DENDRITES (DNDX): A Commerce-Grade Trust Layer for Predictable Fees and Safer Payments

Draft v0.1 — September 18, 2025
Authors: Dendrites Research (Delaware, USA)

Abstract

We present **Dendrites (DNDX)**, a commerce-grade trust layer that delivers **predictable end-user fees** and **safer payments**. DNDX couples a production MVP on an existing L2 with **fee-band /SLA credits**—automatic fee offsets issued when realized costs exceed a published band—and a forward R&D path, the **Stamp & Receipt Layer (SRL)**, which reserves guaranteed-inclusion capacity for a defined class of operations under verifiable resource envelopes. **On this base we expose programmable commerce primitives (UNDO, escrow, AckPay) and delivery modules—including QuickPay for instant payments, an Eco low-cost tier, and inPay for any-to-any routed commerce—implemented on today's L2 with a forward SRL path.** We formalize a quote model

$$\mathrm{q}(a,t) = \mathrm{E}[g(a,t)] + \mathrm{k}\sigma(a,t) + \mathrm{m}$$

where g is realized per-operation cost (execution + DA/proving + infra), σ its dispersion, k a risk loading, and man operational margin. **These primitives operate over a minimal Receipt Ledger—SafetySend (UNDO) for timed cancel, APP escrow (Open→Release/Refund), and AckPay (receiver acknowledgment)—with auditable invariants and bounded liabilities. AI assistants** (Co-Signer, Risk Scorer, SLA Controller, Dispute Triage) are **assistive and non-custodial**, providing pre-flight checks, anomaly scoring, fee-band forecasting, and dispute triage under human governance. Compared to prior work in account abstraction, rollup architectures, data-availability improvements, and private orderflow rails [1–7], DNDX makes **predictable cost bands** and **receipt-level safeguards** the primary user contract, enforced by credits, telemetry, and governance rather than best-effort inclusion alone. Token supply and high-level allocation are disclosed separately; **no sale schedules or pricing** appear here. The result is a measured path from volatile fees to **bounded, auditable costs** and **verifiable service guarantees** that mainstream users and merchants can trust.

Symbols & Notation

This paper uses consistent, unambiguous symbols. Greek letters are rendered as Unicode (not images). Subscripts are semantic (e.g., per-class "c" or per-week "week"). All monetary values are denominated in USD unless stated otherwise.

Symbol	Name / Meaning	Units / Domain	Notes
g	Gas price (effective)	gwei or USD/tx (when converted)	Used only as an input to realized cost.
μ_c	Mean realized cost for class c	USD/tx	Class = {Instant, Standard, Saver}.
σ_c	Dispersion (std. dev.) of realized cost for c	USD	Used to size bands.
B_c^{low}, B_c^{high}	Quoted fee band (min/max) for c	USD	What the user sees before confirm.
m_c	Margin factor for class c	unitless (≥1)	Bias to cover volatility and ops.
k	Band scaling coefficient	unitless	Controls band width aroundμ_c.
r	Realized settlement cost (post-tx)	USD	What it actually cost to execute.
δ	Delta between realized and band	USD	$\delta = max(0, r - B_c^{high})$.
$credit$	/SLA fee credit applied to user	USD	$min(\delta, c_{tx}^{cap}, \kappa_{week} - W)$.
c_{tx}^{cap}	Per-transaction credit cap	USD	Hard ceiling per tx.
κ_{week}	Weekly credit budget (protocol)	USD/week	Moving window budget.
W	Credits already granted this week	USD	Resets/rolls per policy.
τ_{undo}	SafetySend UNDO window	seconds	Default: 180s (configurable).
λ	Admission rate (SRL lanes)	tx/s	Used in 2026+ SRL L2 section.
A	Admission policy (set/function)	—	Gate for lanes & credits.
R	Receipt ledger state	enum	{Open, Released, Refunded}.

Symbol	Name / Meaning	Units / Domain	Notes
ρ	Refund ratio	0–1	Portion returned on policy trigger.
π	Paymaster sponsorship ratio	0–1	Fraction of gas sponsored.
χ	Safety flags bitmask	bitfield	Set by AI/heuristics.
Θ	Weekly safety throttle	tx/week	Anti-abuse guardrail.

Band sizing (reference):

$$B_c^{low} = \mu_c - k \cdot \sigma_c, B_c^{high} = \mu_c \cdot m_c + k \cdot \sigma_c$$

Credit rule (reference):

$$credit = min(max(0, r - B_c^{high}), c_{tx}^{cap}, \kappa_{week} - W)$$

UNDO window: Transactions remain cancel-eligible for τ_{undo} seconds unless finalized or explicitly waived by the user.

Important boundary: Credits are fee offsets, not insurance.

1. Introduction & Related Work

1.1 Motivation

Open blockchains provide neutrality and finality, but everyday commerce still experiences two hard UX failures: **(i) fee volatility** (users can't predict what they'll pay) and **(ii) coordination risk** (refunds, disputes, and staged releases are clumsy or off-chain). For buyers and merchants, "cheap on average" is not enough; they need **bounded fees** and **verifiable rules** for cancel, refund, and milestone release.

1.2 UX gaps we target

- **Unbounded cost variance.** Even when median fees are low, tails (surges) erode trust and cart conversion.
- **Best-effort inclusion.** Users lack guarantees on *when* an operation gets included.
- **Ad-hoc safeguards.** Refunds/escrows live in centralized support tickets or brittle multisigs.
- **Opaque risk handling.** Fraud checks and triage are manual and inconsistent.

1.3 DNDX approach (high level)

DNDX treats **predictable cost** and **receipt-level safety** as first-class protocol outcomes:

- **Predictable fees:** Before submission, the user sees a quote

$$\mathrm{q}(a,t) = \mathrm{E}[g(a,t)] + \mathrm{k}\sigma(a,t) + \mathrm{m}$$

and we publish tiered **fee-bands**. If realized cost breaches the band, **/SLA credits** (fee offsets) are issued automatically to tighten the user's effective cost cone.

- **Receipt-level safety:** Payments run through a minimal **Receipt Ledger** with policies: **SafetySend (UNDO)** for timed cancel, **APP escrow** (Open→Release/Refund), **AckPay** (receiver acknowledgment). These are on-chain transitions with auditable invariants and bounded liabilities.
- **Forward path to guarantees:** The **Stamp & Receipt Layer (SRL)** introduces a **Guaranteed-Stamp (GS)** lane with a clear inclusion SLO N blocks for eligible actions, alongside a market lane for everything else.
- **AI, but bounded: AI Co-Signer**, **Risk Scorer**, **SLA Controller**, and **Dispute Triage** operate as **assistive** checks and forecasters under human governance; they never hold keys or move funds unilaterally.

1.4 Relation to prior work

- **Account Abstraction & Paymasters** lower UX friction and enable sponsored fees; we build on these primitives to enforce **predictable fee-bands** and credits. [1]
- **DA/Throughput improvements** (e.g., blobspace) compress average costs; we add **banding + credits** to control **variance** at the user edge. [2]
- **Rollup architectures** (optimistic/zk) supply security and scalability; DNDX layers **receipt policies** and (future) **GS inclusion SLOs** on top. [3–5]
- **Private orderflow / MEV-aware routing** reduce some griefing; we complement this with **explicit fee bands** and **credit enforcement** when reality diverges. [6,7]

1.5 Contributions

1. **Predictable-fee framework** with user quotes, published bands, and automatic **/SLA credits** for overages.
2. **Receipt-centric commerce primitives** (UNDO, escrow, AckPay) with clear state transitions and bounded liabilities.
3. **SRL design direction** introducing a **Guaranteed-Stamp** lane with inclusion SLO N for eligible classes, enforced via admission envelopes.
4. **AI-assisted safeguards** that are non-custodial and auditable (pre-flight checks, anomaly scoring, fee forecasting, dispute triage).
5. **Solvency & abuse bounds** via weekly budgets W, per-wallet caps, and public telemetry—turning UX promises into measurable service objectives.

2. System Overview

2.1 Actors & Roles

- **Users** — payers/receivers using self-custody wallets or embedded app UIs.
- **Merchants/Creators** — integrate escrow/AckPay for orders, gigs, subscriptions.
- **Sequencer/Relayer** — L2 execution; may expose private orderflow lanes.
- **DNDX Core Services**
    - **Receipt Ledger** (contracts): escrow/UNDO state machine and events.
    - **Quotes & /SLA Oracle** (off-chain → on-chain feeds): bands, parameters, snapshots.
    - **Credits Module**: non-transferable fee-offset credits.
    - **Telemetry**: /status (uptime/incidents), /sla (bands, histograms).
- **AI Assistants (bounded)** — Co-Signer, Risk Scorer, SLA Controller, Dispute Triage; **non-custodial**, human-overridable.

2.2 Environments & Trust Assumptions

- **MVP (Today):** Deployed on a battle-tested L2 with AA/paymaster support. We inherit L1 finality and L2 liveness assumptions.
- **SRL Path (Future):** Adds a **Guaranteed-Stamp (GS)** lane (inclusion SLO N blocks) alongside a **Market Lane (ML)**. GS admission enforces resource envelopes (gas limit per op, max payload), enabling predictable inclusion for eligible actions without weakening base safety.

2.3 User-Visible Operations (Core API Surface)

- Pay(amount, to) — immediate payment (**QuickPay** variant).
- SafetySend(amount, to, T) — Pay with **UNDO** window T.
- EscrowOpen(params) — create **Receipt R** with policy (Timed, Passcode/Preimage, AckPay).
- EscrowRelease(R) / EscrowRefund(R) — terminal transitions.
- AckPay(R, ack_signature) — receiver acknowledges (EIP-712/1271) before Release.
- GetQuote(action) — returns fee quote $q(a, t)$ and active **band**.
  Delivery Modules: QuickPay (instant pay), **Eco** (low-cost/longer latency band), **inPay** (any-to-any routed commerce; post-MVP).

2.4 Lifecycle Examples (Concrete)

A) QuickPay with UNDO

1. **GetQuote** → UI shows all-in price (includes gas via paymaster + margin).
2. **Submit** → paymaster sponsors gas; tx lands; timer T starts.
3. **Cancel** (within T) → Refund path to payer; after T → finalize to receiver.

4. **/SLA check** → if realized cost g > band high + δ, issue credit C (fee offset).

B) APP Escrow with AckPay

1. **Open** → EscrowOpen emits ReceiptOpened(R, policy=AckPay).
2. **Deliver** → receiver signs EIP-712 ack (or EIP-1271 contract-based).
3. **Release** → anyone calls EscrowRelease(R) with ack; funds to receiver.
4. **Dispute** → optional timer or documentation prompt; Refund if conditions met.

2.5 Predictable-Fee Model & /SLA Credits

- **Quote:** $\mathrm{q}(a,t) = \mathrm{E}[g(a,t)] + \mathrm{k}\sigma(a,t) + \mathrm{m}$ (expected cost + risk loading + margin).
- **Bands:** per-action intervals $\left[B_{\{low\}}, B_{\{high\}}\right]$ for **Instant** and **Eco** tiers.
- **Credits (fee offsets):**

$$\mathrm{C} = \min(g - (Bhigh + \delta),\ \kappa tx), C \geq 0$$

with per-wallet weekly cap κ_{week} and program budget W.

- **Testnet vs Mainnet:** On testnet, credits are **points/badges** (off-chain). On mainnet, credits are **non-transferable offsets** redeemable against future fees.

2.6 Receipt Ledger & Policies (FSM with Invariants)

State machine:

Open —▶ Release (funds to receiver) [Invariant: amount conserved]

|

└—▶ Refund (funds back to payer) [Invariant: single terminal outcome]

Policies (composable):

- **Timed/UNDO:** cancel permitted before T; post-T finalization.
- **Passcode/Preimage:** receiver reveals preimage $\mathrm{h}(x) = \mathrm{y}$ to claim.
- **AckPay:** receiver's EIP-712/1271 acknowledgment required before Release.
  Emitted Events: ReceiptOpened, ReceiptReleased, ReceiptRefunded, PolicyAttached, UndoWindowStarted/Ended.

2.7 AI Assistants (Assistive, Logged, Auditable)

- **Co-Signer:** validates pre-conditions (policy checks, anomaly rules). Cannot sign transactions; only gates UI/actions.
- **Risk Scorer:** $\boldsymbol{\rho} \in [\mathbf{0}, \mathbf{1}]$ per receipt; high ρ → velocity limits, holds, or manual review.
- **SLA Controller:** forecasts $\mathrm{E}[g], \sigma$ (time-series) to propose band updates; governance approves.

- **Dispute Triage:** classifies disputes; drafts evidence checklist. All decisions are human-final; assistants never hold keys.

2.8 Gas, Sponsorship & Fallback

- **Default:** user pays **only USDC**; an AA **paymaster** sponsors native gas.
- **Fallback (rare):** if sponsorship is unavailable, UI offers to proceed with a **tiny native gas** amount or retry later. This is a per-tx toggle; no architecture changes.

2.9 Transparency & Telemetry

- **/status** → uptime, incidents, quest/test counts; home page counters (aggregated).
- **/sla** → current bands, example calculations, weekly histograms of realized costs vs bands, credit issuance summaries.
- **/security** → audit badges/links, disclosure policy, security contacts.
- **Contract Registry** → addresses/ABIs, network IDs.
- **Governance Log** → parameter changes $(k, \delta, k_{tx}, k_{tweek}, W)$ with timelocks and human-readable rationales.

2.10 Governance Touchpoints (what's tunable)

- **Bands & Controller:** k, δ, update cadence, EWMA windows (details in §6).
- **Credit Limits:** per-tx cap κ_{tx}, per-wallet weekly cap κ_{week}, budget W.
- **Policy Params:** default UNDO window T ranges, AckPay timeouts, preimage expiry.
- **AI Guardrails:** threshold ranges for ρ, review routing, and logging retention.

2.11 Integration Surfaces (dev-facing)

- **HTTP/WS APIs:** GET /quote?action=..., GET /sla/bands, GET /status, GET /credits/:addr.
- **Events (contracts):** ReceiptOpened, ReceiptReleased, ReceiptRefunded, AckLogged(address,R), UndoStarted/Ended.
- **SDK Helpers:** openEscrow(params), release(R), refund(R), safetySend(to,amount,T), getQuote(action).

2.12 Non-Goals & Deferred Items

- No cross-chain asset bridging at launch; inPay routes are single-chain in MVP.
- No autonomous AI execution; assistants advise only.
- No token pricing/schedules here (separate Tokenomics paper).

3. Threat Model & Assumptions

3.1 System boundary

DNDX executes user-visible primitives (QuickPay, SafetySend/UNDO, APP escrow, AckPay) on a battle-tested L2, with quotes/fee-bands computed off-chain and published via APIs (and, where applicable, on-chain snapshots). AI assistants (Co-Signer, Risk Scorer, SLA Controller, Dispute Triage) are **advisory**—they **never** hold keys or move funds.

3.2 Trust, cryptography, and platform assumptions

- **Base security.** L1 consensus and DA guarantees hold; the selected L2 inherits L1 safety/liveness with typical rollup assumptions [3–5].
- **Wallet model.** Users hold their own keys; approvals/signatures are authentic (EIP-712/EIP-1271).
- **AA/paymaster.** Sponsored gas relies on paymaster solvency and bundler availability (ERC-4337) [1].
- **Time & clocks.** On-chain block time is the source of truth for UNDO windows T and SRL SLOs N.
- **Telemetry integrity.** Off-chain metrics are signed and periodically snapshotted (hash/Merkle) to deter tampering.

3.3 Assets to protect

- **Funds:** escrow balances, refunds, and releases.
- **User cost predictability:** alignment of realized cost g(a, t) with bands $[B_{low},\ B_{high}]$.
- **Service guarantees:** UNDO windows, inclusion SLOs (future SRL), issuance/redemption of /SLA credits.
- **State & logs:** Receipt Ledger state, audit trails, governance parameter history.
- **Privacy & fairness:** order placement, private orderflow paths where used [6,7].

3.4 Adversaries & capabilities

- **Ordering/MEV adversary.** Attempts reordering, sandwiching, or griefing inclusion latency.
- **Sybil/abuse farmer.** Seeks to farm /SLA credits or points via automation or identity churn.
- **Dispute spammer.** Floods refunds/chargebacks to degrade service.
- **Sequencer misbehavior (L2).** Censors or delays eligible ops; halts or reorgs (within L2's model).
- **Model gamer.** Crafts inputs to exploit AI thresholds (false negatives/positives).
- **Insider/ops error.** Misconfigures caps, bands, or paymaster balances.

3.5 Threats mapped to controls

- **Fee volatility → Credit enforcement.** If realized g breaches $B_{high} + \delta$, issue a fee-offset credit

$$\mathrm{C} = \min(g - (Bhigh + \delta), \kappa tx),\ C \geq 0,$$

bounded by per-wallet k_{week} and weekly budget W. Public /sla summaries deter silent drift.

- **Inclusion griefing → Private lanes + SRL path.** Use private orderflow where available [6,7]; future **SRL-GS** lane commits to SLO N blocks for eligible ops; misses trigger credits and, in SRL, objective slashing (specified in the Mechanics doc).
- **Sybil/credit farming → Rate-limits & identity signals.** Email/GitHub/phone signals, velocity caps, IP/device fingerprints, cool-downs, and anomaly scoring; governance can tighten $\kappa_{tx}, \kappa_{week}, W$ quickly (timelocked).
- **Dispute spam → Queues & costs.** Minimum evidence requirements, per-account concurrent case limits, and progressive holds for repeat abusers.
- **Sequencer issues → Pause & transparency.** Emergency pause for specific policies; status page disclosures; credits for SLO misses; optional reroute to alternative relayers.
- **Model gaming → Bounded AI.** Assistants cannot authorize fund movement; all decisions are logged; threshold ranges are governed; sampled human review provides back-pressure.

3.6 Service-level objectives (SLOs) and breach handling

- **Fee SLO (mainnet).** For action class a, publish band $[B_{low}, B_{high}]$. A breach event is$g(a,t) > \mathrm{B}_{\mathrm{high}} + \delta$. Breach ⇒ credit C as above; aggregate issuance $\leq W$ per epoch.$\boldsymbol{\sigma c}$
- **Inclusion SLO (SRL-GS).** For eligible ops, inclusion within NNN blocks. Breach ⇒ credit; SRL mechanics additionally produce proof artifacts for sequencer accountability (slashing/penalties detailed in the Mechanics spec).
- **UNDO window.** If payer requests Refund before on-chain expiry T, refund must succeed or the operation pauses with operator accountability (incident disclosure + remediation).

3.7 Residual risks (acknowledged)

- **Extreme congestion.** In severe DA/execution spikes (e.g., L1 events), credits cap user harm but cannot eliminate latency.
- **Oracle/ops errors.** Mis-set bands $\mathrm{B}.\ \mathrm{or}\ \mathrm{k},\ \delta$ parameters can over/under-issue credits; governance + timelocks + circuit breakers mitigate.
- **AI drift.** Model performance may degrade; bounded authority and review mitigate blast radius.
- **Third-party dependencies.** Paymasters/bundlers and private lanes may experience outages; the **fallback** path (native gas) maintains liveness (Section 2.8).

3.8 Security posture

- **Code & deployment.** Audits prior to mainnet MVP; incremental scope; contract registry with verified ABIs; upgrade timelocks; emergency pause on limited surfaces.
- **Bug bounty & disclosure.** Public policy, clear contact, and rewards tiers; /security page lists audit artifacts.

- **Key management.** Multisig with role separation; least privilege for ops; rotation schedules; HSMs where applicable.
- **Monitoring.** On-chain alerts for abnormal credit issuance, paymaster levels, failure spikes; off-chain SLO dashboards with signed snapshots.

3.9 Privacy & fairness

- **Minimize metadata.** No unnecessary PII on-chain; optional private orderflow reduces exploitability of intent [6,7].
- **Transparent rules.** Uniform fee schedule; no hidden waivers (related parties pay the same).
- **Data retention.** Logs kept per policy; user-visible exports for their receipts and credits.

3.10 Governance & parameter safety

Tunables $(k, \delta, N, \kappa_{tx}, \kappa_{week}, W, T)$ change via timelocked proposals with human-readable rationales and post-change monitoring. Circuit breakers revert to safer defaults if issuance or miss rates exceed thresholds.

4. Mechanisms & Methodology

4.1 Fee-Band Estimation (quotes and bands)

Let g_t be realized unit cost for action a at time t (execution + DA/proving + infra). Maintain EWMAs:

$$\mu t = \lambda \mu t - 1 + (1 - \lambda) gt, vt = \lambda vt - 1 + (1 - \lambda)(gt - \mu t)2, \sigma t = vt,$$

With $\lambda \in [0.85, 0.98]$ and initialization from the first window $(\mu_0,\ v_0)$.

User quote: $\mathrm{qt} = \mu_t + \mathrm{k}\,\sigma_t + \mathrm{m}$, where k is a risk loading and m an ops margin.

Published band for tier $\mathrm{T} \in I\mathrm{nstant}, \mathrm{Eco}$:

$$\left[B_{\{low\}}^{\{(T)\}}, B_{\{high\}}^{\{(T)\}}\right] = [\mu t - \beta_T \sigma_t,\ \mu_t + \alpha_T \sigma_t], \alpha_T > \beta_T \geq 0.$$

Safety margin: $\delta > 0$ (small) used in breach tests.

Outliers: ignore g_t above the rolling **p99** over the last **N** points (e.g., N=1000).

Cadence: recompute continuously; **publish** bands every 15 minutes.

4.2 Credit Issuance (/SLA fee offsets)

For a settled tx with realized cost g and tier T,

$$\mathrm{C} = \min\left(max\left(0, g - \left(B_{high}^{T} + \delta\right)\right),\ \kappa_{tx}\right)$$

with per-wallet weekly cap k_{week} and weekly budget W. Also enforce $\mathrm{C} \leq$ **user's fee paid**.

Testnet: credits are **points/badges** off-chain.
Mainnet: credits accrue to a balance and auto-apply to the next eligible fee.

4.3 Solvency Budget & Controller

Let V_{week} be weekly tx volume; $\mathrm{Fe}\bar{e}$ the average realized fee.

$$\mathrm{W} = \min(\alpha \cdot CreditPool,\ \beta \cdot V_{week} \cdot Fee), \alpha, \beta \in (0,1).$$

Controller trigger: if $\sum C \geq 0.8\,W$, widen bands ($\uparrow_{\alpha T}$) and/or reduce k. Circuit-breaker may pause credits on persistent spikes.

4.4 Abuse Limits (anti-gaming)

$\kappa_{tx}\ per-tx\ cap;\ \kappa_{week}$ per-wallet cap; cooldowns; email/GitHub/phone signals; device/IP fingerprints.

Flag wallets that repeatedly land just above $B_{high}^{(T)} + \epsilon$ for review.

4.5 Receipt Ledger (formal)

Receipt R = $(id, p, r, a, policy, \theta)$. **State machine:**

$$Open(R) \rightarrow \begin{cases} Release(R)(to\ r) \\ Refund(R)(backtop) \end{cases}$$

Invariants: (i) Conservation — amount a transfers exactly once; (ii) Finality — exactly one terminal event; (iii) Uniqueness — id\text{id}id unique; (iv) Auditability — Opened/Released/Refunded events include policy hashes.

4.6 SafetySend (UNDO)

Policy Timed(T): after Pay, start window T (block time).
Before T: Refund must succeed (unless paused by incident controls). After T: only Release succeeds.

4.7 AckPay (receiver acknowledgment)

Receiver provides EIP-712 (or EIP-1271) ack(R); EscrowRelease(R,\textsf{ack}) iff verify(ack,R)=true. If no ack by timeout τ, follow refund/dispute policy per θ.

Default: USDC-only UX via ERC-4337 paymaster.
Fallback (rare): if sponsorship unavailable, user can retry or proceed once with native gas; telemetry alerts ops to refill/tune.

4.9 AI Assistants (bounded authority)

Co-Signer (policy checks), Risk Scorer $(\rho \in [0,1])$**, SLA Controller (band proposals), Dispute Triage (case labeling).**
They do not hold keys or move funds; thresholds are governed; all decisions logged.

4.10 Testnet → Mainnet

Testnet: quotes/bands + points; publish weekly histograms on /sla.
Mainnet: enable credits; set $(W, k_{tx}, k_{week}, k, \delta)$ via governance; publish issuance summaries.

5. Economics

5.1 Cost model (per operation)

For an action a at time t, realized cost decomposes as:

$$g(a,t) = g_{exec}(a,t) + g_{DA}(a,t) + g_{infra}(a,t),$$

where g_{exec} is L2 execution (incl. L1 inclusion amortization), g_{DA} covers blob/data availability amortization, and g_{infra} includes relays, monitoring, and support. Quotes and bands are computed over g (§4.1).

Displayed quote (all-in):

$$q(a,t) = \mu t + k\sigma t + m,$$

With $\mu t = E[g]$, dispersion σ_t, risk loading k ≥ 0, and ops margin m ≥ 0. Users see and pay q (in USDC); if sponsorship is active, native gas is paid by a paymaster (§4.7).

5.2 Fee tiers & monetization

We expose two public tiers (governed; names illustrative):

- **Instant — tighter band $[B^{I}_{low}, B^{I}_{high}]$, faster target latency; higher expected margin m_I.**
- **Eco — wider band $\left[B^{(E)}_{low}, B^{(E)}_{high}\right]$, latency-tolerant; lower m_E.**

Monetization is embedded in m and visible in q. No separate "gotcha" fees. The tier schedule (definitions, example calculations, and governance process) is published on /sla and /fees.

5.3 /SLA credit liability (definition & bounds)

A credit is a fee offset (non-transferable) issued when realized cost breaches the user's band (§4.2). For a tx settled at cost g under tier T:

$C = min(max(0, g - (B^{(T)}_{high} + \delta)),\ \kappa_{tx}), 0 \le C \le$ **fee paid.**

Program-level constraints:

- **Per-wallet rolling cap: k_{week}.**
- **Epoch budget: W (weekly, §5.5).**
- **Controller: if ∑ C ≥ 0.8W in current epoch, widen bands / reduce k (timelocked; §4.3).**

Accounting: We track credited vs redeemed balances. Outstanding credits constitute a deferred fee liability against future gross margin.

5.4 Unit economics (per tx expectations)

Let p be the price user pays (USDC), g the realized cost, and C any credit issued for that tx. Then:

- **Gross margin before credits: $M_{pre} = p - g$.**

- **Effective margin after credits (recognizing liability): $M_{eff} = p - g - E[C^{\wedge}]$, where $C^{\wedge}$ is expected redeemed credit from this tx (credit breakage = unredeemed portion).**

Design objective: choose $(k, m, \alpha_T, \beta_T, \delta)$ so that in steady state:

$$E[M_{eff}] \geq M_{min} > 0$$

for both Instant and Eco tiers, while breach probability and credit issuance remain within governed limits (§5.6).

5.5 Weekly budget W & solvency

Let V_{week} be weekly tx count for the program and $\overline{Fee}$ the average realized user fee. Define:

$$W = min(\alpha \cdot CreditPool,\ \beta \cdot V_{week} \cdot \overline{Fee}), \alpha, \beta \in (0, 1),$$

where CreditPool is a ring-fenced reserve (USDC) dedicated to fee offsets. Policy:

- **Issuance discipline: $\sum_{week} \quad C \leq W$.**
- **Controller trigger: at ∑ C ≥ 0.8W, governance widens bands ($\uparrow_{\alpha T}$), lowers k, or throttles eligible classes until issuance subsides.**
- **Circuit breaker: pause credits if anomalies persist (incident disclosure on /status).**

Runway: with current CreditPool P and budget factor α, minimum guaranteed runway R (weeks) under worst-case full issuance is:

$$R \geq \left[\frac{P}{\alpha P}\right] = \left[\frac{1}{\alpha}\right].$$

(Example: α = 0.10 ⇒ R ≥ 10 weeks of coverage at worst-case issuance.)

5.6 Breach rate targets & band calibration

For a tier T , let breach indicator $b_t = 1\{g_t > B_{high}^T + \delta\}$. *Targets:*

$$\pi_T = E[b_t] \leq \pi_{max}^T,$$

With $\pi_{max}^{(I)} < \pi_{max}^{(E)}$. Controller adjusts $(k, \alpha_T, \beta_T)$to keep π_T within target and maintain $M_{eff} \geq M_{min}$.

Outlier clipping at rolling p99 stabilizes updates.

5.7 Latency–cost trade (Instant vs Eco)

Let expected latency for Instant/Eco be L_I,L_E with $L_I < L_E$. We price via m and band width:

- **Instant: smaller α_I (tighter band), higher m_I.**
- **Eco: larger α_E , lower m_E .**
  Users self-select; the controller keeps both tiers solvent under stochastic cost regimes.

5.8 Credit redemption dynamics (breakage)

Let $r \in [0, 1]$ be redemption rate over a horizon H (fraction of issued credits actually redeemed). Then net credit cost over H:

$$NetCostH \approx r \cdot \sum CH.$$

Design levers that lower r without harming users: expiry windows, per-account caps, and minimum spend thresholds (published on /sla). All levers are governed and timelocked.

Adversary tries to farm C by timing peaks. Our limits ($\kappa_{tx}, \kappa_{week}$, velocity, identity signals) bound expected extraction. Let A be the attacker's feasible weekly tx count after rate limits, and C^{-} the mean credit per farmed tx; then:

$$MaxExtractweek \leq min(A \cdot C^{-}, \kappa_{week}),$$

and ecosystem-wide issuance still respects W. High-frequency edge cases (hovering just above B_{high}) are flagged to review; bands may widen temporarily (§4.4).

5.10 Paymaster economics

When sponsorship is active, DNDX pays native gas; users pay in USDC. Let g_{gas} be native gas cost for a class a. Paymaster solvency condition:

$$E[m] \geq E[g_{gas}] + overhead,$$

over the action mix. If bundlers/paymaster are degraded, fallback to native-gas path (§4.7) preserves liveness; quotes clearly indicate path.

5.11 Treasury & reserves (policy)

- CreditPool segregation: fee-offset reserve segregated from operating cash.
- Denomination: hold reserves in USDC-quality stables to match liabilities.
- Rebalancing: periodic top-ups based on W, breach rates π_T , and redemption r.
- Disclosure: weekly issuance, redemption, and pool levels summarized on /sla; parameter changes logged on /governance.

5.12 Revenue recognition (principle)

Revenue is recognized on settlement of a tx at price p, net of expected credit redemption $E[C^{\wedge}]$. Outstanding credits are carried as a contract liability until redemption or expiry (accounting policy to be finalized with auditors; this paper states product mechanics, not financial statements).

5.13 Sensitivity & stress

- Cost spike: If g shifts up by Δ, bands widen on the next publish; transient breaches are absorbed by W (within controller bounds).
- Volume surge: $Vweek \uparrow V$ increases second term of W, but caps/velocity checks limit per-user extraction; monitoring raises early alerts.

- **Extended volatility: controller lowers k and widens α_T to reduce breach probability; Instant/Eco spreads may temporarily adjust (governed, timelocked).**
- **Liquidity squeeze: circuit breaker pauses credits (not core operations), with disclosure on /status; USDC pricing of q remains live.**

5.14 Guarantees (economic)

Under published parameters and while credits are enabled:

1. **Bounded user harm: for any settled tx, excess cost over the band is offset by C up to κ_{tx} and subject to κ_{week} .**
2. **Program solvency: weekly issuance does not exceed W; controller actions maintain $M_{eff} \geq M\_min$.**
3. **Transparency: issuance, redemption, pool levels, and parameter changes are publicly summarized; breaches and pauses are disclosed.**

6. Security & Assurance

6.1 Objectives & scope

Security for DNDX covers (i) smart-contract correctness (Receipt Ledger, Credits, registry), (ii) AA/paymaster safety, (iii) issuance of /SLA credits, (iv) oracle/telemetry integrity for quotes and bands, (v) key management and operational controls, and (vi) incident response and disclosure. Threats and assumptions are defined in §3; this section specifies the corresponding controls, audits, and guarantees.

6.2 Contract surface & minimality

- **ReceiptLedger — escrow FSM with open, release, refund, policy hooks (Timed/UNDO, Passcode/Preimage, AckPay); events for audit.**
- **Credits — non-transferable fee-offset balances; issue, redeem, per-tx and per-wallet caps; epoch budget checks.**
- **Registry — addresses/ABIs, network IDs, versioning; append-only, timelocked updates.**

Design rules:

- **Single-responsibility modules; no upgradable proxies on funds-holding paths in MVP (upgradeability via facade/router with timelocks).**
- **Pull payments where feasible; zero external calls in state-critical sections.**
- **Reentrancy guards; checks-effects-interactions pattern; bounded loops; constant-time comparisons on preimages.**

6.3 Account Abstraction (AA) & paymaster safety

- **Sponsorship policy: per-app, per-userOp caps; rate limits; daily spend ceilings; deny-lists; allow-listed entrypoints only.**

- **Liquidity controls: automated top-ups; alarms at 30% / 15% thresholds; dual signer approval for transfers > X.**
- **Graceful fallback: if canSponsor(tx)=false, UI offers native-gas path (§4.7). No silent sponsorship of unknown calldata.**
- **Abuse protection: device/IP fingerprints; velocity windows; abnormal refund/UNDO patterns route to manual review.**

6.4 /SLA credits integrity

- **Deterministic issuance: for settled action a at cost g, tier T,**

$$C = min(max(0, g - (B_{high}^{T} + \delta)),\ \kappa_{tx}), C \leq fee\ paid.$$

- **Guards: per-wallet κweek\kappa_{\mathrm{week}}κweek; epoch budget W (enforced on-chain or by a circuit that refuses issuance once W is reached).**
- **Audit trail: CreditIssued(addr, a, g, band, C, epoch); weekly Merkle roots of issuance sets, anchored on-chain for public verification.**
- **Redemption: offsets fees only; no cash out; expiry windows and redemption caps are governed/transparent.**

6.5 Quotes/bands oracle integrity

- **Derivation: EWMA (μ_t, σ_t) (§4.1) with p99 outlier clipping and fixed publish cadence (e.g., every 15 min).**
- **Publication: signed payloads with key rotation; GET /quote includes signature + monotone publish_id.**
- **Anchoring: periodic (e.g., hourly) Merkle roots of published bands anchored on-chain to prevent retroactive manipulation.**
- **Rollbacks: clients reject stale or non-monotone publish_id; governance log records parameter changes ($k, \alpha_T, \beta_T, \delta$) with timelocks.**

6.6 Key management & authorization

- **Multisig for treasury, registry updates, paymaster top-ups; threshold ≥2/3 with role separation.**
- **Operational keys in HSM or equivalent; production secrets in sealed vaults; no hot keys on developer laptops.**
- **Rotation & recovery: documented rotation cadence; social-recovery only for non-custodial app keys; break-glass procedures with post-mortem reporting.**

6.7 Upgrade, config & governance safety

- **Timelocks: ≥48–96h on registry and parameter changes; emergency pause limited to specific surfaces (policy enable/disable, new issues of credits) with public disclosure.**

- **Staged rollouts: testnet ↔ canary ↔ mainnet; feature flags; kill switches on experimental policies.**
- **Config provenance: signed config bundles; hash pinned in the registry; all changes human-readable and archived.**

6.8 Audits, testing & formal methods

- **Audit-first surfaces: ReceiptLedger, Credits, paymaster policies, registry.**
- **Testing: 100% branch coverage on core FSM; fuzzing (property-based) for edge-case state transitions; invariant tests (conservation, single-terminal outcome, uniqueness).**
- **Symbolic analysis: run Slither/Foundry/echidna tools; SMT checks for critical invariants (no double-spend, no stuck funds).**
- **Pre-deploy checklist: compiled bytecode hash pinned; constructor args recorded; addresses verified; Etherscan source verified.**

6.9 Dependency & supply-chain security

- **Version pinning (semver locks); SLSA/SBOM artifacts for builds.**
- **Third-party libraries: permissive licenses; audit history; avoid exotic assembly unless reviewed.**
- **CI/CD hardening: signed artifacts; protected branches; mandatory reviews; no direct pushes to release.**

6.10 Monitoring, detection & telemetry

- **On-chain alerts: abnormal CreditIssued rates; paymaster balance drops; revert spikes; UNDO/refund anomalies; time-to-inclusion regressions.**
- **Off-chain SLOs: /status exposes uptime, incidents; /sla shows histograms of realized vs bands; issuance/redemption summaries each epoch.**
- **Forensics: append-only logs; request/response signing; data retention with privacy controls (§6.12).**

6.11 Incident response & disclosure

- **Severities: Sev-0 (funds at risk), Sev-1 (service breach), Sev-2 (degradation).**
- **Targets: acknowledge ≤1h (Sev-0/1), mitigate ≤24h (Sev-0), RCA ≤72h with remediation plan.**
- **Playbooks: pause limited to affected policy/module; publish incident IDs and timelines on /status; credit policy may be relaxed temporarily to compensate users.**
- **Third-party coordination: with L2 teams, bundlers, RPCs as needed; post-incident joint summary when appropriate.**

6.12 Privacy & data protection

- **Data minimization: no unnecessary PII on-chain; optional private orderflow; off-chain data stored under least-privilege.**
- **User exports: receipts/credits exportable; dispute data redacted per policy.**
- **Compliance posture: region-aware storage and processing; DPA templates for enterprise customers.**

6.13 Bug bounty & responsible disclosure

- **Scope: contracts (ReceiptLedger, Credits, Registry), paymaster policies, quote oracle signing.**
- **Tiers: Critical (funds extraction / invariant break), High (credit over-issuance bypass), Medium (DoS on core flows), Low (info leaks).**
- **Process: report via dedicated channel (PGP); triage ≤72h; payouts in USDC/DNDX; public hall of fame.**
- **Non-qualifying: known issues, rate-limited spam, social-engineering attempts.**

6.14 Policy-specific hardening

- **Timed/UNDO: refund before T must succeed unless pause is active; T derived from block time; all expiries compare against canonical chain time.**
- **Passcode/Preimage: one-way hash (keccak256); constant-time checks; preimage length bounds.**
- **AckPay: EIP-712/EIP-1271 verification; explicit domain separators; replay protection via receipt IDs + chain IDs.**

6.15 Economic & credit-program safety

- **Budget enforcement: epoch issuance $\sum C \leq W$; controller widens bands or reduces k at 0.8 W (§4.3, §5.5).**
- **Caps: $\kappa_{tx}, \kappa_{week}$ enforced at issue time; redemption never exceeds outstanding balance; expiry schedules governed and disclosed.**
- **Abuse analytics: flag wallets clustering at $B_{high}^{T} + \epsilon$; velocity/block-density heuristics; manual review queues.**
- **6.16 Interop & third-party integrations**
- **RPC/bundlers: multi-provider failover; health checks; per-provider circuit breakers.**
- **Bridges/CCTP (deferred): not in MVP; when added, risk-scoped with allow-listed routes and caps.**
- **Partners: SLAs define responsibilities; no privileged fee waivers (related-party apps pay standard fees).**

- **6.17 Residual risks**
- **Macro-congestion: extreme L1/L2 events can delay inclusion; credits bound user harm but cannot remove latency.**
- **Oracle/operator error: misconfigured bands can over/under-issue credits; mitigated via timelocks, reviews, and public anchoring.**
- **Model drift: AI assistants may degrade; bounded authority + human override limit blast radius; periodic evaluations required.**

6.18 Public artifacts (& where to find them)

- **/security: audit reports, bounty policy, contact keys.**
- **/status: uptime, incidents, post-mortems.**
- **/sla: current bands, issuance/redemption summaries, histograms.**
- **/governance: parameter changes $(k, \alpha_T, \beta_T, \delta, k_{tc}, \kappa_{week}, W)$, timelock windows, rationale.**

7. Governance & Transparency

Aim. Make risk-affecting changes predictable and publicly auditable while keeping this Whitepaper free of numeric knobs. DNDX publishes what is changing, why, and when, and anchors key artifacts for independent verification.

7.1 What's governed (surface only)

- **Bands & Controller. Quote/band publish cadence; EWMA windows; qualitative rules that reduce breach probability (actual gains/thresholds are governed configs, not in this paper).**
- **Credit Program. Per-tx issuance cap c_cap_tx, per-wallet weekly cap κ_week, and weekly budget W (mechanics here; parameters governed).**
- **Policy Timers. Default UNDO window τ_undo ranges and AckPay timeouts.**
- **Key Ops. Registry updates, paymaster top-ups, and similar operational authorizations.**

7.2 Change process & timelocks

All governance-scoped changes are proposed with a human-readable rationale, queued behind a timelock, and monitored after execution. Clients reject non-monotone or stale publishes (see §6.5). The /governance page lists pending and executed changes with ETAs.

Circuit breakers. If issuance or miss rates exceed governed thresholds, the controller widens bands and/or reduces risk-loading k; in severe cases, credits pause while core payment paths stay live.

7.3 Public artifacts & logs

We maintain stable public endpoints and on-chain anchors:

- **/status — uptime, incidents, and post-mortems.**
- **/sla — current bands, issuance/redemption summaries, histograms.**
- **/security — audit reports, bounty policy, contact keys.**
- **/governance — parameter-change log with timelock ETAs.**
- **/risk — limits & non-promises (credits are fee-offsets; not insurance).**
- **Contract Registry — addresses/ABIs and network IDs (append-only).**

Anchors. Periodic Merkle roots for (i) fee-band publishes and (ii) credit-issuance sets are recorded on-chain for independent verification.

7.4 Economic program safety (governed envelope)

The /SLA credit system enforces: (i) deterministic issuance per settled operation and tier, (ii) per-wallet weekly caps κ_week, (iii) a weekly program budget W, and (iv) redemption as fee-offset only (no cash/USDC payout). Governance can adjust the envelope (caps, budgets, expiries) via timelocked proposals; issuance sets are auditable via on-chain roots.

7.5 Incident handling & disclosures

Sev-0/1 targets: acknowledge ≤ 1h, mitigate ≤ 24h, publish RCA ≤ 72h with remediation steps. Pauses are scoped to the affected module/policy; temporary policy relaxations may be used to compensate users via credits.

7.6 Roles & key management (summary)

Treasury/registry/paymaster actions use multisig with role separation (≥ 2/3). Operational secrets reside in HSM/secure vaults; no hot keys on developer laptops. Key rotations and access boundaries are documented and auditable.

7.7 Scope boundaries (what stays out of this paper)

This Whitepaper specifies equations, guarantees, and invariants—not numeric tunables, pricing schedules, or token sale calendars. Those live in governed configs and the Mechanics/Tuning documents, with public visibility limited to the change log and observed outcomes (/sla, /status, /risk).

8. Legal, Risk & Compliance (Public-Safe Summary)

Purpose. This section clarifies what DNDX is (and is not), who may use it, how we handle risk, and where formal terms live. It is a summary only; binding terms are in the Protocol Terms, Token T&Cs, and site policies linked from /legal and /governance.

8.1 Nature of the token

- **Utility only. DNDX provides access to protocol functions (fees, credits/offsets, staking or program access where applicable).**

- **No rights. DNDX does not represent equity, debt, dividends, profit-share, voting, or redemption rights in any entity.**
- **No promise of value. Holding DNDX does not entitle holders to cash flows or protocol revenues.**

8.2 Forward-looking statements & non-promises

Plans, timelines, roadmaps, performance targets, and feature names are forward-looking and may change. Nothing herein should be construed as an offer, solicitation, or financial/investment advice.

8.3 Eligibility & geographic restrictions

Access to the token, presales, or certain features may be restricted in specific jurisdictions and to sanctioned or otherwise ineligible persons. Participation requires meeting KYC/AML, sanctions, and eligibility checks where applicable. If you are not permitted under your local laws, you must not participate.

8.4 KYC / AML & source-of-funds

Where applicable, participants must complete identity verification and provide lawful source-of-funds evidence. The issuer and its service providers may monitor, pause, or refuse participation for suspected illicit activity and will comply with lawful requests from competent authorities.

8.5 Consumer protections & "/SLA credits"

Credits issued under /SLA are fee-offsets only—not cash, not a claim on treasury assets, not insurance. Credits are non-transferable, expire per policy, and are bounded by caps/budgets governed under §7.

8.6 Protocol & market risks

Using the protocol involves risks including (non-exhaustive): smart-contract defects, L1/L2 congestion or reorgs, third-party infra outages, oracle/signing/key compromise, market price volatility, regulatory changes, and integration failures. Users should assess their risk tolerance and use only what they can afford to risk.

8.7 Operational safeguards (summary)

We use role-separated multisig for treasury/registry/paymaster actions, publish audits and security advisories on /security, disclose incidents on /status, and timelock governed changes per §7. Bug bounty scope and contact keys are published on /security.

8.8 Treasury, custody & segregation

Protocol treasuries, credit pools, and operational wallets are segregated and subject to internal controls. Custody may involve qualified third-party providers. On-chain registries list official addresses; users must verify against the Contract Registry before interacting.

8.9 Taxes

Participants are solely responsible for any taxes arising from acquiring, holding, or using DNDX or protocol services. We do not provide tax advice.

8.10 Intellectual property

"Dendrites / DNDX," "Predictable Gas," and related marks and visuals are protected by applicable IP laws. Open-source components are licensed under their respective licenses; protocol code that we open-source will include explicit LICENSE files. This Whitepaper is informational and does not create contractual obligations.

8.11 Governing terms & venue

The binding Token Terms, Protocol Terms of Use, and ancillary policies (Privacy, Cookies, Risk) prevail over this summary and specify governing law, dispute resolution, and venue. In case of conflict, those documents control.

8.12 Updates to this section

We may update this section. Material updates are logged on /governance with timelock ETAs (where applicable) and summarized on /status. Users are responsible for reviewing the current terms before participating.

9. Definitions & Notation

This section standardizes terms, symbols, and units used throughout the Whitepaper.

9.1 Core symbols (economics & policy)

Symbol	Meaning	Domain / Units	Notes
g	Realized network cost for a settled operation	currency units	Includes gas + premiums actually paid.
a	Action type (e.g., transfer, escrow.milestone.settle)	enumerated	Used to select bands/tiers.
τ	Quote tier (e.g., Instant, Standard, Eco)	enumerated	Tiers trade tightness vs margin.
$B_{high}(a,\tau)$	Published high band for (a,τ)	currency units	Upper bound of quoted range.
δ	Breach slack (de minimis)	currency units	Avoids trivial issuance for tiny misses.

Symbol	Meaning	Domain / Units	Notes
C_{tx}	Per-tx issued credit (fee offset)	currency units	$C_{tx} = max(0, g - B_{high}(a, \tau) - \delta)$, bounded.
$c_{cap_{tx}}$	Per-tx issuance cap	currency units	Governance-set envelope.
k_{week}	Per-wallet weekly cap	currency units	Governance-set envelope.
W	Weekly program budget (credit pool envelope)	currency units / week	Governance-set; enforces solvency runway.
k	Risk loading	currency units	Covers tail/variance risk.
m	Ops margin	currency units	Funds ops, audits, reserves.
$p^{\wedge}$	All-in price shown to user	currency units	$p^{\wedge} = E[g] + k + m$.
R	Credit runway (weeks)	weeks	$R \geq CreditPool/W$.

9.2 Timers & states

Symbol	Meaning	Domain / Units	Notes
T_{undo}	UNDO window	seconds	Within this, user can cancel (where supported).
T_{final}	Finalization instant	timestamp	Point after which UNDO elapses; state becomes final.
$T_{publish}$	Publish cadence for bands	seconds	Minimum interval between monotone publishes.

States (examples): quoted → submitted → settled → finalized; for escrow: open → funded → in_progress → complete/refund → finalized.

9.3 Governance & integrity primitives

Term	Meaning
publish_id	Monotone identifier for each public band/config publish; clients reject stale/non-monotone updates.
Timelock	Minimum delay between queued governance action and execution.
Anchors	On-chain Merkle roots periodically committing to (i) fee-band publishes and (ii) credit-issuance sets.
Contract Registry	Append-only list of official contract addresses/ABIs per network.

9.4 Programs & pages

Name	Purpose
/sla	Current bands, issuance/redemption summaries, histograms.
/status	Uptime, incidents, post-mortems.
/security	Audits, bounty scope, contact keys.
/governance	Parameter-change log with ETAs; past/proposed actions.
/risk	Limits & non-promises (credits are fee-offsets; not insurance).

9.5 Actors & roles

Actor	Role
User	Initiates actions; receives quotes and potential credits.
Protocol	Quotes, settles, issues credits within governed envelopes.
Paymaster	Sponsors/settles network fees per policy.
Controller	Adjusts bands/risk loading under governance (timelocked).
Governance	Proposes/queues/executes parameter changes; publishes artifacts.

9.6 Safety boundaries (non-promises)

- **Credits are fee-offsets only (non-transferable; no cash/USDC payout).**
- **Quote bands are published with auditable cadence; they are not insurance.**
- **Numeric parameters $(e.g., W, \kappa_{week}, c_{cap_{tx}}, k, m)$ are governed; values are not part of this Whitepaper.**

9.7 Notational conventions

- **Currency amounts are shown in the settlement currency unless explicitly labeled.**
- **Bold caps = published values (e.g., B_{high}); greek letters = tunables/envelope terms.**
- **"Operation" covers both single transfers and multi-step flows (e.g., escrow milestone settle).**
- **Expectations E[·]are over recent windows (EWMA) unless specified otherwise.**

10. References

This Whitepaper cites public standards, background research, and DNDX public artifacts. Final URLs are inserted at publication time; identifiers below are stable.

10.1 Primary standards & specs

[1] EIP-4337: Account Abstraction via EntryPoint & Bundlers.
[2] EIP-712: Typed Structured Data Hashing and Signing.
[3] EIP-1271: Standard Signature Validation Method for Contracts.
[4] ERC-2612: Permit—712-signed approvals for tokens.
[5] Merkle Trees (RFC/standard primer) for commitment schemes.
[6] OP Stack / Rollup Design Docs (execution, derivation, batcher).
[7] Ethereum Yellow Paper (latest canonical revision).

10.2 Paymasters, quotes & fee markets

[8] Paymaster design notes and security considerations (AA ecosystem write-ups).
[9] Gas price oracles and fee estimation methodologies (ETH/Geth docs).
[10] Envelope pricing and risk-loading in congested networks (industry primers).

10.3 MEV, ordering & integrity

[11] Flashbots / MEV-Boost design overview.
[12] Private orderflow / routing literature (builder–relayer separation).
[13] Proposer-Builder Separation (PBS) research notes.

10.4 L2 safety & rollup ops

[14] Fault/zk rollup security models (challenge windows, censorship, reorg risk).
[15] Cross-domain message passing & bridge threat models (general survey).
[16] Data availability primers (EIP-4844 / blobs overview).

10.5 Audits, disclosure & ops

[17] Secure key management & HSM guidance for crypto ops (NIST/industry).
[18] Incident disclosure best practices (RCA templates / sev taxonomy).
[19] Bug bounty program design (scope, safe-harbor norms).

10.6 Policy, KYC/AML & risk

[20] Sanctions/KYC/AML program baselines for token distributions (industry guidance).
[21] Consumer risk disclosures for crypto services (non-insurance, non-deposit).

10.7 DNDX public artifacts (to be hosted at publication)

[22] /security — audits, bounty policy, contact keys (anchor: audit hashes).
[23] /status — uptime, incidents, RCAs.
[24] /sla — current bands, issuance/redemption summaries, histograms.
[25] /governance — parameter-change log, timelock ETAs, executed actions.
[26] /risk — limits & non-promises (credits are fee-offsets; not insurance).
[27] Contract Registry — addresses/ABIs per network (append-only).
[28] Mechanics Manual — governed tunables and controller details (non-whitepaper).
[29] SRL Lanes Spec — roadmap notes for post-2026 R&D (separate document).

Appendix A — Worked Examples

These examples illustrate quotes, breaches → credits, and solvency/runway using the symbols from §9.

A.1 Instant vs. Eco quote (same action, different tiers)

Given

- **Action a = transfer.simple**
- $E[g] = \$0.38$
- **Risk loading k=$0.04 (Instant), k=$0.01 (Eco)**
- **Ops margin m=$0.03 (Instant), m=$0.01 (Eco)**
- **Published highs: $B_{high}(a, Instant) = \$0.50$, B_{high}(a,Eco)=$0.60**

Displayed price

$$P\text{^}Instant = E[g] + k + m = \$0.45$$
$$p\text{^}Eco = \$0.40$$

Interpretation
Instant is tighter + slightly pricier; Eco is wider + cheaper. Users choose predictability vs. price.

A.2 Breach → credit issuance (single transaction)

Given

- **Settled realized cost g=$0.67**
- **Tier: Instant → $B_{high} = \$0.50$**
- **Breach slack δ=$0.02**
- **$Per - tx\ cap\ c_{cap_{tx}} = \$0.10$**

Credit
$Ctx = max(0, g - B_{high} - \delta) = max(0, 0.67 - 0.50 - 0.02) = \$0.15.$
Apply cap → $issued = min(\$0.15, \$0.10) = \$0.10$

Redemption
Credits are fee-offsets only and apply to future protocol fees until expiry.

A.3 Weekly caps & budget envelope

Given

- **User already redeemed $0.80 credits this week**
- **Weekly per-wallet cap κ_{week}=$1.00**
- **New issuance computed by rule: C_{tx}*=$0.30**

Effective issuance
Remaining headroom: κ_{week}−0.80=$0.20
Issued this tx =min($0.30,$0.20) = $0.20

Program budget check
Weekly budget W=$25,000. If cumulative issuance ∑C reaches 0.8W=$20,000, controller widens bands / reduces k per policy until issuance stabilizes.

A.4 Solvency & runway

Given

- **CreditPool balance = $200,000**
- **Weekly budget W=$25,000**

Runway
$R \geq \frac{\$200,000}{\$25,000}$ = 8 weeks of credits at current envelope.

A.5 Escrow flow with UNDO and finalization

Scenario

- **User funds escrow milestone; tier = Standard**
- **UNDO window τ_{undo}= 180 seconds**
- **Finalization instant $t_{final} = t_{settle} + \tau_{undo}$**

Outcomes

- **If the user cancels before t_{final} → refund path engages (subject to policy).**
- **After t_{final} → state final; standard dispute paths apply (no UNDO).**

A.6 Price envelope tightening after a calm week (illustrative)

Given (previous week)

- **Miss rate well below threshold; issuance $\sum C$ only 12% of W.**

Controller action (governed)

- **Slightly reduce B_{high} for common actions (tighter quotes)**
- **Keep k unchanged; review again next publish window.**

Effect

- **Users see narrower ranges; solvency remains well within envelope.**

A.7 Batch effect: many small ops vs. one large

Given

- **Ten micro-ops (each with small variance) vs. one large op with higher variance.**
- **Same expected cost $E[g]$ overall.**

Observation

- **Envelope pricing may present a lower aggregate k across micro-ops (diversification), while a single large op retains a higher k to respect tail risk.**

Appendix B — Threat Model → Controls Matrix

This appendix enumerates key risks and maps them to concrete controls, anchors, and residual risk notes. Scope covers protocol contracts, paymaster, pricing/bands, credits issuance, and ops/disclosure.

B.1 Severity scale (summary)

- **Sev-0 critical loss/custody/irreversible errors; Sev-1 major degradation or widespread impact; Sev-2 limited impact; Sev-3 minor.**

B.2 Controls matrix

Risk (vector)	Likely Impact	Core Controls	Audit / Anchors	Residual Risk (notes)
Contract bug in core settlement/escrow	Funds loss, stuck funds (Sev-0/1)	Minimal surface area; no upgradable proxies on funds paths; staged rollouts; external audits; bug bounty; defense-in-depth invariants	Audit IDs on /security; contract addresses on Contract Registry; changelog in /governance	Non-zero; mitigated by timelocks + gradual feature flags
Paymaster misaccounting or drain	Fee sponsorship outage, credit pool stress (Sev-1)	Strict accounting; rate limits; separate treasuries; multisig for top-ups; monitor balances; emergency kill-switch limited to sponsorship (not user funds)	On-chain balances monitored; events mirrored into /status	Outage risk persists; user funds safe
Quote/band publish spoofing or staleness	Bad quotes, over-issuance of credits (Sev-2)	Monotone publish_id; client rejects stale/non-monotone; cadence limits; Merkle anchoring of publishes; EWMA sanity checks	Publish roots anchored periodically; client logs; /sla history	Timing races possible; bounded by cadence and client checks
Credit issuance inflation (logic bug)	Treasury leakage via over-credits (Sev-1)	Deterministic rule Ctx; per-tx cap $c_{cap_{tx}}$; per-wallet cap κ_{week}; weekly budget W; issuance roots anchored	Issuance Merkle roots; /sla summaries; /governance caps	Residual issuance spikes capped by $c_{cap_{tx}}, \kappa_{week}, W$
Oracle/gas estimate manipulation	Mispriced quotes, credit spikes (Sev-2)	Medianized sources; outlier trimming; controller back-	Publish diffs and band deltas	Manipulation limited by publish

Risk (vector)	Likely Impact	Core Controls	Audit / Anchors	Residual Risk (notes)
		pressure (widen bands, reduce k)	tracked; /sla	cadence + caps
L1/L2 congestion or reorgs	Delays, cost spikes (Sev-2)	Tiers (Instant/Standard/Eco) with envelopes; UNDO window τ_{undo}; post-incident band widening	Incident logs on /status; policy changes via /governance	Volatility cannot be eliminated; credits cushion breaches
Bridge/cross-domain faults (future SRL)	Message loss, replay (Sev-1)	Scoped: bridges optional; audits; challenge windows; delayed enables; limited allowances	Separate SRL spec; addresses in registry; audits on /security	Deferred to SRL timeline; not MVP-critical
Key compromise (ops)	Unauthorized changes/withdrawals (Sev-0)	Role-separated multisig (≥2/3); HSM/secure vaults; no hot keys on dev laptops; least privilege; regular rotations	Key policy on /security; changes timelocked on /governance	Social engineering risk persists; training & process required
Dependency outage (RPC, indexers)	Partial downtime, delayed publishes (Sev-2/3)	Multi-provider redundancy; health checks; degrade-gracefully modes	Uptime on /status	Residual brownouts possible
MEV / ordering games	Adverse fill, latency arbitrage (Sev-2)	Private routing where possible; batch tolerance; no guarantees of exact inclusion time	Method notes in Mechanics Manual	Residual market risk remains
Denial-of-Service (spam, griefing)	Degradation (Sev-2/3)	Rate limits; per-account budgets; proof-of-work/fee	Limits disclosed on /risk	Attacker cost increased; cannot fully

Risk (vector)	Likely Impact	Core Controls	Audit / Anchors	Residual Risk (notes)
		gates on heavy paths		prevent
Misuse/misrepresentation of credits	Legal/optics (Sev-3)	Clear “fee-offset only” labeling; non-transferable; expiry; caps; /risk page	Terms on /legal; summaries on /sla, /risk	Education + UI clarity required
Insider config errors	Bad params pushed (Sev-1/2)	4-eyes review; staging; timelock; automated diff checks; rollback runbooks	Change proposals on /governance; RCAs on /status	Human error reduced, not eliminated

B.3 Incident playbook (pointer)

- **Targets: acknowledge ≤1h, mitigate ≤24h, RCA ≤72h.**
- **Scope of pauses: limit to affected module (e.g., pause credits, keep payments live).**
- **Compensation: use credits (fee-offsets) within governed caps; never cash/USDC payouts via credits.**
- **Disclosure: /status post + Git-tagged diff of any policy change; anchors updated.**

B.4 Assurance artifacts (where to find them)

- **/security — audit reports (hashes), bounty scope, keys.**
- **/status — uptime, incidents, RCAs.**
- **/sla — bands, issuance/redemption histograms.**
- **/governance — queued/executed changes, timelock ETAs.**
- **Contract Registry — canonical addresses/ABIs per network.**
- **Mechanics Manual — governed tunables and controller logic (non-whitepaper).**

Appendix C — Public Artifacts & On-chain Anchors

This appendix defines the artifacts DNDX publishes, the cadence at which they’re refreshed, and how anyone can independently verify them against on-chain anchors.

C.1 What we publish (at a glance)

- **/status: uptime, incidents, RCAs (human-readable).**
- **/security: audit report hashes/ids, bounty scope, contact keys.**
- **/sla: quoted ranges, issuance/redemption summaries & histograms.**
- **/governance: queued/executed parameter changes with timelock ETAs.**
- **/risk: limits & non-promises (credits are fee-offsets; not insurance).**
- **Contract Registry: canonical addresses/ABIs per network (append-only).**
- **Anchors (on-chain): periodic Merkle roots committing to:**
    1. **Fee-band publishes (all (a, τ) entries for the window)**
    2. **Credit-issuance sets (per-tx issuance records within the window)**

C.2 Cadence & retention (normative)

- **Bands root: anchored at a fixed interval (e.g., hourly); each covers [t,t+Δ) with monotone publish_id.**
- **Issuance root: anchored at a fixed interval (e.g., hourly) and rolled up into daily and weekly summary roots.**
- **Contract Registry: anchored on change; never pruned (append-only).**
- **Retention (off-chain pages): raw CSV/JSON for at least 18 months; aggregated stats retained indefinitely.**
- **Clock tolerance: anchors must appear on-chain within Δ_anchor of the stated window end. Clients treat missing or late anchors as *yellow* (see C.6).**

Note: actual intervals (Δ, Δ_anchor) are governed and disclosed on /governance; this appendix defines behavior, not numeric knobs.

C.3 Data model (hash commitments)

We commit to canonical JSON line-items (stable field order). Hashing uses Keccak-256 of the UTF-8 bytes of the canonicalized row.

C.3.1 Fee-band publishes (example fields)

```
{
 "publish_id": "uint64",
 "window_start": "unix_ms",
 "window_end": "unix_ms",
 "action": "string",        // e.g., transfer.simple
 "tier": "string",          // Instant|Standard|Eco
```

"B_{high}": "decimal-string", // currency units

"notes": "string|empty"

}

C.3.2 Credit issuance (example fields)

{

"tx_hash": "0x..",

"ts": "unix_ms",

"action": "string",

"tier": "string",

"g": "decimal-string", // realized cost

"B_{high}": "decimal-string",

"δ": "decimal-string", // breach slack δ at issue time

"C_{tx}": "decimal-string", // issued credit (fee-offset)

"wallet": "0x..(obfuscated)", // privacy-preserving salt+hash

"policy_ver": "semver"

}

Privacy: issuance exports use salted wallet digests (per-period salts) so public proofs cannot trivially deanonymize users. Raw wallet addresses are never published in the off-chain CSV/JSON.

C.4 Merkle construction (deterministic)

- Leaves: leaf_i = keccak256(canonical_json_i)
- Ordering: sort leaves lexicographically by keccak256(canonical_json_i) (tie-break by full JSON string).
- Combiner: node = keccak256(left || right); odd leaf at any level is promoted (left-biased).
- Root commitment: stored on-chain with:
    - type ∈ {BANDS, ISSUANCE_DAILY, ISSUANCE_HOURLY, ISSUANCE_WEEKLY}
    - period_start, period_end (unix_ms)
    - publish_id (for bands) or period_seq (for issuance)
    - uri (content-addressed pointer to off-chain artifact, e.g., IPFS/HTTPS)

C.5 Verifying an artifact (step-by-step)

1. **Fetch the artifact JSON/CSV from the URI listed on /sla or /governance for the period you care about.**
2. **Canonicalize each row (stable key order, no whitespace beyond single separators).**
3. **Hash each row with Keccak-256; sort leaf hashes; build the Merkle tree as per C.4.**
4. **Compare your computed root with the on-chain root for the same (type, period_start, period_end).**
5. **Spot-check: pick a few rows and verify Merkle inclusion proofs supplied alongside the artifact (proof = branch siblings + leaf index).**

Expected outcomes

- **Match: artifact is authentic for that period.**
- **Mismatch: treat as red; see C.6.**
- **No anchor yet (within Δ_anchor): treat as yellow; retry after Δ_anchor.**

C.6 Discrepancies & late anchors (operator policy)

- **Yellow (late/missing within Δ_anchor): page banner on /status; no credits issuance pauses unless two consecutive periods are late.**
- **Red (root mismatch or repeated lateness): immediate banner + incident opened; credits issuance may pause (payments stay live) until corrected; RCA posted within 72h.**

C.7 Registry & audits

- **Registry anchoring: Any change to addresses/ABIs emits a REGISTRY_UPDATED event and updates the on-chain registry root. Old entries remain queryable.**
- **Audit artifacts: We publish report hashes (SHA-256) and store reports off-chain; on /security, each report shows its hash, scope, and date. Hashes are optionally included in a periodic SECURITY_ARTIFACTS root.**

C.8 Testnets vs. mainnet

- **Testnets (preview): we anchor to testnet contracts with shorter cadences for early verification.**
- **Mainnet (production): anchors follow the governed Δ and Δ_anchor; any deviation is disclosed on /status.**

C.9 Content addresses & mirrors

Artifacts are hosted at HTTPS URLs and mirrored to content-addressed storage (e.g., IPFS). The on-chain record carries both; clients should prefer content-addressed URIs when available.

C.10 Deletions & corrections

Artifacts are immutable once anchored. Corrections are published as *superseding* artifacts with a new root and a pointer to the superseded period, plus an entry in /status and /governance (change log).

Appendix D — Change-log & Versioning Template

This appendix defines how we version releases and how editors should write change-logs so external reviewers can diff what changed, when, and why.

D.1 Versioning scheme (SemVer-style, adapted)

- **MAJOR (X.0.0): breaking changes to guarantees or invariants in the Whitepaper (e.g., issuance rule form, safety boundaries).**
- **MINOR (X.Y.0): new features or non-breaking policy extensions (e.g., new tiers, new pages, new anchors).**
- **PATCH (X.Y.Z): clarifications, typo fixes, non-normative edits (no change to guarantees).**

Artifacts track their own versions and roll up into a Release ID:

- **Whitepaper: WP vX.Y.Z**
- **Mechanics Manual: MM vA.B.C**
- **Contracts (per network): CT vN.M.P (tagged commit + addresses)**
- **Public Pages (/status, /sla, /governance, /risk): PUB vR.S.T**
- **Release ID (roll-up across artifacts): REL-YYYYMMDD-<shortname> (e.g., REL-20251012-Aurora)**

D.2 What requires a new version

- **MAJOR: any change to equations, definitions, or scope boundaries (credits stop being fee-offset-only; issuance rule form changes; guarantee wording).**
- **MINOR: adding a new program/page, adding a new action/tier, adding an anchor type, adding examples/appendices.**
- **PATCH: prose clarifications, formatting, fixes to figures/tables, updated references, link corrections.**

Tunable values (e.g., caps, budgets, timers) do not change the Whitepaper version; they are logged on /governance with timestamps and timelock ETAs.

D.3 Release checklist (editor workflow)

- **Assign Release ID (date + codename).**
- **Tag repos/artifacts (whitepaper PDF/Docx, Mechanics, contracts) with versions above.**
- **Update /governance with the change set and links to diffs.**

- **Publish anchors for any period artifacts touched (bands/issuance roots).**
- **Post summary on /status (if user-visible behavior changes).**
- **Regenerate References if IDs or anchors changed.**

D.4 Change-log entry template (copy/paste)

Release: REL-YYYYMMDD-<codename>

Window: 2025-09-22 (UTC) Editor: <name or org handle>

Whitepaper: WP vX.Y.Z → WP vX'.Y'.Z'

Mechanics: MM vA.B.C → MM vA'.B'.C'

Contracts: CT vN.M.P (mainnet: <tx/tag>; testnet: <tx/tag>)

Public: PUB vR.S.T → PUB vR'.S'.T'

Type: MAJOR | MINOR | PATCH

Summary:

- One-sentence summary of the change and user impact.

Details (bulleted):

- §N.N — <what changed> (non-breaking/breaking).

- /sla — <added histogram / field / note>.

- Registry — <added/updated addresses>.

Anchors:

- BANDS root [period_start → period_end, tx_hash]

- ISSUANCE_* [period_start → period_end, tx_hash]

Governance:

- Proposal IDs: GOV-#### (queued/executed)

- Timelock ETA(s): <timestamps>

Migration / Ops Notes:

- <client behavior, caches, any manual steps>

D.5 Examples (concise)

PATCH example (typo & figure fix)

- REL-20251012-Quasar — WP v1.0.2 → v1.0.3. Corrected Fig. 3 caption; no change to equations or guarantees. No anchors affected. /status not required.

MINOR example (new tier added)

- REL-20251026-Nebula — WP v1.0.3 → v1.1.0. Added **Eco+** tier (definition only). Mechanics v0.9.2 → v0.10.0 with controller notes. /governance proposal GOV-121 queued (timelock 48h). New BANDS roots anchored.

MAJOR example (equation form change)

- REL-20251130-Singularity — WP v1.1.0 → v2.0.0. Issuance rule now includes variance guard term; guarantees updated. Mechanics and contracts tags updated. /status post + migration notes required.

D.6 Diffing guidance (for reviewers)

- Prefer **tagged PDFs** for WP/MM diffs; include a plain-text export for semantic diffs.
- Contract diffs reference **commit hashes + addresses**; provide ABI diff links.
- For anchors, link to the on-chain tx and the artifact URI; provide inclusion proofs for spot checks.

D.7 Deprecation policy

- Features marked **Deprecated** remain functional for at least one full publish cycle (unless a security incident forces earlier removal).
- Deprecated items list removal target date and migration steps in /governance and /status.

D.8 Mapping table (where each change is logged)

Change type	Whitepaper	Mechanics	Contracts	/governance	/status	Anchors
Math/guarantee change	✅	✅	↔ (if code)	✅	✅	↔
New program/page	✅ (defn)	✅	❌	✅	↔	↔
Tunable update	❌	✅	❌	✅	↔	↔
Incident/RCA	❌	❌	❌	↔	✅	↔
Address/ABI change	❌	❌	✅	✅	↔	✅ (registry/root)

Legend: ✅ must update; ↔ sometimes; ❌ not applicable.