

Noviscia Protocol Whitepaper

Version: 1.0 (Draft)

Last updated: June 2026

Status: Devnet operational · Mainnet target Q3 2026

Disclaimer: This document describes the Noviscia protocol design and current devnet implementation. It is not financial advice, an offer of securities, or a commitment to future features. Mainnet parameters may change after audit and community review.

Abstract

Noviscia is a **zero-waste perpetual DEX** on Solana. Traders deposit stablecoin margin once, trade USDC-settled perps, and **earn yield on capital that would otherwise sit idle** between trades. Idle margin is deployed across multiple lending venues; when a trader opens a position, funds are **recalled atomically** so margin is available without manual withdrawals.

The protocol separates **yield-bearing margin** (NVSCUSDC vault shares) from **governance** (NVSC). Trading fees and protocol yield feed a **deflationary flywheel**: USDC fees accumulate, NVSC is bought and burned, and stakers receive fee share plus governance over venue allocation.

1. Problem

Perpetual DEX users typically face a trade-off:

Approach	Limitation
Keep USDC in a perp wallet	Zero yield on idle margin
Lend USDC separately (Kamino, etc.)	Manual recall before trading; latency and UX friction
Use a single lending venue	Concentration risk ; suboptimal APY as markets shift

Capital efficiency is left on the table. Noviscia treats margin as a **productive asset** inside the trading stack—not a separate earn product.

2. Solution

One deposit → trade → earn → recall → trade again.

USDC

- | | |
|--------------------------------|----------------------------------|
| → NVSCUSDC Vault (optional) | mint yield-bearing shares at NAV |
| → Escrow PDA (non-custodial) | user-owned margin account |
| → Idle lend (keeper) | deploy free USDC to venue pools |
| → Recall (atomic) | before perp open |
| → Perps (position-tracker) | long/short with oracle marks |
| → Close & settle | PnL + fees routed on-chain |
| → Claim yield / stake / govern | 85% user yield; NVSC flywheel |

Design principles

1. **Non-custodial** — User funds live in program-derived escrow accounts; the protocol never takes discretionary custody.
2. **Atomic recall** — Opening a perp triggers recall of lent margin in the same transaction flow where required.
3. **Dual-token clarity** — NVSCUSDC earns and margins; NVSC governs and captures fee share. No vote inflation from yield shares.
4. **Transparent allocation** — Venue weights are rules-based with optional local AI advisory; stakers govern on-chain caps.
5. **Local-first AI** — Optional Ollama agents for yield routing and risk; no mandatory cloud APIs.

3. Architecture

3.1 On-chain programs

Program	Role
escrow	User USDC / NVSC / NVSCUSDC; lending toggle; idle lend & recall; margin reserve; position settlement
nv-usdc-vault	USDC ↔ NVSCUSDC (ERC-4626-style NAV shares); vault yield accrual
lending-integrator	Multi-venue pools (Kamino / Solend / Marginfi labels); governed weights; deposit/withdraw CPI
position-tracker	Perp positions; dual oracle; open/close; marks; liquidation; fee routing
yield-distributor	Escrow lend yield: 85% user / 15% burn-engine

staking-manager	NVSC stake tiers; governance proposals; trading fee pool
burn-engine	USDC fee accumulation; NVSC buyback & burn (keeper-triggered)
token-nvsc	Fixed-supply NVSC SPL token
liquidation-vault	Insurance / LP layer (program deployed; product UI roadmap)
prediction_market	On-chain prediction markets (adjacent product)

3.2 Off-chain services

Service	Role
Keeper	<code>lend_idle_venue</code> , dual oracle push, position marks, liquidation, yield distribution, vault NAV sync
AI orchestrator (optional)	Local LLM yield router, risk monitor, trade copilot
Web app	Next.js terminal: vault, perps, stake, rewards

3.3 Oracle & marks

- **Dual oracle** (Pyth + optional Switchboard) with on-chain consensus in `position-tracker`.
- Keeper publishes prices; marks must match oracle consensus.
- **Live devnet markets:** SOL-PERP, BTC-PERP, ETH-PERP (catalog-driven; see `perps-catalog.json`).

3.4 Collateral tiers

Tier	Assets	Status
0	USDC	Live
1	NVSCUSDC (vault shares)	Live
2	LSTs (haircut)	Roadmap
3	Staked NVSC (fee discounts)	Stake live; fee discount wiring on perp close = roadmap

4. Token economics

4.1 NVSCUSDC (yield margin share)

Property	Value
----------	-------

Type	Vault share token (not governance)
Mint	On USDC deposit into nv-usdc-vault
Redeem	Burn shares for USDC at NAV
Supply	Elastic
Yield	85% → NAV accrual; 15% → burn-engine
Votes	None

NVSCUSDC is the **primary margin asset** for capital-efficient perps: shares in escrow accrue vault NAV while not reserved for open positions.

4.2 NVSC (governance & utility)

Property	Value
Supply	1,000,000,000 fixed (9 decimals)
Utility	Governance, fee-tier marketing, staking fee share
Staking tiers	Bronze 100 · Silver 1K · Gold 10K · Platinum 100K NVSC
Documented fee discounts	–10% / –25% / –50% / –100% (UI; on-chain perp fee discount = roadmap)
Governance	Proposals to update venue weights & cash buffer

Planned distribution (subject to change pre-TGE):

Allocation	%
Ecosystem	35%
Public sale	25%
Liquidity	15%
Team (vested)	15%
Partners	10%

4.3 Fee flows

Source	User / protocol split
Perp trading fees	40% burn-engine · 60% staking pool (configurable on-chain)
Escrow lend yield	85% user · 15% burn-engine

Vault yield accrual	85% NAV · 15% burn-engine
Liquidation	Keeper incentive (e.g. 1% liquidation_fee_bps)

Burn loop: USDC accumulates in burn-engine → keeper swaps USDC→NVSC off-chain → `trigger_burn` destroys NVSC on-chain.

5. Trading

5.1 Perpetual markets

- **Settlement:** USDC-denominated PnL.
- **Margin:** NVSCUSDC (default) or USDC in escrow.
- **Leverage:** Up to 50× on majors (SOL/BTC/ETH per catalog); lower caps on memecoins when enabled.
- **Markets:** Catalog in `app/web/app/lib/perps/perps-catalog.json` ; `tradeable: true` requires oracle PDA + keeper marks.

5.2 User flow (devnet)

1. Deposit USDC → mint NVSCUSDC (optional) or deposit USDC directly to escrow.
2. Enable idle lending (USDC path) or hold NVSCUSDC shares in escrow.
3. Select market (SOL / BTC / ETH live).
4. Open long or short — protocol recalls lent USDC if needed.
5. Keeper updates marks; user closes → fees split to burn/staking.

5.3 vs typical perps

	Noviscia	Typical Solana perps
Idle margin yield	Automatic multi-venue	Usually none
Yield-bearing collateral	NVSCUSDC vault shares	Plain stables
Venue governance	Staker-weighted on-chain	Rare
Fee → burn	Built-in flywheel	Varies

6. Yield allocation

The **Noviscia Allocator** routes idle escrow USDC across labeled venues (Kamino, Solend, Marginfi):

- **Rules engine** — Transparent coefficients (`ALLOCATION_POLICY.md`).
- **Keeper** — Executes `lend_idle_venue` on eligible escrows.

- **Governance** — NVSC stakers vote venue weights via `set_venue_weights_governed` .
- **AI layer** (optional) — Local Ollama suggests weights; rules engine validates; never sole authority.

Devnet today: Internal venue pools (`lend_mode = internal`). **Mainnet:** External Kamino/Solend/Marginfi CPI (`lend_mode = external`) after audit.

7. Governance

NVSC stakers can:

- Create and vote on proposals (venue weights, cash buffer).
- Execute passed proposals on-chain.
- Receive 60% of perp trading fees (USDC) into the staking fee pool.

NVSCUSDC holders do **not** vote—preventing double influence from yield capital.

8. Security model

- **PDA escrow** per user; no pooled custodial wallet.
- **Oracle consensus** required for marks; deviation caps on-chain.
- **Maintenance margin** + keeper liquidation.
- **CPI boundaries** — Escrow, vault, lending, burn, and staking programs invoked with explicit account constraints.
- **Pre-mainnet:** Independent security audit required (not yet completed in repo).

See `SECURITY.md` for operational guidance.

9. AI layer (optional)

Noviscia ships an **optional, local-first** AI stack:

Agent	Function
Yield Router	Venue weight recommendations
Risk Monitor	Liquidation proximity alerts
Trade Copilot	Perps UI hints (advisory)
Session policy	Off-chain agent scoping (roadmap: on-chain session keys)

No cloud LLM is required. Production deployments can disable AI entirely (`AI_YIELD_ROUTER_IN_KEEPER=0`).

10. Roadmap

Phase	Milestone	Target
0 — Devnet beta	Vault, escrow, perps (SOL/BTC/ETH), keeper, governance	Now
1 — Hardening	Audit, external lending CPI, fee-tier on-chain, insurance vault UI	Pre-mainnet
2 — Mainnet soft launch	NVSC TGE, production oracles, Kamino CPI	Q3 2026
3 — Scale	More perp markets, mobile, session agents, POL	Post-launch

Full checklist: `LAUNCH_ROADMAP.md` .

11. Risks

Risk	Mitigation
Smart contract bugs	Audit, phased rollout, devnet soak
Oracle failure / manipulation	Dual feed, deviation bps, keeper monitoring
Lending venue insolvency	Multi-venue caps, governance weights, recall before trade
Liquidity / slippage on burn	Min/max burn thresholds, keeper cadence
Regulatory uncertainty	Users responsible for local compliance; no investment advice

12. Appendix

Devnet program IDs (June 2026)

Program	Address
escrow	<code>CTmCryJca9cFyMRaGdzrhyZeEnjdGLD8ZkEqNcNbvH2D</code>
position-tracker	<code>3zGRWKZq4V3npHbH9Lati46BwgmstTjynWZFFMxarQgY</code>

nv-usdc-vault	CN92hAtnZxbMxPdho8tugi9GDK86UpGwmnbEvk5yzAWC
lending-integrator	Ea5TXHxsVcnKwMAcAsQkpPN88xr8ndBRpNGDkREWrbSZ
staking-manager	4VDQjH73DiE3zYt66ukyWY7KMMJrxHUZfjkxRHTPDG75
burn-engine	nFgJEQsrKEi7FdAKC6vz5HsQ6f9QjQLBuQcQqQy45id
yield-distributor	CrN1o75FGwcSo6ted7eKxw2kYgkaXDVeWmUaTZCTsLtw
token-nvsc	HSaBJHaGa4Hiv1uBYPHQC4ijmnh8237a5LzM8Lyuz1YT

Related documents

- [USER_GUIDE.md](#) — How to use the app on devnet
- [TOKENOMICS.md](#) — Token separation reference
- [LAUNCH_ROADMAP.md](#) — Launch phases and checklist
- [DEVNET.md](#) — Operator runbook
- [ARCHITECTURE_V2.md](#) — Technical architecture

Noviscia — *The New Science of Capital Efficiency*