Zarin (ZRN) Whitepaper

Abstract. Zarin (ZRN) is a straightforward ERC-20 token focused on clarity and verifiability. The contract exposes a minimal set of on-chain levers — 10% emissions, an owner-set minimum interval between emissions, a permanent finalize-mint switch, and treasury-only burns. Critically, the minimum emission interval is one-way: the owner can only **increase** it (never decrease), which enforces a non-accelerating emission cadence and caps long-run inflation once the project matures. Day-to-day price smoothness is handled off-chain via public liquidity pools (primary on Arbitrum and BSC, optional auxiliary on Ethereum) and a budget-bounded reserve policy. No promises of returns; the project favors simple mechanics over hype.

1. Overview & goals

Zarin prioritizes simplicity, auditability, and explicit controls:

- **Minimal surface.** Standard ERC-20 with 18 decimals; no hidden mint logic or opaque tokenomics.
- **Predictable expansion.** Emission is a fixed 10% of current `totalSupply` and can be permanently disabled.
- **Liquidity first.** Public AMM pools on Arbitrum and BSC (primary), with an optional auxiliary pool on Ethereum.
- **Open policy.** Reserve actions (if any) follow a published corridor and daily USD budget cap.
- **One-way emission cadence.** The minimum emission interval is *monotonic* (non-decreasing): it can only be raised, never lowered — preventing future accelerations in issuance and effectively bounding long-run inflation.

2. Contract & supply

ZRN uses a non-upgradeable ERC-20; constructor mints the initial supply to the treasury on deployment.

- **Symbol:** ZRN **Decimals:** 18
- **Initial supply per network:** 1,000,000,000 ZRN (minted to the treasury at deployment)
- **Treasury:** set at deployment; supply minted to treasury
- **Networks:** Ethereum Mainnet, Arbitrum One, BNB Chain (BSC)

3. Emission (10% of current supply)

The contract exposes a clear emission mechanism:

- `mintNextEmission()` — mints exactly 10% of the current `totalSupply` to the treasury; blocked if `totalSupply` is zero or the computed amount would be negligible.
- `setMintInterval(uint256 newInterval)` — defines the minimum time between emissions while emissions remain enabled. **Safety property:** the interval is *monotonic* — it can only be **increased** (non-decreasing). Calls that do not strictly increase the current interval are **blocked by a require check** and do not change state. This guarantees that future emissions cannot be scheduled more frequently than previously allowed.
- `finalizeMinting()` — a one-way action that permanently disables all future emissions and any interval changes.

In the discovery phase, the interval may be set low. After price stabilizes, the interval will be increased to enforce a slower, more predictable schedule, or minting can be finalized entirely. **Because the interval can only move upward, any increase permanently tightens the maximum emission frequency, establishing an explicit upper bound on long-run inflation.**

4. Liquidity & reserve (off-chain policy)

- **Primary pools:** Arbitrum (Uniswap v3, ZRN/USDC) and BSC (PancakeSwap v3, ZRN/USDT or ZRN/WBNB).
- **Ethereum pool:** optional auxiliary/limited liquidity; can be expanded as volumes grow.
- **Corridor & budget:** public target price with corridor ±δ%; corrective trades follow a daily USD budget cap (e.g., $10–20k).
- **Event-driven:** faster micro-corrections allowed within the same daily cap after large single trades.

5. Post-discovery actions

After a market price forms, the project may:

- **Operate a reserve** (if established) to smooth volatility within public, budget-bounded rules.
- **Burn surplus tokens** via `burnFromTreasury(amount)` if it supports healthier market behavior.
- **Finalize minting** when supply growth is no longer desired.

6. Governance & transparency

Governance begins pragmatically (owner-led) and may evolve toward a multisig or governance wrapper. All chain addresses and key actions (emissions, burns, reserve operations) are published via official channels.

7. Risks

Crypto assets are volatile and carry execution, market, and regulatory risks. Emissions, burns, and reserve operations may influence price dynamics but do not guarantee outcomes. Users should independently assess smart-contract and operational risks.

8. Contract addresses

- **Ethereum Mainnet** — Address: `0xA6C63DC375b0206373fB99F856B4f71B311670B9`
  Explorer: Etherscan (code)
- **Arbitrum One** — Address: `0x9F9d504ca217cfcAfbC250308385AD14b17bfBC1`
  Explorer: Arbiscan (code)
- **BNB Chain (BSC)** — Address: `0x0d431E5A8c290f11a0e63911eC4202FB60D2E80b`
  Explorer: BscScan (code)
- **Sepolia (demo)** — Address: `0xf2695b87924b728e1478Cf9e0F326262c983e6cA`
  Explorer: Etherscan Sepolia (code)

9. Interface & functions (quick spec)

- `mintNextEmission()` — *onlyOwner*; mints exactly 10% of `totalSupply` to the treasury; blocked if supply is zero or the computed amount is too small.
- `setMintInterval(uint256 newInterval)` — *onlyOwner*; sets the minimum spacing between emissions while emissions are enabled; **monotonic:** the new interval must be strictly greater than the current one, otherwise the call is **blocked by a require check** and no state changes occur.
- `finalizeMinting()` — *onlyOwner*; permanently disables further emissions and prevents any future interval updates.
- `burnFromTreasury(uint256 amount)` — *onlyOwner*; burns tokens held by the treasury with event emission and balance checks.
- `transferOwnership(address newOwner)` — *onlyOwner*; standard ownership transfer (e.g., to multisig/governance).

10. Roadmap (high-level)

- **Discovery (launch & discovery):**
  A. **Contracts & verification:** deploy on target networks; verify source on explorers; publish all addresses and fingerprints (code hashes/ABI), update README and `tokenlist.json`; host the Whitepaper page.
  B. **Liquidity & access:** create primary v3 pools (Arbitrum ZRN/USDC; BSC ZRN/USDT; Ethereum auxiliary if needed); seed initial liquidity; add official "Swap on …" buttons/links.
  C. **On-ramp & UX:** embed a fiat on-ramp widget (MoonPay / Transak / Ramp); add a clear "How to Buy" guide; ensure wallet/network checks and a simple demo/read page.
  D. **Marketing & comms:** announce on official channels (Telegram/X); publish a lightweight press kit (logo, addresses, 1-pager); post regular progress updates; consider listings/directories and selective KOL/AMA outreach as readiness improves.
  E. **Monitoring & safety:** set up pool/price monitoring (incl. TWAP/ranges), budget dashboards and alerts; keep an incident-response playbook; provide a permissions-revocation guide for users.
- **Stabilization:** raise `mintInterval` floor in steps; begin discretionary burns when appropriate.
- **Optional reserve:** if community supports it, establish and disclose transparent reserve rules.
- **Governance:** formalize processes; document policies; consider `finalizeMinting()` when emission is no longer needed.

11. How to acquire ZRN

1. Use a fiat on-ramp (MoonPay / Transak / Ramp / etc.) to buy USDC/USDT or native gas into your wallet.
2. Swap to ZRN in the official pools on Arbitrum/BSC (links on the website/README). Ethereum auxiliary pool may be used when available.

Note: Traditional processors often disallow direct token sales via cards/PayPal and involve chargeback risk; the on-ramp → DEX flow minimizes counterparty and compliance risk.

12. Community & links

- Website: zrncoin.com

- Telegram: @zarintoken
- GitHub: github.com/zarincoin/zarincoin
- Token list (repo): `tokenlist.json`

13. Version

Document: Zarin Whitepaper. Last updated: *November 2025*.