WEBSHIELD INC.
PROVISIONAL PATENT APPLICATION

Systems & Methods for Quantum Privacy Catalyst Networks, Tokenized Derivative Settlement & Governed Resource Inheritance

Provisional Filing Date: May 19th, 2026 **Application #**: **64/069,961**.

Inventors: Jonathan Paul Hare (CEO), Richard Arthur Muth (CTO)

Applicant/Assignee: WebShield, Inc. (Delaware)

Type of Application: Provisional Application under 37 C.F.R. § 1.53(c)

Abstract

Systems and methods for trust-verified contribution capture, attribution, settlement, governance inheritance, and capital formation operating within a Quantum Privacy Network (QPN). The invention comprises sixteen integrated subsystems:

- a four-component user-side Sidecar pattern with three-plane decomposition;
- a five-mode signal-input authorization taxonomy;
- a three-stage AI evaluation pipeline;
- a three-ledger/two-log on-ledger record-keeping decomposition with a Permanent Privacy Seal irrevocability primitive;
- a global contribution graph and three-dimensional reputation engine;
- a.qpn Personal Archive format and multi-substrate persistence router;
- a deterministic Settlement Controller with witness-based Cross-Verification Protocol;
- eight domain-specialized Catalyst Vectors;
- a seven-step Governed Agent Loop with Phase-0 DORMANT QPC state;
- a 15-dimensional Premium Framework operating as a Manager-Discretion AI Model;
- a Six-Layer Catalyst Architecture with Behavioral Activation, Multi-Factor Identity Binding, Catalyst Proxy Addresses, and contribution categories taxonomy;
- Senior/Junior QPT Derivatives with dual-hurdle RBF and Encumbrance Authority;
- Stage-Differentiated Revert, Accelerator Lock with Monetization Uplift Multiple, and Governance DNA conditions;
- Privacy Network Exchange liquidity architecture with Exchange Provider/Liquidity Provider role separation, Backing Pool Multiples, and QPT-Collateralized Lending;
- a Quantum DNA/Gene/Genome inheritance architecture with Lamarckian Inheritance and Multi-Layered Aggregation; and (P) Two-Parent PPN and N-Parent Polygenomic Recombination Engines. Each independent claim recites the canonical QPN-infrastructure Wherein clause, inheriting 2016 priority for the recited infrastructure elements from U.S. Patent No. 12,316,610 B1.

Cross-Reference to Related Applications

This application incorporates by reference the following WebShield, Inc. patent filings, each commonly assigned to WebShield, Inc.:

U.S. Patent No. 12,316,610 B1 — "Privacy Network and Unified Trust Model for Privacy Preserving Computation and Policy Enforcement" (granted May 27, 2025)

Appl. No. 17/321,700, filed May 17, 2021; earliest priority March 16, 2016 via U.S. Provisional Application No. 62/309,153; 19 claims (5 independent, 14 dependent). Foundational granted patent: Quantum Privacy, Trust Blocks, Exchange Token issuance, Privacy Network structure, Unified Trust Model, QPCs, Privacy Domains, Trust Criteria, Proof-of-Trust, EasyAccess workflow threads.

U.S. Patent Application No. 19/206,859 — "Quantum Privacy, Proof of Trust, and Privacy Network Exchange" (filed May 13, 2025)
Non-provisional continuation-in-part of U.S. Patent No. 12,316,610 B1; extends the foundational Privacy Network and Unified Trust Model disclosures.

U.S. Provisional Patent Application No. 63/804,583 — "Quantum Privacy, Proof of Trust, and Privacy Network Exchange" (filed May 12, 2025)
Attorney Docket WEBS.08PRO. Foundational provisional disclosure, including its supporting documents entitled "WebShield QP-Drilldown for Provisional Patent Filing (2025-05-12)" and "WebShield Privacy Network — Global Quantum-Safe Cybersecurity Protection." Quantum-safe boundary primitives, Privacy Domain operational details, and narrative architectural overview.

U.S. Provisional Patent Application No. 63/895,861 — "Privacy Network Exchange: Systems and Methods for Trust-Verified Tokenization and Settlement" (filed October 8, 2025)
Attorney Docket WEBS.10PRO. Trust-Block-bound Exchange Token issuance, settlement mechanics, on-ledger attribution, and the Exchange Root allocation primitive.

U.S. Provisional Patent Application No. 63/923,253 — "Systems and Methods for Quantum Privacy-Enabled Self-Funding AI Trust, Safety & Compliance" (filed November 22, 2025)
AI governance primitives: deterministic replay, digital-twin QPCs, zero-knowledge multi-party negotiation, cleanroom synchronization, trust-weight engine, multi-metric governance, and cross-domain policy-bound actuation.

U.S. Provisional Patent Application No. 63/926,629 — "Systems and Methods for Quantum Privacy-Enabled Personalized, Value-Based Universal Exchange for Better Health" (filed November 27, 2025)
Healthcare domain embodiment of the QPN architecture; personalized, value-based exchange primitives.

U.S. Provisional Patent Application No. 63/931,387 — "Systems & Methods for a Self-Funding, Self-Organizing Quantum Privacy Exchange and Accelerator Network" (filed December 4, 2025)
Self-funding mechanics, self-organizing topology dynamics, recursive Accelerator formation, Accelerator topology, Premium framework, Resource Pool formation, and deferred-activation primitives.

U.S. Provisional Patent Application — "Systems and Methods for Governed AI Coordination, Safety, and Derivative Ecosystem Formation within a Quantum Privacy Network" (filed concurrently, on or about May 18, 2026)
Companion application filed concurrently herewith and incorporated by reference in its entirety (the "Q–Z Supplemental Provisional"). Discloses Claim Families Q through Z of the integrated Quantum Privacy Network architecture — governed AI coordination, distributed graph runtime, recursive orchestration, federated synchronization, and derivative AI ecosystem formation. The present application and the Q–Z Supplemental Provisional disclose interoperable and complementary subject matter and together form a coordinated disclosure and priority framework for subsequent continuation and continuation-in-part applications. The Q–Z Supplemental Provisional's application number was not yet assigned as of the filing of the present application; it is identified herein by title, applicant, and filing date, and the application number may be supplied by subsequent amendment.

Each independent claim in this application recites infrastructure elements drawn from the granted U.S. Patent No. 12,316,610 B1 by way of the canonical Wherein clause specified in QPN Context Primer v2.0.3 §22.7. Priority inheritance for the recited QPN-infrastructure elements (Quantum Privacy Cells, Privacy Domains, Trust Criteria, Proof-of-Trust, Trust Blocks, EasyAccess workflow threads) traces to the 2016 priority date of U.S. Patent No. 12,316,610 B1 for §102 and §103 analysis. The new combinations and embodiments disclosed herein carry this application's filing date as their priority date for purposes of those new combinations.

Field of the Invention

The present invention relates to systems and methods for trust-verified contribution capture, attribution, settlement, governance inheritance, and capital formation operating within a Quantum Privacy Network (QPN) as defined in U.S. Patent No. 12,316,610 B1. More particularly, the present invention relates to:

- user-side contribution capture infrastructure (the Quantum Privacy Sidecar Pattern and eight specialized Catalyst Vectors);
- authorization-layer signal classification (Five Signal Input Modes);
- AI-mediated contribution evaluation (Three-Stage Evaluation Pipeline);
- on-ledger record-keeping with cryptographic irrevocability (Three-Ledger/Two-Log Architecture and Permanent Privacy Seal);
- reputation and contribution graph assembly (Global Contribution Graph and DFS Reputation Engine);
- local-storage and multi-substrate persistence (Personal Archive.qpn format with Multi-Substrate Persistence Router);
- deterministic settlement enforcement (Settlement Controller with Cross-Verification Protocol);
- agentic AI execution and deferred-activation compliance (Governed Agent Loop with DORMANT QPC State);
- parameterized reward allocation (15-dimensional Premium Framework with Manager-Discretion AI Model);
- gamification and identity primitives (Behavioral Activation, Multi-Factor Identity Binding, Catalyst Proxy Addresses);
- capital formation derivative instruments (Senior/Junior QPT Derivatives with dual-hurdle RBF, Encumbrance Authority, Accelerator Lock with Monetization Uplift Multiple tracking);
- Privacy Network Exchange liquidity architecture (Exchange Provider / Liquidity Provider role separation, Backing Pool Multiples, Priority Pool absorption, QPT-collateralized lending, existing-FI balance-sheet tokenization integration); and
- cross-resource governance inheritance (Quantum DNA/Gene/Genome architecture with Lamarckian Inheritance, multi-Genome architecture, governance superposition, mitochondrial DNA analog, Premium inheritance, multi-layered aggregation, and two-parent and N-parent recombination).

Background of the Invention

Privacy-preserving digital infrastructure has been substantially advanced by the architecture disclosed in U.S. Patent No. 12,316,610 B1 (the '610 Patent), which establishes the foundational primitives of the Quantum Privacy Network — Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, the Unified Trust Model, and EasyAccess workflow threads. The '610 Patent and subsequent WebShield Inc. provisional filings have established the technical and legal substrate for trust-verified resource exchange, settlement, and AI governance at planetary scale.

Despite this substantial foundation, multiple architectural challenges remain unresolved by the existing patent stack. These challenges fall into three groupings that motivate the present invention.

Challenges in Contribution Capture and Catalyst Network Operation

First, while the '610 Patent establishes the cryptographic primitives for trust-verified resource exchange, it does not specify the user-side contribution capture infrastructure required for individuals and organizations to bring contributions into the network at scale. Conventional contribution-capture systems — browser extensions, telemetry agents, observability sidecars — intermix event observation, passive listening, encrypted storage, and verification bridging in a single process, defeating the cryptographic separability required for participant-controlled attribution. They also fail to provide a structured signal-mode taxonomy, leading to over-restrictive or over-collecting authorization regimes. They lack AI-mediated evaluation pipelines suitable for trust-verified attribution at scale. They lack on-ledger record-keeping decompositions that preserve plane separation. They lack global contribution graph assembly that preserves participant privacy. They lack reputation engines suitable for the Catalyst Network's adoption mechanics. They lack multi-substrate persistence that prevents vendor lock-in. They lack deterministic settlement enforcement with witness-based cross-verification. They lack agent governance suitable for pre-activation compliance. They lack the 15-dimensional Premium Framework

needed to support quid-pro-quo-rebutting reward allocation. And they lack gamification and identity primitives integrated with Trust Block accountability.

Challenges in Capital Formation and Liquidity

Second, the existing patent stack covers the settlement-side mechanics of Exchange Token issuance but does not specify the capital-formation derivative instruments required for pre-settlement institutional capital formation, nor the liquidity architecture required for scale-out operation of the Privacy Network Exchange. Conventional tokenized network capital formation either sells native protocol tokens directly to investors (conflating investment with operational participation) or forgoes pre-settlement capital formation entirely (foreclosing institutional pathways). Conventional liquidity systems conflate settlement counterparty and capital provider roles, lack structural controls on liquidity pool capacity, lack standardized tranche absorption logic, lack Trust Block-bound tokenized collateral lending, and lack integration with existing financial institutions' balance sheets.

Challenges in Cross-Resource Governance Inheritance

Third, the existing patent stack establishes per-resource Trust Block bindings but does not specify how governance characteristics propagate across the resource → derivative → solution → exchange creation chain, across participant creations (new Personal Privacy Networks, Resource Derivatives), or across multiple coexisting governance contexts per participant. Without a structured inheritance architecture, governance characteristics either remain locked at the originating resource (defeating Resource Derivative value creation) or are lost across the creation chain (defeating attribution to upstream contributors). Without multi-context support, participants cannot operate distinct personal, professional, and anonymous Genomes simultaneously. Without acquired-characteristic propagation, accrued Premium contributions are not inherited by descendant entities. Without operational-provenance tracking separate from governance lineage, the operator history of Catalyst Network infrastructure is conflated with the governance history of participants.

Need for an Integrated Solution

The present invention addresses these three groupings of challenges through sixteen integrated Claim Families (A–P) covering contribution capture infrastructure (A–I, K), capital formation and liquidity (L–N), and cross-resource governance inheritance (O–P), together with a parameterized reward allocation framework (J) that bridges the three. All inventions operate within the QPN-enabled infrastructure as defined in U.S. Patent No. 12,316,610 B1; the canonical Wherein clause in each independent claim anchors this dependency and establishes 2016 priority for the recited QPN-infrastructure elements.

Brief Summary of the Invention

The present invention provides systems and methods for trust-verified contribution capture, attribution, settlement, governance inheritance, and capital formation operating within a Quantum Privacy Network (QPN). In a broad aspect, the invention comprises sixteen integrated subsystems — designated Family A through Family P — each addressing a distinct architectural concern, but operating in concert as an integrated system within the QPN-enabled infrastructure.

Family A — Quantum Privacy Sidecar and Witness Architecture

A four-component user-side contribution capture pattern (Witness Agent, Listener Agent, Local Vault, Verification Bridge) operating with cryptographic separability via per-component QPCs, decomposed into three planes (Data, Control, Management), and instantiable as a browser-extension embodiment.

Family B — Five Signal Input Mode Architecture

A five-mode contribution capture taxonomy (Active, Directed, Ambient, Institutional, Evangelized) with mode-specific Trust Criteria and a Mode Discriminator classifying incoming signals at the QPC boundary.

Family C — Three-Stage AI Evaluation Pipeline

An AI-mediated three-stage contribution evaluation (Semantic Classification, Identity Enrichment, Temporal Durability) with each stage executing within a distinct QPC under the deterministic replay primitive of the November 18, 2025 AI Governance Provisional.

Family D — Three-Ledger/Two-Log Architecture and Permanent Privacy Seal

An on-ledger record-keeping decomposition into three Ledgers (Contribution, Authorization, Settlement) plus two Logs (Data Plane, Control Plane), with a Permanent Privacy Seal primitive triggering cryptographic destruction of participant content keys.

Family E — Global Contribution Graph and Reputation Engine

A global contribution graph assembled from heterogeneous Personal Archives without centralizing contribution content, plus a DFS-traversal Reputation Engine emitting reputation scores across five confidence tiers, seven signal types, six badges, and a three-dimensional decomposition (Contribution, Identity, Behavioral).

Family F — Personal Archive, CCP, and Multi-Substrate Persistence

A.qpn Personal Archive file-format primitive, a Contributor Intelligence Module (CCP) for local analytics over the Personal Archive, and a Multi-Substrate Persistence Router/Adapter pattern preventing vendor lock-in across heterogeneous storage substrates.

Family G — Settlement Controller and Cross-Verification Protocol

A deterministic Settlement Controller enforcing non-discretionary Exchange Token issuance upon authorized cross-party Resource reuse satisfying a Settlement Eligibility Predicate, with a witness-based Cross-Verification Protocol enabling third-party attestation without content disclosure.

Family H — Specialized Catalyst Vectors

Eight domain-specialized Catalyst Vectors (WebVector, VoiceVector, CodeVector, DocVector, AgentVector, CommVector, MeetVector, MessageVector) each implementing the Sidecar pattern with surface-specific primitives (origin verification, on-device voice processing, pull-request-level attribution, section-level W3C Verifiable Credentials, Proof of Orchestration, verified engagement loop with Message-ID matching, meeting attendance attestation, privacy-preserving proxy network).

Family I — Governed Agent Loop and Phase-0 DORMANT QPC State

A seven-step canonical agent execution sequence (Sense, Interpret, Propose, Authorize, Execute, Verify, Learn) with Trust Block binding at each step, plus a Phase-0 DORMANT QPC State permitting Manager-Originated contribution accruals for participants who have not yet self-activated, with PPCS Activation Gate enforcement on transition to ACTIVE state.

Family J — Premium Framework as Operative Allocation Mechanism

A 15-dimensional Premium Framework (5 Launch Premiums, 8 Governance Premiums, 2 Adaptive Premiums for Proportionality and Balance) operating as a Manager-Discretion AI Model with quid-pro-quo rebuttal, Premium Multiple Compression Curve governing maturity-arc decay, and runaway prevention via Adaptive Premium Compensation.

Family K — Behavioral Activation, Reputation, and Identity Resilience

A Six-Layer Catalyst Architecture umbrella, a Behavioral Activation System (10 levels, 24 badges, dual-XP, evidence bonus, streak multiplier), Multi-Factor Identity Binding composing multiple attestations into a single Identity Trust Block, Authorized Catalyst Proxy Addresses for delegated capture, and Catalyst Contribution Categories taxonomy (Nine Primary, UX Tier, Program→Objectives).

Family L — Senior/Junior QPT Derivative Capital-Formation Architecture

A Senior QPT Derivative with dual-hurdle RBF (target IRR + target MOIC) and issuer payoff optionality, Encumbrance Authority over the full Exchange Root Token and Accelerator Incentive & Investment Pool allocations under deferred-activation conditions, a Junior QPT Derivative absorbing first-loss with three configurable offset levels, and an Accrual Rights Swap primitive indexed to the Premium Multiple Compression Curve.

Family M — Stage-Differentiated Revert, Accelerator Lock, and MUM Tracking

A capital recovery waterfall varying by ecosystem stage (Pioneer, Cascade, Automated, Self-Funding), an Accelerator Lock Mechanism combining Monetization Uplift Multiple thresholds with Governance DNA

conditions to render Accelerator Incentive & Investment Pool allocations non-dilutable, and per-Accelerator MUM tracking on the Settlement Ledger.

Family N — Privacy Network Exchange Liquidity Architecture

Structural separation of Exchange Provider and Liquidity Provider roles, Backing Pool Multiples Architecture (portfolio-multiple, annual-flow-multiple, cap-amount), Tranche Priority Pool Absorption (two-stage RoC then RoI waterfall), QPT-Collateralized Lending with Trust-Block-bound release and PoT-enforced margining, and Existing-FI Balance-Sheet Tokenization Integration with dual-control.

Family O — Quantum DNA/Gene/Genome Inheritance Architecture

A three-level inheritance hierarchy (Quantum Genes as atomic governance traits composing into Quantum DNA as composite governance profiles composing into Quantum Genomes as full governance identities), with Lamarckian Inheritance of acquired characteristics, Multi-Genome state isolation, Governance Superposition, Mitochondrial DNA Analog for operational provenance, Premium Inheritance via Trust Block Chain, and the Multi-Layered Aggregation Principle.

Family P — Two-Parent PPN Creation and N-Parent Polygenomic Recombination

A Two-Parent Sexual Reproduction Model for Personal Privacy Network creation via deterministic Genome recombination from exactly two parents, plus an N-Parent Polygenomic Recombination primitive for Resource Derivative creation from N ≥ 2 parents with derivation-share weighting, both deterministically replayable via the deterministic replay primitive.

Detailed Description of the Invention

The Detailed Description below presents each of the sixteen Claim Families in sequence. Each Family section follows a consistent structure: Field Summary, Problem Addressed, Solution Overview, Components, Process Flow, Alternative Embodiments, and Enabling Cross-References. All Family operations occur within the QPN-enabled infrastructure of U.S. Patent No. 12,316,610 B1; the Wherein clause in each independent claim (Phase 4) anchors this dependency at the claim-language level.

Family A — Quantum Privacy Sidecar & Witness Architecture

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -1, -4, -32.

Field Summary

User-side contribution capture infrastructure operating within a Quantum Privacy Network (QPN).

Within the broader QPN architecture, Family A sits at the user-side boundary of the contribution-capture stack: it is the participant-facing protocol layer that converts raw, locally-observable user activity into cryptographically attestable Verification Envelopes that the Catalyst Network can ingest under Proof-of-Trust. The Sidecar pattern is therefore foundational to all subsequent Families: every contribution attributed under the Catalyst Contribution Graph (Family E), every Premium evaluation under the AI Pipeline (Family C), every Trust-Block-bound ledger entry under the Three-Ledger architecture (Family D), and every settlement issuance under the Settlement Controller (Family G) depends on the Sidecar having faithfully captured, sealed, and bridged the originating event. Per QPN Context Primer v2.0.5 §§4.2–4.4 and the, the Sidecar is the user-side equivalent of the Catalyst Network's network-side Settlement Controller: each provides a trust-verified protocol surface at its respective boundary.

Problem Addressed

Conventional contribution-capture systems (browser extensions, telemetry agents, observability sidecars) intermix four functions that must, for trust-verified attribution, remain cryptographically separable: event observation, passive listening, encrypted local storage, and verifiable bridging to a network ledger. Where these functions are co-located in a single process, a single compromise yields plaintext access to all four. Where they are merely distributed across services, no architectural guarantee binds their cooperation to the originating participant's authorization. Existing systems also fail to preserve plane separation — data, control, and management traffic share addressable surface, defeating compartmentalized authorization.

The deeper architectural problem the Sidecar addresses is that conventional contribution-capture systems are not merely insecure — they are structurally incapable of producing the kind of evidence required for trust-verified attribution at population scale. A browser extension that captures click events, encrypts them, and posts them to a backend produces evidence that any sufficiently-funded adversary can forge: there is no architectural guarantee that the encrypted blob came from the claimed user, on the claimed device, at the claimed time, observing the claimed surface. The Catalyst Contribution Graph requires evidence at a substantially higher trust grade: evidence that survives subpoena, that survives nation-state-grade adversary modeling, and that survives the eventual public publication of the underlying signals. Existing telemetry architectures fail every one of these tests.

A second dimension of the problem is plane-separation failure. Conventional sidecar architectures (the term originates in service-mesh practice) typically run all three planes — data, control, and management — within a single addressable surface, often within a single process or container. This permits a single compromise of the management plane (e.g., via supply-chain attack on a sidecar dependency) to yield plaintext access to all data flowing through the data plane. The Quantum Privacy Sidecar's Three-Plane Architecture, by contrast, allocates non-overlapping QPC sets to each plane such that no single compromise of one plane yields cryptographic access to another. This plane-separation property is what enables the Sidecar to be trusted with both routine contribution capture (data plane) and high-stakes governance events such as Premium adjustments and Identity Resilience operations (management plane).

Solution Overview

The Quantum Privacy Sidecar is a four-component architectural pattern (Witness Agent, Listener Agent, Local Vault, Verification Bridge) that executes each component within a distinct Quantum Privacy Cell (QPC) as defined in U.S. Patent No. 12,316,610 B1. Each QPC enforces the cryptographic boundary; inter-component communication is mediated through Trust Block-bound message envelopes whose authorization is verified by Proof-of-Trust at every hop. The Three-Plane Architecture (Data / Control / Management Plane) further decomposes Sidecar traffic such that each plane operates in a non-overlapping QPC set with plane-specific Trust Criteria. Browser-extension embodiment binds the Sidecar to the browser's origin model, with extension-specific origin verification supplementing Trust Criteria.

The four-component decomposition (Witness, Listener, Vault, Bridge) is not arbitrary; it derives directly from the four distinct trust roles that any contribution-capture system must implement. Observation (what happened?) must be cryptographically separable from passive listening (what was the ambient context?) which must be separable from storage (what state persists locally?) which must be separable from bridging (what was authorized to leave the device?). Conflating any two of these roles in a single component creates a class of attacks in which a compromise of the conflated component yields more authority than the role's purpose requires. The Sidecar's per-component-QPC decomposition enforces this separation cryptographically, not merely procedurally.

The browser-extension embodiment (*QPN Catalyst Launch Plan & Rewards Framework* -32) is architecturally significant because it grounds the Sidecar in an existing trust boundary that browser vendors already enforce — the extension origin model. By pinning the Sidecar's manifest to a specific QPN origin and supplementing Trust Block-bound signing with extension-specific origin verification, the embodiment achieves a defense-in-depth posture: an adversary who compromises the QPC boundary still faces the browser's extension origin check, and vice versa. The mobile embodiment achieves analogous defense-in-depth using Secure Enclave / StrongBox for the Vault QPC. The server-side organizational embodiment scales horizontally by mapping the Three-Plane Architecture to distinct hosting environments, supporting enterprise deployments where the participant is an organization rather than an individual.

Components

Detailed component-level disclosure follows. Each component is sourced from the *QPN Catalyst Launch Plan & Rewards Framework* (CLP) §§3.1–3.4 and §4.2 (Sidecar four-component decomposition), and is operationalized within the QPC primitives defined in U.S. Patent No. 12,316,610 B1. Component parameterizations and configuration ranges identified below are illustrative; additional embodiments are disclosed in the Alternative Embodiments section.

Witness Agent

Event-observation component that captures user-originated interactions (clicks, keystrokes within a designated input scope, document edits, voice activations triggered by explicit user invocation). Operates within a Witness-scoped QPC. Emits Witness Records — Trust-Block-wrapped event tuples bound to the originating Privacy Domain.

The Witness Agent is implemented as an event-observation component executing within a Witness-scoped QPC (per U.S. Patent No. 12,316,610 B1 QPC primitive) with the following on-disk schema for Witness Records: record_uuid (content-addressed identifier under a Trust-Criteria-specified quantum-safe hash function — typical default SHA3-256 or SHAKE-256 per Trust Criteria configuration), event_class (enum across browser-vector event types — POINTER_DOWN, POINTER_UP, KEYSTROKE, FOCUS_CHANGE, DOCUMENT_EDIT, VOICE_ACTIVATION_TRIGGER, MEDIA_PLAYBACK_BEGIN, etc.), event_payload (typed payload schema-bound to event_class), event_timestamp (Trust-Block-anchored Lamport-ordered timestamp), origin_privacy_domain_uuid (reference to the originating Privacy Domain), origin_trust_block_uuid (reference to the active Trust Block at event observation), signing_key_ref (reference to the participant's Privacy-Domain-master-key-derived signing key). The Agent's input scope is bounded by an event_scope_predicate (Trust-Criteria-bound) that gates which UI surfaces are observable; events outside the scope are dropped before the Witness Record schema is even instantiated, preventing observation-leak attacks. Per, the Witness Agent's observation-frequency bound is configurable (typical: 10–100 events/second under standard contribution profiles; bounded by event_rate_limit_bps parameter to prevent observation-flood attacks against the Local Vault).

Listener Agent

Passive-observation component for ambient signals authorized under the Five Signal Input Modes (Family B). Operates within a Listener-scoped QPC with Trust Criteria limiting observation to authorized ambient surfaces. Does not capture content; emits Listener Attestations describing the existence and class of an observed signal.

The Listener Agent is implemented within a Listener-scoped QPC, distinct from the Witness Agent's QPC, with explicit Trust Criteria constraints forbidding the Listener Agent from capturing event content (only event existence and class). The on-disk schema for Listener Attestations: attestation_uuid, signal_class (enum across the Five Signal Input Modes' ambient signal taxonomy: AMBIENT_AUDIO_PRESENCE, AMBIENT_VIDEO_PRESENCE, GEOLOCATION_CLASS, DEVICE_PROXIMITY, CALENDAR_CONTEXT, etc.), signal_class_confidence (rational $\in [0,1]$ reflecting classifier confidence), mode_tag (the active Signal Input Mode per Family B), observation_window_start and observation_window_end (Trust-Block-anchored timestamps), aggregated_summary_payload (Trust-Criteria-bound; defaults to presence-only metadata; expanded summaries permitted only under Institutional Mode with sponsor attestation). The Listener Agent's content-suppression invariant is structural rather than procedural: the QPC boundary enforces that signal-payload data cannot leave the Listener Agent's address space in plaintext; only the Attestation schema fields traverse the inter-component message bus. Per and, the Listener Agent's Five-Mode admission policy is enforced at the QPC's Trust Criteria gate, not at the application layer.

Local Vault

Encrypted local storage component. Operates within a Vault-scoped QPC whose cryptographic boundary enforces that vault contents never leave the participant's device in plaintext. Storage substrate is abstracted by the Multi-Substrate Persistence layer (Family F).

The Local Vault is implemented within a Vault-scoped QPC with cryptographic-boundary enforcement against the participant's device-resident storage substrate (per Family F Multi-Substrate Persistence). On-disk Vault schema: vault_record_uuid (content-addressed identifier), wrapped_record_payload (authenticated-encryption-wrapped Witness Record or Listener Attestation under a key derived from the Privacy Domain master key per Trust-Criteria-specified KDF — typical default HKDF-SHA3-256 with Trust Block lineage identifier as info parameter), wrap_nonce (96-bit nonce, never reused within a single Privacy Domain key generation), wrap_authentication_tag (128-bit AEAD authentication tag), vault_record_lineage_uuid (reference to the Trust Block under which this record was vaulted). The KDF discipline is critical: derivation includes the Trust Block lineage identifier so that records vaulted under different Trust Blocks have cryptographically distinct encryption keys, enabling Permanent Privacy Seal operations (Family D) to destroy the keys for specific Trust Block lineages

without affecting others. Per, the Vault QPC's storage-substrate-abstraction layer (provided by Family F) permits the underlying substrate to be encrypted-disk-on-local-device, encrypted-cloud-storage, hardware-token-resident encrypted enclave, or other QPC-Trust-Criteria-permitted substrates without breaking the cryptographic-boundary guarantee.

Verification Bridge

PoT bridge to the Catalyst Network. Operates within a Bridge-scoped QPC. Constructs Verification Envelopes from Witness Records and Listener Attestations and presents them to the Catalyst Network's Settlement Controller (Family G) for trust-verified ingestion. The Bridge cannot decrypt Vault contents; the Bridge can only certify that a Verification Envelope was authorized by a specified Trust Block.

The Verification Bridge is implemented within a Bridge-scoped QPC with the following Verification Envelope schema: envelope_uuid, referenced_vault_record_uuid_set (the set of Vault records being verified by this Envelope; ordered per Trust-Criteria specification), referenced_listener_attestation_uuid_set (the set of corresponding Listener Attestations), mode_tag (the active Signal Input Mode per Family B), envelope_signature (signed under the participant's Trust-Block-bound signing key per a Trust-Criteria-specified quantum-safe signature scheme — typical default Dilithium-3 or Falcon-512 per Trust Criteria), envelope_signing_block_uuid (the Trust Block under which the signature was issued). The Bridge's structural-incapacity property: the Bridge QPC's Trust Criteria forbid the Bridge from possessing the Privacy Domain master key or the Vault decryption keys; the Bridge can only sign over content-addressed identifiers of Vault records, not over their plaintext content. Per, the Bridge therefore cannot fabricate Vault records or alter their content; it can only certify that a specified Vault record was observed by the Witness Agent under specified Trust Block authorities. The Catalyst Network ingress verifies each Envelope's signature, the referenced Vault records' Trust Block lineage, and the Listener Attestations' Mode consistency before admitting the contribution to the Contribution Ledger (Family D).

Process Flow

A user-originated interaction is observed by the Witness Agent and packaged as a Witness Record.

The Witness Record is encrypted under the participant's Privacy Domain key and written to the Local Vault.

Concurrently, the Listener Agent emits a Listener Attestation describing any contextual ambient signals authorized under the participant's active Signal Input Mode (Family B).

Upon a synchronization trigger (time-based, event-count-based, or explicit user request), the Verification Bridge constructs a Verification Envelope referencing the Vault-resident Witness Record by content-addressed identifier.

The Verification Envelope is signed under the participant's Trust Block keys; the signature is verified by Proof-of-Trust at the Catalyst Network ingress.

Upon PoT verification, the Settlement Controller (Family G) updates the Contribution Ledger (Family D) with a reference to the Verification Envelope; no plaintext contribution content traverses the network boundary.

Alternative Embodiments

Browser extension embodiment (*QPN Catalyst Launch Plan & Rewards Framework* -32): the Sidecar is delivered as a browser extension whose manifest pins it to a specific QPN origin; the extension's origin verification supplements the Trust Block-bound signing.

Native application embodiment: the Sidecar runs as an OS-resident process group, each component in a separate OS-level sandbox supplementing the QPC boundary.

Mobile embodiment: the Sidecar runs as a system-level service on iOS or Android, leveraging Secure Enclave / StrongBox for the Vault QPC and platform attestation for the Verification Bridge.

Server-side embodiment for organizational endpoints: the Sidecar runs within a server cluster with horizontal scaling across plane boundaries; the Three-Plane Architecture maps directly to distinct hosting environments.

Wearable-device embodiment: The Sidecar may execute within a wearable device (smartwatch, augmented-reality headset, biometric sensor) with the four QPCs running as constrained-resource processes. Wearable embodiments typically configure the Witness Agent for sensor-class events (heart rate above threshold, glance-

direction change, voice activation) and the Listener Agent for ambient environment classification. The Vault QPC leverages the wearable's secure enclave (e.g., Apple Secure Enclave on Apple Watch, Trusty TEE on Android wearables) for cryptographic-boundary enforcement.

IoT-fleet embodiment: An IoT device fleet (smart home sensors, industrial sensors, vehicle telematics) may collectively run the Sidecar in a distributed embodiment where individual devices host subsets of components — e.g., a sensor hosts the Witness Agent; a hub device hosts the Vault QPC; a gateway device hosts the Verification Bridge. The four-QPC decomposition maps onto distinct devices rather than distinct processes, with the inter-component message bus implemented over the local-network substrate under Trust-Block-bound message authentication.

Embedded-device co-location embodiment: For ultra-low-power embedded contexts (medical implants, agricultural sensors, environmental monitors), the Sidecar may operate as a two-QPC reduced variant: a Witness+Listener combined QPC (with logical separation between scope predicates) and a Vault+Bridge combined QPC. Reduced QPC count trades some security separation for operational feasibility; deployments must explicitly Trust-Block-anchor the reduced-QPC variant designation.

Hybrid edge-cloud embodiment: The Sidecar may operate with Witness and Listener Agents at the edge (participant device) and Vault and Bridge components in cloud-resident QPCs under a Trust-Block-bound delegation. The cloud-resident Vault must satisfy participant-controlled-key-material invariants: the Privacy Domain master key never leaves the participant's edge device, but Vault records may be stored in cloud-side encrypted storage. Cloud-side Bridge operations sign under participant-issued ephemeral keys delegated via Trust-Block-bound delegation records.

Browser-native embodiment: Beyond the existing browser-extension embodiment, the Sidecar may execute as a browser-native primitive — built into the browser engine itself rather than as an extension. Browser-native embodiments leverage native browser-process isolation and origin-policy enforcement for QPC-boundary equivalence, with the Bridge integrating into the browser's network stack. This embodiment is forward-compatible with browser-vendor-supported privacy-preserving primitives (e.g., browser-native attestation primitives).

Air-gapped embodiment: For high-security contexts (classified research environments, sensitive corporate intelligence), the Sidecar may operate in an air-gapped embodiment: the Witness, Listener, and Vault QPCs operate on an air-gapped device; the Verification Bridge runs separately on a connected device; cross-device communication uses one-way data diodes with explicit human-mediated Envelope transfer events. The four-QPC decomposition preserves all cryptographic-boundary properties despite the physical air-gap.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Sidecar contribution capture:** (i) the participant's Privacy Domain must be active with valid Trust Block authorities; (ii) the active Signal Input Mode (Family B) must be loaded and verified; (iii) all four QPC boundaries (Witness, Listener, Vault, Bridge) must be initialized with valid Trust Criteria; (iv) the Verification Bridge must hold a valid Trust-Block-bound signing key.
- **Postconditions:** (i) the captured event is encrypted in the Local Vault; (ii) the corresponding Listener Attestation (if applicable per Mode) is emitted; (iii) either the Verification Envelope is signed and submitted to the Catalyst Network ingress (under sync triggers) or remains pending in the Vault for later sync.
- **State transitions:** Witness Record lifecycle: OBSERVED → VAULTED → REFERENCED → SUBMITTED → ADMITTED (Catalyst Network confirmation).
- **Error handling:** Witness-Agent observation failures (event-rate-limit exceeded, scope-predicate-violation) emit Audit Log records without Witness Record emission; Vault-write failures trigger VAULT_WRITE_FAILED states with retry under exponential backoff; Bridge-signing failures (e.g., signing-key rotation in progress) emit BRIDGE_SIGNING_DEFERRED records; ingress-rejection (Catalyst Network rejects the Envelope) emits ADMISSION_REJECTED records and is reported back to the participant for diagnosis.
- **Performance characteristics:** Witness-Record emission is O(1) per event; Vault write is O(1) per record (with periodic O(log N) index updates); Envelope construction is O(K) per K referenced Vault records; Bridge

signing is O(1) per Envelope. Per, the Sidecar is designed for sub-millisecond per-event overhead under typical contribution profiles, supporting real-time interactive use without perceptible latency impact.

Cross-Family Integration

Upstream Dependencies. Family A inherits the foundational QPC, Privacy Domain, Trust Block, Proof-of-Trust, and EasyAccess workflow-thread primitives from U.S. Patent No. 12,316,610 B1. The four-component Sidecar decomposition operates exclusively within QPN-enabled infrastructure as recited in the §22.7 Wherein clause; every Witness Record, Listener Attestation, Vault-resident payload, and Verification Envelope is Trust-Block-bound at construction. Family A also inherits the Five Signal Input Modes from Family B at runtime — the active Mode at event-observation time is the binding admission policy for the Sidecar's Listener Agent.

Downstream Consumers. Family A's primary downstream consumer is Family C (Three-Stage AI Evaluation Pipeline): every Verification Envelope emitted by the Sidecar Bridge is the canonical input to the Pipeline Coordinator at Catalyst Network ingress. Family D (Contribution Ledger) consumes admission outcomes; Family E (Global Contribution Graph) consumes admitted Contribution Ledger Entries as graph-edge endpoints; Family G (Settlement Controller) consumes the resulting Pipeline Completion Records as settlement-eligibility inputs; Family J (Premium Framework) consumes Pipeline outputs for Premium Score computation. Family H (Specialized Catalyst Vectors) inherits the four-component pattern wholesale — every WebVector, VoiceVector, CodeVector, DocVector, AgentVector, CommVector, MeetVector, and MessageVector is a vector-specific instantiation of Family A.

Lateral Interactions. Family A interoperates with Family F (Personal Archive, CCP & Multi-Substrate Persistence) at the Vault-QPC storage-substrate-abstraction boundary: the Vault QPC's storage operations delegate to Family F's Substrate Adapters and Persistence Router. Family A interoperates with Family K (Identity Resilience) at the signing-key boundary: the Verification Bridge's signing keys are managed under the Multi-Factor Identity Binding primitive, with key rotation operations Trust-Block-bound under the Family K Lifecycle Audit Logger.

Emergent System-Level Properties. The Family A + Family C deterministic-replay binding is an emergent system-level property: because every Verification Envelope is content-addressed and Trust-Block-bound, and because every three-stage Pipeline Run produces a Pipeline Completion Record referencing all three stage outputs, any auditor at any future date can deterministically replay the full ingestion-evaluation-admission chain from raw Witness Record through admitted Contribution Ledger Entry. This property does not exist in either Family alone; it emerges from the integration. The browser-extension embodiment (*QPN Catalyst Launch Plan & Rewards Framework* -32) makes this property concrete: a browser-extension-resident Sidecar's Verification Envelope at block height H, combined with the Pipeline's model-snapshot references at H, constitutes complete replay evidence for the contribution's admission decision.

Family B — Five Signal Input Mode Architecture

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework-2.

Field Summary

Authorization-layer taxonomy of contribution capture signals operating within the QPN.

Family B sits between Family A (Sidecar capture) and Family C (AI evaluation pipeline) in the Catalyst Network ingestion stack. Per the and, the Five Signal Input Modes constitute the participant-side authorization vocabulary by which the Sidecar's Listener Agent is permitted to attest to ambient signals at all. Without an active Signal Mode, the Listener Agent emits no Listener Attestations; with an active Signal Mode, the Listener Agent emits attestations bounded by the Mode's specific Trust Criteria. This makes Family B the architectural primitive that converts participant intent into protocol-level admission policy.

Problem Addressed

Contribution-capture systems treat all incoming signals under a single authorization regime, conflating explicit user action with ambient observation and third-party attestation. The result is either (a) over-restriction, in which ambient or third-party signals are excluded entirely, foreclosing valuable attribution pathways; or (b) over-collection, in which all signals are captured under blanket consent, defeating fine-grained authorization. No

prior system provides a structurally separated five-mode taxonomy with mode-specific Trust Criteria enforced by Proof-of-Trust.

The technical problem is that participants in a contribution-capture network do not have a single uniform consent posture: different contribution types warrant different ambient-signal capture rules, different temporal windows, and different downstream attribution permissions. A participant who is actively typing a document (Active Mode) consents to a different ambient envelope than a participant whose device is sitting idle on a desk in the same room as an executive meeting (Ambient Mode). Conventional consent systems collapse this nuance into a binary opt-in / opt-out which is insufficient for trust-verified attribution.

A second problem is that mode transitions must themselves be cryptographically attested. If an adversary could simply spoof a Mode transition ("the participant just entered Institutional Mode and consented to capture this meeting"), the entire Mode framework would be defeated. Family B addresses this by requiring each Mode transition to be Trust-Block-bound and Proof-of-Trust-verified, with deterministic-replay guarantees so that any downstream consumer of the Mode-tagged contribution can independently verify the Mode under which the contribution was admitted.

Solution Overview

The Five Signal Input Mode Architecture decomposes contribution capture into five non-overlapping modes — Active, Directed, Ambient, Institutional, and Evangelized — each governed by mode-specific Trust Criteria and authorized under a distinct Trust Block schema. A Mode Discriminator classifies each incoming signal before capture; mis-classified signals are rejected at the QPC boundary. The taxonomy enables fine-grained participant authorization (e.g., authorize Active and Directed; decline Ambient) while preserving the Settlement Controller's ability to reward contributions across all five modes when authorized.

The five Modes (Active, Directed, Ambient, Institutional, Evangelized) are not arbitrary partitions of the consent space; they correspond to distinct cryptographic admission policies in the Sidecar's message-flow constraints. Active Mode permits the broadest Witness capture but the narrowest Listener attestation; Directed Mode narrows Witness scope to explicitly-invoked input surfaces; Ambient Mode permits broader Listener attestation but Witness capture is suspended; Institutional Mode binds Witness and Listener attestations to a sponsoring organization's Trust Block; Evangelized Mode permits attribution-graph extension to introduced third parties under bounded reach.

Mode-tagged Verification Envelopes are deterministically replayable: any auditor with access to the Trust Block lineage and the Premium weight schedule can recompute the contribution's downstream attribution effects. This deterministic-replay property is what enables Family E's Catalyst Contribution Graph to be cross-verifiable across independent auditors, and what enables foreign-jurisdiction prosecutors to validate that a Mode-tagged contribution was correctly admitted at the time of capture.

Components

The Mode Discriminator, Mode Transition Manager, Mode-Bound Trust Criteria, and Mode-Tag Propagator are detailed below. Source corpus depth for each component is in (Five Signal Input Modes) and §4.4 (Mode-Trust-Block binding mechanics).

Active Mode

Explicit user-initiated contribution. Examples: deliberate authorship, voice command invocation, explicit submission. Trust Criteria: user-action signature plus session-context attestation.

Active Mode is implemented as a Trust-Criteria-bound Mode record with the following schema: mode_uuid, mode_class = ACTIVE, witness_capture_scope_predicate (the set of event classes admissible under Active Mode — typically the full POINTER / KEYSTROKE / DOCUMENT_EDIT / explicit-trigger event taxonomy), listener_attestation_scope_predicate (typically restrictive — only presence-class signals admissible, no aggregated summaries), mode_transition_entry_predicate (the predicate that fires Active Mode entry — typically participant-originated explicit invocation OR an authorized application context entering focus), mode_transition_exit_predicate (the predicate that fires Active Mode exit — typically a configurable inactivity timeout or explicit user action). Per, Active Mode is the broadest Witness-capture envelope but the narrowest Listener-attestation envelope, reflecting that participant attention is on the active task and ambient context is

less relevant. Mode-entry and Mode-exit events are themselves Trust-Block-bound and replayable; any auditor can verify which Mode was active at any specified ledger height.

Directed Mode

Prompted contribution where a system or peer requests the contribution and the user responds. Examples: survey response, agent-prompted clarification. Trust Criteria: prompt-context attestation plus user-response signature.

Directed Mode is a narrowed variant of Active Mode implemented with witness_capture_scope_predicate restricted to a specific input-surface scope — e.g., "only events targeting application window with title-pattern matching regex R" or "only events within DOM subtree of element E". This narrows the capture envelope to participant-directed contexts. Schema includes directed_scope_descriptor (a Trust-Block-anchored descriptor specifying the directed scope), directed_scope_validation_predicate (a predicate evaluated at each event to confirm scope adherence). Per, Directed Mode supports cases where the participant wants Witness capture limited to specific applications or contexts without fully exiting Active Mode for other contexts. Mode-transitions between Active and Directed are themselves Trust-Block-bound.

Ambient Mode

Passive observation under standing authorization. Examples: continuous health-vital sensor stream, ambient meeting context, location-class signal. Trust Criteria: standing-consent Trust Block referencing the specific ambient surface and decay-time.

Ambient Mode is the inverse-emphasis Mode relative to Active Mode: witness_capture_scope_predicate is fully suspended (no Witness Records emitted) while listener_attestation_scope_predicate is expanded to include the full ambient-signal taxonomy. Schema: ambient_signal_class_set (the bitmap of signal classes admissible under Ambient Mode — configurable per participant Trust Criteria), ambient_attestation_aggregation_window_ms (the time window over which ambient attestations are aggregated before emission — typical default 5000-60000 ms balancing freshness against attestation volume). Per and, Ambient Mode supports use cases where the participant is not actively engaged but ambient context (e.g., a meeting in the same room as the device, a presentation being attended) is valuable to attest. The structural Witness suspension invariant is critical: ambient context is observed and attested, but no content is captured.

Institutional Mode

Contribution routed through an enterprise's Privacy Network whose organizational authorization is bound by an Enterprise Trust Block. Examples: enterprise-credentialed employee contribution attributed to both employee and enterprise. Trust Criteria: dual-Trust-Block authorization (individual + organizational).

Institutional Mode binds Witness and Listener attestations to a sponsoring organization's Trust Block. Schema: sponsor_did (the sponsoring institution's decentralized identifier), sponsor_trust_block_uuid (the Trust Block authority under which institutional sponsorship is granted), institutional_scope_predicate (the scope of admissible Witness and Listener events under institutional sponsorship), sponsor_revocation_predicate (the predicate under which the sponsor may revoke Institutional Mode — typically a Trust-Block-bound sponsor-signature requirement). Per, Institutional Mode is the architectural primitive supporting Enterprise Privacy Networks: an enterprise employee participating in an Enterprise Privacy Network operates in Institutional Mode for enterprise-bound contributions; the enterprise's institutional Trust Block authorities govern the admission policy and attribution flow. Mode-entry requires both participant authorization AND sponsor authorization; sponsor revocation is unilateral by design (enterprises may revoke sponsorship instantly upon employment termination or compliance breach).

Evangelized Mode

Third-party attestation of a contribution by a participant who did not directly originate the captured signal. Examples: peer attestation of conference talk attendance, witness attestation of organizational outcome. Trust Criteria: witness Trust Block plus attested-party Trust Block presence (which may be a Manager-Originated DORMANT QPC, Family I).

Evangelized Mode supports attribution-graph extension to introduced third parties under bounded reach. Schema: evangelist_did (the participant operating in Evangelized Mode), evangelized_target_did_set (the set of

introduced third parties), evangelist_reach_bound (the maximum graph-traversal distance from the evangelist that Evangelized Mode attribution may extend — typical default ≤3 hops), evangelist_revocation_predicate (predicate under which Evangelized Mode linkage may be revoked by either party). Per and, Evangelized Mode is the architectural basis for community-formation patterns: a participant introduces a peer to the Catalyst Network, attributing the peer's subsequent contributions back to the evangelist under the bounded reach. The reach bound prevents unbounded attribution chains that could be gamed via sock-puppet networks; the bound is itself Trust-Block-anchored and adjustable only under explicit sponsorship-class Trust Criteria.

Mode Discriminator

Classification component operating within a Discriminator-scoped QPC. Receives candidate signals, applies a deterministic classification function combining signal-source attestation, payload schema, and contextual signature, and emits a Mode Tag bound to the signal. Mode Tag is verified by the receiving Settlement Controller before settlement issuance.

The Mode Discriminator is the protocol component evaluating incoming signals against the active Mode's admission predicates. Implementation: at each signal event arrival, the Discriminator: (i) loads the active Mode record for the originating Privacy Domain; (ii) evaluates the signal class against the Mode's witness_capture_scope_predicate and listener_attestation_scope_predicate; (iii) admits, routes, or rejects the signal accordingly; (iv) emits an Audit Log entry recording the discrimination outcome. The discrimination algorithm is deterministically replayable: any auditor with access to the Audit Log and the Mode record set can verify that each signal was correctly discriminated. Per, the Discriminator's rejection events are themselves Trust-Block-bound (an attempted out-of-Mode capture produces a rejection record), supporting compliance audits and Mode-discipline enforcement.

Process Flow

An incoming signal arrives at the Sidecar's Verification Bridge (Family A) carrying its origin envelope.

The Mode Discriminator extracts source attestation, payload schema, and contextual signature.

The Discriminator applies the deterministic classification function and emits a candidate Mode Tag.

The candidate Mode Tag is checked against the participant's currently-active Mode authorizations (the set of Modes the participant has Trust-Block-authorized for capture).

If the candidate Mode is authorized, the signal is admitted and the Mode Tag is attached as part of the Trust Block envelope; if not, the signal is rejected at the QPC boundary and no record persists.

Settlement Controller (Family G) reads the Mode Tag and applies mode-specific Premium weights from Family J during reward computation.

Alternative Embodiments

- **Domain-specific mode extension:** for healthcare embodiments, an additional Clinical Mode may be defined with HIPAA-bound Trust Criteria. The Mode Discriminator is parameterizable.
- **Per-organization mode customization**: an enterprise may add an Internal Mode to its Enterprise Privacy Network for contributions that do not propagate to the global Contribution Graph.
- **Conditional Mode embodiment:** Modes may be configured with conditional-activation predicates that fire automatically based on external context — e.g., "Active Mode activates automatically when participant device enters Privacy Domain context X AND when calendar context Y is detected". Conditional Mode transitions are themselves Trust-Block-bound; the conditional predicate is evaluated by the Mode Discriminator and produces an automated transition event without requiring explicit user invocation.
- **Multi-Mode concurrent embodiment**: A participant may operate in multiple Modes concurrently across different QPC-segregated contexts — e.g., Active Mode on the participant's primary work device and Ambient Mode on a peripheral device in the same physical environment. Each Mode is bound to its specific Sidecar QPC instance; the cross-Mode coordination is mediated through Trust-Block-bound cross-QPC message envelopes.

- **Mode-Inheritance embodiment:** Inherited Modes per Family O Quantum DNA: a participant's Mode configuration may be inherited from a parent Privacy Domain or parent participant Genome under the Recombination Policy. Inherited Mode configurations preserve the inheritance chain in the Mode record's parent_mode_uuid_set field, supporting cross-generational Mode-policy continuity.
- **Quantum Mode embodiment:** Modes may carry quantum-state semantics under the Family O Governance Superposition primitive: a participant may operate in a superposition of Active and Institutional Mode for contributions that simultaneously qualify as individual contributions and institutional contributions. The superposition resolution at the Contribution Ledger admission stage Trust-Block-anchors the resolved-Mode designation.
- **Stage-Gated Mode embodiment**: Mode availability may be gated by participant's Confidence Tier (Family E) or by stage progression: a participant in TIER_BRONZE may not have Evangelized Mode available; Evangelized Mode unlocks at TIER_SILVER or higher. Stage-gating is itself Trust-Block-bound and revocable upon tier regression.
- **Time-Windowed Mode embodiment**: Modes may carry explicit time-window validity: a participant may enable Institutional Mode for a specific window (e.g., 9am-5pm weekdays during employment) with automatic exit at window end. Window-bound Modes integrate with calendar-context observations and with sponsor-attestation validity periods.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Mode transitions:** (i) the proposed Mode must be in the participant's Trust-Criteria-authorized Mode set; (ii) the Mode-transition predicate must evaluate to true at the proposed transition block height; (iii) any sponsor (for Institutional Mode) or evangelist (for Evangelized Mode) must hold valid current authority.
- **Postconditions:** (i) the Mode-transition event is recorded on the Trust Ledger (Family D); (ii) the previous Mode's exit-handling completes (in-flight Witness Records are flushed per the previous Mode's policy); (iii) the new Mode's entry-handling completes (admission predicates are loaded; ambient attestation aggregation windows are reset if applicable).
- **State transitions:** Mode-state transitions follow the graph permitted by the Trust-Criteria-bound Mode Transition Policy: Active ⇄ Directed, Active ⇄ Ambient, Active → Institutional, Institutional → Active (under sponsor authorization), Active → Evangelized, Evangelized → Active.
- **Error handling:** unauthorized Mode-transition attempts emit Audit Log records and reject the transition; mode-entry-predicate evaluation failures trigger MODE_ENTRY_DENIED states.
- **Performance characteristics:** Mode transition evaluation is O(1) per transition; Mode-discrimination at each event is O(P) per P scope predicates evaluated, typically P ≤ 10 for standard configurations.

Cross-Family Integration

Upstream Dependencies. Family B operates against the QPN-enabled infrastructure of U.S. Patent No. 12,316,610 B1 (QPC, Privacy Domain, Trust Block, Trust Criteria, Proof-of-Trust, EasyAccess workflow threads); the Five Signal Input Modes are themselves Trust-Block-bound Mode records evaluated by the Mode Discriminator. Family B inherits the four-component Sidecar pattern from Family A as the operative substrate for Mode-bound capture: Mode admission policies bind to specific Family A component behaviors.

Downstream Consumers. Family J (Premium Framework) is the primary downstream consumer per Pattern #2 (the architectural analysis): the Mode-Tag attached to each Verification Envelope by Family A under the active Family B Mode propagates through Family C's Pipeline into the Premium Score Record, where Family J's Premium computation references the Mode-Tag as a Premium-dimension input. Family E (Contribution Graph) consumes Mode-Tagged edges differently: Evangelized Mode contributions establish bounded-reach evangelist-to-target attribution edges; Institutional Mode contributions establish sponsor-bound attribution edges. Family G (Settlement Controller) uses Mode-Tag to select the operative Witness Solicitation policy.

Lateral Interactions. Family B's Institutional Mode interoperates with the Family K Authorized Catalyst Proxy Addresses primitive — a sponsor's Institutional Mode authorization may be implemented as a Trust-Block-

bound ACPA delegation from the sponsor to the participant. Evangelized Mode interoperates with Family E Reputation Engine — the bounded-reach attribution graph extension is governed by the Reputation Engine's graph-traversal-depth-bound configuration.

Emergent System-Level Properties. The Family B + Family J Premium-weight propagation discipline produces an emergent property: a deterministic, replayable, and auditable mapping from participant Mode context at contribution time to settlement amount at issuance time. The Mode-Tag is bound at capture (Family A under Family B); the Mode-Tag-conditional Premium weight is bound at Pipeline run (Family C consulting Family J's Mode-conditional schedule); the settlement amount is bound at issuance (Family G consulting the Premium Score Record). An auditor reconstructs this chain by Trust-Block traversal without consulting any single trusted authority.

Family C — Three-Stage AI Evaluation Pipeline

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework-3.

Field Summary

Three-stage AI-mediated contribution evaluation pipeline operating within QPCs and bound by Trust Blocks.

Family C is the network-side evaluation surface of the Catalyst Network: it receives Verification Envelopes from the Sidecar (Family A) under their declared Signal Modes (Family B), applies a three-stage AI evaluation pipeline (Trust Validation, Content Categorization, Premium Computation), and emits AI Evaluation Records that the Settlement Controller (Family G) and the Contribution Graph (Family E) consume. Per and, the three-stage decomposition is essential: collapsing the three stages into a single model would defeat the deterministic-replay and protocol-grade auditability properties that the rest of the Catalyst Network depends upon.

Problem Addressed

Contribution attribution requires reliable evaluation of (a) what was contributed, (b) who contributed it, and (c) when and how durable the contribution is. Conventional pipelines either omit one of these dimensions entirely, or process them in an integrated model that defeats post-hoc auditability. No prior system constructs each evaluation stage as a Trust-Block-bound, deterministically-replayable model operating within a QPC, with cross-stage attestation suitable for §112-grade auditability.

AI-mediated evaluation systems in conventional architectures suffer from three structural defects that the three-stage pipeline addresses. First, single-model evaluation is non-deterministic in ways that defeat audit: a black-box LLM that takes a contribution as input and emits a Premium score as output cannot be replayed by an auditor without access to the original model weights, sampling seeds, and decoding parameters — none of which are guaranteed to be reproducible across model versions. Second, single-model evaluation conflates trust verification (was this contribution admitted under valid Trust Criteria?) with content analysis (what does this contribution actually say?), so a hallucination in the content layer can corrupt the trust layer. Third, single-model evaluation does not permit independent model replacement: upgrading the content model forces re-evaluation of all trust decisions, and vice versa.

The three-stage decomposition addresses all three: each stage has a bounded input/output contract; each stage's output is independently Trust-Block-bound; each stage can be independently replayed with cryptographic guarantees. This converts AI evaluation from a black-box function into a protocol-grade pipeline of auditable transformations.

Solution Overview

The Three-Stage AI Evaluation Pipeline executes three sequential AI evaluations within distinct QPCs, each producing a Trust-Block-bound attestation that is consumed by the next stage: (1) Semantic Classification determines content type and value; (2) Identity Enrichment links the contribution to a participant Trust Block; (3) Temporal Durability assigns a longitudinal weight based on observed contribution persistence and corroborating signals. Each stage employs the deterministic replay primitive from the November 18, 2025 AI Governance Provisional (§5.8) and the trust-weight engine (§5.10), enabling deterministic re-evaluation by any verifier holding the same Trust Block lineage.

Trust Validation (Stage 1) receives the Verification Envelope and validates its Trust Block lineage, Proof-of-Trust signature, and Mode-Tag consistency. Stage 1 emits a Trust Validation Record that is itself Trust-Block-bound. Content Categorization (Stage 2) receives the Validation Record and the original Verification Envelope, and applies content classifiers (modality, topic, claim structure) producing a Categorization Record. Premium Computation (Stage 3) receives the Categorization Record and applies the Premium weight schedule (Family J) to produce a Premium Score Record. Each stage's output is independently auditable, and the pipeline as a whole is deterministically replayable given the stage-specific model snapshots referenced in each Trust Block.

Manager-Discretion AI Model integration allows Catalyst Network Managers to substitute alternative stage-3 Premium models for specific Privacy Domains, subject to Trust Criteria constraints and Governance-Reserve-bound publication. This converts the Premium Framework from a fixed schedule into a parameterizable governance instrument while preserving the deterministic-replay and protocol-grade auditability properties.

Components

Detailed components follow. Sources: (three-stage pipeline architecture), §3.6 (Manager-Discretion model), §4.5 (Trust-Block-bound stage outputs), and (canonical stage definitions).

Semantic Classification Stage

AI model operating within a Classifier-scoped QPC. Inputs: contribution payload, Mode Tag (Family B). Outputs: Semantic Class (taxonomy from *QPN Catalyst Launch Plan & Rewards Framework*-30 Catalyst Contribution Categories), Class Confidence, Class Trust Block attestation. Uses deterministic replay so two verifiers receive identical Class outputs from identical inputs.

Stage 1 (Semantic Classification, equivalent to Trust Validation + Content Categorization in some terminology variants) is the first stage of the three-stage AI pipeline. Implementation: input is a Verification Envelope from the Sidecar Bridge (Family A); output is a Semantic Classification Record with schema: record_uuid, envelope_uuid_reference, semantic_class_set (multi-label classification across a Trust-Block-anchored class taxonomy — typical classes: TOPIC_CATEGORY, INTENT_CLASS, MODALITY_CLASS, CLAIM_STRUCTURE, ENTITY_REFERENCES), classification_confidence_set (per-class confidence values), model_snapshot_reference (the content-addressed reference to the classification model state used). The model snapshot reference is critical for deterministic replay: any auditor can re-evaluate the same Verification Envelope against the same model snapshot and verify identical output. Per, model-snapshot rotation is permitted but each rotation is Trust-Block-anchored; classifications produced under an old snapshot remain valid and replayable indefinitely.

Identity Enrichment Stage

AI model operating within an Identity-scoped QPC. Inputs: Semantic Class output, candidate participant Trust Block keys, Multi-Factor Identity Binding artifacts (*QPN Catalyst Launch Plan & Rewards Framework*-28). Outputs: bound Participant Identifier, Identity Confidence, Identity Trust Block attestation.

Stage 2 (Identity Enrichment) extends the Semantic Classification Record with identity-bound enrichments: participant reputation context (from Family E Reputation Engine), participant Privacy Domain membership set, participant Trust Block authority chain, and cross-Privacy-Domain reputation projection (if Multi-Genome State Isolation per Family O permits cross-domain visibility). Schema: enrichment_record_uuid, semantic_record_reference, participant_reputation_snapshot (Trust-Block-bound reputation value at the enrichment block height), participant_privacy_domain_set (the set of Privacy Domains in which the participant holds membership), cross_domain_reputation_projection (optional, Trust-Criteria-bound). Per, Identity Enrichment is the integration point between content classification and participant identity — it permits Stage 3 Premium computation to weight contributions by participant reputation without requiring the classification stage to know participant identity (separation of concerns).

Temporal Durability Stage

AI model operating within a Durability-scoped QPC. Inputs: Identity Enrichment output, historical contribution lineage from the Contribution Ledger (Family D), corroborating Cross-Verification signals (Family G). Outputs: Durability Weight (longitudinal contribution value), Durability Confidence, Durability Trust Block attestation.

Stage 3 (Temporal Durability, equivalent to Premium Computation in some terminology variants) computes the durability-weighted Premium Score for the contribution. Implementation: input is the Enrichment Record from Stage 2; output is a Temporal Durability Record with schema: durability_record_uuid, enrichment_record_reference, premium_score (rational ∈ [0, max_premium] reflecting the contribution's expected long-term value per the Premium schedule of Family J), compression_curve_position_reference (the Compression Curve position at the computation block height — Trust-Block-anchored), manager_discretion_adjustment_factor (rational ∈ [adjustment_floor, adjustment_ceiling] per Trust-Criteria bounds, permitting Manager-Discretion AI Model substitution), temporal_durability_score (rational reflecting projected long-term value retention vs. ephemeral signal). Per, the Temporal Durability Stage's output is the operative input to the Settlement Controller (Family G) for settlement-issuance computations.

Pipeline Coordinator

Orchestration component within a Coordinator-scoped QPC enforcing stage ordering, dependency satisfaction, and final emission of the composite Pipeline Attestation containing all three stages' outputs and cross-stage bindings.

The Pipeline Coordinator orchestrates the three-stage pipeline with explicit per-stage Trust-Block-bound output emission. Implementation: (i) receives Verification Envelope from Sidecar Bridge; (ii) submits to Stage 1, receives Semantic Classification Record, anchors output Trust Block; (iii) submits Semantic Record to Stage 2, receives Enrichment Record, anchors; (iv) submits Enrichment Record to Stage 3, receives Temporal Durability Record, anchors; (v) emits aggregated Pipeline Completion Record referencing all three stage outputs and binding them as a coherent pipeline run. Schema: pipeline_run_uuid, envelope_reference, stage1_output_reference, stage2_output_reference, stage3_output_reference, pipeline_status (enum: IN_PROGRESS, COMPLETED, FAILED_AT_STAGE_X). The Coordinator enforces strict ordering: a stage cannot be retried under a different model snapshot without producing a new Pipeline Run UUID; output records from a failed-then-restarted pipeline are explicitly distinguished from successful single-run outputs. Per, this ordering discipline prevents an attacker from cherry-picking favorable stage outputs across multiple retries.

Process Flow

An admitted signal carrying a Mode Tag is presented at the Pipeline Coordinator.

The Coordinator invokes Semantic Classification with deterministic seed derived from the signal's Trust Block lineage.

Upon completion of Semantic Classification, the Coordinator invokes Identity Enrichment, supplying the Semantic Class output as part of the input bundle.

Upon completion of Identity Enrichment, the Coordinator invokes Temporal Durability, supplying both prior stages' outputs.

The Coordinator assembles the Pipeline Attestation: a Trust-Block-wrapped tuple of (Semantic Class, Bound Identity, Durability Weight) with stage-specific confidences and cross-stage signatures.

The Pipeline Attestation is presented to the Settlement Controller (Family G) for settlement eligibility evaluation.

Alternative Embodiments

Parallel-stage embodiment: Identity Enrichment and Temporal Durability may execute in parallel after Semantic Classification, with the Coordinator joining results before emission.

Federated-model embodiment: each stage's AI model may be sourced from a different model provider, with model-provider attestation included in the stage Trust Block.

Reduced-confidence embodiment: when corroborating signals are insufficient for high-confidence Durability, the Pipeline may emit a provisional attestation flagged for retry upon receipt of additional Cross-Verification data.

Domain-specialized pipeline embodiment: The three-stage pipeline may be domain-specialized — e.g., a healthcare-specialized variant where Semantic Classification uses a healthcare-domain-tuned classifier (FDA-cleared clinical text classifier), Identity Enrichment incorporates HIPAA-bound participant identity attestations,

and Temporal Durability is indexed to clinical-evidence durability standards. Domain specialization is Trust-Block-anchored at the pipeline-configuration level.

Multi-model ensemble embodiment: Each stage may operate as an ensemble across multiple AI models with Trust-Block-bound aggregation rules — e.g., Stage 1 may run three semantic classifiers (general-purpose, domain-specialized, adversarial-robustness-tuned) with majority-voted output. Ensemble outputs preserve per-model-output references in the Pipeline Completion Record for downstream audit.

Adversarial-robustness embodiment: For contexts subject to adversarial attacks (election integrity, fraud detection, deepfake identification), the pipeline may include an adversarial-robustness sub-stage that evaluates inputs against adversarial-perturbation classifiers. Adversarial detection produces explicit ADVERSARIAL_FLAGGED records in the Pipeline Completion Record.

Streaming pipeline embodiment: For streaming contribution sources (live audio/video feeds, continuous sensor data), the pipeline may operate in streaming mode: stages evaluate incrementally over rolling windows rather than per-discrete-contribution. Streaming pipeline outputs are emitted at configurable cadence intervals with explicit window-boundary Trust-Block-anchors.

Privacy-preserving inference embodiment: Each stage may execute under privacy-preserving inference primitives — homomorphic encryption, secure multi-party computation, trusted-execution-environment-bound inference — such that the input contribution content is never exposed in plaintext during inference. Privacy-preserving inference produces output that is functionally identical to plaintext inference but preserves cryptographic confidentiality.

Verifiable-computation embodiment: Each stage may produce zero-knowledge-proof attestations of correct execution alongside its outputs, supporting third-party verification that the stage output was correctly computed from the input without requiring access to either input or model weights. Verifiable-computation attestations integrate with the Cross-Ledger Verifier (Family D).

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for AI pipeline ingestion:** (i) the Verification Envelope must pass Bridge-side signature verification at Catalyst Network ingress; (ii) the referenced Vault records must be accessible to the participant under the operative Trust Block; (iii) the model-snapshot references for all three stages must be valid at the pipeline-run block height.
- **Postconditions:** (i) a Pipeline Completion Record is emitted with references to all three stage outputs; (ii) the Pipeline Completion Record is anchored in the Authorization Ledger; (iii) downstream consumers (Family D Contribution Ledger admission, Family E edge construction, Family G settlement issuance) consume the Pipeline Completion Record.
- **State transitions:** Pipeline run lifecycle: STARTED → STAGE_1_COMPLETE → STAGE_2_COMPLETE → STAGE_3_COMPLETE → COMPLETED; failure transitions: STARTED → STAGE_X_FAILED → ABORTED or STARTED → STAGE_X_FAILED → RETRY (under new pipeline run UUID).
- **Error handling:** model-snapshot unavailability emits MODEL_SNAPSHOT_UNAVAILABLE states; classification confidence below minimum thresholds emits LOW_CONFIDENCE states with explicit confidence-floor records; stage-2 enrichment failures (e.g., participant identity disputes) emit ENRICHMENT_AMBIGUOUS states requiring resolution.
- **Performance characteristics:** each stage's evaluation cost is dominated by the underlying model inference cost; total pipeline latency is typically 10ms-2s per contribution depending on modality and model class; pipeline throughput is parallelizable across contributions at the Coordinator level.

Cross-Family Integration

Upstream Dependencies. Family C is the network-side evaluation surface for Verification Envelopes admitted via Family A under Family B Mode discipline. The three-stage Pipeline operates within QPN-enabled infrastructure per the §22.7 Wherein clause and depends on Trust-Block-anchored model snapshots for each stage. Family C consumes Family E reputation context at Stage 2 (Identity Enrichment) and Family J Premium schedule at Stage 3 (Temporal Durability).

Downstream Consumers. Pipeline Completion Records are the binding input for Family D Contribution Ledger admission, Family G Settlement Controller eligibility evaluation, Family J Premium Score computation, and Family E Contribution Graph edge construction. Per Pattern #3 (the architectural analysis), the Family C + Family D cross-stage Trust-Block lineage chain is essential: each Pipeline stage's output is Trust-Block-bound; the Three-Ledger / Two-Log records reference the Pipeline Completion Record; downstream queries (e.g., "what were the AI Pipeline outputs for this CLE?") traverse the cross-stage chain deterministically.

Lateral Interactions. Family C's Manager-Discretion AI Model primitive interoperates with Family J's Premium Framework — Manager-Discretion model substitutions adjust the Stage 3 Temporal Durability computation, which feeds into Family J's downstream Premium Score. Family C's Identity Enrichment stage interoperates with Family K's Reputation derivation outputs.

Emergent System-Level Properties. Two emergent properties: (a) the Pattern-#1 deterministic-replay property described in Family A's Cross-Family Integration extends through Family C — every contribution admission decision is reproducible from the Trust-Block-anchored Pipeline state; (b) the Pattern-#3 cross-stage Trust-Block lineage produces an audit chain spanning raw signal observation (Family A) → Mode-admission (Family B) → three-stage evaluation (Family C) → ledger admission (Family D) → graph emission (Family E) → Premium computation (Family J) → settlement issuance (Family G), with each link Trust-Block-anchored to the prior link. The audit chain is the protocol-grade replacement for trust in any single component.

Family D — Three-Ledger/Two-Log Architecture & Permanent Privacy Seal

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework-5, -27.

Field Summary

On-ledger record-keeping decomposition with Permanent Privacy Seal irrevocability primitive.

Family D is the protocol-level state primitive of the Catalyst Network. Per and, the Three-Ledger / Two-Log decomposition is the canonical state surface: the Contribution Ledger (what was admitted), the Settlement Ledger (what was issued), and the Trust Ledger (what authorities were exercised), supplemented by an Audit Log and an Event Log. The Permanent Privacy Seal is the cryptographic destruction primitive that allows participants to invoke an irreversible right-to-be-forgotten on Contribution Ledger entries without breaking downstream attribution chains.

Problem Addressed

Conventional ledgers conflate contribution records, authorization decisions, and settlement events into a single append-only structure, defeating plane separation and limiting cross-record verification. They also lack a structural irrevocability primitive — once a contribution is recorded, no mechanism enables a participant to permanently seal it against subsequent disclosure compulsion. Prior 'right-to-be-forgotten' implementations rely on operator discretion, which is insufficient for participants requiring cryptographic guarantees of permanence-or-destruction.

Conventional ledger architectures conflate state types that must remain cryptographically separable. Combining contribution records with settlement records prevents independent audit of either; combining authority records with content records prevents independent rotation of either; combining append-only logs with mutable state prevents principled state transitions. The Three-Ledger / Two-Log decomposition is the protocol-grade response: each ledger has bounded state semantics, each log has bounded event semantics, and each pair has explicit cross-reference contracts.

The Permanent Privacy Seal addresses a distinct problem: how can a participant invoke a right-to-be-forgotten on previously-attributed contributions without destroying downstream attributions whose own admissibility depends on the original contribution's existence? The Seal solution is to destroy only the participant's content-encryption key for a target Contribution Ledger entry; the entry's Trust-Block lineage and downstream attribution graph remain intact, but the content itself becomes cryptographically unreachable. This is a fundamentally different primitive than either deletion or anonymization, and is required by emerging absolute-privacy regulations in EU, Brazil, and South Korea.

Solution Overview

The Three-Ledger/Two-Log Architecture decomposes the record-keeping substrate into three Ledgers — Contribution, Authorization, and Settlement — plus two Logs — Data Plane Log and Control Plane Log. Each Ledger and Log operates within a plane-scoped QPC; cross-ledger verification primitives enforce consistency without requiring shared cryptographic state. The Permanent Privacy Seal is a Contribution Ledger primitive: a participant may seal a contribution record, which triggers cryptographic destruction of the participant's content-encryption key for that record. The sealed record persists as a Trust-Block header (settlement-bound) with content cryptographically unrecoverable; subsequent disclosure compulsion cannot retrieve content the key for which has been destroyed.

Each of the three ledgers operates as a Trust-Block-bound append-only structure with explicit cross-reference contracts to the other two. Contribution Ledger entries reference Trust Ledger authorities; Settlement Ledger entries reference Contribution Ledger entries; Trust Ledger entries reference both. The Two Logs (Audit and Event) provide temporal indexing and external-event correlation surfaces that do not themselves contain authoritative state but support auditor reconstruction.

The Permanent Privacy Seal primitive operates as a cryptographic destruction event on a participant-specific key, recorded as a Trust Ledger authority entry. Once sealed, the corresponding Contribution Ledger entry's content is unrecoverable even by the participant; downstream attribution graphs preserve the lineage reference but cannot resurface the content. This satisfies both protocol-grade auditability (the seal event is itself auditable) and absolute-privacy semantics (the content is destroyed).

Components

Each ledger and log is detailed below with its state semantics, cross-reference contracts, and Permanent Privacy Seal integration. Sources: (Three-Ledger architecture), §4.7 (Permanent Privacy Seal), and (canonical state surfaces).

Contribution Ledger

Append-only ledger recording each accepted contribution as a Trust-Block-bound entry: contribution identifier, participant Trust Block reference, Mode Tag, Semantic Class, Durability Weight, Permanent Privacy Seal status. Operates within a Contribution-Ledger-scoped QPC.

The Contribution Ledger is implemented as a Trust-Block-bound append-only structure recording each admitted contribution as a Contribution Ledger Entry (CLE). On-ledger CLE schema: cle_uuid (content-addressed identifier), pipeline_completion_record_reference (the Pipeline Coordinator output reference per Family C), originating_envelope_reference (the Sidecar Verification Envelope reference per Family A), originating_participant_did, originating_privacy_domain_uuid, admission_trust_block_uuid (the Trust Block under which admission was authorized), admission_block_height (the Settlement Ledger block height at admission), mode_tag (the active Signal Input Mode per Family B). Per and, the CLE explicitly does NOT contain the plaintext contribution content; content remains in the participant's Local Vault (Family A) and is referenced only by content-addressed identifier. This separation is critical for the Permanent Privacy Seal primitive: a seal destroys content access without destroying the CLE itself.

Authorization Ledger

Append-only ledger recording each authorization decision: requestor, requested capability, Trust Criteria satisfied, decision outcome (grant/deny), expiry. Operates within an Authorization-Ledger-scoped QPC.

The Authorization Ledger (also referred to as the Trust Ledger in some terminology variants) records each Trust Block authorization event. Schema: auth_record_uuid, auth_event_class (enum: TRUST_BLOCK_ISSUANCE, TRUST_BLOCK_REVOCATION, TRUST_BLOCK_AMENDMENT, ENCUMBRANCE_AUTHORITY, PERMANENT_PRIVACY_SEAL, MODE_TRANSITION, etc.), authorizing_party_did, authorized_party_did, auth_scope_descriptor (the scope of authority granted — Trust-Criteria-bound descriptor), auth_block_height, auth_expiry_block_height (optional; for time-bounded authorizations). Per and U.S. Patent No. 12,316,610 B1's Trust Block primitive, the Authorization Ledger is the canonical authority chain — every operation in the QPN that produces protocol-grade state change must reference a valid Authorization Ledger entry. Cross-ledger references from Contribution and Settlement ledgers always cite Authorization Ledger entries for their underlying authority.

Settlement Ledger

Append-only ledger recording each Exchange Token issuance: contribution lineage, recipient participant Trust Block, token amount, allocation waterfall position. Operates within a Settlement-Ledger-scoped QPC. Inherits issuance non-bypassability from Family G Settlement Controller.

The Settlement Ledger records each settlement issuance and economic transfer event. Schema: settlement_record_uuid, settlement_event_class (enum: CONTRIBUTION_SETTLEMENT, DERIVATIVE_ACCRUAL, LIQUIDITY_PROVIDER_DISTRIBUTION, ENCUMBRANCE_RELEASE, SHORTFALL_EVENT, ACCRUAL_RIGHTS_SWAP, etc.), referenced_cle_uuid (if applicable; reference to the underlying Contribution Ledger Entry), referenced_derivative_uuid (if applicable; reference to a Family L derivative record), payer_did, payee_did, settlement_amount, settlement_currency_class, settlement_block_height. Per, the Settlement Ledger is the operative input to all economic-attribution computations: Premium scoring (Family J), Senior Derivative payout (Family L), Backing Pool participation distribution (Family N), Accelerator MUM computation (Family M). Cross-ledger references from the Settlement Ledger to the Contribution Ledger and Authorization Ledger provide the deterministic-replay chain.

Data Plane Log

Plane-scoped operational log recording Data Plane traffic events (read/write/seal). Used for cryptographic destruction enforcement under Permanent Privacy Seal.

The Data Plane Log records data-plane events (Witness Record emissions, Listener Attestations, Verification Envelope constructions, Contribution Ledger admissions) at the temporal granularity required for external-event correlation. Schema: log_entry_uuid, log_event_class, log_event_payload_reference (reference to the canonical record in Contribution Ledger or elsewhere), log_block_height. Per and, the Data Plane Log explicitly does NOT hold authoritative state — it indexes the Contribution Ledger and supports external-system correlation queries (e.g., "give me all data-plane events in this hour") without requiring full Contribution Ledger traversal. The Two-Log decomposition (Data Plane + Control Plane) is essential for temporal-query performance at scale: query patterns that bind to event time rather than authority hierarchy route to the appropriate Log without traversing both.

Control Plane Log

Plane-scoped operational log recording Control Plane traffic events (authorization grants, policy compilations, Mode Tag emissions). Used for governance audit.

The Control Plane Log records control-plane events (Trust Block issuances, Mode transitions, Encumbrance Authority establishments, Permanent Privacy Seal invocations, governance amendments). Schema parallels Data Plane Log but with log_event_class enumerating control-plane event types. Per, the Control Plane Log indexes the Authorization Ledger and supports authority-audit queries (e.g., "give me all Trust Block issuances by sponsor S in this period") without requiring full Authorization Ledger traversal. The strict separation between Data Plane Log and Control Plane Log is a Three-Plane Architecture invariant: data-plane operations cannot directly observe control-plane events and vice versa, preserving the cryptographic separation between attribution and authority.

Cross-Ledger Verifier

Verification component within a Verifier-scoped QPC. Periodically attests cross-ledger consistency: each Settlement Ledger entry must reference a valid Contribution Ledger entry whose Authorization Ledger entries permit the cited settlement. Emits Cross-Ledger Attestations available to the Catalyst Network.

The Cross-Ledger Verifier validates the cross-ledger reference integrity across the Three-Ledger / Two-Log decomposition. Implementation: at any specified ledger height, the Verifier traverses cross-ledger references between Contribution → Authorization, Settlement → Contribution, Settlement → Authorization, and verifies that: (i) every referenced record exists at or before the verifier-invocation block height; (ii) every referenced record's Trust Block lineage is itself verifiable; (iii) no cross-ledger reference points to a revoked or amended record without explicit revocation/amendment-aware semantics. Per and, the Verifier emits a Cross-Ledger Verification Bundle that is itself Trust-Block-bound and reusable: a successful verification at block height H can be referenced by subsequent operations without re-verification, provided the operations do not span ledger state beyond H.

Permanent Privacy Seal

Primitive whose invocation cryptographically destroys the participant's content-encryption key for a target Contribution Ledger record. Implemented as a Trust-Block-bound seal instruction that triggers Data Plane Log-recorded key destruction; the Contribution Ledger record is converted to a sealed-header state.

The Permanent Privacy Seal primitive is implemented as a Trust-Block-bound cryptographic key destruction event. Implementation: at seal invocation, the participant signs a Seal Authorization record naming the target CLE UUIDs and the target Trust Block lineage; the Settlement Controller (Family G) emits a Trust Ledger entry of type PERMANENT_PRIVACY_SEAL referencing the target CLEs; the participant's wallet/key-management subsystem cryptographically destroys the Privacy-Domain-master-key-derived encryption keys for the target Trust Block lineage. Post-seal: the target CLEs remain in the Contribution Ledger (the lineage record is preserved); downstream attribution graphs in Family E preserve the lineage reference; but the plaintext content in the Local Vault becomes cryptographically unreachable even by the participant. Per, the seal is irreversible by design — the Privacy Domain master key derivation function is structurally one-way; key destruction is verifiable through key-attestation primitives that prove the key state cannot be reconstructed without breaking the underlying cryptographic primitives.

Process Flow

Upon Pipeline Attestation acceptance (Family C), the Contribution Ledger appends a new record bound to the participant's Trust Block.

Upon Settlement Controller (Family G) issuance, the Settlement Ledger appends a new entry referencing the Contribution Ledger record.

Authorization Ledger entries are appended throughout, recording every authorization decision in the lifecycle.

Cross-Ledger Verifier periodically scans all three Ledgers and emits Cross-Ledger Attestations.

If a participant invokes Permanent Privacy Seal on a Contribution Ledger record, the Data Plane Log records key destruction, the Contribution Ledger record transitions to sealed-header state, and any subsequent attempt to retrieve content fails cryptographically.

Alternative Embodiments

Multi-substrate ledger embodiment: each Ledger may be backed by a distinct underlying substrate (one Ledger on a permissioned chain, another on an enterprise's internal ledger, the third on a public chain), with Cross-Ledger Verifier emitting substrate-spanning attestations.

Sharded Contribution Ledger: for high-throughput deployments, the Contribution Ledger may be sharded by participant identity, with per-shard QPCs.

Time-locked seal embodiment: the Permanent Privacy Seal may be parameterized with a delay (e.g., seal triggers after 30 days), giving the participant a revocation window.

Federated ledger embodiment: The Three-Ledger / Two-Log structure may operate in a federated embodiment across multiple Catalyst Network instances: each federation member maintains its local Ledger instances with Trust-Block-bound cross-federation reference records. Federation-wide queries traverse cross-federation references under Trust-Criteria-bound federation membership verification.

High-throughput sharded ledger embodiment: For high-throughput contexts (population-scale Catalyst Network deployments), each Ledger may be sharded across multiple partitions with Trust-Block-bound partition assignment. Cross-shard references are explicitly recorded; cross-shard queries traverse shards under the operative shard-routing policy. Shard rebalancing is Trust-Block-bound and produces explicit rebalancing records.

Cross-substrate ledger replication embodiment: Each Ledger may be replicated across multiple substrate technologies (e.g., Hedera Hashgraph for low-latency primary recording, Bitcoin or Ethereum for high-assurance anchoring, IPFS for content-addressable storage of large referenced payloads). Cross-substrate replication preserves the cryptographic integrity chain through cross-substrate hash anchoring.

Delayed-finality embodiment: For contexts where immediate finality is not required (research observations, low-stakes contributions), Ledger admission may operate under delayed finality: admitted entries remain in PROVISIONAL state for a configurable confirmation window, transitioning to FINAL upon Trust-Block-bound confirmation. Provisional-state entries are queryable but downstream operations may opt to wait for finality.

Selective-disclosure ledger embodiment: Ledger entries may carry per-field disclosure policies: certain fields are public-readable; others require Trust-Block-bound disclosure authority. Selective disclosure supports compliance with jurisdiction-specific privacy regulations while preserving the integrity-anchor properties of the underlying entry.

Temporal-locking embodiment: Beyond the time-locked seal embodiment, Ledger entries may carry temporal-locking semantics: an entry may be admitted but invisible to downstream consumers until a specified block height. Temporal locking supports embargoed-disclosure use cases (e.g., research findings under publication embargo, financial disclosures under regulatory release-time discipline).

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for ledger entry admission:** (i) the entry's reference chain (Pipeline Completion → Envelope → Vault records → Trust Block lineage) must be fully verifiable; (ii) the admitting Trust Block must hold sufficient admission authority per the operative Trust Criteria; (iii) no cross-ledger conflict exists (e.g., the underlying CLE is not already sealed by a Permanent Privacy Seal).
- **Postconditions:** (i) the new entry is appended to the appropriate Ledger with a content-addressed UUID; (ii) the corresponding Two-Log indices are updated; (iii) the Cross-Ledger Verifier's incremental state is updated.
- **State transitions:** Each Ledger maintains append-only state; entry-level state is discrete (ADMITTED, REFERENCED_BY_DOWNSTREAM, SEALED — only for CLEs subject to Permanent Privacy Seal).
- **Error handling:** reference-chain verification failures reject the entry with explicit failure-reason records in the Audit Log; cross-ledger conflicts (e.g., admitting an entry that references a revoked Trust Block) emit LEDGER_CONFLICT records and require explicit resolution.
- **Performance characteristics:** entry-admission is O(R) per R cross-ledger references verified; periodic snapshot operations are O(N) per N entries since last snapshot, amortizable to O(1) per entry under incremental-snapshot maintenance. Per, ledger throughput is designed to support 10,000-100,000 entries/second per operational Catalyst Network instance under standard infrastructure provisioning.

Cross-Family Integration

Upstream Dependencies. Family D operates within QPN-enabled infrastructure (per the §22.7 Wherein clause, citing U.S. Patent No. 12,316,610 B1's QPC, Trust Block, Privacy Domain primitives), and consumes Pipeline Completion Records from Family C as the binding admission input. The Three-Ledger / Two-Log decomposition is the canonical state surface; the Permanent Privacy Seal primitive references the Family A Local Vault's Trust-Block-lineage-derived encryption keys.

Downstream Consumers. Every economically-significant Family in the QPN consumes Family D state: Family E (Contribution Graph) consumes Contribution Ledger Entries; Family G (Settlement Controller) consumes both Contribution and Authorization Ledger entries for settlement-eligibility evaluation and issuance; Family J (Premium) consumes Pipeline Completion Records anchored in the ledgers; Family L (Senior/Junior QPT Derivatives) records derivative claims on the Settlement Ledger; Family M (Stage-Differentiated Revert + MUM) computes MUM from Settlement Ledger flows; Family N (Liquidity Architecture) records Backing Pool participation on the Settlement Ledger; Family O (Quantum DNA/Genome) records Genome composition events on the Authorization Ledger; Family P (PPN Reproduction) records reproduction events on the Authorization Ledger.

Lateral Interactions. Family D's Cross-Ledger Verifier component interoperates with every consuming Family by emitting Trust-Block-bound Cross-Ledger Verification Bundles that downstream consumers reference instead of re-verifying the underlying cross-ledger references themselves. The Permanent Privacy Seal interoperates with Family E (graph edges referencing sealed CLEs preserve the lineage reference but the underlying content

becomes cryptographically unreachable) and with Family A (the Vault encryption keys for the sealed Trust Block lineage are destroyed).

Emergent System-Level Properties. The cross-surface Trust Block lineage chain — required by the Phase A reframing recommendation for Family D Claim 93 — emerges from the Three-Ledger / Two-Log decomposition's strict cross-reference contracts. No single Ledger or Log holds the complete state; the complete state is reconstructible only through traversal of cross-Ledger references. This property is what enables the Permanent Privacy Seal: sealing a Contribution Ledger Entry does not break downstream Settlement Ledger references because the Settlement Ledger references the CLE by content-addressed identifier, not by content.

Family E — Global Contribution Graph & Reputation Engine

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -7, -22, -23.

Field Summary

Global Contribution Graph assembly with DFS-traversal Reputation Engine and three-dimensional Quantum Reputation Model.

Family E is the attribution graph engine of the Catalyst Network. Per and, the Global Catalyst Contribution Graph is the protocol-grade structure that converts individual Verification Envelopes (Family A) into a cross-verifiable attribution surface that downstream allocations (Family J Premium, Family L Senior Derivatives, Family N Liquidity Pool) consume. The Reputation Engine derives durable participant-level reputation signals from the attribution graph.

Problem Addressed

Reputation systems built on centralized contribution data sacrifice participant privacy in exchange for graph completeness. Federated approaches preserve privacy but produce fragmented, incomparable reputation signals across silos. No prior system assembles a global contribution graph from heterogeneous Personal Archives without centralizing contribution content, while simultaneously emitting a structured reputation decomposition that separates contribution-derived, identity-derived, and behavior-derived signals.

Conventional attribution systems are non-portable, non-auditable, and non-resilient. Platform-bound attribution graphs (e.g., a social network's like-graph) cannot be migrated, cannot be independently verified, and cannot survive the platform's failure. Citation networks (e.g., academic citation graphs) lack cryptographic integrity guarantees and are vulnerable to citation manipulation. Web hyperlink graphs lack temporal integrity and lack participant-bound attribution semantics.

The Catalyst Contribution Graph addresses each: every edge is Trust-Block-bound (cryptographic integrity); every edge is participant-bound and Privacy-Domain-bound (auditable attribution semantics); the graph as a whole is deterministically reconstructible from the Contribution Ledger and Audit Log (replay-grade resilience); the graph is portable across Catalyst Network deployments under the Trust Block lineage discipline (cross-deployment portability).

Solution Overview

The Global Contribution Graph Assembly Engine constructs a participant-content-blind directed graph from Personal Archives held in heterogeneous storage substrates. Each Personal Archive emits a Graph Manifest (content-addressed identifiers, lineage references, Trust Block bindings) without disclosing payload content. The Assembly Engine merges manifests under cryptographic consistency rules, producing the global graph. The Reputation Engine executes Depth-First Search traversals over the graph to compute reputation scores across five confidence tiers, seven signal types, and six reputation badges. The Three-Dimensional Quantum Reputation Model decomposes the final reputation output into Contribution, Identity, and Behavioral dimensions — each independently weighted and Trust-Block-bound.

Edge construction follows a deterministic-assembly discipline: each edge references its source contribution by content-addressed identifier, its target contribution or participant by Privacy-Domain-bound identifier, and the Trust Block authorities under which the edge was admitted. Edge weights derive from the Premium schedule (Family J). Reputation derivation is itself protocol-grade: reputation scores are deterministically computable

from the graph state at any specified Trust Block height, and reputation transitions are auditable through the Audit Log.

Identity Resilience integration (Family K) is critical: a participant who loses primary identity credentials must be able to re-bind their attribution graph history to a new identity surface without breaking downstream attributions or losing reputation. The Reputation Engine's deterministic-derivation property combined with the Multi-Factor Identity Binding primitive enables this re-binding under Trust Criteria constraints.

Components

Components are detailed below: Edge Constructor, Graph Index, Reputation Engine, Re-Binding Authority. Sources: (Contribution Graph), §3.7 (Reputation derivation), §3.8 (Identity Resilience), and

Graph Manifest Emitter

Personal-Archive-resident component emitting content-addressed graph nodes and edges. Each node carries Trust Block reference and Durability Weight; each edge carries Cross-Verification (Family G) attestation. Payload content is not emitted.

The Graph Manifest Emitter constructs Contribution Graph edge records and emits them to the Assembly Engine. Implementation: at each admitted CLE event, the Emitter constructs candidate edges per the Edge Construction Schema: edge_uuid (content-addressed), source_cle_uuid_reference, target_cle_uuid_reference OR target_participant_did, edge_class (enum: CITATION, REPLY, QUOTE, EXTENSION, ENDORSEMENT, INTRODUCTION, COLLABORATION, etc.), edge_trust_block_uuid (the Trust Block under which the edge was admitted), edge_weight_initial (initial weight per the Premium schedule at edge-creation block height — typically subject to Compression Curve update via Family J). Per and, the Emitter's edge-construction algorithm is deterministically replayable: given the CLE inputs, the operative edge taxonomy, and the Premium schedule snapshot, any auditor can recompute the emitted edges.

Assembly Engine

Operates within an Assembly-scoped QPC. Receives Graph Manifests from many Personal Archives, applies cryptographic consistency rules (Trust Block lineage verification, edge-attestation verification), and emits the canonical Global Contribution Graph.

The Assembly Engine accumulates edge records into the Global Contribution Graph state and maintains the graph indices supporting downstream queries. Implementation: the Engine maintains: (i) an append-only edge log indexed by source and target CLE UUIDs; (ii) a participant-centric edge index supporting per-participant outbound and inbound edge queries; (iii) a temporal index supporting time-bounded graph queries; (iv) an edge-class index supporting class-filtered queries. Per, the Engine's deterministic-assembly property is critical: any auditor with access to the edge log and the index-construction algorithm can independently reconstruct all four indices at any specified block height. The Engine emits a periodic Trust-Block-bound Graph State Snapshot record (typical frequency: hourly to daily, configurable) supporting efficient point-in-time graph queries without full edge-log replay.

Reputation Engine — Graph Traversal Component

Executes Depth-First Search traversals over the Global Contribution Graph from designated source nodes. Aggregates traversal results across the seven signal types (contribution volume, durability decay, cross-verification density, peer-attestation frequency, settlement realization, role-context fit, anti-collusion indicator).

The Reputation Engine's Graph Traversal Component computes participant reputation as a deterministic function of the participant's in-graph position. Implementation: at reputation-evaluation time, the Component traverses inbound edges (citations, endorsements, etc.) from the target participant through a configurable max-depth bound (typical: 3-5 hops), applying per-edge-class weight multipliers and depth-decay factors. Schema: reputation_query_uuid, target_participant_did, evaluation_block_height, traversal_depth_bound, edge_class_weight_map, depth_decay_function_reference, traversal_raw_score. Per and §3.7, the traversal algorithm is parameterized by Trust-Block-anchored configuration: weight maps and decay functions are immutable for the evaluation block height, ensuring deterministic-replay auditability.

Reputation Engine — Confidence Tier Component

Applies a five-tier confidence classification (Tier 1: corroborated, Tier 2: peer-attested, Tier 3: durably-active, Tier 4: provisional, Tier 5: introductory) to each computed score.

The Confidence Tier Component converts raw graph-traversal reputation scores into discretized Confidence Tiers (e.g., TIER_BRONZE, TIER_SILVER, TIER_GOLD, TIER_PLATINUM with configurable thresholds). Schema: tier_assignment_uuid, target_participant_did, evaluation_block_height, raw_score_input, tier_threshold_schedule_reference, assigned_tier. Per, tier thresholds are Trust-Block-anchored and configurable; threshold rotation is permitted but each rotation produces a new threshold-schedule version, and tier assignments under prior versions remain valid for their issuance period. The discretization is structural rather than presentational: downstream consumers (Family J Premium Schedule, Family L Senior Derivative eligibility, etc.) consume the Confidence Tier directly rather than the raw score, supporting protocol-grade tier-based decisioning.

Reputation Engine — Badge Component

Emits six reputation badges (Founder, Steward, Connector, Sentinel, Adept, Apprentice) based on threshold combinations across the seven signals and five tiers.

The Badge Component awards Trust-Block-bound badges for specific contribution patterns or achievements. Schema: badge_uuid, badge_class (enum across a Trust-Block-anchored badge taxonomy — typical classes: FIRST_CONTRIBUTION, MILESTONE_100_CITATIONS, PIONEER_PARTICIPANT, DOMAIN_EXPERT, CROSS_DOMAIN_BRIDGE, etc.), awarded_to_participant_did, award_block_height, badge_award_predicate_reference (the Trust-Block-anchored predicate that triggered the award). Per, badges are non-transferable by default; transferability may be enabled for specific badge classes under explicit Trust-Block-bound transfer rules. Badges are durable across reputation tier rotations and survive Permanent Privacy Seal events (the badge persists; only the specific underlying contributions are sealed).

Three-Dimensional Output Component

Decomposes reputation output into Contribution Dimension (graph-derived signal weight), Identity Dimension (Identity Trust Block density), and Behavioral Dimension (compliance/anti-collusion signals). Emits Trust-Block-bound Reputation Vector consumable by Family J Premium computation.

The Three-Dimensional Output Component emits participant reputation as a three-dimensional vector: (Confidence Tier, Badge Set, Numerical Score). Schema: output_record_uuid, target_participant_did, evaluation_block_height, confidence_tier, badge_set, numerical_score. Per and §3.7, the three-dimensional output supports diverse downstream consumer requirements: Family J Premium Schedule consumes Confidence Tier for tier-bucketed multipliers; Family L Senior Derivative eligibility consumes Badge Set for badge-gated access; Family N Backing Pool participation-rights allocation consumes Numerical Score for fine-grained weighting. The output is Trust-Block-bound and re-computable at any specified block height from the underlying graph state and configuration schedules.

Process Flow

Each Personal Archive emits its current Graph Manifest on a schedule or on Trust-Block-bound request.

The Assembly Engine collects manifests, verifies cryptographic consistency, and incrementally updates the Global Contribution Graph.

The Reputation Engine, on traversal trigger (per-participant query or scheduled batch), executes DFS from the relevant source nodes.

Traversal outputs are aggregated by signal type and assigned a confidence tier.

Badge emission applies the threshold combinations.

The Three-Dimensional Output Component decomposes the final reputation into three independently-weighted dimensions and emits the Trust-Block-bound Reputation Vector.

Alternative Embodiments

Privacy-preserving aggregation embodiment: the Assembly Engine may operate under multi-party computation or zero-knowledge proof regimes for additional privacy guarantees over Graph Manifests.

Time-windowed reputation embodiment: the Reputation Engine may compute reputation over a sliding time window, with windowed Durability decay.

Cross-graph reputation embodiment: where multiple Global Contribution Graphs coexist (e.g., per-jurisdiction or per-Accelerator), the Reputation Engine may compute cross-graph reputation with explicit graph-context attestation.

Domain-stratified reputation embodiment: Reputation may be stratified by domain: a participant may carry distinct reputation scores within healthcare, finance, scientific research, and other domain-specific contexts, with cross-domain reputation projection (Family C Stage 2 Identity Enrichment) governed by Trust-Criteria-bound cross-domain rules. Domain stratification supports cases where reputation in one context should not automatically convey reputation in unrelated contexts.

Adversarial-resistant reputation embodiment: Reputation computation may incorporate adversarial-resistance heuristics: detection of Sybil-style sock-puppet networks via graph-topology analysis, detection of coordinated-citation rings via temporal correlation analysis, detection of reputation-laundering via cross-Privacy-Domain influence flow analysis. Adversarial-detection outputs feed into the Reputation Engine as explicit detection records, reducing the affected participants' reputation under Trust-Block-bound adjustment rules.

Multi-stakeholder reputation embodiment: Reputation may be computed from multiple-stakeholder perspectives: a participant may carry distinct reputation projections for individual peers, institutional sponsors, regulatory authorities, and general-public observers, each indexed to the operative stakeholder class's weight schedule. Multi-stakeholder reputation supports cases where the same underlying contribution graph is consumed by different parties with different evaluation priorities.

Liquid-democracy reputation embodiment: Reputation may flow under delegation patterns: a participant may delegate reputation-influence rights to another participant under Trust-Block-bound delegation records, supporting liquid-democracy-style influence propagation. Delegation is revocable under Trust-Criteria-bound revocation rules; delegation chains are auditable through the cross-reference chain to Trust Ledger entries.

Decay-function reputation embodiment: Reputation may incorporate temporal-decay functions: older contributions contribute reduced reputation weight over time, with decay function parameters Trust-Block-anchored at evaluation time. Decay supports cases where reputation should reflect recent activity rather than historical accumulation. Decay-function parameters are configurable per Privacy Domain Trust Criteria.

Cross-Family-graph reputation embodiment: Reputation may aggregate signals across multiple Family graphs: contributions (Family E Contribution Graph), governance participations (Family K Behavioral Activation graph), liquidity provisions (Family N participation graph), capital deployments (Family L Senior Derivative holdings). Cross-graph aggregation is Trust-Block-bound and supports holistic reputation projections that no single-graph evaluation would produce.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for graph edge admission:** (i) both source and target CLEs (or participants) must exist and be valid at the admission block height; (ii) the edge must be Trust-Block-bound to a valid Trust Block lineage; (iii) the edge class must be in the Trust-Block-anchored edge taxonomy.
- **Postconditions:** (i) the edge record is appended to the Graph Manifest; (ii) the Assembly Engine's indices are updated; (iii) downstream reputation queries reflect the new edge at the next evaluation.
- **State transitions:** edge state is discrete (ACTIVE, REVOKED) — revocation requires Trust-Block-bound revocation authority and is itself recorded in the graph (a revoked edge's existence is preserved for audit). Reputation tier transitions: participant tier may transition discretely between adjacent tiers (e.g., SILVER → GOLD) upon threshold satisfaction; cross-tier jumps are permitted but logged with explicit threshold-satisfaction records.

- **Error handling:** missing source/target CLE references emit EDGE_REFERENCE_INVALID records; cyclic edge constructions (self-citation, immediate-bidirectional citation) are permitted per the taxonomy but may carry reduced or zero edge weight per the Premium schedule.
- **Performance characteristics:** edge admission is O(1) per edge; reputation computation is O(N × D) per N source edges traversed at depth D; snapshot-based reputation queries are O(1) (precomputed snapshots). Per, the Engine is designed to support graphs of 10^9+ edges with sub-second reputation-query latency under standard infrastructure.

Cross-Family Integration

Upstream Dependencies. Family E consumes admitted Contribution Ledger Entries from Family D as the canonical edge endpoints, Pipeline Completion Records from Family C as edge-class and edge-weight inputs, and Mode-Tags from Family B as edge-attribution-context inputs. Edge construction operates within QPN-enabled infrastructure per the §22.7 Wherein clause; edge admission is Trust-Block-bound.

Downstream Consumers. Family J (Premium Framework) consumes the Contribution Graph for graph-based Premium computations (e.g., Synthesis Premium rewards contributions integrating multiple prior contributions per graph topology; Audience Premium rewards contributions with high subsequent-citation density). Family K (Behavioral Activation, Reputation & Identity Resilience) consumes the Reputation Engine's output for Level transitions, Badge awards, and Identity Binding policy decisions. Family G (Settlement Controller) consumes Reputation outputs through the Witness Trust Evaluation component and through Settlement Eligibility Predicates that gate by Confidence Tier. Family L (Senior/Junior QPT Derivatives) consumes Reputation outputs through derivative-eligibility predicates.

Lateral Interactions. Family E's Identity Resilience integration with Family K is the architectural primitive for the Pattern #4 cross-family interaction (the architectural analysis): the Reputation Engine's deterministic-derivation property combined with the Multi-Factor Identity Binding primitive enables a participant who loses primary identity credentials to re-bind their attribution graph history to a new identity surface. Re-binding operations are Trust-Block-bound; the graph state is preserved across identity transitions.

Emergent System-Level Properties. The Family E + Family K graph re-binding property is an emergent property: the Contribution Graph's deterministic-assembly discipline (every edge is Trust-Block-bound and replayable) combined with the Multi-Factor Identity Binding's k-of-n-threshold authentication (Family K) produces a system in which credential loss does not entail attribution history loss. This emergent resilience is essential for population-scale Catalyst Network adoption — without it, the network would face an attrition vector tied to credential-loss events. The Phase A reframing recommendation for Family E Claim 117 (deterministic assembly discipline for edge construction) reinforces this property.

Family F — Personal Archive, CCP & Multi-Substrate Persistence

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -8, -9.

Field Summary

Personal Archive (.qpn) file-format primitive with Contributor Intelligence Module (CCP) and Multi-Substrate Persistence Router/Adapter pattern.

Family F is the participant-side persistence layer of the Catalyst Network. Per and, the Personal Archive and Catalyst Continuity Profile (CCP) provide multi-substrate persistence for participant-owned content with cryptographic continuity guarantees across substrate transitions. This is the architectural primitive that allows participants to relocate, replicate, or migrate their content holdings without breaking the Trust Block lineage that anchors downstream attributions.

Problem Addressed

Personal data-storage systems either (a) lock participants into a single substrate (cloud silo) at substantial portability cost, or (b) provide raw export without preserving the cryptographic bindings and Trust Block lineage required for verified reuse. They also lack a local analytics primitive that can operate over the participant's contribution corpus without exfiltrating data to the operator's analytic infrastructure. The absence of a

substrate-agnostic persistence layer leaves participants exposed to vendor lock-in and substrate-specific availability failures.

Conventional personal-data architectures bind content to a single substrate (a specific cloud provider, a specific local device, a specific encrypted volume) such that substrate failure or substrate migration destroys the content's authentication chain. The Personal Archive addresses this with substrate-agnostic content addressing combined with substrate-specific Trust Block bindings that can be rotated under Trust Criteria.

A second problem is that participant-side content holdings have heterogeneous performance requirements: high-frequency state (active drafts, cursor positions) requires low-latency persistence; medium-frequency state (recently-completed contributions) requires moderate-latency replication; archival state (historical contributions, audit trails) requires high-durability long-tail storage. The Multi-Substrate Persistence layer addresses this by partitioning content across substrate tiers under deterministic policies, with the Catalyst Continuity Profile providing a unified cryptographic continuity guarantee across tiers.

Solution Overview

The.qpn Personal Archive is a canonical file-format primitive comprising (a) an encrypted contribution blob set, (b) Trust Block lineage manifest, (c) Graph Manifest (Family E), (d) participant-control metadata, and (e) substrate-binding attestations. The Contributor Intelligence Module (CCP) is a local analytics engine operating within a CCP-scoped QPC over the.qpn archive; the CCP emits local insights to the participant without ever exporting raw contribution content. The Multi-Substrate Persistence Architecture comprises a Persistence Router that consults a Substrate Policy and dispatches reads/writes to one or more substrate Adapters (local disk, cloud, IPFS, federated storage), with substrate-agnostic Trust Block enforcement preserving cryptographic identity of the archive across substrate transitions.

Content within the Personal Archive is content-addressed under quantum-safe hash functions specified by Trust Criteria. Each content object has a Trust Block lineage record that is rotatable: when content migrates between substrates, a new Trust Block lineage record is emitted referencing the prior lineage, such that the full content history remains auditable but the operative storage substrate can change.

The Catalyst Continuity Profile is a participant-specific configuration object that specifies which content classes live on which substrate tiers, under which Trust Criteria, with which rotation schedules. Updates to the CCP are themselves Trust-Block-bound, allowing participants (and authorized successors under Family K Identity Resilience) to manage substrate allocations as a protocol-grade governance act rather than as an unaudited storage operation.

Components

Components detailed below: Personal Archive, Catalyst Continuity Profile, Multi-Substrate Persistence Layer, Substrate Transition Manager. Sources:, §4.9, and

.qpn File Format

Container format whose normative structure comprises an encrypted contribution blob set (one blob per Contribution Ledger entry), Trust Block lineage manifest (referenced lineages), Graph Manifest (Family E), participant control metadata (consent grants, Permanent Privacy Seal states), and substrate-binding attestations (per-substrate availability proofs).

The.qpn file format is a content-addressed container format with the following on-disk schema: a fixed-length header (magic bytes "QPN1", version, content-hash-algorithm-identifier, total-payload-length), followed by a Trust-Block-bound manifest section (entity binding, schema version, parent.qpn references for Multi-Substrate Persistence chaining), followed by a payload section (AEAD-wrapped content under a Privacy-Domain-master-key-derived encryption key per Trust-Criteria-specified KDF). Container integrity is enforced by a tail authentication tag (typically 128-bit AES-GCM-SIV or ChaCha20-Poly1305 tag per Trust Criteria) covering the full header + manifest + payload. Per, the format supports incremental append (the manifest's append_chain_ref field references a prior.qpn container hash, supporting incremental versions without full re-write) and substrate-transition records (when content migrates between substrates, a new.qpn record is emitted referencing the prior substrate's container hash, preserving lineage).

Contributor Intelligence Module (CCP)

Local analytics engine operating within a CCP-scoped QPC. Reads.qpn archive contents (with participant's key, accessible locally), computes participant-side insights (contribution volume by Mode, Durability trends, reputation contribution decomposition), and emits insights only to the participant's local interface.

The Contributor Intelligence Module (CCP — Catalyst Continuity Profile) is a participant-specific configuration record with schema: ccp_uuid, participant_did, content_class_substrate_map (mapping from content class to substrate tier — e.g., {ACTIVE_DRAFT: LOCAL_DISK, RECENT_COMPLETED: CLOUD_STORAGE, ARCHIVAL: IPFS}), rotation_schedule_set (per-class Trust-Block-anchored rotation policies — e.g., "ACTIVE_DRAFT content migrates to RECENT_COMPLETED tier after 7 days inactivity"), trust_criteria_per_substrate (per-substrate Trust Criteria specifying admissible substrate operations). Per and, CCP modifications are Trust-Block-bound governance acts; the operative CCP at any block height is deterministically reconstructible from the Authorization Ledger. Successor-bound CCP modifications (under Family K Identity Resilience) follow the delegation primitives' Trust-Block-anchored authority chain.

Persistence Router

Routing component within a Router-scoped QPC consulting the participant's Substrate Policy on each read/write. The Policy specifies primary substrate, replication substrates, and migration rules; the Router enforces them.

The Persistence Router implements the content-class → substrate-tier routing decisions specified in the CCP. Routing algorithm: at each.qpn-write request, the Router (i) classifies the content per the operative content-class taxonomy; (ii) consults the CCP for the target substrate tier; (iii) invokes the corresponding Substrate Adapter (Local Disk, Cloud, IPFS, Federated); (iv) records the routing decision as a Trust-Block-bound Routing Record. Routing Record schema: routing_record_uuid, qpn_container_hash, target_substrate_class, routing_decision_predicate_reference, routing_block_height. Per, Router decisions are deterministically replayable: any auditor with access to the Routing Records and the operative CCP can verify each decision.

Substrate Adapter — Local Disk

Adapter for filesystem persistence. Enforces local-only access at the QPC boundary.

The Local Disk Substrate Adapter provides.qpn container persistence on participant-local storage with platform-specific encrypted-disk integration (FileVault on macOS, BitLocker on Windows, dm-crypt on Linux, file-based encryption on iOS/Android). Adapter schema: adapter_uuid, substrate_class = LOCAL_DISK, disk_encryption_attestation_reference (platform-attested encryption state), mount_point_descriptor, quota_bound_bytes. Per, the Adapter supports concurrent access by multiple QPCs on the same device through QPC-bound filesystem-namespace isolation; cross-QPC access requires explicit Trust-Block-bound cross-namespace authorization.

Substrate Adapter — Cloud Storage

Adapter for cloud object storage (S3, GCS, Azure Blob, etc.). Wraps requests in cloud-provider authentication while preserving the Trust Block envelope at the archive layer.

The Cloud Storage Substrate Adapter provides.qpn container persistence on cloud-resident object storage (S3-compatible APIs, Azure Blob Storage, GCS) with client-side-encrypted-at-rest discipline: cloud-resident bytes are AEAD-wrapped under participant-controlled keys before transmission; cloud provider has no plaintext access. Adapter schema: adapter_uuid, substrate_class = CLOUD_STORAGE, cloud_provider_descriptor, bucket_or_container_descriptor, client_side_encryption_key_ref (reference to participant-controlled wrapping key — never the wrapping key itself), compliance_attestation_set (jurisdiction-specific compliance attestations for the cloud region: e.g., GDPR for EU regions, HIPAA-BAA for US healthcare, China data localization for CN regions). Per, multi-region replication is supported through Trust-Block-bound replication policies.

Substrate Adapter — IPFS / Content-Addressed Storage

Adapter for content-addressed substrates with merkle-verification of retrieved content.

The IPFS / Content-Addressed Storage Adapter provides.qpn container persistence on content-addressable distributed storage networks (IPFS, Filecoin, Arweave). Adapter schema: adapter_uuid, substrate_class = CAS, cas_protocol_descriptor (IPFS/Filecoin/Arweave-specific protocol parameters), pinning_service_set (the set of

pinning services contracted for persistence guarantees), replication_policy_descriptor. Per, content-addressing aligns naturally with.qpn's content-hash-bound integrity model: the IPFS CID and the.qpn container hash can be the same value, eliminating cross-substrate integrity reconciliation. Adapter operations are slower-latency than Local Disk or Cloud Storage but provide stronger long-tail persistence guarantees.

Substrate Adapter — Federated Storage

Adapter for participant-controlled federated storage networks (e.g., personal cloud, family-shared NAS) with peer-to-peer access negotiated through Trust Block-bound capability tokens.

The Federated Storage Adapter supports persistence across organizationally-distinct federation nodes — research consortia, healthcare data trusts, journalism collectives. Adapter schema: adapter_uuid, substrate_class = FEDERATED, federation_descriptor (the federation's membership and governance configuration), node_set (participating federation nodes), quorum_policy (the policy specifying minimum-N nodes for write acknowledgment and read availability), federation_trust_block_uuid (the federation's governing Trust Block). Per, federated storage supports cases where neither a single participant nor a single commercial cloud provider is the appropriate persistence custodian — research data, collective archives, public-interest content.

Process Flow

On contribution creation, the Witness Agent (Family A) emits an encrypted blob; the Local Vault (Family A) requests Persistence Router write.

The Persistence Router consults the participant's Substrate Policy and dispatches writes to the configured primary substrate plus any replication substrates.

On CCP analytic invocation, the CCP reads.qpn archive contents via the Persistence Router and computes local insights within its QPC; outputs are displayed only to the participant.

On substrate migration (participant changes Substrate Policy), the Persistence Router orchestrates copy from old substrate to new while preserving Trust Block envelopes and emitting a Migration Attestation referencing both substrates' attestations.

On Graph Manifest emission (Family E request), the.qpn archive emits its Graph Manifest section only; the encrypted contribution blob set is not transmitted.

Alternative Embodiments

Hybrid-substrate embodiment: hot data on local disk, archival data on cloud, with the Router enforcing access-pattern-driven migration.

Multi-party-controlled archive embodiment: a household or small organization may operate a shared.qpn archive whose Substrate Policy requires multi-party authorization for substrate migration.

Quantum-safe-only embodiment: in deployments requiring post-quantum cryptography exclusively, the cloud and federated substrate Adapters apply additional post-quantum wrapping per the November 2025 Global Quantum-Safe Cybersecurity Provisional.

Inheritance-aware persistence embodiment: The CCP may be configured to inherit substrate-tier mappings from a parent CCP (per Family O Quantum DNA inheritance) — supporting cases where a descendant Privacy Domain initially uses its parent's persistence configuration and gradually diverges. Inheritance chains are Trust-Block-bound and traceable through Genome lineage.

Tiered-archive embodiment: Beyond hot/cold partition embodiments, an N-tier archival hierarchy may be configured: Tier 0 (in-memory cache, sub-millisecond access), Tier 1 (Local Disk, single-millisecond), Tier 2 (Cloud Storage, tens of milliseconds), Tier 3 (CAS / IPFS, hundreds of milliseconds), Tier 4 (deep-archive cold storage — e.g., Glacier, tape archives — multi-second access). The CCP specifies per-content-class tier residency and migration thresholds.

End-of-life disposition embodiment: The CCP may specify end-of-life disposition rules: upon participant explicit retirement request, or upon Trust-Block-bound inheritance-event firing,.qpn containers transition through pre-

specified disposition flows — public-archive publication, family-trust-bound inheritance to designated successors, complete cryptographic destruction (Permanent Privacy Seal at scale), or research-trust handover.

Compliance-zone embodiment: Substrate Adapters may be partitioned by compliance zone: an EU-GDPR-zone adapter set, a US-HIPAA-zone adapter set, a China-data-localization-zone adapter set, etc. Per-zone Trust Criteria forbid cross-zone migrations without explicit Trust-Block-bound cross-zone authorization. The Persistence Router consults zone designations before substrate selection.

Verifiable-storage embodiment: Substrate Adapters may emit verifiable-storage proofs — Filecoin Proof-of-Spacetime, Arweave SPoRA proofs, or proprietary verifiable-storage primitives — anchored as Trust-Block-bound storage attestations. Verifiable-storage proofs provide third-party verification that storage commitments are being honored without requiring participant-side verification.

Cross-participant collaboration embodiment: Multiple participants may share a unified persistence layer with cross-participant access controls: a research collaboration's shared.qpn containers may grant designated co-collaborators read access via Trust-Block-bound access-grant records, with the underlying encryption keys remaining individually-derived (no shared master key). Cross-participant access is auditable through the Trust Ledger.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for.qpn-write operations:** (i) the participant's CCP must be loaded and current; (ii) the target Substrate Adapter must report HEALTHY state; (iii) the participant's encryption keys must be derivable from the Privacy Domain master key.
- **Postconditions:** (i) the.qpn container is written to the target substrate with integrity-tag verification; (ii) the Persistence Routing Record is emitted Trust-Block-bound; (iii) substrate-specific replication / pinning operations complete per the operative policy.
- **State transitions:**.qpn container state transitions: PENDING_WRITE → WRITTEN → REPLICATED → ACTIVE → ARCHIVED → MIGRATED → RETIRED.
- **Error handling:** Substrate-Adapter failures (substrate unavailable, quota exceeded, encryption-key unavailable) trigger SUBSTRATE_WRITE_FAILED states with explicit failure-reason records and retry under exponential backoff; encryption-key derivation failures (e.g., HSM unavailability) trigger KEY_UNAVAILABLE states.
- **Performance characteristics:**.qpn-write is dominated by substrate-write latency: Local Disk typical 1-10ms, Cloud Storage typical 10-100ms, IPFS / CAS typical 100ms-2s, Federated typical 100ms-1s. Routing decisions are O(1) per write.

Cross-Family Integration

Upstream Dependencies. Family F provides storage-substrate-abstraction services to Family A's Local Vault QPC. The Personal Archive, Catalyst Continuity Profile (CCP), and Substrate Adapters operate within QPN-enabled infrastructure per the §22.7 Wherein clause. The.qpn file format references Trust-Block-bound content addressing primitives and the Privacy Domain master key derivation hierarchy from U.S. Patent No. 12,316,610 B1.

Downstream Consumers. Family A (Sidecar) is the primary downstream consumer: every Vault QPC operation delegates to Family F's Substrate Adapters and Persistence Router. Family D (Three-Ledger / Two-Log) consumes.qpn container references in Contribution Ledger Entries — the CLE explicitly does not contain plaintext contribution content; content remains in.qpn containers under Family F custody and is referenced only by content-addressed identifier. Family O (Quantum DNA/Genome) consumes CCP records as inheritable Genome elements — a descendant participant's CCP may inherit from a parent participant's CCP per the Recombination Policy. Family P (PPN Reproduction) consumes Family F persistence chains for inheriting.qpn container lineages across reproduction boundaries.

Lateral Interactions. Family F interoperates with Family D's Permanent Privacy Seal primitive — sealed CLEs trigger Family F operations to destroy the corresponding Privacy-Domain-key-derived encryption keys for the affected Trust Block lineage; the.qpn container itself may remain on the substrate but becomes

cryptographically unreachable. Family F interoperates with Family K's Multi-Factor Identity Binding — key-derivation operations across CCP modifications and substrate transitions reference the Family K identity-authentication state.

Emergent System-Level Properties. Family F's Multi-Substrate Persistence architecture produces an emergent property: content portability across substrate boundaries without breaking Trust Block lineage. A.qpn container migrated from Local Disk to Cloud Storage to IPFS preserves its Trust Block lineage record through rotation; downstream consumers (Family D ledger entries, Family E graph edges) reference the content-addressed identifier rather than the substrate location. This decoupling of content addressing from substrate residency is essential for population-scale participant mobility and substrate-vendor diversity.

Family G — Settlement Controller & Cross-Verification Protocol

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -6, -10.

Field Summary

Settlement Controller for deterministic Exchange Token issuance with witness-based Cross-Verification Protocol.

Family G is the network-side issuance authority of the Catalyst Network. Per and, the Settlement Controller is the protocol component that converts admitted contributions (Family D Contribution Ledger entries) into Settlement Ledger issuances under the Premium schedule (Family J), with Cross-Verification Protocol providing the third-party Witness attestation primitive that strengthens issuance against fraud.

Problem Addressed

Conventional settlement systems treat token issuance as a discretionary operator action, exposing the network to discretionary capture and rendering settlement guarantees probabilistic. They also lack a witness-based cross-verification primitive that enables third-party attestation of contribution authenticity without disclosure of underlying contribution content. The result is settlement systems that either (a) over-trust the operator, defeating decentralization, or (b) require contribution disclosure to verifiers, defeating privacy.

Conventional settlement systems for attribution networks suffer from a structural integrity gap: the entity issuing rewards (the settlement authority) is typically the same entity admitting the contributions (the ingestion authority), creating a single point of trust failure. An adversary who compromises the settlement authority can issue fraudulent rewards regardless of contribution integrity.

The Settlement Controller addresses this with explicit separation between ingestion (Sidecar bridge + AI evaluation pipeline) and issuance (Settlement Controller), supplemented by Cross-Verification Protocol attestations from third-party Witnesses who independently confirm contribution integrity by content-addressed reference. The result is that no single entity — not the participant, not the Sidecar, not the AI pipeline, not the Settlement Controller — can unilaterally issue settlement without aggregate Trust Block authorities from each layer.

Solution Overview

The Settlement Controller is a deterministic enforcement primitive: upon authorized cross-party Resource reuse satisfying the Settlement Eligibility Predicate, the Controller issues Exchange Tokens by construction — no operator discretion is involved. The Controller's deterministic function is verifiable by any party holding the relevant Trust Block lineage. The Cross-Verification Protocol is a witness-based attestation primitive: third-party Witnesses may attest to contribution authenticity by signing verification envelopes referencing the contribution's content-addressed identifier; the underlying content is never disclosed to the Witness.

Each settlement issuance operation references its triggering Contribution Ledger entry, its Trust Validation Record (Stage 1 AI output), its Categorization Record (Stage 2), its Premium Score Record (Stage 3), and a Cross-Verification Bundle from at least N third-party Witnesses where N is a Trust Criteria parameter. The Settlement Controller verifies each reference's Trust Block lineage before emitting the Settlement Ledger entry.

QPC isolation enforcement is a protocol-grade requirement: the Settlement Controller executes within a Settlement-scoped QPC with Trust Criteria forbidding direct access to Contribution Ledger content; the

Controller can only verify cross-references, not re-evaluate contributions. This enforces the separation-of-authorities property and prevents settlement-time content tampering.

Components

Components detailed below: Settlement Controller, Cross-Verification Protocol, Witness Solicitation Manager, Issuance Authority. Sources:, §3.10, and

Settlement Eligibility Predicate

Deterministic function over the inputs (Pipeline Attestation from Family C, Authorization Ledger entries from Family D, Cross-Verification attestations from this Family). Returns ELIGIBLE or NOT_ELIGIBLE with a structured reason code.

The Settlement Eligibility Predicate is a Trust-Block-anchored predicate evaluating whether a Contribution Ledger Entry (CLE) qualifies for settlement issuance. Schema: predicate_uuid, predicate_trust_block_uuid, predicate_logic_descriptor (the symbolic predicate logic — typical structure: "CLE is eligible if: (a) Pipeline Completion Record references a successful three-stage run; (b) Confidence Tier of originating participant ≥ threshold T; (c) cross-verification bundle satisfies N-of-M Witness threshold; (d) no Permanent Privacy Seal exists"), threshold_parameter_set (the operative threshold values for the predicate). Per, predicate evaluation is deterministically replayable; predicate amendments are themselves Trust-Block-bound and apply only to CLEs admitted after the amendment block height.

Settlement Controller — Issuance Component

Upon Eligibility Predicate returning ELIGIBLE, the Issuance Component deterministically computes the issuance amount per the canonical waterfall (Exchange Root allocation, Accelerator allocation, Participant Pool allocation) and writes the Settlement Ledger entry.

The Issuance Component executes settlement-issuance operations against eligible CLEs. Algorithm: at each issuance-cycle block height, for each eligible CLE, the Component (i) computes the issuance amount as a deterministic function of the CLE's Pipeline Completion Record's Temporal Durability Score and the operative Premium schedule snapshot; (ii) constructs a Settlement Ledger transfer event Trust-Block-bound to the source CLE; (iii) emits the transfer event to the appropriate payee accounts (participant accounts, Accelerator Participant Pool, Exchange Root allocation, AIIP allocation per the operative allocation schedule). Per and §4.6, the Issuance Component executes within a Settlement-scoped QPC with Trust Criteria forbidding direct Contribution Ledger content access — the Component sees only Pipeline Completion Records, Settlement Eligibility predicate evaluations, and ledger metadata, never the underlying contribution content.

Settlement Controller — Audit Component

Permits any verifier holding the inputs and the Predicate definition to re-execute the Predicate and confirm the Controller's issuance decision. Re-execution is bit-identical via the deterministic replay primitive.

The Audit Component is a passive observation surface that watches Issuance Component operations and emits independent attestations of issuance correctness. Algorithm: at each issuance-cycle block height, the Audit Component independently recomputes the issuance amount for each settled CLE using the same inputs (Pipeline Completion Record, Premium schedule snapshot) and compares against the Issuance Component's emitted amount. Discrepancies emit AUDIT_DISCREPANCY records on the Authorization Ledger. Per, the Audit Component executes within an Audit-scoped QPC distinct from the Issuance Component's QPC, providing structural separation between issuance and audit authorities. Audit Component output is itself Trust-Block-bound and consumable by third-party verifiers.

Cross-Verification Protocol — Witness Solicitation

Component within a Witness-Solicitation-scoped QPC that constructs requests for third-party attestation referencing only content-addressed identifiers.

Witness Solicitation is the protocol component that identifies and solicits third-party Witnesses for cross-verification of admitted contributions. Algorithm: at each CLE admission, the Witness Solicitation component (i) consults the operative Witness Eligibility Policy (Trust-Block-anchored — typical policy: "eligible Witnesses are participants in the same Privacy Domain with Confidence Tier ≥ SILVER and no prior conflict-of-interest edges to the originating participant"); (ii) selects a randomized subset of N candidate Witnesses (typical N = 3-7 for

standard contributions, higher for high-Premium contributions); (iii) emits Witness Solicitation requests to each candidate's Sidecar Bridge. Selection randomization uses a Deterministic Replay Seed (Family P primitive) bound to the CLE UUID so that selection is auditable but unpredictable in advance.

Cross-Verification Protocol — Attestation Aggregation

Component aggregating returned Witness attestations into a Cross-Verification Bundle bound by Trust Block to the original contribution identifier.

Attestation Aggregation collects Witness attestations returned in response to Solicitation requests and aggregates them into a Cross-Verification Bundle. Schema: bundle_uuid, target_cle_uuid_reference, witness_attestation_set (each attestation referencing the witnessing participant's DID, the attestation Trust Block, and the binary or graduated confirmation value), bundle_completion_block_height, bundle_completion_status (enum: PENDING, SATISFIED, INSUFFICIENT). Aggregation Algorithm: SATISFIED requires N-of-M Witness confirmations where N is the operative threshold; INSUFFICIENT outcomes mark the CLE as ineligible for premium settlement under the Settlement Eligibility Predicate. Per, aggregation is deterministic and replayable; attestation submission deadlines are Trust-Block-anchored.

Cross-Verification Protocol — Witness Trust Evaluation

Component evaluating Witness Trust Blocks for reputation and conflict-of-interest signals before incorporating their attestations into the Cross-Verification Bundle.

Witness Trust Evaluation weighs Witness attestations by the Witnessing participant's Confidence Tier (Family E) and historical attestation accuracy. Schema: evaluation_record_uuid, witness_did, historical_attestation_accuracy_score (rational $\in [0,1]$ computed deterministically from the Witness's prior attestation outcomes vs. ground-truth ledger state), current_confidence_tier, evaluation_block_height. Per, Witnesses whose historical attestation accuracy falls below a Trust-Block-anchored threshold are excluded from future Witness Solicitation selections — providing protocol-grade defense against malicious or unreliable Witnesses without requiring centralized Witness curation.

Process Flow

Upon a Resource reuse event triggering a settlement evaluation, the Settlement Controller assembles the input bundle.

If Cross-Verification is required by the Mode (Family B) or Premium framework (Family J), the Witness Solicitation component issues attestation requests to selected Witnesses.

Returned Witness attestations are aggregated and Witness Trust evaluated; the resulting Cross-Verification Bundle is added to the Predicate input.

The Settlement Eligibility Predicate is evaluated.

If ELIGIBLE, the Issuance Component computes the waterfall allocation and writes the Settlement Ledger entry.

If NOT_ELIGIBLE, no settlement occurs and the structured reason code is logged to the Control Plane Log (Family D).

Alternative Embodiments

ZK-bounded Witness embodiment: Witness attestations may be presented as zero-knowledge proofs of contribution authenticity, preserving Witness anonymity.

Multi-jurisdiction settlement embodiment: the Controller may issue per-jurisdiction settlement variants when the contribution spans multiple jurisdictional Privacy Networks, each variant with its own Trust Block jurisdiction binding.

Deferred settlement embodiment: when the participant's QPC is in DORMANT state (Family I), the Settlement Controller may compute and record the settlement but defer Exchange Token release until activation.

Stream settlement embodiment: Settlement may operate in streaming mode for high-volume contribution sources: rather than discrete per-CLE settlement events, the Controller may emit continuous settlement

streams against aggregate CLE batches at configurable cadence intervals (per-second, per-block, per-minute). Streaming settlement requires Trust-Block-anchored aggregation boundaries to preserve per-CLE auditability.

Cross-jurisdiction Witness Solicitation embodiment: Witness Solicitation may select Witnesses across jurisdictions for contributions whose admissibility benefits from cross-jurisdictional verification (e.g., research findings whose validity is enhanced by international peer attestation). Cross-jurisdiction Witness selection follows compliance-attestation chains ensuring each selected Witness is legally permitted to attest under the contribution's substantive content.

Algorithmic Witness embodiment: Witnesses need not be human participants — the protocol supports Algorithmic Witnesses, which are Trust-Block-bound automated attestation systems (e.g., a static-analysis tool attesting to code quality, a fact-checking model attesting to claim accuracy). Algorithmic Witnesses are subject to the same Witness Trust Evaluation discipline (Family G) as human Witnesses; their historical attestation accuracy is similarly tracked.

Settlement-on-Action embodiment: For specific contribution classes (e.g., infrastructure maintenance, code contributions to production systems), settlement may be conditioned on downstream-action observation: settlement triggers only after the contribution is observably consumed (e.g., code deployed to production, infrastructure upgrade verified operational). Settlement-on-Action requires post-admission observation oracles Trust-Block-bound to action-verification systems.

Conditional Cross-Verification embodiment: Cross-Verification thresholds may be conditioned on Premium magnitude: low-Premium contributions may require fewer Witnesses (e.g., 2-of-3); high-Premium contributions may require larger Witness juries (e.g., 7-of-11). Threshold scaling is Trust-Block-anchored and configurable per Privacy Domain.

Reputation-weighted Cross-Verification embodiment: Witness attestations may be weighted by Confidence Tier and historical accuracy in the Cross-Verification Bundle aggregation: a TIER_GOLD Witness's attestation may carry weight equivalent to multiple TIER_BRONZE Witnesses' attestations. Reputation-weighted aggregation preserves N-of-M-style threshold structure with effective N defined in weighted terms.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for settlement issuance:** (i) the target CLE must satisfy the Settlement Eligibility Predicate; (ii) the Pipeline Completion Record must be present and valid; (iii) the Cross-Verification Bundle must be in SATISFIED state.
- **Postconditions:** (i) a Settlement Ledger transfer event is emitted; (ii) Premium Runaway Prevention checks (Family J) have applied; (iii) downstream consumers (Accelerator MUM computation, Backing Pool participation distribution) are notified.
- **State transitions:** CLE settlement-state lifecycle: ELIGIBILITY_PENDING → ELIGIBLE → CROSS_VERIFICATION_PENDING → SATISFIED → SETTLED → AUDITED.
- **Error handling:** eligibility-predicate failures emit CLE_INELIGIBLE records without settlement; Cross-Verification timeouts (no Witness attestations received within the operative deadline) emit CROSS_VERIFICATION_TIMEOUT records and may downgrade settlement amounts per Trust-Block-anchored timeout policy.
- **Performance characteristics:** settlement issuance is O(E) per E eligible CLEs in the issuance cycle; Cross-Verification Bundle aggregation is O(N) per N solicited Witnesses. Per, the Controller is designed for 1,000-10,000 settlements per second per Catalyst Network instance.

Cross-Family Integration

Upstream Dependencies. Family G consumes Pipeline Completion Records from Family C, Contribution and Authorization Ledger state from Family D, Reputation outputs from Family E, Witness solicitation policies bound to Family B Mode-Tags, and Premium Score Records from Family J. The Settlement Controller operates within QPN-enabled infrastructure per the §22.7 Wherein clause; the QPC isolation enforcement (per the Phase A reframing recommendation for Claim 172) is a structural requirement: the Settlement Controller QPC is structurally incapable of direct Contribution Ledger content access.

Downstream Consumers. Family G emits Settlement Ledger transfer events that are consumed by Family L (Senior/Junior QPT Derivatives) for derivative accrual, Family M (Stage-Differentiated Revert + MUM) for MUM computation, Family N (Liquidity Architecture) for Backing Pool participation distribution, and Family K (Behavioral Activation) for XP and Streak Multiplier updates. Per Pattern #5 (the architectural analysis), the Family G + Family J deterministic settlement composition is essential: settlement issuance is a deterministic function of the Pipeline Completion Record's Premium Score, the Compression Curve position, and the operative Settlement Eligibility Predicate. Reproducibility holds across the chain.

Lateral Interactions. Family G's Cross-Verification Protocol interoperates with Family E Reputation Engine — Witness Trust Evaluation references the Reputation Engine's Confidence Tier and historical-accuracy outputs. The Settlement Controller's Audit Component executes within an Audit-scoped QPC distinct from the Issuance Component's QPC, providing structural separation between issuance and audit authorities; this separation is the architectural basis for Family G's downstream consumers' trust in settlement correctness without trust in the issuing entity.

Emergent System-Level Properties. The deterministic settlement composition is an emergent property: given identical Pipeline Completion Records, Premium schedule snapshots, Compression Curve positions, and Settlement Eligibility Predicates, the Settlement Controller's issuance amounts are reproducible by any auditor without privileged access. This property is the architectural replacement for trust in the issuing entity's discretion — issuance is computation, not decision-making.

Family H — Specialized Catalyst Vectors (Web/Voice/Code/Doc/Agent/Comm/Meet/Message)

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -11, -12, -13, -14, -15, -16, -17, -18.

Field Summary

Eight domain-specialized Catalyst Vectors implementing the Sidecar pattern across web, voice, code, docu—— ment, agentic, communication, meeting, and messaging contribution surfaces.

Family H is the vector-specific instantiation layer of the Sidecar. Per and, the Specialized Catalyst Vectors are vector-class-specific implementations of the four-component Sidecar pattern (Family A) tailored to Web, Voice, Code, Document, Agent, Communication, Meeting, and Message contribution modalities. Each vector inherits the four-component decomposition but instantiates vector-specific Witness behaviors, vector-specific Listener attestation surfaces, vector-specific Vault encoding schemes, and vector-specific Bridge envelope formats.

Problem Addressed

Contribution capture surfaces are heterogeneous — browsers, voice assistants, IDEs, document editors, agentic orchestration platforms, communication channels, video conferencing, messaging — and each surface requires domain-specific event extraction, identity binding, and attestation. A monolithic capture agent either over-fits to a single surface or under-serves all of them. Conversely, fully independent capture agents lose architectural coherence — no shared Trust Block schema, no shared Mode discrimination, no shared settlement pathway. The challenge is to provide eight surface-specific Vectors that share the Sidecar pattern (Family A) while adding domain-specific primitives.

A generic Sidecar pattern, while architecturally complete, is insufficient for production deployment because each contribution modality has modality-specific signal characteristics that demand specialized capture and evaluation logic. Web contributions (browser-based reading, writing, transacting) differ fundamentally from Voice contributions (recorded dictation, conversational interaction) which differ from Code contributions (commit events, review comments, test runs) which differ from Document contributions (long-form writing, multi-author edits) and so on. Attempting to handle all modalities through a single generic Sidecar produces either underspecified capture (missing the modality-specific evidence that strengthens trust verification) or overspecified capture (admitting modality-irrelevant signal that defeats privacy).

The Specialized Catalyst Vectors address this by providing modality-specific Sidecar instantiations that share the four-component architecture and the Trust-Block discipline, but specialize the within-component behavior

to the modality. This permits both protocol-grade integrity (every vector retains the four-QPC decomposition and the Trust-Block message envelope discipline) and modality-appropriate capture (each vector emits modality-appropriate Witness Records and Listener Attestations).

Solution Overview

Each of the eight specialized Vectors (WebVector, VoiceVector, CodeVector, DocVector, AgentVector, CommVector, MeetVector, MessageVector) implements the four-component Sidecar pattern with domain-specific primitives layered on. WebVector captures browser interactions and binds page-origin attestation. VoiceVector captures voice activations with on-device speech-to-intent processing; raw audio never leaves the device. CodeVector binds to IDE contributions at the pull-request level. DocVector emits W3C Verifiable Credentials at the document section level. AgentVector emits Proof of Orchestration attestations tying agentic actions to authorizing principal Trust Blocks. CommVector implements the verified engagement loop with Message-ID matching. MeetVector emits attendance attestations. MessageVector implements a privacy-preserving proxy network for messaging contributions.

Each vector is independently disclosed with its specialized Witness behaviors and validation rules. The Web Vector specializes Witness to browser DOM events and Listener to browser ambient context (visible tabs, focus state). The Voice Vector specializes Witness to explicit voice-activation events and Listener to ambient acoustic context with content suppression. The Code Vector specializes Witness to commit and review events with content-addressed source-tree references. The Document Vector specializes Witness to section-granular edit events with structural ancestry. The Agent, Communication, Meeting, and Message Vectors follow analogous specialization patterns.

Witness-validation behaviors are vector-specific: each vector encodes the modality-specific verification rules that strengthen the Witness Record against forgery. Web Vector Witness Records reference browser-specific origin attestations; Voice Vector Witness Records reference platform-attested voice-activation events; Code Vector Witness Records reference content-addressed source-tree state. These specializations are protocol-grade requirements, not implementation conveniences.

Components

Components are detailed per vector below. Sources: (vector taxonomy), §4.10.1–§4.10.8 (per-vector specializations), and

WebVector

Browser-environment Vector. Witness Agent observes user-action events bound to the active tab's origin. Listener Agent observes ambient page-context signals (URL class, content type) authorized by the page-origin Trust Criteria. Local Vault stores per-origin contribution blobs. Verification Bridge emits Origin-Trust-Block-bound Verification Envelopes.

WebVector specializes the four-component Sidecar pattern for browser-based web contributions. Vector-specific Witness Records reference DOM-specific event fields: dom_event_type (click, keydown, edit, focus, scroll), target_element_selector (CSS selector path to the target element), page_url_origin_hash (Trust-Block-bound hash of the page URL origin for deduplication and origin attestation), browser_attestation_token (browser-vendor-attested integrity token where supported — e.g., Chrome Web Authentication, Safari Private Access Tokens). Vector-specific Listener Attestations include visible_tab_count, active_extensions_attestation, focus_state. Per, WebVector is the primary contribution vector for browser-resident reading, writing, and transacting.

VoiceVector

Voice-input Vector. Witness Agent processes wake-word-gated voice activations within an on-device QPC such that raw audio never crosses the device boundary; only post-intent-extraction Trust-Block-wrapped intent records leave the device.

VoiceVector specializes the Sidecar for voice-originated contributions. Vector-specific Witness Records require explicit voice-activation events: activation_trigger_type (wake-word, push-to-talk button, hardware switch), activation_timestamp_attested (platform-attested timestamp with deviation-from-device-clock recorded), voice_recording_handle (reference to a Privacy-Domain-bound encrypted audio segment in the Local Vault —

the recording itself never leaves the Vault QPC in plaintext), speaker_identification_attestation (optional Trust-Block-bound speaker-recognition attestation for multi-speaker contexts). Vector-specific Listener Attestations are heavily content-suppressed: ambient acoustic context is attested only at the presence/absence/class level (e.g., AMBIENT_SPEECH_PRESENT, AMBIENT_MUSIC_PRESENT) without content capture. Per

CodeVector

Code-contribution Vector. Witness Agent binds to IDE events (commit, pull request, review). Identity Binding ties contributions to the developer's Trust Block. Pull-request-level attribution captures multi-author Trust Block bundles.

CodeVector specializes for source-code contributions. Vector-specific Witness Records reference source-tree state: commit_hash (the version-control system commit hash — Git, Mercurial, Pijul), file_path_set, diff_summary_hash (content-addressed hash of the diff, supporting later auditor verification without disclosing diff content), programming_language_set, static_analysis_attestation_set (optional attestations from static-analysis tools — lint passing, security-scan passing, test-suite passing). Vector-specific Listener Attestations include ide_focus_state, active_branch, active_test_suite_state. Per, CodeVector integrates with content-addressed source-tree state supporting deterministic-replay verification of code contributions.

DocVector

Document-authoring Vector. Emits W3C Verifiable Credentials at the document section level, with each section's VC bound to the Trust Block of the contributing author. Section-level binding enables fine-grained attribution and Premium computation per section.

DocVector specializes for long-form document contributions. Vector-specific Witness Records reference section-granular edit events: document_uuid, section_ancestry_path (the document's hierarchical structure ancestry to the edited section), edit_type (INSERT, DELETE, MODIFY, REORDER), co_author_set (other participants concurrently editing the document — supports multi-author attribution), revision_ancestry_chain (the chain of prior revisions, supporting deterministic version-history reconstruction). Vector-specific Listener Attestations include concurrent_collaborator_presence, document_view_state. Per

AgentVector

Agentic AI orchestration Vector. Emits Proof of Orchestration attestations tying each agent action to the authorizing principal's Trust Block. Inherits Governed Agent Loop (Family I) for the action-authorization step.

AgentVector specializes for AI-agent-mediated contributions. Vector-specific Witness Records reference per-step Trust-Criteria gates (Family I): agent_step_uuid, agent_step_type (per the Family I taxonomy: SENSE, INTERPRET, PROPOSE, AUTHORIZE, EXECUTE, VERIFY, LEARN), step_authorization_trust_block_uuid (the Trust Block authorizing this step), step_input_summary (content-addressed summary of inputs), step_output_summary (content-addressed summary of outputs), next_step_trigger_reference. Per and Family I, AgentVector preserves the per-step audit chain — every agent action is a discrete Witness Record rather than an opaque batch event.

CommVector

Communication Vector. Implements the verified engagement loop with Message-ID matching: outbound communications are tagged with Message-IDs; inbound responses referencing the Message-IDs trigger closed-loop engagement attestation.

CommVector specializes for inter-participant communications. Vector-specific Witness Records reference communication-event metadata without content disclosure: communication_event_uuid, participant_set (the communication participants), communication_class (DIRECT, GROUP, BROADCAST), communication_modality (TEXT, VOICE, VIDEO, HYBRID), communication_substrate_descriptor (the underlying substrate — Signal, Matrix, email, in-app messaging), content_hash (content-addressed hash of the message, supporting referenceability without disclosure). Per, CommVector supports cross-substrate communication attribution while preserving the cryptographic-confidentiality properties of the underlying communication substrate.

MeetVector

Meeting attendance attestation Vector. Witness Agent observes meeting join/leave events bound to the meeting's Trust Block; emits attendance attestation upon meeting completion.

MeetVector specializes for meeting-context contributions. Vector-specific Witness Records reference meeting metadata: meeting_uuid, meeting_substrate_descriptor (Zoom, Google Meet, Microsoft Teams, in-person), attendee_set, meeting_duration_seconds, agenda_reference (optional agenda document reference per DocVector), recording_handle (optional encrypted recording reference per VoiceVector — the recording remains in the originating participant's Vault). Vector-specific Listener Attestations include attendee_active_speaking_distribution, slides_or_screenshare_presence. Per

MessageVector

Messaging Vector with Privacy-Preserving Proxy Network: contribution capture flows through a proxy graph such that the messaging substrate cannot correlate contribution attribution to the participant's underlying network identity.

MessageVector specializes for short-form messaging contributions. Vector-specific Witness Records reference message-event metadata: message_uuid, message_thread_ancestry, message_modality (TEXT, EMOJI, MEDIA, REACTION), message_substrate_descriptor, content_hash, thread_participant_set. Vector-specific Listener Attestations include active_thread_count, unread_message_queue_depth. MessageVector differs from CommVector primarily in granularity and thread-ancestry tracking: CommVector captures communication events; MessageVector captures discrete message-level events within communication threads. Per

Process Flow

Each Vector inherits the Sidecar four-component lifecycle from Family A.

Vector-specific Witness Agent applies surface-specific event extraction (browser events, voice intent, IDE events, document edits, agent actions, communication, meetings, messages).

Vector-specific Listener Agent applies surface-specific ambient observation within Trust Criteria.

Vector-specific Verification Bridge applies surface-specific identity binding (origin, device, developer, author, principal, message-thread, meeting, proxy-graph).

Verification Envelopes flow to the Settlement Controller (Family G) via the standard pipeline.

Alternative Embodiments

Cross-Vector aggregation embodiment: a single participant's contributions across multiple Vectors may be aggregated into a composite Pipeline Attestation reflecting cross-surface activity (e.g., a developer who codes, documents, and discusses receives composite Premium reflecting all three).

Domain-specific Vector extension: additional Vectors may be defined for emerging surfaces (e.g., AR/VR interaction, IoT sensor contributions) using the same architectural template.

Pluggable identity binding embodiment: each Vector's Identity Binding primitive is pluggable, supporting multiple identity schemes per surface.

Cross-Vector aggregation pattern embodiment: Beyond the existing Cross-Vector aggregation embodiment, specific aggregation patterns may be Trust-Block-anchored: a Meeting Aggregation Pattern aggregates MeetVector + VoiceVector + DocVector contributions from the same meeting into a unified meeting-attributable contribution bundle; a Research Aggregation Pattern aggregates CodeVector + DocVector + CommVector contributions from the same research project into a unified research-attributable contribution bundle. Aggregation patterns themselves may be inherited per Family O Quantum DNA.

Adaptive Vector embodiment: Vectors may dynamically adapt their capture surfaces based on participant context: a Vector deployed in a healthcare clinical-decision-support context may adapt its Witness Records to capture clinical-decision-relevant metadata; the same Vector deployed in a journalism context may adapt to capture journalism-relevant metadata. Adaptation is Trust-Block-anchored at the deployment configuration level.

Cross-platform Vector embodiment: A single Vector instance may span multiple device platforms simultaneously — e.g., a Document Vector capturing contributions across a participant's laptop, tablet, and smartphone with cross-device deduplication. Cross-platform deduplication uses content-addressed identifiers and Trust-Block-bound device-attestation chains.

Vector-specific Witness Solicitation embodiment: Each Vector may have vector-specific Witness Solicitation policies: CodeVector may solicit Witnesses with software-engineering Reputation; VoiceVector may solicit Witnesses with audio-attestation Reputation; etc. Vector-specific Solicitation increases Cross-Verification quality at modest additional Witness-pool curation cost.

Federated-Vector embodiment: Vectors may operate across organizational federations: a federated CommVector deployment may operate across multiple organizations' communication systems with cross-organization attribution under federation Trust Blocks. Federation memberships are Trust-Block-bound and revocable.

New-Vector instantiation embodiment: The Vector taxonomy is extensible: deployments may instantiate domain-specific Vectors (e.g., BiomedicalSampleVector for laboratory sample attribution, FinancialTransactionVector for transaction-bound attribution, IoTSensorVector for sensor-stream attribution). New-Vector instantiation requires Trust-Block-anchored taxonomy extension and Witness-Solicitation-policy specification.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Vector-Specific Preconditions:** Each Vector has vector-specific preconditions — WebVector requires valid browser-origin attestation; VoiceVector requires explicit voice-activation; CodeVector requires version-control commit reference; DocVector requires document-section ancestry resolution; AgentVector requires Family I gate evaluation; CommVector requires participant-set attestation; MeetVector requires meeting-substrate descriptor; MessageVector requires thread-ancestry resolution.
- **Common Postconditions:** Each Vector emits Witness Records via the same Family A four-component pattern; vector-specific schemas are merged into the common Witness Record envelope.
- **State transitions:** Each Vector preserves the Family A Witness Record state lifecycle (OBSERVED → VAULTED → REFERENCED → SUBMITTED → ADMITTED).
- **Cross-Vector Considerations:** A single participant may operate multiple Vectors concurrently; cross-Vector aggregation (e.g., a meeting captured via MeetVector whose recorded voice content is captured via VoiceVector) is supported through cross-Vector reference records.
- **Performance characteristics:** Per-Vector overhead is bounded by the underlying Sidecar four-component cost; aggregate Vector capacity scales linearly with available QPC instances.

Cross-Family Integration

Upstream Dependencies. Family H inherits the four-component Sidecar pattern from Family A wholesale. Each Specialized Catalyst Vector (Web, Voice, Code, Doc, Agent, Comm, Meet, Message) is a vector-specific instantiation of Family A's Witness Agent, Listener Agent, Local Vault, and Verification Bridge architecture with vector-specific Witness Record schemas, Listener Attestation surfaces, Vault encoding extensions, and Bridge envelope formats. Each Vector operates within QPN-enabled infrastructure per the §22.7 Wherein clause and references the Family B Five Signal Input Modes for admission policy.

Downstream Consumers. Every Vector's outputs are consumed by the same downstream Families that consume Family A outputs — Family C (Pipeline), Family D (Ledger), Family E (Graph), Family G (Settlement), Family J (Premium). The Vector-specific Witness Record schemas propagate downstream into the Pipeline as additional schema fields available to Stage 1 Semantic Classification and Stage 3 Temporal Durability. Family I (Governed Agent Loop) is integrated with AgentVector specifically: each Family I Execute Step emits a Witness Record via AgentVector.

Lateral Interactions. Cross-Vector aggregation patterns (the Pass 2 cross-Vector aggregation embodiment) interoperate across all eight Vectors — e.g., a Meeting Aggregation Pattern combines MeetVector + VoiceVector + DocVector contributions from the same meeting context. Vector-specific Witness-validation behaviors (Phase A reframing recommendations for Claims 185 / 191 / 196 / 217) propagate into the Family C Pipeline as additional verification inputs.

Emergent System-Level Properties. Family H's vector-specific specialization produces an emergent property: modality-appropriate evidence strength. The Sidecar's four-QPC pattern guarantees cryptographic integrity at

the architecture level; Family H's vector-specific Witness-validation behaviors guarantee modality-appropriate evidence strength at the modality level. A WebVector Witness Record references browser-vendor origin attestations; a VoiceVector Witness Record references platform-attested voice-activation events; etc. The integration produces evidence that is both cryptographically integral and modality-appropriately strong — neither property alone is sufficient for population-scale trust-verified attribution.

Family I — Governed Agent Loop & Phase-0 DORMANT QPC State

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -20, -31.

Field Summary

Seven-step Governed Agent Loop and Phase-0 deferred-activation DORMANT QPC compliance state.

Family I is the protocol primitive for AI-mediated participant activity. Per and, the Governed Agent Loop captures AI-agent-originated contributions under a per-step Trust-Block discipline that requires each agent action to pass through a Trust Criteria gate before becoming a Witness Record. The Phase-0 DORMANT QPC State is the compliance primitive that allows a Catalyst Network Manager to record verified contributions on behalf of participants who have not yet self-activated.

Problem Addressed

Agentic AI execution presents two unresolved challenges: (a) per-step authorization, ensuring that each step of an agent's autonomous action is bound to an authorizing principal Trust Block, and (b) pre-activation participant safety, ensuring that contributions attributed to a participant who has not yet self-activated do not create bribery, inducement, or prohibited-benefit exposure. Existing agent frameworks authorize at the session level (insufficient for fine-grained control) and treat unauthorized accruals as either rejected (loss of attribution) or actively distributed (compliance exposure).

AI-agent activity creates an attribution gap that conventional contribution-capture systems cannot bridge. When a user instructs an AI agent to perform a sequence of actions (research a topic, draft a document, send messages), each action produces signal but no human-originated Witness event exists for each step. Conventional systems either attribute all agent action to the original instruction (over-attributing) or attribute nothing (under-attributing). Both extremes defeat the Catalyst Network's attribution discipline.

The DORMANT QPC State addresses a separate but related problem: many beneficial contributions occur on behalf of participants who are not yet active in the Catalyst Network. A teacher whose lecture is recorded and contributed by a student, a public figure whose statements are referenced in research, a community member whose civic activity is attested by a Witness — all of these participants warrant attribution but cannot yet receive settlement because they have not self-activated. The DORMANT QPC primitive allows Manager-Originated QPCs to accumulate contribution records on behalf of these participants, with settlement deferred until self-activation.

Solution Overview

The Governed Agent Loop is a seven-step canonical execution sequence — Sense, Interpret, Propose, Authorize, Execute, Verify, Learn — with each step bound to a Trust Block. The Authorize step gates against the principal's active Trust Criteria and against the participant's QPC activation state. The Phase-0 DORMANT QPC State is a record state in which a Manager-Originated QPC holds contribution accruals for a participant who has not yet self-activated; the DORMANT QPC has no economic function (cannot be transferred, exchanged, or valued) until the participant performs activation under PPCS screening.

The Governed Agent Loop intercepts every agent step at a Trust-Criteria-evaluated gate. Each step's preconditions (intended action, target surface, expected outcome) are evaluated against the participant's standing Trust Criteria; the step is admitted as a Witness Record only if the gate passes. Failed gate evaluations produce Audit Log entries but no Witness Record. This converts agent activity from an opaque sequence of actions into an auditable sequence of Trust-Block-bound decisions, with deterministic-replay guarantees.

The DORMANT QPC State operates as a compliance primitive: a Manager-Originated QPC accumulates Contribution Ledger entries attributed to a participant who has not self-activated, with no economic settlement issued. Upon self-activation (Family K Behavioral Activation), the DORMANT QPC's accumulated entries are

integrated into the participant's active Catalyst Continuity Profile and settlement entitlements are computed retroactively under the prevailing Premium schedule.

Components

Components detailed below: Agent Loop Gate, Trust-Criteria Evaluator, Step Witness Recorder, DORMANT QPC State Manager, Self-Activation Integrator. Sources: (Agent Loop), §4.11 (DORMANT compliance)

Sense Step

Agent observes its environment (data, tool outputs, communications). Within a Sense-scoped QPC; observations are Trust-Block-tagged with timestamp and Mode.

The Sense Step is the agent-loop input observation stage. Schema: sense_step_uuid, agent_did, sense_event_class (enum across environmental observation types — DOCUMENT_READ, API_RESPONSE_RECEIVED, USER_PROMPT_ARRIVED, SCHEDULED_TRIGGER_FIRED), sense_input_summary (content-addressed summary), sense_trust_block_uuid. Per and, each Sense Step is individually Trust-Block-bound; aggregated multi-Sense-step processing is permitted but each input must remain individually identifiable for replay.

Interpret Step

Agent applies its reasoning model to observations, producing an Interpretation Trust Block bound to deterministic-replay seed.

The Interpret Step converts sensed inputs into agent-internal representations. Schema: interpret_step_uuid, sense_step_reference_set, interpretation_payload_hash, interpretation_model_snapshot_reference, interpretation_confidence_score. The Interpret Step's deterministic-replay property: given the same Sense Step inputs and model snapshot, interpretation outputs are reproducible. Low-confidence interpretations (below Trust-Block-anchored threshold) may trigger LOW_CONFIDENCE_INTERPRETATION states requiring escalation.

Propose Step

Agent proposes a candidate action set. Within a Propose-scoped QPC; emits a Proposal Trust Block enumerating candidate actions and their predicted effects.

The Propose Step generates candidate agent actions. Schema: propose_step_uuid, interpret_step_reference, proposed_action_set (each candidate action with action class, target resources, expected effect, risk level), proposal_ranking (relative preference ordering). Per, the Propose Step outputs are subject to the Authorize Step's Trust-Criteria gate before execution; proposal generation is reproducible given the interpretation and model snapshot.

Authorize Step

Authorization component within an Authorize-scoped QPC evaluates the Proposal against the principal's active Trust Criteria. Returns AUTHORIZED set (action subset permitted) and DENIED set (rejected actions). The Authorize step also checks the principal's QPC activation state: actions whose effect would deliver value to a DORMANT QPC are conditionally accepted (effect recorded; value held in escrow until activation).

The Authorize Step is the Trust-Criteria gate that approves or rejects proposed actions before execution. Algorithm: for each proposed action P, the Authorize Step (i) evaluates P against the participant's standing Trust Criteria; (ii) evaluates P against the operative Agent Authorization Policy (Trust-Block-anchored — specifies which action classes require explicit human approval vs. autonomous approval); (iii) emits an Authorization Decision Record (APPROVED, REJECTED, DEFERRED_FOR_HUMAN_APPROVAL). Per, the Authorize Step is the structural primitive distinguishing governed agent operation from ungoverned agent operation; bypass of the Authorize Step is structurally prevented by the QPC boundary.

Execute Step

Agent executes the AUTHORIZED actions. Each action emits an Execution Trust Block.

The Execute Step performs authorized agent actions. Schema: execute_step_uuid, authorize_step_reference, execution_target_reference (the target resource, API endpoint, or device interface), execution_outcome_record_reference, execution_block_height. Execution operations are Trust-Block-bound

and produce Witness Records via AgentVector (Family H); execution failures emit Audit Log entries with explicit failure-reason records.

Verify Step
Verification component checks that execution outcomes match predicted effects. Discrepancies emit a Verify Failure Trust Block triggering remediation.

The Verify Step validates that the Execute Step's outcome matches the proposed/authorized expectation. Schema: verify_step_uuid, execute_step_reference, expected_outcome_signature, actual_outcome_signature, verification_decision (MATCH, MISMATCH, INCONCLUSIVE). Mismatch detection triggers MISMATCH_DETECTED states that may pause subsequent Execute Steps pending human or secondary-authority review. Per

Learn Step
Agent updates its model based on Verify outcomes; the update is recorded as a Learn Trust Block bound to the agent's identity.

The Learn Step incorporates verified outcomes back into agent-internal learning state. Schema: learn_step_uuid, verify_step_reference_set, learning_update_payload_hash, learning_model_snapshot_after_update. Per, Learn Step updates are themselves Trust-Block-bound; model rollback to prior snapshots is supported under Trust-Block-bound rollback authority for cases where Learn Step updates introduce undesirable behavior.

DORMANT QPC State
Compliance state for a participant's QPC prior to activation. Contribution records may be created (Manager-Originated) and bound to the DORMANT QPC, but no economic transfer is possible. Activation requires explicit participant action plus PPCS compliance screening.

The DORMANT QPC State is the protocol primitive for Manager-Originated QPCs accumulating contributions on behalf of pre-activation participants. Schema: dormant_qpc_uuid, originating_manager_did, target_participant_identifier_class (the class of identifier under which the target participant is referenced — typically a hash of identity attestations, not a DID since the target has not yet self-activated), accumulated_cle_set (the set of Contribution Ledger Entries accumulated under this DORMANT QPC), activation_pending_block_height (when the QPC was instantiated and accumulation began). The structural-incapacity invariant: a DORMANT QPC's Trust Criteria forbid economic operations (no settlement issuance, no derivative trading, no liquidity participation) — accumulation is purely attribution-record-keeping. Per and

PPCS Activation Gate
Privacy-Preserving Compliance Screening gate enforcing per-jurisdiction activation eligibility (e.g., bribery/inducement screening for participants who are government officials per the Government Official Participation Framework).

The PPCS (Participant Privacy Cell Self-Activation) Gate is the protocol component that transitions a DORMANT QPC's accumulated state to an ACTIVE participant identity upon self-activation. Algorithm: at participant self-activation, the Gate (i) verifies the participant's identity attestations match the DORMANT QPC's target_participant_identifier_class; (ii) emits a Trust-Block-bound Activation Transition record; (iii) re-binds the DORMANT QPC's accumulated_cle_set to the participant's new ACTIVE Privacy Domain; (iv) computes retroactive settlement entitlements per the operative Premium schedule for each retroactively-bound CLE. Per, the retroactive-settlement computation references the Premium schedule at each CLE's original admission block height, not at the activation block height — preserving Premium-Compression-Curve consistency.

Process Flow
An agent receives a triggering event; the Sense step records the observation.

Interpret produces a Trust-Block-bound interpretation.

Propose enumerates candidate actions.

Authorize evaluates the Proposal: AUTHORIZED actions proceed; DENIED actions are logged. For AUTHORIZED actions whose effect targets a DORMANT QPC, the effect is recorded but value is escrowed.

Execute carries out the AUTHORIZED actions, each emitting an Execution Trust Block.

Verify compares outcomes to predictions; failures trigger remediation flows.

Learn updates the agent model.

Upon a participant invoking activation on a DORMANT QPC, PPCS screening is run; on screening pass, the QPC transitions to ACTIVE state and escrowed value is released.

Alternative Embodiments

Multi-principal Governed Agent Loop: an agent acting on behalf of multiple principals (e.g., enterprise-and-employee dual authorization) executes Authorize with multi-Trust-Block evaluation.

Step-skipping embodiment: for trusted recurrent agent operations, Interpret and Propose may be combined into a single attestation when the operation pattern is pre-authorized.

Pre-activation contribution embodiment: a Manager-Originated DORMANT QPC may accumulate contributions over an extended period; on activation, the Participant retroactively inherits the full attribution lineage.

Multi-agent coordination embodiment: Multiple Governed Agent Loops may operate in coordinated patterns: a primary agent's Propose Step output may feed into a secondary agent's Sense Step input under Trust-Block-bound inter-agent message envelopes. Multi-agent coordination preserves per-agent loop discipline; each agent's seven-step sequence is independently auditable.

Hierarchical agent embodiment: Agents may operate in hierarchical structures: a supervisory agent may delegate sub-tasks to subordinate agents under bounded ACPAs (Family K). Hierarchical delegation chains are Trust-Block-bound and revocable; supervisory agents retain Verify Step authority over subordinate Execute Steps.

Human-in-the-loop embodiment: Specific Authorize Step decision classes may be configured to require human approval — high-stakes actions, low-confidence interpretations, novel proposal types. Human approval is itself Trust-Block-bound; the human approver's attestation is recorded with the Authorize Decision.

Outcome-conditional Learn Step embodiment: Learn Step updates may be conditioned on downstream verification outcomes: the Learn Step's update payload is queued but not applied until Cross-Verification of executed actions yields favorable outcomes. Conditional Learn Steps prevent learning from incorrect or unverified outcomes.

DORMANT QPC mass-activation embodiment: A community or organization may operate a fleet of DORMANT QPCs accumulating contributions on behalf of population-scale pre-activation participant sets. Mass-activation events (e.g., a community launch event, an enterprise rollout) trigger bulk PPCS Activation Gate evaluations with optimized retroactive-settlement computation.

Cross-Privacy-Domain agent embodiment: Agents may operate across multiple Privacy Domains under explicit cross-domain delegation chains. Cross-domain Witness Records are emitted to each operative Privacy Domain's Contribution Ledger; cross-domain settlement is bound by per-domain Trust Criteria.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for agent loop execution:** (i) the agent must hold valid Trust Block authority bindings (delegated from the participant under Family K ACPAs or directly issued); (ii) the agent's model snapshots for each step must be referenced and valid; (iii) the participant's standing Trust Criteria must be loaded for Authorize Step evaluation.
- **Postconditions:** (i) the full seven-step record sequence is emitted Trust-Block-bound; (ii) executed actions' outcomes are verified and learning state is updated; (iii) AgentVector Witness Records (Family H) are emitted for each Execute Step.

- **State transitions:** Per-step state lifecycle: Sense → Interpret → Propose → Authorize → (APPROVED) → Execute → Verify → Learn; Authorize → (REJECTED) → loop-terminate; Authorize → (DEFERRED) → loop-pause-pending-human-review.
- **Error handling:** Step-level failures emit step-specific Audit Log records; Authorize Step rejections terminate the loop without further Execute steps; Verify Step mismatches may pause subsequent loop iterations pending review.
- **Performance characteristics:** Sense, Interpret, Propose are typically <100ms each under standard agent profiles; Authorize is <10ms when no human review needed; Execute latency is dominated by underlying action (API call, document edit, etc.); Verify and Learn are typically <100ms each.

Cross-Family Integration

Upstream Dependencies. Family I operates within QPN-enabled infrastructure per the §22.7 Wherein clause and depends on Family A's Sidecar pattern (the AgentVector from Family H is the operative substrate for emitting per-step Witness Records). Family I's Authorize Step references Family K's Trust Criteria for participant standing and Family K's ACPA records for delegation evaluation. The Phase A reframing recommendation for Family I Claim 226 (moving deterministic-replay + per-step Trust-Criteria gate to the independent claim) reinforces these dependencies as protocol-grade.

Downstream Consumers. Family I's seven-step record sequences flow downstream through the same chain as Family A outputs: Family C Pipeline → Family D Ledger → Family E Graph → Family G Settlement. The DORMANT QPC State primitive consumed by Family K's Behavioral Activation: when a DORMANT QPC transitions to ACTIVE via the PPCS Activation Gate, Family I's accumulated state is integrated into the participant's Family K Behavioral Activation state and retroactive settlement entitlements are computed per Family J's Premium schedule at each CLE's original admission block height.

Lateral Interactions. Family I's per-step Trust-Criteria gate (Authorize Step) interoperates with the QPN-wide Trust Criteria infrastructure — the gate evaluates proposed actions against participant Trust Criteria, sponsor Trust Criteria (for Family B Institutional Mode contexts), and Catalyst Network governance Trust Criteria. Multi-agent coordination (the Pass 2 multi-agent coordination embodiment) interoperates across multiple Family I instances under Trust-Block-bound inter-agent message envelopes.

Emergent System-Level Properties. Family I's Governed Agent Loop produces an emergent property: AI-agent activity at scale without abandoning per-action auditability. Conventional agent architectures produce opaque batch outputs; Family I produces discrete per-step records that downstream Families consume identically to human-originated Witness Records. This protocol-grade uniformity is what permits the Catalyst Network to integrate AI-agent contributions into the same Contribution Graph, Reputation Engine, Premium computation, and Settlement Controller as human contributions, without requiring separate adjudication of agent contributions.

Family J — Premium Framework as Operative Allocation Mechanism

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -24, -25, -26, QPN Universal Exchange, Ownership, AI & Abundance-9.

Field Summary

Premium Framework as operative allocation mechanism: 15-dimensional Premium parameterization, AI-mediated quid-pro-quo-rebutting allocation, Premium Multiple Compression Curve, and Adaptive Premium Compensation.

Family J is the operative allocation mechanism of the Catalyst Network. Per and, the Premium Framework converts AI Pipeline outputs (Family C) into settlement-relevant weights through a multi-dimensional Premium schedule indexed by contribution modality, Mode-Tag, content characteristics, and the Premium Multiple Compression Curve. The Compression Curve is the temporal-discount primitive that compresses early-stage Premium multipliers toward an asymptotic baseline as the Catalyst Network matures.

Problem Addressed

Reward allocation systems face a structural compliance tension: explicit contribution-to-payment exchange exposes the operator to quid-pro-quo characterization (with bribery, inducement, or unregistered-securities consequences depending on context), while ungoverned allocation collapses into operator discretion. Prior systems either (a) tightly couple contribution to payment with explicit pricing, or (b) operate as fully discretionary reward programs with no structural compliance gating. Neither resolves the tension.

Conventional reward systems for contribution-based networks suffer from a calibration problem: fixed reward schedules under-reward high-impact contributions and over-reward low-impact contributions, while dynamic reward schedules without protocol-grade discipline become vulnerable to gaming. The Premium Framework addresses this with a parameterized, deterministically-computable schedule whose parameters are themselves governance-bound and Trust-Block-anchored.

A second problem is temporal: early contributors to a growing network warrant higher Premium multipliers than late contributors (reflecting the marginal-value-of-early-participation property), but those multipliers must compress as the network matures or the system becomes structurally inflationary. Conventional networks address this with ad-hoc reward halvings or governance-discretion adjustments; the Premium Multiple Compression Curve addresses it with a deterministic compression function whose parameters are Trust-Block-anchored and whose progression is independently auditable.

Solution Overview

The Premium Framework is a 15-dimensional parameterized allocation mechanism: 5 Launch Premiums (governing early-stage allocation), 8 Governance Premiums (governing ongoing allocation by governance role), 2 Adaptive Premiums (Proportionality and Balance, governing runtime stability). Each Premium is a multiplier applied during settlement allocation per Family G. The Premium Framework operates as a Manager-Discretion AI Model: the Catalyst Network Manager AI exercises bounded discretion within the Premium parameterization, rebutting any quid-pro-quo characterization by virtue of the AI's structural intermediation. The Premium Multiple Compression Curve governs how Premium Multiples decay over the maturity arc, preventing perpetual premium accrual. Adaptive Premium Compensation (Proportionality + Balance) prevents Premium runaway (autonomous-system instability).

Each contribution's Premium Score is computed as a deterministic function of (i) the AI Pipeline's Categorization Record (modality, topic, claim structure), (ii) the Verification Envelope's Mode-Tag (Family B), (iii) the participant's Reputation derivation (Family E), (iv) the time-of-contribution Compression Curve position, and (v) the Manager-discretion adjustment factor (bounded by Trust Criteria). The function is deterministically replayable: any auditor can recompute a contribution's Premium Score given the inputs and the Trust-Block-anchored Premium schedule version.

Compression Curve integration is critical for the broader QPN architecture: Family L Senior Derivatives index their Accrual Rights Swap valuations to the Compression Curve; Family N Liquidity Pool pricing references the Compression Curve; Family O Quantum DNA inheritance computes resource Premium under the Compression Curve. The Curve is therefore not just an internal Catalyst Network parameter but an architectural primitive shared across Families.

Components

Components detailed below: Premium Schedule, Premium Score Computer, Compression Curve, Manager-Discretion Adjustment, Premium Audit Bundle. Sources:, §3.4.1 (Compression Curve), and

Launch Premiums (5)

Premium dimensions governing early-stage allocation: Pioneer Premium, Cascade Premium, Resource Pool Genesis Premium, Anchor Premium, Liquidity Genesis Premium. Each parameterized by participant role, contribution timing, and Resource Pool genesis status.

The five Launch Premiums are protocol-anchored Premium dimensions activated during the Pioneer stage: (1) Originality Premium (rewarding first-attribution contributions), (2) Audience Premium (rewarding contributions with high subsequent-citation density), (3) Synthesis Premium (rewarding contributions that integrate multiple prior contributions per Family E graph topology), (4) Calibration Premium (rewarding contributions whose claims

match subsequent verification), (5) Cascade Premium (rewarding contributions that trigger high-volume downstream contributions). Each Premium dimension has a Trust-Block-anchored multiplier schedule. Per, the five Launch Premiums collectively shape the Pioneer-stage incentive envelope; their relative weights are configurable per Privacy Domain Trust Criteria.

Governance Premiums (8)

Premium dimensions governing ongoing allocation by governance role: Steward Premium, Auditor Premium, Connector Premium, Curator Premium, Verifier Premium, Resolver Premium, Advocate Premium, Operator Premium. Each parameterized by Reputation Vector (Family E) component.

The eight Governance Premiums extend the Launch Premium set with governance-specific dimensions: (1) Reviewer Premium, (2) Curator Premium, (3) Moderation Premium, (4) Dispute Resolution Premium, (5) Cross-Domain Bridging Premium, (6) Onboarding Premium, (7) Stewardship Premium, (8) Public-Interest Premium. Each Governance Premium recognizes a distinct governance role in the Catalyst Network. Per, Governance Premiums are configured by Privacy Domain governance Trust Criteria — different Privacy Domains may emphasize different governance roles.

Adaptive Premiums (2)

Proportionality Premium: bounds Premium magnitude relative to underlying contribution magnitude. Balance Premium: cross-Premium normalization preventing any single Premium dimension from dominating.

The two Adaptive Premiums are dimensions that adapt dynamically based on Catalyst Network state: (1) Scarcity Premium (boosting Premium for contribution classes where supply is low relative to demand), (2) Balance Premium (adjusting Premium to maintain healthy ratios across Premium dimensions). Adaptive Premium computations reference the current Catalyst Network state at the evaluation block height; adjustments are Trust-Block-bound and bounded by Trust-Criteria-anchored adjustment caps to prevent runaway adaptation. Per

Premium Multiple Compression Curve

Algorithmic schedule reducing Premium Multiples over the network's maturity arc. Parameterized by epoch (Pioneer / Cascade / Automated / Self-Funding) with epoch-specific compression rates.

The Premium Multiple Compression Curve is the protocol-anchored function compressing Premium multipliers over network maturity. Schema: curve_uuid, curve_trust_block_uuid, curve_function_descriptor (typical structure: piecewise-linear or piecewise-exponential decay from initial Pioneer-stage multipliers — typical 5-10× at network launch — toward an asymptotic baseline — typical 1.0× at network maturity), curve_position_at_block_height (deterministic function mapping block height to curve position). Per and Family L Compression-indexed Senior Derivatives, the Curve is referenced across multiple Families as the canonical temporal-discount primitive. Curve parameter rotations are Trust-Block-bound and apply only to contributions admitted after the rotation block height.

Manager-Discretion AI Model

AI model operating within a Manager-AI-scoped QPC. Inputs: contribution Pipeline Attestation, participant Reputation Vector, current Premium parameterization, network epoch. Output: per-Premium allocation decision. Operates deterministically (replay primitive); decisions are Trust-Block-bound.

The Manager-Discretion AI Model is the substitution primitive permitting Catalyst Network Managers to swap alternative Premium-computation models for specific Privacy Domains. Schema: discretion_model_uuid, replacing_default_model_uuid, target_privacy_domain_set, governance_reserve_publication_reference (the Trust-Block-anchored publication binding the model's behavior to public accountability governance discipline), adjustment_factor_bounds (the [floor, ceiling] within which the model's Premium adjustments must remain). Per, Manager-Discretion model substitution is constrained by Trust Criteria — managers cannot unilaterally exceed the adjustment-factor bounds; exceedance attempts emit Audit Log records and are routed to governance review.

Quid-Pro-Quo Rebuttal Layer

Structural compliance feature: the Manager AI's intermediation between contribution and allocation breaks the direct exchange, supporting characterization as discretionary recognition rather than transactional payment.

Reinforced by Premium dimensions that depend on factors other than the immediate contribution (Reputation, network epoch, Balance).

The Quid-Pro-Quo Rebuttal Layer is the defense primitive against coordinated reciprocal-citation rings that artificially inflate Premium scores. Algorithm: at each Premium-computation cycle, the Layer (i) identifies citation pairs (A→B, B→A) within configurable temporal windows; (ii) applies a discount factor to citations meeting reciprocity criteria; (iii) emits REBUTTAL_APPLIED records on the Audit Log specifying the discount applied. Per, the Layer's discount function is Trust-Block-anchored; participants may rebut the rebuttal under explicit Trust-Block-bound rebuttal processes if the citation reciprocity reflects legitimate intellectual exchange rather than coordinated gaming.

Premium Runaway Prevention

Adaptive feedback enforcing Proportionality and Balance: if a Premium dimension's recent allocations exceed Proportionality bounds, the dimension's effective multiplier is reduced for the next allocation cycle.

The Premium Runaway Prevention mechanism caps per-participant aggregate Premium accrual within configurable temporal windows to prevent Premium-multiplier compounding cycles. Schema: prevention_policy_uuid, participant_premium_cap_per_window, window_duration_blocks, cap_excess_routing_policy (the policy specifying where Premium accruals exceeding the cap are routed — typical: redirection to Accelerator Participant Pool, AIIP, or Backing Pool). Per, the prevention is structural rather than discretionary: cap excess routing is deterministic and Trust-Block-bound; cap parameters are governance-amendable but amendments do not retroactively redistribute prior cap excesses.

Process Flow

Upon a Pipeline Attestation reaching the Settlement Controller (Family G), the Premium Framework computes the Premium-adjusted allocation.

The Manager-Discretion AI Model evaluates each of the 15 Premium dimensions for the contribution and the participant.

Each dimension's multiplier is fetched from the current Premium Multiple Compression Curve given the network epoch.

The Adaptive Premium layer applies Proportionality and Balance constraints; out-of-bound multipliers are clipped.

The composite Premium-adjusted allocation is presented to the Settlement Controller's Issuance Component.

The allocation decision and its Premium-dimension breakdown are recorded on the Settlement Ledger.

Alternative Embodiments

Per-Accelerator Premium customization: each Accelerator may parameterize its own Premium overrides within Governance-defined bounds.

Domain-specific Premium addition: domain Accelerators (Healthcare, AI Trust & Safety, etc.) may add domain-specific Premium dimensions per their domain governance.

Time-varying Premium embodiment: Premium dimensions may be configured to vary on schedules (e.g., increased Connector Premium during a campaign).

Reputation-conditional Premium embodiment: Premium dimensions may be conditioned on participant Confidence Tier (Family E): a contribution from a TIER_PLATINUM participant may receive a Premium boost reflecting the participant's track record; a contribution from a TIER_BRONZE participant follows standard Premium computation. Reputation-conditional Premiums are Trust-Block-anchored.

Cross-Domain Premium embodiment: Premium dimensions may apply with different weights across Privacy Domains: a healthcare Privacy Domain may weight Calibration Premium more heavily than Audience Premium; a journalism Privacy Domain may weight Originality Premium more heavily than Synthesis Premium. Cross-domain Premium weights are configurable per domain Trust Criteria.

Cohort-relative Premium embodiment: Premium computation may operate relative to a participant cohort rather than absolute: a contribution's Premium may be computed as percentile-rank within a cohort of contemporary contributions of the same Category and UX Tier. Cohort-relative Premiums normalize for cross-period Premium-scale drift.

Anti-volatility Premium embodiment: Adaptive Premiums may include anti-volatility components dampening rapid Premium-scale shifts: if a Premium dimension's average value swings more than configurable percentage in a temporal window, dampening adjustments apply. Anti-volatility dampening is Trust-Block-anchored and bounded.

Sponsor-bonus Premium embodiment: Sponsors (Family B Institutional Mode) may add sponsor-specific bonus Premium multipliers to their sponsored participants' contributions. Sponsor bonuses are funded from the sponsor's Trust-Block-bound bonus pool; sponsor pools are Trust-Block-anchored at sponsor's commitment level.

Decay-aware Premium embodiment: Premium dimensions may incorporate explicit decay functions: Originality Premium may decay over time as the contribution's claims become widely known; Synthesis Premium may decay as the synthesized components become individually well-known. Decay functions are configurable per Premium dimension and Trust-Block-anchored.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Premium computation:** (i) the Pipeline Completion Record must be present and valid; (ii) the operative Premium schedule snapshot must be Trust-Block-anchored at the computation block height; (iii) the Quid-Pro-Quo Rebuttal Layer's reciprocity windows must be computable.
- **Postconditions:** (i) a Premium Score Record is emitted Trust-Block-bound; (ii) Premium Runaway Prevention has applied; (iii) downstream Settlement Controller computations (Family G) consume the Premium Score Record.
- **State transitions:** Premium computation cycle: SCHEDULED → COMPUTING → REBUTTAL_APPLIED → CAP_CHECKED → EMITTED.
- **Error handling:** Compression Curve evaluation failures (e.g., curve definition unavailable at the evaluation block height) trigger CURVE_UNAVAILABLE states; Manager-Discretion model out-of-bounds adjustments emit MANAGER_DISCRETION_REJECTED records and revert to default model.
- **Performance characteristics:** Premium computation per CLE is O(D) per D Premium dimensions evaluated; Compression Curve lookup is O(1); Quid-Pro-Quo Rebuttal evaluation is O(C) per C concurrent citation pairs in the temporal window.

Cross-Family Integration

Upstream Dependencies. Family J consumes Pipeline Completion Records from Family C, Mode-Tags from Family B, Reputation outputs from Family E, and contribution-category classifications from Family K Catalyst Contribution Categories. The Compression Curve is itself an architectural primitive shared across multiple Families: Family L Accrual Rights Swap valuations index to the Curve; Family L Compression-indexed Senior Derivative payouts index to the Curve; Family N Backing Pool participation distributions reference the Curve; Family O Premium Inheritance Engine references the Curve.

Downstream Consumers. Family G (Settlement Controller) is the primary consumer of Family J Premium Score Records for settlement issuance. Family L (Senior/Junior QPT Derivatives) consumes Premium Score Records for derivative accrual computation. Family M (Stage-Differentiated Revert + MUM) consumes settlement amounts derived from Premium computation. Family K (Behavioral Activation) consumes Premium outputs for XP awards. Family O (Premium Inheritance Engine) is consumed across descendant Resource Derivative Premium computations.

Lateral Interactions. Family J's Manager-Discretion AI Model primitive interoperates with Family C's Stage 3 (Temporal Durability) — Manager-Discretion substitutions adjust Stage 3 outputs that flow into Premium Score computation. The Compression Curve interoperates with Family J's Adaptive Premiums — Adaptive Premium

adjustments are bounded by Compression-Curve-position-conditional limits. The Quid-Pro-Quo Rebuttal Layer interoperates with Family E's Contribution Graph — citation reciprocity is evaluated against graph topology.

Emergent System-Level Properties. Family J participates in four distinct cross-family integration patterns: (a) Pattern #2 Family B + Family J Mode-conditional Premium discipline; (b) Pattern #5 Family G + Family J deterministic settlement composition; (c) Pattern #6 Family J Compression Curve + Family L Accrual Rights Swap valuation; (d) Pattern #12 Family J Premium + Family O Premium Inheritance Engine for Resource Derivative inheritance. The Phase A reframing recommendation for Family J Claim 255 (adding Compression Curve integration to the independent claim) reinforces (c) and (d). Together these patterns make Family J the temporal-economic-discipline backbone of the QPN: every settlement flow, every derivative valuation, every inherited Resource Derivative Premium is deterministically computable and Trust-Block-anchored.

Family K — Behavioral Activation, Reputation & Identity Resilience

Encompassing candidate disclosures: QPN Catalyst Launch Plan & Rewards Framework -19, -21, -28, -29, -30.

Field Summary

Six-Layer Catalyst Architecture umbrella, Behavioral Activation System gamification, Multi-Factor Identity Binding, Authorized Catalyst Proxy Addresses, and Catalyst Contribution Categories taxonomy.

Family K is the participant-lifecycle layer of the Catalyst Network. Per and and §22.1, the Behavioral Activation, Reputation, and Identity Resilience architecture provides protocol-grade primitives for: (i) Behavioral Activation, the deterministic transition of a participant from DORMANT to ACTIVE state; (ii) Reputation derivation from the Contribution Graph; (iii) Multi-Factor Identity Binding for resilience against credential loss; (iv) Authorized Catalyst Proxy Addresses for delegated participation.

Problem Addressed

Catalyst networks must (a) provide a structured architectural decomposition for operator, governance, and participant accountability; (b) sustain participant engagement at scale through gamification primitives that align with Trust Block accountability rather than defeating it; (c) bind multi-factor identity attestations to participant Trust Blocks; (d) support delegated contribution capture (assistant acting on behalf of principal) without breaking attribution; and (e) provide a contribution-category taxonomy supporting category-specific authorization and weighting.

Participant lifecycle in conventional attribution networks is brittle. A participant who loses primary credentials loses their entire attribution history; a participant who wishes to delegate participation to an authorized proxy faces no protocol-grade primitive for doing so; a participant whose activation event is contested has no deterministic audit trail. Identity Resilience is therefore both a security property and an attribution-continuity property.

The Multi-Factor Identity Binding primitive addresses credential loss by binding a participant's protocol identity to a tuple of independent identity factors (cryptographic credential, biometric attestation, sponsor attestation, behavioral signature) such that any K-of-N factors suffice to authenticate. Loss of any single factor does not destroy the participant's protocol identity; rotation of any single factor is itself Trust-Block-bound. Authorized Catalyst Proxy Addresses extend this to delegated participation: a participant may authorize a specific delegate to act on their behalf under bounded Trust Criteria, with the delegation itself auditable.

Solution Overview

The Six-Layer Catalyst Architecture decomposes the network into User-Side, Network-Side, AI-Mediation, Governance, Settlement, and Identity layers, with cross-layer Trust Block bindings. The Behavioral Activation System provides 10 levels, 24 badges, dual-XP currency (contribution XP and behavior XP), evidence bonus (contributions accompanied by Cross-Verification attestation earn bonus XP), and streak multipliers. Multi-Factor Identity Binding combines multiple identity attestations (phone-binding, platform-binding, biometric-binding, government-ID-binding) into a single Identity Trust Block per participant. Authorized Catalyst Proxy Addresses permit delegated contribution capture with proxy-address-level Trust Block scope. Catalyst Contribution Categories provide Nine Primary Categories plus a UX Tier and Program→Objectives mapping.

Behavioral Activation operates as a deterministic state transition from DORMANT QPC State (Family I) to ACTIVE state, triggered by participant-originated self-activation events satisfying Trust Criteria thresholds. The transition itself is Trust-Block-bound and produces a Reputation initialization event in Family E. Reputation derivation thereafter follows the deterministic-assembly discipline of Family E.

The structural-incapacity-to-execute property is critical: a participant in DORMANT state is structurally incapable of executing economic operations (e.g., settlement issuance, derivative trade, liquidity participation). This is not a policy constraint but a cryptographic constraint: the participant's protocol identity does not yet possess the Trust Block authorities required to execute. Self-activation explicitly issues those authorities, and the issuance is itself Trust-Block-bound and auditable.

Components

Components detailed below: Behavioral Activation Manager, Multi-Factor Identity Binder, Authorized Proxy Address Manager, Reputation Initializer, Lifecycle Audit Logger. Sources:, §3.8 (Identity Resilience), and

Six-Layer Catalyst Architecture

User-Side Layer (Sidecar, Vectors,.qpn archive), Network-Side Layer (Catalyst Network ingress/egress, Cross-Verification Protocol), AI-Mediation Layer (Three-Stage Pipeline, Manager-Discretion Model), Governance Layer (Manager controls, Permitted Audit Purposes), Settlement Layer (Settlement Controller, Settlement Ledger), Identity Layer (Multi-Factor Identity Binding, Proxy Addresses). Cross-layer bindings via Trust Block schema.

The Six-Layer Catalyst Architecture decomposes Catalyst Network participation into six functional layers: (1) User Side (Sidecar per Family A), (2) Network Side (Settlement Controller per Family G), (3) AI Mediation (three-stage pipeline per Family C), (4) Governance (Trust Block authority chain per Family D), (5) Settlement (Settlement Ledger and Premium computation per Families D and J), (6) Identity (Multi-Factor Identity Binding per components below). Each layer operates within layer-scoped QPCs with explicit cross-layer Trust-Block-bound message envelopes. Per and

Behavioral Activation System — Levels

10-level progression from Apprentice (Level 1) to Architect (Level 10), with level-specific Premium access and capability grants.

The Levels subsystem categorizes participants into discrete behavioral-activation levels based on cumulative contribution, governance participation, and Reputation indicators. Schema: level_assignment_uuid, participant_did, current_level (enum: NOVICE, ENGAGED, ACTIVE, COMMITTED, STEWARDSHIP, FOUNDATIONAL), level_threshold_satisfaction_reference_set (Trust-Block-anchored references to threshold-satisfaction events), level_transition_block_height. Level transitions confer Trust-Block-bound capability extensions (e.g., STEWARDSHIP level unlocks Cross-Domain Bridging Premium eligibility). Per

Behavioral Activation System — Badges

24 badges distributed across contribution badges (e.g., First Contribution, Hundred Contributions, Diverse Contribution), behavioral badges (e.g., Consistent Streak, Cross-Verifier, Mentor), and governance badges (e.g., Steward, Connector, Auditor).

The Badges subsystem complements Levels with achievement-specific recognition. Badges are independent of Levels (a participant may hold many badges at any Level). Per Family E Reputation Engine — Badge Component for the detailed badge schema and award mechanics. The Behavioral Activation System invokes the Family E Badge Component as a sub-component; Behavioral Activation-specific badges focus on engagement-pattern milestones (e.g., 30-DAY-STREAK, FIRST-CROSS-DOMAIN-CONTRIBUTION, MENTORSHIP-INTRODUCTION).

Behavioral Activation System — Dual XP

Two XP currencies: Contribution XP (earned via accepted Pipeline Attestations weighted by Durability) and Behavior XP (earned via Cross-Verification, Mentorship, and compliance signals). Levels require thresholds in both currencies.

The Dual XP system maintains two parallel experience-point streams for each participant: Contribution XP (accumulated from contribution-type activities) and Governance XP (accumulated from governance-type activities — review, moderation, curation, dispute resolution). Schema: xp_record_uuid, participant_did,

xp_stream (enum: CONTRIBUTION, GOVERNANCE), xp_amount, source_event_reference. Per, Dual XP supports balanced participation incentives — a participant cannot reach FOUNDATIONAL Level on Contribution XP alone without also accumulating Governance XP.

Behavioral Activation System — Evidence Bonus

Contributions presented with Cross-Verification attestation (Family G) earn a 25%-50% Contribution XP bonus per the current Premium Compression epoch.

The Evidence Bonus subsystem awards multiplier bonuses to contributions accompanied by structured evidence (citations, datasets, reproducibility attestations, peer-review references). Schema: evidence_bonus_uuid, contribution_cle_reference, evidence_class_set (enum-set across evidence types), evidence_quality_attestation_set, bonus_multiplier. Per, the Evidence Bonus is Trust-Block-bound and configurable per Privacy Domain — domains with higher evidence standards (e.g., scientific-research domains) may configure higher evidence-bonus multipliers than domains with lower evidence-standard requirements.

Behavioral Activation System — Streak Multiplier

Consecutive-day contribution streaks earn a streak multiplier (capped to prevent grinding) applied to both XP currencies.

The Streak Multiplier rewards sustained contribution rhythm. Schema: streak_record_uuid, participant_did, streak_class (DAILY, WEEKLY, MONTHLY), streak_length, streak_multiplier_current. Per, Streak Multiplier values increase with streak length up to Trust-Block-anchored caps; streak interruption resets the multiplier with grace-period configurations supporting legitimate hiatus events (travel, illness, life events) attested under Trust-Block-bound hiatus declarations.

Multi-Factor Identity Binding

Combines multiple identity attestations into a single Identity Trust Block. Each factor (phone-binding, platform-binding, biometric-binding, government-ID-binding) is a separately-verified attestation; the composite Identity Trust Block records the set of bound factors and emits the composite Identity Confidence Score consumed by Family E.

Multi-Factor Identity Binding maintains a participant's identity binding across multiple independent identity factors. Schema: identity_binding_uuid, participant_did, factor_set (each factor specifying factor_class [CRYPTOGRAPHIC_KEY, BIOMETRIC_ATTESTATION, SPONSOR_ATTESTATION, BEHAVIORAL_SIGNATURE, DEVICE_ATTESTATION], factor_descriptor, factor_authentication_block_height), k_of_n_threshold (the minimum number of factors required for authentication — typical 2-of-4 for individual participants, 3-of-5 for institutional accounts). Factor rotation is Trust-Block-bound; loss of any single factor is recoverable through the remaining factors. Per and

Authorized Catalyst Proxy Addresses

Proxy-attribution primitive: a principal participant may authorize a proxy address (e.g., personal assistant, agentic agent) to capture contributions on their behalf. The proxy address operates with proxy-scope Trust Block; proxy-captured contributions are attributed to the principal subject to the proxy's authorization scope.

Authorized Catalyst Proxy Addresses (ACPAs) are the delegation primitive permitting a participant to authorize a delegate to act on their behalf within bounded scopes. Schema: acpa_uuid, delegating_participant_did, delegate_did, delegated_scope_descriptor (Trust-Criteria-bound scope — e.g., "may submit CLEs in CodeVector but not VoiceVector; may not exercise economic operations"), delegation_expiry_block_height, delegation_revocation_predicate. Per, delegations are unilateral revocable by the delegating participant; delegations may also be configured to require dual-signature for high-stakes actions (both delegating participant and delegate must sign).

Catalyst Contribution Categories — Nine Primary

Nine primary contribution categories: Original Authorship, Curation, Cross-Verification, Mentorship, Governance Action, Resource Provision, Connection, Outreach, Compliance Service. Each category has category-specific Trust Criteria, Premium weights, and authorization rules.

The Nine Primary Catalyst Contribution Categories are the protocol-anchored taxonomy of contribution types: (1) Information Production, (2) Knowledge Synthesis, (3) Decision Support, (4) Calibration & Verification, (5) Connection & Introduction, (6) Mentorship & Onboarding, (7) Governance & Stewardship, (8) Infrastructure Maintenance, (9) Public-Interest Service. Each category has category-specific Premium multipliers (Family J) and category-specific Witness Solicitation policies (Family G). Per

Catalyst Contribution Categories — UX Tier

An additional UX tier for user-experience contributions (design, accessibility, localization) operating across the Nine Primary Categories.

The UX Tier subsystem cross-cuts the Nine Primary Categories with user-experience-quality tiers: BRONZE (minimum qualifying), SILVER (enhanced presentation, accessibility, completeness), GOLD (exemplary user-experience — clear writing, comprehensive citations, accessible formatting), PLATINUM (outstanding contribution serving as exemplar). UX Tier is determined by the three-stage AI Pipeline (Family C) at admission and is Trust-Block-bound. Per

Catalyst Contribution Categories — Program→Objectives Mapping

Mapping from operator-defined Programs (campaigns, initiatives) to underlying contribution categories and objective metrics; permits Program-scoped Premium configurations.

The Program→Objectives Mapping subsystem aligns Catalyst Contribution Categories with sponsor-specific or domain-specific program objectives. Schema: program_uuid, sponsor_did, objective_set (each objective specifying target Contribution Categories, target UX Tiers, target volume / cadence / quality metrics), program_premium_multiplier_set (program-specific Premium adjustments for contributions matching objective criteria). Per, Program →Objectives mappings enable sponsor-specific incentive alignment (e.g., a healthcare research sponsor's Program may boost Premium for Category 2 [Knowledge Synthesis] at GOLD UX Tier addressing specific research questions).

Process Flow

On each accepted Pipeline Attestation, the Behavioral Activation System computes XP credits across both currencies, applies evidence bonus and streak multiplier, and updates participant level if a threshold is crossed.

On level transition, level-specific Premium access and capability grants are activated.

On identity factor binding, a new Identity Trust Block is composed including the new factor; Family E Reputation Engine re-evaluates the participant's Identity Dimension.

On contribution capture, the Mode Discriminator (Family B) and Catalyst Contribution Categories taxonomy classify the contribution into a Mode × Category pair; the Settlement Controller (Family G) applies Category-specific authorization rules.

On proxy authorization, a Proxy Trust Block records the principal-to-proxy delegation and the proxy's authorization scope; proxy-captured contributions are attributed to the principal subject to scope.

Alternative Embodiments

- **Per-Accelerator Six-Layer customization**: each Accelerator may add a domain-specific seventh layer (e.g., Clinical Layer for healthcare) without disturbing the base Six.
- **Seasonal Behavioral Activation configuration**: badge sets and streak multipliers may be reconfigured seasonally for campaigns.
- **Identity factor expansion**: as new identity-attestation modalities emerge (e.g., post-quantum-resistant cryptographic identity), new factor types may be added to the Multi-Factor Identity Binding.
- **Cross-Privacy-Domain Identity Binding embodiment**: Multi-Factor Identity Binding may span multiple Privacy Domains: a participant may bind a single identity to multiple Privacy Domain memberships with consistent k-of-n threshold across domains. Cross-domain Identity Binding supports unified-identity-management for participants active in multiple domains.

- **Recovery-key Identity Binding embodiment**: Identity Binding may include recovery-key factors: cryptographic keys stored with trusted recovery custodians (family members, attorneys, recovery services) that can be invoked to restore identity binding upon loss of primary factors. Recovery-key invocations are Trust-Block-bound and subject to time-delayed activation to prevent unauthorized recovery.
- **Biometric-revocation Identity Binding embodiment**: Biometric factors may be configured with revocation primitives: if biometric data is compromised (e.g., breach of biometric template database), the biometric factor may be revoked and re-issued. Revocation events are Trust-Block-bound and immediate.
- **Delegated-organization ACPA embodiment**: ACPAs may be issued to organizational delegates (a participant's employer, a participant's healthcare provider) under bounded organizational scope. Organizational ACPAs include explicit conflict-of-interest disclosures and are revocable upon organizational relationship termination.
- **Lifetime-bounded ACPA embodiment**: ACPAs may be configured with lifetime bounds rather than expiry-block bounds: lifetime-bounded ACPAs persist until specific events (delegate retirement, participant death, organizational dissolution) — supporting long-duration delegation use cases. Lifetime-bound ACPAs require Trust-Block-bound termination-event detection.
- **Streak-resilient Streak Multiplier embodiment**: The Streak Multiplier may incorporate resilience mechanisms beyond grace-period hiatuses: streak-rescue tokens that participants may invest to preserve streaks during legitimate hiatus events; streak-recovery mechanisms allowing partial streak restoration after interruption; streak-transfer mechanisms supporting team-collaborative streaks.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Behavioral Activation operations:** (i) the participant's Multi-Factor Identity Binding must be authenticated under the operative k-of-n threshold; (ii) the participant's current Level and XP state must be loaded; (iii) the operative Behavioral Activation policy must be Trust-Block-anchored.
- **Postconditions:** (i) XP updates emit Trust-Block-bound XP records; (ii) Level transitions (where triggered) emit Level Transition records; (iii) Badge awards (where triggered) emit Badge records via the Family E Badge Component.
- **State transitions:** Level transitions: NOVICE → ENGAGED → ACTIVE → COMMITTED → STEWARDSHIP → FOUNDATIONAL; rare downgrade transitions permitted under Trust-Block-bound downgrade authority (e.g., for governance-breach events).
- **Error handling:** Identity authentication failures emit AUTH_FAILED records and block subsequent operations; ACPA scope violations emit SCOPE_VIOLATION records and reject the delegated operation.
- **Performance characteristics:** Level / XP updates are O(1) per event; Multi-Factor Identity Binding authentication is O(n) per n active factors (typically <10ms); ACPA scope evaluation is O(1) per delegated operation.

Cross-Family Integration

Upstream Dependencies. Family K's Behavioral Activation, Reputation, and Identity Resilience primitives operate within QPN-enabled infrastructure per the §22.7 Wherein clause. Family K consumes Family A Witness Records for activity tracking, Family E Reputation Engine outputs for Level transitions and Badge awards, Family G Settlement Ledger transfers for XP awards, and Family I PPCS Activation Gate events for DORMANT-to-ACTIVE transitions. The structural-incapacity-to-execute property in DORMANT state (per the Phase A reframing recommendation for Family K Claim 291) is the cryptographic primitive enforcing operational segregation.

Downstream Consumers. Family K's Multi-Factor Identity Binding state is consumed by every Family requiring participant authentication — Family A (Sidecar Bridge signing-key access), Family C (Pipeline ingress verification), Family G (settlement payee verification), Family L (derivative-holder verification), Family N (Liquidity Provider verification). Family K's ACPA delegations are consumed by all Families respecting delegated authority. Family K's Level / XP / Badge state is consumed by Family J as a Premium-eligibility input.

Lateral Interactions. Family K's Six-Layer Catalyst Architecture is the meta-framework within which all other Families operate; the six layers (User Side / Network Side / AI Mediation / Governance / Settlement / Identity) map onto Families A / G / C / D / G+J / K respectively. The Pattern #4 graph re-binding interaction with Family E is detailed in Family E's Cross-Family Integration above.

Emergent System-Level Properties. Family K's k-of-n-threshold Multi-Factor Identity Binding combined with Family E's deterministic-assembly Contribution Graph produces credential-loss resilience at the protocol level. The emergent property is essential for population-scale Catalyst Network adoption: individual participants face credential-loss events (lost phone, hardware failure, compromised biometric template) over multi-decade time horizons; without protocol-grade resilience, those events would entail attribution-history loss and reputation regression. Family K's primitives ensure these are recoverable rather than catastrophic.

Family L — Senior/Junior QPT Derivative Capital-Formation Architecture

Encompassing candidate disclosures: PVRF-1, PVRF-2, PVRF-3, PVRF-10.

Field Summary

Senior/Junior QPT Derivative capital-formation architecture: Senior derivative with dual-hurdle RBF, encumbrance authority over Exchange Root + AIIP allocations, junior derivative offset, accrual rights swap.

Family L is the capital-formation layer of the QPN. Per Participation, Valuation, Rewards & Financing Model (PVRF) §§2.1, 2.6 and, the Senior/Junior QPT Derivative architecture provides pre-settlement institutional capital formation through a dual-instrument derivative stack: Senior QPT Derivatives priced under Dual-Hurdle Revenue-Based Financing (IRR target + MOIC backstop), and Junior QPT Derivatives absorbing first-loss exposure at configurable offset levels. The architecture is anchored by Encumbrance Authority over the Exchange Root Token allocation and the Accelerator Incentive & Investment Pool allocation.

Problem Addressed

Pre-settlement capital formation for tokenized networks faces a structural challenge: investors require exposure to future settlement flows before the network has generated material settlement volume, while the network requires that pre-settlement capital not capture native protocol tokens (which would conflate investment with operational participation and create regulatory exposure). Prior tokenized network designs either (a) sell tokens directly to investors (collapsing the distinction) or (b) decline pre-settlement capital formation entirely (foreclosing institutional investor pathways).

The conventional path for tokenized-network capital formation is direct token issuance to investors, which collapses the distinction between investment exposure and operational participation, generates regulatory exposure under securities classifications across multiple jurisdictions, and prevents the network from preserving its native token allocations for operational participants. The alternative — foregoing pre-settlement capital formation — is incompatible with the capital requirements for Pioneer-stage network bootstrapping at meaningful scale.

The Senior/Junior QPT Derivative architecture addresses this by introducing a derivative-instrument class structurally distinct from the native QPT tokens themselves. Senior Derivatives are bounded-duration capped structured credit backed by the combined Exchange Root + Accelerator Token pool; Junior Derivatives are subordinated first-loss instruments providing additional Senior coverage. Per, the structure's coverage ratios are extraordinary: the combined backing pool's 75-year QPT Derivative NPV exceeds the cap amount by hundreds of thousands of times for the most senior Pioneer-phase tranches (approximately 950,000× for a $1B tranche at 3× cap multiple). The binding investor risk is timing of cap satisfaction, not whether cap satisfaction will occur.

A second problem is the heterogeneity of institutional-investor mandates: pension funds, sovereign wealth funds, endowments, family offices, fund-of-funds, strategic enterprises, hyperscalers, financial institutions, governments, and philanthropic capital each have distinct return-target, duration-tolerance, and risk-allocation requirements. A single fixed-parameter derivative cannot satisfy this mandate universe; a parameterizable derivative class can. The dual-hurdle (IRR + MOIC) plus configurable Payment Term plus configurable PNX Settlement Accrual Preferences plus configurable Junior offset together produce a parameter set that accommodates the full mandate spectrum.

Solution Overview

The Senior/Junior QPT Derivative Capital-Formation Architecture provides a multi-instrument derivative stack: Senior QPT Derivative pays revenue-based financing returns subject to dual hurdles (target IRR + target MOIC) with issuer payoff optionality at hurdle achievement; the issuing entity holds contractual authority to encumber the full Exchange Root Token allocation and Accelerator Incentive & Investment Pool allocation under deferred-activation conditions; Junior QPT Derivative absorbs first-loss exposure for Senior holders at three configurable offset levels; an Accrual Rights Swap primitive enables exchange of accrual rights across Premium-parameterized cash flows. Together, these derivative instruments enable pre-settlement institutional capital formation without diluting native protocol token allocations.

Senior QPT Derivatives carry the following parameter set, each independently configurable per tranche: Payout Return Cap (the total amount of PNX Settlement Accruals distributed by end-of-term, expressed as cap-multiple of investment); MOIC Backstop (minimum absolute return, applicable in rapid-payoff scenarios); IRR Target (minimum annualized return, applicable in slow-payoff scenarios); Payment Term (target duration and distribution pattern, configurable across three standard structures); PNX Settlement Accrual Preferences (priority tier in the settlement waterfall, configurable across Baseline, Expanded, and Maximum tiers).

The dual-hurdle structure ensures that whichever of MOIC or IRR produces the higher lender return applies: the IRR target binds in slow-payoff scenarios (compensating for time-value uncertainty); the MOIC backstop binds in rapid-payoff scenarios (compensating for activation default risk). Pioneer-stage MOIC backstops are typically 1.5–2.0× under institutional-credit-style underwriting, ranging up to 3.0× under venture-style underwriting that prices QPN activation as a high-stakes early-stage technology bet. Pioneer-stage IRR targets are typically 25–35%.

Junior QPT Derivatives provide first-loss coverage at three configurable offset levels (1x, 2x, 3x Senior coverage), with higher offset commanding higher Junior yield. The Accrual Rights Swap primitive enables Senior or Junior holders to exchange accrual rights across Premium-parameterized cash flows, with swap valuations indexed to the Premium Multiple Compression Curve (Family J). Together these instruments enable a Pioneer-stage capital-formation envelope that no conventional structure (venture, private equity, structured credit) could achieve within the same deployment horizon.

Components

Detailed component disclosure follows. Sources: (Market Realizable Value over time, Dual-Hurdle architecture), §2.6 (Capped QPT Derivative architecture, coverage ratios, cap-multiple structure), §2.7 (Governance Reserve and Senior Derivative issuance mechanics with deferred-activation encumbrance), and

Senior QPT Derivative

Derivative instrument paying revenue-based-financing returns to holders subject to dual hurdles: target IRR (e.g., 18%) and target MOIC (e.g., 3.5x). Issuer holds payoff optionality at hurdle achievement: upon achieving both hurdles, the issuer may settle the derivative at par plus hurdle returns, terminating future payment obligations.

The Senior QPT Derivative is implemented as a Trust-Block-bound claim record with the following on-ledger schema fields: tranche_id (content-addressed identifier), issuer_did (issuing entity decentralized identifier, e.g., Quantum Privacy LLC's DID), holder_set (ordered set of holder DIDs and basis-point shares summing to 10,000 bps), payout_return_cap_multiple (rational number, typical range 1.5–5.0 with 1.5–2.0 binding under institutional-credit underwriting), moic_backstop_multiple (rational, typical range 1.5–3.0), irr_target_bps (integer basis points, typical range 1800–3500), payment_term_days (integer, configurable across three standard structures: short ≤2555 days ≈7 years, medium 2556–5475 days ≈7–15 years, long 5476–10950 days ≈15–30 years), settlement_accrual_preference_tier (enum: BASELINE, EXPANDED, MAXIMUM), and encumbrance_authority_uuid (reference to the Encumbrance Authority record). All numeric parameters are tranche-immutable post-issuance; mutations require a new Tranche record and explicit Junior-Senior consent under Trust Criteria. Per, the coverage-ratio computation coverage_ratio = backing_pool_75yr_npv / cap_amount is deterministic and replayable from on-ledger state at any specified ledger height.

Encumbrance Authority

Contractual primitive whereby the issuing entity (Quantum Privacy LLC) holds authority to encumber the full Exchange Root Token (ERT) allocation and Accelerator Incentive & Investment Pool (AIIP) allocation against future settlement flows. Encumbrance operates under deferred-activation conditions: encumbered flows are computed on the Settlement Ledger but value transfer is escrowed against derivative settlement.

The Encumbrance Authority is implemented as a Trust Ledger entry (Family D) of type ENCUMBRANCE_AUTHORITY referencing the issuing entity and the target backing-pool elements (ERT pool, AIIP pool, or specified subsets). Encumbrance operates under the deferred-activation pattern described in: the on-ledger encumbrance record is created at Senior Derivative issuance time, but value-transfer activation is gated by activation_condition predicates that reference Trust-Block-anchored events (e.g., "Settlement Ledger has recorded ≥N PNX Settlement events of class C"). Three configurable Settlement Accrual Preference tiers (BASELINE, EXPANDED, MAXIMUM) map to three target backing-pool subsets: BASELINE encumbers the 15% ERT Pool plus 10% of the AIIP assigned to the Governance Reserve Endowment; EXPANDED encumbers the 15% ERT Pool plus up to 85% of each EP3-managed AIIP not exchanged to the EP3 Network; MAXIMUM encumbers the full ERT Pool plus the full EP3-managed AT Pool. Tier transitions during a Senior tranche's life are forbidden unless a tranche-specific amendment record is signed by all Senior holders and the issuer; this protects Senior-holder priority guarantees from unilateral issuer downgrade.

Junior QPT Derivative

Subordinated derivative absorbing first-loss exposure for Senior derivative holders. Three configurable offset levels: 1x Senior coverage (Junior absorbs first loss up to 1x of Senior face), 2x, 3x. Higher offset levels command higher Junior yield.

The Junior QPT Derivative is implemented as a Trust-Block-bound subordinated claim with on-ledger schema fields paralleling the Senior schema plus a senior_coverage_multiple field (rational, typical values 1.0, 2.0, 3.0 per the three standard offset levels) and a subordination_waterfall_bps field defining the loss-absorption position. Junior loss-absorption is deterministically computed by the Settlement Controller: upon any Senior shortfall event (Senior tranche fails to achieve MOIC backstop within Payment Term), the shortfall amount is applied first against the Junior tranche's outstanding face value, computed as junior_absorption = min(senior_shortfall_amount, junior_face * senior_coverage_multiple). Junior yield is parameterized as a junior_yield_bps integer (typical Pioneer-stage ranges: 4000–7000 bps for 1× offset, 6000–10000 bps for 2× offset, 8000–14000 bps for 3× offset, reflecting the increased loss-absorption obligation). Junior-Senior linkage is enforced through a paired_senior_tranche_id field that cryptographically binds the Junior tranche to its Senior counterpart; unpaired Junior issuance is permitted only for general Backing Pool coverage roles

Accrual Rights Swap

Swap primitive enabling holders of accrual rights against one Premium-parameterized cash flow to exchange for accrual rights against another Premium-parameterized cash flow. Valuation indexed to the Premium Multiple Compression Curve (Family J). Enables holders to reconfigure exposure across Premium dimensions.

The Accrual Rights Swap primitive is implemented as a Trust-Block-bound exchange-of-claims protocol with the following operational sequence: (i) a party initiates a swap request specifying its surrendered accrual-rights bundle (referenced by content-addressed Settlement Ledger identifiers) and its requested accrual-rights bundle; (ii) a counter-party either signs the swap acceptance or proposes a counter-bundle; (iii) the Settlement Controller (Family G) verifies that both bundles' present-value computations under the Premium Multiple Compression Curve (Family J) satisfy a swap-tolerance predicate (typical tolerance: ±2% PV equivalence for standard swaps; configurable up to ±10% for over-the-counter bespoke swaps); (iv) upon verification, the Controller emits paired Settlement Ledger transfer events that surrender each party's prior accrual rights and grant the swapped accrual rights, with both transfers Trust-Block-bound to the swap-request record. Swap valuations reference the Compression Curve at a specified valuation_block_height to ensure deterministic-replay auditability; the Curve's parameters at that height are Trust-Block-anchored and immutable retroactively.

Derivative Ledger Integration

Each derivative is recorded on the Settlement Ledger (Family D) as a Trust-Block-bound claim against future settlement flows; settlement issuance to derivative holders is computed by the Settlement Controller (Family G) per the dual-hurdle / offset / swap parameterization.

Derivative records integrate with the Settlement Ledger (Family D) under the DERIVATIVE_CLAIM record type, with each derivative-class record referencing its issuance Trust Block, its encumbrance authority record, its current outstanding face value, its cumulative settlement-accrual ledger, and its next settlement-trigger predicate. The Settlement Controller (Family G) executes the settlement-issuance algorithm at each ledger height H as follows: for each DERIVATIVE_CLAIM record D, compute D.outstanding_face - D.cumulative_accrual (the remaining cap exposure); evaluate D.next_settlement_trigger_predicate against state at H; if the predicate fires, compute the issuance amount under D's dual-hurdle / offset / swap parameterization and emit a Settlement Ledger transfer event Trust-Block-bound to D. The Controller maintains a per-derivative state machine with states (ISSUED, ACCRUING, HURDLE_ACHIEVED, SHORTFALL_PENDING, SHORTFALL_RESOLVED, TERMINATED) and deterministic transition rules, 2.7. Cross-derivative ordering during shortfall events follows the BASELINE/EXPANDED/MAXIMUM Settlement Accrual Preference tier priority, with intra-tier ordering by issuance-block height.

Process Flow

Issuance: Quantum Privacy LLC issues Senior and Junior QPT Derivatives to qualified investors; the derivative terms are recorded on the Settlement Ledger.

Encumbrance: the Encumbrance Authority pins the ERT and AIIP allocations against the derivative; the Settlement Controller computes settlement allocation reserving the encumbered share.

Settlement flow: as settlement events occur, the Settlement Controller routes the encumbered share to derivative-holder accounts subject to the dual-hurdle and offset parameterization.

Hurdle achievement: when Senior derivative achieves both IRR and MOIC hurdles, the issuer exercises payoff option; settlement to Senior holders terminates and encumbrance releases.

Accrual Rights Swap execution: holders submit swap requests; the Settlement Controller verifies counter-party availability and executes the swap, updating Trust Block bindings.

Alternative Embodiments

- **Multi-tranche Senior embodiment**: Senior derivatives may be tranched by hurdle level (e.g., Senior-Conservative at 12%/2x; Senior-Standard at 18%/3.5x; Senior-Growth at 24%/5x), each with distinct Junior offset coverage.
- **Convertible Senior embodiment**: Senior derivatives may include a conversion right into a Liquidity Pool participation at a Premium-Multiple-Compression-indexed conversion ratio.
- **Cross-jurisdictional Senior embodiment**: Senior derivatives issued in multiple jurisdictions may be linked via cross-jurisdictional Trust Block bindings to maintain unified hurdle accounting.
- **Tax-jurisdictional Senior embodiment**: Senior tranches may be issued under jurisdiction-specific tax-treatment structures: a Cayman-domiciled Senior tranche structured for offshore institutional capital; a Singapore-VCC-domiciled Senior tranche for Asia-Pacific institutional capital; a Delaware-LLC-domiciled Senior tranche for US-onshore institutional capital. Each jurisdictional embodiment carries jurisdiction-specific compliance attestation sets, jurisdiction-specific reporting obligations, and jurisdiction-specific tax-elective optionality. The Trust-Block-bound encumbrance authority is unified across jurisdictions via cross-jurisdictional registry mapping (see Family N Existing-FI Tokenization Integration Adapter).
- **Stage-conditional Senior embodiment**: Senior tranches may be issued with stage-conditional terms that automatically adjust at Accelerator stage transitions (Family M): a Pioneer-stage Senior tranche may carry a Pioneer-stage IRR target of 30% and an automatic step-down to 22% upon Cascade-stage transition, reflecting the reduced execution risk at later stages. The step-down schedule is Trust-Block-anchored at issuance and is itself deterministically replayable; Senior holders may rely on the step-down terms without trust in the issuer's discretion.

- **Co-investment Senior embodiment**: Senior tranches may be structured for co-investment by multiple investor classes — e.g., an institutional anchor (pension fund) co-investing with a strategic enterprise and a sovereign wealth fund into the same Senior tranche. The Junior tranche providing first-loss coverage may itself be cross-class (each co-investor contributing a proportionate Junior share), or it may be a separately-funded Junior held by a third party (a credit-enhancement provider). Co-investment Senior tranches require enhanced compliance attestation: all co-investors must meet the most-restrictive jurisdictional requirement among them.
- **Subscription Senior embodiment**: Senior tranches may be issued under a subscription pattern: rather than a single up-front investment, the investor commits to a series of capital draws over a Subscription Window (typical 12–36 months). Each draw is Trust-Block-bound and increments the tranche's outstanding face; settlement accrual computations adjust pro-rata. This embodiment supports institutional investors with rolling capital-deployment mandates and reduces the up-front capital commitment required for participation.
- **Synthetic Junior embodiment**: Junior tranches may be synthesized from external credit-enhancement instruments (e.g., a bank-issued credit-default swap on the Senior tranche, a sovereign credit-enhancement facility, a parent-company corporate guarantee). The synthetic Junior is represented as a Trust-Block-bound external-attestation record paired with the Senior tranche; loss-absorption events trigger the synthetic Junior's external-instrument payout under cross-jurisdictional enforcement protocols. This embodiment supports cases where conventional Junior issuance is infeasible (e.g., regulatory constraints, capital-allocation constraints).
- **Compression-indexed Senior embodiment**: Senior tranche payout schedules may be directly indexed to the Premium Multiple Compression Curve (Family J): the cap-payment trajectory accelerates in early Curve positions (high Premium multipliers, faster settlement accrual) and decelerates in later Curve positions. This embodiment shifts payout timing without changing total cap amount, supporting investors whose mandates prefer earlier vs later cash-flow profiles. The Compression Curve indexation is Trust-Block-anchored at issuance and immutable thereafter.
- **Permissioned-Junior embodiment**: Junior tranches may be structured as permissioned instruments held only by qualifying parties (e.g., the issuer's affiliated entities, the Accelerator's general partner, the AIIP-managing entity). Permissioned Junior tranches carry restricted transferability — Trust-Block-bound transfer-approval predicates enforce that secondary-market sales are only permitted to similarly-qualifying parties. This embodiment supports cases where Junior loss-absorption obligations entail fiduciary responsibilities incompatible with broad-market holders.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Senior Derivative issuance:** (i) the issuing entity (Quantum Privacy LLC or successor) must hold a valid Trust Ledger entry authorizing Senior issuance under the applicable jurisdiction's compliance posture; (ii) the targeted Backing Pool must satisfy the operative coverage covenant (Portfolio-Multiple ≥ covenant threshold); (iii) the holder set must be jurisdictionally compliant per the operative compliance attestation set.
- **Postconditions:** (i) a DERIVATIVE_CLAIM record is emitted on the Settlement Ledger; (ii) the Encumbrance Authority record reflects the new encumbrance; (iii) holder DIDs are notified per their notification preferences.
- **State transitions:** Senior tranche transitions through ISSUED → ACCRUING → HURDLE_PROGRESS → (HURDLE_ACHIEVED, SHORTFALL_PENDING) → TERMINATED, with each transition Trust-Block-bound.
- **Error handling:** issuance failures (covenant breach, compliance mismatch) emit Audit Log records but no DERIVATIVE_CLAIM record; settlement-execution failures (e.g., Backing Pool insufficiency at a settlement event) trigger SHORTFALL_PENDING transitions with explicit shortfall amount records.
- **Performance characteristics:** issuance operations are O(1) per tranche; settlement-event execution is O(N) per N outstanding tranches; cross-tranche shortfall resolution is O(N log N) reflecting tier-priority ordering. Per, the protocol-grade performance bounds support tens of thousands of concurrent Senior

tranches with per-block-height settlement latency in the sub-second range under Trust-Block-anchored compute-resource provisioning.

Cross-Family Integration

Upstream Dependencies. Family L operates within QPN-enabled infrastructure per the §22.7 Wherein clause and consumes Settlement Ledger flows from Family G, Premium schedule outputs from Family J, Backing Pool composition from Family N, Accelerator Lock state from Family M, and Confidence Tier outputs from Family E for Senior Derivative holder eligibility predicates. The Encumbrance Authority records on the Authorization Ledger (Family D) are the canonical capital-formation governance anchors.

Downstream Consumers. Family L's Senior and Junior QPT Derivative records are consumed by Family G (settlement issuance computations route settled flows to derivative holders under dual-hurdle / offset / swap parameterization), Family N (Backing Pool participation-rights distribution references Senior Derivative encumbrance), and Family M (Stage-Differentiated Revert routes to Senior holders under stage-conditional Revert predicates). Per Pattern #7 (the architectural analysis), the Family L + Family M Encumbrance Authority + Accelerator Lock interaction is the architectural basis for pre-activation encumbrance compatibility with stage-differentiated revert. Per Pattern #8, the Family L + Family N Senior Derivative shortfall routing into the Backing Pool is essential for Senior holder coverage under shortfall scenarios.

Lateral Interactions. Family L's Compression-indexed Senior Derivative embodiment (Pass 2-α) interoperates with Family J's Compression Curve per Pattern #6 — cap-payment trajectories accelerate in early Curve positions and decelerate in later positions. Family L's Accrual Rights Swap primitive (Pattern #6) references the Compression Curve at a specified valuation_block_height for deterministic swap valuation. Family L's tax-jurisdictional Senior embodiment (Pass 2-α) interoperates with Family N's Existing-FI Tokenization Integration Adapter for cross-jurisdictional registry mapping.

Emergent System-Level Properties. Three emergent properties: (a) Pattern #6 Compression-Curve-indexed swap valuation produces deterministic, replayable swap pricing without trust in any single counter-party valuation; (b) Pattern #7 pre-activation encumbrance compatibility produces capital-formation feasibility — Senior Derivative coverage is computable at issuance time, before any settlement has occurred — by binding to Family M's Stage-Differentiated Revert predicates; (c) Pattern #8 Senior shortfall routing into Backing Pool produces Senior-holder protection without trust in any single Senior-issuer entity. Together these properties form the basis for the 950,000× coverage-ratio claim at Pioneer-stage Senior-Conservative tranches

Family M — Stage-Differentiated Revert + Accelerator Lock + MUM Tracking

Encompassing candidate disclosures: PVRF-5, PVRF-8, PVRF-9.

Field Summary

Stage-Differentiated Revert mechanism, Accelerator Lock (MUM + Governance DNA → non-dilutable AIIP), and Monetization Uplift Multiple tracking.

Family M is the Accelerator-discipline layer of the QPN capital-formation architecture. Per, 3.1–3.4 and, the Stage-Differentiated Revert primitive together with the Accelerator Lock and the Monetization Uplift Multiple (MUM) tracking provide the protocol-grade discipline for Private Accelerator capital deployment, performance measurement, and stage-conditional capital recall.

Problem Addressed

Tokenized network capital recovery and topology anchoring face two challenges: (a) uniform capital recovery rules across network maturity stages produce either over-extraction (early stages) or under-recovery (late stages), and (b) Accelerator allocation dilution risk where founder-stage allocations may be diluted by subsequent allocation events, defeating early-mover topology-anchoring incentives. Prior tokenized networks lack stage-aware capital recovery and lack non-dilutable founder-stage anchors.

Conventional accelerator-class capital deployment lacks protocol-grade performance discipline: once capital is committed to an accelerator, recall under underperformance is contractually fraught and often litigated; performance measurement is inconsistent across accelerators, defeating cross-accelerator capital allocation

discipline; stage transitions (incubation → growth → scale → maturity) are managed by accelerator-internal governance rather than by protocol-level primitives.

The Stage-Differentiated Revert primitive addresses recall: pre-activation, the encumbered ERT and AIIP allocations route deterministically to Senior Derivative holders subject to stage-specific revert conditions. The Accelerator Lock binds the capital commitment until specific Trust-Block-anchored performance milestones are achieved. The MUM provides a protocol-grade performance metric — Monetization Uplift Multiple, measured as monetization performance relative to capital deployment — that is independently auditable and cross-accelerator comparable.

Solution Overview

Stage-Differentiated Revert provides a capital recovery waterfall that varies by ecosystem stage — Pioneer / Cascade / Automated / Self-Funding — with each stage applying stage-specific Return-of-Capital priority and waterfall position. Accelerator Lock combines Monetization Uplift Multiple thresholds with Governance DNA conditions to render Accelerator Incentive & Investment Pool allocations non-dilutable: once an Accelerator's MUM crosses a configured threshold under qualifying Governance DNA, the AIIP allocation is permanently sealed against dilution by subsequent Accelerator-network governance actions. MUM is an on-ledger metric tracking each Accelerator's monetization performance relative to its capital contribution.

Each Accelerator capital commitment is recorded on the Settlement Ledger as a Trust-Block-bound commitment referencing the Accelerator's stage parameters, performance targets, and revert conditions. The Accelerator Lock prevents capital deployment changes until the locking conditions are released through Trust-Block-anchored milestone achievements.

MUM tracking operates as a continuous protocol-grade measurement: each Accelerator's monetization events are recorded on the Settlement Ledger, the cumulative monetization is computed against the capital-deployment baseline, and the resulting MUM is published as a Trust-Block-bound metric. Cross-accelerator allocations within the QP Meta Fund reference MUM as a deterministic input.

The Stage-Differentiated Revert mechanism is critical for Senior Derivative coverage: pre-activation, encumbered flows revert deterministically to Senior holders; mid-activation, revert conditions soften as MUM thresholds are achieved; post-activation, revert ceases and Accelerator capital becomes operationally bound. This stage-differentiation is itself parameterizable and Trust-Block-anchored, enabling investor-specific structuring of Senior Derivative tranches with stage-specific revert envelopes.

Components

Components detailed below: Accelerator Lock, Stage-Differentiated Revert Engine, Monetization Uplift Multiple (MUM) Tracker, Stage Transition Manager, Encumbrance Release Authority. Sources: (Governance Reserve mechanics), §3.1–§3.4 (Accelerator architecture and governance), and

Pioneer Stage Revert

Stage-specific Revert configuration for the network's Pioneer Stage. Pioneer participants enjoy preferential Return-of-Capital priority reflecting their early risk-taking; waterfall position places Pioneer recoveries ahead of Cascade and later-stage recoveries.

Pioneer Stage Revert is implemented as a Trust-Block-bound conditional-allocation record with on-ledger schema fields: revert_condition_predicate (symbolic predicate evaluated against the Settlement Ledger and the Catalyst Network performance metrics — e.g., "Accelerator A's MUM is < 0.5 at Stage-end-marker block"); revert_target_tranche (the Senior Derivative tranche to which reverted flows are directed under the Stage-Differentiated Revert mechanism); revert_share_bps (basis points of the would-have-been Accelerator allocation that revert under predicate firing); stage_end_block_height (the canonical block height marking Pioneer Stage end, beyond which Pioneer revert is no longer evaluable). Per, Pioneer Stage revert predicates are typically configured with revert_share_bps = 10000 (full Accelerator allocation reverts to Senior holders) for predicates that fire under severe underperformance (MUM < 0.5), and with revert_share_bps ∈ [3000, 7000] for predicates that fire under moderate underperformance (MUM ∈ [0.5, 1.0]). The predicate evaluation is deterministically replayable: any auditor with access to the Settlement Ledger and the operative predicate definition can compute the revert outcome at any ledger height without privileged access.

Cascade Stage Revert

Cascade Stage configuration emphasizing Return-on-Investment to Pioneer participants and partial RoC to Cascade-Stage participants; waterfall position reflects Cascade-Stage proportional contribution.

Cascade Stage Revert replaces the Pioneer-stage revert predicates with stage-transitioned predicates indexed to the Cascade Propagation stage. The schema fields parallel Pioneer Stage Revert but with two key parameter shifts: (i) revert_share_bps is typically lower (1000–4000 bps) reflecting partial-revert behavior as the Accelerator has now begun generating settlement; (ii) revert_condition_predicate references not only MUM but also Compression Curve position and Cascade Network density metrics — for example, "Accelerator A's MUM is < 0.7 AND Cascade Network density measured at Trust Block height H is below threshold T". This expanded predicate structure permits more nuanced encumbrance release as the broader ecosystem matures.

Automated Stage Revert

Automated Stage configuration shifting waterfall toward broad participant pool; RoC priorities normalize as network capital-formation enters self-sustaining mode.

Automated Stage Revert operates during the Automated Settlement Stage with predicates indexed to autonomous-agent activity volumes (Family I Governed Agent Loop), Resource Derivative Premium velocity (Family J), and Cascade Network maturity. The revert_share_bps is typically further reduced (300–1500 bps) reflecting that, at this stage, Accelerator capital deployment has substantively matured and full revert would unduly penalize underperformance vs minor performance shortfalls. The Automated Stage Revert engine integrates with the Family I governed-agent gate to verify that revert evaluation includes only authorized agent-originated settlement events (preventing forged-agent-flow inflation of MUM).

Self-Funding Stage Revert

Self-Funding Stage configuration optimizing for ongoing participant returns; capital recovery becomes ambient rather than priority-based.

Self-Funding Stage Revert applies during the Self-Funding Stage — the architectural condition where Accelerator capital deployment has fully graduated to internal-cashflow funding. At this stage, the revert_share_bps is typically 0–500 bps (residual coverage only); predicates require severe failure modes (MUM < 0.3 sustained over multiple blocks) to fire. The schema integrates a self_funding_compliance_check field that verifies the Accelerator's internal-cashflow metrics satisfy the Self-Funding Stage compliance criteria; if compliance is breached, the engine reverts the Accelerator's stage classification to Automated Settlement Stage and re-applies the corresponding Automated Stage Revert predicates.

Monetization Uplift Multiple (MUM)

Per-Accelerator on-ledger metric computed as ratio of cumulative settlement volume routed through the Accelerator to cumulative Accelerator capital deployment (financial + bootstrapping contributions). Updated each settlement cycle.

MUM is computed as a deterministic function over the Settlement Ledger: MUM(A, t1, t2) = sum(settlement_events bound to A in [t1, t2]) / capital_deployed_into_A_at_t1. The computation is content-addressed: the input set (settlement events) is identified by their Trust-Block-bound record references, the denominator is fixed by the Accelerator's capital-deployment record at t1, and the resulting MUM value is itself Trust-Block-anchored at publication time. Cross-Accelerator MUM comparisons follow strict normalization rules: the time-window must be identical, the settlement-class scope must be identical (e.g., all PNX-routed settlement vs. only on-ledger tokenized settlement), and the capital-deployment basis must use the same accounting primitive (e.g., committed-capital vs. drawn-capital). MUM publication frequency is configurable per Accelerator within a bounded range (typical: monthly to quarterly); real-time queries against the published MUM record set are supported via the PNX-Wide Registry (Family N).

Governance DNA

Per-Accelerator governance characteristic profile (drawn from the Quantum DNA architecture of Family O). Governance DNA records the Accelerator's governance configuration (Manager set, Permitted Audit Purposes scope, Cross-Verification policies, etc.).

Governance DNA, per the Family O Quantum DNA/Genome architecture, encodes the Accelerator's governance Genes within a Governance Strand inherited at Accelerator creation and modifiable only under Trust-Block-bound governance amendments. Key Governance Genes include: stage_transition_authority (which actors hold authority to authorize stage transitions); mum_threshold_set (the MUM thresholds defining stage transition triggers); revert_predicate_amendment_rules (rules under which revert predicates may be modified); participant_admission_rules (who may participate in the Accelerator under what Trust Criteria). Inheritance of Governance DNA in derived Accelerators (sub-Accelerators, sector-specific spinoffs) follows the Family O Lamarckian Inheritance Engine: acquired governance state is explicitly inheritable, with descendant Accelerators able to override specific Genes under Trust-Block-bound recombination policies.

Accelerator Lock Predicate

Predicate: ACCELERATOR_LOCK_ELIGIBLE = (MUM ≥ LOCK_THRESHOLD) AND (Governance_DNA_qualifies(Accelerator)). Evaluated periodically; first satisfaction triggers permanent non-dilutability of the Accelerator's AIIP.

The Accelerator Lock Predicate is implemented as a compound predicate evaluating: (i) MUM threshold satisfaction at the current stage; (ii) Cross-Accelerator capital-deployment integrity (no unauthorized inter-Accelerator capital migration); (iii) Trust-Block-anchored milestone achievements per the Accelerator's stage-transition rules. The Lock state is binary (LOCKED, UNLOCKED) per Accelerator-tranche pairing; UNLOCKED state permits capital migration, governance amendment, and tranche restructuring; LOCKED state forbids all such operations and routes any attempted operations to the Audit Log. Lock release is itself Trust-Block-bound and requires explicit predicate-firing — Lock cannot be released by issuer fiat. Per, the Lock Predicate's deterministic-replay property is critical for investor protection: any Senior Derivative holder can independently verify that the Accelerator's Lock state is correctly LOCKED or UNLOCKED at any ledger height, without trust in the Accelerator's manager or the issuer.

Process Flow

On each settlement cycle, MUM is computed per Accelerator and recorded on the Settlement Ledger.

The Stage-Differentiated Revert configuration applicable to the current network stage governs the settlement waterfall.

The Accelerator Lock Predicate is evaluated; if satisfied for the first time for a given Accelerator, the AIIP allocation is sealed via a non-dilutable Trust Block marker.

Subsequent governance actions affecting the locked Accelerator's AIIP are rejected at the Settlement Controller.

On network stage transition (governance-recorded event), the active Revert configuration switches; previously-recorded entries retain their stage-of-origin attribution.

Alternative Embodiments

- **Custom-staged embodiment**: networks may define alternative stage sequences (e.g., Bootstrap / Acceleration / Plateau / Sunset) with stage-specific Revert configurations.
- **Conditional Lock embodiment**: Accelerator Lock may be configured to require multiple consecutive MUM threshold satisfactions (e.g., 4 consecutive settlement cycles) before triggering.
- **Partial Lock embodiment**: rather than full AIIP non-dilutability, Accelerator Lock may lock a fraction of AIIP at the first MUM threshold and additional fractions at higher thresholds.
- **Multi-Accelerator Stage Revert embodiment**: Stage-Differentiated Revert may operate across linked Accelerators: a parent Accelerator's revert state may propagate to its child Accelerators (sub-Accelerators, sector-specific spinoffs) under a Linkage Revert Policy. The Policy specifies revert-propagation rules — full propagation (child reverts when parent reverts), partial propagation (child reverts only if child's own MUM is also below threshold), or asynchronous propagation (child revert deferred by N blocks after parent revert). All linkage records are Trust-Block-anchored.

- **Conditional MUM threshold embodiment**: MUM thresholds for stage transitions may be conditional on external observable metrics — e.g., "stage-transition MUM threshold is 1.2 if Cascade Network density at the transition block exceeds T_density, else 1.5". This embodiment supports adaptive stage progression where the broader ecosystem maturity influences individual Accelerator stage criteria. All conditional logic is Trust-Block-anchored and deterministically replayable.
- **Override-with-Audit embodiment**: Accelerator Lock release may be permitted under emergency conditions through an Override-with-Audit primitive: a designated authority (e.g., the Senior Derivative holders by Trust-Block-bound majority) may emit a Lock-Override record that releases the Lock subject to immediate Audit Log emission and a mandatory governance-review-block window (typical 30–90 blocks) during which the Override may be reversed by Senior holders. This embodiment supports legitimate operational flexibility while preserving Trust-Block-bound governance.
- **Premium-Curve-indexed Revert embodiment**: Revert predicates may be indexed to the Premium Multiple Compression Curve position: revert shares may be computed as revert_share_bps = base_revert_share × compression_factor(current_block) where compression_factor diminishes over Curve progression. This embodiment ties revert magnitude to ecosystem maturity, with later-stage reverts (in compressed-Premium regimes) absorbing smaller revert shares.
- **Cross-Stage MUM averaging embodiment**: Stage-transition MUM evaluation may use cross-stage averaged MUM rather than current-block MUM: the transition MUM is MUM_avg = weighted_average(MUM at last 10 blocks, weights per a Trust-Block-anchored weight schedule). This embodiment smooths transient MUM fluctuations and prevents single-block MUM spikes from triggering improper stage transitions.
- **Pooled-Accelerator MUM embodiment**: MUM may be computed at the pooled-Accelerator level: multiple Accelerators may be grouped into a Pool with shared MUM computation, supporting cases where individual Accelerator MUM is too noisy or volume-limited for reliable stage-transition assessment. Pool composition is Trust-Block-anchored; Pool MUM is the volume-weighted average of constituent Accelerator MUMs.
- **Forfeiture-on-Reset embodiment**: If an Accelerator's stage classification is reverted (e.g., SELF_FUNDING → AUTOMATED under compliance breach), a Forfeiture-on-Reset primitive may forfeit a portion of the Accelerator's accumulated MUM credit — forfeiture_bps = configurable basis points of accumulated MUM — disincentivizing speculative stage advancement. The forfeited MUM is redirected to the Backing Pool under the Family N participation-rights distribution algorithm.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for stage-transition operations:** (i) the Accelerator must hold a valid current-stage record; (ii) the operative stage-transition predicate must evaluate to true at the current ledger height; (iii) the Accelerator Lock must be UNLOCKED.
- **Postconditions:** (i) the Accelerator's stage record is updated; (ii) active Revert predicates are switched to the new-stage Revert predicate set; (iii) MUM computation parameters are re-anchored to the new stage.
- **State transitions:** Accelerator stage transitions follow PIONEER → CASCADE → AUTOMATED → SELF_FUNDING, with explicit reverse transitions permitted under compliance-breach conditions (SELF_FUNDING → AUTOMATED reversion described in the Self-Funding Stage Revert technical depth above).
- **Error handling:** stage-transition predicate failures emit Audit Log records and leave the Accelerator in its current stage; revert-predicate evaluation errors (e.g., insufficient settlement-event data) trigger SHORTFALL_PENDING states deferring revert resolution to subsequent blocks.
- **Performance characteristics:** stage-transition evaluation is O(1) per Accelerator per block; revert-predicate evaluation is O(N) per N active predicates; MUM computation is O(M) per M settlement events in the operative window.

Cross-Family Integration

Upstream Dependencies. Family M operates within QPN-enabled infrastructure per the §22.7 Wherein clause and consumes Family G Settlement Ledger flows for MUM computation, Family L Senior Derivative records for Revert predicate evaluation, Family N Backing Pool state for Accelerator participation tracking, Family O Governance DNA from the Accelerator's Genome, and Family I governed-agent settlement-event records for Automated Stage Revert evaluation.

Downstream Consumers. Family M's MUM publication records are consumed by Family L (Senior Derivative stage-conditional terms reference MUM thresholds), Family N (Liquidity Provider participation rates may reference MUM as a quality metric), Family K (Behavioral Activation Level transitions for Accelerator-affiliated participants may reference MUM), and the broader QPIIN / QP Meta Fund deployment logic consumes MUM as a deterministic input for cross-Accelerator capital allocation. Stage-Differentiated Revert records consumed by Family L Senior Derivative holders.

Lateral Interactions. Family M's Pioneer / Cascade / Automated / Self-Funding stage taxonomy interoperates with the broader Catalyst Network stage taxonomy. Stage transitions are Trust-Block-bound and may require Accelerator Lock release evaluation. Per Pattern #9 (the architectural analysis), the Family N + Family M liquidity-architecture-driven Accelerator performance metrics interaction is essential: Family N Backing Pool flow data feeds Family M MUM computation; Family M MUM publication feeds Family N Liquidity Provider participation decisions.

Emergent System-Level Properties. Two emergent properties: (a) Pattern #7 Encumbrance + Lock compatibility (detailed in Family L Cross-Family Integration) — Family M's contribution is the stage-transition discipline that releases the Lock under MUM-threshold satisfaction; (b) Pattern #9 liquidity-driven performance metrics produce protocol-grade Accelerator quality signals that propagate across the Backing Pool and the broader QPIIN deployment logic without requiring centralized Accelerator evaluation.

Family N — Liquidity Architecture

Encompassing candidate disclosures: PVRF-4, PVRF-6, PVRF-7, PVRF-11, PVRF-12.

Field Summary

Privacy Network Exchange liquidity architecture: distinct Exchange Provider and Liquidity Provider roles, Backing Pool Multiples, Tranche Priority Pool absorption, QPT-collateralized lending, existing-FI balance-sheet tokenization integration.

Family N is the liquidity layer of the QPN. Per, 2.6, and, the Privacy Network Exchange (PNX) Liquidity Architecture provides multi-tier liquidity pool primitives, multi-tier pricing under Liquidity-as-a-Service framing, Backing Pool mechanics integrating with Senior Derivative shortfall routing, and the PNX-wide registry of liquidity-pool participations that is publishable as a protocol-grade transparency surface.

Problem Addressed

Tokenized network liquidity faces several unresolved problems: (a) settlement counterparties and capital providers are conflated, producing principal-agent conflicts where settlement counterparties have incentive to favor their own positions; (b) liquidity pool capacity parameterization lacks structural controls and tends toward either over-extension (collapse risk) or under-extension (illiquidity); (c) tranche absorption logic is ad-hoc and varies across implementations; (d) tokenized collateral lending lacks Trust Block-bound release conditions; (e) integration with existing financial institutions' balance sheets is treated as out-of-scope, foreclosing significant institutional liquidity pathways.

Conventional tokenized-network liquidity provision suffers from a structural defect: liquidity pools and protocol native-token allocations are typically managed by the same entity under conflicting incentives. The liquidity-provider role rewards minimizing transaction costs and maximizing pool depth; the native-token-issuer role rewards maximizing protocol revenue. Conflating the two produces either under-provisioned liquidity (degrading user experience) or overissuance (degrading native-token value).

The PNX Liquidity Architecture addresses this with structural separation: liquidity pools operate as Trust-Block-bound participations against a Backing Pool whose composition is governed by protocol-level allocations, with

multi-tier pricing converting the pricing function from a single-parameter curve into a parameterizable governance instrument. Per, the Pioneer-Stage Launch Capital and Liquidity-as-a-Service framework provides protocol-grade liquidity provisioning during the price-discovery window without conflating the provisioning role with the native-token-issuer role.

Solution Overview

The Liquidity Architecture structurally separates Exchange Provider (settlement counterparty) and Liquidity Provider (capital provider) roles, each with distinct compensation and Trust Block authorization scope. Backing Pool Multiples Architecture parameterizes Backing Pool capacity along three dimensions: portfolio-multiple, annual-flow-multiple, and cap-amount, each governed by Trust-Block-bound governance. Tranche Priority Pool Absorption applies a two-stage waterfall: Return-of-Capital absorbed first by Priority Pool, then Return-on-Investment distributed per participation rights. QPT-Collateralized Lending uses Quantum Privacy Tokens as collateral with Trust Block-bound release conditions and PoT-enforced margining. Existing-FI Balance-Sheet Tokenization Integration permits financial institutions to retain custody keys while QPN-issued tokens represent on-ledger claims on FI-held assets, with Trust-Block-bound dual-control.

Each liquidity-pool participation is recorded on the Settlement Ledger as a Trust-Block-bound claim. The Backing Pool aggregates the underlying Exchange Root + Accelerator Token + AIIP allocations against which Senior and Junior Derivative claims are computed; Senior Derivative shortfalls route into the Backing Pool under stage-conditional protocols.

Multi-tier pricing operates across three pricing strata indexed to liquidity depth, time-of-day, and Privacy-Domain class. Pricing transitions are themselves Trust-Block-anchored, enabling deterministic-replay auditability of every price quote. PNX-wide registry publishability is a protocol-grade transparency property: the aggregate liquidity-pool participation surface is publishable as a Trust-Block-anchored snapshot at any specified ledger height, providing third-party auditors with a deterministic view of total liquidity provisioning. This is materially stronger than conventional exchange transparency, which provides only point-in-time order-book snapshots without cryptographic integrity guarantees.

Components

Components detailed below: PNX Liquidity Pool, Backing Pool Manager, Multi-Tier Pricing Engine, Shortfall Routing Engine, PNX-Wide Registry, Registry Publishability Manager. Sources: (Early Liquidity Architecture and Liquidity-as-a-Service), §2.6 (QPT Derivative coverage from Backing Pool), and

Exchange Provider Role

Settlement counterparty role. Holds Trust-Block-bound authorization to participate in settlement matching but does not provide working capital. Compensated via per-settlement fee from the Settlement Controller's waterfall.

Exchange Provider Role is a Trust-Block-bound role-record with on-ledger schema fields: provider_did, accreditation_tier (enum: PIONEER, CASCADE, AUTOMATED, SELF_FUNDING per stage taxonomy), permitted_settlement_classes (bitmap over settlement-class taxonomy: SPOT, FORWARD, OPTION, DERIVATIVE_BUNDLE, etc.), fee_schedule_uuid (reference to a Trust-Block-anchored fee schedule), compliance_attestation_set (jurisdiction-specific compliance attestations: e.g., MiFID-II for EU jurisdictions, FINRA for US, MAS for Singapore). The Exchange Provider's PNX-side execution surface is constrained by permitted_settlement_classes; attempts to execute settlement of a class not in the bitmap are rejected at PNX ingress with Trust-Block-bound rejection records. Per, accreditation-tier transitions are governed by the Liquidity-as-a-Service compliance criteria: a Pioneer-tier provider may graduate to Cascade tier upon demonstrating MUM-thresholded performance across a configurable observation window (typical: 6–12 months).

Liquidity Provider Role

Capital provider role. Holds Trust-Block-bound authorization to deposit working capital into Liquidity Pools and to claim a share of liquidity-provision fees. Does not act as settlement counterparty.

Liquidity Provider Role provides protocol-grade liquidity into the PNX Backing Pool subject to the following on-ledger schema: provider_did, deposit_amount, deposit_currency_class (enum: FIAT_USD, FIAT_EUR,..., QPT,

QPT_DERIVATIVE, EXTERNAL_TOKEN), lockup_period_blocks (integer block-count for lockup), withdrawal_priority_tier (enum: STANDARD, EXPANDED, MAXIMUM with same priority semantics as Family L Settlement Accrual Preference tiers), participation_rights_share_bps (computed at deposit time, typically proportional to deposit amount weighted by lockup period and currency class). Per, Pioneer-stage Liquidity-as-a-Service Provider tier confers durable structural advantages similar to QPIIN Tier 1 status: Pioneer LP allocations confer priority withdrawal rights and Compression-Curve-indexed Premium uplift on associated Resource Derivative flows.

Backing Pool Portfolio-Multiple

Parameter governing Backing Pool capacity as a multiple of underlying portfolio value held by participants. Trust-Block-bound; updates require governance approval.

The Backing Pool Portfolio-Multiple is computed as portfolio_multiple = backing_pool_aggregate_npv / aggregate_cap_amount where backing_pool_aggregate_npv is the sum across the Backing Pool of NPVs of held instruments (QPTs, QPT Derivatives, externally-held collateral) under a specified valuation methodology, and aggregate_cap_amount is the sum of all outstanding Senior Derivative cap amounts. Per, the Portfolio-Multiple at Pioneer-stage Senior-Conservative tranches is approximately 950,000× for $1B at 3× cap multiple, reflecting the long-horizon (75-year) backing pool NPV vs the bounded cap amount. The Portfolio-Multiple computation is deterministically replayable from the Settlement Ledger and the operative valuation methodology snapshot; replay-grade auditability is a protocol-level requirement, not an operational convenience.

Backing Pool Annual-Flow-Multiple

Parameter governing Backing Pool capacity as a multiple of annual settlement flow. Trust-Block-bound.

The Annual-Flow-Multiple is computed as annual_flow_multiple = annualized_settlement_flow_into_pool / aggregate_cap_payment_obligation. Where the Portfolio-Multiple measures the static balance-sheet coverage, the Annual-Flow-Multiple measures the dynamic coverage rate — how quickly the cap-payment obligations are being satisfied by ongoing settlement flow. The two metrics together define the coverage envelope: a high Portfolio-Multiple with a low Annual-Flow-Multiple indicates an extended cap-payment horizon (the cap will be paid eventually, slowly); a low Portfolio-Multiple with a high Annual-Flow-Multiple indicates a near-term cap-payment horizon. Per, the Dual-Hurdle structure's IRR target binds under the former regime, MOIC backstop under the latter.

Backing Pool Cap-Amount

Absolute cap on Backing Pool capacity, providing structural ceiling regardless of multipliers.

The Backing Pool Cap-Amount is the aggregate sum of outstanding Senior Derivative cap obligations against the Backing Pool, computed as cap_amount = sum across senior tranches T of (T.tranche_face * T.payout_return_cap_multiple). The Cap-Amount is the binding ceiling on Senior holders' aggregate nominal claim against the Backing Pool, and it is the canonical denominator for the Portfolio-Multiple computation. Per, Cap-Amount additions (new Senior tranche issuance) are subject to compliance with Backing-Pool coverage covenants: a new tranche may not be issued if the resulting Portfolio-Multiple would fall below a Trust-Block-anchored covenant threshold.

Priority Pool

Senior pool that absorbs Return-of-Capital before any Return-on-Investment distribution. Operates as the first stage of the two-stage waterfall.

The Priority Pool is the subset of Backing Pool assets specifically reserved against highest-tier Senior Derivative obligations (BASELINE-tier Settlement Accrual Preference per Family L). Priority Pool assets are: (i) subject to first-claim priority under shortfall scenarios; (ii) restricted from general-purpose Backing Pool deployment; (iii) reported separately in the PNX-Wide Registry. Per, the Priority Pool composition is governed by Trust-Block-anchored selection rules: only QPTs and QPT Derivatives whose underlying source flows are deemed protocol-grade (Settlement Ledger entries with N-of-M Cross-Verification Bundle confirmations) qualify for Priority Pool inclusion.

Participation Rights Distribution

Second-stage waterfall: after Priority Pool RoC absorption, residual flows are distributed per participation rights of Liquidity Providers.

Participation Rights Distribution is the protocol-grade mechanism by which Backing Pool participation rights are allocated to Liquidity Providers, Senior Derivative holders, and Junior Derivative holders. The distribution algorithm operates as: at each settlement-event block, the Settlement Controller (Family G) computes the participation-rights deltas for each role-record and emits Trust-Block-bound transfer events on the Settlement Ledger. Per, the distribution algorithm is deterministically replayable: any auditor can recompute participation-rights state at any ledger height. Distribution priorities follow the Settlement Accrual Preference tier hierarchy (BASELINE > EXPANDED > MAXIMUM in priority of claim, inverse priority of yield), with intra-tier ordering by deposit-block height.

QPT-Collateralized Lending Module

Lending architecture using Quantum Privacy Tokens as collateral. Trust-Block-bound collateral release: on loan repayment, the Trust Block governing the collateral is amended to release the encumbrance. PoT-enforced margining: if collateral value falls below margin threshold, a PoT-triggered margin call is issued.

The QPT-Collateralized Lending Module extends the Backing Pool with overcollateralized lending functionality: a participant may pledge QPTs (or QPT Derivatives) as collateral against fiat or token-denominated loans, with the collateral held in a Lending-Sub-Pool subject to Trust-Block-bound loan-to-value (LTV) covenants. The schema includes: pledged_collateral_record_uuid, loan_principal, loan_currency_class, ltv_ratio_bps (typical Pioneer-stage range 2000–4000 bps for QPT collateral, 4000–6000 bps for QPT Derivative collateral), liquidation_trigger_predicate, liquidation_priority_tier. Per, liquidation events route the collateral back into the Backing Pool under a specified liquidation_routing_policy that preserves Priority Pool integrity.

Existing-FI Tokenization Integration Adapter

Adapter permitting financial institutions to retain custody keys on underlying assets while QPN-issued tokens represent on-ledger claims. Dual-control: any redemption of QPN tokens against FI-held assets requires both FI authorization and QPN Trust Block release.

The Existing-FI Tokenization Integration Adapter provides protocol-grade interoperability with existing financial-institution tokenization platforms (e.g., Hedera Token Service, Stellar Asset Issuance, custody-platform-bound tokenized securities). The Adapter's schema includes: external_fi_identifier, external_token_class, mapping_policy_uuid (Trust-Block-anchored mapping rules), compliance_attestation_set (jurisdiction-specific attestations linking the external FI's compliance posture to the PNX's compliance requirements), ingress_attestation and egress_attestation (cross-chain attestations binding the external token to its PNX-side representation). Per and the QPN Hedera Catalyst Partnership Overview, the Adapter design supports both ingress (external tokens entering the PNX as PNX-side tokenized representations) and egress (PNX-side tokens reflecting onto external platforms). All ingress/egress operations are deterministically replayable; the Trust-Block lineage chain is preserved across substrate boundaries.

Process Flow

On a settlement event, the Settlement Controller (Family G) matches Exchange Provider counterparties; matched counterparties earn Exchange Provider fees.

Liquidity Pool capacity is consulted: Backing Pool Multiples are evaluated against current portfolio value, annual flow, and cap; effective capacity is the most restrictive of the three.

If liquidity is consumed beyond Backing Pool capacity, the Settlement Controller defers or rejects the settlement.

On liquidity-provider returns, the two-stage waterfall applies: Priority Pool absorbs RoC, then Participation Rights Distribution distributes RoI.

For QPT-collateralized lending: collateral is deposited under a Trust Block lock; loan is issued; on repayment, lock is released; on default-triggering PoT signal, collateral is liquidated.

For existing-FI integration: an FI deposits an asset and the Integration Adapter mints corresponding QPN tokens; on redemption, FI authorization and Trust Block release are jointly required.

Alternative Embodiments

- **Tiered Liquidity Provider embodiment**: Liquidity Providers may be tiered by deposit size and lock duration, with tier-specific Premium weights.
- **Cross-asset Backing Pool embodiment**: Backing Pools may hold heterogeneous backing assets (QPT, stable tokens, existing-FI tokenized assets) with asset-class-specific multiples.
- **Insurance-Pool embodiment**: an additional Insurance Pool may absorb Priority Pool shortfalls, with Insurance Pool capacity parameterized by its own multiples.
- **Multi-Substrate Backing Pool embodiment**: The Backing Pool may span multiple ledger substrates: Hedera-Token-Service-issued tokens, Stellar-issued assets, Ethereum-issued tokens, and PNX-native tokens may all coexist within the Backing Pool subject to cross-substrate valuation rules anchored in the Existing-FI Tokenization Integration Adapter. Each substrate-class is reported separately in the PNX-Wide Registry and is subject to substrate-specific compliance attestation sets.
- **Time-Tiered Liquidity Provider embodiment**: Liquidity Providers may deposit under time-tiered classifications: 30-day tier with lower yield and faster withdrawal; 1-year tier with intermediate yield and intermediate withdrawal; 5-year tier with highest yield and longest lockup. Tier transitions during a deposit's lifetime are forbidden absent Trust-Block-bound amendment; tier-specific Compression-Curve-indexed Premium uplift applies.
- **Pooled-Insurance embodiment**: Beyond the Insurance Pool described in the existing alternative embodiment, a Pooled-Insurance primitive may aggregate insurance coverage from multiple external insurance providers — sovereign insurance funds, commercial insurers, mutual insurance cooperatives — into a unified PNX-side Pooled-Insurance record. Pooled-Insurance coverage applies to specific shortfall classes (e.g., Operational Shortfall, Regulatory Shortfall, Catastrophic Shortfall) with class-specific coverage amounts.
- **Cross-Jurisdiction Backing Pool embodiment**: Backing Pool composition may be partitioned by jurisdiction: an EU-domiciled Backing Pool subset under MiFID-II compliance attestation; a US-domiciled subset under US securities-regulator attestation; an Asia-Pacific subset under MAS / JFSA / FSC attestation. Cross-jurisdictional reallocation requires explicit Trust-Block-bound compliance attestation chains.
- **Smart-Routing PNX embodiment**: PNX settlement routing may incorporate smart-routing logic: a settlement event may be deterministically routed to the Backing Pool subset with the most-favorable Premium-Curve position, the most-available Priority Pool capacity, and the most-compliant jurisdictional attestation chain. Smart-routing decisions are Trust-Block-anchored and replayable.
- **DeFi-Bridge Liquidity embodiment**: The Backing Pool may interface with permissionless DeFi liquidity pools (Uniswap, Curve, Balancer) via a DeFi-Bridge Adapter that wraps DeFi-side liquidity as PNX-side participation-rights records. Bridge-Adapter operations are Trust-Block-bound; DeFi-side liquidity is non-Priority-Pool by default but may be promoted to Priority Pool under explicit Trust-Block-anchored promotion attestations.
- **Sovereign-Liquidity embodiment**: Sovereign actors (national central banks, sovereign wealth funds, multilateral institutions) may participate as specialized Liquidity Providers under Sovereign-Liquidity Records: enhanced Priority-Pool tier, regulatory-coordination protocols, sovereign-specific compliance attestation chains. Sovereign-Liquidity records may carry settlement-priority privileges (e.g., national-security-event routing priority) under explicit Trust-Block-anchored sovereign attestation.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Liquidity Provider deposit operations:** (i) the Provider's compliance-attestation set must include all jurisdictional attestations required for the target Backing Pool; (ii) the deposit-currency-

class must be in the Backing Pool's accepted-class set; (iii) the proposed lockup-period must satisfy the operative minimum-lockup covenant.

- **Postconditions:** (i) the Backing Pool's aggregate state is updated; (ii) the Provider's participation-rights record reflects the deposit; (iii) Priority Pool / Standard Pool allocations are rebalanced per the operative rebalancing policy.
- **State transitions:** Provider state transitions through PENDING_COMPLIANCE → DEPOSITED → LOCKED → WITHDRAWABLE → WITHDRAWN, with lockup-period satisfaction triggering the LOCKED → WITHDRAWABLE transition.
- **Error handling:** compliance-attestation failures emit Audit Log records without state change; deposit failures (e.g., external-substrate transfer failures via the Existing-FI Tokenization Integration Adapter) trigger DEPOSIT_FAILED states with explicit failure-reason records.
- **Performance characteristics:** deposit/withdrawal operations are O(log N) per N concurrent providers reflecting Priority-Pool / Standard-Pool ordering; Backing-Pool-Multiple recomputation is O(P) per P held instruments; participation-rights distribution is O(P × R) per P participants and R role-records, optimizable to O((P + R) log(P + R)) under Trust-Block-anchored incremental computation.

Cross-Family Integration

Upstream Dependencies. Family N operates within QPN-enabled infrastructure per the §22.7 Wherein clause and consumes Family L Senior and Junior Derivative records for Backing Pool composition computation, Family G Settlement Ledger flows for PNX settlement routing, Family M MUM publications for Accelerator-class Liquidity Provider tier evaluation, Family O Quantum DNA for Genome-bound Liquidity Provider Role records, and Family K Multi-Factor Identity Binding for Liquidity Provider authentication.

Downstream Consumers. Family N's Backing Pool state and PNX-Wide Registry records are consumed by every Family with capital-market-facing operations: Family L (coverage-ratio computations reference Backing Pool aggregate state), Family M (cross-Accelerator capital-deployment integrity checks reference Backing Pool reconciliation), and the broader QPIIN / Meta Fund logic. The PNX-Wide Registry publishability (per the Phase A reframing recommendation for Family N Claim 384) supports third-party auditor and regulator queries against aggregate Backing Pool state.

Lateral Interactions. Family N's Exchange Provider and Liquidity Provider Roles interoperate with Family B Institutional Mode for sponsor-bound Provider relationships. Family N's QPT-Collateralized Lending Module interoperates with Family L Senior Derivative collateralization. Family N's Existing-FI Tokenization Integration Adapter interoperates with Family L's tax-jurisdictional Senior embodiment (Pass 2-α) for cross-jurisdictional registry mapping. Per Patterns #8 and #9 (the architectural analysis), Family N's Backing Pool integrates with Family L Senior Derivative shortfall routing and Family M MUM tracking.

Emergent System-Level Properties. Two emergent properties: (a) Pattern #8 Senior shortfall routing produces Senior-holder protection at the protocol level; (b) Pattern #9 liquidity-driven Accelerator metrics produce a capital-market-grade feedback loop in which liquidity flows are governed by deterministic MUM signals rather than discretionary Accelerator-manager assessments. Together these properties produce a self-disciplining capital market in which Senior Derivative holders, Liquidity Providers, and Accelerator participants share a common Trust-Block-anchored view of system state.

Family O — Quantum DNA/Genome Inheritance Architecture

Encompassing candidate disclosures: UE-1, UE-2, UE-3, UE-4, UE-7, UE-8, UE-10.

Field Summary

Quantum DNA/Gene/Genome inheritance architecture: three-level hierarchy, Lamarckian inheritance, multi-Genome architecture, governance superposition, mitochondrial DNA analog, Premium inheritance, multi-layered aggregation.

Family O is the inheritance-architecture layer of the QPN. Per Universal Exchange, Ownership & AI (UE) §2.7 and and §22.1 (Quantum Genome vocabulary cluster), the Quantum DNA / Quantum Gene / Quantum Genome

architecture provides cryptographically-anchored inheritance of governance, attribution, and economic state across participant, resource, and Personal Privacy Network (PPN) generations. The Quantum Genome is the full governance identity comprising multiple Quantum DNA strands; Quantum Genes are the individually-inheritable units within each strand.

Problem Addressed

Trust-Block-based systems require a structured representation of governance characteristics that can (a) be inherited from upstream resources to downstream derivatives, (b) be propagated across participant creations (new Personal Privacy Networks, Resource Derivatives), (c) support multiple coexisting governance contexts per participant, (d) record acquired governance characteristics for inheritance, and (e) track operational infrastructure provenance separately from primary governance lineage. Prior systems either treat governance as a flat per-resource property (no inheritance) or as a rigid hierarchical namespace (no multi-context support).

Conventional digital inheritance architectures address only narrow slices of the problem: estate planning systems address account transfer but not attribution-graph continuity; corporate succession systems address organizational identity but not protocol-level state inheritance; biological-analog inheritance metaphors (e.g., "DNA of a system") are figurative rather than protocol-grade. None provide a cryptographically-anchored, deterministically-computable inheritance primitive that survives the originating participant or resource and remains auditable across generations.

The Quantum DNA/Genome architecture addresses this with a protocol-grade inheritance primitive: each participant and each resource possesses a Quantum Genome comprising multiple Quantum DNA strands; each strand comprises Quantum Genes corresponding to specific governance, attribution, and economic state elements; inheritance operates as deterministic Genome recombination from one or more parent Genomes under a configurable Recombination Policy. The full inheritance graph is Trust-Block-anchored and deterministically replayable. Lamarckian inheritance — direct inheritance of acquired governance state — is a protocol primitive rather than a contested biological-evolution claim, because the protocol explicitly defines which state elements are inheritable and under which Recombination Policy.

Solution Overview

The Quantum DNA Architecture provides a three-level inheritance hierarchy: Quantum Genes are atomic governance traits (e.g., 'requires-HIPAA-attestation', 'allows-secondary-derivative', 'sealed-after-365-days'); Quantum DNA is a composite governance profile composed of multiple Genes; Quantum Genome is the full participant-or-resource governance identity composed of multiple DNA strands. Lamarckian Inheritance propagates acquired governance characteristics (Premium accruals, attestation events, reputation events) into descendant entities' Genome composition. Multi-Genome Architecture permits a participant to hold multiple Genomes simultaneously (personal/professional/anonymous) with strict state isolation. Governance Superposition permits DNA expression across multiple overlapping governance contexts simultaneously, with context-specific projection of the relevant Genes. Mitochondrial DNA Analog tracks operational infrastructure provenance separately. Premium Inheritance propagates Premium accruals through Trust Block chains. Multi-Layered Aggregation Principle permits governance-preserving composition of contributions across resource → derivative → solution → exchange layers.

The Quantum Genome is structured as a directed acyclic graph of Quantum DNA strands, each strand comprising a set of Quantum Genes. Strands are typed (Governance Strand, Attribution Strand, Economic Strand, Reputation Strand, Resource-Specific Strand, etc.); Genes within each strand are independently parameterizable.

Inheritance Engine integration is the protocol-grade mechanism that composes descendant Genomes from parent Genomes. Two-parent reproduction (Family P) is the canonical case; N-parent polygenomic recombination is the generalized case. Recombination Policies are Trust-Block-anchored and themselves auditable: any descendant Genome can be deterministically derived from its parents and the operative Recombination Policy, providing replay-grade auditability of inheritance across multiple generations.

Mitochondrial-analog inheritance is the architectural primitive for asymmetric inheritance: certain Genome elements (typically Operational Lineage state) follow a single-parent inheritance path analogous to mitochondrial DNA, while the bulk of the Genome follows multi-parent recombination. This asymmetry is

essential for preserving Operational Lineage continuity across descendant generations while still permitting polygenomic governance recombination. Resource Derivative Premium inheritance is computed under the Compression Curve (Family J), tying inheritance economics into the Catalyst Network's Premium discipline.

Components

Components detailed below: Quantum Genome, Quantum DNA Strand, Quantum Gene, Inheritance Engine, Recombination Policy Manager, Mitochondrial-Analog Inheritance Manager, Operational Lineage Tracker, Resource Derivative Premium Inheritance Engine. Sources: (Universal Ownership and Inheritance Architecture), §§2.8–2.11 (Quantum Privacy Domain inheritance) (canonical Quantum Genome vocabulary cluster), and §22.1 (Universal Ownership architectural primitives).

Quantum Gene

Atomic governance trait. Each Gene has a defined trait type, valid value set, and Trust Block schema. Examples: 'jurisdiction:US-HIPAA', 'consent-decay-days:365', 'cross-border-permitted:false', 'minor-data-flagged:true'.

A Quantum Gene is implemented as a Trust-Block-bound addressable state element with the schema: gene_uuid (content-addressed identifier), gene_type (enum from a Trust-Block-anchored Gene taxonomy: GOVERNANCE_GENE, ATTRIBUTION_GENE, ECONOMIC_GENE, REPUTATION_GENE, RESOURCE_SPECIFIC_GENE,...), gene_value (typed payload), parent_gene_uuid_set (optional, for inherited Genes — ordered set of parent Gene UUIDs from which this Gene was recombined), recombination_policy_uuid (reference to the Recombination Policy that produced this Gene if inherited), gene_creation_block_height (Settlement Ledger block height at which this Gene was first recorded). Per and, the Gene-level granularity is essential for inheritance computation: descendant Genome composition operates Gene-by-Gene under the operative Recombination Policy, not at coarser Strand or Genome granularity. Gene immutability post-creation is a protocol invariant; Gene modification requires emitting a new Gene with a gene_supersedes_uuid reference.

Quantum DNA

Composite governance profile composed of multiple Genes. DNA is a Trust-Block-bound set of Gene values plus inheritance metadata.

A Quantum DNA Strand is an ordered set of Quantum Genes grouped under a common Strand Type, implemented as: strand_uuid, strand_type (GOVERNANCE_STRAND, ATTRIBUTION_STRAND, ECONOMIC_STRAND, REPUTATION_STRAND, RESOURCE_SPECIFIC_STRAND, OPERATIONAL_LINEAGE_STRAND), gene_uuid_ordered_set (ordered list of Gene UUIDs comprising the Strand), strand_metadata (Trust-Block-anchored metadata including version, schema_version, inheritance_policy_default). Strand-level operations include: Strand recombination (Gene-by-Gene composition from parent Strands under the operative Recombination Policy), Strand override (replacing a Strand wholesale with a new Strand under explicit Trust-Block-bound authority), Strand freeze (marking a Strand as inheritance-only, no further direct modification). Per and, the canonical Strand Types listed above are the protocol-grade taxonomy; deployments may extend the taxonomy via Trust-Block-anchored type registrations but must not collapse Strand boundaries.

Quantum Genome

Full participant-or-resource governance identity composed of multiple DNA strands. Strands may represent different facets (personal-DNA, professional-DNA, anonymous-DNA), and the Genome maintains strict state isolation across strands.

A Quantum Genome is the directed acyclic graph of Quantum DNA Strands comprising a participant's or resource's full governance identity. The schema is: genome_uuid, entity_did (the participant or resource bound to this Genome), strand_uuid_set (set of constituent Strand UUIDs), genome_metadata (Trust-Block-anchored metadata including genome_version, schema_compatibility_set, parent_genome_uuid_set if this is a descendant Genome). Genome operations include: Genome composition (constructing a descendant Genome from parent Genomes via Strand-by-Strand recombination), Genome serialization (emitting a content-addressed canonical Genome representation for content-addressed referencing), Genome diff (computing the Strand-level and Gene-level diff between two Genome versions). Per, the Genome is the canonical inheritance unit at the entity level; cross-entity inheritance (Genome composition across multiple entity Genomes) follows the polygenomic recombination patterns disclosed in Family P.

Lamarckian Inheritance Engine

Component within an Inheritance-scoped QPC that propagates acquired governance characteristics into descendant entity Genomes. On a recorded event (Premium accrual, attestation event, reputation event), the Engine updates the participant's Genome to record the acquired characteristic, which is then inherited by entities subsequently created from that participant.

The Lamarckian Inheritance Engine is the protocol component that performs Gene-level inheritance from parent Genomes to descendant Genomes under the operative Recombination Policy. The algorithm operates as: for each Gene G_target in the descendant Genome, evaluate the Recombination Policy's gene_resolution_rule(G_target) over the parent Genomes' Gene sets; emit the resolved Gene value as the descendant Genome's Gene at the target position. Resolution rules include: SINGLE_PARENT (one designated parent's Gene is inherited), DOMINANT (one parent's Gene dominates per a dominance Trust-Block reference), RECOMBINED (multiple parents' Gene values are recombined under a Gene-type-specific recombination function), DERIVED (the descendant Gene is computed as a function of parents' Genes — e.g., weighted average for numeric Genes), ACQUIRED (the descendant inherits a Gene value explicitly added by the inheritance policy author rather than computed from parents). Per, the Lamarckian property (inheritance of acquired state) is a protocol-level affirmative design choice — descendant Genomes may inherit acquired governance state that did not exist in any parent.

Multi-Genome State Isolation

Mechanism enforcing strict state separation across Genome strands. Cross-strand information flow requires explicit Trust-Block-bound bridging operations that record provenance.

Multi-Genome State Isolation provides cryptographic isolation between concurrent Genomes operating in the same protocol context. A participant may hold multiple Genomes — for example, one Personal Genome bound to their primary PPN, one Operational Genome bound to a temporary delegation, one Inheritable Genome bound to a planned successor PPN. State Isolation enforces that Gene values, Strand modifications, and Genome operations on one Genome do not leak into another Genome without explicit Trust-Block-bound cross-Genome operations. Implementation: each Genome operates within a Genome-scoped QPC (per U.S. Patent No. 12,316,610 B1 QPC primitive) with Trust Criteria forbidding cross-QPC state references except through Trust-Block-bound message envelopes.

Governance Superposition

Algorithmic primitive permitting a participant's DNA to be expressed in multiple overlapping governance contexts simultaneously. Each context projects only the relevant Genes; cross-context interference is prevented by per-context Trust Criteria.

Governance Superposition is the architectural primitive by which a single participant or resource may be governed simultaneously under multiple Genome contexts, each Genome contributing partial governance authority over specific operations. Implementation: a Governance Context Resolver evaluates the active Genome context at each operation invocation site and computes the operative governance authority as a Trust-Block-bound aggregation function over the contributing Genomes' Governance Strands. Per, Governance Superposition is the architectural basis for cross-Privacy-Domain governance: a resource exists in multiple Privacy Domains simultaneously, each Privacy Domain contributing its own Governance Strand to the resource's aggregate governance state.

Mitochondrial DNA Analog

Auxiliary inheritance line tracking operational infrastructure provenance (which Catalyst Manager, which Accelerator) separate from primary governance Genome lineage. Inherited maternally — i.e., from the operational-infrastructure parent, not the governance parent.

The Mitochondrial DNA Analog provides asymmetric single-parent inheritance for the Operational Lineage Strand, distinct from the polygenomic multi-parent inheritance applied to other Strand Types. Implementation: at descendant Genome composition, the Recombination Policy designates one parent Genome as the Operational Lineage Parent; the descendant's Operational Lineage Strand is inherited single-parent from that parent without recombination. Per, this asymmetry preserves Operational Lineage continuity (audit trails, operational state history, lineage-bound trust ratings) across descendant generations while permitting the bulk

of the Genome to undergo polygenomic recombination. The Operational Lineage Parent designation is a Trust-Block-bound act and is itself auditable.

Premium Inheritance via Trust Block Chain

Mechanism propagating Premium accruals through Trust Block chains. When a Resource Derivative is created from an upstream Resource, the derivative's Genome inherits the upstream Resource's Premium accruals (proportional to derivation share) such that downstream settlement reflects upstream contribution.

Premium Inheritance is the Family J–Family O integration primitive by which Resource Derivative Premium values are inherited by descendant Resource Derivatives. Implementation: at descendant Resource Derivative creation, the Premium Inheritance Engine evaluates the operative Compression Curve at the descendant-creation block height, computes the parent-Premium-to-descendant-Premium transformation under a Compression-Curve-indexed inheritance function, and emits the descendant Resource Derivative's Premium as a Trust-Block-bound function of the parent Premiums and the Curve position. Per Family J Premium Schedule and the Compression Curve definition in the Catalyst Launch Plan, Premium inheritance preserves the compression discipline: descendant Premiums cannot exceed parent Premiums adjusted for Curve compression over the elapsed block-height interval.

Multi-Layered Aggregation Principle

Foundational aggregation primitive permitting governance-preserving composition across layers: contributions aggregate into resources, resources into derivatives, derivatives into solutions, solutions into exchange-routable artifacts. At each layer, governance Genome composes from constituent Genomes via deterministic merge rules.

The Multi-Layered Aggregation Principle governs cross-Strand and cross-Genome aggregation under protocol-grade auditability constraints. Implementation: each aggregation operation (e.g., "compute the aggregate Reputation across this Privacy Domain's participant Genomes") references an Aggregation Function Record that is itself Trust-Block-anchored, specifying: the input set (which Genomes / Strands / Genes feed into the aggregation), the aggregation function (algebraic, weighted-sum, threshold, max, etc.), the output schema (single-value, vector, histogram), the aggregation-block-height (at which ledger height the aggregation is computed). The Multi-Layered property: aggregations may themselves be inputs to further aggregations, with each layer Trust-Block-anchored independently. Per, this supports population-scale inheritance and reputation derivation without sacrificing individual-participant audit transparency.

Process Flow

On Resource creation: the Resource's Genome is composed from constituent contribution Genomes via the Multi-Layered Aggregation Principle.

On a governance event (Premium accrual, attestation): the Lamarckian Inheritance Engine updates the relevant Genome.

On Resource Derivative creation: the derivative inherits the parent Resource's Genome (including Lamarckian acquired characteristics) plus per-derivative additions; Premium Inheritance ensures upstream Premium accruals propagate proportionally.

On new PPN creation: Family P's reproduction model invokes Genome recombination from parents.

On context-specific operation (e.g., enterprise context requiring professional-DNA): Governance Superposition projects only the relevant Genes from the active Genome strand.

On infrastructure-provenance query: the Mitochondrial DNA Analog provides the operational lineage independently of the governance lineage.

Alternative Embodiments

- **Sparse-Gene embodiment**: in deployments where most Genes are default-valued, the Genome may store only deviations from default, with default Genes inferred at evaluation time.
- **Versioned DNA embodiment**: DNA strands may carry version metadata; descendant entities may pin to specific upstream DNA versions, supporting governance-stability guarantees.

- **Cross-Accelerator Genome migration**: a participant may migrate Genome strands across Accelerators with Trust-Block-bound migration attestations.
- **Genome Versioning embodiment**: Beyond the existing Versioned DNA embodiment, Genomes themselves may carry version metadata at the whole-Genome level: a Genome at version v1.2 may be referenced as a specific snapshot; a descendant Genome composition may operate against a specified parent-Genome-version rather than the parent's current state. This embodiment supports point-in-time inheritance and temporal-snapshot composition without requiring full Genome immutability.
- **Conditional Gene Resolution embodiment**: Recombination Policies may include conditional resolution rules: a Gene's resolution may depend on observable Genome state — "Gene G inherits from parent_A if parent_A's Reputation Strand value at block H exceeds threshold T, else from parent_B". This embodiment supports performance-conditioned inheritance where descendant Genomes preferentially inherit from higher-performing parents.
- **Federated Genome embodiment**: Multiple distinct Genomes belonging to a federation of participants may be aggregated into a Federated Genome record reflecting the federation's collective governance state. The Federated Genome is itself Trust-Block-bound and supports federation-scale operations (collective voting, collective reputation derivation, collective Resource Pool admission rules) without requiring identity-level homogenization.
- **Cross-Domain Genome Mirror embodiment**: A Genome may be mirrored across Privacy Domains: a participant's Personal Genome bound to one Privacy Domain may be mirrored as a read-only reference in another Privacy Domain, supporting cross-domain reputation propagation and cross-domain governance acknowledgment without full Genome migration. Mirror records are Trust-Block-bound and explicit; cross-domain mirror authorization is governed by both Privacy Domains' Trust Criteria.
- **Quantum-Safe Gene Encoding embodiment**: Gene values may be encoded under quantum-safe cryptographic primitives — lattice-based commitments, hash-based signatures over Gene values, post-quantum-safe Trust Block signature schemes. This embodiment provides forward-secrecy against future quantum-cryptographic attacks; encoding migration to post-quantum-safe schemes is itself Trust-Block-bound and reversible only under quorum-Senior governance signatures.
- **Lazy-Materialized Genome embodiment**: Descendant Genomes may be lazily materialized: the descendant Genome's parent-references and Recombination Policy are emitted at composition time, but individual Gene values are materialized only on first reference. This embodiment reduces composition cost for large Genomes where most Genes are never queried; on-demand materialization is deterministically replayable from the parent references and Policy.
- **Multi-Generation Inheritance Audit embodiment**: An auditor may invoke a Multi-Generation Inheritance Audit primitive that traverses an arbitrary number of generations of Genome inheritance from a current descendant Genome back to ancestral root Genomes, verifying each generation's Trust-Block-bound composition. Audit results are themselves Trust-Block-bound and reusable: an attested audit record may be referenced by subsequent auditors without re-traversal.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for Genome inheritance operations:** (i) all parent Genomes must be Trust-Block-anchored and verifiable at the recombination-block height; (ii) the operative Recombination Policy must be Trust-Block-anchored and not in a future-only-valid state; (iii) the Replay Seed must be derivable from the parent Genome UUIDs and the Policy UUID.
- **Postconditions:** (i) the descendant Genome is emitted as Trust-Block-bound with parent-references; (ii) each constituent Gene reflects the resolved value with parent-Gene references; (iii) the Operational Lineage Strand is inherited single-parent per the Mitochondrial DNA Analog.
- **State transitions:** Genome states include DRAFT (under composition), COMPOSED (Trust-Block-bound but not yet bound to an entity), BOUND (bound to a participant or resource), FROZEN (no further modification), RETIRED (no longer operative).

- **Error handling:** parent-Genome-verification failures emit Audit Log records and abort composition; Recombination Policy ambiguity (under-specified resolution rules) triggers POLICY_AMBIGUOUS states requiring explicit resolution-rule extension.
- **Performance characteristics:** Genome composition is O(G) per G Gene positions; replay verification is O(G) per Gene; cross-Genome aggregation under the Multi-Layered Aggregation Principle is O(L × G) per L aggregation layers.

Cross-Family Integration

Upstream Dependencies. Family O operates within QPN-enabled infrastructure per the §22.7 Wherein clause. Quantum Gene, Quantum DNA, Quantum Genome primitives are Trust-Block-bound records on the Authorization Ledger (Family D). Inheritance operations consume Family D state, Family K identity binding state, and Family J Premium schedule snapshots for Premium-Inheritance computation per Pattern #12 (the architectural analysis).

Downstream Consumers. Family O is consumed by virtually every other Family for inheritance and governance purposes: Family A (Sidecar configurations inherit through Quantum DNA), Family B (Mode configurations inherit per the Pass 2-β Mode-Inheritance embodiment), Family F (CCP inheritance), Family J (Resource Derivative Premium inheritance per Pattern #12), Family L (Senior Derivative governance Genes for issuer-side decision rules), Family M (Accelerator Governance DNA), and most directly Family P (PPN Reproduction inherits all of Family O's primitives wholesale). Per Patterns #10 and #11, the Family O + Family P Inheritance Engine integration is the architectural basis for PPN reproduction; the Mitochondrial DNA Analog primitive binds Operational Lineage inheritance to single-parent semantics distinct from polygenomic governance recombination.

Lateral Interactions. Family O's Governance Superposition primitive interoperates with Family K's Six-Layer Catalyst Architecture — a participant operating in multiple Privacy Domains is governed by a superposition of those domains' Governance Strands. Family O's Multi-Genome State Isolation interoperates with Family K's Authorized Catalyst Proxy Addresses — a participant may hold distinct Genome contexts for their primary PPN and their delegated proxy operations.

Emergent System-Level Properties. Three emergent properties: (a) Pattern #10 + Pattern #11 Family O + Family P Inheritance Engine + Mitochondrial Analog produces protocol-grade lineage continuity across PPN generations; (b) Pattern #12 Family J + Family O Premium Inheritance Engine produces Resource Derivative Premium continuity under inheritance — descendant Resource Derivatives' Premium values are deterministically derived from parent Premiums adjusted for Compression Curve progression. Together these properties make the QPN inheritance-aware at every layer.

Family P — Two-Parent PPN Creation + Polygenomic Resource Derivative Recombination

Encompassing candidate disclosures: UE-5, UE-6.

Field Summary

Two-parent sexual reproduction model for Personal Privacy Network creation and N-parent polygenomic recombination for Resource Derivative creation.

Family P is the reproduction-architecture layer of the QPN. Per and, the Two-Parent PPN Creation and N-Parent Polygenomic Resource Derivative Recombination architecture provides protocol-grade primitives for the instantiation of new Personal Privacy Networks (PPNs) from one or more parent PPNs through deterministic Genome recombination. The architecture extends Family O's Quantum DNA inheritance primitives to the network-instantiation case: an entire PPN is a descendant Genome composition rather than a parameterized clone.

Problem Addressed

Creation of new entities (Personal Privacy Networks, Resource Derivatives) from existing entities requires a deterministic Genome recombination function that (a) preserves governance characteristics from parents, (b) supports multi-parent creation for Resource Derivatives that aggregate multiple sources, (c) structurally

distinguishes the two-parent case (relevant for PPN creation, e.g., joint household) from the N-parent case (relevant for multi-source Derivatives), and (d) is deterministically replayable for audit. Prior systems either lack a structured recombination primitive entirely or apply ad-hoc merge rules with no replay guarantees.

Conventional digital-network instantiation architectures provide either copy-on-write replication (which inherits no governance state) or parametrized configuration (which inherits no attribution state). Neither is sufficient for PPN reproduction in the QPN context: a descendant PPN must inherit the governance, attribution, and economic state of its parent or parents in a cryptographically-anchored, deterministically-computable manner, while still permitting the descendant PPN to diverge from its parents through its own operational lineage.

The two-parent case (the canonical reproduction primitive) addresses this directly: a descendant PPN is instantiated from a pair of parent PPNs through deterministic Genome recombination, with each Quantum DNA strand recombined under the parent-pair-specific Recombination Policy. The N-parent generalized case addresses polygenomic situations in which the descendant PPN inherits from more than two parents — e.g., a community-formed PPN instantiated from contributions of N founding participants, or an organizationally-spun-off PPN inheriting from N predecessor PPN states.

Solution Overview

The Two-Parent Sexual Reproduction Model is a deterministic Genome recombination primitive specialized for new Personal Privacy Network creation from exactly two parent participants. Genome recombination follows configurable inheritance rules (typically: equal weighting of dominant Genes, recessive Gene handling per inheritance pattern). The Polygenomic Recombination primitive generalizes to N-parent recombination (N ≥ 2) for Resource Derivative creation: each parent contributes a weighted Genome share (weights derive from derivation share), and the resulting derivative Genome composes via deterministic merge rules ensuring governance preservation.

Reproduction Initiator components solicit and validate the parent Genomes, evaluate the Recombination Policy, and emit a Reproduction Trust Block authorizing the descendant Genome composition. The Recombination Engine performs the deterministic Genome composition from the parent Genomes under the operative Recombination Policy. The Descendant Instantiator binds the resulting Genome to a newly-instantiated PPN structure, recording the lineage on the Settlement Ledger as Trust-Block-bound parent-references.

PNX-wide registry publishability extends to PPN reproduction: the full graph of PPN parent-descendant relationships is publishable at any specified ledger height as a protocol-grade transparency surface. This is essential for inheritance auditing at population scale: any participant can verify the lineage of any PPN against the published registry, with full cryptographic integrity guarantees.

Resource Derivative Premium recombination is integrated with Family J: when a descendant PPN inherits Resource Derivatives from its parents, the descendant's Premium computation references the parent-Premiums under a Premium Inheritance Engine that applies the Compression Curve appropriately. This ties PPN reproduction economics into the Catalyst Network's Premium discipline and the broader QPN capital-formation architecture.

Components

Components detailed below: Reproduction Initiator, Recombination Engine, Descendant Instantiator, Reproduction Trust Block Manager, PNX-Wide Registry Publishability Manager, Premium Inheritance Engine, Lineage Audit Logger. Sources: (reproduction architecture), §§2.8–2.11 (PPN inheritance), and

Two-Parent PPN Recombination Engine

Operates within a Reproduction-scoped QPC. Inputs: parent A Genome, parent B Genome, recombination policy. Outputs: child PPN initial Genome bound to both parents' Trust Blocks. Recombination policy specifies per-Gene inheritance rules.

The Two-Parent PPN Recombination Engine is implemented as follows: input is two parent Genomes (parent_A, parent_B) and a Recombination Policy R; output is a descendant Genome composed Gene-by-Gene under R. The algorithm: for each Gene position G in the descendant Genome schema, evaluate R.gene_resolution_rule(G, parent_A.gene_at(G), parent_B.gene_at(G)) producing the descendant Gene value; emit the descendant Gene as Trust-Block-bound with parent_gene_uuid_set = {parent_A.G.uuid, parent_B.G.uuid}. The descendant Genome

is itself emitted as a Trust-Block-bound record with parent_genome_uuid_set = {parent_A.uuid, parent_B.uuid}. Per, the two-parent case is the canonical PPN reproduction primitive; N-parent reproduction follows the same algorithmic pattern with R extended to evaluate over N parents.

N-Parent Polygenomic Recombination Engine

Generalized recombination engine for N ≥ 2 parents. Inputs: parent Genomes, parent derivation-share weights, recombination policy. Outputs: derivative Genome bound to all N parents' Trust Blocks. Weighted Gene composition reflects derivation-share weighting.

The N-Parent Polygenomic Recombination Engine generalizes two-parent recombination to arbitrary parent counts. Implementation: input is an ordered set of N parent Genomes {parent_1,..., parent_N} and a Recombination Policy R; output is a descendant Genome composed under R. The Recombination Policy specifies, per Gene position: the N-parent resolution rule (SINGLE_PARENT_FROM_SET with parent index, WEIGHTED_AVERAGE with per-parent weights, MAJORITY with N/2+1 threshold, UNANIMOUS with N-of-N threshold, or DERIVED with a Trust-Block-anchored function over all N parents' Genes). Per, the polygenomic primitive supports community-formed PPNs (N founding members each contributing partial Genome state), organizationally-spun-off PPNs (inheritance from N predecessor PPN states), and merger-style PPN composition (N predecessor PPNs merging into a single descendant PPN). All Gene-level inheritance lineage is preserved in the descendant Genome's parent_gene_uuid_set references for replay-grade auditability.

Recombination Policy

Configurable rule set specifying per-Gene inheritance: dominant inheritance (any parent's dominant Gene propagates), recessive inheritance (Gene propagates only if all parents share it), weighted-average (Gene value computed as weight-averaged parent values), or custom function (operator-defined).

A Recombination Policy is a Trust-Block-anchored record specifying per-Gene-type resolution rules. The schema includes: policy_uuid, policy_authority_did (the entity authorized to author / modify this Policy), gene_resolution_rule_set (a mapping from Gene Type → resolution rule), strand_inheritance_default_set (Strand-level inheritance defaults applicable when per-Gene rules are absent), mitochondrial_strand_set (the Strand Types that follow single-parent inheritance per Family O), policy_immutable_after_block_height (optional immutability marker beyond which the Policy may not be modified). Per, Recombination Policies are inheritance-time bindings: at descendant Genome composition, the operative Policy at that block height governs; subsequent Policy modifications do not retroactively affect prior descendant Genomes.

Deterministic Replay Seed Binding

Each recombination event seeds its deterministic function with a Trust-Block-derived seed, ensuring any verifier holding the inputs can replay the recombination bit-identically. Inherits the deterministic replay primitive from the November 18, 2025 AI Governance Provisional §5.8.

Deterministic Replay Seed Binding is the protocol primitive ensuring that recombination operations involving randomness or stochastic Gene resolution are deterministically replayable. Implementation: at recombination initiation, a replay_seed (typically a 256-bit value derived deterministically from the parent Genome UUIDs, the Recombination Policy UUID, and the recombination-block height) is computed and bound to the descendant Genome's Trust Block. Any subsequent auditor can recompute the descendant Genome from the parent Genomes, the Policy, and the seed without privileged access. Per, the Seed-Binding property is critical for population-scale inheritance audit: in deployments with millions of PPNs across multiple generations, only deterministic-replay-capable composition can support third-party verification at scale.

Process Flow

PPN Creation: two participants invoke joint PPN creation; both parent Genomes are presented to the Two-Parent Recombination Engine; the child PPN's initial Genome is composed; the child PPN is bound to both parents' Trust Blocks.

Resource Derivative Creation: N parents (typically including upstream Resources and contributing participants) present Genomes plus derivation-share weights; the Polygenomic Recombination Engine composes the derivative Genome.

Deterministic replay: any verifier holding the parent Genomes and the recombination policy can replay the Recombination Engine to obtain the bit-identical child or derivative Genome.

Cross-reference to Family O: the recombination output Genome enters the Multi-Layered Aggregation pipeline at the appropriate layer.

Alternative Embodiments

- **Asymmetric two-parent embodiment**: in two-parent PPN creation, parents may agree to asymmetric weighting (e.g., 70/30) reflecting differential consent or contribution.
- **Time-staged recombination embodiment**: a Resource Derivative may be created in stages, with each stage adding a new parent and re-running the recombination; useful for collaborative Resource creation where contributors join over time.
- **Limited-recombination embodiment**: certain Gene types may be marked non-recombinable (e.g., jurisdiction Genes that must match across all parents), with recombination rejected if non-matches occur.
- **Asymmetric-Strand recombination embodiment**: Beyond the existing Asymmetric two-parent embodiment, Recombination Policies may specify Strand-level asymmetry: the Governance Strand may follow majority recombination, the Economic Strand may follow weighted-average recombination, the Reputation Strand may follow single-parent inheritance from the higher-reputation parent. Each Strand-level resolution rule is independently Trust-Block-anchored.
- **Phased-Reproduction embodiment**: PPN reproduction may proceed in phases: Phase 1 composes the Governance Strand and Operational Lineage Strand; Phase 2 composes the Economic Strand and Reputation Strand; Phase 3 composes Resource-Specific Strands. Each phase is Trust-Block-bound separately, supporting cases where parent-PPN consent may be staged (e.g., one parent consents to immediate Governance inheritance but defers Economic Strand inheritance pending operational maturity).
- **Inheritance-Veto embodiment**: Designated authorities may exercise an Inheritance-Veto primitive over specific Genes or Strands during recombination: a participant's testamentary-executor analog may veto inheritance of specified Genes; a regulatory authority may veto inheritance of jurisdictional-compliance Genes that conflict with the descendant's target jurisdiction. Veto invocations are Trust-Block-bound and explicit; the descendant Genome reflects the post-veto resolved state.
- **Adoptive-PPN embodiment**: An existing PPN may be adopted into a descendant relationship with a newly-designated parent PPN: rather than reproducing a new descendant, an existing PPN may have its parent-references retroactively bound to a new parent under explicit Trust-Block-bound adoptive consent. This embodiment supports merger-and-acquisition-style PPN combinations where pre-existing PPNs gain new ancestor relationships.
- **Time-Bounded Reproduction Window embodiment**: Reproduction Policies may carry time-bounded validity: a Policy may be valid only between specified block heights, after which Policy-bound reproductions are no longer permitted absent Policy renewal. This embodiment supports finite-duration reproduction programs (e.g., a community-PPN founding window that closes once N founding members have contributed).
- **Reverse-Mitochondrial embodiment**: Beyond the Mitochondrial DNA Analog single-parent inheritance for Operational Lineage Strand, a Reverse-Mitochondrial embodiment supports cases where a different Strand (e.g., the Governance Strand) follows single-parent inheritance while Operational Lineage follows polygenomic recombination. Mitochondrial-Strand designation is Trust-Block-anchored at the Recombination Policy level.
- **Stochastic Recombination embodiment**: For specified Gene types (typically tie-breaking Genes, randomized identifier Genes), Recombination Policies may specify stochastic resolution under Deterministic Replay Seed Binding: a deterministic pseudo-random function of (parent_Genome_UUIDs, Policy_UUID, recombination_block_height, gene_position) selects the resolved value. This embodiment supports fair-tie-breaking in symmetric-parent scenarios without sacrificing replay-grade auditability.

Process Flow Expansion — Preconditions, Postconditions, State Transitions, Error Handling.

- **Preconditions for PPN reproduction operations:** (i) all designated parent PPNs must be in BOUND state with current operational PPN records; (ii) all parent Genomes must be verifiable at the reproduction-block height; (iii) the Recombination Policy must be Trust-Block-anchored; (iv) the Reproduction Trust Block authorization must be signed by all parent PPNs' authorizing parties (per parent-specific governance Strand).
- **Postconditions:** (i) the descendant Genome is composed and bound to the newly-instantiated descendant PPN; (ii) the descendant PPN's structural primitives (Resource Pools, Trust Criteria, Privacy Domain bindings) are initialized per the Recombination Policy; (iii) the parent-descendant lineage is recorded in the PNX-Wide Registry.
- **State transitions:** descendant PPN states transition through REPRODUCTION_PENDING → RECOMBINATION_IN_PROGRESS → DESCENDANT_INSTANTIATED → BOUND.
- **Error handling:** missing parent authorization signatures emit Audit Log records and abort reproduction; Recombination Policy failures (incompatible parent Genome schemas) trigger SCHEMA_INCOMPATIBLE states requiring schema-bridging Recombination Policy extension.
- **Performance characteristics:** PPN reproduction is dominated by Genome composition cost (O(G) per G Gene positions across all Strands); registry-update propagation is O(R) per R registry-replicated nodes; for large-N polygenomic reproduction with N parents, the per-Gene resolution-rule evaluation cost grows linearly with N, yielding total reproduction cost O(N × G).

Cross-Family Integration

Upstream Dependencies. Family P operates within QPN-enabled infrastructure per the §22.7 Wherein clause and depends wholesale on Family O's Quantum Gene, Quantum DNA, Quantum Genome, Inheritance Engine, Recombination Policy, and Mitochondrial DNA Analog primitives. Family P consumes Family D Authorization Ledger state for Reproduction Trust Block authorization records, Family K Multi-Factor Identity Binding state for parent-PPN governance authorization, and Family F.qpn container lineages for substrate-side reproduction continuity.

Downstream Consumers. Family P's Reproduction events are recorded on the Authorization Ledger (Family D) and consumed by every Family requiring lineage awareness — Family E (Contribution Graph edges may reference descendant PPN's parent lineage), Family J (Resource Derivative Premium inheritance per Pattern #12), Family K (participant Levels and Reputation may carry through descendant PPN bindings), Family L (Senior Derivative holder relationships may transfer to descendant PPNs under explicit Trust-Block-bound transfer authorities), Family M (Accelerator spawning operations create descendant Accelerators per Family P primitives). The PNX-Wide Registry publishability of PPN parent-descendant relationships is consumed by inheritance auditors, regulators, and population-scale Catalyst Network analytics.

Lateral Interactions. Family P's Two-Parent and N-Parent Polygenomic Recombination Engines interoperate with Family O Recombination Policy definitions and Mitochondrial DNA Analog single-parent designations. The Pass 2-α adoptive-PPN embodiment interoperates with Family K's identity-rebinding primitives — an adopted PPN's new parent-references are themselves Trust-Block-bound under explicit adoptive consent.

Emergent System-Level Properties. Pattern #10 and Pattern #11 are the canonical Family P-anchored emergent properties: (a) Family O + Family P Inheritance Engine integration produces deterministic-replay-auditable PPN reproduction across arbitrary generations; (b) Mitochondrial-Analog single-parent Operational Lineage inheritance preserves audit-trail continuity even under polygenomic governance recombination. Together these properties make the QPN structurally capable of population-scale, multi-generational reproduction with full lineage auditability — a property essential for the long-horizon coverage and inheritance claims in (75-year QPT Derivative NPV horizon).

CLAIMS

What is claimed is:

The following 457 claims (54 independent + 403 dependent) are organized into 16 Claim Families (A through P). Each independent claim incorporates the canonical §22.7 Wherein clause anchoring 2016 priority on the recited QPN infrastructure under U.S. Patent No. 12,316,610 B1.

May 12, 2025 Quantum Privacy Drilldown Provisional — discloses Privacy Network instantiation primitives that PPN Reproduction extends (Drilldown §§5-7).

October 7, 2025 PNX Provisional — discloses PNX-wide registry primitives for publishability of cross-PPN lineage (PNX Provisional §§6-8).

December 2025 Self-Funding QPX Provisional — discloses N-Parent Polygenomic Recombination Engine, Reproduction Trust Block authorization, and Deterministic Replay Seed Binding primitives (QPX Provisional §§7-9).

Family A — Quantum Privacy Sidecar & Witness Architecture

Claims 1–40 (40 claims: 3 independent + 37 dependent)

Family Introduction

Family A captures the Quantum Privacy Sidecar pattern, the three-plane decomposition, and the browser-extension embodiment. Each independent claim recites the four-component Sidecar (Witness Agent, Listener Agent, Local Vault, Verification Bridge) executing within distinct Quantum Privacy Cells, with the §22.7 canonical Wherein clause anchoring 2016 priority on the QPN infrastructure.

Claims List

Claim 1. A trust-verified contribution capture system, comprising:

a Witness Agent executing within a first Quantum Privacy Cell (QPC), configured to capture user-originated interactions and emit Witness Records bound to an originating Privacy Domain;

a Listener Agent executing within a second QPC distinct from the first QPC, configured to emit Listener Attestations for ambient signals authorized under standing Trust Criteria;

a Local Vault executing within a third QPC distinct from the first and second QPCs, configured to store Witness Records under encryption keys whose private components do not leave the Local Vault QPC boundary in plaintext;

a Verification Bridge executing within a fourth QPC distinct from the first, second, and third QPCs, configured to construct Verification Envelopes referencing Vault-resident Witness Records by content-addressed identifier, sign the Verification Envelopes under Trust Block keys, and present the signed Verification Envelopes to a Catalyst Network ingress for Proof-of-Trust verification;

an inter-component message bus configured to mediate communication among the Witness Agent, Listener Agent, Local Vault, and Verification Bridge exclusively through Trust-Block-bound message envelopes whose authorization is verified by Proof-of-Trust at each hop;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a single compromise of any component QPC does not yield plaintext access to Vault-resident contribution content and does not enable unauthorized issuance of Verification Envelopes to the Catalyst Network.

Claim 2. The system of claim 1, wherein the Witness Agent captures at least one of: pointer interactions within a designated input scope, keystroke events within an authorized application surface, document-edit events at section granularity, and explicit voice-activation events.

Claim 3. The system of claim 1, wherein the Listener Agent emits Listener Attestations describing existence and class of an ambient signal without capturing content of the ambient signal.

Claim 4. The system of claim 1, wherein the Local Vault is configured such that encryption keys for stored Witness Records are derived from a Privacy Domain master key by a key derivation function whose inputs include a Trust Block lineage identifier.

Claim 5. The system of claim 1, wherein the Verification Bridge is configured to compute the content-addressed identifier of a Witness Record using a quantum-safe hash function specified in a Trust Criteria of the originating Privacy Domain.

Claim 6. The system of claim 1, wherein the inter-component message bus enforces a unidirectional message-flow constraint from the Witness Agent and Listener Agent toward the Local Vault and from the Local Vault toward the Verification Bridge, and rejects messages flowing in the reverse direction.

Claim 7. The system of claim 1, further comprising a Synchronization Scheduler component executing within a fifth Quantum Privacy Cell and configured to emit the synchronization trigger upon at least one of: an elapsed-time condition, an event-count condition, and an explicit user-invocation condition.

Claim 8. The system of claim 1, wherein the Verification Envelope additionally references a Mode Tag identifying one of: an Active Mode, a Directed Mode, an Ambient Mode, an Institutional Mode, and an Evangelized Mode.

Claim 9. The system of claim 1, wherein the Verification Bridge is unable to decrypt Local Vault contents and is constrained by Trust Criteria to certify only that the Verification Envelope was authorized by a specified Trust Block.

Claim 10. The system of claim 1, wherein each component QPC enforces a quantum-safe cryptographic boundary specified by Trust Criteria of the originating Privacy Domain.

Claim 11. The system of claim 1, wherein the system is embodied as a browser extension whose manifest pins each component QPC to a specified QPN origin, and wherein the browser extension supplements Trust Block authorization with extension-origin verification.

Claim 12. The system of claim 1, wherein the system is embodied as a native application of an operating system, and wherein each component QPC additionally executes within a respective operating-system sandbox.

Claim 13. The system of claim 1, wherein the system is embodied as a system-level service of a mobile operating system, and wherein the Local Vault QPC is bound to a hardware-isolated cryptographic enclave of the mobile device.

Claim 14. The system of claim 1, wherein the system is embodied as a server-side process group of an organizational endpoint, and wherein each plane of a Three-Plane Architecture is hosted in a distinct hosting environment of the server-side process group.

Claim 15. The system of claim 1, further comprising a Replay Verifier configured to deterministically replay Verification Envelope construction from Local Vault contents and Trust Block lineage and assert byte-for-byte identity with a prior Verification Envelope.

Claim 16. The system of claim 1, wherein the Witness Agent additionally records, in association with each Witness Record, a user-action signature derived from at least one of: a user-input event signature and a session-context attestation.

Claim 17. The system of claim 1, wherein, upon a Permanent Privacy Seal being asserted by the originating participant for a specified Witness Record, the Local Vault destroys the encryption key associated with the specified Witness Record under control of Trust Block-bound key destruction logic.

Claim 18. A computer-implemented method for trust-verified contribution capture, the method executing within a Quantum Privacy Network (QPN) and comprising:

observing, by a Witness Agent executing within a first Quantum Privacy Cell, a user-originated interaction and packaging the interaction as a Witness Record bound to an originating Privacy Domain;

encrypting the Witness Record under a Privacy Domain key and writing the encrypted Witness Record to a Local Vault executing within a second Quantum Privacy Cell;

concurrently emitting, by a Listener Agent executing within a third Quantum Privacy Cell, a Listener Attestation describing an ambient signal authorized under a Trust Block-bound Signal Input Mode;

upon a synchronization trigger, constructing, by a Verification Bridge executing within a fourth Quantum Privacy Cell, a Verification Envelope referencing the encrypted Witness Record by content-addressed identifier;

signing the Verification Envelope under Trust Block keys and presenting the signed Verification Envelope to a Catalyst Network ingress for Proof-of-Trust verification;

upon Proof-of-Trust verification, updating, by a Settlement Controller, a Contribution Ledger with a reference to the Verification Envelope without traversal of plaintext contribution content beyond the originating Privacy Domain boundary;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby contributions are admitted to the Catalyst Network with cryptographic authorization while plaintext contribution content remains within the participant's Privacy Domain.

Claim 19. The system of claim 18, wherein the user-originated interaction comprises at least one of: a click event, a keystroke event within a designated input scope, a document-edit event, and a voice-activation event invoked by explicit user action.

Claim 20. The system of claim 18, wherein the synchronization trigger is determined by the Verification Bridge as the earliest of: an elapsed-time condition, an event-count condition, and an explicit user-invocation condition.

Claim 21. The system of claim 18, further comprising verifying, by the Catalyst Network ingress, that the Verification Envelope's signature was produced by a Trust Block whose Trust Criteria includes a current authorization for capture of the indicated Signal Input Mode.

Claim 22. The system of claim 18, wherein the Settlement Controller updates the Contribution Ledger only upon completion of a Three-Stage AI Evaluation Pipeline output authorizing the Verification Envelope.

Claim 23. The system of claim 18, further comprising, prior to writing the encrypted Witness Record to the Local Vault, classifying the user-originated interaction under a Mode Discriminator component executing within a fifth Quantum Privacy Cell and emitting a Mode Tag.

Claim 24. The system of claim 18, wherein the content-addressed identifier is computed as a quantum-safe cryptographic hash of the Witness Record under a hash function specified in the Trust Criteria of the originating Privacy Domain.

Claim 25. The system of claim 18, further comprising, upon Proof-of-Trust verification failure, rejecting the Verification Envelope at the Catalyst Network ingress and emitting a Trust Block-bound rejection record to the originating Verification Bridge without disclosing the Witness Record contents.

Claim 26. The system of claim 18, further comprising recording, by the Settlement Controller, a per-Witness Mode Tag in the Contribution Ledger and applying mode-specific Premium weights from a Premium Framework component during subsequent reward computation.

Claim 27. The system of claim 18, wherein the Privacy Domain key is derived by a key derivation function whose inputs include a participant-specific seed and a Trust Block lineage identifier, and wherein the key derivation function output is bound to the Privacy Domain QPC.

Claim 28. The system of claim 18, wherein the method is performed by a browser extension whose manifest pins each Quantum Privacy Cell to a specified QPN origin.

Claim 29. The system of claim 18, wherein the method is performed by a native operating-system application whose Quantum Privacy Cells additionally execute within respective operating-system sandboxes.

Claim 30. The system of claim 18, further comprising, upon assertion of a Permanent Privacy Seal by the originating participant for a specified Witness Record, destroying the encryption key associated with the specified Witness Record under control of Trust Block-bound key destruction logic.

Claim 31. The system of claim 18, further comprising verifying that the inter-component message bus rejects every message not bound to a Trust Block.

Claim 32. The system of claim 18, further comprising reflecting, in the Contribution Ledger, a content-addressed reference to the Verification Envelope and not the Verification Envelope's plaintext contents.

Claim 33. A three-plane trust-verified capture architecture, comprising:

a Data Plane comprising a first set of Quantum Privacy Cells configured exclusively to transport contribution payloads under Trust Block authorization;

a Control Plane comprising a second set of Quantum Privacy Cells, disjoint from the first set, configured exclusively to compute authorization decisions and emit Trust Blocks;

a Management Plane comprising a third set of Quantum Privacy Cells, disjoint from the first and second sets, configured exclusively to perform operator administration including configuration updates, key rotation events, and audit retrieval;

an inter-plane gateway configured to mediate inter-plane traffic exclusively through Trust-Block-bound envelopes whose authorization is verified by Proof-of-Trust against plane-specific Trust Criteria;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a compromise of any single plane does not yield authorization capability of another plane, and management actions cannot read data-plane payloads.

Claim 34. The system of claim 33, wherein the inter-plane gateway enforces a directional traffic constraint such that Management Plane components may issue control messages to the Control Plane but may not read Data Plane payloads.

Claim 35. The system of claim 33, wherein each plane's Trust Criteria specifies a plane-specific quantum-safe cryptographic boundary, and wherein the Data Plane Trust Criteria additionally prohibits Trust Block emission.

Claim 36. The system of claim 33, wherein the Control Plane is configured to emit Trust Blocks pursuant to authorization decisions and is structurally incapable of decrypting Data Plane payloads.

Claim 37. The system of claim 33, wherein, in a single physical host embodiment, each plane is mapped to a distinct hardware-isolated execution environment within the physical host.

Claim 38. The system of claim 33, wherein, in a distributed embodiment, each plane is mapped to a distinct hosting cluster and inter-plane traffic traverses the inter-plane gateway over a quantum-safe network protocol.

Claim 39. The system of claim 33, further comprising a Plane Auditor component executing within a Management Plane QPC and configured to verify that no traffic crossing the inter-plane gateway lacks a valid Trust-Block-bound envelope.

Claim 40. The system of claim 33, wherein the Three-Plane Architecture is integrated with a Quantum Privacy Sidecar such that the Sidecar's four components are mapped across the Data, Control, and Management Planes by component function.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family A closes the Quantum Privacy Sidecar **&** Witness Architecture gap, binding 3 independent claims and 37 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism*.

Family B — Five Signal Input Mode Architecture

Claims 41–65 (25 claims: 2 independent + 23 dependent)

Family Introduction

Family B captures the Five Signal Input Mode Architecture (Active, Directed, Ambient, Institutional, Evangelized) together with the Mode Discriminator and mode-specific Trust Criteria. The Family is structurally integrated because the discriminator and the five modes must operate together to preserve fine-grained authorization.

Claims List

Claim 41. A mode-discriminated contribution authorization system, comprising:

a Mode Discriminator executing within a Discriminator Quantum Privacy Cell and configured to classify each incoming signal among five non-overlapping Signal Input Modes consisting of Active Mode, Directed Mode, Ambient Mode, Institutional Mode, and Evangelized Mode;

a first Trust Criteria specifying authorization requirements for Active Mode comprising a user-action signature and a session-context attestation;

a second Trust Criteria specifying authorization requirements for Directed Mode comprising a prompt-context attestation and a user-response signature;

a third Trust Criteria specifying authorization requirements for Ambient Mode comprising a standing-consent Trust Block referencing a specific ambient surface and a decay-time;

a fourth Trust Criteria specifying authorization requirements for Institutional Mode comprising dual Trust Block authorization from both an individual Trust Block and an organizational Trust Block;

a fifth Trust Criteria specifying authorization requirements for Evangelized Mode comprising a witness Trust Block and an attested-party Trust Block, wherein the attested-party Trust Block is permitted to comprise a Manager-Originated dormant participant record;

a Mode Tag emitter component configured to bind a Mode Tag to each authorized signal and provide the Mode Tag to a downstream Settlement Controller;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby each contribution signal is captured under a mode-specific authorization regime and rejected at the QPC boundary if its Mode Tag does not satisfy the corresponding Trust Criteria.

Claim 42. The system of claim 41, wherein the Mode Discriminator's classification function is deterministic such that identical inputs yield identical Mode classifications across replays.

Claim 43. The system of claim 41, wherein the Active Mode signature comprises at least one of: a pointer-input signature, a keystroke-input signature, an authenticated voice-activation signature, and an explicit submit-event signature.

Claim 44. The system of claim 41, wherein the Directed Mode prompt-context attestation identifies an originating system component or peer participant Trust Block.

Claim 45. The system of claim 41, wherein the Ambient Mode standing-consent Trust Block specifies a decay-time after which standing authorization automatically expires.

Claim 46. The system of claim 41, wherein the Institutional Mode is structured such that the organizational Trust Block must be presented alongside the individual Trust Block at signal admission, and neither alone is sufficient.

Claim 47. The system of claim 41, wherein the Evangelized Mode permits the attested-party Trust Block to comprise a Manager-Originated DORMANT Quantum Privacy Cell record per a Catalyst Network Manager attribution framework.

Claim 48. The system of claim 41, further comprising a Mode Customization Interface component permitting an Enterprise Privacy Network to define an Internal Mode whose contributions do not propagate to a global Contribution Graph.

Claim 49. The system of claim 41, further comprising a Mode Customization Interface component permitting a healthcare embodiment to define a Clinical Mode whose Trust Criteria includes HIPAA-compliant authorization requirements.

Claim 50. The system of claim 41, wherein the Mode Tag emitter produces a Mode Tag comprising at least: the assigned Mode identifier, the participant Trust Block identifier, and a temporal binding.

Claim 51. The system of claim 41, wherein each Trust Criteria specifies a mode-specific decay-time after which an authorization is automatically revoked.

Claim 52. The system of claim 41, wherein the Mode Discriminator additionally enforces mutual exclusion among the five Modes such that no signal may carry more than one Mode Tag at admission.

Claim 53. The system of claim 41, wherein the system is integrated with a Quantum Privacy Sidecar of a contribution capture system, and wherein the Mode Discriminator operates upon signals emitted by the Sidecar's Witness Agent and Listener Agent.

Claim 54. The system of claim 41, wherein the Settlement Controller's mode-specific Premium weights are bound to a Premium Framework component, and wherein the Premium weights for Ambient Mode are bounded below the Premium weights for Active Mode pursuant to a Proportionality constraint.

Claim 55. The system of claim 41, wherein the Mode Discriminator emits, upon each rejection, a rejection attestation bound to a Control Plane log of a Three-Plane Architecture, without recording any contribution payload.

Claim 56. A computer-implemented method for mode-discriminated contribution authorization in a Quantum Privacy Network (QPN), the method comprising:

receiving, at a Mode Discriminator executing within a Quantum Privacy Cell, a candidate contribution signal carrying an origin envelope;

extracting from the origin envelope at least: a source attestation, a payload schema identifier, and a contextual signature;

applying a deterministic classification function over the extracted fields to assign the candidate signal to exactly one of five Signal Input Modes consisting of Active, Directed, Ambient, Institutional, and Evangelized;

comparing the assigned Mode against a participant-active Mode authorization set bound to a Trust Block;

if the assigned Mode is within the participant-active Mode authorization set, admitting the signal and emitting a Mode Tag bound to the signal; otherwise rejecting the signal at the Quantum Privacy Cell boundary without persistence;

providing the Mode Tag to a Settlement Controller for downstream mode-specific Premium weight application;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby fine-grained per-mode authorization is enforced at signal admission, and rejected signals leave no persistent record beyond a rejection attestation.

Claim 57. The system of claim 56, wherein extracting the contextual signature comprises computing a quantum-safe hash of contextual envelope fields under a hash function specified in Trust Criteria.

Claim 58. The system of claim 56, wherein the deterministic classification function returns identical Mode assignments across replays performed by independent verifiers having access to the same inputs.

Claim 59. The system of claim 56, further comprising emitting, upon rejection, a Trust Block-bound rejection attestation specifying a rejection reason and not specifying the rejected payload contents.

Claim 60. The system of claim 56, further comprising verifying, prior to admission, that a current temporal binding of the participant-active Mode authorization set has not exceeded a Trust-Block-specified decay-time.

Claim 61. The system of claim 56, wherein the participant-active Mode authorization set is mutable only by explicit user action authorized under a Trust Block.

Claim 62. The system of claim 56, further comprising emitting a Mode Tag whose temporal binding identifies a specific epoch for which the Mode was authorized.

Claim 63. The system of claim 56, wherein, for Evangelized Mode, admitting the signal comprises additionally verifying the presence of an attested-party Trust Block, the attested-party Trust Block being permitted to be a Manager-Originated DORMANT record having no economic function.

Claim 64. The system of claim 56, wherein, for Institutional Mode, admitting the signal comprises verifying that the organizational Trust Block's authorization scope encompasses the contribution category indicated by the payload schema.

Claim 65. The system of claim 56, wherein the deterministic classification function uses a feature vector derived from the source attestation, the payload schema identifier, and the contextual signature, and applies a fixed-policy classifier whose policy is bound to a Management Plane Trust Block.

Whereby Clause (Family-Level Structural Effect)

Whereby Family B closes the Five Signal Input Mode Architecture gap, binding 2 independent claims and 23 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family C — Three-Stage AI Evaluation Pipeline

Claims 66–92 (27 claims: 2 independent + 25 dependent)

Family Introduction

Family C captures the Three-Stage AI Evaluation Pipeline (Semantic Classification, Identity Enrichment, Temporal Durability) operating within distinct QPCs with deterministic-replay AI models. The pipeline is structurally integrated; each stage produces a Trust-Block-bound attestation consumed by the next.

Claims List

Claim 66. A three-stage AI-mediated contribution evaluation pipeline system, comprising:

a Semantic Classification Stage comprising a first AI model executing within a first Quantum Privacy Cell (QPC), configured to receive a contribution payload and an associated Mode Tag and to output a Semantic Class, a Class Confidence, and a Class Trust Block attestation;

an Identity Enrichment Stage comprising a second AI model executing within a second QPC distinct from the first QPC, configured to receive the Class Trust Block attestation, candidate participant Trust Block keys, and Multi-Factor Identity Binding artifacts, and to output a bound Participant Identifier, an Identity Confidence, and an Identity Trust Block attestation;

a Temporal Durability Stage comprising a third AI model executing within a third QPC distinct from the first and second QPCs, configured to receive the Identity Trust Block attestation, a historical contribution lineage from a Contribution Ledger, and corroborating Cross-Verification signals, and to output a Durability Weight, a Durability Confidence, and a Durability Trust Block attestation;

a Pipeline Coordinator executing within a fourth QPC configured to enforce stage ordering, verify dependency satisfaction across stages, and emit a composite Pipeline Attestation comprising the three stages' outputs and cross-stage bindings;

a Deterministic Replay primitive bound to each AI model and configured such that identical inputs yield identical outputs across replays performed by independent verifiers holding the same Trust Block lineage;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby each contribution is evaluated across three independent dimensions with cryptographically-attestable cross-stage bindings, and the entire evaluation is deterministically replayable for audit.

Claim 67. The system of claim 66, wherein the Semantic Classification Stage classifies the contribution payload into a taxonomy comprising at least nine primary Catalyst Contribution Categories.

Claim 68. The system of claim 66, wherein the Semantic Classification Stage additionally emits a sub-category specifier within the primary Category and a UX-tier annotation.

Claim 69. The system of claim 66, wherein the Identity Enrichment Stage's Multi-Factor Identity Binding artifacts include at least two independently-attested identity factors.

Claim 70. The system of claim 66, wherein the Identity Enrichment Stage rejects an Identity binding whose Identity Confidence is below a Trust Block-specified minimum.

Claim 71. The system of claim 66, wherein the Temporal Durability Stage's Durability Weight is computed by a function whose inputs include contribution age, corroboration count, and a Premium-bearing field of an originating Trust Block.

Claim 72. The system of claim 66, wherein the Temporal Durability Stage queries the Contribution Ledger through an authenticated read interface bound to a Pipeline Trust Block, and wherein the read interface returns only Trust-Block-bound lineage descriptors.

Claim 73. The system of claim 66, wherein each AI model's deterministic replay primitive is bound to a fixed model snapshot identified by a content-addressed identifier recorded in the Class, Identity, or Durability Trust Block attestation respectively.

Claim 74. The system of claim 66, wherein the Pipeline Coordinator verifies that the Class Trust Block, Identity Trust Block, and Durability Trust Block attestations form a strictly-ordered chain.

Claim 75. The system of claim 66, wherein each AI model executes within a respective QPC whose cryptographic boundary prohibits exfiltration of model parameters.

Claim 76. The system of claim 66, further comprising a Trust-Weight Engine component executing within a fifth QPC and configured to compute a composite trust weight over the three stages' confidence outputs and emit the composite trust weight as a field of the composite Pipeline Attestation.

Claim 77. The system of claim 66, wherein the Pipeline Coordinator rejects a contribution upon failure of any single stage and emits a rejection attestation specifying the failing stage without disclosing intermediate model outputs.

Claim 78. The system of claim 66, wherein, upon a Permanent Privacy Seal of the contribution payload, the Pipeline Coordinator destroys all intermediate stage outputs while preserving the composite Pipeline Attestation's cryptographic structure.

Claim 79. The system of claim 66, wherein the Pipeline is integrated with a Quantum Privacy Sidecar such that the Pipeline Coordinator consumes Verification Envelopes emitted by the Sidecar's Verification Bridge.

Claim 80. The system of claim 66, wherein the Pipeline is integrated with a Settlement Controller such that the Settlement Controller withholds Exchange Token issuance until receipt of a composite Pipeline Attestation whose composite trust weight exceeds a Trust Block-specified threshold.

Claim 81. The system of claim 66, wherein each AI model is replaceable under a Management Plane Trust Block, and wherein model replacement requires a fresh deterministic replay descriptor recorded in subsequent Pipeline Attestations.

Claim 82. The system of claim 66, wherein the Temporal Durability Stage's corroborating Cross-Verification signals include witness attestations conforming to a Cross-Verification Protocol of a Settlement Controller.

Claim 83. A computer-implemented method for three-stage AI-mediated contribution evaluation within a Quantum Privacy Network (QPN), the method comprising:

presenting an admitted signal carrying a Mode Tag to a Pipeline Coordinator;

executing, by the Pipeline Coordinator, a Semantic Classification Stage AI model within a first Quantum Privacy Cell that emits a Semantic Class, a Class Confidence, and a Class Trust Block attestation;

executing, by the Pipeline Coordinator, an Identity Enrichment Stage AI model within a second Quantum Privacy Cell that consumes the Class Trust Block attestation and emits a Participant Identifier, an Identity Confidence, and an Identity Trust Block attestation;

executing, by the Pipeline Coordinator, a Temporal Durability Stage AI model within a third Quantum Privacy Cell that consumes the Identity Trust Block attestation and a Contribution Ledger lineage and emits a Durability Weight, a Durability Confidence, and a Durability Trust Block attestation;

emitting a composite Pipeline Attestation comprising the three stages' outputs, the cross-stage bindings, and a Deterministic Replay descriptor sufficient to permit deterministic re-execution;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the contribution is evaluated under cryptographic separation of concerns and any verifier holding the Pipeline Attestation may deterministically reproduce the three evaluations.

Claim 84. The system of claim 83, wherein each AI model output is computed by a deterministic procedure such that two verifiers presented with identical inputs and the same Pipeline Attestation's Deterministic Replay descriptor produce byte-identical outputs.

Claim 85. The system of claim 83, wherein the Semantic Classification Stage applies a fixed model snapshot identified by a content-addressed identifier recorded in the Class Trust Block attestation.

Claim 86. The system of claim 83, wherein the Identity Enrichment Stage's emission depends, in part, on a Multi-Factor Identity Binding artifact set whose members each carry a Trust Block attestation.

Claim 87. The system of claim 83, wherein the Temporal Durability Stage's Durability Weight is increased upon receipt of additional Cross-Verification signals and decreased upon detected contradiction with prior lineage.

Claim 88. The system of claim 83, further comprising emitting, by the Pipeline Coordinator, a Trust-Block-bound replay receipt enabling third-party deterministic replay of the entire three-stage evaluation.

Claim 89. The system of claim 83, wherein the method is integrated with a Premium Framework component such that the composite Pipeline Attestation's composite trust weight is provided as an input to a Premium computation.

Claim 90. The system of claim 83, further comprising, upon any stage's confidence falling below a Trust Block-specified minimum, withholding emission of the composite Pipeline Attestation and emitting a rejection attestation identifying the failing stage.

Claim 91. The system of claim 83, wherein the Pipeline executes within a Catalyst Network ingress between a Verification Bridge of a Quantum Privacy Sidecar and a Settlement Controller of a Settlement-enforcement-layer component.

Claim 92. The system of claim 83, wherein the Pipeline's three QPCs are mapped to a Data Plane, a Control Plane, and a Management Plane of a Three-Plane Architecture, and wherein inter-stage attestations cross plane boundaries exclusively through Trust-Block-bound envelopes.

Whereby Clause (Family-Level Structural Effect)

Whereby Family C closes the Three-Stage AI Evaluation Pipeline gap, binding 2 independent claims and 25 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family D — Three-Ledger/Two-Log Architecture & Permanent Privacy Seal

Claims 93–116 (24 claims: 3 independent + 21 dependent)

Family Introduction

Family D captures the Three-Ledger / Two-Log Architecture (Contribution / Authorization / Settlement Ledgers + Data Plane / Control Plane Logs) and the Permanent Privacy Seal irrevocability primitive. Ledger decomposition is structurally integrated with the Three-Plane Architecture of Family A.

Claims List

Claim 93. A multi-ledger trust-verified record-keeping system, comprising:

a Contribution Ledger recording, for each authorized contribution, a Trust-Block-bound reference to a Verification Envelope and a Mode Tag, without recording any plaintext contribution content;

an Authorization Ledger recording, for each authorization decision, a Trust Block hash, a Trust Criteria identifier, and a Proof-of-Trust verification outcome;

a Settlement Ledger recording, for each settlement event, an Exchange Token issuance record bound to a Contribution Ledger reference and an Authorization Ledger reference;

a Data Plane Log recording transport-layer events within a Data Plane Quantum Privacy Cell set of a Three-Plane Architecture;

a Control Plane Log recording authorization-decision events within a Control Plane Quantum Privacy Cell set distinct from the Data Plane Quantum Privacy Cell set;

a Cross-Ledger Verifier component configured to verify, by Proof-of-Trust, that each Settlement Ledger record references a valid Authorization Ledger record and a valid Contribution Ledger record, and that no record set is mutated absent a Trust-Block-bound authorization;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby record-keeping functions are decomposed such that no single component can record settlement without independent contribution and authorization records, and the Three-Plane Architecture's plane separation is reflected in the log structure.

Claim 94. The system of claim 93, wherein the Contribution Ledger and the Authorization Ledger are physically separated such that no single host writes to both ledgers, and wherein write authority is bound by Trust Criteria.

Claim 95. The system of claim 93, wherein the Settlement Ledger records, in addition to the Exchange Token issuance, an Exchange Root allocation share and a Participant Pool allocation share computed pursuant to a settlement waterfall.

Claim 96. The system of claim 93, wherein the Data Plane Log records transport-layer events without recording payload contents, and wherein the Control Plane Log records authorization decisions including the Trust Criteria identifier evaluated.

Claim 97. The system of claim 93, wherein the Cross-Ledger Verifier is configured to detect, by Proof-of-Trust, any Settlement Ledger record whose Contribution Ledger reference or Authorization Ledger reference fails to verify.

Claim 98. The system of claim 93, wherein each ledger is structured as an append-only sequence whose write authority is bound by a respective Trust Block of a Settlement-enforcement-layer component.

Claim 99. The system of claim 93, wherein the Contribution Ledger reference to a Verification Envelope is a content-addressed identifier computed by a quantum-safe hash function.

Claim 100. The system of claim 93, wherein, upon detection of a verification failure by the Cross-Ledger Verifier, the affected Settlement Ledger record is withheld from subsequent reward computation pending a Trust-Block-bound dispute resolution.

Claim 101. The system of claim 93, wherein each ledger is replicated across a quorum of Quantum Privacy Cells operating under distinct Privacy Domains, and replication consistency is enforced by Proof-of-Trust.

Claim 102. The system of claim 93, wherein each ledger is configured to enable lineage queries returning a Trust-Block-bound reference graph without disclosing payload contents.

Claim 103. The system of claim 93, wherein the Cross-Ledger Verifier is operated by a Catalyst Network Manager pursuant to a Permitted Audit Purpose specified in a governing participation framework.

Claim 104. A Permanent Privacy Seal system, comprising:

a Seal Request Interface configured to receive, from an originating participant, a Permanent Privacy Seal assertion identifying a Contribution Ledger reference and bound to a Trust Block of the originating participant;

a Seal Verifier configured to verify, by Proof-of-Trust, that the Permanent Privacy Seal assertion is authorized under a Trust Criteria of the originating Privacy Domain;

a Key Destruction Component configured to, upon Seal Verifier approval, destroy the encryption key associated with a Vault-resident Witness Record identified by the Contribution Ledger reference, under control of Trust-Block-bound key destruction logic, such that no future decryption of the Vault-resident Witness Record is possible;

a Seal Ledger Recorder configured to record, in the Contribution Ledger, a Permanent Privacy Seal annotation referencing the destroyed key without disclosing the prior contents of the Witness Record;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a participant may render a contribution permanently undisclosable through cryptographic key destruction while preserving the Contribution Ledger's structural completeness with respect to settlement and audit references.

Claim 105. The system of claim 104, wherein the Permanent Privacy Seal assertion identifies a specific Witness Record by a content-addressed identifier and is bound to the originating participant's Trust Block by an explicit user-action signature.

Claim 106. The system of claim 104, wherein the Key Destruction Component is structurally incapable of recovering the destroyed key after destruction, and wherein the destruction operation is logged in the Control Plane Log with a Trust-Block-bound destruction attestation.

Claim 107. The system of claim 104, wherein the Permanent Privacy Seal annotation in the Contribution Ledger preserves a content-addressed identifier of the Witness Record while indicating the Seal state, such that Settlement Ledger references remain structurally valid.

Claim 108. The system of claim 104, wherein the Seal Verifier additionally verifies that no outstanding compliance hold prevents Sealing of the identified Witness Record under a Trust Block-bound hold policy.

Claim 109. The system of claim 104, wherein the Permanent Privacy Seal system is integrated with a Three-Stage AI Evaluation Pipeline of a contribution-evaluation system such that, upon Sealing, the Pipeline Coordinator destroys intermediate stage outputs while preserving the composite Pipeline Attestation's cryptographic structure.

Claim 110. The system of claim 104, wherein the Seal Verifier emits a Trust-Block-bound Seal Receipt to the originating participant evidencing successful key destruction.

Claim 111. A computer-implemented method for cross-ledger record-keeping in a Quantum Privacy Network (QPN), comprising:

upon receipt of a Verification Envelope at a Catalyst Network ingress, recording, in a Contribution Ledger, a Trust-Block-bound reference to the Verification Envelope and a Mode Tag;

upon completion of a Proof-of-Trust verification of the Verification Envelope, recording, in an Authorization Ledger, a Trust Block hash and a Proof-of-Trust outcome;

upon issuance of an Exchange Token by a Settlement Controller, recording, in a Settlement Ledger, an Exchange Token issuance record bound to the Contribution Ledger reference and the Authorization Ledger record;

concurrently recording, in a Data Plane Log, the transport-layer events of the Verification Envelope and, in a Control Plane Log, the authorization-decision events of the Trust Block emission;

verifying, by a Cross-Ledger Verifier, that each Settlement Ledger record references a valid Authorization Ledger record and a valid Contribution Ledger record before exposure of the Settlement Ledger record to a subsequent reward computation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby ledger record dependencies are structurally enforced and settlement records cannot reference unauthorized or undocumented contributions.

Claim 112. The system of claim 111, further comprising verifying, by the Cross-Ledger Verifier, the temporal ordering of the Contribution Ledger record, the Authorization Ledger record, and the Settlement Ledger record such that the Settlement Ledger record's timestamp is not earlier than either of the other two records.

Claim 113. The system of claim 111, further comprising, upon receipt of a Permanent Privacy Seal request, executing a key destruction operation as recited in the Permanent Privacy Seal system, and annotating the Contribution Ledger record without invalidating the corresponding Settlement Ledger record.

Claim 114. The system of claim 111, wherein the Data Plane Log and the Control Plane Log are stored in physically separated Quantum Privacy Cells whose access is mediated by plane-specific Trust Criteria.

Claim 115. The system of claim 111, further comprising emitting a Cross-Ledger Audit Bundle on demand of an authorized Catalyst Network Manager, the Cross-Ledger Audit Bundle comprising Trust-Block-bound references to the corresponding Contribution Ledger record, Authorization Ledger record, and Settlement Ledger record without disclosing payload contents.

Claim 116. The system of claim 111, wherein the Settlement Ledger records, in addition to the Exchange Token issuance, an Exchange Root allocation share and an Accelerator Incentive & Investment Pool allocation share where applicable.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family D closes the Three-Ledger/Two-Log Architecture **&** Permanent Privacy Seal gap, binding 3 independent claims and 21 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism*.

Family E — Global Contribution Graph & Reputation Engine

Claims 117–141 (25 claims: 3 independent + 22 dependent)

Family Introduction

Family E captures the Global Contribution Graph Assembly Engine, the DFS-traversal Reputation Engine, and the Three-Dimensional Quantum Reputation Model. The three members are functionally cohesive: the Reputation Engine consumes the Contribution Graph, and the Three-Dimensional Model is the Reputation Engine's output schema.

Claims List

Claim 117. A global contribution graph assembly system, comprising:

a plurality of Personal Archives, each residing within an originating participant's Privacy Domain and storing Trust-Block-bound references to Witness Records of contributions captured under a contribution capture system;

a Graph Assembly Engine executing within a Graph Quantum Privacy Cell (QPC) and configured to query each Personal Archive through an authenticated read interface bound to a Trust Block of the originating Privacy Domain and to assemble a global contribution graph from the returned references, without ingesting any plaintext contribution content;

a Graph Trust Block recording, for each global contribution graph snapshot, the set of Personal Archive references contributing to the snapshot and the Proof-of-Trust verification outcomes for each authenticated read interface invocation;

a Graph Query Interface configured to expose the global contribution graph to authorized consumer components under per-consumer Trust Criteria, and to return only Trust-Block-bound graph fragments authorized by the queried consumer's Trust Block;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a global contribution graph is constructed without centralization of contribution content, and queries against the graph cannot extract content outside the querying Trust Block's authorization scope.

Claim 118. The system of claim 117, wherein each Personal Archive is encrypted under keys held within the originating Privacy Domain and exposes content only as Trust-Block-bound references through the authenticated read interface.

Claim 119. The system of claim 117, wherein the Graph Assembly Engine assembles the global contribution graph as a directed acyclic graph whose nodes are content-addressed references to Witness Records and whose edges are Cross-Verification attestations.

Claim 120. The system of claim 117, wherein the Graph Trust Block additionally records a temporal-window descriptor scoping the snapshot to a specified time interval.

Claim 121. The system of claim 117, wherein the Graph Query Interface enforces per-consumer Trust Criteria that limit returned graph fragments to those whose constituent Trust Blocks intersect the consumer's authorization scope.

Claim 122. The system of claim 117, wherein the Graph Assembly Engine maintains an incremental update mechanism such that newly admitted contributions are integrated into the global contribution graph without re-querying every Personal Archive.

Claim 123. The system of claim 117, wherein each authenticated read interface invocation is logged in a Control Plane Log of a Three-Plane Architecture with a Proof-of-Trust outcome recorded against the invoking Trust Block.

Claim 124. The system of claim 117, wherein the global contribution graph is replicated across a quorum of Graph Quantum Privacy Cells operating under distinct Privacy Domains, and replication consistency is enforced by Proof-of-Trust.

Claim 125. The system of claim 117, wherein the Graph Assembly Engine is structurally incapable of decrypting Personal Archive contents.

Claim 126. A trust-verified reputation engine system, comprising:

a Reputation Engine executing within a Reputation Quantum Privacy Cell and configured to compute reputation scores by depth-first traversal of a global contribution graph;

a Three-Dimensional Reputation Schema comprising a Contribution Dimension, an Identity Dimension, and a Behavioral Dimension, each independently weighted under a Trust Block-specified weighting policy;

a Confidence Tier Classifier configured to assign each computed reputation score to one of at least five confidence tiers based on signal corroboration;

a Signal Type Registry enumerating at least seven reputation signal types, wherein each reputation signal type is bound to a respective Trust Criteria specifying admission rules;

a Reputation Trust Block emitter configured to bind each computed reputation score to a Trust Block recording the three dimension components, the confidence tier, and the signal-type set contributing to the computation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby reputation is computed deterministically over the global contribution graph and decomposed across three independently-attestable dimensions with confidence-tier classification.

Claim 127. The system of claim 126, wherein the Contribution Dimension is computed by aggregating Trust-Block-bound contribution counts weighted by a Premium-bearing field of each contribution's originating Trust Block.

Claim 128. The system of claim 126, wherein the Identity Dimension is computed from Multi-Factor Identity Binding artifacts and is increased upon additional independently-attested identity factors and decreased upon detected identity conflicts.

Claim 129. The system of claim 126, wherein the Behavioral Dimension is computed from a Behavioral Activation System's level, badge set, and streak multiplier records.

Claim 130. The system of claim 126, wherein the five or more confidence tiers comprise at least: a base tier, a corroborated tier, a multi-corroborated tier, a witnessed tier, and an authoritatively-witnessed tier, each tier requiring a respective minimum count of independently-attested signals.

Claim 131. The system of claim 126, wherein the at least seven reputation signal types include at least: a contribution-count signal, a corroboration signal, a witness signal, an identity-factor signal, a behavioral-activity signal, a cross-Privacy-Network signal, and a settlement-frequency signal.

Claim 132. The system of claim 126, wherein the Reputation Engine's depth-first traversal is bounded by a Trust-Block-specified maximum depth and a Trust-Block-specified maximum visited-node count.

Claim 133. The system of claim 126, wherein each Reputation Trust Block is deterministically reproducible by a verifier holding the same global contribution graph snapshot and the same weighting policy.

Claim 134. The system of claim 126, wherein the Reputation Trust Block additionally records a reputation-badge set selected from at least six reputation badges, each badge specifying a Trust Block-bound award condition.

Claim 135. The system of claim 126, wherein the Reputation Engine emits reputation scores under a Privacy-Preserving Differential mechanism such that no participant's individual reputation score is revealed to an unauthorized querier.

Claim 136. A computer-implemented method for global contribution graph assembly and reputation computation within a Quantum Privacy Network (QPN), comprising:

querying, by a Graph Assembly Engine executing within a Graph Quantum Privacy Cell, each of a plurality of Personal Archives through an authenticated read interface bound to a Trust Block of the originating Privacy Domain;

assembling, from references returned by the authenticated read interfaces, a global contribution graph without ingesting any plaintext contribution content;

computing, by a Reputation Engine, reputation scores by depth-first traversal of the global contribution graph and decomposing the reputation into a Contribution Dimension, an Identity Dimension, and a Behavioral Dimension;

assigning each computed reputation score to one of at least five confidence tiers;

emitting Reputation Trust Blocks binding each reputation score, three-dimension components, and confidence tier;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby reputation computation preserves Personal Archive locality of contribution content and produces Trust-Block-bound, decomposed reputation scores.

Claim 137. The system of claim 136, wherein computing reputation scores includes traversing the global contribution graph by depth-first search up to a Trust-Block-specified maximum depth.

Claim 138. The system of claim 136, further comprising emitting, for each computed reputation score, a deterministic-replay descriptor enabling third-party reproduction of the reputation computation.

Claim 139. The system of claim 136, wherein each authenticated read interface invocation returns no plaintext contribution content and returns only Trust-Block-bound references and metadata.

Claim 140. The system of claim 136, wherein the Behavioral Dimension's input includes records from a Behavioral Activation System comprising at least ten activity levels.

Claim 141. The system of claim 136, further comprising periodically refreshing the global contribution graph snapshot under a Trust-Block-specified refresh policy and emitting a corresponding Graph Trust Block.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family E closes the Global Contribution Graph **&** Reputation Engine gap, binding 3 independent claims and 22 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism*.

Family F — Personal Archive, CCP & Multi-Substrate Persistence

Claims 142–160 (19 claims: 2 independent + 17 dependent)

Family Introduction

Family F captures the Personal Archive (.qpn) file format primitive, the Contributor Intelligence Module (CCP) local analytics engine, and the Multi-Substrate Persistence Architecture (Persistence Router + Adapter pattern). The three are co-located at the participant-local storage layer.

Claims List

Claim 142. A participant-local trust-verified contribution archive system, comprising:

a Personal Archive comprising a structured file conforming to a content-addressed archive format and storing Trust-Block-bound references to Witness Records, the Personal Archive being encrypted under a Privacy Domain key whose private component does not leave an originating Privacy Domain in plaintext;

a Contributor Intelligence Module (CCP) executing locally within a CCP Quantum Privacy Cell of the originating Privacy Domain and configured to perform analytics over the Personal Archive without externalizing any plaintext contribution content;

a Persistence Router executing within a Router Quantum Privacy Cell and configured to select, on a per-Archive-segment basis, a persistence substrate from a plurality of registered persistence substrates;

a plurality of Persistence Adapters, each Adapter implementing a substrate-specific write interface and a substrate-specific read interface, and each Adapter being bound to a Trust Block whose Trust Criteria specifies substrate-acceptable cryptographic properties;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby contribution records are stored locally and analyzed locally under cryptographic boundaries, while persistence is selectable across multiple substrates without exposing plaintext contribution content to any substrate operator.

Claim 143. The system of claim 142, wherein the Personal Archive's content-addressed archive format specifies a manifest header recording Trust Block lineage and Privacy Domain identifiers, and a body comprising encrypted segments addressed by quantum-safe hash.

Claim 144. The system of claim 142, wherein the Contributor Intelligence Module executes analytics queries over the Personal Archive and emits aggregated results that are Trust-Block-bound and that do not contain plaintext contribution content.

Claim 145. The system of claim 142, wherein each Persistence Adapter is bound to a substrate selected from at least: a local disk substrate, a private cloud substrate, an inter-planetary file system substrate, a federated peer substrate, and an organizational private-cloud substrate.

Claim 146. The system of claim 142, wherein the Persistence Router selects a substrate per segment based on Trust Block-specified policies including at least: a confidentiality-required policy, an availability-required policy, and a cost-bounded policy.

Claim 147. The system of claim 142, wherein the Persistence Router maintains a substrate availability map under quantum-safe attestations of each substrate's accessibility and rotates segments across substrates upon detected unavailability.

Claim 148. The system of claim 142, wherein, upon a Permanent Privacy Seal assertion, each Persistence Adapter is instructed by Trust Block-bound key destruction logic to render its locally-stored ciphertext segments undecryptable, and to confirm destruction by an attestation logged in a Control Plane Log.

Claim 149. The system of claim 142, wherein the Personal Archive format supports incremental append such that new contributions add new content-addressed segments without rewriting prior segments.

Claim 150. The system of claim 142, wherein the CCP supports a query language exposing Trust-Block-bound aggregates including at least: contribution counts by category, temporal histograms, and reputation-dimension summaries.

Claim 151. The system of claim 142, wherein the system is integrated with a Sidecar contribution capture system such that the Sidecar's Local Vault is implemented by the Personal Archive.

Claim 152. The system of claim 142, wherein each Persistence Adapter is configured to refuse write operations whose Trust Block does not satisfy the Adapter's substrate-acceptable cryptographic-property Trust Criteria.

Claim 153. A computer-implemented method for multi-substrate persistence of a trust-verified Personal Archive in a Quantum Privacy Network (QPN), comprising:

writing, by a Persistence Router executing within a Router Quantum Privacy Cell, each segment of a Personal Archive to a substrate-specific Persistence Adapter selected from a plurality of Persistence Adapters;

binding the write operation to a Trust Block of the originating Privacy Domain such that the substrate operator receives only ciphertext segments and does not receive Privacy Domain keys;

upon read, retrieving the ciphertext segments by the selected Persistence Adapter, decrypting the segments within the Router Quantum Privacy Cell using Privacy Domain keys held within the originating Privacy Domain, and re-constructing the Personal Archive locally;

executing, by a Contributor Intelligence Module within a CCP Quantum Privacy Cell of the originating Privacy Domain, analytics over the reconstructed Personal Archive without externalization of plaintext contribution content;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the Personal Archive is substrate-portable while preserving Privacy Domain locality of plaintext contribution content and local-only analytics.

Claim 154. The system of claim 153, further comprising recording, in a Data Plane Log, each write operation's substrate selection and resulting Trust-Block-bound attestation, without recording the ciphertext segment contents.

Claim 155. The system of claim 153, further comprising verifying, upon each read, that the retrieved ciphertext segment's content-addressed identifier matches the manifest header's reference for the requested Personal Archive segment.

Claim 156. The system of claim 153, further comprising rotating, by the Persistence Router, ciphertext segments across substrates pursuant to a Trust-Block-specified rotation policy without requiring re-encryption.

Claim 157. The system of claim 153, wherein the analytics performed by the Contributor Intelligence Module include emission of Trust-Block-bound reputation inputs that are subsequently consumed by a Reputation Engine.

Claim 158. The system of claim 153, wherein, upon a Permanent Privacy Seal request, the method further comprises destroying the encryption keys for the affected Personal Archive segments and emitting a Seal Receipt evidencing destruction.

Claim 159. The system of claim 153, wherein each Persistence Adapter is operated under a substrate-operator Trust Block, and wherein the substrate-operator Trust Block's Trust Criteria do not include any authority to decrypt the stored ciphertext.

Claim 160. The system of claim 153, wherein the originating Privacy Domain key is rotated under a Trust Block-specified rotation policy and prior ciphertext segments are re-keyed by the Persistence Router without externalization of the prior key.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family F closes the Personal Archive, CCP **&** Multi-Substrate Persistence gap, binding 2 independent claims and 17 dependent claims to the foundational QPN primitives via the §22.7 canonical **Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism**.*

Family G — Settlement Controller & Cross-Verification Protocol

Claims 161–184 (24 claims: 3 independent + 21 dependent)

Family Introduction

Family G captures the Settlement Controller (deterministic Topic-9 non-bypassability enforcement) and the Cross-Verification Protocol (witness-based contribution authenticity attestation). The two members are co-located at the settlement enforcement layer; the Cross-Verification Protocol feeds the Settlement Controller's eligibility check.

Claims List

Claim 161. A trust-verified settlement enforcement system, comprising:

a Settlement Controller executing within a Controller Quantum Privacy Cell and configured to detect each authorized cross-party reuse of a Quantum Privacy Resource and to deterministically issue Exchange Tokens upon detection;

an Eligibility Verifier configured to verify, prior to Exchange Token issuance, at least: the presence of a valid Authorization Ledger record, the presence of a valid Contribution Ledger record, and the presence of a Cross-Verification attestation associated with the reuse event;

a Token Allocation Calculator configured to compute, for each authorized cross-party reuse event, an Exchange Root allocation share fixed at a system-specified percentage of total settled value and a Participant Pool allocation share comprising the remainder, and configured to inherit Premium-bearing fields of upstream Trust Blocks into the issued Exchange Token records;

a Settlement Trust Block emitter configured to record each issuance event in a Settlement Ledger with cross-references to the Authorization Ledger record and the Contribution Ledger record;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Exchange Token issuance is deterministically triggered upon authorized cross-party reuse and is structurally incapable of bypass within the compliant operational perimeter.

Claim 162. The system of claim 161, wherein the system-specified percentage of total settled value comprising the Exchange Root allocation share is fixed at 7.5%.

Claim 163. The system of claim 161, wherein the Settlement Controller is structurally incapable of issuing Exchange Tokens absent valid Authorization Ledger record, Contribution Ledger record, and Cross-Verification attestation.

Claim 164. The system of claim 161, wherein the Token Allocation Calculator applies a maturity-stage waterfall such that the Participant Pool allocation is further split between an Accelerator Incentive & Investment Pool share and an Accelerator Participant Pool share where the cross-party reuse occurs within an Accelerator scope.

Claim 165. The system of claim 161, wherein the Token Allocation Calculator inherits Premium-bearing fields of upstream Trust Blocks into the issued Exchange Token records pursuant to a Premium Inheritance via Trust Block Chain mechanism.

Claim 166. The system of claim 161, wherein the Settlement Controller emits an issuance-rejection attestation upon failure of the Eligibility Verifier, the issuance-rejection attestation being recorded in the Authorization Ledger and being structurally distinguishable from a successful issuance record.

Claim 167. The system of claim 161, wherein the Settlement Controller is operated under a Settlement Trust Block whose Trust Criteria prohibits discretionary withholding of Exchange Token issuance upon Eligibility Verifier success.

Claim 168. The system of claim 161, wherein the Settlement Controller's detection logic identifies an authorized cross-party reuse event as occurring when a Quantum Privacy Resource is accessed by a participant Trust Block other than the resource's originating participant Trust Block under a valid Trust Block-bound access grant.

Claim 169. The system of claim 161, wherein the system enforces a deferred-activation invariant such that issued Exchange Tokens directed to a DORMANT participant record have no economic function until the participant self-activates pursuant to a separate participation framework.

Claim 170. The system of claim 161, wherein each issuance is deterministically reproducible by an independent verifier holding the same Authorization Ledger, Contribution Ledger, and Cross-Verification Record set.

Claim 171. The system of claim 161, wherein the Settlement Controller emits a Trust-Block-bound settlement receipt to each settlement counterparty evidencing successful issuance.

Claim 172. A cross-verification protocol system, comprising:

a Witness Registration Interface configured to register, in a witness Trust Block, an attesting participant's authorization to provide cross-verification attestations under a specified Trust Criteria;

a Witness Attestation Emitter configured to receive, from a registered witness, an attestation of a third-party contribution's authenticity without disclosure of the third-party contribution's plaintext content, and to emit the attestation as a Trust-Block-bound Cross-Verification Record;

an Attestation Aggregator configured to collect Cross-Verification Records associated with a specified Contribution Ledger reference and to compute a quorum-based authenticity indicator;

a Settlement Controller interface exposing the quorum-based authenticity indicator to a Settlement Controller's Eligibility Verifier as an input to settlement eligibility determination;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby third-party attestation of contribution authenticity is admissible to the settlement system without disclosure of contribution payload contents.

Claim 173. The system of claim 172, wherein the Witness Attestation Emitter accepts witness attestations bound by the witness's Trust Block and rejects attestations whose witness Trust Block has not been registered.

Claim 174. The system of claim 172, wherein the quorum-based authenticity indicator is computed by an aggregation function that yields a positive authenticity indicator only upon receipt of a Trust-Block-specified minimum count of independent witness attestations.

Claim 175. The system of claim 172, wherein each Cross-Verification Record carries a Trust-Block-bound temporal binding, and the Attestation Aggregator discards Cross-Verification Records whose temporal binding has exceeded a Trust-Block-specified expiration.

Claim 176. The system of claim 172, wherein the Witness Registration Interface validates that an attesting participant's Trust Block is authorized to provide attestations of the specified contribution category pursuant to a Trust Criteria.

Claim 177. The system of claim 172, wherein the Attestation Aggregator emits a Trust-Block-bound aggregated authenticity Trust Block, which is recorded in an Authorization Ledger and which is consumable by the Settlement Controller's Eligibility Verifier.

Claim 178. The system of claim 172, wherein, where the third-party contribution is bound to a Manager-Originated DORMANT Quantum Privacy Cell record, the Cross-Verification Protocol additionally requires that the witness Trust Block carries no economic interest in the DORMANT record.

Claim 179. A computer-implemented method for deterministic settlement enforcement in a Quantum Privacy Network (QPN), comprising:

detecting, by a Settlement Controller executing within a Controller Quantum Privacy Cell, an authorized cross-party reuse of a Quantum Privacy Resource;

verifying, by an Eligibility Verifier, the presence of a valid Authorization Ledger record, the presence of a valid Contribution Ledger record, and the presence of a Cross-Verification attestation associated with the reuse event;

computing, by a Token Allocation Calculator, an Exchange Root allocation share fixed at a system-specified percentage of total settled value and a Participant Pool allocation share comprising the remainder;

issuing Exchange Tokens deterministically pursuant to the computed allocation and recording the issuance in a Settlement Ledger with cross-references to the Authorization Ledger and Contribution Ledger records;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Exchange Token issuance is enforced deterministically upon authorized cross-party reuse with structural non-bypassability within the compliant operational perimeter.

Claim 180. The system of claim 179, wherein detecting an authorized cross-party reuse comprises observing a Trust Block-bound access grant invoking a Resource Trust Block of an originating participant.

Claim 181. The system of claim 179, further comprising emitting an issuance-rejection attestation upon failure of Eligibility Verifier, the issuance-rejection attestation being recorded in the Authorization Ledger.

Claim 182. The system of claim 179, further comprising inheriting Premium-bearing fields of upstream Trust Blocks into the issued Exchange Token records pursuant to a Premium Inheritance mechanism.

Claim 183. The system of claim 179, wherein the Exchange Root allocation share is fixed at a percentage of total settled value such that the Exchange Root allocation is computed before the Participant Pool allocation in the settlement waterfall.

Claim 184. The system of claim 179, further comprising recording, in a Settlement Trust Block bound to the issuance record, a content-addressed reference to the Cross-Verification quorum-based authenticity indicator.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family G closes the Settlement Controller **&** Cross-Verification Protocol gap, binding 3 independent claims and 21 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.*

Family H — Specialized Catalyst Vectors (Web/Voice/Code/Doc/Agent/Comm/Meet/Message)

Claims 185–225 (41 claims: 8 independent + 33 dependent)

Family Introduction

Family H captures eight specialized Catalyst Vectors implementing the Family A Sidecar pattern across distinct contribution domains: WebVector, VoiceVector, CodeVector, DocVector, AgentVector, CommVector, MeetVector, MessageVector. Each Vector is a domain-specific specialization adding domain-particular capture, attestation, and authorization primitives.

Claims List

Claim 185. A WebVector contribution capture system, comprising:

a browser-resident WebVector Sidecar component implementing a Witness Agent, Listener Agent, Local Vault, and Verification Bridge configured to operate within a respective Quantum Privacy Cell (QPC);

an Origin Verifier configured to bind each captured web-page interaction to a verified browser origin under an extension-manifest pinning;

a Page-Context Attestor configured to emit, for each captured web-page interaction, a Trust-Block-bound page-context attestation comprising at least a content-addressed page identifier, a temporal binding, and a user-agent fingerprint;

a Trust-Block Signing Engine configured to sign Verification Envelopes under the participant's Trust Block keys for transmission to a Catalyst Network ingress;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby web-page interactions are captured with browser-origin verification and Trust-Block-bound page-context attestation, while content extraction beyond the captured interaction is structurally prevented.

Claim 186. The system of claim 185, wherein the Origin Verifier additionally verifies, against a Trust-Block-specified origin allowlist, that the browser origin of the captured web-page interaction is authorized.

Claim 187. The system of claim 185, wherein the Page-Context Attestor's user-agent fingerprint is computed by a deterministic hash whose inputs include browser version, extension version, and rendering-engine identifier.

Claim 188. The system of claim 185, wherein the WebVector Sidecar is implemented as a browser extension whose manifest pins each Quantum Privacy Cell to a specified QPN origin.

Claim 189. The system of claim 185, wherein the Trust-Block Signing Engine signs Verification Envelopes using quantum-safe digital signatures whose verification keys are recorded in the participant's Trust Block.

Claim 190. The system of claim 185, wherein the WebVector Sidecar enforces a content-scope constraint such that only user-originated input events within an explicitly-designated input scope are captured.

Claim 191. A VoiceVector contribution capture system, comprising:

an on-device VoiceVector Sidecar component implementing a Witness Agent and a Local Vault configured to operate within a respective Quantum Privacy Cell of a participant's device, wherein raw audio data does not leave the participant's device in plaintext;

an On-Device Transcription Module configured to convert authorized voice activations to a Trust-Block-bound transcript and to bind the transcript to a content-addressed identifier of the raw audio that remains on the participant's device;

a Voice-Activation Authorizer configured to admit voice activations only upon detection of an explicit user wake-word or wake-gesture authorized under a Trust Criteria;

a Verification Bridge configured to construct Verification Envelopes referencing the Trust-Block-bound transcript without exporting the raw audio;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby voice-originated contributions are captured under on-device privacy preservation with cryptographic attestation linking the transcript to the originating audio without exporting the audio.

Claim 192. The system of claim 191, wherein the On-Device Transcription Module's transcript is computed by a model whose snapshot is identified by a content-addressed identifier recorded in the Trust Block.

Claim 193. The system of claim 191, wherein the Voice-Activation Authorizer admits voice activations only upon detection of a Trust-Block-authorized wake-word.

Claim 194. The system of claim 191, wherein the raw audio is stored within a hardware-isolated cryptographic enclave of the participant's device and is destroyed upon a Trust-Block-bound retention-window expiration.

Claim 195. The system of claim 191, wherein the VoiceVector Sidecar's Trust Criteria prohibits transmission of raw audio across any network interface.

Claim 196. A CodeVector contribution capture system, comprising:

a CodeVector Sidecar component bound to an integrated development environment (IDE) and configured to capture code-authoring events including at least: source edits at file-and-line granularity, commit events, and pull-request authorship events;

a Pull-Request-Attribution Component configured to bind each captured pull-request authorship event to a Trust-Block-bound participant identifier and to a content-addressed reference to a target repository;

an Authorship-Lineage Tracker configured to record, for each captured code-authoring event, a lineage of upstream Trust-Block-bound code-authoring events on which the event derives;

a Verification Bridge configured to construct Verification Envelopes for code-authoring events and transmit them to a Catalyst Network ingress;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby code-authoring contributions are captured with pull-request-level attribution and authorship-lineage tracking, enabling derivative attribution downstream.

Claim 197. The system of claim 196, wherein the CodeVector Sidecar binds each captured code-authoring event to a content-addressed identifier of the target source file and a line-range descriptor.

Claim 198. The system of claim 196, wherein the Pull-Request-Attribution Component identifies upstream commit Trust Blocks that the pull request derives from and records the upstream lineage in the Authorship-Lineage Tracker.

Claim 199. The system of claim 196, wherein the Authorship-Lineage Tracker emits, for each captured pull-request authorship event, a Trust-Block-bound derivative-attribution record subsequently consumed by a Settlement Controller for derivative reward computation.

Claim 200. The system of claim 196, wherein the CodeVector Sidecar additionally captures code-review events authored by a reviewing participant and binds each code-review event to the reviewing participant's Trust Block.

Claim 201. The system of claim 196, wherein the CodeVector Sidecar additionally captures continuous-integration test-pass events and binds each test-pass event to a Trust-Block-bound test descriptor.

Claim 202. A DocVector contribution capture system with native Privacy Network signing, comprising:

a DocVector Sidecar component bound to a document-authoring application and configured to capture document-edit events at section-level granularity;

a Section-Level Trust-Block Binder configured to bind each section-level document-edit event to a Trust Block of the originating participant and the originating section's section identifier;

a Verifiable Credential Issuer configured to issue W3C Verifiable Credentials bound to each Trust Block, the Verifiable Credentials being suitable for attestation by a Catalyst Network ingress and by third-party verifiers;

a Verification Bridge configured to transmit Trust-Block-bound section-level edit events and associated Verifiable Credentials to a Catalyst Network ingress;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby document-authoring contributions are captured at section-level granularity with W3C Verifiable Credential attestation suitable for both QPN and external verifiers.

Claim 203. The system of claim 202, wherein the Section-Level Trust-Block Binder additionally records a parent-section reference such that nested section structure is reflected in the Trust Block lineage.

Claim 204. The system of claim 202, wherein the Verifiable Credential Issuer issues W3C Verifiable Credentials conforming to the W3C Verifiable Credentials data model and signed under quantum-safe digital signatures.

Claim 205. The system of claim 202, wherein the DocVector Sidecar additionally captures co-author events for documents authored by multiple participants and binds each section-level edit event to all co-authoring participants' Trust Blocks pursuant to a co-authorship Trust Block.

Claim 206. The system of claim 202, wherein the DocVector Sidecar additionally captures document version events and emits a Trust-Block-bound version attestation referencing the prior version's content-addressed identifier.

Claim 207. An AgentVector contribution capture system with Proof of Orchestration, comprising:

an AgentVector Sidecar component bound to an agentic AI orchestration runtime and configured to observe orchestration events including at least: tool invocations, sub-agent delegations, and result aggregations;

a Proof-of-Orchestration Attestor configured to emit, for each observed orchestration sequence, a Trust-Block-bound Proof of Orchestration attestation comprising a deterministic-replay descriptor sufficient to reconstruct the orchestration;

a Tool-Authorization Verifier configured to verify, prior to admission of each orchestration event, that each invoked tool's Trust Criteria authorizes the participant's Trust Block to invoke the tool;

a Verification Bridge configured to transmit Proof of Orchestration attestations to a Catalyst Network ingress for trust-verified ingestion;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby agentic AI orchestrations are captured with deterministic-replay attestation and tool-level authorization verification.

Claim 208. The system of claim 207, wherein the Proof-of-Orchestration Attestor records, for each orchestration event, a Trust-Block-bound execution trace sufficient to deterministically reconstruct the orchestration's tool-invocation sequence.

Claim 209. The system of claim 207, wherein the Tool-Authorization Verifier rejects orchestration events whose invoked tool's Trust Criteria does not authorize the participant's Trust Block, and emits a Trust-Block-bound rejection attestation.

Claim 210. The system of claim 207, wherein the AgentVector Sidecar additionally captures sub-agent delegation events and binds each delegation event to a sub-agent Trust Block.

Claim 211. The system of claim 207, wherein the Proof-of-Orchestration attestation incorporates a deterministic-replay descriptor permitting third-party verification that the orchestration outputs are byte-identical upon replay with identical inputs.

Claim 212. The system of claim 207, wherein the AgentVector Sidecar is integrated with a Three-Stage AI Evaluation Pipeline such that orchestration attestations are consumed by the Pipeline's Semantic Classification Stage.

Claim 213. A CommVector contribution capture system with verified engagement loop, comprising:

a CommVector Sidecar component bound to a communication endpoint and configured to capture outbound and inbound communication events at the message granularity;

a Message-ID Matcher configured to bind paired outbound and inbound communications by a Trust-Block-bound message identifier such that a verified engagement loop is recorded only upon receipt of a matching inbound communication referencing the originating outbound message identifier;

an Engagement Loop Attestor configured to emit, for each closed loop, a Trust-Block-bound engagement attestation comprising the originating message identifier, the responsive message identifier, and a temporal binding;

a Verification Bridge configured to transmit engagement attestations to a Catalyst Network ingress;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby communication-based contributions are credited only upon verified bidirectional engagement evidenced by Trust-Block-bound message identifier matching.

Claim 214. The system of claim 213, wherein the Message-ID Matcher additionally enforces a Trust-Block-specified temporal window such that an inbound communication arriving after the temporal window's expiration does not close the engagement loop.

Claim 215. The system of claim 213, wherein the Engagement Loop Attestor additionally records a content-addressed reference to the outbound and inbound messages without including message contents.

Claim 216. The system of claim 213, wherein the CommVector Sidecar additionally captures communication-channel attestations identifying the communication channel under a Trust Block.

Claim 217. A MeetVector contribution capture system, comprising:

a MeetVector Sidecar component bound to a meeting attendance endpoint and configured to capture meeting attendance events including at least: join events, leave events, and active-participation indicators;

a Meeting-Identity Binder configured to bind each meeting attendance event to a Trust Block of the originating participant and a content-addressed reference to the meeting's Trust-Block-bound identifier;

an Active-Participation Attestor configured to emit, for each capture instance, a Trust-Block-bound attestation that the participant was actively engaged for a Trust-Block-specified minimum duration;

a Verification Bridge configured to transmit attendance and engagement attestations to a Catalyst Network ingress;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby meeting attendance contributions are credited only upon Trust-Block-attested active participation.

Claim 218. The system of claim 217, wherein the Active-Participation Attestor's active-participation indicator includes at least one of: audio-presence indicator, video-presence indicator, and text-channel participation indicator.

Claim 219. The system of claim 217, wherein the Meeting-Identity Binder records the meeting's Trust-Block-bound identifier as a content-addressed hash of meeting metadata authorized under a Meeting-Host Trust Block.

Claim 220. The system of claim 217, wherein the MeetVector Sidecar emits, in addition to attendance attestations, presenter attestations bound to a Presenter Trust Block.

Claim 221. A MessageVector contribution capture system with privacy-preserving proxy network, comprising:

a MessageVector Sidecar component bound to a messaging endpoint and configured to capture message authorship events without exporting message payload content;

a Privacy-Preserving Proxy Network comprising at least two relay nodes, each operating under a Trust Block whose Trust Criteria prohibits decryption of relayed messages and prohibits linkage of relayed message identifiers to participant identifiers;

a Content-Addressed Message Identifier Emitter configured to compute a quantum-safe hash of each captured message authorship event and to emit the resulting message identifier to a Catalyst Network ingress through the Privacy-Preserving Proxy Network;

a Verification Bridge configured to construct and transmit Verification Envelopes referencing the content-addressed message identifiers;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby messaging contributions are captured with proxy-network unlinkability and content-addressed referencing, while no relay node can correlate message identifiers to participant identifiers.

Claim 222. The system of claim 221, wherein the Privacy-Preserving Proxy Network comprises at least three relay nodes operating under distinct Trust Blocks, and wherein each relay node enforces unlinkability between inbound and outbound transmissions.

Claim 223. The system of claim 221, wherein the Content-Addressed Message Identifier Emitter computes the message identifier under a quantum-safe hash function specified in a Trust Criteria.

Claim 224. The system of claim 221, wherein the MessageVector Sidecar additionally records, in a participant's Personal Archive, a Trust-Block-bound reference to each message identifier emitted to the Catalyst Network ingress.

Claim 225. The system of claim 221, wherein the Privacy-Preserving Proxy Network's relay nodes are selected per message under a Trust-Block-specified routing policy that randomizes relay selection.

Whereby Clause (Family-Level Structural Effect)

Whereby Family H closes the Specialized Catalyst Vectors (Web/Voice/Code/Doc/Agent/Comm/Meet/Message) gap, binding 8 independent claims and 33 dependent claims to the foundational QPN primitives via the §22.7

canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family I — Governed Agent Loop & Phase-0 DORMANT QPC State

Claims 226–254 (29 claims: 3 independent + 26 dependent)

Family Introduction

Family I captures the Governed Agent Loop (seven-step Sense → Interpret → Propose → Authorize → Execute → Verify → Learn) and the Phase-0 Deferred-Activation DORMANT QPC Record State. The two members are structurally integrated: the Loop's Authorize step gates against DORMANT-QPC presence, and DORMANT-state activation interacts with the Loop's Verify step.

Claims List

Claim 226. A trust-verified governed agent execution loop system, comprising:

a Sense Component executing within a Sense Quantum Privacy Cell (QPC) and configured to receive environmental observations and Trust-Block-bound participant inputs;

an Interpret Component executing within an Interpret QPC distinct from the Sense QPC and configured to construct, from received inputs, a Trust-Block-bound situational model;

a Propose Component executing within a Propose QPC distinct from the Sense and Interpret QPCs and configured to enumerate candidate actions and to attach a candidate-action Trust Block to each;

an Authorize Component executing within an Authorize QPC distinct from the foregoing QPCs and configured to apply, to each candidate action, a Trust Criteria check and to reject any candidate action whose Trust Criteria is not satisfied or whose participant target Trust Block carries a DORMANT-record marker absent additional activation conditions;

an Execute Component executing within an Execute QPC distinct from the foregoing QPCs and configured to perform authorized actions under a Trust-Block-bound execution envelope;

a Verify Component executing within a Verify QPC distinct from the foregoing QPCs and configured to verify, post-execution, that each executed action's effects conform to the originating Trust Block constraints;

a Learn Component executing within a Learn QPC distinct from the foregoing QPCs and configured to incorporate verified outcomes into a participant Trust Block-bound learning state;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby agentic actions are executed under per-step cryptographic authorization, post-hoc verification, and Trust-Block-bound learning, with structural prevention of unauthorized action emission.

Claim 227. The system of claim 226, wherein the Authorize Component's Trust Criteria check additionally enforces a Three-Stage AI Evaluation Pipeline's composite Pipeline Attestation as a necessary input to authorization of any candidate action that materially affects a participant Trust Block.

Claim 228. The system of claim 226, wherein the Sense Component receives environmental observations through a Sidecar contribution capture system's Verification Bridge, and rejects observations that lack a valid Verification Envelope.

Claim 229. The system of claim 226, wherein the Interpret Component is structurally incapable of emitting actions and is restricted to emitting situational models.

Claim 230. The system of claim 226, wherein the Propose Component emits each candidate action with a deterministic-replay descriptor sufficient to deterministically reconstruct the candidate-action generation.

Claim 231. The system of claim 226, wherein the Authorize Component rejects candidate actions whose target Trust Block is marked DORMANT, except where the candidate action constitutes a Manager-Originated record contribution-attribution write authorized by a Catalyst Network Manager Trust Block.

Claim 232. The system of claim 226, wherein the Execute Component records, for each executed action, a Trust-Block-bound execution receipt linking the executed action to the authorizing candidate-action Trust Block and the originating Verify-bound expectation.

Claim 233. The system of claim 226, wherein the Verify Component emits a verification non-conformance attestation upon detected deviation from the originating Trust Block constraints and triggers a Trust-Block-bound corrective action.

Claim 234. The system of claim 226, wherein the Learn Component is configured to incorporate verified outcomes into the participant's Quantum DNA pursuant to a Lamarckian inheritance mechanism, with bounded incorporation governed by an Adaptive Premium Compensation Proportionality constraint.

Claim 235. The system of claim 226, wherein each Component executes within a respective Quantum Privacy Cell whose cryptographic boundary prohibits cross-Component plaintext access except through Trust-Block-bound envelopes.

Claim 236. The system of claim 226, wherein the Loop is integrated with a Three-Plane Architecture such that the Sense and Interpret Components operate within Data Plane QPCs, the Authorize Component operates within a Control Plane QPC, and the Execute, Verify, and Learn Components operate within designated plane QPCs pursuant to plane-specific Trust Criteria.

Claim 237. The system of claim 226, wherein each Component is replaceable under a Management Plane Trust Block, and replacement requires emission of a Component-Provenance attestation recorded in a Control Plane Log.

Claim 238. The system of claim 226, wherein the Loop's Authorize Component additionally enforces a Permanent Privacy Seal check such that any candidate action affecting a Sealed contribution is rejected.

Claim 239. A deferred-activation dormant participant record system, comprising:

a Record Creator Component executing within a Creator Quantum Privacy Cell and configured to instantiate a Quantum Privacy Cell record bound to a participant identifier without the participant's knowledge or contemporaneous authorization, the record being marked with a DORMANT state indicator;

a DORMANT State Enforcer configured to prohibit, while the record is in the DORMANT state, any economic function of the record, including any market value, any transferability, and any ability to issue Exchange Tokens to the participant;

an Activation Interface configured to receive, from the participant, an explicit activation invocation accompanied by Trust-Block-bound authorization;

an Activation Verifier configured to, prior to state transition from DORMANT to ACTIVE, execute a Privacy-Preserving Compliance Screening against a Trust-Block-specified screening policy and to record the screening outcome in an Authorization Ledger;

a State Transition Recorder configured to record, upon successful activation, the state transition in an Authorization Ledger with a Trust-Block-bound transition attestation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby contribution attribution may be recorded for participants without their contemporaneous knowledge under cryptographic guarantee that no economic function attaches absent the participant's explicit activation and successful Privacy-Preserving Compliance Screening.

Claim 240. The system of claim 239, wherein the Record Creator Component records, in a Manager-Originated Provenance Ledger, an attestation identifying the Catalyst Network Manager responsible for the record's creation.

Claim 241. The system of claim 239, wherein the DORMANT State Enforcer is structurally incapable of conveying economic value to the participant absent successful activation, and any attempt to issue Exchange Tokens to a DORMANT-marked participant Trust Block is rejected at the Settlement Controller boundary.

Claim 242. The system of claim 239, wherein the Activation Verifier's Privacy-Preserving Compliance Screening includes at least: an identity-confirmation step, a sanctions-screening step, and an authorization-eligibility step under a Trust-Block-specified screening policy.

Claim 243. The system of claim 239, wherein the State Transition Recorder records, upon successful activation, the prior DORMANT state, the new ACTIVE state, the activation timestamp, and a Trust-Block-bound screening-outcome attestation.

Claim 244. The system of claim 239, wherein the system additionally maintains a Public Benefit Distribution Reserve allocation rule such that Exchange Tokens accrued to a DORMANT record whose participant declines activation or fails Privacy-Preserving Compliance Screening are redirected to a Public Benefit Distribution Reserve pursuant to a Trust-Block-bound redirection policy.

Claim 245. The system of claim 239, wherein the Activation Interface enforces an explicit user-action signature requirement such that no DORMANT record may transition to ACTIVE absent a user-action signature bound to the participant's Trust Block.

Claim 246. The system of claim 239, wherein the Privacy-Preserving Compliance Screening is performed within a Screening Quantum Privacy Cell whose cryptographic boundary prohibits disclosure of screening inputs and outputs outside the Screening QPC except as Trust-Block-bound attestations.

Claim 247. The system of claim 239, wherein the DORMANT State Enforcer is integrated with a Governed Agent Loop's Authorize Component such that no authorized action of the Governed Agent Loop may transfer value to a DORMANT-marked participant Trust Block.

Claim 248. The system of claim 239, wherein the Record Creator Component is operated under a Catalyst Network Manager Trust Block whose Trust Criteria includes a Permitted Audit Purpose authorization for the record's creation.

Claim 249. A computer-implemented method for trust-verified governed agent execution within a Quantum Privacy Network (QPN), comprising:

receiving, by a Sense Component executing within a Sense Quantum Privacy Cell, environmental observations and Trust-Block-bound participant inputs;

constructing, by an Interpret Component executing within an Interpret Quantum Privacy Cell, a Trust-Block-bound situational model from the received inputs;

enumerating, by a Propose Component executing within a Propose Quantum Privacy Cell, candidate actions with respective candidate-action Trust Blocks;

applying, by an Authorize Component executing within an Authorize Quantum Privacy Cell, a Trust Criteria check to each candidate action, and rejecting candidate actions failing the Trust Criteria check or whose target Trust Block carries a DORMANT-record marker absent additional activation conditions;

executing, by an Execute Component executing within an Execute Quantum Privacy Cell, authorized actions under Trust-Block-bound execution envelopes;

verifying, by a Verify Component executing within a Verify Quantum Privacy Cell, post-execution conformance of each executed action's effects to the originating Trust Block constraints;

incorporating, by a Learn Component executing within a Learn Quantum Privacy Cell, verified outcomes into a participant Trust Block-bound learning state;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the agent's actions are bound to per-step Trust Block authorization, post-hoc verification, and Trust-Block-bound learning, while interactions with DORMANT-marked participant records are constrained.

Claim 250. The system of claim 249, further comprising emitting, by each Component, a deterministic-replay descriptor recorded in an associated Trust Block such that the entire Loop execution is deterministically reproducible by independent verifiers.

Claim 251. The system of claim 249, further comprising rejecting, by the Authorize Component, any candidate action whose target Trust Block is marked DORMANT except where the candidate action constitutes a Manager-Originated record contribution-attribution write authorized by a Catalyst Network Manager Trust Block.

Claim 252. The system of claim 249, wherein executing, by the Execute Component, includes recording an execution-time attestation comprising at least: the candidate-action Trust Block identifier, an execution timestamp, and a content-addressed reference to executed-action effects.

Claim 253. The system of claim 249, further comprising, upon a verification non-conformance, executing a Trust-Block-bound rollback procedure that reverts the action's effects and records a Rollback Trust Block attestation in an Authorization Ledger.

Claim 254. The system of claim 249, wherein incorporating verified outcomes into the participant Trust Block-bound learning state is bounded by Proportionality and Balance constraints of an Adaptive Premium Compensation policy.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family I closes the Governed Agent Loop **&** Phase-0 DORMANT QPC State gap, binding 3 independent claims and 26 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.*

Family J — Premium Framework as Operative Allocation Mechanism

Claims 255–281 (27 claims: 3 independent + 24 dependent)

Family Introduction

Family J captures the Premium Framework as the operative allocation mechanism: 15-dimensional Premium parameterization (5 Launch + 8 Governance + 2 Adaptive), the AI-mediated allocation model with quid-pro-quo-rebutting compliance design, the Premium Multiple Compression Curve, and the Adaptive Premium Compensation runaway-prevention primitive.

Claims List

Claim 255. A multi-dimensional Premium parameterization system for QP Rewards allocation, comprising:

a Premium Registry recording, for each of fifteen Premium dimensions consisting of five Launch Premium dimensions, eight Governance Premium dimensions, and two Adaptive Premium dimensions, a Trust-Block-bound parameter set and a permitted value range;

a Premium Computation Engine executing within a Premium Quantum Privacy Cell and configured to compute, for each Resource Trust Block and each Settlement Ledger event involving the Resource Trust Block, a Premium vector comprising values for each of the fifteen Premium dimensions;

a Premium Validation Component configured to reject Premium vectors whose values exceed the permitted value range and to enforce, for each Adaptive Premium dimension, an Adaptive Compensation Proportionality bound such that the Adaptive Premium value does not exceed a Trust-Block-specified function of an underlying contribution magnitude;

a Premium Inheritance Component configured to propagate Premium-bearing fields of an upstream Trust Block to a downstream Resource Trust Block pursuant to a Premium Inheritance via Trust Block Chain;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby reward allocation depends on multi-dimensional Premium parameterization computed under Trust-Block authorization with Proportionality bounds preventing runaway compensation, and Premium inheritance propagates upstream attribution.

Claim 256. The system of claim 255, wherein the five Launch Premium dimensions comprise at least: a launch-timing Premium, an early-cascade Premium, an anchor-participation Premium, a distribution-activation Premium, and a regulatory-enablement Premium.

Claim 257. The system of claim 255, wherein the eight Governance Premium dimensions comprise at least: a contribution-quality Premium, an attestation-density Premium, a reuse-frequency Premium, an audit-cooperation Premium, an attribution-coverage Premium, a participation-continuity Premium, a governance-discipline Premium, and a regulatory-conformance Premium.

Claim 258. The system of claim 255, wherein the two Adaptive Premium dimensions comprise at least: an Adaptive Proportionality Premium and an Adaptive Balance Premium.

Claim 259. The system of claim 255, wherein the Adaptive Compensation Proportionality bound is computed as a function of contribution magnitude such that Adaptive Premium values cannot exceed a Trust-Block-specified multiple of contribution magnitude.

Claim 260. The system of claim 255, wherein the Premium Inheritance Component propagates the upstream Premium-bearing fields with a Trust-Block-specified decay factor such that derivative Resources inherit upstream Premiums attenuated by the decay factor.

Claim 261. The system of claim 255, wherein the Premium Validation Component records, in a Premium Validation Ledger, each rejected Premium vector with a Trust-Block-bound rejection reason.

Claim 262. The system of claim 255, wherein the Premium Computation Engine accepts inputs from a Three-Stage AI Evaluation Pipeline's composite Pipeline Attestation and from a Reputation Engine's Reputation Trust Block.

Claim 263. The system of claim 255, wherein each Premium dimension's value is constrained to a Trust-Block-specified permitted value range and integer Premium dimensions are enforced as integer values.

Claim 264. The system of claim 255, wherein the Premium Registry is updated under a Management Plane Trust Block, and each registry update emits a Premium-Registry-Update attestation recorded in a Control Plane Log.

Claim 265. The system of claim 255, wherein the system is integrated with a Settlement Controller such that the Premium vector for each Settlement Ledger event is provided as an input to Exchange Token allocation share computation.

Claim 266. A quid-pro-quo-rebutting AI-mediated rewards allocation system, comprising:

an AI Allocation Model executing within an Allocator Quantum Privacy Cell and configured to receive Premium vectors and Trust-Block-bound contribution attestations and to emit reward allocation determinations;

a Discretion Boundary Component configured to constrain the AI Allocation Model's discretion such that no allocation determination is dictated by any single participant's direct request and such that the allocation function combines multiple contributions and Premium dimensions before emitting any determination;

a Deterministic Replay Component configured to record, for each allocation determination, a deterministic-replay descriptor sufficient to permit independent reconstruction of the determination by third-party verifiers;

an Allocation Attestation Emitter configured to bind each allocation determination to a Trust Block recording the contributing Premium dimensions, the model snapshot identifier, and the deterministic-replay descriptor;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the AI Allocation Model produces discretionary reward allocations whose construction structurally rebuts a direct-exchange interpretation while preserving deterministic replayability for audit.

Claim 267. The system of claim 266, wherein the AI Allocation Model is a fixed model snapshot identified by a content-addressed identifier recorded in the Allocation Attestation, and replacement of the model snapshot requires a Management Plane Trust Block authorization.

Claim 268. The system of claim 266, wherein the Discretion Boundary Component enforces a Multi-Contribution Combination rule such that no allocation determination is computed from fewer than a Trust-Block-specified minimum number of distinct contribution attestations.

Claim 269. The system of claim 266, wherein the Discretion Boundary Component prohibits any direct-pairing between a single contribution and a single allocation amount, ensuring the allocation function combines multiple inputs.

Claim 270. The system of claim 266, wherein the AI Allocation Model executes within an Allocator QPC whose cryptographic boundary prohibits exfiltration of model parameters.

Claim 271. The system of claim 266, wherein the Deterministic Replay Component records a deterministic-replay descriptor sufficient that two independent verifiers presented with the same inputs and the same descriptor produce byte-identical allocation determinations.

Claim 272. The system of claim 266, wherein the Allocation Attestation Emitter records, in addition to the inputs and snapshot identifier, a Trust-Block-bound timestamp and the deterministic-replay descriptor.

Claim 273. The system of claim 266, wherein the AI Allocation Model is integrated with a Premium Multiple Compression Curve such that effective Premium Multiples are provided to the AI Allocation Model after Compression Curve application.

Claim 274. The system of claim 266, wherein the system is integrated with a Quid-Pro-Quo Rebuttal Audit Component configured to periodically attest, under a Management Plane Trust Block, that no allocation determination over a sampled set was structurally tied to a single direct request.

Claim 275. The system of claim 266, wherein the AI Allocation Model is replaceable under a Management Plane Trust Block, and each replacement event emits a Model-Replacement attestation recorded in a Control Plane Log.

Claim 276. A computer-implemented method for Premium Multiple compression across a maturity arc of a Quantum Privacy Network (QPN), comprising:

maintaining, by a Premium Compression Controller, a Trust-Block-bound Compression Curve specifying a compression schedule that reduces Premium Multiples over a maturity arc identified by Trust-Block-bound epoch markers;

computing, for each Settlement Ledger event, an effective Premium Multiple by applying the Compression Curve to the underlying Premium dimensions;

binding each effective Premium Multiple computation to a Trust Block recording the curve identifier, the epoch marker, and a deterministic-replay descriptor;

transmitting the effective Premium Multiple to an AI Allocation Model for inclusion in a reward allocation determination;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Premium Multiples are predictably reduced over the maturity arc under Trust-Block-bound compression policy, with each computation deterministically reproducible.

Claim 277. The system of claim 276, wherein the Trust-Block-bound Compression Curve is a piecewise function over epoch markers, each piecewise segment specifying a compression coefficient between zero and one.

Claim 278. The system of claim 276, wherein the Compression Curve's epoch markers correspond to Trust-Block-bound network maturity events including at least: a launch-completion event, a critical-mass-attainment event, an automated-operations-transition event, and a self-funding-maturation event.

Claim 279. The system of claim 276, wherein the Compression Controller emits, for each effective Premium Multiple computation, a deterministic-replay descriptor referencing the curve identifier, the active epoch marker, and the input Premium dimensions.

Claim 280. The system of claim 276, wherein the Compression Curve is updateable under a Management Plane Trust Block, and updates do not retroactively alter prior effective Premium Multiple computations.

Claim 281. The system of claim 276, wherein the Compression Curve enforces, at each epoch transition, a non-increasing constraint on effective Premium Multiples such that effective Premium Multiples cannot increase across epoch transitions.

Whereby Clause (Family-Level Structural Effect)

Whereby Family J closes the Premium Framework as Operative Allocation Mechanism gap, binding 3 independent claims and 24 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family K — Behavioral Activation, Reputation & Identity Resilience

Claims 282–308 (27 claims: 3 independent + 24 dependent)

Family Introduction

Family K captures the Six-Layer Catalyst Architecture (umbrella), the Behavioral Activation System (10 levels / 24 badges / dual-XP / evidence bonus / streak multiplier), the Multi-Factor Identity Binding, the Authorized Catalyst Proxy Addresses primitive, and the Catalyst Contribution Categories taxonomy. The Six-Layer Architecture provides the umbrella under which the gamification and identity primitives operate.

Claims List

Claim 282. A six-layer trust-verified contribution attribution architecture, comprising:

a Capture Layer comprising a contribution capture system implementing Trust-Block-bound capture pursuant to a Quantum Privacy Sidecar;

a Verification Layer comprising a Cross-Verification Protocol component, a Multi-Factor Identity Binding component, and a Witness Registration Interface, each executing within respective Quantum Privacy Cells;

a Ledger Layer comprising a Three-Ledger / Two-Log architecture wherein Contribution, Authorization, and Settlement Ledgers are maintained with Data Plane and Control Plane Logs;

a Reputation Layer comprising a Reputation Engine emitting Reputation Trust Blocks under a Three-Dimensional Reputation Schema;

an Allocation Layer comprising a Premium Framework Premium Registry and an AI Allocation Model;

a Governance Layer comprising a Catalyst Network Manager interface, a Permitted Audit Purposes registry, and a Management Plane Trust Block authority structure;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the six layers collectively form a trust-verified contribution attribution architecture with structural separation among capture, verification, ledger, reputation, allocation, and governance concerns.

Claim 283. The system of claim 282, wherein the Capture Layer's Quantum Privacy Sidecar comprises a Witness Agent, Listener Agent, Local Vault, and Verification Bridge each executing within a respective Quantum Privacy Cell.

Claim 284. The system of claim 282, wherein the Verification Layer's Cross-Verification Protocol component requires a Trust-Block-specified minimum count of independent witness attestations before emitting a positive authenticity indicator.

Claim 285. The system of claim 282, wherein the Ledger Layer enforces, by Cross-Ledger Verifier, that each Settlement Ledger record references a valid Authorization Ledger record and a valid Contribution Ledger record.

Claim 286. The system of claim 282, wherein the Reputation Layer's Three-Dimensional Reputation Schema comprises a Contribution Dimension, an Identity Dimension, and a Behavioral Dimension, each emitting Trust-Block-bound scores.

Claim 287. The system of claim 282, wherein the Allocation Layer's Premium Framework comprises fifteen Premium dimensions divided into five Launch, eight Governance, and two Adaptive Premium dimensions.

Claim 288. The system of claim 282, wherein the Governance Layer's Catalyst Network Manager interface enforces Permitted Audit Purposes including at least: attribution-accuracy verification, attribution-dispute resolution, pipeline integrity audit, fraud investigation, and Universal Trust Model and applicable-law compliance verification.

Claim 289. The system of claim 282, wherein inter-layer communication is mediated exclusively through Trust-Block-bound envelopes whose authorization is verified by Proof-of-Trust.

Claim 290. The system of claim 282, wherein each layer's components operate within respective Quantum Privacy Cells under layer-specific Trust Criteria.

Claim 291. A behavioral activation system for participant engagement within a QPN, comprising:

a Level Tracker maintaining, for each participant Trust Block, an integer activity level among at least ten levels and configured to advance the activity level upon satisfaction of a Trust-Block-bound level-advancement condition;

a Badge Registry recording at least twenty-four distinct badge definitions, each badge bound to a Trust-Block-bound award condition;

a Dual Experience Currency Tracker maintaining, for each participant Trust Block, two distinct experience currencies, each currency accruing under a Trust-Block-bound accrual policy and being separately bounded;

an Evidence-Bonus Component configured to apply an experience-currency multiplier upon detection of corroborating evidence under a Cross-Verification Protocol;

a Streak Multiplier Component configured to apply an experience-currency multiplier upon detection of consecutive Trust-Block-bound participation events within a Trust-Block-specified streak window;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby participant engagement is gamified under cryptographically-attestable level, badge, and currency mechanics, with evidence-bonus and streak amplification driving engagement frequency.

Claim 292. The system of claim 291, wherein the at least ten activity levels are arranged in a non-decreasing progression and each level specifies a Trust-Block-bound level-advancement condition.

Claim 293. The system of claim 291, wherein the Badge Registry records at least twenty-four badges including at least: a streak badge, a verified-contribution badge, an evangelization badge, a corroboration badge, and an identity-factor badge, each bound to a Trust-Block-specified award condition.

Claim 294. The system of claim 291, wherein the Dual Experience Currency Tracker's two currencies comprise a Contribution Experience currency and a Behavioral Experience currency, each separately bounded.

Claim 295. The system of claim 291, wherein the Evidence-Bonus Component's multiplier is computed from a Trust-Block-specified function of corroborating evidence count.

Claim 296. The system of claim 291, wherein the Streak Multiplier Component's multiplier is reset upon detection of an unmet streak condition recorded under a Trust-Block-bound break event.

Claim 297. The system of claim 291, wherein the Behavioral Activation System emits, for each level advancement, a Trust-Block-bound Level-Up attestation recorded in a Behavioral Ledger.

Claim 298. The system of claim 291, wherein the Behavioral Activation System integrates with a Reputation Engine such that the Behavioral Dimension of the Three-Dimensional Reputation Schema is derived in part from the Level Tracker, Badge Registry, and Dual Experience Currency Tracker records.

Claim 299. The system of claim 291, wherein the Behavioral Activation System enforces a Proportionality bound on currency accrual such that no single participation event accrues more than a Trust-Block-specified maximum currency increment.

Claim 300. A multi-factor identity binding system, comprising:

a Factor Registration Interface configured to record, in a participant Trust Block, identity factors selected from at least: a credentialed-identity factor, a biometric-identity factor, a device-bound factor, a behavioral-identity factor, and a witness-attested-identity factor;

an Identity Composition Component configured to compose registered identity factors into a single Identity Trust Block whose Trust Criteria specifies a minimum count of factors required for identity binding;

a Factor Verification Component configured to verify, for each factor, a Trust-Block-bound proof of factor authenticity, with rejection of factors failing verification;

an Authorized Catalyst Proxy Address Component configured to issue, under a Proxy Trust Block, a proxy address authorizing a delegated participant or system component to capture contributions on behalf of the principal under a Trust-Block-specified scope;

a Catalyst Contribution Category Registry recording at least nine primary contribution categories together with a UX-tier annotation and a Program-to-Objectives mapping;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby participant identity is bound to multiple cryptographically-attested factors, contribution categories are taxonomically structured, and delegated capture is enabled through Trust-Block-scoped proxy addresses.

Claim 301. The system of claim 300, wherein the Factor Registration Interface verifies, for each registered factor, a Trust-Block-bound proof of factor authenticity and rejects factors failing verification.

Claim 302. The system of claim 300, wherein the Identity Composition Component's Trust Criteria specifies a minimum count of at least two independently-attested identity factors required for identity binding.

Claim 303. The system of claim 300, wherein the Factor Verification Component records, for each factor verification, a Trust-Block-bound verification attestation in an Authorization Ledger.

Claim 304. The system of claim 300, wherein the Authorized Catalyst Proxy Address Component issues a proxy address whose Proxy Trust Block carries a Trust-Block-specified scope including at least: a permitted contribution category, a temporal validity window, and a permitted Privacy Network scope.

Claim 305. The system of claim 300, wherein the Catalyst Contribution Category Registry's at least nine primary categories comprise at least: a creative-contribution category, a curation category, an attestation category, an organization-membership category, an evangelization category, a coordination category, an execution category, a witnessing category, and a learning category.

Claim 306. The system of claim 300, wherein the UX-tier annotation classifies the category among at least: an active-participation tier, a passive-engagement tier, and an organizational tier.

Claim 307. The system of claim 300, wherein the Program-to-Objectives mapping records the contribution category's association with at least one Trust-Block-bound program objective specified by a Catalyst Network program.

Claim 308. The system of claim 300, wherein the Authorized Catalyst Proxy Address Component is configured to revoke a proxy address under a Trust-Block-bound revocation event, with revocation recorded in a Control Plane Log.

Whereby Clause (Family-Level Structural Effect)

*Whereby Family K closes the Behavioral Activation, Reputation **&** Identity Resilience gap, binding 3 independent claims and 24 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.*

Family L — Senior/Junior QPT Derivative Capital-Formation Architecture

Claims 309–335 (27 claims: 4 independent + 23 dependent)

Family Introduction

Family L captures the senior/junior QPT Derivative capital-formation architecture: Senior QPT Derivative with dual-hurdle RBF, contractual authority to encumber the full Exchange Root Token allocation and Accelerator Incentive & Investment Pool allocation under deferred activation, Junior Derivative Offset Mechanism (three configurable levels), and Accrual Rights Swap.

Claims List

Claim 309. A senior derivative instrument system for revenue-based financing of a Quantum Privacy Network, comprising:

a Senior Derivative Registry recording, for each issued Senior Derivative, a Trust-Block-bound principal amount, a target internal rate of return (IRR) hurdle, a target multiple of invested capital (MOIC) hurdle, and an issuer payoff option;

a Settlement-Linked Distribution Component configured to direct, from a Settlement Ledger's Exchange Token issuance flow, a Trust-Block-bound distribution share to each Senior Derivative holder until both the target IRR hurdle and the target MOIC hurdle are achieved;

an Issuer Payoff Option Component configured to permit the issuer, upon achievement of a Trust-Block-specified payoff condition, to extinguish the Senior Derivative by paying out a Trust-Block-specified payoff amount;

a Hurdle Verification Component configured to compute, deterministically and under Trust Block authorization, the achievement of the IRR and MOIC hurdles from Settlement Ledger records;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Senior Derivative holders receive revenue-based-financing returns under dual cryptographically-attestable hurdles, with issuer payoff optionality preserving capital efficiency.

Claim 310. The system of claim 309, wherein the target IRR hurdle is computed deterministically from Settlement Ledger records and is achieved when cumulative distributions to the Senior Derivative holder yield an internal rate of return at or above the specified hurdle.

Claim 311. The system of claim 309, wherein the target MOIC hurdle is achieved when cumulative distributions to the Senior Derivative holder reach or exceed the specified multiple of the principal amount.

Claim 312. The system of claim 309, wherein the issuer payoff option is exercisable only upon achievement of both the target IRR hurdle and the target MOIC hurdle, and only under a Trust-Block-bound issuer authorization.

Claim 313. The system of claim 309, wherein the Hurdle Verification Component records, in a Hurdle-Verification Ledger, each hurdle-achievement event under a Trust-Block-bound attestation.

Claim 314. The system of claim 309, wherein the Settlement-Linked Distribution Component prioritizes distributions to Senior Derivative holders over Junior Derivative holders pursuant to a settlement waterfall recorded in a Settlement Trust Block.

Claim 315. The system of claim 309, wherein the Senior Derivative Registry records, in addition to the principal amount, a Trust-Block-bound issuance timestamp and a Trust-Block-bound maturity descriptor.

Claim 316. The system of claim 309, wherein the Senior Derivative is bound to a Resource Trust Block of the issuing entity, and wherein the Resource Trust Block specifies the underlying Exchange Token allocation share over which the Senior Derivative may claim distributions.

Claim 317. The system of claim 309, wherein the Hurdle Verification Component is deterministically reproducible such that two independent verifiers presented with the same Settlement Ledger records produce identical hurdle-achievement outcomes.

Claim 318. A pre-activation encumbrance system for ERT and Accelerator Incentive & Investment Pool allocations under deferred activation, comprising:

an Encumbrance Authority Registrar configured to record, under an Encumbrance Trust Block, a contractual authority of an issuing entity to encumber a specified Exchange Root Token allocation share and a specified Accelerator Incentive & Investment Pool allocation share prior to participant activation;

an Allocation Reservation Component configured to mark, in a Settlement Ledger, future-issuance allocation shares as encumbered such that, upon issuance, the encumbered shares are directed to a holder identified by the Encumbrance Trust Block;

a Deferred-Activation Compatibility Component configured to ensure that encumbrance does not transfer economic function of participant allocations from DORMANT to ACTIVE participants and that participant allocations remain bound by Phase-0 DORMANT-state constraints until participant self-activation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby pre-activation capital formation is enabled against future settlement flows while DORMANT-state participant allocations retain their pre-activation compliance properties.

Claim 319. The system of claim 318, wherein the Encumbrance Trust Block specifies a maximum encumbered share of the Exchange Root Token allocation and a maximum encumbered share of the Accelerator Incentive & Investment Pool allocation.

Claim 320. The system of claim 318, wherein the Allocation Reservation Component is structurally incapable of redirecting an allocation share absent a corresponding Encumbrance Trust Block.

Claim 321. The system of claim 318, wherein the Deferred-Activation Compatibility Component prohibits the Encumbrance Trust Block from impacting any DORMANT-marked participant record such that DORMANT records' allocations remain pre-activation-constrained.

Claim 322. The system of claim 318, wherein the Encumbrance Authority Registrar records, in an Encumbrance Ledger, each encumbrance event with a Trust-Block-bound timestamp and a content-addressed reference to the underlying contractual instrument.

Claim 323. The system of claim 318, wherein the Encumbrance Trust Block is revocable under a Management Plane Trust Block upon a Trust-Block-specified revocation condition.

Claim 324. A junior derivative offset mechanism system, comprising:

a Junior Derivative Registry recording, for each issued Junior Derivative, a Trust-Block-bound principal amount and an offset-level selection from a set of at least three configurable offset levels;

a First-Loss Absorption Component configured to absorb, from a Settlement-Linked Distribution Component, an amount equal to a Trust-Block-specified function of senior shortfall up to a per-Junior-Derivative limit;

a Settlement Ledger annotation component configured to record, for each loss-absorption event, a Trust-Block-bound annotation referencing the Senior Derivative shortfall and the Junior Derivative absorbing the loss;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Junior Derivative holders provide first-loss capital to Senior Derivative holders under cryptographically-attestable offset levels, with each loss-absorption event recorded for audit.

Claim 325. The system of claim 324, wherein the at least three configurable offset levels comprise at least: a one-times-senior offset level, a two-times-senior offset level, and a three-times-senior offset level.

Claim 326. The system of claim 324, wherein the First-Loss Absorption Component caps total loss absorption at the Junior Derivative principal amount.

Claim 327. The system of claim 324, wherein the Settlement Ledger annotation component records, for each loss-absorption event, a content-addressed reference to the Senior Derivative shortfall record.

Claim 328. The system of claim 324, wherein the Junior Derivative Registry additionally records a Trust-Block-bound subordination position relative to other Junior Derivatives within the same offset level.

Claim 329. The system of claim 324, wherein the Junior Derivative is bound to a Resource Trust Block of the issuing entity, and wherein the Resource Trust Block specifies the underlying allocations against which the Junior Derivative offsets shortfall.

Claim 330. An accrual rights swap system for Premium-parameterized cash flows, comprising:

a Swap Registry recording, for each Accrual Rights Swap, a first counterparty Trust Block, a second counterparty Trust Block, a swapped Premium dimension set, and a Trust-Block-bound valuation derived from a Premium Multiple Compression Curve;

a Swap Settlement Component configured to redirect, in a Settlement Ledger, accruals on the swapped Premium dimensions from the first counterparty to the second counterparty pursuant to the Swap Registry;

a Swap Audit Component configured to record, in an Authorization Ledger, each Swap registration and Swap settlement event under a Trust-Block-bound audit attestation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby accrual rights on Premium-parameterized cash flows are exchangeable under cryptographically-attestable terms with valuation indexed to the Compression Curve.

Claim 331. The system of claim 330, wherein the Trust-Block-bound valuation is computed as a function of the Premium Multiple Compression Curve at the time of Swap registration and Premium dimension values bound to the underlying Resource Trust Block.

Claim 332. The system of claim 330, wherein the Swap Settlement Component redirects accruals at the granularity of individual Settlement Ledger events such that each event's accrual is split between the first and second counterparty pursuant to the swapped Premium dimensions.

Claim 333. The system of claim 330, wherein the Swap Audit Component records, for each Swap settlement, a content-addressed reference to the source Settlement Ledger event.

Claim 334. The system of claim 330, wherein the Accrual Rights Swap is revocable under a Trust-Block-bound mutual-consent revocation event recorded in the Swap Registry.

Claim 335. The system of claim 330, wherein the Swap Registry additionally records a swap-maturity descriptor specifying the temporal scope of the Swap, after which accrual redirection ceases.

Whereby Clause (Family-Level Structural Effect)

Whereby Family L closes the Senior/Junior QPT Derivative Capital-Formation Architecture gap, binding 4 independent claims and 23 dependent claims to the foundational QPN primitives via the §22.7 ***canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.***

Family M — Stage-Differentiated Revert + Accelerator Lock + MUM Tracking

Claims 336–362 (27 claims: 3 independent + 24 dependent)

Family Introduction

Family M captures the Stage-Differentiated Revert mechanism (Pioneer / Cascade / Automated / Self-Funding), the Accelerator Lock Mechanism that combines Monetization Uplift Multiple (MUM) thresholds with Governance DNA conditions to render Accelerator Incentive & Investment Pool allocations non-dilutable, and MUM tracking. Functional cohesion: all three members govern Accelerator-level economic state transitions. Each independent claim recites concrete Trust-Block-bound state transitions executed by a Settlement Controller within QPC infrastructure, with the §22.7 canonical Wherein clause anchoring 2016 priority on the QPN infrastructure.

Claims List

Claim 336. A stage-differentiated capital-recovery system for a Privacy Network Exchange, comprising:

a Stage Indicator executing within a first Quantum Privacy Cell (QPC) and configured to identify a current ecosystem stage of an Accelerator from a stage set consisting of Pioneer, Cascade, Automated, and Self-Funding;

a Revert Waterfall Engine executing within a second QPC distinct from the first QPC and configured to evaluate, upon a capital-recovery event, a Trust-Block-bound waterfall whose priority ordering is selected as a function of the current ecosystem stage identified by the Stage Indicator;

a Trust Block-bound waterfall specification recording, for each stage of the stage set, a distinct ordering of Return of Capital and Return on Investment priorities;

a Settlement Controller configured to deterministically issue Exchange Tokens to a Priority Pool, a Backing Pool, and a Participant Pool in accordance with the waterfall ordering selected by the Revert Waterfall Engine;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby capital recovery is determined by ecosystem maturity rather than by negotiation, and the same Trust-Block-bound waterfall specification deterministically governs recovery across all participants of the Accelerator.

Claim 337. The system of claim 336, wherein the Stage Indicator identifies the Pioneer stage by reference to a Trust-Block-bound indicator of pre-cascade adoption, the Cascade stage by reference to an indicator of cross-Accelerator reuse exceeding a Trust-Block-specified threshold, the Automated stage by reference to an indicator of AI-mediated settlement exceeding a Trust-Block-specified threshold, and the Self-Funding stage by reference to an indicator of settlement-funded operational sustainability.

Claim 338. The system of claim 336, wherein, in the Pioneer stage, the Trust-Block-bound waterfall prioritizes Return of Capital to a Priority Pool over Return on Investment to a Participant Pool.

Claim 339. The system of claim 336, wherein, in the Cascade stage, the Trust-Block-bound waterfall introduces a Backing Pool absorption priority between Return of Capital and Return on Investment.

Claim 340. The system of claim 336, wherein, in the Automated stage, the Trust-Block-bound waterfall introduces an algorithmic re-tranching priority that absorbs uneven settlement flows before Participant Pool distribution.

Claim 341. The system of claim 336, wherein, in the Self-Funding stage, the Trust-Block-bound waterfall removes the Priority Pool priority and distributes settlement directly to a Participant Pool in proportion to participation rights.

Claim 342. The system of claim 336, wherein the Revert Waterfall Engine is configured to detect a stage transition event by observing a sustained crossing of a Trust-Block-specified threshold, and to commit the stage transition to the Stage Indicator only after a confirmation window specified in the Trust Block.

Claim 343. The system of claim 336, wherein each capital-recovery event is recorded with a stage tag identifying the ecosystem stage in effect at the time of the recovery event.

Claim 344. The system of claim 336, wherein the Stage Indicator publishes the current ecosystem stage to a Catalyst Network registry under a Privacy Network Exchange-wide naming convention.

Claim 345. The system of claim 336, wherein a stage transition causes the Trust Block-bound waterfall specification to be re-loaded by the Revert Waterfall Engine without operator intervention, and an audit record of the re-load event is written to a Control Plane Log.

Claim 346. A computer-implemented method for non-dilutable Accelerator Incentive & Investment Pool allocation, the method executing within a Quantum Privacy Network and comprising:

computing, by a Monetization Uplift Multiple Engine executing within a first Quantum Privacy Cell, a current Monetization Uplift Multiple (MUM) value for an Accelerator from settlement records of the Accelerator over a specified measurement window;

evaluating, by a Governance DNA Evaluator executing within a second Quantum Privacy Cell, a Governance DNA condition vector specified in a Trust Block associated with an Accelerator Incentive & Investment Pool allocation;

upon both (a) the current MUM value meeting or exceeding a MUM threshold recorded in the Trust Block and (b) the Governance DNA condition vector evaluating to true, asserting an Accelerator Lock state in the Trust Block governing the Accelerator Incentive & Investment Pool allocation;

rejecting, by a Settlement Controller, any subsequent attempt to dilute the allocation while the Accelerator Lock state is asserted, regardless of operator instructions;

permitting, by the Settlement Controller, ongoing accrual of Exchange Tokens to the allocation in accordance with the Trust Block while the Accelerator Lock state is asserted;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby the Accelerator Incentive & Investment Pool allocation becomes economically non-dilutable as a deterministic consequence of measured performance and governance compliance, anchoring topology and removing operator discretion from the dilution decision.

Claim 347. The system of claim 346, wherein the Governance DNA condition vector includes at least one of: a compliance attestation, a Premium-allocation conformance attestation, an attribution-integrity attestation, and a Trust Criteria-conformance attestation.

Claim 348. The system of claim 346, wherein the MUM threshold is specified as a multi-window threshold requiring the MUM value to remain at or above a specified level across a Trust-Block-specified minimum number of consecutive measurement windows.

Claim 349. The system of claim 346, wherein, upon assertion of the Accelerator Lock state, the Settlement Controller emits an Accelerator Lock Notice to a Catalyst Network registry, and Trust Blocks referencing the allocation acquire a Lock-Aware status that propagates to derivative Trust Blocks created thereafter.

Claim 350. The system of claim 346, wherein dissolution of the Accelerator Lock state requires both (a) the MUM value falling below a Trust-Block-specified de-Lock threshold and (b) the Governance DNA condition vector evaluating to false, with both conditions sustained across a Trust-Block-specified confirmation window.

Claim 351. The system of claim 346, wherein, while the Accelerator Lock state is asserted, attempted operator-initiated dilution instructions are recorded as rejected events on a Control Plane Log and produce no change to the Accelerator Incentive & Investment Pool allocation.

Claim 352. The system of claim 346, wherein the Accelerator Lock state is independently observable by Catalyst Network participants through inspection of the Trust Block, without disclosure of underlying participant-identifying information.

Claim 353. The system of claim 346, wherein the Governance DNA condition vector is composed of dimension-specific predicates, and each dimension's predicate is independently evaluable from Control Plane Log entries without joint disclosure of dimension values to any single observer.

Claim 354. A Monetization Uplift Multiple tracking system for Accelerator-level performance evaluation, comprising:

a Settlement Sampler executing within a first Quantum Privacy Cell and configured to sample, from a Settlement Ledger of an Accelerator, settlement records over a specified measurement window;

a Baseline Calculator executing within a second QPC and configured to compute, from a baseline Trust Block specified at Accelerator launch, an expected baseline settlement magnitude for the measurement window;

a Multiple Computer executing within a third QPC and configured to compute a current MUM value as a ratio of observed settlement magnitude to expected baseline settlement magnitude, with the computation traceable to specific Settlement Ledger entries through a deterministic-replay procedure;

a MUM Ledger configured to record MUM values per measurement window with cryptographic linkage to the Settlement Ledger entries used to compute each value;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Monetization Uplift Multiple values are deterministic, replayable, and tamper-evident, and so are suitable for use as threshold inputs to non-discretionary economic state transitions.

Claim 355. The system of claim 354, wherein the baseline Trust Block is specified at Accelerator launch and may not be modified thereafter except by a Trust-Block-bound governance event that emits a new baseline Trust Block and supersedes the prior baseline in a forward-only manner.

Claim 356. The system of claim 354, wherein the Multiple Computer applies a quantum-safe hash to each Settlement Ledger entry used to compute the MUM value and records the hash in the MUM Ledger.

Claim 357. The system of claim 354, wherein the measurement window is specified in the Trust Block as one of: a fixed-duration calendar window, a sliding-event-count window, and a hybrid window combining duration and event-count criteria.

Claim 358. The system of claim 354, wherein the MUM Ledger is written under a write-once cryptographic discipline such that prior MUM values are inspectable by any participant authorized under a Trust Block but cannot be retroactively modified by any participant.

Claim 359. The system of claim 354, wherein the Settlement Sampler is configured to reject Settlement Ledger entries that fail Proof-of-Trust verification prior to inclusion in the MUM computation.

Claim 360. The system of claim 354, wherein the MUM computation is performed by deterministic replay such that any participant authorized under a Trust Block may independently reproduce the MUM value from the recorded Settlement Ledger entries.

Claim 361. The system of claim 354, wherein, when an Accelerator operates across multiple Privacy Network Exchange substrates, the MUM Engine aggregates Settlement Ledger entries across substrates by reference to a substrate-agnostic Trust Block lineage identifier.

Claim 362. The system of claim 354, wherein the MUM Engine emits a deterministic-replay receipt for each MUM computation, said receipt being recorded on the MUM Ledger and being usable to challenge the MUM value via a re-replay procedure.

Whereby Clause (Family-Level Structural Effect)

Whereby Family M closes the Stage-Differentiated Revert + Accelerator Lock + MUM Tracking gap, binding 3 independent claims and 24 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family N — Liquidity Architecture

Claims 363–395 (33 claims: 5 independent + 28 dependent)

Family Introduction

Family N captures the Liquidity Architecture of the Privacy Network Exchange: (i) the architectural separation of the Exchange Provider role from the Liquidity Provider role; (ii) the Backing Pool Multiples Architecture; (iii) the Tranche Priority Pool Absorption mechanism; (iv) QPT-Collateralized Lending; and (v) Existing-FI Balance-Sheet Tokenization Integration. Functional cohesion: all five members address liquidity provision and risk-tranching for the PNX. Each independent claim recites concrete Trust-Block-bound liquidity-provisioning primitives, with the §22.7 canonical Wherein clause anchoring 2016 priority on the QPN infrastructure.

Claims List

Claim 363. A liquidity-role-separation system for a Privacy Network Exchange, comprising:

an Exchange Provider role authorized by a first Trust Block to act as settlement counterparty for cross-party Resource reuse and to receive a settlement counterparty share of Exchange Tokens issued upon such reuse;

a Liquidity Provider role authorized by a second Trust Block distinct from the first Trust Block to commit capital to a Backing Pool and to receive a liquidity-provision share of Exchange Tokens issued from the Backing Pool waterfall;

a Role Separation Enforcer executing within a Quantum Privacy Cell and configured to reject any Trust Block authorizing the same participant simultaneously in both the Exchange Provider role and the Liquidity Provider role with respect to a single Resource Pool, except upon a Trust-Block-bound governance authorization providing dual-role disclosure;

a Settlement Controller configured to issue Exchange Tokens to the Exchange Provider role and to the Liquidity Provider role from non-overlapping settlement streams whose flows are independently auditable on a Settlement Ledger;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby settlement-counterparty risk and capital-provision risk are architecturally separated and may be priced, traded, and audited independently.

Claim 364. The system of claim 363, wherein the Role Separation Enforcer additionally rejects, in the absence of a Trust-Block-bound governance authorization providing dual-role disclosure, any Trust Block in which a single beneficial-owner identifier appears in both the Exchange Provider role and the Liquidity Provider role.

Claim 365. The system of claim 363, wherein the Exchange Provider role share of Exchange Tokens and the Liquidity Provider role share of Exchange Tokens are recorded under distinct Settlement Ledger account scopes, such that aggregate amounts are independently queryable.

Claim 366. The system of claim 363, wherein a dual-role disclosure authorization additionally requires recording, in the governance Trust Block, a conflict-management attestation specifying participant-protection terms in effect during dual-role operation.

Claim 367. The system of claim 363, wherein the Role Separation Enforcer is configured to detect role conflicts spanning Resource Pools by reference to a Resource Pool lineage identifier preserved in Trust Blocks.

Claim 368. The system of claim 363, wherein each role's Trust Block additionally specifies a role-revocation procedure executable upon a Trust-Block-bound governance event, and revocation under said procedure is recorded on a Control Plane Log.

Claim 369. A Backing Pool Multiples system for liquidity capacity parameterization, comprising:

a Trust Block specifying, for a Backing Pool, three Backing Pool Multiples parameters: a portfolio multiple, an annual-flow multiple, and a cap-amount;

a Capacity Engine executing within a first Quantum Privacy Cell and configured to compute, from the three Backing Pool Multiples parameters and from observed Settlement Ledger inputs, a current Backing Pool capacity governing the maximum derivative shortfall the Backing Pool may absorb;

a Shortfall Router executing within a second QPC and configured, upon a derivative shortfall event recorded by a Settlement Controller, to route the shortfall to the Backing Pool up to the current Backing Pool capacity and to overflow remaining shortfall to a fallback waterfall position specified in the Trust Block;

a Backing Pool Ledger configured to record Backing Pool inflows and outflows with cryptographic linkage to the Trust Block-bound capacity parameters in effect at each event;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Backing Pool capacity is determined by three independent risk-management dimensions that admit independent parameter audit while preserving deterministic shortfall routing.

Claim 370. The system of claim 369, wherein the portfolio multiple specifies a maximum ratio of Backing Pool capacity to a portfolio-wide settlement-token outstanding amount recorded on the Settlement Ledger.

Claim 371. The system of claim 369, wherein the annual-flow multiple specifies a maximum ratio of Backing Pool capacity to an annualized settlement-flow rate computed by the Capacity Engine from Settlement Ledger entries over a Trust-Block-specified prior window.

Claim 372. The system of claim 369, wherein the cap-amount specifies an absolute upper bound on Backing Pool capacity expressed in a unit recorded in the Trust Block.

Claim 373. The system of claim 369, wherein the Capacity Engine selects the minimum of the three Backing Pool Multiples parameters as the operative Backing Pool capacity at any point in time.

Claim 374. The system of claim 369, wherein adjustment of any Backing Pool Multiples parameter requires a Trust-Block-bound governance event, and the adjusted parameter takes effect on a forward-only basis from a Trust-Block-specified effective time.

Claim 375. The system of claim 369, wherein the Shortfall Router additionally records, for each shortfall event, the Backing Pool Multiples parameters in effect at the time of the event, such that historical shortfall absorptions are reconstructible by reference to the recorded parameters.

Claim 376. The system of claim 369, wherein the Backing Pool Ledger is configured under a write-once cryptographic discipline preventing retroactive modification of recorded inflows and outflows.

Claim 377. The system of claim 369, wherein, when no parameter is finite, the Shortfall Router rejects the shortfall event and emits an exception receipt to the Control Plane Log specifying the unfilled shortfall.

Claim 378. A computer-implemented method for two-stage tranche priority absorption, the method executing within a Privacy Network Exchange and comprising:

receiving, by a Settlement Controller executing within a first Quantum Privacy Cell, a settlement event for a Resource Pool;

absorbing, in a first stage, a Return of Capital portion of the settlement event into a Priority Pool until participant Return of Capital obligations recorded in a Trust Block are satisfied;

distributing, in a second stage commenced upon completion of the first stage, a Return on Investment portion of the settlement event to a Participant Pool in proportion to participation rights recorded in the Trust Block;

recording, on a Settlement Ledger, the first-stage absorption and the second-stage distribution under cryptographic linkage to the Trust Block in effect at the time of the settlement event;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Return of Capital obligations are satisfied with priority and predictability before Return on Investment distribution begins, with both stages auditable from a single Settlement Ledger.

Claim 379. The system of claim 378, wherein the Trust Block additionally specifies, for the first stage, a Priority Pool composition identifier indicating which participant accounts comprise the Priority Pool at the time of the settlement event.

Claim 380. The system of claim 378, wherein the Trust Block additionally specifies, for the second stage, a Participant Pool composition identifier indicating the participation rights distribution in effect at the time of the settlement event.

Claim 381. The system of claim 378, wherein the Settlement Controller emits, between the first stage and the second stage, a stage-transition receipt recording Priority Pool satisfaction on the Settlement Ledger.

Claim 382. The system of claim 378, wherein, when the first stage exhausts the settlement event without fully satisfying Priority Pool obligations, the Settlement Controller defers the second stage and records the unfilled Priority Pool obligation as a forward-carried claim against subsequent settlement events.

Claim 383. The system of claim 378, wherein the Trust Block enforces that Return of Capital obligations of the Priority Pool are denominated in a unit identical to the unit of settlement events, eliminating a unit-conversion exposure within the waterfall.

Claim 384. A QPT-collateralized lending system, comprising:

a Collateral Vault executing within a first Quantum Privacy Cell and configured to lock a specified quantity of Quantum Privacy Tokens (QPTs) under a Trust Block authorizing a lending event;

a Loan Issuance Engine executing within a second QPC and configured to issue a loan obligation, in an amount determined by the locked QPT quantity and a loan-to-value parameter specified in the Trust Block, to a borrower designated in the Trust Block;

a Margin Monitor executing within a third QPC and configured to monitor a margin condition specified in the Trust Block, said margin condition being computed from a QPT reference value publishable on a Privacy Network Exchange-wide registry;

a Liquidation Controller configured, upon the margin condition crossing a Trust-Block-specified liquidation threshold, to liquidate a Trust-Block-specified fraction of the locked QPTs and to apply the proceeds to the loan obligation;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby QPTs are usable as collateral for lending events under Trust-Block-bound terms, with margin monitoring and liquidation occurring deterministically without operator discretion.

Claim 385. The system of claim 384, wherein the loan-to-value parameter is specified in the Trust Block as a static value, a piecewise function of QPT reference value, or a function of additional Trust-Block-bound risk parameters.

Claim 386. The system of claim 384, wherein the Margin Monitor publishes, to a Privacy Network Exchange-wide registry, the QPT reference value computed under a Trust-Block-specified reference methodology.

Claim 387. The system of claim 384, wherein the Liquidation Controller is configured to liquidate locked QPTs in tranches specified in the Trust Block rather than as a single liquidation event, with each tranche recorded on the Settlement Ledger.

Claim 388. The system of claim 384, wherein, upon repayment of the loan obligation in full, the Collateral Vault releases the locked QPTs to a recipient address specified in the Trust Block.

Claim 389. The system of claim 384, wherein the Trust Block additionally specifies a forbearance procedure executable upon a Trust-Block-bound governance event, said forbearance procedure suspending margin monitoring for a Trust-Block-specified duration.

Claim 390. A balance-sheet tokenization integration system for an existing financial institution, comprising:

a Balance-Sheet Witness executing within a first Quantum Privacy Cell of an existing financial institution and configured to attest, under a Trust Block authorized by the financial institution, to balance-sheet states of designated asset classes without disclosure of underlying account-level data;

a Tokenization Bridge executing within a second QPC and configured to issue, in correspondence with attested balance-sheet states, Privacy-Network-Exchange-recognized Tokenized Balance-Sheet Claims under Trust Blocks that inherit the financial institution's attestation lineage;

a Reconciliation Engine executing within a third QPC and configured to reconcile, on a Trust-Block-specified cadence, attested balance-sheet states with outstanding Tokenized Balance-Sheet Claims and to emit reconciliation receipts to a Settlement Ledger;

a Redemption Controller configured to honor redemption requests for Tokenized Balance-Sheet Claims against the financial institution's balance sheet in accordance with the Trust Block governing each Claim;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby existing-financial-institution balance sheets are integratable with Privacy Network Exchange liquidity without disclosure of underlying account-level data and with deterministic reconciliation receipts.

Claim 391. The system of claim 390, wherein the Balance-Sheet Witness attests to balance-sheet states by reference to a zero-knowledge proof of compliance with a balance-sheet predicate specified in the Trust Block, without disclosure of underlying transaction-level data.

Claim 392. The system of claim 390, wherein the Tokenization Bridge issues each Tokenized Balance-Sheet Claim under a Trust Block whose lineage chain references both the attestation Trust Block and the financial institution's authorization Trust Block.

Claim 393. The system of claim 390, wherein the Reconciliation Engine emits a reconciliation receipt recording (a) the attested balance-sheet state, (b) the aggregate outstanding Tokenized Balance-Sheet Claims, and (c) a delta between (a) and (b), with the receipt cryptographically linked to the underlying Settlement Ledger entries.

Claim 394. The system of claim 390, wherein the Redemption Controller is configured to enforce a Trust-Block-specified redemption window, redemption notice procedure, and settlement-currency designation for each redemption request.

Claim 395. The system of claim 390, wherein, upon detection of a reconciliation delta exceeding a Trust-Block-specified threshold, the Tokenization Bridge suspends issuance of additional Tokenized Balance-Sheet Claims until the delta is resolved.

Whereby Clause (Family-Level Structural Effect)

Whereby Family N closes the Liquidity Architecture gap, binding 5 independent claims and 28 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family O — Quantum DNA/Genome Inheritance Architecture

Claims 396–437 (42 claims: 5 independent + 37 dependent)

Family Introduction

Family O captures the Quantum DNA/Genome Inheritance Architecture: (i) the three-level inheritance hierarchy (Gene / DNA / Genome); (ii) Lamarckian inheritance of acquired governance characteristics; (iii) multi-Genome architecture; (iv) governance superposition of DNA expression; (v) mitochondrial DNA analog for operational infrastructure lineage; (vi) Premium inheritance via Trust Block chain; and (vii) the Multi-Layered Aggregation Principle. Functional cohesion: all seven members address how governance characteristics propagate across participants and resources within a QPN. Each independent claim recites concrete Trust-Block-bound inheritance and aggregation primitives, with the §22.7 canonical Wherein clause anchoring 2016 priority on the QPN infrastructure.

Claims List

Claim 396. A three-level governance inheritance system for a Quantum Privacy Network, comprising:

a Gene Layer executing within a first set of Quantum Privacy Cells (QPCs) and configured to store atomic governance characteristics as Trust-Block-bound key-value records, each Gene Record being independently authorizable and revocable;

a DNA Layer executing within a second set of QPCs and configured to compose Gene Records into DNA Strands, each DNA Strand recording a governance composition under a Trust Block whose lineage chain references the included Gene Records;

a Genome Layer executing within a third set of QPCs and configured to compose DNA Strands into Genome Records, each Genome Record recording a participant-level or resource-level governance composition under a Trust Block whose lineage chain references the included DNA Strands;

an Inheritance Engine executing within a fourth set of QPCs and configured, upon creation of a new Genome Record from one or more source Genome Records, to propagate selected Gene Records and DNA Strands from the source Genome Records into the new Genome Record under Trust-Block-bound inheritance rules;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby governance characteristics are composable at three architectural granularities, inheritance is governed by Trust-Block-bound rules rather than operator discretion, and lineage of each governance characteristic is independently auditable from the Trust Block chain.

Claim 397. The system of claim 396, wherein the Inheritance Engine is configured to propagate Gene Records selectively in accordance with a Trust-Block-bound inheritance weight, said weight governing the probability or determinism of propagation as specified in the Trust Block.

Claim 398. The system of claim 396, wherein each Gene Record is independently revocable by a Trust-Block-bound revocation event without invalidating other Gene Records of the same DNA Strand.

Claim 399. The system of claim 396, wherein each DNA Strand additionally records a composition signature derived from the Gene Records included in the DNA Strand, said signature being verifiable independently of the DNA Strand's contents.

Claim 400. The system of claim 396, wherein each Genome Record additionally records a Genome signature derived from the DNA Strands included in the Genome Record, said Genome signature being computable by a deterministic-replay procedure.

Claim 401. The system of claim 396, wherein the Inheritance Engine emits, for each inheritance event, an Inheritance Receipt recording (a) the source Genome Records, (b) the propagated Gene Records and DNA Strands, and (c) the Trust-Block-bound inheritance rule that governed the event.

Claim 402. The system of claim 396, wherein the Trust Block of each Genome Record permits independent audit of the lineage chain without disclosure of unrelated Genome Records of the same or other participants.

Claim 403. The system of claim 396, wherein the three layers operate under a layer-isolation discipline such that Gene Layer QPCs do not have direct access to Genome Layer state and vice versa, except through Trust-Block-bound inter-layer message envelopes.

Claim 404. A computer-implemented method for Lamarckian inheritance of acquired governance characteristics, the method executing within a Quantum Privacy Network and comprising:

monitoring, by a Behavioral Witness executing within a first Quantum Privacy Cell, governance-relevant behavioral signals of a participant whose Genome Record is recorded in a Genome Layer;

upon a signal-aggregation condition specified in a Trust Block evaluating to true, computing a candidate Gene Record encoding an acquired governance characteristic derived from the aggregated behavioral signals;

presenting the candidate Gene Record to a Lamarckian Governance Gate executing within a second Quantum Privacy Cell, said Gate enforcing a Trust-Block-bound acceptance predicate;

upon acceptance by the Lamarckian Governance Gate, integrating the candidate Gene Record into the participant's Genome Record under a Trust Block whose lineage chain records (a) the prior Genome Record, (b) the aggregated behavioral signals, and (c) the acceptance event;

upon a subsequent reproduction event in which the participant's Genome Record contributes to a descendant Genome Record, propagating the integrated Gene Record to the descendant Genome Record in accordance with Trust-Block-bound inheritance rules;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby acquired governance characteristics are propagable to descendants under deterministic, Trust-Block-bound acceptance and inheritance rules, with the acceptance event recorded as part of the descendant Genome's lineage chain.

Claim 405. The system of claim 404, wherein the signal-aggregation condition specified in the Trust Block is configured as a multi-window stability requirement, requiring the behavioral signals to satisfy a stability predicate across a Trust-Block-specified minimum number of measurement windows.

Claim 406. The system of claim 404, wherein the Lamarckian Governance Gate is configured to evaluate the acceptance predicate by deterministic-replay procedure such that any participant authorized under the Trust Block may independently verify the acceptance decision.

Claim 407. The system of claim 404, wherein the acquired governance characteristic encoded in the candidate Gene Record is selected from a Trust-Block-specified Gene Schema enumerating permissible acquired characteristics.

Claim 408. The system of claim 404, wherein the integrated Gene Record additionally records a derivation receipt enabling the candidate Gene Record's derivation to be reproduced from the aggregated behavioral signals.

Claim 409. The system of claim 404, wherein the propagation to a descendant Genome Record is governed by a Trust-Block-bound dampening function, said dampening function reducing the inheritance weight of acquired Gene Records across successive generations.

Claim 410. The system of claim 404, wherein the Lamarckian Governance Gate is operable in two modes, a proposal mode emitting candidate Gene Records without integration and an integration mode integrating accepted candidates, with mode selection recorded in the Trust Block.

Claim 411. The system of claim 404, wherein, upon rejection by the Lamarckian Governance Gate, the candidate Gene Record is recorded on a Control Plane Log with the reason for rejection, without integration into the participant's Genome Record.

Claim 412. A multi-Genome governance architecture for a single participant, comprising:

a Genome Multiplexer executing within a first Quantum Privacy Cell and configured to maintain, for a single participant, a plurality of Genome Records each scoped to a distinct Privacy Domain or distinct governance context, with no Genome Record being implicitly merged into any other;

a Context Selector executing within a second Quantum Privacy Cell and configured, upon a participant operation, to select the operative Genome Record from the plurality by reference to a Trust-Block-bound context predicate associated with the operation;

a Cross-Genome Isolation Enforcer executing within a third Quantum Privacy Cell and configured to reject any read or write operation that would cause data of one Genome Record to be incorporated into a different Genome Record, except upon a Trust-Block-bound governance authorization expressly permitting the cross-Genome flow;

an Audit Producer configured to emit, for each Genome Record selection event, an audit record recording the context predicate evaluation without disclosure of unrelated Genome Records;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a single participant operates a plurality of governance Genomes under context-specific Trust-Block-bound rules, with cross-Genome flows blocked by default and auditable when permitted.

Claim 413. The system of claim 412, wherein the context predicate evaluated by the Context Selector is composed of dimension-specific predicates including at least one of: a Privacy Domain identifier, a Trust Criteria identifier, a Resource Pool scope, and an Accelerator scope.

Claim 414. The system of claim 412, wherein the Cross-Genome Isolation Enforcer additionally enforces directional isolation, permitting flows from Genome Record A to Genome Record B but not the reverse, where authorized by a Trust-Block-bound directional flow grant.

Claim 415. The system of claim 412, wherein the Audit Producer emits the audit record under a participant-scoped log scope such that aggregate context-selection patterns are not externally inferable.

Claim 416. The system of claim 412, wherein the Genome Multiplexer is configured to refuse creation of a new Genome Record that duplicates an existing context predicate, except upon a Trust-Block-bound override permitting context-predicate overlap with a deterministic precedence rule.

Claim 417. The system of claim 412, wherein each Genome Record additionally records a cross-Genome reference list enumerating other Genome Records of the same participant from which Trust-Block-bound cross-Genome flows have been authorized.

Claim 418. A governance-superposition system for DNA expression, comprising:

a DNA Strand recording a plurality of expression alternatives, each expression alternative being a Trust-Block-bound governance characteristic and being annotated with an activation predicate;

a Superposition Resolver executing within a first Quantum Privacy Cell and configured, upon a participant operation, to evaluate the activation predicate of each expression alternative under the operation's Trust-Block-bound context and to determine an operative expression alternative for the operation;

a Coherence Constraint Engine executing within a second Quantum Privacy Cell and configured to reject any DNA Strand in which two or more activation predicates evaluate to true simultaneously for a given context, except upon a Trust-Block-bound coherence override specifying a deterministic tiebreaker;

an Expression Audit Ledger configured to record each expression-resolution event with cryptographic linkage to the DNA Strand and the operative context predicate, such that historical expression resolutions are reconstructible from the ledger;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby a DNA Strand expresses different governance characteristics in different Trust-Block-bound contexts under deterministic resolution and coherence rules, with historical expression auditable from the Expression Audit Ledger.

Claim 419. The system of claim 418, wherein the activation predicate of each expression alternative is composed of dimension-specific sub-predicates evaluable from Trust-Block-bound context fields without disclosure of unrelated Trust-Block-bound state.

Claim 420. The system of claim 418, wherein the Coherence Constraint Engine is additionally configured to reject any DNA Strand in which no activation predicate evaluates to true for a Trust-Block-specified mandatory context, ensuring expression coverage of mandatory contexts.

Claim 421. The system of claim 418, wherein the Trust-Block-bound coherence override specifies a deterministic tiebreaker as one of: a static priority ordering, a recency-based ordering, and a Trust-Block-specified arithmetic combination of expression alternatives.

Claim 422. The system of claim 418, wherein the Expression Audit Ledger is written under a write-once cryptographic discipline preventing retroactive modification of recorded expression-resolution events.

Claim 423. The system of claim 418, wherein the Superposition Resolver is configured to emit a resolution receipt enabling deterministic replay of the resolution decision from the DNA Strand contents and the operative context predicate.

Claim 424. The system of claim 418, wherein, upon resolution of an expression alternative, the Superposition Resolver invalidates a cached resolution of any DNA Strand whose activation predicates reference the same context dimensions, ensuring fresh evaluation on subsequent operations.

Claim 425. A mitochondrial-analog operational-infrastructure inheritance system, comprising:

an Operational Genome Layer executing within a first Quantum Privacy Cell and configured to record operational-infrastructure characteristics of a participant or resource as Operational Gene Records distinct from governance Gene Records;

a Single-Line Inheritance Gate executing within a second Quantum Privacy Cell and configured to enforce a Trust-Block-bound inheritance rule constraining Operational Gene Records to propagate only from a designated Operational Lineage Parent at the time of reproduction;

an Operational Lineage Ledger configured to record each Operational Genome inheritance event with cryptographic linkage to the designated Operational Lineage Parent;

an Operational Genome Verifier configured to confirm, upon participant or resource activation, that the Operational Genome lineage chain traces to a Trust-Block-recognized origin, and to refuse activation otherwise;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby operational-infrastructure characteristics follow a deterministic single-line inheritance distinct from the bilateral inheritance of governance characteristics, with lineage verifiable from a dedicated Operational Lineage Ledger.

Claim 426. The system of claim 425, wherein the designated Operational Lineage Parent for a participant is one of: a participant Personal Privacy Network of which the participant is the founding participant, an Enterprise Privacy Network of which the participant is an enrolled participant, and a Sovereign Privacy Network of which the participant is a citizen-participant.

Claim 427. The system of claim 425, wherein the Operational Genome Verifier is configured to refuse activation if any Operational Gene Record in the lineage chain fails Proof-of-Trust verification or if the lineage chain references a Trust-Block-recognized origin that has been revoked.

Claim 428. The system of claim 425, wherein the Operational Lineage Ledger emits a lineage-verification receipt for each activation event, enabling external participants to independently verify Operational Genome integrity without disclosure of underlying Operational Gene Records.

Claim 429. The system of claim 425, wherein the Single-Line Inheritance Gate additionally enforces a non-substitution rule preventing modification of the designated Operational Lineage Parent after the initial inheritance event, except upon a Trust-Block-bound emergency override executed under a Privacy-Network-Exchange-wide governance procedure.

Claim 430. The system of claim 425, wherein the Operational Genome Layer is structurally isolated from the governance Genome Layer such that Operational Gene Records do not interfere with the bilateral inheritance of governance Gene Records.

Claim 431. The system of claim 396, further comprising a Premium Inheritance Engine configured, upon creation of a Resource Derivative under a Trust Block whose lineage chain references one or more upstream Resource Trust Blocks, to compute a Premium for the Resource Derivative as a Trust-Block-bound function of the Premiums of the upstream Resource Trust Blocks.

Claim 432. The system of claim 396, wherein the Premium Inheritance Engine is configured to enforce a Premium-non-decrease constraint such that the Premium of the Resource Derivative is not less than a Trust-Block-specified function of the Premiums of the upstream Resource Trust Blocks.

Claim 433. The system of claim 396, wherein the Premium Inheritance Engine emits a Premium-inheritance receipt recording the upstream Premiums and the computed downstream Premium, with the receipt cryptographically linked to the Resource Derivative's Trust Block.

Claim 434. The system of claim 396, further comprising a Multi-Layered Aggregation Engine configured to compute aggregate governance state across a plurality of Genome Records under a Trust-Block-bound aggregation procedure, said procedure producing an aggregate state observable to authorized observers without disclosure of underlying Genome Records.

Claim 435. The system of claim 396, wherein the Multi-Layered Aggregation Engine is configured to produce aggregate states at multiple layers of granularity simultaneously, including at least: a Privacy Domain layer, a Resource Pool layer, an Accelerator layer, and a Privacy Network Exchange-wide layer.

Claim 436. The system of claim 396, wherein the Multi-Layered Aggregation Engine emits aggregation receipts at each layer of granularity, with each receipt cryptographically linked to the Genome Records contributing to that layer's aggregate state.

Claim 437. The system of claim 396, wherein the Multi-Layered Aggregation Engine refuses production of an aggregate state at a given layer if the number of contributing Genome Records is below a Trust-Block-specified k-anonymity threshold, preventing inference of individual-Genome state from the aggregate.

Whereby Clause (Family-Level Structural Effect)

Whereby Family O closes the Quantum DNA/Genome Inheritance Architecture gap, binding 5 independent claims and 37 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Family P — Two-Parent PPN Creation + Polygenomic Resource Derivative Recombination

Claims 438–457 (20 claims: 2 independent + 18 dependent)

Family Introduction

Family P captures the Two-Parent Sexual Reproduction Model for Personal Privacy Network creation and the N-Parent Polygenomic Recombination Model for Resource Derivative creation. Functional cohesion: both

members address the creation of new Privacy Networks or Resource Derivatives via recombination of source Genomes. Architectural separation from Family O: while inheritance (O) governs how characteristics propagate, reproduction (P) governs how new entities are created from existing entities. Each independent claim recites concrete Trust-Block-bound recombination primitives, with the §22.7 canonical Wherein clause anchoring 2016 priority on the QPN infrastructure.

Claims List

Claim 438. A two-parent Personal Privacy Network reproduction system, comprising:

a Reproduction Initiator executing within a first Quantum Privacy Cell (QPC) and configured, upon a reproduction request specifying two source Genome Records, to evaluate a Trust-Block-bound consent predicate against authorization records of each source Genome Record;

a Recombination Engine executing within a second QPC and configured, upon consent predicate satisfaction, to construct a descendant Genome Record by combining selected Gene Records and DNA Strands from each source Genome Record under a Trust-Block-bound recombination rule;

a Personal Privacy Network Constructor executing within a third QPC and configured to instantiate, from the descendant Genome Record, a new Personal Privacy Network whose Privacy Domain is initialized with the recombined governance characteristics;

a Lineage Ledger configured to record the reproduction event with cryptographic linkage to both source Genome Records and the descendant Genome Record, such that the lineage of the descendant Personal Privacy Network is independently auditable;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby new Personal Privacy Networks are creatable from two source Genome Records under Trust-Block-bound consent and recombination rules, with descendant lineage independently auditable from the Lineage Ledger.

Claim 439. The system of claim 438, wherein the consent predicate is composed of consent records from each source Genome Record, each consent record being authorized by a participant identified by the source Genome Record's Trust Block.

Claim 440. The system of claim 438, wherein the Trust-Block-bound recombination rule specifies, for each Gene Record and each DNA Strand of each source Genome Record, a propagation outcome selected from: propagate-to-descendant, omit-from-descendant, and combine-with-counterpart-from-other-source.

Claim 441. The system of claim 438, wherein the Recombination Engine is configured to detect Gene Record conflicts in which both source Genome Records contribute Gene Records of the same Gene Schema field with incompatible values, and to resolve such conflicts under a Trust-Block-bound conflict-resolution procedure.

Claim 442. The system of claim 438, wherein the Personal Privacy Network Constructor instantiates the descendant Personal Privacy Network with a fresh Privacy Domain identifier distinct from the Privacy Domain identifiers of both source Personal Privacy Networks.

Claim 443. The system of claim 438, wherein the Lineage Ledger emits a reproduction receipt enabling deterministic replay of the recombination procedure from the recombination request and Trust-Block-bound recombination rule.

Claim 444. The system of claim 438, wherein the Reproduction Initiator is configured to refuse reproduction if either source Genome Record's Trust Block has a revoked status at the time of the reproduction request, with the refusal recorded on a Control Plane Log.

Claim 445. The system of claim 438, wherein the descendant Personal Privacy Network's governance composition is configured with an initial Lamarckian acceptance scope inherited from the source Genome Records but with a Trust-Block-specified initial dampening factor.

Claim 446. The system of claim 438, wherein the descendant Personal Privacy Network is initialized with operational-infrastructure characteristics inherited under a single-line inheritance procedure as specified in Family O for the mitochondrial-analog inheritance, with the Operational Lineage Parent selected from the two source Personal Privacy Networks per a Trust-Block-bound selection rule.

Claim 447. The system of claim 438, further comprising a Reproduction Audit Producer configured to emit, for each reproduction event, an audit record observable to authorized observers without disclosure of underlying Gene Records or DNA Strands of either source.

Claim 448. The system of claim 438, wherein the Personal Privacy Network Constructor verifies, prior to instantiation, that the descendant Genome Record satisfies a Trust-Block-bound minimum-governance predicate ensuring baseline governance coverage of the descendant.

Claim 449. A computer-implemented method for N-parent polygenomic Resource Derivative recombination, the method executing within a Quantum Privacy Network and comprising:

receiving, by a Recombination Engine executing within a first Quantum Privacy Cell, a recombination request specifying N source Resource Trust Blocks where N is at least two;

evaluating, by a Consent Aggregator executing within a second Quantum Privacy Cell, an N-fold Trust-Block-bound consent predicate against each of the N source Resource Trust Blocks, with the predicate including, for each source, a use-rights consent, an attribution-rights consent, and a Premium-rights consent;

upon satisfaction of the N-fold consent predicate, constructing, by the Recombination Engine, a Resource Derivative Trust Block whose lineage chain references each of the N source Resource Trust Blocks and whose governance composition is computed under a Trust-Block-bound recombination procedure;

computing, by a Premium Inheritance Engine executing within a third Quantum Privacy Cell, a Premium for the Resource Derivative as a Trust-Block-bound function of the Premiums of the N source Resource Trust Blocks;

recording, on a Lineage Ledger, the recombination event with cryptographic linkage to each of the N source Resource Trust Blocks and the Resource Derivative Trust Block;

wherein the system executes within a QPN-enabled infrastructure comprising at least one of: Quantum Privacy Cells (QPCs), Privacy Domains, Trust Criteria, Proof-of-Trust (PoT), Trust Blocks, or EasyAccess workflow threads.

whereby Resource Derivatives are creatable from arbitrarily many source Resource Trust Blocks under deterministic consent, recombination, and Premium inheritance procedures, with lineage and Premium computation independently auditable.

Claim 450. The system of claim 449, wherein the recombination procedure is composed of dimension-specific operators including a use-rights union operator, an attribution-rights merge operator, and a Premium-rights aggregation operator, each operator being independently Trust-Block-specified.

Claim 451. The system of claim 449, wherein the Consent Aggregator is configured to require an unanimous evaluation of the N-fold consent predicate, refusing recombination upon any single source's consent failure.

Claim 452. The system of claim 449, wherein the Trust-Block-bound Premium-inheritance function is configured to enforce a Premium-non-decrease constraint such that the Resource Derivative's Premium is at least the maximum of the source Resource Trust Blocks' Premiums, scaled by a Trust-Block-specified inheritance factor.

Claim 453. The system of claim 449, wherein the Lineage Ledger records, for each source Resource Trust Block, the contribution weight under which the source contributed to the Resource Derivative, with the contribution weight being computed deterministically from the recombination procedure.

Claim 454. The system of claim 449, wherein the Resource Derivative Trust Block additionally records a recombination receipt enabling deterministic replay of the recombination procedure from the source Trust Blocks and the Trust-Block-bound recombination rule.

Claim 455. The system of claim 449, wherein, upon revocation of any source Resource Trust Block subsequent to recombination, the Recombination Engine is configured to emit a revocation-propagation event to a Catalyst Network registry, with the propagation outcome specified by the Trust-Block-bound recombination rule.

Claim 456. The system of claim 449, wherein the Recombination Engine is operable in two modes: an aggregation mode constructing the Resource Derivative as a union of source contributions, and a transformation mode constructing the Resource Derivative as a Trust-Block-specified function over source contributions; mode selection is recorded in the Resource Derivative Trust Block.

Claim 457. The system of claim 449, wherein the Resource Derivative Trust Block is configured to be the source for further N-parent recombination events under the same procedure, enabling recursive Resource Derivative formation with lineage chains preserved at each depth.

Whereby Clause (Family-Level Structural Effect)

Whereby Family P closes the Two-Parent PPN Creation + Polygenomic Resource Derivative Recombination gap, binding 2 independent claims and 18 dependent claims to the foundational QPN primitives via the §22.7 canonical Wherein clause, and so inherits 2016 priority on the recited QPN infrastructure under the Wherein Clause Inheritance Mechanism.

Glossary of Terms

The following terms are used throughout this specification with the meanings indicated. Where a term is defined in U.S. Patent No. 12,316,610 B1 or in any of the WebShield Inc. provisional filings identified in the Cross-Reference to Related Applications, the term has the meaning given in that document; the entries below either summarize those meanings or define new terms introduced by the present invention.

Accelerator
A trust-verified accreditation and incubation environment within the QPN under which Exchange Networks and Resource Pools are launched. May be EP3-Managed or Private.

Accelerator Incentive & Investment Pool (AIIP)
The 20% Accelerator-level allocation of Exchange Tokens settled through the PNX that is reserved for Accelerator financial-investment and bootstrapping-contribution participants.

AI Governance Provisional
Refers to the November 18, 2025 WebShield Inc. provisional patent application titled 'AI Governance & Self-Funding Operation,' comprising 432 claims across 16 Claim Groups.

Catalyst Network
The contribution-capture, attribution, and reward layer of the QPN, comprising the user-side, network-side, AI-mediation, governance, settlement, and identity layers (Family K).

Contribution Ledger
On-ledger record-keeping component within Family D recording each accepted contribution as a Trust-Block-bound entry.

Cross-Verification Bundle
A Trust-Block-bound aggregate of third-party Witness attestations referencing a contribution by content-addressed identifier.

DORMANT QPC State
A Phase-0 compliance state in which a Manager-Originated QPC holds contribution accruals for a participant who has not yet self-activated; no economic function until activation.

EasyAccess workflow threads
Authorization workflow primitive defined in U.S. Patent No. 12,316,610 B1.

Exchange Root Token (ERT)
Perpetual economic claim on 7.5% of total PNX-settled value flows.

Genome
A participant-or-resource's full governance identity comprising multiple DNA strands (Family O).

Genome Recombination
Deterministic Genome composition from one or more parent Genomes per a configurable Recombination Policy (Family P).

Manager-Originated QPC
A QPC created by a Catalyst Network Manager without the originating participant's knowledge or authorization, used to record verified contributions for individuals who have not yet self-activated.

Mode Discriminator
Component within Family B classifying incoming signals among the Five Signal Input Modes prior to admission.

Monetization Uplift Multiple (MUM)
Per-Accelerator on-ledger metric tracking monetization performance relative to capital deployment (Family M).

Permanent Privacy Seal
Primitive triggering cryptographic destruction of the participant's content-encryption key for a target Contribution Ledger record (Family D).

Personal Privacy Network (PPN)
A participant's QPC-bound network of personal Resources and governance Genome strands.

Pipeline Attestation
Trust-Block-bound composite attestation emitted by the Three-Stage AI Evaluation Pipeline (Family C).

PPCS
Privacy-Preserving Compliance Screening, the activation-gate compliance screening applied on transition of a DORMANT QPC to ACTIVE state (Family I).

Premium
A multiplicative parameter applied during settlement allocation; the Premium Framework (Family J) defines 15 Premium dimensions.

Privacy Network Exchange (PNX)
The settlement, attribution, and exchange layer of the QPN.

Proof-of-Trust (PoT)
Cryptographic verification primitive defined in U.S. Patent No. 12,316,610 B1 enforcing Trust Criteria at every authorization hop.

Quantum DNA
Composite governance profile composed of multiple Quantum Genes (Family O).

Quantum Gene
Atomic governance trait (Family O).

Quantum Privacy Cell (QPC)
Foundational cryptographic-boundary primitive defined in U.S. Patent No. 12,316,610 B1.

Quantum Privacy Network (QPN)
The integrated trust-verified privacy-preserving infrastructure defined in U.S. Patent No. 12,316,610 B1.

Settlement Controller
Deterministic Exchange Token issuance enforcement component within Family G.

Settlement Ledger
On-ledger record of Exchange Token issuances (Family D).

Sidecar
The four-component user-side contribution capture pattern of Family A.

Three-Plane Architecture
Decomposition of Catalyst Network traffic into Data, Control, and Management planes with plane-specific QPCs.

Trust Block
Foundational authorization-bearing record primitive defined in U.S. Patent No. 12,316,610 B1.

Trust Criteria
Foundational authorization-policy primitive defined in U.S. Patent No. 12,316,610 B1.

Unified Trust Model (UTM)
Foundational cross-domain trust framework defined in U.S. Patent No. 12,316,610 B1.

Verification Envelope
Trust-Block-signed envelope emitted by the Sidecar's Verification Bridge for ingestion at the Catalyst Network boundary.

Wherein clause
The canonical claim-language primitive specified in QPN Context Primer v2.0.3 §22.7: 'wherein the system executes within a QPN-enabled infrastructure comprising at least one of: QPCs, Privacy Domains, Trust Criteria, Proof-of-Trust, Trust Blocks, or EasyAccess workflow threads.'

Witness
Third party providing Cross-Verification attestation under Family G.