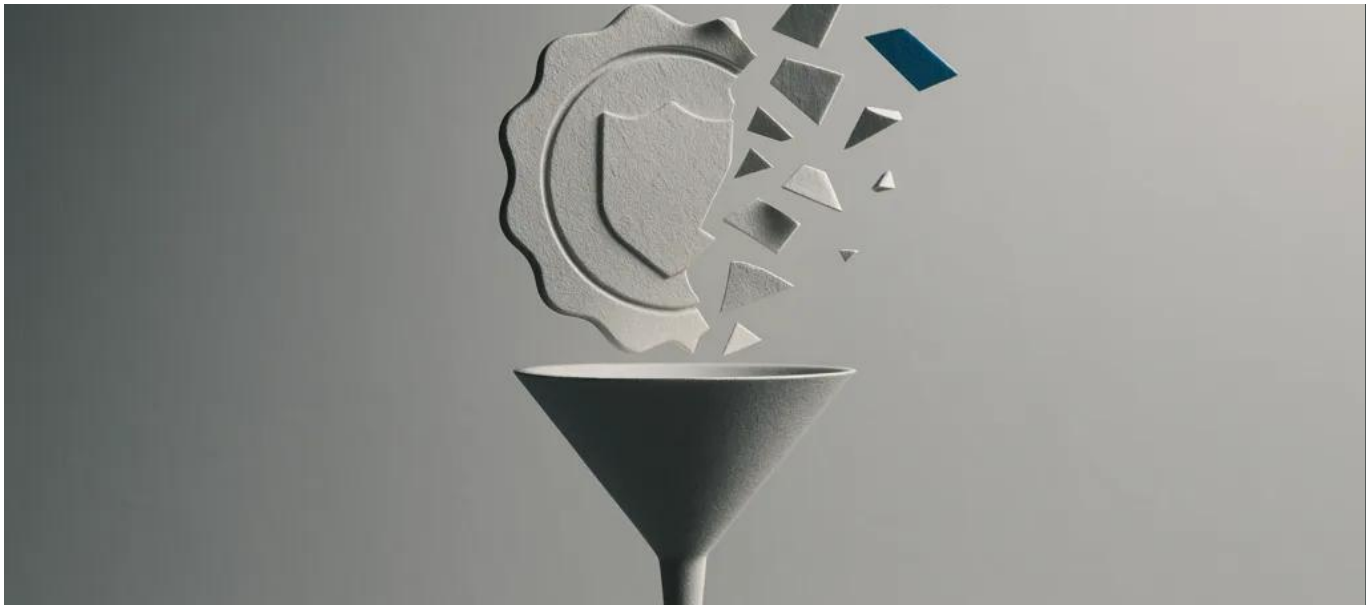

# C2PA, Explained: How Content Credentials Actually Work, and Why 5 of 6 Platforms Strip Them

Five of six social platforms deleted the C2PA manifest from uploaded images entirely. Only Facebook kept anything, and only on certain creator flows.

THE SHORT VERSION

C2PA signs a cryptographic manifest into the file itself, so anything that re-encodes the file can drop it. As of 14 May 2026, 54 products from 31 companies had passed conformance testing, but an AFIP metadata stripping study (15 January 2025, updated 19 July 2026) found five of six platforms strip manifests completely on upload. Conformance is a producer property. Survival is a delivery property, and nobody certifies that.

## C2PA is the only marking mechanism already deployed at scale

There is no prescribed technical mechanism for content marking in the text of the rules, which in practice means the industry converges on whatever is already shipping. That is C2PA. OpenAI became a Conforming Generator Product and joined the steering committee on 19 May 2026, and the deployment guidance from 3 August 2026 lists 120+ additional companies expressing interest in conformance on top of the 54 products that have passed.

One caveat worth holding: the ISO version, ISO/IEC 22144 "Authenticity of information: Content credentials", based on C2PA 2.1, was still at stage 30.99 as ISO/CD 22144 as of 1 August 2026. So the de facto standard is a consortium spec, and anyone writing "ISO-certified provenance" into a procurement document is writing fiction.

I have written before about how the Omnibus shifts the watermarking dates. The mechanics below are what you have to ship regardless of which date lands.

## A manifest is a code-signing certificate that travels inside the pixels

That is the closest analogy for an engineer coming from another domain: the thing being signed is the image data, and the signature rides inside the container rather than next to it.

Concretely, a C2PA manifest bundles one claim, one claim signature and one or more assertions, encoded as JUMBF boxes with CBOR payloads. Assertions are the statements (who captured this, what model generated it, what edits were applied). The claim is a JUMBF superbox labeled c2pa.claim.v2 carrying a CBOR payload that follows the deterministic encoding rules in RFC 8949 clause 4.2.1. Deterministic encoding matters because you are hashing that payload: two serializations of the same map must produce identical bytes, or verification fails for no good reason.

The signature is COSE_Sign1, with X.509 certificates carried in the COSE headers. Verification walks that chain against a trust list. All of this is in the C2PA 2.4 Content

Credentials spec.

1 **Assert**
The producing tool writes assertions: capture device, generative model, edit history. Each assertion gets hashed.

2 **Bind to the bytes**
A hard binding hashes the raw pixel data or the container segments. This is what ties the claim to this specific file and not a lookalike.

3 **Claim and sign**
The claim collects the assertion hashes plus the hard binding, serializes as deterministic CBOR, and gets signed with COSE_Sign1. The X.509 chain rides in the COSE headers.

4 **Embed as JUMBF**
The whole manifest goes into the file as JUMBF boxes. Which is also the step where the entire thing becomes deletable by anything that rewrites the container.

5 **Optionally soft-bind**
Register a fingerprint or invisible watermark so the manifest can be recovered from a repository after the file is transformed. C2PA calls this a Durable Content Credential.

## Image pipelines drop JUMBF boxes because they do not recognise them

Mostly not out of malice. Platforms re-encode uploads aggressively: resize for multiple viewport tiers, transcode to a cheaper codec, strip metadata. A JUMBF box full of CBOR and X.509 chains is, to an image pipeline written years ago, unrecognized ancillary data, and the default behaviour of most encoders is to drop what they do not understand.

The AFIP study uploaded signed images to six platforms and found manifests fully stripped in five. Facebook showed partial preservation, and only on certain creator flows. That is the gap between having a conforming generator and having a verifiable image in front of a reader.

Hard bindings make this worse. Because the binding hashes the raw pixels, any crop, overlay, resize or re-encode invalidates verification, including entirely benign edits. So even a platform that preserves the manifest byte for byte while resizing the image hands the reader a credential that fails validation. The spec's own security considerations say this plainly.

> ! **The catch with soft bindings**
> Soft bindings answer hard-binding fragility: fingerprint or watermark the content, then re-find the manifest in a repository after transformation. They are also vulnerable to collision attacks, per C2PA guidance and a 2026 CEUR workshop paper on C2PA and eIDAS 2.0. A fingerprint that two different images can match is a fingerprint that can attribute someone else's provenance to your content.

## Releases 2.3 and 2.4 are both about surviving delivery

C2PA 2.3, published December 2025, extended provenance to live streaming through CMAF segment signing. C2PA 2.4 adds a Soft Binding API and crJSON, a compact representation for constrained environments such as camera firmware and streaming segments. Both are admissions that the original assumption (one file, one manifest, end to end) does not survive contact with a real CDN.

On the infrastructure side, Cloudflare became the first major CDN to support Content Credentials in February 2025 and offers a provenance-passthrough mode that preserves manifests across delivery. At the capture end, Google's Pixel Camera app reached C2PA Assurance Level 2, the highest security tier, using hardware-backed keys in the Titan M2 chip. Both details come from a C2PA analysis published 7 September 2026.

> WHERE I LAND
>
> *My take: the conformance number (54 products, 31 companies) measures the wrong end of the pipeline for anyone whose content reaches users through social platforms. Conformance certifies that a tool produces a valid manifest. It says nothing about whether the manifest arrives. I think the compliance-relevant metric will be soft-binding recovery rate against a manifest repository rather than conformance status, because that is the only number that survives a platform re-encode. I could be wrong about the timing, but I expect the Soft Binding API in 2.4 to become the part of the spec everyone actually integrates, and* ***the hard***

***binding to end up as a nice-to-have for unmodified originals***.

## Three roles, three different things to build

If you produce AI content, conformance testing is the cheap part, and OpenAI's approach of embedding SynthID alongside C2PA credentials is the pattern to copy: a watermark that survives re-encoding plus a manifest that carries the detail. One mechanism per failure mode.

If you distribute content, the question is whether your image pipeline preserves unknown ancillary boxes, and the honest answer for most older pipelines is no. Cloudflare's passthrough mode solves it at the edge; everything upstream of the CDN is still yours to fix.

If you verify content, plan for the case where no manifest is present, because in five of six platform paths that is the normal case. A verification UI that shows "no credential" as a red flag will mark almost the entire internet as suspicious, which trains users to ignore it. **Absence of a credential has to read as absence of information.**

And treat ISO/CD 22144 as what it is, a draft still sitting at stage 30.99. Build against C2PA 2.4 and the conformance program, and write your contracts so a later ISO edition does not require re-signing your archive.

WRITTEN BY

Artur Markus

Software engineer in Basel building AI infrastructure and agentic systems, with a background in pharmaceutical automation, GxP compliance, and validation. I help teams turn AI from a chatbot into a reliable operational layer.

Book a meeting →