1. Multi-dimensional coloring of transactions and accounts

Let:

- **C** = an arbitrarily large (but finite) set of attributes (“colors”)
- **N** = an arbitrarily large (but finite) set of dimensions (jurisdiction, entity type, risk class, policy state, asset class, etc.)

Each account and transaction is annotated across **C × N**, producing a **multi-dimensional labeled graph**, not a single risk score.

Why this matters

- Regulators and risk teams can observe **subnetwork views, projections, and heatmaps** along the dimensions they care about on first pass.
- This enables **policy-relevant visibility without data extraction**.
- In escalated cases, **selective extraction** occurs only via valid legal process (law-enforcement order, subpoena, FIU / regulator signing, or jurisdiction-specific authority).

This is strictly more expressive — and safer — than post-hoc analytics or monolithic risk scoring.

---

2. Control levers across the full transaction lifecycle

Design toward explicit control and observation points across the transaction lifecycle:

- **Transaction ingress**
  Client-side construction and submission, preflight checks, signature validity, and initial metadata / attestation attachment.
- **Policy evaluation**
  Deterministic checks over transaction context, accounts, and attestations — enforced at the RPC, validator ingress, or pre-execution runtime layer without altering execution semantics.

- **Execution / ordering**
  Parallel execution and scheduling (SVM), with ordering constraints derived from account locks and dependencies — a natural enforcement point for asset- and counterparty-specific constraints.

- **Propagation (gossip)**
  Validator-to-validator dissemination of transactions and blocks, enabling future-state controls such as selective visibility, jurisdiction-aware propagation, and policy-informed "mempool" like behavior.

- **Settlement and finality**
  State commitment and finalization, producing authoritative artifacts for audit, regulatory reporting, and dispute resolution.

This is a **future-state architecture**, but the framing lands because it maximizes **optionality** for regulators, financial institutions, and central banks.

Key insight: controls are layered, contextual, and composable — applied where they are cheapest, safest, and most legible.

---

3. Proof-driven attestations, verifier trees, and economics

A robust proof system enables **attestation trees** across:

- identity and entity status
- jurisdictional eligibility
- policy compliance
- asset provenance
- transactional constraints

These attestations are **reusable economic artifacts**, not one-off checks.

Why the economics matter
Compliance today is slow and expensive because verification is duplicated, manual, and non-reusable.
Proof-native attestations turn verification into a **market**:

- Verifiers incur real cost (data acquisition, validation, liability).
- Consumers pay because verification is **cheaper, faster, and more defensible** than bespoke diligence.
- Attestations are produced once and consumed many times, collapsing discovery and onboarding costs.

Phased approach

- **Initial phase:** centralized verification (us / our APIs) to move fast, standardize, and reduce time-to-approval.
- **Medium–long term:** verifier roles decentralize via:
  - economic incentives (specialized verifier marketplace), and/or
  - jurisdictional designation (licensed or mandated verifiers).

Core insight:
The highest-value compliance data is data that is **hard to obtain, expensive to reproduce, and immediately actionable**. Making it verifiable and transferable turns compliance from a cost center into a **throughput accelerator**.

Unifying principle

Move from opaque ledgers and post-hoc surveillance to **real-time, policy-aware, proof-native financial infrastructure** — observable through a veil by default, extractable only when verifiably required, and flexible enough to satisfy multiple sovereign regimes without fragmenting the network.