Protect. Renew. Empower.

C2PA, the Security Under the Hood

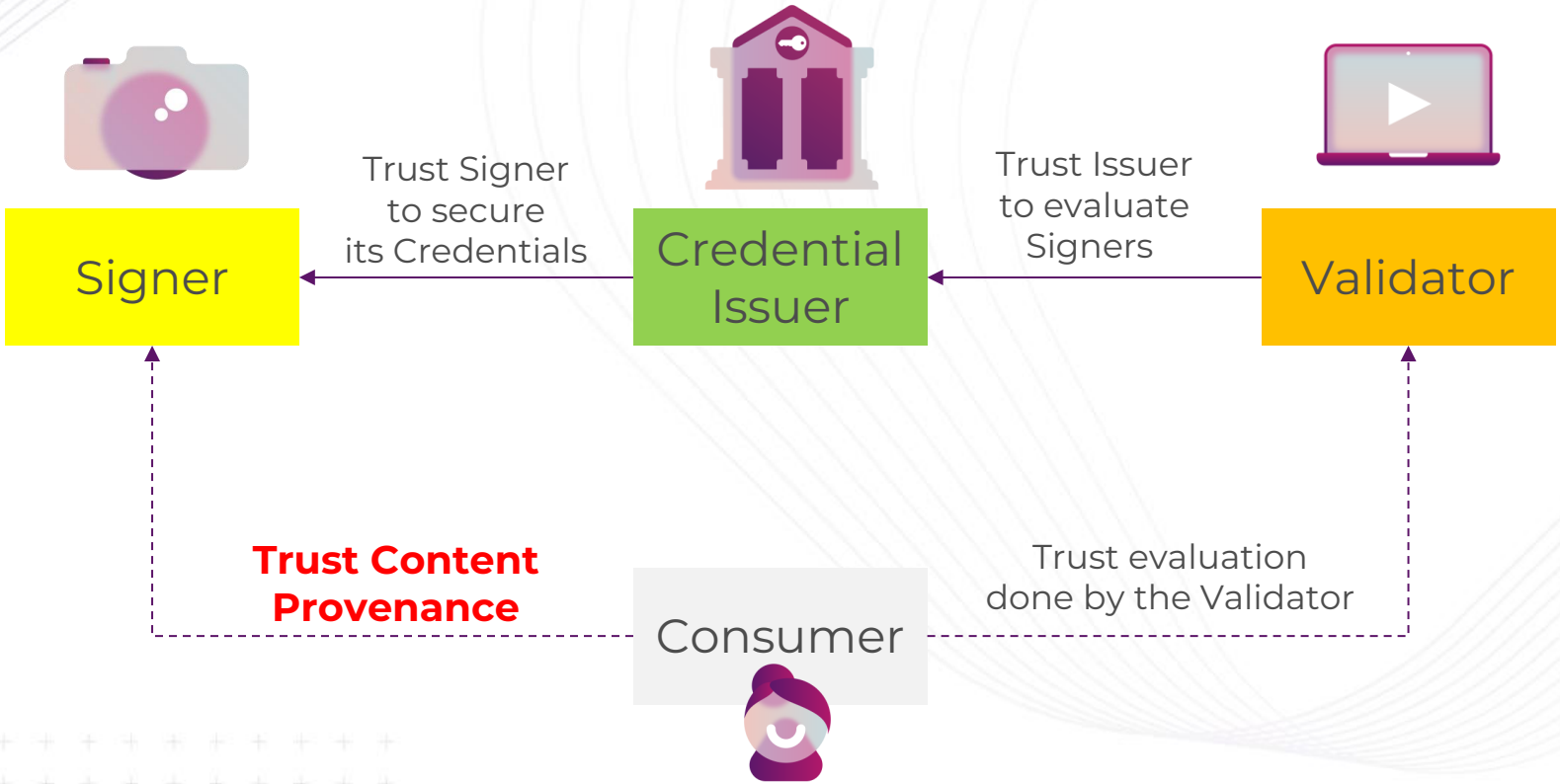
Ettore Benedetti

June 23, 2026

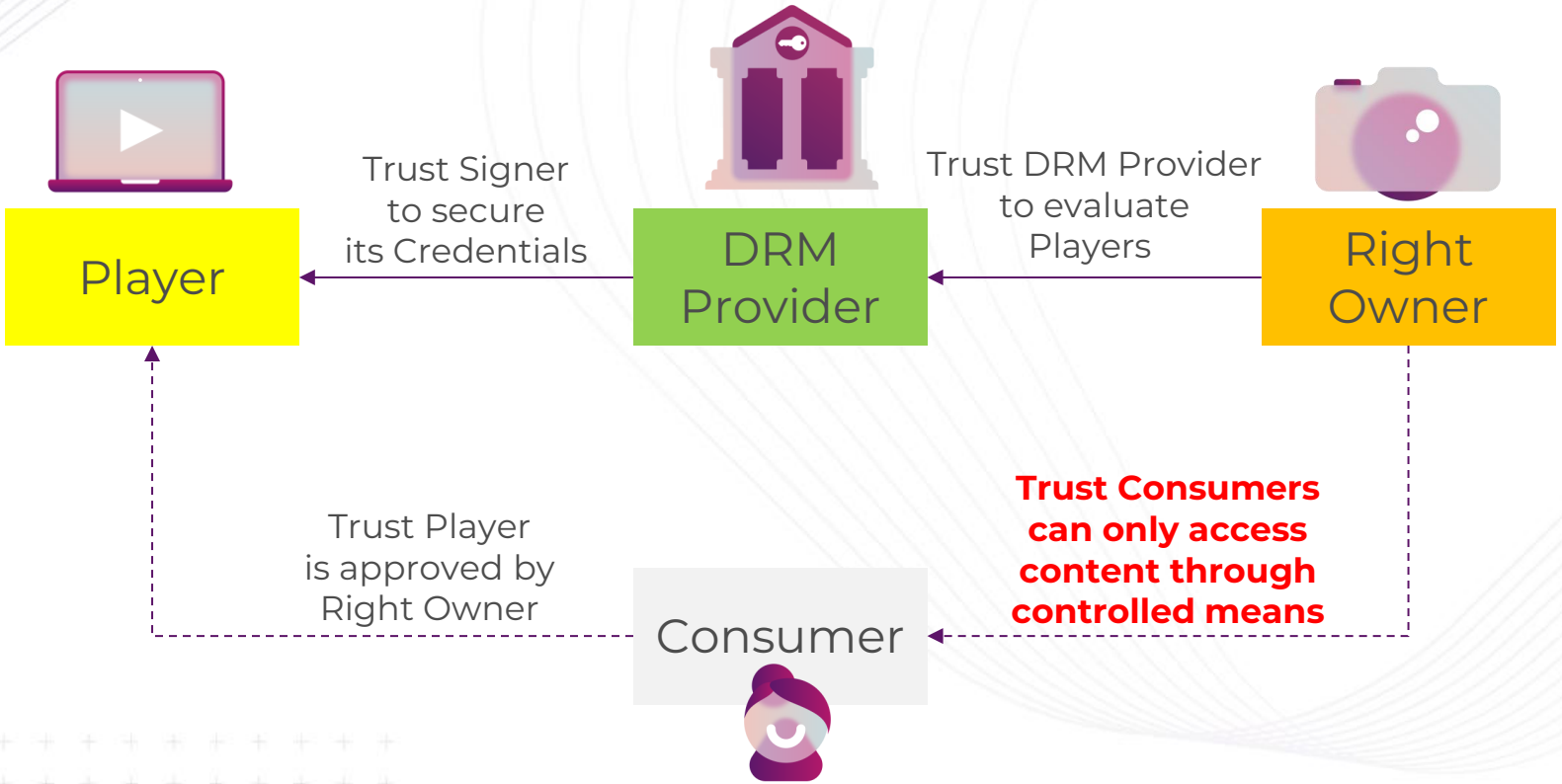
Agenda

- C2PA Trust Model vs DRM Trust Model
- Cryptographic basis of C2PA
- C2PA Conformance Program
- Revocation
- Secure Timestamping
- Conclusion

Trust Model: C2PA



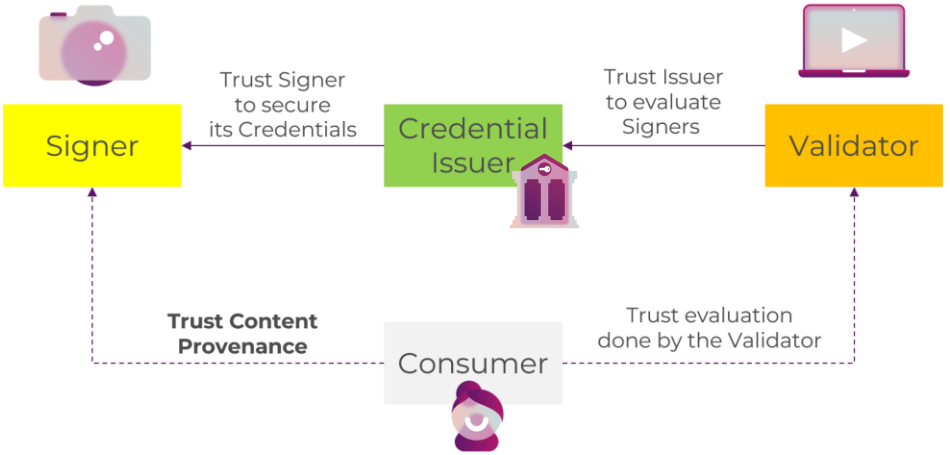
Trust Model: DRM

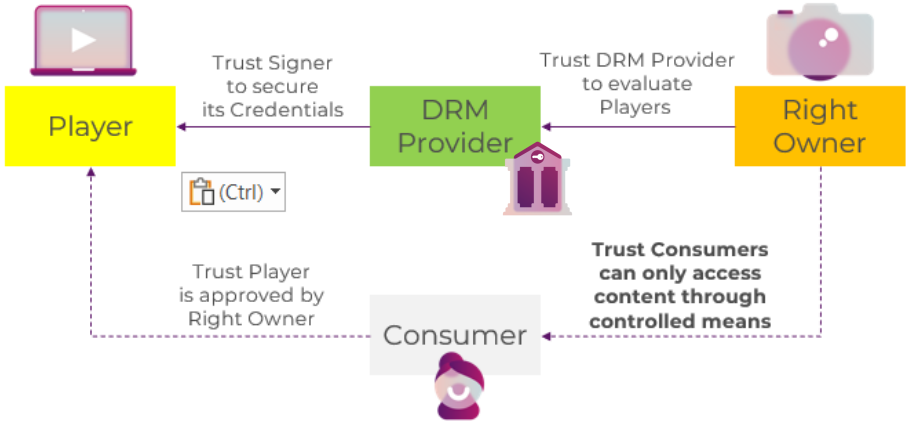


C2PA and DRM side-by-side

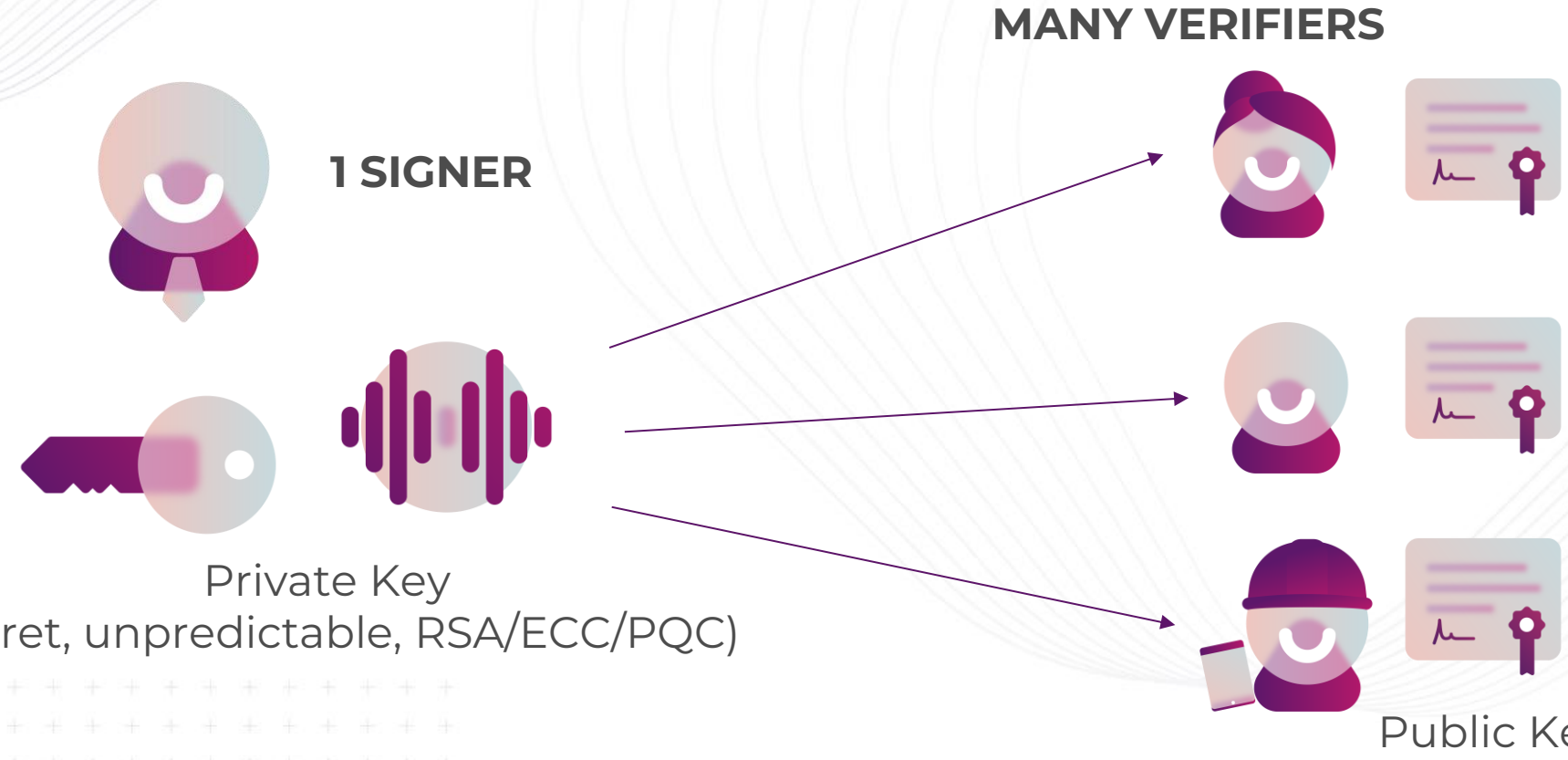
False Propaganda,
Scammers, AI slop

Content Pirates

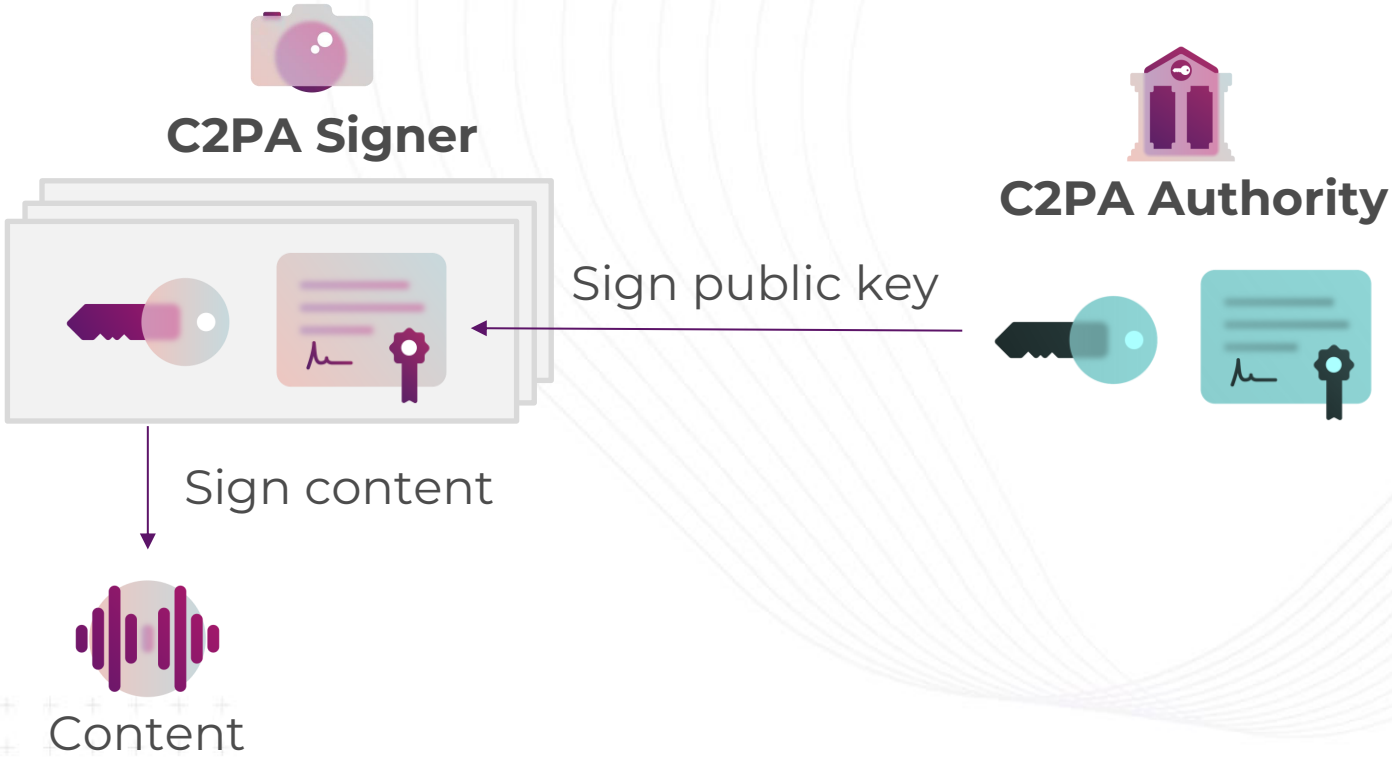




Cryptographic Keys for Signing



Scaling Cryptography (PKI)

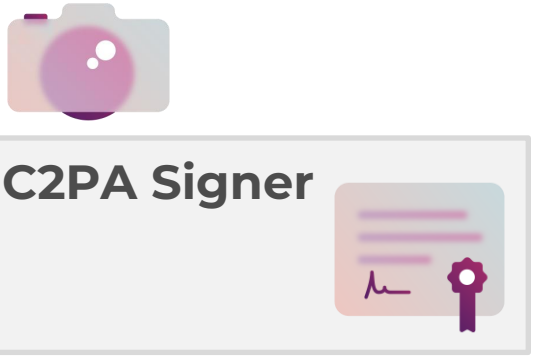


Scaling Cryptography (PKI)

STEP 2: Verify the Signer's public key with Authority's public key

C2PA Verifier

RESULT: Verifiers just need to hold 1 public key



Content

STEP 1: Verify the C2PA Manifest with the Signer's public key

C2PA Specifications are not enough

C2PA Conformance =

Interoperability

+

Establish Trust

Trust by Conformance

Security Requirements for

- **Authorities** 
- **Generator Products** 

Trust Level 1: lightweight
Example: app can use key in the clear

(~ Widevine L3 for HD content)

Trust Level 2: more secure
Example: keys must be isolated

(~ Widevine L1 for Full HD content)

Putting Everything Together (for now)

C2PA Trust List

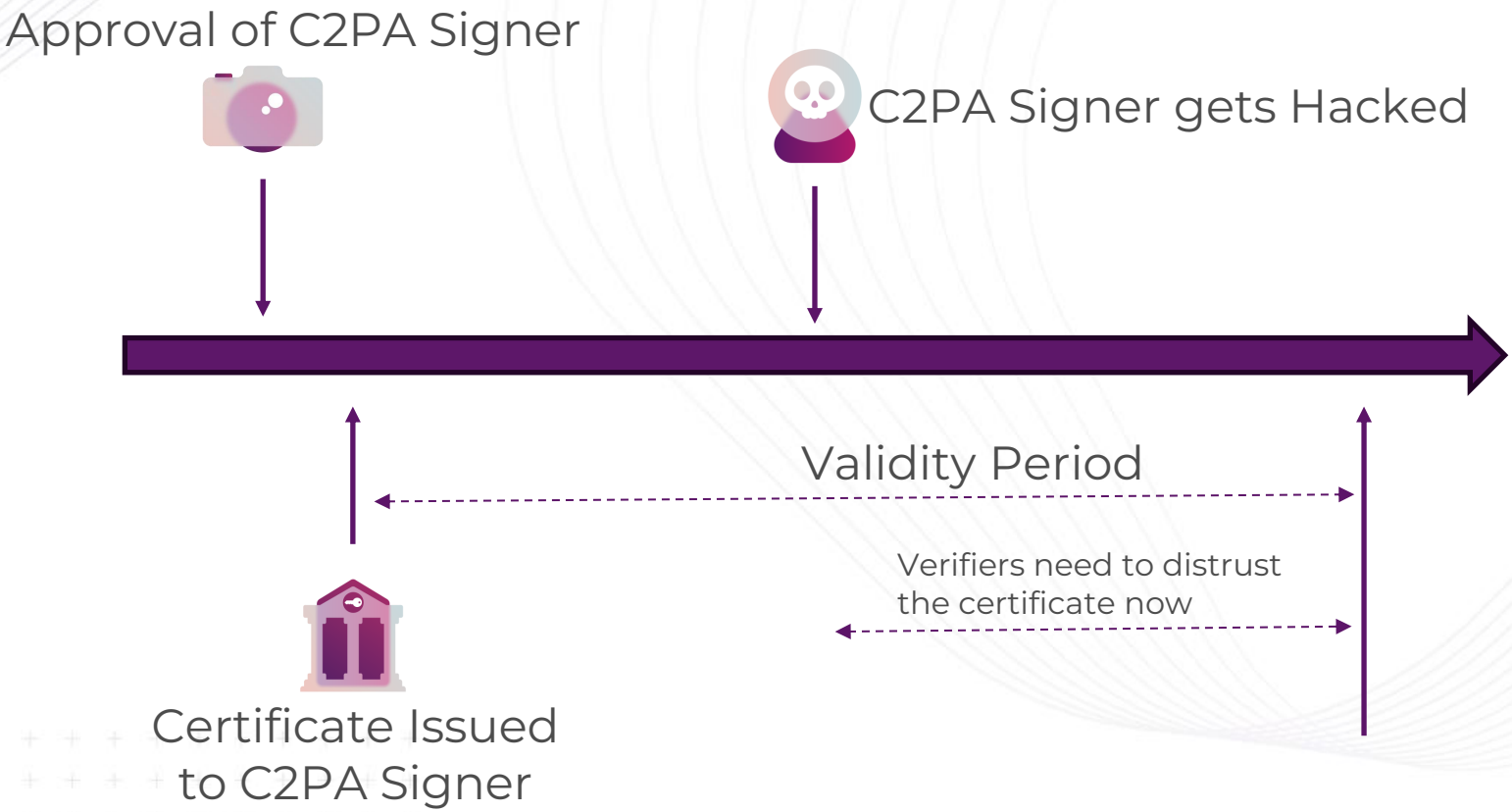
Issue certificates to

C2PA Generator On Product List

Trust Level 1
Certificate validity: <1 year

Trust Level 2
Certificate validity: <3 months

Revocation: Introduction



Trust Status (and Revocation) Checks in C2PA

WELL-FORMED → functionally OK

VALID → WELL-FORMED
+ cryptographic integrity
+ claimSignature.validated == TRUE
+ claimSignature.insideValidity == TRUE
+ **signingCredential.ocsp.revoked == FALSE**

TRUSTED → Valid + signingCredential.trusted == TRUE

Enabling Revocation

C2PA Signer

OCSP Responder

Am I Revoked?

Signed Proof that you are NOT revoked (valid for ~24 hours)

Embed Proof in C2PA Manifest

RESULT: Verifiers do not need to perform any online check

Expired Certificates Invalidate C2PA Manifests

WELL-FORMED → functionally OK

VALID → WELL-FORMED
+ cryptographic integrity
+ claimSignature.validated == TRUE
+ claimSignature.insideValidity == TRUE
+ signingCredential.ocsp.revoked == FALSE

TRUSTED → Valid + signingCredential.trusted == TRUE

Secure Timestamps as a Time Machine

C2PA Signer

TimeStamp Authority

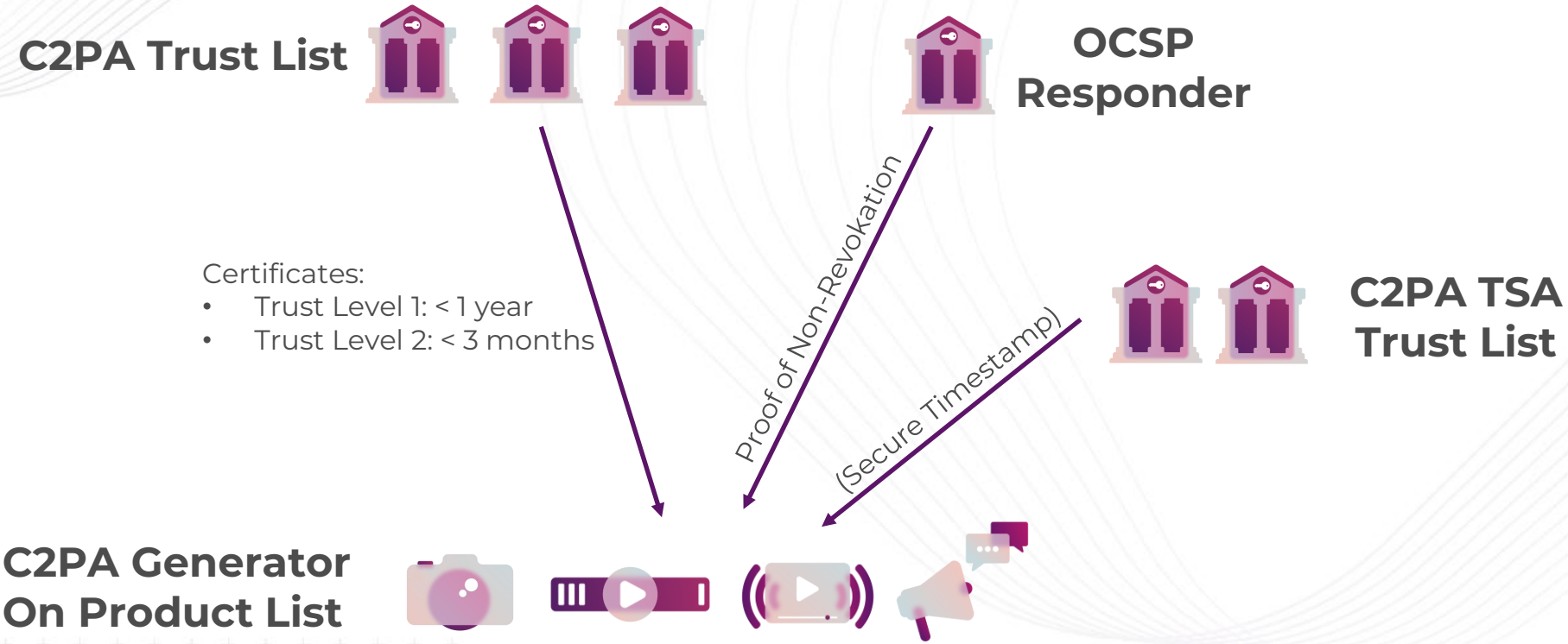
Notarize this C2PA Manifest (hash)

Signed Proof that C2PA Manifest existed at this time

Embed Timestamp in C2PA Manifest

RESULT: C2PA Manifest Remains Verifiable in the Future

Putting Everything Together



Summary

- The Security of C2PA is remarkably similar to the Security for DRM
- C2PA Manifests are signed with standard tools (PKI)
- Interoperability and Trust require programs beyond C2PA alone
- C2PA Signers must embed Revocation Status information in the Manifests
- Secure Timestamps: to ensure the C2PA Manifest remains verifiable

Protect. Renew. Empower.