KASSTR — Kaspa Strategy Token (Draft Litepaper)

> **Status:** Community draft v0.1 • **Chain:** Kasplex L2 (EVM/KRC-20) • **Ticker:** KASSTR • **Objective:** Algorithmically accumulate **Kaspa (KAS)** and return value to holders via **buyback-and-burn** cycles.

---

1) One-pager Summary

KASSTR is a strategy token on Kasplex L2 that uses a **transaction fee (10%)** to continuously acquire **KAS** into a transparent on-chain treasury. When price conditions are met (e.g., **+20%** over batch cost), the contract sells a portion of its KAS and uses proceeds to **market-buy KASSTR and burn it**, reducing supply. Over time, this aims to **increase KAS per KASSTR** (intrinsic backing) while **shrinking supply**.

Who it's for: Kaspa holders who want an algorithmic, perpetual KAS accumulation engine with upside via deflation.

Why now: Kasplex enables low-fee EVM smart contracts and AMMs, making automated DCA + buyback loops feasible directly in the Kaspa ecosystem.

---

2) Design Goals

- **Simple & transparent**: Minimal moving parts, on-chain metrics, predictable rules.
- **Kaspa-aligned**: A portion of fees optionally **burns KAS** or funds public goods.
- **Self-reinforcing loop**: Trading → fees → buys KAS → sells some at profit → buys & burns KASSTR → repeat.
- **Guardrails first**: Pausable, slippage caps, no raw treasury withdrawals.

---

3) Mechanics (How it works)

1. **Fee on transfer (10%)**
2. **8%**: Accrues to the contract, periodically swapped to **KAS** (treasury growth).
3. **1%**: **Ops/Dev** multisig for audits, maintenance, infra.

4. **1%**: **Kaspa Alignment** — community-decided: **(A)** burn KAS, **(B)** send to public Kaspa wallet, **(C)** reserve for future ecosystem feed.

5. **Accumulation**

6. Fees accumulate in KASSTR. When **buffer ≥ threshold**, contract swaps for **KAS** via AMM (e.g., Zealous Swap).

7. **Profit realization**

8. Strategy tracks average **batch cost** of acquired KAS. When price ≥ **target multiple** (default **1.2×**), contract sells **a fraction** of treasury KAS (default **15%** of treasury) with slippage limits.

9. **Buyback & Burn**

10. Proceeds market-buy **KASSTR** and send directly to **burn address**. Supply falls; each remaining token represents a larger share of the growing KAS treasury.

11. **Cycle repeats**

12. Trading volume funds new KAS buys; rallies fund burns.

    No redemptions: KASSTR is **not** redeemable for KAS. Value accrues via backing growth and supply reduction, not a direct claim.

---

4) Core Parameters (defaults) — subject to community vote

Parameter	Default	Rationale
Transaction fee (tax)	**10%**	Mirrors successful strategy tokens; funds accumulation.
Fee split	**8% KAS buys / 1% Ops / 1% Kaspa Alignment**	Maximizes KAS growth while funding upkeep & alignment.
Fee buffer threshold	**0.10% of total supply** (in KASSTR)	Batches swaps to reduce gas/slippage.
Profit trigger	**+20%** vs. batch VWAP	NFT-strategy analogue (1.2× relist). Simple & clear.
Sell fraction on trigger	**15%** of treasury KAS	Realizes profit without depleting treasury.
Max swap slippage	**0.5–1.0%**	Avoids toxic price impact.
Cooldown between ops	**30 minutes**	Prevents rapid fire loops/manipulation.
Exempt addresses	LP pair, treasury, burn, router, multisig	Avoids self-taxing + plumbing.

---

5) Open Decisions for Community Governance

Please discuss and vote on the following. We'll lock v1 after community signaling.

D1. Kaspa alignment (1% of each trade) - **A. Burn KAS** (deflationary for Kaspa; strong alignment signal) - **B. Send to a public Kaspa community wallet** (ecosystem dev grants; transparent) - **C. Hold for future cross-strategy feed** (if more strategy tokens emerge)

D2. Profit skim per event - **10%**, **15% (default)**, or **20%** of treasury KAS sold when trigger hits.

D3. Trigger logic - **Fixed +20%** vs batch cost (**default**) - **Volatility-aware** (e.g., ATR-based; ramps target in high vol)

D4. Launch & distribution - **Fair launch LBP** (price discovery) vs. **Seeded LP** + **community airdrop** to KAS addresses. - **Initial liquidity target** (e.g., ~$250k equivalent in KAS) and lock period.

D5. Governance & keys - **Multisig composition** (e.g., 3/5 with community reps). List nominees. - **Upgrade window** (30–60 days), then **renounce/lock** critical controls.

D6. Ops/Dev budget usage (1%) - Audit(s), monitoring infra, indexer & dashboard hosting, bug bounties. - Quarterly transparency report commitment.

D7. Buyback venue & routing - Primary AMM, acceptable route(s), max % of pool per buy, TWAP vs. market orders.

---

6) Tokenomics

- **Name/Symbol:** Kaspa Strategy / **KASSTR**
- **Standard:** KRC-20 (Kasplex EVM)
- **Max Supply: 1,000,000,000 KASSTR** (minted at deploy)
- **Initial Liquidity:** KASSTR/KAS AMM pool; LP tokens time-locked.
- **Tax:** 10% on transfers (buys/sells/transfers), exemptions apply per §4.
- **Treasury:** On-chain KAS holdings; no arbitrary withdrawals; only whitelisted swap/buyback flows.
- **Deflation:** Buybacks send KASSTR directly to burn; total supply decreases over time.

---

7) Smart Contract Architecture (high level)

- **KASSTR Token**
- Fee-on-transfer, exemptions, fee accrual buffer.
- Events: `FeesAccrued`, `SwapToKAS`, `ProfitTrigger`, `Buyback`, `Burn`.
- **Treasury/Broker Module**
- Executes swaps via AMM router; guards: slippage cap, cooldown, size cap.
- Batch ledger for average cost (compact rolling windows).

- **Admin/Guardrails**
- **Pausable**, router whitelist, timelock + multisig for parameter nudges.
- No raw KAS withdrawals to EOA; only strategy functions.

---

8) Transparency & Metrics (public dashboard)

- **Treasury KAS balance** & batch cost basis
- **mNAV (KAS per KASSTR)** and market premium/discount
- **Total KASSTR burned** (deflation curve)
- Recent **buys/sells** (tx links), fees collected, ops wallet spends

---

9) Roadmap

1. **Community Signaling** (this doc): lock D1–D7.
2. **Testnet Dry-Run**: deploy, simulate fees, verify swaps/buybacks.
3. **Audit + Guarded Mainnet**: small caps on swap size/day, live dashboard.
4. **Governance Handoff**: finalize multisig, publish runbook & KPIs.
5. **Ecosystem Integrations**: CEX/DEX listings, indexers, potential staking module.

---

10) Risks & Disclosures

- **Market risk:** KAS volatility can reduce treasury value; triggers may miss optimal moves.
- **Liquidity/volume risk:** Low activity reduces fee inflow and execution quality.
- **Smart contract risk:** Bugs or AMM/router issues; mitigated via audits, caps, pause.
- **Governance risk:** Misconfigured parameters; mitigated via timelocks, community oversight.
- **Speculation risk:** Token may trade far from mNAV; no redemption mechanism.

This is **not** an investment contract or promise of profit. KASSTR is an experimental mechanism; participate only if you understand and accept the risks.

---

11) Launch Checklist (to be filled with community)

- [ ] Finalize **D1–D7** decisions
- [ ] Publish addresses: **KASSTR**, **Treasury**, **Multisig**, **Burn**
- [ ] Publish AMM **router** & **pair** addresses
- [ ] Lock LP tokens; publish lock tx
- [ ] Dashboard live (treasury, mNAV, burns)
- [ ] Audit report link + bug bounty scope

---

12) Appendix

A. Pseudocode sketch

```
onTransfer(sender, recipient, amount):
  if sender/recipient in exempt: transfer full
  else:
    fee = amount * 10%
    take(fee)
    allocate(8% -> feeBuffer, 1% -> ops, 1% -> kaspaAlign)
    transfer(recipient, amount - fee)

maybeSwapToKAS():
  if feeBuffer >= threshold and cooldown passed:
    swap(feeBuffer -> KAS) respecting slippage; add to treasury; record batch
cost

maybeTakeProfit():
  if price(KAS) >= 1.2 * batchVWAP and cooldown passed:
    sell = treasuryKAS * 15%
    swap(KAS -> stable/route -> KASSTR) recipient = burn
```

B. Glossary

- **Batch VWAP:** Volume-weighted average price of last acquired KAS tranche.
- **mNAV:** Modified net asset value; KAS per KASSTR including expectations.
- **TWAP:** Time-weighted average price; can reduce sandwich risk on buys.

Feedback wanted: Please comment on **D1–D7**. After a 5–7 day discussion window, we’ll freeze v1, deploy testnet, and publish audit scope.