Coalition for Content Provenance and Authenticity

Leonard Rosenthol, Adobe
Chair of the C2PA Technical Working Group

What are we building?

C2PA

Attribution is….

Basics

- A simple structure for storing and accessing cryptographically verifiable metadata combined with both hard and soft bindings to the asset's content
- This metadata comprises statements regarding asset creation, authorship, edit actions, capture device details, software used and many other subjects. This makes up the provenance of a given asset.

(Some of our) Design Goals

- Create only the minimum required novel technology by relying on prior, battle-tested techniques.
- Do not require cloud storage but allow for it.
- Maintain an audit trail of claims across multiple tools, from asset creation through all subsequent modification and publication/distribution.
- Support all standard asset formats supported by common authoring tools, across media types such as images, videos, audio, and documents.

Some key terms

- Assertion
  - A data structure which represents a statement, asserted by the actor concerning the asset, at a specific time and for a specific reason. This data is a part of the manifest.
- Claim
  - A digitally signed and tamper-evident data structure that references a set of assertions by one or more actors, concerning an asset, and the information necessary to represent the content binding. This data is a part of the manifest.
- Claim signature
  - The digital signature on the claim using the private key of an actor. This data is a part of the manifest.
- Content binding
  - Information that uniquely associates digital content to the asset in which it is contained.
- Manifest
  - The representation of a combination of assertions, claims (and their content bindings) and claim-signatures that includes the complete set of information about the provenance of an asset.

Core Technologies

- JSON
- CBOR
- eXtensible Metadata Platform (XMP)
- Multihash
- JPEG universal metadata box format (JUMBF)
- Cryptographic Message Syntax (CMS)
- CMS Advanced Electronic Signatures (CAdES)
- CBOR Object Signing and Encryption (COSE)

Some types of Assertions

- Asset Hashes
- Identity
- Date of Claim
- Thumbnails
- Locations (Broad, Precise)
- Camera Information
- Depthmap
- Copyright
- Actions
- Cloud Data
- Ingredients
- XMP
- ClaimReview

Example Assertion data

Precise Location


```
{
    "exif:GPSVersionID": "2.2.0.0",
    "exif:GPSLatitude": "39,21.102N",
    "exif:GPSLongitude": "74,26.5737W",
    "exif:GPSAltitudeRef": "0",
    "exif:GPSAltitude": "100963/29890",
    "exif:GPSTimeStamp": "2019-09-22T18:22:57Z",
    "exif:GPSSpeedRef": "K",
    "exif:GPSSpeed": "4009/161323",
    "exif:GPSImgDirectionRef": "T",
    "exif:GPSImgDirection": "296140/911",
    "exif:GPSDestBearingRef": "T",
    "exif:GPSDestBearing": "296140/911",
    "exif:GPSHPositioningError": "13244/2207",
}
```


Camera Information


```
{
    "crs:HasCrop": false,
    "crs:WhiteBalance": "As Shot",
    "crs:IncrementalTemperature": 0,
    "crs:IncrementalTint": 0,
    "crs:Saturation": "+3",
    "crs:Sharpness": 0,
    "crs:LuminanceSmoothing": 0,
    "crs:ColorNoiseReduction": 0,
    "exif:ColorSpace": 1,
    "exif:DigitalZoomRatio": 2.0,
    "exif:LensMake": "Apple",
    "exif:Lens": "iPhone 7 Plus back dual camera 3.99mm f/1.8",
    "aux:LensInfo": "4183519/1048501 33/5 9/5 14/5",
}
```


Adobe

Claim


```
{
    "recorder" : "Photoshop",
    "parent_claim" : "self#jumbf=cai/cb.truepic_1/cai.claim?hl=6E6DD0923B57DCE",
    "signature" : "self#jumbf=cai/cb.adobe_1/cai.signature",
    "assertions" : [
        "self#jumbf=cai/cb.adobe_1/cai.assertions/cai.identity?hl=45919681DCCAF6ABAD",
        "self#jumbf=cai/cb.adobe_1/cai.assertions/cai.claim.thumbnail.jpeg?hl=76142BD62363F"
    ],
    "redacted_assertions" : [
        "self#jumbf=cai/cb.truepic_1/cai.assertions/cai.location.precise"
    ],
    "xmp_hash": "EiBSR+oTdpKu/9/2UqnD1STX6aKVNwpqRJWE4ncOlWySkwA=",
    "asset_hashes": [
        {
            "start": "0x0000000000000000",
            "length": "0x0000000000009959",
            "name": "JFIF SOI-APP0",
            "url": "",
            "value": "EiAuxjtmax46cC2N3Y9aFmBO9Jfay8LEwJWzBUtZOsUM8gA="
        }
    ]
}
```


Adobe

Content Binding (hashing)

- File Offset
  - Useful for non-box-based formats such as JPEG or PDF
  - Non-specific
- (BMFF) Box-Based
  - Enables binding to specific functionality (and ignoring others)
  - Supports situations where some boxes may change or be created post-creation
  - May need to also support partial box hashing

Hashing BMFF Examples

```
// simple example
{
  "BoxExclusions" : [
    "Item" : { "Path" : "//c2pa" },                    // Contains manifest
    "Item" : { "Path" : "//moov/pssh" }       // Path to non-root box
      ]
}

// Exclude a box based on its Guid
{
  "BoxExclusions" : [
      "Item" : { "Path" : "//moov/uuid",
      "UUID" : "0IpPGBDzSoK2yDLYq6GD0w=="     // Defines box semantics
    }
  ]
}

// Exclude a box based on arbitrary data within the box.
{
  "BoxExclusions" : [
    "Item" : { "Path" : "//moof/emsg",        // 'emsg' etc are weird
      "BoxVersion" : 1,                       // All BMFF boxes are versioned
      "BoxLength": 1234,                      // Size of box
      "Data" : [
        { "Offset" : 20           // If data at offset 20 matches the binary, don't hash box
          "Value" : "dXJuOm1wZWc6Y29udG9zbzpmb29iYXI=" },            // URN as binary
           … ]
           }
}
```

Embedding the Manifest

- Non-BMFF formats
  - Format-specific sections of the file
    - JPEG APP11
    - PDF EmbeddedFiles
- BMFF-based formats
  - 'c2pa' box vs. 'emsg'
  - Advantages of emsg
    - Existing box (at the root to avoid size updates)
    - Working Prototype! (dash.js with minimal changes, streaming)
    - HTML5 support planned – **FOR VIDEO ONLY**

How do we get browsers to expose the C2PA information for images?

Not only for BMFF-based but all formats (JPEG, PNG, etc.)

Adobe

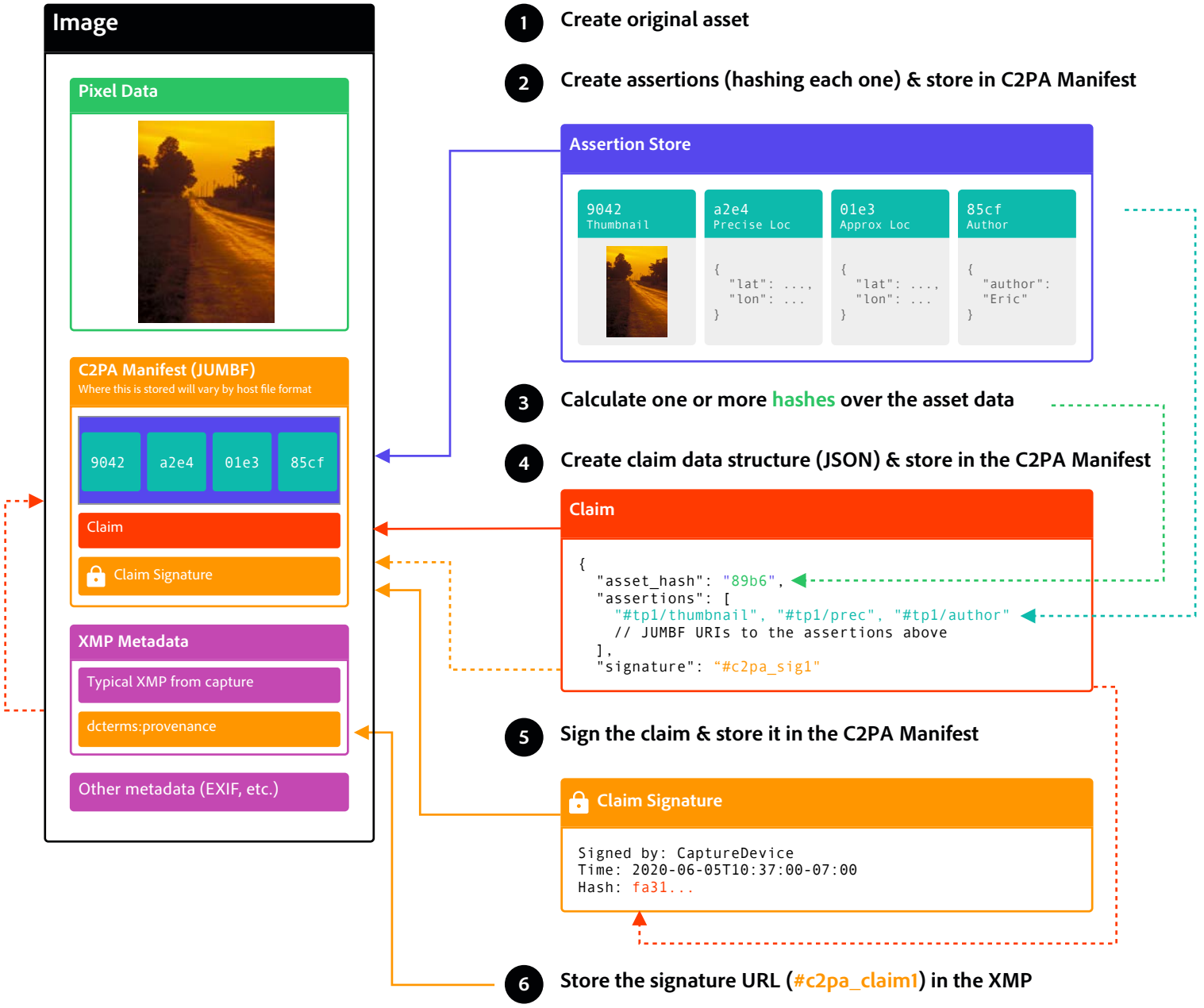
Image
Pixel Data
C2PA Manifest (JUMBF)
Where this is stored will vary by host file format
9042
a2e4
01e3
85cf
Claim
Claim Signature
XMP Metadata
Typical XMP from capture
dcterms:provenance
Other metadata (EXIF, etc.)
1
Create original asset
2
Create assertions (hashing each one) & store in C2PA Manifest
Assertion Store
9042
Thumbnail
a2e4
Precise Loc
{
"lat": ...,
"lon": ...
}
01e3
Approx Loc
{
"lat": ...,
"lon": ...
}
85cf
Author
{
"author":
"Eric"
}
3
Calculate one or more hashes over the asset data
4
Create claim data structure (JSON) & store in the C2PA Manifest
Claim
{
"asset_hash": "89b6",
"assertions": [
"#tp1/thumbnail", "#tp1/prec", "#tp1/author"
// JUMBF URIs to the assertions above
],
"signature": "#c2pa_sig1"
5
Sign the claim & store it in the C2PA Manifest
Claim Signature
Signed by: CaptureDevice
Time: 2020-06-05T10:37:00-07:00
Hash: fa31...
6
Store the signature URL (#c2pa_claim1) in the XMP

Live Content – Oy!

- One area we are putting off from our initial deliverables is support for Live Content delivery.
  - We would welcome participation and input to help in this area!

Questions

Adobe

C2PA