SBOM Attestation for Software Health

by Nick Clark | Published April 25, 2026

What It Specifies

Each unit declares its SBOM: software components, versions, and hashes. The architecture verifies the running software against the declared SBOM; deviations enter as credentialed monitoring events.

SBOM attestation events carry: unit identity, declared SBOM, observed SBOM, deviation analysis, monitoring authority signature. Downstream operations admit the events against admissibility.

Why It Matters Structurally

Software-integrity monitoring without SBOM faces structural ambiguity. The expected software composition is implicit; deviations cannot be evaluated against explicit expectation.

SBOM attestation produces structural specificity. The expected composition is declared; deviations are evaluated against the declaration; monitoring events carry structured semantics.

How It Composes With Mesh Operation

The architecture defines the SBOM-declaration format, the deviation-evaluation primitives, and the event recording. Implementations apply the architecture; monitoring operations proceed within the framework.

SBOM composes with other features. Cross-mesh SBOM federation, byzantine-robust SBOM evaluation, and dispute mechanism for SBOM disputes all build on the SBOM primitive.

What This Enables

Defense software integrity gains structurally-supported attestation. Civilian critical-infrastructure software integrity gains the same.

The architecture also supports SBOM evolution. As SBOM standards (SPDX, CycloneDX) mature, attestation protocols update through governance procedures.