Coalition for Content Provenance and Authenticity

Use of Content Credentials to identity synthetic and non-synthetic content

1.0, 2026-07-30: Release

Table of Contents

Chapter 1. Abstract

This white paper clarifies how use of C2PA's Content Credentials can better ensure that end users, people who are consuming media assets, are better able to discern when those assets have been created or modified using generative AI. It does so by describing how the C2PA's Technical Specification for Content Credentials may be used to establish tamper-evident, verifiable, machine-readable provenance labels for assets created or modified by Generative AI systems. The paper examines three key components:

- the manifest structures and assertion types relevant to AI-generated content, specifically the `c2pa.actions.v2` assertion and the `c2pa.ai-disclosure` assertion;
- the use of Regions of Interest (ROI) to locate AI-modified portions of an asset across image, document, audio, and video formats;
- the use of `inputTo` ingredient assertions to record the prompts and reference assets that constitute an AI generation recipe.

Guidance on an end-to-end labelling workflow is provided throughout, with worked examples drawn from multiple asset formats.

IMPORTANT	Systems and products that utilize Generative AI technology vary from completely stand-alone environments to those where it is embedded into a traditional authoring environment. As such, the information available to the system about what it is starting with and how the edits are being applied, will vary significantly. Accordingly, the following should be considered as some common scenarios and not necessary the only possible cases. Developers should use the information here as guidance and not as normative requirements.

Chapter 2. Introduction

2.1. Background and Motivation

The rapid proliferation of Generative Artificial Intelligence (Generative AI) tools capable of producing photorealistic images, synthetic audio, and convincing video has created a critical requirement for interoperable mechanisms to declare and verify the provenance of digital media assets. Without such interoperable mechanisms, consumers and platforms are unable to reliably distinguish content captured by sensors from content synthesised by AI models — or to identify the extent to which a human-originated asset has subsequently been modified by AI-assisted post-processing.

The **C2PA Technical Specification for Content Credentials** international standard addresses this challenge. It was developed by the Coalition for Content Provenance and Authenticity (C2PA) and defines a comprehensive framework for recording the provenance of digital assets through cryptographically signed manifests containing structured assertions. This enables an end-to-end, policy-driven labelling and verification ecosystem that is both human-readable and machine-actionable.

2.2. Scope of This Document

This white paper covers the data structures and assertion types defined by the C2PA specification for Content Credentials that are relevant to Generative AI content. It presents a suggested end-to-end implementation workflow for labelling assets that are created from scratch by Generative AI or modified using AI techniques, including guidance on using Regions of Interest (ROI) to record the extents of AI-applied modifications across various media formats.

The paper also covers the use of `inputTo` ingredient assertions to capture the prompts, reference images, and other inputs that constitute an AI generation recipe, and illustrates all of these concepts with worked examples of manifests and assertions drawn from multiple asset formats.

This document is informative in nature. In all matters of conformance, the normative requirements of the C2PA specification take precedence.

2.3. Intended Audience

This document is intended for:

- Developers and implementers of Generative AI tools who wish to integrate Content Credential generation

into their workflows.

- Digital asset creators and publishers seeking to label their AI-generated or AI-modified content with verifiable provenance information.
- Platform operators and content distributors who want to verify the provenance of incoming assets, possibly to enforce policy decisions based on that provenance.

2.4. Normative References

Identifier	Document
Content Credentials	C2PA Technical Specification (v2.4), Coalition for Content Provenance and Authenticity
IPTC Digital Source Types	IPTC Digital Source Type NewsCodes
JUMBF	ISO 19566-5:2023, "Information technology — JPEG Universal Metadata Box Format (JUMBF)"
RFC 3161	Adams, C. et al., "Internet X.509 PKI Time-Stamp Protocol (TSP)", RFC 3161, August 2001

Chapter 3. The C2PA Provenance Model

The C2PA Technical Specification defines a provenance model centred on three core artifacts [C2PA §2.3]:

Assertion

A data structure representing a statement — either created by the signer or gathered at claim generation time — concerning an asset. Assertions cover areas such as asset creation, edit actions, capture device details, bindings to content, and metadata.

Claim

A digitally signed, tamper-evident data structure that references a set of assertions and the information necessary to represent the content binding [C2PA §2.3.2].

Claim Signature

The digital signature on the claim created using the private key owned by a signer [C2PA §2.3.3].

These artifacts are combined into a **C2PA Manifest** and stored within a **C2PA Manifest Store**, which in non-technical usage is referred to as a **Content Credential** [C2PA §2.3.6].

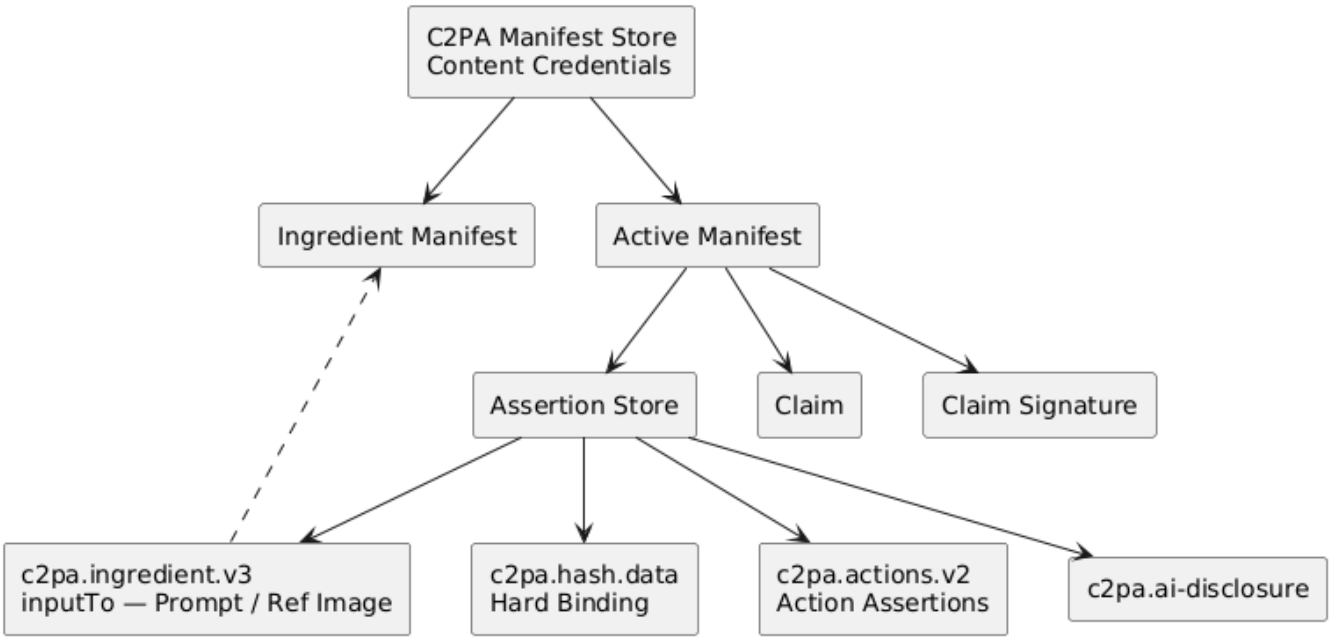

Figure 1. C2PA Manifest Store Structure (derived from C2PA §11.1.4)

The Manifest Store is serialised using the **JPEG Universal Metadata Box Format (JUMBF)** [C2PA §11.1], which provides a hierarchical, referenceable, typed container format which can be embedded within numerous asset file formats including JPEG, PDF, video and audio containers. Additionally, the Manifest Store

can be stored as a detached "sidecar file", which can be stored alongside the asset, in the cloud, on a blockchain or wherever desired. This allows for use with asset formats that do not support embedding or where embedding is not desired.

Chapter 4. Generative AI Labelling

4.1. Standard Manifest Requirements

The primary unit of provenance in C2PA is the **Standard Manifest** [C2PA §11.2.2]. Every standard manifest shall contain exactly one hard binding (to content) assertion; `c2pa.hash.data`, `c2pa.hash.boxes`, or one of the other supported options - selected according to asset type [C2PA §11.2.2]. It shall include either a `c2pa.created` or `c2pa.opened` action (known as an inception action) in its actions assertion [C2PA §15.10.1.2].

> **NOTE** An inception action is the first action in the set of actions assertions in a C2PA Manifest, representing how this asset "came into existence".

4.2. The `digitalSourceType` Field

The `digitalSourceType` field on an action is the primary machine-readable signal for classifying AI-generated content [C2PA §18.15.4.5]. Additionally, there are use cases where the `digitalSourceType` field will be present on an ingredient assertion, as described in Section 4.6, “Use of `digitalSourceType` fields on Ingredient Assertions Without a Manifest”.

For Generative AI content, the following values are normatively applicable:

Table 1. IPTC `digitalSourceType` Values for Generative AI Content (C2PA §18.15.4.5)

URI	Interpretation
`http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia`	The asset was created entirely by a trained AI/ML model without the direct use of pre-existing media assets as primary inputs. This is the canonical value for pure Generative AI output.
`http://cv.iptc.org/newscodes/digitalsourcetype/compositeWithTrainedAlgorithmicMedia`	The asset has undergone augmentation, correction or enhancement using a Generative AI model (e.g., a human-originated photograph combined with AI-generated imagery).
`http://cv.iptc.org/newscodes/digitalsourcetype/compositeSynthetic`	The asset is a mix or composite of several elements, at least one of which is Generative AI.

When identifying content produced by a human, the `http://cv.iptc.org/newscodes/digitalsourcetype/humanEdits` value is used.

4.3. Newly generated Generative AI Assets

For a newly generated Generative AI asset, the appropriate inception action is `c2pa.created`.

> **NOTE** Newly generated means that the system is creating something completely new, ala "File→New" in an interactive authoring tool or a "Text→Image" in an AI system.

A standard manifest for such a newly generated Generative AI asset shall contain, at minimum:

A `c2pa.created` action specifying: . The creation timestamp in the `when` field [C2PA §18.15.4.3]. . The `softwareAgent` identifying the Generative AI tool or system [C2PA §18.15.4.4]. . One of the `digitalSourceType` values from the IPTC vocabulary that signal the use of Generative AI [C2PA §18.15.4.5], listed in Section 4.2, “The `digitalSourceType` Field”.

> **NOTE** If additional information is required, consider using custom-named fields in the `parameters`, e.g., `com.fabrikam.assetId`.

A crJSON-based example of such an assertion is provided in `c2pa.actions.v2` Assertion for a Purely AI-Generated Asset.

`c2pa.actions.v2` *Assertion for a Purely AI-Generated Asset*

```
"c2pa.actions.v2": {
  "actions": [
    {
      "action": "c2pa.created",
      "when": "2026-03-15T14:30:00Z",
      "softwareAgent": { "name": "Fabrikam Text2Image" "version": "3.5" },
      "digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia"
    }
  ]
}
```

A claim generator can provide additional information about the model(s) used as part of the AI generation process by including a `c2pa.ai-disclosure` assertion and referencing it from the `relatedAssertions` field of the `c2pa.created` action, as described in Section 4.7, “The `c2pa.ai-disclosure` Assertion”. An example can be found here

`c2pa.actions.v2` *Assertion for a Purely AI-Generated Asset with* `c2pa.ai-disclosure` *Reference*

```
"c2pa.actions.v2": {
```

```
  "actions": [
    {
      "action": "c2pa.created",
      "when": "2026-03-15T14:30:00Z",
      "softwareAgent": { "name": "Fabrikam AI", "version": "2.0" },
      "digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia",
      "parameters": {
        "relatedAssertions": [
          {
            "url": "self#jumbf=c2pa.assertions/c2pa.ai-disclosure",
            "hash": "b64'ikvBZJWWN5l69j1+/GwgnCJ8DAvlHKSWiF0ZTnWKAUY=",
            "pad": "b64'"
          }
        ]
      }
    }
  ]
}

"c2pa.ai-disclosure": {
  "modelType": "diffusion",
  "modelName": "Fabrikam AI 2.0",
  "modelIdentifier": "https://fabrikam.example/models/fabrikam-ai-2.0",
}
```

4.4. Generative AI Modification of Assets

For an asset (whether human-created or from Generative AI) that was subsequently modified using Generative AI, the appropriate inception action for the manifest covering the modification is `c2pa.opened`, followed by one or more additional actions such as `c2pa.edited`, `c2pa.placed` or other applicable action types. At least one of those subsequent actions or associated ingredients shall have the `digitalSourceType` field that is populated according to the IPTC vocabulary, as described in Section 4.2, “The `digitalSourceType` Field”, along with the other fields (`when` & `softwareAgent`) described in the previous section.

For an asset that has undergone modifications, the manifest associated with the modification shall begin with a `c2pa.opened` action referencing the parent ingredient (`parentOf`), followed by one or more additional actions describing the changes.

A crJSON example of an actions assertion representing starting with an existing asset and then modifying it with Generative AI is provided in `c2pa.actions.v2` Assertion for an Asset Modified by Generative AI.

`c2pa.actions.v2` *Assertion for an Asset Modified by Generative AI*

```
{
  "action": "c2pa.opened",
```

```
  "parameters": {
    "ingredients": [
      {
        "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3",
        "hash": "b64'SdSVV2KWCssld25s1HaJd1alYQ4hICNuMR4Yl9icL50=",
        "pad": "b64'"
      }
    ]
  }
},
{
  "action": "c2pa.enhanced",
  "when": "2026-04-01T10:00:00Z",
  "softwareAgent": { "name": "Fabrikam AI", "version": "2.0" },
  "digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia"
}
```

A claim generator can provide additional information about the model(s) used as part of the AI generation process by including a `c2pa.ai-disclosure` assertion and referencing it from the `relatedAssertions` field of the specified action(s), as described in Section 4.7, "The `c2pa.ai-disclosure` Assertion".

In the case where the Generative AI content is added indirectly, such as the placement/addition of a Generative AI-based asset, that would be represented through the use of a `c2pa.placed` action and its associated `componentOf` ingredient. A crJSON example can be found at `c2pa.actions.v2` Assertion for an Asset with a placed Generative AI.

`c2pa.actions.v2` *Assertion for an Asset with a placed Generative AI*

```
{
  "action": "c2pa.opened",
  "parameters": {
    "ingredients": [
      {
        "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3",
        "hash": "b64'SdSVV2KWCssld25s1HaJd1alYQ4hICNuMR4Yl9icL50=",
        "pad": "b64'"
      }
    ]
  }
},
{
  "action": "c2pa.placed",
  "when": "2026-04-01T10:00:00Z",
  "softwareAgent": { "name": "Fabrikam Editor", "version": "2.0" },
  "parameters": {
    "ingredients": [
      {
```

```
        "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3__1",
        "hash": "b64'U2RTVlYyS1dDc3NsZDI1czFIYUpkMWFsWVE0aElDTnVNUjRZbDlpY0w1MA=",
        "pad": "b64'"
      }
    ]
  },
  "digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia"
}
```

In the example at `c2pa.actions.v2` Assertion for an Asset with a placed Generative AI, the ingredient doesn't have an associated Manifest, and so it uses the approach documented at Section 4.6, "Use of `digitalSourceType` fields on Ingredient Assertions Without a Manifest". Most commonly, the asset will have an associated Manifest which will be referenced from the `activeManifest` field of the action.

4.5. Human Edits on a Generative AI asset

Another use case is where the asset was originally created by Generative AI system and is subsequently modified by a human, utilize the `humanEdits` source type [C2PA §18.15.4.2, §18.15.4.5].

An example of such a manifest is provided in `c2pa.actions.v2` Assertion for an Asset Created by Generative AI and Subsequently Modified by a Human.

`c2pa.actions.v2` *Assertion for an Asset Created by Generative AI and Subsequently Modified by a Human*

```
{
  "actions": [
    {
      "action": "c2pa.opened",
      "when": "2026-04-01T10:00:00Z",
      "softwareAgent": { "name": "Fabrikam Studio", "version": "3.1.0" },
      "parameters": {
        "ingredients": [{
          "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3",
          "alg": "sha256",
          "hash": "<base64-encoded-hash>"
        }]
      }
    },
    {
      "action": "c2pa.edited",
      "when": "2026-04-01T10:01:30Z",
      "softwareAgent": { "name": "Fabrikam Studio", "version": "3.1.0" },
      "digitalSourceType": "http://cv.iptc.org/newscodes/digitalsourcetype/humanEdits",
    }
  ]
}
```

4.6. Use of `digitalSourceType` fields on Ingredient Assertions Without a Manifest

A significant use case arises when an ingredient asset does not carry its own C2PA Manifest. In this situation the claim generator is unable to include an `activeManifest` reference in the ingredient assertion. However if the claim generator knows that it is a generative image, such as in the case where the claim generator invoked the Generative AI tool to produce the asset but did not receive a manifest from that tool, it may still wish to label the ingredient as generative. Another possible scenario is where a Claim Generator is pruning, optimizing or summarizing a branch of the provenance graph, and needs to express key AI labelling information about what happened inside that branch.

The `c2pa.ingredient.v3` schema accommodates these uses cases through an optional `digitalSourceType` field on the ingredient assertion itself [C2PA §18.16.6] as shown in Ingredient Assertion with `digitalSourceType` for an asset w/o provenance (C2PA §18.16.6).

Ingredient Assertion with `digitalSourceType` for an asset w/o provenance (C2PA §18.16.6)

```
{
  "dc:title": "Generative Background Photo",
  "dc:format": "image/jpeg",
  "relationship": "componentOf",
  "digitalSourceType":
    "https://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia",
  "metadata": {
    "dateTime": "2025-10-01T12:00:00Z"
  },
  "description":
    "Generative image, without a Content Credential, used as background layer; source type
declared by the claim signer as generative AI."
}
```

NOTE

The `digitalSourceType` field on an ingredient assertion cannot be combined with an `activeManifest` field — the presence of both constitutes a validation failure [C2PA §15.11.3.2]. Its purpose is specifically to characterise ingredients that lack their own provenance record, preserving the descriptive integrity of the overall provenance chain. This mechanism is particularly relevant in Generative AI workflows where a model may incorporate style-reference assets or training exemplars that predate this ecosystem.

To incorporate a date-time value onto an ingredient assertion, the `dateTime` field of the ingredient assertion's `metadata` field can be used.

4.7. The `c2pa.ai-disclosure` Assertion

C2PA §18.28 introduces a dedicated **AI Disclosure assertion** (`c2pa.ai-disclosure`) to provide machine-readable transparency information beyond that carried by `digitalSourceType`. This assertion provides three critical provenance primitives not previously captured in the ecosystem [C2PA §18.29]:

- **Model provenance** — full disclosure of the AI model(s) used.
- **Scientific routing metadata** — content domain characterisation using the arXiv taxonomy.
- **Human oversight level** — a structured signal of the degree of human involvement in the generation pipeline.

The `modelType` field is mandatory; all other fields are optional but should be populated where information is available [C2PA §18.28.1.1]:

`c2pa.ai-disclosure` *Assertion (schema from C2PA §18.29.1)*

```
{
  "modelType": "c2pa.types.model.huggingface.transformers",
  "modelName": "ExampleDiffusion v4",
  "modelIdentifier": "https://example.org/models/exampleddiffusion-v4",
  "contentProfile": {
    "humanOversightLevel": "prompt_guided"
  },
  "scientificDomain": ["cs.AI", "cs.CV"]
}
```

A `c2pa.ai-disclosure` assertion does not stand on its own, but instead is referenced using the `relatedAssertion` field of either an action assertion (for the primary asset) or an ingredient assertion (for included/referenced assets). An example of what that might look like for an action assertion is shown in Relating a `c2pa.ai-disclosure` Assertion to an Action Assertion.

Relating a `c2pa.ai-disclosure` *Assertion to an Action Assertion*

```
{
  "action": "c2pa.created",
  "description": "Created by Fabrikam Generative AI.",
  "digitalSourceType":
"http://cv.iptc.org/newscodes/digitalsourcetype/trainedAlgorithmicMedia",
  "parameters": {
    "relatedAssertions": [
      {
        "url": "self#jumbf=c2pa.assertions/c2pa.ai-disclosure",
        "hash": "b64'ikvBZJWWN5l69j1+/GwgnCJ8DAvlHKSWiF0ZTnWKAUY=",
        "pad": "b64'"
      }
```

```
    ]
  }
}
```

4.8. Identifying AI-Modified Regions Using Regions of Interest

4.8.1. Overview

When Generative AI is applied not to the whole asset but to a specific portion of it — for example, AI modification of a face in an image, AI-generated paragraphs inserted into a document, or AI enhancement of a segment of audio — it is both possible and recommended to record the precise extent of that modification. C2PA §18.15.4.6 provides for this through the **`changes` field** of an action, whose value is an array of `region-map` objects as defined in C2PA §18.2.

Some content authoring and editing tools do not operate within clearly defined regions, so precise region data cannot always be recorded. In such cases, the `changes` field may be omitted without invalidating the provenance record.

> **NOTE**
>
> Regions of Interest are not just for identifying AI-generated content; they can be used to specify the scope of any action or assertion that applies to a sub-region of an asset. However, they are particularly valuable for AI provenance as they enable precise labelling of which parts of an asset were modified by AI or a human.

The region of interest mechanism supports four orthogonal range types that may be combined within a single region definition [C2PA §18.2.2.1]:

Table 2. Region of Interest Range Types (C2PA §18.2.2)

Range Type	Description
`spatial`	A physical area of the asset described by a shape (rectangle, circle, or polygon) in pixel or percentage coordinates [C2PA §18.2.2.2].
`temporal`	A time period described using Normal Play Time (NPT, per RFC 2326) or wall-clock timestamps [C2PA §18.2.2.3].
`frame`	A range of frames or pages identified by ordinal number, suitable for documents and animations [C2PA §18.2.2.4].

Range Type	Description
`textual`	A range of content identified by W3C Web Annotation fragment selectors, supporting structured document formats [C2PA §18.2.2.5].

4.8.2. Use in Action Assertions

The `changes` field should be included on an action whenever the action was specific to only a portion of the asset and the authoring tool can record precise region data. For Generative AI modification, this allows a manifest consumer to present a precise spatial, temporal, or textual overlay indicating which regions of an asset were AI-generated or AI-modified.

The `regionOfInterest` field in assertion metadata [C2PA §18.3.8] additionally enables any assertion — not only an action — to be associated with a region. When an ingredient assertion is relevant only to a specific sub-region of the ingredient asset, the ingredient assertion's metadata should include a `regionOfInterest` field specifying that sub-region [C2PA §18.16.13].

4.8.3. Examples

4.8.3.1. Example: JPEG Image — Rectangular Spatial Region

A common Generative AI use case is the replacement of a region of interest within an image, such as removing unwanted content. The following example records a `c2pa.edited` action with a rectangular spatial region of interest in pixel coordinates describing the area where AI modifications were applied:

Spatial Region of Interest for AI modification of a JPEG Image

```
{
  "actions": [
    {
      "action": "c2pa.opened",
      "when": "2025-04-01T10:00:00Z",
      "softwareAgent": { "name": "ExampleGenAI Studio", "version": "3.1.0" },
      "parameters": {
        "ingredients": [{
          "url": "self#jumbf=c2pa.assertions/c2pa.ingredient.v3",
          "alg": "sha256",
          "hash": "<base64-encoded-hash>"
        }]
      }
    },
    {
      "action": "c2pa.edited",
      "when": "2025-04-01T10:01:30Z",
```

```
      "softwareAgent": { "name": "ExampleGenAI Studio", "version": "3.1.0" },
      "digitalSourceType":

"http://cv.iptc.org/newscodes/digitalsourcetype/compositeWithTrainedAlgorithmicMedia",
      "changes": [
        {
          "region": [
            {
              "type": "spatial",
              "shape": {
                "type": "rectangle",
                "unit": "pixel",
                "origin": { "x": 320.0, "y": 85.0 },
                "width": 415.0,
                "height": 390.0
              }
            }
          ],
          "name": "AI-Modified Region",
          "identifier": "modified-region-001",
          "description":
            "Region modified using generative AI to remove unwanted content; original
content was a tree that was removed from the background."
        }
      ]
    }
  ]
}
```

A validator or display application can use the spatial region data to overlay a visible indicator on the image at coordinates `(320, 85)` with dimensions `415 x 390` pixels, clearly signalling to users that this area was AI-generated.

4.8.3.2. Example: PNG Image — Polygon Region (AI Object Placement)

Where the AI-generated region is not rectangular, a polygon may be more appropriate. The following example records an AI-generated object placed within an irregularly shaped region of a PNG image:

Polygon Spatial Region for AI-Generated Object Placement in a PNG Image

```
{
  "region": [
    {
      "type": "spatial",
      "shape": {
        "type": "polygon",
        "unit": "percent",
        "origin": { "x": 42.0, "y": 18.0 },
        "vertices": [
```

```
          { "x": 42.0, "y": 18.0 },
          { "x": 67.0, "y": 15.0 },
          { "x": 78.0, "y": 44.0 },
          { "x": 63.0, "y": 61.0 },
          { "x": 38.0, "y": 55.0 }
        ]
      }
    }
  ],
  "name": "AI-Generated Vehicle",
  "description":
    "Vehicle in foreground generated by AI and composited into scene"
}
```

NOTE

Using `"unit": "percent"` makes the region definition resolution-independent, which is preferable where the asset may be re-encoded or scaled downstream.

4.8.3.3. Example: PDF Document — Tagged PDF Textual Region

In a PDF document, Generative AI may be used to generate or revise specific paragraphs or sections. For **tagged PDFs**, where the document structure tree is present, textual regions are identified using W3C fragment selector paths into the tag tree [C2PA §18.2.2.5, §18.2.4]:

Textual Region of Interest for an AI-Written Section in a Tagged PDF

```
{
  "actions": [
    {
      "action": "c2pa.edited",
      "when": "2025-04-02T09:15:00Z",
      "softwareAgent": {
        "name": "ExampleAI Writer",
        "version": "1.5.0"
      },
      "digitalSourceType":

"http://cv.iptc.org/newscodes/digitalsourcetype/compositeWithTrainedAlgorithmicMedia",
      "changes": [
        {
          "region": [
            {
              "type": "textual",
              "text": {
                "selectors": [
                  [
                    {
                      "selector": { "fragment": "path=/Document/Sect[1]/P[3]" },
                      "end": { "fragment": "path=/Document/Sect[1]/P[5]" }
```

```
                    }
                  ]
                ]
              }
            }
          ],
          "name": "AI-Generated Executive Summary",
          "description":
            "Paragraphs 3 to 5 of Section 1 were generated by an AI writing assistant"
        }
      ]
    }
  ]
}
```

The selector pair `path=/Document/Sect[1]/P[3]` (start) to `path=/Document/Sect[1]/P[5]` (end) addresses paragraphs 3 through 5 of the first section as represented in the PDF tag tree.

4.8.3.4. Example: PDF Document — Non-Tagged PDF (Page and Rectangle)

For **non-tagged PDFs**, where no structure tree is available, the textual region may be expressed using a W3C fragment selector consisting of the page number and rectangle specification [C2PA §18.2.4]:

Textual Region in a Non-Tagged PDF Using Page and Rectangle (C2PA §18.2.4)

```
{
  "region": [
    {
      "type": "textual",
      "text": {
        "selectors": [
          [
            {
              "selector": {
                "fragment": "page=3,rect=72,600,540,750"
              }
            }
          ]
        ]
      }
    }
  ],
  "description": "AI-generated content on page 3, upper body text area"
}
```

The `page=3,rect=72,600,540,750` fragment identifies page 3 with a bounding rectangle expressed as `left,bottom,right,top` in PDF user space units (points, with origin at the lower-left corner of the page).

4.8.3.5. Example: Video — Combined Temporal and Spatial Region

For video assets, AI modification may occur over a defined time window within a defined spatial area — for example, an AI-generated face replacement applied to a specific speaker during a specific segment. Combined temporal and spatial ranges may be used [C2PA §18.2.4]:

Combined Temporal and Spatial Region for AI Face Replacement in Video (C2PA §18.2.4)

```
{
  "region": [
    {
      "type": "temporal",
      "time": {
        "type": "npt",
        "start": "32.5",
        "end": "47.0"
      }
    },
    {
      "type": "spatial",
      "shape": {
        "type": "rectangle",
        "unit": "percent",
        "origin": { "x": 15.0, "y": 5.0 },
        "width": 30.0,
        "height": 45.0
      }
    }
  ],
  "name": "AI Face Replacement – Speaker A",
  "description":
    "Face of Speaker A replaced by AI during seconds 32.5 to 47.0 of the recording"
}
```

Two other range types are useful for video.

frame-based

This is useful when frame numbers are more meaningful than time offsets, such as in the case of an animation or a video with variable frame rate.

item-based

When it is important to identify a single "track" of the video that was modified, such as only the caption or the audio. It can also be used to identify more semantic content items, such as subject’s face in the above example. See [C2PA §18.2.2.6] for more detail.

4.8.3.6. Example: Audio Temporal Region for AI-Replaced Vocal Phrase

For audio and music assets, AI modification may apply only to a defined time range. For example, AI enhancement of a lead vocal, stem separation cleanup, noise reduction, or generative extension of a musical passage. A temporal range identifies the affected segment [C2PA §18.2.2.3, §18.2.2.6]:

Temporal Region for AI-Replaced Vocal Phrase in Audio (C2PA §18.2.4)

```
{
  "region": [
    {
      "type": "temporal",
      "time": {
        "type": "npt",
        "start": "68.00",
        "end": "72.00"
      }
    }
  ],
  "name": "AI-Replaced Vocal Phrase",
  "description": "Vocal phrase replaced by AI during seconds 68.00 to 72.00 of the
recording"
}
---
```

Given that most audio is usually "multi-track", it is useful to also use item-based regions to identify a single "track" of the audio that was modified.

Chapter 5. Capturing the AI Generation Recipe Using `inputTo` Ingredients

5.1. Overview

One piece of information that can be useful as part of AI provenance is the ability to record not just *what* was generated, but *how* it was instructed. The generation recipe — comprising the text prompts, negative prompts, seed values, reference images, style exemplars, and any other inputs provided to the AI system — constitutes essential provenance information for understanding the origin of an AI-generated asset.

> **NOTE** These pieces of information may contain personally identifiable information (PII) or other sensitive data. They should be treated accordingly.

C2PA provides a direct mechanism for this through the **`inputTo` relationship** value of the `c2pa.ingredient.v3` assertion [C2PA §18.16.3, Table 10]:

inputTo

> This ingredient was used as input to a computational process, such as an AI/ML model, that led to the creation or modification of this asset.

The `inputTo` relationship is explicitly designed to represent inputs to AI/ML inference processes, including both training dataset references and inference-time inputs [C2PA §18.16.1]. Unlike `parentOf` (which requires `c2pa.opened`) or `componentOf` (which requires `c2pa.placed`), the `inputTo` relationship carries no mandatory associated action in the `c2pa.actions.v2` assertion. The AI inference process itself produces the output; the inputs are contextual provenance rather than compositional derivation.

> **NOTE** During ingredient validation, a validator encountering an ingredient that lacks an `activeManifest` field will record `ingredient.unknownProvenance` rather than treating the absence as a validation failure [C2PA §15.11.3.3]. For `inputTo` ingredients, this reflects the reality that many AI inputs — raw text prompts, hyperparameter sets — do not have associated C2PA Manifests.

5.2. Representing Text Prompts as Ingredients

A text prompt used to instruct a Generative AI model may be embedded directly within the manifest using an

Embedded Data assertion referenced from the ingredient's `data` field [C2PA §18.12, §18.16.6].

> **NOTE**
>
> A text prompt may contain personally identifiable information (PII) or other sensitive data. When embedding prompt text, claim generators should ensure that they have obtained appropriate consent from the prompt author and that the inclusion of such data complies with applicable privacy regulations and policies.

`inputTo` Ingredient Assertion for a Text Prompt shows an `inputTo` ingredient assertion for a text prompt, with the `data` field referencing an embedded data assertion containing the prompt text, and the `dataTypes` field characterising the content as a generator prompt [C2PA §18.16.5].

`inputTo` *Ingredient Assertion for a Text Prompt*

```
{
  "dc:title": "Generation Prompt",
  "dc:format": "text/plain",
  "relationship": "inputTo",
  "description":
    "Text prompt provided to the diffusion model for this generation",
  "data": {
    "url": "self#jumbf=c2pa.assertions/c2pa.embedded-data",
    "alg": "sha256",
    "hash": "<base64-encoded-hash-of-embedded-data>"
  },
  "dataTypes": [
    {
      "type": "c2pa.types.generator.prompt"
    }
  ]
}
```

The `dataTypes` field provides typed characterisation of the embedded data, enabling manifest consumers to identify prompt data without parsing its content [C2PA §18.16.5].

5.3. Representing Reference Images as Ingredients

Where a reference image (e.g., a style reference, a seed image, etc.) is used as input to the AI inference process, it should be recorded as an `inputTo` ingredient. If the reference image carries its own C2PA Manifest, the `activeManifest` and `claimSignature` fields shall be included to preserve the provenance chain. `inputTo` Ingredient Assertion for a Manifested Reference Image (C2PA §18.16.3, §18.16.6) shows what such an ingredient assertion might look like.

`inputTo` *Ingredient Assertion for a Manifested Reference Image (C2PA §18.16.3, §18.16.6)*

```
{
  "dc:title": "Style Reference — Watercolour Exemplar",
  "dc:format": "image/jpeg",
  "relationship": "inputTo",
  "description":
    "Reference image used to guide the diffusion model's visual style",
  "activeManifest": {
    "url":
      "self#jumbf=c2pa/urn:c2pa:A3F7-0012-4E9B-BBCD-001122334455/c2pa.claim.v2",
    "alg": "sha256",
    "hash": "<base64-encoded-hash>"
  },
  "claimSignature": {
    "url":
      "self#jumbf=c2pa/urn:c2pa:A3F7-0012-4E9B-BBCD-001122334455/c2pa.signature",
    "alg": "sha256",
    "hash": "<base64-encoded-hash>"
  }
}
```

If the reference image does not carry a C2PA Manifest, the `digitalSourceType` field shall be used on the ingredient assertion to characterise its source, as described in Section 4.2, “The `digitalSourceType` Field”.

5.4. Representing Multiple Inputs — A Complete Recipe

Common Generative AI workflows can involve multiple distinct inputs: a positive prompt, a negative prompt, seed and hyperparameter values, a style reference image, and possibly a mask. Each of these should be recorded as a separate `inputTo` ingredient assertion, labelled according to the standard multiple-instance labelling rules:

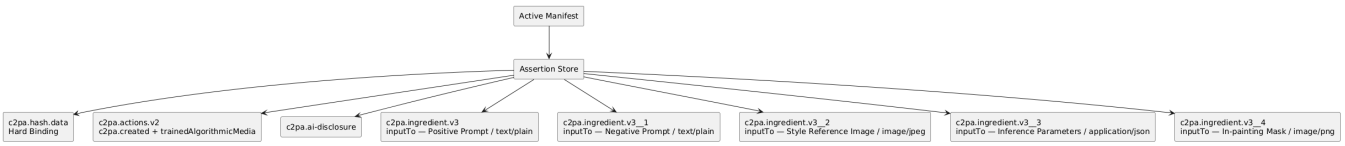

Figure 2. Multiple `inputTo` Ingredient Assertions Representing a Generation Recipe

5.5. Representing Non-Media Inputs — Inference Hyperparameters

Scalar values and structured parameters (e.g., random seed, guidance scale, number of inference steps) that do not have a natural file-based representation may be embedded as JSON in an embedded data assertion and referenced as an `inputTo` ingredient:

`inputTo` *Ingredient for Inference Hyperparameters*

```
{
  "dc:title": "Inference Parameters",
  "dc:format": "application/json",
  "relationship": "inputTo",
  "description": "Inference hyperparameters used for this generation run",
  "data": {
    "url": "self#jumbf=c2pa.assertions/c2pa.embedded-data__1",
    "alg": "sha256",
    "hash": "<base64-encoded-hash>"
  },
  "dataTypes": [
    {
      "type": "c2pa.types.generator.parameters"
    }
  ]
}
```

The corresponding embedded data assertion would carry a JSON object such as:

Embedded Inference Parameters JSON

```
{
  "seed": 2847391052,
  "guidance_scale": 7.5,
  "num_inference_steps": 50,
  "scheduler": "DPMSolverMultistepScheduler"
}
```

5.6. Training Data References

In addition to inference-time inputs, the `inputTo` relationship may be used to reference the dataset used to train the AI model. C2PA §9.2.5 describes the `c2pa.hash.collection.data` assertion specifically for use with collections such as AI/ML training datasets, noting that this assertion can describe each folder of a training dataset for an AI/ML model [C2PA §9.2.5]. This enables verifiable, hash-bound association between the generated asset and the training corpus.

Chapter 6. Conclusion

The C2PA Technical Specification for Content Credentials provides mechanisms for the verifiable labelling of Generative AI assets with its foundational data structures — manifests, assertions, claims, and cryptographic bindings.

The key integration points described in this paper are:

1. **Standard Manifests** which mandate the `c2pa.created` inception action with `digitalSourceType` as the primary machine-readable AI creation classification signal along with `c2pa.opened` for subsequent modifications. It also accommodates `digitalSourceType` on ingredient assertions for un-manifested source assets.
2. The **`c2pa.ai-disclosure` assertion**, which provides granular model provenance, scientific domain classification, and human oversight level signals complementing the `digitalSourceType` classification.
3. **Regions of Interest** in the `changes` field of action assertions, enabling precise spatial, temporal, and textual localisation of AI-modified portions of an asset across image, document, and video formats.
4. **`inputTo` ingredient assertions** (`c2pa.ingredient.v3`), which provide a normative, verifiable mechanism for capturing the complete AI generation recipe — prompts, reference assets, seed values, and model parameters — as cryptographically bound provenance data.

This specification for Content Credentials provides the technical foundation for a trustworthy, scalable, and interoperable ecosystem for Generative AI provenance disclosure.

Appendix A: Abbreviations

Abbreviation	Expansion
AI	Artificial Intelligence
C2PA	Coalition for Content Provenance and Authenticity
IPTC	International Press Telecommunications Council
JUMBF	JPEG Universal Metadata Box Format
ML	Machine Learning
NPT	Normal Play Time (RFC 2326)
ROI	Region of Interest