NorthernPay (NPY)

Shariah-Compliant, Gas-Relayer Sponsored Fast Payment Token on Polygon

NorthernPay Foundation

Version 1.1 | August 17, 2025

One-line summary: NorthernPay (NPY) is a Shariah-compliant ERC-20 token (mintable, burnable, 8-decimal precision) deployed on Polygon via Thirdweb. End users pay **no on-chain NPY transaction fee**; gas is subsidized by a relayer network (in development).

Core parameters: Total supply $S_{total} = 2.1 \times 10^9$ NPY   Decimals = 8   No premine   Mintable/Burnable   Relayer-subsidized gas (gasless UX).

Contents

Abstract

This document is a comprehensive technical and economic specification for NorthernPay (NPY), a Shariah-compliant retail payment token designed for instant, low-friction payments and remittances. NPY is engineered to deliver a gasless user experience by employing an off-chain/on-chain relayer architecture that subsidizes gas costs for users. The token is issued on Polygon (EVM-compatible) using Thirdweb Token Creator, enabling rapid deployment, integrations, and modular governance. This whitepaper details vision, technical architecture, tokenomics, compliance, governance, security, and an extensive roadmap. It includes illustrative Solidity code for core operations—mint, burn, transfers—and a sample relayer/forwarder pattern. All tokenomic formulas and unit conversions are documented precisely.

1 Introduction & Vision

1.1 Background

Digital payments have transformed commerce but many audiences still face frictions: onboarding complexity, transaction costs, and trust mismatches with cultural or religious constraints. NorthernPay addresses these frictions with a single objective: bring an accessible, auditable, and Shariah-aware payment token to everyday users and merchants.

1.2 Vision Statement

NorthernPay's vision is to become the default digital payment medium for communities requiring Shariah compliance while providing an exceptional user experience: instant settlement, near-zero friction, and native blockchain transparency. The network will be developer-friendly, merchant-integrable, and governed with strong audit controls.

1.3 Design Principles

- **Sovereignty:** Users hold tokens in their wallets; custodial models are optional.
- **Simplicity:** Minimal on-chain complexity in the token contract; complexity moved to modular relayer and middleware layers.
- **Compliance-first:** Built-in operational processes for Shariah review and KYC/merchant onboarding.
- **Gasless UX:** End users experience no transaction fees; relayers sponsored by ecosystem participants handle gas.
- **Economic prudence:** Transparent minting/burning with governance oversight; no premine.

2 Use Cases

2.1 Retail Point-of-Sale

Merchants accept NPY for everyday purchases. Relayers sponsor gas and batch transactions to minimize on-chain overhead, enabling instant-like settlement.

2.2 Micro-remittances

Low friction, gasless transfers for remittances between individuals across borders.

2.3 Online Payments and Subscriptions

Integrations via SDKs enable recurring payments with on-chain receipts and optional off-chain settlement agreements.

2.4 Charitable Giving and Zakat

Transparent disbursement flows and tools to compute zakat and charitable distributions, with Shariah board oversight.

3 Technology Stack & Architecture

3.1 Core Choices

- **Base Layer:** Polygon (EVM-compatible) for low cost, fast confirmations.
- **Token Issuance:** Thirdweb Token Creator for standardized ERC-20 deployment with administrative GUI and multisig compatibility.
- **Smart Contracts:** OpenZeppelin patterns (ERC-20, AccessControl), audited modules, and relayer-forwarder contracts (EIP-2771 / MinimalForwarder pattern).
- **Relayer Network:** Off-chain relayer services that submit meta-transactions to Polygon, covering gas for users.

3.2 High-level Architecture

Architecture diagram placeholder: Wallets ↔ Signed Meta-Transactions → Relayers → Trusted Forwarder → Token Contract (Polygon)

Figure 1: High-level Relayer and Token Interaction

3.3 Relayer Model (Gasless UX)

To provide a no-fee experience to end users, NorthernPay uses a **relayer-sponsored meta-transaction** model. Key components:

1. **Client Wallet:** User signs a meta-transaction (EIP-712 structured data) authorizing an action (transfer/mint/burn).
2. **Relayer Node:** Relayer collects signed meta-transactions and submits them on-chain, paying gas in MATIC.
3. **Trusted Forwarder / MinimalForwarder:** Verifies signature, nonce, and payload; then calls the token contract with the original sender's context.
4. **Subsidy Model:** Relayers are incentivized or compensated off-chain (via merchant sponsorship, treasury grants, or relayer staking rewards). Importantly, the whitepaper's canonical policy: **end users do not pay token transaction fees on-platform**.

3.3.1 Relayer Security & Anti-Abuse

Relayer network design considerations:

- **Whitelisting and Rate-limits:** To prevent spam relayer operations, relayers enforce per-wallet rate-limits and require authorization for certain high-value actions.
- **Staking and Slashing:** Optional relayer staking model; relayers stake deposits to guarantee service quality; slashing penalties for proven misbehavior.
- **Replay Protection:** Nonce per user and strict replay-prevention in forwarder contract.
- **Monitoring:** Active telemetry, alerts, and anomaly detection to protect users and merchants.

4 Token Specification & Tokenomics

4.1 Canonical Parameters

- **Name:** NorthernPay
- **Symbol:** NPY
- **Standard:** ERC-20 (EVM-compatible)
- **Total supply (nominal cap):** $S_{total} = 2.1 \times 10^9$ NPY
- **Decimals:** $d = 8$
- **Initial circulation:** 0 (No premine; tokens enter circulation via controlled minting events)
- **Mintable:** Yes (authorized minters; multisig / governance)
- **Burnable:** Yes (holders may burn; treasury may burn)
- **Transaction Fees to End-Users: None.** Gas is subsidized by relayers (in progress). There will be **no on-chain fee charged in NPY to end users** by the token contract.

4.2 Unit Conversion

Define smallest indivisible unit (base unit) as 10^{-d} NPY. The total supply in base units:

$$S_{\text{units}} = S_{total} \times 10^d = 2.1 \times 10^9 \times 10^8 = 2.1 \times 10^{17}$$

4.3 Supply Dynamics

Circulating Supply Let B denote cumulative burned tokens (nominal units), and M denote cumulative minted tokens since genesis (nominal units). Circulating supply:

$$S_{circulating} = M - B$$

Note: Since there is no premine, M starts at 0 and only increases under governance-authorized events up to the nominal cap defined in policy.

Inflation / Deflation Metric A succinct metric I to capture net supply change relative to the nominal cap:

$$I = \frac{M - B}{S_{total}} \times 100\%$$

This metric helps governance quantify dilution/inflation against the stated cap.

Market Price Relationship Given market capitalization V_{market} (USD), the exchange value per token is:

$$P_{NPY} = \frac{V_{market}}{S_{circulating}}$$

This relation is useful for analytics, but note that liquid market price may differ due to liquidity, slippage, and exchange spreads.

4.4 Distribution Policy (Governance-controlled)

Although there is **no premine**, distribution events are planned and governed. The high-level allocation goals (subject to governance approval and vesting schedules):

Category	**Percent of S_{total}**	**Purpose / Notes**
Ecosystem & Developer Grants	25%	Funding integrations, SDKs, bounties, DApp grants
Merchant Incentives	20%	Subsidies to onboard merchants, relayer sponsorships
Liquidity	15%	Market making and initial liquidity pools (timed release)
Treasury	15%	Strategic reserves, emergency stabilizer
Team & Advisors	10%	Vesting schedule, time-locked
Community	10%	Airdrops, community initiatives (subject to governance)

Vesting All allocations to team, advisors, and treasury will be subject to time-based vesting with cliffs and linear release to avoid sudden token dumps. Exact schedules are to be published in governance proposals and multisig-executed timelocked transactions for transparency.

4.5 Monetary Authority & Governance Controls

- **Minting Controls:** Only authorized minter role(s) may mint; these roles are governed by a multisig(3/5 or higher) and subject to on-chain governance processes.
- **Burn Authority:** Any holder may burn their tokens (standard ERC-20 burn). Treasury burns require multisig.
- **Parameter Changes:** Any change to token policies, allocation, or relayer economics must be proposed via governance and executed with time-lock to allow audits.

5 Compliance: Shariah Framework

5.1 High-level Shariah Approach

NorthernPay aims to adhere to leading Shariah principles:

- **No Riba:** No interest-bearing obligations inherent to the token.
- **Risk Transparency:** Clear, auditable token mechanics; deterministic mint/burn rules.
- **Avoiding Gharar:** No opaque, excessively uncertain contractual mechanisms.
- **Prohibition of Haram Financing:** Merchant onboarding policies to prevent facilitation of prohibited activities.

5.2 Shariah Board & Audit

A formal Shariah board will review tokenomics, products, and new financial primitives. Regular public memos and fatwas (where applicable) will be published. Smart contract audits and Shariah compliance assessments will be dual-tracked.

5.3 Operational KYC/AML

While the token itself is censorship-resistant, NorthernPay ecosystem services (merchant onboarding, relayer sponsorships, fiat on/off ramps) will follow KYC/AML best practices per jurisdiction to reduce abuse and align with compliance expectations.

6 Security & Audits

6.1 Smart Contract Security

- Use battle-tested libraries (OpenZeppelin).
- Role-based access (AccessControl) and multisig custody for all critical operations.
- Minimal attack surface in token contract; relayer/forwarder architecture isolates signature verification from token logic.

6.2 Audit Program

1. Pre-deployment internal audits and formal external audits by top-tier firms.
2. Bug bounty program and continuous monitoring.
3. Formal verification for critical modules (forwarder, signature verification).

6.3 Operational Security (Relayers)

Relayers are off-chain services; their security posture includes:

- Private key management (HSMs and hardware wallets).
- Rate limiting, IP allowlists, and logging.
- Monitoring of transaction pools to detect abnormal behavior.

7 Governance

7.1 Governance Philosophy

NorthernPay's governance blends on-chain proposals with off-chain deliberation and Shariah board oversight. The initial governance model is foundation/multisig-led with gradual decentralization.

7.2 Governance Modules

- **Proposal Submission:** Community or foundation submits proposals describing parameter changes, fund allocations, or upgrades.
- **Review Process:** Technical review + Shariah review (if relevant) + public comment period.
- **Execution:** Approved proposals executed by multisig or timelocked executor.

7.3 Decision Rights

Certain classes of decisions (core security patches, emergency actions) remain under immediate multisig control, with public disclosure and auditability.

8 Roadmap (Detailed)

Year 0 — Preparation (Months 0–3)

- Team hiring and core architecture finalized.
- Shariah board recruitment (initial advisors identified).
- Whitepaper v1 release and community kickoff.
- Development of relayer prototype and minimal forwarder tests on testnet.

Year 1 — Launch (Months 3–12)

- Thirdweb token deployment on Polygon testnet; rigorous audits.
- Alpha relayer rollout; merchant pilot programs (selected merchants).
- Mobile/web wallet SDKs and merchant SDKs release.
- Public beta with limited liquidity events and community grants.

Year 2 — Growth (Months 12–24)

- Mainnet feature expansion; cross-border remittance partners.
- Fiat on/off ramp integrations with compliant partners.
- Formal relayer marketplace with staking incentives.

- Governance expansion: token-holder voting model (if adopted).

Year 3 — Maturity (Months 24–36)

- Ecosystem maturity: partners, merchant adoption, developer community.
- Shariah-reviewed financial products (non-interest-bearing savings products).
- International expansions, regional hubs, and regulated entity establishment if required.

9 Implementation: Example Contracts & Relayer Patterns

9.1 Abridged ERC-20 Token (Mintable & Burnable)

Listing 1: NorthernPay ERC-20 (abridged)

```
pragma solidity ^0.8.17;

/*
  Illustrative only. Use audited OpenZeppelin contracts and production-ready access control.
*/
import "@openzeppelin/contracts/token/ERC20/ERC20.sol";
import "@openzeppelin/contracts/access/AccessControl.sol";

contract NorthernPay is ERC20
AccessControl {
    bytes32 public constant MINTER_ROLE = keccak256("MINTER_ROLE");
    uint8 private constant _decimals = 8;

    constructor(address admin) ERC20("NorthernPay", "NPY") {
        _setupRole(DEFAULT_ADMIN_ROLE, admin);
        // No premine: initial supply is zero
    }

    function decimals() public pure override returns (uint8) {
        return _decimals;
    }

    // Mint: only accounts with MINTER_ROLE can mint
    function mint(address to, uint256 amount) external {
        require(hasRole(MINTER_ROLE, msg.sender), "Not a minter");
        // amount is expressed in base units (10^-8)
        _mint(to, amount);
    }

    // Burn: holder can burn their tokens
    function burn(uint256 amount) external {
        _burn(msg.sender, amount);
    }

    // Administrative burn from treasury (requires minter or admin role
       ↪ )
    function burnFrom(address from, uint256 amount) external {
        require(hasRole(MINTER_ROLE, msg.sender), "Not authorized");
        _burn(from, amount);
    }
}
```

9.2 Minimal Forwarder (EIP-2771 / MinimalForwarder pattern)

This forwarder contracts verifies signatures and nonces, enabling relayers to submit meta-transactions preserving original sender context.

Listing 2: MinimalForwarder (illustrative)

```
pragma solidity ^0.8.17;

contract MinimalForwarder {
    struct ForwardRequest {
        address from;
        address to;
        uint256 value;
        uint256 gas;
        uint256 nonce;
        bytes data;
    }

    mapping(address => uint256) public nonces;

    // EIP-712 domain separator and typehash omitted for brevity

    function verify(ForwardRequest calldata req, bytes calldata
       ↪ signature) public view returns (bool) {
        // verify signature using EIP-712 (omitted: implement typed
           ↪ data hash and ecrecover)
        // also verify nonce
        return true;
    }

    function execute(ForwardRequest calldata req, bytes calldata
       ↪ signature) public payable returns (bool, bytes memory) {
        require(verify(req, signature), "Invalid signature or nonce");
        nonces[req.from] = req.nonce + 1;
        // perform call preserving 'from' as the original sender; token
           ↪  contracts using _msgSender() hooks will see the
           ↪ forwarded from
        (bool success, bytes memory returndata) = req.to.call{gas: req.
            ↪ gas, value: req.value}(req.data);
        require(success, "Forwarded call failed");
        return (success, returndata);
    }
}
```

9.3 Relayer (Off-chain) — Operational Outline

Relayers operate off-chain but must follow strict behaviors:

1. Collect signed ForwardRequest payloads from wallets.
2. Validate signatures and check nonce/anti-replay.
3. Optionally perform eligibility checks (KYC/blacklist/limits).
4. Submit transactions to MinimalForwarder.execute with appropriate gas price and gas limit.
5. Record and telemetry-report transactions; reconcile off-chain compensation (if any) with merchant or treasury.

9.4 Example: Transfer via Meta-Transaction (User Experience)

1. Wallet signs meta-transaction for a `transfer(to, amount)` call of the NPY contract.
2. Wallet sends signed request to a relayer endpoint.
3. Relayer submits to forwarder; forwarder calls NPY contract on-chain with relayed context.
4. On-chain event emitted; end user receives confirmation in wallet UI (no MATIC spent by user).

10 Economics of Relayer Sponsorship (Descriptive)

Relayer costs (MATIC gas) are borne off-token from the perspective of end users. Potential sponsorship models include:

- **Merchant sponsorship:** Merchants pay relayers off-chain or through settlement agreements to process customer transactions.
- **Treasury-funded onboarding:** Initial relayer subsidies paid from ecosystem funds to bootstrap merchant adoption.
- **Relayer marketplace:** Relayers may sell service packages to merchants with SLA guarantees.

Each model must be carefully audited for Shariah-compliance (no riba, transparent agreements).

Mathematical Appendix

Total Supply (nominal)

$$S_{total} = 2.1 \times 10^9$$

Decimals and Base Units If $d = 8$:

$$S_{\text{units}} = S_{total} \times 10^d = 2.1 \times 10^{17}$$

Circulating Supply

$$S_{circulating} = M - B$$

where M is total minted and B is total burned (nominal units).

Inflation/Deflation Rate

$$I = \frac{M - B}{S_{total}} \times 100\%$$

Market Price Relationship

$$P_{NPY} = \frac{V_{market}}{S_{circulating}}$$

11 Governance, Legal, and Risk Considerations

11.1 Legal Structure

NorthernPay Foundation will maintain legal entities in compliant jurisdictions to manage treasury, relayer sponsorship agreements, and partner contracts.

11.2 Regulatory Risks

Regulatory classification of tokens varies by jurisdiction. The Foundation will proactively engage with regulators, adopt KYC/AML where required by integrations, and update policy with legal counsel.

11.3 Operational Risks

- Relayer centralization risk — mitigate via multiple relayers and open relayer marketplace.
- Oracle risks — minimize by using multiple feeds and fallback strategies.
- Smart contract bugs — mitigate via audits, bounty programs, formal verification where feasible.

12 Technical Implementation Roadmap (Milestones)

1. Complete core token contract and thirdweb deployment pipeline.
2. Implement, test, and audit MinimalForwarder and meta-transaction flows.
3. Build relayer infrastructure (scalable node software, API endpoints, telemetry).
4. Merchant SDKs for web and mobile (React, Android, iOS).
5. Testnet pilot with selected merchants performing gasless transactions.
6. Security audits, Shariah board reviews, and legal compliance checks.
7. Mainnet launch on Polygon with phased merchant onboarding.

13 Conclusion

NorthernPay (NPY) is designed to deliver a frictionless, culturally-aware payment token that eliminates on-chain transaction fees for end users via a relayer-sponsored architecture. With a conservative tokenomic framework (no premine, transparent distribution, mint/burn controls), strong emphasis on security, and a governance model that integrates Shariah oversight, NorthernPay aims to be the go-to solution for everyday digital payments within communities that value compliant financial products. The project roadmap emphasizes careful, audited rollout and practical merchant-first integrations to ensure sustainable adoption.

Appendix: Events & Contact

Implement the following events in token contract for observability and off-chain reconciliation:

```
event Mint(address indexed to, uint256 amount);
event Burn(address indexed from, uint256 amount);
event RelayedTransfer(address indexed from, address indexed to, uint256
    ↪  amount, bytes32 indexed relayTx);
```

For inquiries, audits, or governance proposals contact: info@northernstudios.cc