Creator Assertions Working Group

Eric Scouten · Identity Standards Architect · Adobe

CAWG identity assertion working session

Content Authenticity 101 lightning round

C2PA data model lightning round

CAWG identity assertion data model

The interesting challenges ...

Looking for a note-taker

Slides will be posted online

Help us (OK, me) remember interesting discussion

Who’s Who?

Content Authenticity Initiative

C2PA

Coalition for Content Provenance and Authenticity

Creator Assertions Working Group

Who’s Who?

Content Authenticity Initiative

Outreach · Advocacy · Education

** also name of Adobe’s team*

C2PA

Coalition for Content Provenance and Authenticity

Technical Standards: **What / How**

Creator Assertions Working Group

Technical Standards: **Who**

Scope for today

Content Authenticity Initiative

C2PA Coalition for Content Provenance and Authenticity

Creator Assertions Working Group

Review:
ericscouten.dev/cai-101

Identity assertion is a CAWG project

C2PA data model (lightning round edition)

C2PA data model

Overview

An **asset** is any piece of digital media that we wish to describe.

It is described by a **C2PA Manifest**. Each asset in C2PA has an *active manifest* which describes the current asset.

That C2PA Manifest may refer to *ingredient manifests* when earlier content is incorporated.

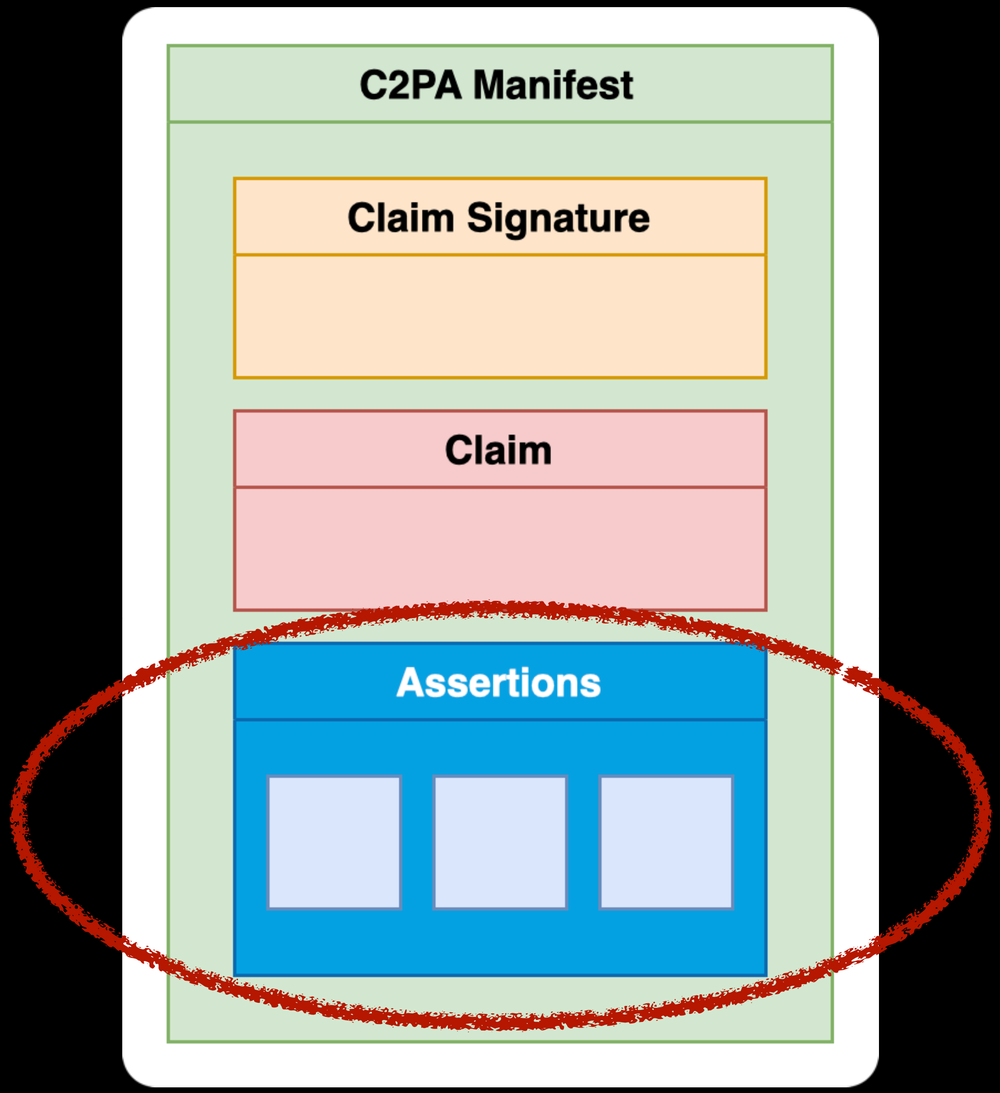
The collection of C2PA Manifests is referred to as a **C2PA Manifest Store.**

C2PA data model

Assertions

Assertions are opt-in statements that cover areas such as:

- hard binding to asset's binary content *(required – provides tamper evidence)*
- capture device details
- identity of the content creator(s) *(hello, CAWG!)*
- edit actions
- thumbnail of the content
- other content (ingredients) that were incorporated into this content



CAWG identity assertion data model

Identity assertion

Status

November 2023:

Initial private drafts for review

February 2024:

Transition to public working draft
under Community Specification License process

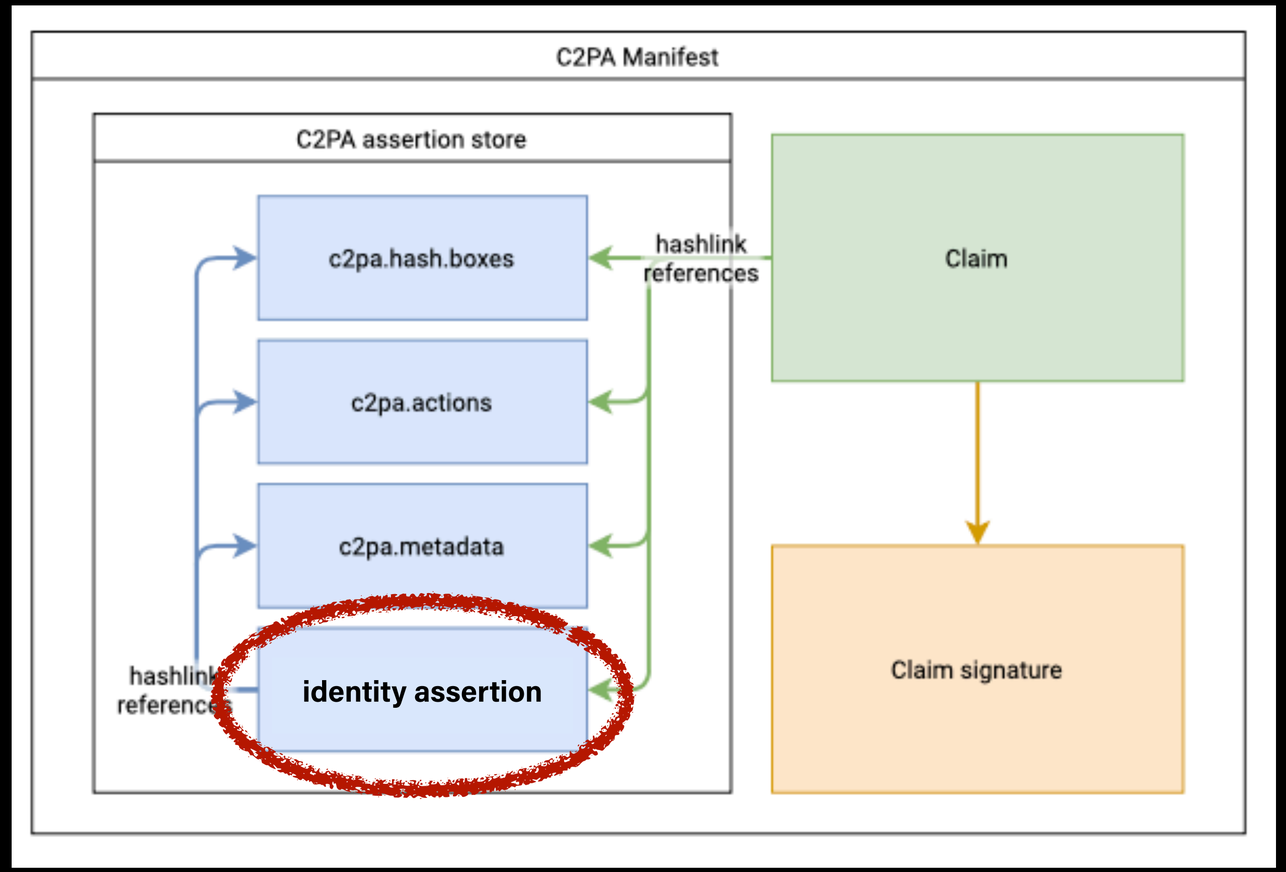
Weekly meetings with ~30 regular contributors

Identity assertion

Overview

Identity assertion allows a credential holder to sign a data structure we call **signer_payload**, which contains:

- Tamper-evident references to one or more other assertions in the same C2PA Manifest (including hard-binding assertion)
- Role of credential subject with regard to the content
- Other items TBD ...

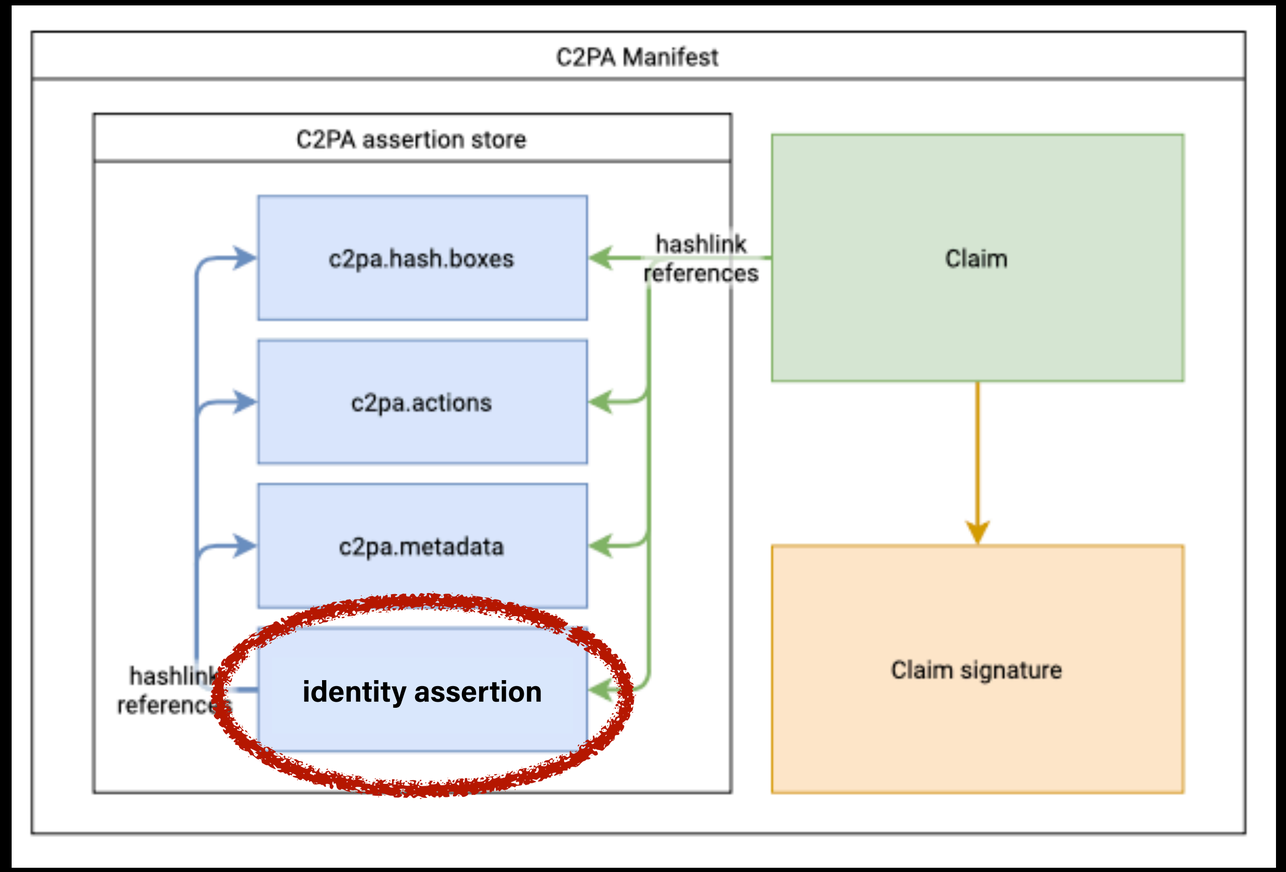


Identity assertion

Overview

New and separate trust signal over and above the C2PA claim generator signal.

Typically meant to indicate **subject's authorization or active participation** in production of the asset.



Identity assertion

CDDL

```
identity = {
  "signer_payload": $signer-payload-map,          ; content to be signed by credential holder
  "sig_type": tstr .size (1..max-tstr-length),    ; identifies the data type of the signature
  "signature": bstr,                              ; byte string of the signature
  "pad1": bstr,
  ? "pad2": bstr,
}

signer-payload-map = {
  "referenced_assertions": [1* $hashed-uri-map],
    ; more coming soon ... (credential holder's role in relation to asset, etc.)
}
```

Identity assertion

CBOR-Diag

```
{
  "signer_payload": {
    "referenced_assertions": [
      { "url": "…/c2pa.assertions/c2pa.hash.data", "hash": b64'U9Gyz05…' },
      { "url": "…/c2pa.assertions/c2pa.thumbnail.claim.jpeg", "hash": b64'G5hfJwY
      { "url": "…/c2pa.assertions/c2pa.ingredient.v2", "hash": b64'Yzag4o5…' }
    ]
  },
  "sig_type": "cawg.w3c.vc", <— based on type of credential presented
  "signature": b64'....',    <— varies based on sig_type
  "pad1": b64'....',
  "pad2": b64'....'
}
```

Credential types currently supported in draft

X.509 Certificate

```
sig_type: "cawg.x509.cose"
signature: (COSE signature over signer_payload)
```

W3C Verifiable Credential (or VP)

```
sig_type: "cawg.w3c.vc"
signature: (new VC that specifically describes the C2PA asset)
```

Framework allows for experimentation and evolution

Credential types currently supported in draft

X.509 Certificate

```
sig_type: "cawg.x509.cose"
signature: (COSE signature over signer_payload)
```

W3C Verifiable Credential (or VP)

```
sig_type: "cawg.w3c.vc"
signature: (new VC that specifically describes the C2PA asset)
```

Framework allows for experimentation and evolution

W3C VC walkthrough

- Actor holds VC
- Actor is asked for VP that references assertions
- `signature` is the VP

NOPE!

W3C VC walkthrough

- Acto[illegible]s V[illegible]
- Actor is asked to is[illegible]a new[illegible]C that references assertions

- `signature` is the new VC

ALSO NO!

W3C VC walkthrough (version 3 – current)

`signature` is a new VC

that describes the **asset**

and its creator

W3C VC walkthrough (version 3 – current)

- Actor holds a VC with assertion_method
- Actor issues new VC that is asset-specific

- Actor responds to a presentation request from HW/SW
- HW/SW uses VP as part of a new VC including asset-specific description

- HW/SW somehow knows about actor
- HW/SW uses that info to generate a new VC that describes actor and content

`signature` is the new VC

W3C VC walkthrough (version 3)

```
{
  "@context": [
    "https://www.w3.org/ns/credentials/v2",
    "https://creator-assertions.github.io/tbd/tbd"
  ],
  "type": [
    "VerifiableCredential",
    "CreatorIdentityAssertion"
  ],
  …
}
```

W3C VC walkthrough (version 3)

```
{
  …,
  "issuer": {
    id: "did:example:2g55q912ec3476eba2l9812ecbfe",
    name: "Adobe Photoshop 2024"
      // could also be the person or organization creating the content
  },
  …
}
```

W3C VC walkthrough (version 3)

```
{
  …,
  "credentialSubject": {
    "id": "did:example:ebfeb1f712ebc6f1c276e12ec21",
    "name": "Jane Doe",
    "c2pa_asset": {
      "referenced_assertions": [
        {
          "url": "…/c2pa.hash.data",
          "hash": "U9Gyz05t…"
        },
        { "url": "…", "hash": "…" },
        { "url": "…", "hash": "…" }
      ],
      // TO DO: Add other members of `signer_payload` structure as they are defined.
    }
  },
  "proof": …,
  …
}
```

The interesting challenges ...

Broadcast application

Relying party is unknown, which is … interesting …

Also, time of verification is unknown

Identity threat model

Posit: VCs (and any digital credential, really) are *themselves* subject to misinformation and disinformation.

So ...

Identity verification

Who attests to the identity?

What is the threat model for credential issuance?

What is the trust model for credential in a broadcast environment?

Does this lead to *recentralized identity*?

Interoperability

How to navigate the rather enormous DID method space?

Given that we don’t know a priori, who is playing role of subject, issuer, and relying party, how can we ensure that credentials will be *understood* when it counts?

Duplicate identity

How do I (as a relying party) differentiate John Smith from another person also named John Smith?

Bulk signing

Creating one asset is fine, but what about 1000 at a time?

What needs to be presented to credential holder when requesting consent for signature?

What about social media?

How to document control over / affiliation with various social media accounts?

Help us build the identity assertion!

- **https://creator-assertions.github.io**
- Weekly meetings:
  - Typically on Mondays at 0830 Pacific / 1130 Eastern / 1530 UTC
  - Contact me for invitation