Next-Gen Store of Value

Privacy, Proofs, Compute

Private Money & AI Money:
A thesis on monetary primitives for the AI era

Jason St George

> *A cypherpunk monetary stack: privacy by default, proofs by construction, and compute that only gets paid when anyone can verify it.*

Version 1.0

2026

Abstract

Post-Bretton Woods money is increasingly backed by *compliance* and *regulatory enforcement* rather than reserves or convertibility. Debt stocks that cannot be honored in real terms make financial repression (fiscal dominance) arithmetically attractive; synthetic media and platform moderation undermine the assumption that "seeing is believing"; hardware, networks, and identity systems are progressively integrated into state and platform surveillance infrastructure. In that environment, stores of value that depend on soft guarantees (custodians, editorial gatekeepers, reputation) are brittle.

This thesis makes a concrete claim: **three cryptographic capacities can function as monetary primitives** for a dense digital civilization: **Privacy** (censorship-resistant settlement that preserves agency), **Proofs** (portable attestations of computation and provenance), and **Compute** (useful work such as matmul, inference, and ZK proving, all wrapped in succinct guarantees). We build a threat model that assumes intentional repression, not benevolence, and introduce a **seven-layer architecture** grounded in Layer 0 verifiable machines and energy, extending through communications, software distribution, identity, truth & work, value & settlement, and a telemetry/governance layer that measures drift rather than denying it.

At the economic core, we formalize **verification asymmetry** and define **VerifyPrice** (the time-and-cost vector for independently checking a claim) as the hinge that turns proofs and verified FLOPs into commodities rather than platform IOUs. We propose a **Create/Compute → Prove → Settle → Verify** loop, a modular stack of **twelve primitives** and **four reference applications** (private treasury & payroll, media provenance, verified inference, proof/compute procurement), and an operator/investor telemetry program (including extensions like **VerifyReach** and **VerifySettle**) that makes neutrality and repression-resilience falsifiable.

The result is not a single chain, but a **research and engineering agenda**: a "Bell Labs" for proof-of-useful-work and lawful privacy, aimed at making Privacy, Proofs, and Compute behave like a next-generation store of value (verifiable digital necessities that the world must keep buying, on rails anyone can audit).

Contents

Reader's Map

This thesis is structured as a ladder of seven interlocking layers. Each layer has one or more "home" sections, and most layers also surface in the **Create/Compute → Prove → Settle → Verify** loop and in the telemetry program.

Use the following table as your compass:

Layer	Role in the Stack	Primary Sections	Key Metrics / Concepts
Layer 0 — Verifiable Machines & Energy	Physical base reality: open silicon, sampled supply chains, and the power plants that keep provers and routers alive. *If the machine can lie or cannot stay on, nothing above it matters.*	§14 *Layer 0* (open hardware, profiles, lot sampling, FERs); §28.1 (L0 implementation sketches); Appendix D (Energy & Plant); Appendix E (Hardware Profiles)	Hardware profiles (HIDs); lot-sampling coverage; Facility Energy Receipts (FERs); PUE/ERE/WUE; kWh/receipt; $kgCO_2e$/receipt; profile incidents/deprecations
Layer 1 — Reachability: Communications & Transport	Keeps packets flowing under DPI, filtering, and shutdowns. *Without Layer 1, the loop cannot run and "public verification" becomes theoretical.*	§15 *Layer 1* (threats, design rules, mechanisms); §15.1–15.5 (VerifyReach); §28.1 (comms sketch); Appendix F (Comms Resilience)	**VerifyReach(N,R)**: p50/p95 time-to-first-connection; $succ_1$/$succ_2$ reachability; failure causes; share encrypted/obfuscated traffic; peer-set diversity (geo/ASN); swap success under network stress
Layer 2 — Distribution & Execution: Software Supply & Runtime	Ensures honest clients and updates can be shipped even when app stores, CDNs, and DNS are weaponized. *Protects the code that speaks the protocol and runs PaL/PRK.*	§16 *Layer 2* (threats, design rules, multi-home architecture); §16.1–16.9 (update pipeline, key mgmt, fallback playbooks); §28.2 (distribution tooling); Appendix B (update KPIs)	Update health: p50/p95 metadata & artifact fetch times; success/rollback/signature-failure rates; mirror/CDN top-N share; geographic/ASN diversity; key-rotation & incident history

Continued on next page…

Layer	Role in the Stack	Primary Sections	Key Metrics / Concepts
Layer 3 — Identity & Claims: Humans & Machines Without Doxxing	Lets humans and machines prove capabilities and rights (age, residency, uniqueness, model ownership, device profile) without turning identity into a global dossier. *Reputation is receipts, not biographies.*	§17 *Layer 3* (identity kernel, human & machine identity, reputation, guardrails); §17.9–17.11 (how identity uses the loop + SDK sketch); §24.1 (lawful-privacy requirements); Appendix C (PaL/PRK SDK)	Identity-predicate VerifyPrice (p50/p95 verify times for age/residency/set-membership proofs); unlinkability tests; reputation from PIDL receipts (SLA fulfillment vs slashes); hardware-anchored machine attestations
Layer 4 — Truth & Work: Proof Systems and PoUW	Converts expensive work into cheap-to-check receipts. *This is where verification asymmetry lives and where VerifyPrice is defined.*	Conceptual: §9 *Compute as AI Money*, §10 *Work Credits*; Structural: §18 *Infrastructure to Economics*; §19 *Layer 4* (canonical workloads, PoUW, proof factories, PaL); §21.1 (primitive catalog); Appendix A (formal VerifyPrice)	**VerifyPrice(W)** per workload: p50/p95 verify times/costs; failure rates; $r(W) = v(W)/p(W)$; prover concentration; SLA attainment (Bronze/Silver/Gold); MatMul-PoUW economics; Work-Credit issuance
Layer 5 — Value & Settlement: Privacy Rails & Non-Custodial Flow	Moves value non-custodially and privately, with auditability via receipts. *This is where "Private Money" and "AI/Proof Money" actually settle.*	Monetary: §7 *Privacy as Private Money*, §10 (Work Credits); Stack: §20 *Layer 5* (settlement as workload, PRK, bridge-safety, corridors); §21.1–21.5 (treasury/payroll, media, inference, procurement flows); §28.2 (privacy toolkit); Appendix F (swaps, routing)	**VerifySettle(C)**: swap success ≥ target; **refund_safe(C) = 1.0**; p50/p95 time-to-finality; anonymity-set size & churn; corridor LP/route concentration; non-custodial vs custodial flow share

Continued on next page…

Layer	Role in the Stack	Primary Sections	Key Metrics / Concepts
Layer 6 — Governance & Telemetry	The immune system and constitution: keeps drift and capture visible and forces responses. *Turns "trustlessness" into dashboards and runbooks.*	Metrics: §19.6–19.7 (VerifyPrice observatory); §23 *Extended Telemetry*; Governance: §22 *Layer 6* (SLOs as constitution, control surfaces, Bell-Labs vs on-chain vs norms); Legal: §24; Ops: §25 (checklist & red flags); Dynamics: §26–§27; Appendix A & B	VerifyPrice dashboards per workload; **VerifyReach** & **VerifySettle** boards; decentralization metrics (entry latency, top-N share, Nakamoto coefficients, geo/ASN spread); fee+burn coverage; repression-beta; incident reports

Thesis in Plain Language

The core claim: Three cryptographic capacities—**Privacy** (censorship-resistant settlement), **Proofs** (portable attestations), and **Compute** (verified useful work)—can function as monetary primitives for a dense digital civilization. Networks that supply these capacities at scale, with cheap public verification, can earn a durable store-of-value premium.

The mechanism: Value accrues to holders of triad assets (Work Credits, network tokens) through four channels:

1. fees paid in the native asset for proof generation, privacy settlement, and verified compute;
2. burns that permanently remove supply when capacity is consumed;
3. required collateral for provers, routers, and liquidity providers; and
4. scarcity constraints tied to energy and hardware, not fiat decree.

Demand for these capacities is structural—AI needs verified compute, commerce needs private settlement, trust needs proofs—so fee revenue persists through cycles.

The falsifiable test: If VerifyPrice (verification cost) stays low, VerifyReach (reachability under censorship) stays high, and VerifySettle (settlement success and refund safety) remains robust, the thesis holds. If any of these degrade, the asset becomes just another platform IOU.

The thesis examines the triad from three complementary angles:

- **Angle 1 — Monetary:** the triad (Privacy, Proofs, Compute) as a next-gen store of value (Private Money & AI Money).
- **Angle 2 — Stack:** the seven-layer cypherpunk stack that actually supplies these capacities.
- **Angle 3 — Telemetry & Governance:** VerifyPrice/Reach/Settle + ops as the thing that keeps "repression-resilient neutrality" falsifiable.

Key Definitions

Before proceeding, we establish precise definitions for the core constructs that recur throughout this thesis. These are not metaphors; they are operationally specified primitives.

Definition: VerifyPrice(W) — Specification Stub

For a canonical workload W, **VerifyPrice** is the public KPI vector:

$$\textsc{VerifyPrice}(W) \equiv (p_{50,t}(W),\ p_{95,t}(W),\ p_{50,c}(W),\ p_{95,c}(W),\ \text{fail}(W))$$

Where:

- $p_{50,t}(W)$, $p_{95,t}(W)$: median and 95th-percentile verification **time** (seconds)
- $p_{50,c}(W)$, $p_{95,c}(W)$: median and 95th-percentile verification **cost** (see cost vector below)
- $\text{fail}(W)$: verification failure rate (fraction of attempts that fail or timeout)

Why this matters: VerifyPrice is the hinge that determines whether proofs and verified compute behave as commodities (publicly checkable) or as platform IOUs (trust someone's claim). If $r(W) = v(W)/p(W) \ll 1$, verification is cheap relative to production and markets can form; if $r(W) \to 1$, we're back to "trust the prover."

Definition: Work Credits

A **Work Credit** is an energy-anchored claim on a standardized unit of triad work (privacy settlement, proof generation, or verified compute) that has been produced and attested under public SLOs.

Issuance: Credits are minted only when:

1. A valid proof of workload W at tier T is accepted by the network.
2. Telemetry confirms VerifyPrice(W,T) and other SLOs (latency, failure rate, decentralization) are within bounds.

Redemption semantics: Implementation-dependent. Work Credits can be designed across a spectrum:

- **Non-redeemable but scarce:** pure SoV instruments where credits represent historical work (like BTC tied to historical hashes). Value derives from scarcity and demand, not redemption rights.

- **Redeemable vouchers:** credits burnable for future proofs, compute, or settlement capacity. Provides direct utility claim.
- **Fee/collateral/governance medium:** credits required for network operations:
  - *Fee prepayment:* credit burns in lieu of per-call fees.
  - *Collateral:* credit staked as skin-in-the-game for provers, routers, and LPs.
  - *Governance weight:* credit-weighted voting in telemetry disputes and parameter changes.

These options are not mutually exclusive; a single network may support multiple redemption paths for different use cases.

Energy anchoring: Marginal cost of minting one credit is bounded below by energy and hardware required to pass verification. Unlike SHA-256 PoW, the work is *useful*. Each credit references a Facility Energy Receipt (FER) chain; if the referenced plant drifts out of profile (PUE > 1.5, carbon intensity $>$ threshold, etc.), downstream credits are flagged.

Non-debt property: Work Credits do not promise fixed coupons or redemption in fiat terms. Value floats with demand for triad capacity.

Failure mode: If VerifyPrice regresses materially, new issuance halts until SLOs recover. Existing credits remain valid but may trade at a discount, reflecting the network's degraded utility.

Definition: Lawful Privacy

Lawful privacy is the design principle: *default privacy with optional, user-controlled disclosure*.

Concretely:

- **Default state:** Transactions, identities, and flows are encrypted and unlinkable without explicit consent.
- **Disclosure mechanisms:** Viewing keys, auditable receipts, and selective-disclosure proofs allow holders to prove specific facts (e.g., "I paid X to Y for purpose Z") without exposing the full transaction graph.
- **No backdoors:** The protocol has no master keys, regulatory escrow, or "lawful intercept" APIs. Disclosure is always at the holder's discretion.

Coercion boundary: Lawful privacy is a *technical* guarantee. It cannot prevent social or legal coercion to disclose viewing keys. What it guarantees is that (1) non-custodial routes exist, (2) disclosure cannot be forced at the protocol level, and (3) coercion surface is minimized by keeping data encrypted by default.

Part I
Context & Claims

This Part sets the stage: it explains why soft guarantees (central banks, broadcast media, credentialed authority) are failing under debt, AI, and platform control, and why we need new monetary primitives. It introduces the triad (Privacy, Proofs, Compute), the threat model, and the layered "cypherpunk stack" that the rest of the thesis will fill in.

Chapter 1

Introduction

Every monetary epoch begins with an argument about what is real. James Dale Davidson and William Rees-Mogg, in their seminal work *The Sovereign Individual*, argued that the decentralization of computer networks would erode centralized power, and that governments, in their death throes, would reach for more aggressive tools of control: *capital controls, nationalization, outright authoritarianism.*

The old guarantees (central banks, broadcast media, credentialed authority) no longer hold their shape under the pressure of digital networks and foundation models. They are under attack, not by a single conspirator, but by the physics of decentralization itself, which dissolves the monopolies that once defined our shared reality.

Marshall McLuhan saw an even deeper shift: the move from print, a one-to-many medium that centralizes narrative, to electronic networks, a many-to-many mesh that fragments it. A print-created reality is a centralized reality: he who controls the press, controls the narrative. This underwrote 20th-century monetary supremacy and global settlement currency backed by a monopoly on violence: the post-1971 US petrodollar.

In the 20th century, citizens got their singular cultural feed from *The New York Times* over breakfast, and later their dualistic, pseudo-antagonistic programming from CNN and Fox News. In the 21st, the internet (many-to-many) erodes this centralization of narrative. Take the network formerly known as Twitter: authoritarian regimes that attempt to censor information find it very hard to stop leakage. Within minutes, people with "smart" phones can produce a preponderance of evidence that either supports or invalidates the prevailing story.

But a countervailing force has arrived: AI. The quality of realistic generated content has crossed a critical threshold, approaching the asymptote of believability. Synthetic faces, voices, and scenes now compete with direct sensory experience.

This creates a world wealthy in symbols but poor in anchors: a sea of abstractions with no obvious way back to sensible shores.

> So we look for new anchors.

Cypherpunk cryptography is the unifying answer: privacy by default, proof by construction, and compute that only gets paid when anyone can verify it. It replaces authority with

protocols and gossip with receipts. In this frame, money is not a promise from a platform; it is what remains when verification is cheap and permission is irrelevant. Cypherpunks don't ask for integrity; they **instrument** it. They don't trust platforms; they price receipts. The triad is that credo made economic: privacy that preserves agency, proofs that travel, and compute that earns only when anyone can verify it.

Bitcoin's SHA-256 puzzle was a brilliant bootstrap: it minted digital scarcity by tying consensus to thermodynamic cost. The next stage in the evolution of blockchains keeps what made PoW legitimate (open admission and unpredictable leader election), but swaps the work function so the "lottery ticket" is earned by producing succinct, publicly verifiable receipts of useful compute. Miners don't win by burning cycles on nonce search; they win by attaching a proof that some market-demanded computation was done correctly.

A natural anchor workload is matrix multiplication. MatMul-PoUW constructions make verification asymptotically cheaper than naive production (for example, $O(n^2)$ verification vs. $O(n^3)$ multiplication) while keeping verifier overhead at $(1 + o(1))$ relative to the best-known randomized checker. That turns AI's core primitive into **verified FLOPs**: a commodity unit anyone can check cheaply. Projects like Nockchain show the zk-PoW design space operating in the wild (fair-launch ethos, scope-minimized "dumbnet," explicit issuance), illustrating that proof-carrying work can be wired into consensus without appointing gatekeepers. Keep the lottery; change the work. Make the prize a receipt the public can verify, not heat no one can use.

> Operationally, the goal is simple: bind useful work to the hash race **without introducing trust**.

Two deployable patterns are in scope:

- **(A) Hash-gated useful work.** A SHA-family threshold confers short-lived eligibility; a block is valid only if it carries a PoUW artifact (e.g., MatMul proof or zk-proof) seeded by header randomness.
- **(B) Proof-first selection.** Miners race to post useful-work receipts to a mempool; header entropy resolves ties and timing.

Both patterns must prevent precomputation (epoch seeds, commit-reveal), avoid closed-hardware dependencies, and expose verifier-light clients plus dispute/slash routes for junk artifacts.

When blocks routinely carry proofs that clear public SLOs and decentralization telemetry remains healthy (time-to-first-proof, top-N share, geo/ASN spread), block rewards stop subsidizing waste and start underwriting capacity the world already buys: ZK proving, matrix multiplication, verified inference. That is the philosophical and operational upgrade: a puzzle that mints receipts, not heat, so **Privacy, Proofs, and Compute begin to behave like money**.

The rest of this Part translates that intuition into claims, contributions, a threat model, and

a layered architecture.

1.1 Central Claims

In this thesis we argue:

1. **Post-Bretton Woods money is compliance-backed, not reserve-backed.**
   The modern monetary system relies more on surveillance, capital controls, and regulatory force than on convertibility or reserves. When compliance becomes weaponized and asset freezing becomes policy, neutral stores of value become necessary infrastructure, not ideological luxuries.
2. **Three cryptographic capacities can function as monetary primitives.**
   Privacy (censorship-resistant settlement that preserves agency), **Proofs** (portable attestations of computation and provenance), and **Compute** (useful work wrapped in succinct guarantees) are scarce capacities the world must continually buy. Each can be engineered to be credibly scarce, permissionless, censorship-resistant, and cheap to verify.
3. **A modular stack can operationalize this triad.**
   We propose four reference applications (private treasury & payroll, media provenance, verified inference, proof/compute procurement) built on twelve primitives, with **verification asymmetry** and **VerifyPrice** as the key economic metrics that turn proofs and verified FLOPs into commodities rather than platform IOUs.
4. **Verifiable machines (Layer 0) are the precondition.**
   Without open, auditable hardware designs and sampled supply chains, the entire stack devolves to "trust the vendor." Open hardware is not ornament; it is the base layer that achieves common knowledge of security among mutually distrusting actors.

In the rest of the thesis we make this concrete in modular form. Four reference applications exercise the stack (private treasury & payroll, media provenance & authenticity, verified inference as AI service, and proof/compute procurement), supported by a small toolkit of primitives: a proofs-as-a-library SDK ("PAL") that compiles claims to proofs, a privacy-rails kit that executes non-custodial, refund-safe settlement over BTC↔ZEC/XMR corridors, a minimal receipt schema ("PIDL") that turns every proof or settlement into a portable artifact, neutral router logic that keeps useful-work markets open, and a VerifyPrice observatory that measures how cheap verification really is. Later sections develop these primitives and applications in detail; here we refer to them by name once the context is clear.

The overall thesis is simple: the next reserve asset is not a single object but a **triad of verifiable necessities**: Privacy, Proofs, and Compute. Treat this triad explicitly as cypherpunk monetary primitives:

- **Privacy**: the right to hold and move value without chokepoints (to preserve agency).
- **Proofs**: cryptographic attestations of origin, integrity, identity, or computation (to crystallize

truth).

- **Compute**: useful work (ZK proving, matrix multiplication, verified inference) that applications demand and that blockchains can verify cheaply (to power intelligence).

These are not slogans. They are the cypherpunk stack distilled into assets: what cannot be forged, what no one has to bless, and what everyone can check. They are *scarce capacities* the world must continually buy because life in a dense, digital civilization requires them. Each has its own economy; together they form a monetary base.

Institutionally, this is not just a design for another chain. It is a **research and engineering agenda**: a Bell Labs for proof-of-useful-work and the zk economy. Where the original Bell Labs turned Shannon's information theory into cables, switches, and semiconductors, the mandate here is to *turn privacy primitives, zero-knowledge, and verifiable compute into everyday infrastructure*: receipts, rails, and verifiable machines that other people build on without thinking about it. This document is the charter for that lab.

Each primitive is designed to be credibly scarce, permissionless, censorship-resistant, and cheap to verify. Where gold condensed geology and Bitcoin condensed randomness, this triad condenses the utilities of the information era into assets: things that cannot be faked and do not ask permission, cheap for anyone to verify yet costly to produce or deny.

> Networks that supply these at scale will earn a durable store-of-value premium because the world must keep buying their utility through every cycle, even (and especially) under financial repression and currency debasement.

1.2 Contributions

This thesis makes four main contributions:

1. **Threat model and layered architecture.**
   We build a threat model that assumes intentional repression rather than benevolence: financial repression (YCC, capital controls), censorship and shutdowns, hardware and identity capture, and platform-mediated reality. On top of that we propose a seven-layer "cypherpunk stack":
   - Layer 0: Verifiable Machines & Energy
   - Layer 1: Reachability (communications & transport)
   - Layer 2: Distribution & Execution (software supply & runtime)
   - Layer 3: Identity & Claims (humans and machines without doxxing)
   - Layer 4: Truth & Work (proof systems, PoUW, VerifyPrice)
   - Layer 5: Value & Settlement (privacy rails, non-custodial flow)
   - Layer 6: Governance & Telemetry (keeping neutrality and resilience measurable)

   The rest of the thesis walks this stack from silicon and power up through proofs, settlement,

and governance.

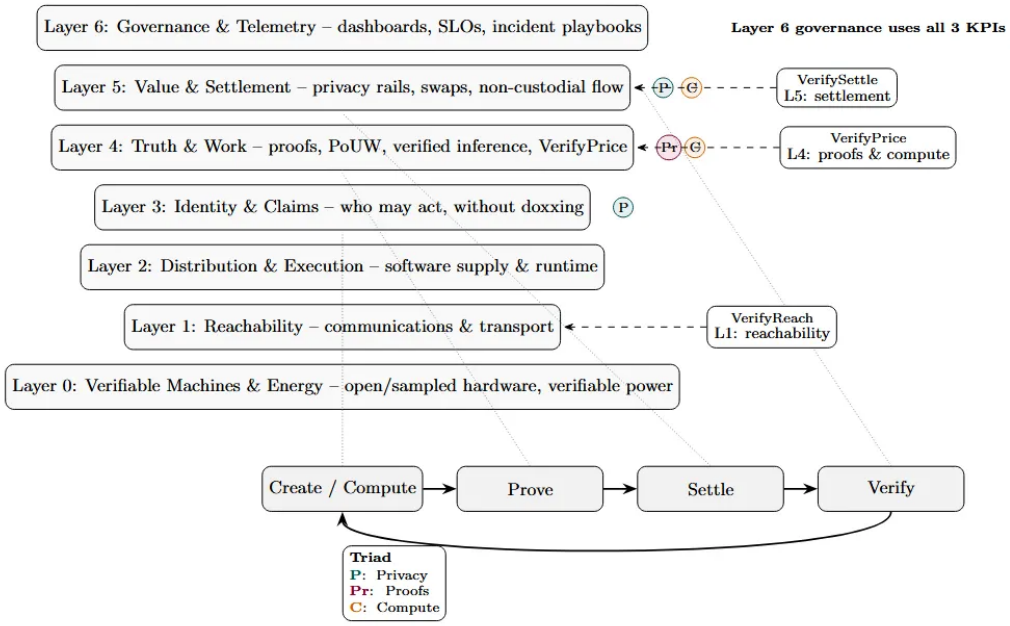

Figure 1: Seven-layer cypherpunk stack with Create/Compute→Prove→Settle→Verify loop, triad overlay (Privacy, Proofs, Compute), and telemetry KPIs (VerifyReach, VerifyPrice, VerifySettle).

Figure 1.1: Architecture Map: The seven-layer cypherpunk stack.

2. **Economic formalization: verification asymmetry and VerifyPrice.**
   We formalize verification asymmetry as the ratio

$$r(W) = \frac{v(W)}{p(W)}$$

   between verification and production cost for a workload W, and introduce **VerifyPrice** (a public KPI vector of p50/p95 verify times, costs, and failure rates) as the hinge that turns proofs and verified FLOPs into commodities rather than platform IOUs. This gives a quantitative basis for treating Privacy, Proofs, and Compute as monetary primitives and for defining "AI money" and "ZK money" as assets backed by verifiable work.
3. **Modular stack: four reference applications and twelve primitives.**
   We propose a **Create/Compute → Prove → Settle → Verify** loop and instantiate it with four reference applications (private treasury & payroll, media provenance & authenticity, verified inference, and proof/compute procurement), built on a reusable kit of primitives. These include a Proofs-as-a-Library SDK (PAL), a Privacy Rails Kit (PRK) for non-custodial settlement, Proof Interface Definition Language (PIDL) as a minimal receipt schema, MatMul-

PoUW and verified-inference harnesses, canonical workload registries, multi-ZK adapters, SLA escrow/slashing, neutral routers, bridge-safety templates, and a telemetry layer that keeps useful work and neutrality measurable across chains and vendors.

4. **Telemetry, governance, and implementation playbook.**
   We extend VerifyPrice into a broader observability regime, including **VerifyReach** (reachability under censorship) and **VerifySettle** (settlement success and refund safety), and propose a "no dashboards, no trust" governance posture where protocol changes and incident responses are driven by SLOs and public receipts rather than foundation fiat. We complement this with an adoption curve, an operator/investor checklist for SoV evaluation, and implementation sketches (Layer-0 hardware, privacy corridors, proof factories, developer SDKs) that make the stack actionable for builders and allocators over the next 12–36 months.

1.3 End-to-End Vignette: The Loop in Action

Before diving into theory, here is a concrete story that walks the **Create/Compute → Prove → Settle → Verify** loop. This vignette shows how the stack works as a felt reality, not just a framework.

Scenario: A Company Under Capital Controls Runs Payroll

TechCo is a software company with employees in three countries. Local banks are increasingly unreliable: one jurisdiction has imposed capital controls, another requires intrusive reporting, and exchange rates are manipulated. TechCo's CFO wants to pay staff in a stable, private, auditable way—without exposing salary data to competitors or governments, but with the ability to prove to auditors that payroll was run correctly.

Step 1: Create (Intent) The CFO opens TechCo's treasury dashboard (built on PRK, the Privacy Rails Kit). She creates a payroll batch:

- 47 employees
- Total: 142,000 USD-equivalent in Work Credits
- Policy: "Salaries are private; aggregate spend is auditable; individual amounts disclosed only with employee consent."

The dashboard compiles this into a **claim**: "Pay these 47 recipients the specified amounts, under this policy, by end-of-day Friday."

Step 2: Prove (Compliance + Privacy) The claim is sent to PAL (Proofs-as-a-Library). PAL compiles it into two proofs:

1. **Compliance proof:** A ZK proof that the batch satisfies TechCo's internal policy (aggregate

under budget, recipients are on the approved list, no single payment exceeds threshold). This proof reveals *nothing* about individual amounts or identities—only that the rules were followed.

2. **Integrity proof:** A hash commitment binding the full payroll data. This hash is stored on-chain; the actual data stays encrypted in TechCo's vault.

Both proofs are packaged into a **PIDL receipt**: claim hash, proof hashes, workload ID, SLA tier, timestamps, and a prover signature.

VerifyPrice checkpoint: The compliance proof took 2.3 seconds and cost $0.004 to generate. Independent verification takes 0.8 seconds on a laptop. This is well within the SLO ($p_{95,t} \leq 5$s, $p_{95,c} \leq \$0.01$).

Step 3: Settle (Private Execution) With proofs in hand, PRK executes the payroll:

- For employees in the capital-controlled jurisdiction: atomic swap via **BTC↔XMR corridor**. TechCo's BTC is swapped for XMR, which is sent to employee wallets. The corridor is non-custodial; if anything fails, funds return to TechCo (refund-safe).
- For employees elsewhere: direct settlement over a **shielded pool** (e.g., Zcash). Each payment is encrypted; only the recipient and TechCo (via viewing keys) can see the details.

VerifySettle checkpoint:

- Swap success rate: 98% (2 retries due to network latency, both succeeded).
- Refund safety: 100% (all potential failures would have returned funds).
- Time to finality: median 4 minutes, p95 11 minutes.
- Anonymity set: 12,000+ active notes in the shielded pool.

Step 4: Verify (Audit Trail) After settlement, the following are publicly verifiable:

- **On-chain:** The PIDL receipt exists, the proofs verify, the aggregate amount was transferred at the specified time.
- **By TechCo's auditors:** Using viewing keys, auditors can see the full breakdown (who got paid how much) and confirm it matches the compliance proof.
- **By employees:** Each employee can verify their own payment arrived, using their private key.
- **By anyone:** The compliance proof demonstrates policy was followed, without revealing any private data.

What the adversary sees:

- A capital-controls regulator in Country A sees: "TechCo moved some BTC into an XMR corridor." They cannot see amounts, recipients, or purposes—unless TechCo chooses to disclose.
- A competitor monitoring the blockchain sees: "Some shielded transactions occurred." They learn nothing about TechCo's payroll structure or employee compensation.

- A hacker who compromises a single employee's device sees: that employee's payment. They cannot reconstruct the full payroll or identify other employees.

What TechCo gains:

- **Privacy:** Payroll data is not exposed to competitors, governments, or hackers by default.
- **Compliance:** Auditors get cryptographic proof that policy was followed, without needing to trust TechCo's word.
- **Resilience:** Even if banks freeze accounts or exchanges delist, the privacy corridor remains operational.
- **Receipts:** Every step produced a verifiable artifact (PIDL receipts, proofs) that can be archived, audited, or used in disputes.

Stage	What Happens	Output
Create	CFO defines intent + policy	Claim
Prove	PAL compiles compliance + integrity proofs	PIDL receipt
Settle	PRK executes via privacy corridors	Value transferred
Verify	Anyone can check proofs; auditors use viewing keys	Audit trail

Table 1.1: The Create/Compute → Prove → Settle → Verify loop in the payroll scenario.

The loop in summary: This is **lawful privacy**: default-private, optional-disclosure, with receipts that anyone can verify. The triad (Privacy for settlement, Proofs for compliance, Compute for proof generation) works together to make the flow possible.

Variation: Media Provenance

The same loop applies to a journalist publishing a video:

1. **Create:** Camera with secure element captures footage, emitting a signed provenance attestation (device ID, timestamp, geolocation hash).
2. **Prove:** Edit suite issues proofs of each transformation (crop, color grade, caption). Each edit is a new PIDL receipt referencing the parent.
3. **Settle:** (Optional) If the video is monetized, payment flows over privacy rails to the creator, with receipts linking payment to provenance.
4. **Verify:** Any viewer or platform can verify the chain: "This video originated from camera X at time T, was edited as follows, and has not been tampered with since." Deepfake detectors can check against registered provenances.

Platforms that strip metadata cannot strip the on-chain PIDL receipt. The proof persists even if the platform delists the content.

These vignettes are not hypothetical futures; they are the concrete scenarios the rest of this

thesis is engineered to support. The stack exists to make these flows **cheap, verifiable, and non-custodial** under adversarial conditions.

Chapter 2

The Failure of Soft Guarantees

The twentieth century ran on soft guarantees: the promise that money would preserve purchasing power; that televised images would correspond to facts; that institutions would be slow, sober, and legible. The twenty-first runs on different physics. Monetary systems must service accumulated promises faster than real productivity can grow. Media systems can synthesize plausible realities faster than editors can adjudicate them. Intelligence, once gated by human attention, now scales with silicon. In that environment, guarantees that depended on slow institutions and centralized editors begin to fray.

Since the end of Bretton Woods, *compliance*, not convertibility, has been the hidden backing of money. When compliance becomes weaponized and asset freezing becomes policy, savings need privacy the way ships need hulls.

> *When the **symbols of trust** are cheaper to counterfeit than to maintain, the market seeks a new substrate.*

This is not moral decline; it is entropic arithmetic. When verification is costly and authority is centralized, cheaters arbitrage the gap. When verification becomes cheap and public at scale, the equilibrium flips. Our civilization is now crossing that threshold.

"Money is memory with consequences."

It records who did work and who is owed work, then carries that record across time, space, and politics. Gold solved this with geology; fiat with law, taxation, and war; Bitcoin with thermodynamics and code. Each regime answered the same question (what counts as real work, and how do we know?) with a different ontology.

Our era answers: **work is what machines can do and humans can verify cheaply**. The new memory must therefore be backed by capacities essential to digital life and secured by proofs anyone can check. The unit must survive not because a state decrees it, but because the network will keep buying its utility through booms, busts, and repression cycles.

> "A store of value is not a museum; it is a power plant that stays online."

The same forces that erode the old legitimacy (AI, ubiquitous networks, programmable markets) also supply the antidote. Cryptography can wrap data, identity, and computation in

attestations that travel; privacy tech can re-sacralize the personal and the intimate; verifiable compute can make expensive work legible to cheap verification. The system that dissolves trust also furnishes the reagents to precipitate it again, this time as math, not authority. Verification, not violence, becomes the final arbiter of truth.

In the rest of this thesis we make that sketch precise. We start from a threat model that assumes repression, not benevolence: debt stocks that make financial repression arithmetically attractive; a communications stack that can be filtered or shut off; hardware and identity systems that can be integrated into surveillance infrastructure. Against that backdrop we propose a layered architecture, from verifiable machines and energy up through communications, distribution, identity, truth, settlement, and telemetry, and show how three capacities (Privacy, Proofs, and Compute) can be engineered across those layers as monetary primitives. The task is not to find a new story to believe in, but to build a stack where cheap, public verification and indispensable utility do the work that soft guarantees used to perform.

Chapter 3

The World Forces New Monetary Primitives

3.1 Macro Playbook: Debt → Repression → Flight to Neutrality

Total global debt (private + public) according to the IMF's latest Global Debt Database now sits at a roughly ~2.4× multiple of world GDP, with some economies (such as Japan) exceeding 400% of GDP when public and private debt are combined. Standard scenarios from multilateral institutions point to global public debt approaching ~100% of GDP by decade-end, after already exceeding $100 trillion in the mid-2020s.

Historically, such overhangs are closed via **financial repression**: inflation plus regulation that imposes negative real returns, including overt capital controls and captive-balance-sheet rules, pushing savers toward neutral, censorship-resistant rails. "Voting with your feet" is harder when the exits are policed at gunpoint.

The Bondholder Kill Box (How Repression Works)

Definition. Financial repression is a stealth tax on savings: cap nominal rates, let inflation run, and force captive balance sheets to hold the paper. The result is systematically negative real returns for depositors and bondholders; an engineered wealth transfer to the sovereign. In the post-WWII advanced economies, real rates were negative in roughly *half the years from 1945–1980*, and the "liquidation tax" (interest-expense savings on the public debt) averaged on the order of 1–5% of GDP per year.

How the cap is enforced. Overt or implicit yield-curve control (YCC), administered rate ceilings, capital/liquidity rules that force government paper into banks, insurers, pensions, and collateral boxes, plus capital controls to slow leakage. In the U.S. WWII episode, the Fed capped T-bills at 3/8% and long bonds at 2½% until the 1951 Accord; when the cap ended, long-duration holders took capital losses as yields normalized.

Why now. With global debt at several times world GDP, public debt ratios trending toward ~100% of GDP by decade-end, and interest costs compounding, repression is arithmetically attractive to policymakers versus explicit default or politically costly aus-

terity.

Two equations every saver should know:

1. Repression wedge (real carry):

$$r_{\text{real}} \approx \frac{1 + i_{\text{capped}}}{1 + \pi} - 1$$

If $i_{\text{capped}} = 2.5\%$ and inflation $\pi = 6\%$, the real return is

$$\frac{1.025}{1.06} - 1 \approx -3.3\% \text{ per year.}$$

2. Duration burn (when the peg breaks): For a bond with duration D, a yield move Δy produces a price move

$$\Delta P \approx -D \cdot \Delta y.$$

A 10-year with $D \approx 8$ takes about -24% on a $+300$ bps repricing.

A worked example: three years at -3.3% real/yr is $\approx -9.6\%$ real loss compounded; then a $+300$ bps jump deals another $\approx -24\%$ price loss, roughly -31% cumulative in purchasing-power terms. The design is deliberate: bleed bondholders while buying time for the sovereign balance sheet.

3.2 The Repression Playbook, Then and Now

The lesson is simple: post-Bretton-Woods money is upheld more by regulation, violence (or its threat), and surveillance than by reserves. When repression becomes the spread, capital quietly rotates to neutral, bearer-like rails. We are already seeing a partial recollateralization of the financial system with gold following events like the seizure of Russia's reserves in early 2022, and a visible rotation of some marginal liquidity toward privacy-preserving assets such as Zcash (ZEC) and Monero (XMR).

If "safe" nominal assets become de facto taxes on savings, rational capital routes (not protests) to assets whose issuance cannot be decreed and whose verification is cheap and public.

In that environment, a credible store of value must:

- Avoid being a duration instrument whose real return can be pinned negative by policy; and
- Live on rails that cannot easily be gated by a single jurisdiction.

This is where the triad enters:

- **Privacy** as an anti-seizure hull that makes savings bearer-like again.
- **Proofs** as cheap public verification that replaces platform vouching.

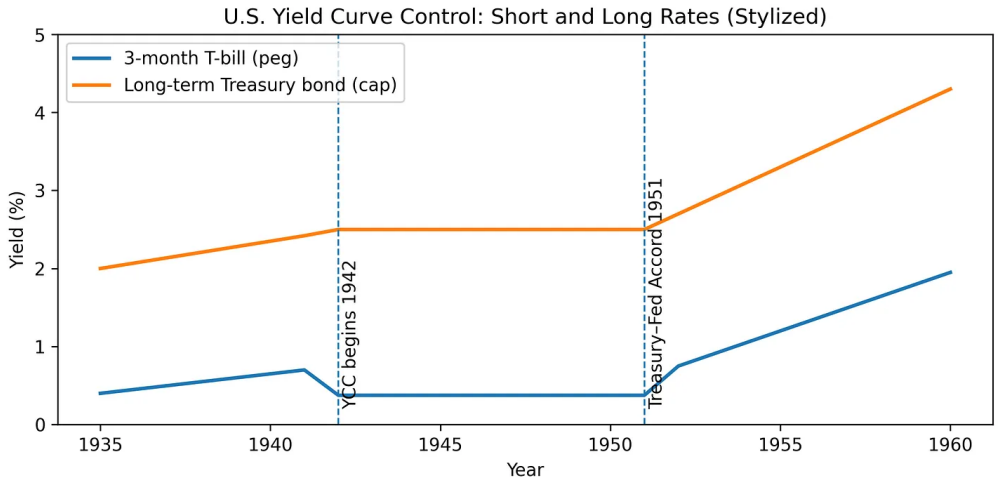

Figure 3.1: U.S. Treasury bill peg and long-bond cap under WWII-era yield-curve control (stylized). Peg at 3/8% on bills, cap at 2½% on long bonds; dashed line marks the 1951 Treasury–Fed Accord. Source: Rose (2021), Federal Reserve Bank of Chicago.

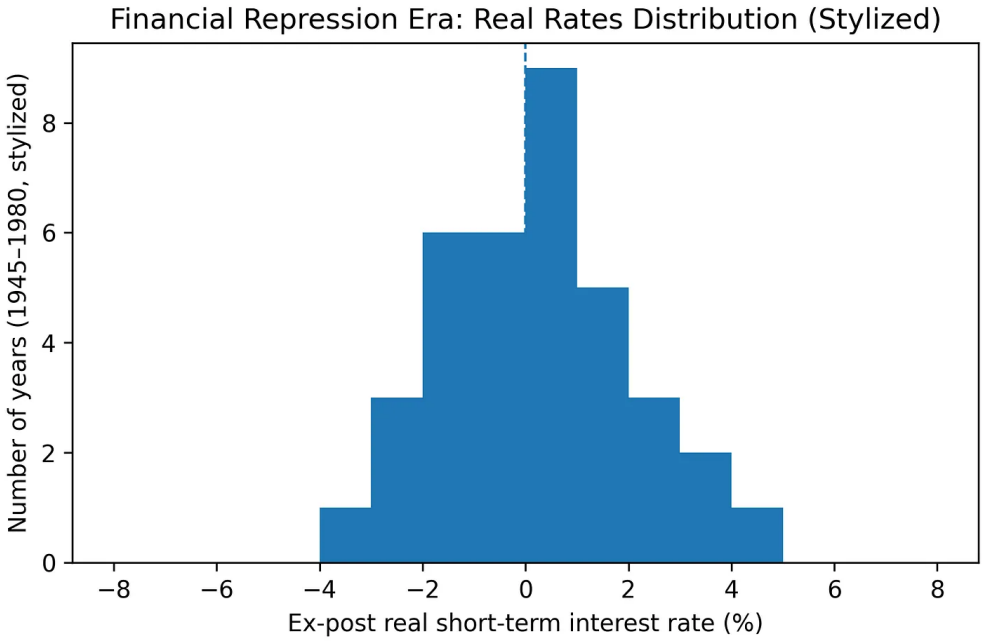

Figure 3.2: Stylized distribution of ex-post real short-term interest rates in advanced economies, 1945–1980. Real rates are negative in roughly half the years, consistent with Reinhart & Sbrancia's estimates of the financial-repression era.

- **Compute** as a non-coupon revenue base whose clearing price floats with nominal budgets rather than being pegged by decree.

Each primitive is designed to remain neutral and verifiable even when traditional financial infrastructure is weaponized.

3.3 Social: The Web's Trust Default Has Flipped

The same technological forces reshaping money are also reshaping trust.

The web used to borrow its epistemic norms from broadcast: seeing was believing, and the job of editors was to gate what got seen. Deepfakes and generative models invert that. AI systems can now synthesize faces, voices, and scenes that are indistinguishable from genuine footage to lay observers. Exposure itself doesn't inoculate people; survey data across multiple countries suggests prior exposure to deepfakes can actually increase susceptibility to misinformation.

In parallel, we get the **liar's dividend**: once everyone knows deepfakes are possible, any inconvenient real video can be dismissed as fake. The result is a crisis of knowing, not just a rise in error rates.

This isn't entirely new (authoritarians have been editing history since Stalin airbrushed Trotsky out of photos), but the cost curve and scale have changed. Deepfake tools now let anyone with a laptop fabricate a leader conceding an election, a riot that never happened, or "footage" of a historical event that subtly rewrites who was present and who was not. The fear is not just short-term hoaxes; it is revisionist history at machine speed, where archives, livestreams, and "receipts" themselves become contestable. In that world, the substrate of trust shifts from *memory* ("I saw the clip") to *mechanism* ("I can verify how this artifact came to be").

At the same time, the feeds themselves are no longer obviously "neutral pipes." Investigations and hearings around government–platform coordination have documented extensive relationships between agencies, NGOs, and major platforms for content moderation in the name of combating misinformation and foreign influence. Whether one views this as necessary harm reduction or problematic overreach, the empirical point stands: the old story that your feed is simply "what's popular" no longer survives contact with the evidence. What you see (and don't see) is the result of political, institutional, and algorithmic priors you don't control.

Content authentication standards such as C2PA and watermarking prototypes exist on paper, but real-world implementations are voluntary, inconsistent, and often invisible to end-users. Tests across major platforms such as the Washington Post's 2025 C2PA test show provenance metadata being stripped in transit, and the few disclosures that do survive are buried in UI that most users never touch.

In other words, **the trust default has flipped**. The practical baseline for online media is now "untrusted unless proven," and the "proven" part cannot safely be left to platform labels

or government–platform partnerships. Platform UX is inconsistent; cryptographic provenance and computation proofs are the scalable, cross-platform backstops.

In a SoV context, this matters because monetary systems sit on top of communication systems: if claims about ownership, origin, and behavior are cheap to fake and expensive to audit, then both money and memory become soft targets. Anchoring value in *receipts that anyone can verify*, rather than narratives that someone must vouch for, is the only stable equilibrium.

Recent work constructs PoUW for arbitrary matrix multiplication with verification that is asymptotically cheaper than production (e.g., $O(n^2)$ verify vs. $O(n^3)$ produce; "$\sim 1 + o(1)$" denotes a constant-factor overhead relative to the best known verification baseline). Matrix multiplication is the operation that bottlenecks modern AI. This is the asymmetry PoW always needed: hard to produce, cheap to check.

When verification is very cheap relative to production, **honesty becomes a market equilibrium** and commodities emerge (standardized proofs, verified FLOPs) that can be priced, saved, and eventually used as collateral and stores of value.

Chapter 4

First Principles: What a SoV Must Survive

A credible store of value must endure time, space, and politics. In practice, that means:

1. **Credible scarcity**: issuance cannot be tweaked at will.
2. **Cheap, public verification**: authenticity is verifiable by anyone, not decided by a platform.
3. **Censorship-resistance & portability**: no chokepoints; global movement by default.
4. **Neutrality & permissionlessness**: open access; rules apply equally.
5. **Native demand**: the asset does something indispensable, beyond serving as a symbol.
6. **Lawful privacy by design**: default privacy with optional disclosure (viewing keys, auditable receipts) so regulated actors can comply without re-introducing custodians or surveillance chokepoints.
7. **Duration-neutrality**: no fixed nominal cash flows to peg. A repression-resistant SoV cannot be a duration instrument whose real return can be driven persistently negative or whose price can be easily mass-managed by policy. Value should come from scarce capacity purchased every cycle, not from a promised coupon.

Each of these conditions is just the "money as memory" idea made operational: if money is the record of who did work and who is owed work, then scarcity, verifiability, censorship-resistance, neutrality, native demand, lawful privacy, and duration-neutrality are the properties that keep that record honest across time, space, and politics.

> **Thesis:** Privacy, Proofs, and Compute can be engineered to meet all seven, and the world now needs exactly those properties.

We can sketch how the triad maps to SoV requirements:

- **Credible scarcity →** issuance schedules and realized issuance vs. schedule for triad assets and Work Credits.
- **Cheap, public verification → VerifyPrice(W)** dashboards for canonical workloads W.
- **Censorship-resistance & portability → VerifyReach** + **VerifySettle** metrics: reachability under censorship and settlement success/refund safety across corridors.
- **Neutrality & permissionlessness →** decentralization telemetry: house share, geo/ASN dis-

tribution, time-to-first-proof.

- **Native demand →** Work Credit utilization, proof/compute fee share of the security budget, and application-level usage.
- **Lawful privacy →** corridor compliance receipts, viewing-key usage, anonymity-set health (shielded-pool size, churn, volume).
- **Duration-neutrality →** no fixed coupons; revenues from priced workloads and triad usage, with explicit "repression beta" rather than fixed-income-like promises.

Table 4.1: Store-of-Value Requirements vs. Triad Capabilities

SoV Requirement	Privacy (P)	Proofs (Pr)	Compute (C)	Metrics / SLOs
Credible scarcity	Shielded balances; no selective debasement; predictable issuance schedules	Auditable supply without doxxing; proof capacity tied to hardware and energy	Verified FLOPs limited by physical compute; energy-tied issuance	FERs; facility telemetry; VerifyPrice
Cheap, public verification	Privacy proofs verifiable by anyone; private paths for ordinary users	Succinct, cheap-to-check proofs; VerifyPrice targets $p95 \leq 5s$, $r(W) \leq 0.3$	Bounded verify cost for heavy workloads; MatMul $O(n^2)$ vs $O(n^3)$	VerifyPrice; VerifyReach
Censorship-resistance	Non-custodial atomic swaps and shielded pools; harder for censors to see flows	Receipts portable across chains; proofs of inclusion/exclusion expose filtering	Open-admission prover/miner markets; diverse operators	VerifyReach; VerifySettle
Neutrality	Privacy by default, not by permission; same privacy for all	Anyone can verify; open circuits and PIDL interfaces; no gatekeepers	Useful-work mining with open hardware paths; no single vendor chokepoint	Regional VerifyPrice; dispersion metrics
Native demand	Enterprises and individuals need privacy for operations, payments, savings	Provenance and compliance mandates; proofs demanded by AI, finance, regulators	AI workloads, ZK proving as budget line items	Fee volume; Work Credit utilization
Lawful privacy	Viewing keys and auditable receipts; default-private with narrow exceptions	Proof anchors satisfy audits; policy-aware proofs reveal facts, not full data	Compute SLAs with receipts; policy-tagged workloads	Compliance flows; stress-test results
Duration-neutrality	No fixed coupon; revenues track priced capacity; survives regime change	Proof unit pricing floats with budgets; long-run telemetry	Verified FLOPs clear at market rates, not pegs; anchored in infrastructure	Multi-year FER and SLO stability

In a world that will likely choose *stealth default* (negative real yields and capital controls) over explicit default, "store of value" can't just be a narrative; it has to clear visible, adversarial stress tests. If the whole point of the stack is to survive yield-curve control, on-/off-ramp throttling, and peg breaks, then we should write down the conditions under which it continues to function and accrue value.

The checklist below turns those claims into operator SLOs that treasuries and allocators can actually underwrite and monitor. Here are several repression stress-tests to keep in mind:

- **YCC shock.** 24–36 months of −300 to −500 bps real yields → fee+burn share of the security budget remains ≥ a target threshold (e.g., 40–60%) and VerifyPrice (p95) remains < 5s for core workloads.
- **On-/off-ramp squeeze.** Non-custodial BTC↔XMR/ZEC routes maintain ≥ 95% success with 100% safe refunds; shielded-pool anonymity sets remain large and growing (no collapse in active notes or volume).
- **Peg break.** If global yields gap +300 bps, triad revenues (proofs/FLOPs, privacy rails usage) track buyer budgets; there are no coupon-like revenue shortfalls that turn the asset into a synthetic bond.

Publish these in dashboards; **no dashboards, no trust**. Later sections turn these stress tests into explicit SLAs and telemetry: VerifyPrice, reachability, settlement success, and decentralization metrics that operators and allocators can track in the open.

With these seven requirements as design constraints, we can now turn to the primitives that might satisfy them. Before naming those primitives, however, we need to be explicit about who will attack this system and at what layers. The next section states that threat model; the one after lays out the layered architecture that the rest of the thesis will fill in. The entire document can be read as an attempt to engineer Privacy, Proofs, and Compute to satisfy this seven-point contract under adversarial conditions.

We can summarize the whole design posture in one line: our North Star is to pay the machine only for work anyone can verify cheaply, on rails that give humans lawful privacy by default.

Chapter 5

Threat Model

Why a threat model? This thesis argues that Privacy, Proofs, and Compute can function as monetary primitives when verification is cheap and public. A threat model makes that claim falsifiable. It defines what we must protect, who is trying to break it, where the trust boundaries lie, and which metrics decide if the system remains credible under pressure. It sits between first principles and implementation so that every later design choice can be traced to an explicit adversary and invariant.

Every architecture is, implicitly, a bet about who will attack it and how. Up to this point we have argued from first principles: what a store of value must survive, and why Privacy, Proofs, and Compute can be engineered to meet those conditions. The next step is to be explicit about the pressure this stack will live under.

The backdrop is already in view. Macroeconomic arithmetic makes financial repression attractive: debt stocks in the hundreds of percent of GDP, negative real rates by policy, capital controls and captive balance sheets as standard tools rather than emergency measures. At the same time, the web's trust default has flipped from "trusted unless flagged" to "untrusted unless proven" as deepfakes and coordinated moderation make memory itself a soft target. The combination is simple and brutal: states are incentivized to squeeze balance sheets and communications; platforms are incentivized to mediate reality; hardware and networks are increasingly treated as levers of policy, not neutral infrastructure. Against that field, a "next-gen store of value" is not being asked to survive price volatility; it is being asked to survive a world that optimizes for control.

5.1 Scope and Assets

The scope is the full **Create/Compute → Prove → Settle → Verify** loop and its hardware base layer. At minimum, we need to protect four things:

1. **Integrity of receipts & state.** Proofs, provenance, settlement artifacts, and consensus state must not be silently corrupted.
2. **Confidentiality of flows.** Private settlement with optional disclosure, not privacy that evaporates at the first audit request.
3. **Availability & neutrality.** Open admission for provers and users, anti-capture, and non-

seizable settlement paths.

4. **Verifiability economics.** Verification must remain cheap in absolute and relative terms; otherwise "anyone can verify" collapses into "someone we have to trust."

These map directly to the SoV properties enumerated in Chapter 4. The assets at risk are not just balances on ledgers. They are the whole loop and the matter it runs on.

- At the bottom lie **machines**: chips, RNGs, secure elements, TEEs, capture devices, modems, routers, and the power infrastructure that feeds them.
- Up one level sit the **proofs and receipts** that describe what those machines did: provenance attestations, computation proofs, settlement artifacts, audit trails.
- Above that are the **flows of value and obligation** that ride on those receipts: private payrolls, cross-chain settlements, inference contracts, collateral arrangements.
- Threaded through everything are the **communications substrate** that carries claims and proofs, the **software distribution channels** that keep clients coherent, and the **identity and key material** that bind actions to actors without doxxing them.

All of these must remain sufficiently intact that savings, truth, and compute can continue to clear when old guarantees fail.

5.2 Adversary Classes

We care about several classes of adversaries:

State-level repressors. Impose negative real yields, capital controls, and hardware mandates; gate on-/off-ramps; require closed attestation; seek to turn banks, clouds, app stores, and IXPs into enforcement arms.

Platform cartels. Strip or obscure provenance; collude on labels; prefer vendor-mediated "trust" and walled-garden verification.

Economic attackers. Miner/prover cartels, router capture, MEV/censorship, liquidity games on swap corridors, front-running and soft-forking that tilt rules in their favor.

Hardware/supply-chain adversaries. Trojans, biased RNGs, covert debug paths that turn "proof" into theater; opaque TEEs whose attestation keys terminate in unaccountable HSM farms.

Cryptanalytic / PQ attackers. Proof forging or signature breaks; long-horizon post-quantum risk.

UX / vaporware risk. Spec-drift and unverifiable performance claims that corrode credibility, even if the cryptography is sound.

Of these, the primary adversary in this frame is not a cartoon hacker but the **rational sovereign under stress**. A state that must service promises larger than its productive base will reach, as history shows, for the levers it actually controls: interest-rate caps, yield-curve control, capital controls, regulated custody, and mandatory "secure" hardware and identity schemes. It will be tempted to turn banks, clouds, app stores, hardware vendors, and IXPs into

enforcement arms: require real-name registration at the edge; throttle or shut down networks in the name of stability; mandate TEEs whose attestation keys terminate in state-controlled HSMs; define "safety" as surveillance.

In parallel, platforms facing regulatory and reputational risk will centralize moderation and provenance: which media is shown, which credentials count, which proofs are recognized. The danger is not that any one actor becomes an obvious villain, but that their combined incentives re-create a soft but totalizing chokepoint architecture.

Beneath that sovereign–platform axis are **economic adversaries**: miners, provers, routers, and liquidity providers who prefer rent to work. They will collude if they can, capture matching engines and orderflow, quietly prioritize their own routes, soft-fork the rules in their favor, or simply withdraw service when it suits them.

> If verification becomes expensive or gated, a priesthood of "trusted verifiers" will emerge, and with it all the familiar pathologies of rent extraction and arbitrary censorship.

And below them, like bedrock or landmines, are **hardware and supply-chain adversaries**: fabs and vendors (sometimes complicit, sometimes merely compromised) who can bias randomness, slip trojans into IP blocks, leave "debug modes" wired to secrets, or ship opaque enclaves that amount to remote-controlled kill switches. If the machine can lie, the proof becomes theater.

5.3 Threats by Loop Stage

We can slice the threat surface by stage in the Create/Compute → Prove → Settle → Verify loop:

Create/Compute.

- *Threats:* poisoned models, mislabeled workloads, biased hardware profiles, hidden accelerators.
- *Mitigation:* canonical workload registries, hardware profiles bound to receipts, open benchmarking, attested hardware provenance.

Prove.

- *Threats:* junk proofs, prover cartels, selective service, "house-only" pricing.
- *Mitigation:* multi-ZK adapters, neutral routers with fairness tests, SLA escrows & slashing; publishing entry-latency, top-N share, and geography/ASN distributions.

Settle.

- *Threats:* corridor censorship, refund failure, deanonymization, bridge compromise.

- *Mitigation:* adaptor-signature swap kits, lawful-privacy corridors (viewing keys + auditable receipts), bridge-safety templates, mandatory refund-safety, public **VerifySettle** metrics.

Verify.
- *Threats:* verification-cost creep; opaque clients; "trust our node" monocultures.
- *Mitigation:* laptop-grade verifiers, reference harnesses, verify predicates embedded in SDKs, public p50/p95 telemetry via the **VerifyPrice** observatory.

Telemetry & Governance.
- *Threats:* silent centralization, KPI gaming, governance capture.
- *Mitigation:* open dashboards for VerifyPrice, VerifyReach, VerifySettle; decentralization metrics (house share, geo/ASN spread, time-to-first-proof); incident reports; hard rules about what must be public before changes ship.

Layer 0.
- *Threats:* "trust the vendor" cliff, mandatory closed TEEs, unmeasured side-channels.
- *Mitigation:* open RTL or microarchitectures where possible, lot sampling and imaging where not, SNARK-wrapped attestations, and explicit side-channel budgets.

5.4 Mapping the Authoritarian Playbook

We can sketch several canonical "playbook moves" and the corresponding counter-moves. The table below maps real-world censorship and surveillance tactics to the stack's countermeasures:

Beyond these network-level moves, the financial and institutional playbook includes yield-curve control, capital controls, mandatory "secure" hardware mandates, platform-mediated reality attacks, and proof priesthoods. For each of these macro attacks, the stack provides specific countermeasures that are detailed throughout the document and formalized as testable stress harnesses.

The high-level pattern:
- **Yield-curve control + captive balance sheets →** triad assets with no fixed coupons, Work Credits priced by market demand for Privacy/Proofs/Compute rather than by decree.
- **Capital controls + KYC choke points →** non-custodial privacy corridors (BTC↔ZEC/XMR) that remain viable without centralized exchanges; VerifySettle metrics that make corridor health public.
- **Mandatory "secure" hardware + identity schemes →** Layer-0 verifiable machines with open or sampled designs; hardware-profile receipts; identity schemes at Layer 3 that separate accountability from doxxing.

Playbook Move	What It Breaks	Counter
DNS/IP/SNI/QUIC filtering & active probing	Reaching verifiers/bridges; provenance fetch	Default to Tor Snowflake/obfs4, refraction networking; pin ECH; fall back to domain-front-free paths.
Domain fronting curtailed by major clouds	"Collateral freedom" paths disappear	Use Snowflake's WebRTC and ISP-partner decoy routing; keep multiple CDNs with content-addressed updates.
App-store pressure	Mobile distribution	Parallel update rails (direct, IPFS, USB/QR) + detached sigs; keep APK/IPA sideload guides ready.
Real-name + SIM registration	Pseudonymous ops	VC/anon-cred kit; receipt-based reputation; no SIM/face gates for access.
Government-ordered shutdowns	All of the above	Sat/mesh/sneakernet paths, store-and-forward receipts; pre-provisioned peer lists; publish "blackout drills" results.
Backbone/seabed sabotage	Regional isolation	Multi-landing routing, regional proof markets, diaspora relays; monitor cable incidents.
Bitcoin network metadata capture	Wallet/settlement mapping	BIP-324 everywhere; BTC↔XMR/ZEC adaptor-sig corridors with refund-safety; publish corridor telemetry.

Table 5.1: Authoritarian playbook moves and stack countermeasures.

- **Platform-mediated reality + provenance stripping →** PIDL-encoded receipts for media and computation; PAL to compile provenance claims into portable proofs that survive platform stripping.
- **Proof priesthoods + closed verifiers →** laptop-grade reference verifiers; multi-backend proof systems; VerifyPrice observatories that publish verifier costs and failure rates.

The thesis responds by treating neutrality, privacy, and verifiability as **service-level objectives**, not vibes.

5.5 Explicit Tensions and Boundary Conditions

Two tensions within this thesis deserve explicit acknowledgment. Ignoring them would make the framework less credible; addressing them honestly strengthens it.

Tension 1: Privacy by Default vs. Coercion at the Social Layer

The concern: Lawful privacy proposes "optional disclosure via viewing keys." But if disclosure becomes mandatory at chokepoints (exchanges, employers, landlords, visa applications), "optional" becomes fiction. The technical guarantee of privacy could be circumvented by social or legal coercion.

What the system can and cannot guarantee:

Layer	Guarantee	Limitation
Protocol	No backdoors, no master keys, no escrow. Disclosure requires the holder's key.	Cannot prevent a holder from being coerced to share their key.
Architecture	Non-custodial routes exist. Users can transact without intermediaries.	If all practical routes require KYC, non-custodial option may be theoretical.
Corridor design	Multiple asset paths with redundant liquidity in multiple jurisdictions.	If all corridors are choked simultaneously, exit paths narrow.
Telemetry	VerifySettle makes corridor health public; users can see which routes are viable.	Does not prevent coercion; only makes it visible.

How the stack reduces coercion surface:

1. **Data minimization:** By default, nothing is revealed. Coercion must extract keys, not merely subpoena records that already exist.
2. **Decentralized verification:** No single auditor or platform must see all flows. Selective disclosure can be scoped narrowly.

3. **Multi-jurisdiction design:** Corridors, mirrors, and provers are distributed so no single legal regime can compel universal disclosure.
4. **Exit optionality:** Even under local coercion, users retain technical ability to move assets to less-coerced jurisdictions—if corridors remain open.

Honest acknowledgment: Lawful privacy is a *technical* property, not a social guarantee. It cannot prevent a sufficiently powerful adversary from coercing individuals. What it does is (a) raise the cost of mass surveillance (each disclosure requires individual coercion), (b) preserve optionality for those in less-coerced environments, and (c) make the coercion visible through telemetry rather than hidden in platform logs.

Tension 2: Verified Compute vs. Hyperscaler Dominance

The concern: Compute is dominated by a handful of hyperscalers (AWS, Azure, GCP) and chip vendors (NVIDIA, AMD). If verified compute becomes valuable, won't these incumbents simply dominate issuance and markets, making "decentralized" PoUW a fiction?

What the system offers vs. hyperscalers:

Property	Hyperscaler Compute	Triad Verified Compute
Cost per raw FLOP	Lower (economies of scale)	Higher (proof overhead, smaller operators)
Verifiability	Trust the vendor	Anyone can verify receipts
Permissionlessness	Vendor can ban workloads, customers, regions	Open admission; no TOS-based exclusion
Censorship surface	Single legal entity; can be compelled	Distributed operators; no single chokepoint
Receipts	Vendor-issued invoices	PIDL receipts verifiable by anyone

Why decentralized verified compute wins in specific scenarios:

1. **Repression scenarios:** When a hyperscaler is compelled to ban certain workloads or customers, decentralized alternatives remain available. The value proposition is not "cheaper" but "still available."
2. **Proof-heavy applications:** For workloads where *verifiability* is the product (compliance proofs, provenance, audit trails), the receipt is the value—not raw FLOP cost. Hyperscalers don't currently sell receipts; they sell capacity.
3. **Certain procurement regimes:** Governments, NGOs, and enterprises with sovereignty constraints may prefer compute that doesn't route through foreign hyperscalers or vendor-controlled enclaves.
4. **Composability with privacy:** Hyperscaler inference is logged. Verified inference over privacy rails is not. For sensitive workloads (medical, financial, personal), the privacy premium

justifies higher cost.

Honest acknowledgment: Decentralized verified compute will not beat hyperscalers on raw cost. It competes on *verifiability, permissionlessness, and censorship-resistance*. In a benign environment, hyperscalers win on price. In a repressive or high-stakes environment, verified compute wins on trust properties. The thesis bets that demand for these trust properties is structural and growing.

Telemetry response: If hyperscaler concentration in verified compute markets exceeds thresholds (e.g., >50% of VerifyPrice-tracked capacity from 3 vendors), this is flagged in dashboards. The decentralization metrics in §22–23 make this drift visible, not hidden.

These tensions are not defeaters of the thesis; they are **boundary conditions**. The stack does not promise to solve all social and political problems. It promises:

1. Technical guarantees that are real within their scope.
2. Telemetry that makes boundary violations visible.
3. Economic and architectural design that expands the scope of those guarantees over time.

Chapter 6

Layers of the Cypherpunk Stack

The threat model makes one thing clear: this is not a single-layer contest between "good" and "bad" money. It is a contest across an entire stack of machinery, from silicon and power all the way up to governance and law. States under stress will not only try to tax balances or censor individual transactions; they will reach for routing, hardware mandates, identity schemes, update channels, and legal definitions of "safety." If we want Privacy, Proofs, and Compute to behave like monetary primitives rather than platform features, we have to meet them on all of those planes.

It is useful, therefore, to make the architecture explicit as a set of layers. We already speak of **Layer 0: Verifiable Machines**, which means open hardware and sampled supply chains as base reality. Above that, the thesis has described triad-centric layers in terms of Create/Compute → Prove → Settle → Verify, with Telemetry & Governance threaded through. We can now refine this into a seven-layer stack that stretches from matter to institutions:

- **Layer 0:** Verifiable Machines (and power)
- **Layer 1:** Reachability (communications and transport)
- **Layer 2:** Distribution & Execution (software supply and runtime)
- **Layer 3:** Identity & Claims (who may act, without doxxing)
- **Layer 4:** Truth & Work (proofs, VerifyPrice, and useful compute)
- **Layer 5:** Value & Settlement (privacy rails and non-custodial flow)
- **Layer 6:** Governance & Telemetry (keeping the system honest)

The rest of the document can then be read as a tour through these layers.

Layer 0 remains the bedrock. **Verifiable Machines** are the point where cryptography stops being metaphor and touches matter. The chips that sign, prove, randomize, and capture must not be opaque priesthood artifacts; they must be at least partially inspectable and sampled. That means open RTL or microarchitectures where we can get them, open PDKs where politics allow, and structured lot-sampling and imaging where we cannot. It also means treating power and physical plant as part of Layer 0 rather than an afterthought: a prover farm or router that cannot withstand a planned blackout is as brittle as a closed chip. In practice, Layer 0 is "verifiable machines on verifiable power," which means open designs, measurable side-channel budgets, lot attestations, and micro-grids or backup arrangements that keep the hardware honest and powered when stress arrives.

On top of that sits **Layer 1: Reachability**. If Layer 0 asks "can we trust what the machine does?", Layer 1 asks "can we talk to it at all?" A repression toolkit that has already learned to pull levers on subsea cables, IXPs, DNS roots, and mobile carriers will aim squarely at transport. Here the goal is simple: make it hard to turn the network off without turning the country off. Concretely, that means encrypted transports that blend into ordinary traffic, obfuscation that defeats naive DPI and active probing, alternate paths through satellite and radio when fiber is throttled, and enough diversity of routes that there is no single "kill switch." In the stack metaphor, Layer 1 is the oxygen line—if it fails, proofs and privacy become academic. The communications section we introduce later makes this concrete: Tor-class pluggable transports, encrypted handshakes, refraction networking where cooperators exist, plus sat/mesh/sneakernet fallbacks all belong here.

Layer 2: Distribution & Execution answers a quieter but equally potent adversary move: "if we cannot block the packets, we will block the programs." App stores, CDNs, corporate MDM policies, and automatic update systems are powerful levers. A network whose clients can only be installed or updated with the blessing of a small set of platforms is not neutral; it is merely waiting to be deputized. Layer 2 is where we insist that the code that speaks triad-money can circulate without anyone's permission. Content-addressed binaries, signed manifests mirrored across jurisdictions, USB/QR/audio installers, and runtimes that can be fetched and verified over whatever transport is available are the practical elements. Execution environments matter too: if the only viable runtime is a locked-down phone OS wired to one app store and one identity scheme, the repression toolkit has already won. The software distribution section we add later sits here, making update and execution independence a first-class requirement.

If Layer 2 makes sure the code can run, **Layer 3: Identity & Claims** makes sure it can express "who" is acting without collapsing into real-name KYC. The adversary here is the growing web of SIM registration, network identity, and "safety" regimes that treat anonymity itself as suspicious. The stack does not deny that some actions must be accountable; it refuses to equate accountability with global doxxing. At this layer, keys, credentials, and policies are defined in a way that allows entities to prove the right to act (spend, sign, operate a prover) without binding that right to a civil identity unless absolutely necessary. Anonymous or pseudonymous credentials, selective-disclosure proofs, and reputation linked to receipts and behavior rather than to phone numbers belong here. When we later speak of viewing keys, lawful-privacy corridors, and audit-friendly receipts, we are concretely filling in Layer 3.

Only once machines are honest, packets can move, code can run, and actors can be named without being exposed do we reach **Layer 4: Truth & Work**, the core of the original thesis. This is where proofs, VerifyPrice, and useful work live. The job of this layer is to answer, cheaply and publicly, the question "did this actually happen as claimed?" It is here that verification asymmetry is engineered: MatMul-PoUW constructions and ZK proof systems that make checking an order of magnitude cheaper than doing; harnesses that measure p50/p95 verifier time and

cost for each workload; and markets that pay only for receipts that pass those checks. In the earlier language of the loop, Layer 4 is the Prove/Verify spine: the circuits, proofs, and reference verifiers that turn "trust me" into "verify me" at scale. Much of the existing text (Verification Asymmetry & VerifyPrice, Proof Factories, AI-PoUW, the modular stack) already describes this layer in detail.

With Layer 4 in place, **Layer 5: Value & Settlement** can safely be money-like. This is where privacy rails, atomic swaps, shielded pools, and cross-chain bridges live. Its task is double: move value without custody or censorship, and do so in a way that preserves privacy by default while leaving a path for auditable disclosure. Here the adversary is both the classical financial repression toolkit (negative real yields enforced through custodians and captives) and the newer chokepoints of KYC'd exchanges, surveilled payment processors, and compliant stablecoin issuers. The response is not rhetoric; it is architectures that keep custody at the edge, settlement neutral in the middle, and content private unless a viewing key is invoked. Adaptor-signature swaps between BTC and privacy assets, lawful-privacy corridors for treasuries, bridge designs that avoid trusted relays, and settlement telemetry (success rates, refund safety, anonymity-set health) are the concrete pieces. The Privacy Rails sections, and the new communications and identity work, all culminate here.

Finally, **Layer 6: Governance & Telemetry** closes the loop. If the lower layers are the nervous system, this is the cortex and immune system: how the network notices drift, resists capture, and adapts under stress. Without this layer, even beautifully engineered proofs and privacy corridors will decay into the usual pattern of oligopolies and black boxes. In practice, Layer 6 is where VerifyPrice observatories, decentralization dashboards, corridor-health monitors, and incident reports live, along with the institutions that respond to them. It is also where operational and legal posture is set: how routers are constrained so they cannot quietly privilege "house" flow; how upgrades are staged so Layer 0 and Layer 4 stay aligned; how incident response works when a corridor breaks or a hardware profile is compromised; how policy engagement happens without creating a single political point of failure. The mantra "no dashboards, no trust" belongs to this layer: if the public cannot see verification cost, neutrality, censorship, and settlement health, the system has already slid back toward platform trust.

Seen this way, the cypherpunk monetary stack is not one clever consensus mechanism but a ladder of seven interlocking layers. Layer 0 and Layer 1 keep the silicon honest and the packets flowing. Layer 2 and Layer 3 ensure that code and identity cannot be quietly turned into chokepoints. Layer 4 turns work into truth with receipts anyone can verify. Layer 5 turns those receipts into private, non-custodial settlement. Layer 6 watches the whole organism and pushes it back toward neutrality when it drifts. The sections that follow simply walk this ladder: beginning with verifiable machines, extending upward into communications resilience, distribution, and identity, and then re-entering the territory already mapped (proofs, compute, settlement, governance) with a clearer sense of what each layer must survive.

6.1 Modules in Outline: Applications and Primitives

The layered picture tells us **where** things must live; the modular picture tells builders **what** they actually touch. We will use four reference applications as running examples throughout the rest of the thesis:

1. **Private treasury & payroll:** Pays staff and vendors non-custodially over privacy rails, emits receipts that auditors can check, and treats "who got paid what" as a matter of viewing keys, not public gossip.
2. **Media provenance & authenticity:** Cameras, sensors, and editors emit cryptographic lineage that survives platform stripping; payments to creators and data providers are conditioned on the presence of such receipts.
3. **Verified inference:** A market where model owners and service providers sell inferences that come with succinct proofs or hybrid proofs-of-logits, all priced and measured via VerifyPrice.
4. **Proof/compute procurement:** A rail where developers and treasuries can buy standardized units of "proofs" and "verified FLOPs" as futures or spot capacity and treat them as balance-sheet items next to BTC.

To support these, we rely on a small set of reusable primitives:

- A **Proofs-as-a-Library SDK (PAL)** that lets developers declare claims ("prove this computation," "prove this provenance") and compiles them to multiple proving backends and useful-work miners.
- A **Privacy Rails Kit (PRK)** that expresses pay-for-proof intents and executes refund-safe settlement over BTC↔ZEC/XMR corridors or shielded pools.
- A **Proof Interface Definition Language (PIDL)** that defines the minimal receipt that binds a claim, a proof, a workload ID, an SLA tier, and timestamps into a portable object.
- A **VerifyPrice observatory** that provides public metrics for how cheap verification actually is for each canonical workload.
- A set of **market and telemetry primitives** (SLA escrows, neutral routers, bridge-safety templates) that make it possible to treat proofs, privacy capacity, and verified FLOPs as commodities rather than favours.

Later sections turn this into a full "substrate kit" of twelve primitives and four reference applications. Until then, we will use the names PAL, PIDL, privacy-rails kit, proof factories, and VerifyPrice observatory as shorthand for these building blocks.

Part II
The Triad as Monetary Base

Part I argued that the old guarantees are failing under debt arithmetic, AI, and platform control, and that a seven-layer cypherpunk stack is needed to survive repression. This Part looks at the same system from the **monetary angle**.

The claim is straightforward: **Privacy, Proofs, and Compute can be treated as a monetary base for a dense digital civilization.** They are not simply features of a protocol; they are verifiable necessities that economic actors must keep buying every cycle. This Part explains how that triad behaves as a store of value (SoV), and how **Work Credits** turn verifiable work into energy-anchored monetary instruments (Private Money and AI Money).

Chapter 7

The Triad as Monetary Base

Traditional monetary regimes pick a **base reality** and build promises on top of it:

- Gold: geology + metallurgy.
- Fiat: law, taxation, and war.
- Bitcoin: thermodynamics and code.

Each regime implicitly answers two questions:

1. *What counts as real work?*
2. *What object or capacity will we treat as the canonical memory of that work?*

In a civilization where intelligence is largely machine-executed and verification can be cheap and public, the answer shifts: **work is what machines can do and humans can verify cheaply**, and the "object" that records that work no longer has to be a metal or a pure ledger entry. It can be a *capacity*.

This thesis treats three capacities as monetary primitives:

- **Privacy**: censorship-resistant settlement that preserves agency.
- **Proofs**: portable attestations of computation and provenance.
- **Compute**: useful work wrapped in succinct guarantees.

Each is:

- **Indispensable** in a dense digital economy (no safe commerce without privacy; no safe coordination without proofs; no AI without compute).
- **Verifiably scarce** at any point in time (bandwidth, cycles, proof capacity are bounded by physics and capital).
- **Cheap to verify** (you can check whether you have privacy, a valid proof, or a verified FLOP without trusting a platform).

The **monetary base** in this frame is not "the token" but the **networked capacity to deliver Privacy, Proofs, and Compute under adversarial conditions**. Tokens, credits, and instruments are just ways of slicing claims on that capacity.

A few distinctions help:

Base vs. wrappers: The *base* is "how much verifiable Privacy/Proofs/Compute per unit time this stack can deliver at target SLOs." *Wrappers* are things like Work Credits, corridor tokens, and staking positions that slice this capacity into transferable claims.

Nominal vs. real: Nominal pricing of the triad will oscillate in terms of fiat and BTC. Real value

is "does this capacity still buy me censorship-resistant settlement, proofs, and verified FLOPs when repression and AI get worse?"

Symbol vs. utility: Gold and BTC partly trade as symbols. The triad trades as **plumbing**: "can I still pay people, prove things, and run intelligence without asking permission?"

From a monetary perspective, the key property is that **the world must keep buying the triad's utility**:

- Treasuries and individuals purchase **private settlement** to escape surveillance and yield-curve control.
- Platforms, enterprises, and states purchase **proofs** to secure provenance, compliance, and audit trails in an AI-polluted information environment.
- AI labs, agents, and applications purchase **verified compute** to sell trustworthy services.

The claim is not that the triad *replaces* all money, but that it behaves like a **reserve asset**: a base layer of verifiable capacity that other instruments reference, hedge with, or settle into. Private Money and AI Money are just two lenses on how that base is held and used.

7.1 Monetary Objects and Value Capture

The thesis claims that triad capacity can earn a "durable store-of-value premium." For that claim to be testable, we must answer four questions without poetry:

1. **What exactly is the asset?**
2. **What do users pay in?**
3. **How does holding capture value?**
4. **Why doesn't value leak entirely to operators?**

Question 1: What is the asset? The triad stack can issue several kinds of holdable instruments:

Instrument	What It Represents	Scarce?	Transfer?	SoV?
Base Token	Native unit; required for fees, staking, governance	Yes	Yes	Primary
Work Credits	Claims on verified capacity; minted against work	Yes	Yes	Secondary
LP/Staking	Rights to fee share from corridors, pools, validators	Yes	Sometimes	Derivative
Capacity Vouchers	Prepaid access at fixed rates	No	Yes	No
PIDL Receipts	Proof that work was done	No	Yes	No

Table 7.1: Triad instruments and their SoV candidacy.

The **primary SoV candidate** is the **base token**: the unit in which fees are paid, burns occur,

and collateral is posted.

Question 2: What do users pay in? All triad services are priced in the **base token**:

- **Proofs:** Pay base tokens to provers; portion burned.
- **Privacy settlement:** Pay base tokens for corridor fees; portion burned.
- **Verified compute:** Pay base tokens for inference/MatMul; portion burned.
- **Staking/collateral:** Provers, routers, and LPs must lock base tokens to participate.

This creates **structural demand**: every use of the triad requires acquiring the base token.

Question 3: How does holding capture value? Value accrues to holders through four channels:

1. **Fee burns** (30–50% of fees): Deflationary pressure proportional to usage.
2. **Staker yield** (20–40% of fees): Income for holders who stake.
3. **Operator share** (20–40% of fees): Incentive for provers/routers/LPs.
4. **Collateral requirements** (10–20% of capacity value): Locks supply proportional to network size.

Question 4: Why doesn't value leak entirely to operators? The "utility token trap" occurs when operators capture all economic value while token holders merely provide exit liquidity. The stack avoids this through:

- **Fee burns:** 30–50% of fees are permanently destroyed, benefiting all holders.
- **Staking yields:** Passive holders can stake to earn fee share without operating infrastructure.
- **Collateral requirements:** Operators must hold significant tokens, aligning their interests with holders.
- **Governance rights:** Token holders vote on fee splits, issuance changes, and protocol upgrades.
- **Issuance constraints:** New tokens cannot be minted arbitrarily; issuance is tied to capacity growth.

Net effect: At steady state, fee burns $\geq$ new issuance, so total supply is flat or declining. Holders benefit from both yield (if staked) and appreciation (via burns).

7.2 Three Reference Designs

To make the thesis testable, we commit to three concrete designs. Implementations may vary, but at least one must be viable for the SoV claim to hold.

Design A: Base Token as SoV (Primary)

- **Issuance:** Fixed schedule with halvings (like BTC) or capacity-linked ceiling.
- **Fee medium:** All triad services priced in base token.
- **Burns:** 40% of fees permanently destroyed.
- **Staking:** 30% of fees to stakers; 10–15% collateral requirement.
- **Operators:** 30% of fees to provers/routers/LPs.
- **Governance:** Token-weighted voting on parameters.

SoV properties: Credible scarcity (capped + burns), native demand (fees), duration-neutral (no coupons), cheap verification (VerifyPrice dashboards).

Risk: If demand stalls, burns decline and scarcity weakens.

Design B: Work Credits as Capacity Vouchers (Hedge Instrument)

- **Issuance:** Minted against verified work (FERs + proofs); no fixed cap.
- **Redemption:** Burnable for priority access to proofs/compute/settlement at SLA-guaranteed rates.
- **Expiry:** Credits decay or expire after N years to prevent hoarding and bank-run dynamics.
- **Transferable:** Yes, but primarily used for cost hedging, not long-term savings.

Use case: Enterprises hedge against compute cost spikes; treasuries lock in future settlement capacity.

Not a SoV: Supply expands with capacity; expiry prevents indefinite accumulation. This is infrastructure hedging, not a store of value.

Design C: Triad Index Token (SoV Candidate)

- **Backing:** Diversified revenue streams across privacy corridors, proof pools, and compute markets.
- **Value capture:** Protocol revenue used for periodic buyback-and-burn or dividend distribution.
- **Issuance:** Fixed supply; no new minting after genesis.
- **Governance:** Index holders vote on revenue allocation and rebalancing.

SoV properties: Diversified exposure to triad demand; fixed supply; yield via buybacks or dividends.

Risk: Concentration in specific corridors or proof markets; governance capture.

Which design does the thesis endorse? The thesis is compatible with all three, but **Design A (Base Token as SoV)** is the primary reference design because:

1. It is closest to the BTC model that has demonstrated SoV behavior.

2. Fee burns create clear, measurable scarcity.
3. Staking and collateral create structural demand beyond speculation.
4. VerifyPrice and telemetry make the value linkage auditable.

Design B is useful for enterprises but is explicitly *not* pitched as a SoV. Design C is viable but adds complexity (index construction, rebalancing).

The rest of Part II assumes Design A as the default when discussing "the asset" or "Work Credits as SoV." Where Design B semantics apply (vouchers, hedging), we will note it explicitly.

Chapter 8

Privacy as Private Money

Cash used to be **default private money**: anonymous, bearer, final on receipt. In a world of KYC'd banks, programmable payments, and networked surveillance, that role has decayed. At the same time, the repression playbook (negative real rates + capital controls) makes **bearer-like savings** economically necessary, not ideologically optional.

"Privacy" in this thesis is not romantic opacity; it is **the ability to hold and move value without chokepoints, plus the option to disclose on your own terms.**

Concretely:

Bearer-like holding: Keys, not accounts, define control. Custodians may exist, but custody is an *optional service*, not a mandatory chokepoint.

Non-custodial settlement: Cross-asset flows (e.g., BTC↔ZEC/XMR) execute as atomic swaps or corridor protocols; neither side needs to trust a centralized intermediary.

Auditable by consent: Viewing keys and structured receipts allow specific flows to be disclosed to auditors without exposing the entire graph.

When these properties hold, **private settlement capacity itself** starts to behave like a monetary asset:

- A treasurer facing capital controls is not just asking "what's the yield?" but rather, "*can I still get value to my people next year if on-/off-ramps are throttled?*"
- A dissident journalist or NGO cares less about upside and more about "*will this still be here and spendable if my local banks freeze?*"

> **Blockchain may be waiting for its SSL moment.** The analogy is suggestive: in 1994, putting credit card information on the Web seemed reckless until SSL made encrypted commerce viable. E-commerce grew into a multi-trillion-dollar industry once confidentiality became default. Today, most public blockchains are "public by default" the way HTTP was—every transaction visible, every address linkable. Privacy is plausibly the bottleneck for mass institutional adoption.

The institutional version is straightforward: no enterprise wants payroll, vendor rates, or treasury operations visible by default. The unlock is one-click, non-custodial BTC↔XMR/ZEC with refund-safe UX and clear settlement analytics—privacy as a product, not a promise.

In that environment:

- The **rails** (privacy corridors, shielded pools) earn fees for providing unseizable, auditable settlement capacity.
- The **claims** on those rails (e.g., corridor LP positions, Work Credits) can be held as a store of value.

You can think of **Private Money** as:

> *"Rights to future, censorship-resistant, auditable settlement capacity."*

The SoV properties from Chapter 4 map naturally:

- **Credible scarcity.** Capacity is constrained by bandwidth, cryptographic verification costs, and capital at risk.
- **Cheap public verification.** Anyone can verify that a transaction was correctly formed, swapped, or refunded.
- **Censorship-resistance & portability.** Flows ride over non-custodial corridors; exit options span multiple assets and jurisdictions.
- **Neutrality & permissionlessness.** Adversarially diverse relays, routers, and LPs; public metrics on concentration.
- **Native demand.** Demand is created by repression itself.
- **Lawful privacy.** Viewing keys + PIDL receipts mean regulated entities can prove compliance without deanonymizing entire networks.
- **Duration-neutrality.** Claims participate in fee flows and scarcity premiums, not fixed coupons.

On a balance sheet, Private Money can coexist with BTC and fiat, but it plays a different role:

- **BTC**: thermodynamic base, global risk asset, macro hedge.
- **Fiat**: near-term unit of account, legal tender, credit medium.
- **Private Money**: repression-hedged rail, a way to ensure you can still pay and be paid without being fully seen.

Chapter 9

Proofs as Truth Money

If Privacy is the hull, **Proofs** are the **receipts**.

In a world of synthetic media, platform moderation, and "liar's dividend," the scarce thing is no longer *content* but **trustworthy provenance and computation history**. Proofs are the mechanism layer that answers, cheaply and publicly:

> *"Did this actually come from where it claims, and did the computation actually run as stated?"*

Clarification: Proofs vs. Proof-Backed Instruments Proofs themselves are *infinitely replicable outputs* once generated. A ZK proof can be copied and verified anywhere. On its own, a proof is not a scarce bearer asset—it is an attestation.

What *is* scarce is:

- **The ability to produce valid proofs at scale** (requires compute, hardware, energy).
- **Rights embedded in instruments** that reference proofs (Work Credits, staking positions, capacity claims).

So we distinguish three roles proofs play in the monetary stack:

1. **Proofs as a commodity market:** Standardized attestations priced by VerifyPrice. You buy proofs the way you buy bandwidth—as a priced input to operations.
2. **Proofs as collateral enablers:** Proofs make *other* contracts collateralizable. An inference SLA backed by proofs of correct execution can be used as collateral because the proof makes default detectable.
3. **Proof-backed instruments:** The "money-like" object is not "a proof" but an instrument whose integrity is enforced by proofs—e.g., Work Credits tied to verified proof workloads.

"Truth Money" in this thesis refers primarily to **proof-backed instruments and claims on proof capacity**, not to proofs themselves.

From a monetary perspective, two features matter:

1. **Proofs travel:** A PIDL receipt can be moved across systems, archived, or collateralized. It outlives any single platform's UX, TOS, or reputation.
2. **Proofs can be priced and standardized:** Once you know the VerifyPrice of a workload, you can treat "valid proof of workload W" as a commodity unit.

This gives rise to **Truth Money**:

- Media platforms, insurers, and courts demand proofs of origin, editing history, and custody.
- AI services demand proofs that a model of a given hash ran on given inputs with given bounds.
- Enterprises demand proofs that compliance computations actually ran.

Initially, proofs are purchased as **opex** ("we pay per proof"). Over time, markets will create **claims on future proof capacity**:

- Reservations or futures on proof-of-provenance capacity for a media network.
- Proof pool shares that entitle holders to a portion of fees from high-value workloads.
- Work Credits minted against proof workloads that meet certain VerifyPrice and SLO thresholds.

These instruments function as **Truth Money**:

> *"Rights to future, standardized, verifiable attestations about data and computation."*

From the triad perspective:

- Privacy protects *who* is involved.
- Proofs protect *what* actually happened.
- Compute powers *how* we arrive at outputs.

Truth Money is what you hold when you believe the long run belongs to systems where **verification, not vibe, mediates trust**.

Chapter 10

Compute as AI Money

Compute has always been an economic input: first as muscle, then as steam and electricity, now as FLOPs. The AI boom made this explicit: GPUs, TPUs, and datacenters are priced like oil fields.

But raw compute is **not yet money-like**:

- Most compute markets are opaque capacity rentals (cloud contracts, colo deals).
- There is no standardized unit of **verified work**; only hours and instance types.
- There is no cheap, public way to verify that a claimed computation actually ran correctly.

The triad reframes compute as **AI Money** by insisting on:

- **Canonical workloads:** Matrix multiplications, FFTs, and core model primitives.
- **Proofs of useful work (PoUW):** Miners/provers earn rewards for producing proofs that these workloads ran correctly, with verification asymmetry.
- **Verification economics:** VerifyPrice turns "one unit of verified MatMul at dimension n" into something that can be priced and traded.

In that context, **verified compute capacity** becomes a monetary primitive:

> *"Rights to future, standardized, cheaply verifiable units of useful compute (verified FLOPs)."*

10.1 Compute Deflation and What Remains Scarce

A skeptical reader will object: *"Compute gets cheaper every year. Moore's Law drives raw FLOP cost down. How can 'rights to compute' be a store of value when the underlying commodity deflates?"*

This is a serious objection. Here is the response:

What deflates:

- Raw FLOPs per dollar (secular decline).
- Cost of unverified, permissioned compute from hyperscalers.
- Commodity inference for non-sensitive workloads.

What remains scarce:

- **Verified FLOPs with receipts:** Compute where correctness is cryptographically proven, not vendor-asserted.
- **Policy-constrained capacity:** Compute that runs under specific jurisdictional, privacy, or compliance constraints.
- **Censorship-resistant access:** Compute that cannot be denied by TOS changes, sanctions, or platform bans.
- **Priority access under congestion:** During demand spikes, priority slots are scarce even if raw capacity is abundant.
- **Specific hardware profiles:** Compute on L0-C or L0-D grade hardware may remain scarce even as closed alternatives proliferate.

The instrument should reference capacity share, not "one timeless FLOP": AI Money is not "rights to 1 FLOP forever." It is:

> *"Rights to X% of verified compute throughput under SLA Y on hardware profile Z."*

This is analogous to how oil futures reference "barrels of WTI crude delivered at Cushing"—not "energy" in the abstract. The specificity (verification, SLA, profile) is what creates persistent scarcity.

Telemetry that detects deflation risk: If raw compute deflation outpaces the scarcity premium of verification/censorship-resistance, the following metrics will signal it:

- VerifyPrice for verified FLOPs converges toward unverified market rates.
- Capacity utilization on verified networks falls below thresholds.
- Fee revenue from compute workloads declines as a share of total fees.

These are observable. The thesis predicts that the verification premium will persist because (a) demand for trustworthy AI is structural, and (b) the compliance/sovereignty premium will grow as AI becomes more consequential. But if telemetry shows otherwise, the "compute as SoV" leg weakens relative to privacy and proofs.

AI Money instruments can include:

- **Work Credits**: minted against verified-compute contributions, redeemable for future workloads or tradeable as a savings asset.
- **Compute futures**: for specific workloads (e.g., "X verified inferences at model M with SLO Y").
- **Stake in proof factories**: whose entire business is converting energy + silicon into verified FLOPs with transparent VerifyPrice.

Why is this money-like rather than just "another utility token"? Because:

- **Demand is global and secular:** As long as AI is an input to value creation, there is demand for FLOPs.

- **Verification is public:** Anyone can check that a unit of AI Money corresponds to a valid proof.
- **It's duration-neutral:** No fixed coupon; value is backed by the ability to sell compute into whatever nominal budgets exist.

Chapter 11

Work Credits: Energy-Anchored Claims

The triad gives us three capacities. To turn them into concrete monetary instruments, we need a **unit of account for work**: something that binds energy, hardware, and verification into a transferable claim.

Call these **Work Credits** (**WC**).

Informally:

> *A Work Credit is a claim on a standardized unit of triad work (privacy settlement, proof generation, or verified compute) that has been produced and attested under public SLOs.*

Clarification. Throughout this section, "Work Credits," "AI Money," and "proof money" are generic, notional instruments (a class of designs that any future chain or protocol could implement), not product branding for a particular network. The point is to specify what *kind* of asset can sit on top of the triad, not to name one specific chain or ticker.

We can describe Work Credits along a few axes:

11.1 Definition: Work Receipts vs. Work Credits

A common source of confusion is conflating "proof that work was done" with "transferable claim on future capacity." These are different financial objects. We split them cleanly:

Work Receipt (WR)

A **Work Receipt** is a PIDL artifact proving that a specific unit of work was completed under attested conditions.

- **Content:** Claim hash, proof hash, workload ID, SLA tier, timestamps, hardware profile, prover signature.
- **Properties:** Copyable, verifiable by anyone, *not scarce*.
- **Analogy:** A receipt from a completed transaction. It proves the past but confers no future rights.

Work Receipts are *not* money. They are evidence.

Work Credit (WC)

A **Work Credit** is a transferable instrument issued against Work Receipts under protocol-defined issuance rules.

- **Issuance:** Minted when (a) a valid Work Receipt is accepted by the network, and (b) telemetry confirms SLOs are met.
- **Properties:** Scarce (supply bounded by issuance rules), transferable, fungible within workload class.
- **Rights:** Depend on the design variant (see below).

Variant	Rights Conveyed	Scarce?	SoV?
WC-Base (monetary)	No direct redemption. Represents historical verified work. Required for fees, staking, governance. Supply capped; burns create scarcity.	Yes	Yes (primary)
WC-Voucher (prepaid)	Redeemable for future proofs/compute/settlement at SLA-guaranteed rates. May expire or decay.	No (supply tracks capacity)	No (hedge instrume

Two design variants for Work Credits: **Recommendation:** The thesis adopts **WC-Base** as the primary reference design for SoV analysis. WC-Voucher is useful for enterprises hedging cost volatility, but is explicitly *not* pitched as a store of value.

Issuance mechanics for WC-Base: For a canonical workload W (e.g., "MatMul of size n with error bound ε," "provenance proof for content type C," "corridor settlement of size S with anonymity set $\geq A$"), define:

- $p(W)$: production cost (energy + hardware amortization + opex) to generate one unit and produce a valid proof.
- $v(W)$: verification cost.
- $r(W) = v(W)/p(W)$: verification asymmetry.

A **Work Credit of type W, tier T** is issued only when:

- A valid Work Receipt for workload W at tier T is accepted by the network.
- **Telemetry confirms** that VerifyPrice(W,T) and other SLOs (latency, failure rate, decentralization) are within bounds.
- Issuance does not exceed the **issuance envelope** for the current period (see Section 11.6).

The credit does *not* represent a "future option" on its own—that is the WC-Voucher variant. WC-Base represents **historical verified work** and derives its value from being the fee/staking medium, not from redemption rights.

11.2 Energy Anchoring

Like PoW, Work Credits are ultimately **energy-anchored**:

- The marginal cost of producing one more credit is bounded below by the energy and hardware required to pass the verification threshold.
- Unlike SHA-256 PoW, the work is **useful**: it powers privacy settlements, proofs of provenance, and AI computation.

This anchoring gives Work Credits:

- **Credible scarcity:** You cannot mint Work Credits without expending real resources to produce proofs and settle flows.
- **Economic meaning:** One credit corresponds to something the world actually cares about (anonymized payrolls, authentic media, verified inference), not just burned electricity.

11.2.1 Layer 0 Maturity and Economic Consequences

Work Credits are only as trustworthy as the hardware that produces them. A world where all proving runs on opaque, vendor-controlled hardware is different from one with diversified, partially open designs.

§14 (Part III) defines a **Layer 0 maturity ladder** with grades L0-A through L0-D. Here we preview how those grades affect Work Credit economics:

L0 Grade	Hardware Trust Level	Economic Treatment
L0-A	Closed hardware + attestation + multi-party audits + diversity	WC accepted at full value if diversity thresholds met; risk premium priced into fee schedules
L0-B	Lot sampling + imaging + bounded side-channel budgets	WC accepted at full value; lower risk premium
L0-C	Partial open (open RTL for critical components)	WC may qualify for "open-profile" premium or priority tiers
L0-D	Fully open designs	Highest trust tier; may command premium in markets that value sovereignty

How this affects issuance and pricing:

1. **Tiered issuance caps:** WC minted on L0-A profiles may face stricter issuance limits than L0-C/D profiles. This prevents the system from becoming dependent on opaque hardware.
2. **Market pricing:** WC from different L0 grades can trade at different prices if markets distinguish them. Treasuries and institutions may pay premiums for L0-C/D-backed capacity.
3. **Risk disclosure:** Every Work Receipt includes the hardware profile (HID) used. Aggregated telemetry shows what fraction of total WC is backed by each grade.

4. **Deprecation impact:** If an L0-A profile is compromised (TEE backdoor discovered), WC minted on that profile can be quarantined, discounted, or excluded from certain uses. This is the "bond downgrade" analog.

Why this matters for SoV: If 90% of Work Credits are minted on L0-A hardware and a major TEE is compromised, the SoV thesis takes a hit—not because the cryptography failed, but because the base assumption (honest machines) was violated.

By making L0 grade explicit and tying it to economic consequences, the system:

- Creates incentives to invest in open hardware.
- Bounds the impact of hardware compromises.
- Gives users and allocators the information to price risk appropriately.

This is covered in detail in §14.3.1 (Layer 0 Feasibility Ladder). The key point for Part II: **hardware trust is not binary; it is graded, measured, and priced**.

11.3 Robots, AI, and the Demand for Work Money

In a robot- and AI-heavy economy the demand story for Work Credits becomes almost embarrassingly straightforward.

Consider a warehouse or factory where most of the physical activity is carried out by robots, and most of the planning and oversight is handled by models. The day-to-day budget splits into:

- Energy to power robots and data centers.
- Compute to run models and generate proofs.
- Privacy and settlement to pay workers, suppliers, tax authorities, and investors without leaking trade secrets or exposing everyone's graph.

Each of those budgets expresses itself as a recurring need for receipts:

- Proof of correct inference for high-stakes decisions.
- Proof of compliance and risk calculations for regulators and insurers.
- Privately settled wages and vendor payments with lawful audit trails.

Today that flow is mediated through cloud bills, payroll files, bank wires, and audit PDFs. In the stack described here, it is mediated through Work Credits and receipts. The robots and services are paid in Work Credits; in exchange they produce receipts that can be verified cheaply and settled over privacy rails. The more of the economy runs this way, the more natural it becomes to hold savings directly in the unit that pays for these things instead of in a separate, compliance-backed currency that must be constantly converted.

This is one concrete way to read remarks like "money will just be energy" in a non-handwavey way. Energy, compute, and hardware are the inputs that enable robots and AI to work; Work Credits are the units in which their work is paid for and stored. The credits do not mag-

ically become Joules; they become a convenient way to denominate future access to verifiable, robot-mediated output.

11.4 Why This Is Still "Money," Not Just Coupons

If we revisit the seven SoV criteria from §3, the mapping is straightforward:

- **Credible scarcity.** Issuance is programmed and publicly documented, either as an explicit schedule (halvings, caps) or as a capacity-linked minting rule. Deviations are detectable on-chain and in telemetry.
- **Cheap, public verification.** Every unit of capacity that credits buy is backed by receipts that any honest party can verify under published VerifyPrice targets.
- **Censorship-resistance & portability.** Settlement uses non-custodial privacy rails and atomic swaps; Layer 1 and 2 ensure clients can reach the network under filtering; Layer 3 ensures users are not doxxed by default.
- **Neutrality & permissionlessness.** Open admission for provers, miners, routers, and mirrors is enforced by decentralization telemetry and fairness tests.
- **Native demand.** The demand for Work Credits comes from real workloads: proofs, verified FLOPs, private settlement, robotics and AI tasks. These are line items in budgets, not speculative self-reference.
- **Lawful privacy by design.** Viewing keys, receipts, and provenance proofs allow institutions to satisfy audits without re-introducing custodians or surveillance chokepoints.
- **Duration-neutrality.** Work Credits do not promise fixed nominal coupons that can be pinned negative by policy. Their value comes from the intersection of energy/hardware costs and buyer budgets for capacity.

Seen in this light, Useful Work Money is not a loyalty program or a scrip. It is a monetary instrument whose backing is precisely the capacities this thesis has been concerned with all along: privacy, proofs, and compute, grounded in verifiable machines and energy, and measured by VerifyPrice and FERs. It earns a store-of-value premium not because it tells a compelling story, but because in a dense digital civilization it is one of the few things the world must keep buying, on rails anyone can audit.

11.5 Monetary Role

Work Credits sit at the junction between **base capacity** and **financial representation**:

- For operators, they are **revenue in kind**: miners/provers/routers earn credits by contributing triad capacity.
- For users, they can be **pre-paid capacity** or **savings**: hold credits to ensure future access to triad services, or to speculate on increased demand.

Their SoV behavior depends on:

- **Demand for specific workloads:** Work Credits tied to high-value workloads (e.g., compliance proofs, LLM inference) may command higher premia.
- **Governance & telemetry honesty:** If VerifyPrice and decentralization metrics are falsified or gamed, the link between credits and real work weakens.
- **Repression intensity:** As repression rises, demand (and thus value) for Private Money credits (privacy/settlement workloads) and Truth Money credits (provenance workloads) should increase.

In portfolio terms, Work Credits are the **equity-like layer of the triad**; they express exposure to the growth and usage of Privacy, Proofs, and Compute under verifiable rules.

11.6 Issuance: Tying Credits to Real Capacity

There are many possible issuance schemes; what matters here is not choosing a particular curve, but enforcing two principles:

1. **Issuance is legible.**
2. **Issuance is constrained by real capacity.**

One extreme is the Bitcoin model: a fixed schedule, regardless of demand, with the understanding that price will equilibrate. Another extreme is a pure capacity-linked model: Work Credits are minted only when new proving, compute, and settlement capacity comes online and is registered with telemetry; they behave almost like tokenized capacity reservations. In practice, a hybrid is likely: a predefined issuance envelope over time, modulated by capacity growth and burn.

In such a system, adding a new proving cluster, plant, or corridor is not just a marketing slide; it is an event that expands the envelope of Work Credits the system can credibly support. The converse is also true: if plant retires or corridors die and are not replaced, issuance that continues on autopilot will show up as VerifyPrice drift, SLA breaches, and deteriorating energy metrics. The governance layer's job is not to guarantee any particular price, but to keep issuance and capacity in rough proportion and to make any departures visible.

From an allocator's perspective this yields a familiar pattern: Work Credits look like equity in (or claims on) a portfolio of infrastructure (proving farms, AI chains, privacy corridors, and plants) rather than a purely arbitrary balance sheet. The difference from conventional "infra tokens" is the insistence on receipts and KPIs: if claimed capacity and observable behavior diverge, the discrepancy is not a rumor; it is a datapoint.

11.6.1 The Supply Balance Equation

A critical question for any SoV claim is: **why does demand not simply expand supply, neutralizing scarcity?**

If Work Credits are minted whenever verified work is done, and demand for triad services grows, supply grows too. This could make WC behave like a **labor-backed scrip** rather than a scarce reserve asset.

The answer lies in the **net supply dynamics**. We can express this as a balance equation:

$$\Delta\text{Supply} = \text{Issuance} - \text{Burns} - \text{Lost} \pm \text{Governance}$$

Term	Definition	Typical Magnitude
Issuance	New WC minted against Work Receipts, subject to issuance envelope	Bounded by schedule or capacity ceiling
Burns	WC permanently destroyed when used for fees	30–50% of fee volume (reference design)
Lost	WC in lost/forgotten wallets	~1–2% of supply per year (empirical from BTC)
Governance	Adjustments via protocol upgrades (rare, requires supermajority)	Near zero in steady state

For WC-Base to behave as SoV, the design must ensure:

$$\text{Burns} + \text{Lost} \geq \text{Issuance} \quad \text{(at steady state)}$$

This means:

- **During growth phase:** Issuance > Burns, but capped by schedule/capacity. Supply grows, but at a predictable, declining rate (like BTC halvings).
- **At maturity:** Burns ≥ Issuance. Supply is flat or declining. Holders benefit from scarcity.

What prevents runaway issuance?

1. **Issuance envelope:** Total WC mintable per epoch is capped, regardless of work submitted. Excess work earns priority, not extra tokens.
2. **Capacity linkage:** Issuance envelope expands only when new verified capacity (FERs, hardware profiles) is registered. You can't mint by decree.
3. **Halving schedule:** Many designs use BTC-style halvings to ensure long-run issuance approaches zero.
4. **Governance friction:** Changes to issuance rules require supermajority and are visible in dashboards before activation.

Telemetry that detects supply risk:

- **Inflation rate:** Issuance / circulating supply. Should decline over time.
- **Burn rate:** Burns / fee volume. Should stay within target range.
- **Net supply change:** ΔSupply per epoch. Positive during growth, flat or negative at maturity.
- **Issuance vs. capacity:** If issuance grows faster than verified capacity, this signals potential dilution.

If these metrics drift outside healthy ranges, it becomes visible in dashboards—holders and operators can respond before the SoV thesis is undermined.

11.7 Energy-Priced, Not Energy-Pegged

In popular discourse one often hears that "money will just be energy," especially in the context of robotics and AI. As shorthand, this is attractive: robots and data centers run on electricity, so why not quote everything in kWh and be done with it? The problem is that not all kilowatt-hours are created equal. Time of day, grid node, reliability, carbon intensity, and siting constraints all affect their economic and political meaning. A winter-peaking kWh on a stressed urban grid is not the same as a curtailed hydro kWh in a remote valley. If we pretend otherwise, we smuggle a lot of hidden politics and risk into the unit of account.

The discipline in this thesis is to admit that complexity and route around it with **receipts**.

Energy is measured and wrapped into Facility Energy Receipts (FERs), which record, for each facility and time window:

- kWh in,
- kWh delivered to IT,
- heat reused,
- PUE/ERE/WUE,
- water use,
- carbon intensity,
- and outages/curtailments.

On top of that, we measure **verified work per kWh**:

- η_{vFLOP} for FLOPs,
- proofs-per-kWh for proof workloads,
- swaps-per-kWh for settlement.

VerifyPrice then tells us how much it costs, in time and money, for an independent verifier to check that a given amount of work was done.

The result is an implicit conversion path:

> **energy → FERs → verified work (proofs/FLOPs/swaps) → receipts → Work Credits**

We never pretend that *one Work Credit is one kilowatt-hour*; instead we make it easy for anyone to estimate, at any given time, **how many kilowatt-hours of which quality and where in the world** sit behind a portfolio of Work Credits, via the receipts.

Pricing in energy then becomes an **inference problem for markets**:

- Work Credits are implicitly energy-priced because their production and redemption depend on energy-intensive workloads whose cost curves are public.
- FERs and VerifyPrice make those curves legible without forcing a brittle kWh peg.

What distinguishes Work Credits from naive "energy tokens" is precisely this separation:

- The unit of account is denominated in **work**, not in kWh.
- Energy enters through FERs, η_{vFLOP}, and VerifyPrice, not through a one-dimensional peg.

Markets, regulators, and builders can look at those receipts and say, with some confidence, "a Work Credit currently corresponds to about this much capacity, with this energy and carbon profile, under these SLAs." That is enough to make the asset priceable and analyzable without forcing it into a crude energy standard.

11.8 Why Work Credits Are SoV, Not Just a Utility Token

It is easy to misread Work Credits as another "utility token" with a story stapled on top. §3 and Section 11.4 already showed that they inherit the full SoV checklist (scarcity, cheap public verification, censorship-resistance, neutrality, native demand, lawful privacy, and duration-neutrality) from the stack. Here we make that contrast explicit.

Four properties do most of the work:

1. **Issuance is constrained by energy and installed capacity.**
   Work Credits do not float on arbitrary governance votes. Issuance is bounded by measured energy and verifiable capacity: FERs and plant telemetry cap the budget of subsidized work over a given interval. To print more, someone has to build more verifiable machines or increase real capacity. Narrative cannot override thermodynamics.
2. **Demand is driven by budgeted workloads, not vibes.**
   The "use" of a Work Credit is paying for proofs, inference, settlement, and related workloads that already sit on OPEX lines. AI labs, exchanges, custodians, and regulators must keep buying these workloads each quarter, regardless of token price. That places Work Credits *upstream* of compulsory spend, not downstream of hype.
3. **Verification asymmetry keeps cheating costlier than honesty.**
   For each workload W, the verification ratio $r(W)$ is explicitly bounded: it is always much cheaper to check a PIDL receipt than to fake it at scale. A Work Credit backed by a portfolio of such workloads inherits this property: to counterfeit the base, you must fake receipts, and faking receipts is provably more expensive than doing the work. Most "utility tokens" rely on soft notions of "activity"; Work Credits rely on receipts that anyone can cheaply reject if

they are bogus.

4. **Returns are duration-neutral and fee-driven, not coupon-driven.**
   Work Credits do not promise fixed coupons or redemptions at par. Their economic value comes from a share of fee+burn and spread on a growing, compulsory workload budget. As more proofs, compute, and private settlements clear over the stack, more fees flow in and more units are retired. That makes them duration-neutral: they behave more like equity in a necessary utility than a bond whose real yield can be pinned negative.

Taken together, these properties push Work Credits into the store-of-value bucket described in §3 and Section 11.4:

- constrained by energy and capacity,
- demanded by necessity rather than fashion,
- protected by verification asymmetry, and
- paid in a fee stream tied to indispensable workloads.

In that sense they are not tickets to a theme park; they are metered claims on the **power plant** that keeps the park (and the rest of the AI + ZK economy) online.

Work Credits are "utility tokens" only in the most literal sense: they buy utility the world cannot stop buying. Under the SoV lens, that is exactly what you want: an asset whose backing is quantitatively visible in receipts and KPIs, and whose long-run value is rooted in capacities that must be purchased through every regime.

11.9 Economic Linkage: How Triad Demand Becomes Asset Value

This subsection answers the question that separates a *systems manifesto* from a *monetary thesis*: **Why does demand for Privacy, Proofs, and Compute raise the value of *holding* the asset, rather than merely rewarding *consuming* a service?**

The answer has four parts.

1. What exactly is the asset? The **asset** in this framework can take several forms, but all share a common structure:

- **Network tokens:** Native units of the protocol (analogous to ETH or BTC) that are required to pay fees, post collateral, and participate in governance.
- **Work Credits:** Claims on verified capacity, minted against energy-anchored work.
- **Corridor/Pool shares:** LP positions or staking rights in specific privacy corridors, proof factories, or compute networks.

What matters is that **all uses of the triad must flow through the asset**. You cannot get a proof, settle a private payment, or buy verified compute without either holding or acquiring the native unit.

2. Why is it scarce in a monetary sense? Scarce *capacity* does not automatically imply a scarce *asset*. The link is forged through:

Mechanism	How It Creates Monetary Scarcity
Capped issuance	Total supply follows a predefined schedule (e.g., halvings) or is bounded by capacity growth, not governance fiat.
Fee burns	A portion of every fee is permanently destroyed, removing units from circulation as usage grows.
Collateral lockups	Provers, routers, and LPs must bond assets to participate; this removes circulating supply proportionally to network activity.
Energy-anchored minting	New credits are issued only against verified work backed by FERs; you cannot mint by decree.

The result: **supply is bounded by physics (energy, hardware) and shrinks with usage (burns), while demand is driven by compulsory workloads.** This is the scarcity structure of a commodity, not an IOU.

3. Why does demand for triad capacity raise the value of holding the asset? The causal chain:

1. Demand for Privacy/Proofs/Compute
2. ↓ Users must acquire native tokens to pay fees
3. ↓ Fees paid → portion burned, portion to stakers/provers
4. ↓ Burns reduce supply; staking rewards require holding
5. ↓ Increased demand + reduced supply → price appreciation
6. ↓ Holding the asset captures the economics of the triad

More concretely:

- **Fee revenue:** Every proof, every private settlement, every verified inference pays a fee denominated in the native asset. This creates continuous buy pressure.
- **Burn mechanics:** A fraction (e.g., 30–70%) of fees is burned. As throughput grows, more units are destroyed than minted, creating deflationary pressure during growth phases.
- **Required collateral:** Provers, routers, and LPs must stake tokens proportional to their capacity. This locks supply and aligns incentives with network health.
- **Priority/governance rights:** Holding grants access to premium SLA tiers, governance votes, and first-mover allocation of scarce capacity.

Key insight: The holder is not buying "exposure to price" (reflexive speculation). The holder is buying **a share of the fee stream from indispensable workloads**, denominated in an asset whose supply shrinks as those workloads grow. This is closer to equity in a utility than to a collectible.

4. What prevents capacity providers from capturing all value while holders get diluted? This is the classic "utility token trap": if operators earn all the fees and governance can inflate supply, holders are just exit liquidity.

The stack avoids this through:

Risk	Mitigation
Operator rent extraction	Fees are split: burn + stakers + provers. Operators cannot capture 100%; a structural portion goes to asset retirement.
Governance inflation	Issuance is constrained by FERs and capacity telemetry. Minting beyond real capacity triggers SLO breaches visible in dashboards.
Holder dilution	Burns offset new issuance. Net supply is designed to be flat or declining during steady-state usage.
Value leakage to fiat	Core operations (fees, collateral, rewards) are denominated in the native asset, not fiat. Fiat is an off-ramp, not the unit of account.

The economic design goal: At steady state, fee burns $\geq$ new issuance, so holders benefit from both yield (staking rewards) and capital appreciation (supply contraction).

Summary: The Value Accrual Lemma

> Demand for triad capacity (Privacy, Proofs, Compute) translates into store-of-value premium for the native asset because:
>
> 1. **All usage requires the asset** (fee medium).
> 2. **Usage burns supply** (deflationary pressure).
> 3. **Capacity provision requires staking** (supply lockup).
> 4. **Issuance is energy-constrained** (no governance inflation).
> 5. **Workload demand is structural** (AI, commerce, compliance budgets, not hype cycles).
>
> If these five conditions hold, and VerifyPrice/Reach/Settle remain within SLOs, holding the asset captures the economics of the triad rather than merely consuming a service.

This is the bridge between "these capacities are indispensable" and "therefore they can function as monetary primitives and earn a durable store-of-value premium."

Chapter 12

Monetary Design Space

Once Work Credits exist as a generic mechanism for binding work to claims, we can sketch the **design space of monetary instruments** built on top of the triad.

12.1 ZK Money

ZK Money instruments primarily reference **Privacy + Proofs**:

- Claims on shielded settlement capacity (corridor LP positions, privacy-rail credits).
- Claims on zero-knowledge proof capacity (e.g., SNARK/STARK service tiers).

They behave like:

- **Private Money**: bearer-like, censorship-resistant settlement rights.
- Plus **Truth Money**: bundled attestations of compliant behavior.

ZK Money is attractive to:

- Treasuries under repression.
- Regulated actors who need both privacy and attestable compliance.
- Individuals who want "cash-like" assets in digital form.

12.2 AI Money

AI Money instruments primarily reference **Compute + Proofs**:

- Claims on verified FLOPs.
- Futures on model-specific inference capacity with proof guarantees.
- Equity-like positions in proof factories whose output is standardized, verified compute.

They are attractive to:

- AI labs and application builders wanting hedges against compute shortage.
- Investors who believe "AI demand will outlive this particular model cycle."

12.3 Hybrid Instruments

Hybrids reference all three legs:

- **Triad baskets:** Index-like instruments backed by diversified Work Credits across privacy, proof, and compute workloads.
- **Corridor-linked AI Money:** Instruments that couple private settlement rights with AI inference capacity.
- **Agent treasuries:** On-chain agents that hold combinations of ZK Money and AI Money to fund their own operation.

The key design constraint is always the same: **claims must stay tightly coupled to verifiable work and capacity**, not to governance mood or marketing narrative.

Chapter 13

ZK Money and the ZK + AI Economy

ZK is the accounting engine of this stack. If privacy is the right to speak softly and compute is the ability to think loudly, zero-knowledge proofs are the receipts that make both negotiable. They turn "trust me" into "verify me" without exposing the underlying state. A world that routes value and decisions through proofs rather than paperwork is, in effect, a **zk-economy**.

A zk-economy is not just "more proofs." It is day-to-day life running on attestations: every payment, inference, bridge, and audit backed by succinct evidence that any party can check.

13.1 Why Open Hardware Is a Precondition for the ZK-Economy

If the prover can cheat, the proof is theater.

Today's largest ZK systems run on stacks that are almost entirely closed: proprietary GPUs, opaque microcode, black-box TEEs, firmware that can be updated silently overnight. In that world, a "fast prover" with a hidden trapdoor can mint convincing bogus proofs. A government-mandated "secure enclave" with a secret backdoor can exfiltrate witnesses from supposedly private circuits.

You get math-flavored trust, not actual trust.

A genuine zk-economy therefore begins one layer below the circuits, with **verifiable machines** (Layer 0). At least part of the proving path must be open from the RTL up through the software stack.

We will never get perfect certainty about every chip in circulation, but we can move to a regime where:

> *"This proof came from this class of machine, built from this design, and it would have been extraordinarily expensive to tamper with it without being caught."*

That is enough to make "trust the hardware" a falsifiable claim instead of a sacrament.

Concretely, a zk-economy that wants to underpin **ZK Money** and **AI Money** needs three things from Layer 0:

- **Open proving paths:** For canonical circuits, at least one proving stack must be open from RTL through firmware and prover binaries.

- **Sampled, attestable devices:** Lots can be sampled with verifiable randomness, imaged, and subjected to structured tests.
- **Hardware-level SLOs for proof honesty:** Profiles with weak sampling or poor audit history should be priced differently.

13.2 How This Opens a ZK + AI Economy

The same logic extends to AI. If proofs are the receipts of the digital order, AI is the industrial plant that consumes energy and data and emits capabilities.

The zk + AI economy is built on three interacting strata:

1. **AI blockchains (“model chains” – AI Money substrate):** Duplex-style and Ambient-style PoUW systems treat model training and inference as work functions. Work Credits minted against these workloads become **AI Money**.
2. **ZK blockchains (“proof chains” – ZK Money substrate):** Nockchain-style zk-PoW systems coordinate global prover markets. They mint **ZK Money** (claims on future proof capacity).
3. **Privacy rails as connective tissue (Private Money substrate):** BTC↔XMR/ZEC swaps, shielded pools, and private rollups let capital move without custody.

The daily-life version looks almost boring:

- Your phone proves you paid a toll **without revealing your route**.
- A DAO buys inference on a model chain; a proof chain clears the resulting proofs; payment moves over privacy rails.
- An exchange settles cross-chain obligations with batches of zk-proof-backed settlement receipts.
- An insurer prices climate risk from sensor networks whose readings are baked into proofs from open hardware profiles.

Underneath the surface, three demand curves reinforce one another:

1. **Proof demand grows** as systems move from “trusted unless flagged” to “untrusted unless proven.”
2. **Verified compute demand grows** as AI saturates workflows.
3. **Privacy demand grows** as financial repression and surveillance tighten.

We can view these as three economic flywheels that, if the thesis is right, will spin up over time:

Proof flywheel. More systems demand proofs → prover markets deepen → VerifyPrice falls or stabilizes → more systems can afford to demand proofs.

Compute flywheel. More AI in workflows → more willingness to pay for verified compute → more Work Credits minted against AI workloads → more capital available to fund model chains.

Privacy flywheel. More repression and surveillance → more demand for lawful private rails →

deeper anonymity sets and corridor liquidity → cheaper and safer private settlement → more users adopt privacy by default.

Open hardware is what keeps these flywheels from collapsing into a handful of "trust me" platforms. zk-PoW is what coordinates proof work into a measurable, meterable commodity. AI-PoUW is what turns model work into an asset class whose revenues are natively tied to **triad usage**. Privacy rails are what keep capital flowing between them without recentering custody.

Put in one line:

> **Open hardware gives us honest machines. ZK turns that honesty into portable guarantees. AI gives us something valuable to spend that honesty on. zk-PoW and AI-PoUW are the mechanisms that weld the three into an economy rather than a collection of clever demos.**

Chapter 14

SoV Evaluation Framework

Later, Part V will present an operator/investor checklist that spans the whole stack. Here we sketch the **monetary lens**: how to evaluate whether a particular triad-based instrument is a credible store of value.

For any candidate instrument X (a Work Credit type, a ZK Money basket, an AI Money token), ask:

1. **What triad workloads back it?**
   - Exactly which Privacy, Proof, and Compute flows does X reference?
   - Are those workloads canonical and well-specified?
2. **How is issuance tied to work?**
   - Is issuance formulaic and transparent (e.g., one credit per verified unit of W)?
   - Can governance mint outside of those rules? Under what constraints?
3. **Can anyone verify the backing?**
   - Are VerifyPrice and workload telemetry public?
   - Can a third party recompute or sample the proofs/settlements that justify supply?
4. **How does it behave under repression?**
   - If real yields are -300 bps for 3 years, what happens to demand for X?
   - If on-/off-ramps are throttled, can X still be acquired, held, and spent?
5. **Is privacy lawful by design?**
   - Are there clean paths for voluntary disclosure that do not reintroduce custody or KYC chokepoints?
   - Are the receipts (PIDL) expressive enough for auditors?
6. **What is the duration profile?**
   - Does X promise fixed nominal coupons (bond-like) or variable participation in fee flows (equity/commodity-like)?
   - Can a sovereign push its real return negative by fiat?
7. **Who can be locked out?**
   - Is access to X gated by one jurisdiction, cloud, or app store?
   - Do Layer-0/1/2 assumptions create hidden chokepoints?

A triad-based instrument that scores well on these questions behaves like a **next-generation SoV**:

- It is backed by verifiable work, not by decree.
- Its value is rooted in capacities that a digital civilization must keep buying: privacy, proofs, and compute.
- It is measurable under stress via telemetry, not defended by rhetoric.

The rest of the thesis will shift back to the **stack angle** (Layers 0–6) and the **telemetry angle** (VerifyPrice/Reach/Settle, decentralization metrics, governance). But the monetary picture stays in the background: every design choice is judged by how well it supports the triad as a monetary base and how well Work Credits and derived instruments can function as Private Money and AI Money when the world turns adversarial.

Part III
Infrastructure Layers 0–3

Part I argued that soft guarantees are breaking under debt, AI, and platform control, and set out a threat model plus a seven-layer cypherpunk stack. Part II treated **Privacy, Proofs, and Compute** as a **monetary base**, with Work Credits as energy-anchored claims on that base.

Part III switches to the **infrastructure angle**. It asks a blunt question:

> *What has to be physically true, and stay true under repression, for the triad to be real?*

The answer starts at **Layer 0 – Verifiable Machines & Energy**, then climbs through **Layer 1 – Reachability**, **Layer 2 – Distribution & Execution**, and **Layer 3 – Identity & Claims**. These layers do not directly "store value," but they determine whether any higher-layer SoV story survives contact with hardware, networks, and identity regimes.

Chapter 15

Layer 0: Verifiable Machines & Energy

Layer 0 is where cryptography stops being metaphor and touches matter.

PoW had an implicit Layer 0: silicon, power, and warehouses hashing in the dark. The trust assumption was: *"ASICs will do what the SHA-256 spec says."* In practice that meant: *"we trust the vendor, the fab, the firmware, and the power company, and we hope no one has a better ASIC they haven't told us about."*

That was barely acceptable for **useless work**. It is not acceptable when the puzzle mints **receipts**, not heat; when proofs and verified compute become monetary primitives.

15.1 Why Layer 0 Is a Monetary Question

Gold worked as money because geology is hard to fake at scale. Bitcoin worked because hashing cost was hard to fake at scale. In both cases, the monetary story rested on silent assumptions: rocks behave; fabs behave; physics behaves.

The triad inherits those assumptions and tightens them:

- Privacy is only as real as the devices that hold keys and speak on the wire.
- Proofs are only as real as the machines that generate entropy, execute circuits, and sign receipts.
- Compute is only as real as the GPUs/ASICs that claim to have run workloads.

If those machines are opaque, remotely steerable, or quietly biased, **Work Credits degrade into theater**:

- A compromised RNG can turn "unpredictable leader election" into a slow rug-pull.
- A backdoored prover can emit "valid" proofs that leak secrets to its designer.
- A mandated TEE can become a kill switch for entire clusters of provers and routers.

All of this math still runs on matter.

Every proof, every encrypted wallet, every verified FLOP ultimately lives on a sliver of doped silicon that almost nobody is allowed to audit. Today's "trusted hardware" stack is a daisy chain of NDAs: closed-source EDA tools, proprietary IP blocks, opaque PDKs, black-box fabs, sealed packaging, vendor-run attestation services.

> **If the machine can lie, the proof becomes theater.**

From a monetary standpoint, Layer 0 asks:

> *"Can we treat triad capacity as collateral if we don't know what the machines are really doing?"*

The answer is "no." A triad backed by machines we cannot interrogate is not a reserve asset; it is an IOU on a hardware cartel plus the jurisdictions that regulate it.

So the mandate of Layer 0 is:

> **Translate "trust the vendor" into "trust these verifiable claims about the machine and its power," or don't pretend it's money.**

15.2 Design Goals and Non-Goals

Layer 0 has to be ambitious enough to matter and humble enough not to LARP full supply-chain omniscience.

Goals

1. **Verifiability over purity.** We aim for *checkable claims* about machines, not for metaphysical purity. Open RTL where we can; structured sampling where we cannot.
2. **Common knowledge of security.** Different actors should be able to agree on *facts* about hardware profiles, even if they disagree about policy.
3. **Energy anchoring, not energy worship.** Power use should be measurable enough that "Work Credit per joule" is meaningful.
4. **Degradability under attack.** When assumptions fail, the system should *degrade visibly*: telemetry spikes, profiles are deprecated, Work Credits tied to broken profiles are quarantined.
5. **Composable exports.** Layer 0 should emit artifacts that higher layers can consume mechanically.

Non-goals

1. **Perfect trustlessness.** We will not "solve" global hardware and supply chains.
2. **Single-vendor dependence.** Heterogeneity is a feature, not a bug.
3. **Total hardware transparency on day one.** Political and commercial realities exist.
4. **Magical protection against all side-channels.** We assign **budgets** to attack surfaces.

15.3 Hardware as Base Reality for Work Credits

In Part II, Work Credits were defined as **claims on standardized units of triad work** (privacy settlement, proofs, verified compute) anchored to energy and VerifyPrice.

Layer 0 defines the **hardware profile** that each Work Credit type rests on. For a canonical workload W, a **hardware profile** H might specify:

- Chip family and stepping.
- Microarchitectural features (e.g., presence of certain accelerators or TEEs).
- RNG source and test regimen.
- Power metering and thermal envelope.
- Known limitations (e.g., "avoids TEE X due to backdoor Y; uses open core Z instead").

When a Work Credit of type (W, T) is minted, the receipt can say:

> *"This unit of work was performed on hardware profile H under conditions C, with proof P and VerifyPrice statistics V."*

Monetarily, that matters because:

- Profiling makes **hardware risk priced** instead of hidden. Credits from "sketchy profile H'" can trade at a discount.
- It lets different actors pick their risk tolerance: some will only hold Work Credits linked to fully open cores; others will accept mixed profiles in exchange for lower cost or higher performance.

Layer 0's job is not to tell everyone what risk to take; it is to make the **risk legible and instrumentable**.

15.4 The Layer 0 Feasibility Ladder

A common objection to Layer 0 is: "Open silicon and sampled supply chains sound like moonshots. What can we actually do *this decade* given real-world fabs, opaque GPU stacks, and geopolitical constraints?"

The answer is a **graded trust ladder**. Layer 0 does not require perfection on day one; it requires **measurable progress and falsifiable claims** at each grade. Higher grades provide stronger guarantees; lower grades are acceptable for less sensitive workloads, with telemetry that detects when you're relying on weaker grades.

Key principles:

1. **Grades are explicit.** Every hardware profile is tagged with its Layer 0 grade. Work Credits, VerifyPrice dashboards, and SLA tiers reference these grades so users know what trust assumptions they're accepting.

Grade	Name	What It Means	Timeline	Trust Residual
L0-A	Best Available Today	Attestation + multi-party audits + reproducible benchmarking + diversity	Now	Trust vendor + third-party auditors; side-channel budget measured but not minimized
L0-B	Sampled & Bounded	Lot sampling + imaging + side-channel budgets + independent lab inspections	1–3 years	Trust sampling methodology; residual risk is unsampled units
L0-C	Partial Open	Open RTL for critical components; open firmware; proprietary accelerators wrapped with proofs	2–5 years	Trust fab + packaging; open logic is inspectable
L0-D	Fully Open	Fully open designs + open PDKs + multi-fab production	5–10+ years	Trust physics + fab process; no single-vendor chokepoint

Table 15.1: Layer 0 Feasibility Ladder

2. **Telemetry detects reliance on weaker grades.** If 80% of proving capacity is L0-A (vendor-attested) and only 5% is L0-C (partial open), that concentration is visible in dashboards. Users can decide whether to hold Work Credits tied to such a distribution.
3. **Higher grades earn lower risk premiums.** Markets should price Work Credits from L0-D profiles more favorably than L0-A profiles, creating economic gravity toward openness as it becomes available.
4. **Migration paths are first-class.** When L0-B or L0-C options become viable, there are documented procedures to migrate workloads off L0-A profiles without catastrophic downtime.
5. **Failure modes are bounded.** If a specific grade is compromised (e.g., a TEE used in L0-A profiles is broken), the impact is contained to that grade. Higher-grade capacity continues to function; affected Work Credits are quarantined or hair-cut.

15.4.1 Policy Hooks: Economic Treatment by Grade

The following table specifies how each L0 grade affects Work Credit eligibility, pricing, and collateral treatment. These are reference parameters; implementations may adjust within bounds.

Interpretation:

- **Issuance weight:** Higher grades earn more WC per unit of verified work, incentivizing investment in open hardware.
- **Collateral haircut:** When WC are used as collateral (e.g., for staking, LP positions, or DeFi), lower grades may face haircuts. L0-A collateral is discounted if a single vendor/TEE dominates.
- **VerifyPrice tier:** Dashboards stratify measurements by L0 grade. If most verification runs

Grade	WC Eligibility	Issuance Weight	Collateral Haircut	Verif
L0-A	All workloads; capped at 60% of total issuance	1.0× (baseline)	0% (if diversified) to 15% (if concentrated)	Repo
L0-B	All workloads; no issuance cap	1.0×	0%	Stanc
L0-C	All workloads; eligible for "open-profile" premium tier	1.1× (bonus for open hardware investment)	0%; may qualify as "pristine collateral"	Prem
L0-D	All workloads; highest trust tier	1.2×	0%; pristine collateral	Gold

Table 15.2: Economic treatment by L0 grade.

on L0-A hardware, this is flagged as concentration risk.

- **Issuance cap (L0-A only):** To prevent over-reliance on vendor-attested hardware, L0-A profiles are capped at 60% of new issuance. Excess work at L0-A earns priority for future slots but not extra tokens.

15.4.2 Quantitative Thresholds (Reference Design)

Metric	Threshold	Consequence if Breached
L0-A share of issuance	>60%	New L0-A issuance paused until ratio falls
Single vendor share within L0-A	>40%	Affected profiles moved to "watch list"; 5% coll
Sampling coverage for L0-B	<80% of units per quarter	Grade downgraded to L0-A until coverage resto
Open-source audit currency (L0-C/D)	>12 months since last audit	Grade downgraded one level

Table 15.3: Quantitative thresholds for L0 grades.

These thresholds are governance parameters, adjustable via protocol upgrade with supermajority. Changes are announced 90 days in advance and visible in dashboards.

15.4.3 Pragmatic Starting Point (L0-A)

Today's stack can achieve L0-A by combining:

- **Multi-vendor diversity:** No single chip family or TEE dominates more than X% of critical capacity.
- **Third-party attestation audits:** Independent labs verify that attestation claims match device behavior.
- **Reproducible firmware and benchmarks:** All prover/router firmware is built reproducibly; benchmark results are publicly verifiable.
- **Side-channel measurement:** Known side-channel leakage is measured and published as a "leakage budget" per profile.

- **Cryptographic agility:** Profiles document which primitives can be upgraded via firmware vs. require hardware swap.

This is not trustless. It is *trust-bounded and measured*. The residual trust is explicit: "We trust these vendors, these auditors, and this sampling methodology, and here is the evidence."

15.4.4 The Path Forward

- **L0-A → L0-B:** Fund independent lot-sampling programs. Publish inspection results. Build statistical models of detection confidence.
- **L0-B → L0-C:** Invest in open RISC-V cores, open RNG designs, open firmware stacks. Wrap proprietary accelerators with proof interfaces so their internal logic doesn't need to be trusted.
- **L0-C → L0-D:** Support open PDK initiatives. Diversify fab sources across jurisdictions. Build ecosystem gravity so open designs become economically competitive.

15.4.5 Why This Prevents "Layer 0 Is Impossible, Therefore Thesis Fails"

The thesis does not require L0-D today. It requires:

1. Clear grading of what trust assumptions each profile carries.
2. Telemetry that makes those assumptions visible.
3. Economic and technical paths toward stronger grades over time.
4. Failure modes that degrade visibly, not silently.

If these four conditions hold, Layer 0 becomes a *progress metric* rather than a moonshot prerequisite. The stack can operate today at L0-A while building toward L0-C/D, and users can decide what risk they're willing to accept at each stage.

15.5 Concrete Components of Layer 0

Layer 0 is not a single device. It is a **bundle of practices and mechanisms** that give higher layers a surface to stand on.

15.5.1 Open Designs Where Possible

The gold standard is **open cores and toolchains**: open RTL or microarchitectures for CPUs/accelerators, open PDKs where geopolitical conditions allow, and reproducible build infrastructure.

Where open options exist, they are **first-class citizens** in hardware profiles. On the margin, this creates **economic gravity**: as open hardware matures, Work Credits tied to open profiles should command a lower risk premium.

15.5.2 Attested Randomness and Entropy

Randomness is the hidden spine of consensus and proofs. A biased RNG can leak signing keys, predict leader election, and make PoUW "fairness" an illusion.

Layer 0 requires:

- Hardware RNG designs that are **documented and testable**.
- **Attested entropy tests**: periodic statistical test suites.
- **Blended randomness**: mixing hardware entropy with VRFs, commit-reveal, and cross-machine aggregation.

15.5.3 Attestation Without Priesthood

Modern hardware ecosystems push TEEs and attestation as the answer to everything. Used naively, they simply move "trust the vendor" into "trust the vendor's signing key."

Layer 0's posture:

- TEEs and hardware attestation are **useful tools**, not **root of trust**.
- Attestations should be **wrapped and sampled**, not taken as gospel.
- Devices emit **local attestations** wrapped in **SNARKs or STARKs** where possible.

15.5.4 Lot Sampling and Destructive Audits

Because we cannot open every chip, Layer 0 leans on **lot sampling**:

- For each hardware profile, a fraction of units are randomly selected for **deep inspection**: decapping, imaging, side-channel probing.
- Results are published as part of the profile's dossier.
- A protocol-level **Layer 0 Assurance Fund** (2% of fee revenue) finances sampling.

This is how Layer 0 turns "we hope the vendor is honest" into:

> *"We have inspected a statistically relevant sample of this profile, and here are the findings and residual risks."*

Sampling Economics and Sufficiency

Critics will ask: "How much sampling is enough? Who pays? What confidence do we actually get?"

Sample rate model (reference design):

Statistical sufficiency: For a profile with N deployed units, sampling n units provides confidence that:

$$P(\text{detect if } \geq k\% \text{ are compromised}) \geq 1 - (1 - k/100)^n$$

Profile Criticality	Definition	Min Sample Rate	Funding Source
Tier 1 (Critical)	Top 5 profiles by WC issuance; >10% of total capacity	≥50 units/quarter	Protocol assurance budget (2% of fee revenue)
Tier 2 (Standard)	Profiles with 1–10% of capacity	≥20 units/quarter	Protocol assurance budget
Tier 3 (Emerging)	New profiles in probation; <1% of capacity	≥5 units/quarter	Profile sponsor (operator or vendor)

Table 15.4: Sample rate model by profile criticality.

For Tier 1 profiles ($n = 50, k = 5\%$), detection probability is ∼92%. This is not certainty—it is **bounded uncertainty**, documented and priced.

What "Gold tier" requires: A profile achieves Gold tier (L0-B or higher) when:

1. ≥80% of deployed units have been covered by sampling programs over the profile's lifetime.
2. No critical findings (backdoors, RNG bias >threshold) in the last 4 quarters.
3. Side-channel leakage remains within published budget under independent testing.
4. At least 2 independent labs have conducted inspections.

Gold tier unlocks:

- Full WC eligibility (no issuance cap).
- Zero collateral haircut.
- Eligibility for "pristine collateral" designation in DeFi integrations.

Funding mechanism: The protocol allocates 2% of fee revenue to a **Layer 0 Assurance Fund**. This fund:

- Contracts with independent hardware security labs.
- Publishes RFPs for sampling campaigns.
- Maintains a public registry of inspection results.
- Is governed by a multisig of hardware security researchers (not protocol developers).

Transparency: All sampling results are published within 30 days. Raw data (images, test logs) is archived and available for independent verification. If a lab's findings are disputed, a second lab can be commissioned for arbitration.

15.5.5 How This Opens New Economies

Verifiable machines don't just make today's cloud slightly less sketchy. They unlock whole categories of economic arrangement that aren't viable when hardware is a black box.

Verifiable cloud and compute co-ops. If you can spin up a rack of open-design TEEs or accelerators and have them produce attestations and ZK receipts that anyone can verify, then:

- A small data center in Nairobi or Reykjavík can sell the same class of "trusted inference" or "trusted proving" as a hyperscaler in Virginia.
- A co-op of households can pool "AI appliances" in their basements and earn by running verified work for others.
- Regulators and enterprises can enforce compliance through proofs and telemetry, not vendor logos.

Compute ceases to be a winner-takes-all brand game and becomes a commodity with open admission.

Civic infrastructure that doesn't depend on one vendor's conscience. Voting machines, digital ID kiosks, public-health dashboards: today they are RFPs to a short list of contractors. With open hardware and proof-wrapped attestation, you can build ballot boxes and ID hardware whose entire stack is open to public inspection and require that each device emit public proofs of correct behavior. Democratic legitimacy becomes a property of math and sampling, not of which vendor's logo sits on the plastic.

Tamper-evident telemetry. As the physical world fills with sensors and actuators, the ability to sell tamper-evident telemetry becomes critical. An environmental sensor built on an open profile, with attested firmware and ZK-wrapped readings, can sell CO_2 or temperature data as evidence into climate, insurance, and industrial hedging markets—not just as numbers on a dashboard.

Hardware-native monetary instruments. Once machines themselves are verifiable actors, we can imagine "miner-notes" backed by the future output of a specific open-design proving farm, or municipal bonds whose coupon is denominated in SLA-backed capacity: X proofs per second, Y verified FLOPs, Z private swaps per month, all attested by open hardware meters. These become new kinds of collateral: claims on capacity the world must keep buying.

Open profiles manufactured at multiple fabs in multiple jurisdictions turn chip supply into a multi-polar fabric instead of a single chokepoint. Neutral money needs neutral hardware; verifiable machines produced on a diversified manufacturing base make that a reachable design goal rather than a slogan.

15.6 Verifiable Power: Energy as First-Class Input

If Work Credits are energy-anchored claims on triad work, then **power** is not just an environmental footnote; it is part of the monetary base.

Layer 0 treats power in three ways:

1. **Measurement.** Prover farms track power draw, mapping between workloads and power, and local generation vs. grid intake. These feed into **Work Credit metadata** and **VerifyPrice** models.
2. **Resilience.** Micro-grids or backup generation for critical infrastructure; geographic and jurisdictional dispersion of power sources.
3. **Policy hedging.** Transparent power telemetry allows answering "how much of this is actually training/inference/proof that humans pay for?"

For Work Credits, this means credits can carry **energy provenance tags** and markets can price credits differently based on energy profile.

15.7 Operational Patterns: Profiles, Upgrades, and Failure Modes

Layer 0 is not static; hardware evolves, breaks, and gets deprecated. We need patterns for **living with that churn**.

15.7.1 Hardware Profiles and Workload Binding

Each canonical workload W in Layer 4 is associated with one or more **acceptable hardware profiles** $(H_1, H_2, ...)$. Profiles define:

- Minimum performance characteristics (to keep VerifyPrice in target bands).
- Acceptable side-channel leakage budgets.
- Attestation/sampling histories.
- Known caveats (e.g., "avoid profile H3 for workloads with secret inputs; leaks are too strong").

When PAL compiles a workload or the router assigns work, it can target specific profiles, diversify across profiles, or refuse high-sensitivity workloads to marginal profiles. This is how **Layer 0 informs the market** rather than hiding under it.

15.7.2 Upgrades and Deprecations

Hardware ages; bugs and backdoors are discovered; fabs change hands. Layer 0 needs clear **life-cycle rules**:

Onboarding: New profiles go through a probation period with extra sampling and conservative Work Credit weights.

Deprecation: When a profile is compromised or obsolete, higher layers stop accepting new

Work Credits minted from it, quarantine or hair-cut existing credits, and publish an incident report at Layer 6.

Migration: Proof factories and corridor operators need technical paths to migrate workloads off deprecated profiles without massive downtime.

In monetary terms, this is the **hardware analog of a bond downgrade**: transparent, describable, and priced, rather than silently swept under the rug.

Non-Discretionary Downgrade Rules

To prevent "ex post discretionary default" critiques, deprecation and haircut decisions follow a **predictable severity framework**, not ad hoc governance.

Severity	Definition	Examples
S0 (Watch)	Potential issue; under investigation	Anomalous side-channel readings; unverified third-party report
S1 (Warning)	Confirmed issue with limited impact	RNG bias below 1%; isolated firmware bug
S2 (Critical)	Confirmed issue with systemic impact	Backdoor in >5% of sampled units; key extraction demonstrated
S3 (Emergency)	Active exploitation or catastrophic risk	Widespread key compromise; vendor collusion confirmed

Table 15.5: Severity levels for hardware profile issues.

Process guarantees:

1. **Evidence threshold:** S1 requires independent lab confirmation; S2 requires reproducible demonstration; S3 requires active exploitation observed.
2. **Multi-party decision:** Severity escalation requires sign-off from ≥3 of 5 designated hardware security reviewers.
3. **Appeal process:** Profile sponsors can challenge findings within 14 days.
4. **Sunset, not confiscation:** Even at S3, WC holders retain 75% of value after haircut.
5. **Transparency:** All severity determinations are published.

15.7.3 PQ and Cryptographic Agility

Some Layer-0 assumptions are about **cryptography**, not just silicon: signature schemes in ROMs, hash functions in hardware accelerators, and RNG primitives.

Layer 0 insists that hardware and firmware expose enough **configurability** to migrate to post-quantum or new primitives without throwing away entire fabs. Hardware profiles document cryptographic agility (e.g., "can switch hash from X to Y via firmware; signature scheme fixed").

15.8 Stress Tests for Layer 0

Every claim in Layer 0 should map to a testable stress scenario:

Claim	Stress Test
Profile H is honest within bounds X	Feed adversarial inputs; measure deviation from spec; decap sample devices
Sampling methodology detects tampering	Red-team: insert known-bad units into supply; measure detection rate
RNG entropy meets threshold	Run standard test suites (NIST, Dieharder); inject synthetic bias; verify detection
Power telemetry is honest	Cross-check metering with grid records; test resilience to meter spoofing
Migration from H1 to H2 works	Simulate H1 deprecation; measure latency and failure rate during migration

Table 15.6: Layer 0 stress tests.

If a stress test fails, Layer 0 **degrades visibly**: profiles are flagged, Work Credits tied to them are quarantined, and Layer 6 publishes an incident report.

15.9 What Layer 0 Exports to Higher Layers

Higher layers consume a **small set of artifacts and APIs**:

1. **Hardware profiles (HIDs).** Compact identifiers + dossiers describing profile properties, sampling history, and current status.
2. **Attestation receipts.** Machine-level statements binding device → profile, firmware → hash, measurement → nonce/time.
3. **Power and health telemetry.** Streams of power usage, failure rates, and uptime patterns.
4. **Incident and status flags.** Signals like "Profile H3 compromised; do not accept new work."

Chapter 16

Layer 1: Reachability

All of this math still travels as packets.

Every proof, every private swap, every verified FLOP ultimately crosses a handful of cables and radios that a small number of operators can see and shape. A store of value that survives yield-curve control but dies when a few IXPs collude is not a store of value; it is an overlay.

Comms resistance is the stack's oxygen: the property that air still flows when someone leans on the hose.

16.1 Threat Model: What We Must Survive

The comms threat model has four main faces:

Backbone controls. IP and prefix blocking, BGP blackholing, DNS and SNI filtering, DPI-based resets, QoS throttles on known P2P handshakes.

Exposure and linkage. Static addresses and reusable invoices let adversaries map who is getting paid; mempool and gossip surveillance reveal patterns.

Eclipse and routing capture. A node that only talks to a small set of peers in the same ASN can be effectively isolated.

Platform risk at the edge. App-store takedowns, CDN firewalls, corporate endpoint policies that classify P2P clients as malware.

16.2 Design Rules: Protocol Posture Under Pressure

1. **Transports must be encrypted by default and boring on the wire.** Clear-text, distinctive handshakes invite classification. BIP-324-class v2 encrypted transport is the template.
2. **The stack must be transport-agile.** No single path (TCP/TLS, Tor, I2P, QUIC) should be a single point of failure.
3. **Receiver privacy must be the default.** Payment rendezvous should be "addressless"—BOLT12 Offers, path-blinding, and shielded pools.
4. **Settlement must be refund-safe under squeeze.** Atomic, adaptor-signature flows with clear timeout and refund semantics.

5. **Edge admission must remain open.** No "special" relays or whitelisted entry points.
6. **Comms health is part of the telemetry regime.** If reachability silently degrades, neutrality degrades with it.

16.3 Mechanisms: What Needs to Ship

Encrypted, multi-path P2P transports. Full nodes speak encrypted P2P by default, with Tor and I2P as first-class citizens. Camouflage layers (Noise-style patterns, obfs-class shims) allow relays in high-interdiction ASNs to keep moving traffic.

Receiver-private routing for payments. BOLT12 Offers: receiver-private, reusable invoices. Path-blinding and rendezvous routing ensure intermediaries see only their hop. Federated ecash and shielded pools provide endpoint firebreaks.

Settlement survivability. Adaptor-signature atomic swaps as the default for cross-asset payouts. "Abort & refund" as a first-class, well-signposted action.

Topology and anti-eclipse hardening. Peer-set diversity and rotation; connections spread across geographies and ASNs; gossip protocols favor multiplicity of paths.

16.4 How Layer 1 Anchors the Triad

Communications resilience is not an orthogonal concern; it is the medium in which the triad either lives or suffocates.

For Privacy, encrypted transports and receiver-private endpoints are what keep the network itself from nullifying ledger-level secrecy. Shielded pools and privacy coins are only as private as their membrane to the outside world. If every shielded payout can be traced to a stable IP graph and a static address, "lawful privacy" collapses into a thin veneer over a rich flow-of-funds analysis. Layer 1's job is to ensure that settlement paths are as hard to pin down as the flows they carry: private not only in state, but in motion.

For Proofs, communications resilience keeps verification a public act rather than a priestly privilege. A world where only a small set of well-positioned nodes can fetch and check proofs is a world where VerifyPrice has quietly become an internal KPI rather than a public commodity. If any honest machine with modest connectivity can still reach a verifier over at least one path, proof markets and receipt ledgers remain subject to universal scrutiny. When reachability fragments, the "public" in public verification becomes aspirational.

For Compute, Layer 1 ensures that useful-work mining and verified inference do not devolve into "whoever still has a clear line to the router wins." Proof-of-Useful-Work schemes depend on open admission: a wide, geographically and topologically diverse set of provers and miners competing to satisfy claims. If adversaries can choke ingress to a few ASNs or clouds, PoUW degenerates into a club good. Comms health metrics and open-admission design at the edge keep compute supply neutral and keep the token's claim ("backed by globally demanded work") from turning into "backed by whichever datacenter the regulator likes."

All of this math still travels as packets. Layer 0 keeps the machines honest and powered; Layer 1 keeps them in conversation when it is no longer convenient for them to be. The triad's monetary ambitions depend on both. Without verifiable machines, we do not know what happened. Without resilient communications, we do not know it in time, or at all.

16.5 VerifyReach: Communications Telemetry

Layer 1 introduces **VerifyReach** as the communications analogue of VerifyPrice:

- **Reachability metrics**: fraction of vantage points from which key services are reachable.
- **Degradation patterns**: which networks experience blocking or throttling.
- **Transport diversity**: percentage of traffic over each transport class.

Target SLOs:

- Core infrastructure reachable from $\geq 95\%$ of sampled ASNs (uncensored regions).
- $\geq 70\%$ reachability for known-censored regions.
- p95 time-to-first-connection < 10s uncensored; < 30s censored.
- No country-level view sees more than $Y\%$ persistent reachability degradation without triggering incident handling.

VerifyReach feeds into Layer 6 governance: if reachability collapses in a region, incident response kicks in (alternative transports are promoted, routing is adjusted, and the degradation is visible on public dashboards).

16.5.1 VerifyReach Measurement Specification

Like VerifyPrice, VerifyReach requires a rigorous measurement methodology to prevent gaming and ensure credibility.

Sampling Frame:

Vantage Points:

Dimension	Minimum Coverage	Rationale
ASNs	$\geq$ 500 distinct ASNs globally	ISP-level blocking
Regions	$\geq$ 50 countries; $\geq$ 10 per continent	Regional censorship
Censorship regimes	Coverage of known filtering states (CN, IR, RU, etc.)	Worst-case reachability
Network types	Residential, mobile, enterprise, datacenter	Filtering varies by type

Table 16.1: VerifyReach sampling frame requirements.

- **Community nodes:** Volunteer-run measurement agents (similar to OONI probes). Incentivized via small WC rewards.
- **Independent labs:** At least 3 organizations (academic, NGO, commercial) run measurement infrastructure. No single operator controls > 30% of vantage points.
- **Diversity requirement:** Vantage points must span $\geq$ 20 countries and $\geq$ 100 ASNs for measurements to be considered valid.

Metric	Definition	Target SLO
succ_1(N,R)	Fraction reaching service N via primary transport within 30s	$\geq$ 95% / $\geq$ 70%
succ_2(N,R)	Fraction reaching via any transport (incl. fallbacks) within 60s	$\geq$ 99% / $\geq$ 85%
ttfc(N,R)	Time-to-first-connection (p50, p95)	p95 $<$ 10s / $<$ 30s
failure_class	Taxonomy: DNS, TCP RST, TLS, timeout, active probe	Published per region

Table 16.2: VerifyReach metrics and targets. Targets shown as uncensored / known-censored.

Metrics:

Threat	Mitigation
Spoofed vantage points	Periodic challenges (fetch and sign specific data); anomalous behavior triggers exclusion
Selective treatment	"Canary" requests that should succeed; if canaries fail but targets succeed, point is flagged
Measurement capture	Cross-check results from different operators; divergence > 10% triggers investigation
Temporal gaming	Continuous measurements (not snapshots); 24-hour rolling averages published

Table 16.3: VerifyReach adversarial robustness measures.

Adversarial Robustness:

Publication:

- Raw measurement data (anonymized to protect vantage point operators) published daily.
- Aggregated dashboards updated hourly.

- Quarterly reports summarizing regional trends, incidents, and transport effectiveness.
- All measurement code is open-source and reproducible.

Monetary Consequences: If VerifyReach for a region falls below thresholds:

- WC minted by operators in that region may face **regional risk premiums** (higher collateral requirements).
- Corridors primarily serving that region are flagged; users see warnings before transacting.
- Incident response is triggered: alternative transports promoted, routing adjusted.

This makes VerifyReach not just a dashboard metric, but an **input to economic risk pricing**.

Chapter 17

Layer 2: Distribution & Execution

Software distribution is where lofty protocol guarantees meet the boring reality of phones, laptops, and routers that have to run code. In a repression cycle, app stores are throttled, DNS is poisoned, and "safety" policies become chokepoints.

It is far easier to lean on an app store, certificate authority, or CDN than to break a new primitive. If privacy, proofs, and compute are monetary primitives, then the binaries that implement them must keep moving even when networks are hostile.

17.1 Threat Model & Objectives

An adversary's repertoire at this layer:

- App-store removals and policy bans.
- DNS poisoning, SNI and IP blocking, TLS interception.
- BGP hijacks or route blackholing for update servers.
- Certificate revocations; captive portals that substitute their own "secure updates."
- Targeted developer coercion and "emergency directives."

Objectives:

- **Reachability**: users can discover and fetch releases despite interference.
- **Authenticity**: users can verify what they fetched without trusting a platform.
- **Safety**: upgrades are atomic, reversible, and survive power loss.
- **Neutrality**: no single jurisdiction or vendor can gate updates.
- **Telemetry without doxxing**: reliability is measurable without surveillance.

17.2 Design Rules

Authenticity before reachability. Releases ship with threshold signatures (3-of-5) from independent maintainers, plus inclusion in an append-only transparency log.

Reproducible builds as default. At least one independent builder reproduces binaries from source; the build process emits deterministic artifact hashes and proofs.

Threshold keys and real revocation. Root keys live offline and are sharded; release keys rotate regularly; revocation is a signed artifact in the transparency log.
Binary transparency anchored to a proof ledger. The update log behaves like certificate transparency for releases, periodically anchored to a neutral proof ledger.
Atomic updates with safe rollback. Clients update into an inactive slot, flip a pointer only after verification, and revert automatically on failure.
Graceful degradation. Offline packages (USB, SD, QR) use the same receipts and keys.

Supply-Chain Invariants (Must Hold)

I1. Authenticity: Every release requires threshold signatures ($\geq$3-of-5) from independent maintainers in $\geq$2 jurisdictions, plus transparency log inclusion.
I2. Freshness: Clients enforce monotonic version counters. Rollback attacks are detected and rejected.
I3. Reproducibility: At least 2 independent builders reproduce each release. Divergence is release-blocking.
I4. Compromise response: If a signing key is compromised: revocation within 24 hours, affected releases quarantined, post-mortem within 14 days.
I5. Multi-path availability: Releases available via $\geq$3 independent channels. No single channel's failure blocks updates for >24 hours.

17.3 Reference Distribution Architecture

The architecture is deliberately multi-homed. We assume any single channel can become a chokepoint.

Control plane (discovery and metadata):
- Standard HTTPS endpoints behind diverse DNS providers.
- Tor onion services exposing the same manifests.
- Signed release announcements gossiped over generic protocols (Matrix, Nostr).
- Append-only transparency log with roots checkable by light clients.

Data plane (artifact delivery):
- Traditional CDNs and anycast mirrors in multiple jurisdictions.
- P2P swarms (IPFS-like or BitTorrent-class) using content addressing.
- Community micro-mirrors with signed manifests.
- Offline channels: USB/SD bundles, QR-encoded update chunks, radio relays.

All of these are interchangeable from the client's perspective. Whoever delivers the bytes fastest wins, but no one is trusted beyond the hash: the client verifies the final digest and only

then installs.

17.4 Key Management & Release Process

Key management is where many otherwise sophisticated systems quietly re-centralize. A single HSM in a single jurisdiction becomes a soft power lever: compromise the humans or the box, and you own the distribution pipeline.

A capture-resistant release process needs:

- **Public release ceremony.** Changes land on protected branches; CI produces deterministic binaries; independent builders confirm bit-for-bit equality. A quorum of maintainers signs the targets manifest.
- **Emergency freeze and unfreeze.** Any maintainer can propose a freeze, but activation requires a threshold of signatures. Unfreeze requires an explicit, signed artifact explaining the incident.
- **Compromise playbook.** Detect, revoke, rotate, and re-sign last-known-good releases. All steps produce receipts that anyone can audit.
- **Jurisdictional diversity.** Key shards and signers are distributed across legal zones, making it difficult for one government to unilaterally subvert the process.

17.5 Update Economics & Neutrality

Distribution has an economics layer just as proofs and compute do. Instead of pretending bandwidth and storage are free, we let mirrors be market actors: they meter what they serve, emit receipts, and get paid for honest work.

- **Receipts for bytes**: mirrors export metered receipts signed by their nodes; a broker aggregates and pays out via privacy rails with SLA escrow.
- **Open admission**: any party can become a mirror by publishing bandwidth and uptime commitments and staking a small bond.
- **Anti-capture telemetry**: publish house-share, top-N mirror share, geographic/ASN diversity. If a CDN dominates beyond thresholds, route weight decays automatically.

17.6 Fallback Playbooks

Having explicit, rehearsed playbooks keeps censorship events from becoming existential crises:
Soft network blocks (DNS/SNI/IP): Bias clients toward alternative control-plane channels (DoH/DoT, onions), and toward content-addressed P2P for artifacts.
Hard regional blocks: Rely on offline kits (USB/SD bundles, QR sets) seeded through civic

institutions and NGOs.

App-store bans: Publish signed sideload bundles with clear, localized installation instructions.

Key compromise: Hit the freeze button; revoke; rotate; and re-sign last-known-good releases, with all steps recorded in the transparency log.

17.7 Lawful Privacy in Distribution

The same architectural moves that make distribution censorship-resistant also make it more legible to good-faith auditors. Transparent logs, reproducible builds, and signed receipts give regulators something objective to look at:

- Which releases were shipped, and when.
- Which vulnerabilities were patched, and how quickly.
- Whether there were jurisdiction-specific forks or backdoors.

Crucially, auditors can examine this evidence without sitting in the middle of every update flow. The system does not ask to be exempt from scrutiny; it insists that scrutiny be applied at the level of receipts and logs rather than taps and implants.

17.8 Update Telemetry

To make distribution part of the triad's credibility, networks should publish continuously:

- **Update health**: p50/p95 metadata and artifact fetch times; per-path success rates; rollback and signature-failure rates.
- **Transparency evidence**: current log roots; inclusion proofs for each release.
- **Decentralization metrics**: top-N mirror share; geographic and ASN diversity.
- **Key material and incidents**: root fingerprints; release-key roster; revocation lists; incident reports.

Chapter 18

Layer 3: Identity & Claims

Identity in a repression-prone, AI-saturated internet cannot be "a file with your name on it." It must be capabilities you can prove on demand (age, jurisdiction, uniqueness, solvency, model ownership) without handing over the dossier.

Reputation should be the trail of receipts from prior correct behavior, not a database of personally identifying facts.

18.1 Why Identity Must Decouple from Doxxing

The web's trust default has flipped: seeing is no longer believing, and platforms strip provenance labels inconsistently. Accounts and government IDs are brittle anchors; they leak power to gatekeepers.

What scales instead is **mechanism over memory**: claims backed by cryptographic evidence any honest party can check cheaply and without permission. This is the same shift the thesis makes for money—verification, not authority, is the arbiter.

18.2 Design Goals

Identity and reputation in this stack satisfy six goals:

1. **Privacy by default, proof by construction.** No standing, globally queryable identity graphs. All predicates are proven as needed, most often with zero-knowledge proofs.
2. **Selective disclosure and unlinkability.** Verifiers learn only the outcome of a predicate, not your name or the rest of your credential bundle.
3. **Cheap, public verification.** Any honest laptop can check evidence quickly, measured the same way we measure proofs with VerifyPrice.
4. **Open admission for issuers and verifiers.** Anyone can issue attestations bound to their own reputation and slashing.
5. **Hardware honesty for machine identity.** Machine claims are anchored in verifiable machines with open designs and sampled supply chains.
6. **Lawful privacy.** Regulators get provable, time-bounded visibility via receipts and viewing

keys, not through custodial chokepoints.

18.2.1 Issuer Pluralism Without Issuer Anarchy

"Open admission for issuers" (goal 4) sounds like chaos: anyone can issue AML credentials, so criminals just issue their own. The resolution is that **issuer sets are policy objects**, not global permissions.

Policy Context	Example Predicate	Issuer Set Definition
AML-screened	"User is not on OFAC/EU sanctions lists"	Threshold of ≥2 from {Exchange A, Exchange B, Compliance Provider C, Regulator D}
Age-verified	"User is 18+"	Any licensed identity provider in user's jurisdiction, or threshold of ≥2 from global providers
Accredited investor	"User meets SEC accredited investor definition"	Licensed broker-dealer or accredited investor verification platform
Device integrity	"Device is L0-B or higher"	Hardware attestation from ≥2 independent TEE vendors or open-source attestation

Table 18.1: Issuer sets by policy context.

Issuer sets by policy context:

How issuers enter/exit sets:

1. **Stake:** Issuers post collateral (WC-Base) proportional to the value of credentials they issue. If credentials are later found fraudulent, stake is slashed.
2. **Audit receipts:** Issuers must publish periodic audit receipts proving their verification processes meet policy requirements. Audits are conducted by independent third parties.
3. **Slashing:** If an issuer's credentials are shown to be systematically false (e.g., issuing "not on sanctions list" to sanctioned entities), their stake is slashed and they are removed from relevant issuer sets.
4. **Jurisdictional diversity:** For high-stakes predicates (AML, accredited investor), issuer sets require representation from ≥2 jurisdictions to prevent single-government capture.

How verifiers select policy bundles:

- Verifiers declare which policy contexts they require (e.g., "AML_screened AND Age_verified").
- The protocol resolves these to issuer sets.
- Presentations must include credentials from issuers in the relevant sets.
- Verifiers can add custom issuer requirements (e.g., "must include credential from Regulator D for EU users").

Why this isn't centralization:

- No single issuer is required for any predicate.
- New issuers can join sets by meeting stake and audit requirements.
- Issuer sets are transparent and auditable.
- Users can choose which issuers to use within the set.

This model preserves open admission while ensuring that predicates in regulated contexts are backed by accountable issuers.

18.3 The Identity Kernel

We treat identity as a claim machine rather than a name registry.

Identifiers are cheap, ephemeral, and scoped—pairwise keys or DIDs, not global user IDs.

Credentials are issued by many parties (exchanges, employers, DAOs, regulators, devices). Each is a capability statement, not a dossier.

Presentations are one-off proofs of predicates over credentials: membership, attributes in a range, conjunctions of statements. Typically zero-knowledge.

Receipts are the portable artifacts. Every accepted presentation produces a PIDL receipt binding the claim, proof hash, SLA tier, and timestamps.

This kernel compiles directly into the developer surfaces: PAL SDK to express "prove this predicate," PIDL receipts to carry results, and privacy corridors to pay without doxxing.

18.4 Human Identity: Personhood and Compliance Without Dossiers

What to prove, privately:

- **Uniqueness / Sybil resistance.** Rate-limited nullifiers, stake-and-slash, or attested-liveness ceremonies.
- **Eligibility.** Age, residency, licensing, or "member-of a KYC-screened set," proved with set-membership or range proofs.
- **Standing & solvency.** Proofs of reserves/income ranges, revealed under viewing keys during credit underwriting.

How it composes with the stack: A wallet invokes PAL to compile predicates to proofs, pays over privacy rails, and gets a PIDL receipt the counterparty can verify on-chain or in a browser in < 5s p95.

18.4.1 Sybil Resistance Menu

Sybil resistance (preventing one entity from creating unlimited fake identities) is a core challenge. Different mechanisms offer different tradeoffs; the stack supports a menu of approaches rather than mandating one.

Mechanism	How It Works	VerifyPrice	Unlinkability	Best For
Rate-limited nullifiers	Each identity can perform N actions per epoch; nullifiers prevent reuse without linking	Low	High	Spam prevention, voting, airdrops
Stake-based	Identity actions require WC collateral; bad behavior → slashing	Low	Medium	High-stakes actions, market making
Social/liveness ceremonies	Periodic video calls, in-person events, or social graph vouching	Medium	Low-Medium	Proof of personhood, high-value credentials
Device-based attestation	Hardware attestation limits identities per device	Low	Low	Machine identity; bounded human use
Economic proof-of-work	Solving computational puzzles to create identity	Medium-High	High	Bot prevention, rate limiting

Table 18.2: Sybil resistance mechanism menu.

Tradeoff guidance:

- **For maximum privacy:** Rate-limited nullifiers + economic PoW. No linkage, but limited action rate.
- **For regulated contexts:** Stake-based + issuer-set credentials. Some linkage to stake address, but accountability.
- **For machine identity:** Device attestation bounded by L0 grade. Accept fingerprinting risk for hardware honesty.
- **For proof of personhood:** Social ceremonies with ZK proofs. Higher coordination cost, but strong Sybil resistance.

Composability: Mechanisms can be combined. A high-stakes action might require:

1. Rate-limited nullifier (prevents spam),
2. Stake (ensures accountability),
3. Issuer credential (meets regulatory requirement).

Each layer adds cost and reduces privacy; the protocol allows verifiers to specify requirements and users to choose compliant paths.

18.5 Machine Identity: Capability, Not Brand

Machines (provers, miners, sensors, model endpoints) need identity, but it should look nothing like a cloud vendor account. Their "identity" is a statement of capability on a particular hardware profile coupled with receipts for past behavior.

When a device participates, it attests to what it is and what it did. Attestations link to open hardware profiles. Those attestations are then wrapped in succinct proofs so any chain can verify them cheaply.

This is **reputation for robots**: SLA-backed capacity that accrues a track record via receipts, not by vendor logo.

18.6 Reputation: Priced Behavior, Not Personal Data

- **Positive reputation** accumulates as fulfilled obligations with proofs: deliveries within SLA, honest inferences, timely settlements, clean audits.
- **Negative reputation** accrues as rejects and slashes: failed audits, invalid proofs, settlement timeouts.
- **Scope and decay** are explicit: reputation is contextual and decays unless renewed.

Because receipts are portable and verifiable, reputation becomes a market primitive. A prover can advertise: "over the last 30 days I delivered PROOF_2^{20} within p95 $<$ 1s at cost $<$ \$0.001 with 0.04% failure"—and anyone can check.

18.7 Patterns That Keep Identity Private and Usable

zK-KYC / zK-AML. Users prove membership in a regulated allow-list without revealing which issuer or which entry.

Rate-limited actions. Per-relationship nullifiers cap spam without linking handles.

Credit without doxxing. A lender demands a proof set plus a bond; the loan contract auto-slashes on missed proofs. No one learns the borrower's legal name unless escrow triggers.

Creator & data provenance. Cameras on open hardware sign origin; receipts embed these proofs in a way that survives platform stripping.

Work credentials for models. Model stewards prove they own specific weights and that inferences came from those weights under defined parameters.

18.8 Failure Modes & Guardrails

Any identity layer that matters will eventually be attacked. What distinguishes a durable architecture is whether failures are observable, bounded, and repairable without appealing to a central authority.

Centralized attesters. If one vendor or agency controls machine attestation or anon-cred issuance, identity becomes permissioned by decree. **Countermeasure:** Pluralism—open hardware profiles, lot sampling, and ZK-wrapped quotes that any verifier can check; multiple issuers whose credentials are interchangeable or combined via threshold logic.

Proof cost creep. If p95 verify time drifts up, only gateways can afford to check receipts. **Countermeasure:** Treat identity workloads as first-class citizens of VerifyPrice; treat regressions as Sev-1 incidents.

Bridge and settlement doxxing. If corridors leak metadata, identity collapses into routing tables. **Countermeasure:** Cryptographic atomicity (adaptor signatures), refund safety, and public settlement telemetry.

Reputation blacklists. Global "bad lists" are honey traps for political pressure. **Countermeasure:** Scoped, revocable policies tied to evidence; actors retain the ability to present counter-receipts.

18.9 How Identity Fits the Modular Stack

Identity and reputation are just another set of workloads in the same loop:

1. **Create / Compute.** A human or machine emits a claim: "I am unique and 18+," "this device captured this video," "this miner computed MATMUL_4096 under profile P."
2. **Prove.** The PAL SDK compiles the predicate into a circuit or useful-work audit, routes it to a prover market, and returns a PIDL receipt.
3. **Settle.** The privacy rails kit executes refund-safe payouts across BTC↔ZEC/XMR corridors or shielded pools.
4. **Verify.** Anyone runs `verify(receipt)` locally, on-chain, or inside another proof.
5. **Telemetry.** VerifyPrice, anonymity-set size, swap success rates, and hardware-profile distributions are public.

The stack stops being "crypto plus KYC" and becomes one fabric where work, truth, and permission all pass through the same narrow gate: proofs anyone can check.

18.10 Identity/Reputation SLOs

For operators and allocators, identity becomes legible through metrics:

- **Predicate VerifyPrice.** Median and tail verify time and cost per standard identity predicate.

Target: p95 under a few seconds, sub-cent cost.

- **Unlinkability.** Rate at which presentations can be linked by passive adversaries.
- **Settlement privacy.** Corridor success rates and anonymity-set sizes for identity-linked flows.
- **Hardware honesty.** Share of machine-signed receipts tied to open profiles.
- **Reputation health.** Fraction of decisions based on receipts vs. account flags.

Identity ≠ name; it is a set of provable capabilities. **Reputation ≠ biography**; it is a ledger of receipts. Bind both to privacy by default, publish VerifyPrice so anyone can check them cheaply, and anchor machine attestations in verifiable hardware.

18.11 Implementation Sketch

All of the above shows up in three simple patterns that developers wire into their apps:

Human → compliance without names. A wallet encodes a claim like: prove(age ≥ 18) $\wedge$ prove(residency $\in$ EU) $\wedge$ prove(member_of AML_screened_set). It sends this to PAL, which compiles the predicates, sends them to a prover market, and returns a PIDL receipt. The merchant verifies in <5s p95 without ever seeing the passport.

Machine → capability without brand. A sensor on hardware profile `open_camera_v1` captures video. The device emits an attested capture, PAL wraps it in a succinct proof, producing a provenance proof and receipt. The camera's "identity" is simply "a device of profile P that has produced N correct receipts."

Reputation → receipts, not profiles. A prover's handle accumulates receipts. When it advertises itself, it publishes: "Last 30 days: delivered PROOF_2^{20} under SLA Gold with p95 verify-time 0.8s, 99.9% success, zero slashes." Anyone can pull the underlying receipts and run the reference verifier.

18.12 Part III Summary: Minimum Viable Stack

Before proceeding to the economic layers (4–5), we consolidate the **minimum viable requirements**:

Telemetry Reproducibility Principle

Telemetry must be reproducible from raw public artifacts by multiple independent parties, or it is not part of the trust base.
Specifically:

Layer	Minimum Viable Requirement	Falsification Test
L0	At least one L0-B grade profile; sampling for L0-A	If all proving runs on L0-A with <20% coverage → thesis weakened
L1	Encrypted P2P + ≥2 fallback transports; VerifyReach with ≥3 operators	If >20% ASNs unreachable for >7 days → thesis fails
L2	Threshold-signed releases + transparency log + ≥2 builders	If single key compromise can push malicious updates → thesis fails
L3	≥3 identity predicates with p95 VerifyPrice <5s; issuer pluralism	If identity requires centralized issuer → thesis weakened

Table 18.3: Minimum viable requirements for Layers 0–3.

1. **Raw data availability:** All inputs to telemetry metrics are published or archived.
2. **Open methodology:** Aggregation formulas and statistical procedures are documented.
3. **Multi-party verification:** At least 3 independent parties publish telemetry; divergence >10% triggers investigation.

Identity ≠ name; it is a set of provable capabilities. Reputation ≠ biography; it is a ledger of receipts. Bind both to privacy by default, publish VerifyPrice so anyone can check them cheaply, and anchor machine attestations in verifiable hardware.

Part IV
Truth, Work, and Settlement: Layers 4–5

Part III started at the basement: Layer 0 as verifiable machines and energy. Layers 1–3 keep packets moving, code running, and actors named without being doxxed.

Part IV is where the stack becomes **explicitly economic**. Layers 4 and 5 are the point where:

- raw compute becomes **truth** (via proofs and verification asymmetry), and
- truth becomes **money** (via private, non-custodial settlement).

This is the layer pair that turns the abstract triad (Privacy, Proofs, Compute) into something you can actually buy and sell: **proof commodities, verified FLOPs, and privacy-preserving flows**.

Chapter 19

From Infrastructure to Economics

Layers 0–3 answer four questions:

1. *Can we trust the machine?* (Layer 0)
2. *Can we reach it under censorship?* (Layer 1)
3. *Can we get code onto it and keep it updated?* (Layer 2)
4. *Can actors prove "who" they are without doxxing themselves?* (Layer 3)

Layers 4 and 5 take those answers as given and ask two more:

5. *Can we turn machine work into claims that anyone can verify cheaply?* (Layer 4)
6. *Can we settle those claims as value flows without chokepoints?* (Layer 5)

If Layers 0–3 are the **nervous system and limbs**, Layers 4–5 are the **cortex and circulatory system**: they decide what counts as action and how it is remembered in economic form.

Chapter 20

Layer 4: Truth & Work

Layer 4 is the **Prove/Verify spine** of the stack.

Its job is to take arbitrary claims—"this media object came from camera C at time T," "this inference was computed by model M on input X," "this corridor swap executed atomically"—and turn them into **portable proofs** that:

- anyone can check with **cheap, public verification**, and
- can be **standardized as commodities** (canonical workloads, SLAs, prices).

20.1 What Layer 4 Is (and Isn't)

Layer 4 is:

- The layer of **circuits, proofs, and verification economics**.
- The place where **verification asymmetry** is engineered and measured.
- The home of **Proofs-as-a-Library (PAL)**, proof factories, and **canonical workload registries**.

Layer 4 is not:

- A specific proof system (SNARK vs. STARK vs. something else); it assumes **multi-ZK**.
- A single PoUW design; it supports several patterns as long as VerifyPrice and anti-capture constraints are met.
- A truth oracle; it can attest to **provenance and computation**, not metaphysical correctness of content.

20.2 Verification Asymmetry Revisited

Recall the definition from Part I. For a workload W:

- $p(W)$ = cost (time, energy, hardware) to **produce** a result + proof.
- $v(W)$ = cost to **verify** that result + proof.
- $r(W) = \frac{v(W)}{p(W)}$ = **verification asymmetry**.

Layer 4's goal is simple:

> **Make $r(W) \ll 1$ for the workloads that matter, and keep it that way in production.**

Why this matters:

- If $v(W)$ is small and stable, **anyone** (including small nodes) can check proofs.
- That makes proofs and verified FLOPs **commodities**: units of work that any counterparty can accept without trusting a platform.

Layer 4 introduces:

$$\text{VerifyPrice}(W) = \{\text{p50 time}, \text{p95 time}, \text{p50 cost}, \text{p95 cost}, \text{failure rate}, \text{hardware profile mix}\}$$

20.3 Canonical Workloads and Proof Types

To avoid an unbounded zoo of bespoke proofs, Layer 4 maintains **canonical workloads**:

- **MatMul**$(n, k, m; \varepsilon)$ – matrix multiplication at specified dimensions and error bounds.
- **Inference**$(M, X$**; policy**$)$ – run model M on input X under constraints.
- **Provenance**$(C$**, chain**$)$ – provenance chain for content type C.
- **Settlement**$(S$**, policy**$)$ – settlement and refund logic for a corridor.

For each canonical workload, the stack defines:

- **Circuits / arithmetization**: how the workload is represented for proving systems.
- **Proof schemas**: which proof systems are supported.
- **SLO tiers**: latency and reliability classes (Bronze, Silver, Gold).

Canonical workloads matter economically:

- They become the **SKUs** for proof and compute markets.
- Work Credits are minted against units of these workloads.
- VerifyPrice and $r(W)$ are tracked per workload and tier.

20.3.1 Canonical Workload Definition Template

For a workload to become a tradable SKU (and eligible for Work Credit issuance), it must be defined with sufficient precision. The following template ensures standardization:

Canonical Workload Definition Template

WorkloadID: Unique identifier (e.g., `MATMUL_4096_FP32`, `INFER_LM_70B_256TOK`)

Statement: What is being proved (natural language + formal predicate)

- *Example:* "The matrix product $C = A \times B$ was computed correctly, where $A, B \in \mathbb{R}^{4096\times4096}$, and $\|C - A \cdot B\|_\infty \leq \varepsilon$."

Public Inputs:

- Commitment to A, B (hash or Merkle root)
- Claimed result commitment (hash of C)
- Error bound ε

- Timestamp range

Private Inputs (Witness):

- Full matrices A, B, C
- Intermediate computation trace (if required by proof system)

What Is Verified:

- Correctness (computation matches claim)
- Bounds (result within specified limits)
- Freshness (timestamp within allowed window)
- Liveness (proof generated within epoch, not precomputed)

Verifier Complexity Class:

- Time: $O(\log n)$ for succinct proofs; $O(n)$ for non-succinct
- Memory: specified peak (e.g., ≤512MB for Laptop-Class)
- Proof size: max allowed (e.g., ≤1MB)

Policy Hooks: (machine-verifiable predicates)

- Hardware profile requirements (e.g., L0-B or higher)
- Prover stake requirements
- Membership/non-membership predicates (allowlist proofs, not graph inspection)

Allowed Hardware Profiles:

- L0-A: Yes (with issuance weight 0.9×)
- L0-B: Yes (1.0×)
- L0-C/D: Yes (1.1×)

Tier	Latency Target	Redundancy	Fee Multiplier
Bronze	p95 ≤ 60s	1× verification	1.0×
Silver	p95 ≤ 10s	2× verification	1.5×
Gold	p95 ≤ 2s	3× verification + audit	2.5×

Table 20.1: SLA tiers for canonical workloads.

SLA Tiers:

Example: MATMUL_4096_FP32 **Why this matters:** Without this template, "canonical workload" risks becoming "whatever the prover says it is." With it, workloads are **precisely specified** (anyone can implement a conforming prover/verifier), **auditable** (claims can be checked against the template), and **tradable** (markets can price and exchange standardized units).

Field	Value
WorkloadID	`MATMUL_4096_FP32`
Statement	Matrix multiply $C = A \times B$, dimensions 4096×4096, FP32, $\|C - A \cdot B\|_\infty \leq 10^{-5}$
Public Inputs	Hash(A), Hash(B), Hash(C), ε, timestamp
Private Inputs	A, B, C, intermediate products
Verified	Correctness, Bounds, Freshness
Verifier Complexity	$O(\log n)$, ≤256MB, ≤500KB proof
L0 Requirement	L0-A minimum; L0-B+ for Gold tier
VerifyPrice Target	$t_{95} \leq 2$s (Laptop-Class)

Table 20.2: Example canonical workload: MATMUL_4096_FP32.

20.4 PoUW Design Patterns

Layer 4 doesn't pick a single consensus recipe. It supports several **proof-of-useful-work patterns**, as long as they satisfy:

- **Open admission**: commodity participants can join.
- **Unpredictable leader election**: no one can cheaply bias the lottery.
- **Useful work binding**: you can't precompute offline or reuse stale work.
- **Proof quality & anti-spam**: junk proofs can't flood the system without penalty.

Pattern 1: Hash-gated useful work

- Miners perform a cheap hash race; crossing a threshold gives short-lived eligibility.
- To produce a valid block, the miner attaches a PoUW artifact seeded from header randomness.

Pattern 2: Proof-first selection

- Provers race to produce useful-work proofs and post them to a mempool.
- A lightweight mechanism selects which proofs get included and rewarded.

Both patterns are compatible with MatMul-PoUW, Inference-PoUW, and hybrid schemes.

PoUW Security Properties (Must Hold)

P1. Precomputation Resistance: Proofs seeded by unpredictable randomness; valid only for $N \leq 10$ blocks after seed.

P2. Grinding Resistance: Attacker with X% hashrate gains at most $(1 + \delta) \cdot X$% rewards, where $\delta \leq 0.1$.

P3. Network Advantage Bound: p95 propagation delay ≤ 2s for proof announcements.

P4. Spam Resistance: Deposit ≥ 10× expected verification cost; 100% slash for invalid proofs.

P5. Cartel Detection: Top-1 share < 20%; Top-5 share < 50%; entry latency ≤ 7 days.

P6. Useful Work Binding: $\geq 10\%$ of proofs independently re-verified by random auditors.

20.5 Proof Factories and PaL

Most developers don't want to think about circuits; they want to say:

> *"Prove that this computation happened, then get me a receipt and pay whoever did the proving."*

Layer 4 provides this via:

- **Proof factories**: infrastructure clusters specialized in generating proofs.
- **Proofs-as-a-Library (PaL)**: an SDK that compiles high-level claims into proofs.

PaL exposes interfaces like:

```
prove_compute(f, inputs, policy)
prove_provenance(asset_id, lineage)
prove_settlement(tx, corridor_policy)
```

Under the hood, PaL:

1. Maps the request to a canonical workload W.
2. Selects suitable proof systems and hardware profiles.
3. Submits the job via neutral routers to proof factories.
4. Returns a PIDL receipt plus a proof artifact.

20.6 VerifyPrice Observatory

Verification asymmetry is a design goal; VerifyPrice turns it into a dashboard.

The **VerifyPrice observatory** continuously measures:

- p50/p95 verify times,
- estimated energy per verification,
- failure rates and mismatch rates,
- diversity metrics (how many independent verifiers are actually checking).

Reference Verifier Classes:

Class	Hardware Spec	Use Case
Laptop-Class	4-core CPU, 16GB RAM, SSD, no GPU	Baseline for "anyone can verify"
Mobile-Class	ARM SoC, 8GB RAM, flash	Edge verification, IoT
Server-Class	32-core CPU, 128GB RAM, optional GPU	High-throughput

Table 20.3: Reference verifier classes for VerifyPrice measurement.

20.6.1 VerifyPrice Measurement Specification

VerifyPrice is now a monetary KPI—it determines whether Work Credits retain their value proposition. That requires a rigorous measurement harness, not just a dashboard slogan.

Cost Vector Definition: For each workload W and verifier class V, VerifyPrice is a 5-tuple:

$$\text{VerifyPrice}(W, V) = (t_{50}, t_{95}, e, m, f)$$

Where:

- t_{50}, t_{95}: median and 95th-percentile verification **time** (seconds)
- e: **energy** per verification (Joules, measured via hardware counters or watt-meter)
- m: peak **memory** (MB)
- f: **failure rate** (timeouts, invalid proof rejections, crashes)

USD cost is derived: $c = e \times p_{\text{energy}} + t_{95} \times p_{\text{opp}}$, where p_{energy} is the reference energy price (USD 0.10/kWh) and p_{opp} is opportunity cost (USD 0.001/s), both published quarterly.

Requirement	Rationale
≥2 independent implementations per workload tier	Prevents single-implementation bugs from corrupting measurements
Open-source, reproducible builds	Anyone can audit and rebuild
Deterministic output	Same proof → same result, always
Versioned and tagged	Measurements tied to specific verifier version

Table 20.4: Verifier implementation requirements.

Verifier Implementations:

Sampling Methodology:

- **Random selection:** Proofs selected uniformly at random from recent submissions (not cherry-picked).
- **Stratified by region/profile:** Measurements cover ≥10 regions and ≥3 hardware profiles per workload.

- **Adversarial corpus:** 10% of test proofs are malformed or worst-case (max witness size, pathological inputs) to measure failure handling.
- **Continuous measurement:** Not periodic snapshots; rolling 24-hour windows published hourly.

Condition	How Tested
Network latency	200ms RTT injected for proof fetch
Packet loss	5% random packet loss during verification
Worst-case proof size	Max allowed witness for workload class
Malformed proofs	10% of test corpus intentionally invalid
Resource exhaustion	Verification under 80% memory pressure

Table 20.5: Adversarial conditions for VerifyPrice testing.

Adversarial Conditions:

Reproducibility Standard:

- **Benchmark harness:** Open-source, versioned, deterministic. Anyone can run the same tests.
- **Signed results:** Each measurement batch is signed by ≥2 independent measurement operators.
- **Divergence alerts:** If independent operators diverge by >5%, investigation triggered.
- **Archived raw data:** All proof samples and timing logs archived for 1 year.

Why this matters: Without this spec, "VerifyPrice observatory" is an oracle claim. With it, VerifyPrice becomes **reproducible consensus**—any skeptic can run the harness, check the measurements, and falsify the dashboard if it's wrong.

Workload	Metric	Target (Laptop)	Sev-1 Threshold
PROOF_2^{20}	t_{95}	$\leq$ 5s	$>$ 10s for 7 days
MATMUL_4096	t_{95}	$\leq$ 2s	$>$ 5s for 7 days
INFER_LM_70B	t_{95}	$\leq$ 30s	$>$ 60s for 7 days
All	failure rate	$\leq$ 0.1%	$>$ 1% for 7 days

Table 20.6: Constitutional VerifyPrice targets.

Physical VerifyPrice SLOs (Constitutional):

20.7 VerifyPrice in Practice

At the North-Star level, VerifyPrice answers a simple allocator question:

> *"If I hold this asset for a cycle, does one unit still buy at least as much verification as it used to?"*

We care about VerifyPrice in three dimensions, designed to avoid circular reasoning:

20.7.1 Physical VerifyPrice SLOs (Constitutional)

These are **non-negotiable** targets, measured in real resources on reference hardware. They are **exogenous to token price**. Whether the token rallies or dumps, these targets must hold. If verification takes 30s on a laptop, the "anyone can verify" promise is broken.

20.7.2 Protocol Affordability SLOs (Operational)

These measure whether verification remains affordable as a fraction of typical transaction costs:

$$\text{AffordabilityRatio}(W) = \frac{\text{VerifyCost}(W)}{\text{MedianFee}(W)}$$

- **Target:** AffordabilityRatio $\leq 5\%$ for all workload classes.
- **Settlement:** Verification cost $\leq 1\%$ of median transaction value.

20.7.3 Token-Quoted VerifyPrice (Market Signal, Not Target)

Token-quoted VerifyPrice is a **useful market signal**, but not a constitutional target:

$$\text{VerifyPower}(\text{token}) = \frac{1}{\text{VerifyPrice}(W) \times \text{FeeSchedule}(W)}$$

Token price is endogenous. Making this a *target* creates circular reasoning. The protocol commits to **physical SLOs and affordability ratios**. Token-quoted metrics are dashboards for market participants, not governance constraints.

Level	What It Measures	Who Enforces	Consequence of Breach
Physical (Constitutional)	Real-world verification cost	Protocol governance	Sev-1; remediation required
Affordability (Operational)	Verification as % of fees	Fee policy	Fee schedule review
Token-quoted (Market)	Purchasing power signal	Market participants	Informational only

SLO Hierarchy:

20.8 Layer-4 Stress Tests

Layer 4 passes its SoV audition if it survives several adversarial scenarios:

Circuit bloat. Can we detect when proofs become too expensive to verify? Is there a migration path to leaner circuits?
Prover cartel. Can neutral routers and SLA/slashing mechanisms prevent a small set of proof factories from monopolizing high-value workloads?
Proof system break / new attack. Can we deprecate a proof system, rotate to alternatives, and quarantine affected Work Credits?

If Layer 4 remains **cheap to verify, open to participate, and instrumented enough to handle change**, then Proofs and Compute deservedly move closer to "monetary primitive" rather than "platform feature."

20.9 Minimum Viable Layer-4 Economy

WorkloadID	Description	VerifyPrice Target	WC Tier
`PROOF_2^20`	Generic ZK proof	$t_{95} \leq 5$s	A (full)
`MATMUL_4096_FP32`	Matrix multiply 4096×4096	$t_{95} \leq 2$s	A (full)
`INFER_LM_7B_512TOK`	7B param inference	$t_{95} \leq 10$s	B (discounted)
`PROVENANCE_MEDIA`	Media provenance chain	$t_{95} \leq 3$s	A (full)
`SETTLE_ATOMIC`	Atomic swap settlement	$t_{95} \leq 5$s	A (full)

Table 20.7: Canonical workload starter set.

Canonical Workload Starter Set:

Tier	Verification Type	WC Weight	Collateral Grade
Tier A	Full cryptographic proof (ZK-SNARK/STARK)	1.0×	Pristine
Tier B	Probabilistic verification (audited transcripts)	0.6×	Standard
Tier C	Attestation-backed (TEE + sampling)	0.3×	Discounted

Table 20.8: Work Credit issuance tiers.

Tier Rules for Work Credit Issuance:

Fee + Burn + Slashing Logic:
- 70% of fee to prover (reward)
- 20% burned (supply reduction)
- 10% to protocol treasury (security budget)

Fee formula (reference):

$$F(W) = \text{BaseFee}(W) \times (1 + \text{CongestionMultiplier}) \times \text{SLATierMultiplier}$$

Chapter 21

Layer 5: Value & Settlement

Layer 4 turns work into truth. Layer 5 turns truth into **money movement**.

It is the layer where balances change, salaries get paid, cross-asset swaps execute, and collateral gets posted or released.

The constraints are harsh:

Repression-resilience. Default assumption: yield-curve control, capital controls, KYC chokepoints.

Privacy & lawfulness. Users need *privacy by default* and *auditability by consent*.

Non-custodial safety. Settlement protocols must be self-custodial and refund-safe.

21.1 Settlement as a First-Class Workload

The stack treats settlement itself as a **canonical workload**:

- Workload type `Settlement(S, policy)` includes chain(s) involved, transfer size, corridor routing, and policy hooks.
- Proofs at Layer 4 can attest that settlement followed policy and the final state matches expectations.

This means:

- Layer 5 isn't "just payments"; it is a **proven computation** over state transitions.
- Settlement events generate PIDL receipts that can be audited, archived, or collateralized.

21.2 Design Constraints for Privacy Rails

Privacy rails must satisfy a tight triangle:

1. **Non-custodial.** Users never relinquish both principal and unilateral control to intermediaries. Adaptor signatures, HTLC-like constructs, or scriptless scripts enforce this.
2. **Refund-safe.** If a swap fails, both sides can unilaterally reclaim funds after bounded time. Refund logic is testable and provable.
3. **Privacy-by-default with optional disclosure.** Default flows have strong anonymity sets; viewing keys allow specific flows to be revealed.

This is where the **Privacy Rails Kit (PRK)** lives.

Policy = Predicates, Not Surveillance

Policy compliance in Layer 5 is achieved through **predicate proofs**, not graph inspection.

What can be proved (ZK-compatible):

- Not on sanctions list → Membership proof: "sender ∈ AML-cleared set S"
- Within limits → Range proof: "amount ≤ \$10,000"
- Authorized counterparty → Set membership proof
- Tax reporting → Viewing key + receipt disclosure

What cannot be proved (and we don't try):

- "Did not interact with blacklisted address X" — requires global graph knowledge
- "Full transaction history disclosure" — defeats privacy model
- "Real-time surveillance feed" — centralizes and doxxes

21.3 The Privacy Rails Kit (PRK)

PRK is the Layer-5 sibling of PAL. It gives builders a way to express **pay-for-proof** and **pay-for-compute** intents and execute them over privacy-preserving, non-custodial corridors.

At a high level, PRK supports:

- **BTC↔ZEC/XMR corridors.** Using adaptor signatures or HTLC-like patterns, ensuring atomicity.
- **Shielded pools and internal flows.** For in-asset privacy with proofs for policy compliance.
- **Conditional payment flows.** "Pay address A if proof P of workload W is posted within time T; else refund."

PRK exposes intents like:

```
pay_for_proof(claim_id, max_price, corridor_policy)
execute_corridor(btc_input, zec_output, privacy_policy)
batch_payroll(payroll_blob, corridor_set, audit_policy)
```

21.4 Settlement Safety: Refund and Bridge Invariants

Non-custodial rhetoric is cheap; invariants are not.

Refund Safety Invariant: For any corridor C and transaction T:

$$\text{refundSafe}(T) = 1 \iff (T \text{ succeeds} \vee T \text{ refunds within timeout})$$

Where:

- Timeout is bounded (e.g., $\leq$ 24 hours for BTC↔ZEC).
- Refund means unilateral recovery without counterparty cooperation.
- Success means both legs complete atomically.

Bridge Safety: Cross-chain bridges are historically the single biggest loss vector. Layer 5 imposes:

- **No "magic multisigs."** Bridges must not rely solely on small, permissioned signer sets.
- **Light-client or proof-based validation where possible.**
- **Explicit trust classifications** reflected in profiles and telemetry.

Bridge Class	Trust Model	Collateral Eligibility	Example
Class A: Trustless	Cryptographic only	Full	Atomic swap
Class B: Threshold	m-of-n MPC	10% haircut	5-of-9 threshold
Class C: Federated	Known, bonded federation	25% haircut	Liquid-style
Class D: Custodial	Single custodian	**Not eligible**	CEX-wrapped

Table 21.1: Bridge trust classification.

21.5 VerifySettle: Settlement Telemetry

VerifySettle tracks:

- **Success rates** across corridors (p50/p95, by size bucket).
- **Refund outcomes** — how often refunds are exercised and their timing.
- **Anonymity-set health** — shielded pool sizes, churn, volume.
- **Concentration metrics** — top-N LPs, corridor operators, jurisdictional spread.

Metric	Definition	Target	Sev-1
swap_success(C)	% of swaps that complete	$\geq$ 95%	$<$ 90% for 7 days
refund_safe(C)	% of failures that refund	100%	Any violation
ttf(C)	Time-to-finality (p95)	$\leq$ 30 min	$>$ 2 hours
anon_set(C)	Anonymity set size	$\geq$ 1000	$<$ 500 for 30 days
lp_concentration(C)	Top-3 LP share	$<$ 50%	$>$ 70% for 14 days

Table 21.2: VerifySettle corridor health SLOs.

Corridor Health SLOs:

21.6 Layer-5 Stress Tests

Layer 5 earns the label "Value & Settlement" if it can navigate:

On-/off-ramp strangling. Can users still move value between BTC and privacy assets without centralized exchanges in country C? Do non-custodial corridors maintain $\geq 95\%$ success with 100% refund safety?

Corridor LP exit. If a major liquidity provider disappears, do corridors collapse, or does routing adapt?

Regulatory attack on privacy. Can lawful-privacy corridors (viewing keys + receipts) keep enough demand to sustain anonymity sets when shielded assets are labeled "high risk"?

21.7 Minimum Viable Layer-5 Corridor

Scope Type	What Is Revealed	Who Holds Key	Use Case
Transaction-scoped	Single transaction details	Counterparty or auditor	Receipt for specific payment
Time-bounded	All transactions in window $[t_1, t_2]$	Auditor with consent	Tax audit
Amount-bounded	Transactions > threshold	Compliance officer	AML monitoring
Full	All transactions	User only (default)	Personal records

Table 21.3: Viewing key disclosure scope model.

Viewing Key Disclosure Scope Model: **Key management:** Users generate and control all viewing keys. Keys are derived hierarchically; revoking a parent key invalidates children. No "master key" exists.

Requirement	Threshold
refund_safe	1.0
swap_success	$\geq 90\%$
ttf (p95)	≤ 2 hours
anon_set	≥ 500
Bridge class	A, B, or C only
Viewing key support	Transaction-scoped minimum

Table 21.4: Minimum viable corridor requirements.

Minimum Viable Corridor Spec: Corridors failing these requirements are flagged as "experimental" and excluded from default PRK routing.

Chapter 22

The Modular Stack

Parts III and IV described the stack layer-by-layer. This section flips the view: instead of layers, think in terms of **modules** builders actually touch and the **primitives** the system enforces underneath.

22.1 Primitive Catalog: The Twelve Pieces

The twelve primitives fall into four clusters:

Compute / Consensus Primitives (AI-Money Substrate)

1. **MatMul-PoUW kit** — A work function that turns matrix multiplication into a useful lottery with verification asymmetry. Canonical sizes (e.g., MATMUL_4096), seeding rules, and adversarial tests ensure that:
   - production cost scales $\sim O(n^3)$,
   - verification cost scales $\sim O(n^2)$,
   - and cheating is more expensive than being honest.

   This is the Duplex-style substrate: when proofs from this kit clear, we mint **verified FLOPs** rather than heat.
2. **Verified-inference harness** — An Ambient-style proof-of-logits layer for AI workloads: deterministic transcripts, randomized audits, peer-prediction, and stake-and-slash for dishonesty. Where full ZKML is not yet practical, the harness gives us:
   - probabilistic guarantees that outputs came from model M under policy P;
   - receipts per inference that VCO and PFS can route and price.

   As ZKML matures, the same harness becomes the "front door" for full proofs.
3. **Canonical workload registry** — A dictionary of standardized SKUs for useful work: `MATMUL_4096`, `INFER_LM_70B_256TOK`, `PROOF_2^20`, `PROVENANCE_VIDEO_V1`, `SETTLEMENT_BTC_XMR_V1`, etc.

Inference Proof Tier Taxonomy Verified inference is the most economically significant—and currently most fragile—part of the "AI Money" claim. To prevent readers from concluding "so AI Money is backed by trust," we tier the guarantees explicitly:

Tier	Verification Type	Description	Error Bound	WC Weight
Tier A	Cryptographic correctness	Full ZK proof of inference (ZKML)	0 (exact)	1.0×
Tier B	Probabilistic soundness	Audited transcripts + randomized checks + peer prediction	Explicit (e.g., 99% confidence)	0.6×
Tier C	Attestation-backed	TEE attestation + sampling audits	Implicit (trust TEE + sampling)	0.3×

Table 22.1: Inference proof tier taxonomy.

Use Case	Minimum Tier	Rationale
WC-Base issuance (full weight)	Tier A	Monetary issuance requires highest assurance
WC-Base issuance (discounted)	Tier B	Acceptable with explicit error bounds and discount
Service market only	Tier C	Can sell inference; cannot mint WC-Base
Collateral (pristine)	Tier A only	DeFi integrations require cryptographic certainty
Collateral (standard)	Tier A or B	With appropriate haircut
Collateral (ineligible)	Tier C	Not acceptable as collateral

Table 22.2: Tier eligibility rules.

Why this matters: Without this taxonomy, "verified inference" conflates cryptographic proof with probabilistic audit with vendor attestation. By making tiers explicit and tying them to WC issuance weights, the thesis acknowledges engineering reality (full ZKML isn't always practical) while preserving the monetary story (only high-assurance verification backs money-like instruments).

Tier B requirements (reference): For Tier B verification to qualify:

- **Transcript determinism:** Given the same model + input, transcript must be reproducible.
- **Audit rate:** $\geq 10\%$ of inferences are independently re-executed by random auditors.
- **Peer prediction:** Multiple independent provers; divergence triggers investigation.
- **Stake-and-slash:** Provers stake collateral; fraudulent transcripts → 100% slash.
- **Error bound publication:** Explicit statistical guarantee (e.g., "99% confidence that output matches claimed model within ε").

Migration path: As ZKML matures: (1) Tier A coverage expands to more model classes; (2) Tier B issuance weight increases toward 1.0× as probabilistic guarantees tighten; (3) Tier C is deprecated from WC eligibility entirely. The goal is for inference verification to converge to Tier A over time, but the system functions with tiered guarantees in the interim.

Proof Primitives (Receipts as First-Class Objects)

4. **PAL SDK (Proofs-as-a-Library)** — Developer surface for Layer 4.
5. **Multi-ZK adapter & auto-selector** — Policy engine keeping proofs a commodity, not a ven-

dor feature.

6. **PIDL (Proof Interface Definition Language)** — Minimal receipt schema binding claim, workload, proof, SLA, timestamps, and signatures.

Privacy & Settlement Primitives (Private-Money Substrate)

7. **Adaptor-signature atomic-swap kit** — Library for non-custodial BTC↔ZEC/XMR settlement with refund-safety.
8. **Lawful-privacy corridor pack** — Hooks for viewing keys + receipts enabling regulated actors to prove compliance.
9. **Bridge-safety templates** — Pattern book for cross-chain settlement avoiding "magic multisigs."

Market & Telemetry Primitives

10. **SLA escrow & slashing contracts** — Standard contracts for Bronze/Silver/Gold tiers with automatic refunds or slashes.
11. **VerifyPrice telemetry & methodology** — Measurement side of verification asymmetry.
12. **Neutral router & fairness tests** — Matching logic with house-share caps and entry-latency metrics.

PFS (Proof Factory Stack) bundles primitives 4–6, 10–12 to turn claims into routed proof jobs.

VCO (Verified Compute Orchestrator) bundles 1–3, 5, 10–12 to turn useful workloads into Work Credits.

22.2 Reference Application: Private Treasury & Payroll

Scenario. A globally distributed company wants to pay staff in a way that avoids single-jurisdiction custody risk, preserves employee privacy, and remains auditable.

Flow:

1. **Create.** Finance defines a payroll batch with policy.
2. **Compute.** Application logic computes net pays and corridor selection.
3. **Prove.** PAL generates proofs that payments are within policy.
4. **Settle.** PRK executes payments over BTC↔ZEC/XMR corridors.
5. **Verify.** Auditors verify correctness via proofs and receipts.

22.3 Reference Application: Media Provenance

Scenario. A media network guarantees that "gold channel" content is accompanied by verifiable provenance.

Flow:

1. **Create.** Cameras capture footage with Layer-0 attested hardware.
2. **Prove.** Proof factories generate `Provenance(C, chain)` proofs.
3. **Settle.** Advertisers pay creators over privacy rails, conditioned on valid proofs.
4. **Verify.** End-users verify that content passed through the stated chain.

22.4 Reference Application: Verified Inference

Scenario. An AI service offers "verified inference" to enterprises who don't want to trust a black-box API.

Flow:

1. **Create.** Client submits inference request with constraints.
2. **Compute.** Proof factory runs the inference.
3. **Prove.** PAL compiles into `Inference(M, X; policy)` with proof.
4. **Settle.** Payment via PRK contingent on valid proof.
5. **Verify.** Client verifies proof and accepts output.

AI Money is just Work Credits and derivatives for these inference workloads.

22.5 Reference Application: Proof/Compute Procurement

Scenario. A DAO wants to pre-buy proof and compute capacity for future needs.

Flow:

1. **Create.** DAO defines demand curves for workloads over time.
2. **Compute.** Procurement module computes optimal schedules.
3. **Prove.** As Work Credits are minted, DAO purchases them.
4. **Settle.** Payments over privacy rails; Work Credits held in treasury.
5. **Verify.** Anyone audits backing via VerifyPrice, profiles, and proof metadata.

The DAO's treasury becomes **triad-backed**: part BTC/ETH, part ZK Money, part AI Money.

22.6 Part IV Summary

Part IV is where the triad learns to **speak economics**:

- **Layer 4** turns computation and provenance into standardized, verifiable work units.
- **Layer 5** turns those work units into private, non-custodial value flows.

Only with these in place does it make sense to talk about **Private Money** and **AI Money** as more than metaphors. In Part V, we climb to **Layer 6 – Governance & Telemetry**, where neutrality and repression-resilience are kept falsifiable, not just promised.

Part V
Governance, Telemetry & Neutrality

Parts I–IV argued that the old soft guarantees are failing; that Privacy, Proofs, and Compute can function as monetary primitives; that a seven-layer cypherpunk stack can supply these primitives; and that Layers 4–5 turn work into truth and truth into private, non-custodial settlement.

Part V looks at the same system from the **operations and governance** angle. If the triad is to behave like a store of value under repression, then **"we promise" is not enough**. Neutrality, censorship-resistance, and verification economics must be **measured** and **governed** in ways that are falsifiable, not vibes.

Chapter 23

Layer 6: Governance & Telemetry

Layer 6 is the **control plane** and the **nervous system** of the stack.

Everything below it—hardware, comms, distribution, identity, proofs, settlement—can drift, centralize, or quietly break without anyone noticing until it's too late. Layer 6 refuses to let that drift remain invisible. It insists that:

- **Neutrality** (no favored flows, no hidden house edge),
- **Repression-resilience** (still works under capital controls and censorship), and
- **Verification economics** (VerifyPrice, VerifyReach, VerifySettle)

  are **service-level objectives (SLOs)**, not marketing copy.

23.1 "No Dashboards, No Trust": Public SLOs as Constitution

Most protocols ship documents called "constitutions." In practice, the real constitution is whatever you cannot silently violate without getting caught.

For this stack, the constitution is:

- a **set of SLOs** on triad capacity and neutrality, plus
- a **set of dashboards and receipts** that make violations obvious.

Examples:

- "p95 VerifyPrice for canonical workloads stays below T seconds on commodity hardware."
- "Top-N prover/LP/router share remains below X% by hashpower/FLOPs/liquidity."
- "BTC↔ZEC/XMR corridors achieve $\geq 95\%$ success with 100% refund safety."
- "At least K distinct jurisdictions and hardware profiles are actively verifying."

These are not just operational goals; they are the **monetary invariants**. Layer 6's first job is to **publish and maintain these SLOs as public contracts**.

23.2 Control Surfaces: Parameters, Upgrades, Emergency Powers

The second job of Layer 6 is to define the **control surfaces**: the knobs that can be turned, by whom, and with what receipts.

Economic Parameters:

- Block reward curves and Work Credit issuance schedules.
- Fee policies (floor, burn, split between security and dividends).
- PoUW weighting across workloads.

Technical Parameters:

- Circuit versions and proof-system parameters.
- Accepted hardware profiles (additions, deprecations).
- Corridor policies (timelocks, refund windows, anonymity thresholds).

Operational Parameters:

- Telemetry granularity and reporting requirements.
- Incident severity levels and response playbooks.
- Key ceremonies and quorum sizes.

For each surface, Layer 6 specifies: (1) who can propose a change, (2) who can ratify it, (3) what data must be presented, (4) what delays and rollback mechanisms exist, and (5) which changes are "emergency powers."

23.3 Bell-Labs-Style R&D vs. On-Chain Governance vs. Off-Chain Norms

There are three governance "forces" that need to be reconciled:

1. **Bell-Labs-style R&D.** A research org with autonomy to explore new circuits, proof systems, hardware profiles, and corridor designs.
2. **On-chain governance.** Token- or Work-Credit-weighted voting, parameter changes baked into protocol logic.
3. **Off-chain norms and institutions.** Foundations, labs, dev collectives, and community norms.

Layer 6's thesis is not "pick one." It is:

> **Use Bell-Labs-style R&D to discover, on-chain governance to ratify and constrain, and off-chain norms to fill the gaps—but keep all three under telemetry.**

Concretely:

The R&D lab:

- Maintains public roadmaps and risk registers.
- Publishes upgrade proposals with quantified impacts on VerifyPrice, VerifyReach, VerifySettle, decentralization, and hardware profiles.

- Runs testnets and shadow deployments.

On-chain governance:

- Controls **scarce resources**: issuance, reward splits, canonical workloads, acceptance/deprecation of proof systems and hardware profiles.
- Is bounded by **constitutional SLOs**: certain changes are simply not allowed if they push metrics beyond thresholds unless higher-order safety processes trigger.

Off-chain institutions:

- Operate under **public charters** that explicitly state their mandate and constraints.
- Are expected to publish minutes, risk assessments, and incident reports.

This three-body system is inherently unstable; Layer 6 keeps it from drifting into pure plutocracy or pure priesthood by insisting that **all three bodies are visible in telemetry**:

- Changes in code and parameters appear on-chain.
- Lab work appears in open repos and benchmarks.
- Institutional decisions appear in charters, public calls, and reports.

No dark corners; no "just trust us, we're the stewards."

23.4 Governance as SLOs: The Five Invariants

Governance and operations exist to keep one invariant true under stress:

> **Pay the machine only for work anyone can verify cheaply—and give humans lawful privacy by default.**

Everything below—roles, metrics, runbooks, and legal posture—turns that sentence into procedures, with receipts. Where policy pressure rises, we default to measurement and non-custodial design rather than new gatekeepers. "No dashboards, no trust" is not a slogan; it is the rule for deciding when the system is still safe to treat as money.

Governance and operations are about what happens when the world pushes back. They are the difference between a beautiful mechanism that works in a friendly lab and an actual monetary substrate that survives YCC (yield-curve control), capital controls, app-store bans, and "secure enclave" mandates.

We can summarize the protocol's "constitution" as **five invariants**:

1. **Verification asymmetry.** For each canonical workload W, verification stays much cheaper than production. Formally, the verification overhead $r(W) = v(W)/p(W)$ remains at or below a published bound (e.g., 0.3, with a stretch goal of 0.1), and this is reported via VeriFyPrice(W) on the Proof & Compute Board. If $r(W)$ drifts up or p95 verify times blow past

targets, the economic hinge of the triad fails; this is a Sev-1 condition.

2. **Open admission.** Anyone who brings correct work clears. Admission is not gated by identity, licensing, or proprietary hardware. Routers are neutral and open-source; house share and time-to-first-fill for new provers/LPs are visible on the Neutrality & Admission Board. If honest new entrants cannot join and get paid within a bounded window, decentralization is already drifting.
3. **Useful-work security budget.** Block rewards and fees underwrite **useful work**—MatMul-PoUW, verified inference, provenance and settlement proofs—under explicit SLAs, not waste. The security budget is tied to canonical workloads with published VerifyPrice, not to arbitrary hash puzzles with no external demand.

   Inference Verification Tiers and WC Eligibility: To prevent the "AI Money is backed by spot checks" critique, inference workloads are tiered by verification strength:

Tier	Verification Type	WC Eligibility
Tier A	Cryptographic (full ZK)	Full eligibility (1.0×)
Tier B	Probabilistic (bounded error, audited)	Discounted (0.6×)
Tier C	Attestation-only (TEE + sampling)	Service market only; no WC issuance

Table 23.1: Inference verification tiers and WC eligibility.

   Rule: Only Tier A and Tier B verified inference is eligible for Work Credit issuance. Tier B issuance is discounted by a published "soundness haircut" reflecting error bounds and audit rate. Tier C can exist as a service market but is not collateral-grade—it represents trust in TEE vendors, which is exactly what the thesis aims to minimize.
4. **Lawful privacy by design.** Settlement is neutral and non-custodial by default—adaptor-sig atomic swaps, shielded pools, privacy rails—but comes with **viewing keys and auditable PIDL receipts** so law-abiding users can evidence obligations without re-introducing chokepoints. The Settlement & Privacy Board reports swap success, refund safety, and anonymity-set health; flows that never touch custodians should still leave enough receipts that auditors can do their jobs.
5. **Verifiable machines at Layer 0.** Canonical proving and attestation paths run on open designs with sampled supply chains; hardware attestation becomes one input to proofs, not a vendor priesthood. Hardware profiles, lot-sampling coverage, and profile incidents show up on the Hardware/Layer-0 panes of the boards.

These invariants are **hard to change and easy to measure**. Everything else—fee curves, circuit versions, work-function parameters, corridor lists—is configuration.

To keep configuration from degenerating into a governance circus, we tie changes to **observable triggers** instead of vibes:

- If **VerifyPrice p95** for a workload W drifts above its target band for a sustained window, that

workload's circuit and parameters are queued for revision. Blocking the change requires an explicit override that cites alternative SLOs and appears in the governance log.

- If **entry latency** for new provers/miners/LPs rises beyond a threshold, or the **top-N share** and house share breach caps, neutral routers automatically ratchet down house share and prioritize small bidders until the metric recovers.
- If a **swap corridor's refund-safety** ever falls below 100% in VerifySettle, that corridor is removed from admissible routes by default. Re-listing requires a synthetic regression suite to pass (including stress tests) plus a public post-mortem.

In this frame, "governance" does not micromanage every adjustment; it **writes the SLOs and the triggers**. Operations enforces them and publishes the receipts. Most changes are **data-driven roll-forward**: parameters evolve in response to Verify* drift, with clear before/after numbers.

To keep rule-making separate from rent-collection, we distinguish two broad roles:

- A **spec & telemetry steward** that publishes ABIs, PIDL schemas, canonical workloads, hardware profiles, and the VerifyPrice/Reach/Settle dashboards. It does not run routers or take custody. Its mandate is to keep the bill of materials and metrics honest and up to date.
- **Market participants**—miners, provers, routers, corridor LPs—who compete on price and reliability under those public SLOs, with SLA escrow and slashing keyed to PIDL receipts. Their incentives are economic; their constraints are the invariants and metrics.

This "two-chamber" structure keeps the job of defining rules and SLOs distinct from the job of selling capacity under those rules. It avoids the familiar pattern in which the entity that can edit parameters also happens to own the biggest fleet of nodes.

Governance as SLOs, not personality, is what lets the system answer policy pressure with dashboards and runbooks instead of press releases. When asked "how do you survive X?", the right answer is not "trust the foundation," but "look at this board, this trigger, and this incident playbook."

23.5 Constitutional Enforcement

"SLO-bounded governance" is only credible if the enforcement mechanism is specified.

23.5.1 Machine-Enforced Constraints (Hard)

Certain invariants are enforced **on-chain or in protocol code**:

23.5.2 Override Path (Slow, Expensive)

Some constraints need flexibility. The override path is **deliberately expensive**:

- **Supermajority:** $\geq 67\%$ of governance weight.

Constraint	Enforcement
Issuance cap per epoch	Smart contract rejects minting transactions exceeding cap
Corridor refund timeout	Atomic swap contracts enforce timeout
House-share cap	Neutral router contracts reject bids exceeding share
VerifyPrice bound violation	Workload suspended from WC eligibility

Table 23.2: Machine-enforced constitutional constraints.

- **Long timelock:** $\geq$ 30 days (90 days for issuance-related).
- **Risk memo:** Mandatory written analysis, hash anchored on-chain.
- **Sunset clause:** Override auto-expires after 12 months.

23.5.3 Emergency Path (Narrow)

True emergencies (proof system break, active exploit) require faster action. The emergency path is **narrow and auditable**:

Allowed	Not Allowed
Deprecate proof system	Increase issuance
Delist corridor	Confiscate user funds
Quarantine hardware profile	Bypass refund safety
Freeze workload class	Mint unbacked credits

Table 23.3: Emergency path scope limitations.

Requirements: $\geq$ 3 of 5 security signers; auto-expire in 7 days; postmortem within 14 days.

23.6 Monetary Constitution

23.6.1 Issuance Envelope

Work Credit issuance is governed by:

- **Base schedule:** Halving every N blocks (e.g., 4-year halvings).
- **Capacity modulator:** $\pm$10% based on verified capacity growth.

  Capacity modulator formula:

$$\text{Issuance}_{\text{epoch}} = \text{BaseSchedule}_{\text{epoch}} \times (1 + 0.1 \times \tanh(\text{CapacityGrowthRate} - \text{TargetGrowthRate}))$$

23.6.2 Fee Routing

Why burn? Burn creates deflationary pressure tied to usage. High demand → high fees → high burn → supply reduction → value appreciation. This closes the loop between utility and asset

Destination	Share	Mechanism
Capacity providers	70%	Paid via SLA escrow
Protocol burn	20%	Permanently removed
Assurance budget	10%	Funds sampling, audits, security

Table 23.4: Fee routing split.

value.

23.6.3 Retirement Rule

In addition to fee burn, WC-Base is retired in specific circumstances:

Trigger	Retirement
WC-Voucher redemption	Underlying WC-Base used to fulfill capacity claim is retired
Slashing	100% of slashed collateral is burned (not redistributed)
Failed workloads	Fees for failed proofs (prover fault) are burned, not paid

Table 23.5: WC-Base retirement triggers.

23.6.4 Risk Haircuts

Work Credits minted under weaker assurance levels receive discounted issuance:

Factor	Haircut	Example
Hardware profile (L0 grade)	L0-A: 0.9×; L0-B: 1.0×; L0-C/D: 1.1×	Lower-assurance hardware earns fewer credits
Verification tier	Tier A: 1.0×; Tier B: 0.6×; Tier C: 0.3×	Probabilistic verification earns less than cryptographic
Prover concentration	>20% share: 0.9×; >30%: 0.8×	Concentrated provers earn progressively less

Table 23.6: Risk haircuts for Work Credit issuance.

These haircuts are protocol parameters (not discretionary); they appear in dashboards and apply automatically.

23.6.5 Coverage Target

The **fee-coverage ratio** measures what share of the security budget comes from fees vs. issuance:

$$\text{FeeCoverage} = \frac{\text{FeeRevenue} + \text{Burn}}{\text{TotalSecurityBudget}}$$

Why this matters: A system that relies entirely on issuance is inflating its way to security.

Target	Timeline
FeeCoverage $\geq$ 30%	Year 1
FeeCoverage $\geq$ 50%	Year 3
FeeCoverage $\geq$ 80%	Year 5+

Table 23.7: Fee coverage trajectory.

A system where fees cover the budget has structural demand. The trajectory from issuance-funded to fee-funded is the path from "speculative token" to "productive asset."

23.6.6 Value Capture Loop

```
+-------------------------------------------------------------+
|                    VALUE CAPTURE LOOP                       |
+-------------------------------------------------------------+
|  DEMAND (users need Privacy, Proofs, Compute)               |
|    |                                                        |
|    v                                                        |
|  Users pay fees in WC-Base for triad services               |
|    |                                                        |
|    +-- 70% -> Capacity providers (stakers must hold WC-Base)  |
|    +-- 20% -> Burned (supply reduction)                     |
|    +-- 10% -> Assurance budget (sampling, audits)           |
|    |                                                        |
|    v                                                        |
|  Provers/routers must stake WC-Base as collateral           |
|    |                                                        |
|    v                                                        |
|  Issuance capped by schedule + capacity modulator           |
|    |                                                        |
|    v                                                        |
|  NET EFFECT: Demand -> Fees -> Burns + Staking -> Scarcity  |
|              Scarcity + Utility -> Store of Value Premium   |
+-------------------------------------------------------------+
```

Why demand doesn't simply expand supply proportionally:

1. **Issuance cap is hard:** Protocol enforces epoch-level cap regardless of demand.
2. **Burn is automatic:** Higher demand → higher fees → higher burn → lower net supply.
3. **Staking locks supply:** More provers/routers → more collateral locked → less circulating supply.
4. **Haircuts constrain issuance:** Weak verification or concentrated provers earn less, limiting supply growth.

This is the “bond indenture” that makes the SoV claim auditable, not asserted.

Chapter 24

Extended Telemetry

"No dashboards, no trust" is the mantra. But dashboards themselves can be gamed. If Layer 6 claims "measurement is the constitution," then the measurement regime itself must be bullet-proof.

24.1 Reference Verifier Classes

All Verify* metrics are measured on **standardized reference hardware**:

Class	Hardware Spec	Use Case
Laptop-Class	4-core CPU, 16GB RAM, SSD, no GPU	Default for SLOs
Mobile-Class	ARM SoC, 8GB RAM	Edge/mobile
Server-Class	32-core CPU, 128GB RAM	High-throughput

Table 24.1: Reference verifier classes.

Rule: All constitutional SLOs (e.g., "VerifyPrice p95 $\leq$ 5s") are defined against **Laptop-Class** unless otherwise specified. This ensures the "anyone can verify" promise is falsifiable.

24.2 Reference Verifier Implementations

Requirement	Specification
Independent implementations	≥ 2 independent verifier codebases per canonical workload tier
Open source	All reference verifiers must be open-source with reproducible builds
Versioned and hashed	Binary hashes are part of the canonical workload registry
Deterministic	Same proof + same verifier version $\rightarrow$ same result, always

Table 24.2: Reference verifier implementation requirements.

Why this matters: A single verifier implementation can have bugs that inflate or deflate measurements. Two independent implementations provide a cross-check.

24.3 Cost Vector Definition

VerifyPrice is not a single number. It is a **cost vector** for each workload W and verifier class V:

$$\text{VerifyPrice}(W, V) = (t_{50}, t_{95}, m, b, e, f)$$

Where:

- t_{50}, t_{95}: Median and 95th-percentile verification **time** (seconds)
- m: Peak **memory** usage (MB)
- b: **Bandwidth** consumed (bytes)
- e: **Energy** estimate (Joules)
- f: **Failure rate**

24.4 Sampling and Anti-Cherry-Pick

The most common attack on telemetry is selective reporting: only show receipts from good regions, fast hardware, or favorable conditions. The sampling regime prevents this:

Mechanism	Specification
Public randomness beacon	Receipt selection seeded by on-chain randomness (e.g., block hash, VRF output)
Stratified sampling	Samples drawn proportionally from: workload class, hardware profile, geographic region, time-of-day
Adversarial corpus	10% of test proofs are malformed or worst-case (max witness size, pathological inputs)
Continuous measurement	Rolling 24-hour windows; no periodic snapshots that can be gamed
Multi-operator	≥ 3 independent measurement operators; divergence $> 5\%$ triggers investigation

Table 24.3: Sampling mechanisms for Verify* telemetry.

Rule: Any dashboard point published without a verifiable sampling seed and stratification breakdown is not part of the official Verify* record.

24.5 Data Availability and Anti-Memory-Hole

Dashboards are views. The source of truth is the **receipt corpus**. Receipts must be:

Why this matters: If a government or CDN can delete "bad months," the telemetry regime is captured. Content-addressing + neutral anchoring makes deletion detectable.

Requirement	Specification
Content-addressed	Receipt sets identified by Merkle root
Anchored	Merkle roots periodically committed to a neutral ledger (e.g., Bitcoin, Ethereum)
Archived	Raw receipt data retained for ≥ 2 years by ≥ 3 independent archivists
Queryable	Any observer can request receipts corresponding to a dashboard point

Table 24.4: Data availability requirements.

24.6 Reproducibility Contract

A valid dashboard datapoint is defined as:

$$Datapoint = (d_{\text{root}}, v_{\text{harness}}, v_{\text{verifier}}, r)$$

Validity check: Anyone can:

1. Fetch receipts by d_{root} (dataset root)
2. Run v_{harness} (harness version) with v_{verifier} (verifier version)
3. Verify r (result) matches

If the results don't match, the datapoint is invalid and the discrepancy is a Sev-1 incident.

Published artifacts:

- Harness code (open source, tagged)
- Verifier binaries (reproducible builds, hashed)
- Dataset roots (timestamped, anchored)
- Aggregation scripts (deterministic)

This transforms "no dashboards, no trust" from philosophy into a **reproducible measurement contract**.

24.7 How to Measure: Receipts, Not Vibes

VerifyPrice was introduced at Layer 4; VerifyReach at Layer 1; VerifySettle at Layer 5. Layer 6 treats them as a **coherent observability regime** rather than three unrelated graphs.

The observability story only works if it is itself verifiable. That is why every claim, proof, settlement, and SLA outcome is turned into a **receipt** with enough structure that anyone can replay the metrics on a reference verifier.

In practice, this means:

- Every proof receipt carries not only the proof artifact but also start/end timestamps, resource usage, and a hardware profile tag.
- Settlement receipts encode corridor, timing, refund status, and failure codes.
- Routers emit anonymized samples of order books, matched and unmatched bids, and house

vs. third-party flags.

- Hardware vendors publish lot-attestation artifacts that receipts can reference as part of their profile.

A public "VerifyPrice Observatory" then ingests these receipts and emits the aggregate vectors we refer to in the text. Anyone can pull the raw receipts corresponding to a dashboard point, feed them into the same open-source verifier suite, and see whether they obtain the same numbers. If they do not, the mismatch is itself an incident.

This architecture enforces a simple discipline: **if it matters, there is a receipt for it.** If a claim about performance, neutrality, or safety cannot be tied back to receipts that anyone can re-check, it is out of scope for the monetary thesis.

They are the three primary dials for answering:

- *Is verification still cheap?* (VerifyPrice)
- *Can users still reach the network under pressure?* (VerifyReach)
- *Can users still move value privately and safely?* (VerifySettle)

We can think of this observability regime as watching the stack across five planes:

Proof & compute plane Are receipts reliable, and does verification asymmetry actually hold under load?

Settlement & privacy plane Do non-custodial payouts actually clear, with privacy and refund-safety matching the promise?

Neutrality & admission plane Can new provers, miners, routers, and mirrors join on equal footing, or has the system ossified into a club?

Economic coverage plane Are real fee flows, not issuance, paying a growing share of the security and operations budget?

Layer-0 hardware plane Do "verifiable machines" remain falsifiable in practice, or did we slide quietly back into "trust the vendor"?

Each plane maps directly to the SoV requirements from Part II: credible scarcity, cheap public verification, censorship-resistance, neutrality, native demand, lawful privacy, and duration-neutrality. If any plane drifts too far from its targets, the monetary thesis weakens, no matter how elegant VerifyPrice looks in isolation.

24.8 The Four Public Boards

Metrics surface as four public "boards":

Proof & Compute Board. Summarizes VerifyPrice per workload, SLA attainment, queue depths, and failure modes.

Settlement & Privacy Board. Reports swap success, refund safety by corridor, time-to-finality, and anonymity-set health.

Neutrality & Admission Board. Tracks time-to-first-proof for new entrants, top-N share, geo/ASN

distribution, and house share.

Economic Coverage Board. Charts fee-coverage ratio, physical VerifyPrice against SLOs, and demand curves indexed to policy events.

These boards are not marketing pages; they are part of the protocol's public interface.

24.9 Detecting Drift

Telemetry detects **drift**:

Verification cost creep. VerifyPrice shows p95 verify time creeping up. Response: R&D lab proposes fixes; governance ratifies changes.

Verifier concentration. 80–90% of verifications performed by one hardware profile. Response: adjust rewards to subsidize diverse profiles.

Reachability degradation. VerifyReach shows a major country blocking protocol ports. Response: shift transports, promote alt routes.

Corridor fragility. VerifySettle shows corridor success dropping. Response: re-weight corridors, onboard new LPs.

24.10 Making Neutrality and Repression-Resilience Falsifiable

Neutrality and repression-resilience are usually sold as moral properties. In this thesis, they are **falsifiable hypotheses**:

Neutrality hypothesis: "No actor or group can reliably censor, privilege, or front-run specific flows across Layers 1–5 without being detected and without other actors being able to route around them."

Repression-resilience hypothesis: "Under YCC, capital controls, DPI, and blacklists, the triad's core capacities remain usable enough that Work Credits still behave as stores of value."

Telemetry makes these hypotheses testable:

- If VerifyReach shows certain flows consistently blocked and no alternatives emerge, neutrality is falsified.
- If VerifySettle shows privacy corridors collapsing under regulation, repression-resilience is falsified.
- If VerifyPrice shows verification cost pushed beyond reach, "anyone can verify" is falsified.

A system that cannot tolerate bad news cannot be money; it can only be marketing.

24.11 Policy-Attack Stress Harness

A monetary stack designed for repression should publish the exact conditions under which it considers itself passing or failing. For each attack scenario, we define tests and pass criteria:

Yield-curve control & negative real yields:

- **Test:** Over 24–36 months of -300 to -500 bps real yields, fee+burn share of security budget is stable or rising.
- **Pass:** Tokens behave like claims on useful capacity, not synthetic bonds.

On-/off-ramp throttling:

- **Test:** Non-custodial corridors maintain $\geq$ 95% success and 100% refund safety; anonymity sets remain above thresholds.
- **Pass:** Users can move capital via non-custodial rails even when custodial ramps are hostile.

Provenance mandates:

- **Test:** Canonical provenance workloads have healthy VerifyPrice; regulators can run `verify(receipt)` with commodity tooling.
- **Pass:** The triad is the cheapest way to comply with provenance mandates.

Hardware backdoor mandates:

- **Test:** A non-trivial share of receipts are tied to open or sampled hardware profiles, not a single opaque TEE.
- **Pass:** Hardware mandates become one option among many, not a kill switch.

When the stack claims "we can survive policy attacks," the proof is not a blog post; it is a corpus of **stress-harness runs with metrics and receipts**.

Chapter 25

Legal, Policy, and Jurisdictional Posture

Layer 6 sits where protocol meets law and politics. If the triad is to be sustainable, it must support **lawful privacy**, defend itself against bans without being defined by them, and avoid capture by any single jurisdiction.

25.1 Lawful Privacy as Protocol Design

"Lawful privacy" is often a euphemism. Here it means something precise:

- **By default**, settlement and compute flows are private.
- **By design**, participants can opt into **selective disclosure** via viewing keys and receipts.

Constitutional Constraint: Policy = Predicates

Policy compliance is defined as **ZK-checkable predicates** over credentials and receipts, plus **selectively disclosed audit windows** via viewing keys.
It is **not** defined as global graph inspection or universal traceability. Any policy requiring ubiquitous transaction tracing is treated as **incompatible with the monetary design**.

Requirement	How Satisfied	Cannot Require
AML screening	ZK proof of "sender $\in$ cleared-set"	Full sender identity
Jurisdictional limits	ZK range proof of amount	Exact amount
Tax audit	Time-bounded viewing key	Universal history
Counterparty verification	ZK set-membership proof	Real-name disclosure

Table 25.1: Lawful privacy: what can and cannot be required.

25.2 Defense-in-Depth Against Bans and Sanctions

Regimes may attempt to ban privacy assets, sanction addresses, mandate KYC at all edges, or classify triad services as "unlicensed financial institutions."

Layer 6 cannot prevent law from existing, but it can:

- Avoid giving any single jurisdiction a **kill switch**.
- Maintain **jurisdictional diversity** in hardware profiles, proof factories, corridor LPs, and foundation incorporation.
- Provide **fallback modes**: local-first clients, mesh/satellite relays, minimal CLI modes that look like ordinary traffic.

The legal posture is **defensive, not antagonistic**: do not advertise the stack as a tool to evade law; do insist that users retain agency and lawful privacy.

25.3 Labs, Foundations, and Neutral Router Commitments

Key commitments:

Neutral router charters. Router operators commit to content-agnostic routing. Deviation is punished in-protocol and reputationally.

Foundation / lab structure. Legally separate entities with public charters that define roles and explicitly renounce certain powers.

Multi-jurisdiction footprint. Incorporations and key staff spread across legal regimes to avoid single-point capture.

Chapter 26

Operator & Investor Checklist

The last piece of Layer 6 is a **practical checklist**: how builders, operators, and allocators decide whether this stack qualifies as a next-gen store of value.

26.1 Triad Supply: Privacy, Proofs, Compute

For each leg of the triad:

Privacy:
- Are there live, non-custodial privacy corridors with measurable anonymity sets?
- Are shielded pools growing in depth and churn?
- Are lawful-privacy patterns (viewing keys + receipts) actually used?

Proofs:
- Is there robust supply of proofs for canonical workloads?
- Are VerifyPrice SLOs met consistently?
- Is proof production diversified across hardware profiles and geographies?

Compute:
- Are meaningful amounts of useful compute being proven?
- Is there a liquid market for Work Credits?
- Are triad workloads tied to real-world demand?

26.2 Stack Health: Layers 0–6

Layer 0. Documented hardware profiles? Open hardware participation? Visible incident reports?
Layer 1. VerifyReach shows resilient connectivity? Alternative transports exist?
Layer 2. Clients distributed via multiple channels? Reproducible and signed binaries?
Layer 3. Pseudonymous credentials integrated? Actors can prove rights without doxxing?

Layer 4. Canonical workloads documented? VerifyPrice within SLOs? Proof production decentralized?
Layer 5. Non-custodial corridors healthy and diversified? Refund safety enforced?
Layer 6. Public dashboards? Documented governance processes? Recent incident reports?

26.3 Telemetry Honesty

Telemetry itself can be gamed. Checklist for honesty:

- **Open data.** Are raw metrics public? Can independent teams reproduce Verify* metrics?
- **Client diversity.** Metrics gathered by multiple operators?
- **Incentive alignment.** Any incentive to under-report problems?
- **History.** Historical series for metrics, or only recent snapshots?

26.4 Red Flags and Failure Patterns

Finally, some patterns that should trigger skepticism:
Closed hardware monoculture. One TEE, one vendor, no sampling, no open profiles → Layer-0 cliff.
Opaque bridges. "Magic multisig" bridges with no proofs, unclear jurisdiction, and no incident history → not settlement rails, just custodial risk.
Foundation fiat. Major parameter or policy changes via blog post, with no on-chain trace, no telemetry, and no incident report → governance capture.
Zombie corridors. Privacy rails that haven't moved meaningful volume in months but still appear in marketing. Anonymity sets that are effectively dead.
Proof theater. Lots of "ZK" branding but no public VerifyPrice metrics, no canonical workloads, no commodity verifiers.
Governance theater. Token votes with single-digit participation deciding fundamental parameters; no constraints; no SLOs.

Any one of these is survivable in the short run; taken together, they say:

> *"This is not a triad-backed money system; it is a platform dressed as one."*

A system with **no telemetry** is untrustworthy. A system with **one vendor's telemetry** is fragile. A system with **multi-source, reproducible telemetry** has a shot at being money.

Part V closes the governance loop:

- It treats SLOs and dashboards as the **constitution**.
- It makes **Verify* telemetry** the lifeblood of neutrality and repression-resilience.

- It frames law and politics as **constraints to be engineered around**, not as deities to be appeased.
- It gives builders and allocators a **practical checklist** for deciding whether a system's "store of value" claim is mathematically grounded or just well-typeset.

With Layer 6 in place, the stack has all three angles: Money (triad as SoV), Stack (Layers 0–6), and Telemetry (Verify* keeping it honest). Part VI can now focus on dynamics (adoption curves, risk, and implementation sketches): how this thing actually gets built, traded, attacked, and, if it works, quietly becomes part of what the world calls "money."

Part VI
Dynamics, Risk & Implementation

Parts I–V defined the triad as monetary base, built a seven-layer stack to supply it, and wired in governance and telemetry so neutrality and repression-resilience are falsifiable. Part VI asks the unromantic question:

> *What happens when this thing hits the real world? Who uses it first, how does it fail, and under what conditions does it quietly become "money"?*

Chapter 27

Adoption Curve & Ecosystem Dynamics

The stack is not adopted by "the market" in one step. It's pulled in by different constituencies, each with their own pain.

27.1 Four Phases of Adoption

The stack is not adopted by "the market" in one step. It's pulled in by different constituencies, each with their own pain.

We sketch the curve as four overlapping phases. These are **metric-anchored, not calendar-anchored**—transitions happen when telemetry shows they have happened, not when a date arrives:

Phase I: Cypherpunk-led. Plumbing proves out publicly; loop works without permission.
Phase II: Demand-led. Budgeted workloads close the fee loop; capacity becomes a line item.
Phase III: Composability-led. Receipts become default integration; invisible infrastructure.
Phase IV: Policy-led. Receipts become recognized evidence; jurisdictional differentiation.

These phases overlap and vary by geography and sector. The pattern is robust: early idealists, then necessity-driven budgets, then invisible defaults, and finally institutional and policy recognition. **Calendar horizons are intentionally omitted**—they age badly and invite cheap dunking. What matters is whether the phase gate metrics (below) are met.

27.1.1 Phase Gates and Failure Gates

The following table makes the adoption curve **auditable**. Each phase has entry gates (observable thresholds) and failure gates (conditions that falsify the phase or regress to a prior phase).

Phase Entry Gates:

Failure Gates (Phase Regression or Thesis Falsification):

Phase	Gate Metrics	Threshold
I → II	VerifyPrice exists for ≥ 3 canonical workloads with ≥ 6 months history	p95 within SLO
	VerifySettle for ≥ 1 corridor with refund_safe = 1.0	success ≥ 95%
	VerifyReach published for major regions	$\text{succ}_1 > 0.7$
	Receipt volume non-trivial	≥ 1000 receipts/day
II → III	Fee+burn covers ≥ 30% of security budget	Sustained 6+ months
	Budgeted workloads ≥ speculative in volume	Visible in receipt tags
	VerifySettle survives ≥ 1 policy shock	SLOs hold
	Capex/OPEX lines reference verification	Public disclosures exist
III → IV	Collateral/reserve usage in multiple venues	≥ 3 independent
	VerifyPrice/Settle stable through macro stress	SLOs hold during crisis
	Duration-neutral holding cohort	Measurable in on-chain data
	Fee coverage ≥ 50%	Sustained
IV	Legal codification of receipts	≥ 2 jurisdictions
	Public-sector participation	Visible in dashboards
	Geographical differentiation	Measurable in VerifyReach

Table 27.1: Phase entry gates for adoption curve.

Condition	Consequence
VerifyPrice p95 exceeds SLO for core workloads for ≥ 3 months with no credible remediation	Phase regression; SoV thesis weakens
Corridor refund_safe < 100% on admissible route (repeated, not isolated)	Phase regression; Private Money claim fails
Verification concentrates > 70% on single profile/jurisdiction for ≥ 3 months	Phase regression; decentralization claim fails
Receipt data becomes unverifiable (datasets unavailable, dashboards dark)	Telemetry capture; thesis unverifiable
Fee coverage collapses and workload mix becomes > 80% speculative	Phase II → I regression

Table 27.2: Failure gates for adoption phases.

Phase regression logic: Phases can regress. If Phase II metrics collapse (fee coverage falls, budgeted workloads disappear, corridors fail stress tests), the system slides back to Phase I. This is not failure of the technology—it's failure of adoption. The telemetry makes it visible so stakeholders can respond.

27.1.2 Adoption Curve To Expect

The adoption curve for this new monetary substrate will not be a single "flip" moment. It will unfold in discernible phases, each with its own constituencies, failure modes, and telemetry.

The through-line is simple. Under Bretton Woods II, you ran money by buying reserves, hiring lawyers, and trusting gatekeepers. Under Verification, you run money by buying capacity: privacy capacity, proof capacity, verified FLOPs. **Budgets stop paying for promises and start paying for work that anyone can check.**

"Phase I–IV" are not just vibes. Each phase has rough, observable thresholds in the same metrics we have already defined (VerifyPrice, VerifyReach, VerifySettle, FERs, Work Credit utilization). The point of the curve is that you can tell, from dashboards and receipts, which world you are in.

27.1.3 Phase I: Cypherpunk-Led (Prove the Plumbing in Public)

Phase I belongs to the people who already live in the future: open-source cryptographers, hardware hackers, privacy-chain communities, and early AI+ZK builders. Their job is to prove that the loop (Create/Compute → Prove → Settle → Verify) can be made real **without permission**.

In this phase, the main artifacts are reference implementations and receipts:

- **Open-admission prover markets** stand up on top of existing chains and zk networks. They publish live VerifyPrice dashboards for canonical workloads: PROOF_2^{20}, MATMUL_4096, basic PoL fingerprints.
- **MatMul-PoUW testnets** (Duplex-style patterns) show that verification overhead can remain $r(W) = v/p \leq 0.3$ on commodity hardware and that miners without favoured hardware profiles can still win blocks.
- **PoL pilots** (Ambient-like patterns) demonstrate hybrid verified inference with honest-output rates that can be measured and contested, not marketed.
- **zk-PoW networks** (Nockchain-style patterns) act as public receipt ledgers (places where proofs from many domains can be anchored and timestamped).
- **Non-custodial BTC↔XMR/ZEC swaps with refund-safe UX** move from GitHub curiosities to tools that people actually use.

(*Note: "Duplex-style," "Ambient-like," and "Nockchain-style" are referenced as design patterns, not endorsements of any specific project or ticker.*)

Nothing is "mainstream" yet. Most users are still speculators and hobbyists. But a few

things become hard facts rather than hopes: you can buy proofs as a service from permissionless markets; you can pay for them over privacy rails without custody; and anyone with a laptop can verify the receipts.

Phase I telemetry and triggers: **Phase I is real** (not hypothetical) once:

- **VerifyPrice exists for a small set of canonical workloads.** At least a handful of public markets publish VerifyPrice(W) dashboards for $W \in \{\text{PROOF\_}2^{20}, \text{MATMUL\_4096}, \text{INFER\_LM\_70B}\}$ with months of history, and p95 times stay within stated SLOs under stress.
- **VerifySettle is measured for at least one serious corridor.** For some BTC↔XMR or BTC↔ZEC corridor C, VerifySettle(C) is public and hits targets like success(C) ≥ 0.95, refund_safe(C) $= 1.0$.
- **VerifyReach is non-degenerate.** At least one verifier network publishes VerifyReach(N, R) for major regions with real multi-path reachability.
- **Receipt volume is non-trivial.** Receipt ledgers anchor a steady flow of PIDL receipts per day (≥ 1000) across multiple workloads.

Once those metrics are visible, the question is no longer "can this exist?" but "can it scale?"

27.1.4 Phase II: Allocator-Led (Proofs and Privacy Become Budget Lines)

Phase II begins when **allocators** (fund managers, corporate treasuries, exchanges, and large web platforms) start to treat privacy and proofs the way they once treated bandwidth: as recurring operating costs, not science projects.

By this point (**after Phase I gate metrics are met**, not by calendar date):

- Proof and compute networks publish open VerifyPrice and reliability dashboards with historical data. You can see, month by month, how p95 verification times, costs, and failure rates behave under load.
- The PaL SDK and settlement adapters have been integrated into serious applications: analytics pipelines, custody stacks, compliance tooling, provenance layers for media.
- Privacy corridors (particularly BTC↔XMR/ZEC) have polished GUIs, reference libraries, and documented latencies. Refund failures are statistical outliers with post-mortems, not routine hazards.

Allocators do what they always do when facing repression and technological change: they **reclassify**. Instead of framing privacy and proof capacity as speculative tokens, they treat them as line items required to keep operating:

- A bank's AI risk model must be run on verifiable compute with audit-friendly receipts, because regulators now ask for them.
- A media platform must attach cryptographic provenance to high-stakes content, because the liability of not doing so is too high.

- A trading venue must use private settlement rails to avoid leaking its entire order book and client graph.

They do not buy these capacities because they have converted to cypherpunk ideology, but because compliance and risk management now require math, not memos. Under Verification, "do nothing" is no longer the conservative option; it is reckless.

Phase II telemetry and triggers: **Phase II is real** once:

- **Fee+burn covers a meaningful slice of security budget.** For at least one serious PoUW/proof network, fees and burns tied to real workloads cover $\geq 30\%$ of miner/prover revenue over 12–24 months.
- **Budgeted workloads dominate speculative ones in volume.** In receipt analytics, the *number* of proofs purchased by enterprises grows steadily, even if speculative volume remains higher in nominal terms.
- **VerifySettle stays inside SLOs through at least one policy shock.** For at least one high-volume corridor, VerifySettle remains within bounds across a visible policy or regulatory event without catastrophic failure.
- **Capex/OPEX lines reference verification capacity explicitly.** At least some institutions treat Work Credits as a line item (verification spend, privacy spend), not as a speculative asset bucket.

At that point, the triad is no longer "crypto" from the allocator's perspective; it is infrastructure they **have** to pay for.

27.1.5 Phase III: Composability-Led (The Stack Disappears into Infrastructure)

Once Phase II gate metrics are met and sustained, composability becomes second nature. Developers no longer think in terms of "ZK project X" or "PoUW chain Y"—they think in terms of the loop.

Provenance, verified compute, and private settlement start to resemble TLS on the web:

- New applications default to emitting claims and receiving receipts via the SDK, simply because it is easier than hand-rolling trust.
- Major frameworks and toolchains ship with verification modules and privacy adapters bundled: verifying a receipt feels as ordinary as opening an HTTPS socket.
- Chains and rollups routinely outsource heavy computation to PoUW or proof factories and anchor receipts on shared ledgers.

From the outside, nothing dramatic happens. There is no "flippening." What changes is the **default**:

- Sensitive data is processed either under enclaves with open silicon profiles or under ZK, and accompanied by attestations.

- Payments that cross borders or touch politically exposed persons quietly route over privacy rails, with receipts that satisfy auditors but not censors.
- AI systems whose outputs matter are either run with PoL-style verifiable inference or backed by helmets of audits and shadow runs, with failure rates visible rather than buried.

Networks that can demonstrate a clear, measurable link between work and value, paired with credible scarcity and decentralization telemetry, begin to trade at a **durable store-of-value premium** alongside Bitcoin.

Phase III telemetry and triggers: **Phase III is real** once:

- **Non-trivial collateral and reserve usage.** Work Credits appear as collateral and reserves in $\geq$ 3 independent venues: lending markets, derivatives, structured products, custody mandates. A visible share of supply is locked in long-duration roles.
- **Macro sensitivity flips.** Price and flows react as much to changes in workload budgets (AI capex, compliance requirements, settlement volume) as to crypto-native news.
- **VerifyPrice and VerifySettle stability through macro stress.** During interest-rate shocks or liquidity crunches, the underlying utility, capacity, and safety metrics do not collapse.
- **Duration-neutral holding behavior.** Holder data shows a meaningful cohort who hold primarily for access to future capacity and SoV properties, with holding periods measured in years.
- **Fee coverage** $\geq 50\%$**.** Sustained over multiple quarters.

When these show up in the dashboards and market structure, the triad has joined the shortlist of things serious allocators treat as monetary base, not just tech.

27.1.6 Phase IV: Policy-Led and Path-Dependent (Verification as Public Good)

Once Phase III gate metrics are sustained through macro stress, the curve becomes more path-dependent and political. If the stack delivers on its risk promises (decent decentralization, cheap verification, robust privacy corridors that are visibly abuse-resistant), states and institutions will begin to treat parts of it as **public goods** rather than threats.

Some jurisdictions will move first. They will:

- Codify cryptographic receipts as acceptable evidence in courts and regulatory filings.
- Mandate provenance proofs for certain classes of media or AI systems rather than ad hoc labelling.
- Recognize privacy rails with viewing-key regimes as compliant infrastructure rather than as dark pools.

Others will resist, preferring the comfort of chokepoints and legacy gatekeepers. That is where the anti-repression design matters. If privacy rails remain non-custodial and censorship-resistant, if proof and compute markets remain globally addressable, capital and talent can route around laggard jurisdictions the way data routed around telcos.

In the best case, the monetary role of the triad becomes self-reinforcing:

- Each wave of repression (negative real yields, capital controls, information crackdowns) pushes more savings and more workflows onto rails where verification is cheap and permission is irrelevant.
- Each wave of adoption increases the depth and liquidity of proof and compute markets, which in turn makes it cheaper and more obvious to use them for new domains.

The triad earns a store-of-value premium not because everyone suddenly shares a philosophical vision, but because the world has learned, through trial and error, that capacities which are verifiable, censorship-resistant, and continually demanded are safer long-term anchors than promises that can be administered or revoked.

Phase IV telemetry and triggers: **Phase IV is real** once:

- **Legal codification of receipts and corridors.** Multiple jurisdictions explicitly recognize PIDL-style receipts as valid evidence in regulation and courts; some codify acceptable VerifySettle or provenance requirements into statute.
- **Regulatory reliance on cryptographic proofs.** Policies and enforcement actions reference proof primitives (ZK proofs, PoL audits, receipt formats) rather than merely "records" or "reports."
- **Public-sector participation.** Public institutions (development banks, treasuries, public broadcasters) use triad-aligned rails and publish their own receipt and corridor metrics.
- **Geographical differentiation.** A measurable share of volume and capital migrates toward jurisdictions and networks that respect lawful privacy and verifiable compute, as seen in regional splits in VerifyReach, settlement volume, and Work Credit usage.

This adoption curve is not guaranteed. Verification cost can creep; hardware or router oligopolies can re-centralize control; regulation can pinch on- and off-ramps harder than expected. But if we anchor the stack in verifiable machines, keep verification cheap and public, treat privacy as infrastructure rather than vice, and force our own claims through telemetry and receipts, this is the curve we can plausibly aim at.

27.2 Failure Gates

Phases can regress. If Phase II metrics collapse, the system slides back to Phase I.

Condition	Consequence
VerifyPrice p95 exceeds SLO for ≥ 3 months	Phase regression; SoV thesis weakens
Corridor refund_safe $< 100\%$ (repeated)	Phase regression; Private Money claim fails
Verification concentrates $> 70\%$ on single profile	Decentralization claim fails
Receipt data becomes unverifiable	Telemetry capture; thesis unverifiable
Fee coverage collapses; $> 80\%$ speculative	Phase II $\rightarrow$ I regression

Table 27.3: Failure gates and consequences.

Chapter 28

Risk Analysis & Failure Modes

The risks to this emerging triad are not merely technical; they are structural, political, and economic.

28.1 Technical Risks

Proof system failures. A SNARK/STARK is broken or exploited. Mitigation: multi-ZK support; tagging proofs with system/parameter IDs; migration paths.
Hardware capture. A major fab or TEE platform has a backdoor. Mitigation: Layer-0 profiles, lot sampling, open hardware alternatives.
Protocol bugs. Consensus bugs, bridge flaws, privacy leaks. Mitigation: reference implementations, staged rollouts, incident response.

28.2 Economic Risks

Verification cost creep. If verifying proofs ceases to be much cheaper than producing them, the entire asymmetry collapses. Shows up as rising $r(W)$ in VerifyPrice.
Centralization. When specialized hardware or closed routers dominate, neutrality erodes. Shows up in concentrated facility IDs, rising entry latency.
Fee death-spirals. If demand falls, fees collapse, security budgets shrink. Mitigation: conservative base issuance, Work Credits encoding long-term demand.

28.3 Political Risks

Bans & sanctions. Jurisdictions may declare privacy assets illegal, sanction contracts, or criminalize the use of certain clients. Mitigation: jurisdictional diversity (labs, foundations, hardware profiles, LPs); client modes that degrade gracefully (offline, mesh, sat-links); legal defense resources; clear separation between core protocol and specific front-ends so that UI bans do not equal protocol death.
Info-ops and framing. Media and governments can frame the stack as "criminal tech" or "na-

tional security threat." Mitigation: lawful-privacy narratives grounded in receipts and compliance primitives; visible legitimate use cases (payroll, provenance, AI verification, critical-infrastructure audit trails); and an insistence on evidence ("show the receipts") rather than slogans.

Regulatory chokepoints at the edge. App stores, banks, and ISPs can be pressured to block access even when the protocol is neutral. Mitigation: the Layer-1/2 work described earlier (alternate transports, side-loading, content-addressed and offline distribution) so that no single storefront, bank, or carrier can become a kill switch.

Political risk cannot be "engineered away." It must be **distributed and prepared for**: diversified jurisdictions, many independent implementations, multiple access paths, and a culture that expects (and drills) for attempted bans and smear campaigns rather than treating them as unthinkable.

28.4 Spec Drift, Vaporware, and Scoreboard Capture

Spec drift and vaporware can hollow out trust long before censorship does. Whitepapers without shipped code or inflated performance claims corrode the ecosystem's credibility. In metrics, this shows up as:

- a widening gap between claimed and measured $\textsc{VerifyPrice}(W)$;
- a mismatch between Work Credit issuance and actual workload utilization;
- chains with high market caps and low receipt volume;
- clients that are nominally "live" but barely process real workloads.

Active liveness tracking (clients, explorers, throughput under real load, settlement safety metrics) should replace social metrics or TVL as the industry's scoreboard. Projects that publicly document testnet-to-mainnet milestones and publish SLOs and incident reports exemplify the discipline needed.

No dashboards, no trust. If $\textsc{VerifyPrice}$, swap success and refund safety, decentralization telemetry, and corridor health are not public, treat claims as unpriced risk. A system that refuses to show its receipts is asking you to underwrite the very soft guarantees this thesis is designed to escape.

28.5 Red Lines: When the SoV Thesis Fails

The thesis is falsifiable. If any of these conditions persist without credible remediation, the store-of-value claim is **no longer defensible**:

Red Line 1: Verification Affordability Breaks

Condition: Physical VerifyPrice p95 exceeds SLO bounds for core workloads for ≥ 3 consecutive months, with no credible remediation path.
Why it kills the thesis: "Anyone can verify" is the hinge. If verification becomes expensive, proofs become platform claims, not public facts.

Red Line 2: Refund Safety Breach

Condition: Corridor refund_safe $< 100\%$ on any admissible route, with repeated incidents and no automatic delist + remediation.
Why it kills the thesis: Non-custodial settlement is the "Private Money" foundation.

Red Line 3: Verification Monoculture

Condition: $> 70\%$ of verifications on a single hardware profile, TEE vendor, or jurisdiction for ≥ 3 consecutive months.
Why it kills the thesis: Monoculture means "trust the dominant vendor."

Red Line 4: Telemetry Capture

Condition: Receipt datasets become unavailable, unverifiable, or controlled by a single party.
Why it kills the thesis: If telemetry can be captured, the entire observability regime is theater.

Red Line 5: Fee Coverage Collapse

Condition: Fee+burn coverage falls below 10% of security budget and workload mix becomes $> 80\%$ speculative for ≥ 12 months.
Why it kills the thesis: The SoV story requires structural demand, not pure narrative.

Pattern: The red line is not "bad thing happens once." It is **"sustained breach + no recovery."** Isolated incidents with rapid remediation are expected in any complex system. Persistent degradation without response is thesis failure.

What happens at a red line: These are not punishments—they are **truth in advertising**. An asset that claims SoV properties must demonstrate them. If it can't, it should stop claiming.

Condition	Response
Red line breached	Incident declared; governance must publish remediation plan within 14 days
Remediation fails after 90 days	Asset reclassified from “SoV candidate” to “speculative/-experimental” in protocol communications
Multiple red lines breached simultaneously	Crisis mode; independent review commissioned; results published

Table 28.1: Red line response protocol.

Chapter 29

Implementation Sketches for Builders

Everything above is the blueprint. This section is about what to actually ship first.

29.1 Minimum Viable Stack (MVS) Sequencing

Builders will ask: "What is the minimal sequence?" The answer matters because dependencies are real.

Priority	Component	What It Enables	Dependency
1	PIDL + verifier harness + anchoring	Telemetry cannot be faked	Foundation
2	One corridor kernel + VerifySettle	Non-custodial settlement	PIDL
3	One proof factory + VerifyPrice	Proof commodity market	PIDL
4	PAL SDK + PRK SDK	Developer surfaces	Corridor + factory
5	Neutral router + fairness tests	Open participation	SDKs + dashboards
6	PoUW pilots	Useful-work security	All above stable

Table 29.1: Minimum Viable Stack sequencing.

Why this order:

1. **Receipts first:** Without verifiable receipts, all claims are unpriced. Ship PIDL, reference verifiers, and dataset anchoring before anything else. This is the foundation that makes "no dashboards, no trust" real.
2. **One corridor before many:** A single refund-safe, non-custodial corridor with public VerifySettle is worth more than ten "coming soon" announcements. Ship BTC↔ZEC or BTC↔XMR with 100% refund safety before expanding.
3. **One proof factory before PoUW:** Prove that canonical workloads can be produced, verified, and priced before adding consensus complexity. PROOF_2^{20} and MATMUL_4096 are good starting points.
4. **SDKs after primitives:** PAL and PRK are developer surfaces, not primitives. Ship them once the underlying corridors and proof factories are stable.
5. **Neutral routing after basic markets:** Fairness tests and admission metrics matter only once there's enough activity to measure. Don't over-engineer routing before you have provers to

route.

6. **PoUW last:** Proof-of-useful-work consensus is the most complex piece. It requires stable proof factories, receipts, and measurement infrastructure. Don't attempt it until Priorities 1–5 are solid.

Hardware profile clarity: "Commodity GPUs with well-understood drivers" is an "open-ish hardware profile." To be explicit: this is **Profile Class: Partially Closed, High Familiarity**. GPUs are not verifiable machines in the same sense as open RTL—they are "known black boxes" whose behavior is well-characterized but not inspectable. Label profiles clearly so users understand the trust assumptions.

29.2 Layer-0 Verifiable Machines

Start with small, fully open "security evaluation" chips: key-storage/signing cores, simple TRNG/PUF arrays. For each tape-out, commit to a fixed sampling plan.

Minimal Layer-0/4 pattern:

- Pick one open-ish hardware profile (e.g., commodity GPUs).
- Define a hardware profile HID with chip family, driver versions, entropy tests, power metering.
- Stand up a proof factory cluster with reference prover binaries.
- Publish basic telemetry and VerifyPrice samples.

29.3 Privacy Rails: Making Settlement Safe and Boring

Standardize adaptor-signature swap flows. Take BTC↔XMR and BTC↔ZEC as canonical corridors. Write down message formats, time-lock conventions, refund procedures.

Ship wallet UX with recovery, not bravado. Clear progress indicators, explicit timeouts, visible "abort and refund" buttons.

Document settlement latencies and success rates. Publish historical p50/p95 time-to-finality, success rates, and failure breakdowns.

29.4 Proof Factories: Receipts as a Service

The next implementation target is the Prove stage: turning raw work into portable receipts that any chain, app, or institution can consume.

Unify proof outputs under a receipt schema. Start by defining a compact, chain-agnostic receipt format (PIDL): the claim hash, workload ID, circuit or model hash, proof hash or tran-

script commitments, SLA tier, start/end timestamps, resource usage, and provider signatures. Whether the underlying artifact is a MatMul transcript, a logits fingerprint, or a SNARK, all of them fit into this envelope.

Expose a single programming surface. For developers, the SDK should offer one set of verbs: `declare_claim`, `request_proof`, `await_receipt`, `verify_receipt`. Underneath, it can target multiple proof systems and networks. The developer doesn't wire to "Platform X"; they attach their claim and policy to the SDK and let it find anyone who can satisfy the SLA.

Integrate multiple zkVMs and proof systems from day one. Monoculture is the enemy of both security and economics. The factory should be multi-ZK and multi-backend by construction: support at least one pairing-based SNARK, one STARK or FRI-based system, and one zkVM at launch, with a clear path to adding more.

Make telemetry a first-class product. Every proof request and receipt should emit metrics into the VerifyPrice Observatory: p50/p95 verifier times and costs for each workload, failure modes, queue depths per backend, variance under adversarial mixes.

29.5 Compute Consensus Pilots

With privacy rails and proof factories in place, you can begin to anchor consensus itself in useful work. Think in terms of reference pilots, not one chain to rule them all.

MatMul PoUW pilot (Duplex-style): Launch a devnet where the block-making puzzle is a MatMul instance drawn from a canonical distribution. Use the workload registry/WF-ABI to describe tasks; derive instances from header randomness; implement low-rank noise so that miners can return both a correct product and a succinct proof. Wire block validity to the presence of a valid MatMul receipt whose VerifyPrice parameters are below published thresholds. Instrument everything: $r(W)$ ratios, verifier times on consumer hardware, time-to-first-proof for new miners, and centralization metrics.

Verified-inference pilot (Ambient-style PoL): In parallel, build a network where the work function is "serve model inferences with proofs of honesty." Commit to models and datasets; design logits-fingerprint schemes that are simple, deterministic, and uniform; add randomized audits and peer-prediction-based slashing. Here full ZKML may be too expensive, so hybrid verification is acceptable, but only if it is exposed honestly: publish honest-output rates, audit coverage, and failure patterns as SLOs, not marketing.

zk-Proof-of-Work / receipt ledger pilot (Nockchain-style): Finally, either integrate with or prototype a chain whose primary job is to produce and timestamp zk proofs themselves. This

network can serve as a common receipt ledger: proofs from many domains are anchored here with minimal metadata and availability guarantees.

These pilots need not be over-promised as final destinations. Their job in the early years is to act as laboratories: to establish that verification asymmetry holds in practice at scale, that decentralization can be preserved under useful-work mining, and that receipts remain cheap to check.

29.6 Developer Kit: Make "Import Proofs" the Default

The last implementation sketch is about developer experience. If every application needs a team of cryptographers to participate, nothing scales.

Claims & proving library: A library that plugs into common languages and frameworks and exposes a simple API: mark this function as "must be proven," mark this data stream as "must be provenance-tracked," specify acceptable backends and SLAs, and let PAL handle task generation and proof requests.

Settlement adapter: A component that connects to existing wallets and key-management setups, and that exposes "pay-for-proof" and "pay-for-compute" as simple intents: no manual swap logic, no bespoke bridges, just a predictable interface plus receipts, backed underneath by PRK and the atomic-payout kernel.

Verification module: Modules that can be compiled into smart contracts, browser bundles, and command-line tools, so that end-users and auditors can check receipts without standing up new infrastructure. `verify(receipt)` should be as embedded in the tooling as `log()` is today.

The work of the next few years is, in large part, to make these three pieces boring. When "import proofs" feels like "import TLS," and when "settle privately" feels like "call the payments API," the cypherpunk monetary stack stops being an argument and starts being the default way serious systems are built.

Chapter 30

Objections & Responses

Any thesis that ends with "this becomes money" deserves hard pushback.

30.1 "This is just another chain / coin."

Objection: We've seen this movie: every new system claims to be hard money with a fancy narrative. This is just buzzwords on top of a token.

Response:

- The core claim is not "this token pumps," but "Privacy, Proofs, and Compute are verifiable necessities that can be priced and audited."
- The focal metrics are **VerifyPrice/Reach/Settle and triad usage**, not just TVL or number of wallets.
- Work Credits are minted against **measurable work**: proofs, private settlements, verified FLOPs, with energy and hardware profiles tied in.

If, in practice, the system behaves like "just another chain" (no canonical workloads, no PoUW, no privacy corridors, no telemetry), then the objection is correct. The point of the architecture is to make that failure **obvious**, not to hide it.

30.2 "Users don't care about privacy or proofs."

Objection: People trade convenience for privacy all the time; most don't verify anything. Why build a SoV thesis on properties most users won't touch?

Response:

- Users may not **ask** for privacy or proofs, but they **suffer** when they're absent: identity theft, surveillance, misinformation, censorship, frozen funds.
- The stack is not asking end-users to run verifiers or design circuits; it's asking:
  - infra operators to run verifiers,
  - institutions to demand receipts, and
  - developers to call PAL/PRK instead of opaque APIs.
- The "user" that cares most may be a treasury, DAO, insurer, or regulator—actors who must

explain themselves.

If, after a decade of increasing repression and AI-mediated reality, nobody is willing to pay for privacy, proofs, or verified compute, then the triad doesn't become money. The thesis is that the opposite is more likely.

30.3 "PoUW will centralize / can't compete with hyperscalers."

Objection: Useful-work mining sounds good, but hyperscalers and incumbents will always dominate; you'll just rebuild cloud oligopolies inside a blockchain.

Response:

- Hyperscalers already dominate raw compute; the point of PoUW is not to out-compete them on price, but to:
  - turn **verified units of useful work** into a commodity;
  - ensure **anyone** can verify;
  - keep entry for provers open at the margin.
- Layer-0/4/6 design fights centralization by diversifying hardware profiles; measuring and publishing prover concentration; using rewards and Work Credit policies to favor diversity; allowing hyperscalers to participate, but not to be **the only ones**.

If the market decides that centralized compute is always "good enough" and verifiability never matters, then AI Money stays a nice phrase. The thesis is that high-stakes actors—finance, defense, safety-critical infra—will demand **verifiable** compute from multiple vendors.

30.4 "Governments will never allow lawful privacy at scale."

Objection: Sovereigns will not tolerate unbreakable privacy and bearer-like digital money; they will regulate and block until these systems are marginal.

Response:

- Some governments will indeed oppose; others will see advantages in verifiable, receipt-backed compliance; lower settlement risk; reduced dependence on foreign platforms and currencies.
- The architecture is about **resilience**, not legal victory: it assumes partial repression; spreads hardware, comms, and governance across jurisdictions; keeps protocol-level custody and identity neutral.
- Lawful privacy is engineered, not begged for: viewing keys and receipts allow compliance without re-centralizing custody.
- **Critically:** The system designs *neutral infrastructure with consented disclosure*, not "dark finance." Policy compliance is predicate-based (ZK proofs of allowlist membership, authorization, jurisdiction), not graph-inspection.

If all major jurisdictions converge on banning any form of non-custodial digital value, a lot more breaks than this stack. The design goal is: **if even a few open jurisdictions remain, and some gray-market paths exist, can the triad continue to function?**

30.5 "Receipts will become surveillance tools."

Objection: All these receipts and proofs will just become a new surveillance layer. Governments will demand access; the system will comply; privacy dies.

Response:

- Receipts are **privacy-preserving by design**: they prove predicates (membership, compliance, authorization), not identities. A receipt proves "sender is in cleared set" without revealing which entry.
- Disclosure is **consented and scoped**: viewing keys reveal specific flows to specific auditors for specific time windows, not universal access. There is no "master key."
- **Constitutional constraint:** "Policy = predicates, not graph inspection" is a hard constraint. Any policy requiring universal tracing is treated as incompatible with the monetary design.
- The system **explicitly rejects** policies that require global traceability. If a regulator demands "show me all transactions," the answer is: "We can prove compliance with specific predicates; we cannot provide a surveillance feed, because the architecture doesn't support it."

If receipts become surveillance tools, the design has failed. The telemetry should make this visible: if most flows require real-name disclosure, the Settlement & Privacy Board will show it.

30.6 "They'll just shut off the internet / app stores."

Objection: Sovereigns can simply block the network or remove apps from stores.

Response:

- Total, permanent shutdowns are blunt and politically costly; partial, targeted throttling is more likely. The stack assumes this and designs for it:
  - Multiple obfuscated transports (Layer 1).
  - Content-addressed updates and offline installers (Layer 2).
  - Sideload paths for clients.
- Treat reachability and update-health as SLOs (VerifyReach): if clients in censored regions can still fetch proofs and updates via at least one path, comms resistance is doing its job.

30.7 "This burns too much energy."

Objection: PoUW is just PoW with extra steps; it still wastes energy.

Response:

- All monetary substrates consume scarce resources—geology, enforcement, balance-sheet capacity, or energy. The point of PoUW is not to justify waste; it is to redirect energy into **useful work** whose receipts the world must keep buying.
- The right question is energy per verified FLOP or per proof unit, and whether that trend is improving. If energy-per-receipt falls while verified-capacity-per-token rises, the system is doing strictly better than random hashing per unit of trust delivered.
- **Measurable answer:** Facility Energy Receipts (FERs) and "work per kWh" metrics make energy usage auditable. Energy efficiency shows up in VerifyPrice (cost component) and in Layer-0 telemetry.

30.8 "Governance will just re-centralize."

Objection: The stack is too complex; whoever runs it will become the new chokepoint.

Response:

- Complexity does not have to mean opaqueness. The governance layer is deliberately thin: decisions are about parameters and SLOs, not about picking winners.
- Multi-jurisdictional foundations, transparent config changes, and slashing rules keyed to public telemetry (VerifyPrice, decentralization stats, corridor health) make governance legible.
- If a network cannot show **who changed what, when, and in response to which metrics**, it is not an SoV candidate—however elegant its whitepaper.

Chapter 31

Why These Become Money

It is fashionable to say that money is a shared hallucination. That line flatters our cleverness while excusing our passivity. Hallucinations cannot settle debts across adversarial jurisdictions; hallucinations do not finance supply chains. Money works because it is backed by a machine—sometimes a literal machine of war, sometimes a machine of law, and now verification machines that continuously produce scarce, indispensable utilities.

Post-Bretton-Woods, the machine was compliance. The next machine is verification. Cypherpunks did not abolish trust; they automated it. When privacy, proofs, and compute clear across neutral rails, money stops asking for favors and starts paying for facts.

We can restate the thesis as a conditional:

> *If* a dense digital civilization continues to rely on AI, global networks, and programmable markets, *and if* states continue to use repression and narrative control rather than explicit default and humility, *then* capacities that deliver Privacy, Proofs, and Compute cheaply and verifiably will behave like money.

31.1 As a Store of Value

From the SoV lens, a credible store of value must be credibly scarce, cheap to verify in public, resist censorship and capture, have native demand that is not purely narrative, and avoid being a duration instrument whose real return can be pinned negative by policy.

Privacy, Proofs, and Compute meet that brief, but in a way very different from gold or Bitcoin:

- **Privacy** is purchased because some people and institutions must pay without broadcasting their graph: dissidents, NGOs, treasuries under capital controls, enterprises with sensitive payroll and vendor relationships. In a repression-heavy world, privacy is not a luxury good; it is the hull that keeps savings from becoming an option owned by someone else.
- **Proofs** are purchased because "seeing is believing" has failed. Deepfakes, platform curation, and the liar's dividend make any unproven artifact suspect. Regulated AI and finance regimes require auditable provenance and computation. In that world, proofs are not a niche; they are the affidavit layer of the digital order.

- **Compute** is purchased because intelligence is now a first-class input to production. FLOPs for training, inference, and proving are line items in budgets. Once those FLOPs are wrapped in proofs and standardized as canonical workloads, verified compute becomes a commodity that can be priced, hedged, and stored.

In a world that will likely choose **stealth default**—negative real yields aided by regulation—over explicit default, a durable SoV must be duration-neutral, peg-proof, and paid for by recurring, indispensable utility. Privacy, proofs, and compute meet that brief: issuance that can't be decreed; revenues that reprice with fiat budgets; verification that stays cheap and public. **Bonds can be anesthetized; utility cannot.**

31.2 As Stack

Seen from the stack angle, the triad is backed not by a metaphor but by a supply chain:

- Layer 0 keeps **machines** honest and powered (verifiable hardware, FERs).
- Layer 1 keeps **packets** flowing under censorship (VerifyReach).
- Layer 2 keeps **code** moving even when app stores and CDNs are hostile.
- Layer 3 keeps **identity** accountable without doxxing.
- Layers 4–5 make **work** and **value** flow through proofs and privacy rails.
- Layer 6 keeps **governance** and **telemetry** legible.

Triad instruments are wrappers around claims on this supply chain. When you hold Work Credits, you hold future access to this stack.

31.3 As Telemetry

VerifyPrice, VerifyReach, VerifySettle, and decentralization metrics are the constitution. They keep "trustlessness" from decaying into "trust the custodians."

A system that cannot show its own health cannot be money; it can only be marketing.

31.4 Trading One Base Reality for Another

What changes and what stays the same?

What changes: The base reality that backs money. Instead of "the sovereign will repay" or "gold is scarce because geology," the base reality becomes "these capacities are scarce, necessary, and verifiable."

What stays the same: Money is still a claim on work. The work just becomes specific and measurable: proofs that anyone can check, privacy that anyone can use, compute that anyone can verify.

As stack: The triad is backed not by a metaphor but by a supply chain: Layer 0 keeps machines honest; Layer 1 keeps packets flowing; Layers 2–6 complete the pipeline. Triad instruments are wrappers around claims on this supply chain.

As telemetry: VerifyPrice, VerifyReach, VerifySettle, and decentralization metrics are the constitution. They keep "trustlessness" from decaying into "trust the custodians."

Chapter 32

Conclusion: A Bell Labs for Privacy, Proofs, and Compute

The original Bell Labs turned information theory into cables, switches, and semiconductors that quietly reshaped the world. Nobody needed to know the math; they just made phone calls.

The mandate here is similar, but for a different substrate:

> **Turn privacy primitives, zero-knowledge, and verifiable compute into everyday infrastructure—receipts, rails, and verifiable machines—such that Privacy, Proofs, and Compute behave like a next-generation store of value.**

That mandate breaks into three concrete programs:

1. **Monetary program.** Design and issue Work Credits so that claims on triad capacity are credibly scarce, easy to verify, hard to seize, and worth holding through repression cycles.
2. **Stack program.** Build and harden Layers 0–6: verifiable machines, resilient comms, pseudonymous identity, PoUW and proof factories, privacy corridors, governance and telemetry.
3. **Telemetry program.** Keep VerifyPrice, VerifyReach, VerifySettle, and decentralization metrics live, public, and blunt. Treat dashboards as constitution.

32.1 The Quiet Mic Drop

If it does work, the moment of success will not look like an ICO or a conquest. The mic drop is quiet.

There will just be a decade in which:

- auditors ask for **receipts**, not screenshots;
- regulators accept **cryptographic attestations** as first-class evidence;
- enterprises budget for **proofs and privacy** the way they once budgeted for bandwidth;
- wallets and apps call `verify(receipt)` as casually as they call `https://`;
- and allocators treat triad assets not as exotic bets, but as part of the boring hedge against repression.

At that point, the thesis will have stopped being a thesis. It will have collapsed into plumbing.

32.2 Final Thoughts

A decade ago, Bitcoin taught us that **digital scarcity alone** can be money. The next epoch adds **native utility that the world must keep buying**, secured by proofs anyone can check.

Privacy will preserve agency.

Proofs will anchor truth in a synthetic world.

Compute will power intelligence—verifiable and honest, not merely performed.

When these three clear across neutral settlement, money stops being an article of faith and becomes what it always wanted to be: **a record of work that cannot be faked and does not need permission to move**.

In the decade ahead we will learn that digital necessities, wrapped in proofs anyone can verify, are better money still.

Sources

Theory & Framing

Davidson, J. D., & Rees-Mogg, W. (1997). *The Sovereign Individual: How to Survive and Thrive During the Collapse of the Welfare State*. New York: Simon & Schuster / Touchstone.

McLuhan, M. (1964). *Understanding Media: The Extensions of Man*. New York: McGraw-Hill.

Macro: Debt, Financial Repression, Real Yields

Gaspar, V., Gonçalves, C. E., & Poplawski-Ribeiro, M. (2025, September 17). Global Debt Remains Above 235% of World GDP. *IMF Blog / Global Debt Monitor*. International Monetary Fund.

International Monetary Fund. (2025). *Global Debt Monitor 2025*. IMF Global Debt Database.

Reinhart, C. M., & Sbrancia, M. B. (2015). The Liquidation of Government Debt. *IMF Working Paper 15/7*. International Monetary Fund.

Rose, J. (2021). Yield Curve Control in the United States, 1942 to 1951. *Economic Perspectives*, Federal Reserve Bank of Chicago, 45(2).

Social / Disinformation / Deepfakes / Provenance

"The Twitter Files." (overview). Twitter Files entry, Wikipedia (summarizing 2022–2023 document releases and reporting).

U.S. House Committee on the Judiciary (Republican Staff). (2023). *The Weaponization of CISA: How a "Cybersecurity" Agency Colluded with Big Tech and "Disinformation" Partners to Censor Americans*. Staff Report.

U.S. Department of State. (2021–2025). Global Engagement Center (GEC): About Us. U.S. Department of State.

Harwell, D., & Patton, D. (2025, October 22). We uploaded a fake video to 8 social apps. Only one told users it wasn't real. *The Washington Post*.

UNESCO. (2025). *Deepfakes and the Crisis of Knowing*. UNESCO briefing.

Ahmed, S., et al. (2024). Social Media News Use Amplifies the Illusory Truth Effects of Deepfakes. *Journalism & Mass Communication Quarterly*.

Policy Clock: EU AI Act

European Union. (2024). *Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act)*. Official Journal of the European Union.

Verification Asymmetry & PoUW (MatMul)

Komargodski, I., Lin, S., & Weinstein, O. (2025). Proofs of Useful Work from Arbitrary Matrix Multiplication. *arXiv:2504.09971*.

ZKML / Verifiable ML

Peng, Z., Wang, T., Zhao, C., et al. (2025). A Survey of Zero-Knowledge Proof Based Verifiable Machine Learning. *arXiv:2502.18535*.

FHE / FHE Rollups / OpenFHE / Fhenix

Zyskind, G., Erez, Y., Langer, T., Grossman, I., & Bondarevsky, L. (2024). FHE-Rollups: Scaling Confidential Smart Contracts on Ethereum and Beyond. *Proceedings of BSCI '24*. ACM.

OpenFHE Project. (2025). OpenFHE v1.4.0 – Development Release Announcement. GitHub.

Fhenix. (2023). Introducing FHE-Rollups: Scaling Confidential Smart Contracts on Ethereum and Beyond. Fhenix blog.

Bitcoin Transport Privacy (BIP-324, Core 27.0)

Bitcoin Core Developers. (2024, April 16). Bitcoin Core Version 27.0 Released. BitcoinCore.org.

Mehta, D., Ruffing, T., Schnelli, J., & Wuille, P. (2023). BIP-324: Version 2 P2P Encrypted Transport Protocol. Bitcoin Improvement Proposals.

Lightning Receiver Privacy (BOLT12 Offers)

No Bullshit Bitcoin. (2024, July 4). Phoenix Wallet v2.3.1 & Phoenixd v0.2.0: BOLT12 Offers. NoBSBitcoin.com.

No Bullshit Bitcoin. (2024, September 24). BOLT12: Offers Officially Merged into Lightning Specification.

Ecash / Federations (Fedimint / Fedi)

Bitcoin Magazine. (2025, October 29). From Stealth to Scale: Fedi Unveils Multi-Sig Guardians for Federated Bitcoin Ecash Mints.

Fedimint Project. (2024–2025). What Does the Federation Do? Fedimint documentation.

Privacy Rails: Zcash, BTC↔XMR Swaps

Electric Coin Co. / Zcash Foundation. (2024). *Zcash Protocol Specification – NU6 Activation and Second Halving*. Zcash Protocol Specification, Rev. 2025.6.0-105.

Reitwießner, C. et al. (COMIT Network). (2020). Monero–Bitcoin Atomic Swaps. COMIT blog and GitHub.

UnstoppableSwap / eigenwallet. (2025). UnstoppableSwap GUI / eigenwallet: Non-custodial XMR↔BTC Atomic Swaps. GitHub.

Zcash Roadmap / NU7 Discussions

Zcash Community Forum. (2024–2025). Accelerate NU7: Let's Deliver a Smaller Network Upgrade Sooner. Governance thread.

Ethereum Privacy L2 (Aztec)

Steinrueck, D., & Sammara, A. (2025, May 2). Testnet Retro – Launch Day: Aztec Public Testnet Went Live. Aztec Network blog.

Coindesk. (2025, May 1). "Everything Is Encrypted": Aztec's Privacy Rollup Hits Testnet Amid Growing Demand.

IBC-Wide Shielding (Namada)

Namada Team. (2025, June 20). Namada Mainnet Launch Is Complete! Namada blog.

Namada Team. (2023, January 31). Understanding the MASP and CC Circuits. Namada blog.

Proof Markets / PoUW / Privacy-First L1s

Aleo. (2025). Consensus and Proof of Succinct Work (PoSW). Aleo developer documentation.

RISC Zero / Boundless. (2025). Boundless: A Universal ZK Compute Layer. Binance Research.

Succinct Labs. (2025, August 5). Succinct Prover Network Mainnet Launch (PROVE).

Nockchain Foundation. (2025). Nockchain Is Now Open Source; Nockchain Issuance Schedule and Dumbnet Launch. Nockchain.org.

Psy Protocol (formerly QED). (2025). Psy: A Scalable, Privacy-First L1 Secured by Proof-of-Useful-Work 2.0.

Bittensor. (2025). Yuma Consensus. Bittensor Docs.

Ambient. (2024). Ambient Litepaper v1: Proof of Logits and Consensus.

Transport, Privacy, SDKs & Related Infrastructure

No Bullshit Bitcoin. (2024, July 11). The New Phoenix: A 3rd Generation Lightning Wallet.

Fedimint. (2025). Fedimint Lightning Gateway Uses LDK Node to Simplify Deployment and Liquidity Management. Lightning Dev Kit blog.

Ethereum Privacy Roadmaps (PSE / Privacy Cluster)

Privacy Stewards of Ethereum (PSE). (2025, September 12). PSE Roadmap: 2025 and Beyond. PSE.dev blog.

Ethereum Foundation. (2025, October 8). The Ethereum Foundation's Commitment to Privacy. Ethereum.org blog.

CoinMarketCap Academy. (2025, October 9). Ethereum Foundation Launches Privacy Cluster for Enhanced Blockchain Security.

Vitalik on Openness & Verifiable Hardware

Buterin, V. (2025, September 24). The Importance of Full-Stack Openness and Verifiability. Personal blog (vitalik.eth.limo).

C2PA / Provenance (Supporting)

Creative Commons / Coalition for Content Provenance and Authenticity (C2PA). Content Credentials Standard Documentation.

Formal Model of Verification Asymmetry & VerifyPrice

.1 Definition 1: VerifyPrice(W) and Verification Overhead

For each canonical workload W (for example, a constraint system $\text{PROOF}_{2^{20}}$, a matrix multiplication size MATMUL_4096, or a verified-inference task INFER_LM_70B_256TOK), we care about two things:

1. How expensive it is to verify a purported result on a reference verifier, and
2. How that verification cost compares to producing the result from scratch.

VerifyPrice(W): verification profile Fix a reference verifier and adversarial mix of inputs. From a sample of verification runs for workload W, we estimate:

- $p_{50,t}(W), p_{95,t}(W)$: median and 95th-percentile wall-clock verify time from "artifact received" to "accept/reject";
- $p_{50,c}(W), p_{95,c}(W)$: corresponding median and 95th-percentile verify cost in a reference unit (e.g., USD of cloud compute);
- $\text{fail}(W)$: observed fraction of artifacts that fail verification (invalid, malformed, or timed out).

The VerifyPrice vector for W is then:

$$\text{VerifyPrice}(W) \equiv (p_{50,t}(W), p_{95,t}(W), p_{50,c}(W), p_{95,c}(W), \text{fail}(W))$$

ProduceProfile(W): production profile Similarly, we can describe the prover's cost to produce correct outputs for W. From a sample of production runs (or cost models) we estimate median and 95th-percentile produce time and cost:

$$\text{ProduceProfile}(W) \equiv (p^{\text{prod}}_{50,t}(W), p^{\text{prod}}_{95,t}(W), p^{\text{prod}}_{50,c}(W), p^{\text{prod}}_{95,c}(W))$$

Verification overhead $r(W)$ We define separate time- and cost-based overhead ratios:

$$(v/p)^{\text{time}}_W = \frac{p_{50,t}(W)}{p^{\text{prod}}_{50,t}(W)}, \quad (v/p)^{\text{cost}}_W = \frac{p_{50,c}(W)}{p^{\text{prod}}_{50,c}(W)}$$

For convenience, we can summarize these as a single scalar:

$$r(W) \equiv \max\left((v/p)_W^{\text{time}}, (v/p)_W^{\text{cost}}\right)$$

.1.1 Implementation Note: VerifyPrice Telemetry Schema (Sketch)

The exact encoding of VerifyPrice is implementation-dependent, but for operational dashboards and APIs it helps to standardize a minimal JSON-style schema per canonical workload. A skeletal example:

```
{
  "workload_id": "PROOF_2^20",
  "window_start": "2025-06-01T00:00:00Z",
  "window_end":   "2025-06-07T00:00:00Z",

  "p50_verify_time_sec": 0.40,
  "p95_verify_time_sec": 0.90,

  "p50_cost_usd": 0.00030,
  "p95_cost_usd": 0.00055,

  "failure_rate": 0.0004,

  "r_time": 0.18,
  "r_cost": 0.21,
  "r_max": 0.21,

  "sample_size": 125000,
  "hardware_profiles": ["open_tee_v2", "gpu_cluster_Y_v1"]
}
```

This schema is not normative; it is illustrative. The key properties are:

- Each canonical workload (workload_id) has its own series of VerifyPrice snapshots over sliding windows.
- VerifyPrice is expressed both in time and cost percentiles, with the overhead ratio $r(W)$ reported explicitly.
- Enough context is included to interpret shifts (sample size, hardware profile mix, time window), but no user-identifying data is required.

Clients and observatories can compute VerifyPrice from raw receipts and emit them in this or a similar format. The important thing is that:

1. The mapping from receipts → VerifyPrice is public and reproducible, and
2. The resulting metrics are exposed in a way that any third party can verify and alert on.

A system that claims "cheap, public verification" but cannot produce machine-readable VerifyPrice series for its canonical workloads is, by construction, asking you to trust what it cannot or will not measure.

We say that W is **verification-asymmetric** if $r(W) \ll 1$: checking honest work is much cheaper than re-doing it from scratch, both in time and cost.

A network (or facility) satisfies the VerifyPrice SLO for workload W if, for a published target (e.g., $p_{95,t}(W) < 5\text{s}$, $p_{95,c}(W) < 10^{-3}$ USD), it maintains:

- $\text{fail}(W)$ below a small, explicit threshold, and
- $r(W) \leq 0.3$ (with a stretch goal $r(W) \leq 0.1$),

over adversarial mixes of inputs and artifacts.

.2 Definition 2: Facility Energy Receipt (FER)

A Facility Energy Receipt (FER) is a signed, machine-readable summary of a site's energy use and availability over a fixed time interval Δt. Each FER is identified by a unique hash `fer_id` and contains at least:

- **facility_id**: a stable identifier for the site or plant;
- **interval_start, interval_end**: timestamps delimiting the reporting window;
- **energy_in_kwh**: total electrical energy consumed at the site boundary over Δt;
- **energy_it_kwh**: energy delivered to IT loads (provers, routers, storage);
- **heat_reused_kwh**: energy captured and reused (for ERE metrics);
- **pue, ere, wue**: standard efficiency and water-use indicators over Δt;
- **outages**: a list of curtailment or outage events during Δt;
- **carbon_intensity_kg_per_kwh**: average emissions factor for supplied energy;
- **signature**: one or more digital signatures attesting to the report's correctness.

FERs are produced at regular cadence (e.g., every 5 minutes or hour). Other receipts (e.g., PIDL receipts for proofs) can reference a FER via `fer_id`, allowing allocators, auditors, and protocols to reconstruct the energy context in which a unit of verified work was performed.

.3 Definition 3: PIDL Receipt (Proof Interface Definition Language)

A PIDL receipt is the minimal, canonical record of a verified interaction in the Create/Compute → Prove → Settle → Verify loop. It binds together the claim, the evidence, and the operational context into a single, serializable object.

At a minimum, a PIDL receipt contains:

- **receipt_id**: a unique identifier (e.g., a hash of the serialized payload);
- **claim_id**: an identifier or hash of the claim being attested;

- **workload_id**: a canonical tag for the workload type and parameters (e.g., PROOF_2^{20}, MATMUL_4096);
- **proof_ref**: a hash or pointer to the proof or transcript artifact;
- **sla_tier**: the service class under which the work was promised (e.g., Bronze/Silver/Gold);
- **verify_time_sec, verify_cost_unit**: measured time and cost for verification;
- **result**: an outcome flag (e.g., accept, reject, timeout);
- **optional context**: hardware_profile, fer_id, settlement_txid, policy/jurisdiction tags;
- **signatures**: digital signatures from the prover, verifier, and (if applicable) broker or router.

PIDL receipts are designed to be portable and composable: they can be embedded in blocks, stored off-chain with only their hashes committed on-chain, included in higher-level proofs, or presented to auditors and counterparties.

.4 Definition 4: VerifyReach(N, R)

Reachability and path diversity for independent verifiers in network N and region R.

For a given network configuration N, client region R, and a target time budget T (e.g., "within 30 seconds"), VerifyReach summarizes how reliably an honest client can reach at least one honest verifier within T seconds over multiple transport classes.

The VerifyReach KPI for network N in region R is the vector:

$$\text{VerifyReach}(N,R) \equiv (p_{50,\text{conn}}(N,R), p_{95,\text{conn}}(N,R), \text{succ}_1(N,R), \text{succ}_2(N,R), \text{fail}_{\text{mix}}(N,R))$$

Where:

- $p_{50,\text{conn}}, p_{95,\text{conn}}$: median and 95th-percentile time-to-first-connection;
- succ_1: 1-path reachability rate;
- succ_2: multi-path reachability rate (resilience to single-path blocking);
- fail_{mix}: distribution over connection failure causes.

Example SLOs:

- $\text{succ}_1(N,R) \geq 0.98$ and $\text{succ}_2(N,R) \geq 0.90$ for major regions;
- $p_{95,\text{conn}}(N,R)$ under a few seconds or tens of seconds (depending on transport mix).

.5 Definition 5: VerifySettle(C)

Time, reliability, and privacy quality of non-custodial settlement on a corridor C.

For a given settlement corridor C (e.g., BTC↔XMR atomic swaps), VerifySettle(C) summarizes whether non-custodial settlement actually clears under its advertised SLOs.

The VerifySettle KPI is:

$$\text{VerifySettle}(C) \equiv (p_{50,t}(C), p_{95,t}(C), s(C), r_{\text{safe}}(C), f_{\text{mix}}(C), a(C))$$

Where:

- $s(C)$ (success): fraction of settlements that complete as intended within the SLA window;
- $r_{\text{safe}}(C)$ (refund-safe): fraction of failed attempts where all funds return safely;
- $f_{\text{mix}}(C)$ (failure mix): distribution of failure causes;
- $a(C)$ (anonset): summary of anonymity-set size and churn.

A corridor satisfies its VerifySettle SLO if:

- success$(C) \geq$ target (e.g., 0.95), and
- refund_safe$(C) = 1.0$ over adversarial mixes,

with no hidden custodian or discretionary freeze paths.

.6 Definition 6: Repression Wedge

The repression wedge is the realized real return on rate-capped instruments when nominal yields are held below inflation:

$$r_{\text{real},t} \equiv \frac{1 + i_{\text{cap},t}}{1 + \pi_t} - 1 \approx i_{\text{cap},t} - \pi_t$$

A negative $r_{\text{real},t}$ indicates that holders of capped instruments are being taxed in real terms via financial repression rather than explicit default.

.7 Definition 7: Liquidation Effect

The liquidation effect measures the effective "tax" on bondholders from negative real rates (after Reinhart–Sbrancia):

$$\text{Liquidation}_t \approx \max(0, -r_{\text{real},t}) \times \frac{\text{Domestic Gov't Debt}_t}{\text{GDP}_t}$$

Advanced economies saw negative real rates ~1/2 the time, with average savings ~1–5% of GDP/yr during 1945–1980.

Practical KPIs & Telemetry Templates

This appendix provides a practical checklist of key performance indicators and telemetry metrics organized by operational domain. These templates support the VerifyPrice, VerifyReach, and VerifySettle observability regime described in the main text.

Prover markets: Queue depth, clearing price per proof type, cancellation rates, p50/p95 VerifyPrice.

Compute consensus pilots: MatMul throughput (GFLOPs/s), verifier cost ratio, invalid-work rejection rate under adversarial load.

Privacy settlement: Atomic swap success %, time-to-finality, stuck-flow causes, wallet UX friction metrics.

Decentralization & capture risk:

- Nakamoto coeff. (by stake/hash/prover share); top-N concentration; Gini.
- Entry latency: time-to-first-proof/mined-block for a new node.
- Geographic/ASN diversity; % over Tor/VPN; relay/MEV-builder diversity.

Liveness & safety:

- Reorg/orphan rate; effective finality time (p50/p95); incident MTTR/MTBF.
- SLA attainment (% proofs/settlements meeting latency/uptime targets).
- DoS/churn resilience: success % under adversarial load mixes.

Verify/produce economics:

- Verifier cost ratio (verify/produce) by workload; energy per proof/inference.
- Hardware mix & utilization (GPU/CPU/ASIC) vs. throughput; cost/FLOP ($/GFLOP-verified).
- Failure taxonomy: prover faults, invalid proofs, circuit timeouts.

Privacy rails health:

- Anonymity-set size & churn (shielded pool/epoch); linkability regressions found.
- Swap slippage & fees (p50/p95); cross-venue route diversity; K-fail root causes.
- UX friction: successful first-run rate, steps/clicks to complete, abandonment rate.

Bridges & cross-chain settlement:

- Atomicity violations (0-conf leaks, partial completes); retry rate; liquidity depth per route.
- Watchtower/guardian coverage; light-client verification share vs. multisig.

Provenance & policy demand:

- C2PA attach rate, strip rate by platform; "verified views" share.
- Proof volume indexed to policy milestones (EU AI Act dates); % budgeted vs. ad-hoc spend.

Security & abuse:

- Sybil detection hit rate; staking/miner collusion alerts; MEV/censorship events.
- Key/attestation hygiene: validator/prover key rotations, slashing/penalties.

Token & issuance telemetry:

- Realized issuance vs. schedule; fee/burn coverage of security budget; % rewards tied to useful work.
- Velocity vs. locked/collateralized supply; SoV signals (holder age bands, exchange outflows).

Customer success:

- Enterprise proof spend: $/month, SLA breaches, churn; Net Proof Retention (NPR).
- Time-to-integration (SDK→prod); support tickets per 1k proofs/settlements.

Hardware honesty:

- Share of receipts tied to open or sampled hardware profiles (by workload and volume).
- Lot-sampling coverage: percentage of lots and devices subject to destructive audits (SEM/optical imaging, side-channel tests), and maximum time since the last audit per profile.
- Incidents and deprecations: count and severity of profile-level compromises, time-to-deprecation, and migration progress away from affected profiles.

The SDK (Proofs-as-a-Library)

At the top of the stack, the application layer, you work with a small vocabulary:

- "Prove this computation."
- "Prove this provenance."
- "Settle this payment under these privacy and finality constraints."
- "Anchor this result to hardware with this security profile."

The SDK turns those sentences into types. A claim is a first-class object: ComputationClaim, ProvenanceClaim, SettlementClaim, optionally decorated with requirements. The compiler and runtime then map those claims to whatever combination of proving backends, settlement rails, and verifiable machines currently clear the SLA at the best VerifyPrice.

Core types:

```
class Policy:
    max_verify_time_sec: float
    max_verify_cost_usd: float
    privacy_level: str              # e.g. "encrypted_io", "public"
    finality_target: str            # e.g. "1m", "30m", "2h"
    allowed_hardware_profiles: list # e.g. ["open_tee_v2", "pure_zk_only"]

class ComputationClaim:
    fn: Callable
    inputs: Any
    policy: Policy

class ProvenanceClaim:
    media_bytes: bytes
    device_profile: str
    policy: Policy

class SettlementClaim:
    payments: list  # [(recipient, amount, asset), ...]
    policy: Policy

class Receipt:
    claim_id: str
    proof_bytes: bytes
    settlement_txid: str | None
```

```
    metadata: dict   # verify_time, verify_cost, hw_profile, etc.
```

SDK surface:

```
sdk = ProofSDK(networks=[...])  # PoUW chains, zk rollups, open-TEE clusters

claim   = sdk.make_claim(...)
receipt = sdk.prove_and_settle(claim, pay_with=user_wallet)
result  = sdk.verify(receipt)   # -> {accept: bool, metadata: ...}
```

Everything else—choice of proving backend, choice of settlement rail, choice of hardware profile—is a routing decision made under the hood, constrained only by the policy you declared.

Example 1: Proofed medical inference on verifiable machines

```
policy = Policy(
    max_verify_time_sec = 1.0,
    max_verify_cost_usd = 0.001,
    privacy_level = "encrypted_io",
    finality_target = "30s",
    allowed_hardware_profiles = ["open_tee_v2", "pure_zk_only"]
)

@proofed(policy=policy)
def diagnose(image: EncryptedImage) -> Diagnosis:
    return model.predict(image)

# In request handler:
claim   = sdk.make_computation_claim(fn=diagnose, inputs=enc_image, policy=policy)
receipt = sdk.prove_and_settle(claim, pay_with=clinic_wallet)
result  = sdk.verify(receipt)
```

Example 2: Camera provenance anchored in open hardware

```
# On capture (running on "open_camera_v1" device):
raw_bytes = camera.capture()

policy = Policy(
    max_verify_time_sec = 0.5,
    max_verify_cost_usd = 0.0005,
    privacy_level = "hide_location_and_identity",
    finality_target = "5m",
    allowed_hardware_profiles = ["open_camera_v1"]
```

```
)

prov_claim = sdk.make_provenance_claim(
    media_bytes=raw_bytes,
    device_profile="open_camera_v1",
    policy=policy
)

receipt = sdk.prove_and_settle(prov_claim, pay_with=newsroom_wallet)
bundle = MediaBundle(media=raw_bytes, provenance_receipt=receipt.export())
```

Example 3: Private payroll over neutral rails

```
policy = Policy(
    max_verify_time_sec = 3.0,
    max_verify_cost_usd = 0.002,
    privacy_level = "shielded_settlement_with_auditable_receipts",
    finality_target = "2h",
    allowed_hardware_profiles = ["pure_zk_only"]
)

payroll_batch = [
    Payment(recipient=alice_addr, amount=1000, asset="USDt"),
    Payment(recipient=bob_addr,   amount=1200, asset="USDt"),
]

settle_claim = sdk.make_settlement_claim(payments=payroll_batch, policy=policy)
receipt = sdk.prove_and_settle(settle_claim, pay_with=treasury_wallet)
```

Energy & Plant Architecture

Plant architecture for verifiable work Objective: deliver verifiable, dispatchable, and composable power to proving and inference clusters so that proof units and verified FLOPs remain cheap to check and neutral to route.

Electrical topology (MV → LV):

- Medium-voltage interconnect with dual utility feeds where available; 2N or N+1 step-down to rack power.
- UPS/BESS for ride-through and proof-preserving shutdowns.
- Power-quality SLOs (voltage/frequency/THD) bound worst-case verifier latency inflation.

Thermal topology:

- Air to liquid (direct-to-chip/immersion) as default for high-density provers.
- Heat-reuse loops to district heating, greenhouses, or absorption chillers to improve ERE.
- Publish PUE and ERE: if we monetize heat, ERE < 1.0 is both possible and economically material.

Network & provenance:

- Redundant backhaul with clock discipline (PTP/Stratum) so receipts carry tight timing jitter bounds.
- Facility attestation path: meters and controllers sign energy/thermal telemetry.

Control plane:

- Workload-aware orchestration couples grid signals → job scheduler.
- Interruptible Bronze proofs pause first; Gold proofs ride through with BESS and on-site firming.

Energy procurement & grid posture:

- Portfolio, not a point bet: long-dated PPAs, nodal index exposure, behind-the-meter firming, and BESS.
- Co-location with stranded or curtailed energy (hydro spill, wind/solar curtailment, flare-gas).
- Demand response as a feature: treat prover/inference farms as dispatchable loads.
- Carbon & provenance: bind certificates into FER format so "kgCO_2e per proof" is auditable.

SLA tiers that map power quality to receipts:

- **Bronze (interruptible):** Shed priority 1. Eligible for demand-response; refunds/penalties via SLA escrow if p95 VerifyPrice breached. Typical use: PoL audits, non-urgent prover workloads.
- **Silver (curtailable with notice):** Shed priority 2 with N+1 firming; curtailment windows posted ahead of time. Typical use: batch MatMul-PoUW, rollup proving backlogs.
- **Gold (firm):** 2N electrical path or N+1 + BESS ride-through; zero curtailment except force majeure. Typical use: settlement-critical proofs, receipt ledger finalization.

Hardware Profiles

Throughout this thesis we treat "hardware profiles" as first-class objects. They are the way we talk about machines in the same language we use for proofs and receipts.

.8 Concept

A hardware profile is a public description of a class of devices that are interchangeable for security and performance purposes. It is not a single physical unit; it is the specification for what counts as a valid unit of that type.

Informally: a profile says, "devices that meet this description behave like this, within these error bars, under this sampling regime."

We use hardware profiles to:

- express trust and threat-model assumptions at the protocol and application layer;
- bind proofs and receipts to the types of machines that produced them; and
- make the honesty (and failure rates) of machines measurable over time.

.9 Naming

Profiles are identified by short, stable strings. Three broad kinds:

1. **Open-hardware profiles** — devices whose RTL/microarchitecture and sampling regime are publicly documented.
   - `open_tee_v2` — second-generation open TEE design with published RTL, side-channel budget, and lot-sampling plan.
   - `open_signer_v1` — simple key-storage/signing chip with open RTL and sampling.
   - `open_camera_v1` — camera/capture pipeline with documented sensor path, secure element, and sampling method.
2. **Logical or pure-cryptography profiles** — paths that do not rely on special hardware.
   - `pure_zk_only` — verification/proving path relying only on general-purpose CPUs/GPUs and cryptographic assumptions.
   - `software_only_v1` — profile used for testing or where hardware attestations are unavailable.
3. **Vendor or mixed profiles** — existing devices not fully open but characterized and sampled.
   - `vendor_tee_X_v1` — proprietary TEE profile with documented side-channel and attestation

scheme.

- `gpu_cluster_Y_v1` — particular GPU farm configuration with known performance and sampling behavior.

.10 Hardware Profile Specification

Each profile has a specification that is part of the public "bill of materials" for the stack. At a minimum:

- **profile_id**: string (e.g., "open_tee_v2")
- **device_class**: what the device is used for (e.g., "TEE", "signer", "accelerator", "camera")
- **trust_mode**: "open", "mixed", "opaque", or "logical"
- **design_commit**: identifiers for design artifacts (hashes of RTL, GDS, firmware images)
- **side_channel_budget**: qualitative/quantitative bounds
- **sampling_plan**: how lots and devices are checked
- **attestation_scheme**: how devices prove identity and state
- **security_target**: high-level security goal

.11 Hardware Profiles in Receipts and Policies

Profiles appear in three main places:

1. **Policies** — Developers express requirements using profiles:

```
allowed_hardware_profiles = ["open_tee_v2", "pure_zk_only"]
```

2. **Receipts (PIDL)** — Every PIDL receipt includes a hardware_profile field.
3. **Telemetry** — The observability layer aggregates metrics by profile.

.12 Example Profiles

open_tee_v2:

- device_class: "TEE"
- trust_mode: "open"
- design_commit: hashes of RTL, layout, firmware
- sampling_plan: random lot sampling, SEM/optical inspection
- intended use: high-value proving, key storage, secure computation

open_camera_v1:

- device_class: "camera"

- trust_mode: “open”
- sampling_plan: device sampling and destructive testing per lot
- intended use: origin-anchored capture for media provenance

pure_zk_only:

- device_class: “logical”
- trust_mode: “logical” (no hardware assumptions)
- intended use: contexts where hardware trust is unacceptable

Communications Resilience Mechanisms

P2P transport set:

- v2 Encrypted P2P (BIP-324) as default for all full nodes and relays; expose QUIC/TLS fallbacks.
- Onion/I2P modes first-class in clients; auto-failover if clearnet handshakes exhibit DPI resets.
- Handshake camouflage (Noise/obfs-style) for relays in high-interdiction ASNs.

Receiver-private routing for payments:

- BOLT12 Offers (receiver-private, reusable invoices) in wallets and merchant stacks.
- Path-blinding where supported.
- Federated ecash and shielded pools as optional "last-mile" sinks/sources.

Settlement survivability:

- Adaptor-signature atomic swaps for BTC↔XMR/ZEC corridors with typed failure & refund flows.
- Wallet UX exposes "abort & refund" as a first-class path.

Topology & anti-eclipse hardening:

- Peer-set diversity targets (geo/ASN spread), randomized peer rotation, inbound slot quotas.
- Gossip-path multiplicity to reduce single-jurisdiction capture.
- Light-client modes that verify receipts over any reachable path (browser, mobile, enclave).

Glossary of Terms & Notation

Triad Privacy, Proofs, and Compute — the three cryptographic capacities that can function as monetary primitives.

VerifyPrice(W) A public KPI vector measuring the cost and time to verify workload W.

VerifyReach(N,R) Metrics for network reachability under censorship conditions.

VerifySettle(C) Metrics for settlement success and safety on corridor C.

$r(W)$ Verification overhead ratio — $v(W)/p(W)$, where v is verification cost and p is production cost.

Work Credit A claim on standardized units of triad work (privacy settlement, proof generation, or verified compute) produced and attested under public SLOs.

PIDL (PIDL) Proof Interface Definition Language — the minimal receipt schema for proofs and settlements.

PaL (PaL) Proofs-as-a-Library — SDK that compiles claims to proofs.

PRK (PRK) Privacy Rails Kit — executes non-custodial, refund-safe settlement over privacy corridors.

FER Facility Energy Receipt — signed summary of a site's energy use over a time interval.

PoUW Proof of Useful Work — consensus mechanism where block rewards are earned by producing verifiable receipts of useful compute.

SLO Service Level Objective — published targets for system performance and availability.

Bronze/Silver/Gold SLA tiers for proof and settlement services with different latency, redundancy, and interruptibility characteristics.

MatMul-PoUW Proof of Useful Work construction based on matrix multiplication verification.

ZK Money Instruments primarily referencing Privacy + Proofs (shielded settlement capacity, ZK proof capacity).

AI Money Instruments primarily referencing Compute + Proofs (verified FLOPs, inference capacity).

Layer 0 Verifiable Machines & Energy — open hardware and sampled supply chains as base reality.

Layer 1 Reachability — communications and transport resilience.

Layer 2 Distribution & Execution — software supply and runtime.

Layer 3 Identity & Claims — pseudonymous credentials without doxxing.

Layer 4 Truth & Work — proof systems, PoUW, VerifyPrice.

Layer 5 Value & Settlement — privacy rails and non-custodial flow.

Layer 6 Governance & Telemetry — keeping neutrality and resilience measurable.