100,000	25%	threshold
Minister (Governor)	> 500,000	32%	threshold
Vice Minister (Vice Governor)	> 1,500,000	35%	threshold
State Councilor / Vice Premier	> 5,000,000	40%	threshold
Premier / President / General Secretary	> 10,000,000 · rank #1	50%	single seat · cap 1

Rank is determined by **ve-weighted stake**, not raw balance: a stake of size s under committed lock duration ℓ carries effective weight

$$w = s \cdot \mu(\ell), \quad \mu(0) = 0,\ \mu(1\text{ mo}) = 1.0,\ \mu(6\text{ mo}) = 2.5,\ \mu(24\text{ mo}) = 4.0 \tag{22}$$

A flexible, no-lock stake carries a 0× multiplier — lowest rung, no dividend, a mode intended for market makers that require mobile capital. The multiplier applies from the moment of commitment, so a smaller stake under a long lock can outrank a larger stake that retains the option to exit. Every rung except one is an uncapped threshold; the top seat is competitive — held by the single largest holder by ve weight and reassigned in real time. The validator dividend requires at least a one-month lock, and two timing parameters — an activation delay before a new tier takes effect and an exit cooldown on unstaking — are governance-set above a hard 24-hour code floor, so no parameter change can make staking instantly reversible or a discount retroactive.

11. Security model

MetaFlux is a stake-weighted BFT chain: the relevant adversary thresholds are one-third and two-thirds of stake, not a Nakamoto 51%. There is no probabilistic finality and no reorg window.

Byzantine stake	Safety	Liveness	Consequence
< 1/3	`holds`	`holds`	Normal operation.
[1/3, 2/3)	`holds`	`attackable`	Censorship or halt — but no conflicting finality can be forged.
≥ 2/3	`broken`	—	Conflicting blocks could finalize. This is the BFT limit.

No protocol can make a two-thirds colluding quorum safe; that is the boundary of any BFT system, and defense at the boundary is economic and social — stake decentralization, slashing, transparency — not algorithmic. Within the boundary, the design addresses the threats that matter most for an exchange specifically:

- **Censorship of liquidations (≥ 1/3).** The most reachable attack and the most under-appreciated: a one-third coalition could selectively suppress time-critical actions — notably its own liquidations — to push bad debt onto the protocol. The defense is forced inclusion: a proposal is invalid if it omits eligible time-critical actions that were available to the proposer. This lives in proposal-validity rules, above the commit rule, so it strengthens censorship-resistance without touching safety.
- **Consensus safety break (≥ 2/3).** The canonical double-spend scenario. It is deterred economically — double-signing is provable from two conflicting signatures and slashed with self-verifying evidence — and structurally: the consensus model is formally specified, with TLA+ and Stateright verification covering safety and liveness invariants. Slashing deters a rational attacker; against an adversary that shorts the token before attacking, the operative defense is stake decentralization, which is why the validator set opens progressively.
- **Bridge double-spend.** The largest concrete surface on any chain that custodies external assets. Every MetaBridge transfer is validator-cosigned to two-thirds, gated on source-chain finality, idempotent by message id, and released only after a dual dispute window during which a single honest validator can freeze it. There is no administrative key and no external attestation committee.
- **Determinism failure.** A node that computes divergent state undermines safety as surely as an equivocating one. Determinism is enforced mechanically — floating point, unordered iteration, and wall-clock time are prohibited in the consensus path — and every block's `app_hash` cross-checks the entire validator set's state agreement every round.

If the chain halts, nothing is lost: reads keep serving the last finalized state, writes refuse cleanly, liquidations pause against a frozen mark, and validators resume by replay from the last committed block. Release binaries are GPG-signed against a pinned offline root key, and node supervisors refuse unsigned builds. Vulnerability disclosure runs through security@mtf.exchange with 48-hour acknowledgment and 90-day coordinated disclosure.

12. Conclusion

We have specified a Layer 1 blockchain that is an exchange in its state machine rather than in its application layer. Consensus finality is the settlement guarantee; the order book, clearinghouse, and risk engine are one deterministic program; a parallel EVM gives on-chain strategies the same primitives the protocol uses to police them; and a validator-cosigned bridge extends the chain's own trust assumption to external assets rather than importing a third party's. The economics are mechanical throughout: protocol revenue

purchases the token and removes it from circulation, validators and their delegators are paid from revenue rather than from dilution, and market listing is priced by burned-bid auction.

The premise throughout is that derivatives infrastructure is won on capability — market structure that resists extraction, margin that recognizes hedges, liquidation that bounds cascades — and that each of these capabilities must be a property of the protocol, provable from its rules, rather than a commitment by its operator.

Derivatives, on first principles.

13. References

[1] M. Yin, D. Malkhi, M. K. Reiter, G. Golan-Gueta, and I. Abraham. "HotStuff: BFT Consensus with Linearity and Responsiveness." In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC)*, 2019.

[2] R. Gelashvili, L. Kokoris-Kogias, A. Sonnino, A. Spiegelman, and Z. Xiang. "Jolteon and Ditto: Network-Adaptive Efficient Consensus with Asynchronous Fallback." In *Financial Cryptography and Data Security (FC)*, 2022.

[3] The Diem Team. "DiemBFT v4: State Machine Replication in the Diem Blockchain." Technical report, 2021.

[4] R. Gelashvili, A. Spiegelman, Z. Xiang, G. Danezis, Z. Li, D. Malkhi, Y. Xia, and R. Zhou. "Block-STM: Scaling Blockchain Execution by Turning Ordering Curse to a Performance Blessing." In *Proceedings of the ACM Symposium on Principles and Practice of Parallel Programming (PPoPP)*, 2023.

[5] E. Budish, P. Cramton, and J. Shim. "The High-Frequency Trading Arms Race: Frequent Batch Auctions as a Market Design Response." *The Quarterly Journal of Economics*, 130(4), 2015.

[6] Chicago Mercantile Exchange. "SPAN: Standard Portfolio Analysis of Risk." Margining methodology, 1988.

[7] "Autodeleveraging as Online Learning." arXiv:2602.15182, 2026.

[8] R. Bloemen, L. Logvinov, and J. Evans. "EIP-712: Typed Structured Data Hashing and Signing." Ethereum Improvement Proposals, 2017.

[9] V. Buterin, E. Conner, R. Dudley, M. Slipper, I. Norden, and A. Bakhta. "EIP-1559: Fee Market Change for ETH 1.0 Chain." Ethereum Improvement Proposals, 2019.