Content
Authenticity
Initiative

Binding Identity to Publicly-Visible Content

21 September 2023 · working draft

Eric Scouten
Senior Engineering Manager

Agenda

- The grand plan ...
- Introduction to CAI and C2PA
- Existing data model
- Proposal: Extending data model to support SSI and strongly-vetted identity
- Discuss!

How do you combat misinformation?

- education
- detection
- **attribution**

Content Authenticity Initiative (CAI)

A **community** of 1000+ media and tech companies, device manufacturers, NGOs, academics, and others working to promote adoption of an open industry standard for content authenticity and provenance.

contentauthenticity.org

Content Authenticity Initiative (CAI)

Also the team at Adobe that provides:

- open-source implementations
- product integrations (Photoshop, Firefly, etc.)
- hosted services for CAI at Adobe

Coalition for Content Provenance and Authenticity (C2PA)

C2PA addresses the prevalence of misleading information online through the development of **technical standards** for certifying the source and history (or provenance) of media content.

c2pa.org

What is Content Authenticity?

CAI allows **content creators** to make tamper-evident, digitally-signed *assertions* about what they've created.

CAI allows **content consumers** to evaluate those statements and use them to make trust decisions.

What is Content Authenticity?

Content Authenticity is **not:**

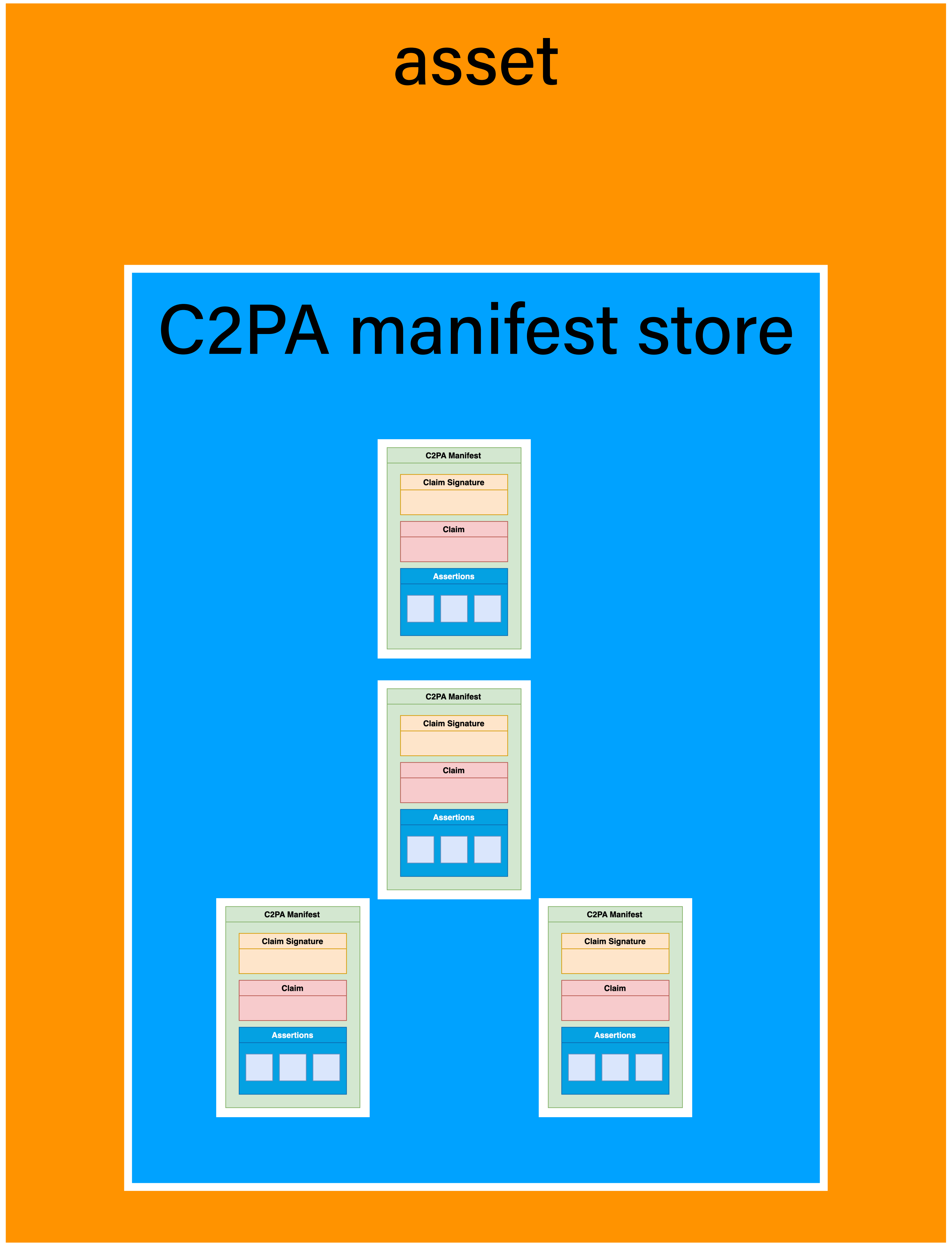
- fact-checking
- fake image detection
- politically opinionated

What is Content Authenticity?

CAI/C2PA metadata is *similar to* Exif and XMP metadata, **but** comes with tamper-evident binding to the content it describes.

So ... how does it work?

Very quick summary of data model



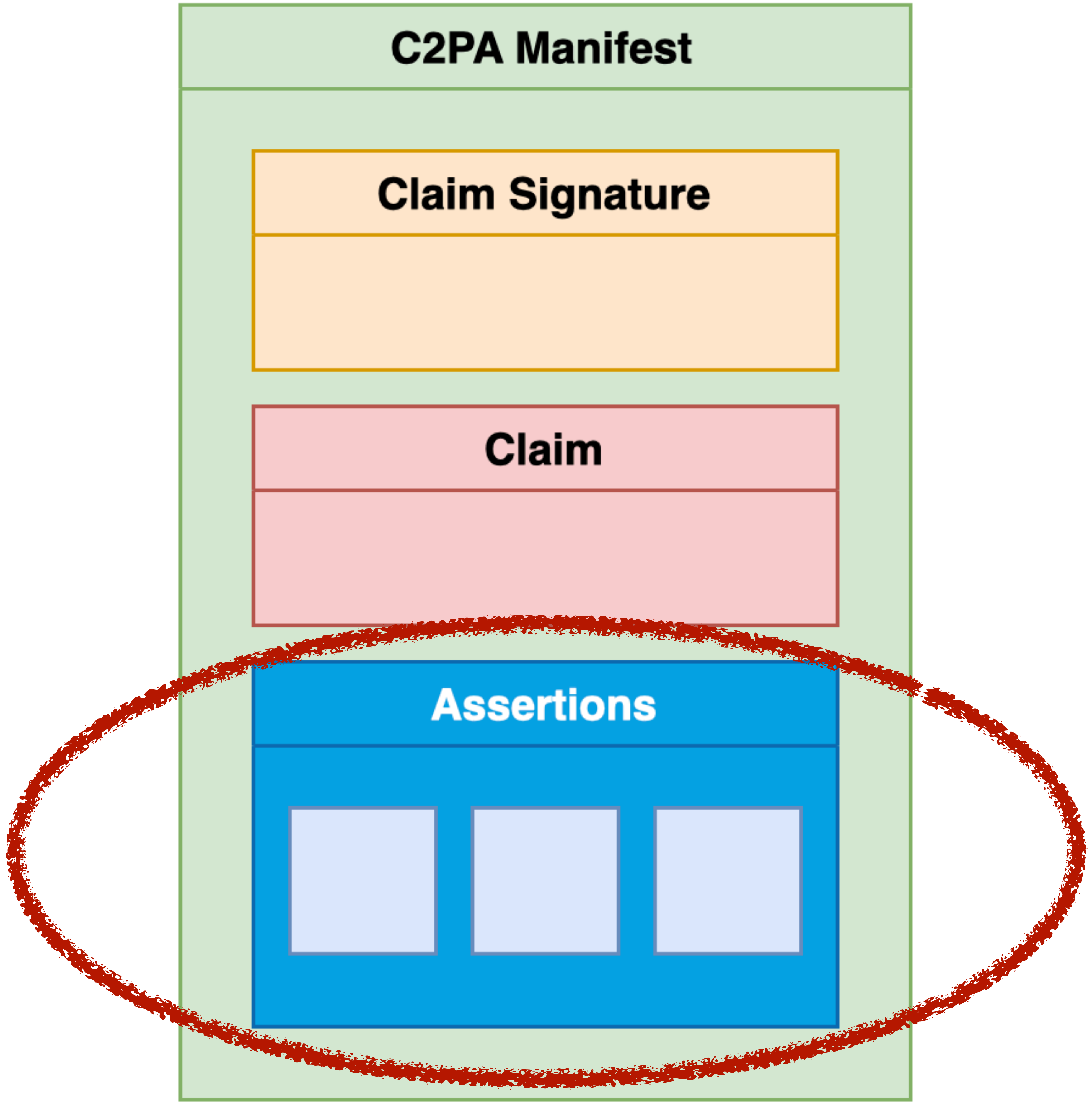
An **asset** is any piece of media (photo, video, audio, PDF, etc.).

It is described by a **manifest** which describes the most recent act of creation. That manifest may refer to *other* manifests when earlier content is incorporated.

The collection of manifests is referred to as a **manifest store.**

What's in a manifest, anyway?

Very quick summary of data model

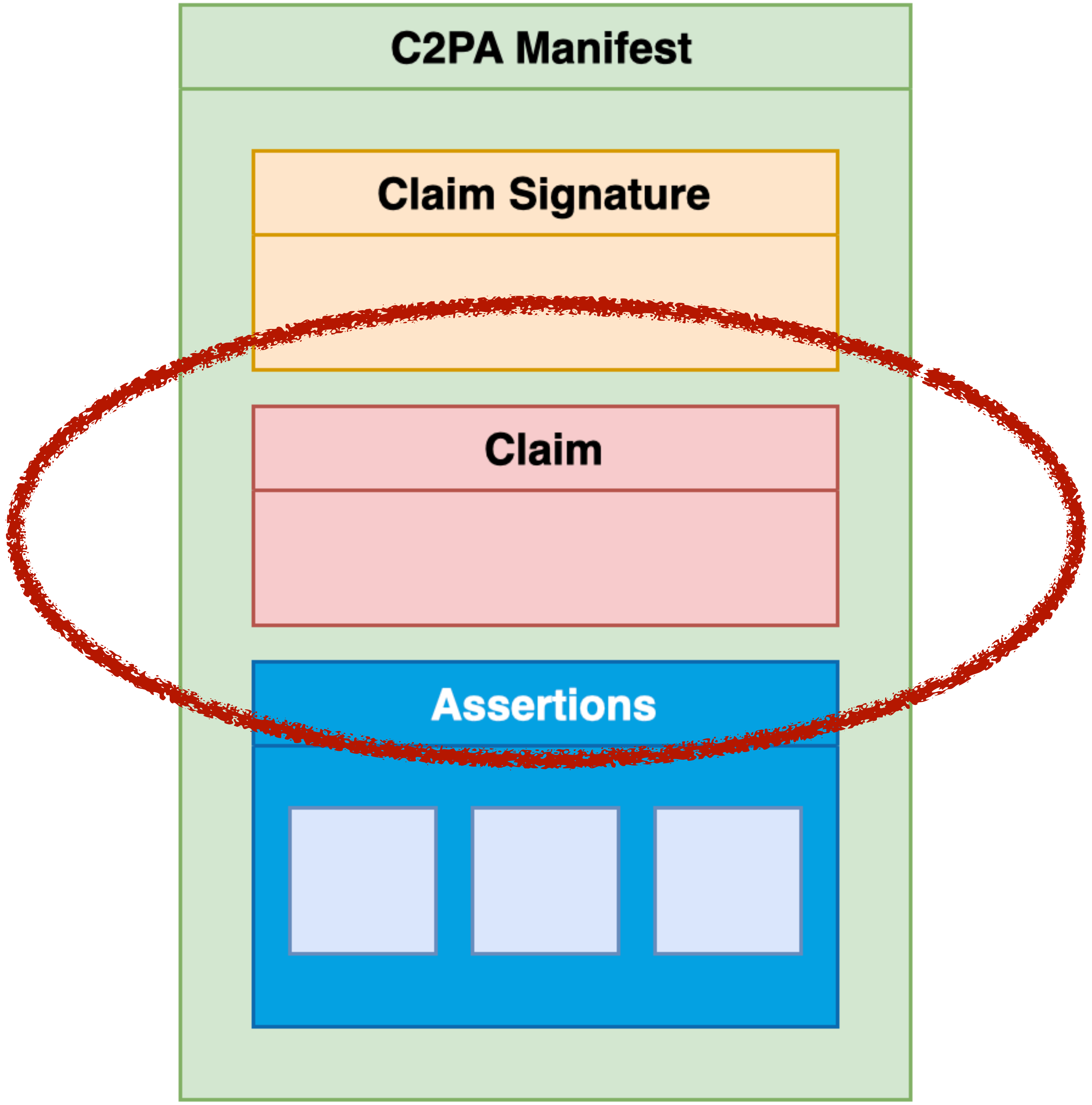


Assertions are *opt-in* statements that cover areas such as:

- capture device details
- identity of the content creator ⛳
- edit actions
- binding hash over content (req'd)
- thumbnail of the content
- **other content (ingredients)** that were incorporated into this content

Assertions can be **redacted** in later claims. ⛳

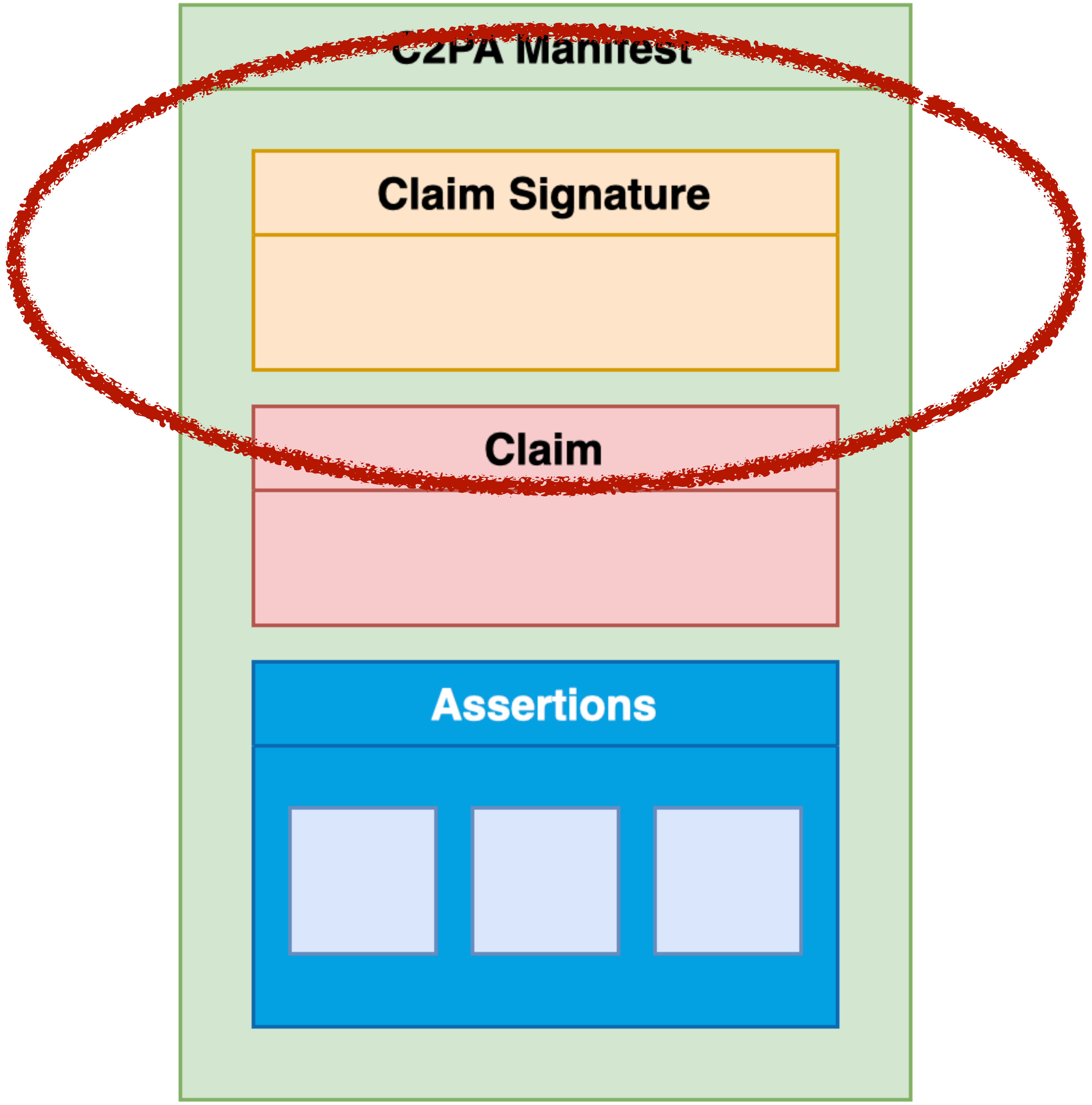
Very quick summary of data model



A **claim** contains:

- a list of its assertions (via hashed JUMBF URI) ⛳
- information about who created the claim (typically tool vendor)
- a list of assertions from *prior* claims that are being redacted ⛳

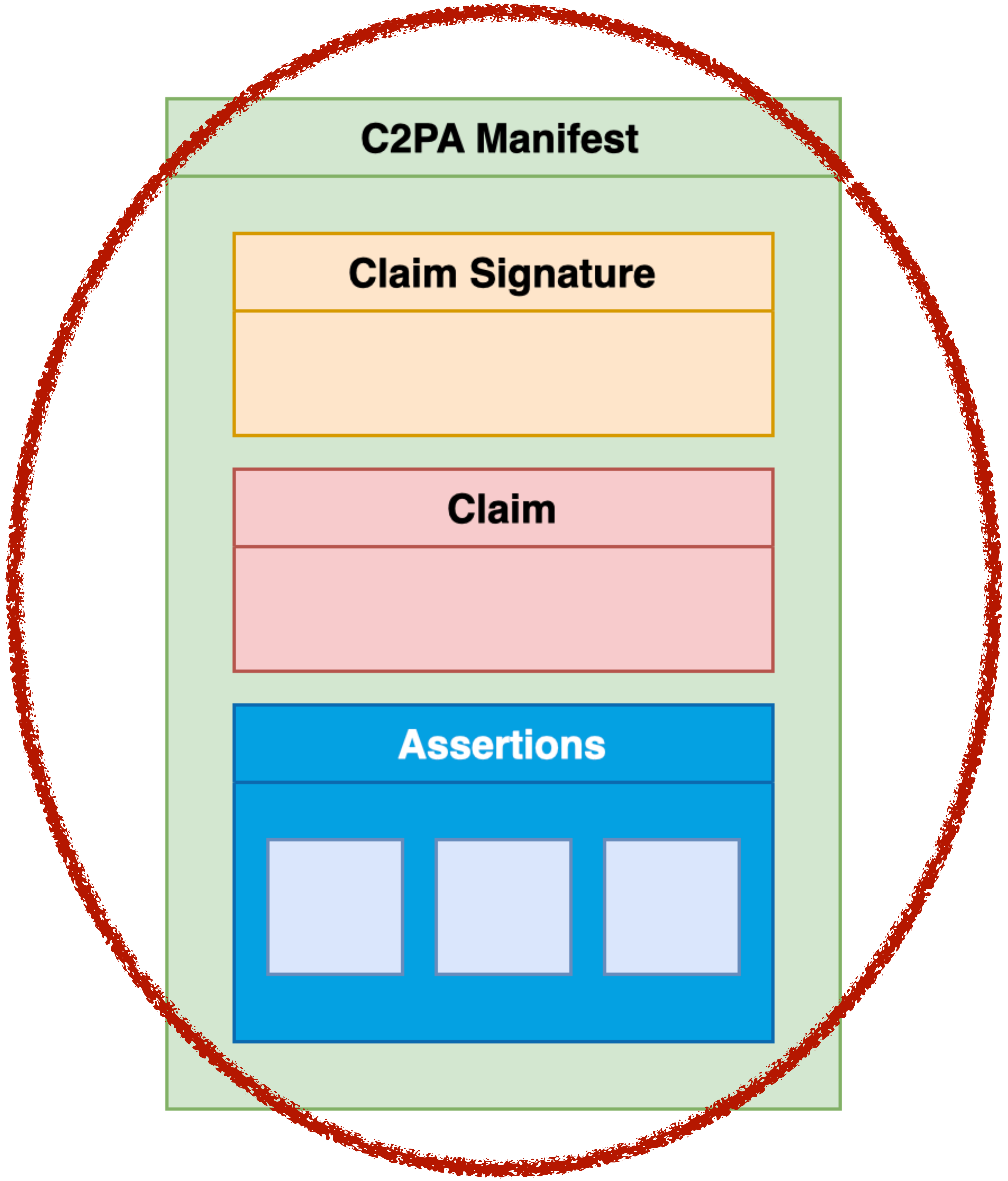
Very quick summary of data model



A **claim signature** is a COSE signature over the claim data structure.

In practice, the signature is issued by the tool vendor, though any X.509 certificate that rolls up to a known/trusted CA is accepted.

Very quick summary of data model



A **manifest** is a JUMBF data structure which contains the claim signature, claim, and assertions.

A **manifest store** (shown earlier) is a JUMBF data structure which contains one or more manifests.

A manifest store may be **embedded** in the asset it describes, **externally referenced** (via HTTPS hashlink), or both.

Questions so far?

About C2PA’s use of X.509 for signing claims

- X.509 trust model well understood
- Current practice: X.509 cert held by tool vendor (Adobe, device manufacturers, etc.), not by content creators ⛳
- X.509 is feasible for larger businesses,
  less so for individuals and smaller businesses
- Baked into standard that is in production;
  not likely to change
- But …

New questions/concerns about identity

- How should we define strongly-vetted identity within the C2PA ecosystem?

- How can subjects of such an identity prove that they were participants in the creation of each asset? Conversely, how can a content creator *disprove* a false assertion of their participation in an asset that they did not help to create?

Remember those ⛳ flags?

Now for the fun part ...

- VCs can be embedded in a manifest and referenced through CreativeWork assertions as a representation of authorship

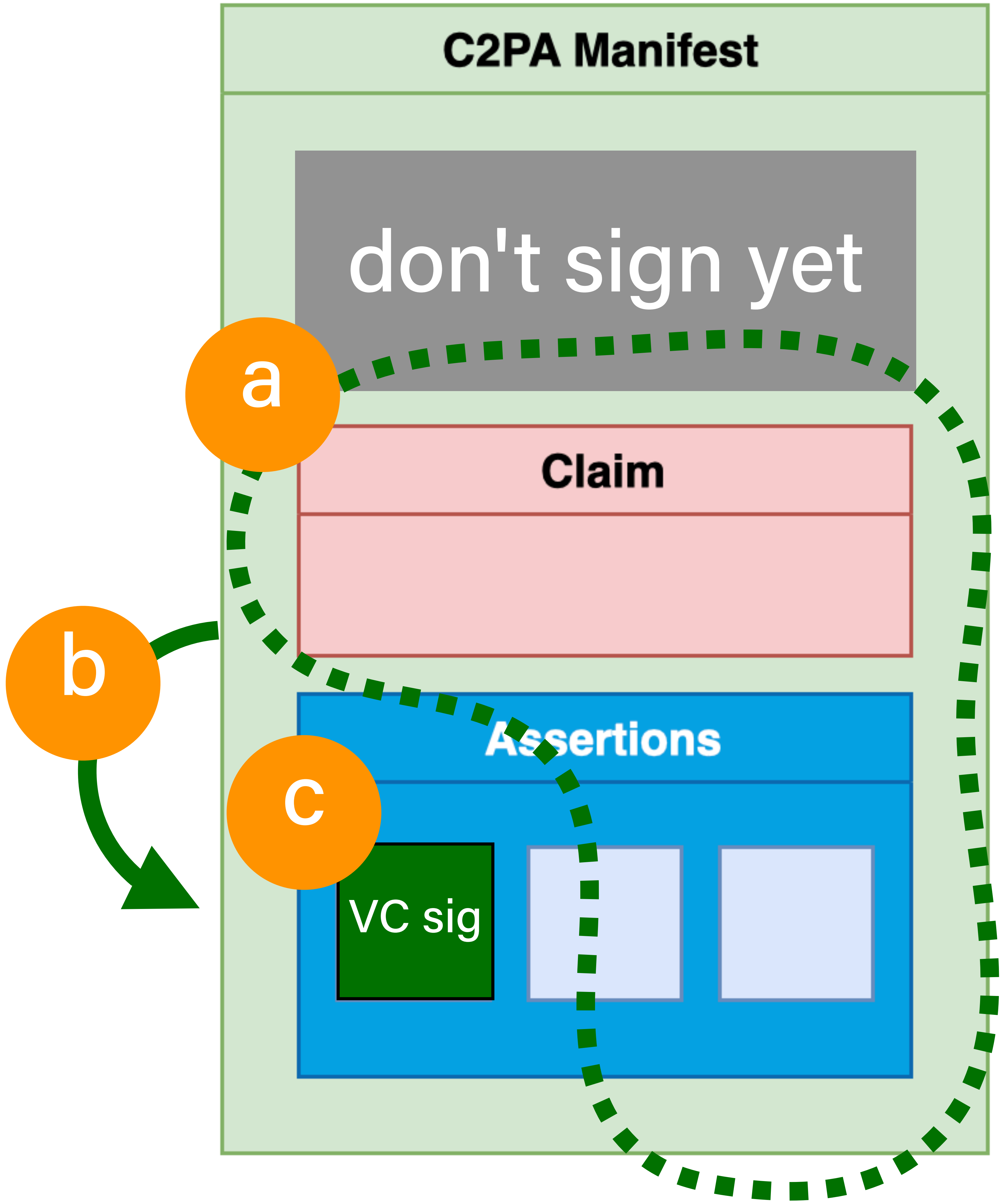
- 👍 Those assertions and VCs can be redacted if identity needs to be masked by a subsequent editor

- 👎 VCs, as currently used, are subject to replay attacks

A sketch of a proposal (1 of 4)

- Deprecate the existing mechanism of simply including VCs in CreativeWork assertion
- Add a new assertion type which incorporates a VerifiablePresentation binding the content creator (VC holder) to the content (likely via a new `did:c2pa` method)

A sketch of a proposal (2 of 4)

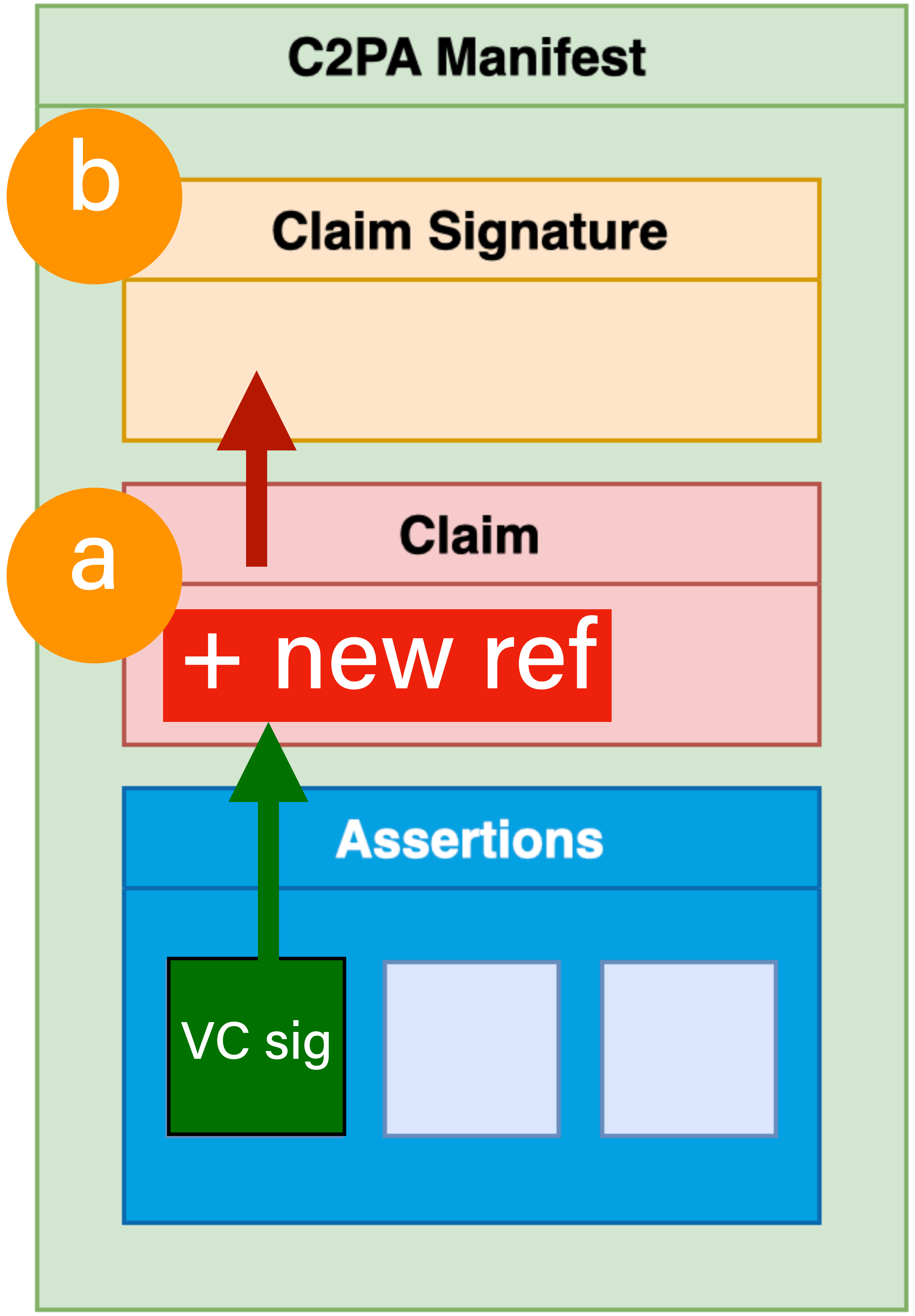


(NEW) Two stage signature process.

Signature stage 1: VC holder(s) sign a “preliminary” claim.

a. Construct assertions and claim, but don't create X509 signature.
b. VC holder (content creator) signs a Verifiable Presentation request binding VC subject to preliminary claim.
c. Create a new assertion containing that Verifiable Presentation.

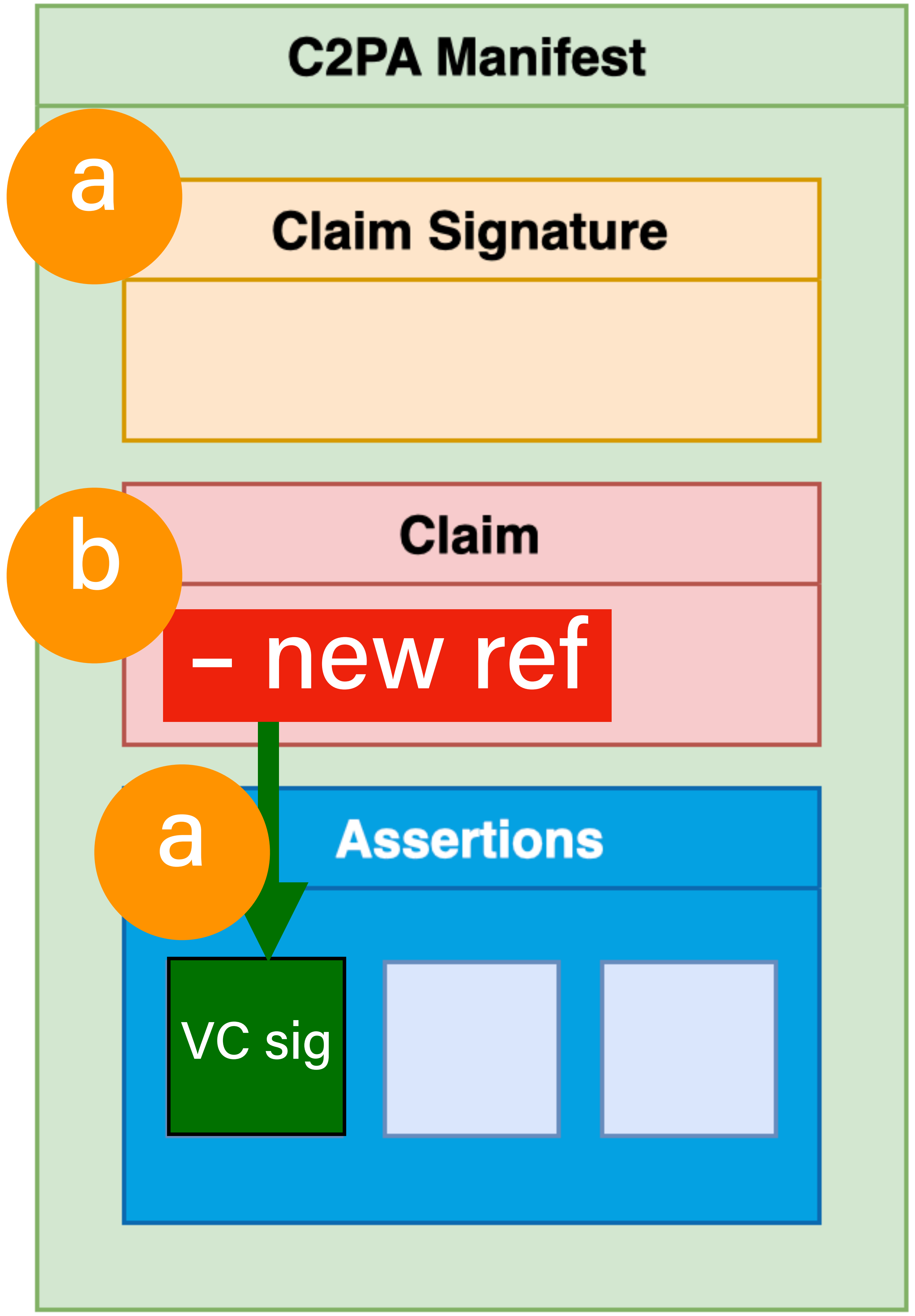
A sketch of a proposal (3 of 4)



Signature stage 2: Add new assertion; X509 holder signs full claim

a. Rebuild claim *adding* reference to VC sig assertion
b. Generate COSE signature over new claim as before

A sketch of a proposal (4 of 4)



Verifying the signature means:

a. Verify COSE signature (as before)
b. Now reverse the addition of the VC sig assertion from the claim
c. Verify the VC holder's signature against (b)

Discussion topics / contact info / links

- Suggestions for user experience?
  - Variation: What about mass-production cases?
- Is wallet adoption sufficient for this use case?
  - Will it be in __ years?
- What DID methods to support?

Eric Scouten (scouten@adobe.com, LinkedIn, IIW 37)

contentauthenticity.org · c2pa.org