Nick Clark · Published April 25, 2026 · PDF

Structural Provenance for Software Supply Chains: Binary and Firmware Identity Independent of SBOM Metadata

Software supply-chain integrity rests on metadata that travels beside the artifact and not within it: SBOM entries, signatures, and provenance attestations bind to file names, hashes, and build records that break the moment a binary is recompiled, repacked, or stripped. This application closes that gap by deriving identity from the internal byte structure of a binary or firmware object, so an artifact carries verifiable derivation and provenance on its own terms. It is built on the home inventive step Content Anchoring, disclosed in PCT International Application No. PCT/US26/28630.

What This Application Specifies

This application specifies how the structural content identity disclosed in PCT International Application No. PCT/US26/28630 applies to binary and firmware artifacts in a software supply chain. The core mechanism is a multi-axis variance vector extracted from the internal structure of an artifact rather than from its file name, storage path, signature, or build metadata. The disclosure describes a normalization procedure for binary object artifacts, including executable files, compiled archives, container payloads, and source code files, that maps the byte sequence of the artifact to a two-dimensional scalar field by reshaping the byte stream into a square or near-square matrix at a canonical resolution.

From that scalar field the pipeline computes per-cell statistics from sliding-window byte variance, byte-frequency entropy approximated as the variance of byte-frequency counts within the window, and structural-section profile values where the artifact carries a recognized container structure such as a portable executable, an executable and linkable format, or an archive index. The same multi-axis variance vector extraction pipeline that operates on images and audio then derives a unique identifier (UID) encoding the byte-level structural texture, sectional layout, and statistical regularity of the binary. Because the UID encodes a position in a continuous variance space, cosine similarity between two identifiers is directly computable. The disclosure states that this supports comparison across recompiled, repacked, or partially patched variants of the same underlying payload.

On top of identity, the disclosure specifies a provenance layer: multi-root lineage graphs that link a derivative UID to one or more parent UIDs with edge weights proportional to variance inheritance, a version history record ordered by registration timestamp, and a pre-release admissibility engine that evaluates a candidate artifact against cryptographically signed, versioned policy objects before any commitment. Applied to a supply chain, the artifact under evaluation is a binary or firmware image, and the commitment boundary is a release, a registry publication, or admission into a downstream build.

Why It Matters

Conventional supply-chain integrity references artifacts by static identifiers. A cryptographic hash pins one exact byte sequence; change a single bit through a recompilation, a different toolchain version, a reproducible-build mismatch, or a vendor repack, and the hash no longer matches even though the payload is materially the same. An SBOM entry names a component and a version, but that record is decoupled from the binary's structure and requires persistent external storage to stay attached. A code signature attests that some authority signed some bytes, yet signatures are severable: strip them, re-sign with a different key, or reconstruct the artifact, and

the binding is gone. The disclosure makes exactly this point about identifiers derived from storage location or transmission metadata, which are invalidated by mutation and replication and produce identity fragmentation across versions and derivatives, and about watermark-style and metadata-tag approaches whose signals are removable or externally stored.

A structurally derived identity behaves differently. Two builds of the same source that differ only in timestamps, padding, or symbol order produce variance vectors that remain close in slope space, so cosine similarity registers them as near-identical even when their hashes diverge. A repacked or partially patched variant lands within a measurable distance of its parent, which is precisely what a lineage edge needs. This is the property a supply chain wants and a hash cannot give: an identity that survives the benign transformations a binary undergoes between source and shelf, while still diverging predictably when the payload itself changes.

How It Composes With the Domain

A binary or firmware object enters the pipeline through the modality path the disclosure assigns to binary artifacts. Its byte stream is reshaped into a normalized scalar field, sliding-window byte variance and byte-frequency entropy are computed per cell, and recognized container structure (portable executable, executable and linkable format, or archive index) contributes structural-section profile values. The shared multi-axis extraction stage produces the variance vector and the UID. Nothing is embedded in the binary and no enrollment step is required; the identity is computed from the content itself.

That UID is quantized into a variance band and registered with anchor nodes governing the band. Registration places the artifact in a slope-band-indexed network where resolution is routed by variance proximity rather than by network address, which means a querying party that has only a candidate binary can compute its UID locally and ask the network what it matches. The UID resolution query protocol returns ranked

matches with cosine similarity scores, lineage annotations, and policy constraints, and it defines an orphan resolution mode that returns an empty match set when a candidate has no registered lineage in the governed corpus. For a supply chain, an orphan result is a signal in itself: a binary that claims to derive from a governed component but anchors to nothing.

Derivation is recorded in the lineage graph. When a build produces a new artifact from one or more sources, the new UID is linked to its parent UIDs with edge weights proportional to cosine similarity, and the directed edges, slope deltas, contribution weights, and anchor endorsement signatures are committed to the graph. The disclosure is explicit that these weights are structural signals and not legal determinations of authorship, but they are exactly the signals a supply-chain audit needs: a verifiable, content-derived account of what an artifact descended from. The version history record then captures successive builds as slope-delta-annotated transitions, so a firmware image and its patched successor sit on a traceable lineage rather than as two unrelated hashes.

At the release boundary, the pre-release admissibility engine evaluates the candidate binary against signed, versioned policy objects before commitment, where a commitment includes public release, marketplace publication, or cross-platform provenance anchoring. A policy object can declare admissible categories, restricted classes, jurisdictional constraints, similarity tolerance thresholds, override authorities, and escalation paths. The structural similarity evaluator compares the candidate against a governed corpus and rejects, regenerates, or escalates when similarity to an excluded reference exceeds the policy threshold. Admissibility decisions are reproducible and auditable: given the UID, the structural signatures, and the policy object version, an authorized party can replay the evaluation and reach the same result, which is the property the disclosure contrasts with opaque post-hoc moderation.

What This Enables

Concretely, this composition supports a set of supply-chain capabilities grounded in the disclosed mechanisms:

- Metadata-independent component matching. Because identity is structural, a binary can be recognized as a variant of a known component across recompilation, repacking, and partial patching, so SBOM-style component inventories can be reconciled against what the binary actually is rather than against the label it carries.
- Derivation-grade provenance. The multi-root lineage graph and version history record give an artifact a verifiable account of its parents and predecessors, expressed as content-derived edges with similarity-proportional weights and anchor signatures.
- Orphan and drift detection. The orphan resolution mode flags artifacts with no registered lineage in the governed corpus, and the version history's slope deltas surface a successor that has drifted further from its declared parent than policy allows.
- Pre-release gating at the commitment boundary. The admissibility engine interposes a reproducible, policy-driven evaluation between a candidate binary and its release, so an artifact that matches an excluded reference (for example, a known-bad payload class indexed in the exclusion corpus) is rendered non-committable before publication.
- Client-side and disconnected evaluation. The disclosure's client-side architecture computes the UID and evaluates similarity against a locally cached, cryptographically signed corpus fragment and policy object without transmitting the raw artifact, which suits build agents and air-gapped or intermittently connected pipelines where signed corpus fragments and policy objects are verifiable offline.
- Bulk resolution for ingestion pipelines. The protocol's bulk mode routes a batch of candidate UIDs to their bands in parallel, fitting registry-scale ingestion and continuous-integration flows that must resolve many artifacts per request.

Boundary Conditions

The disclosure supports binary-object identity through one normalization recipe and a stated comparison property; it does not assert that structural identity replaces cryptographic signing, reproducible-build verification, or formal SBOM standards. Those remain complementary, and the disclosure positions its identity as derived from content rather than embedded in or attached to it, not as a substitute for authority-based trust.

The comparison guarantee is bounded. The disclosure describes stability under controlled transformations within defined thresholds and predictable divergence as variance-shifting mutations occur; it states that binary-object UIDs support cosine-similarity comparison across recompiled, repacked, or partially patched variants, but it does not claim invariance to arbitrary obfuscation, deliberate adversarial restructuring of a binary, or transformations that materially alter byte-level structure. Slope proximity is a structural signal subject to policy-declared continuity and similarity thresholds, and lineage edges are candidate links confirmed under anchor quorum, not authorship or ownership findings.

No performance numbers are claimed here because the specification states none for binary throughput or detection rates. The variance-band thresholds, the canonical resolutions, and the hashing scales recited above are the disclosure's own parameters for the general pipeline; the byte-domain normalization reuses that pipeline. Domain framing such as SBOM practice, code signing, and registry publication is external context describing where the mechanism plugs in, not additional disclosed technology.

Disclosure Scope

The technical mechanisms described in this article are disclosed in PCT International Application No. PCT/US26/28630, including the multi-axis variance vector, the binary-object normalization that reshapes a byte stream into a scalar field and computes

sliding-window byte variance, byte-frequency entropy, and structural-section profiles for portable executable, executable and linkable format, and archive structures, the cosine-similarity comparison across recompiled, repacked, and partially patched variants, the multi-root lineage and version history records, the UID resolution query protocol with its orphan mode, the pre-release admissibility engine evaluating signed versioned policy objects, and the client-side and bulk-resolution architectures. All references in this article to software supply-chain practice, SBOM frameworks, code signing, container and firmware distribution, and registry publication are external domain and regulatory context provided to show how the disclosed invention composes with an existing field; they are not claims of the application and do not extend the disclosure. Claims to binary-object support are limited to the extent the specification discloses them, and nothing here should be read to assert structural identity beyond the byte-domain normalization and comparison properties the specification recites.

Content Anchoring (/content-anchoring)

All 40 steps → (/inventive-steps)

Computable identity for media. Provenance from structural variance.

PCT/US26/28630 (/patents/pct-us26-28630)

PRIMARY TECHNICAL DISCLOSURE

- Content Anchoring: Computable Identity for Media That Changes (/articles/content-anchoring-computable-identity-for-media-that-changes)

SECONDARY TECHNICAL

- Multi-Axis Variance Vector Extraction: Nine Dimensions of Structural Content Identity (/articles/content-anchoring/variance-vector)
- Quadrant Decomposition: Spatial Sub-Region Fingerprinting for Partial Similarity Detection (/articles/content-anchoring/quadrant-decomposition)
- 320-Bit UID Construction: Multi-Segment Hashing for Negligible Collision Probability (/articles/content-anchoring/uid-construction)

- Structure Signature: Background-Invariant Matching Through Gradient-Only Descriptors (/articles/content-anchoring/structure-signature)
- Constellation Signature: Geometry-Invariant Matching Across Crop, Scale, and Occlusion (/articles/content-anchoring/constellation-signature)
- Five-Band Variance Classification: Content Routing by Structural Complexity (/articles/content-anchoring/variance-classification)
- Variance Saturation-Governed Cache Eviction: UID Density Replacing Static TTL (/articles/content-anchoring/cache-eviction)
- Multi-Root Composite Lineage Graphs: Provenance Through Variance Vector Similarity (/articles/content-anchoring/composite-lineage)
- Multi-Modal Content Identity: Unified Pipeline Across Image, Audio, Text, and Video (/articles/content-anchoring/multi-modal-identity)
- Rights-Grade Pre-Release Admissibility: Policy Evaluation Before Content Commitment (/articles/content-anchoring/pre-release-admissibility)
- Training Corpus Governance: Verifiable Lineage From Training Data to Model (/articles/content-anchoring/training-corpus-governance)
- Consultation Event Logging: Deterministic Records of Every Generation Reference (/articles/content-anchoring/consultation-logging)
- Model Output Provenance Fingerprint: Structural Proximity Without Model Access (/articles/content-anchoring/output-provenance)
- Creator Attribution and Compensation Routing: Payment From Consultation Lineage (/articles/content-anchoring/creator-attribution)
- Adversarial Robustness and Deepfake Detection: Content Identity as Detection Substrate (/articles/content-anchoring/adversarial-robustness)
- Client-Side Execution Architecture: Privacy-Preserving Variance Computation on Device (/articles/content-anchoring/client-side-execution)
- UID Resolution Query Protocol: Distributed Lookup Across Anchor Node Networks (/articles/content-anchoring/uid-resolution)
- Orientation Canonicalization: Rotation-Invariant Processing Through Gradient Normalization (/articles/content-anchoring/orientation-canonicalization)
- Cross-Band Resolution Pathfinding: Traversal Between Variance Bands Under Mutation (/articles/content-anchoring/cross-band-resolution)
- Identity by Position: Media as a Third Navigable Space (/articles/content-anchoring/identity-by-position)

APPLICATIONS · GENERAL

- Forbidden-Content Blocking at Upload and Generation Time: Pre-Release Exclusion Against Signed Policy (/articles/content-anchoring/forbidden-content-blocking)
- **Structural Provenance for Software Supply Chains: Binary and Firmware Identity Independent of SBOM Metadata (/articles/content-anchoring/software-supply-chain-provenance)**
- Rights-Grade Generative AI: How to Pay Creators, Exclude Forbidden Content, and Prevent Infringement Before Release (/articles/content-anchoring/rights-grade-generative-ai)
- Deepfake Detection by Structural Provenance: Verifying Synthetic Media Without Watermarks (/articles/content-anchoring/deepfake-provenance)
- Creator Economy Attribution Without Platform Intermediaries (/articles/content-anchoring/creator-attribution-economy)
- Verifying Source Photos and Video in the Newsroom: Content Anchoring for Journalism (/articles/content-anchoring/journalism-verification)
- Detecting Image Manipulation and Proving Figure Provenance in Research Publications (/articles/content-anchoring/academic-research-integrity)
- Content Anchoring for Legal Evidence Chains (/articles/content-anchoring/legal-evidence-chain)
- Content Anchoring for Insurance Claims Evidence (/articles/content-anchoring/insurance-claims-evidence)
- Content Anchoring for Real Estate Documentation (/articles/content-anchoring/real-estate-documentation)
- Art Authentication and Provenance Verification with Content Anchoring (/articles/content-anchoring/art-authentication)

APPLICATIONS · SPECIFIC

- C2PA Attaches Provenance to Content. The Content Itself Has No Identity. (/articles/content-anchoring/c2pa)
- Google SynthID Watermarks AI Output. Watermarks Are Not Identity. (/articles/content-anchoring/google-synthid)
- Shutterstock Tracks Licensed Media. The Media Itself Cannot Prove Its Own Identity. (/articles/content-anchoring/shutterstock)
- Spotify Tracks Every Stream. The Music Itself Has No Computable Identity. (/articles/content-anchoring/spotify)
- Getty Images Built the World's Largest Licensed Image Library. Image Identity Still Depends on Metadata. (/articles/content-anchoring/getty-images)
- Adobe Stock Integrates Licensed Content Into Creative Workflows. Content Identity Is Still External. (/articles/content-anchoring/adobe-stock)

- YouTube Content ID Matches Audio and Video. The Content Has No Intrinsic Identity. (/articles/content-anchoring/youtube-content-id)
- Audible Magic Identifies Audio Content. The Audio Has No Self-Identifying Properties. (/articles/content-anchoring/audible-magic)
- Digimarc Embeds Invisible Watermarks. The Watermark Is Added, Not Intrinsic. (/articles/content-anchoring/digimarc)
- Irdeto Protects Digital Content Through DRM. The Protection Is Applied, Not Intrinsic. (/articles/content-anchoring/irdeto)

Content Anchoring overview → (/content-anchoring)